The Pennsylvania State University

The Graduate School

Eberly College of Science

DYNAMIC SINGULARITY EXCISION IN NUMERICAL RELATIVITY

A Thesis in

Physics

by

Kenneth L. Smith

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2004

The thesis of Kenneth L. Smith was reviewed and approved* by the following:

Pablo Laguna
Professor of Physics
Professor of Astronomy & Astrophysics
Thesis Adviser
Chair of Committee

Abhay Ashtekar
Eberly Professor of Physics

Bernd Brügmann
Associate Professor of Physics

Steinn Sigurdsson
Assistant Professor of Astronomy & Astrophysics

Jayanth Banavar
Professor of Physics
Head of the Department of Physics

*Signatures are on file in the Graduate School.

Abstract

We present here recent developments in the technique of singularity excision. With the ultimate goal of developing a generic code which will evolve single or multiple-black hole spacetimes until late times (with reference to the typical dynamical timescales of the problem being studied), we focus our attention on the numerical treatment of the "inner" boundaries of the black hole problem. The physical singularities inherent to black hole spacetimes pose enormous difficulties to the numerical implementations of the evolution equations, and, in general, should not be allowed to be part of the computational domain of evolution. Of the several methods available for enforcing this, singularity excision is one of the most straightforward to apply. We discuss the background, implementation, and results of our dynamic singularity algorithm which allows for movement of black holes through the computational grid and allows for long-lived evolutions of single and binary black hole spacetimes under certain circumstances to be discussed within.

Table of Contents

List of Figures

Notation/Conventions

In this work, we will adopt the following notation and conventions. The spacetime metric will have the signature $(-+++)$. Wherever possible, we will adopt the abstract index notation with Latin indices ranging over the letters a–h and o–z, such that the symbol v^a will denote a vector without reference to any particular basis. Occasionally, where it aids in the discussion, we will implement the MTW-style of denoting geometrical objects in boldface without indices, *i.e.*, $\mathbf{\Omega}$ or $\mathbf{d}\tau$.

Whenever we do introduce a basis, we will use Greek letters μ–σ to indicate the components of tensors on a spacetime basis in which the indices run over $0, 1, 2, 3$, and Latin letters in the middle of the alphabet i–n to indicate the components of tensors on a purely spatial, three-dimensional basis in which the indices will run over $1, 2, 3$.

The typical ADM quantities, the lapse, the shift, the spatial (three-)metric, and the extrinsic curvature will be denoted, respectively, by α, β^a, γ_{ab}, K_{ab}.

The Riemann tensor, Ricci tensor, and Ricci scalar associated with four-dimensional spacetime will be denoted, respectively, by ${R_{abc}}^d$, R_{ab}, and R, while the same tensors associated with a spatial slice will employ a sans-serif font for the 'R', ${\mathsf{R}_{abc}}^d$, R_{ab}, and R.

Preface

Over the past few years, I have worked with a talented group of computational physicists on developing a three-dimensional code known as Maya for numerically integrating Einstein's equations. As this is a group-built code with a handful of developers resulting in several-authored publications, I wish to take this opportunity to single out my contributions which form the central topic of this work.

First and foremost, I have, under the guidance of my thesis supervisor, designed and implemented the dynamic singularity excision algorithm which is discussed in detail in Chapter 4 of this work. This technique represents a novel contribution to the field of numerical relativity in which we allow for the possibility of having a black hole dynamically move to new coordinate locations in time, thus moving through the computational grid. Allowing for this possibility requires a substantial development of the infrastructure of the code, which I have implemented as a module in the Maya code known as `GridPointLabel`.

In order to test this algorithm, I have also implemented a time-dependent analytic solution to Einstein's equations in which a time-varying spatial translation is applied to the common Schwarzschild solution in Ingoing Eddington-Finkelstein coordinates. By forcing a single black hole to move throughout the computational grid along a pre-determined path, this module, named `MovingIEF`, has been invaluable as a proving-ground for the dynamic singularity excision algorithm. I discuss this method in §5.2.2.

The natural next step after creating an application in which the black hole and its associated excision region was required to move along a pre-determined path was to generalize the infrastructure so that the location of the hole could be determined "on the fly" by the physics being simulated. One can think of several criteria by which one might determine the location of a black hole: monitoring the location of the apparent horizon (*cf.* §2.4.2), following a pre-determined trajectory (as with `MovingIEF`), or monitoring some fall-off condition of some of the field variables. I developed the infrastructure to allow for these methods in terms of another module in the code known as `BHInfo`, whose purpose is to collect information about the location of the black hole from a set of possible black-hole "tracker" methods.

I also proposed a tracker method which monitored the fall-off of a field variable as a way of "sensing" where the black hole was located, but the implementation of my idea proved not to be fruitful. Ultimately, Ulrich Sperhake was able to develop a similar algorithm which we have used with a great deal of success. I discuss both my failed method and Sperhake's "GaussTracking" method in §4.3.

While I have maintained and developed many other components of the Maya code, these

are the most relevant and important contributions for the purpose of the research presented herein. To summarize, I have (1) developed a novel dynamic excision algorithm which allows us to move our excision region through the computational domain, (2) created a testbed application which creates this motion along a pre-determined path, and (3) implemented a sophisticated infrastructure which allows any generic "tracker" method to inform the excision algorithm how it should move the excision region.

Acknowledgments

Both during the work that led to, and the writing of this thesis, I have received an enormous amount of support and encouragement from friends, family, and colleagues. The support of two people deserves special noting. First, I would particularly like to thank my supervisor, Pablo Laguna, for his support, advice, and enthusiasm (which was never in short supply) in the research which forms the subject of this thesis. Second, I want to share my admiration and gratitude for my wife, Amanda, who has made this thesis possible through her love, patience, and understanding.

I am indebted to Ulrich Sperhake for the massive amount of work he put into learning the intricacies of the Maya code and the diligence with which he has brought this code to its current state of stability and flexibility. Much of this work would not have been possible had it not been for his careful eye and attention to detail.

I would also like to express my sincerest appreciation to

- my parents, for their never-ending love and support,
- Erik Schnetter, for giving birth to the Maya code,
- David Fiske, for his help in breaking the Cactus mold and making Maya more modular,
- Deirdre Shoemaker, for protecting me when I needed it,
- Bernard Kelly, for being a friend, compatriot, patient listener, and valued colleague,
- Marcus Ansorg, Ian Jones, Badri Krishnan, Patrick Sutton, and Manuel Tiglio for many interesting scientific (and non-scientific) discussions, and
- the friendly waitstaff at the All-American Rathskeller for their constant supply of liquid sanity.

I thank everyone who read early drafts of this work for their comments and corrections.

Finally, I would like to thank Pablo Laguna, Bernd Brügmann, and the Center for Gravitational Wave Physics at Penn State for their contributions to the computational hardware that has made development of our code possible. The work which forms the subject of this thesis was largely funded by the National Science Foundation under grant PHY 01-14375.

For my grandfather, Harold Lee Smith,
who set me on my career path the
first time he called me "Professor."

Chapter 1

Introduction

Physics, to some degree or another, is the science of predicting the future.

Simply put, a very large portion of physics research is concerned with *time evolution*, the description of how a given physical system at an instant of time will change as time marches on. From the early days of the "natural philosophers," we have wanted to understand the principles by which the universe operates, and there is no better test of our understanding than to see if we can accurately predict "what comes next."

From the point of view of classical physics, this is a fairly straightforward procedure. First, we must describe our system, whether it be a sub-atomic particle, the air inside a kiln, a distant solar system, or the observable universe, as precisely as we possibly can at a single instant of time. The set of variables necessary for this description, of course, varies depending upon what is being studied. It may be the temperature of the kiln, it may be the energy of the subatomic particle, it may be the positions and velocities of the planets in the solar system. Whatever the description, we attempt to pin down precisely what the system is doing at that instant.

Let us consider a very simple system for the sake of concreteness. We are handed a photograph in which we see an elderly, fuzzy-haired man standing in the middle of an apple orchard, his hand reaching slightly forward with his palm open to the sky. His eyes are staring at an apple, seemingly suspended in mid-air roughly half a meter above his hand. Looking just at this photograph (and presuming that a strange new phenomena that causes apples to levitate is not at work here), we cannot say exactly what was happening just before or just after this photograph was taken. It seems likely, given our everyday experiences, that the man was recently in the process of tossing the apple up into the air, or was about to catch it as it fell. With this single photograph, though, we do not know if the apple is on its way up or down. Obviously, then, just knowing where the apple "was," *i.e.*, its position, is not sufficient to determine "what comes next."

If we are now handed a photograph taken a mere instant after the first picture, we have something to go on. We can look very closely, and perhaps we can see that in the second picture, the apple is slightly higher than it was in the first. We have determined then that the apple is moving upward. If there is an accurate enough time-stamp on the pictures, we

can even estimate the apple's upward velocity.

In this simplistic example, these two pieces of information, the apple's position and its velocity, (if known to arbitrary precision) represent a sufficient set of variables that describe what our system was "doing" essentially at one instant of time. (Here, we consider the limit in which the time between the two photographs goes to zero.) The set of variables that completely describe what is happening at an instant of time represents what we call the **initial data** of the system.

Then, if we think we know the physical theory or theories well enough that describe interactions within our system, we can ask what those interactions predict for the subsequent evolution of the system, given specific initial data. In other words, we must determine the **dynamical theory** that governs the interactions between elements of our system. In the simple case of our flying apple, we would probably assume that the only interactions involving the apple might be the "pull of gravity" on it and perhaps some air resistance. Neglecting the air resistance, we would normally assume then that Newton's law of gravitation, describing the pull two massive bodies exert on one another, would be sufficient to calculate the path the apple will take as time advances from the moment the photographs were taken. We could even figure out, working backward in time, where it may have been shortly before the photographs were taken.

Given a set of initial data describing a system and the dynamical theory governing its interactions, determining the history of events before or after the initial conditions is just a matter of turning the mathematical crank and watching the results spew out (classically speaking, anyway). Of course, we are being grotesquely simple in these early, motivating remarks, but the ideas which we have put forth above lend themselves to the studies we undertake within this work.

Sometimes, the "mathematical crank" is not as easy to turn as one might like and one must settle for approximate solutions. When determining the future history of the system under study seems untenable by normal analytical techniques, we often resort to computational techniques and modern computer hardware to use "brute force" to tackle the problem. Non-perturbative, analytical techniques are almost always preferable to computational techniques because they allow one to calculate solutions to arbitrary precision. Oftentimes, though, a problem is so complex, that when one attempts to find solutions without overly restrictive assumptions and with generic initial data, computational techniques seem the only viable alternative.

Such, sadly, is the case with *our* studies of general relativity.

The theory put forth by Einstein, which pulls on the heartstrings of so many physicists because of its beauty and elegance, is a veritable nightmare when it comes time to perform

calculations. It is not simply that the theory is nonlinear. Others have dealt with nonlinearities before, for example, in studies of hydrodynamics. While it is certainly sad to see the dear old superposition principle of linear theories pack its bags and leave, we are brave souls and are willing to work with whatever nonlinearities present themselves along the way.

Rather, the difficulties in solving Einstein's equations do not stem simply from the mathematical hurdles. Understanding and interpreting the solutions to Einstein's equations are sometimes as difficult as finding them. (On the order of a half a century passed between the derivation of the first black-hole solution in general relativity and the understanding of what a black hole is. Some would argue that we still do not have a satisfying notion of how we should define a black hole.)

In attempting to treat Einstein's equations as a time-evolution problem like the simple Newtonian man-and-apple example from above, we are confronted with a most unusual problem. Usually, when we construct a time-evolution problem, we would describe that problem on some background. The very notion of "instant of time" implies some knowledge of what "time" is. Saying precisely what we mean by this in the context of general relativity takes a bit of work.

Still, assuming we can overcome these obstacles (and we can), the question remains, why might we wish to do so? Why should we be interested in casting Einstein's equations into the context of a time-evolution problem? Simply, because general relativity is our best candidate for *the* dynamical theory of gravity. We want to understand the implications of this theory, and we want to be able to predict, like those natural philosophers before us, "what comes next."

As we approach the centennial of the introduction of the special theory of relativity, it seems astonishing that, even with modern computer hardware and computational techniques, we are still struggling with the two-body problem in general relativity. For at least the past thirty years, various individuals and groups of researchers have attempted to solve this complicated problem numerically, with varying degrees of success. It is certainly safe to say that we are not there yet.

And so we continue to explore numerical simulations of general relativistic systems, out of our innate academic curiosity, but also because these simulations are on the verge of becoming very, very relevant to a much larger community. One of the most exciting elements to come out of Einstein's theory is the prediction of the existence of gravitational waves, and a nearly world-wide network of gravitational wave observatories sits poised, waiting to open a new window on our universe. Whether numerical simulations of highly relativistic systems will play a role in helping to detect these waves or simply help interpret them well after gravitational wave observations have become commonplace is difficult to predict. If history has taught us anything, it is to be cautious when making wagers on the course of scientific progress.

Thus, independent of the role our simulations may or may not play in gravitational wave observations, we will march on, resolved to obtain the best numerical simulations we can. We will find whatever techniques are necessary to extend the reliability of these simulations. We will investigate the myriad possibilities of formulations of general relativity and coordinate choices, and so on, until we have the solution within our grasp.

In the grand scheme of all this research and all of the different efforts committed to improving the state-of-the-art in the field of numerical relativity, we report in this work on a technique which we have found quite helpful in our calculations. It is by no means the final word on the matter, more like a piece in a great puzzle.

Simply put, we have developed a technique for treating the physical singularities that come along with simulations of black hole spacetimes. We will refer to this technique as "dynamic singularity excision." It is an extension of earlier works by different authors, but with it, and a collection of interesting time-dependent applications, we hope to provide a singularity-handling method that will eventually help contribute to longer, more stable, and more accurate simulations of astrophysically-interesting binary black hole spacetimes.

In this text, we will describe the foundation on which our work is based, both in terms of long-established results such as the standard 3+1 decomposition of general relativity and also more recent examples of singularity excision implementations that motivate our treatment. The organization of this thesis will be as follows.

In chapter two, (and a related, lengthy analysis in the first appendix) we will provide the necessary background for a researcher familiar with general relativity but not necessarily the intricacies of numerical relativity to follow the presentation herein. In particular, we will present a very brief summary of the standard 3+1 decomposition, the popular BSSN formulation, recent results on gauge stability in the context of numerical relativity and some elementary discussions of black hole horizons.

In chapter three, we will present an overview of singularity-handling methods in numerical relativity. We will describe some of the commonly used alternatives to singularity excision. This will form a nice backdrop against which we can present some of the historical developments in singularity excision techniques.

This leads to our own method, which we will describe in detail in chapter four. We will discuss the motivation for developing this method, and present its implementation in detail. We will end by discussing a novel method for estimating the location of a black hole when other methods are too costly or inaccurate.

In chapter five, we will present the details of our code and results computed with it. In particular, we will present simulations of single non-spinning black holes that last indefinitely when the black holes remain at fixed coordinate positions for all times. We also study scenarios in which the black holes are designed to move throughout the coordinate grid, and see how recent improvements in our understanding of gauge conditions has drastically

increased the lifetime of these runs.

Finally, we will briefly summarize our work in the final chapter and present some thoughts on future work.

Chapter 2

Theoretical Framework

In this chapter, we will briefly summarize the most important theoretical considerations employed in numerical relativity. Essentially, the program is to take Einstein's equations and collapse them down to two sets of equations, those valid at a "moment of time," and those which tell us how to evolve geometry from moment to moment. This construction is known as the 3+1 decomposition, and it will be presented here in brief, and in an appendix in great detail.

As many numerical experiments in recent years have demonstrated, the standard form in which this decomposition is performed, resulting in the so-called ADM system [12], is not ideal for numerical treatment. Thus, we will adopt the ADM system *only* as a common ground on which all discussion may be based, but we will actually implement one of the many analytically equivalent, yet numerically more stable, systems, the BSSN system [13, 65].

We will discuss some of the nuances associated with implementing this system. We will then briefly highlight the results of a recent work by Khokhlov and Novikov [48] that offers guidance for numerical relativity by pointing out that some of the simple gauge choices one might make are, in fact, ill-posed. This will explain difficulties experienced in our earlier simulations [66] and offer suggestions for alternative treatments, for which we will discuss the results in §5.

Finally, we will collect a few definitions and analytic expressions regarding concepts of immense use to us in our studies of black hole spacetimes, namely horizons. Of these, the two most important to consider are the event and the apparent horizon, defined below.

As will also be the case in the next chapter, no new results will be stated here; we will simply review those concepts a reader versed in general relativity may need in order to appreciate some of the details of the results stated later.

2.1 The 3+1 Decomposition

Most texts introducing students to the revolutionary ideas of special and general relativity make a special point of stressing that we live in a four-dimensional universe, *spacetime*, and that the previously dearly held concepts of space and time as separate notions are, in the words of Minkowski, "doomed to fade away into mere shadows" (quoted in [53]).

It is somewhat humorous, then, given this emphasis, that in order to utilize Einstein's equations to predict new phenomena for which no closed-form solutions exist, we find it so useful to reintroduce time as a unique and special dimension. We thus carve spacetime (which, for our purposes, will always be considered to be a four-dimensional manifold, $\mathcal{M}$, and a Lorentz-signature metric, g_{ab}) into a collection (or foliation) of non-intersecting spacelike hypersurfaces $\{\Sigma\}$, parameterized by a scalar field t, which we will use as a time coordinate. This simple treatment yields a fairly rich mathematical development which allows us to recast Einstein's equations into what is known as the "3+1" decomposition.

This elegant formalism is such an elementary part of so many disciplines within gravity research, that presentations of it are almost ubiquitous. As the derivation of the relations are quite interesting (and involved), we have chosen to provide a complete derivation of the 3+1 decomposition. Due to the fact, though, that we have worked out essentially all of the intermediate steps explicitly, this presentation is quite lengthy and so we have decided to place the derivation in an appendix of this work. Please refer to App. A for the full treatment; here, we will summarize the major results.

The standard 3+1 decomposition provides us with an initial-value formulation of general relativity, in which we must provide as initial data a three-dimensional manifold Σ, a three-dimensional metric γ_{ij} with Euclidean signature defined on Σ, and a symmetric rank-two tensor known as the extrinsic curvature, K_{ij}.

Similar in many ways to the treatment in electrodynamics, the initial-value formulation of general relativity provides us with gauge freedom, constraint equations that must be satisfied at every point in time, and evolution equations. The gauge freedom will be prescribed via a scalar function known as the lapse, α, and a spatial vector field known as the shift, β^i.

The constraint equations are time-independent equations that must be satisfied if the set $(\Sigma, \gamma_{ij}, K_{ij})$ is to be a solution of Einstein's equations. For the case of vacuum spacetimes, $T_{ab} = 0$, they are given by

$$\mathsf{R} + K^2 - K_{ij}K^{ij} = 0, \tag{2.1}$$

$$D_j K_i{}^j - D_i K = 0, \tag{2.2}$$

which are known, respectively, as the Hamiltonian and momentum constraints. In the above, K is the trace of the extrinsic curvature, $K = \gamma^{ij}K_{ij}$, D is the covariant derivative operator compatible with γ_{ij}, and R is the Ricci scalar of the space (Σ, γ_{ij}).

The evolution equations will advance the spatial metric and extrinsic curvature from one slice of constant coordinate time, t, to another. These equations are given, again for

vacuum, by

$$\frac{\partial}{\partial t}\gamma_{ij} = -2\alpha K_{ij} + \pounds_{\boldsymbol{\beta}}\gamma_{ij}, \tag{2.3}$$

$$\frac{\partial}{\partial t}K_{ij} = -D_i D_j \alpha + \alpha\Big[\mathsf{R}_{ij} - 2K_{ki}K_j{}^k + K_{ij}K\Big] + \pounds_{\boldsymbol{\beta}}K_{ij}. \tag{2.4}$$

In the above, R_{ij} is the Ricci tensor associated with (Σ, γ_{ij}), and $\pounds_{\boldsymbol{\beta}}$ is the Lie derivative along the shift vector β^i. The equations (2.1)–(2.4), together, are typically known as the ADM equations [12].

The general method of performing an evolution using these equations numerically is then as follows. On a given three-dimensional manifold (which is quite often taken to be $\mathbb{R}^3$), one specifies γ_{ij} and K_{ij} satisfying the constraints, Eqs. (2.1), (2.2). We will not consider, in any great detail, the issue of constructing suitable initial data in this work. For a summary of methods in initial-data construction, the reader is referred to the summary by Cook [35].

With initial data in hand, one must prescribe a lapse and shift so that one may evolve into the future. These functions are freely specifiable and represent the coordinate freedom in general relativity. Their role is purely to determine how the coordinates on a slice with constant coordinate time t_0 map into coordinates on the next, infinitesimally advanced slice with constant coordinate time $t_0 + \delta t$. The lapse allows one to specify how much proper time elapses at a given point for evolution along a direction (in spacetime) orthogonal to the spatial slices and the shift allows one to move around the spatial coordinates within the spatial slices. See Fig. A.4 in App. A for an illustration of this freedom.

Finally, then, the evolution equations, (2.3), (2.4) provide a method for determining the spatial metric and extrinsic curvature on the next coordinate-time slice given the geometry on the current slice. In the numerical implementation, we will approximate the right-hand-sides of these equations with a suitable technique (either finite-differences as discussed herein, or pseudo-spectral methods, finite-element approximations, *etc.*). We will then numerically integrate each point forward in time to the next time-slice according to the approximated right-hand-sides, and repeat. We will discuss the numerical issues in more detail in §5, but for now, this completes our specification of the program.

The ADM system formed the basis for most simulations in numerical relativity for close to a quarter of a century. Many of the codes implementing these equations had difficulty generically evolving spacetimes to late times. In [3] it was shown, in a linearization around flat-space, that the ADM system admits as solutions gauge and constraint-violating modes that do not propagate (so-called zero-speed modes). It is conjectured there that these zero-speed modes ultimately cause the observed instabilities in ADM evolutions because in the linearized solution, they correspond to an extrinsic curvature that is constant in time with a spatial metric that grows linearly in time. It is thought that, in the fully-nonlinear evolutions, these linear growths are amplified by the nonlinearity, leading to a blow-up of

the solutions.

Around the dawn of the new millennium then, the general failure of three-dimensional simulations implementing the ADM equations to provide long-term stable and convergent results led many groups to explore alternative formulations of the 3+1 formulation. Of the many formulations that began appearing, they could generally be divided into two groups, (1) those that were explicitly hyperbolic, for which the wealth of theoretical knowledge regarding uniqueness, well-posedness, and so on, was predicted to yield substantial improvement and (2) modifications of the ADM system, especially those that focused on conformal-traceless (CT) decompositions of the ADM system. Of the latter category, one system in particular has become exceedingly popular in the numerical relativity community, namely the BSSN system.

2.2 The BSSN system

Originally introduced in 1995 by Shibata and Nakamura [65], and modified slightly in 1999 by Baumgarte and Shapiro [13], this system has at its heart a conformal-traceless decomposition of the ADM equations. (This system should probably be referred to as the 'SNBS' system, but seems to have first picked up the abbreviation 'BSSN' in [4], despite the reversed chronological order of the authors. This ordering of the authors, however, does facilitate the pronunciation of the abbreviation.) Transverse-traceless and conformal-transverse-traceless decompositions are well-known in the context of attempting to isolate the physical degrees of freedom in general relativity or the polarizations of a gravitational wave signal. One might argue that the conformal-traceless decomposition applied in constructing the BSSN system plays a similar role.

