UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA

Aspectos de Relatividade Numérica

Campos escalares e estrelas de nêutrons

Leonardo Rosa Werneck

Orientador: Prof. Dr. Elcio Abdalla

Uma tese apresentada ao Instituto de Física da Universidade de São Paulo como parte dos requisitos necessários para obter o título de doutor em Física.

Banca examinadora:

Prof. Dr. Elcio Abdalla (IF-USP) – Presidente da banca
Prof. Dr. Arnaldo Gammal (IF-USP)
Prof. Dr. Daniel A. Turolla Vanzela (IFSC-USP)
Prof. Dr. Alberto V. Saa (IFGW-UNICAMP)
Profª. Drª. Cecilia Bertoni Chirenti (UFABC/UMD/NASA)

São Paulo
2020

FICHA CATALOGRÁFICA
Preparada pelo Serviço de Biblioteca e Informação do Instituto de Física da Universidade de São Paulo

Werneck, Leonardo Rosa

Aspectos de relatividade numérica: campos escalares e estrelas de nêutrons / Aspects of Numerical Relativity: scalar fields and neutron stars. São Paulo, 2020.

Tese (Doutorado) – Universidade de São Paulo. Instituto de Física. Depto. Física Geral.

Orientador: Prof. Dr. Elcio Abdalla

Área de Concentração: Relatividade e Gravitação

Unitermos: 1. Relatividade numérica; 2. Campo escalar; 3. Fenômeno crítico; 4. Estrelas de nêutrons; 5. Equações diferenciais parciais.

USP/IF/SBI-057/2020

UNIVERSITY OF SÃO PAULO
INSTITUTE OF PHYSICS

Aspects of Numerical Relativity

Scalar fields and neutron stars

Leonardo Rosa Werneck

Advisor: Prof. Dr. Elcio Abdalla

A thesis submitted to the Institute of Physics of the University of São Paulo in partial fulfillment of the requirements for the title of Doctor of Philosophy in Physics.

Examination committee:

Prof. Dr. Elcio Abdalla (IF-USP) – Committee president
Prof. Dr. Arnaldo Gammal (IF-USP)
Prof. Dr. Daniel A. Turolla Vanzela (IFSC-USP)
Prof. Dr. Alberto V. Saa (IFGW-UNICAMP)
Prof. Dr. Cecilia Bertoni Chirenti (UFABC/UMD/NASA)

São Paulo
2020

This page intentionally left blank.

Unthinking respect for authority is the greatest enemy of truth.

Albert Einstein

This page intentionally left blank.

Contents

List of Figures

List of Tables

This page intentionally left blank.

Resumo

Nesta tese apresentamos os resultados de nossa pesquisa no campo de relatividade numérica. Inicialmente, focaremos no estudo de fenômenos críticos no colapso gravitacional de campos escalares não massivos, mostrando que conseguimos obter a solução crítica correta utilizando tanto o formalismo ADM quanto o BSSN. Nosso estudo se concentra no caso esfericamente simétrico, onde encontramos valores para o expoente crítico e período de eco em excelente concordância com a literatura. A universalidade da solução crítica também é verificada. Posteriormente, apresentamos os resultados de nossas implementações de equações de estado avançadas que melhor descrevem estrelas de nêutrons. Validamos nossos códigos realizando o teste de "hidrodinâmica sem hidrodinâmica" e calculando as relações "massa vs raio" para as estrelas de nêutrons. Finalmente, aplicamos a nova implementação para o estudo de uma simulação de fusão de um binário de estrêlas de nêutrons.

Palavras-chave: relatividade numérica, campo escalar, fenômeno crítico, estrelas de nêutrons, equações diferenciais parciais

This page intentionally left blank.

Abstract

In this thesis we present the results from our research in the field of numerical relativity. We initially focus on critical phenomena in the gravitational collapse of massless scalar fields, showing that we are able to obtain the correct critical solution using both the ADM and BSSN formalisms. We study the spherically symmetric case to great extent, finding critical exponents and echoing periods in excellent agreement with the literature. The universality of the critical solution has also been explored and verified. Then, we present results from our implementations of advanced equations of state that more accurately describe neutron stars. We validate our codes by performing the "hydro without hydro" test and computing "mass vs radius" relations for the neutron stars. Finally, we apply the new implementation to the study of a binary neutron star merger simulation.

Keywords: numerical relativity, scalar field, critical phenomena, neutron stars, partial differential equations

This page intentionally left blank.

Acknowledgements

I thank my wife, Natali, for being by my side during all my years as a graduate student. She motivated and inspired me to keep pushing, no matter how difficult things got, and I will be forever grateful for her companionship and her patience.

I thank my parents, Marco and Fatima, for always valuing my education and giving me the means to be able to pursue an academic career.

I would like to thank Elcio for all his help and his friendship. Numerical relativity has become my passion and I have him to thank for it, since I doubt I would've joined the field if it wasn't for him.

I thank Zach for his friendship, patience, and support over the past year. His kindness and hospitality are refreshing, which coupled with his natural ability to produce beautiful, efficient, and well documented codes, are continuous sources of inspiration to me.

I would also like to thank Carlos Eduardo for the time he dedicated to help me with the implementation of the numerical algorithm that started this study. I also thank him for kindly receiving me in Natal and for all the long talks we had trying to make sense of what we were doing.

I thank Bertha for the weekly meetings where I could share my successes and failures. This research project would not have started without the initial proposal by prof. Eleftherios Papantonopoulos, so I offer him my gratitude as well.

I would also like to thank Thomas Baumgarte and Matt Choptuik for useful inputs in my studies of critical phenomena.

My gratitude goes to the members of the CPG/IFUSP that helped me with the numerous requests for documents and forms throughout this doctorate. I also thank the secretaries from the Department of Mathematical Physics and the Department of General Physics at the IFUSP for dealing with departamental affairs. I also thank the secretaries from the Physics and Astronomy at WVU, who have helped making the time I spent there very pleasant.

This study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil* (CAPES) - Finance Code 001, and the National Aeronautics and Space Administration (NASA), grant number NASA-TCAN-80NSSC18K1488.

This page intentionally left blank.

Aspects of Numerical Relativity

Scalar fields and neutron stars

This page intentionally left blank.

Chapter 1

Introduction

As of early 2019, the Laser Interferometer Gravitational-Wave Observatory (LIGO) and Virgo Scientific Collaboration have detected gravitational waves which have been emitted from different black hole mergers [1–7] and neutron star binary inspirals [8–10]. More recently still, an event has been detected that involves a 23 solar mass black hole and a 2.6 solar mass compact object [11], and our theoretical understanding of this event does not tell us conclusively whether the compact object is the least massive black hole or the most massive neutron star ever detected.

The newest updates to the detectors increase their signal-to-noise ratio, allowing us to detect more and more gravitational waves. Some of these signals, however, will be weaker since they will come from sources which are further away. This generates an increasing demand for theoretical models capable of accurately describing these phenomena, since this description is of paramount importance if we are to understand and predict what was the physical system that originated a detected gravitational wave, and may result in the discovery of new physics.

Gravitational waves are emitted when gravity is extremely strong and the equations of general relativity become highly nonlinear. Numerical relativity consists of numerical techniques used to solve Einstein's field equations in this extreme regime, though this is far from a trivial task.

Over the last few decades a few different techniques have been used in numerical relativity codes. The Arnowitt-Deser-Misner (ADM) formalism [12], a popular approach, decomposes spacetime into space plus time, the so called *3+1 ADM decomposition*. The formalism turns the problem of solving Einstein's field equations into an initial value

problem, so that once an initial hypersurface satisfying Einstein's equations at an instant in time is specified, the formalism constructs the spacetime at later times one spatial hypersurface at a time.

The 3+1 ADM decomposition is, theoretically, suitable for evolving Einstein's equations on the computer, but the resulting set of partial differential equations (PDEs) is only *weakly hyperbolic* [13]. As a result, simulations adopting this formalism are largely unstable due to its propensity to allow high-frequency modes generated from numerical noise to grow without bounds.

Given these known issues of the ADM formalism, the Baumgarte-Shapiro-Shibata-Nakamura (BSSN) formalism [14–16] has superseded it as the go-to approach to numerical relativity. This formalism poses the Einstein's field equations in a *strongly hyperbolic* [13] form and has proven to yield robust and stable simulations of dynamical spacetimes. The BSSN formalism has been used to produce gravitational wave profiles for black hole binary mergers (see, e.g., [17]) and neutron star binary mergers (e.g., [18]).

The ever increasing demand for accurate predictions of gravitational waves emitted from mergers of compact objects binaries require the development of robust codes that are capable of solving Einstein's field equations both stably and reliably. In this thesis we present results obtained from, and contributions made to, three different such codes.

The first of them is a code developed during this research project which is intended to solve Einstein's field equations using the ADM formalism and study the gravitational collapse of massless scalar fields: the `SFcollapse1D` code [19]. This code is faithfully based on Matt Choptuik's original studies on the subject [20–22], but makes use of a simpler infrastructure that allows it to run efficiently even on laptop computers, replacing the original adaptive mesh refinement (AMR) algorithm [23] with a clever coordinate transformation that results in the same resolution increase. This code has been open-sourced [24] recently as well.

The second one is the `NRPy+` [25, 26], an open-source [27] infrastructure which allows us to generate highly optimized and memory efficient `C` code to evolve the BSSN equations. We have been in active collaboration with the authors of `NRPy+` and have made contributions to it ourselves. The fruitful results of this collaboration are shown in this thesis, where we use our codes to study both critical phenomena and neutron stars.

The third one is `IllinoisGRMHD` [28–30], a code specialized in solving the general

relativistic magnetohydrodynamics equations to study binary neutron stars. We have made significant contributions to updating and documenting `IllinoisGRMHD` over the past few months. Our latest contributions have been using `NRPy+` to generate `IllinoisGRMHD`, while also optimizing some of its routines, and adding support for piecewise polytropic (see [31] for the ones we are interested in) and tabulated equations of state [32]. We note that out of all of these modifications, only the tabulated equations of state support has not been completely validated and debugged, hence this version of the code has not yet been open-sourced.

We emphasize that the contents of this thesis, as well as the research projects that lead to its writing, have been developed with the goal of creating a new research subgroup inside Prof. Abdalla's research group, with the specific intent of studying Numerical Relativity. To this end, most of the work we have completed is related to code development, since we started completely from scratch. We are glad to report that we have succeeded in this enterprise, while also establishing active collaborations with well stablished Numerical Relativity research groups, including the groups which compose the Theoretical and Computational Astrophysics Network on Binary Neutron Stars (TCAN) [33]. Our most direct and active collaboration is with the research group of Prof. Dr. Zach B. Etienne, from the West Virginia University.

This thesis is divided into two parts: I) theoretical background and II) research work. Although we will not label the appendices as a "part", we think they contain valuable information for those who are encountering the topics discussed in this thesis for the first time, and should viewed as more technical extensions of some of the chapters. Below we give suggestions regarding when each appendix should be read.

Chapter 2 provides a theoretical introduction to the equations commonly used in numerical relativity. Section 2.1 sets the notation used throughout the thesis. In section 2.2 we briefly review Einstein's field equations. In section 2.3 we decompose Einstein's equations into their (3+1) form using the so-called ADM formalism. Then, in section 2.4 we provide an introduction to the more robust BSSN formalism, studying its covariant form in section 2.5. Finally, in section 2.6 we briefly discuss the general relativistic (ideal) magnetohydrodynamics equations. We recommend appendices A and B as additional reading material for this chapter.

In chapter 3 we present our results on the gravitational collapse of a massless scalar

field. We discuss how to solve the Einstein-Klein-Gordon equations numerically, using the ADM formalism, in section 3.1. Solving the Einstein-Klein-Gordon equations using the BSSN formalism is discussed in section 3.2. In section 3.3 we present results that indicate that a single parameter separates solutions that form black holes from those that do not. Building on these results, we discuss critical phenomena in section 3.4. Further detail on the implementations we have used to get to the results presented in this chapter can be found in appendices C, E, and F.

Neutron stars are discussed in chapter 4. We focus on how to describe a neutron star on ever increasing levels of accuracy by studying equations of state of increasing sophistication. Starting from a simple polytropic equation of state in section 4.1. Tabulated equations of state are discussed in section 4.2. The motivation and description for our study of piecewise polytropic equations of state is found in section 4.3. We provide results that validate our implementations in section 4.4. Finally, binary neutron stars are discussed in section 4.5.

Part I
Theoretical Background

This page intentionally left blank.

Chapter 2

An introduction to numerical relativity

Einstein's field equations are a set of nonlinear, second-order, partial differential equations that must be solved in order to determine the spacetime metric $g_{\mu\nu}$. Some exact solutions to these equations are known, for example the famous Schwarzschild [34], Kerr [35], Reissner–Nordström [36–39], and Friedmann–Lemaître–Robertson–Walker [40–47] solutions. It is worth noting, however, that exact solutions of Einstein's field equations are indeed quite rare, but approximations and perturbative approaches can still provide us insight on how spacetime should behave under certain conditions.

When we study the strong field regime of gravity, e.g. on the surface of neutron stars, our analytic and perturbative methods tend to break down completely. Numerical relativity is one way to overcome these limitations, where we solve Einstein's field equations, *in full*, using numerical techniques. For anyone who has worked with general relativity in the past, it is probably quite obvious that numerically solving Einstein's field equations is no trivial task either, but as the field of numerical relativity evolved, new formalisms and techniques allowed us to accurately solve Einstein's field equations and probe even the most extreme regimes of gravity. We present in this chapter two such formalisms.

The first and oldest of the two is due to Arnowitt, Deser, and Misner and is called the ADM formalism [12]. In the ADM formalism, Einstein's field equations are rewritten in order to split spacetime into "space plus time". Spacetime is folliated into hypersurfaces of constant coordinate time, turning the problem of solving Einstein's field equations into an *initial value problem*. Once an initial hypersurface satisfying Einstein's field equations

at an instant in time is specified, the formalism constructs the spacetime at later times one spatial hypersurface at a time.

Robust integration of the ADM equations can be obtained by using the so-called *implicit*, or *constrained*, evolution schemes, and we will give an in-depth example of such a scheme in chapter 3. *Explicit*, or *unconstrained*, time integration of the ADM equations, however, is *unstable*. This is due to the fact that the ADM equations are only weakly hyperbolic,[1] making the initial value problem ill-posed. As a result, simulations adopting this approach tend to allow high-frequency modes generated from numerical noise to grow without bounds.

To overcome this ill-posedness, a new formalism has been developed by Baumgarte, Shapiro, Shibata, and Nakamura, the so-called BSSN formalism [14–16]. The BSSN formalism rewrites the ADM equations into a strongly hyperbolic system of partial differential equations by introducing a new set of auxiliary variables. The hyperbolicity of the ADM and BSSN equations is studied, e.g., in [50–53]. Simulations using the BSSN equations, coupled with appropriate gauge conditions [54, 55], have been able to produce gravitational wave profiles for black hole binary mergers (e.g. [17]) and neutron stars binary mergers (e.g. [18]).

Most objects studied in numerical relativity tend to be either spherical or quasi-spherical. Despite this fact, the original BSSN formalism was developed with Cartesian coordinates in mind. It should be obvious that Cartesian coordinates are not the most efficient way of modeling spherical objects. Moreover, the BSSN equations are not generally covariant, resulting in very complicated transformations rules for certain tensorial quantities and, therefore, making it difficult to use them with a different coordinate system.

To remedy all of these difficulties, Brown has proposed a modification to the BSSN formalism that uses a reference metric approach [56], making it generally covariant. With this generalization, new strategies were developed to handle the coordinate singularities that arise when using curvilinear coordinates, such as the rescaling of tensors component by component [57–60] (see also appendix F).

In this chapter we review both the ADM and BSSN formalisms. All the equations will

[1] See e.g. [13, 48] for relatively intuitive discussions on the concepts of *hyperbolicity* and [49] for a more rigorous discussion. We also provide a brief introduction on the subject, based on the first two references mentioned here, in appendix A.

be written in full generality, without any specific matter source in mind. Section 2.1 sets the notation we will use throughout the thesis, followed by a brief review of Einstein's equations in section 2.2. In section 2.3 we will show the ADM decomposition of Einstein's field equations. In section 2.4 we make the ADM equations strongly hyperbolic by deriving the BSSN equations. We then present the BSSN equations in covariant form in section 2.5. Finally, in section 2.6 we make a brief introduction to the general relativistic magnetohydrodynamics formalism we will use later on to study neutron stars.

2.1 Notation and conventions

Throughout this thesis we will adopt the metric to have the mostly positive sign convention $(-,+,+,+)$. We will also adopt, unless otherwise stated, units where $G_{\rm N} = 1 = c$, called *geometrized units*. This means that, for example, the metric for a Schwarzschild black hole is written, in spherical coordinates, as

$$ds^2 = -\left(1 - \frac{2M}{r}\right) dt^2 + \left(1 - \frac{2M}{r}\right)^{-1} dr^2 + r^2 \left(d\theta^2 + \sin^2\theta \, d\varphi^2\right) \ . \tag{2.1}$$

We adopt index notation throughout the dissertation with Einstein's summation convention implied. We use Greek letters $(\mu, \nu, \ldots)$ to label indices that describe all of spacetime, thus running through $(0, 1, 2, 3)$. Meanwhile, Latin letters $(i, j, \ldots)$ represent the spatial components only and therefore run through $(1, 2, 3)$. The spacetime metric, $g_{\mu\nu}$, is used to raise and lower spacetimes indices from vectors and tensors, i.e.

$$x^\mu = g^{\mu\nu} x_\nu \ , \tag{2.2}$$

and

$$x_\mu = g_{\mu\nu} x^\nu \ . \tag{2.3}$$

We adopt the shorthand notation

$$\frac{\partial \psi}{\partial x^\mu} \equiv \partial_\mu \psi \ , \ \frac{\partial \psi}{\partial x_\mu} \equiv \partial^\mu \psi \ , \ \frac{\partial f}{\partial t} \equiv \partial_t f \ , \ \frac{\partial f}{\partial r} \equiv \partial_r f \ , \tag{2.4}$$

to express derivatives.

On our discussion of the ADM formalism in section 2.3, we will adopt, as is usual in

numerical relativity, *abstract index notation* to represent tensors. Specifically, we will use the convention that "early" latin indices, $(a, b, c, \ldots)$, do not represent a tensor component, but instead the abstract, coordinate-free tensor itself. Thus, while we view $T_{\mu\nu}$ as the components of the tensor $T = T_{\mu\nu} dx^\mu dx^\nu$ on a particular coordinate basis, the object T_{ab} is to be thought of as the tensor T, not its components. Thus, these letters will be used to represent tensors that live in spacetime. The "middle" latin indices, $(i, j, k, \ldots)$, however, will continue to be used on tensors which live only on spatial surfaces of constant time.

2.2 Einstein's field equations

Einstein's equations relate the curvature of spacetime, as described by the Einstein tensor G_{ab}, with the energy-momentum tensor, T_{ab}, by

$$G_{ab} \equiv R_{ab} - \frac{1}{2} g_{ab} R = 8\pi T_{ab} \ . \tag{2.5}$$

The left-hand side (LHS) of equation (2.5) is determined uniquely after specifying the metric g_{ab}, with R_{ab} and R, the Ricci tensor and Ricci scalar, respectively, being combinations of the metric and its derivatives. The right-hand side (RHS) of equation (2.5) is determined uniquely after specifying the matter content of the spacetime. Thus, paraphrasing one of the most well known sentences physicists use, matter tells spacetime how to curve, while the curvature tells matter how it should move.

To understand a bit more about the tensors in equation (2.5), it is useful to remind ourselves of the *Riemann curvature tensor*, defined as

$$R^d{}_{acb} \equiv \partial_c \Gamma^d_{ab} - \partial_b \Gamma^d_{ac} + \Gamma^d_{ce} \Gamma^e_{ab} - \Gamma^d_{be} \Gamma^e_{ac} \ , \tag{2.6}$$

where we have already adopted the torsion-free metric connection whose components are the *Christoffel symbols*, Γ^c_{ab}, given by

$$\Gamma^c_{ab} = \frac{1}{2} g^{cd} \left(\partial_a g_{bd} + \partial_b g_{ad} - \partial_d g_{ab} \right) \ . \tag{2.7}$$

The Ricci tensor is then defined as the contraction between the first and third indices of the Riemann tensor[2]

[2]While at first this may seem arbitrary, this is the only index contraction of the Riemann tensor that is relevant, the others yielding either zero or minus the same result. This is related to the symmetry

$$R_{ab} \equiv R^{c}{}_{acb} = g^{cd} R_{cabd} \ . \tag{2.8}$$

Finally, the Ricci scalar is defined as the trace of R_{ab}

$$R \equiv R^{a}{}_{a} = g^{ab} R_{ab} \ . \tag{2.9}$$

Einstein's equations from a variational principle

Here we sketch the derivation of Einstein's equations by applying the principle of least action to the *Einstein-Hilbert action*, $S_{\rm EH}$, given by

$$S_{\rm EH} = \frac{1}{16\pi} \int d^4x \, \sqrt{-g} \, R \ , \tag{2.10}$$

where g is the determinant of the metric g_{ab}. Einstein's equations can then be derived by considering a small variation of the metric

$$g_{ab} \to g_{ab} + \delta g_{ab} \ , \tag{2.11}$$

assuming δg_{ab} infinitesimal. This variation implies a variation in the action of the form

$$\delta S_{\rm EH} = \frac{1}{16\pi} \int d^4x \, \sqrt{-g} \, \left[g^{ab} \delta R_{ab} + \delta g^{ab} R_{ab} - \frac{1}{2} g_{ab} \delta g^{ab} \, R \right] \ , \tag{2.12}$$

where we have already used the identity $\delta\sqrt{-g} = +\frac{1}{2}\sqrt{-g}\, g^{ab}\delta g_{ab} = -\frac{1}{2}\sqrt{-g}\, g_{ab}\delta g^{ab}$ and kept only terms linear in δg^{ab}. Dropping the surface terms[3] that arise from the first term in square brackets in (2.12) we have

$$\delta S_{\rm EH} = \frac{1}{16\pi} \int d^4x \, \sqrt{-g} \left[R_{ab} - \frac{1}{2} g_{ab} R \right] \delta g^{ab} \ . \tag{2.13}$$

Coupling the Einstein-Hilbert action with a matter action, $S_{\rm matter}$, and defining the canonical energy-momentum tensor as

$$T_{ab} \equiv -\frac{2}{\sqrt{-g}} \frac{\delta S_{\rm matter}}{\delta g^{ab}} \ , \tag{2.14}$$

properties of the indices of the Riemann tensor, which we will not go into detail here. See e.g. chapter V.1 of [61] for a rather pedagogical introduction/review of the subject.

[3] Again, a more complete description can be found in [61].

yields, by invoking the principle of least action,

$$R_{ab} - \frac{1}{2} g_{ab} R = 8\pi T_{ab} \ , \tag{2.15}$$

which are Einstein's field equations (2.5).

2.3 The Arnowitt-Deser-Misner decomposition

In this section we will discuss the *Arnowitt-Deser-Misner (ADM) decomposition* [12] of Einstein's field equations. Our notation and exposition will follow closely those of [48], but we will omit some of the derivation and try to make some of the explanations clearer.

Spacetime has no preferred folliation. We could choose to slice spacetime along a particular spatial direction, say z, and consider then surfaces of constant (t, x, y), in Cartesian coordinates. However, as humans we tend to choose a folliation which is intuitive to us. A more "natural" choice would be to slice spacetime into spatial surfaces of constant time. Why would we choose to folliate spacetime at all is then the next question we should address. Einstein's equations make no distinction between space and time, so why should we? The answer is because by doing this, we turn the problem of solving Einstein's equations into a Cauchy problem: we specify some initial condition by telling what is the initial spatial surface, $\Sigma(t = 0)$, then evolve this surface in time. A folliation of this kind is illustrated in figure 2.1.

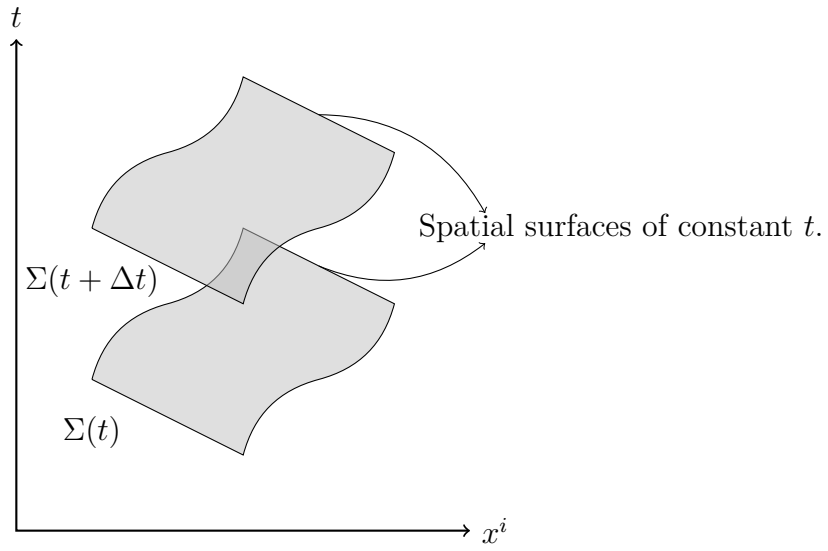


Figure 2.1: Sketch of a (3+1) spacetime folliation. The gray surfaces, labelled by Σ, are surfaces of constant coordinate time.

Let us consider the decomposition of a $4d$ spacetime manifold $\mathcal{M}$ with metric g_{ab} into $3d$ space plus time. This is what we will call a *(3+1)d decomposition*. In this decomposition, we will assume that the spacetime $(\mathcal{M}, g_{ab})$ can be foliated into a family of non-intersecting spatial surfaces, Σ, which can be viewed as level surfaces of a scalar function t, which can be interpreted as a time function. Let us define the 1-form

$$\Omega_a \equiv \nabla_a t \ , \tag{2.16}$$

where ∇_a is the covariant derivative compatible with the spacetime metric g_{ab}. Using the spacetime metric, we can compute the norm of this 1-form

$$\Omega^2 \equiv g^{ab}\nabla_a t \nabla_b t \equiv -\frac{1}{\alpha^2} \ , \tag{2.17}$$

defined above to have the value $-\alpha^{-2}$. This value, α, measures how much proper time elapses between time slices along the normal vector to the slice, Ω^a, and is called the *lapse function*. We assume $\alpha > 0$ so that Ω^a is timelike and the surfaces Σ are spacelike.

We state here that α depends on the spatial and time coordinates, thus is may take different values at each point of the surface, which do not necessarily remain constant throughout the time evolution of the surface. The lapse function encodes our freedom to slice the time-like evolution as we see fit – the best way to look at it is as a *gauge variable*.

Let us define the normalized 1-form $\omega_a \equiv \alpha\Omega_a$. Then, the unit normal vector to the slices is

$$n^a \equiv -g^{ab}\omega_b \ . \tag{2.18}$$

This vector is timelike by construction

$$n^a n_a = g^{ab}\omega_a\omega_b = -1 \ , \tag{2.19}$$

where we have used

$$n_a = g_{ab}n^b = -g_{ab}g^{bc}\omega_c = -\delta_a{}^c\omega_c = -\omega_a \ . \tag{2.20}$$

Now we are ready to define a *projection operator*, P^a_b, which projects the spacetime components of a tensor onto the spatial surface

$$P^a_{\ b} \equiv \delta^a_{\ b} + n^a n_b \ . \tag{2.21}$$

We can apply this projection operator to our 4-dimensional metric

$$\begin{aligned} P^c_{\ a} P^d_{\ a} g_{cd} &= \left(\delta^c_{\ a} + n^c n_a\right)\left(\delta^d_{\ b} + n^d n_b\right) g_{cd} \\ &= \left(\delta^c_{\ a}\delta^d_{\ b} + \delta^c_{\ a} n^d n_b + n^c n_a \delta^d_{\ b} + n^c n_a n^d n_b\right) g_{cd} \\ &= g_{ab} + g_{ac} n^c n_b + n^c n_a g_{cb} + n_c n_a n^c n_b \\ &= g_{ab} + n_a n_b + n_b n_a - n_a n_b \\ &= g_{ab} + n_a n_b \\ &= \gamma_{ab} \ . \end{aligned} \tag{2.22}$$

The last equality in (2.22) defines the metric γ_{ab}, which is purely spatial, i.e. resides entirely in Σ, as

$$\gamma_{ab} = g_{ab} + n_a n_b \ . \tag{2.23}$$

It is easy to check that the vector n^a is indeed normal to γ_{ab}

$$n^a \gamma_{ab} = n^a \left(g_{ab} + n_a n_b\right) = n_b - n_b = 0 \ . \tag{2.24}$$

2.3.1 The extrinsic curvature

The extrinsic curvature, K_{ab}, tells us how our spatial surfaces are embedded in $4d$ spacetime. We can think of K_{ab} as the "apparent curvature" due to embedding. The canonical example of this is the cylinder: the cylinder is flat, i.e. parallel transport[4] of *any* vector on the surface of the cylinder is trivial. In other words, the cylinder has zero *intrinsic curvature*. However, when we look at a bottle of water, we perceive it as curved. This is due to its embedding into $3d$ space: an ant living on the surface of our cylindric bottle cannot make an experiment which detects this curvature.

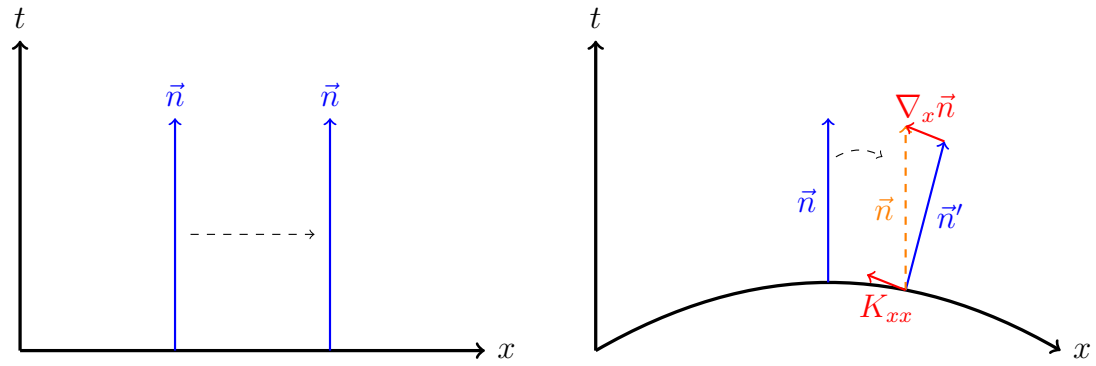
The most intuitive definition of extrinsic curvature is the following. Consider the normal vector to our surface at a given point. If we now move our normal vector a small

[4] See e.g. [61] for a good review.

distance away from our starting point via parallel transport, the extrinsic curvature will measure the projection onto the spatial surface of the change in n^a caused by this parallel transport:

$$K_{ab} = -P^c{}_a \nabla_c n_b = -\left(\nabla_a n_b + n_a n^c \nabla_c n_b\right) \ . \tag{2.25}$$

By projecting this change to the spatial surface, we ensure that K_{ab} is both symmetric and intrinsic to the spatial surface. With this projection, K_{ab} is defined as if all normal observers followed the geodesic congruence $\alpha = 1$, which means that it is a gauge independent quantity. This projection process is illustrated in figure 2.2.



(a) A scenario where the extrinsic curvature is zero.

(b) A scenario where the extrinsic curvature is not zero.

Figure 2.2: The figures above illustrate the concept of extrinsic curvature. On figure (a), we have the normal vector to the surface, $\vec{n}$, being transported to the right. The normal vector does not change and, thus, the extrinsic curvature is zero. On figure (b), however, the vector $\vec{n}$ changes to the vector $\vec{n}'$ when moved to the right. The orange dashed arrow indicates what $\vec{n}'$ should be if no change occured. The difference between these vectors is $\nabla_x \vec{n}$, illustrated in red. The extrinsic curvature, K_{xx}, is the projection of this difference onto the surface, illustrated in red as well.

Alternatively, but equivalently, the extrinsic curvature may be defined as the Lie derivative of the metric along the normal direction:

$$K_{ab} = -\frac{1}{2}\mathcal{L}_n \gamma_{ab} \ . \tag{2.26}$$

We can also compute the trace of K_{ab}, $K = g^{ab}K_{ab}$, from (2.25), which yields

$$K = -\nabla_c n^c \ , \tag{2.27}$$

since n^a is unitary and thus orthogonal to its gradient. In the case of a homogeneous and

isotropic universe, K is related to the Hubble constant, H, by $K = -3H$. In this sense, negative values of K correspond to an expanding universe, while positive values of K to a collapsing one.

2.3.2 The constraint and evolution equations

The objective of this subsection is to project Einstein's equations onto the spatial surfaces. To do so, we will make use of two well known equations: Gauss' equation and the Codazzi equation. *Gauss' equation* is nothing more than the projection of each index of the Riemann tensor onto the spatial surface

$$P^{p}{}_{a} P^{q}{}_{b} P^{r}{}_{c} P^{s}{}_{d}\, R_{pqrs} = {}^{(3)}R_{abcd} + K_{ac}K_{bd} - K_{ad}K_{cb} \ , \tag{2.28}$$

and relates the $4d$ Riemann tensor, R_{abcd}, with the $3d$ Riemann tensor, ${}^{(3)}R_{abcd}$, constructed from γ_{ab}, and terms quadratic in the extrinsic curvature. The *Coddazi equation* represents a projection of three indices of the $4d$ Riemann tensor and then the contraction of the last index with the normal vector

$$P^{p}{}_{a} P^{q}{}_{b} P^{r}{}_{c} n^{s}\, R_{pqrs} = D_b K_{ac} - D_a K_{bc} \ , \tag{2.29}$$

and relates the $4d$ Riemann tensor with the spatial derivatives of the extrinsic curvature. In equation (2.29), D_a are covariant derivatives compatible with the induced metric γ_{ab}. These two equations can be thought of as integrability conditions, allowing the embedding of a $3d$ slice Σ with initial data (γ_{ab}, K_{ab}) inside a $4d$ manifold with metric g_{ab}.

The first equation we are interested in is found by contracting (2.28) with $g^{ac}g^{bd}$:

$$n^a n^b G_{ab} = {}^{(3)}R + K^2 - K_{ab}K^{ab} \ . \tag{2.30}$$

Replacing $G_{ab} = 8\pi T_{ab}$ gives

$$n^a n^b \left(8\pi T_{ab}\right) = {}^{(3)}R + K^2 - K_{ab}K^{ab} \ . \tag{2.31}$$

Then, defining the *energy density* $\rho \equiv n^a n^b T_{ab}$, we get

$$\boxed{\mathcal{H} \equiv {}^{(3)}R + K^2 - K_{ab}K^{ab} - 16\pi\rho = 0} \ . \tag{2.32}$$

Equation (2.32) is known as the *Hamiltonian constraint*. This equation involves no time derivatives and is gauge independent. It is not related to the evolution of the functions contained in it, but rather with enforcing a relation that must be satisfied on each spatial surface. If we specify initial data such that it does not satisfy the Hamiltonian constraint, then it will not satisfy Einstein's equations.

The second equation of interest is obtained contracting (2.29) with $g^{ac}g^{bd}$:

$$P^{ab}n^{c}G_{bc} = D_b\left(\gamma^{ab}K - K^{ab}\right) \ . \tag{2.33}$$

Then, defining the *momentum density* $S_a \equiv -\gamma^{q}_{\ a}n^{s}T_{qs}$, and replacing the Einstein tensor in favour of the energy-momentum tensor we get

$$\boxed{\mathcal{M}_a \equiv D_b K^{b}_{\ a} - D_a K - 8\pi S_a = 0} \ . \tag{2.34}$$

Equation (2.34) is known as the *momentum constraint*. This equation, again, gives us no information on how to move from one slice to the next, but it must be satisfied on each slice if our data is to satisfy Einstein's equations.

Having found the relations that must be satisfied on each surface of constant time, we must now find the equations that take us from one spatial surface to the next. There is one caveat when doing this. If we start at a particular spatial point (X^1, X^2, X^3) and travel in the direction normal to the surface $\Sigma(t)$ at that point to the surface $\Sigma(t + \Delta t)$, there is no guarantee that we will end in the same spatial point. This is due to the fact that n^a is not dual to the surface 1-form Ω_a, since their dot product is not unity

$$n^{a}\Omega_a = -\alpha g^{ab}\nabla_a t \nabla_b t = \frac{1}{\alpha} \ . \tag{2.35}$$

Instead, we consider the vector

$$t^a \equiv \alpha n^a + \beta^a \ , \tag{2.36}$$

which is dual to Ω_a,

$$t^{a}\Omega_a = \alpha n^{a}\Omega_a + \beta^{a}\Omega_a \equiv 1 \ , \tag{2.37}$$

for any spatial *shift vector* β^a. It is useful to choose t^a to be the congruence along which

we propagate the spatial coordinate grid from one time slice to the next. This means that t^a will connect points with the same spatial coordinates on neighboring slices. We can also view β^a as the projection of t^a onto the spatial surface:

$$\begin{aligned} P^a{}_b t^b &= (\delta^a{}_b + n^a n_b)\left(\alpha n^a + \beta^b\right) \\ &= \beta^a + \alpha n^a - \alpha n^a \\ &= \beta^a \ . \end{aligned} \tag{2.38}$$

Hence, the shift vector will measure the amount by which the spatial coordinates are shifted within a slice with respect to the normal vector. The situation is illustrated in figure 2.3.

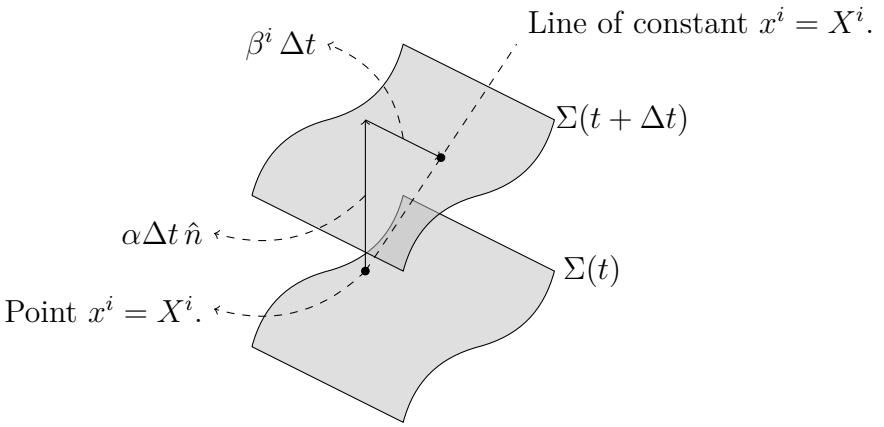


Figure 2.3: Sketch to illustrate the roles of the lapse function, α, and the shift vector, β^i. The figure can be misleading because the two surfaces, $\Sigma(t)$ and $\Sigma(t + \Delta t)$, are drawn as exact copies of one another, but it is sufficient to illustrate the concept.