The crucial element of the construction, though, seems to be its introduction of certain multiples of the constraint equations (which would be satisfied analytically for valid initial data) into the evolution equations. This has the effect of causing, in particular, the violations of the momentum constraint to obey a form of the wave equation, which was noticed already in the linearization around flat space in [3]. (Also, the effect of the introduction of the constraints into the evolution equations and the subsequent effect on evolutions of the constraint violations are illustrated very clearly in an analogy with electrodynamics in [49].) In a recent work by Gentle *et al.* [43], deriving evolution equations for the constraint violations was proposed, and the effective implementation of this idea in the BSSN system was demonstrated.

Heuristically, then, one can argue that the BSSN behaves as well as it does because it has, in some sense, highlighted the physical degrees of freedom though the conformal-traceless decomposition, and has introduced a method by which violations of the constraints may leave a computational domain by causing them to obey a wave equation. It is sometimes

said that the BSSN system has a certain "hyperbolic" flavor, although it has only been demonstrated that it is *equivalent* to a certain class of strongly hyperbolic systems in a study of an enlarged BSSN system in linearizations about flat space [58].

At the end of the day, it is the empirical evidence of improved stability behavior that brings so much attention to the BSSN system (*cf.*, for instance, [4] and references therein). Why this system behaves so much better than the ADM system is becoming better understood. For our purposes, its worth has been demonstrated, and we will adopt it for all of the work to follow.

Let us now go about establishing the equations that describe the system and that are implemented in our three-dimensional evolution code, MAYA. We can easily define the BSSN quantities in terms of the typical "ADM" quantities presented in the previous section,

$$\phi = \tfrac{1}{12} \ln \left(|\boldsymbol{\gamma}| \right), \tag{2.5}$$

$$\widetilde{\gamma}_{ij} = e^{-4\phi} \gamma_{ij}, \tag{2.6}$$

$$K = \gamma^{ij} K_{ij}, \tag{2.7}$$

$$\widetilde{A}_{ij} = e^{-4\phi} \left(K_{ij} - \tfrac{1}{3} \gamma_{ij} K \right), \tag{2.8}$$

where $|\boldsymbol{\gamma}|$ is the determinant of the physical three-metric, $\widetilde{\gamma}_{ij}$ is the conformal three-metric, conformally scaled such that its determinant is unity, K is simply the trace of the extrinsic curvature, and $\widetilde{A}_{ij}$ is the conformally-rescaled trace-free portion of the extrinsic curvature. In addition to these variables which are just natural elements of the conformal-traceless decomposition, the BSSN system also introduces a new "conformal connection" variable, given by

$$\widetilde{\Gamma}^i = \widetilde{\gamma}^{jk} \widetilde{\Gamma}^i{}_{jk}, \tag{2.9a}$$

$$= -\widetilde{\gamma}^{ij}{}_{,j}, \tag{2.9b}$$

where the second version comes from the fact that the conformal three-metric has unit determinant ($|\widetilde{\boldsymbol{\gamma}}| = 1$). This variable is introduced as an auxiliary variable, but promoted to an evolved field, complete with its own evolution equation derived by permuting a time-derivative with its definition above and replacing the time-derivatives of the other variables where they appear.

Given the introduction of these variables, one can then derive expressions for the evolution and constraint equations entirely in terms of the BSSN variables. We will not work through the results here, merely citing the results (again, for vacuum spacetimes).

The constraint equations, in terms of the BSSN variables, become

$$R - \widetilde{A}_{ij}\widetilde{A}^{ij} + \tfrac{2}{3}K^2 = 0, \tag{2.10}$$

$$\widetilde{D}_j\widetilde{A}^{ij} + 6\widetilde{A}^{ij}\widetilde{D}_j\phi - \tfrac{2}{3}\widetilde{\gamma}^{ij}\widetilde{D}_j K = 0. \tag{2.11}$$

The evolution equations take on the form

$$\frac{\partial}{\partial t}\phi = -\tfrac{1}{6}\alpha K + \pounds_{\boldsymbol{\beta}}\phi, \tag{2.12}$$

$$\frac{\partial}{\partial t}\widetilde{\gamma}_{ij} = -2\alpha\widetilde{A}_{ij} + \pounds_{\boldsymbol{\beta}}\widetilde{\gamma}_{ij}, \tag{2.13}$$

$$\frac{\partial}{\partial t}K = -e^{-4\phi}\widetilde{\gamma}^{ij}\left(D_iD_j\alpha\right) + \alpha\left(\widetilde{A}_{ij}\widetilde{A}^{ij} + \tfrac{1}{3}K^2\right) + \pounds_{\boldsymbol{\beta}}K, \tag{2.14}$$

$$\frac{\partial}{\partial t}\widetilde{A}_{ij} = e^{-4\phi}[-D_iD_j\alpha + \alpha\mathsf{R}_{ij}]^{\mathrm{TF}} + \alpha\left(K\widetilde{A}_{ij} - 2\widetilde{A}_{ik}\widetilde{A}^k{}_j\right) + \pounds_{\boldsymbol{\beta}}\widetilde{A}_{ij}, \tag{2.15}$$

$$\begin{aligned}\frac{\partial}{\partial t}\widetilde{\Gamma}^i &= -2\widetilde{A}^{ij}\alpha_{,j} + 2\alpha\left(\widetilde{\Gamma}^i{}_{jk}\widetilde{A}^{jk} + 6\widetilde{A}^{ij}\phi_{,j} - \tfrac{2}{3}\widetilde{\gamma}^{ij}K_{,j}\right) + \widetilde{\gamma}^{jk}\beta^i{}_{,jk} \\ &\quad + \tfrac{1}{3}\widetilde{\gamma}^{ij}\beta^k{}_{,jk} + \beta^j\widetilde{\Gamma}^i{}_{,j} - \widetilde{\Gamma}^j\beta^i{}_{,j} + \tfrac{2}{3}\widetilde{\Gamma}^i\beta^j{}_{,j},\end{aligned} \tag{2.16}$$

where we have used the notation that $[\cdot]^{\mathrm{TF}}$ denotes that the trace-free (with respect to either the physical or conformal metric) portion of the expression within the brackets should be used.

Because of the many factors of the determinant of the metric that have been introduced, one must be careful with the $\pounds_{\boldsymbol{\beta}}$ terms. (Some of these tensors have taken the form of tensor densities.) The Lie derivatives in the equations above are given by:

$$\pounds_{\boldsymbol{\beta}}\phi = \beta^k\phi_{,k} + \tfrac{1}{6}\beta^k{}_{,k} \tag{2.17}$$

$$\pounds_{\boldsymbol{\beta}}\widetilde{\gamma}_{ij} = \beta^k\widetilde{\gamma}_{ij,k} + 2\widetilde{\gamma}_{k(i}\beta^k{}_{,j)} - \tfrac{2}{3}\widetilde{\gamma}_{ij}\beta^k{}_{,k} \tag{2.18}$$

$$\pounds_{\boldsymbol{\beta}}K = \beta^k K_{,k} \tag{2.19}$$

$$\pounds_{\boldsymbol{\beta}}\widetilde{A}_{ij} = \beta^k\widetilde{A}_{ij,k} + 2\widetilde{A}_{k(i}\beta^k{}_{,j)} - \tfrac{2}{3}\widetilde{\gamma}_{ij}\beta^k{}_{,k} \tag{2.20}$$

The above equations, (2.12)–(2.20), determine the set of equations used to evolve from one time step to the next in the BSSN system. Some subtleties still remain, however, in how some of these terms are computed. For completeness, we present the precise expressions used in the Maya code to calculate some of the more complicated terms.

First of all, note that for notational convenience, the second covariant derivative of the lapse, $D_iD_j\alpha$, in Eqs. (2.14) and (2.15) has been left in terms of the *physical* three-metric-compatible derivative D_i. In practice, we want to be able to express all expressions in terms of only the BSSN (conformal) variables. Thus, we will use the derivative operator compatible with the *conformal* three-metric, $\widetilde{D}_i$, when calculating any covariant derivatives.

The expression $D_i D_j \alpha$ in these equations is then just a shorthand for

$$D_i D_j \alpha = \widetilde{D}_i \widetilde{D}_j \alpha - 4\alpha_{,(i}\phi_{,j)} + 2\widetilde{\gamma}_{ij}\widetilde{\gamma}^{kl}\alpha_{,k}\phi_{,l}. \tag{2.21}$$

Also for notational convenience, we have left the *physical* three-dimensional Ricci tensor R_{ij} and Ricci scalar R in Eqs. (2.10) and (2.15). Again, during an evolution, we wish to concern ourselves only with the BSSN quantities and not have to continually re-calculate the ADM (physical) ones. Thus, the physical Ricci tensor in that equation is again considered to be just a shorthand for

$$\mathsf{R}_{ij} = \widetilde{\mathsf{R}}_{ij} + \mathsf{R}^{\phi}_{ij}, \tag{2.22}$$

where R_{ij} is the Ricci tensor associated with the physical metric (γ_{ij}), $\widetilde{\mathsf{R}}_{ij}$ is the Ricci tensor associated with the conformal metric ($\widetilde{\gamma}_{ij}$), and R^{ϕ}_{ij} is a set of terms which relate the Ricci tensors of two conformally related metrics (found, for example, in App. D of [77]). For the sake of being explicit, these are given by

$$\widetilde{\mathsf{R}}_{ij} = -\tfrac{1}{2}\widetilde{\gamma}^{kl}\widetilde{\gamma}_{ij,kl} + \widetilde{\Gamma}^{k}\widetilde{\Gamma}_{(ij)k} + \widetilde{\gamma}_{k(i}\widetilde{\Gamma}^{l}{}_{,j)} + \widetilde{\gamma}^{kl}\left(2\widetilde{\Gamma}^{m}{}_{k(i}\widetilde{\Gamma}_{j)lm} + \widetilde{\Gamma}^{m}{}_{ik}\widetilde{\Gamma}_{mjl}\right), \tag{2.23}$$

$$\mathsf{R}^{\phi}_{ij} = -2\widetilde{D}_i\widetilde{D}_j\phi - 2\widetilde{\gamma}_{ij}\widetilde{\gamma}^{kl}\widetilde{D}_k\widetilde{D}_l\phi - 4\widetilde{\gamma}_{ij}\widetilde{\gamma}^{kl}\phi_{,k}\phi_{,l} + 4\phi_{,i}\phi_{,j}. \tag{2.24}$$

The Ricci scalar R may be calculated easily from $\mathsf{R} = e^{-4\phi}\widetilde{\gamma}^{ij}\mathsf{R}_{ij}$.

Equation (2.23) for the conformal Ricci tensor just follows from the standard expression written in terms of the metric and and Christoffel symbols, with the "conformal connection" variables ($\widetilde{\Gamma}^{i}$) introduced where appropriate. This serves to make the conformal Ricci tensor have only one term containing an explicit second partial derivative of the conformal metric, namely, $\frac{1}{2}\widetilde{\gamma}^{kl}\widetilde{\gamma}_{ij,kl}$. This "Laplacian" term appears in the right-hand side of the evolution equation for $\widetilde{A}_{ij}$, Eq. (2.15), which itself appears in the right-hand side of the evolution equation for $\widetilde{\gamma}_{ij}$, Eq. (2.13). Thus, with this choice of variables, the conformal metric evolves roughly according to a wave equation (with a complicated mess of source terms).

In the calculation of the conformal Ricci tensor, it has been observed [4, 79] that the behavior of the system changes according to exactly where the conformal connection variables ($\widetilde{\Gamma}^{i}$) are used. Specifically, in the term $\widetilde{\Gamma}^{k}\widetilde{\Gamma}_{(ij)k}$ in Eq. (2.23), one can use the variables as written, or replace them with their definition as contracted Christoffel symbols, Eq. (2.9a). In the Maya code, we can control this behavior with a Boolean parameter, `use_gamma_in_ricci`, which implements Eq. (2.23) as written when set to 'true' and replaces the un-differentiated $\widetilde{\Gamma}^{i}$ term with the contracted Christoffel symbols from Eq. (2.9a) when set to 'false.'

Finally, note that the conformal connection variables $\widetilde{\Gamma}^{i}$ appear in the right-hand side of their own evolution equation, (2.16), in the last term, $(+)\frac{2}{3}\widetilde{\Gamma}^{i}\beta^{j}{}_{,j}$. If all the other terms

on the right-hand side were negligible, and the divergence of the shift ($\beta^j{}_{,j}$) were positive, this would lead to exponentially growing solutions. To counter this, Yo *et al.* proposed a modification of the $\widetilde{\Gamma}^i$-evolution equation in which they add a multiple of the quantity $(\widetilde{\Gamma}^i - \widetilde{\gamma}^{jk}\widetilde{\Gamma}^i{}_{jk})$. From Eq. (2.9a), this quantity must vanish identically at the continuum level, but may be non-zero due to numerical error. Specifically, to the $\widetilde{\Gamma}^i$-evolution equation, Eq. (2.16), they add the term

$$-\left(\chi + \tfrac{2}{3}\right)\left(\widetilde{\Gamma}^i - \widetilde{\gamma}^{jk}\widetilde{\Gamma}^i{}_{jk}\right)\beta^j{}_{,j}. \tag{2.25}$$

By tuning the 'χ' parameter, they can eliminate or change the sign of the potentially exponentially growing term. In the Maya code, we have allowed for the use of this alternative evolution equation, referring to it as the "Yo trick."

While the BSSN system has become quite popular in recent years in numerical relativity, one of the difficulties in implementing it in a code is that the shorthand notations presented in many publications make it unclear exactly which equations are being solved. Analytically, the subtle differences vanish, and all implementations of BSSN are identical. With the errors introduced in the numerical methods, though, the differences in which the way the equations are written result in different observed properties of the simulations performed. We have attempted here to point out those subtleties that have had the greatest impact on the stability of our simulations and to present explicitly the form of the equations we use most commonly, along with the parameters that change that behavior. We will return to the details of how the equations are treated numerically in §5.

2.3 Gauge stability of 3+1 Formulations

As was mentioned in the first section of this chapter, there are three basic components to performing a numerical evolution of Einstein's equations: providing constraint-solving initial data, prescribing a lapse and a shift to build a foliation of the spacetime, and using the evolution equations to march forward in time. For the simulations considered in this thesis, initial data will be chosen to be analytic solutions of Einstein's equations, and thus, this step is trivial. For the evolution equations, we have committed ourselves to the BSSN system, with all the subtleties and issues described above. The remaining open issue, then, is how to prescribe the gauge (the choice of lapse and shift) during an evolution.

In a work by Khokhlov and Novikov [48], the impact of gauge choices is studied (independent of which particular 3+1 formulation is used), and certain classes of gauge choices are found to be ill-posed in the short-wavelength approximation. This provides us with a criterion by which we can exclude many gauge choices one might apply in a numerical simulation. Here, a well-posed problem is one for which the solution exists, is unique, and

depends continuously on the initial data. An ill-posed problem is simply defined to be a problem that is not well-posed.

In their work, they attempt to ascertain, independent of which 3+1 formulation is used, whether or not a given choice of gauge (lapse and shift) is well-posed at the analytical level. They do this by considering the impact of an infinitesimal coordinate transformation,

$$x^\mu \to x^\mu + \xi^\mu. \tag{2.26}$$

Under this transformation, the metric will transform as

$$g_{\mu\nu} \to g_{\mu\nu} + \delta g_{\mu\nu} \quad \text{with} \quad \delta g_{\mu\nu} = -\big(\nabla_\mu \xi_\nu + \nabla_\nu \xi_\mu\big). \tag{2.27}$$

Using the ADM line-element (*cf.* Eq. (A.78)), one can express the variations $\delta g_{\mu\nu}$ in terms of variations of the ADM quantities, $\delta\alpha$, $\delta\beta_i$, and $\delta\gamma_{ij}$. Comparing these variations with the variations in the metric, Eq. (2.27) above, one establishes a set of quasi-linear differential relations relating the variations in the ADM quantities and the coordinate transformation functions ξ^μ. (Note that these steps are common to first-order perturbation theory as well — no specialization has been introduced yet). The relations between the $\delta\alpha$, $\delta\beta_i$, *etc.*, and the ξ^μ can be re-written in the form

$$\frac{\partial \xi^\mu}{\partial t} = F^\mu \left(\alpha, \beta^i, \gamma_{ij}, \delta\alpha, \delta\beta^i, \xi^\nu, \xi^\nu{}_{,j}, \Gamma^\sigma{}_{\nu\rho}\right). \tag{2.28}$$

Again, this just follows from working through the definitions of the ADM quantities within the four-metric, and the variation of the four-metric under an infinitesimal coordinate transformation.

The benefit of their analysis is to compare the relations derived above with the actual choices made in numerical relativity for conditions on the lapse and the shift. Without loss of generality, these conditions can be written in the form,

$$G^\mu \left(x^\nu, \alpha, \frac{\partial\alpha}{\partial x^\nu}, \ldots, \beta_i, \frac{\partial\beta_i}{\partial x^\nu}, \ldots, \gamma_{ij}, \frac{\partial\gamma_{ij}}{\partial x^\nu}, \ldots, \right) = 0, \tag{2.29}$$

where the ellipses above indicate higher-order derivatives of the lapse, shift, and three-metric. Again, the relations G^μ will be provided by the numerical relativist. By varying this expression with respect to α, β_i, and γ_{ij}, an additional set of four quasi-linear relations can be obtained, and, when combined with Eq. (2.28), a total system of eight equations is available that describe the evolution of the coordinate perturbations ξ^μ and the corresponding perturbations of the lapse and shift in time.

As these relations are quite involved, Khokhlov and Novikov then proceed to look for solutions where the coordinate perturbations are considered to vary on a much shorter length

scale than the original four-metric (*i.e.*, taking the high-frequency limit). This allows for a relatively straightforward Fourier mode analysis (similar to a von-Neumann stability analysis). Those modes for which the real part of the eigenfrequencies are greater than zero will grow exponentially (this is still considered well-posed if the solutions depend continuously on the initial data). Ill-posed prescriptions will be those for which the real part of at least one of the eigenfrequencies tends to infinity. Thus their construction can determine whether someone's preferred choice of lapse and shift will lead to an ill-posed gauge.

In particular, they distinguish between three different types of gauge conditions, and then consider their well-posedness in turn. **Fixed** gauges are those for which the lapse and shift are prescribed functions of only the coordinates, *i.e.*, $\alpha = \alpha(x^\mu)$ and $\beta_i = \beta_i(x^\mu)$. **Algebraic** gauges are those in which the lapse and shift can also depend upon the three-metric and its spatial derivatives,

$$\alpha = \alpha(x^\mu, \gamma_{ij}, \gamma_{ij,k}, \ldots) \quad \text{and} \quad \beta_i = \beta_i(x^\mu, \gamma_{ij}, \gamma_{ij,k}, \ldots). \tag{2.30}$$

Finally, a **differential** gauge is one involving relations of α, β_i, and their derivatives that cannot be reduced to an algebraic gauge.

We will not touch upon the remainder of their derivations, other than to summarize their results for a few key gauge choices. First, they show that, of all the allowed *fixed* gauge choices, only gauges for which the shift is zero ($\beta_i = 0$) and the lapse is purely a function of the time coordinate ($\alpha = \alpha(t)$) are well-posed (they call such gauges "synchronous"). In many simulations, we have applied "analytic" gauges in which we choose the lapse and shift to be the functions of coordinates given by an analytic solution, such as the Schwarzschild solution in ingoing Eddington-Finkelstein coordinate (*cf.* §B.4). This is a "fixed" gauge in the terminology of [48], and thus is ill-posed. Using such analytic gauges had been empirically observed to behave poorly [4, 50], but in lieu of a more intelligent choice for dynamic spacetimes, we employed it in our early evolutions with dynamic singularity excision [66]. In that work, we had observed that for both simulations in which the black hole moved across the computational domain and simulations in which it remained fixed, our evolution times were limited to approximately $150M$. The work of Khokhlov and Novikov, then, has provided analytic support for the claim that analytic (fixed in their terminology) gauge conditions are poor choices for numerical work.

At the next level of complication, they also study algebraic gauge conditions. Again, these are conditions in which the lapse and shift can also depend upon the spatial metric and its derivatives. A very simple example of such a condition is the so-called densitized lapse condition, $\alpha = \alpha(x^\mu, |\boldsymbol{\gamma}|)$, with the shift a fixed function of coordinates $\beta_i = \beta_i(x^\mu)$.

With this class of gauges, they show that the densitized lapse will be ill-posed unless

$$\frac{\partial \alpha^2}{\partial |\boldsymbol{\gamma}|} > 0. \tag{2.31}$$

Based on this, and the experience of Laguna and Shoemaker in their studies of a BSSN-like system [50], we implemented a lapse condition

$$\alpha = |\boldsymbol{\gamma}|^{n/2} q, \tag{2.32}$$

where q can be a prescribed function of the coordinates, $q = q(x^\mu)$, or can be the solution to a differential equation. Here n typically takes on integer values, and q is what is referred to as the **densitized lapse**, as it is a scalar density of weight n. We see that the conditions derived in [48] for the well-posedness of such gauges requires that $n > 0$. It is exactly for this choice of parameters ($n = 1$ in particular) that we were able to achieve drastically longer evolution times in our moving-black-hole simulations [71].

For our purposes, the introduction of a densitized lapse was the simplest modification to the code that could be implemented within the classes that Khokhlov and Novikov found to be well-posed. It is extremely encouraging that such a condition worked as well as it did "out of the box." We will defer further discussion of this work until §5.

Let us conclude this discussion by noting one very disturbing aspect of the analysis of [48], namely, that their work only serves to provide a filter by which gauge conditions can be determined to be well- or ill-posed. Well-posedness, however, *does not guarantee stability*. They comment that the densitized lapse can be shown to be unstable for a "wide range of background solutions." While their results are extremely helpful in eliminating patently bad choices of gauge conditions, we are still in the situation that the true measure of a conditions' success will be found by implementing and performing numerical experiments. The proof is in the pudding, so to speak.

2.4 Black Hole Horizons

Turning our attention now from the process by which a numerical evolution of a spacetime is constructed, we will consider below how we discuss the black holes which we are interested in studying in our numerical experiments. In numerical relativity, we are so accustomed to dealing with physical variables that are local, *i.e.*, defined at a given instant of time, that we sometimes forget that not everyone thinks the way we do. An example of this is what we actually mean by the term "black hole." In the relativity community at large, a black hole is commonly defined as the region inside an **event horizon**.

As we will discuss below, however, the event horizon is an essentially useless concept

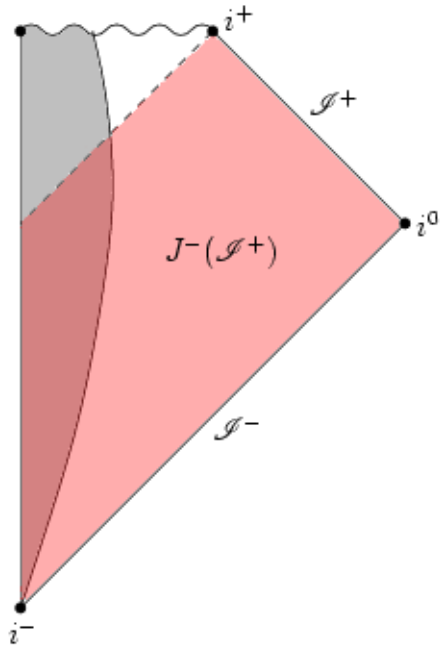


Figure 2.1: Conformal diagram for a black hole formed from the spherically symmetric collapse of a star (dark shaded region). Labeled are spatial infinity (i^0), past/future timelike infinity (i^-/i^+), and past/future null infinity ($\mathscr{I}^-/\mathscr{I}^+$). The event horizon is the dashed line, at the future boundary of $J^-(\mathscr{I}^+)$ (the causal past of future null infinity, lightly shaded here).

while one is performing the numerical calculations that are constructing the time-evolution of a spacetime. After we have finished the calculations (imagining a ludicrous limit in which we can perform our simulations for an infinite amount of time), we can then go back and talk about the event horizon and concretely say where the black holes were. During an evolution, though, the only real handle we have on the black hole is provided by the **apparent horizon**. We will define both of these terms, and discuss their utility in numerical simulations. In the following, we have found it helpful to refer to the text by Hawking and Ellis [45] and the Ph.D. thesis of Shoemaker [67].

2.4.1 Event Horizons

In simple terms, the event horizon is a boundary which separates what can be seen by arbitrarily distant observers from what cannot be seen. To be more concrete, it is the future boundary of $J^-(\mathscr{I}^+)$ (the causal past of future null infinity). An example of these terms is presented in a conformal diagram for the spherically symmetric collapse of a star in Fig. 2.1.

Because of this definition, the event horizon requires knowledge of the entire future

history of spacetime. A program for finding the event horizon could be summarized as: (1) wait for the end of the universe, then (2) retrospectively follow causal (timelike or null) geodesics backward in time from future null infinity, (3) determine which, if any, regions of the entire spacetime manifold were not covered by step two, and (4) label the boundary between the regions described in steps two and three as an event horizon. The event horizon is thus a *global* notion, and knowledge of the entire spacetime is required to locate it.

As an example, it is not inconceivable that we are inside an event horizon right now. A week, a month, or thousands of years from now, the curvature of spacetime in our immediate vicinity could increase (perhaps because of some infalling shell of matter that we have not yet suspected is on its way toward us), becoming great enough that null geodesics emitted here and now are focused back toward the source of the curvature such that they never "get away." The message here is that events in the (potentially distant) future will determine the true location of the event horizon.

Because of the arguments we will make later with regard to singularity excision, the event horizon does have an important role to play in the context of numerical relativity. If we know completely where an event horizon is, then we know that nothing inside that horizon can have any causal impact on the spacetime outside of it. This observation is the fundamental argument for implementing singularity excision — within an event horizon, one can get away with murder, never having to answer to the authorities out in the exterior spacetime. The prime failing, then, of the notion of the event horizon (from our perspective) is that we simply will not know where it is until everything is said and done. Thus, we turn our attention to the apparent horizon.

2.4.2 Apparent Horizons

While the notion of the event horizon has played a seminal role in the theoretical development of black hole physics, another type of boundary has also played an important role, namely, the concept of a **trapped surface**. A trapped surface is a smooth, compact, two-dimensional spacelike surface with the property that both sets (one set outgoing and one set ingoing) of future-pointing null geodesics orthogonal to the surface have negative **expansion**. There are quite a lot of pieces to this definition, so we will take them each in turn to see what is meant by a trapped surface and why it such a useful concept. In this section, we will generally adopt the terminology and definitions of Wald [77] (with the warning that we swap the labeling of outgoing and ingoing null vectors relative to that text).

Let us imagine a spherical surface S (or anything topologically a two-sphere), sitting in Minkowski spacetime $\mathcal{M}$ within a $t =$ const. spatial slice Σ. To keep things simple, let us say that this is the surface $r =$ const., with r the usual radial coordinate of spherical polar coordinates. At every point on this sphere, we can talk about two sets of orthogonal null rays, one pointing outward and one pointing inward. (We can imagine that our sphere S

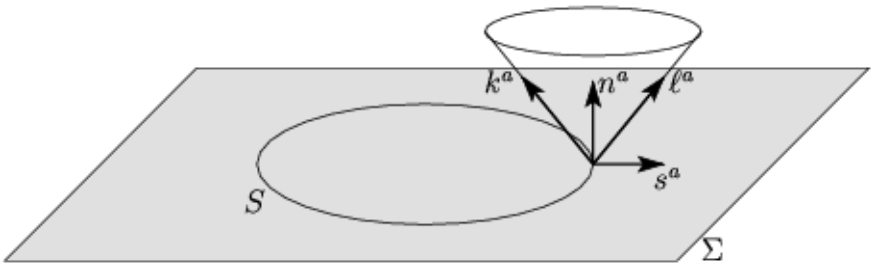


Figure 2.2: A spatial two-sphere S embedded in a spatial slice Σ (which is in turn, embedded in spacetime $\mathcal{M}$), with its two sets of orthogonal null vector fields. The vector field n^a is the unit timelike normal to Σ, s^a is the unit spatial normal to S, and ℓ^a and k^a are, respectively, the outgoing and ingoing null vectors orthogonal to S.

"fluoresces" for an instant, and we wish to follow the wavefronts of the light emitted from the surface as they propagate outward and inward.) With one spatial dimension suppressed, this surface and its orthogonal null vectors are depicted in Fig. 2.2, where we have labeled the outgoing null vector ℓ^a and the ingoing one k^a.

If we follow the null geodesics forward in time, instant by instant, we expect to see that the outward rays (geodesics along ℓ^a) will take us along a sequence of spheres increasing in radius (and area), while the inward rays (geodesics along k^a) will take us along a sequence of spheres whose radius (and area) decreases. Thus, we would say that the generators of the congruence of null geodesics orthogonal to the spacelike surface in the outgoing direction diverge while the generators in the ingoing direction converge.

Now, we imagine this same sphere of fixed radius sitting inside the black-hole region ($r < 2M$) of the Schwarzschild spacetime. Again, we consider the set of outgoing and ingoing null geodesics orthogonal to the surface S. Here, though, both sets of null rays emanating from S will follow a sequence of spheres whose radius and area decrease, because r is the timelike coordinate inside $r < 2M$, with r decreasing into the future. Thus, inside the event horizon, the outgoing and ingoing null vectors are both converging. More formally, we would say that the expansion θ of both sets of future-pointing null geodesics orthogonal to S is everywhere negative. This is what we mean by a "trapped surface."

With this physical interpretation in mind, let us then attempt to calculate the expansion of a congruence of null geodesics in terms of quantities readily available in a numerical simulation. By expansion, here, we are really asking, by how much do neighboring null vectors change along the surface of our two-sphere S? This is very similar to our discussion of the extrinsic curvature K_{ab} of spacelike slices embedded in a spacetime $\mathcal{M}$ presented in §A.4. There, we defined the extrinsic curvature as the the covariant derivative of the surface-orthogonal vector field, projected back into the surface. Here, we will do essentially the same thing.

The metric induced on the two-sphere S by its embedding in Σ will be

$$q_{ab} = \gamma_{ab} - s_a s_b, \tag{2.33}$$

where $\gamma_{ab} = g_{ab} + n_a n_b$ is the metric induced in the spacelike slice Σ by its embedding in $\mathcal{M}$, and s_a is the unit spacelike normal to S. Again, these vector fields and their relations are depicted for a single point on S in Fig. 2.2. As was also the case for our discussion of the 3+1 split, the induced metric q_{ab} functions as a projection tensor into the submanifold. Thus, we use the induced metric to project the divergence of the null vectors, defining the expansion θ of the geodesic congruence with orthogonal null vector ℓ^a to be

$$\theta_{(\ell)} = q^{ab} \nabla_a \ell_b. \tag{2.34}$$

(Compare this with the trace of the extrinsic curvature, $K = \gamma^{ab} K_{ab} = -\gamma^{ab} \nabla_a n_b$.) We could calculate the expansion of the ingoing null vectors, $\theta_{(k)}$, by substituting k^a for ℓ^a above.

In either case, if the expansion is positive, we say that the congruence is diverging, and if negative, we say that it is converging. A trapped surface is then a spacelike two-surface meeting the criteria defined above for which $\theta_{(\ell)} < 0$ and $\theta_{(k)} < 0$. A **marginally trapped surface** relaxes the above restriction such that the expansions are merely non-positive, *i.e.*, $\theta_{(\ell)}, \theta_{(k)} \leq 0$. The last building block in arriving at a clear definition of an apparent horizon is the definition of an **outer marginally trapped surface**, which is a two-dimensional surface S that is the boundary of a three-dimensional volume with the property that the expansion of the outgoing family of null geodesics orthogonal to S is everywhere non-positive, $\theta_{(\ell)} \leq 0$. In the common usage of the numerical relativity community, then, a closed, spacelike, two-surface is an **apparent horizon** if it is the *outermost* outer marginally trapped surface.

This final result thus offers a concrete definition for what we mean by a "black hole" at a single instant of time (on a single spatial slice of spacetime in our numerical evolution). As we will see below, it also provides us with a method by which we can search our spatial slice for these apparent horizons. In practice, one typically looks for a two-surface S in Σ where the outgoing, orthogonal null vectors ℓ^a have exactly vanishing expansion, $\theta_{(\ell)} = 0$. (Depending upon how carefully this search is done, one might then check for other surfaces with vanishing expansion farther out, or simply stop after one is found.)

We will briefly demonstrate how the zero-expansion condition can be re-written in terms of local-in-time quantities. Again referring to Fig. 2.2, we can, up to some scaling factor, write the outgoing null vector orthogonal to S as

$$\ell^a = n^a + s^a. \tag{2.35}$$

Then, the expansion, Eq. (2.34) becomes

$$\begin{aligned}\theta_{(\ell)} &= q^{ab}\nabla_a(n_b + s_b) \\ &= \left(\gamma^{ab} - s^a s^b\right)\nabla_a(n_b + s_b) \\ &= \gamma^{ab}\nabla_a n_b + \gamma^{ab}\nabla_a s_b - s^a s^b \nabla_a n_b - s^a s^b \nabla_a s_b \\ &= \gamma^{ab}\left(-K_{ab} - n_a a_b\right) + \gamma^{ab}\nabla_a s_b - s^a s^b\left(-K_{ab} - n_a n_b\right) - s^a s^b \nabla_a s_b,\end{aligned}$$

where we have used Eq. (A.25) from App. A to replace the projected $\nabla_a n_b$ terms. The spatial inverse metric γ^{ab} and unit normal to the sphere s_a are all spatial, and thus will vanish when contracted with n^a. Also, the remaining spacetime covariant derivative terms are all acting on and contracted with spatial tensors; thus, we can replace them with the spatial-metric-compatible derivative operator D_a. This leaves us with

$$\begin{aligned}\theta_{(\ell)} &= -\gamma^{ab}K_{ab} + \gamma^{ab}D_a s_b + s^a s^b K_{ab} - s^a s^b D_a s_b \\ &= -K + D_a s^a + K_{ab}s^a s^b - \tfrac{1}{2}s^a D_a\left(s_b s^b\right) \\ &= -K + D_a s^a + K_{ab}s^a s^b. \end{aligned} \tag{2.36}$$

The last term vanishes because $s_a s^a = 1$ everywhere on S, and hence the projected derivative vanishes. To find an apparent horizon in a numerically-generated spacetime, then, one looks for surfaces for which the expansion of the outgoing null vectors is exactly zero, yielding

$$D_a s^a - K + K_{ab}s^a s^b = 0 \tag{2.37}$$

Equation (2.37) is known in the numerical relativity community as the **apparent horizon (AH) equation**.

This equation is then in form suitable for implementation on a given time-slice, and thus is a way of defining the location and size of a black hole appropriate for numerical relativity. Still, the apparent horizon has some unsettling properties. For one, it is not necessary that every time-slice through a black hole spacetime need contain an apparent horizon. Also, apparent horizons can jump discontinuously between time-slices — note that this is not a problem with poor temporal resolution in our computational simulations; at the continuum level, the world-tube formed by an apparent horizon can be discontinuous.

Even with these caveats, in general relativity, the apparent horizon will always lie inside the event horizon, and thus is a solid local measure of the existence of a black hole. Because of this, and the ability to search for the apparent horizon using only three-dimensional quantities available within a single time-slice, we turn to the AH time and time again to give us a handle on where our black holes are located within a numerical simulation.

Let us conclude then by mentioning that several readily-available numerical implemen-

tations of apparent-horizon finders exist today that one can use freely to track horizons in a numerical spacetime [61, 75]. These finders are optimized and efficient, yet apparent-horizon finding in three dimensions is still a relatively slow process, especially when implemented as part of an evolution scheme. For this reason, in §4.3, we will construct a quick (and dirty) method of estimating the coordinate location of the black hole so that we know where to employ singularity excision. We will always verify periodically with these AH finders, though, that our excision region is, in fact, contained within an apparent horizon, and so it remains a very powerful and useful concept for us.

Having presented the most relevant elements of the theoretical background we need for numerical evolutions of black-hole spacetimes, we will move on to consider in the next chapter how we can treat the singular behavior of black hole solutions, before we go on in the subsequent chapter to detail our method for dynamic singularity excision.

Chapter 3

Singularity-handling techniques

Simply put, simulations of black holes in a fully-nonlinear, general relativistic setting suffer from a problem that does not plague simulations of other astrophysical bodies: the presence of **spacetime singularities**.

Without delving too deeply into the rich arena of mathematical relativity dealing with with singularities and singularity theorems, we will simply think of spacetime singularities as being points in spacetime beyond or through which null or timelike geodesics cannot be extended. (In other words, we are adopting the working definition that a spacetime with a singularity is not timelike- or null-geodesic complete). Most of the time, of course, we are thinking of the simple picture of the Schwarzschild $r = 0$ singularity or the Kerr ring singularity with their associated blow-up of the curvature scalars. As such, we may occasionally alternate through the descriptors "spacetime," "physical," and "curvature" to describe these singularities in the discussion to follow.

In contrast to these physical singularities, we are, in any study of general relativity, also confronted with the notion of *coordinate singularities*, *i.e.*, locations in coordinate space where calculations become degenerate or divergent, due solely to "bad coordinates." The prototypical example of this is the Schwarzschild radius $r = 2M$ in Schwarzschild coordinates. A radially-infalling observer takes an infinite amount of time in the Schwarzschild time coordinate to cross the Schwarzschild radius. Calculations of the proper time that the observer would actually measure, of course, show no such problem — an observer can happily cross the Schwarzschild radius and be none the wiser (provided the tidal forces do not crush him first). In some sense, one discovers that an apparent singularity is merely a coordinate singularity by finding that "nothing special" happens there.

Operationally then, coordinate singularities are just a nuisance, a bad choice. Physical singularities signify not a mere breakdown in coordinates, but rather a breakdown in our understanding of the physical universe. They are, in effect, endpoints to spacetime. The problem is that it is not always obvious when one encounters divergent results whether one is faced with a coordinate or a physical singularity. Over the years, rules of thumb have been developed that typically allow one to make these distinctions. Quite often, these involve checking whether certain curvature invariants remain finite or whether one can find

a clever way of extending geodesics beyond these putative singularities. From an analytic standpoint then, coordinate singularities are harmless; they can be transformed away by a "better" choice of coordinates.

From the standpoint of a numerical evolution of a general relativistic system, however, both types of singularities will wreak havoc on a simulation. In simple terms, computers can not handle calculations involving infinite terms. While the result of such a calculation will vary by architecture and machine-specific finite-precision representations of real numbers, one typically encounters "`Inf`" (meaning infinity) or "`NaN`" (meaning Not a Number). Not surprisingly, the result of a calculation of a `NaN` and a finite number is also a `NaN`. This will mean, in the context of an evolution code, that once an infinite quantity appears at any point on the computational grid, it will quickly propagate out to all other points. Obviously, this will not lead to the maximal development of the initial data that we are seeking in a numerical evolution, and so we will need to develop special methods for handling singularities such that they will not bring an end to our simulations.

Although this means that coordinate singularities can be just as damaging to the ability of a numerical code to evolve for long times as physical singularities, we will not devote much of our attention to them in this work. Constructing coordinate-pathology-avoiding conditions is by no means a trivial affair (this is still an active area of research), but we will concentrate on methods for treating physical singularities as they are a nastier breed of beast.

It should be mentioned, if only in passing, that this inclination to avoid curvature singularities in our numerical simulations is motivated by the physics we are interested in modeling. We are, by and large, interested in evolving the strong-field regions of highly dynamic, relativistic spacetimes so that we may determine what observers far away will be able to detect in the form of gravitational radiation. This is not to say that this is true for all numerical relativists — many researchers are interested in using numerical relativity as a tool for *studying* singularities, such as testing numerically if naked singularities can form under certain conditions or studying properties of cosmological singularities, just to name two examples. The interested reader is referred to a recent review article by Berger [16]. For this work, we will assume that physical singularities will be an undesirable yet unavoidable feature of our spacetimes and will go about discussing methods for handling them in our numerical simulations.

Regarding the numerical treatment of curvature singularities, we are essentially presented with two options. Either we can tailor the simulation such that the singularity be required to never intersect the computational time-slices (in which case the problem is effectively avoided altogether), or, if a time-slice does contain a singularity, we may employ causality arguments to isolate it from the numerical calculations.

The method which will be the focus of this work, namely singularity excision, falls into

the latter category. The method gets its name from the fact that a region containing the singularity is removed, or excised, from the computational domain. Depending upon the exact nature of the region excised, this may necessitate establishing boundary conditions on field variables outside the region. These issues will be discussed in detail herein.

The organization of this chapter will be as follows. We will briefly consider alternative singularity-handling techniques, highlighting their advantages and drawbacks. We will then present the general concept behind singularity excision and trace its historical development over roughly the past two decades. Among these developments, we will pause to highlight a particular treatment of singularity excision by Alcubierre and Brügmann known as "simple excision" [4] that forms the conceptual basis for our treatment. This will provide the historical backdrop against which we can present our dynamic excision algorithm in the next chapter.

3.1 Alternatives to singularity excision

In order to understand why singularity excision has become so popular of late, we find it important to consider what alternatives exist, weighing their advantages and weaknesses against the physics we are interested in studying.

As was mentioned above, only two options for handling the physical singularities of a spacetime exist. One can either excise them from the spacetime or ensure that they are avoided at all times during an evolution. In this vein, we will discuss the use of singularity-avoiding slicings which serve in this function. Because of the difficulty in handling coordinate singularities, we will also take this opportunity to discuss two particular techniques that are frequently used to do away with the coordinate singularities found in many black-hole solutions, namely throat/isometry conditions and puncture methods.

3.1.1 Singularity-avoiding slicings

Historically, the long-standing alternative to excision has been the use of singularity-avoiding slicings. The basic idea of this technique is that one can use the gauge or slicing freedom inherent to the 3+1 decomposition to control the flow of time across the grid, such that time slows to a halt in those regions approaching a singularity. In general, this means that the slicing conditions chosen during an evolution (provided in terms of the lapse and shift) cannot be prescribed functions of space and time, but must be able to respond to the changes in the geometry of the spatial slices as they evolve. These conditions are often motivated by physical considerations, such as attempting to ensure that the local volume element not be allowed to vanish. In general, then, these slicing conditions are designed to prevent coordinate singularities from forming, and their avoidance of physical singularities is a "bonus" property that is established afterward.

It stands to reason that, for any singularity-avoiding slicing condition to work, one must begin with initial data that does not already contain a singularity — a requirement that is typically not difficult to satisfy. Most general-relativistic simulations of matter begin with a perfectly nonsingular initial slice, and for black-hole spacetimes, various wormhole constructions are possible that do not intersect a physical singularity.

For instance, by introducing the coordinate transformation from the Schwarzschild radial coordinate r to a new coordinate $\bar{r}$ via $r = \bar{r}(1 + M/2\bar{r})^2$, one can write the Schwarzschild metric in spatially isotropic coordinates,

$$ds^2 = -\left(\frac{2\bar{r} - M}{2\bar{r} + M}\right)\mathbf{d}t^2 + \psi^4\Big(\mathbf{d}\bar{r}^2 + \bar{r}^2\mathbf{d}\theta^2 + \bar{r}^2\sin^2\theta\,\mathbf{d}\phi^2\Big). \tag{3.1}$$

where $\psi = \psi(\bar{r}) = 1 + M/2\bar{r}$ is the spatial conformal factor. From this it is immediately obvious that spatial metric is conformally flat, *i.e.* $\gamma_{ij} = \psi^4\delta_{ij}$.

In these coordinates, $\bar{r} = M/2$ corresponds to the Schwarzschild radius $r = 2M$, which is typically known as the 'throat' or Einstein-Rosen bridge [39] in these contexts. One can easily see that the isotropic radial coordinate, $\bar{r}$, defined on the interval $[0, \infty]$, covers the Schwarzschild radial coordinate r only over the interval $[2M, \infty]$. Effectively, this coordinate comes in from spatial infinity, reaches the throat at $\bar{r} = M/2$, and heads back out to spatial infinity. General relativity fixes the geometry, but not the topology, so an appropriate question is, "To which spatial infinity does the radial coordinate $\bar{r}$ return?"

To answer this, we note that these coordinates possess an isometry at the throat that maps $\bar{r} \in [0, M/2]$ to $\bar{r}' \in [M/2, \infty]$ by

$$\bar{r} \to \bar{r}' = \frac{M^2}{4\bar{r}}. \tag{3.2}$$

Under this transformation, every point inside the throat is mapped to a corresponding point outside and vice-versa. In fact, the origin of this coordinate system, $\bar{r} = 0$ is mapped to spatial infinity, $\bar{r}' = \infty$. This leads to the situation depicted in Fig. 3.1. It is obvious from the Kruskal-Szekeres diagram that the regions inside and outside the throat are causally disconnected from each other. Regardless of what happens in the "other" asymptotically flat spacetime (corresponding to $\bar{r} < M/2$), it can have no effect on "our" universe, $\bar{r} > M/2$. In this way, one can describe the spatial hypersurface corresponding to $t = 0$ as being two separate, three-dimensional, asymptotically flat spaces, spliced together at a common two-sphere.