As we mentioned before, the lapse function α allows us to measure how much proper time elapses when moving from one surface of constant time to the next. As with the lapse function, the shift vector β^a can also be viewed as a gauge variable, one that allows us to relable spatial coordinates on each slice in an arbitrary way. These four[5] gauge variables embody the four-fold coordinate degrees of freedom inherent to general relativity. It is worth mentioning that the choices for α and β^a are quite arbitrary, but there are some common choices used in numerical relativity, some of which we will discuss later.

From equation (2.26), we can show, using (2.36), that

[5]Remember that $n^a \beta_a = 0$ because β^a is a purely spatial quantity, thus only 3 of its components are independent.

$$\boxed{\mathcal{L}_t \gamma_{ab} = -2\alpha K_{ab} + \mathcal{L}_\beta \gamma_{ab}} \ . \tag{2.39}$$

This can be viewed as an evolution equation for the metric, but also as the definition of the extrinsic curvature. To find our final equation, we contract (2.28) with g^{ac} to get

$$P^f_{\ b} P^h_{\ d} \left({}^{(4)}R_{fh} + n^e n^g \, {}^{(4)}R_{efgh} \right) = {}^{(3)}R_{bd} + K K_{bd} - K_{bc} K^c_{\ d} \ . \tag{2.40}$$

It is possible to show that [48]

$$P^f_{\ b} P^h_{\ d} \left(n^e n^g \, {}^{(4)}R_{efgh} \right) = \mathcal{L}_n K_{bd} - K_{bc} K^c_{\ d} + \frac{1}{\alpha} D_b D_d \alpha \ . \tag{2.41}$$

Then, rewriting Einstein's equations in terms of the trace of T_{ab}, $R_{ab} = 8\pi \left(T_{ab} - \frac{1}{2} g_{ab} T \right)$, and defining $S_{ab} \equiv P^c_{\ a} P^d_{\ b} T_{cd}$, we can rearrange (2.41) as

$$\boxed{\begin{aligned} \mathcal{L}_t K_{ab} = \mathcal{L}_\beta K_{ab} - D_a D_b \alpha + \alpha \left({}^{(3)}R_{ab} + K K_{ij} - 2 K_{ac} K^c_{\ b} \right) \\ + 4\pi\alpha \left[\gamma_{ab} \left(S - \rho \right) - 2 S_{ab} \right] \end{aligned}} \ , \tag{2.42}$$

which gives us an evolution equation for the extrinsic curvature. We now have a full set of equations to evolve the metric and the extrinsic curvature in time.

2.3.3 Choice of basis vectors

We are now going to consider a particular choice of basis vectors $e^a_{(b)}$,[6] which have been completely arbitrary so far. We will introduce a basis of three spatial vectors $e^a_{(i)}$ that reside in a particular surface Σ,

$$\Omega_a e^a_{(i)} = 0 \ . \tag{2.43}$$

These basis vectors move from one spatial surface to the next along the t^a direction. The fourth basis vector we will choose is $e^a_{(0)} = t^a$. Because we want the other basis vectors to connect the same spatial points on neighboring slices, condition (2.37) implies

$$t^a = e^a_{(0)} = (1, 0, 0, 0) \ . \tag{2.44}$$

[6] Jargon alert: when writing $e^a_{(b)}$ as the basis vectors, the index a represents the components of each basis vector (b).

From equation (2.43) we find that

$$\Omega_a e^a_{(i)} = -\frac{1}{\alpha} n_a e^a_{(i)} = 0 \ , \tag{2.45}$$

which implies that the covariant spatial components of the normal vector have to vanish

$$n_i = (0, 0, 0) \ . \tag{2.46}$$

Since spatial tensors vanish when contracted with the normal vector, this also means that all components of spatial tensors with a contravariant index equal to zero must vanish. For example, the shift vector must have $n_a \beta^a = 0$, and since $n_i = 0$ we get

$$\beta^a = (0, \beta^i) \ . \tag{2.47}$$

For the sake of the reader, let us rewrite here equation (2.36)

$$t^a \equiv \alpha n^a + \beta^a \ , \tag{2.36}$$

which, together with our choice (2.44) and result (2.47), can now be solved for n^a, yielding

$$n^a = \left(\frac{1}{\alpha}, -\frac{\beta^i}{\alpha} \right) \ . \tag{2.48}$$

From the normalization condition $n_a n^a = -1$ we find

$$n_a = (-\alpha, 0, 0, 0) \ . \tag{2.49}$$

The $(3+1)d$ form of the $4d$ metric

Now we are ready to determine our metric in its $(3+1)d$ decomposed form. Remember that $\gamma_{ij} = g_{ij}$. Then, using the relation

$$g^{ab} = \gamma^{ab} - n^a n^b \ , \tag{2.50}$$

and the fact that spatial components of spatial contravariant tensors have to vanish (and also $\gamma^{a0} = 0$), we have

$$g^{ab} = \begin{pmatrix} -1/\alpha^2 & \beta^i/\alpha^2 \\ \beta^i/\alpha^2 & \gamma^{ij} - \beta^i\beta^j/\alpha^2 \end{pmatrix} . \tag{2.51}$$

We can now invert this matrix

$$g_{ab} = \begin{pmatrix} -\alpha^2 + \gamma_{mn}\beta^m\beta^n & \gamma_{ik}\beta^k \\ \gamma_{jk}\beta^k & \gamma_{ij} \end{pmatrix} , \tag{2.52}$$

which, combined with $\beta_i = \gamma_{ij}\beta^j$, gives

$$g_{ab} = \begin{pmatrix} -\alpha^2 + \beta_\ell\beta^\ell & \beta_i \\ \beta_j & \gamma_{ij} \end{pmatrix} . \tag{2.53}$$

With all these considerations, our $4d$ metric has been decomposed into its $(3+1)d$ form, which we can express by the line element

$$\boxed{ds^2 = (-\alpha^2 + \beta_\ell\beta^\ell)dt^2 + 2\beta_i dx^i dt + \gamma_{ij}dx^i dx^j} . \tag{2.54}$$

2.3.4 The ADM equations

Finally, we can use our choice of basis vectors to eliminate some of the components in our constraint equations, (2.32) and (2.34), and evolution equations, (2.39) and (2.42). The Hamiltonian and momentum constraints are given by

$$\boxed{{}^{(3)}R + K^2 - K_{ij}K^{ij} = 16\pi\rho} , \tag{2.55}$$

and

$$\boxed{D_j K^j{}_i - D_i K = 8\pi S_i} , \tag{2.56}$$

respectively. The evolution equations for the extrinsic curvature and the metric are given by

$$\boxed{\begin{aligned} \partial_t K_{ij} = {} & \beta^\ell\partial_\ell K_{ij} + K_{i\ell}\partial_j\beta^\ell + K_{j\ell}\partial_i\beta^\ell - D_i D_j\alpha \\ & + \alpha\left({}^{(3)}R_{ij} + KK_{ij} - 2K_{i\ell}K^\ell{}_j\right) + 4\pi\alpha\left[\gamma_{ij}\left(S-\rho\right) - 2S_{ij}\right] \end{aligned}} . \tag{2.57}$$

and

$$\boxed{\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i\beta_j + D_j\beta_i}\,, \tag{2.58}$$

respectively. This set of equations is equivalent to Einstein's equations and in this decomposed format is often referred to as the *ADM equations*.

2.3.5 The polar/radial gauge in spherical coordinates

Here we will fix a coordinate system such that we specify the values of the gauge parameters α and β^i. Our first assumption will be that of spherical symmetry, so that we use coordinates (t, r, θ, φ) and our spacial metric has the general form

$$\gamma_{ij} = \begin{pmatrix} a^2 & 0 & 0 \\ 0 & b^2 r^2 & 0 \\ 0 & 0 & b^2 r^2 \sin^2\theta \end{pmatrix}\,, \tag{2.59}$$

where both a and b are functions of t and r. Furthermore, we must impose the condition of *elementary flatness* [62] at $r = 0$. From a physical perspective, this condition demands that no conical singularities form around the origin. From a numerical point of view, this imposes regularity at the origin by demanding that fields do not blow up at $r = 0$.

Now we want to make sure that our coordinate r is indeed an areal coordinate, meaning that a surface of constant t and constant r has proper area $4\pi r^2$. This is satisfied if $b(t, r) = 1$, which is known as the *radial coordinate condition*.

Regularity at the origin for the radial metric function means setting it to a finite value, and we choose it to

$$a(t, 0) = 1\,, \tag{2.60}$$

which will is used as a boundary condition.

With these choices we now have the *radial gauge*, for which $\beta^\theta = 0 = \beta^\varphi$. At this stage we have the following:

1. Our spatial metric is $\gamma_{ij} = \mathrm{Diag}\,(a^2(r, t), r^2, r^2 \sin^2\theta)$.

2. Our shift vector is $\beta^i = (\beta^r, 0, 0)$.

3. The function $a(t, r)$ must satisfy the boundary condition $a(t, 0) = 1$.

Now we are ready to impose a condition on α, the *polar slicing condition*. The polar slicing is defined by[7]

$$K^{\theta}_{\ \theta} = 0 = K^{\varphi}_{\ \varphi} \ , \tag{2.61}$$

so that the trace of the extrinsic curvature is $K = K^{r}_{\ r}$. This condition implies $\beta^r = 0$ [48]. Therefore our full, $(3+1)d$ metric can be written in the simple form

$$ds^2 = -\alpha^2(r,t)\, dt^2 + a^2(r,t)\, dr^2 + r^2 \left(d\theta^2 + \sin^2\theta\, d\varphi^2\right) \ , \tag{2.62}$$

which is known as the $(3+1)d$ metric in the *polar/radial gauge*.

2.3.6 A brief comment on degrees of freedom

We here make a simple count of our degrees os freedom to show that we have properly used all our gauge freedom. We started with 10 degrees of freedom, the 10 components of $g_{\mu\nu}$. Then, we enforce 4 constraint equations, the Hamiltonian and momentum constraints. This implies that 4 of the 10 components are no longer independent. Then, we have our gauge freedom: we can specify α and β^i, the lapse function and shift vector, respectively, and remove 4 degrees of freedom. Thus, $10-4-4=2$ degrees of freedom is what we are left with, which are the two polarizations of gravitational waves, as expected.

2.4 The BSSN formalism

We will now look at an extension of the ADM equations due to Baumgarte, Shapiro, Shibata, and Nakamura known as the *BSSN formalism*. The BSSN formalism introduces new auxiliary variables to improve the stability of the original ADM equations, which, when coupled with the so-called standard gauge conditions [54, 55], have proven to be incredibly robust [50–53].

We highly encourage readers who are encountering the BSSN formalism for the first time to read through appendices A and B, where we discuss the classification of partial

[7] This might seem confusing since we are imposing the condition on the extrinsic curvature, but this essentially changes the equation for α, namely the contracted version of (2.39),

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

so that we are effectivelly imposing a condition on our lapse function.

differential equatoins, notions of hyperbolicity, and go over some of the ideas behind the modifications we will do to the ADM equations in the simpler case of Maxwell's equations. For those interested in a slightly more in-depth exposition of these ideas, but that is still pedagogical, we recommend [13, 48], while a more rigorous discussion on the hyperbolicity of partial differential equations can be found in [49].

We start by introducing a new metric, $\bar{\gamma}_{ij}$, which we refer to as the *conformal metric*. This conformal metric is related to the ADM 3-metric, γ_{ij}, via the conformal transformation

$$\bar{\gamma}_{ij} = e^{-4\phi}\gamma_{ij} \ , \tag{2.63}$$

where ϕ is referred to as the *evolved conformal factor*. We then demand that the determinant of the conformal metric be equal to that of the flat metric, η_{ij}, in whatever coordinate system we are using. In the original BSSN formulation, η_{ij} is chosen to be the flat metric in Cartesian coordinates, imposing the constraint

$$\bar{\gamma} = 1 = \eta \ . \tag{2.64}$$

Taking the determinant of equation (2.63) we find the identity

$$\bar{\gamma} = 1 = e^{-12\phi}\gamma \implies \phi = \frac{1}{12}\ln\gamma \ . \tag{2.65}$$

Instead of evolving the extrinsic curvature directly, the BSSN formalism decomposes the extrinsic curvature into its traceless and pure trace parts, namely

$$K_{ij} = A_{ij} + \frac{1}{3}\gamma_{ij}K \ , \tag{2.66}$$

where $K \equiv \gamma^{ij}K_{ij}$. Furthermore, we introduce a conformal counterpart to A_{ij} which is defined as

$$\bar{A}_{ij} = e^{-4\phi}A_{ij} \quad \textit{and} \quad \bar{A}^{ij} = e^{4\phi}A^{ij} \ . \tag{2.67}$$

We note that the indices of $\bar{A}_{ij}$ are raised and lowered using the conformal metric $\bar{\gamma}_{ij}$, hence the transformation equation for $\bar{A}^{ij}$ above.

To find an evolution equation for ϕ, we start with the evolution equation for γ_{ij},

equation (2.58). Contracting that equation with γ^{ij} we get

$$\gamma^{ij}\partial_t\gamma_{ij} = \partial_t \ln\gamma = -2\alpha K + 2D_i\beta^i \ . \tag{2.68}$$

Expanding the covariant derivative of the shift vector we get

$$\begin{aligned} D_i\beta^i &= \partial_i\beta^i + \Gamma^i_{ij}\beta^j \\ &= \partial_i\beta^i + \frac{\beta^j}{2}\left[\gamma^{i\ell}\left(\partial_i\gamma_{j\ell} + \partial_j\gamma_{i\ell} - \partial_\ell\gamma_{ij}\right)\right] \\ &= \partial_i\beta^i + \frac{\beta^j}{2}\gamma^{i\ell}\partial_j\gamma_{i\ell} \\ &= \partial_i\beta^i + \frac{1}{2}\beta^j\partial_j \ln\gamma \ . \end{aligned} \tag{2.69}$$

Substituting result (2.69) and the identity (2.65) into (2.68) we obtain

$$\partial_t\phi = \beta^j\partial_j\phi - \frac{1}{6}\alpha K + \frac{1}{6}\partial_i\beta^i \ . \tag{2.70}$$

An evolution equation for K can be found by taking the trace of equation (2.57), which yields[8]

$$\partial_t K = -D^2\alpha + \alpha\left[K_{ij}K^{ij} + 4\pi\left(\rho + S\right)\right] + \beta^i D_i K \ . \tag{2.71}$$

Consider, also

$$\begin{aligned} K_{ij}K^{ij} &= \left(A_{ij} + \frac{1}{3}\gamma_{ij}K\right)\left(A^{ij} + \frac{1}{3}\gamma^{ij}K\right) \\ &= A_{ij}A^{ij} + \frac{2}{3}K\gamma^{ij}A_{ij} + \frac{1}{3}K^2 \\ &= \bar{A}_{ij}\bar{A}^{ij} + \frac{1}{3}K^2 \ , \end{aligned} \tag{2.72}$$

where we have employed the fact that A_{ij} is traceless and the transformations (2.67) in the last equality. Combining (2.71) and (2.72) we get, finally, the evolution equation for K in terms of the BSSN variables

$$\partial_t K = -D^2\alpha + \alpha\left(\bar{A}_{ij}\bar{A}^{ij} + \frac{1}{3}K^2\right) + 4\pi\alpha\left(\rho + S\right) + \beta^i\partial_i K \ . \tag{2.73}$$

[8]Recall the the trace is a linear operation, so that $\partial\left(\mathrm{tr}M\right) = \mathrm{tr}\partial M$. We also use the Hamiltonian

We can obtain an evolution equation for $\bar{\gamma}_{ij}$ by considering

$$\partial_t \bar{\gamma}_{ij} = \partial_t \left(e^{-4\phi} \gamma_{ij} \right) = -4 \bar{\gamma}_{ij} \partial_t \phi + e^{-4\phi} \partial_t \gamma_{ij} \ . \tag{2.74}$$

We can manipulate this equation to get it into a nicer form, but we omit this derivation because it is tedious and not very illuminating. The result is

$$\partial_t \bar{\gamma}_{ij} = -2\alpha \bar{A}_{ij} + \beta^k \partial_k \bar{\gamma}_{ij} + \bar{\gamma}_{ik} \partial_j \beta^k + \bar{\gamma}_{kj} \partial_i \beta^k - \frac{2}{3} \bar{\gamma}_{ij} \partial_k \beta^k \ . \tag{2.75}$$

Similarly, one can write

$$\partial_t \bar{A}_{ij} = \partial_t \left(e^{-4\phi} A_{ij} \right) = -4 \bar{A}_{ij} \partial_t \phi + e^{-4\phi} \partial_t \left(K_{ij} - \frac{1}{3} \gamma_{ij} K \right) \ , \tag{2.76}$$

and use the evolution equations for ϕ, K_{ij}, K, and γ_{ij} to obtain

$$\begin{aligned} \partial_t \bar{A}_{ij} = {} & \beta^k \partial_k \bar{A}_{ij} + \bar{A}_{ik} \partial_j \beta^k + \bar{A}_{kj} \partial_i \beta^k - \frac{2}{3} \bar{A}_{ij} \partial_k \beta^k \\ & + \alpha \left(K \bar{A}_{ij} - 2 \bar{A}_{ik} \bar{A}^k{}_j \right) + e^{-4\phi} \left[-D_i D_j \alpha + \alpha \left(R_{ij} - 8\pi S_{ij} \right) \right]^{\mathrm{TF}} \ , \end{aligned} \tag{2.77}$$

The superscript TF indicates the trace-free part of a tensor, e.g.

$$R_{ij}^{\mathrm{TF}} = R_{ij} - \frac{1}{3} \gamma_{ij} R \ . \tag{2.78}$$

We now introduce the *conformal connection functions*[9]

$$\bar{\Gamma}^i \equiv \bar{\gamma}^{jk} \bar{\Gamma}^i_{jk} = -\partial_j \bar{\gamma}^{ij} \ . \tag{2.79}$$

The last equality above holds because of the constraint $\bar{\gamma} = 1$. In terms of $\bar{\Gamma}^i$, the Ricci tensor associated with the conformal metric can be written as

$$\bar{R}_{ij} = -\frac{1}{2} \bar{\gamma}^{lm} \partial_m \partial_l \bar{\gamma}_{ij} + \bar{\gamma}_{k(i} \partial_{j)} \bar{\Gamma}^k + \bar{\Gamma}^k \bar{\Gamma}_{(ij)k} + \bar{\gamma}^{lm} \left[2 \bar{\Gamma}^k_{l(i} \bar{\Gamma}_{j)km} + \bar{\Gamma}^k_{im} \bar{\Gamma}_{klj} \right] \ . \tag{2.80}$$

Note that, with the introduction of the conformal connection functions, the only explicit

constraint (2.55) to obtain equation (2.71).

[9] This is analogous to introducing the auxiliary variable Γ, as we do in appendix B for Maxwell's equations

second derivative of the conformal metric that appears in the Ricci tensor is in the form of a Laplace operator, $\bar{\gamma}^{lm}\partial_m\partial_l$.

If we were to fix the values of $\bar{\Gamma}^i$, this would be equivalent to choosing a particular gauge. Instead, we will promote them to new, independent evolution variables. To obtain an evolution equation for $\bar{\Gamma}^i$ we simply take a time derivative of definition (2.79). Using the evolution equation for $\bar{\gamma}_{ij}$ and momentum constraints it is possible to write this evolution equation as

$$\begin{aligned}\partial_t\bar{\Gamma}^i =& -2\bar{A}^{ij}\partial_j\alpha + 2\alpha\left(\bar{\Gamma}^i_{jk}\bar{A}^{kj} - \frac{2}{3}\bar{\gamma}^{ij}\partial_j K - 8\pi\bar{\gamma}^{ij}S_j + 6\bar{A}^{ij}\partial_j\phi\right)\\ &+ \beta^j\partial_j\bar{\Gamma}^i - \bar{\Gamma}^j\partial_j\beta^i + \frac{2}{3}\bar{\Gamma}^i\partial_j\beta^j + \frac{1}{3}\bar{\gamma}^{li}\partial_l\partial_j\beta^j + \bar{\gamma}^{lj}\partial_j\partial_l\beta^i\ .\end{aligned} \tag{2.81}$$

In terms of these new variables, the Hamiltonian and momentum constraints can be written as

$$\mathcal{H} = \bar{\gamma}^{ij}\bar{D}_i\bar{D}_j e^{\phi} - \frac{e^{\phi}}{8}\bar{R} + \frac{e^{5\phi}}{8}\bar{A}_{ij}\bar{A}^{ij} - \frac{e^{5\phi}}{12}K^2 + 2\pi e^{5\phi}\rho = 0\ , \tag{2.82}$$

and

$$\mathcal{M}^i = \bar{D}_j\left(e^{6\phi}\bar{A}^{ji}\right) - \frac{2}{3}e^{6\phi}\bar{D}^i K - 8\pi e^{6\phi}S^i = 0\ . \tag{2.83}$$

respectively. In these equations $\bar{D}_i$ are covariant derivatives compatible with the conformal metric $\bar{\gamma}_{ij}$.

The evolution equations (2.70), (2.73), (2.75), (2.77), and (2.81), as well as the constraints (2.82) and (2.83) form a system of equations known as the *BSSN equations* [14–16].

With the BSSN equations, we are almost ready to solve Einstein's field equations. What we are missing now are gauge conditions that will allow us to determine the lapse function and the shift vector. As the field of numerical relativity evolved, different gauge conditions have been tested and the best ones became the standard for the field. The standard gauge condition for the lapse function is known as the "1+log" slicing condition [54], and is given by

$$\partial_t\alpha = \beta^j\partial_j\alpha - 2\alpha K\ . \tag{2.84}$$

The standard gauge condition for the shift vector is known as the Gamma-driver shift [55], and is given by the following system of equations

$$\partial_t \beta^i = \beta^j \partial_j \beta^i + \mu_S B^i \ , \tag{2.85}$$

$$\partial_t B^i = \beta^j \partial_j B^i + \partial_t \bar{\Gamma}^i - \beta^j \partial_j \bar{\Gamma}^i - \eta_S B^i \ , \tag{2.86}$$

where B^i is yet another auxiliary variable and $\mu_S = \frac{3}{4}$ and $M_{\rm ADM} < \eta_S < 2M_{\rm ADM}$ are constants. In practice, the term $\partial_t \bar{\Gamma}^i$ is replaced by the RHS of equation (2.81).

2.5 Brown's covariant BSSN formulation

The BSSN equations we have derived in the previous section are specialized to Cartesian coordinates. This is rather obvious since we impose the constraint (2.64), where η_{ij} is chosen to be the flat space metric in Cartesian coordinates. One can also immediately see that the Gamma-driver shift condition is not generally covariant, since β^i and $\partial_t \beta^i$ transform as contravariant vectors, but the advection term $\beta^j \partial_j \beta^i$ does not, spoiling the covariance of (2.85).

It is relevant to mention that the advection term is not always present in the Gamma-driver shift condition (see e.g. [63]), but even without the advection term we have problems. For example, the connection functions $\bar{\Gamma}^i$ *do not* transform as a contravariant vector, but instead as the derivative of a type $\left(\frac{0}{2}\right)$ tensor density of weight $-2/3$ — a transformation which we will not even bother writing down.

To fix this issue, we introduce a background metric, $\hat{\gamma}_{ij}$, which is independent of the coordinate time t, i.e. $\partial_t \hat{\gamma}_{ij} = 0$. Henceforth, all hatted quantities are associated with this background or *reference metric*.

The covariant metric can now be thought of as a deformation about the reference metric,

$$\bar{\gamma}_{ij} = \hat{\gamma}_{ij} + \epsilon_{ij} \ , \tag{2.87}$$

[10] This is a standard terminology to distinguish condition (2.89) from the second option given by Brown, which is given by

$$\partial_\perp \bar{\gamma} = (\partial_t - \mathcal{L}_\beta)\, \bar{\gamma} = 0 \ . \tag{2.88}$$

This second condition is referred to as the "Eulerian" condition [56].

where ϵ_{ij} is not necessarily small. The relation between the physical and covariant metric, however, remains the same and is given by equation (2.63).

We now modify condition (2.64) by imposing, instead, Brown's "Lagrangian"[10] condition [56]

$$\partial_t \bar{\gamma} = 0 \ . \tag{2.89}$$

This condition states that the determinant of the covariant metric remains equal to its initial value, which we still choose to be the determinant of the flat spacetime metric η_{ij}. The key difference here is that we *do not* restrict η_{ij} to be the flat space metric *in Cartesian coordinates*. We then set $\hat{\gamma}_{ij} = \eta_{ij}$, i.e. the reference metric is chosen to be the flat space metric in any particular coordinate system, and thus we have the constraint $\bar{\gamma} = \hat{\gamma}$.

The reason why a reference metric is introduced at all is perhaps more apparent when we consider the modification of the BSSN variable $\bar{\Gamma}^i$ in this new formalism. It is a well known result from differential geometry that Christoffel symbols do not transform covariantly between coordinate systems. However, the difference between two Christoffel symbols is, in fact, a tensor.[11] This means that instead of considering the Christoffel symbol $\bar{\Gamma}^i_{jk}$ to construct a new auxiliary variable, we can introduce the quantity

$$\Delta^i_{jk} \equiv \bar{\Gamma}^i_{jk} - \hat{\Gamma}^i_{jk} \ . \tag{2.90}$$

The indices of Δ^i_{jk} are raised and lowered using the conformal metric. Then, define the contraction

$$\Delta^i \equiv \bar{\gamma}^{jk} \Delta^i_{jk} \ , \tag{2.91}$$

which is guaranteed, by construction, to transform as a contravariant vector. Finally, we can define a new independent evolution variable, $\bar{\Lambda}^i$, which satisfies the initial constraint

$$\mathcal{C}^i \equiv \bar{\Lambda}^i - \Delta^i = 0 \ . \tag{2.92}$$

With these new considerations, and defining the notation

[11] See e.g. the discussion leading to equation (3.15) in chapter 3 of [64] for a pedagogical derivation of this result.

$$\partial_\perp \equiv \partial_t - \mathcal{L}_\beta \ , \tag{2.93}$$

where $\mathcal{L}_\beta$ is the Lie derivative along the shift vector β^i, Brown's "Lagrangian" formulation of the BSSN equations is given by

$$\partial_\perp \bar{\gamma}_{ij} = -\frac{2}{3}\bar{\gamma}_{ij}\bar{D}_k\beta^k - 2\alpha\bar{A}_{ij} \tag{2.94}$$

$$\begin{aligned}\partial_\perp \bar{A}_{ij} = &-\frac{2}{3}\bar{A}_{ij}\bar{D}_k\beta^k - 2\alpha\bar{A}_{ik}\bar{A}^k{}_j + \alpha\bar{A}_{ij}K \\ &+ e^{-4\phi}\Big[-2\alpha\bar{D}_i\bar{D}_j\phi + 4\alpha\bar{D}_i\phi\bar{D}_j\phi + 4\bar{D}_{(i}\alpha\bar{D}_{j)}\phi \\ &\qquad - \bar{D}_i\bar{D}_j\alpha + \alpha(\bar{R}_{ij} - 8\pi S_{ij})\Big]^{\mathrm{TF}}\end{aligned} \tag{2.95}$$

$$\partial_\perp \phi = \frac{1}{6}\bar{D}_k\beta^i - \frac{1}{6}\alpha K \tag{2.96}$$

$$\partial_\perp K = \frac{\alpha}{3}K^2 + \alpha\bar{A}_{ij}\bar{A}^{ij}e^{-4\phi}(\bar{D}^2\alpha + 2\bar{D}^i\alpha\bar{D}_i\phi) + 4\pi\alpha(\rho + S) \tag{2.97}$$

$$\begin{aligned}\partial_\perp \bar{\Lambda}^i = &\ \bar{\gamma}^{jk}\hat{D}_j\hat{D}_k\beta^i + \frac{2}{3}\Delta^i\bar{D}_j\beta^j + \frac{1}{3}\bar{D}^i\bar{D}_j\beta^j - \frac{4}{3}\alpha\bar{\gamma}^{ij}\partial_j K \\ &- 2\bar{A}^{jk}(\delta^i{}_j\partial_k\alpha - 6\alpha\delta^i{}_j\partial_k\phi - \alpha\Delta^i_{jk}) - 16\pi\alpha\bar{\gamma}^{ij}S_j.\end{aligned} \tag{2.98}$$

In the set of equations above, $\bar{D}_i$ are covariant derivatives compatible with the conformal metric $\bar{\gamma}_{ij}$, while $\hat{D}_i$ are covariant derivatives compatible with the reference metric $\hat{\gamma}_{ij}$. Note that Brown's original formulation of equations (2.94)–(2.98) did not include the source terms above, which we have included according to [65].

Brown also suggests covariant versions of the standard gauge conditions, namely[12]

$$\partial_t\alpha = \beta^j\bar{D}_j\alpha - 2\alpha K \ , \tag{2.99}$$

$$\partial_t\beta^i = \beta^j\bar{D}_j\beta^i + \frac{3}{4}B^i \ , \tag{2.100}$$

$$\partial_t B^i = \beta^j\bar{D}_j B^i + \left(\partial_t\bar{\Lambda}^i - \beta^j\bar{D}_j\bar{\Lambda}^i\right) - \eta_S B^i \ . \tag{2.101}$$

This new set of equations, including the gauge conditions, is known as the *moving puncture approach* [55, 66, 67]. We will use these equations, or close modifications of them, to study both the gravitational collapse of massless scalar fields, in chapter 3, and

[12]The "1+log" condition is not actually modified, since $\bar{D}_j\alpha = \partial_j\alpha$.

neutron stars, in chapter 4.

2.6 General relativistic magnetohydrodynamics

We will now provide a very brief introduction to the equations of general relativistic (ideal) magnetohydrodynamics (GRMHD). Einstein's equations in the presence of a matter source are given by equation (2.5). We will now consider the case where the energy-momentum tensor is composed of a perfect fluid and that of electromagnetism, for which we must then solve the Einstein-Maxwell equations. We will also assume the limit of infinite conductivity (ideal magnetohydrodynamics).

The first equation we consider, following the assumptions above, is the conservation of baryon number

$$\nabla_\mu \left(\rho_0 u^\mu\right) = 0 \ , \tag{2.102}$$

where ∇_μ is the covariant derivative compatible with $g_{\mu\nu}$, ρ_0 the fluid rest-mass density, and u^μ the fluid four-velocity. A second conservation equation is that of conservation of energy-momentum,

$$\nabla_\mu T^{\mu\nu} = 0 \ , \tag{2.103}$$

where $T_{\mu\nu} = T^{\rm matter}_{\mu\nu} + T^{\rm EM}_{\mu\nu}$ is the total energy-momentum tensor, composed of the perfect fluid, $T^{\rm matter}_{\mu\nu}$, and electromagnetic, $T^{\rm EM}_{\mu\nu}$, energy-momentum tensors in the ideal magnetohydrodynamics (MHD) limit ($u_\mu F^{\mu\nu} = 0$).

The dynamics of the eletromagnetic field are captured by the homogeneous Maxwell's equations,

$$\nabla_\nu F^{*\mu\nu} = \frac{1}{\sqrt{-g}} \partial_\nu \left(\sqrt{-g} F^{*\mu\nu}\right) = 0 \ , \tag{2.104}$$

where $F^{*\mu\nu} \equiv \frac{1}{2}\epsilon^{\mu\nu\rho\sigma} F_{\rho\sigma}$ is the dual of the Faraday tensor $F^{\mu\nu}$, with $\epsilon^{\mu\nu\rho\sigma}$ the totally antisymmetric Levi-Civita symbol. Coupled with the BSSN equations we have seen in the previous section, we have the full set of GRMHD equations.

Notice that Einstein's equations in their standard form, equation (2.5), were not suited for numerical integration. Analogously, the MHD equations (2.102), (2.103), and (2.104)

are not suitable for numerical integration. Instead, we write down these equations in *conservative form* (as in [68], for example)

$$\partial_t \vec{C} + \nabla_j \vec{F}^j = \vec{S} \ , \tag{2.105}$$

where $\vec{F}^j$ is the flux vector along the spatial direction x^j, $\vec{C}$ is the vector of conservative variables, and $\vec{S}$ is the vector of source terms. These vectors depend directly on the "primitive" variables of GRMHD, which are chosen to be

$$\vec{P} = \begin{bmatrix} \rho_0 \\ P \\ v^i \\ B^i \end{bmatrix} \ , \tag{2.106}$$

where P is the pressure, $v^i = \frac{u^i}{u^0}$ is the fluid three-velocity, and B^i are the spatial components of the magnetic field as measured by normal observers. We then make a choice of conservative variables which can be written in terms of these primitives as [68]

$$\vec{C} = \begin{bmatrix} \rho_\star \\ \tilde{\tau} \\ \tilde{S}_i \\ \tilde{B}^i \end{bmatrix} = \begin{bmatrix} \alpha\sqrt{\gamma}\rho_0 u^0 \\ \alpha^2\sqrt{\gamma}T^{00} - \rho_\star \\ \left(\rho_\star h + \alpha u^0 \sqrt{\gamma} b^2\right) u_i - \alpha\sqrt{\gamma} b^0 b_i \\ \sqrt{\gamma} B^i \end{bmatrix} \ , \tag{2.107}$$

where $h = 1 + \epsilon + \frac{P}{\rho_0}$ is the specific enthalpy, ϵ the specific internal energy, and $b^\mu = B^\mu_{(u)}/\sqrt{4\pi}$, with $B^\mu_{(u)}$ the magnetic field measured by an observer comoving with the fluid. The total energy-momentum tensor can be written as

$$T^{\mu\nu} = \left(\rho_0 h + b^2\right) u^\mu u^\nu + \left(P + \frac{b^2}{2}\right) g^{\mu\nu} - b^\mu b^\nu \ . \tag{2.108}$$

We will be using the code `IllinoisGRMHD` to solve the GRMHD equations [28]. In `IllinoisGRMHD`, the evolution equation for the conservative variables are

$$\partial_t \begin{bmatrix} \rho_\star \\ \tilde{\tau} \\ \tilde{S}_i \end{bmatrix} + \partial_j \begin{bmatrix} \rho_\star v^j \\ \alpha^2\sqrt{\gamma}T^{0j} - \rho_\star v^j \\ \alpha\sqrt{\gamma}T^j{}_i \end{bmatrix} = \begin{bmatrix} 0 \\ s \\ \frac{1}{2}\alpha\sqrt{\gamma}T^{\alpha\beta}\partial_i g_{\alpha\beta} \end{bmatrix} \ , \tag{2.109}$$

where

$$s = \alpha\sqrt{\gamma}\left[\left(T^{00}\beta^i\beta^j + 2T^{0i}\beta^j + T^{ij}\right)K_{ij} - \left(T^{00}\beta^i + T^{0i}\right)\partial_i\alpha\right] \ . \tag{2.110}$$

The remaining evolution equation is the magnetic induction equation, which in conservative form may be written as

$$\partial_t\tilde{B}^i + \partial_j\left(v^j\tilde{B}^i - v^i\tilde{B}^j\right) = 0 \ . \tag{2.111}$$

However, if this equation is used as above, numerical truncation errors may cause the violation of the no-monopoles constraint

$$\partial_i\tilde{B}^i = 0 \ . \tag{2.112}$$

To remedy this, we introduce a magnetic 4-vector potential [69, 70],

$$\mathcal{A}_\mu = n_\mu\Phi + A_\mu \ . \tag{2.113}$$

The magnetic field can be written in terms of the vector potential A_i as

$$\tilde{B}^i = \epsilon^{ijk}\partial_j A_k \ . \tag{2.114}$$

We use special finite difference operators in this last equation, which guarantee that condition (2.112) is satisfied to roundoff error.

We now want to rewrite the induction equation in terms of $\mathcal{A}_\mu$. First, note that A_μ is purely spatial, i.e. $n^\mu A_\mu = 0$, which imply that

$$n^\mu A_\mu = 0 = n^t A_t + n^i A_i = \frac{1}{\alpha}\left(A_t - \beta^i A_i\right) \implies A_t = \beta^i A_i \ . \tag{2.115}$$

We now need the Faraday tensor (cf. equation (28) in [71])

$$F_{\mu\nu} = \partial_\mu\mathcal{A}_\nu - \partial_\nu\mathcal{A}_\mu = n_\mu E_\nu - n_\nu E_\mu + \epsilon_{\mu\nu\sigma}\tilde{B}^\sigma \ , \tag{2.116}$$

and the ideal MHD condition, $u_\mu F^{\mu\nu} = 0$, which can be written as (see equation (46) in [71])

$$\alpha E_i = \epsilon_{ijk}\left(v^j + \beta^j\right)\tilde{B}^k \ . \tag{2.117}$$

Now consider equation (2.116) with $\mu = t$ and $\nu = i$,

$$F_{ti} = \partial_t \mathcal{A}_i - \partial_j \mathcal{A}_t = n_t E_i - n_i E_t + \epsilon_{tij}\tilde{B}^j \ . \tag{2.118}$$

Furthermore, consider the identities

$$\mathcal{A}_t = n_t \Phi + A_t = -\alpha\Phi + \beta^j A_j \ , \tag{2.119}$$

$$\mathcal{A}_i = n_i \Phi + A_i = A_i \ , \tag{2.120}$$

$$n_t E_i = -\alpha E_i = -\epsilon_{ijk}\left(v^j + \beta^j\right)\tilde{B}^k \ , \tag{2.121}$$

$$n_i E_t = 0 \ , \tag{2.122}$$

$$\epsilon_{tij} B^j = \epsilon_{ijk}\beta^j \tilde{B}^k \ , \tag{2.123}$$

where in the first equation we have used relation (2.115), in the third one we have used (2.117), and in the last one we used

$$\begin{aligned}
\epsilon_{tij} &= n^\mu \epsilon_{\mu tij} \\
&= n^t \epsilon_{ttij} + n^k \epsilon_{ktij} \\
&= -\frac{\beta^k}{\alpha}\epsilon_{ktij} \\
&= +\frac{\beta^k}{\alpha}\epsilon_{tijk} \\
&= +\beta^k n^\mu \epsilon_{\mu ijk} \\
&= +\beta^k \epsilon_{ijk} \ ,
\end{aligned} \tag{2.124}$$

where we use the projection $n^\lambda \epsilon_{\lambda\mu\nu\sigma} = \epsilon_{\mu\nu\sigma}$ and, in the fifth equality we have used the identity

$$\begin{aligned}\beta^k n^\mu \epsilon_{\mu ijk} &= \beta^k \left(n^t \epsilon_{tijk} + n^m \epsilon_{mijk} \right) \\ &= \beta^k \left(\alpha^{-1} \epsilon_{tijk} - \alpha^{-1} \beta^m \epsilon_{mijk} \right) \\ &= \frac{\beta^k}{\alpha} \epsilon_{tijk} - \frac{1}{\alpha} \beta^m \beta^k \epsilon_{mijk} \\ &= \frac{\beta^k}{\alpha} \epsilon_{tijk} \ , \end{aligned} \tag{2.125}$$

since $\beta^m \beta^k$ is symmetric under $m \leftrightarrow k$ but ϵ_{mijk} is antisymmetric. Then, returning to equation (2.118), but considering all these new identities, we have

$$\partial_t A_i - \partial_i \left(-\alpha \Phi + \beta^j A_j \right) = -\epsilon_{ijk} \left(v^j + \beta^j \right) \tilde{B}^k + \epsilon_{ijk} \beta^j \tilde{B}^k \ , \tag{2.126}$$

which yields the evolution equation

$$\partial_t A_i = \epsilon_{ijk} v^j \tilde{B}^k - \partial_i \left(\alpha \Phi - \beta^j A_j \right) \ . \tag{2.127}$$

We are now left with the task of choosing a gauge condition. `IllinoisGRMHD` adopts the generalized Lorenz gauge conditon [72], which yields the evolution equation

$$\partial_t \left[\sqrt{\gamma} \Phi \right] + \partial_j \left(\alpha \sqrt{\gamma} A^j - \beta^j \left[\sqrt{\gamma} \Phi \right] \right) = -\xi \alpha \left[\sqrt{\gamma} \Phi \right] \ , \tag{2.128}$$

for the scalar potential Φ. In this last equation, ξ is a parameter with dimensions of inverse length, chosen so that the Courant-Friedrichs-Lewy (CFL) condition (see appendix D) remains satisfied, and we note that the evolved variable is $\left[\sqrt{\gamma} \Phi \right]$.