We went through this in some detail because many of the various single or multiple black-hole initial data sets that one might construct share common features with the simple isotropic Schwarzschild example. Specifically, many of the initial-data constructions used over the years can be understood in the context of a multi-sheet topology, where the sheets

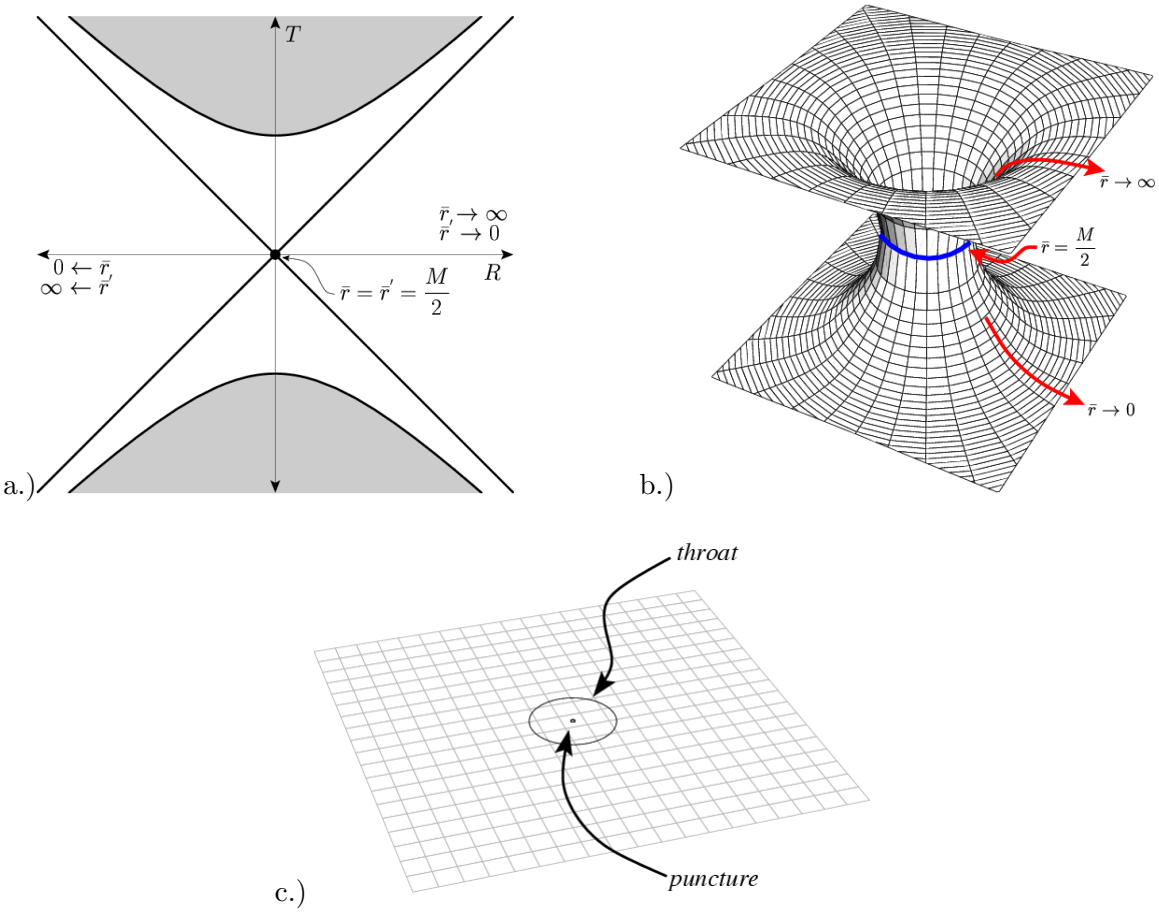


Figure 3.1: Spacetime structure of the Schwarzschild solution in the context of isotropic coordinates. a.) Kruskal-Szekeres diagram in which the $t = 0$ slice in isotropic coordinates coincides with the $T = 0$ slice. We see the inversion symmetry present at the throat as the radial coordinate $\bar{r}$ for "our" universe (sheet) at $\bar{r} = 0$ corresponds to spatial infinity in the connected asymptotically flat universe. In the Kruskal-Szekeres coordinates, null geodesics are represented by lines at $\pm 45°$; thus, we can immediately see that the two universes are causally disconnected. The throat is represented by the single point at the origin (recalling that each point on the diagram is a two-sphere). b.) An isometric embedding of the $t = 0$ hypersurface in a fictitious flat three-space (with one rotational dimension restored). c.) A representation of the numerical initial data (with one spatial dimension suppressed again). Fields are perfectly regular aside from a coordinate singularity at $\bar{r} = 0$ (corresponding, again, to spatial infinity of the other sheet), here indicated as the "puncture".

are joined at common, closed two-surfaces with minimal area, the throats. In general, these throats will lie somewhere *within* the event horizon and not coincide with it, as is the case for isotropic Schwarzschild. The total number of sheets involved and whether they are isometric across the throats are freedoms one has in constructing the initial data. In short, these multi-sheet or wormhole constructions provide initial data that do not contain physical singularities and, as they are typically expressed, have no more than n coordinate singularities, where n is the number of throats in the data. The reader is referred to Cook [35] for an extensive review of the construction of initial data, which we will not consider in detail here.

Having initial data thus free from physical singularities, one typically wishes to evolve it in time. In any general black-hole spacetime, though, one faces the possibility that during the evolution, one of the time slices might intersect a physical singularity and subsequently end the simulation. As has been mentioned in many contexts, evolving the initial data corresponding to $t = 0$ in the isotropic Schwarzschild coordinates using geodesic slicing, (*i.e.*, $\alpha = 1$ and $\beta^i = 0$), leads to a very rapid crash of the code as the evolution must intersect the $r = 0$ singularity within a finite proper (and coordinate) time of $\tau = \pi M$. In many numerical relativity groups, this fact is exploited as a test of the code: in these so-called "crash tests," the time at which the code crashes should converge to πM as the spatial and temporal resolution are increased [11, 25].

For a more interesting development of the spacetime, one generally wants to employ a singularity-avoiding slicing condition, *i.e.*, a condition which chooses the lapse (and possibly the shift), such that singularity-containing or singularity-developing regions are avoided. Here we discuss the perennial example, maximal slicing, wherein one requires that the trace of the extrinsic curvature vanish, $K = 0$.

As early as 1944, Lichnerowicz had advocated the use of coordinate anti-focusing conditions and proposed maximal slicing as an option, as it has the property of avoiding the vanishing of the volume element for observers moving normal to the slices (*cf.* [80] and references therein). Many years later, others would find, especially in the extended Schwarzschild spacetime, that maximal slicing is also a curvature singularity-avoiding condition [37, 41].

Heuristically, one can see that by requiring $K = 0$ at all times (which therefore also means that $\partial_\tau K = 0$), the convergence of observers normal to the slices will vanish. (Recalling the definition of the extrinsic curvature as the spatially-projected covariant derivative of these observers' four-velocity, $K_{ab} = \gamma^c{}_a \gamma^d{}_b \nabla_c n_d$, we find that, for the trace of the extrinsic curvature to vanish, we must have $\nabla_a n^a = 0$.) In other words, the observers will not converge (or diverge), effectively avoiding the focusing of any coordinates (provided the coordinates are also dragged normal to the hypersurfaces).

In any given numerical implementation, one obtains maximal slicing by choosing an initial slice satisfying $K = 0$ (a trivial condition if one adopts a moment of time-symmetry,

$K_{ij} = 0$, for the initial data) and then enforcing $\partial_\tau K = 0$ at all subsequent times. From the evolution equation for K, seen for instance, in the BSSN system of §2.2, one can take this condition to be an elliptic equation on the lapse, given (in vacuum) by

$$D^i D_i \alpha = \alpha K_{ij} K^{ij}. \tag{3.3}$$

Typically, this means that one must solve an elliptic equation on every time-step, or at least, fairly often, in order to have maximal slices. In three dimensions, even on modern computer hardware, this is a very expensive operation and is the most unsettling disadvantage of the method from the perspective of computational efficiency. Various work-arounds have been proposed, mostly in terms of converting the elliptic equation into a parabolic one, evolved in a "fake" time coordinate, such as Shibata's construction of the "approximate maximal slicing condition" [64],

$$\partial_\lambda \ln \alpha = D^k D_k \ln \alpha - (D_k \ln \alpha)(D^k \ln \alpha) - K_{ij} K^{ij}, \tag{3.4}$$

rewritten here for vacuum, and converting back to the typical ADM variables via $K_{ij}K^{ij} = \tilde{A}_{ij}\tilde{A}^{ij} + \frac{1}{3}K^2$. In the limit of large λ, this tends to the maximal slicing solution, and can be solved approximately during an evolution at a significantly reduced computational cost. For the Schwarzschild and Reissner-Nordström spacetimes, it also is possible to construct analytic solutions to the maximal slicing condition [15, 37, 56]. These are not typically used as replacements for the elliptic-solved conditions due to their complexity, but they do offer a fair deal of insight as to why the numerically computed maximal slices take on the features they do.

The stress here thus far has been on maximal slicing, as it has been perhaps one of the simplest conditions to understand conceptually, and was used quite extensively in the early one- and two-dimensional (and even three-dimensional, *cf.* [11]) black hole simulations. There do exist many alternatives, however, that do not require solving an elliptic equation. Harmonic slicing (*i.e.*, $\alpha = h(x^i)\sqrt{|\gamma|}$, where $h(x^i)$ is a time-independent function only of spatial coordinates and $|\boldsymbol{\gamma}|$ is again, the determinant of the spatial metric) is an example of an algebraic slicing condition that is singularity avoiding [19]. Among the class of hyperbolic slicing conditions, (*i.e.*, first-order-in-time differential equations on the lapse), the Bona-Massó [20] family of slicing conditions,

$$\partial_\tau \alpha = -\alpha^2 f(\alpha) K + \pounds_{\boldsymbol{\beta}} \alpha, \tag{3.5}$$

(with $f(\alpha)$ a positive, but otherwise unrestricted function of α) has become quite popular of late. Alcubierre [2] has performed an analytical study of this family for various choices of the function $f(\alpha)$ and in many cases finds that the conditions do tend to be anti-focusing

(coordinate-singularity avoiding).

Within this family of lapse conditions, one which has become a *de facto* standard in the community is the '1+log' condition, utilized recently, for instance in [4, 5, 6, 66, 71, 78]. This condition corresponds to the choice of $f(\alpha) = N/\alpha$ above. For this choice, one can integrate the differential relation and obtain an algebraic one, namely, $\alpha = h(x^i) + \ln|\boldsymbol{\gamma}|^{N/2}$, for any time-independent function $h(x^i)$. (Choosing this function h to be unity, one sees whence the name '1+log' came.) There is strong motivation to use this condition as it is computationally inexpensive, and empirically well-behaved for many black-hole spacetimes. In his analysis, Alcubierre has shown that '1+log' slicing is strongly singularity avoiding, defined in this context to mean that locally, the slices stop advancing a finite coordinate time before reaching a singularity, signaled by the vanishing of the volume element, $|\boldsymbol{\gamma}|^{1/2}$.

To illustrate both the benefits and the most severe drawback of these singularity-avoiding slicing conditions, we schematically illustrate in Fig. 3.2 the case of maximal slicing of the extended Schwarzschild solution. There, for the symmetric (even) case, the slices have the property that they slow the evolution normal to the slices as they approach the singularity. This means that the lapse α will tend to zero in these inner regions, a phenomenon known as the "collapse of the lapse."

Far away from the horizon-penetrating region, however, the lapse will generally be required to asymptote to its Minkowskian value of unity, and so time far away from the central region will evolve normally. This typically leads to "grid stretching" or "slice stretching," as the slices must become more and more distorted as they "stretch" back to connect the frozen-time regions with the outer, normal-time regions. The metric on the slices naturally develops large gradients, which will grow without bound, and eventually the numerics will fail to adequately resolve the gradients, at which point the code will crash. Increasing the spatial resolution can postpone these crashes, but hardware limits make this more difficult in three-dimensional simulations [11]. In his presentation of the first 3+1 numerical relativity code implementing AMR, Brügmann points out that with an adaptive re-gridding algorithm, one could continue to add resolution where the metric gradients are growing and hence potentially under-resolved [25]. Even with dynamic re-gridding, increased resolution can only buy time; this is a battle that cannot be won by brute force.

Singularity-avoiding slicings seemed doomed to lead to short-lived simulations, only buying singularity-avoidance at the price of slice-stretching. It was not until recently [5], that singularity avoiding slicings had been able to perform long-term black hole evolutions. There the key ingredient appears to be a non-vanishing shift vector which serves to "pull" grid points out of the region where the lapse is collapsing. Whereas the slice-stretching effects with zero shift typically limited earlier run-times (in three-dimensional simulations) to approximately 30–40M (in units of the black hole mass M), Alcubierre *et al.* were able to evolve a single black hole using a singularity-avoiding slicing condition ('1+log') and a

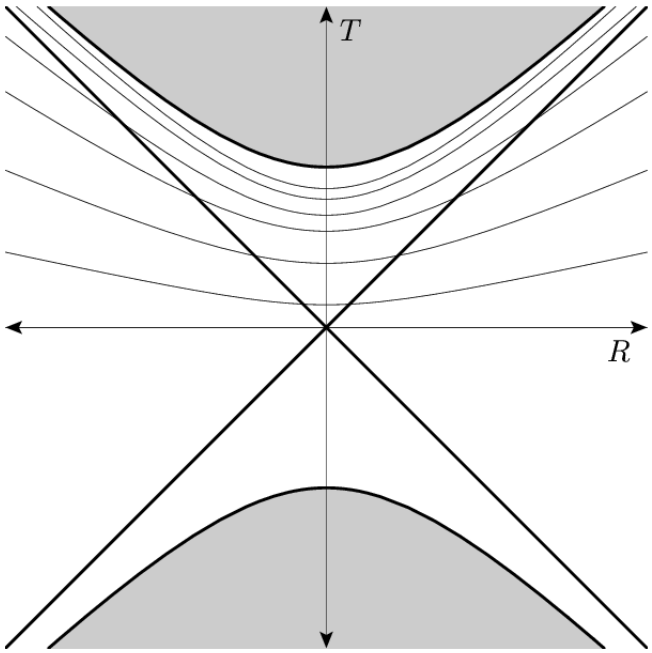


Figure 3.2: Illustration of the singularity-avoiding properties of the symmetric (even) maximal slicing of the extended Schwarzschild solution in Kruskal-Szekeres coordinates. The bottom-most horizontal slice represents the initial data coinciding with $t = 0$ in the Schwarzschild time coordinate, appropriately extended to the other asymptotically flat region (to the left). The lapse tends to zero in the inner-most region, causing the slices to pile up at $r = 3M/2$ before hitting the singularity at $r = 0$, while at spatial infinity of either asymptotic region, the lapse reaches unity. The figure above is only schematic.

non-vanishing, time-dependent shift condition (a version of the so-called 'Gamma-driver') for times on the order of $3000M$ [5].

In the field today, then, one finds that many of these conditions are employed, with a great deal of success. Many, such as the '1+log' condition, are not as easily motivated as the maximal slicing condition, yet it has been shown, both through numerical experiment and analytic treatments, that they share many of the desirable properties of maximal slicing without the computational overhead of requiring the solution of elliptic equations. We turn then, from this discussion of ways of avoiding coordinate and physical singularities to methods for dealing with coordinate singularities which may not be avoidable (especially those that exist in an initial data hypersurface).

3.1.2 Isometry/throat conditions

Recall that, in the last section, we assumed that one could begin with an initial hypersurface that was free of singularities. The various slicing conditions we discussed would do their best to prevent an evolution from developing a new coordinate singularity, and in many cases, would also prevent it from hitting a physical singularity.

For the simple case of the isotropic Schwarzschild coordinates established in Eq. (3.1), however, the $t = 0$ initial time-slice already has a coordinate singularity at $\bar{r} = 0$. Again, nothing is "wrong" there; this is merely the single point to which spatial infinity of the other asymptotic space is mapped. Yet, as was mentioned in the introduction of this chapter, the computer will not discriminate between coordinate and physical singularities, and the code will crash.

For this reason, many of the early axisymmetric (two-dimensional) simulations of black holes [1, 9, 10, 17, 40, 69, 70] made sure that the coordinate-singularity containing region inside of the throat was not part of their computational domain. They did this by choosing the throats to be boundaries of their spacetime. For single black holes, regular spherical polar or cylindrical polar coordinates made this easy to accomplish. For the binary black hole problem, they typically utilized the specially-adapted Čadež coordinates [30, 68] in order to ensure that the throats would lie along constant-coordinate lines.

This, combined with their choice of initial data that is isometric across the throat(s), (either isotropic Schwarzschild or Misner data [54]), allowed them to derive very simple boundary conditions for the various geometrical quantities (lapse, shift, metric, extrinsic curvature, conformal factor, *etc.*) which usually amounted to requiring that a normal derivative of a metric function across the throat or the function itself vanish at the throat (*cf.*, for example, the explicit relations given in [17]). This conveniently removed the coordinate singularity and the other asymptotically flat space entirely from their calculations.

When Anninos *et al.* performed the first three-dimensional simulations of black holes in 1995 [11], they abandoned specialized coordinates in favor of simple Cartesian coordinates,

as it was felt that these would offer the freedom to explore a broader range of physically-interesting scenarios while being significantly easier to implement. They are not without drawbacks, however, and one such problem is that imposing boundary conditions is a much more complicated issue when the throat no longer sits along constant coordinate lines. (Interestingly, non-boundary-conforming grids are a difficulty that must be surmounted in three-dimensional singularity excision as well.) Like the axisymmetric simulations they had performed earlier, they chose the Schwarzschild solution in isotropic coordinates as their initial data. Here, though, the space inside the throat *would* be represented on their computational domain. In order to avoid the coordinate singularity in the conformal factor at $\bar{r} = 0$, they staggered the grid with respect to the origin, as is depicted in the lower panel of Fig. 3.1. This sounds, perhaps, like too easy a fix, yet it is quite effective, and, to this day, remains one of the prescriptions in the puncture evolution method described in the next section.

With this simple staggering, one has initial data that is regular across the entire computational grid, and Anninos *et al.* argue that this allows them to evolve the entire grid, both inside and outside the throat, without any special treatment whatsoever. Given that they use a uniformly-spaced grid everywhere, the region inside the throat (representing all of the other asymptotically-flat sheet) is extremely poorly resolved. Still, this is not a reason for concern as they were only interested in the exterior solution.

Anninos *et al.* did find, in the case of geodesic slicing, that they needed to explicitly enforce the isometry condition to prevent the code from crashing prematurely (*i.e.*, before πM) [11]. Again, because of the Cartesian grid, it is a non-trivial issue to construct accurate finite-differenced equations on the grid points near the throat that preserve the isometry. (See the discussion of finite-difference stencils and excision of spheres on Cartesian grids in §4.2 and Fig. 4.3.) They ultimately found that by mapping the field variables at points outside the throat to points inside, thus actively forcing the geometry inside to be isometric to that outside, they could avoid the mess of using different finite-difference stencils and logical switches such as Cook *et al.* used in their calculation of isometric multiple-black-hole initial data on a Cartesian grid [36]. Note that mapping external data onto data inside the throat will require interpolation (grid-points inside the throat will be mapped to locations outside the throat that, in general, will not lie on grid-points). Anninos *et al.*. used a "volume-weighted" interpolation scheme for this task, whereas Brügmann reports using a typical polynomial interpolation for this task [25].

The upshot of these early works in three-dimensional black-hole evolutions seems to be that, when dealing with data that has an isometry condition across the throat(s), the numerical relativist may choose to enforce or ignore this isometry during an evolution as he or she sees fit. When the isometry condition is employed, it allows the code to effectively avoid the coordinate-singularity-containing inner region, which may be desirable

for a specific application.

3.1.3 Puncture evolutions

Very much related to the issue of isometry conditions across throats, Brandt and Brügmann [22] introduced a novel technique for constructing multiple-black-hole initial data that was significantly simpler to implement than the popular conformal-imaging method. As initial data is not the focus of the current work, but rather singularity-handling in evolutions, we will only highlight the main features of the puncture method in its initial-data context, and concentrate on its role in evolutions.

If one seeks to construct initial data representing n black holes at a moment of time-symmetry, *i.e.*, $K_{ij} = 0$, under the further assumptions of spatial conformal flatness ($\gamma_{ij} = \psi^4 \delta_{ij}$) and asymptotic flatness ($\psi \to 1$ as $r \to \infty$), there is very little work to do. The momentum constraint, Eq. (2.2), is trivially satisfied, while conformal flatness implies that the Hamiltonian constraint, Eq. (2.1), reduces to the flat-space Laplace's equation acting on the conformal factor ψ, *i.e.* $\nabla^2 \psi = 0$. The regular flat-space Laplacian operator is linear, and so linear superpositions of solutions are also solutions. Using this fact, one finds the Brill-Lindquist-type solutions [24],

$$\psi = 1 + \sum_{a=1}^{n} \frac{M_a}{2|\vec{r} - \vec{r}_a|}, \tag{3.6}$$

where M_a is a mass parameter for the ath black hole and $\vec{r}_a$ is its position. For n=1, we recover the Schwarzschild solution in isotropic coordinates, Eq. (3.1). We can see that this solution has a coordinate singularity at the position of each black hole, r_a, which Brandt and Brügmann term "punctures." (For the regularity of the conformal factor ψ, the solutions exist on the domain $\mathbb{R}^3 - \{r_a\}$, and so one can say that domain is a "punctured" Euclidean space.)

The topology of this solution is quite interesting — for n holes, one finds $n+1$ distinct, asymptotically flat sheets, connected by n throats, which will not, in general, coincide with the apparent or event horizons of the individual holes as was the case in the isotropic Schwarzschild example. There is an inversion map through every throat, such that each puncture corresponds to spatial infinity of the asymptotic sheet connected by the throat, but the metric will not be isometric under these maps other than for the trivial example of $n = 1$, for which Schwarzschild in isotropic coordinates is recovered.

The major drawback with this data is that the feature which makes the solution so simple, namely, requiring a moment of time-symmetry, precludes the individual black holes from having any linear or angular momentum, a huge failing if the aim is to create astrophysically-relevant data. In 1980, Bowen and York [21] derived a non-trivial extrinsic curvature for a

point source with momentum parameter P_i and spin parameter S_i satisfying the momentum constraint, under the less restrictive condition that the initial time-slice be merely maximal, *i.e.*, $K = 0$, rather than completely time-symmetric. Keeping the assumptions of conformal flatness, and asymptotic flatness, the momentum constraint reduces to a flat-space *vector* Laplacian on a vector potential. Bowen and York explicitly provide a formula for the conformal extrinsic curvature, which we will denote as $\bar{K}_{ij}^{\mathrm{BY}}$ for a single source with the momentum and spin as free parameters. Again, by virtue of the linearity of the flat-space operator, linear superpositions of solutions are also solutions, and we can describe n point sources with arbitrary spin and linear momentum by simply adding the extrinsic curvature tensors associated with each source.

We will not restate the Bowen-York extrinsic curvature here; it suffices to know that the solutions are analytic and superposable. The difficulty, then, is that the Hamiltonian constraint takes on a more complicated form, *i.e.*,

$$\nabla^2\psi = -\tfrac{1}{8}\psi^{-7}\bar{K}_{ij}\bar{K}^{ij}. \tag{3.7}$$

For arbitrary numbers of holes with arbitrary linear and angular momenta, one only expects to find solutions for the conformal factor numerically. Then the issue becomes, "Over what domain and with what boundary conditions should the elliptic equation for ψ be solved?" In the context of the multi-sheet picture, it will be unfeasible to solve the Hamiltonian constraint if the domain is not reduced somehow from the $n + 1$ distinct Euclidean spaces joined at n throats as in the Brill-Lindquist setup.

In the conformal-imaging approach, one chooses a two-sheeted topology such that both sheets are isometric across *every* throat (*cf.* [35] and references therein and [36] for three distinct numerical implementations). For this to work, one must add more terms to the extrinsic curvature in an analogous fashion to the method of images in electrostatics. Because the two sheets are isometric to each other, one of them can be ignored altogether, and the domain of the elliptic solver can be restricted to $\mathbb{R}^3$ minus the volumes enclosed by the throats. The throats then become part of the "inner" boundaries of the domain, and the boundary condition is simply that the conformal factor be isometric across each throat.

The conformal-imaging approach is generally considered to be a very elegant approach to constructing initial data; its greatest drawbacks are that it formally requires the addition of an infinite number of image terms and that the numerical implementation of the elliptic solver is quite cumbersome because of the generally non-coordinate conforming boundaries in three spatial dimensions. Again, the work presented by Cook *et al.* in 1993 is a testament to the amount of complexity involved in providing a numerical implementation of this method [36].

Within this context, Brandt and Brügmann presented a very simple construction for

initial data, essentially combining the best aspects of the Brill-Lindquist and conformal-imaging data sets. Their solution was to assume that the extrinsic curvature is that given by the original, uncorrected sum of individual Bowen-York extrinsic curvatures, *i.e.*,

$$\bar{K}_{ij} = \sum_{a=1}^{n} \bar{K}^{\mathrm{BY}}_{ij(a)}. \tag{3.8}$$

In the conformal-imaging approach, this is also a starting point. In Brandt and Brügmann's construction, however, this completely specifies the extrinsic curvature — no corrections to it will be made.