Finally, to close the system of equations, we must specify an equation of state (EOS), which relates the pressure to other primitive variables, typically the fluid rest-mass density, ρ_0. More sophisticated equations of state may also include other dependencies, for example on the temperature T or on the electron fraction Y_e. We will talk more about equations of state when we discuss neutron stars in chapter 4.

This page intentionally left blank.

Part II
Research work

This page intentionally left blank.

Chapter 3

Gravitational collapse of massless scalar fields

In 1993, Matt Choptuik published a seminal paper announcing the discovery of critical phenomena in his studies of a massless scalar field propagating in a dynamical spacetime [20]. Choptuik found solutions of the Einstein-Klein-Gordon (EKG) equations, $\mathcal{S}[\eta]$, which depended on a parameter η. He found a *critical* value for this parameter, η_*, such that solutions with $\eta > \eta_*$ were comprised of spacetimes where the scalar field collapsed and formed a black hole, while solutions with $\eta < \eta_*$ were characterized by full dispersion of the scalar field, resulting in flat space as the end state.

By performing several *supercritical* runs (i.e. with $\eta > \eta_*$), Choptuik conjectured that the masses of the black holes formed near criticality followed the power law

$$M_{\rm BH} \propto |\eta - \eta_*|^{\gamma} \ , \tag{3.1}$$

where γ, the critical exponent, is *universal*.[1] The sense in which the critical exponent is said to be universal is the following: given a particular matter source, in this case a massless scalar field, the numerical value of the critical exponent is the same for all possible initial conditions. In his paper, Choptuik performed runs with four different initial data families and they all resulted in the same numerical value $\gamma \approx 0.37$.

Furthermore, there is a periodicity on the behaviour of the scalar field and metric quantities which is observed on ever smaller spatiotemporal scales. This periodicity is

[1]Note that γ here is just a number and *not* the determinant of the physical metric γ_{ij}. This is a standard terminology and we hope there is no confusion.

better observed in terms of logarithmic variables, such as the logarithmic proper time (to be defined below). Choptuik found, for example, that given a value of the logarithmic proper time, Λ, the same behaviour of the scalar field is observed at $\Lambda+\Delta$, with $\Delta = 3.43$, but on a scale which is $e^{\Delta} \approx 30$ times smaller than the first one. The reappearance of the solution at logarithmic time intervals Δ is known as *echoing*, and Δ is the echoing period, which is also universal.

After Choptuik's discovery, many other studies were performed, considering, for example, different matter sources and spacetime dimensions (we recommend [73, 74] for good reviews). Choptuik's results have also been confirmed by other, independent studies. Using perturbation theory, the universal exponent has been found to be $\gamma \approx 0.374$ [75, 76], while the echoing period has been shown to be $\Delta = 3.445452402(3)$ using semi-analytical methods [77].

The power law behaviour discovered by Choptuik has also been modified to [75, 78]

$$\ln M_{\rm BH} = C + \gamma \ln|\eta - \eta_*| + f\left(\ln|\eta - \eta_*|\right) \ , \tag{3.2}$$

where $f(x)$ is a function which is periodic in x with angular frequency

$$\omega = \frac{4\pi\gamma}{\Delta} \ . \tag{3.3}$$

Given the generality of the arguments used to show the behaviour (3.2) for the black hole mass near criticality, a quantity $\mathcal{A}$ with dimension of length α can be inferred to satisfy

$$\mathcal{A} \propto |\eta - \eta_*|^{\alpha\gamma} \ , \tag{3.4}$$

and, therefore,

$$\ln \mathcal{A} = C + \alpha\gamma \ln|\eta - \eta_*| + f\left(\ln|\eta - \eta_*|\right) \ . \tag{3.5}$$

It is also relevant to mention that Choptuik's original discovery made use of a numerical code that solved the EKG equations using the ADM formalism. However, similar results have also been found using the BSSN formalism (e.g. in [79, 80]), and, in particular, cases where the spherical symmetry condition is relaxed have been studied in [80].

In this chapter we report our results on the gravitational collapse of massless scalar

fields. We studied this problem using both the ADM and BSSN formalisms, and our results are in excellent agreement with the literature. In section 3.1 we will talk about solving the EKG equations using the ADM formalism and the polar/radial gauge. We will also talk about the development of our own numerical code to solve these equations. In section 3.2 we will talk about solving the EKG equations using the BSSN formalism. We note that we discuss the Klein-Gordon equations separately for each formalism, since some auxiliary variables are defined in slightly different ways. In section 3.3 we report preliminary results that characterize subcritical ($\eta < \eta_*$) and supercritical ($\eta > \eta_*$) runs, and in section 3.4 we discuss critical phenomena.

3.1 The ADM approach

The gravitational collapse of a massless scalar field is studied by minimally coupling it to gravity, which is described by the action

$$S = \int d^4x\sqrt{-g}\left[\frac{1}{16\pi}R - \frac{1}{2}g^{\mu\nu}\partial_\mu\psi\partial_\nu\psi\right] \ , \tag{3.6}$$

where $g^{\mu\nu}$ is the inverse of the physical 4-metric $g_{\mu\nu}$, g is the determinant of $g_{\mu\nu}$, and ψ is a massless scalar field. The energy-momentum tensor for a massless scalar field is given by

$$T^{\psi}_{\mu\nu} = -\frac{2}{\sqrt{-g}}\frac{\delta\mathcal{L}_\psi}{\delta g^{\mu\nu}} = \partial_\mu\psi\partial_\nu\psi - \frac{1}{2}g_{\mu\nu}\partial_\alpha\psi\partial^\alpha\psi \ , \tag{3.7}$$

where $\mathcal{L}_\psi = \frac{\sqrt{-g}}{2}g^{\mu\nu}\partial_\mu\psi\partial_\nu\psi$ is the Lagrangian density of a massless scalar field.

We adopt the metric in the polar/radial gauge, equation (2.62). This means that our metric variables are $(\alpha, a, K^r{}_r)$, with the shift vector and all other components of the extrinsic curvature being zero. With this metric choice, the Klein-Gordon equation, which is obtained by varying action (3.6) with respect to ψ, reads

$$\partial_t\left(\frac{a}{\alpha}\partial_t\psi\right) = \frac{1}{r^2}\partial_r\left(r^2\frac{\alpha}{a}\partial_r\psi\right) \ . \tag{3.8}$$

At this point, we introduce the two auxiliary fields

$$\Psi\left(t,r\right) \equiv \partial_r\psi\left(t,r\right) \ , \ \Pi\left(t,r\right) \equiv \frac{a\left(t,r\right)}{\alpha\left(t,r\right)}\partial_t\psi\left(t,r\right) \ , \tag{3.9}$$

so that the Klein-Gordon equation can be traded by two first-order (in time and space) partial differential equations

$$\partial_t \Psi = \partial_r \left(\frac{\alpha}{a} \Pi \right) \ , \tag{3.10}$$

$$\partial_t \Pi = \frac{1}{r^2} \partial_r \left(r^2 \frac{\alpha}{a} \Psi \right) \ . \tag{3.11}$$

The energy density, $\rho = n^\mu n^\nu T^\psi_{\mu\nu} = \alpha^2 T^\psi_{tt}$, is given by

$$\rho = \frac{\Psi^2 + \Pi^2}{2a^2} \ . \tag{3.12}$$

In this coordinate system, $K = K^r{}_r$, and therefore the Hamiltonian constraint is reduced to

$$\mathcal{H} = {}^{(3)}R - 16\pi\rho \ . \tag{3.13}$$

Our gauge choice implies that the spatial Ricci scalar associated γ_{ij} is given by

$${}^{(3)}R = \frac{4}{ra^2} \left(\frac{\partial_r a}{a} + \frac{a^2 - 1}{2r} \right) \ , \tag{3.14}$$

which finally imply that the Hamiltonian constraint is given by

$$\mathcal{H} = \frac{4}{ra^2} \left(\frac{\partial_r a}{a} + \frac{a^2 - 1}{2r} \right) - 8\pi \left(\frac{\Psi^2 + \Pi^2}{a^2} \right) \ . \tag{3.15}$$

Since we know that $\mathcal{H} = 0$ *analytically*, we can write the equation above as a constraint that must be satisfied at each hypersurface, namely

$$\frac{\partial_r a}{a} + \frac{a^2 - 1}{2r} = 2\pi r \left(\Psi^2 + \Pi^2 \right) \ . \tag{3.16}$$

The last equation used by Choptuik follows from the *polar slicing condition* $K^\theta{}_\theta = 0 = K^\varphi{}_\varphi$ for *all* times. This implies that $\partial_t K_{\theta\theta} = 0$, which in turn implies the relation

$$\partial_t K_{\theta\theta} = -D_\theta D_\theta \alpha + \alpha \, {}^{(3)}R_{\theta\theta} + 4\pi\alpha \left[r^2 \left(S - \rho \right) - 2 S_{\theta\theta} \right] = 0 \ , \tag{3.17}$$

[2]The following identities are useful to obtain this relation:

$$S_{ii} = T^\phi_{ii} \ , \ S = \frac{\left(3\Pi^2 - \Phi^2 \right)}{2a^2} \ , \ {}^{(3)}R_{\theta\theta} = \frac{r}{a^3} \partial_r a + \frac{a^2 - 1}{a^2} \ , \ D_\theta D_\theta \alpha = \frac{r}{a^2} \partial_r \alpha \ .$$

leading to the constraint equation[2]

$$\frac{\partial_r \alpha}{\alpha} - \frac{\partial_r a}{a} - \frac{a^2 - 1}{r} = 0 \ . \tag{3.18}$$

Equations (3.10), (3.11), (3.16), and (3.18) are equations (3)-(5) in Choptuik's original paper [20]. Notice that in this approach we are explicitly evolving the massless scalar field in time using the Klein-Gordon equations, but α and a are "evolved" by finding consistent values of them that satisfy the conservation of energy and polar slicing condition. This evolution scheme is referred to as being *fully constrained*, since all ADM variables are "evolved" using constraint equations. Notice, also, that we did not present an evolution equation for $K^r{}_r$. This is because, in this gauge, the equation that $K^r{}_r$ must satisfy is algebraic (i.e. not a differential equation), and thus it does not need to be evolved.

One disadvantage of choosing this fully constrained approach is that equations (3.16) and (3.18) are *elliptic* (see appendix A). Solving elliptic equations is, generally, far more costly from a computational point of view. Moreover, equation (3.16) is also non-linear, requiring a Newton-Raphson solver to obtain the values of a. This makes the overall integration algorithm a bit awkward, since we are not able to simply implement, say, a standard fourth-order Runge-Kutta time stepper.

However, this choice is not an accident. As we have discussed in chapter 2, the ADM equations are only weakly hyperbolic (see also appendix A), and using them to explicitly evolve γ_{ij} and K_{ij} in time will lead to instabilities that we cannot remove. Thus we opt for a less efficient, but stable, numerical evolution scheme.

3.1.1 The `SFcollapse1D` code

We solve the EKG equations, in the ADM formalism, using the `SFcollapse1D` code [19]. `SFcollapse1D` has been developed using the `C++` programming language, using best programming practices. It is an open-sourced [24], well-documented, and user friendly code. This code has been specially developed for solving the EKG equations using the ADM formalism in the polar/radial gauge. Therefore, it is highly optimized at solving this particular system of equations, but *only* this system. Changing the problem even slightly, by considering a different gauge choice or changing the matter content, would require

rewriting, or making major modifications, to `SFcollapse1D`. This means it is not an ideal code for us to use moving forward, and we refer the reader to the next section for more on this subject.

A typical run of the `SFcollapse1D` code starts by specifying an initial condition for the scalar field, ψ. This usually means setting $\psi(0, r)$, which is then translated to initial conditions that the auxiliary fields must satisfy. Using the initial values for Ψ and Π, the code then solves the Hamiltonian constraint (3.16) and polar slicing condition (3.18) to obtain consistent values of a and α, respectively, that satisfy Einstein's equations on the initial hypersurface.

Since our integration scheme uses centered, second-order accurate finite differences, it is a "three time levels" scheme. This means that specifying (Ψ, Π, a, α) at $t = 0$ is not enough to perform a "standard" integration using the scheme. To overcome this, Ψ and Π are integrated to $t = \Delta t$ using a second-order accurate Runge-Kutta method, after which a and α are computed. This is only done on the very first integration step. After obtaining all the fields at $t = 0$ and $t = \Delta t$, a set of values we refer to as *initial data*, the code evolves them in time using the procedure described below.

`SFcollapse1D` solves the EKG equations by using centered, second-order accurate finite differences to discretize equations (3.10), (3.11), (3.16), and (3.18) (see appendix E for more information on the discretization scheme). The code then evolves the scalar field auxiliary variables, Ψ and Π, explicitly in time using equations (3.10) and (3.11), respectively. The Courant–Friedrichs–Lewy factor[3] used is 0.5. After having Ψ and Π on the next hypersurface, the procedure to obtain the initial data is repeated to determine a and α. The Hamiltonian constraint is solved by introducing a new variable

$$A \equiv \ln a \ . \tag{3.19}$$

A Newton-Raphson solver is then used to determine the value of A on the hypersurface, from which a is computed via $a = e^A$. The polar slicing condition is simpler to solve, resulting in a linear equation for the lapse function in terms of the radial metric.

Throughout the evolution, we impose Neumann boundary conditions for the scalar field variables at the origin,

[3] See appendix D.

$$\Psi(t,0) = 0 \ , \tag{3.20}$$

$$\partial_r \Pi(t,0) = 0 \ , \tag{3.21}$$

and outgoing radiation boundary conditions at the outer boundary,

$$\Big(\partial_t \left[r\psi(t,r)\right] + \partial_r \left[r\psi(t,r)\right]\Big)\Big|_{r=r_{\max}} = 0 \ , \tag{3.22}$$

which is applied to the scalar field ψ itself. Then, values of Ψ and Π which are consistent with this boundary condition are computed from ψ. The discretization of this equation, in spherical coordinates, reads

$$\psi_J^{n+1} = \left(\frac{3}{\Delta t} + \frac{2}{r_J} + \frac{3}{\Delta r}\right)^{-1} \left[\frac{4\psi_J^n - \psi_J^{n-1}}{\Delta t} + \frac{4\psi_{J-1}^{n+1} - \psi_{J-2}^{n-1}}{\Delta r}\right] \ , \tag{3.23}$$

where the index J corresponds to $r_{\max}$. Note that this is also a $\mathcal{O}\left(\Delta r^2 + \Delta t^2\right)$ approximation.

To solve the elliptic equations that determine the radial metric and lapse functions iteratively, we must provide their values at the inner boundary. The boundary condition for a follows from the elementary flatness condition discussed in section 2.3.5,

$$a(t,0) = 1 \ . \tag{3.24}$$

For the lapse function we set its value to unity at the origin,

$$\alpha(t,0) = 1 \ , \tag{3.25}$$

which is a regularity condition that also sets the coordinate time at the origin is equal to the proper time. We also perform a so-called *rescaling of the lapse function* [81], which multiplies the lapse on the entire hypersurface by

$$\kappa = \min_{0 \le j \le J} \left(\frac{a_j}{\alpha_j}\right) \ . \tag{3.26}$$

Note that the polar slicing condition is homogeneous in α, so multiplying α by a constant means that the condition is still satisfied. This procedure ensures that no signals propagate

faster than the speed of light, by typically setting $\alpha = a^{-1}$ at the outer boundary [81].

Some extra features of the `SFcollapse1D` code are listed below, while algorithm 3.1 shows how `SFcollapse1D` solves the EKG equations in pseudocode form.

- **Collapse of the lapse**: the code periodically checks whether or not the lapse function α has dropped below a user-specified threshold ϵ_α. When the criterion $\alpha < \epsilon_\alpha$ is met, the code interupts the time integratiion and characterizes the run as being in the strong field regime. We are able to infer this because around the singularity we have $\alpha \to 0$, which is known as the *freezing of the lapse* or *collapse of the lapse*.

- **NaN checker**: the code also periodically checks whether or not any of the evolved functions, $(\psi, \Psi, \Pi, a, \alpha)$, have become NaN, ending the run if that is the case.

- **Run information**: integration progress is constantly printed so that the user is able to track how much of the run has already been completed. The program also shows the time elapsed during the run and estimates the amount of time required to complete the run.

- **Central values**: *Central values* (i.e. values at $r = 0$) of certain quantities, such as ψ, α, and ρ, the energy density, are periodically outputted so that we can study critical phenomena.

```
1  begin Initial condition
2      Set the initial condition for ψ_j^0, Ψ_j^0, and Π_j^0;
3      Impose inner boundary conditions to Ψ_0^0, a_0^0, and α_0^0;
4      Solve the Hamiltonian constraint to determine a_j^0;
5      Solve the polar slicing condition to determine α_j^0;
6      Perform the rescaling of the lapse function;
7  begin Initial data
8      Apply inner boundary conditions to set Ψ_0^{1/2}, a_0^{1/2}, and α_0^{1/2};
9      for j = 1 to N_r − 1 do
10         if j = 1 then start by computing ψ_0^{1/2};
11         Perform a half-step integration, compute ψ_j^{1/2}, Ψ_j^{1/2}, and Π_j^{1/2};
12     Apply outgoing radiation boundary conditions to ψ_{N_r}^{1/2};
13     Apply inner boundary conditions to find Π_0^{1/2};
14     Compute Ψ_{N_r}^{1/2} and Π_{N_r}^{1/2};
15     for j = 0 to N_r do
16         Solve the Hamiltonian constraint to determine a_j^{1/2};
17         Solve the polar slicing condition to determine α_j^{1/2};
18     Perform the rescaling of the lapse function;
19     Apply inner boundary conditions to set Ψ_0^1, a_0^1, and α_0^1;
20     for j = 1 to N_r − 1 do
21         if j = 1 then start by computing ψ_0^1;
22         Perform a single-step integration, compute ψ_j^1, Ψ_j^1, and Π_j^1;
23     Apply outgoing radiation boundary conditions to ψ_{N_r}^1;
24     Apply inner boundary conditions to find Π_0^1;
25     Compute Ψ_{N_r}^1 and Π_{N_r}^1;
26     for j = 0 to N_r do
27         Solve the Hamiltonian constraint to determine a_j^1;
28         Solve the polar slicing condition to determine α_j^1;
29     Perform the rescaling of the lapse function;
30 begin Evolution
31     while t < t_final do
32         Apply inner boundary conditions to set Ψ_0^n, a_0^n, and α_0^n;
33         for j = 1 to N_r − 1 do
34             if j = 1 then start by computing ψ_0^n;
35             Compute ψ_j^n, Ψ_j^n, and Π_j^n;
36         Apply outgoing radiation boundary conditions to ψ_{N_r}^n;
37         Apply inner boundary conditions to find Π_0^n;
38         Compute Ψ_{N_r}^n and Π_{N_r}^n;
39         for j = 1 to N_r do
40             Solve the Hamiltonian constraint to determine a_j^n;
41             Solve the polar slicing condition to determine α_j^n;
42         Perform the rescaling of the lapse function;
```

Algorithm 3.1: The `SFcollapse1D` program's pseudocode.

3.2 The BSSN approach

In the BSSN formalism, we will need a more general form of the EKG equations, since we do not assume γ_{ij} to be in a specific gauge. We then start over from the Klein-Gordon equation,

$$\nabla^\mu \left(\nabla_\mu \psi\right) = 0 \ . \tag{3.27}$$

The auxiliary variable Π from the previous section is now defined as

$$\Pi \equiv -\frac{1}{\alpha}\left(\partial_t \psi - \beta^i \partial_i \psi\right) \ . \tag{3.28}$$

Note that this definition is not equivalent to the previous one. We then use Π to write the Klein-Gordon equation as two first order (in time) partial differential equations,

$$\begin{aligned} \partial_t \psi &= \beta^i \partial_i \psi - \alpha\, \Pi \ , \\ \partial_t \Pi &= \beta^i \partial_i \Pi + \alpha K\, \Pi - \gamma^{ij}\left(\partial_j \psi\, \partial_i \alpha - \alpha\, \Gamma^k{}_{ij}\, \partial_k \psi + \alpha\, \partial_i \partial_j \psi\right) \ . \end{aligned} \tag{3.29}$$

We now must write these equations in terms of the BSSN variables,

$$\left(\bar{\gamma}_{ij}, \bar{A}_{ij}, K, \phi, \bar{\Lambda}^i, \alpha, \beta^i, B^i\right) \ . \tag{3.30}$$

Let us consider the identity

$$\Gamma^k{}_{ij} = \bar{\Gamma}^k{}_{ij} + 2\left(\delta^k{}_i \bar{D}_j \phi + \delta^k{}_j \bar{D}_i \phi - \bar{\gamma}_{ij} \bar{\gamma}^{k\ell} \bar{D}_\ell \phi\right) \ . \tag{3.31}$$

Then, consider the term that contains $\Gamma^k{}_{ij}$ on the right-hand side of $\partial_t \Pi$:

$$\alpha\, \gamma^{ij} \Gamma^k{}_{ij}\, \partial_k \psi = \alpha e^{-4\phi} \bar{\gamma}^{ij} \left[\bar{\Gamma}^k{}_{ij} + 2\left(\delta^k{}_i \bar{D}_j \phi + \delta^k{}_j \bar{D}_i \phi - \bar{\gamma}_{ij} \bar{\gamma}^{k\ell} \bar{D}_\ell \phi\right)\right] \partial_k \psi \ . \tag{3.32}$$

Focusing on the term in parenthesis, we have (ignoring, for now, a few non-essential multiplicative terms and replacing $\bar{D}_i \phi = \partial_i \phi$)

$$\begin{aligned} 2\bar{\gamma}^{ij}\left(\delta^k{}_i \bar{D}_j \phi + \delta^k{}_j \bar{D}_i \phi - \bar{\gamma}_{ij} \bar{\gamma}^{k\ell} \bar{D}_\ell \phi\right) \partial_k \psi &= 2\left(\bar{\gamma}^{kj} \partial_j \phi + \bar{\gamma}^{ki} \partial_i \phi - 3\bar{\gamma}^{k\ell} \partial_\ell \phi\right) \partial_k \psi \\ &= 2\left(\bar{\gamma}^{ij} \partial_i \phi + \bar{\gamma}^{ij} \partial_i \phi - 3\bar{\gamma}^{ij} \partial_i \phi\right) \partial_j \psi \\ &= -2\bar{\gamma}^{ij} \partial_j \psi \partial_i \phi \ , \end{aligned} \tag{3.33}$$

so that

$$\alpha\,\gamma^{ij}\Gamma^{k}_{\ ij}\,\partial_k\psi = e^{-4\phi}\bar{\gamma}^{ij}\left(\alpha\bar{\Gamma}^{k}_{\ ij}\,\partial_k\psi - 2\alpha\partial_j\psi\partial_i\phi\right) \tag{3.34}$$

For the rest of the equation, all we need to do is replace $\gamma^{ij} \to e^{-4\phi}\bar{\gamma}^{ij}$, so that the Klein-Gordon equation becomes

$$\partial_t\Pi = \beta^i\partial_i\Pi + \alpha K\,\Pi - e^{-4\phi}\bar{\gamma}^{ij}\left(\partial_j\psi\,\partial_i\alpha - \alpha\,\bar{\Gamma}^{k}_{\ ij}\,\partial_k\psi + \alpha\,\partial_i\partial_j\psi + 2\alpha\,\partial_j\psi\,\partial_i\phi\right)\ . \tag{3.35}$$

Note that the evolution equation for ψ is left unchanged,

$$\partial_t\psi = \beta^i\partial_i\psi - \alpha\,\Pi\ . \tag{3.36}$$

3.2.1 The NRPy+ infrastructure

Coding up the BSSN equations by hand is extremely difficult. Not only it is a laborous task, but prone to human error. On top of that, the code must be extensively debugged in order to be trustworthy, and, unless you have a personal supercomputer, you must ensure the code is efficient. Equally, or perhaps even more, important is to make sure your code is well-documented, so that other users, students in particular, can start use your code with minimal difficulty.

To minimize user error and maximize code performance, we opted to use the NRPy+ infrastructure [25, 26]. NRPy+ ("Python-based code generation for numerical relativity and beyond") which is an open-source [24, 27], permissively licensed package for generating highly optimized numerical relativity codes in singular, curvilinear coordinate systems like spherical coordinates. NRPy+ is written entirely in Python[4] and depends only on the standard Python computer algebra package SymPy [82] for symbolic algebra. Using NRPy+, outputting Einstein's equations into any spherical-like, Cartesian-like, or cylindrical-like coordinate system as highly optimized C code is both well-documented and intuitive to new users.

Although we were not involved in the development of the initial infrastructure, we are contributing to its development on a daily basis. Implementation and debugging of

[4]Both Python 2.7+ and Python 3.0+ are supported.

source terms to the BSSN equations and the Klein-Gordon equations were some of our contributions to the infrastructure. We have also made several contributions to the study of neutron stars, but we will discuss that in the next chapter.

To show how easy it is to implement the BSSN equations, we will give an example by implementing the "1+log" lapse condition, equation (2.84), which we repeat here

$$\partial_t \alpha = \beta^j \partial_j \alpha - 2\alpha K \ . \tag{3.37}$$

We note that advective terms of the form $\beta^j \partial_j \alpha$ are handled using "upwinded" finite difference stencils (see the footnote in section 3.4.2 for more details), instead of centered one. This is reflected in the implementation below.

```
# Part 0: Load all needed Python and NRPy+ packages
import sympy as sp
import indexedexp as ixp
import BSSN.BSSN_quantities as Bq
# Part 1: Define all BSSN variables,
# i.e. (γ̄_ij, Ā_ij, K, φ, Λ̄^i, α, β^i, B^i)
Bq.BSSN_basic_tensors()
# Part 2: Define local variables to unclutter the notation
# Part 2.a: We need the lapse
alpha = Bq.alpha
# Part 2.b: The trace of the extrinsic curvature
trK = Bq.trK
# Part 2.c: The shift vector
betaU = Bq.betaU
# Part 2.d: And the derivative of the lapse
alpha_
```

Listing 3.1: Implementation of the "1+log" lapse condition in `NRPy+`.

```
dupD = ixp.declarerank1(''alpha_dupD'')
# Part 3: Implement the ''1+log'' lapse condition.
# Part 3.a: Initialize the rhs to the term  −2αK
alpha_rhs = - 2 * alpha * trK
```

```
# Part 3.b: Add the advection term β^j ∂_j α
for j in range(DIM):
    alpha_rhs += betaU[j] * alpha_dupD[j]
```

Listing 3.2: Implementation of the "1+log" lapse condition in `NRPy+`.

Commenting on the code above, tensorial quantities in `NRPy+` are stored as lists of `SymPy` symbols or expressions. Some of these quantities, such as the lapse function α, actually correspond to the `SymPy` symbol "alpha". Other quantities, for example the physical metric γ_{ij}, are placeholder variables, defined in terms of other symbols that will appear in the final `C` code. Tensor quantities, regardless of whether or not they are defined in terms of more fundamental tensors, are appended with "U"'s or "D"'s to indicate they have "up" or "down" indices, respectively. For example,

$$\begin{aligned}
\alpha &\leftrightarrow \text{alpha} \ , \\
\beta^i &\leftrightarrow \text{betaU[i]} \ , \\
\gamma_{ij} &\leftrightarrow \text{gammaDD[i][j]} \ , \\
\Gamma^k_{ij} &\leftrightarrow \text{GammaUDD[k][i][j]} \ .
\end{aligned}$$

Derivatives of a quantity are appended with a "_d" to indicate that it is a derivative. Thus, for example,

$$\begin{aligned}
\partial_i \alpha &\leftrightarrow \text{alpha_dD[i]} \ , \\
\partial_j \beta^i &\leftrightarrow \text{betaU_dD[i][j]} \ , \\
\partial_k \partial_l \gamma_{ij} &\leftrightarrow \text{gammaDD_dDD[i][j][k][l]} \ , \\
\partial_m \partial_n \Gamma^k_{ij} &\leftrightarrow \text{GammaUDD_dDD[k][i][j][m][n]} \ .
\end{aligned}$$

`NRPy+` implements Brown's covariant formulation of the BSSN equations in "Lagrangian" form, as described in section 2.5. The `C` code generated by `NRPy+` has been validade against trusted implementations of the BSSN equations and shown to produce roundoff error agreement.

3.3 The different field regimes

In this section we present results from different runs in the so-called *weak* and *strong* field regimes obtained using the `SFcollapse1D` code. Note that the terminology "subcritical" run corresponds to runs that lie in the weak field regime, while "supercritical" runs belong to the strong field regime. We emphasize here that the results presented in this section were not obtained with accuracy and stability in mind, but simply to illustrate the different regimes. This means that our grid choices are not meant to be used during production runs, since they underesolve the structures that emerge once the scalar field propagates and interacts with the underlying geometry.

We present here results obtained using the initial condition

$$\psi(0,r) = \eta e^{-r^2} \ . \tag{3.38}$$

Although we are evolving the set of variables $(\psi, \Psi, \Pi, a, \alpha)$, we will be interested in looking only at the scalar field $\psi(t,r)$, the lapse function $\alpha(t,r)$, and the *mass-aspect function*, $M(t,r)$, which is defined in analogy to the Schwarzschild metric

$$a^2(t,r) \equiv \frac{1}{1-\frac{2M(t,r)}{r}} \implies M(t,r) = \frac{r}{2}\left[1 - \frac{1}{a^2(t,r)}\right] \ . \tag{3.39}$$

The mass-aspect function computes the amount of mass/energy inside a sphere of radius r. In the limit $r \to \infty$, we have $M \to M_{\rm ADM}$, the ADM mass.

Another important aspect of the code is the use of a different coordinate system, which we will refer to as "SinhSpherical" coordinates.[5] In this coordinate system, we make the coordinate change

$$r = r_{\rm max}\frac{\sinh\left(\frac{x}{w}\right)}{\sinh\left(\frac{1}{w}\right)} \ , \tag{3.40}$$

where $r_{\rm max}$ represents the outer boundary of the computational domain, w is a free parameter that controls the sampling density around $r \sim 0$, and $x \in [0,1]$ is a new *uniformly sampled* coordinate. We refer the reader to appendix E.2 for further details on this coordinate system and our particular implementation of it whitin `SFcollapse1D`.

The weak field regime is obtained by choosing a value of η which is smaller than the

[5] We first encountered this coordinate system when working with `NRPy+` [25, 26, 83]. It is similar to

critical value η_*, and is characterized by full dispersion. Since we do not know η_* *a priori*, this is done by trial and error. In the run whose results are presented in figures 3.1, 3.3, and 3.4, we have set $\eta = 0.05$. It can be seen that the metric quantities α and a differ at most $\sim 2\%$ from flat space, the deviation being more prominent for the lapse than the radial metric. Because of this, the backreaction from the geometry on the scalar field is negligible and we see it propagating as if it was not coupled to gravity at all, i.e. just a simple scalar wave in spherical coordinates.

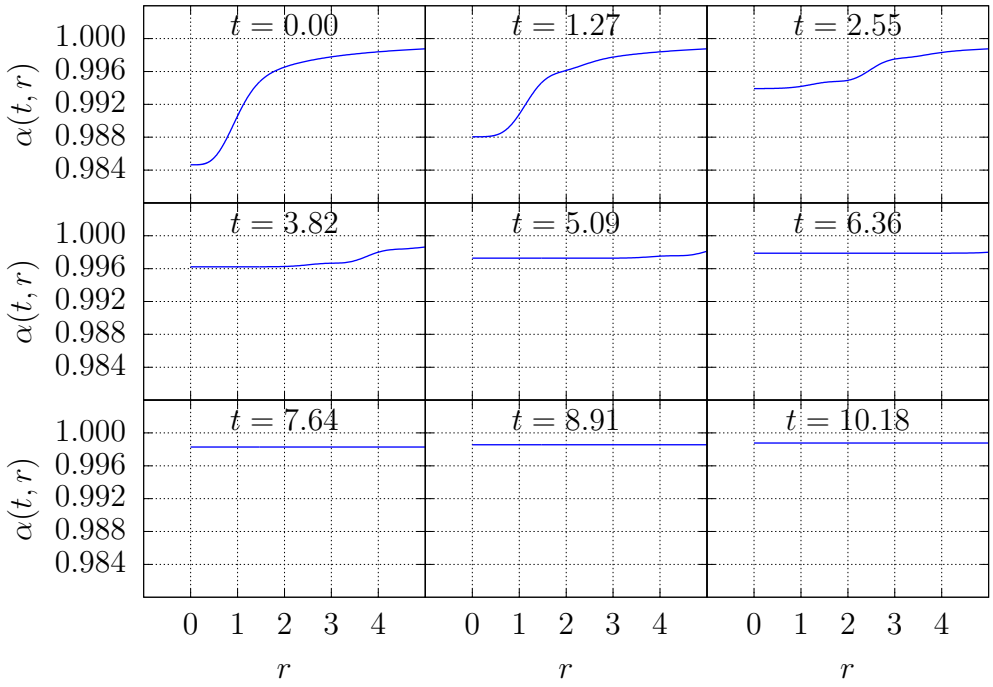


Figure 3.1: Weak field results for the lapse function $\alpha(t, r)$. The parameters of the run are $\eta = 0.05$, $r_{\max} = 64$, $w = 0.15$, and $N_r = 640$.

The strong field regime is obtained by choosing a value of η which is larger than the critical value η_*, and is characterized by the gravitational collapse of the scalar field which leads to black hole formation. For the results presented in figures 3.2, 3.5, and 3.6 we have set $\eta = 0.6$. After a black hole forms, the code eventually crashes. The more we increase the sampling around the black hole horizon, which can be determined by checking where $\frac{2M}{r} \approx 1$, the longer it will take for the code to crash, and the better we can estimate the mass of the black hole and the horizon position. Our resolution was just high enough to keep to simulation from crashing immediately after black hole formation, but certainly our grid choice was not the best to resolve the horizon, as our grid is more densily sampled around $r \approx 0$, not around $r \approx r_{\text{horizon}}$.

prolate/oblate spheroidal coordinates, but only the radial component is transformed.

The lapse function goes to zero[6] in this regime, which tells us that around that region there is virtually no proper time being elapsed between one hypersurface of constant time and the next. This is known as the *freezing of the lapse function*, and is a very useful feature to detect black hole formation.

Figure 3.5 shows clearly that no time is passing for the scalar field after it collapses, since its evolution gets frozen. The freezing of the lapse function is evident in figure 3.2, while the mass of the black hole formed (in code units) can be seen as the plateau of the function $M(t, r)$ in figure 3.6, and its value is $M_{\rm BH} \approx 0.473$.

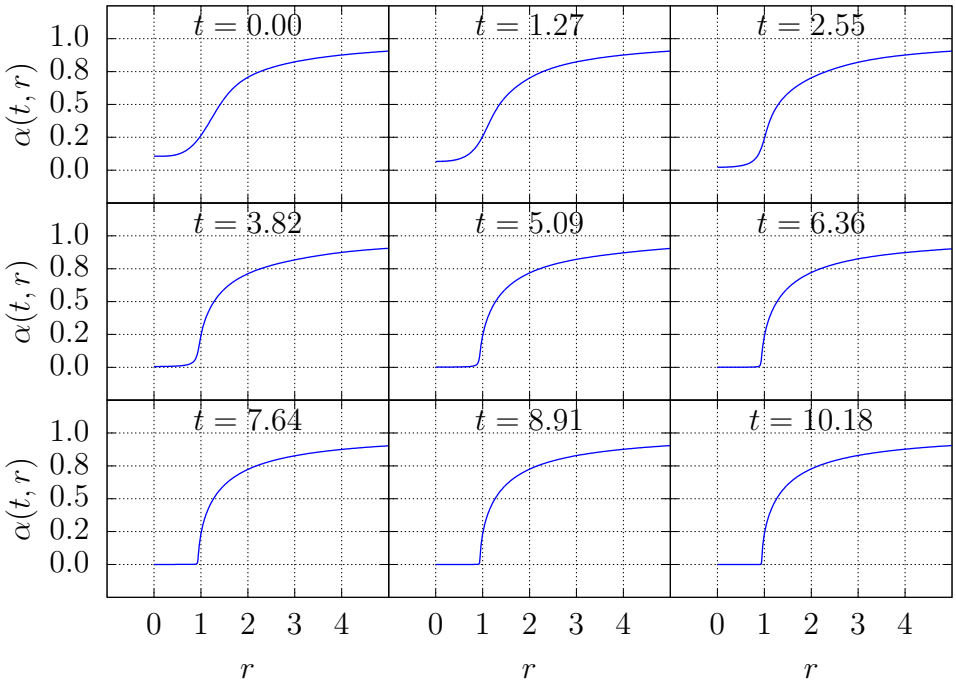


Figure 3.2: Strong field results for the lapse function $\alpha(t, r)$. The parameters of the run are $\eta = 0.6$, $r_{\rm max} = 64$, $w = 0.15$, and $N_r = 640$.

3.4 Critical phenomena

As is clear from the previous section, it is possible to choose values of η, the strength (amplitude) of the initial condition (3.38), which result in either full dispersion or gravitational collapse. We then consider the existence of a "critical value", η_*, which separates the two cases. This means that a value of $\eta < \eta_*$ will result in full dispersion, while a value of $\eta > \eta_*$ will result in gravitational collapse and black hole formation.

[6] In the actual code the lapse reaches a very small value, below 10^{-3} for example, which is enough for us to characterize black hole formation.

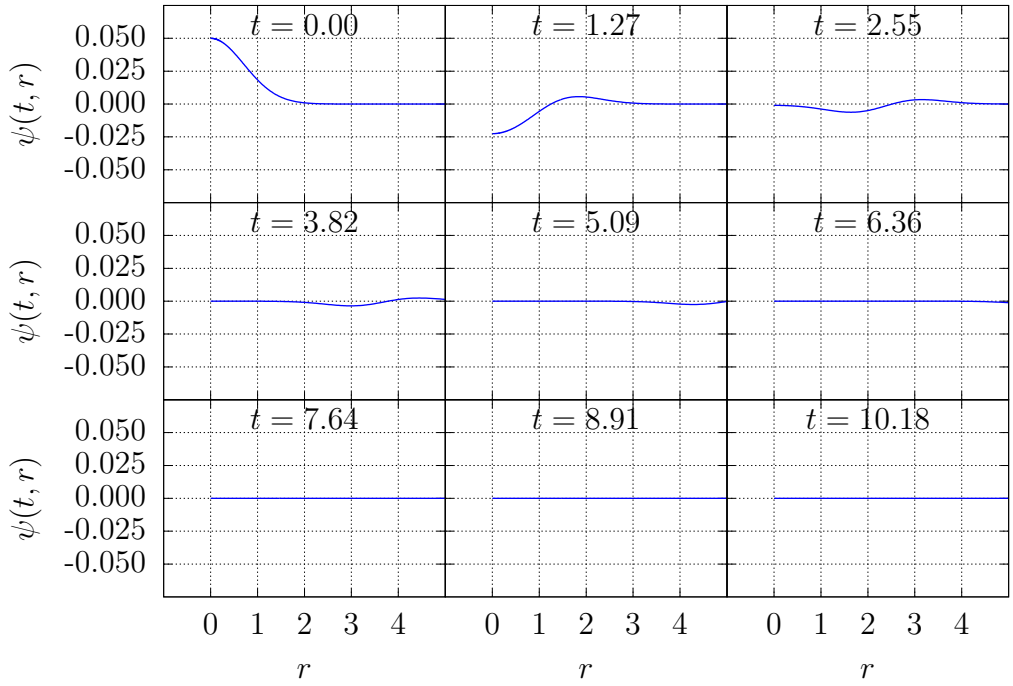


Figure 3.3: Weak field results for the scalar field $\psi(t, r)$. The parameters of the run are $\eta = 0.05$, $r_{\max} = 64$, $w = 0.15$, and $N_r = 640$.

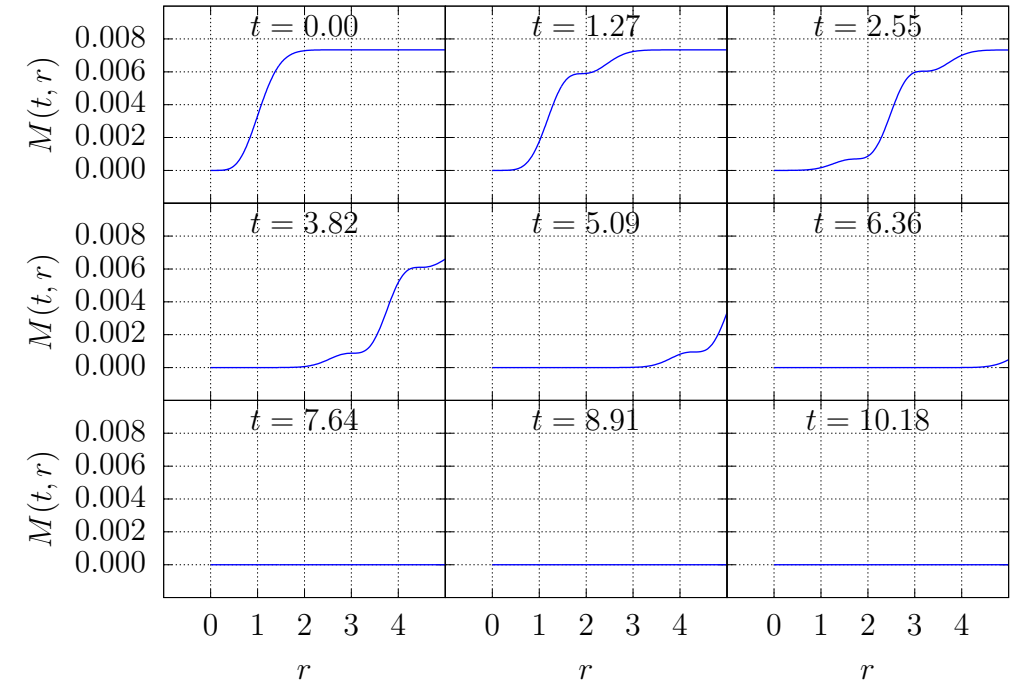


Figure 3.4: Weak field results for the mass-aspect function $M(t, r)$. The parameters of the run are $\eta = 0.05$, $r_{\max} = 64$, $w = 0.15$, and $N_r = 640$.

We are now interested in the behaviour of the scalar field, and metric quantities, as $\eta \to \eta_*$. Choptuik reported [20], after performing a series of supercritical runs, that the masses of the black holes, near the critical solution, follow the power law (3.1). However, given the considerations in [75, 78] and the more recent results presented in [79, 80], we

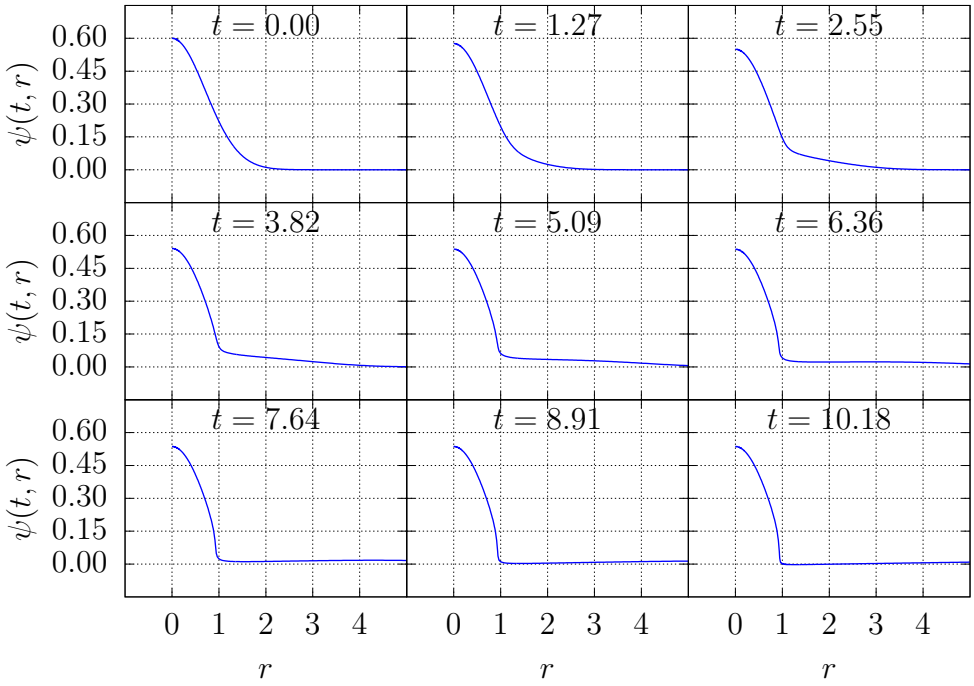


Figure 3.5: Strong field results for the scalar field $\psi(t, r)$. The parameters of the run are $\eta = 0.6$, $r_{\max} = 64$, $w = 0.15$, and $N_r = 640$.

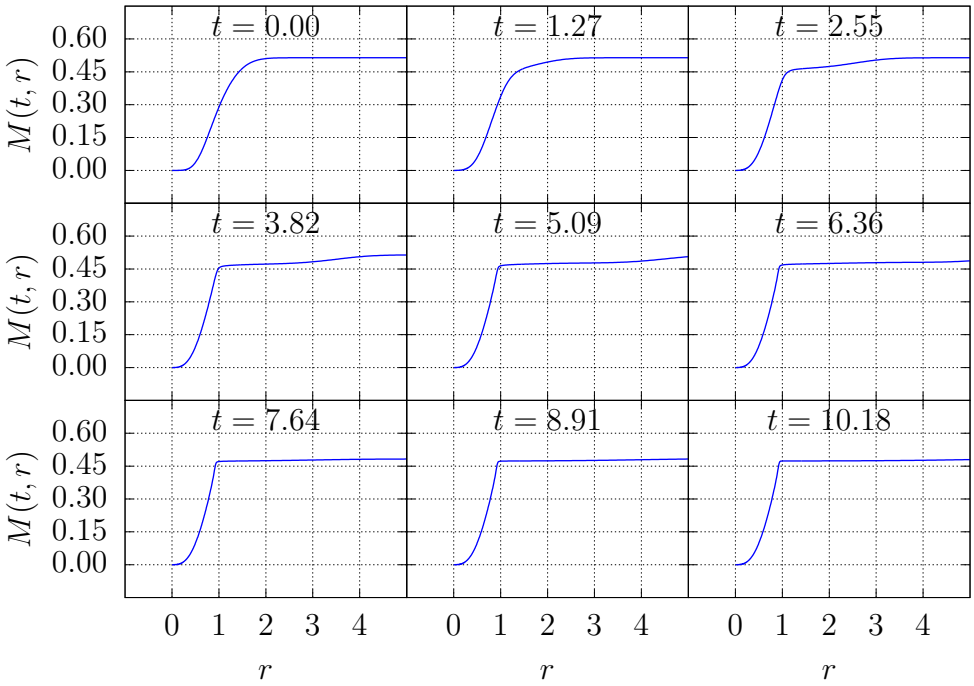


Figure 3.6: Strong field results for the mass-aspect function $M(t, r)$. The parameters of the run are $\eta = 0.6$, $r_{\max} = 64$, $w = 0.15$, and $N_r = 640$.

have opted to perform an analysis of the "near criticality" behaviour of the scalar field in *subcritical* runs instead. Subcritical runs do not form a black hole, so we cannot look at $M_{\rm BH}$ to satisfy a power law in this case. Instead, we will analyse the behaviour of the energy density, ρ, at the origin, which we will refer to as *central value*.

To obtain the near critical results, we perform a *fine-tuning* of the critical parameter η_*. We start by determining two values of η, η_{weak} and η_{strong}, which bracket η_*. We then perform a bisection search to determine the critical parameter. A useful quantity to define is the relative error

$$\delta\eta \equiv \frac{\eta_{\text{strong}} - \eta_{\text{weak}}}{\eta_{\text{weak}}} \ , \tag{3.41}$$

which is also refered to as *fine-tuning*.

After having a good estimate for η_*, we perform a series of runs with $\eta < \eta_*$ and extract the maximum value of the central energy density, $\rho^{\text{max}}_{\text{central}}$. The energy density has units of inverse length squared (L^{-2}), thus relation (3.5) states that it should behave as

$$\ln\left(\rho^{\text{max}}_{\text{central}}\right) = C - 2\gamma\, z + f(z) \ , \tag{3.42}$$

where γ is the critical exponent, $f(z)$ is periodic in z with angular frequency ω , and C is a constant. The variable z is defined as

$$z \equiv \ln|\eta_* - \eta| \ , \tag{3.43}$$

and we assume that the function $f(z)$ has the form [80]

$$f(z) = A \sin\left(\omega z + \phi_{\text{ph}}\right) \ , \tag{3.44}$$

where A and ϕ_{ph} are constants. The angular frequency ω is related to the logarithmic period Δ via the relation [84]

$$\omega = \frac{4\pi\gamma}{\Delta} \ , \tag{3.45}$$

and the oscillations are conjectured to have the universal oscillation period [84]

$$T = \frac{\Delta}{2\gamma} \approx 4.6 \ . \tag{3.46}$$

To fit the numerical data with equation (3.42), we fit the linear and oscillatory behaviours of $\ln\left(\rho^{\text{max}}_{\text{central}}\right)$ separately. First, we drop the oscillatory term altogether and assume the original power law behaviour discovered by Choptuik

$$\rho^{\max}_{\text{central}} \propto |\eta_* - \eta|^{-2\gamma} \ , \tag{3.47}$$

which results in the relation

$$\ln\left(\rho^{\max}_{\text{central}}\right) = C - 2\gamma \ln|\eta_* - \eta| \ . \tag{3.48}$$

We then use (3.48) to obtain the parameters η_*, γ, and C using `SciPy` [85]. Having determined the linear behaviour of the function, we use equation (3.42) and compute

$$f(z) = \ln\left(\rho^{\max}_{\text{central}}\right) - C + 2\gamma \ln(z) \ , \tag{3.49}$$

using the known values of η_*, γ, and C and the numerical data. We then use equation (3.44) to fit the resulting curve and obtain the remaining fit parameters.

3.4.1 Results using the ADM formalism

We solve the EKG equations using the ADM formalism by using the fully constrained system, discussed in section 3.1, with the `SFcollapse1D` code. We assume spherical symmetry and SinhSpherical coordinates. The code performs second-order in time and space discretization using centered finite differences stencils, which allow us to explicitly evolve the Klein-Gordon equations in time. The Hamiltonian constraint is solved on each slice in a pointwise fashion, with the aid of a Newton-Raphson solver, to determine the radial metric function γ_{rr}. Finally, the polar slicing condition is solved to determine the lapse on the slice.

We determine the critical parameter η_* using bisection. We have an automated routine to determine whether or not the endstate of the run is full dispersal or collapse. Throughout the run we keep track of the central value of the lapse function, $\alpha_{\text{central}}(t) \equiv \alpha(t, 0)$, and we use it to detect black hole formation. At late times, if $\alpha_{\text{central}} \to 0$ we classify the run as supercritical, while when $\alpha_{\text{central}} \to 1$ the run is subcritical. In figure 3.7 we show the behaviour of α_{central} after fine-tuning the critical parameter to $\delta\eta \sim 10^{-13}$, a few orders of magnitude above machine precision (10^{-16}). Notice that in both cases, the solution follows the critical one until some *accumulation event*, where the solution departs from the critical one and either disperses or collapses.

To obtain such a fine-tuning, we have used SinhSpherical coordinates to dramatically

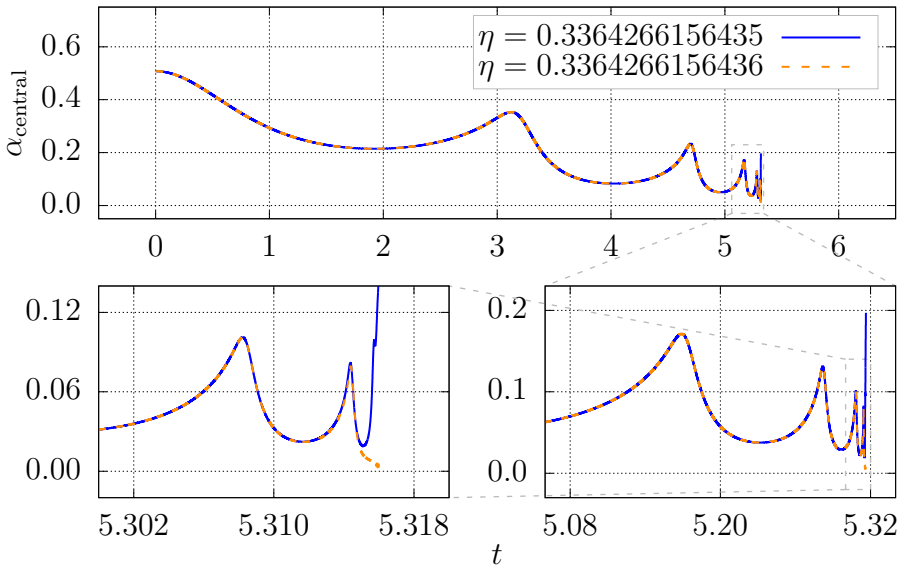


Figure 3.7: α_{central} results near criticality. The top panel shows the central value of the lapse as a function of coordinate time t throughout the entire run, while the bottom two panels are zooms of increasing magnitude towards the late behaviour of the lapse. The simulation parameters used were $N_r = 320$, $w = 0.08$, and $r_{\text{max}} = 16$. Note that we have fine-tuned the critical parameter to $\delta\eta \sim 10^{-13}$.

increase the resolution around the origin, while keeping the outer boundary relatively far away, keeping it out of causal contact with the center of the simulation. A typical run uses 320 radial points, $w = 0.08$, and $r_{\text{max}} = 16$. The fine-tuning obtained was the maximum our grid choice allowed us to obtain, with features developing beyond this point that we could no longer resolve. We note that, in principle, it should be possible to fine-tune the critical parameter even further by increasing the resolution around the origin, but we did not think it was necessary since this level of fine-tuning is enough to capture all the interesting behaviour of the critical solution.

With the parameters above, our grid has $\Delta r_{\text{min}} \approx 4.66 \times 10^{-6}$ and $\Delta r_{\text{max}} \approx 6.13 \times 10^{-1}$, which are the inner most and outer most radial step sizes, respectively. The ratio of the two is ${}^{\Delta r_{\text{max}}}/_{\Delta r_{\text{min}}} \approx 1.32 \times 10^5$. To give some perspective, a code using Berger-Oliger adaptive mesh refinement (AMR) [23] would require about 17 levels of refinement by factors of 2 to resolve the same range of length scales.

In figure 3.7, the self-similarity of the solution is evident, where the same pattern repeats itself but at ever smaller spatiotemporal scales. To visualize the behaviour of the scalar field, we look at its central value as a function of proper time,

$$\tau(t) = \int_0^t \alpha(t')\, dt' \ . \tag{3.50}$$

Figure 3.8 shows $\psi_{\text{central}}(\tau)$. The scalar field also oscillates towards the accumulation event with an amplitude of 0.61, consistent with what was found in [20, 79, 80] The accumulation time, τ_*, is defined as the time beyond which the solution no longer follows the critical one.

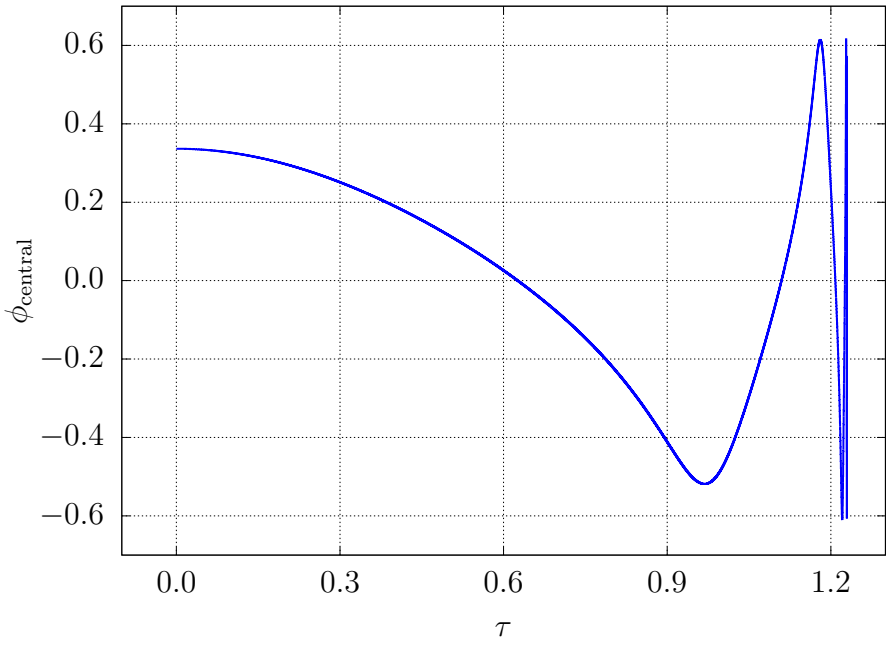


Figure 3.8: Central value of the scalar field, ψ_{central}, near criticality as a function of proper time, τ.

We now define the logarithmic time variable

$$\Lambda \equiv -\ln(\tau_* - \tau) \ . \tag{3.51}$$

Let τ_n represent a particular value of proper time for which ψ_{central} crosses zero. Consider, still, two pairs of consecutive zero crossings, (τ_n, τ_{n+1}) and (τ_m, τ_{m+1}), and assume that *in the logarithmic time*, Λ, these two pairs of points correspond to the same half-period of oscillation $\frac{\Delta}{2}$, i.e.

$$\Lambda_{n+1} - \Lambda_n = \frac{\Delta}{2} = \Lambda_{m+1} - \Lambda_m \ , \tag{3.52}$$

where $\Lambda_n \equiv \Lambda(\tau_n)$. From equations (3.51) and (3.52) we can estimate the accumulation time τ_* using

$$\ln\left(\frac{\tau_* - \tau_n}{\tau_* - \tau_{n+1}}\right) = \ln\left(\frac{\tau_* - \tau_m}{\tau_* - \tau_{m+1}}\right) \implies \tau_* = \frac{\tau_n\tau_{m+1} - \tau_m\tau_{n+1}}{\tau_n - \tau_{n+1} - \tau_m + \tau_{m+1}} \,. \tag{3.53}$$

Using the same reasoning, we can use a single pair (τ_n, τ_{n+1}) to estimate the period Δ using

$$\frac{\Delta}{2} = -\left[\ln\left(\tau_* - \tau_{n+1}\right) - \ln\left(\tau_* - \tau_n\right)\right] \implies \Delta = 2\ln\left(\frac{\tau_* - \tau_n}{\tau_* - \tau_{n+1}}\right) \,. \tag{3.54}$$

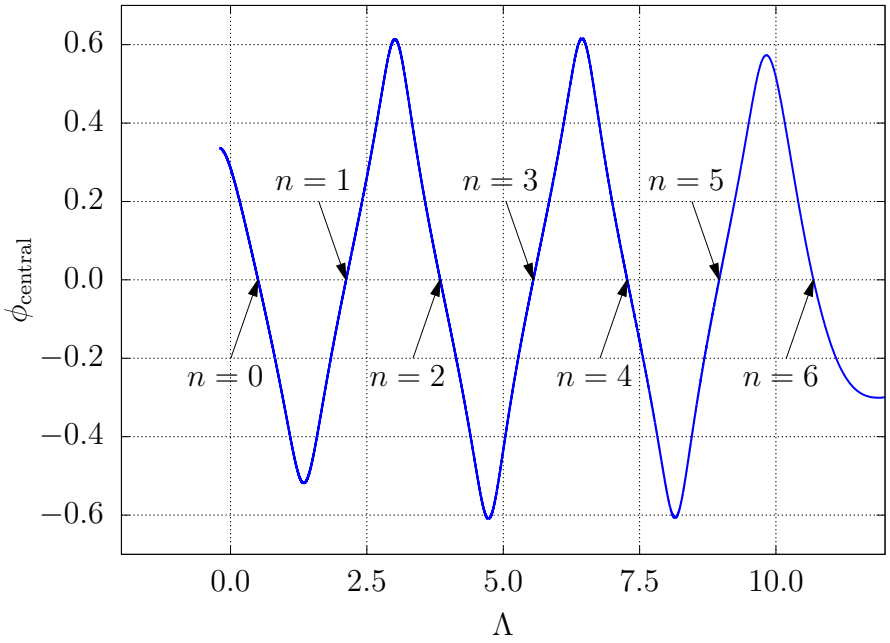


Figure 3.9: Same as figure 3.8, but the x-axis is now the logarithmic proper time $\Lambda \equiv -\ln(\tau_* - \tau)$. The accumulation time, τ_*, was determined as the average value obtained from equation (3.53) after using all possible pairs (τ_n, τ_{n+1}) for which the scalar field was already in the critical regime. The figure also displays the number of times the central value of the scalar field changes sign over the evolution. The average oscillation period was estimated to be $\Delta = 3.43(4)$, in agreement with the literature. We also observe that the central value of the scalar field oscillates with amplitude 0.61, consistent with what was found in [20, 79, 80].

In figure 3.9 we show $\psi_{\text{central}}(\Lambda)$ and the zero crossings of the scalar field. The oscillatory behaviour of the scalar field in the logarithmic time is obvious, and the oscillation period, obtained using (3.54), has the average value $\Delta = 3.43(4)$, where the uncertainty is the sample standard deviation from all possible pairs considered to compute the period. The initial and final zero crossings are neglected to avoid transient contamination of the estimate.

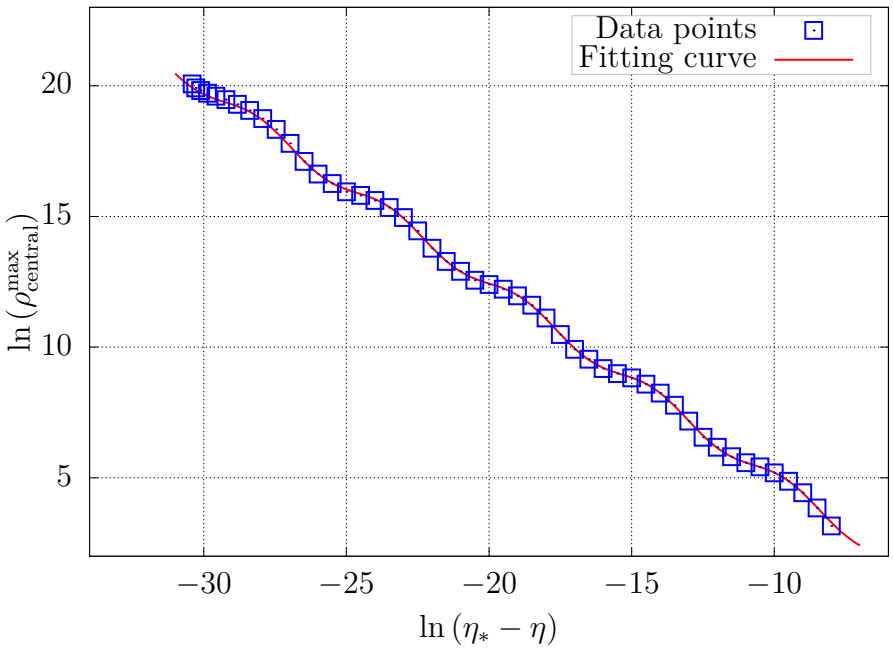


Figure 3.10: The plot shows the maximum central density for several subcritical runs. The critical exponent $\gamma \approx 0.374(1)$, oscillation period $T_{\text{data}} \approx 4.6$, and logarithmic oscillation period $\Delta = 3.47^{+0.02}_{-0.04}$, obtained using the fitting function (3.42), are all in excellent agreement with the literature.

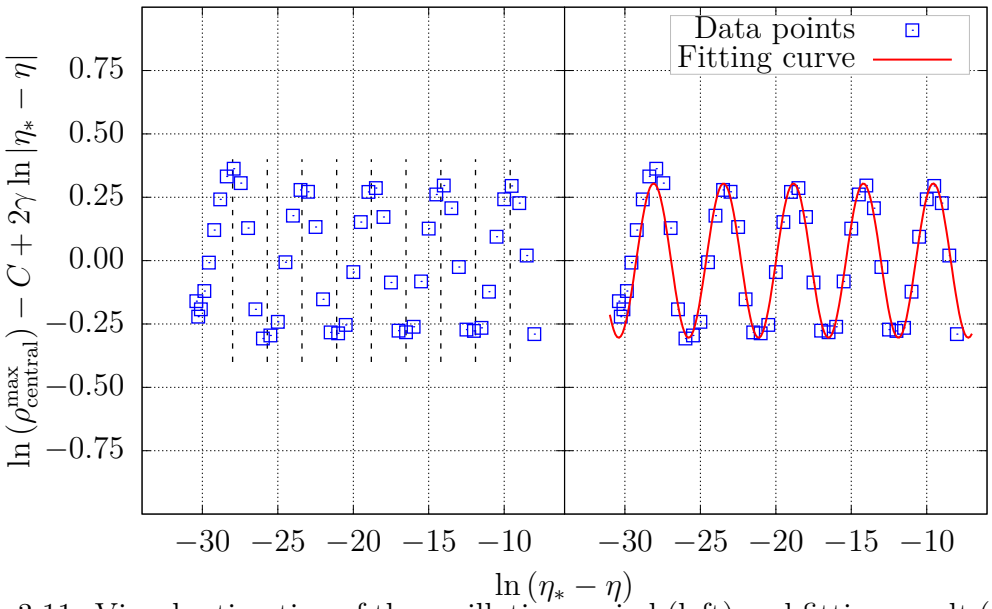


Figure 3.11: Visual estimation of the oscillation period (left) and fitting result (right). On the left panel, the vertical dashed lines are equally spaced with $\Delta z = 2.3$ and starting at $z = -28$. This allows us to visually estimate the period of oscillation to be $T_{\text{data}} \approx 4.6$, as expected. With this estimate of T_{data}, we compute $\omega = \frac{2\pi}{T_{\text{data}}}$ and use the value obtained to restringe our search for the fitting value of ω. The function $A \sin\left(\omega z + \phi_{\text{ph}}\right)$ is then used to fit the data points, and the result is displayed on the right panel.

We then perform a series of subcritical runs, assuming η_* to be the average between the smallest value of η we found to produce a black hole and the largest value of η which

results in full dispersion (these values are shown in figure 3.7). We keep track of the central energy density and extract its maximum value in each run. Figure 3.10 shows the behaviour of $\ln\left(\rho^{\max}_{\text{central}}\right)$ as a function of $\ln|\eta-\eta_*|$, as well as the result of a fit performed using `SciPy` [85] and equation (3.42). From the fit we obtain the estimates $\gamma = 0.375(1)$ and $\Delta = 3.47^{+0.02}_{-0.04}$, both in excellent agreement with the literature. Figure 3.11 shows how well our fitting function captures the oscillatory behaviour of the central density. The figure also shows our visual estimate of $T_{\text{data}} \approx 4.6$, in agreement with [84].

3.4.2 Results using the BSSN formalism

We now solve the EKG equations using Brown's covariant formulation of the BSSN formalism, discussed in section 3.2, using `NRPy+` to perform the discretization of the evolution equations for the BSSN and scalar field variables. We use fourth-order accurate, centered finite differences to approximate the spatial derivatives, and the equations are integrated in time using the method of lines and a fourth-order accurate Runge-Kutta method. The advection terms, which can be identified as having the contraction "$\beta^j\partial_j$", are discretized using fourth-order upwinded finite differences (see appendix C.3). We impose spherical symmetry by making all derivatives in the angular directions zero.

We use the "1+log" slicing condition for the lapse,

$$\partial_t\alpha = \beta^j\partial_j\alpha - 2\alpha K \ , \tag{3.55}$$

and a Gamma-driver condition for the shift vector. In the literature, we found that similar studies made use of either a frozen shift condition [79] or a first-order Gamma-driver condition [79, 80]. In contrast to them, we choose the standard, second-order, covariant Gamma-driver condition, as suggested by Brown [56],

$$\partial_t\beta^i = \beta^j\bar{D}_j\beta^i + \frac{3}{4}B^i \ , \tag{3.56}$$

$$\partial_t B^i = \beta^j\bar{D}_j B^i + \left(\partial_t\bar{\Lambda}^i - \beta^j\bar{D}_j\Lambda^i\right) - \eta_S B^i \ , \tag{3.57}$$

with

$$\eta_S = \frac{1}{2M_{\text{ADM}}} \approx 2.3 \ , \tag{3.58}$$

where $M_{\rm ADM} \approx 0.218$ is the ADM mass.[7] We note that it was quite challenging to keep the runs stable as we approached the critical solution, and this choice of η_S was empirically determined to be the one that produced the best results.

We use the same initial condition (3.38) for the scalar field ψ. The conformal metric is initialized to the flat metric, $\bar{\gamma}_{ij} = \hat{\gamma}_{ij}$, while the other quantities are assumed to be at an instant of time symmetry, which sets

$$\Pi = 0 \ , \ \bar{A}_{ij} = 0 \ , \ K = 0 \ , \ \bar{\Lambda}^i = 0 \ , \ \beta^i = 0 \ , \ B^i = 0 \ . \tag{3.60}$$

Given this choice, we must find a consistent value for the the evolved conformal factor, ϕ, so that Einstein's equations are satisfied on the initial slice. We start by defining

$$W \equiv e^{\phi} \ , \tag{3.61}$$

so that the Hamiltonian constraint (2.82) can be written as

$$\hat{D}^2 W = -2\pi W^5 \rho \ , \tag{3.62}$$

which is an elliptic equation to be solved for W. We solve this equation as described in [79], by imposing the boundary conditions

$$\partial_r W|_{r=0} = 0 \ , \tag{3.63}$$

$$\lim_{r\to\infty} W = 1 \ , \tag{3.64}$$

which are regularity at the origin and asymptotic flatness, respectively. Finally, we choose the "precollapsed lapse" condition, $\alpha = W^{-2}$, which completes our initial data specification.

Our numerical grid again uses SinhSpherical coordinates to increase the resolution

[7] The ADM mass is a gauge independent quantity. To obtain it with minimal computational effort, we use the `SFcollapse1D` code and equation (8) in [20], namely

$$M_{\rm ADM} = \int_0^\infty \left(X^2 + Y^2\right) dr \ , \tag{3.59}$$

where $X \equiv \sqrt{2\pi}\frac{r}{\sqrt{\gamma_{rr}}}\partial_r\psi$ and $Y \equiv \sqrt{2\pi}\frac{r}{\alpha}\partial_t\psi$.

around the origin. In constrast to the `SFcollapse1D` code, however, we choose a cell-centered grid,[8] which avoids the origin $r = 0$. We also perform a rescaling of tensorial quantities to remove coordinate singularities [57–60] (see also appendix F).

To make our runs more time efficient, we perform a series of *regrids* throughout the run. During a regrid, data from the current, coarser grid is interpolated onto a new, finer grid, which then becomes the new grid used in the run. A similar strategy could not be used in `SFcollapse1D`, since the EKG equations we solved using the ADM formalism contained elliptic equations, which are very sensitive to changes in the grid structure.

A typical run starts with 320 radial points, $r_{\rm max} = 64$, and $w = 0.2$. Throughout the run, we increase the radial sampling around the origin by changing $w \to w - \Delta w$ until we get $w = 0.09$. This is done over a series of 11 regrids, with $\Delta w = 0.01$. We also increase the resolution and linearity of the grid around the origin by shrinking the outer boundary to $r_{\rm max} = 8$ over an additional 8 regrids. The number of regrids was determined empirically, with more regrids added whenever the current resolution could no longer resolve the solution accurately. The initial grid has $\Delta r^{\rm initial}_{\rm max} \approx 1.003$, and the final grid has $\Delta r^{\rm final}_{\rm min} \approx 8.334 \times 10^{-6}$, with ratio $\Delta r^{\rm initial}_{\rm max}/\Delta r^{\rm final}_{\rm min} \approx 1.204 \times 10^5$. Again, a code that uses a standard Berger-Oliger AMR algorithm would require about 17 levels of refinement by factors of 2 to resolve the same range of length scales.

We emphasize that obtaining the critical parameter η_* was a lot more challenging using the BSSN formalism than the ADM one. Based on the discussions in [79, 86], it may very well be that our gauge conditions contributed to this difficulty, increasing the instability of the critical solution. Nevertheless, we have still obtained a fine-tuning of $\delta\eta \sim 10^{-10}$ with very efficient runs,[9] and using standard gauge conditions.

Kreiss-Oliger dissipation [87] has been added to the RHSs of the evolution equations, and has been essential to obtain the results presented here. The RHS for a given grid-function f is modified as

[8] A uniformly sampled *cell-centered* grid chooses to discretize $r \in [0, r_{\rm max}]$ as

$$r_j = \left(j + \frac{1}{2}\right)\Delta r \ , \tag{3.65}$$

in contrast to a *vertex-centered* grid, which sets

$$r_j = j \cdot \Delta r \ . \tag{3.66}$$

[9] A typical run using the BSSN formalism as described here took less than 10 minutes to complete, which allowed us to experiment quite a bit with the gauge and regrid conditions.

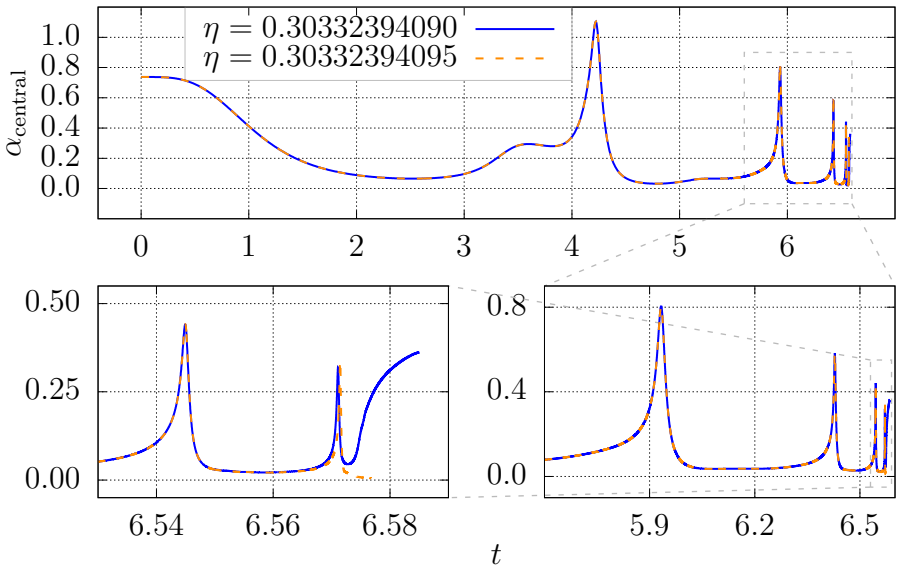


Figure 3.12: α_{central} results near criticality using the BSSN formalism. We have fine-tuned the critical parameter to $\delta\eta \sim 10^{-10}$.

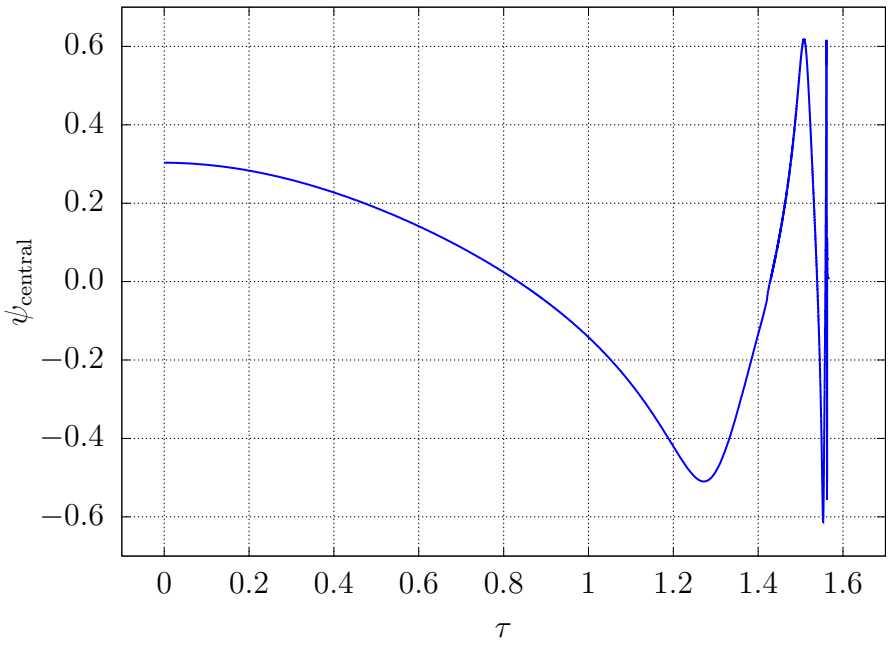


Figure 3.13: Central value of the scalar field, ψ_{central}, near criticality as a function of proper time, τ, using the BSSN formalism.

$$\partial_t f \to \partial_t f + Qf \ , \tag{3.67}$$

where Q is the Kreiss-Oliger dissipation operator, which has the general form

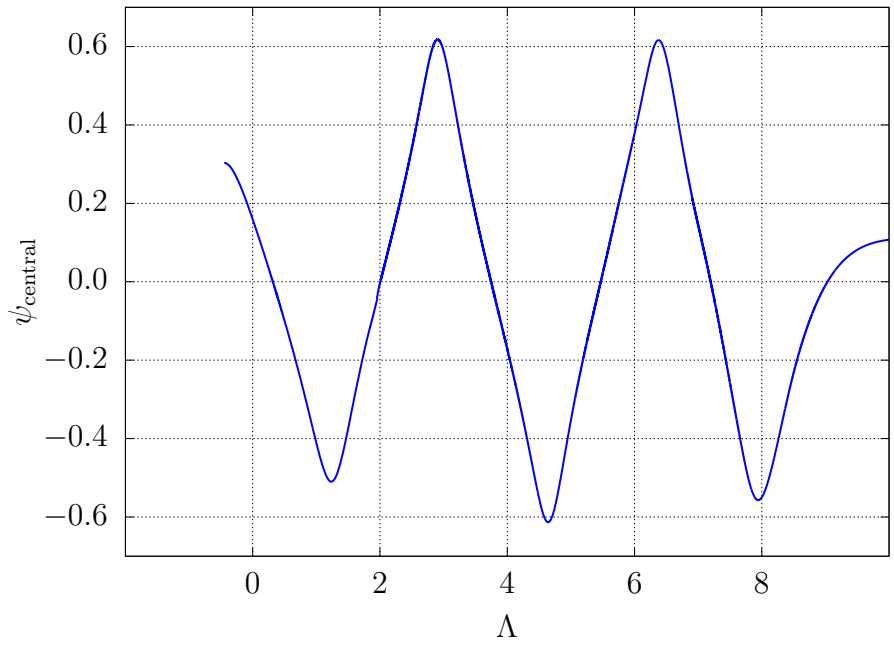


Figure 3.14: Central value of the scalar field, ψ_{central}, near criticality as a function of proper time, Λ, using the BSSN formalism.

$$Q \equiv \frac{\sigma}{2^{2p}} h^{2p-1} \left(D_+\right)^p \left(D_-\right)^p \ , \tag{3.68}$$

where σ the dissipation strength, $2(p-1)$ is the accuracy of the evolution scheme, h is the step size, and $D_\pm$ are forward and backward finite difference operators (see appendix C). Our scheme is fourth-order accurate, implying that $p = 3$, and hence

$$Qf = \frac{\sigma}{64} \Delta r^5 \frac{\partial^6 f}{\partial r^6} \ , \tag{3.69}$$

because of the assumption of spherical symmetry. This term acts as a artificial viscosity, or "numerical viscosity", and is responsible for supressing high frequency oscillations in the grid. Note that the addition of this term does not alter the solution in the continuum limit, where $\Delta r \to 0$, and should not greatly distort the numerical solution so long as σ is small.