To specify the domain over which the numerical elliptic solver for ψ will operate, they specify that the conformal factor ψ will take on a form very similar to the Brill-Lindquist solution,

$$\psi = \frac{1}{c} + u \quad \text{where} \quad \frac{1}{c} = \sum_{a=1}^{n} \frac{M_a}{2|\vec{r} - \vec{r}_a|}. \tag{3.9}$$

Recalling simple calculations in electrostatics, we see that

$$\nabla^2 \left(\frac{1}{c}\right) = -4\pi \sum_{a=1}^{n} M_a \delta^3(\vec{r} - \vec{r}_a), \tag{3.10}$$

which means that on the "punctured" Euclidean space, $\mathbb{R}^3 - \{r_a\}$, the Laplacian of that piece of the conformal factor vanishes, $\nabla^2(1/c) = 0$. This leads to a simplification in the Hamiltonian constraint,

$$\nabla^2 u = -\tfrac{1}{8} c^7 (1 + cu)^{-7} \bar{K}_{ij} \bar{K}^{ij}. \tag{3.11}$$

The final element of this prescription is to establish boundary conditions on u for the elliptic solver. The assumption of asymptotic flatness requires simply that $u \to 1$ as $r \to \infty$ with an appropriate $\mathcal{O}(1/r)$ falloff. The surprising result from their paper is that one need not specify a boundary condition on u at the punctures, r_a. It is sufficient to solve over the entire un-punctured $\mathbb{R}^3$, and one achieves a regular solution to the Hamiltonian constraint everywhere on the grid (albeit, with a loss in differentiability at the punctures, where the solutions are only C^2, while C^∞ everywhere else).

Let us pause here and see why this result is so useful. After the Bowen and York paper in 1980, initial data was no longer restricted to the astrophysically-uninteresting configurations defined at a moment of time-symmetry. They provided a fairly succinct prescription for the extrinsic curvature, satisfying the momentum constraint, leaving the initial-data creator with the task of of performing an elliptic solve for the conformal factor, which, due to

the complexity of the equation, must typically be calculated numerically. Without any additional assumptions, the domain of integration for the elliptic solver is unworkable — it requires the solver to be capable of performing integration over $n+1$ copies of $\mathbb{R}^3$ connected at n closed two-surfaces. The memory requirements alone are daunting enough to scare away most solvers.

What is then required is the imposition of a clever construction on top of the existing structure in order to drastically simplify the domain. The conformal-imaging method reduces the domain to $\mathbb{R}^3$ with n (topologically) spherical inner boundaries by imposition of isometries on the throats. The puncture method reduces the domain to vanilla $\mathbb{R}^3$ via a clever *ansatz* for the conformal factor. As we will see in the next section, Thornburg also developed a method which restricts the solver domain to that of the conformal-imaging method, but without the infinite sum of correction terms [73].

We come now to the role of the puncture method with regard to singularity-handling. While the method is a fantastic tool allowing the numerical relativist to essentially use off-the-shelf elliptic solvers to construct interesting initial data (by virtue of its simple domain), Brügmann presented a framework in which one could utilize the same ideas to treat the coordinate singularities at the punctures during an evolution [26]. Briefly, the premise is that one can use an analytically-known function to factor out the singular behavior of the geometrical variables (usually in the form of a conformal factor), and treat all differentiation as a combination of numerical (finite-difference) derivatives and analytically-evaluated derivatives.

This idea was not new to the field of numerical relativity — the first black hole simulation in three dimensions by Anninos *et al.* also used this technique, which they dubbed "conformal differentiation" [11], and Brügmann had also implemented this technique for his initial AMR simulations [25]. Focusing on the earlier of the two works, we recall that Anninos *et al.* evolved a single Schwarzschild black hole in isotropic coordinates (which, again, are spatially conformally flat), providing them with the foreknowledge that all of the singular behavior in the metric functions was contained within the conformal factor. Their implementation of conformal differencing proceeded as follows.

1. When constructing the initial data, they calculate the conformal factor ψ and its spatial derivatives analytically, and store them in memory up to machine precision.

2. During an evolution, whenever derivatives of the spatial metric are needed, they divide out the conformal factor, *e.g.* $\tilde{\gamma}_{ij} = \gamma_{ij}/\psi^4$ and then compute derivatives of the conformal functions $\tilde{\gamma}_{ij}$ via standard finite-difference equations.

3. The derivative for the physical metric variable is then calculated by retrieving the conformal factor and its derivatives from memory and combining this with the numerically-

computed derivative of the conformal functions. Thus,

$$\partial_k \gamma_{ij} = \underbrace{\partial_k(\psi^4)}_{\text{(a)}}\underbrace{\tilde{\gamma}_{ij}}_{\text{(n)}} + \underbrace{\psi^4}_{\text{(a)}}\underbrace{\partial_k \tilde{\gamma}_{ij}}_{\text{(n)}} \tag{3.12}$$

where '(n)' indicates a numerically-computed quantity and '(a)' denotes an analytically-known result retrieved from memory.

Anninos *et al.* remarked that they needed this implementation in order to have accurate and stable evolutions. They also comment that the "known" singular function that one factors out need not be a purely analytic function. One can use, for instance, the conformal factor given in the elliptic-solved initial data, calculated to arbitrarily high accuracy, with its derivatives calculated on the initial slice numerically.

Brügmann, on the other hand, argued in favor of keeping the singularity-absorbing conformal factor an analytically known function. Recall from the discussion of puncture initial data above that the conformal factor will take on the form $\psi = u + 1/c$, where u is the solution to the modified Hamiltonian constraint, and $1/c$ is essentially the Brill-Lindquist-type conformal factor (up to an additive constant). While derivatives of u could be computed numerically after the elliptic solve, the prescription given suggests that one instead use the Brill-Lindquist form, *i.e.*,

$$\psi_{\rm BL} = 1 + \sum_{a=1}^{n} \frac{M_a}{2|\vec{r} - \vec{r}_a|}, \tag{3.13}$$

along with its easily determined (analytic) derivatives, as the singularity-absorbing conformal factor, that is then used in the same fashion as described above for the work by Anninos *et al.*

To be perfectly clear, the prescription is to construct puncture initial data by solving the Hamiltonian constraint for the conformal factor using the *ansatz* $\psi = u + 1/c$, where u is determined by a numerical elliptic solver. Then, all of the physical variables are constructed from their conformally-given variables (the flat metric δ_{ij} and the Bowen-York extrinsic curvature $\bar{K}_{ij}^{\rm BY}$) and this conformal factor. During an evolution, one factors out a *different* conformal factor, namely the Brill-Lindquist-form $\psi_{\rm BL}$, so that its derivatives are given analytically. These differently-scaled conformal variables and the known $\psi_{\rm BL}$ are used then in tandem to compute spatial derivatives of the physical variables.

Alcubierre *et al.* extend these methods by constructing well-tailored choices of gauge conditions such that they can evolve puncture initial data (for a single black hole, a black hole plus Brill wave, and a head-on collision of an equal-mass black hole binary) until late times [5]. The gauge conditions used there were found to be successful in earlier simulations employing singularity excision [6], and so it is somewhat surprising that these conditions

worked so well in a simulation in which no portion of the computational domain is excised. They do introduce some nontrivial modifications to the conditions so as to enforce the desired behavior that there be no evolution at the punctures themselves (vanishing lapse and shift), making a direct comparison of the two sets of simulations difficult.

In summary, we have presented here the most commonly used alternative singularity-handling techniques in a fair amount of detail so that singularity excision may be understood within its proper context. Any technique that can extend the lifetime of simulations (while at the same time not introducing spurious content into the physics one is simulating) represents a positive contribution to the field. (Of course, it always must be demonstrated that a given method does not introduce spurious effects).

With these methods as a backdrop, in the next section, we will examine how the technique of singularity excision became a standard tool-of-the-trade in numerical relativity, and how the different implementations have varied in their emphasis and implementation over the years. We do this to lead up to modern implementations of singularity excision, which are generally quite simple in nature and will form a conceptual starting point for our dynamical excision algorithm in the next chapter.

3.2 Historical development of excision

Strangely, to begin our discussion of the technique which we find so useful in the context of black hole evolutions, we must turn again to the construction of initial data. As was mentioned in the previous section, with the introduction of the Bowen-York extrinsic curvature in 1980, interesting initial data sets were essentially in hand, but clever choices of the domain of numerical integration for the conformal factor were still needed. The conformal-imaging approach was developed during the 1980's, but in 1987, Thornburg introduced an alternative construction [73], which forms one of the first concrete implementations of what has come to be known as the apparent-horizon boundary condition.

Attributing the idea to a suggestion by Unruh in 1984 (*cf.* ref. in [73]), Thornburg suggested that one chose inner boundaries of the domain to be marginally trapped surfaces. (Recall our discussion of trapped surfaces and apparent horizons in §2.4.2.) This choice reduces the domain on which the elliptic solver needs to operate to a single $\mathbb{R}^3$ with a closed two-surface cut out for every black hole. (Or, for close separations, it is possible that only one two-surface would be necessary to enclose all "black holes." Note also that this domain is the same as that used in the conformal-imaging approach.) This is an interesting choice for a boundary condition because it does away with the issue of wormholes, throats and multiple asymptotic spaces. Also, in general relativity, we know that when an apparent horizon exists on a given slice, it is guaranteed to be coincident with, or contained within

an event horizon [45].

Thornburg argues that this construction is desirable on three fronts. The first is a conceptual argument: the conformal imaging method requires that the other asymptotically-flat sheet be isometric to "our" sheet. In [73], he asks, "[W]hy should the (modelled) [*sic*] interior of a black hole depend at all on the contents of the 'outside world'?" By choosing a marginally trapped surface to be the boundary condition, we lose all ability to comment on the interior of the black hole. Our knowledge of space ends on "our" sheet. Second, because the interior of the black hole can not have any causal effect on the exterior, choosing the boundary to be an apparent horizon means that one is not "wasting" computational power computing the evolution for a region that cannot impact us. The final point, and the one which we will find a strong motivation for this work, is that "censored" singularities are implicitly excluded from the numerical domain in this method. This frees the lapse from its usual requirement of collapsing to zero before a singularity is intersected, and so slice-stretching need not necessarily be an issue in evolutions of this data.

Thornburg offers some ideas for how this boundary condition might be used in a numerical evolution, but he only treats the initial data construction (with an interesting choice of a multiple coordinate-patch method for performing the elliptic solve). It would not be until 1992 that an implementation of an "apparent horizon boundary condition" would be realized in the context of an evolution, when Seidel and Suen [63] presented in a brief letter two novel elements of black hole evolutions that they felt were instrumental in working toward a "singularity-proof" evolution, namely, a "horizon-locking" coordinate system, and "causal differencing." (The details of this framework are explained more completely in a follow-up paper with several additional authors in 1995, [8].)

This work does represent one of the first *working* numerical realizations of what we would now call singularity excision. They introduced the horizon-locking coordinates (HLC) and causal differencing (CD) techniques largely for the sake of facilitating the implementation of the excision algorithm. For our purposes, it suffices to say that their HLC conditions offer a variety of constructions for determining a non-vanishing shift such that the apparent horizon stays fixed in terms of its coordinate location and thus the grid-points are not allowed to fall into the black hole. (One essentially gets this "for free" with the Kerr-Schild slicing, *cf.* App. B.) We will comment more on the use of causal differencing below. In both works mentioned above [8, 63], they simulate a single Schwarzschild black hole in isotropic coordinates, using a spherically-symmetric (one-dimensional) code.

Their method can be summarized as follows. For approximately $1M$, they evolve the initial data with the shift set to zero and the lapse determined by maximal slicing. At approximately $t = 1M$, they begin to phase in their apparent-horizon locking shift, such that by $t = 3M$, the shift is entirely determined by the HLC conditions (which are algebraic or purely spatial differential relations). During this initial evolution governed by maximal

slicing, the lapse will begin to collapse to zero at the inner-most grid points, and as the coordinates "fall in" to the black hole, the location of the apparent horizon will grow to larger and larger values of the radial coordinate. Once the HLC condition is fully active, they freeze the evolution of the lapse for the remainder of the evolution, thus locking in the lapse profile created by evolving for $3M$ with maximal slicing. The lapse will have begun its collapse at the inner-most points, but they found that, even after $3M$, the smallest value of the lapse on the grid was 0.3. At this point they also implement excision by ignoring the evolution for the inner-most grid points, maintaining on the order of ten grid-points inside the horizon as a "buffer."

Let us pause here and consider this last point, the introduction of singularity excision in practice. Many of the issues faced in this early work will be encountered in other implementations of excision as well. The arguments that suggest that one should be able to remove or excise a region of the spacetime from numerical computations via causality arguments are common to all singularity excision techniques, and so we will discuss them here:

1. The event horizon (EH) of a black hole represents a one-way membrane through which timelike or null geodesics can enter but never leave.

2. The numerically problematic physical (curvature) singularity will lie hidden within an event horizon.

3. The event horizon can only be determined at the end of a simulation (assuming the simulations are sufficiently "calm" at their end) by finding the boundary between those null geodesics that reach future null infinity and those that do not. Without the entire future history of spacetime, the event horizon cannot be located. In this sense we say that the event horizon is a *global* concept.

4. A *local* (in time) measure of this causal surface is represented by the apparent horizon (AH). Again, the apparent horizon is the outermost outer marginally trapped surface as explained in §2.4.2. The AH may or may not exist on a given spatial slice. If it does exist, it will lie inside the EH, and as the spacetime approaches stationarity, the AH and the EH will become arbitrarily close. Thus we can use the AH, if it exists, as an indicator, distinguishing between points than can send causal signals to future null infinity and those that cannot.

5. If it is numerically convenient, then, we can remove all or some of the grid-points within the AH from our numerical calculations because they will never be able to transmit *physical information* that can affect the exterior spacetime.

6. Information that does not propagate causally (such as, for instance, gauge modes in some formulations of GR) can and will escape from the apparent horizon to interact with the exterior computational domain.

This last point is one which deserves special attention, which we will postpone until our discussion of the dynamic singularity excision algorithm in §4.1.2. For now, the itemized list of issues above encapsulate the case one typically makes in favor of employing singularity excision. We will now return then to the discussion of the work presented by Seidel and Suen in [63].

By choosing to ignore, or throw away, the inner points of their grid, Seidel and Suen have created an inner boundary to their computational domain where one did not exist before. Two crucial issues then, are the location of this boundary and the nature of the boundary condition applied to the field variables nearby.

Regarding this first issue, almost all implementations of excision in numerical relativity choose their inner boundary to remain *within* the apparent horizon, with a number of "buffer zones" between the horizon and the boundary, as Seidel and Suen did. The issue of the existence and size of this buffer region of grid-points that are still considered part of the evolution and yet are within the apparent horizon is one that has been debated quite often (*cf.* discussion in [74]). Basically, one wishes to find a fine balance between keeping the inner boundary of the grid close enough to the horizon such that no points on the grid run the risk of intersecting the singularity and yet far enough inside the horizon that any possible errors created near the inner boundary are unlikely to be able to escape and ruin the evolution outside the horizon. In general, prescribing how large the buffer region should be becomes an issue to be determined empirically by numerical experimentation.

Regarding the issue of the boundary condition imposed there, Seidel and Suen argue that the introduction of their causal differencing scheme allows them to use a "boundary condition without a boundary condition." Their shift condition used to lock the coordinate location of the apparent horizon has the effect that, inside the horizon, the future light cone is tilted inward toward smaller and smaller values of the radial coordinate. (Note that local future-pointing light-cones will always tip in the opposite direction as the shift-vector, as viewed in the constant-coordinate (grid) frame.) For every point on their grid inside the AH, data will depend only on past data from grid points at equal or larger radial coordinate values — their causal finite-difference stencils have effectively become entirely one-sided. In other words, they impose *no* condition at all on the inner boundary by virtue of their finite-differencing method. We will discuss how causal differencing accomplishes this below.

With this method, Seidel and Suen [63] and Anninos *et al.* [8] demonstrate single-black hole evolutions lasting approximately $150M$, with considerably better behavior at late times than simulations implemented with maximal slicing and zero shift. (In both papers, they compare their results to the work by Bernstein, Hobill, and Smarr [18], considered at the time to be the state-of-the-art in one-dimensional maximal slicing simulations.) They also show results of a simulation lasting up to $1000M$ with a coarser resolution than used in their other results and a constant, but seemingly linear, drift in the apparent horizon mass

and Hamiltonian constraint. They conclude that progress in numerical relativity over the previous thirty years had been hindered by the difficulties involved in avoiding spacetime singularities (*i.e.*, the use of maximal slicing) and that their apparent-horizon boundary condition effectively circumvented these difficulties.

Even though the simulations were performed only in a one-dimensional code, the improvements over using singularity-avoiding slicings were so dramatic that, in many ways, this work would form the basis of excision for almost the next eight years. In the later works, there are few appeals to the need for a horizon-locking coordinate scheme, but what is generally known as causal differencing persisted as a commonly used technique until the work of Alcubierre and Brügmann in 2001 [4]. In addition to the presentation in [8, 63], one can find treatments of causal differencing in [34, 51, 59], or "causal reconnection" in [7].

Following the presentation in Scheel *et al.* [59], we briefly describe the motivation behind causal differencing. Essentially, causal differencing attempts to reconcile the local physical and numerical (grid) light cones such that the finite-difference stencils never require information to be transmitted acausally. Referring to the discussion of the 3+1 decomposition in App. A, note that we choose to evolve along the integral curves of the vector field t^a, which has a component orthogonal to the time-slices and a component tangential to them given by $t^a = \alpha n^a + \beta^a$ (*cf.* §A.8). We require that our coordinate basis be Lie-dragged along t^a which means that a grid-point labeled by values i, j, k will have the same coordinate basis (and hence the same coordinate value) on one time-slice as it does on the next. In the frame of the computational grid, we have the (relatively boring) spacetime picture depicted in Fig. 3.3a. In other words, our grid-points sit at constant coordinate values.

The reason for concern is that a curve through spacetime made by a point sitting at constant spatial coordinates is not required by any condition we have imposed to be a timelike. We can see this easily by noting that the norm of our time-evolution vector field t^a is given by

$$|\boldsymbol{t}| = g_{ab} t^a t^b = -\alpha^2 + \beta^a \beta_a. \tag{3.14}$$

Obviously, for large enough values of the shift, t^a can become null or even spacelike, as is illustrated in Fig. 3.3b. Again, the shift just represents a coordinate freedom, and thus we cannot possibly "break" the physics in any way by imposing an arbitrarily large shift. The problem is that the finite-difference stencils have their own intrinsic notion of causality. An explicit forward-in-time, centered-in-space (FTCS) time-update stencil, for instance has the numerical light-cone shaded in Fig. 3.3a, meaning that the value on the $(n+1)$-th time-slice depends upon those values on the n-th level falling within its shaded area. With a large enough shift, numerical and physical causality will no longer coincide.

In order to reconcile this, various implementations of causal differencing have been pro-

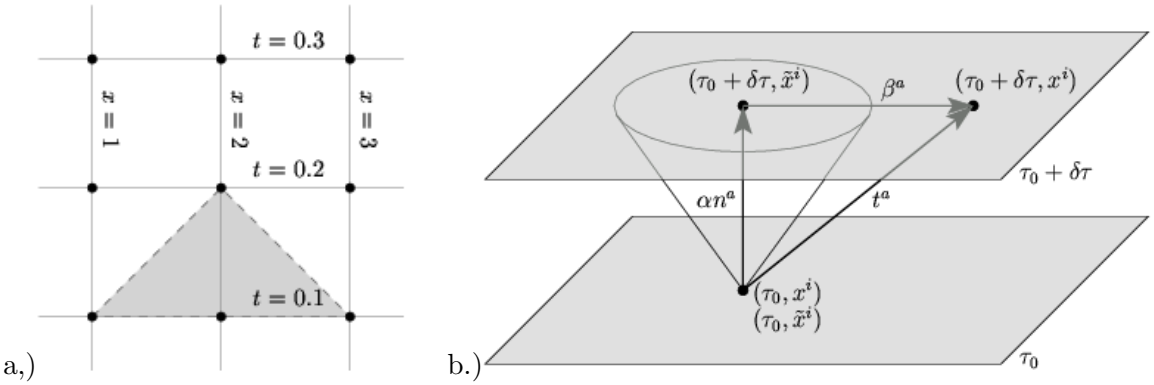


Figure 3.3: Spacetime diagrams illustrating the use of causal differencing. a.) The spacetime as seen by the computational grid. From time-slice to time-slice, any given grid-point will always have the same spatial coordinate. A typical forward-in-time, centered-in-space finite-difference stencil's domain of dependence is depicted to illustrate that stencil's numerical past light-cone. b.) The spacetime in terms of geometric quantities. For large shifts, the time-evolution vector field t^a can be null or spacelike. The spatial coordinates x^i are Lie-dragged along t^a, so the point with coordinates (τ, x^i) is taken to the point $(\tau + \delta\tau, x^i)$. One can also define another set of coordinates $\tilde{x}^i$ that coincide with x^i on an initial slice, but are Lie-dragged along αn^a rather than t^a.

posed, all essentially suggesting the same idea: rather than evolve spatial tensors along t^a from slice to slice, one should evolve them along the the hypersurface-orthogonal vector $\tilde{t}^a = t^a - \beta^a$ $(= \alpha n^a)$, and then once all quantities are available in the new slice, interpolate to the true location of the grid-points. As Scheel *et al.* point out, in spherical symmetry (1+1-dimensional simulations), this procedure is not untenable, but for the generic 3+1 simulations, causal differencing will require a significant investment in computational infrastructure and an increase in computational time to accommodate the added evolution variables and interpolations. Singularity excision had developed a reputation for being difficult to implement, but this probably was due to the use of causal differencing than excision itself. (What could be simpler than choosing to ignore calculations at some points?)

Nonetheless, excision with causal differencing was quite successful in 1D [8, 59, 63], and had a fair amount of success in 3D, being featured in the Alliance [34] simulation of a single, boosted Kerr-Schild hole, the null-formulation simulations of "wobbling" black holes [44], and the Pitt-Penn-Texas grazing collision [23]. By the end of the last decade, singularity excision was beginning to become an accepted (and according to some authors, necessary) method for singularity-handling in black hole spacetimes.

Of these simulations, none but the null code could demonstrate evolutions to late times ($\sim 1400M$, in that case), and it was not clear what features of the numerical implementations or coordinate choices were to blame. In hindsight, we can look back at these simulations and suggest that the difficulties they encountered stemmed from their use of the ADM

formulation, their use of a purely algebraic gauge, use of a fixed (Dirichlet) outer boundary condition, and/or any number of other issues. Still, the production of generically stable and accurate evolutions in 3+1-dimensional numerical relativity remains an open problem today; we should not be haughty and scoff at any of these early setbacks or limited-duration simulations.

Moving on, and entering the "modern" era of numerical relativity at the turn of the millennium, the field went through quite a change. Perhaps because of the difficulties the Grand Challenge Alliance had encountered in producing binary black hole collision simulations in three dimensions, or perhaps because of the large number of postdoctoral scholars working in the field who found themselves in a job market in which few universities were hiring, the field changed its focus, but not its overall aim.

With the binary black-hole problem looming overhead, many groups set themselves to understanding why single black-hole simulations were so difficult to evolve stably in three dimensions. This time saw the the ADM formulation being phased out in favor of explicitly hyperbolic formulations or the BSSN system. The overwhelming success of the personal computer industry ushered in the era of the Beowulf cluster, shifting the focus somewhat away from supercomputers and onto trying to use more limited resources to discover the causes of instabilities and how to treat them. And, as we will discuss below, it saw the introduction of a significantly simpler way of implementing singularity excision.