Our results for the central energy density as a function of $\ln|\eta - \eta_*|$ are shown in figure 3.15. We note that the points where $\ln|\eta - \eta_*| < -20$ have much higher uncertainties than the points where $\ln|\eta - \eta_*| > -20$, since we could not find the critical parameter as accurately as in the ADM formalism. This produces a higher uncertainty on the critical exponent and echoing period, which we have found to be $\gamma = 0.374(3)$ and

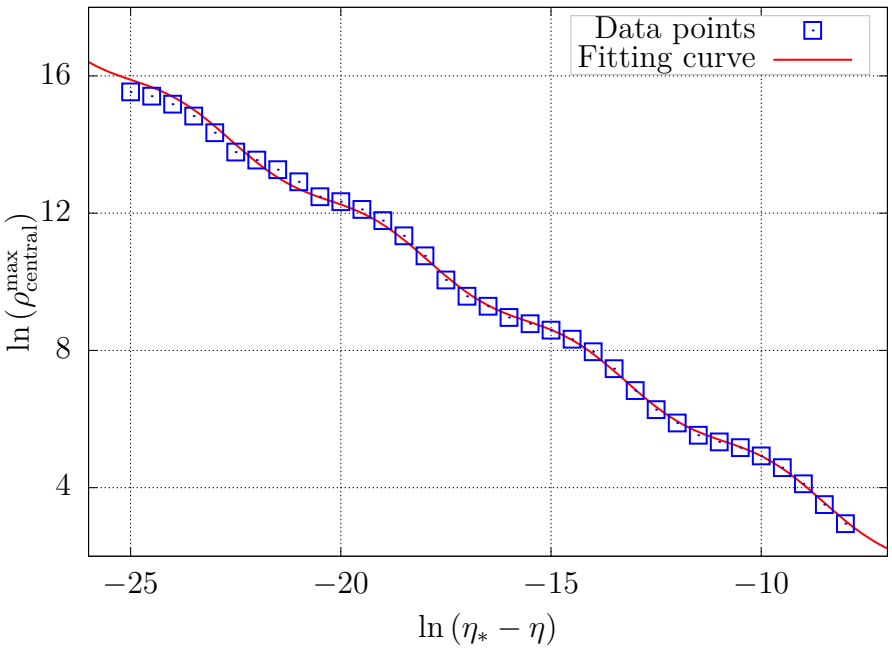


Figure 3.15: The plot shows the maximum central density for several subcritical runs. The critical exponent $\gamma = 0.374(3)$, and logarithmic oscillation period $\Delta = 3.51^{+0.01}_{-0.06}$, obtained using the fitting function (3.42), are all in excellent agreement with the literature.

$\Delta = 3.51^{+0.01}_{-0.06}$, respectively. Using equation (3.54), the echoing period has been estimated to be $\Delta = 3.46(2)$, again discarding the first and last zero crossings of the scalar field to avoid transient contamination.

3.4.3 Universality of the critical solution

We now study the universality of the critical solution by performing the same analysis of the previous two sections but with different initial conditions. These new runs have been performed using the ADM formalism and the `SFcollapse1D` code, for convenience, but we will compare them with the results in section 3.4.2 as well.

We consider the following initial conditions for the scalar field

$$\psi(0,r) = \eta \exp\left(-\frac{r^2}{\delta^2}\right) \ , \tag{3.70a}$$

$$\psi(0,r) = \eta\, r^3 \exp\left[-\left(\frac{r-r_0}{\delta}\right)^2\right] \ , \tag{3.70b}$$

$$\psi(0,r) = \eta\left\{1 - \tanh\left[\left(\frac{r-r_0}{\delta}\right)^2\right]\right\} \ , \tag{3.70c}$$

Formalism	Family	Equation	r_0	δ
BSSN	I	(3.70a)	0	1
ADM	I	(3.70a)	0	1
ADM	II	(3.70b)	2	2
ADM	III	(3.70c)	0	2.5

Table 3.1: Parameters chosen for the different initial data families.

which we will refer to as *families* I, II, and III, respectively. Notice that η is still chosen to be the parameter that is tuned to find the critical solution, while the parameters r_0 and δ are kept fixed for each family, and our choices for them are shown in table **??**.

In figures 3.16 and 3.17 we show our fine-tuning results for the families II and III, which were both of order $\delta\eta \sim 10^{-13}$. In figures 3.18 and 3.19 we show a side-by-side comparison of the near critical behaviour of the the scalar field for all the initial data in table **??**. Notice that these results indicate that the scalar field follows *the same* critical solution *for all* the initial data families we considered.

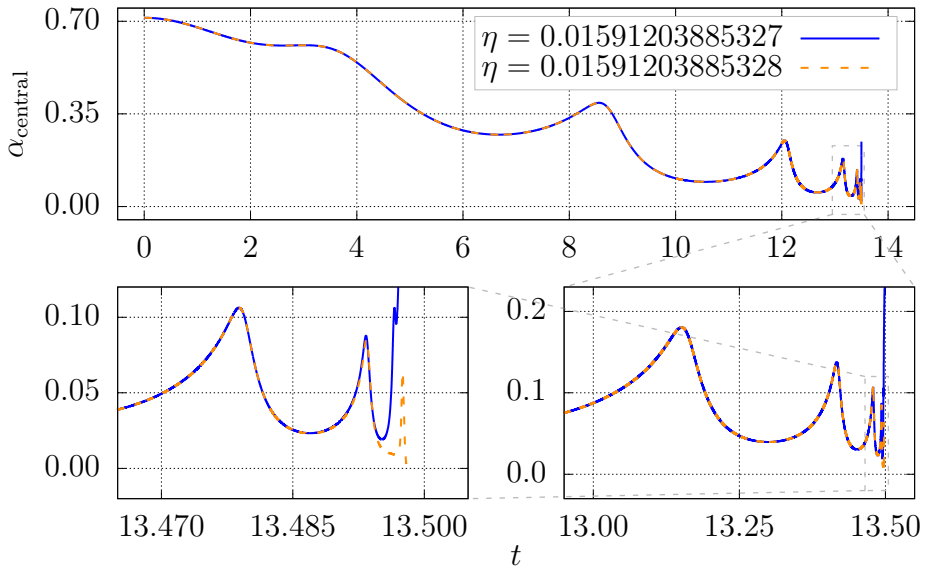


Figure 3.16: α_{central} results near criticality using a different grid setup. The simulation parameters used were $N_r = 320$, $w = 0.086$, and $r_{\text{max}} = 24$. Note that we have fine-tuned the critical parameter to $\delta\eta \sim 10^{-13}$.

Then, we perform a series of subcritical runs for all the initial data families in table **??**, extracting the maximum central energy density in each run. Figure 3.20 shows the combined results obtained for all different initial data families. In table 3.2 we present the results obtained for the critical exponent, γ, and echoing period, Δ, which shows the

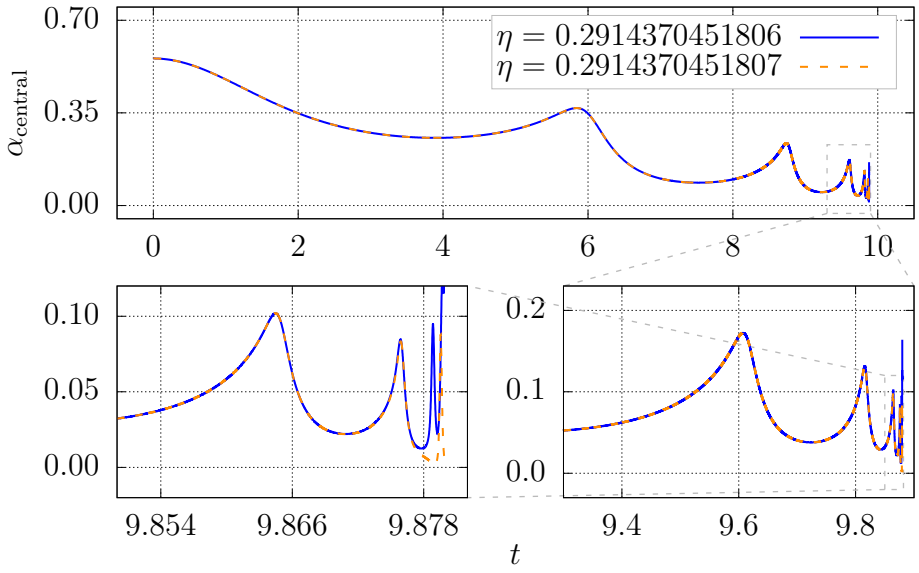


Figure 3.17: α_{central} results near criticality using a different grid setup. The simulation parameters used were $N_r = 320$, $w = 0.086$, and $r_{\max} = 24$. Note that we have fine-tuned the critical parameter to $\delta\eta \sim 10^{-13}$.

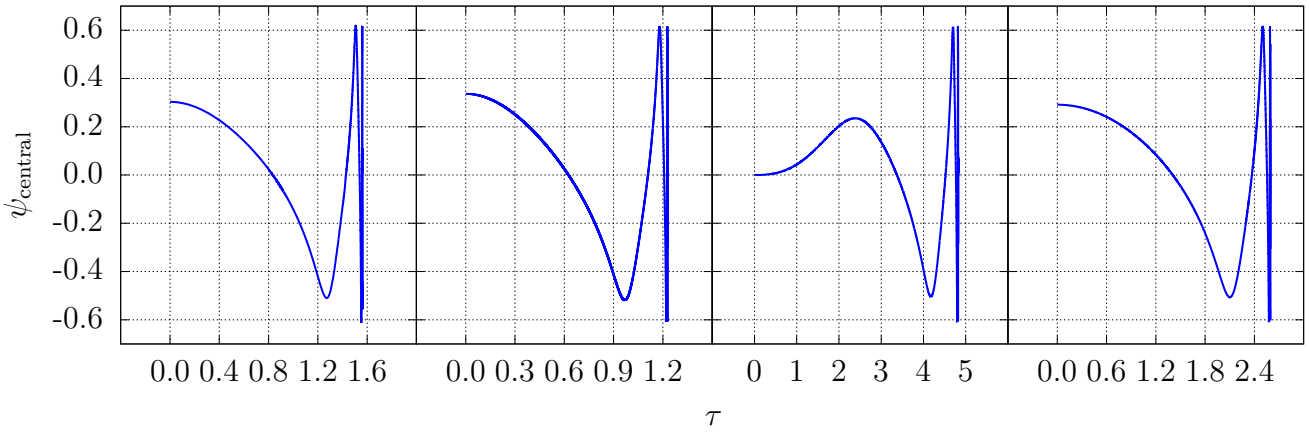


Figure 3.18: ψ_{central} results near criticality for different initial data, as a function of proper time. From left to right, we have the following formalism/families: BSSN/I, ADM/I, ADM/II, and ADM/III.

universal character of the solution.

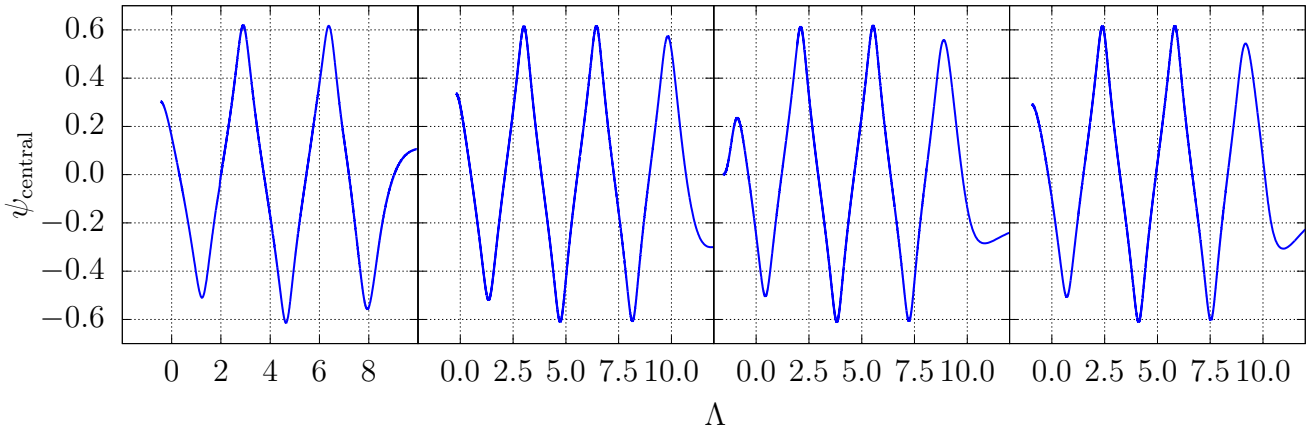


Figure 3.19: $\psi_{\rm central}$ results near criticality for different initial data, as a function of logarithmic proper time. From left to right, we have the following formalism/families: BSSN/I, ADM/I, ADM/II, and ADM/III. It is clear that the scalar field follows *the same* critical solution.

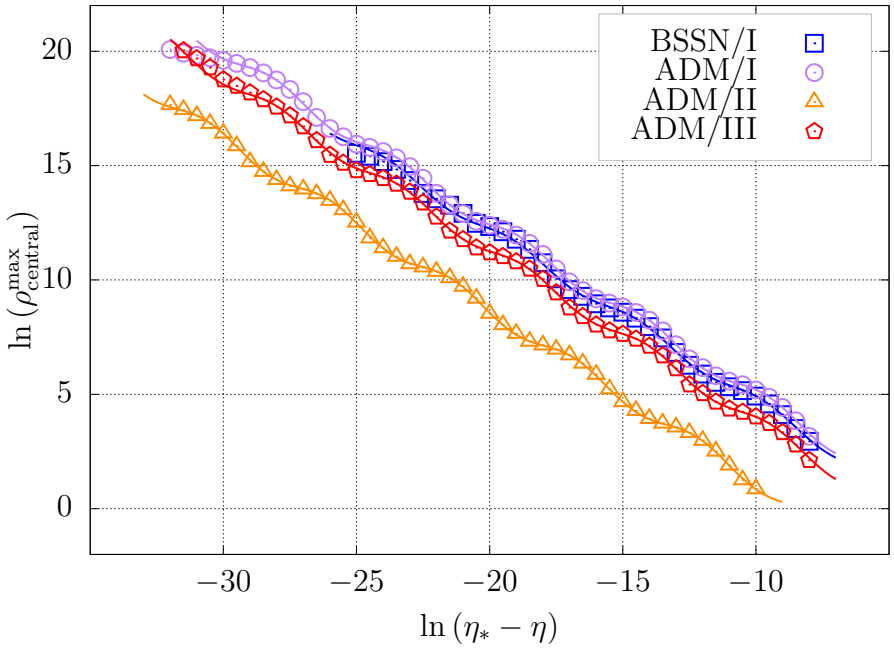


Figure 3.20: The plot shows the maximum central density for several subcritical runs for different initial data.

Formalism	Family	γ	Δ	
			Eq. (3.45)	Eq. (3.54)
BSSN	I	0.374(3)	$3.51^{+0.01}_{-0.06}$	3.46(2)
ADM	I	0.375(1)	$3.47^{+0.02}_{-0.04}$	3.43(4)
ADM	II	0.376(1)	$3.45^{+0.02}_{-0.03}$	3.42(4)
ADM	III	0.375(1)	$3.47^{+0.03}_{-0.05}$	3.44(6)

Table 3.2: Results for the critical exponent and echoing period for different formalisms and initial data families.

This page intentionally left blank.

Chapter 4

Neutron stars

We will now discuss neutron stars (NS) and, more specifically, how to accurately model them in a numerical simulation. We will focus on the way NSs were originally described in `IllinoisGRMHD` [28, 29], and how we modified the code to improve this description [30].

Equations of state (EOS) that accurately describe a NS must be valid for a very wide range of densities, temperatures, and composition. To give some perspective, we need to be able to describe densities ranging from nuclear densities of order $\sim 10^{15}$ g/cm^3 up to atmospheric densities which tend to be up to 10 orders of magnitudes smaller[1], while the temperature typically varies from zero to hundreds of MeV, and, for example, the electron fraction from zero up to 60% (see [88, 89] for good reviews).

The simplest EOS used to model NS is the so-called *polytropic EOS*, which relates the pressure and the density via a simple power law. However, these are very crude descriptions of the matter distribution inside a NS. A much more complete description is obtained using the so-called *tabulated EOS* (e.g. the ones in [32]), which take into account many features of nuclear matter physics, astrophysics, and experiments. This results in a more sophisticated EOS, relating the pressure with not only the density, but also other thermodynamics quantities.

Implementation of tabulated EOSs, however, is not trivial. As its name suggests, the EOS is constructed as a numerical *table*. The tables we are interested in are those available at [90], for which a particular numerical value of the pressure is found when particular numerical values of density, temperature, and electron fraction are given. This

[1] In reality, if we assume a single, isolated NS, and we start moving radially from its center, we would eventually hit vacuum. In numerical codes, however, we tend to introduce an artificial "atmosphere" that extends all the way to the edge of the computational domain.

results in a discrete, three-dimensional table which is computed ahead of time and then interpolated inside GRMHD codes whenever the pressure (and/or other quantities) needs to be determined.

To avoid dealing with the more complicated algorithms that must be implemented in order to use tabulated EOSs, we have the option of using the so-called *piecewise polytropic EOS* (PPEOS). PPEOSs are approximations of some aspects of tabulated EOSs that provide an analytic expression for the pressure-density relation. Usually this involves considering fixed values for other thermodynamic quantities and assuming that they remain constant throughout a dynamical run.

In this chapter we describe these different EOSs in some detail and discuss how we have implemented them in both the `NRPy+` infrastructure and the `IllinoisGRMHD` code. When we started collaborating with the West Virginia University Numerical Relativity group, lead by Prof. Dr. Zach B. Etienne, lead author of `IllinoisGRMHD`, the code supported hybrid EOSs [91], which assumed a simple polytrope-based cold component. Some functions of the code had been designed to support for more general EOSs, e.g. a piecewise polytrope-based cold component, but the code lacked all the low level infrastructure to handle them.

Over the past fall (in the Northern hemisphere), we have implemented PPEOS support in `IllinoisGRMHD`, making our implementation publicly available at [29]. A new version of `IllinoisGRMHD`, which is fully generated using `Jupyter` notebooks [92] and makes use of the `NRPy+` infrastructure, is also publicly available at [30]. The results obtained from this implemetation, in particular the ones discussed in section 4.5, will be the subject of a publication that is currently under preparation.

We are currently working on getting `IllinoisGRMHD` to work with tabulated EOSs, particularly the SRO EOS [32]. While the source code of this modified version is currently not available, we do plan on open-sourcing it `IllinoisGRMHD` once it has been properly debugged and our results have been published.

The main motivation to get `IllinoisGRMHD` working with the SRO EOS is to make it compatible with `HARM3D` [93]. `HARM3D` is a GRMHD code designed to evolve black hole accretion-disks, and we intend to use it alongside `IllinoisGRMHD` in a hand-off approach. The dynamics of a binary neutron star (BNS) system will be initially evolved using `IllinoisGRMHD`, from inspiral, to merger, to the formation of a hypermassive neu-

tron star, and eventual collapse into a black hole. Once the black hole settles into a nearly spherically symmetric and stable configuration, `IllinoisGRMHD` will interrupt its evolution and hand-off the evolution data to `HARM3D`, which will then be responsible for evolving the accretion-disk that forms around the black hole.

In this chapter we thus present our progress in getting `NRPy+` and `IllinoisGRMHD` to work with tabulated EOSs, while describing our implementation of PPEOS support, which was a necessary intermediate step to familiarize our group with `IllinoisGRMHD`. In section 4.1 we talk about simple polytropic EOSs. Tabulated EOSs are discussed in section 4.2, while the PPEOS approximation of tabulated EOSs is discussed in section 4.3. Section 4.4 is dedicated to results that validate the implementations we have done so far, specifically the the "mass vs radius" test (section **??**) and the "hydro without hydro" test (section 4.4.2). Finally, section 4.5 is dedicated to describing our studies of a particular BNS system using PPEOS. The results of this last section will be the subject of a publication which is currently under preparation.

4.1 Simple polytropic equation of state

`IllinoisGRMHD` adopts a *hybrid* equation of state (EOS) of the form [91]

$$P = P_{\rm cold} + P_{\rm th} \ , \tag{4.1}$$

where $P_{\rm cold}$ and $P_{\rm th}$ are the cold and thermal components of the pressure P, respectively. Typically, we set

$$P_{\rm th} = (\Gamma_{\rm th} - 1)\,\rho\,(\epsilon - \epsilon_{\rm cold}) \ , \tag{4.2}$$

where ρ is the baryonic density, $\Gamma_{\rm th}$ is a constant parameter which determines the conversion efficiency of kinetic to thermal energy at shocks, ϵ is the specific internal energy and $\epsilon_{\rm cold}$ its cold part. The function $\epsilon_{\rm cold}$ is related to $P_{\rm cold}$ by the first law of thermodynamics,

$$\epsilon_{\rm cold} = \int d\rho\, \frac{P_{\rm cold}}{\rho^2} \ . \tag{4.3}$$

The cold pressure, $P_{\rm cold} = P_{\rm cold}(\rho)$, can be modeled using a *polytropic EOS*,

$$P_{\text{cold}} = \kappa\rho^{\Gamma} \ . \tag{4.4}$$

Using equation (4.3), we find that the cold part of the specific internal energy, for a simple polytrope, is given by

$$\epsilon_{\text{cold}} = \int d\rho\, \kappa\rho^{\Gamma-2} = C + \frac{\kappa\rho^{\Gamma-1}}{\Gamma-1} \ , \tag{4.5}$$

with C an integration constant which is set to zero by demanding that $\lim_{\rho\to 0} \epsilon_{\text{cold}} = 0$.

This particular choice of equation of state is scale free, allowing for tuning of the arbitrary constants κ and Γ. For example, let us say that for a particular value of density, ρ_c, we know that the pressure is $P(\rho_c) = P_c$. Then, once $\Gamma = \Gamma_c$ is specified, κ is determined automatically via the constraint

$$\kappa = \frac{P_c}{\rho_c^{\Gamma_c}} \ . \tag{4.6}$$

We will use this in section 4.4.2 to produce neutron stars which are more closely related with the ones we obtain when using piecewise polytropic and tabulated EOS.

4.2 Tabulated equations of state

Tabulated EOS are the most advanced and best EOS we currently have available to describe the matter content of a neutron star. These EOS, in particular those described in [32], take into account, in their construction, microscopic interactions as well as astrophysical and experimental inputs. This means that the EOS becomes a lot more general than polytropic ones, and, in particular, the pressure now becomes a function of additional thermodynamics variables, for example[2]

$$P = P\left(\rho, T, Y_e\right) \ , \tag{4.7}$$

where T is the temperature and Y_e the electron fraction. Determining P at a given tuple (ρ, T, Y_e) is not trivial, and we no longer have an analytical expression for the function $P\left(\rho, T, Y_e\right)$. Instead, we settle to a *table*, containing particular numerical values of the

[2]This is specific to the tabulated EOS we are using and should not be taken to be the most general possibility.

tuple (ρ, T, Y_e) and the value of P at those particular values. Similarly, the internal energy, ϵ, the entropy S, and many other thermodynamics quantities are also given in this way.

To construct the tables in [32], it is assumed a composition of free nucleons, electrons, positrons, and α-particles. A liquid drop model is used to determine thermodynamics variables in the high density regime, while a nuclear statistical equilibrium model is used for the low density regime.[3] The regimes are then smoothly "glued" together, creating a single table that is able to describe a wide range of densities, temperatures, and compositions.

To determine a particular value of $P(\rho, T, Y_e)$ that is not readily available in the table, we can interpolate it from the known values of the pressure in the table. Using tabulated EOS usually leads to a slower code, since many interpolations need to be performed over the course of a single run. It is worth noting, though, that if we use a very high resolution table, we can use lower interpolation orders, which may result in significant speedups.

The implementations of tabulated EOS that we are implementing in `NRPy+` and `IllinoisGRMHD` use the SRO tables [32], which are available at [90, 95]. These tables give us the pressure and density in cgs units, while the temperature is given in MeV.

The thermodynamics quantities in these tables are all given as functions of density, ρ, temperature, T, and electron fraction, Y_e. The most relevant quantities that we have available from the EOS tables are described in table 4.1.

Quantity	**Variable**
Pressure	P
Entropy	S
Specific internal energy	ϵ
Neutron chemical potential	μ_n
Proton chemical potential	μ_p
Electron chemical potential	μ_e
Neutrino chemical potential	$\mu_\nu = \mu_n - \mu_p + \mu_e$
Adiabatic index	$\Gamma \equiv \frac{d\log P}{d\log \rho}$

Table 4.1: Most relevant thermodynamic quantities available in the SRO tabulated EOS.

All quantities in the table are given, by default, in cgs units, with the temperature given in MeV. To avoid having to deal with very large numbers[4], we rewrite all table quantities in dimensionless units. From the mass of the Sun, $M_\odot$, the gravitational constant,

[3] For a good introduction on the thermodynamics of neutron stars, see e.g. [94].

[4] For example, a density $\rho \sim 10^{15}$ g/cm^3 at $T \sim 0.1$ MeV and $Y_e \sim 0.1$ results in $P \sim 10^{38}$ dyn/cm^2.

$G_{\rm N}$, and the speed of light, c, we construct the quantities

$$\tilde{\ell} \equiv \frac{M_\odot G}{c^2} \implies [\tilde{\ell}] = \text{Length} , \tag{4.8}$$

$$\tilde{t} \equiv \frac{\ell}{c} \implies [\tilde{t}] = \text{Time} , \tag{4.9}$$

$$\tilde{P} \equiv \frac{M_\odot}{\tilde{\ell}\tilde{t}^2} \implies [\tilde{P}] = \text{Pressure} , \tag{4.10}$$

$$\tilde{\rho} \equiv \frac{M_\odot}{\tilde{\ell}^3} \implies [\tilde{\rho}] = \text{Density} , \tag{4.11}$$

$$\tilde{\epsilon} \equiv c^2 \implies [\tilde{\epsilon}] = \text{Specific internal energy} , \tag{4.12}$$

which we then use to convert the table to code (dimensionless) units. For example, the pressure in the table, $P_{\rm cgs}$, is converted to code units, $P_{\rm code}$, using

$$P_{\rm code} = \frac{P_{\rm cgs}}{\tilde{P}} . \tag{4.13}$$

To use the EOS tables with `NRPy+`, we use interpolation routines from `scipy` [85], which allow us to compute P for any tuple (ρ, T, Y_e). Our experience shows that the errors we obtain with linear interpolation, for the case of a single independent variable variable (i.e. when the other two independent variables are kept fixed), are not significantly worse than those obtained with higher order interpolations, allowing us to use these less computationally expensive routines. Similarly, bilinear and trilinear interpolations also produce reasonable results for the more general cases.

To use the EOS tables in `IllinoisGRMHD`, on the other hand, we have adapted interpolation routines provided by the authors of the tables, which are written in `C/C++`, are completely open-sourced, and are available at [96]. Most of the infrastructure has already been implemented and the code already compiles after all the additions. We are currently subjecting the code to a series of tests to validate our implementation before we start studying the physics of binary neutron stars (BNS) mergers.

4.3 Piecewise polytropic equation of state

Our motivation to study piecewise polytropic EOS (PPEOS) was twofold. The main goal of our work with `IllinoisGRMHD` is to implement a more realistic description of NSs,

which better models the matter content of a NS and adds new physics into the simulation, for example neutrino emission/absorbtion. To this end, we need to add routines to the code so that it can support tabulated EOS, which are more complicated than routines for simpler, analytic EOS. Moreover, our group at IF-USP had little experience with `IllinoisGRMHD`, and thus decided to go over the code, improving its documentation and adding PPEOS support, which required less effort for new users and would allow us to familiarize ourselves with the code.

A piecewise polytropic EOS is constructed from a set o simple polytropic EOS that are "glued" together so that we obtain a pressure-density relation that approximates a particular tabulated EOS. Note, however, that the pressure is treated as $P = P(\rho)$, which means that we must choose a particular value for the other thermodynamic quantites and assume that they remain fixed throughout the run. If we fix the electron fraction, Y_e, for example, we are then unable to accurately study the physics of neutrino emission/absorbtion, which results in the cooling/heating of the system. Note, however, that the hybrid EOS used in `IllinoisGRMHD`, namely (4.1), is intended to account for some thermalization at shocks, even though we do not explicitly keep track of the temperature when using it.

To formalize what we discussed in the previous paragraph, a PPEOS consider a set of densities, $\{\rho_0, \rho_1, \ldots, \rho_{N-1}, \rho_{N-2}\}$, such that the cold pressure, $P_{\rm cold}$, is given by

$$P_{\rm cold} = \begin{cases} \kappa_0 \rho^{\Gamma_0} & , \quad \rho \leq \rho_0 \\ \kappa_1 \rho^{\Gamma_1} & , \quad \rho_0 \leq \rho \leq \rho_1 \\ \vdots & \vdots \\ \kappa_j \rho^{\Gamma_j} & , \quad \rho_{j-1} \leq \rho \leq \rho_j \\ \vdots & \vdots \\ \kappa_{N-2} \rho^{\Gamma_{N-2}} & , \quad \rho_{N-3} \leq \rho \leq \rho_{N-2} \\ \kappa_{N-1} \rho^{\Gamma_{N-1}} & , \quad \rho \geq \rho_{N-2} \end{cases} , \tag{4.14}$$

where the constants κ and Γ in (4.4) are now replaced by the sets of constants $\{\kappa_0, \kappa_1, \ldots, \kappa_{N-2}, \kappa_{N-1}\}$ and $\{\Gamma_0, \Gamma_1, \ldots, \Gamma_{N-2}, \Gamma_{N-1}\}$.

We then assume that the pressure is everywhere continuous, though we do not demand that the different pieces be smoothly connected.[5] The continuity condition of the pressure

[5] This will result in discontinuities on the first derivative of the pressure, as we will see later on.

implies that

$$P_{\rm cold}(\rho_0) = \kappa_0 \rho_0^{\Gamma_0} = \kappa_1 \rho_0^{\Gamma_1} \implies \kappa_1 = \kappa_0 \rho_0^{\Gamma_0 - \Gamma_1} \ , \tag{4.15}$$

imposing a constraint on the value of the constant κ_1. In general, we have a series of constraints which is given by

$$\kappa_j = \kappa_{j-1} \rho_{j-1}^{\Gamma_{j-1} - \Gamma_j} \ , \ j \in [1, N-1] \ . \tag{4.16}$$

The specific internal energy is still given by equation (4.3), but using the PPEOS (4.14) we obtain

$$\epsilon_{\rm cold} = \begin{cases} \frac{\kappa_0 \rho^{\Gamma_0 - 1}}{\Gamma_0 - 1} + C_0 & , \quad \rho \leq \rho_0 \\ \frac{\kappa_1 \rho^{\Gamma_1 - 1}}{\Gamma_1 - 1} + C_1 & , \quad \rho_0 \leq \rho \leq \rho_1 \\ \vdots & \vdots \\ \frac{\kappa_j \rho^{\Gamma_j - 1}}{\Gamma_j - 1} + C_j & , \quad \rho_{j-1} \leq \rho \leq \rho_j \\ \vdots & \vdots \\ \frac{\kappa_{N-2} \rho^{\Gamma_{N-2} - 1}}{\Gamma_{N-2} - 1} + C_{N-2} & , \quad \rho_{N-3} \leq \rho \leq \rho_{N-2} \\ \frac{\kappa_{N-1} \rho^{\Gamma_{N-1} - 1}}{\Gamma_{N-1} - 1} + C_{N-1} & , \quad \rho \geq \rho_{N-2} \end{cases} \ , \tag{4.17}$$

where C_j are integration constants. As in the simple polytrope case, we fix $C_0 = 0$ by demanding that $\epsilon_{\rm cold}(\rho = 0) = 0$. As with the pressure, we demand that $\epsilon_{\rm cold}$ be everywhere continuous, which constraints the value of the integration constants C_j. For example,

$$\frac{\kappa_0 \rho_0^{\Gamma_0 - 1}}{\Gamma_0 - 1} + C_0 = \frac{\kappa_1 \rho_0^{\Gamma_1 - 1}}{\Gamma_1 - 1} + C_1 \implies C_1 = C_0 + \frac{\kappa_1 \rho_0^{\Gamma_1 - 1}}{\Gamma_1 - 1} - \frac{\kappa_0 \rho_0^{\Gamma_0 - 1}}{\Gamma_0 - 1} \ , \tag{4.18}$$

or, in general,

$$C_j = \begin{cases} 0 & , \text{ if } \quad j = 0 \\ C_{j-1} + \frac{\kappa_j \rho_{j-1}^{\Gamma_j - 1}}{\Gamma_j - 1} - \frac{\kappa_{j-1} \rho_{j-1}^{\Gamma_{j-1} - 1}}{\Gamma_{j-1} - 1} & , \text{ if } \quad 1 \leq j \leq N-1 \end{cases} \ . \tag{4.19}$$

Our implementation of PPEOS is quite general, in the sense that it can handle an arbitrary number of simple polytropes and constants $\{\rho_j, \kappa_j, \Gamma_j\}$. However, we focus on the particular set of PPEOS described in [31], since it is widely adopted by other

groups and thus it becomes easy to validate our implementation. It is worth emphasizing that PPEOS are neither the most realistic EOS to describe a NS, nor are they completely consistent with their tabulated cousins. However, they offer an arguably better description of $P_{\rm cold}(\rho)$ and $\epsilon_{\rm cold}(\rho)$ than just a simple polytrope.

The PPEOS in [31] have $N = 7$, meaning that we have seven simple polytropic EOSs. The values of

$$\{\log_{10}(P_4), \rho_0, \rho_1, \rho_2, \rho_4, \rho_5, \Gamma_0, \Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \Gamma_5, \Gamma_6, \kappa_0\} \ , \tag{4.20}$$

where $P_4 \equiv P_{\rm cold}(\rho_4)$, are taken from tables II and III in [31], while the values of the constants $\{\rho_3, \kappa_1, \ldots, \kappa_6\}$, as well as the specific internal energy integration constants, $\{C_j\}$, must be determined using the constraints (4.16) and (4.19).

Initially, we are able to compute $\{\kappa_1, \kappa_2, \kappa_3\}$ using (4.16) directly. Computing κ_4 directly is not possible,

$$\kappa_4 = \kappa_3 \rho_3^{\Gamma_3 - \Gamma_4} \ , \tag{4.21}$$

since we do not know the value of ρ_3. To determine ρ_3, we use

$$P_4 = \kappa_4 \rho_4^{\Gamma_4} = \left(\kappa_3 \rho_3^{\Gamma_3 - \Gamma_4}\right) \rho_4^{\Gamma_4} \implies \rho_3 = \left[\frac{P_4}{\kappa_3 \rho_4^{\Gamma_4}}\right]^{\frac{1}{\Gamma_3 - \Gamma_4}} \ . \tag{4.22}$$

Finally, we are able to compute the remaining values of κ_j using, again, the continuity condition (4.16). Having determined all constants $\{\rho_j, \kappa_j, \Gamma_j\}$, we are able to determine $\{C_j\}$ using the constraint (4.19).

As with tabulated EOSs, it is desirable to work with nuclear densities and pressures which are of order unity.[6] To this end, we perform a unit conversion from cgs to code units. The units of our code are a combination of geometrized units, where $G_{\rm N} = 1 = c$, and a normalization of the nuclear density to unity, i.e.

$$\rho_{\rm nuc} \equiv 1 \times 10^{15} \ {\rm g/cm^3} \to 1 \ . \tag{4.23}$$

The densities ρ_j are then adjusted by performing a ratio-preserving rescaling, i.e.

[6] Note that we are not really referring to the density of the nucleus of an atom, which can be estimated to be $\approx 2 \times 10^{14}$ g/cm^3, but instead to the value given in (4.23). This is an arbitrary, but convenient, value used to normalize the densities in the code. That being said, the use of the term "nuclear density" is loosely used in the literature to refer to densities of order 10^{15} g/cm^3.

$$\frac{\rho_j^{\rm cgs}}{\rho_{j-1}^{\rm cgs}} = \frac{\rho_j^{\rm code}}{\rho_{j-1}^{\rm code}} \implies \rho_{j-1}^{\rm code} = \left(\frac{\rho_{j-1}^{\rm cgs}}{\rho_j^{\rm cgs}}\right) \rho_j^{\rm code} \ . \tag{4.24}$$

The pressure is then written in units of density (i.e. they get divided by c^2), so that they obey a similar ratio-preserving rescaling,

$$P_{j-1}^{\rm code} = \left(\frac{\rho_{j-1}^{\rm cgs}}{\rho_j^{\rm cgs}}\right) P_j^{\rm code} \ , \tag{4.25}$$

which finally allows us to also compute

$$\kappa_j^{\rm code} = \frac{P_j^{\rm code}}{\left(\rho_j^{\rm code}\right)^{\Gamma_j}} \ . \tag{4.26}$$

Lastly, we compute the integration constants C_j, already in code units, using equation (4.19).

The PPEOS-related routines that we use to produce the results in section 4.4 are publicly available at [26, 30], along with a new version of `IllinoisGRMHD` that supports them. The `IllinoisGRMHD` version available in [30] has been also modified to make use of the `NRPy+` infrastructure, and validated against a trusted version.

We illustrate how well a piecewise polytropic EOS represents its tabulated EOS cousin in figure 4.1 by plotting $P_{\rm cold}(\rho)$ and $\epsilon_{\rm cold}(\rho)$ for both of them. We choose the SLy4 EOS [97], as described by [31] (PPEOS) and [32] (tabulated EOS). Since the code units conventions differ for each type of EOS, we plot them using physical (cgs) units. We note that to generate the plots of P and ϵ as a function of density alone, we have fixed the temperature at $T = 0.1$ MeV and have determined the appropriate value of Y_e imposing neutrino free beta-equilibrium, which means finding a value of Y_e, for a given pair (ρ, T), such that

$$\mu_\nu \left(\rho, T, Y_e\right) = 0 \ . \tag{4.27}$$

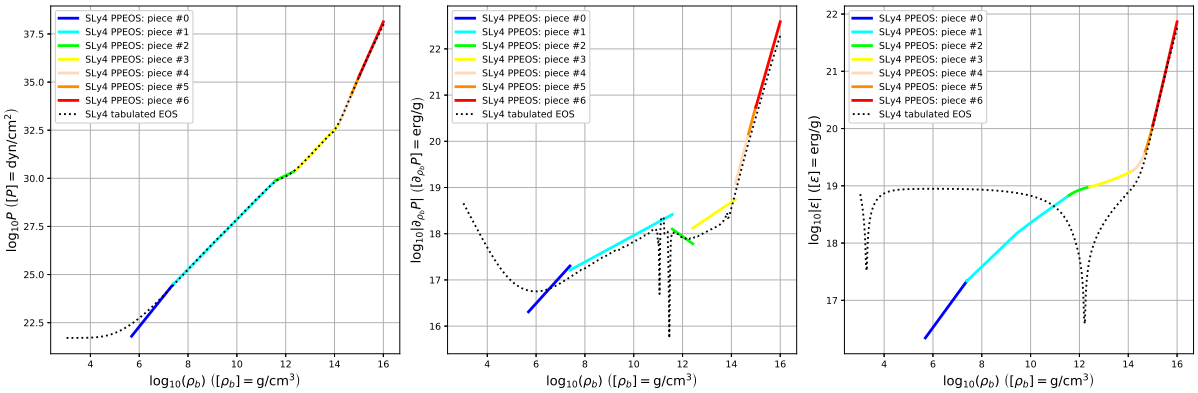


Figure 4.1: Pressure vs density (left), first derivative of the pressure vs density (center) and specific internal energy vs density (right) for the SLy4 PPEOS of [31] and the SLy4 tabulated EOS of [32]. We note that the pressure-density relation of the PPEOS is a very good approximation of the tabulated, particularly at high densities. The energy density also matches its tabulated EOS cousin for high densities, but the agreement for low densities is very poor. The discontinuities in the first derivative of the pressure, with respect to the density, are also evident from the figure.

4.4 Code validation

4.4.1 The Tolman-Oppenheimer-Volkoff equations

We will now perform an initial validation of our implementation of the functions that handle tabulated EOS and PPEOS by solving the Tolman-Oppenheimer-Volkoff (TOV) equations [98–100]. We start from the line element

$$ds^2 = -e^{\nu}dt^2 + \left(1 - \frac{2m}{r}\right)^{-1} dr^2 + r^2 d\Omega^2 \ , \tag{4.28}$$

or, using the isotropic radius

$$\bar{r} \equiv e^{-2\phi} r \ , \tag{4.29}$$

we have

$$ds^2 = -e^{\nu}dt^2 + e^{4\phi}\left(d\bar{r}^2 + \bar{r}^2 d\Omega^2\right) \ , \tag{4.30}$$

with ϕ the conformal factor of the BSSN equations and ν defined through the constraint equation below. In terms of these variables the TOV equations can then be written as

$$\frac{dP}{dr} = -\frac{1}{r}\left(\frac{\rho + P}{2}\right)\left(\frac{2m}{r} + 8\pi r^2 P\right)\left(1 - \frac{2m}{r}\right)^{-1} , \tag{4.31}$$

$$\frac{d\nu}{dr} = \frac{1}{r}\left(1 - \frac{2m}{r}\right)^{-1}\left(\frac{2m}{r} + 8\pi r^2 P\right) , \tag{4.32}$$

$$\frac{dm}{dr} = 4\pi r^2 \rho , \tag{4.33}$$

$$\frac{d\bar{r}}{dr} = \left(1 - \frac{2m}{r}\right)^{-1/2}\frac{\bar{r}}{r} . \tag{4.34}$$

Solutions of the TOV equations are static stars of mass M and radius R. To integrate the TOV equations, we use a simple fourth-order Runge-Kutta routine. An initial condition is given in the form of the density at the center of the star, $\rho_{\rm central}$, from which the central pressurre, $P_{\rm central} \equiv P(\rho_{\rm central})$ is computed. Note that for hybrid polytrope-based EOSs, the star is assumed to be cold (i.e. we neglect the thermal contribution to the pressure), so when considering tabulated EOSs we set the temperature to a small value as well, typically to $T = 0.1$ MeV, as mentioned in the previous section. We also assume neutrino free beta-equilibrium.

Integration of the TOV equations happen along the radial direction until some stop criterion is met. One may impose, for example, an *ad hoc* value for the density at the surface of the star, $\rho_{\rm surface}$, and stop integrating the equations whenever $\rho = \rho_{\rm surface}$. Alternatively, one may stop integration once a sudden drop in pressure occurs, indicating that we have reached the surface of the star and are now in vacuum. In `NRPy+`'s TOV solver we opt for the latter.

One may have the reasonable expectation that increasing the central density $\rho_{\rm central}$ will result in a more massive star, but that is not always true. For a given EOS, one will reach a critical value of the central density which results in the most massive star that can be obtained using that particular EOS. Increasing the central density beyond the critical value will result in stars which are less massive and unstable.

Bounds to the masses of NSs are determined experimentally by observing, e.g., pulsars. The most massive NS to date was observed in pulsar PSR J0740+6620 [101] and has a mass of $2.14^{+0.10}_{-0.09} M_\odot$ (to 1σ; $2.14^{+0.20}_{-0.19} M_\odot$ to 2σ). An EOS that is capable of modeling NSs must therefore be able to produce NSs with masses $\gtrsim 2M_\odot$.

Formalizing the description above, one selects an EOS and specifies a central density

ρ_{central}. One then solves the TOV equations and determines the star's mass, $M(\rho_{\text{central}})$, and radius, $R(\rho_{\text{central}})$, both as a function of the initial central density. One then increases the central density by a factor $\delta\rho$ and solves the TOV equations again, obtaining the new values $\left\{M(\rho_{\text{central}}+\delta\rho), R(\rho_{\text{central}}+\delta\rho)\right\}$. We repeat these steps until we find the critical central density, ρ_{crit}, for which $M(\rho_{\text{crit}}+\delta\rho) < M(\rho_{\text{crit}})$, and conclude that $M(\rho_{\text{crit}}) = M_{\text{max}}$.

We have selected the PPEOSs in [31] which are able to produce stars with masses $\gtrsim 2M_{\odot}$, as well as a particular SRO tabulated EOS, SLy4, to demonstrate this behaviour. We integrate the TOV equations many times for each of the EOSs selected until we obtain the maximum mass for that EOS.[7] In figure 4.2 we present these results, as well as the mass range of PSR J0740+6620. We note how well the PPEOS SLy4 EOS matches the results of its tabulated version.

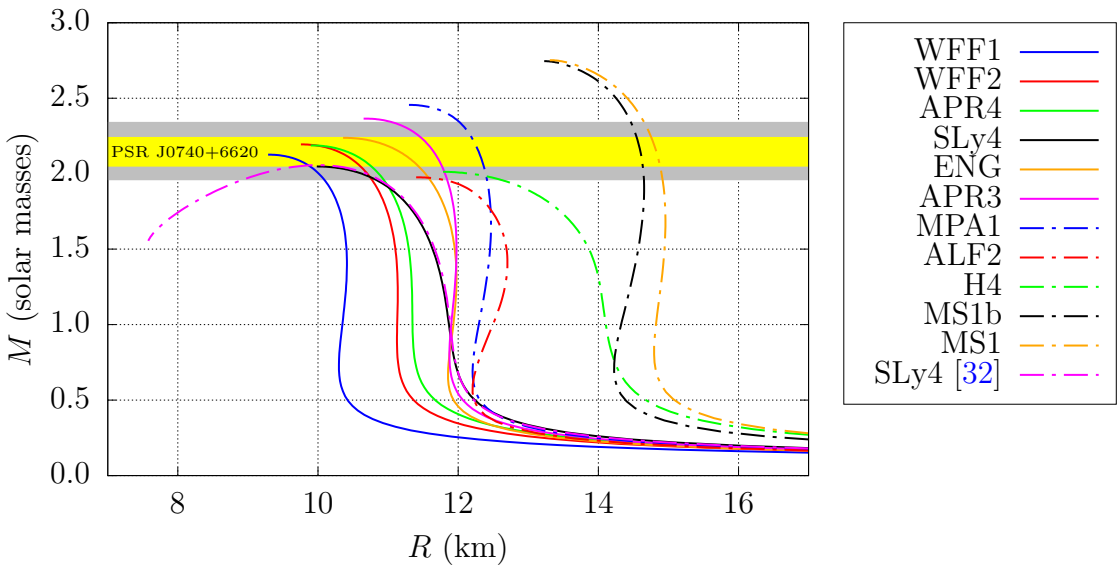


Figure 4.2: NSs mass-radius relations obtained by solving the TOV equations for the PPEOSs in [31] which are able to produce neutron stars with $M_{\text{NS}} \gtrsim 2M_{\odot}$. We have also included the SLy4 EOS from [32], with $T = 0.1$ MeV in neutrino free beta-equilibrium, to illustrate how well its PPEOS cousin matches its results. The yellow/gray strip represents the mass of PSR J0740+6620 with 68.3%/95.4% credibility [101]. This figure reproduces results from [32, 102], and is a robust validation test of the low level functions used to handle PPEOSs and tabulated EOSs.

[7] This criterion was not enforced for the tabulated EOS used to generate figure 4.2, where instead we kept solving the TOV equations until the central density was out of table bounds.

4.4.2 The "hydro without hydro" test

Having validated our low level functions by solving the TOV equations, we performed yet another validation test which indicates how well our TOV solution satisfies Einstein's field equations: the "hydro without hydro" test [103]. The hydro without hydro test was originally devised to demonstrate the stability of the BSSN equations, by using it to evolve static solutions of Einstein's equations (i.e. solutions where $\partial_t \gamma_{ij} = 0$). Because the solution is static, the matter variables should remain at their initial values throughout the run. Therefore, by injecting the initial value of the hydrodynamic variables into the source terms of the BSSN equations, we do not need to solve the GRHD equations at every time step.

The original results in [103] were obtained by performing this test only for a simple polytropic EOS, and we repeat their analysis here as a "control test". We then perform tests using more advanced EOSs, namely PPEOS and tabulated EOS.

We focus here on TOV initial data. The TOV solution is a static solution, and therefore we can use equation (2.58) to show that

$$\partial_t \gamma_{ij} = 0 = -2\alpha K_{ij} + D_i \beta_j + D_j \beta_i \implies K_{ij} = \frac{1}{2\alpha} \left(D_i \beta_j + D_j \beta_i \right) \ . \tag{4.35}$$

If we adopt a frozen shift condition,

$$\beta^i = 0 = B^i \ , \tag{4.36}$$

and take the trace of equation (4.35), we find that

$$K = 0 \ . \tag{4.37}$$

Thus with this gauge condition, the mean curvature $K = \gamma^{ij} K_{ij}$ is analytically zero for a TOV star. As a diagnostic of how well our numerical run satisfies this analytical constraint, we will keep track of the central value of the mean curvature, $K_{\rm central}$, i.e. the value of K at the very center of the star.

We also choose the harmonic slicing condition for the lapse (see eq. (32) in [16])

$$\partial_t \alpha = \partial_t \left(e^{6\phi} \right) = 6 e^{6\phi} \partial_t \phi \ . \tag{4.38}$$

We note here that we perform this test using `NRPy+` to discretize and evolve the BSSN equations, and emphasize the fact that this test provides validation for routines implemented in both `NRPy+` and `IllinoisGRMHD`, since the `IllinoisGRMHD` routines are `C/C++` adaptations of the `Python` routines implemented in `NRPy+`.

For all the tests performed below, we evolve the BSSN variables in time, keeping the matter quantities fixed at their initial values, up to $t/M = 3000$, using SinhSpherical coordinates to push the outer boundary to $r_{\rm max} \sim 3000M$, while still maintaining about 30 points along a diameter of the star. We integrate in time using a standard RK4 integrator and the method of lines, while spatial discretization of the BSSN equations uses fourth-order accurate, centered finite differences. We use a 3d grid with 96 points in the radial direction, 16 points in θ and 2 points in φ. We impose axisymmetry by setting all derivatives along the φ-direction to zero. We again use Kreiss-Oliger dissipation [87], as in equation (3.69), with dissipation strength $\sigma = 0.1$.

The star is assumed to be "cold", so that for the polytrope based hybrid EOSs we assume $P = P_{\rm cold}$, and for tabulated EOSs we assume $T = 0.1$ MeV. Initial data for the TOV star is obtained by solving the TOV equations (4.31–4.34), tuning the central density of the star so that it produces a star with the mass and radius we are interested in studying.

For the control test, we consider a simple polytrope EOS with $\kappa = 1$ and $\Gamma = 2$. We choose $\rho_{\rm central} = 0.2$, producing a star with mass $M \approx 0.157$ and radius $R \approx 0.866$, with isotropic radius $\bar{R} \approx 0.700$, exactly as in [103]. In figure 4.3 we show $K_{\rm central}$ as a function of time for this control test. We notice that some numerical noise develops quite early in the run, which is a result of both the finite differences errors and noise at the surface of the star. To avoid an extra source of noise, which are a result of imposing outer boundary conditions at finite distances, we have pushed the outer boundary out of causal contact with the center of the numerical grid. We emphasize here that the evolution is stable to the point that we can keep it running indefinetely.

We track the numerical errors of the evolution by looking at the Hamiltonian constraint (2.82). Since $\mathcal{H} = 0$ analytically, we look at violations of this result as another independent diagnostic of how well our solution satisfies Einstein's equations. Figure 4.4

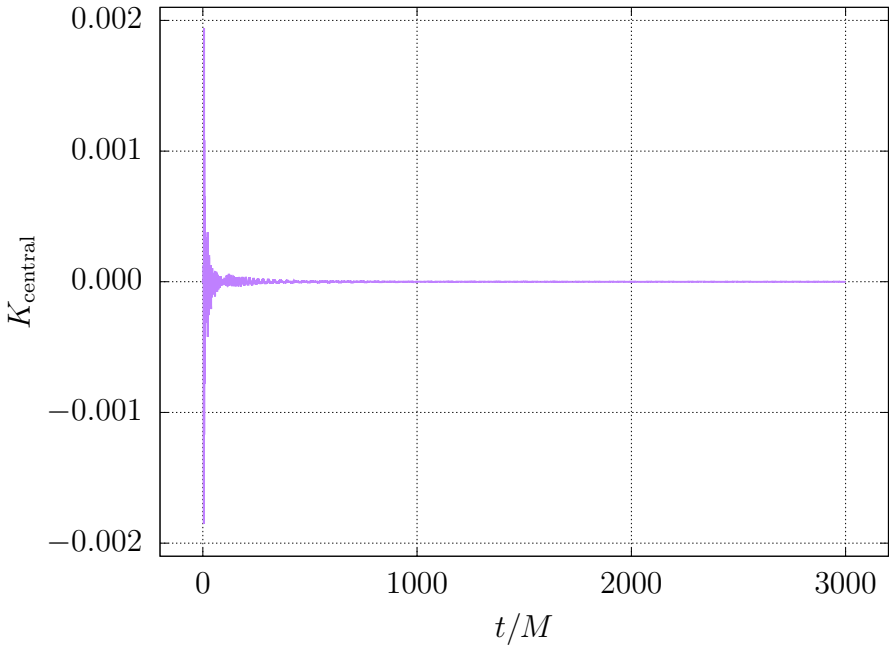
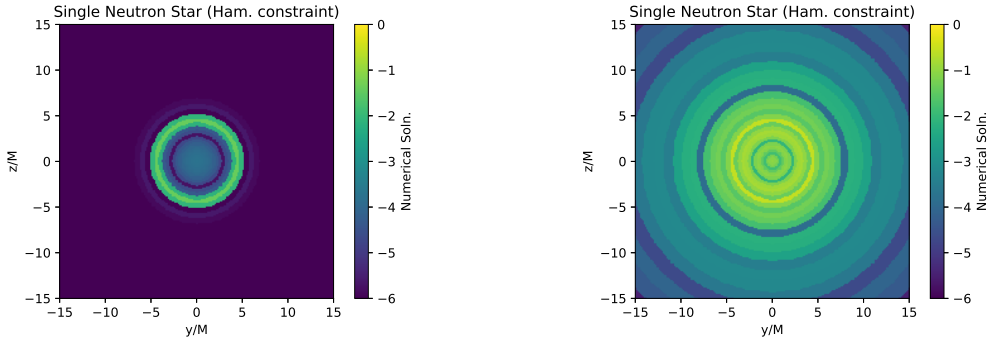


Figure 4.3: "Hydro without hydro" control test results. We expect the value of K to be zero throughout the evolution, and which it is after some initial numerical noise disperses (more in the text).

shows the violation of the Hamiltonian constraint at $t/M = 0$ (i.e. the initial data) and $t/M = 3000$. We note that the violations are larger around the surface of the NS, which is expected considering there is a sudden drop of many orders of magnitude in the magnitude of the pressure and the density.



(a) $t/M = 0$. (b) $t/M = 3000$.

Figure 4.4: Hamiltonian constraint violation for the evolution of a OV static star initial data using a simple polytrope EOS. The plots show a slice of the yz-plane and the colors represent $\log(\mathcal{H})$.

We now want to perform tests that compare simple and piecewise polytrope based EOSs, as well as tabulated EOSs. We choose the SLy4 in its PPEOS version [31] and tabulated EOS version [32] to perform the tests. Since simple polytropes are scale free,

we adjust its parameters by keeping $\Gamma = 2$ but determining a new value of the constant κ, $\kappa_{\rm new}$, such that its nuclear pressure-density relation matches that of SLy4. This means that

$$\kappa_{\rm new} \equiv \frac{P_{\rm nuc}^{\rm SLy4}}{\left(\rho_{\rm nuc}^{\rm SLy4}\right)^{\Gamma}} \ . \tag{4.39}$$

where we fix $\rho_{\rm nuc}^{\rm SLy4} \equiv 1 \times 10^{15}$ g/cm^3 and

$$P_{\rm nuc}^{\rm SLy4} = 1.85 \times 10^{35} \ {\rm dyn/cm^2} \ , \tag{4.40}$$

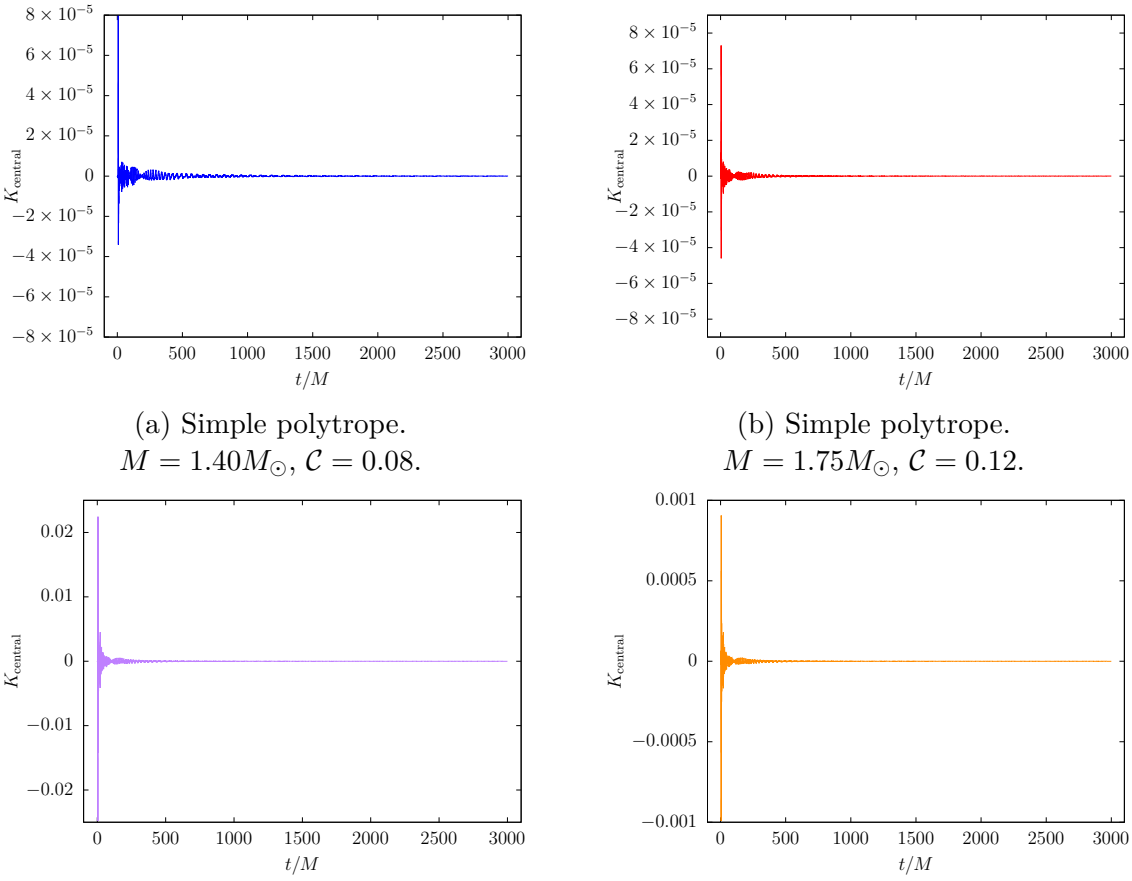
using the tabulated version of the SLy4 at $T = 0.1$ MeV and neutrino free beta-equilibrium.

For the next battery of tests, we set the central density values in order to obtain the stars described in table 4.2. For the SLy4 EOS, we perform the test for a star with $M = 1.40 M_\odot$ and $R = 11.7$ km. The simple polytrope with $\Gamma = 2$ and $\kappa_{\rm new}$ is, however, still quite different from the SLy4 EOS, and thus we cannot produce a star with same mass *and* radius we had for the SLy4 stars. We thus consider, also, the compactification of the star, $\mathcal{C} \equiv M/R$, and perform two separate runs: an equal mass, but lower compactification, star and an equal compactification, but higher mass, star. This allows us to make "apples-to-apples" comparisons between the different EOSs. The behaviour of the central mean curvature, $K_{\rm central}$, for these tests is presented in figure 4.5, while Hamiltonian constraint violations are shown in figure 4.6.

EOS type	**EOS name**	$\boldsymbol{M/M_\odot}$	$\boldsymbol{R}$ **(km)**	$\boldsymbol{\mathcal{C} \equiv M/R}$
Simple polytrope	$\Gamma = 2$	1.40	16.9	0.08
Simple polytrope	$\Gamma = 2$	1.75	14.7	0.12
Piecewise polytrope	SLy4 [31]	1.40	11.7	0.12
Tabulated	SLy4 [32]	1.40	11.7	0.12

Table 4.2: Characteristic of the different TOV stars used in the hydro without hydro test.

From the results of the test, particularly the ones shown in figure 4.6, it is clear that the tabulated version of the SLy4 EOS provides data that better satisfies Einstein's field equations. Of course, one may argue that by looking at the errors alone the simple polytropes satisfy Einstein's field equations even better than tabulated EOS. This is to be expected, since the simple polytrope provides a simple power law pressure-density



(a) Simple polytrope. $M = 1.40M_\odot$, $\mathcal{C} = 0.08$.

(b) Simple polytrope. $M = 1.75M_\odot$, $\mathcal{C} = 0.12$.

(c) PPEOS SLy4. $M = 1.40M_\odot$, $\mathcal{C} = 0.12$.

(d) Tabulated SLy4. $M = 1.40M_\odot$, $\mathcal{C} = 0.12$.

Figure 4.5: Central mean curvature, K_{central}, as a function of time for a TOV initial data using the hydro without hydro test. The subfigure (d) shows the result for the tabulated EOS SLy4, while the subfigure (c) shows the result for its PPEOS approximation, both for a star of $M = 1.4M_\odot$ and $R = 11.7$ km. With the simple polytrope EOS that matches the SLy4 pressure-density relation at nuclear densities, a $1.4M_\odot$ star does not have the same compactification as the SLy4 star. Thus we show results for this star, as well as for a star with higher mass, $1.75M_\odot$, but same compactification as the SLy4 stars considered in the test.

relation and is therefore quite smooth. However, it provides an objectivelly less realistic description of a NS, and therefore the fact that it solves Einstein's equations does not mean it does so with the same physical significance as the tabulated EOS solution.

4.5 Binary neutron stars

We now briefly discuss results we have obtained from a SLy4 PPEOS binary neutron star (BNS) simulation. These results will be used as part of a code comparison project and will result in a publication which is currently under preparation. The main goal here is

to show that multiple codes available in the Numerical Relativity community are able to produce reliable results.

Our run consists of solving the GRHD equations in a dynamical spacetime, using the BSSN formalism and the conservative scheme discussed in section 2.6. Note that we are solving the GRHD equations, not the GRMHD equations, which means that we set all magnetic fields to zero for this particular run.

We perform this run using the Einstein Toolkit [104, 105]. The Einstein Toolkit is built using the Cactus framework [106, 107], which is itself composed of multiple modules called "thorns". To mention a few of the most useful thorns, we have adaptive mesh refiment (AMR) via the Carpet driver [108, 109], time integration using the method of lines is provided by the MoL thorn [104, 105], aparent horizon[8] finding routines are provided by the AHFinderDirect thorn [111, 112], useful volume integrals, such as the ADM mass and angular momentum, can be computed using the QuasiLocalMeasures thorn [113, 114], and the multipole expansion of the real and imaginary parts of ψ_4 in spin-weighted spherical harmonics [115] is performed using the Multipole thorn [104, 105]. `IllinoisGRMHD` itself is a thorn of the Einstein Toolkit, and provides the necessary algorithms to evaluate the right-hand sides of the GRMHD equations, so that the GRMHD variables can be evolved together with the BSSN variables using the MoL thorn.

The initial data for our BNS run is setup using `LORENE` [116–122], and consists of two NSs with $M = 1.5M_\odot$ each and seperated initially by 50 km. The initial ADM mass of the system is $M_{\rm ADM} = 2.68M_\odot$. The inspiral phase of the the two stars consists of almost 10 complete orbits before they finally merge and give rise to a hypermassive neutron star (HMNS). The inspiral phase is depicted in figure 4.7. After oscillating for quite some time, the HMNS finally collapses into a black hole.[9]

Gravitational waves for the simulation are extracted by looking at the scalar

$$\psi_4 \equiv -C_{\mu\nu\rho\sigma} k^\mu \bar{m}^\nu k^\rho \bar{m}^\sigma \ , \tag{4.41}$$

where $C_{\mu\nu\rho\sigma}$ is the Weyl tensor, and l^μ and k^μ are real, radially outgoing and ingoing null vectors, respectively. m^μ is a complex vector constructed from two spatial vectors

[8] See e.g. [110].

[9] A neat movie of the results described here can be found in [123], more specifically here [124] (this second reference is subject to change, but you can still find the movie from the first reference in this footnote by searching for the "IllinoisGRMHD BNS simulation; hybrid EOS with initially cold piecewise-polytrope SLy EOS" movie).

that are orthogonal to l^μ and k^μ in such a way that the only nonvanishing inner products between these 4-vectors, or *null tetrad*, are

$$-l^\mu k_\mu = 1 = m^\mu \bar{m}_\mu \ , \tag{4.42}$$

and $\bar{m}^\mu$ is the complex conjugate of m^μ.

The scalar ψ_4, which is one of the five scalars that can be constructed from the Weyl tensor, corresponds to outgoing gravitational waves. For practical applications, it is often useful to decompose it into $s = -2$ spin-weighted spherical harmonics [115] (see also [48] for a briefer introduction)

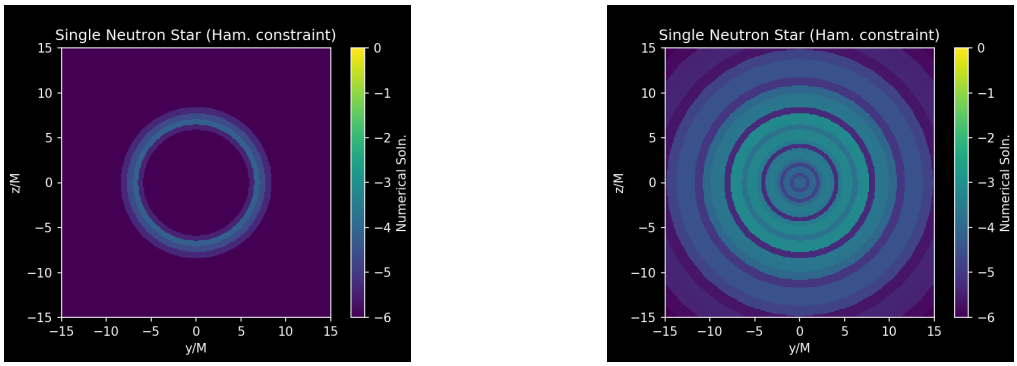
$$\psi_t(t, r, \theta, \varphi) = \sum_{l=2}^{\infty} \sum_{m=-l}^{l} \psi_4^{l,m}(t, r) {}_{-2}Y^*_{lm}(\theta, \varphi) \ . \tag{4.43}$$

The coefficients $\psi_4^{l,m}$ are computed using the Multipole thorn, and are extracted at multiple extraction radii $R_{\rm ext}$. The amplitude of the real part of these coefficients, in particular of the dominant $l = 2$, $m = 2$ mode, is expected to behave as

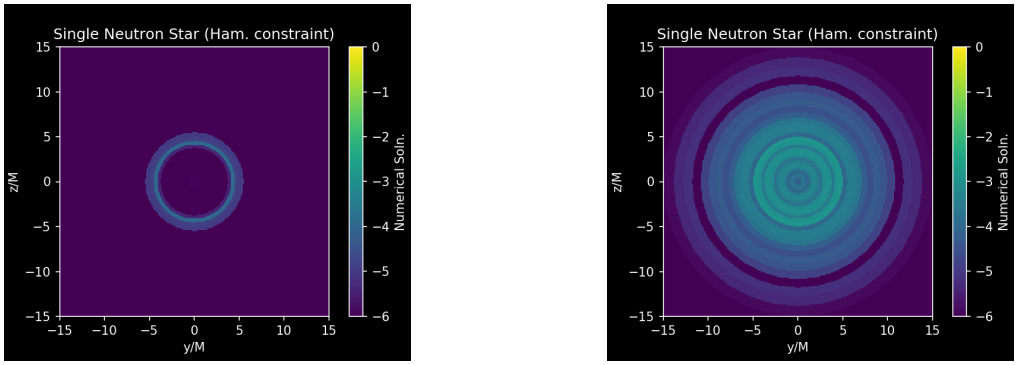
$$\mathrm{Re}\left(\psi_4^{2,2}\right) \sim \frac{1}{r} \ , \tag{4.44}$$

for large r, and we can test this falloff behaviour by performing the extraction at different radii. In figure 4.8 we show $R_{\rm ext}\mathrm{Re}\left(\psi_4^{2,2}\right)$ as a function of time at extraction radii $R_{\rm ext} = 500$ and $R_{\rm ext} = 850$, showing that the falloff behaviour is indeed satisfied. The figure also depicts quite clearly the different regimes of the BNS run: the long inspiral, followed by the formation of the HMNS with high frequency oscillations, and finally its collapse into a black hole.

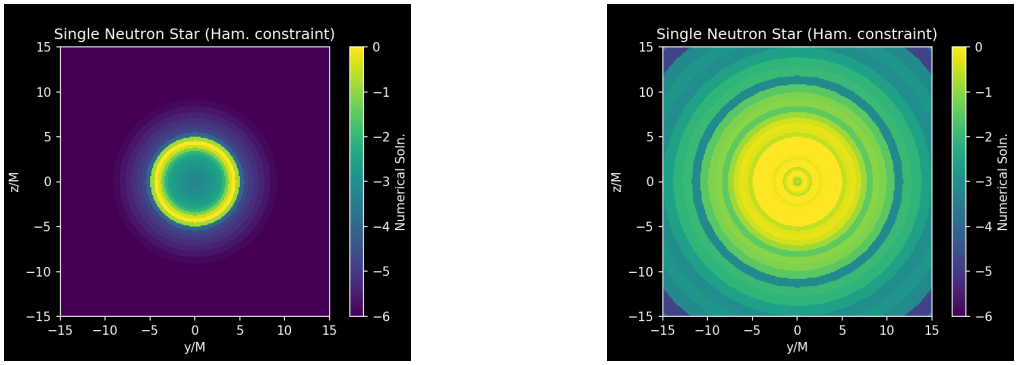
We conclude this section by emphasizing that this serves as a very stringent test of the PPEOS low level functions that have been implemented into `IllinoisGRMHD`. A publication that compares these results with the ones obtained by other trusted codes in the Numerical Relativity community is currently under preparation.



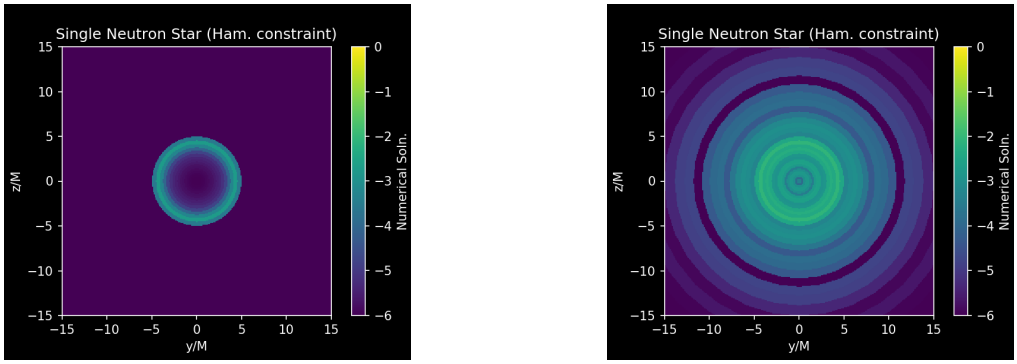
(a) Simple polytrope, $M = 1.40M_\odot$, $\mathcal{C} = 0.08$, initial (left, $t = 0$) and final (right, $t = 3000M$) Hamiltonian constraint violation.



(b) Simple polytrope, $M = 1.75M_\odot$, $\mathcal{C} = 0.12$, initial (left, $t = 0$) and final (right, $t = 3000M$) Hamiltonian constraint violation.



(c) PPEOS SLy4, $M = 1.40M_\odot$, $\mathcal{C} = 0.12$, initial (left, $t = 0$) and final (right, $t = 3000M$) Hamiltonian constraint violation.



(d) Tabulated SLy4, $M = 1.40M_\odot$, $\mathcal{C} = 0.12$, initial (left, $t = 0$) and final (right, $t = 3000M$) Hamiltonian constraint violation.

Figure 4.6: Hydro without hydro test: Hamiltonian constraint violation.

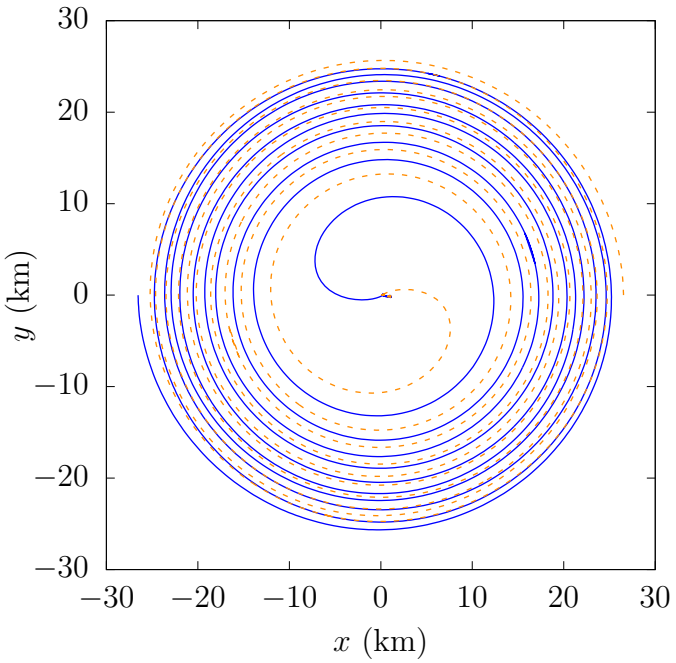


Figure 4.7: Inspiral trajectory for a SLy4 PPEOS, equal mass BNS merger. Each star has $M = 1.5M_\odot$, the initial separation is 50 km, and the total ADM mass of the system is $M_{\rm ADM} = 2.68M_\odot$.

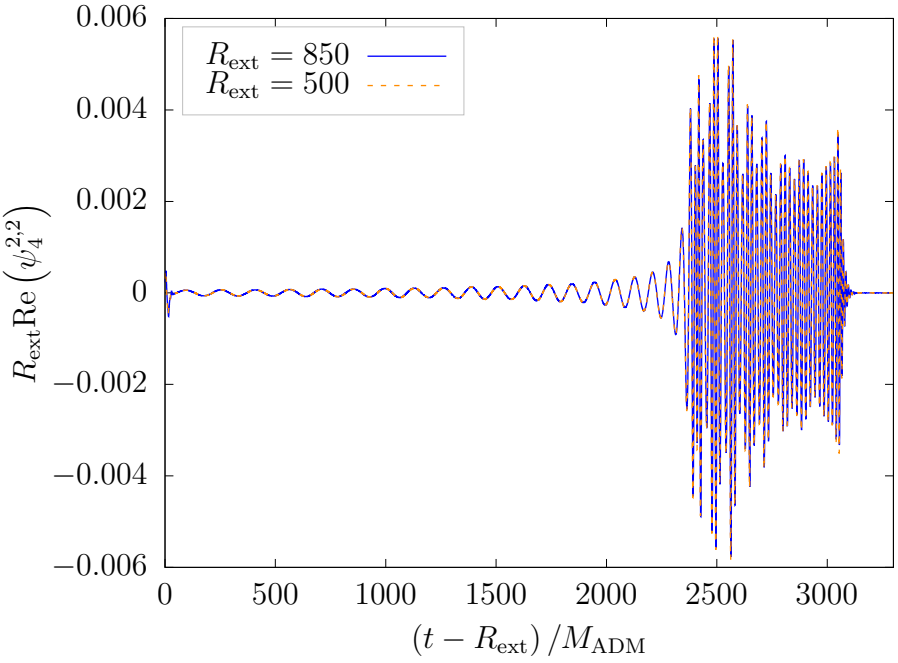


Figure 4.8: Dominant outgoing gravitational wave mode, $\psi_4^{2,2}$, for a SLy4 PPEOS, equal mass BNS merger, at different extraction radius. The wave amplitude and frequency increase as the stars get closer to each other. Once they merge, a hypermassive NS forms around $(t - R_{\rm ext})/M_{\rm ADM} \approx 2300$, and oscillates for quite some time before collapsing into a black hole at $(t - R_{\rm ext})/M_{\rm ADM} \approx 3100$.