3.3 "Simple" Excision

In 2001, Alcubierre and Brügmann [4] focused on one aspect of three-dimensional numerical simulations that they thought were perhaps overly complicated, namely, singularity excision, and demonstrated how a very simple implementation led to stable evolutions of a single Schwarzschild black hole in octant symmetry.

They expressed the sentiment that causal differencing and spherical excision regions on Cartesian grids were overly complicated and that their introduction into codes muddied the waters when one had to eventually discover what ingredient was causing a code to go unstable. They decided to simplify the algorithm considerably by introducing the following simplifications:

1. They excised a shape adapted to Cartesian coordinates, namely, a cube.

2. They suggested using an extremely simple boundary condition rather than leaving the inner boundary entirely unfixed (in the "Boundary Condition without a Boundary Condition" ideal of Seidel and Suen).

3. They did not implement causal differencing. Rather, they used standard centered differencing for all spatial derivatives of field variables except advective terms ($\beta^i \partial_i$),

and for those, they used upwind differencing.

Their choice of boundary condition for their static evolution was to copy the time derivative of every field onto their excision cube. They refer to this implementation as "Simple Excision".

With the ingredients described above, and the change in focus to implementing the *simplest* conditions they could, Alcubierre and Brügmann were able to demonstrate a three-dimensional simulation of a single Schwarzschild black hole in IEF coordinates that evolved until the time derivatives of the field variables had reached machine-precision (on the order of 10^{-16}). We will discuss some of these design choices in more detail in the next chapter when we describe our implementation of dynamic singularity excision.

Yo *et al.* [78] took these simplifying assumptions and implemented them in an evolution of a scalar field on a fixed Kerr-Schild background. They argued that the simple derivative-copying employed by Alcubierre and Brügmann was fine for static cases, but was inadequate to handle boosted black holes. They introduced a modified method of determining along which direction one does the copying or extrapolation of derivatives, and they increased the order of the extrapolation scheme to fourth order, as well as returning to a spherical excision region. They used the scalar field on a moving (but not numerically evolved) black hole background as a testbed for their excision techniques and method of handling the newly emerging points as the boosted hole moves across the domain.

The treatment in these works is closely related to our implementation, and so we will not discuss their implementations individually, but will rather focus our attention in the next chapter on our treatment of excision, as it builds upon both of these works, extending them to not just static or fixed black-hole spacetimes, but also dynamic ones, in which we introduce the freedom to allow the black hole to move across the computational domain.

Chapter 4

Dynamic singularity excision

Having highlighted some of the various theoretical considerations involved with working in numerical relativity and discussed in detail the various alternative singularity-handling techniques employed over the years, we are now in a position to discuss the algorithm which has been the focus of the author's research over the past few years.

4.1 Overview

While most of the simulations we will describe in this work involve single black holes, we are always attempting to work toward a code that can evolve arbitrary black-hole initial data. In the case of the binary-black-hole in-spiral problem, it is expected that, at large enough separation, one can construct an approximately co-rotating frame such that the black holes sit at approximately constant coordinate values (*cf.* [27] for a recent application). As the system radiates gravitational energy, and the black holes drift toward each other, these coordinates would have to become increasingly distorted to keep the black holes at their original coordinate positions.

And while it is certainly possible to use the coordinate freedom to hold the holes in place, we feel that it is, in some ways, more intuitive to allow the holes to move across the grid (occupying different coordinate values in time), and to construct an excision algorithm that allows for this. Again, some combination of this and the method above wherein co-rotating coordinates for a binary are constructed with the deviations from perfectly circular orbits corrected by motions on the coordinate grid are likely to be extremely useful. With this in mind, we have invested our effort in constructing a dynamic singularity excision algorithm and various testbeds and practical working examples that can handle both the static scenarios already explored as well as interesting new dynamic scenarios.

4.1.1 Goals of the method

Within the context of the developments in excision techniques over the past two decades, the goals of our work are the following:

- Keep the algorithm as simple as possible. Numerical relativity is already riddled with

enough complexity by itself; we do not wish to introduce new errors whose origins we cannot determine.

- Allow greater flexibility. The method should present the numerical relativist with the ability to tune the excision to the problem he/she is tackling.

- Preserve inherent symmetry. When the problem being solved has an inherent symmetry (e.g. spherical, axisymmetric, equatorial, etc.) the excision technique should do its best to preserve this.

- Handle newly emerging points. As the holes move across the domain, points which had previously been excised will emerge from the excision region and a method will be needed to provide these points with correct data. (We call this step 'populating' the grid-points.)

In accordance with these goals, we have developed an excision algorithm along the lines of [4] and [78], that is easy to understand and implement, and works for static and dynamic spacetimes. While, in many ways, we have followed the logical progression from earlier works, we present a few novel elements which seem to aid in increasing the duration of runs.

It should be noted that the way in which we have implemented the dynamic singularity excision algorithm is inspired by and perhaps even coupled to the finite-difference approximations we use to calculate spatial derivatives. The fact that our simulations are three-dimensional in nature and are performed on a Cartesian grid prompted us to focus attention on an implementation that was well-suited for that environment. Other methods for estimating spatial derivatives, such as the pseudo-spectral methods or finite-element/volume methods, are much better at adapting to irregular (*i.e.*, spherical) boundaries. By virtue of this, it is almost trivial to implement excision within those frameworks — one can simply adapt the boundaries of one's computational domain such that a small region near the singularity of a black hole but within the apparent horizon is not present. Or, one can choose the boundary to be the apparent horizon itself. It is only in the finite-difference approximation, and even then, typically only in the case of Cartesian grids, that one need develop techniques such as these.

That having been said, it is likely that finite-difference codes are likely to remain quite popular in three-dimensional numerical relativity, especially now that readily-available fixed-mesh-refinement implementations are being introduced [62].

4.1.2 Causality considerations

In this method, we will continue to insist (as other implementations of excision have done before) that the single- or multiple-black hole spacetime being evolved numerically be repre-

sented in horizon-penetrating coordinates by the time excision is applied. Again, in choosing the location of the excision boundary, we are always trying to strike a balance between being well-enough inside the event/apparent horizon such that we are confident in the causal "isolation" of our inner boundary with respect to the exterior solution and being far enough away from the singularity so that gradients of the field variables will not be too severe for the finite-differencing routines to handle. Hence we will continue the time-honored tradition of keeping a buffer region of a few grid-points between the apparent horizon and the excision boundary.

With the inner boundary of the computational domain then located sufficiently inside the apparent horizon, we assume, as Seidel and Suen did [63], that all of the *physical* signals will travel within or along light-cones which are tilted with respect to the coordinate grid toward the singularity such that no boundary condition need be applied at these points.[1]

As we alluded in §3.2, a feature of general relativity is that the gauge freedom potentially allows gauge modes (which can carry no physical information) to propagate acausally. These modes would then not necessarily be subject to the considerations above regarding our ability to proceed without boundary conditions on the excision boundary, and they would have the potential to penetrate the apparent horizon and enter the exterior spacetime. Gauge modes are a sticky subject because, on the one hand, we can argue that they carry no physically meaningful information and thus their behavior can have no impact on the observable content of an evolution, while on the other hand, these gauge modes can dramatically affect the stability of the numerical simulations.

In this context, the utility of explicitly hyperbolic formulations of general relativity becomes apparent (*cf.* review by Reula [57]). With a manifestly hyperbolic formulation, one can determine the characteristic variables and determine their propagation speeds, thus allowing the identification of those fields requiring conditions at the excision boundary. Some hyperbolic formulations have the exceptionally nice behavior that *all* modes, physical and gauge modes alike, propagate along the light-cone or normal to the spatial hypersurfaces (*cf.* implementation by Scheel *et al.* [59] and references therein). This would have the feature that the problem would be formally well-posed with no boundary condition applied to any of the field variables at the inner boundary.

In this work, however, we perform calculations exclusively in the BSSN system, which, for arbitrary gauges and background spacetimes, has not been demonstrated to be explicitly

[1] In spherical symmetry (as Seidel and Suen studied), it is easy to guarantee that, on the excision boundary, the future-directed light cones are all contained entirely within the excision region, *i.e.*, the boundary condition is purely outflow there. With the cubical excision regions we will often use, a very tight restriction is imposed if one requires that the boundary be purely outflow. Namely, the cube must have edges smaller than $4M/(3\sqrt{3})$ for simulations of a Schwarzschild black hole in Kerr-Schild (IEF) coordinates [31, 32]. We will frequently violate this restriction, as we have seen no evidence in our numerical experiments that having cubical excision regions larger than this size leads to unstable evolutions or lack of convergence for these simulations.

hyperbolic. (The hyperbolicity of BSSN has been established for fixed densitized lapse and fixed shift in the linear case [58].) We will not explicitly demonstrate that all modes are outgoing (as we cannot identify characteristic variables), but will brazenly proceed, assuming that we need not apply a boundary condition. Evolutions of single black-hole spacetimes for essentially infinite run-times with this excision algorithm or slight variations thereof offer some empirical evidence [4, 66, 71, 79] that this assumption is not as cavalier as it may sound. While this assumption may formally present an ill-posed problem, we will argue, as other authors have [76], that this does not prevent us from learning much about the solution and its numerical properties.

In summary, we work under the assumption that there is no need for boundary conditions at the excision boundary. The only task is to design a discretization of the evolution equations that is appropriate at the excision boundary where centered finite differencing is no longer feasible. With these considerations in mind, we will now describe our implementation in detail.

4.2 Details of the algorithm

Roughly speaking, our dynamic singularity excision algorithm proceeds in three steps for each step forward in time. First, given the location and size of a black hole on the coordinate grid, all grid-points are labeled (via this author's `GridPointLabel` routines) to indicate whether they are excised or not. Once the labeling is complete, information about the boundary of the excision region is recorded so that data from the computational interior can later be extrapolated onto it. Second, when it comes time to provide data on the excision region boundary, simple one-dimensional polynomial extrapolation is employed to do so, either extrapolating the solution field variables or their time-derivatives (right-hand-sides of the evolution equations). The third step is to check (via routines external to the excision algorithm) whether the excision region should be moved before the next time step. If it should be moved, we must alter the labels of the grid-points, redefine the boundary meta-data, and provide data for those points which had been previously excised, but now are uncovered. We will discuss each of these steps in detail.

4.2.1 Establishing the excision mask

Because the regions of spacetime outside the black hole(s) will typically occupy much more volume on the computational grid than those inside, we do not go to special lengths to construct a computational grid that does not contain our excised points. Rather, we construct a simple "masking" function, as is common practice in these discussions, that indicates whether a certain point on our grid is to be treated as part of the numerical evolution, or whether it is excised from these calculations. This treatment is illustrated in Fig. 4.1.

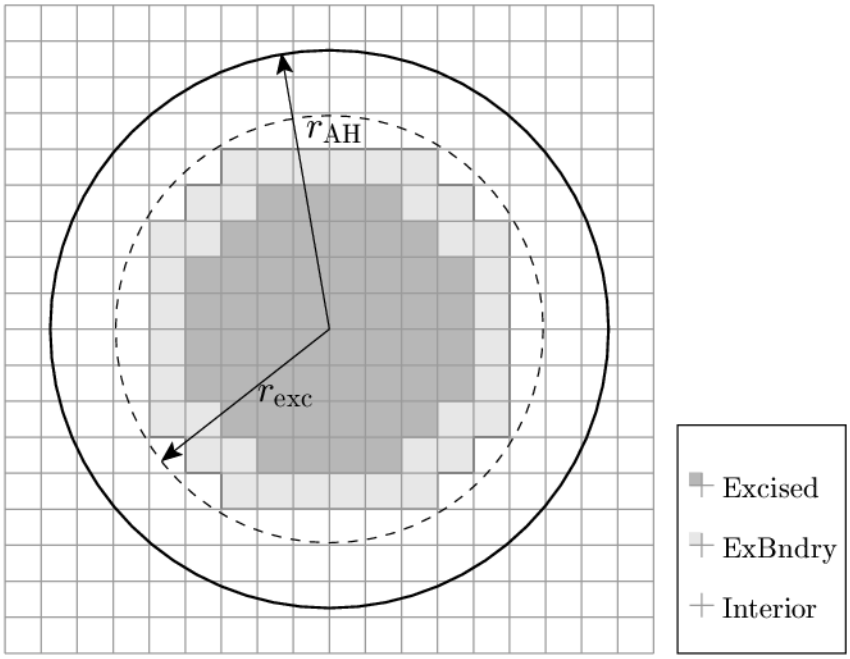


Figure 4.1: Schematic of the mask function which labels whether a grid-point residing on the coordinate grid is to be considered in computations, represented here in two dimensions for clarity. The apparent horizon is denoted by the circle (sphere) with radius $r_{\rm AH}$. The size of our excision region is parameterized by the circle (sphere) with radius $r_{\rm exc}$ such that any grid-points within this sphere are marked as 'excised,' here shaded dark grey. Those excised points which have non-excised nearest-neighbors are relabeled 'excision boundary,' here shaded light grey. Grid-points lie at the nodes (intersections) of grid lines, and our shading indicates that the grid-points *at* and within the boundary of a shaded region are labeled with the same mask value.

Our mask function, which lives at the same grid-points as the field variables, is initialized to indicate that all points on the grid are available for computations. We will give these points the mask label '`Interior`,' meaning that they are part of the computational interior. Any points which we have excised, we will label '`Excised`.' Finally, the excised points which have `Interior` nearest neighbors will be relabeled '`ExBndry`,' to indicate that they lie on the boundary of the excision region. As described below, we will treat the `ExBndry` points specially.

We parameterize our choice for the size of the excised region by the radius $r_{\rm exc}$ of a sphere in which the excised region can be inscribed, rather, than say, the number of "buffer" grid-points between the horizon and the excised region. Our choice is as good a parameterization as any, but it does have the advantage of ensuring that, regardless of what shape we choose for the excision region, we are guaranteed that no excised grid-point can be farther away from the center of the region than $r_{\rm exc}$.

A brief comment is warranted regarding the choice of the shape of the excision region.

Historically, three-dimensional simulations implementing excision used a spherical excision region [23, 34]. When the spatial coordinates used in a simulation are relatively distortion-free, the apparent horizon is often spherical or ellipsoidal. With this in mind, choosing the excision region to be approximately spherical represents the best available compromise between guaranteeing that the excision region is everywhere within the apparent horizon and yet far enough away from the physical singularity such that the large gradients of field variables near the central singularity do not become unmanageable. So, while a spherical region nicely encapsulates the behavior we would like, it is approximated quite poorly on Cartesian grids at the spatial resolutions commonly used in three-dimensional simulations. Such a construction is known within the community colloquially as a "LEGO" sphere.

With this in mind, we have explored alternatives to the sphere as candidates for our excision-region shape. As mentioned in our list of motivating factors in designing this algorithm, we sought to preserve any inherent symmetry whenever possible. In three-dimensional simulations on Cartesian grids, we often impose reflection or rotation symmetry conditions across some axes in order to decrease the memory requirements of a given simulation (when the underlying symmetry of the problem allows it). Thus, we typically refer to evolving an octant, a quadrant, or a "bitant" of the full three-dimensional space. In order to preserve, at the very least, these simple symmetries, we have focused our attention on implementing excision shapes that possess at least an octahedral symmetry. Obviously, a sphere possesses this symmetry, but we have also implemented two others which are particularly easy to use, namely the cube and the cuboctahedron. (Note that [4] implemented a cubical excision region, while [78] decided to return to spherical excision regions.) In Fig. 4.2, we illustrate these simple shapes, and typical representations of them on a Cartesian grid. In many simulations, the evolutions are independent of the excision shape used, and thus we predominantly focus our attention on the cube and the sphere.

4.2.2 Treating the boundary of the excision region

As with the technique introduced in [4], we adopt standard centered-differencing stencils on all spatial derivatives (except for the advective terms taking the form of $\beta^i \partial_i$, for which we use "upwind", *i.e.*, one-sided, derivatives). Immediately next to the excised region, the centered differencing stencils will have insufficient information to calculate their derivatives, as depicted in Fig. 4.3. Note that the advective derivatives do not pose a difficulty because they always take values from points in the same coordinate directions as the shift vector. In the black-hole spacetimes we typically study (*cf.* App. B), the shift vector always points radially outward from the central singularity, preventing grid-points from falling in. This seems to be a common feature of shift conditions for horizon-penetrating coordinate systems that seek to avoid slice-stretching [5].

One possible approach to handle spatial derivatives at the excision boundary is to mod-

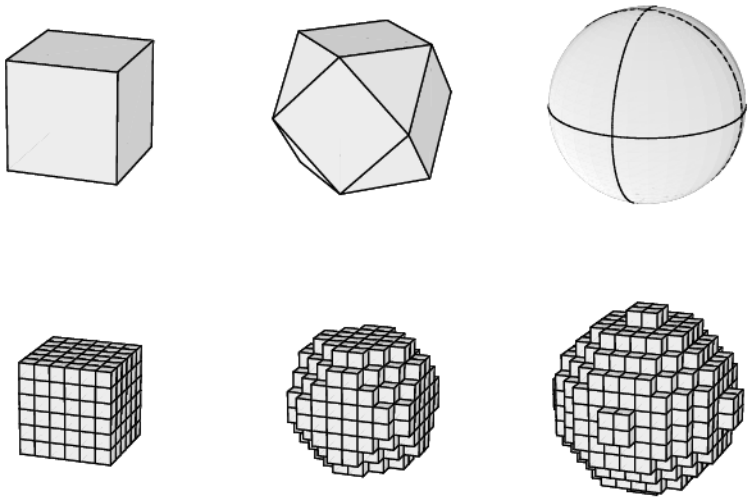


Figure 4.2: The continuum excision shapes one may wish to use for excision — a cube, a cuboctahedron, and a sphere, along with their representations on a finite-resolution Cartesian grid. All of the shapes are normalized such that they are inscribed within the continuum unit sphere at the top-right. The commonly-seen "LEGO" sphere at the bottom right is the largest of the three excision shapes in terms of the volume it encloses.

ify all finite-difference stencils there so as to avoid using grid-points from within the excised region, while keeping the same order of accuracy. (Note that, in the spherically symmetric simulations of [63], the causal-differencing method selected such one-sided stencils automatically.) In three-dimensional simulations, however, selecting one-sided differencing stencils requires an unpleasantly large number of conditional checks. For each type of boundary interface encountered (faces, edges, corners, *etc.*), the code must select an appropriate stencil, which might not be uniquely determined. Several common second-order accurate finite-difference stencils that must change from centered to one-sided are depicted in Fig. 4.3.

To circumvent the complications of this stencil-modification method, we have opted to leave all centered-difference stencils as they are. To do this, we must somehow provide valid data on the excision boundary (lightly grey-shaded grid-points in Figs. 4.1 and 4.3). Our method for providing this data will be to extrapolate the necessary quantities onto the boundary of the excision region based on the data available at the nearby `Interior` grid-points. Extrapolation onto a boundary followed by a centered-difference calculation using that extrapolated data is effectively equivalent to performing a one-sided difference calculation. With proper balancing of the order of the truncation error of the extrapolation and the truncation error of the centered-difference, one can perform this calculation without

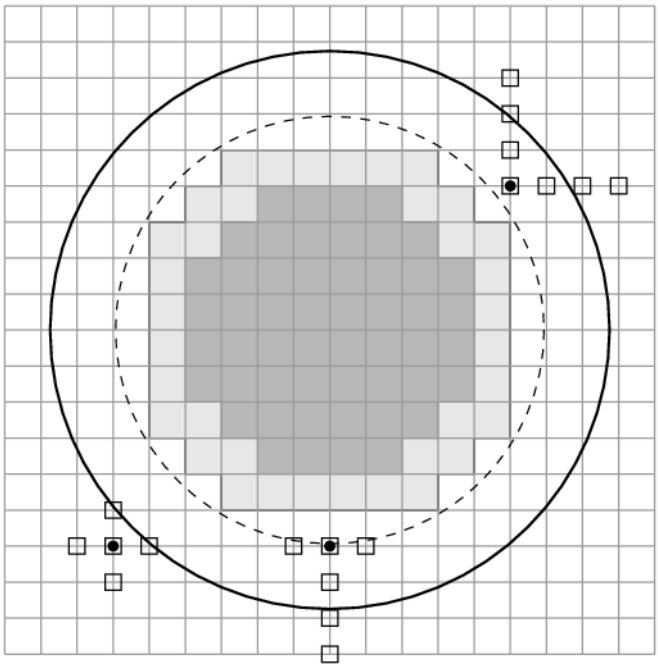


Figure 4.3: Schematic of the mask function as in Fig. 4.1, with several second-order accurate finite-difference stencils overlaid. A solid point denotes a location at which a spatial derivative calculation is desired, and the boxes denote the points that are involved in calculating that derivative (the stencil). Values for the field variables are unknown at the `Excised` and `ExBndry` points, so at each point near the excision boundary, the stencils must be modified to become one-sided and use only available data.

a loss in accuracy (although the coefficient of the error term will not be the same as if a standard one-sided difference stencil had been used).

Let us restate our method simply. We establish our excision mask by choosing a location on the coordinate grid to represent the center, a radius, and one of the three shapes in Fig. 4.2 such that the excision region is suitably within the apparent horizon (assuming for now we know where that is). All points fitting within the chosen excision shape (which is scaled to fit within a sphere with radius $r_{\rm exc}$) are labeled `Excised`, and these points will not take part in any calculation. For the sake of visualization, we typically set the values of all quantities there to zero. Those `Excised` points which have `Interior` nearest-neighbors are re-labeled `ExBndry`. These points will not have any spatial derivatives computed *at* them, but values at these points will be used in the finite-difference calculations for spatial derivatives at nearby `Interior` points. To continue to provide valid data at the `ExBndry` points so that this last step may proceed, we will extrapolate from the quantities computed at the `Interior` points during or after every time-step. Again, the motivation for this is that we want to avoid having to alter our finite-difference stencils to accommodate the irregular nature of the excision boundary, a step which has always complicated excision algorithms in the past. By comparison, extrapolating the data and using centered-differences across the remainder of the computational grid is fantastically simple. This, in essence, is the extent of the algorithm. We will now discuss the specifics of the extrapolation, the update scheme, and how we allow for the excision region to move through the coordinate grid.

In order to keep the algorithm as simple as possible, we perform only one-dimensional polynomial extrapolation onto the excision boundary. This allows us to update the values of the field variables on the `ExBndry` points quite rapidly. The issue then becomes a simple question of along which direction we choose to extrapolate the data. There is a fair amount of flexibility in this prescription, and different authors have selected different methods of choosing their extrapolation directions.

In this work, we are motivated by the continuum excision shapes of Fig. 4.2 to extrapolate along the normals to the surfaces of those shapes, which we will denote as s^i (*e.g.*, for the sphere, we have $s^i = x^i/r$). In general these continuum normals will not coincide with our coordinate grid. To keep the algorithm fast, we will only allow extrapolation along "lattice vectors," *i.e.*, those vectors that only point from one grid-point to another grid-point (borrowing terminology from solid-state physics). If we did not restrict ourselves to this choice, we would need to interpolate. We will select whichever lattice (or grid) vector k^i that is closest (in a Euclidean dot-product sense of closest) to the normal to our continuum surface s^i.

At each point on the excision boundary, we record its location on the grid (its grid labels $(i, j, k) \in \mathbb{Z}$), and the grid locations of its four closest neighbors along the lattice vector k^i, selected as described above, and store this for future use (in the form of an array of data

structures containing this information). When we need to extrapolate data onto the excision boundary, we can quickly loop over all the points in this array, and access the meta-data stored within it to tell us the locations of the points we will use in the extrapolation.