Chapter 5

Conclusions

In this thesis we have discussed our results when studying aspects of Numerical Relativity. We have presented an introduction of the theoretical framework used to study Einstein's field equations numerically, as well as the contributions we made to the development of numerical codes that use them.

We have studied the gravitational collapse of the massless scalar field using both the ADM and BSSN formalisms. Our results with the ADM formalism made use of the `SFcollapse1D` code [19], which we have developed based on Matt Choptuik's original studies on the subject [20–22]. To study Einstein's field equations using the BSSN formalism, we have used the `NRPy+` infrastructure [25, 26]. We note that, as far as we are aware, our results with the BSSN formalism have been the first to use the standard moving puncture approach, with a second-order Gamma-driver shift condition, in contrast to other similar studies [79, 80, 86]. Our results for both formalisms are in excellent agreement with the literature, as discussed in chapter 3.

More general cases, where we relax, for example, the spherical symmetry of the problem or an expanding universe, will be the subject of future studies. We also intend to study gravitational collapse of different matter sources other than the massless scalar field.

We have also made significant contributions to the `IllinoisGRMHD` code [28, 29], most notably by thoroughly documenting the code in pedagogical `Jupyter` notebooks [92] and extending its equation of state (EOS) support to cold piecewise polytrope-based hybrid EOSs (PPEOS). We have also made use of `NRPy+` to generate a new and improved version of the code [30]. Work is ongoing for the addition of tabulated EOS support and

neutrino physics to `IllinoisGRMHD`. `IllinoisGRMHD` is one of the main codes in the TCAN collaboration [33], and will be used to study multi-messenger astrophysics problems along with `HARM3D` [93].

Our implementation of PPEOS low level functions has been fully validated. In particular, we have performed a full binary neutron star (BNS) run that considered a SLy4 PPEOS from [31] and the results obtained, which will be the subject of a publication which is currently under preparation, have been discussed in section 4.5. Further compact binary simulations, in particular BNSs, will be performed in the near future using `IllinoisGRMHD`.

The implementation of tabulated EOS low level functions has been partially validated and the results of these validations have been discussed in section 4.4. As far as we are aware, ours has been the first hydro without hydro test that has considered advanced EOSs. A tabulated version of the BNS simulation discussed in section 4.5, which will also include neutrino physics, will likely be the subject of a publication in the near future.

References

This page intentionally left blank.

References

[1] Benjamin P Abbott et al. "Observation of gravitational waves from a binary black hole merger". In: *Physical review letters* 116.6 (2016), p. 061102.

[2] Benjamin P Abbott et al. "GW151226: Observation of gravitational waves from a 22-solar-mass binary black hole coalescence". In: *Physical review letters* 116.24 (2016), p. 241103.

[3] LIGO Scientific et al. "GW170104: observation of a 50-solar-mass binary black hole coalescence at redshift 0.2". In: *Physical Review Letters* 118.22 (2017), p. 221101.

[4] Benjamin P Abbott et al. "GW170814: a three-detector observation of gravitational waves from a binary black hole coalescence". In: *Physical review letters* 119.14 (2017), p. 141101.

[5] Benjamin P Abbott et al. "GW170608: Observation of a 19 solar-mass binary black hole coalescence". In: *The Astrophysical Journal Letters* 851.2 (2017), p. L35.

[6] B. P. Abbott et al. "GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs". In: (2018). arXiv: 1811.12907 [astro-ph.HE].

[7] LIGO Scientific Collaboration, Virgo Collaboration, et al. "GW190412: Observation of a Binary-Black-Hole Coalescence with Asymmetric Masses". In: *arXiv preprint arXiv:2004.08342* (2020).

[8] Benjamin P Abbott et al. "Multi-messenger observations of a binary neutron star merger". In: *Astrophys. J. Lett* 848.2 (2017), p. L12.

[9] Benjamin P Abbott et al. "GW170817: observation of gravitational waves from a binary neutron star inspiral". In: *Physical Review Letters* 119.16 (2017), p. 161101.

[10] BP Abbott et al. "GW190425: Observation of a compact binary coalescence with total mass 3.4 M ". In: *The Astrophysical Journal Letters* 892.1 (2020), p. L3.

[11] R Abbott et al. "GW190814: Gravitational waves from the coalescence of a 23 solar mass black hole with a 2.6 solar mass compact object". In: *The Astrophysical Journal Letters* 896.2 (2020), p. L44.

[12] Richard L. Arnowitt, Stanley Deser, and Charles W. Misner. "Dynamical Structure and Definition of Energy in General Relativity". In: *Phys. Rev.* 116 (1959), pp. 1322–1330. DOI: 10.1103/PhysRev.116.1322.

[13] Miguel Alcubierre. *Introduction to 3+ 1 numerical relativity*. Vol. 140. Oxford University Press, 2008.

[14] Takashi Nakamura, Kenichi Oohara, and Yasufumi Kojima. "General relativistic collapse to black holes and gravitational waves from black holes". In: *Progress of Theoretical Physics Supplement* 90 (1987), pp. 1–218.

[15] Masaru Shibata and Takashi Nakamura. "Evolution of three-dimensional gravitational waves: Harmonic slicing case". In: *Physical Review D* 52.10 (1995), p. 5428.

[16] Thomas W Baumgarte and Stuart L Shapiro. "Numerical integration of Einstein's field equations". In: *Physical Review D* 59.2 (1998), p. 024007.

[17] John G. Baker et al. "Gravitational wave extraction from an inspiraling configuration of merging black holes". In: *Phys. Rev. Lett.* 96 (2006), p. 111102. DOI: 10.1103/PhysRevLett.96.111102. arXiv: gr-qc/0511103 [gr-qc].

[18] Masaru Shibata and Kōji Uryū. "Gravitational Waves from the Merger of Binary Neutron Stars in a Fully General Relativistic Simulation". In: *Progress of Theoretical Physics* 107.2 (Feb. 2002), pp. 265–303. ISSN: 0033-068X. DOI: 10.1143/PTP.107.265. eprint: http://oup.prod.sis.lan/ptp/article-pdf/107/2/265/5313264/107-2-265.pdf. URL: https://dx.doi.org/10.1143/PTP.107.265.

[19] *SFcollapse1D github repository*. URL: https://github.com/leowerneck/SFcollapse1D.

[20] Matthew W. Choptuik. "Universality and scaling in gravitational collapse of a massless scalar field". In: *Phys. Rev. Lett.* 70 (1 Jan. 1993), pp. 9–12. DOI: 10.1103/PhysRevLett.70.9. URL: https://link.aps.org/doi/10.1103/PhysRevLett.70.9.

[21] M. W. Choptuik. ""Critical" behaviour in massless scalar field collapse". In: *Approaches to Numerical Relativity*. Ed. by Ray Editor d'Inverno. Cambridge University Press, 1992, pp. 202–222. DOI: 10.1017/CBO9780511524639.019.

[22] M. W. Choptuik. "Critical Behaviour in Scalar Field Collapse". In: *Deterministic Chaos in General Relativity*. Ed. by David Hobill, Adrian Burd, and Alan Coley. Boston, MA: Springer US, 1994, pp. 155–175. ISBN: 978-1-4757-9993-4. DOI: 10.1007/978-1-4757-9993-4_10. URL: https://doi.org/10.1007/978-1-4757-9993-4_10.

[23] Marsha J Berger and Joseph Oliger. "Adaptive mesh refinement for hyperbolic partial differential equations". In: *Journal of computational Physics* 53.3 (1984), pp. 484–512.

[24] The GNU Public License version 3. URL: https://www.gnu.org/licenses/gpl-3.0.txt.

[25] Ian Ruchlin, Zachariah B. Etienne, and Thomas W. Baumgarte. "SENR/NRPy+: Numerical relativity in singular curvilinear coordinate systems". In: *Physical Review D* 97.6 (Mar. 2018). ISSN: 2470-0029. DOI: 10.1103/physrevd.97.064036. URL: http://dx.doi.org/10.1103/PhysRevD.97.064036.

[26] *NRPy+ github repository*. URL: https://github.com/zachetienne/nrpytutorial/.

[27] The 2-Clause BSD License. URL: https://opensource.org/licenses/BSD-2-Clause.

[28] Zachariah B Etienne et al. "IllinoisGRMHD: an open-source, user-friendly GRMHD code for dynamical spacetimes". In: *Classical and Quantum Gravity* 32.17 (Aug. 2015), p. 175009. ISSN: 1361-6382. DOI: 10.1088/0264-9381/32/17/175009. URL: http://dx.doi.org/10.1088/0264-9381/32/17/175009.

[29] *IllinoisGRMHD github repository*. URL: https://github.com/zachetienne/nrpytutorial/tree/master/IllinoisGRMHD.

[30] *NRPy+IllinoisGRMHD github repository*. URL: https://github.com/leowerneck/NRPyIGM.

[31] Jocelyn S. Read et al. "Constraints on a phenomenologically parametrized neutron-star equation of state". In: *Physical Review D* 79.12 (June 2009). ISSN: 1550-2368. DOI: 10.1103/physrevd.79.124032. URL: http://dx.doi.org/10.1103/PhysRevD.79.124032.

[32] 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". In: *Physical Review C* 96.6 (Dec. 2017). ISSN: 2469-9993. DOI: 10.1103/physrevc.96.065802. URL: http://dx.doi.org/10.1103/PhysRevC.96.065802.

[33] *Compact Binaries website*. URL: https://compact-binaries.org/.

[34] K. Schwarzschild. "Uber das Gravitationsfeld eines Massenpunktes nach der Einstein'schen Theorie". In: *Berlin. Sitzungsberichte* 18 (1916). URL: https://ci.nii.ac.jp/naid/10028154624/en/.

[35] Roy P. Kerr. "Gravitational Field of a Spinning Mass as an Example of Algebraically Special Metrics". In: *Phys. Rev. Lett.* 11 (5 Sept. 1963), pp. 237–238. DOI: 10.1103/PhysRevLett.11.237. URL: https://link.aps.org/doi/10.1103/PhysRevLett.11.237.

[36] Hans Reissner. "Über die Eigengravitation des elektrischen Feldes nach der Einsteinschen Theorie". In: *Annalen der Physik* 355.9 (1916), pp. 106–120.

[37] Hermann Weyl. "Zur gravitationstheorie". In: *Annalen der Physik* 359.18 (1917), pp. 117–145.

[38] G Nordstrom. "On the energy of gravitation field in Einstein's theory". In: *Proc. Ned. Ac. Wet.* 20 (1918), pp. 1238–1245.

[39] George Barker Jeffery. "The field of an electron on Einstein's theory of gravitation". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 99.697 (1921), pp. 123–134.

[40] Alexander Friedman. "Über die krümmung des raumes". In: *Zeitschrift für Physik* 10.1 (1922), pp. 377–386.

[41] Alexander Friedmann. "Über die Möglichkeit einer Welt mit konstanter negativer Krümmung des Raumes". In: *Zeitschrift für Physik* 21.1 (1924), pp. 326–332.

[42] Georges Lemaître. "Expansion of the universe, A homogeneous universe of constant mass and increasing radius accounting for the radial velocity of extra-galactic nebulae". In: *Monthly Notices of the Royal Astronomical Society* 91 (1931), pp. 483–490.

[43] Georges Lemaître. "L'univers en expansion". In: *Publications du Laboratoire d'Astronomie et de Geodesie de l'Universite de Louvain, vol. 10, pp. 1-19* 10 (1937), pp. 1–19.

[44] Howard Percy Robertson. "Kinematics and world-structure". In: *The Astrophysical Journal* 82 (1935), p. 284.

[45] Howard P Robertson. "Kinematics and World-Structure II." In: *The Astrophysical Journal* 83 (1936), p. 187.

[46] Howard P Robertson. "Kinematics and World-Structure III." In: *The Astrophysical Journal* 83 (1936), p. 257.

[47] Arthur Geoffrey Walker. "On Milne's Theory of World-Structure". In: *Proceedings of the London Mathematical Society* 2.1 (1937), pp. 90–127.

[48] Thomas W. Baumgarte and Stuart L. Shapiro. *Numerical Relativity: Solving Einstein's Equations on the Computer*. Cambridge University Press, 2010. DOI: 10.1017/CBO9781139193344.

[49] Heinz-Otto Kreiss and Jens Lorenz. *Initial-boundary value problems and the Navier-Stokes equations*. Vol. 47. Siam, 1989.

[50] Olivier Sarbach et al. "Hyperbolicity of the Baumgarte-Shapiro-Shibata-Nakamura system of Einstein evolution equations". In: *Physical Review D* 66.6 (2002), p. 064002.

[51] Gen Yoneda and Hisa-aki Shinkai. "Advantages of a modified ADM formulation: Constraint propagation analysis of the Baumgarte-Shapiro-Shibata-Nakamura system". In: *Physical Review D* 66.12 (2002), p. 124003.

[52] Carsten Gundlach and Jose M Martin-Garcia. "Hyperbolicity of second order in space systems of evolution equations". In: *Classical and Quantum Gravity* 23.16 (2006), S387.

[53] Carsten Gundlach and Jose M Martin-Garcia. "Well-posedness of formulations of the Einstein equations with dynamical lapse and shift conditions". In: *Physical Review D* 74.2 (2006), p. 024016.

[54] Carles Bona et al. "New formalism for numerical relativity". In: *Physical Review Letters* 75.4 (1995), p. 600.

[55] Miguel Alcubierre et al. "Gauge conditions for long-term numerical black hole evolutions without excision". In: *Physical Review D* 67.8 (2003), p. 084023.

[56] J. David Brown. "Covariant formulations of Baumgarte, Shapiro, Shibata, and Nakamura and the standard gauge". In: *Physical Review D* 79.10 (May 2009). ISSN: 1550-2368. DOI: 10.1103/physrevd.79.104029. URL: http://dx.doi.org/10.1103/PhysRevD.79.104029.

[57] Silvano Bonazzola et al. "Constrained scheme for the Einstein equations based on the Dirac gauge and spherical coordinates". In: *Physical Review D* 70.10 (2004), p. 104007.

[58] Masaru Shibata, Kōji Uryū, and John L Friedman. "Deriving formulations for numerical computation of binary neutron stars in quasicircular orbits". In: *Physical Review D* 70.4 (2004), p. 044044.

[59] Pedro J Montero and Isabel Cordero-Carrión. "BSSN equations in spherical coordinates without regularization: Vacuum and nonvacuum spherically symmetric spacetimes". In: *Physical Review D* 85.12 (2012), p. 124037.

[60] Thomas W Baumgarte et al. "Numerical relativity in spherical polar coordinates: Evolution calculations with the BSSN formulation". In: *Physical Review D* 87.4 (2013), p. 044026.

[61] A. Zee. *Einstein Gravity in a Nutshell*. New Jersey: Princeton University Press, 2013. ISBN: 069114558X, 9780691145587.

[62] Hans Stephani et al. *Exact solutions of Einstein's field equations*. Cambridge Monographs on Mathematical Physics. Cambridge: Cambridge Univ. Press, 2003. ISBN: 9780521467025, 0521467020, 9780511059179, 9780521467025. DOI: 10.1017/

CBO9780511535185. URL: http://www.cambridge.org/uk/catalogue/catalogue.asp?isbn=0521461367.

[63] James R van Meter et al. "How to move a black hole without excision: gauge conditions for the numerical evolution of a moving puncture". In: *Physical Review D* 73.12 (2006), p. 124011.

[64] Sean M. Carroll. *Lecture Notes on General Relativity*. 1997. arXiv: gr-qc/9712019 [gr-qc].

[65] Thomas W. Baumgarte et al. "Numerical relativity in spherical polar coordinates: Evolution calculations with the BSSN formulation". In: *Physical Review D* 87.4 (Feb. 2013). ISSN: 1550-2368. DOI: 10.1103/physrevd.87.044026. URL: http://dx.doi.org/10.1103/PhysRevD.87.044026.

[66] Manuela Campanelli et al. "Accurate evolutions of orbiting black-hole binaries without excision". In: *Physical Review Letters* 96.11 (2006), p. 111101.

[67] John G Baker et al. "Gravitational-wave extraction from an inspiraling configuration of merging black holes". In: *Physical Review Letters* 96.11 (2006), p. 111102.

[68] Matthew D Duez et al. "Relativistic magnetohydrodynamics in dynamical spacetimes: Numerical methods and tests". In: *Physical Review D* 72.2 (2005), p. 024028.

[69] Zachariah B Etienne, Yuk Tung Liu, and Stuart L Shapiro. "Relativistic magnetohydrodynamics in dynamical spacetimes: A new adaptive mesh refinement implementation". In: *Physical Review D* 82.8 (2010), p. 084031.

[70] Zachariah B Etienne et al. "Relativistic magnetohydrodynamics in dynamical spacetimes: Improved electromagnetic gauge condition for adaptive mesh refinement grids". In: *Physical Review D* 85.2 (2012), p. 024013.

[71] Thomas W Baumgarte and Stuart L Shapiro. "General relativistic magnetohydrodynamics for the numerical construction of dynamical spacetimes". In: *The Astrophysical Journal* 585.2 (2003), p. 921.

[72] Brian D Farris et al. "Binary black-hole mergers in magnetized disks: simulations in full general relativity". In: *Physical Review Letters* 109.22 (2012), p. 221102.

[73] Carsten Gundlach. "Critical phenomena in gravitational collapse". In: *Physics Reports* 376.6 (2003), pp. 339–405.

[74] Carsten Gundlach and José M Martín-García. "Critical phenomena in gravitational collapse". In: *Living Reviews in Relativity* 10.1 (2007), p. 5.

[75] Carsten Gundlach. "Understanding critical collapse of a scalar field". In: *Phys. Rev. D* 55 (2 Jan. 1997), pp. 695–713. DOI: 10.1103/PhysRevD.55.695. URL: https://link.aps.org/doi/10.1103/PhysRevD.55.695.

[76] Jose M Martin-Garcia and Carsten Gundlach. "All nonspherical perturbations of the Choptuik spacetime decay". In: *Physical Review D* 59.6 (1999), p. 064031.

[77] Jose M Martin-Garcia and Carsten Gundlach. "Global structure of Choptuik's critical solution in scalar field collapse". In: *Physical Review D* 68.2 (2003), p. 024011.

[78] Shahar Hod and Tsvi Piran. "Fine structure of Choptuik's mass-scaling relation". In: *Physical Review D* 55.2 (1997), R440.

[79] Arman Akbarian and Matthew W. Choptuik. "Black hole critical behavior with the generalized BSSN formulation". In: *Phys. Rev. D* 92 (8 Oct. 2015), p. 084037. DOI: 10.1103/PhysRevD.92.084037. URL: https://link.aps.org/doi/10.1103/PhysRevD.92.084037.

[80] Thomas W. Baumgarte. "Aspherical deformations of the Choptuik spacetime". In: *Physical Review D* 98.8 (Oct. 2018). ISSN: 2470-0029. DOI: 10.1103/physrevd.98.084012. URL: http://dx.doi.org/10.1103/PhysRevD.98.084012.

[81] Matthew William Choptuik. "A study of numerical techniques for radiative problems in general relativity". PhD thesis. University of British Columbia, 1986.

[82] Aaron Meurer et al. "SymPy: symbolic computing in Python". In: *PeerJ Computer Science* 3 (2017), e103.

[83] *NRPy+ interactive tutorial*. URL: http://astro.phys.wvu.edu/bhathome/nrpytutorial.html.

[84] Shahar Hod and Tsvi Piran. "Fine structure of Choptuik's mass-scaling relation". In: *Phys. Rev. D* 55 (2 Jan. 1997), R440–R442. DOI: 10.1103/PhysRevD.55.R440. URL: https://link.aps.org/doi/10.1103/PhysRevD.55.R440.

[85] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: https://doi.org/10.1038/s41592-019-0686-2.

[86] Katy Clough and Eugene A Lim. "Critical phenomena in non-spherically symmetric scalar bubble collapse". In: *arXiv preprint arXiv:1602.02568* (2016).

[87] Heinz Kreiss, Heinz-Otto Kreiss, and Joseph Oliger. *Methods for the approximate solution of time dependent problems*. 10. International Council of Scientific Unions, World Meteorological Organization, 1973.

[88] Micaela Oertel et al. "Equations of state for supernovae and compact stars". In: *Reviews of Modern Physics* 89.1 (2017), p. 015007.

[89] James M Lattimer and Madappa Prakash. "The equation of state of hot, dense matter and neutron stars". In: *Physics Reports* 621 (2016), pp. 127–164.

[90] *Stellar collapse website*. URL: https://stellarcollapse.org/.

[91] H-Th Janka, Th Zwerger, and Ralph Moenchmeyer. "Does artificial viscosity destroy prompt type-II supernova explosions?" In: *Astronomy and Astrophysics* 268 (1993), pp. 360–368.

[92] *Project Jupyter website*. URL: https://jupyter.org/.

[93] Scott C. Noble, Julian H. Krolik, and John F. Hawley. "Direct Calculation of the Radiative Efficiency of an Accretion Disk Around a Black Hole". In: *The Astrophysical Journal* 692.1 (Feb. 2009), pp. 411–421. DOI: 10.1088/0004-637X/692/1/411. arXiv: 0808.3140 [astro-ph].

[94] Stuart L Shapiro and Saul A Teukolsky. *Black holes, white dwarfs, and neutron stars: The physics of compact objects*. John Wiley & Sons, 2008.

[95] *SROEOS bitbucket repository*. URL: https://bitbucket.org/andschn/sroeos/src/master/.

[96] Zelmani eosdrivercxx bitbucket repo. *Interpolation drivers for the SRO tabulated EOS*. URL: https://bitbucket.org/zelmani/eosdrivercxx.

[97] F Douchin and P Haensel. "A unified equation of state of dense matter and neutron star structure". In: *Astronomy & Astrophysics* 380.1 (2001), pp. 151–167.

[98] Richard C Tolman. "Effect of inhomogeneity on cosmological models". In: *Proceedings of the national academy of sciences of the United States of America* 20.3 (1934), p. 169.

[99] Richard C Tolman. "Static solutions of Einstein's field equations for spheres of fluid". In: *Physical Review* 55.4 (1939), p. 364.

[100] J Robert Oppenheimer and George M Volkoff. "On massive neutron cores". In: *Physical Review* 55.4 (1939), p. 374.

[101] H Thankful Cromartie et al. "Relativistic Shapiro delay measurements of an extremely massive millisecond pulsar". In: *Nature Astronomy* 4.1 (2020), pp. 72–76.

[102] Paul B Demorest et al. "A two-solar-mass neutron star measured using Shapiro delay". In: *nature* 467.7319 (2010), pp. 1081–1083.

[103] Thomas W. Baumgarte, Scott A. Hughes, and Stuart L. Shapiro. "Evolving Einstein's field equations with matter: The "hydro without hydro" test". In: *Physical Review D* 60.8 (Sept. 1999). ISSN: 1089-4918. DOI: 10.1103/physrevd.60.087501. URL: http://dx.doi.org/10.1103/PhysRevD.60.087501.

[104] Frank Löffler et al. "The Einstein Toolkit: a community computational infrastructure for relativistic astrophysics". In: *Classical and Quantum Gravity* 29.11 (2012), p. 115001.

[105] *Einstein Toolkit Consortium Homepage*. URL: http://www.einsteintoolkit.org.

[106] Tom Goodale et al. "The Cactus Framework and Toolkit: Design and Applications". In: *Vector and Parallel Processing – VECPAR'2002, 5th International Conference, Lecture Notes in Computer Science*. Berlin: Springer, 2003. URL: http://edoc.mpg.de/3341.

[107] *Cactus Computation Toolkit homepage*. URL: http://cactuscode.org/.

[108] Erik Schnetter, Scott H Hawley, and Ian Hawke. "Evolutions in 3D numerical relativity using fixed mesh refinement". In: *Classical and quantum gravity* 21.6 (2004), p. 1465.

[109] *Carpet: Adaptive mesh refinement for the Cactus framework*. URL: http://carpetcode.org/.

[110] Jonathan Thornburg. "Event and apparent horizon finders for 3+ 1 numerical relativity". In: *Living Reviews in Relativity* 10.1 (2007), p. 3.

[111] J Thornburg. "Classical Quantum Gravity 21, 743 (2004)". In: *AIP Conf. Proc.* Vol. 686. 2003, p. 247.

[112] Erik Schnetter, Frank Herrmann, and Denis Pollney. "Horizon pretracking". In: *Physical Review D* 71.4 (2005), p. 044033.

[113] Erik Schnetter, Badri Krishnan, and Florian Beyer. "Introduction to dynamical horizons in numerical relativity". In: *Physical Review D* 74.2 (2006), p. 024028.

[114] Olaf Dreyer et al. "Introduction to isolated horizons in numerical relativity". In: *Physical Review D* 67.2 (2003), p. 024018.

[115] Kip S Thorne. "Multipole expansions of gravitational radiation". In: *Reviews of Modern Physics* 52.2 (1980), p. 299.

[116] LORENE. *Language Objet pour la RElativité NumériquE*. URL: https://lorene.obspm.fr/.

[117] Eric Gourgoulhon et al. "Quasiequilibrium sequences of synchronized and irrotational binary neutron stars in general relativity: Method and tests". In: *Physical Review D* 63.6 (2001), p. 064029.

[118] Keisuke Taniguchi, Eric Gourgoulhon, and Silvano Bonazzola. "Quasiequilibrium sequences of synchronized and irrotational binary neutron stars in general relativity. II. Newtonian limits". In: *Physical Review D* 64.6 (2001), p. 064012.

[119] Keisuke Taniguchi and Eric Gourgoulhon. "Equilibrium sequences of synchronized and irrotational binary systems composed of different mass stars in Newtonian gravity". In: *Physical Review D* 65.4 (2002), p. 044027.

[120] Keisuke Taniguchi and Eric Gourgoulhon. "Quasiequilibrium sequences of synchronized and irrotational binary neutron stars in general relativity. III. Identical and different mass stars with $\gamma = 2$". In: *Physical Review D* 66.10 (2002), p. 104019.

[121] Keisuke Taniguchi and Eric Gourgoulhon. "Various features of quasiequilibrium sequences of binary neutron stars in general relativity". In: *Physical Review D* 68.12 (2003), p. 124025.

[122] M Bejger et al. "Impact of the nuclear equation of state on the last orbits of binary neutron stars". In: *Astronomy & Astrophysics* 431.1 (2005), pp. 297–306.

[123] *Compact-binaries website, video repository*. URL: https://compact-binaries.org/videos.

[124] *Compact-binaries website, BNS PPEOS SLy4 movie*. URL: https://compact-binaries.org/node/2596.

[125] Lawrence E Kidder, Mark A Scheel, and Saul A Teukolsky. "Extending the lifetime of 3D black hole computations with a new hyperbolic system of evolution equations". In: *Physical Review D* 64.6 (2001), p. 064017.

Appendices

This page intentionally left blank.

Appendix A

Classification of second-order partial differential equations

In this thesis we study how to efficiently solve non-linear partial differential equations using numerical techniques. To this end, it is useful to be able to identify what category a given partial differential equation falls into. We note that this exposition is presented here for the convenience of the reader and closely follows the ones in sections 6.1 and 11.1 of [48].

The prototype for an *elliptic* equation is Poisson's equation,

$$\partial_x^2 u + \partial_y^2 u = \rho \ . \tag{A.1}$$

In this equation, ρ is a source term that may depend on the function u, its first derivatives, and the spatial coordinates x, y. When $\rho = 0$, the resulting equation is the famous Laplace's equation. An example of this kind of equation is the Hamiltonian constraint, equations (3.16) and (3.62).

An example of a *parabolic* equation is the diffusion equation,

$$\partial_t u - \partial_x \left(\kappa \partial_x u\right) = \rho \ , \tag{A.2}$$

where κ is the diffusion coefficient.

The prototypical example of a hyperbolic equation is the wave equation,

$$\partial_t^2 u - c^2 \partial_x^2 u = \rho \ , \tag{A.3}$$

where c is the constant wave speed.

We have encountered equations very similar to the wave equation in this thesis, but they are usually disguised as a pair of equations instead of a single one. For example, let $\partial_t u \equiv v$. Then equation (A.3) is equivalent to the pair of equations

$$\partial_t u = v \ , \tag{A.4}$$

$$\partial_t v = c^2 \partial_x^2 u + \rho \ , \tag{A.5}$$

which resemble, but are not equivalent to, the ADM equations that evolve K_{ij} and γ_{ij}, (2.57) and (2.58) respectively.

The pair of equations (A.4) and (A.5) can be brought to full first-order form by introducing yet another variable that absorbs the first spatial derivative of the function u. For example, let $w \equiv \partial_x u$. Then we get the first-order system

$$\partial_t u = v \ , \tag{A.6}$$

$$\partial_t v = c^2 \partial_x w + \rho \ , \tag{A.7}$$

$$\partial_t w = \partial_x v \ , \tag{A.8}$$

where the third equation is obtained by taking the time derivative of the definition of w

$$w = \partial_x u \implies \partial_t w = \partial_t \partial_x u = \partial_x \partial_t u = \partial_x v \ , \tag{A.9}$$

where in the last equality we replaced $\partial_t u$ by the RHS of equation (A.6). We can now express this system of equations in the compact form

$$\partial_t \vec{f} + A \cdot \partial_x \vec{f} = \vec{S} \ , \tag{A.10}$$

where $\vec{f}$ is the solution vector and $\vec{S}$ is the source vector, which are given by

$$\vec{f} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad \text{and} \quad \vec{S} = \begin{bmatrix} v \\ \rho \\ 0 \end{bmatrix} \ . \tag{A.11}$$

The matrix A is known as the *velocity matrix* and, for this particular system of equations

is given by

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -c^2 \\ 0 & -1 & 0 \end{bmatrix} . \tag{A.12}$$

We know that the general solution to the wave equation is of the form $u(t,x) = g(x + ct) + h(x - ct)$. This means that part of the solution u travels along lines $x + ct =$ constant, while the other part travels along $x - ct =$ constant. These lines are called the *characteristic curves*. The *characteristic speeds*, dx/dt, of the solution can be obtained from the eigenvalues of the matrix A, which in our example are given by

$$\det\left(A - 1 \cdot \lambda\right) = \lambda^3 - \lambda c^2 = \lambda\left(\lambda^2 - c^2\right) = 0 \implies \lambda = \begin{cases} +c \\ -c \\ 0 \end{cases} , \tag{A.13}$$

as expected.

A.1 General classification

We now generically classify second-order partial differential equations which are given by

$$A\partial_\xi^2\phi + 2B\partial_\xi\partial_\eta\phi + C\partial_\eta^2\phi = \rho \ , \tag{A.14}$$

where the coefficients A, B, and C are real, differentiable, and do not vanish simultaneously. The classification of the differential equation then depends on these coefficients:

- If $AC - B^2 > 0$, then there is a coordinate transformation $(\xi,\eta) \to (x,y)$ such that equation (A.14) can be written in the form (A.1); thus, the equation is *elliptic*.
- If $AC - B^2 = 0$, then there is a coordinate transformation $(\xi,\eta) \to (x,y)$ such that equation (A.14) can be written in the form (A.2); thus, the equation is *parabolic*.
- If $AC - B^2 < 0$, then there is a coordinate transformation $(\xi,\eta) \to (x,y)$ such that equation (A.14) can be written in the form (A.3); thus, the equation is *hyperbolic*.

Only hyperbolic equations have real (i.e. not imaginary) characteristics. It is also useful to note that hyperbolic equations come in varying levels of *hyperbolicity*, which we

will discuss in more detail in the next section of this appendix.

A.2 Hyperbolicity

Consider now equation (A.10) generalized to more than one spatial dimension, namely

$$\partial_t \vec{f} + A^i \partial_i \vec{f} = \vec{S} \ . \tag{A.15}$$

If the solution vector, $\vec{f}$, has n components, then *each* matrix A^i is an $n \times n$ matrix. For the sake of our discussion, we will consider now that the source vector is zero, i.e. $\vec{S} = 0$.

A problem is called *well-posed* if there is some norm, $||\ldots||$, such that the solution vector satisfies

$$\left|\left|\vec{f}(t, x^i)\right|\right| \leq k e^{\alpha t} \left|\left|\vec{f}(0, x^i)\right|\right| \ . \tag{A.16}$$

The constants k and α are independent of the initial data $\vec{f}(0, x^i)$. Condition (A.16) states that solutions of a well-posed problem cannot increase more rapidly than exponentially. However, this property is not guaranteed for all hyperbolic systems.

To classify these different hyperbolic systems based on their *hyperbolicity*, we consider an arbitrary unit vector n^i and construct the matrix

$$P = A^i n_i \ , \tag{A.17}$$

which is known as the *principal symbol* or *characteristic matrix* of the system. Based on the properties of the characteristic matrix, we can distinguish between the kind of hyperbolicity of a system. In particular, we say that the system is

- *symmetric hyperbolic* if P can be symmetrized in a way that is independent of n^i,
- *strongly hyperbolic* if, for all unit vectors n^i, P has real eigenvalues and a complete set of eigenvectors,
- *weakly hyperbolic* if P has real eigenvalues but not a complete set of eigenvectors.

There are a few results that are interesting to us, which we summarize now. A symmetric hyperbolic system is automatically strongly hyperbolic. Strongly hyperbolic systems are well-posed, while weakly hyperbolic systems are ill-posed [49].

Analysing the hyperbolicity of the ADM equations with the techniques showed here is not possible, because we have second spatial derivatives of the evolved fields in the evolution equations. An analysis that rewrites the ADM equations in full first-order form, by absorbing the first spatial derivatives of the fields into new auxiliary variables, and shows that the system is only weakly hyperbolic, and thus ill-posed, can be found in [125]. The BSSN formulation of Einstein's field equations, which we study in section 2.4, introduces new, auxiliary variables, such that the resulting system of equations becomes strongly hyperbolic, and hence well-posed. The hyperbolicity of the BSSN equations has been studied in [50–53].

This page intentionally left blank.

Appendix B

Maxwell's equations

B.1 The similarities between Maxwell's equations and the ADM equations

To acquire some intuition about the changes we will do to the ADM equations to cast them into a strongly hyperbolic form, we will first pause and look at Maxwell's equations. Notice that Maxwell's equations share great resemblance to the ADM equations, in the sense that we have two constraint equations,

$$\mathcal{C}_E \equiv D_i E^i - 4\pi\rho_e = 0 \ , \tag{B.1}$$

$$\mathcal{C}_B \equiv D_i B^i = 0 \ , \tag{B.2}$$

and two evolution equations,

$$\partial_t E_i = \epsilon_{ijk} D^j B^k - 4\pi j_i \ , \tag{B.3}$$

$$\partial_t B_i = -\epsilon_{ijk} D^j E^k \ . \tag{B.4}$$

In the equations above, ρ_e is the electric charge density, j^i is the charge 3-current, and D_i is a covariant derivative with respect to the coordinate x^i. In flat space and Cartesian coordinates, $D_i \to \partial_i$. Taking the divergence of equation (B.3) yields the continuity equation

$$\partial_t \rho + D_i j^i = 0 \ . \tag{B.5}$$

To rewrite these equations in a form that resembles the ADM equations even further, we introduce the vector potential $A^a \equiv (\Phi, A^i)$, and write

$$B_i = \epsilon_{ijk} D^j A^k \ . \tag{B.6}$$

Then, by construction, B^i automatically satisfies the contraint (B.2). Using (B.6), we can rewrite Maxwell's equations as

$$\mathcal{C}_E = D_i E^i - 4\pi\rho_e = 0 \ , \tag{B.7}$$

$$\partial_t A_i = -E_i - D_i \Phi \ , \tag{B.8}$$

$$\partial_t E_i = D_i D^j A_j - D^j D_j A_i - 4\pi j_i \ , \ . \tag{B.9}$$

When we rewrite Maxwell's equations in this form, we introduce a gauge freedom into the problem, namely the freely specifiable scalar potential Φ. An initial value problem involving the set of equations (B.7)–(B.9) begins by specifying the initial data (A_i, E_i, ρ, j^i) such that the constraint (B.7) is satisfied. Before we can start the evolution with equations (B.8) and (B.9), we must specify a gauge condition, for example the *Coulomb* or *transverse* gauge where

$$D_i A^i = 0 \ . \tag{B.10}$$

Recall that in the ADM formalism, the evolution equations for the spatial metric and extrinsic curvature are, respectively,

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i \beta_j + D_j \beta_i \ , \tag{B.11}$$

$$\begin{aligned} \partial_t K_{ij} = {} & \beta^\ell \partial_\ell K_{ij} + K_{i\ell} \partial_j \beta^\ell + K_{j\ell} \partial_i \beta^\ell - D_i D_j \alpha \\ & + \alpha \left({}^{(3)}R_{ij} + K K_{ij} - 2 K_{i\ell} K^\ell{}_j \right) + 4\pi\alpha \left[\gamma_{ij} \left(S - \rho \right) - 2 S_{ij} \right] \ . \end{aligned} \tag{B.12}$$

Now let us identify the vector potential A_i with the spatial metric γ_{ij} and the electric field E_i with the extrinsic curvature K_{ij}. Notice that the RHSs of equations (B.8) and (B.11) both contain the other field variable and the spatial derivative of a gauge variable. As for equations (B.9) and (B.12), we see that they both contain source terms as well as second derivatives of the other field variable, which in the case of equation (B.12) are hidden inside the Ricci tensor ${}^{(3)}R_{ij}$, which contains terms of the form $\gamma^{k\ell} \partial_\ell \partial_i \gamma_{kj}$.

Notice that equations (B.8) and (B.9) can *almost* be combined to yield a wave equation. Taking the time derivative of (B.8) and substituting the RHS of (B.9) into the resulting equation yields

$$\left(-\partial_t^2 A_i + D^j D_j A_i\right) - D_i D^j A_j = D_i \partial_t \Phi - 4\pi j_i \ . \tag{B.13}$$

The term in parenthesis in (B.13) is the "wave operator", $(-\partial_t^2 + \Box)$, acting on the vector potential A_i, while the RHS of (B.13) is simply a source term. However, the mixed derivative term $D_i D^j A_j$ prevents us from writing it as a wave equation. The same thing happens for the ADM equations, where the Ricci tensor contains mixed derivative terms of the spatial metric, as well as the d'Alembertian $\gamma^{k\ell}\partial_k \partial_\ell \gamma_{ij}$. Without the mixed derivatives terms, the ADM equations could also be written as a wave equation for the spatial metric.

Let us now explore a few ways of eliminating the mixed derivative term in Maxwell's equation, which will be useful to understand the BSSN reformulation of the ADM equations.

B.2 Improving the hyperbolicity of Maxwell's equations

B.2.1 Making an appropriate gauge choice

Let us consider the simplest approach to eliminate the mixed derivative term: making a gauge choice such that

$$D^i A_i = 0 \ . \tag{B.14}$$

This choice is known as the *Coulomb gauge*.

This turns Maxwell's equations into a wave equation for the vector potential,

$$\left(-\partial_t^2 + D^j D_j\right) A_i = D_i \partial_t \Phi - 4\pi j_i \ . \tag{B.15}$$

However, consider now taking the divergence of equation (B.8)

$$D^i \partial_t A_i = \partial_t D^i A_i = 0 = -D^i E_i - D^i D_i \Phi \ . \tag{B.16}$$

Using the constraint (B.7), we see that the scalar potential, Φ, must satisfy the elliptic

equation

$$D^i D_i \Phi = -4\pi\rho_e \ , \tag{B.17}$$

which is the famous *Poisson's equation*. This equation must be satisfied for all times and it can be extremely inconvenient to solve this equation over and over during a time evolution.

B.2.2 First-order hyperbolic formulation

An alternative approach is to consider taking the time derivative of equation (B.9) instead of (B.8),

$$\begin{aligned}
\partial_t^2 E_i &= \left(D_i D^j\right) \partial_t A_j - \left(D^j D_j\right) \partial_t A_i - 4\pi \partial_t j_i \\
&= \left(D_i D^j\right) \left(-E_j - D_j \Phi\right) - \left(D^j D_j\right) \left(-E_i - D_i \Phi\right) - 4\pi \partial_t j_i \\
&= D^j D_j E_i - D_i D_j E^j - D_i D^j D_j \Phi + D^j D_j D_i \Phi - 4\pi \partial_t j_i \\
&= D^j D_j E_i - D_i D_j E^j - 4\pi \partial_t j_i \ .
\end{aligned} \tag{B.18}$$

Using the contrainst (B.7) we end up with a wave equation for the electric field:

$$-\partial_t^2 E_i + D^j D_j E_i = 4\pi \left(D_i \rho_e + \partial_t j_i\right) \ . \tag{B.19}$$

Notice how this approach eliminates completely the gauge variables Φ and A_i. Equation (B.19) is *symmetric hyperbolic*.

To obtain the first-order hyperbolic formulation, we need to introduce new variables that absorb the first derivatives of the electric field. For the sake of this exposition, let us do this only for the time derivative. Let

$$\mathcal{E}_i \equiv \partial_t E_i \ . \tag{B.20}$$

Then equation (B.19) can be written as a system of two first-order (in time) partial differential equations

$$\partial_t E_i = \mathcal{E}_i \ , \tag{B.21}$$

$$\partial_t \mathcal{E}_i = D^j D_j E_i - 4\pi \left(D_i \rho_e + \partial_t j_i\right) \ , \tag{B.22}$$

which are very suitable for integration using the method of lines, for example.

B.2.3 Introducing auxiliary variables

Another approach, which is very similar to the one we explored in section B.2.1, is to introduce a new variable

$$\Gamma \equiv D^i A_i \ , \tag{B.23}$$

and consider it as a new, independent variable that must be evolved in time as well. An evolution equation for Γ can be derived using equation (B.8) and the constraint (B.7):

$$\partial_t \Gamma = \partial_t D^i A_i = D^i \partial_t A_i = -D^i E_i - D^i D_i \Phi = -D^i D_i \Phi - 4\pi \rho_e \ . \tag{B.24}$$

Using this new variable, we can also eliminate the mixed derivative term in the evolution equation of the electric field, yielding the new system of evolution equations

$$\partial_t A_i = -E_i - D_i \Phi \ , \tag{B.25}$$

$$\partial_t E_i = -D_j D^j A_i + D_i \Gamma - 4\pi j_i \ , \tag{B.26}$$

$$\partial_t \Gamma = -D^i D_i \Phi - 4\pi \rho_e \ . \tag{B.27}$$

The constraints that must be satisfied during the evolution are now equations (B.7) and (B.23).

This particular approach bears great resemblance to the BSSN reformulation of the ADM equations, which is discussed in section 2.4.

This page intentionally left blank.

Appendix C

Finite differences

C.1 Basic notation and terminology

Here we briefly review the method of *finite differences*, used to approximate derivatives of a function. Let $f(x)$ be a function of a single variable, $x \in [x_{\min}, x_{\max}]$, and consider the definitions

$$x_j \equiv j \cdot \Delta x \ , \tag{C.1}$$

$$\Delta x \equiv \frac{x_{\max} - x_{\min}}{N_x + 1} \ , \tag{C.2}$$

$$f_j \equiv f(x_j) \ , \tag{C.3}$$

where N_x is a positive integer. Assume, also, that $f(x)$ is 2π-periodic in x.

Jargon: The points x_j are said to define a *grid* where we evaluate the function $f(x)$, which is therefore refered to as *gridfunction*.

We now define the *finite difference operators*

$$D_+ f_j \equiv \frac{f_{j+1} - f_j}{\Delta x} \ , \tag{C.4}$$

$$D_- f_j \equiv \frac{f_j - f_{j-1}}{\Delta x} \ , \tag{C.5}$$

$$D_0 f_j \equiv \frac{f_{j+1} - f_{j-1}}{2\Delta x} \ . \tag{C.6}$$

Notice that these operations are well-defined *everywhere* in the grid, since, for example, the

operation $D_0 f_j$ can be computed at $j = 0$ since $f(x)$ is 2π-periodic and thus $f_{-1} = f_{N-1}$. Similarly, at $j = N$, we can identify $f_{N+1} = f_1$.

These operators provide different approximations to the *first* derivative of the function $f(x)$ with respect to the variable x. D_+ is known as the *forward* operator, D_- the *backward* operator, and D_0 the *centered* operator.

The points j used to approximate a derivative are often referred to as a *stencil*. Thus, the operation $D_+ f_j$ uses the stencil $(j, j+1)$, while $D_0 f_j$ uses the stencil $(j-1, j+1)$. Notice that knowing the finite difference stencil is not enough to write down the expression for the derivative, since the coefficients that multiply each of the stencil points are, at this point, unknown. In the next section we provide a more practical and, in my opinion, more intuitive way of evaluating finite difference derivatives.

C.2 A practical way of determining the coefficients of a finite difference approximation

Consider, again, the same function $f(x)$ above. But now, consider the following expansions of this function:

$$f_{j+2} = f_j + 2\Delta x \partial_x f_j + 4\frac{\Delta x^2}{2!}\partial_x^2 f_j + 8\frac{\Delta x^3}{3!}\partial_x^3 f_j + 16\frac{\Delta x^4}{4!}\partial_x^4 f_j + \mathcal{O}\left(\Delta x^5\right) , \tag{C.7}$$

$$f_{j+1} = f_j + \Delta x \partial_x f_j + \frac{\Delta x^2}{2!}\partial_x^2 f_j + \frac{\Delta x^3}{3!}\partial_x^3 f_j + \frac{\Delta x^4}{4!}\partial_x^4 f_j + \mathcal{O}\left(\Delta x^5\right) , \tag{C.8}$$

$$f_{j-1} = f_j - \Delta x \partial_x f_j + \frac{\Delta x^2}{2!}\partial_x^2 f_j - \frac{\Delta x^3}{3!}\partial_x^3 f_j + \frac{\Delta x^4}{4!}\partial_x^4 f_j + \mathcal{O}\left(\Delta x^5\right) , \tag{C.9}$$

$$f_{j-2} = f_j - 2\Delta x \partial_x f_j + 4\frac{\Delta x^2}{2!}\partial_x^2 f_j - 8\frac{\Delta x^3}{3!}\partial_x^3 f_j + 16\frac{\Delta x^4}{4!}\partial_x^4 f_j + \mathcal{O}\left(\Delta x^5\right) . \tag{C.10}$$

We are now ready to find approximations to the derivative of $f(x)$. Consider, for example, subtracting (C.2) from (C.2). We get

$$f_{j+1} - f_{j-1} = 2\Delta x \partial_x f_j + \mathcal{O}\left(\Delta x^3\right) \implies \partial_x f_j = \frac{f_{j+1} - f_{j-1}}{2\Delta x} + \mathcal{O}\left(\Delta x^2\right) . \tag{C.11}$$

Thus, we find that the operator D_0 provides a *second-order* accurate approximation to

the first derivative of $f(x)$. Similarly, the operators D_+ and D_- provide *first-order* approximations to the first derivative of $f(x)$, since by subtracting f_j from and yield

$$f_{j+1} - f_j = \Delta x \partial_x f_j + \mathcal{O}\left(\Delta x^2\right) \implies \partial_x f_j = \frac{f_{j+1} - f_j}{\Delta x} + \mathcal{O}\left(\Delta x^1\right) \ , \tag{C.12}$$

and

$$f_{j-1} - f_j = -\Delta x \partial_x f_j + \mathcal{O}\left(\Delta x^2\right) \implies \partial_x f_j = \frac{f_j - f_{j-1}}{\Delta x} + \mathcal{O}\left(\Delta x^1\right) \ , \tag{C.13}$$

respectively.

To obtain a *fourth-order* accurate approximation of the first derivative of $f(x)$, we must find a combination of $(f_{j-2}, f_{j-1}, f_{j+1}, f_{j+2})$ which cancels all other lower order terms. For example, consider the combination

$$-f_{j+2} + 8f_{j+1} - f_{j-1} + f_{j-2} = 12\Delta x \partial_x f_j + \mathcal{O}\left(\Delta x^5\right) \ , \tag{C.14}$$

which can be rearranged slightly to yield

$$\partial_x f_j = \frac{-f_{j+2} + 8f_{j+1} - f_{j-1} + f_{j-2}}{12\Delta x} + \mathcal{O}\left(\Delta x^4\right) \ . \tag{C.15}$$

The problem can then be recast to the following form. Let the desired stencil size be 5. Then what is the centered approximation we can get to the second derivative of $f(x)$ using this stencil size? Well, we know it will be a combination of the form

$$\partial_x^2 f_j = a_2 f_{j+2} + a_1 f_{j+1} + a_0 f_j + a_{-1} f_{j-1} + a_{-2} f_{j-2} \ . \tag{C.16}$$

Using the expansions (C.7)–(C.10), we find

$$\begin{aligned}\partial_x^2 f_j = &\left(a_2 + a_1 + a_0 + a_{-1} + a_{-2}\right) f_j \\ &+ \left(2a_2 + a_1 - a_{-1} - 2a_{-2}\right) \Delta x \partial_x f \\ &+ \left(4a_2 + a_1 - a_{-1} - 4a_{-2}\right) \frac{\Delta x^2}{2!} \partial_x^2 f \\ &+ \left(8a_2 + a_1 - a_{-1} - 8a_{-2}\right) \frac{\Delta x^3}{3!} \partial_x^3 f \\ &+ \left(16a_2 + a_1 - a_{-1} - 16a_{-2}\right) \frac{\Delta x^4}{4!} \partial_x^4 f \ . \end{aligned} \tag{C.17}$$

For the equation above to be true, the coefficients $\{a_j\}$ must satisfy the following system of equations

$$0 = a_2 + a_1 + a_0 + a_{-1} + a_{-2} \ , \tag{C.18}$$

$$0 = 2a_2 + a_1 - a_{-1} - 2a_{-2} \ , \tag{C.19}$$

$$1 = 4a_2 + a_1 - a_{-1} - 4a_{-2} \ , \tag{C.20}$$

$$0 = 8a_2 + a_1 - a_{-1} - 8a_{-2} \ , \tag{C.21}$$

$$0 = 16a_2 + a_1 - a_{-1} - 16a_{-2} \ , \tag{C.22}$$

or, equivalently,

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \\ -4 & -1 & 0 & 1 & 4 \\ -8 & -1 & 0 & 1 & 8 \\ -16 & -1 & 0 & 1 & 16 \end{bmatrix} \begin{bmatrix} a_{-2} \\ a_{-1} \\ a_0 \\ a_1 \\ a_2 \end{bmatrix} \ . \tag{C.23}$$

Note that the coefficient matrix may be rewritten to make the pattern more explicit

$$M = \begin{bmatrix} (-2)^0 & (-1)^0 & 0^0 & 1^0 & 2^0 \\ (-2)^1 & (-1)^1 & 0^1 & 1^1 & 2^1 \\ (-2)^2 & (-1)^2 & 0^2 & 1^2 & 2^2 \\ (-2)^3 & (-1)^3 & 0^3 & 1^3 & 2^3 \\ (-2)^4 & (-1)^4 & 0^4 & 1^4 & 2^4 \end{bmatrix} \ . \tag{C.24}$$

Finding the solution then amounts to finding the inverse of the matrix M, which in this case is given by

$$M^{-1} = \begin{bmatrix} 0 & 1/12 & -1/24 & -1/12 & 1/24 \\ 0 & -2/3 & 2/3 & 1/6 & -1/6 \\ 1 & 0 & -5/4 & 0 & 1/4 \\ 0 & 2/3 & 2/3 & -1/6 & -1/6 \\ 0 & -1/12 & -1/24 & 1/12 & 1/24 \end{bmatrix} . \tag{C.25}$$

The second derivative coefficients are then given by column 2 (start counting from left to write at 0). We must, however, multiply the coefficients by $2!/\Delta x^2$, which we absorbed into the derivative along the way. We then have[1]

$$\partial_x^2 f_j = \frac{-f_{j+2} + 16f_{j+1} - 30f_j + 16f_{j-1} - f_{j-2}}{2\Delta x^2} . \tag{C.26}$$

Note, also, that column 1 gives the coefficients to the first derivative, when multiplied by $1!/\Delta x^1$:

$$\partial_x f_j = \frac{-f_{j+2} + 8f_{j+1} - 8f_{j-1} + f_{j-2}}{12\Delta x} . \tag{C.27}$$

This is the way finite difference coefficients are computed in `NRPy+` (see [83] for more details).

C.3 Upwinded and downwinded derivatives

From our previous discussion, it is clear that the stencils associated with *centered* finite differences are *symmetric*, i.e. if we include the points (f_{j-2}, f_{j-1}), we can immediately infer that the points (f_{j+1}, f_{j+2}) will also be included.

An *upwinded* derivative starts from the most general stencil that a centered finite differences approximation has, $(f_{j-N}, f_{j-N+1}, \ldots, f_j, \ldots, f_{j+N-1}, f_{j+N})$, and then shifts the indices of the functions used by 1 in the *positive* direction. This means that we get the new stencil $(f_{j-N+1}, \ldots, f_j, \ldots, f_{j+N-1}, f_{j+N}, f_{j+N+1})$. Consider the case where $N = 2$

[1] Look at equation (C.23) again if you get confused about the order of the coefficients. In the way we have written the matrix equation (C.23), each column of the matrix M^{-1} yields, from top to bottom, the coefficients $(a_{-2}, a_{-1}, a_0, a_1, a_2)$.

and the approximation is fourth-order. Then we must find the combination

$$\partial_x f_j = a_{-1} f_{j-1} + a_0 f_j + a_1 f_{j+1} + a_2 f_{j+2} + a_3 f_{j+3} \ . \tag{C.28}$$

Compare equations (C.28) and (C.16). We then need to find the inverse of the matrix

$$M = \begin{bmatrix} (-1)^0 & (0)^0 & 1^0 & 2^0 & 3^0 \\ (-1)^1 & (0)^1 & 1^1 & 2^1 & 3^1 \\ (-1)^2 & (0)^2 & 1^2 & 2^2 & 3^2 \\ (-1)^3 & (0)^3 & 1^3 & 2^3 & 3^3 \\ (-1)^4 & (0)^4 & 1^4 & 2^4 & 3^4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 4 & 9 \\ -1 & 0 & 1 & 8 & 27 \\ 1 & 0 & 1 & 16 & 81 \end{bmatrix} \ , \tag{C.29}$$

which is given by

$$M^{-1} = \begin{bmatrix} 0 & -1/4 & 11/24 & -1/4 & 1/24 \\ 1 & -5/6 & -5/6 & 5/6 & -1/6 \\ 0 & 3/2 & 1/4 & -1 & 1/4 \\ 0 & -1/2 & 1/6 & 1/2 & -1/6 \\ 0 & 1/12 & -1/24 & -1/12 & 1/24 \end{bmatrix} \ , \tag{C.30}$$

and therefore we have

$$\partial_x f_j = \frac{-3f_{j-1} - 10f_j + 18f_{j+1} - 6f_{j+2} + f_{j+3}}{12\Delta x} \ . \tag{C.31}$$

Similarly, a *downwinded* derivative shifts the indices of the functions used in the centered stencil by 1 in the *negative* direction, i.e.

$$\partial_x f_j = a_{-3} f_{j-3} + a_{-2} f_{j-2} + a_{-1} f_{j-1} + a_0 f_j + a_1 f_{j+1} \ . \tag{C.32}$$

Repeating the reasoning above we get

$$\partial_x f_j = \frac{3f_{j+1} + 10f_j - 18f_{j-1} + 6f_{j-2} - f_{j-3}}{12\Delta x} \ . \tag{C.33}$$

Note that upwinded/downwinded derivatives retain the accuracy of their centered cousins with same stencil size. To see this for our particular example, consider the expansion

$$f_{j+3} = f_j + 3\Delta x \partial_x f_j + 9\frac{\Delta x^2}{2!}\partial_x^2 f_j + 27\frac{\Delta x^3}{3!}\partial_x^3 f_j + 81\frac{\Delta x^4}{4!}\partial_x^4 f_j + \mathcal{O}\left(\Delta x^5\right) \ . \tag{C.34}$$

Then, writing the combination given in equation (C.31) in terms of the explicit expansions (C.7), (C.2), (C.2), and (C.34) yields

$$-3f_{j-1} + 18f_{j+1} - 6f_{j+2} + f_{j+3} = 10f_j + 12\partial_x f_j + \mathcal{O}\left(\Delta x^5\right) \ , \tag{C.35}$$

as anticipated above.

This page intentionally left blank.

Appendix D

Von Neumann stability analysis

We will now discuss how to perform a von Neumann stability analysis of one or more partial differential equations (PDE). This will naturally lead to the definition of the Courant-Friedrichs-Lewy (CFL) factor.

D.1 Single PDE

Consider the PDE

$$\partial_t f(t,x) = v\partial_x f(t,x) \ , \tag{D.1}$$

where $v > 0$ is the wave speed. Then, consider a foward-time, centered-space (FTCS) finite-difference discretization of this PDE (refer to appendix C.1 for a description of the notation below),

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = v\left(\frac{f_{j+1}^n - f_{j-1}^n}{2\Delta x}\right) \ , \tag{D.2}$$

which is a $\mathcal{O}\left(\Delta t + \Delta x^2\right)$ accurate scheme. We then pose the question: is this an acceptable scheme for solving the PDE (D.1)?

To answer this question, we will perform the *von Neumann stability analysis* of the discretization scheme (D.2). We perform this analysis by assuming that the function f_j^n behaves as

$$f_j^n = \xi^n e^{ikj\Delta x} \ , \tag{D.3}$$

where k is the real spatial wave number and $\xi = \xi(k)$ is a complex number. Notice that ξ^n in the above equation means "ξ to the n^{th} power", not "ξ at time $t = n \cdot \Delta t$". This means that if $|\xi(k)| > 1$ for *some* k, the discretization scheme is *unstable*, since they will have exponentially growing modes.

Inserting relation (D.3) into (D.2) we get

$$\frac{e^{ikj\Delta x}}{\Delta t}\left(\xi^{n+1} - \xi^n\right) = \frac{v}{\Delta x}\xi^n e^{ikj\Delta x}\left(\frac{e^{ik\Delta x} - e^{-ik\Delta x}}{2}\right) . \tag{D.4}$$

Dividing through by $\xi^n e^{ikj\Delta x}$ we get

$$\frac{\xi - 1}{\Delta t} = \frac{v}{\Delta x}\left(\frac{e^{ik\Delta x} - e^{-ik\Delta x}}{2}\right) = \frac{iv}{\Delta x}\sin(k\Delta x) \implies \xi(k) = 1 + i\frac{v\Delta t}{\Delta x}\sin(k\Delta x) . \tag{D.5}$$

Computing the module of $\xi(k)$ we get

$$|\xi(k)| = \sqrt{1 + \left[\frac{v\Delta t}{\Delta x}\sin(k\Delta x)\right]^2} , \tag{D.6}$$

which shows that $|\xi(k)| > 1$ (remember $v > 0$) for *all* values of k. The FTCS discretization (D.2) of equation (D.1) is thus *unconditionally unstable*.

We have no other option but to choose a different discretization scheme. Consider, then, a centered-time, centered-space (CTCS) scheme. This scheme has a special name: the *leapfrog* scheme. Using it to discretize equation (D.1) we get

$$\frac{f_j^{n+1} - f_j^{n-1}}{2\Delta t} = v\left(\frac{f_{j+1}^n - f_{j-1}^n}{2\Delta x}\right) . \tag{D.7}$$

Replacing relation (D.3) into (D.7) we get

$$\frac{e^{ikj\Delta x}}{2\Delta t}\left(\xi^{n+1} - \xi^n\right) = \frac{v}{\Delta x}\xi^n e^{ikj\Delta x}\left(\frac{e^{ik\Delta x} - e^{-ik\Delta x}}{2}\right) . \tag{D.8}$$

Dividing through by $\xi^{n-1}e^{ikj\Delta x}$ we have

$$\frac{\xi^2 - 1}{2\Delta t} = \frac{iv}{\Delta x}\xi\sin(k\Delta x) \implies \xi^2 - \frac{2iv\Delta t}{\Delta x}\sin(k\Delta x)\,\xi - 1 = 0 . \tag{D.9}$$

Solving for $\xi(k)$ we find that

$$\xi(k) = i\left[\frac{v\Delta t \sin(k\Delta x)}{\Delta x}\right] \pm \sqrt{1 - \left[\frac{v\Delta t \sin(k\Delta x)}{\Delta x}\right]^2} \ . \tag{D.10}$$

Let us focus on the particular case where $\sin(k\Delta x) = \pm 1$, so that its absolute value is maximized. Then, if $1 - \left[\frac{v\Delta t}{\Delta x}\right]^2 \geq 0$, the second term above is real and thus

$$|\xi(k)| = \sqrt{\left(\frac{v\Delta t}{\Delta x}\right)^2 + 1 - \left(\frac{v\Delta t}{\Delta x}\right)^2} = 1 \ , \tag{D.11}$$

and the system is stable. Since $\Delta t > 0$, $\Delta x > 0$, and $v > 0$, we can write the condition $1 - \left[\frac{v\Delta t}{\Delta x}\right]^2 \geq 0$ as

$$\left[\frac{v\Delta t}{\Delta x}\right]^2 \leq 1 \implies \boxed{\frac{v\Delta t}{\Delta x} \leq 1} \ , \tag{D.12}$$

which is known as the *Courant-Friedrichs-Lewy* (CFL) stability condition. One can define

$$C \equiv \frac{v\Delta t}{\Delta x} \ , \tag{D.13}$$

which is known as the *CFL factor*. The leapfrog scheme is only stable if the CFL condition is satisfied, and thus the scheme is said to be *conditionally stable*.

Yet another option would be to choose a backward-time, centered-space (BTCS) discretization scheme,

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = v\left(\frac{f_{j+1}^{n+1} - f_{j-1}^{n+1}}{2\Delta x}\right) \ . \tag{D.14}$$

Notice that we can no longer solve for f_j^{n+1} in terms of functions at the time level t^n alone, since the scheme couples it to its neighbors f_{j+1}^{n+1} and f_{j-1}^{n+1} at time level t^{n+1}. This is what is known as an *implicit* discretization scheme. Instead of sweeping over the grid, we must find f at t^{n+1} for all values of x_j at once, similar to what we do when solving an elliptic equation.

Why choose such a complicated integration scheme? Well, if we perform a von Neumann stability analysis we find that

$$\xi - 1 = \frac{iv\Delta t \sin(k\Delta x)}{\Delta x}\xi \implies \xi(k) = \frac{1}{1 - i\frac{v\Delta t}{\Delta x}\sin(k\Delta x)} \ . \tag{D.15}$$

Considering, again, $\sin(k\Delta x) = \pm 1$, we have that

$$
\begin{aligned}
|\xi(k)| &= \sqrt{\left[\frac{1}{1+\left(\frac{v\Delta t}{\Delta x}\right)^2}\right]^2 + \left[\frac{\frac{v\Delta t}{\Delta x}}{1+\left(\frac{v\Delta t}{\Delta x}\right)^2}\right]^2} \\
&= \sqrt{\frac{1+\left(\frac{v\Delta t}{\Delta x}\right)^2}{\left[1+\left(\frac{v\Delta t}{\Delta x}\right)^2\right]^2}} \\
&= \sqrt{\frac{1}{1+\left(\frac{v\Delta t}{\Delta x}\right)^2}} \ . \qquad \text{(D.16)}
\end{aligned}
$$

Notice that since $\left(\frac{v\Delta t}{\Delta x}\right)^2 > 0$, $|\xi(k)| < 1$ for all k, and thus the discretization scheme is *unconditionally stable*.

D.2 System of PDEs

We now illustrate the von Neumann stability analysis for a system of PDEs. Consider the wave equation

$$\partial_t^2 u(t,x) = c^2 \partial_x^2 u(t,x) \ , \qquad \text{(D.17)}$$

and the auxiliary variables

$$v(t,x) \equiv \partial_t u(t,x) \ , \qquad \text{(D.18)}$$

$$w(t,x) \equiv c\partial_x u(t,x) \ . \qquad \text{(D.19)}$$

Then we can rewrite the wave equation as the first-order system of PDEs

$$\partial_t v = c\partial_x w \ , \qquad \text{(D.20)}$$

$$\partial_t w = c\partial_x v \ . \qquad \text{(D.21)}$$

If we choose, say, a FTCS scheme, we have

$$v_j^{n+1} = v_j^n + \frac{c\Delta t}{2\Delta x}\left(w_{j+1}^n - w_{j-1}^n\right) \ , \tag{D.22}$$

$$w_j^{n+1} = w_j^n + \frac{c\Delta t}{2\Delta x}\left(v_{j+1}^n - v_{j-1}^n\right) \ . \tag{D.23}$$

Assuming, now, that

$$\begin{bmatrix} v_j^n \\ w_j^n \end{bmatrix} = \xi^n e^{ikj\Delta x} \begin{bmatrix} v^0 \\ w^0 \end{bmatrix} \ , \tag{D.24}$$

we can find the system of equations

$$\xi v^0 = v^0 + i\frac{c\Delta t}{\Delta x}\sin\left(k\Delta x\right) w^0 \ , \tag{D.25}$$

$$\xi w^0 = w^0 + i\frac{c\Delta t}{\Delta x}\sin\left(k\Delta x\right) v^0 \ . \tag{D.26}$$

We can rewrite this system in matrix form as

$$\begin{bmatrix} 1-\xi & i\frac{c\Delta t}{\Delta x}\sin\left(k\Delta x\right) \\ i\frac{c\Delta t}{\Delta x}\sin\left(k\Delta x\right) & 1-\xi \end{bmatrix} \cdot \begin{bmatrix} v^0 \\ w^0 \end{bmatrix} = 0 \ , \tag{D.27}$$

which can only be solved if

$$\left(1-\xi\right)^2 + \left[\frac{c\Delta t}{\Delta x}\sin\left(k\Delta x\right)\right]^2 = 0 \implies \xi(k) = 1 \pm i\frac{c\Delta t}{\Delta x}\sin\left(k\Delta x\right) \ . \tag{D.28}$$

Thus, we again find that $|\xi(k)| > 1$ for all values of k and the FTCS discretization of the wave equation is unconditionally unstable.

This page intentionally left blank.

Appendix E

Discretization of the EKG system in the polar/radial gauge

We will present here the discretized version of equations (3.10), (3.11), (3.16), and (3.18), as they are implemented in the `SFcollapse1D` code. The code supports both spherical and SinhSpherical versions of the equations, so we present both of them here.

In the equations that follow, the variable A is defined as

$$A \equiv \ln a \ . \tag{E.1}$$

E.1 Spherical coordinates

In spherical coordinates, the EKG+ADM system of equations, in the polar/radial gauge, is discretized as

$$\frac{\Phi_j^{n+1}-\Phi_j^{n-1}}{2\Delta t} = \frac{1}{2\Delta r}\left[\frac{\alpha_{j+1}^n}{a_{j+1}^n}\Pi_{j+1}^n - \frac{\alpha_{j-1}^n}{a_{j-1}^n}\Pi_{j-1}^n\right] , \tag{E.2}$$

$$\frac{\Pi_j^{n+1}-\Pi_j^{n-1}}{2\Delta t} = \frac{3}{r_{j+1}^2-r_{j-1}^3}\left[r_{j+1}^2\frac{\alpha_{j+1}^n}{a_{j+1}^n}\Pi_{j+1}^n - r_{i-1}^2\frac{\alpha_{j-1}^n}{a_{j-1}^n}\Pi_{j-1}^n\right] , \tag{E.3}$$

$$\begin{aligned}\frac{A_{j+1}^{n+1}-A_j^{n+1}}{\Delta r} &+ \frac{\exp\left(A_{j+1}^{n+1}+A_j^{n+1}\right)-1}{2r_{j+1/2}} \\ &- 2\pi r_{j+1/2}\left\{\left[\frac{1}{2}\left(\Phi_{j+1}^{n+1}+\Phi_j^{n+1}\right)\right]^2 + \left[\frac{1}{2}\left(\Pi_{j+1}^{n+1}+\Pi_j^{n+1}\right)\right]^2\right\} = 0 ,\end{aligned} \tag{E.4}$$

$$\frac{\alpha_{j+1}^{n+1}-\alpha_j^{n+1}}{\Delta r} + \frac{1}{2}\left(\alpha_{j+1}^{n+1}+\alpha_j^{n+1}\right)\left\{\frac{1-\left[\frac{1}{2}\left(a_j^{n+1}+a_{j+1}^{n+1}\right)\right]^2}{r_{j+1/2}} - \frac{2\left(a_{j+1}^{n+1}-a_j^{n+1}\right)}{\Delta r\left(a_{j+1}^{n+1}+a_j^{n+1}\right)}\right\} = 0 , \tag{E.5}$$

to order $\mathcal{O}\left(\Delta t^2+\Delta r^2\right)$.

E.2 SinhSpherical coordinates

In SinhSpherical coordinates, we introduce a new coordinate variable x which is uniformly sampled in the interval $[0,1]$ via the transformation

$$r = \mathcal{A}\frac{\sinh\left(\frac{x}{w}\right)}{\sinh\left(\frac{1}{w}\right)} , \tag{E.6}$$

with $\mathcal{A}$ and w user-specified parameters. The interpretation of $\mathcal{A}$ is that of $r_{\rm max}$, the position of the outer boundary of the computational domain, since when $x=1$, $r=\mathcal{A}$. w is a parameter that regulates the sampling density in the radial coordinate. Lowering the value of w increases the sampling near the origin *in the radial coordinate* (remember that x is uniformly sampled!). An illustration of this is given in figure E.1.

The EKG+ADM system in SinhSpherical coordinates is obtained from equations (3.10), (3.11), (3.16), and (3.18) by replacing

$$r \to \mathcal{A}\frac{\sinh\left(\frac{x}{w}\right)}{\sinh\left(\frac{1}{w}\right)} \quad \text{and} \quad \partial_r \to \frac{w}{\mathcal{A}}\frac{\sinh\left(\frac{1}{w}\right)}{\cosh\left(\frac{x}{w}\right)}\partial_x , \tag{E.7}$$

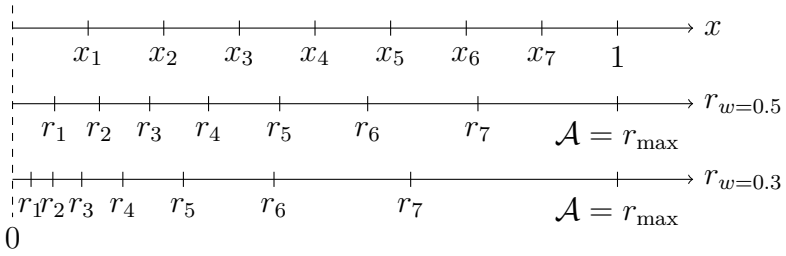


Figure E.1: Illustration of the grid structure for SinhSpherical coordinates. The top axis corresponds to the x-coordinate, which is uniformily sampled in the interval $[0, 1]$. The middle and bottom axis present the corresponding r-coordinate when the values $w = 0.5$ and $w = 0.3$, respectively, are plugged into equation (E.6). We emphasize that the grid is *not* uniformily sampled in the r-coordinate, as we can clearly see in the figure, since $\Delta r_j \equiv r_j - r_{j-1} < \Delta r_{j+1}$.

Thus, the EKG+ADM system of equations, in the polar/radial gauge and SinhSpherical coordinates, is written as

$$\partial_t \Phi = \frac{w}{\mathcal{A}} \frac{\sinh\left(\frac{1}{w}\right)}{\cosh\left(\frac{x}{w}\right)} \partial_x \left(\frac{\alpha}{a}\Pi\right) \ , \tag{E.8}$$

$$\partial_t \Pi = \frac{w}{\mathcal{A}} \frac{\sinh\left(\frac{1}{w}\right)}{\sinh^2\left(\frac{x}{w}\right)\cosh\left(\frac{x}{w}\right)} \partial_x \left[\sinh^2\left(\frac{x}{w}\right)\frac{\alpha}{a}\Phi\right] \ , \tag{E.9}$$

$$\frac{\partial_x \alpha}{\alpha} - \frac{\partial_x a}{a} - \frac{a^2-1}{w\tanh\left(\frac{x}{w}\right)} = 0 \ , \tag{E.10}$$

$$\frac{\partial_x a}{a} + \frac{a^2-1}{2w\tanh\left(\frac{x}{w}\right)} = 2\pi \frac{\mathcal{A}^2}{w\sinh^2\left(\frac{1}{w}\right)} \sinh\left(\frac{x}{w}\right)\cosh\left(\frac{x}{w}\right)\left(\Phi^2+\Pi^2\right) \ , \tag{E.11}$$

to order $\mathcal{O}\left(\Delta t^2 + \Delta r^2\right)$. The discretization of these equations is complete analogous to the one we have performed for the system in spherical coordinates and will not be repeated here because it is just a long and tedious, albeit straightforward, exercise.

This page intentionally left blank.

Appendix F

Rescaling of tensorial quantities in the BSSN formalism

When we use Brown's covariant formulation of the BSSN equations [56], we are able to choose any coordinate system we like and use the same form of the equations. Of particular interest are curvilinear coordinate systems, such as spherical-like and cylindrical-like coordinates. Here we show how to handle the coordinate singularities that arise when using curvilinear coordinate systems. Our discussion will be closely related to the `NRPy+` implementation and, hopefully, will be fairly intuitive. For a more rigurous description we recommend [25] and references therin.

Let us consider here, for example, the case of spherical coordinates, where we use the standard variables (r, θ, φ). The transformations between spherical and Cartesian coodinates are given by

$$\begin{aligned} x &= r\sin\theta\cos\varphi \ , \ r &&= \sqrt{x^2+y^2+z^2} \ , \\ y &= r\sin\theta\sin\varphi \ , \ \theta &&= \arccos\frac{z}{\sqrt{x^2+y^2+z^2}} \ , \\ z &= r\cos\theta \qquad , \ \varphi &&= \arctan\frac{y}{x} \ . \end{aligned}$$

Now consider a spatial vector U^i and assume that the components (U^x, U^y, U^z) of this vector in Cartesian coordinates are smooth. Then, transforming this vector from Cartesian to spherical coordinates is done using

$$U^i_{\rm Sph} = \frac{\partial x^i_{\rm Sph}}{\partial x^j_{\rm Cart}} U^i_{\rm Cart} \ , \tag{F.1}$$

so that, for example,

$$\begin{aligned} U^\varphi &= \frac{\partial \varphi}{\partial x} U^x + \frac{\partial \varphi}{\partial y} U^y + \frac{\partial \varphi}{\partial z} U^z \\ &= -\left(\frac{y}{x^2+y^2}\right) U^x + \left(\frac{x}{x^2+y^2}\right) U^x \\ &= -\left(\frac{\sin\varphi}{r\cos\theta}\right) U^x + \left(\frac{\cos\varphi}{r\cos\theta}\right) U^y \ . \end{aligned} \tag{F.2}$$

Similarly,

$$U^r = (\sin\theta\cos\varphi)\, U^x + (\sin\theta\sin\varphi)\, U^y + (\cos\theta)\, U^z \ , \tag{F.3}$$

$$U^\theta = \left(\frac{\cos\theta\cos\varphi}{r}\right) U^x + \left(\frac{\cos\theta\sin\varphi}{r}\right) U^y - \left(\frac{\sin\theta}{r}\right) U^z \ . \tag{F.4}$$

Thus, we see that U^r is also smooth everywhere, but U^θ diverges when $r \to 0$, while U^φ diverges when $r\sin\theta \to 0$. However, if we define

$$U^r \equiv u^r \ , \tag{F.5}$$

$$U^\theta \equiv \frac{1}{r} u^\theta \ , \tag{F.6}$$

$$U^\varphi \equiv \frac{1}{r\sin\theta} u^\varphi \ , \tag{F.7}$$

then the new vector u^i is guaranteed to be smooth everywhere as well.

We thus define a scale factor, S^i, which, in spherical coordinates, is given by

$$S^i = (1, r, r\sin\theta) \ , \tag{F.8}$$

and say that we *rescale* the vector U^i, component-by-component, using

$$u^i = S^i U^i \ , \tag{F.9}$$

where no summation of indices is performed. This means that we have

$$u^r = S^r U^r \ , \tag{F.10}$$

$$u^\theta = S^\theta U^\theta \ , \tag{F.11}$$

$$u^\varphi = S^\varphi U^\varphi \ . \tag{F.12}$$

Notice that a covariant vector, V_i, is rescaled using

$$v_i = \frac{V_i}{S^i} \ , \tag{F.13}$$

a rank-2 contravariant tensor, T^{ij}, is rescaled using

$$t^{ij} = S^i S^j T^{ij} \ , \tag{F.14}$$

a rank-3 mixed tensor, Ξ^i_{jk}, is rescaled using

$$\xi^i_{jk} = \frac{S^i}{S^j S^k} \Xi^i_{jk} \ , \tag{F.15}$$

and so on. In all these relations *we are not performing index summation.* It should be clear, also, that scalars need not be rescaled.

When evolving the BSSN equations in curvilinear coordinates, we introduce a new set of *rescaled* BSSN variables, and evolve them in time instead. This new set of variables, usually coupled with cell-centered grids, can be used to perform robust time evolutions that avoid coordinate singularities. Notice that the spatial discretization is performed for the *rescaled* variables, so that (we know the notation is confusing here, but remember that there is no index summation)

$$\partial_j U^i = \partial_j \left(\frac{u^i}{S^i} \right) = -\frac{u^i}{(S^i)^2} \partial_j S^i + \frac{1}{S^i} \partial_j u^i \ , \tag{F.16}$$

is the actual equation on which we apply finite differences. For notational sake, the BSSN rescaled variables we use in `NRPy+` are called, respectively,

$$\left(\bar{\gamma}_{ij}, \bar{A}_{ij}, \phi, K, \bar{\Lambda}^i, \alpha, \beta^i, B^i \right) \to \left(h_{ij}, a_{ij}, \phi, K, \lambda^i, \alpha, \mathcal{V}^i, \mathcal{B}^i \right) \ . \tag{F.17}$$