This is a particularly nice way of treating the excision boundary update procedure because, in general evolutions, we will have a large number of field variables that will all need to be extrapolated onto the excision boundary. By storing all the information about which points are involved in the extrapolation process, we can quickly and efficiently perform the same extrapolation procedure on different field variables.

Using one-dimensional polynomial extrapolation along our lattice vector k^i, then, the value of a field variable f extrapolated to a point $\mathbf{x}$ on our boundary will be given by

$$f(\mathbf{x}) = \sum_{j=1}^{N} a_j \; f(\mathbf{x} + j\mathbf{k}), \tag{4.1}$$

where the a_j are the extrapolation weighting coefficients, given below (simply determined by evaluating Lagrange's interpolating polynomials for uniformly spaced data), and N is the number of points used in the extrapolation. Terminology may vary a bit within the field: an extrapolation using N known values results in a Nth order accurate estimate for the value, while the order of the polynomial passing through the points is $N-1$. In this work, the extrapolation order will refer to the the accuracy of the scheme. We only use extrapolations up to fourth-order accurate (using four points), and the coefficients for these operations are given by

$$f(x) = \begin{cases} 1\, f(x+k) & +\mathcal{O}(|\Delta x|) \\ 2\, f(x+k) - 1\, f(x+2k) & +\mathcal{O}(|\Delta x|^2) \\ 3\, f(x+k) - 3\, f(x+2k) + 1\, f(x+3k) & +\mathcal{O}(|\Delta x|^3) \\ 4\, f(x+k) - 6\, f(x+2k) + 4\, f(x+3k) - 1 f(x+4k) & +\mathcal{O}(|\Delta x|^4) \end{cases}, \tag{4.2}$$

where we have dropped the vector notation for convenience, as it is understood that this extrapolation will occur along the grid vector k^i.

Note that higher-order extrapolation coefficients can be easily calculated by constructing Pascal's triangle with alternating sign. Also, increasing the polynomial extrapolation to higher orders by including more points is generally a bad idea because of the rigidity of the Lagrange interpolating polynomial — these polynomials are required to pass through every point in the data set, and thus tend to develop large oscillations between grid-points when fitting noisy data while becoming wildly inaccurate at the ends of the interval of known data (*cf.* discussions in [28, 55]).

Having established a method for constructing valid data on the excision boundary, we now need to ask when to apply this method. To understand this, let us introduce a simple

example. Suppose we are implementing the excision algorithm above in a 1+1-dimensional simulation. For one of the field variables u, we have an evolution equation of the form

$$\partial_t u = \rho = F(u, \partial_i u, \ldots), \tag{4.3}$$

where F is just some function of the field variables and their spatial derivatives. Using the method of lines (MoL), described in §5.1.1, one computes the spatial derivatives on a given time-slice using, here, the finite-difference approximation, and thus constructs ρ. With ρ thus considered to be "given," one then evolves to the next time-step by performing an ODE integration of the equation above forward in time.

So, we suppose we are given data everywhere on the one-dimensional spatial grid of our simple example at some time t^n. We excise a collection of points, and set the values of the field variable u to zero there. In order to update values of u_i^n to u_i^{n+1} on the next time-level, we must compute spatial derivatives to construct ρ_i. As we have mentioned, computing the spatial derivatives at the points labeled `Interior` is trivial; standard centered differences suffice. However, once we are at an `ExBndry` point, we do not have enough data to compute our difference, as is depicted in Fig. 4.4.

As illustrated in that figure, we have two options for proceeding. In the first, we may ignore the `ExBndry` points altogether, and perform an update without them. (We had no difficulty computing the right-hand-side quantities ρ_i for $i > 1$). Once our ODE integration has provided data on the t^{n+1} time-level, we can perform an extrapolation of the new field variables on that time-slice. We call this solution-extrapolation, and it is illustrated by the bold black arrow in 4.4. Our second option for proceeding is to extrapolate the right-hand-side quantities ρ_i for $i > 1$ in order to construct ρ_1. Then, the ODE integrator can integrate all points $i \geq 1$ forward in time, providing the full data on t^{n+1}. This technique, which we call rhs-extrapolation is illustrated by the two open arrows in the figure.

Alcubierre and Brügmann [4] found that they achieved stable evolutions of a Schwarzschild black hole in IEF coordinates when they implemented a cubical excision region and *copied* the right-hand-side quantities computed immediately outside the cube onto the excision boundary. This is effectively a first order rhs-extrapolation in our method. Yo, Baumgarte, and Shapiro, on the other hand, found it necessary to use higher-order extrapolations (specifically, fourth order) when their spacetime was not static. Again, in their work, the black-hole background was given analytically, and they evolved a scalar field on top of this stationary or boosted background.

We have also found in our numerical experiments with a moving black hole that using first-order extrapolations quickly leads to an instability. One of the appealing features of the excision algorithm as we have designed it is that it is very easy to choose solution- or right-hand-side-extrapolation as we see fit.

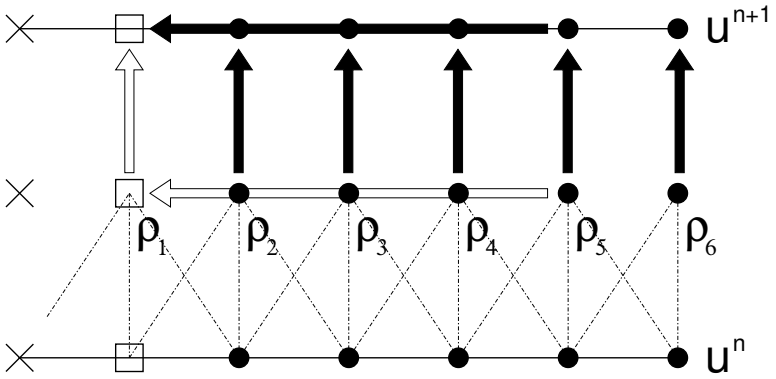


Figure 4.4: Illustration of the use of `ExBndry` extrapolations in the context of an evolution step. Here, a cross ($\times$) denotes an `Excised` point, a box denotes an `ExBndry` point, and filled dots denote `Interior` points. We can provide valid data for the `ExBndry` on the next time step by updating followed by sol-extrapolation (solid arrow) or rhs-extrapolation followed by updating (open arrows).

4.2.3 Dynamic situations: moving the excision region

Having established the general algorithm, let us now discuss how we allow for the excision region to move across the coordinate grid. As the coordinate location of a black hole changes, so does the position of the excision region. We will always control the rate of time-advance in our simulations such that the excision region never moves by more than one grid-point within one time-step. The major concern in allowing the black hole (and hence the excision region) to move across the coordinate grid is that points which were formerly excised (and therefore contain no data for any of the field variables) necessarily become "uncovered." We illustrate this in Fig. 4.5.

The order in which this all happens is important. Let us assume that a single evolution step has just completed. At this point, we check to see where the black hole is located. (The method by which this location is determined could be the use of a "God-given" trajectory, the result of an numerical apparent horizon finder, or any other method which gives a reasonable estimate for the hole location.) At this point, we move our "continuum limit" excision sphere to the new location. We excise all grid-points conforming to our choice of excision shape that fit within the continuum sphere by giving them the label `Excised`. We then relabel those excised grid-points which have non-excised nearest neighbors as `ExBndry`. Having established which points are excision boundary points, we can then construct our database of extrapolation information (location of `ExBndry` points, location of neighbors to be used in extrapolation, *etc.*).

Having done this, there are a few points which were previously excised which are sitting now on the newly redefined excision boundary. These points are indicated as the solid dots in Fig. 4.5. This is the key reason for only allowing the excision region to move by

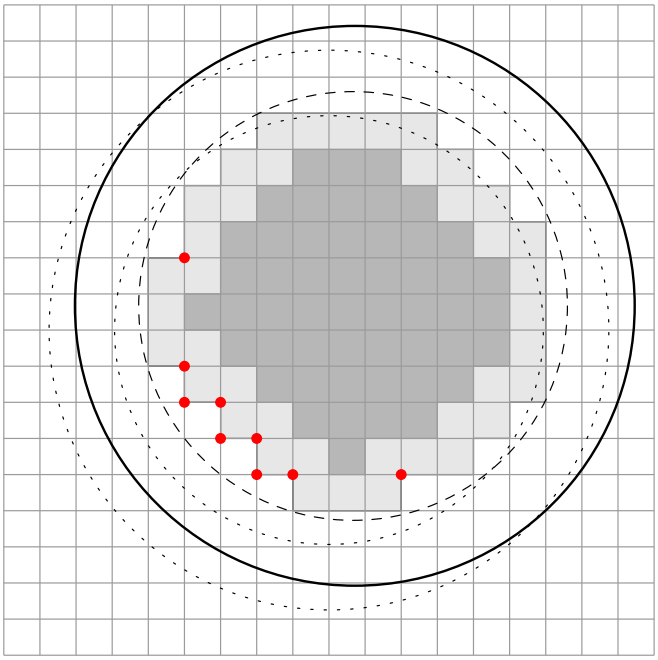


Figure 4.5: Illustration of the need for "populating" recently uncovered points when the excision region is allowed to move. Here the apparent horizon and the continuum excision sphere have moved from their previous, dotted positions. Points which were previously labeled `Excised` (thus containing no data) have now re-entered the computational domain and are re-labeled `ExBndry` (denoted here as solid dots). Before the evolution can proceed, these points must have data provided on them. The dynamic singularity excision algorithm presented herein facilitates the extrapolation onto these newly emerged points.

one point between time-steps. By doing so, the newly uncovered points will always lie on the new excision boundary, where our algorithm naturally constructs a database of points from which the boundary point may extrapolate data. Thus, at this point, we perform an extrapolation onto these newly uncovered points, "populating" them with data. These steps must take place every time the excision region moves.

As a side remark, we notice that in Fig. 4.5, the excision region has changed shape from the way it looked, for example, in Fig. 4.1. This is due to the fact that we allowed the mask center to move to an arbitrary point rather than a grid-point. Any time this happens, the excision region has the possibility to change shape as new points on the grid suddenly fit within the continuum limit sphere and others suddenly are excluded. As the hole moves across the grid, this tends to lead to a rapid fluctuation in the shape of the excision region. While this does not necessarily lead to numerical trouble, it is, at the very least, visually unattractive. For this reason, we introduce a simple boolean parameter (`snap_mask_to_gridpoint`) which, when active, only permits the mask center to "snap" to a grid-point, thus preserving the same excision shape from time-step to time-step as the black hole moves.

4.2.4 Summary of dynamic singularity excision algorithm

Let us now summarize our specification of the dynamic singularity excision method as we have presented it. In the context of performing an evolution update step, we have a very simple iterative loop, in which a single time step is completed via the following procedure. Assuming that the excision mask has been initialized properly immediately after the initial data for the simulation was prescribed, each time step proceeds as follows:

1. For all points labeled `Interior`, compute spatial derivatives with centered-difference approximations (and add in advective pieces).

2. Construct the right-hand-sides of the evolution equations, *e.g.* Eqs. (2.12)–(2.16) from the field variables and their numerically computed spatial derivatives.

3. If rhs-extrapolation is to be used, extrapolate onto the `ExBndry` points from the rhs variables already computed. If not, set the rhs variables to zero on the `ExBndry` points.

4. Perform an ODE time integration to calculate data one time-step into the future, *e.g.*, $\mathbf{u}^{n+1} = \mathbf{u}^n + \boldsymbol{\rho}\Delta t$.

5. If sol-extrapolation is to be used, extrapolate the new solutions for the field variables onto the `ExBndry` points. If not, the evolution step is complete; continue.

6. Check whether the excision mask needs to move. If not, proceed to the next time iteration at Step 1. If so, continue.

7. Re-calculate which points are contained within the continuum-limit sphere, centered at the new black hole location. Label these points as `Excised`. Find non-excised nearest neighbors to the excised points and label these `ExBndry`. Create database of boundary points and their associated extrapolation neighbors.

8. For those points which have been newly uncovered, extrapolate solution variables onto them.

9. Proceed to next time iteration step at Step 1.

4.3 Black hole tracking

For the dynamic evolutions we wish to study in which the black hole is allowed to move across the computational grid, the dynamic excision algorithm of the previous section allows us to handle the bookkeeping and data management necessary to make this process relatively quick and efficient. One issue which we have not addressed, however, is how one tracks the motion of the black hole so that the excision region is properly told "where to go." In order to be justified in excising a region of our spatial slices, it is imperative that the excision region be located within the apparent horizon of the black hole (assuming an apparent horizon does exist in a given time-slice).

Ideally one would perform a search for an apparent horizon at each time step and adjust the location of the excision region correspondingly. Even with the modern batch of freely-available, highly performance-tuned apparent horizon finders [61, 75], this process can be slow and inaccurate. (We do not know of examples of false positives when the tracker finds horizons that do not exist, but quite frequently we have experienced situations in which the horizon finder fails to find horizons that should be present. We have often encountered instances when the tracker under study will fail to find a horizon for a few time-steps, only to find it again without difficulty. Presumably, this happens when some internal iteration scheme fails to converge, subject to slight perturbations in the data coming from the numerical evolution.)

Thus it appears very desirable to have available a fairly reliable, computationally inexpensive method of dynamically adjusting the location of the excision regions to accommodate the movement of the black holes. One may then calculate the apparent horizon at regular intervals to monitor the causal consistency of the excision procedure.

As an implementation of such a scheme, the author had originally proposed an algorithm by which one could use the database of extrapolation points stored in the singularity excision algorithm to perform a fitting of the fall-off behavior of some of the field variables along the extrapolation directions as one approached the singularity. It was thought that by using a fitting of simple polynomials (much like in the extrapolations themselves), a very quick fit to

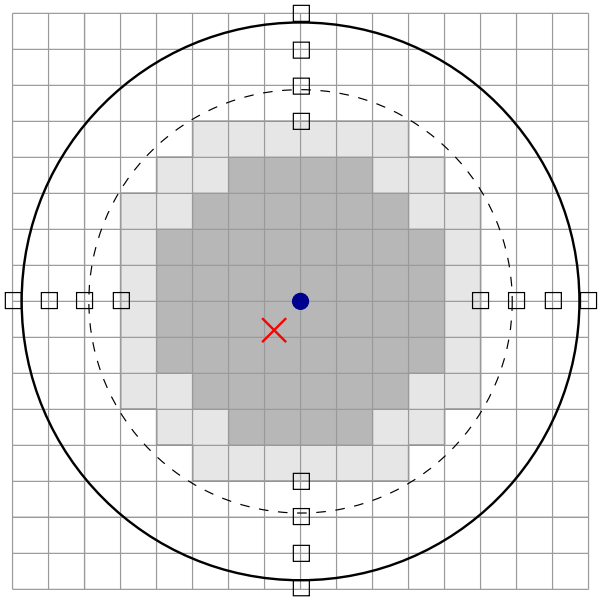


Figure 4.6: A schematic illustration of the excision region, when used in the context of tracking the black hole. The solid dot denotes the current center of the excision region whereas the center of the black hole is located at the '×'. The points commonly used for extrapolation onto the excision boundary, including the boundary points themselves, are denoted by open boxes. Using the dynamic singularity excision infrastructure, we can quickly retrieve the values of a (scalar) field variable at these points and analyze the symmetry of the field with respect to the current excision center in order to estimate the location of the black hole's center.

the data around the excision region could be found, after which we could use an isosurface of some value to estimate a location for the black hole's horizon (assuming large degrees of symmetry). This idea was ill-fated because of the poor fit polynomials make with inverse power law functions, such as the field variables yield near the central singularity in a single, IEF black hole.

Fortunately, we found that we could rather use a simple Gaussian to fit the data of a field variable across the excision region. Prior experience working with data analysis of astronomical spectra had demonstrated that Gaussians can be quite robust at finding the centers of peaks, even when dealing with noisy data. Thus, we describe here, as in [71], the algorithm which we actually employ.

The premise is the same as this author's original idea — to construct a simple scheme to analyzes the fall-off and symmetry behavior of evolution variables near the black hole singularity in order to predict a reasonable value for the central position of the excision region. In Fig. 4.6 we have schematically plotted a 2-dimensional slice of a spherical excision region (the precise shape of that region is irrelevant for our method). The current excision center is marked by a solid dot. Assuming that the black hole is slightly offset with respect

to this position, we have marked by a $\times$ the desirable location of the excision center. In a loose sense we might call this the current black hole position. We note, however, that a precise definition of a black hole center, as for example in terms of a point-like singularity, can only be given for a restricted subclass of black holes.

For concreteness, we now consider analytic data for a static black hole of mass $M = 1$ in IEF coordinates located at $x = -0.105$, $y = 0$, $z = 0$. We describe this data in terms of an excision region with radius 0.7 centered at $x = 0$, $y = 0$, $z = 0$. Quite obviously this is not the optimal position of the excision region and the data will show some asymmetry in their fall-off behavior in the vicinity of the excision boundary with respect to the excision center.

The idea is then to construct some combination of the field variables which adequately exhibits this behavior. Such a combination will in general not be coordinate-invariant and thus depends on the scenario under investigation. For all runs presented in this work, we have found the trace of the extrinsic curvature K a perfectly adequate choice.

A 1-dimensional plot in Fig. 4.7 of K along the x-axis through the excision center reveals the asymmetry. The $\times$ symbols in this figure correspond to the values of K at the 8 points marked by boxes on the x-axis in Fig. 4.6. At the end of each evolution step in our code we fit a Gaussian

$$a\, e^{(x-b)^2/c^2} \tag{4.4}$$

through these points. The parameters a, b and c are obtained from χ^2-minimization and the value for b gives us the x-coordinate of the updated excision center. We then proceed similarly for the y and z direction. If further excision regions are present in the computational domain they are treated in the same way. The total number of points used for this method is a free parameter but we typically find 8 (as in this example) to be sufficient.

With regard to non-stationary scenarios it is, of course, possible that a certain asymmetry of the data might arise from reasons unrelated to the black hole location, as for example in the case of boosted black holes. We emphasize, however, that the purpose of this algorithm is not to provide as accurate as possible an estimate of a black hole center (which in many cases will not be well-defined anyway), but to prescribe a recipe for centering the excision region.

The only requirement for a healthy evolution is that the excision region be contained entirely within the apparent horizon. This may be monitored, for example, by regular calculations of the apparent horizon. In the numerical evolutions presented above we have used a few buffer zones (layers of grid points inside the apparent horizon which are not excised) and verified that the excision region indeed remains confined to the interior of the apparent horizon.

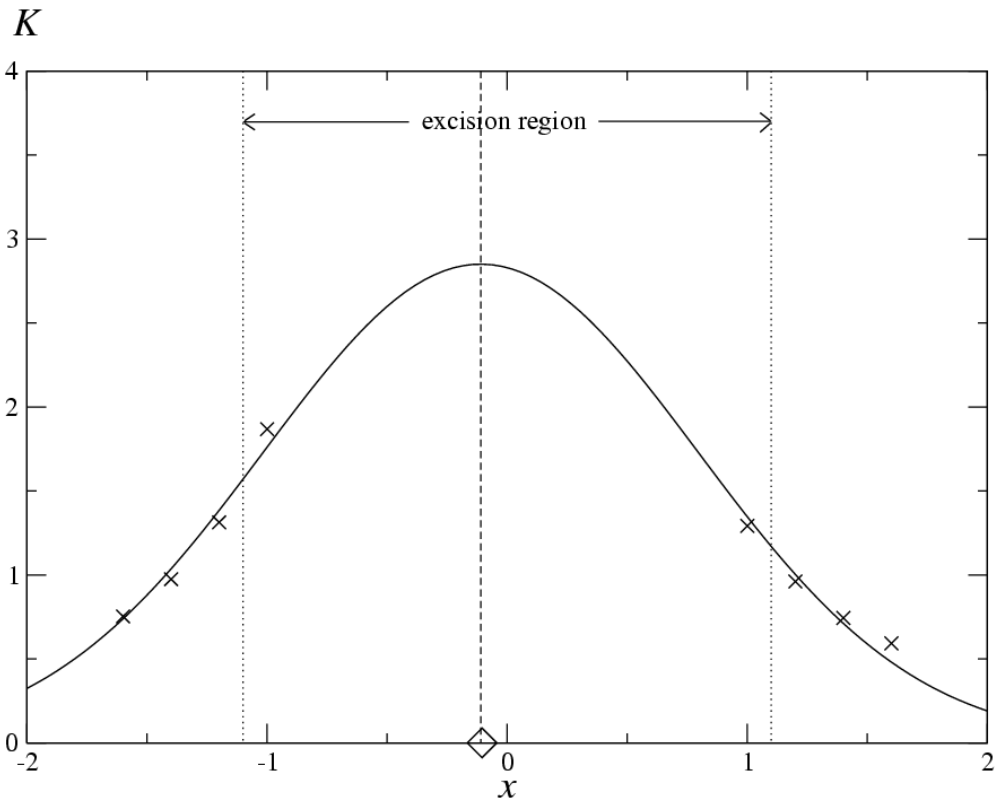


Figure 4.7: Numerical data (×) on the x-axis (*cf.* Fig. 4.6) around the excision region is fitted with a Gaussian curve (solid line). The central position of the Gaussian is used as the updated x value of the excision center. The diamond marks the exact location of the singularity used for these data and is well-approximated by the fitting algorithm.

Thus, with this "GaussTracking" method, we can crudely follow the motion of black holes across our grid, at least insofar as the data shares some of the features of Kerr-Schild type data. We have not fully explored how well this method generalizes to black hole solutions in other coordinates or highly distorted black holes. In summary, though, this method provides an alternative to constant apparent-horizon finding schemes.

Note also that the recent calculation by Brügmann *et al.* employs a similar method for tracking the location of their black holes (they find an isosurface of the lapse) so that they know what shift vector to apply to force their holes back into co-rotation [27]. It would seem, then, that quick, simple-to-implement alternatives to apparent-horizon finders are a useful commodity within the community.

Chapter 5

Applications of dynamic singularity excision

In order to demonstrate the utility of the dynamic singularity excision algorithm detailed in the previous chapter, we will present results from some testbeds and practical applications of the method. As a brief overview, for all of the three-dimensional simulations presented here, we evolve the initial data representing a single Schwarzschild black hole written in ingoing Eddington-Finkelstein coordinates (see App. B). In order to test the dynamic capabilities of the excision algorithm, we have devised a time-dependent coordinate transformation which causes the black hole to continually change its coordinate location.

We find this to be an interesting test of the algorithm, and we report here (as well as in [71]) on how the implementation of a densitized lapse allows for runs of a significantly improved duration over previous simulations reported in [66] in which we employed a fixed (analytic) lapse. We can understand the success of this gauge choice in terms of the results of Khokhlov and Novikov [48], summarized in §2.3.

In this chapter, we will briefly discuss the numerical implementation of the BSSN equations of §2.2, with reference to the dynamic singularity excision module of the previous chapter, and then proceed to describe the nature of our numerical experiments and their outcomes.

5.1 Description of the Maya code

Building on the experience many members of the Penn State numerical relativity group gained during the Binary Black Hole Grand Challenge Alliance during the 1990's, a code dubbed Agave was developed, culminating in a study of the grazing collision of two black holes at close separation [23]. Around the time of that publication, work on a new code was begun, utilizing the new Cactus 4.0 infrastructure being developed at the AEI [29]. As originally implemented, this new code, named Maya, was an updated and drastically simplified ADM 3+1 evolution code. Having gone through many redesigns, new developers, and a change of formulation, it has been a useful tool for studying dynamic spacetimes in the fully nonlinear regime. Currently, it implements the BSSN evolution system, Eqs. (2.12)–(2.16).

5.1.1 Infrastructure

One advantage of MAYA is its emphasis on simplicity and readability. Designed with the the numerical method of lines [60] in mind, it views the evolution system in the form:

$$\partial_t \mathbf{u} = \boldsymbol{\rho} = \beta^i \partial_i \mathbf{u} + \mathbf{T}(\mathbf{u}, \partial\beta) + \mathbf{R}(\mathbf{u}, \partial\mathbf{u}, \partial^2\mathbf{u}), \tag{5.1}$$

where $\mathbf{u}$ is a vector containing the evolution variables, $\mathbf{u} = \{\phi, \widetilde{\gamma}_{ij}, K, \widetilde{A}_{ij}, \widetilde{\Gamma}^i\}$. In the method of lines (MoL), one chooses a numerical approximation technique to calculate approximations to spatial derivatives, constructs the right-hand-sides of the evolution equations algebraically from the variables and their appropriate numerically approximated derivatives, and then treats the entire system as a set of coupled ordinary differential equations (ODE's) to integrate forward in time. This allows for a very flexible code infrastructure in that the same approximation routines may be used for all of the different evolution variables. In MAYA, we have implemented standard second-order-accurate finite-differencing approximations to calculate our spatial derivatives.

Notice also that in Eq. (5.1), we have identified special features within these spatial derivatives that may be treated differently. We have separated the Lie derivatives along β^i into distinct pieces. The $\beta^i\partial_i$ are referred to as the **advective** terms as they share many features with the $v^i\partial_i$ terms one encounters in numerical hydrodynamics simulations. These are treated specially with the so-called upwind (one-sided) derivatives. The $\mathbf{T}$ terms above are labeled the "twist" terms, and involve all of the derivatives of β^i, with appropriate contractions with the evolution variables. Finally the remaining $\mathbf{R}$ terms are simply the rest of the right-hand-side expressions not already explicitly mentioned. In simulations in which the shift is zero, we would have $\boldsymbol{\rho} = \mathbf{R}$. Aside from the advective terms, which, again, are computed using second-order accurate or higher one-sided finite-difference stencils, the other terms $\mathbf{T}$ and $\mathbf{R}$ are computed using standard second-order centered-differencing stencils. Looking at the BSSN evolution equations, Eqs. (2.12)–(2.16), and the corresponding equations for the Lie derivative terms, Eqs. (2.17)–(2.20), we can see that this nicely encapsulates the different features of these equations.

While there has been much recent work on the subject of appropriate boundary conditions in numerical relativity, we typically will employ either Dirichlet conditions (setting the field variables to prescribed values, which may be time-dependent) or a modified Sommerfeld condition. In the latter, one assumes that near the edges of the domain, a given field has the behavior

$$u = u_0 + \frac{w(t-r)}{r^n}, \tag{5.2}$$

where u_0 represents a prescribed function, which is typically taken to be the analytic solution

(if known) or the initial data, and n is a fall-off parameter assumed to be known for the given field. For simplicity, we assume all fields propagate at the speed of light ($v = 1$). Numerically, the condition above translates into the differential relation

$$\partial_t u = \partial_t u_0 - \frac{x^i}{r}\partial_i(u - u_0) - n\frac{(u - u_0)}{r}\,. \tag{5.3}$$

The right-hand-side of this equation is discretized with an upwind derivative as it also has the form of an advective derivative, with the time-derivative of the prescribed function ($\partial_t u_0$) given.

In summary, then, all spatial derivatives across the interior points are computed using either centered differences or upwind differences. At the outer boundary, either Dirichlet conditions or the radiative boundary condition Eq. (5.3) are used. On the excision boundary, we use the infrastructure explained in the previous chapter to provide valid data at all `ExBndry` points.

With the spatial derivatives computed, the right-hand-sides of the evolution equations can be constructed algebraically. Then, to integrate forward in time, we use the Iterative Crank Nicolson (ICN) method [72] as an ODE integrator. The method for handling the excision boundaries for situations in which extrapolation of the right-hand-sides is desired was explained in the previous chapter. This allows us to march the data forward into the future, time-slice by time-slice as desired.

This completes the description of the main evolution loop. We have implemented various different diagnostic tools and schemes for determining the black hole location (such as coupling an apparent horizon finder or the 'Gauss' tracker of the previous chapter into our evolutions), but the core of the code follows the schematic laid out above.

5.1.2 Gauge choices implemented

One of the most difficult tasks in numerical relativity is constructing an intelligent choice for the gauge/slicing conditions. Very few of the conditions have the simple physical interpretation that, for example, one finds for the maximal slicing condition for the lapse or the minimal shear condition for the shift. In general, we are content if we find a condition that works for the case at hand. In manifestly hyperbolic formulations, work has progressed in attempts to fold evolution equations for the lapse and shift into an existing hyperbolic system such that the enlarged system is also manifestly hyperbolic [52].

Other than wishing to impose symmetric hyperbolicity, most motivations for constructing gauge conditions come from attempting to derive computationally efficient evolution equations that share similar behavior with maximal slicing or minimal shift.

In single black-hole simulations, the '1+log' evolution equation for the lapse (discussed in §3.1.1) and the 'Gamma-driver' evolution equation for the shift are frequently employed

[4, 79], namely

$$\partial_t \alpha = 6\partial_t \phi = -\alpha K + D_i \beta^i, \tag{5.4}$$

$$\partial_t \beta^i = \lambda \partial_t \widetilde{\Gamma}^i, \tag{5.5}$$

where $\partial_t \phi$ and $\partial_t \widetilde{\Gamma}^i$ are the evolution equations for the BSSN variables ϕ and $\widetilde{\Gamma}^i$, Eq. (2.12) and Eq. (2.16). These conditions have been seen to work particularly well for these evolutions, but we have had limited success in applying them to our dynamic spacetimes.

Thus, in lieu of a gauge more suited to dynamic scenarios, we often fall back on fixed gauge choices, where we can use some degree of foreknowledge such as the analytic solution to prescribe a lapse and shift as a function of the coordinates. As we now know from [48], discussed in §2.3, this gauge choice is ill-posed. To fold in some knowledge of the analytic solution without selecting an ill-posed lapse, we implement an algebraic condition, namely the densitized lapse condition, Eq. (2.32).

To be clear, prescribing the lapse of an analytic solution directly, unless it happens to be a spatially-constant function of the time coordinate, is an ill-posed choice. Yet, when our evolved gauge conditions that serve us so well in stationary spacetimes fail us in dynamic scenarios, we have few places left to turn. The densitized lapse condition allows us a way out — we can impose the *densitized lapse*, q, corresponding to an analytic solution, and then calculate the real lapse from this and the determinant of the metric, which will evolve. (Note that we can calculate the determinant of the metric from the definition of the BSSN variable ϕ, Eq. (2.5).) Because ϕ is a numerically evolved quantity, in a qualitative manner of speaking, we enable the gauge conditions to respond in a more flexible way to the numerical evolution.

For our purposes, we also find it useful to reformulate Eq. (5.4) in terms of the densitized lapse. This leads to

$$\partial_t q = e^{-6n\phi}(1 - nq^{-1}) 6\partial_t \phi. \tag{5.6}$$

5.2 Numerical evolutions of a single black hole

Using the computational infrastructure described above, we perform evolutions of single black hole spacetimes in the BSSN system. Particular to that system, in all simulations we actively enforce $\widetilde{A}^i{}_i = 0$ after each time step. We always use a Courant factor (dt/dx) of 0.25. The other grid parameters as well as gauge and boundary conditions will be discussed separately for each case studied below. Many of the results displayed here have also been reported in our recent paper [71].

5.2.1 Static black hole evolutions

We first consider the numerical evolution of spacetimes containing a single Schwarzschild black hole, in which no modifications are made to the IEF coordinates so that the hole remains eternally at a fixed coordinate position. Other simulations of this configuration have recently been evolved to essentially infinite times in the works by Alcubierre *et al.* and Yo, Baumgarte and Shapiro [6, 79], without the usual imposition of reflection symmetries in the latter work. In [66], we were also able to demonstrate the ability to evolve a single black-hole to a numerically static configuration (*i.e.*, the time-derivatives of the field variables approach zero to within machine precision). We will refer to this condition as **numerical staticity**.

With these works representing the current state-of-the-art in three-dimensional simulations of static black holes, we consider them an effective benchmark against which we can compare our simulations. Specifically, we will require that regardless of what modifications are made to the code, we preserve the ability to perform these single-black hole evolutions to numerical staticity.

For the evolutions presented here, we have adopted a cubical excision region chosen to fit within a radius of $r_{\rm exc} = 1.5M$. The computational domain assumes octant symmetry, and the outer boundary is located at $x, y, z = 12M$. Choosing one of the scalar field variables, the trace of the extrinsic curvature K, to focus our attention on, Fig. 5.1 shows the ℓ_2-norm of the time derivative of K for three simulations. The results show that, as expected for static solutions, $\partial_t K \to 0$.

The solid and dashed lines in this figure (almost coincident for most of the plot) are from runs with resolution $0.4M$ in which the simple excision algorithm of [6] is used. That is, the grid-functions on the excision boundary are updated by extrapolating the time derivative from the `Interior` neighbor closest to the normal direction. For the solid line, the standard first-order-accurate extrapolation of that reference is used (copying), whereas for the dashed line we use a third-order accurate extrapolation.

Besides the way updating at the excision boundary is handled, these three runs differ in the recipe for fixing the slicing. For the solid line, we use the classical 1+log condition in terms of α; that is, the lapse is computed from Eq. (5.4). In the dashed line, we apply instead the densitized 1+log version given by Eq. (5.6). It is then clear in Fig. 5.1 that for simple excision the choice of classical versus densitized 1+log slicing condition does not affect the quality of the simulation.

The dotted line represents a novel experiment, which, to our knowledge has not been reported in other works. In most of these single-black hole runs in which numerical staticity is demonstrated, evolved gauges *must* be used to achieve staticity. This run was intended to address the question of whether such long-term stable evolutions can also be achieved if

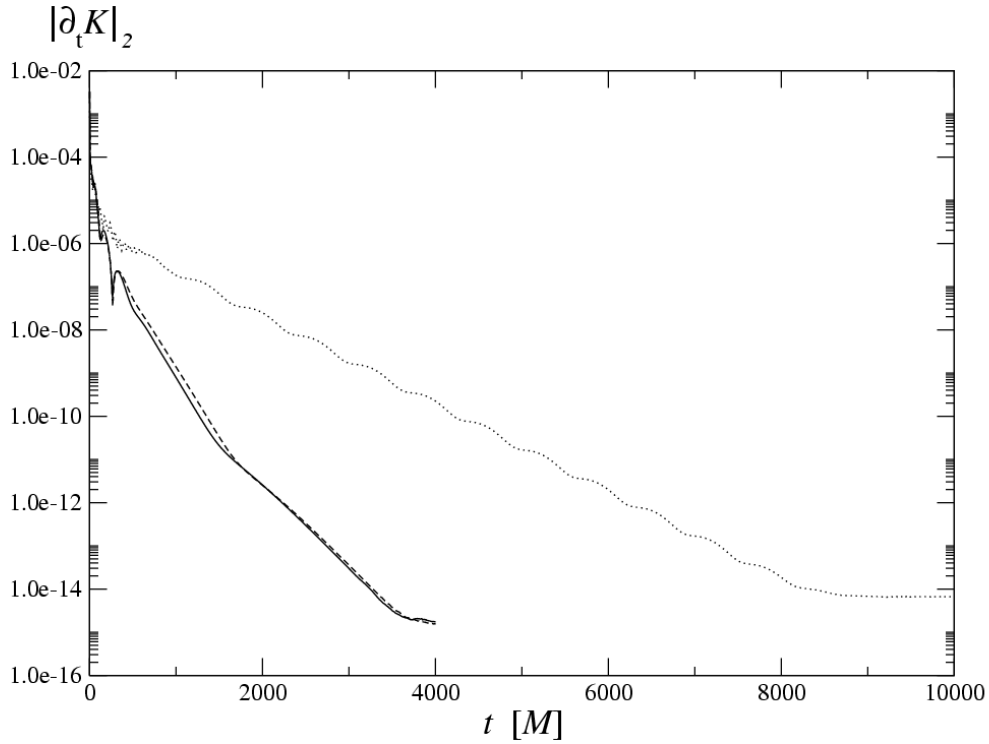


Figure 5.1: The ℓ_2-norm of the time derivative of K for the evolution of a static IEF black hole in octant symmetry using 1+log slicing expressed in terms of the lapse (solid curve) and densitized lapse (dashed curve), both with spatial resolutions of $0.4M$. The dotted line is from a simulation using an algebraic densitized lapse, with resolution $0.2M$.

we use the purely algebraic gauge conditions given by Eq. (2.32), with q obtained from the exact IEF solution. The answer is affirmative; that is, condition (2.32) in conjunction with second-order extrapolation on the excision boundary yields qualitatively similar results to those depicted by the solid and dashed lines. The motivation for switching for this run from simple excision to an excision with extrapolation of the grid-functions lies in anticipation of the wobbling black hole simulations. When the evolved black hole solution is time-dependent (as is the case of a wobbling black hole), simple excision is no longer suitable. The copying of time-derivatives at the excision boundary used in simple excision becomes effectively a boundary condition on the spatial derivatives of grid-functions. This boundary condition is incompatible with the outflow nature of the excision boundary.

Comparison of the dotted line with the solid and dashed lines in Fig. 5.1 shows that the algebraic densitized lapse run possesses long-term stability. The difference is mostly in the variation of the time when machine precision is reached. In summary, using an analytic densitized lapse q or equivalently an algebraic lapse α, we are able to evolve a single black hole with stability properties comparable to those obtained with differential conditions such as 1+log slicing and the Γ-driver condition for the shift vector. While it is not clear how helpful such analytic slicing conditions will be for simulations of a merging binary black hole, the introduction of a densitized lapse makes them at least available for serious consideration in long-term stable simulations carried out with the BSSN scheme.

5.2.2 Moving black hole evolutions

We now turn our attention to single black holes moving across the numerical grid. One way to obtain such a scenario is to evolve a single boosted black hole. The ensuing motion will, however, move the black hole off the computational domain. With the sizes of the computational domain currently restricted by available hardware resources, this will happen on time scales significantly shorter than those considered relevant for black hole orbits or for testing long-term stability of simulations, namely simulations lasting more than $1000M$.

In [66], we introduced therefore an alternative approach which facilitates the motion of single black holes with trajectories confined to the computational domain. We transformed the IEF black hole solution to coordinates $\bar{x}^\mu$ such that

$$t = \bar{t}, \tag{5.7}$$

$$x^i = \bar{x}^i + \xi^i(\bar{t}). \tag{5.8}$$

In terms of the new coordinates, the line element becomes

$$ds^2 = -\alpha^2 d\bar{t}^2 + \gamma_{ij}\left(d\bar{x}^i + \bar{\beta}^i\, d\bar{t}\right)\left(d\bar{x}^j + \bar{\beta}^j\, d\bar{t}\right), \tag{5.9}$$

where

$$\bar{\beta}^i = \beta^i + \dot{\xi}^i. \tag{5.10}$$

In our earlier work [66], this method was used to move black holes on circular and bouncing trajectories. Using either a cubical or spherical excision region, these runs lasted for about $130\,M$, though the apparent horizon started to intersect the excision region at about $90\,M$ and the apparent horizon finder failed to give reasonable results afterward. The fixed gauge conditions used in these evolutions were considered a crucial limiting factor in those evolutions.

Using the densitized lapse condition (2.32) with $n = 1$, we performed similar simulations to those in [66]. We fix the value of q in Eq. (2.32) from the exact single IEF black hole solution transformed according to Eqs. (5.7), (5.10). In Fig. 5.2, we plot the ℓ_2-norm of the Hamiltonian constraint (upper panel) and normalized Hamiltonian constraint (lower panel) for simulations of a black hole on a circular path with radius $2\,M$ and orbital angular frequency $(4\,M)^{-1}$. The normalized version of the Hamiltonian constraint is defined by

$$\mathrm{n}H = \frac{e^{-4\phi}R + \frac{2}{3}K^2 - \hat{A}_{ij}\hat{A}^{ij}}{\sqrt{(e^{-4\phi}R)^2 + (\frac{2}{3}K^2)^2 + (\hat{A}_{ij}\hat{A}^{ij})^2}}. \tag{5.11}$$

Equatorial symmetry is assumed for all the simulations, and the outer boundary conditions consist of setting the values of the grid-functions to the exact analytic solution.

For comparison, in Fig. 5.3 we also show the results from our earlier work in which simulations lasted $< 130\,M$ [66]. In that work, we used a domain size of $20 \times 20 \times 7\,M^3$ for all runs, and so a direct comparison can be made for the circling case. This comparison clearly demonstrates the tremendous improvement on the duration of the simulations when a densitized lapse is used. Some of the evolutions are stable at least for $6000\,M$, when the runs were terminated due to limitations of computational resources. Fig. 5.2 also demonstrates that, even though the durations of the simulations have been improved by at least an order of magnitude, the simulations continue to be affected by boundary effects. This is expected since it is well known that setting boundary conditions to the exact analytic solution is conducive to numerical instabilities.

In Fig. 5.4 we show snapshots of the evolution obtained on the domain $20 \times 20 \times 7\,M^3$ at $t = 0$, $2000\,M$, $4000\,M$, $6000\,M$. Here we plot the variable K on the xy-plane at $z = 0$, namely the orbital plane. The excision region is clearly visible and has been checked to lie within the apparent horizon. We find the apparent horizon area to remain within a few percent of its analytic value $16\,\pi\,M^2$ and the deviation from spherical shape to be of the same order. Here we have used the "Gauss" tracker described in §4.3 to enable the excision region to follow the motion of the black hole. This method is in general not coordinate

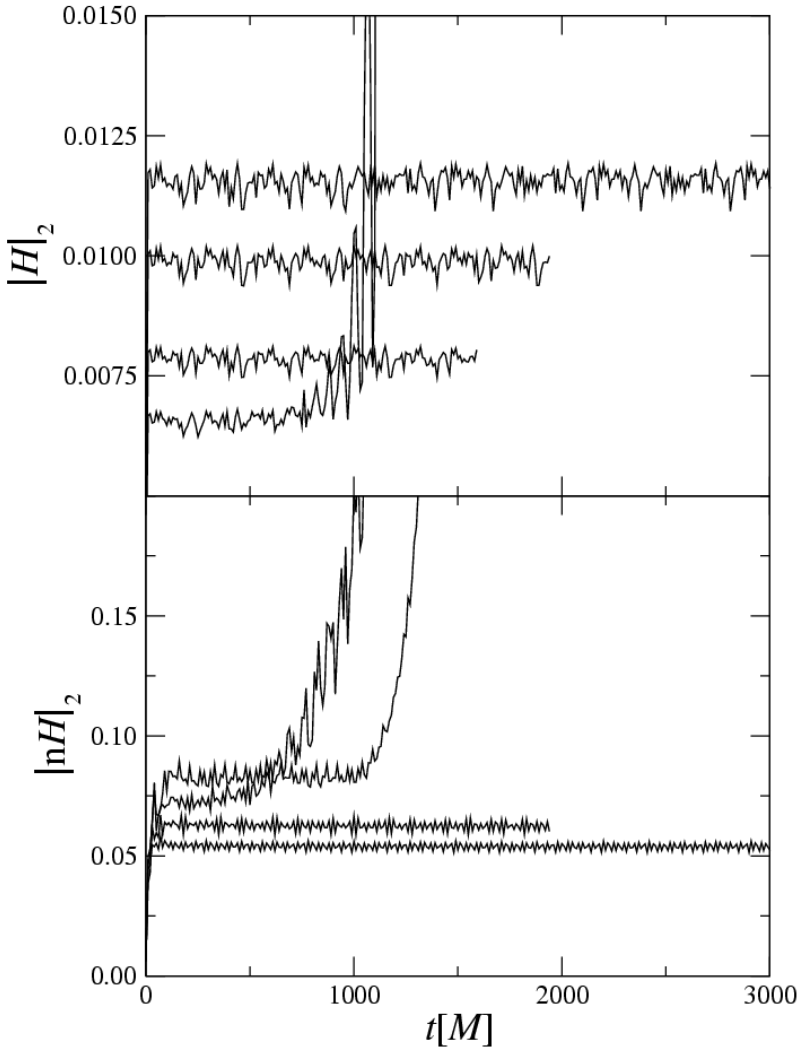


Figure 5.2: The ℓ_2-norm of the Hamiltonian constraint (upper panel) and normalized Hamiltonian constraint (lower panel) are shown as a function of time for the circling black holes for different sizes of the computational domain, using resolution $0.2\,M$. In the upper panel from top to bottom, each data set corresponds to computational domains $20 \times 20 \times 7\,M^3$, $20 \times 20 \times 10\,M^3$, $30 \times 30 \times 7\,M^3$ and $30 \times 30 \times 10.5\,M^3$, respectively. In the lower panel, the correspondence is reversed. That is, from top to bottom at early times, each data set is for domains $30 \times 30 \times 10.5\,M^3$, $30 \times 30 \times 7\,M^3$, $20 \times 20 \times 10\,M^3$ and $20 \times 20 \times 7\,M^3$, respectively. The shortest scale is along the direction perpendicular to the orbital plane.

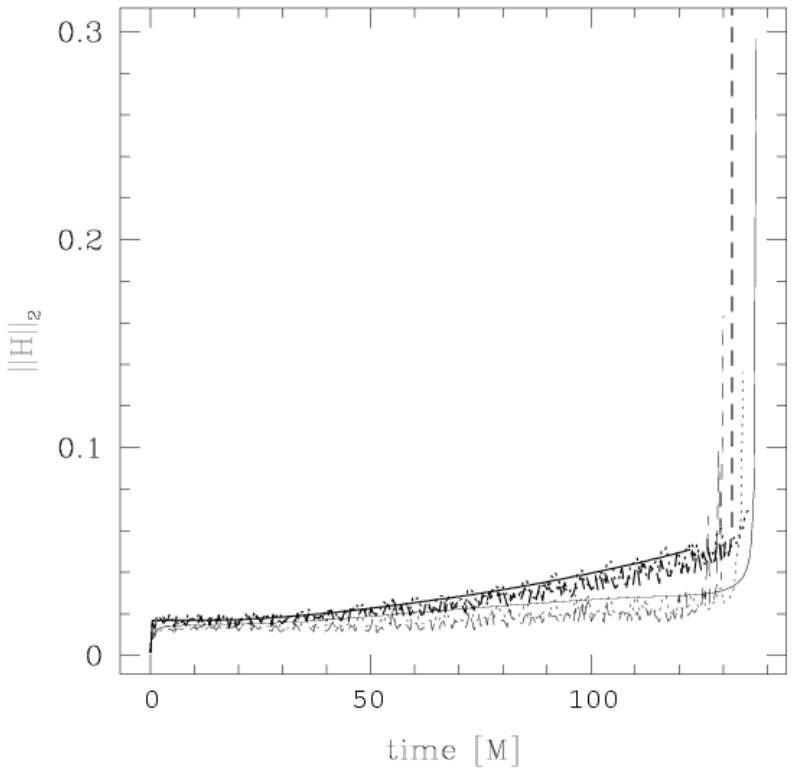


Figure 5.3: Data originally presented in [66], in which our simulations are thought to have been limited by the choice of a fixed gauge condition. Here, for a domain of $20 \times 20 \times 7\,M^3$, we plot the ℓ_2-norm of the Hamiltonian constraint for three different simulations. The solid line depicts data from an unmoving black hole sitting at the origin, the dotted line from a hole which oscillates sinusoidally along the x-axis, and the dashed line from the circling black-hole simulations, which can be directly compared with the curve in Fig. 5.2 with the corresponding domain size.

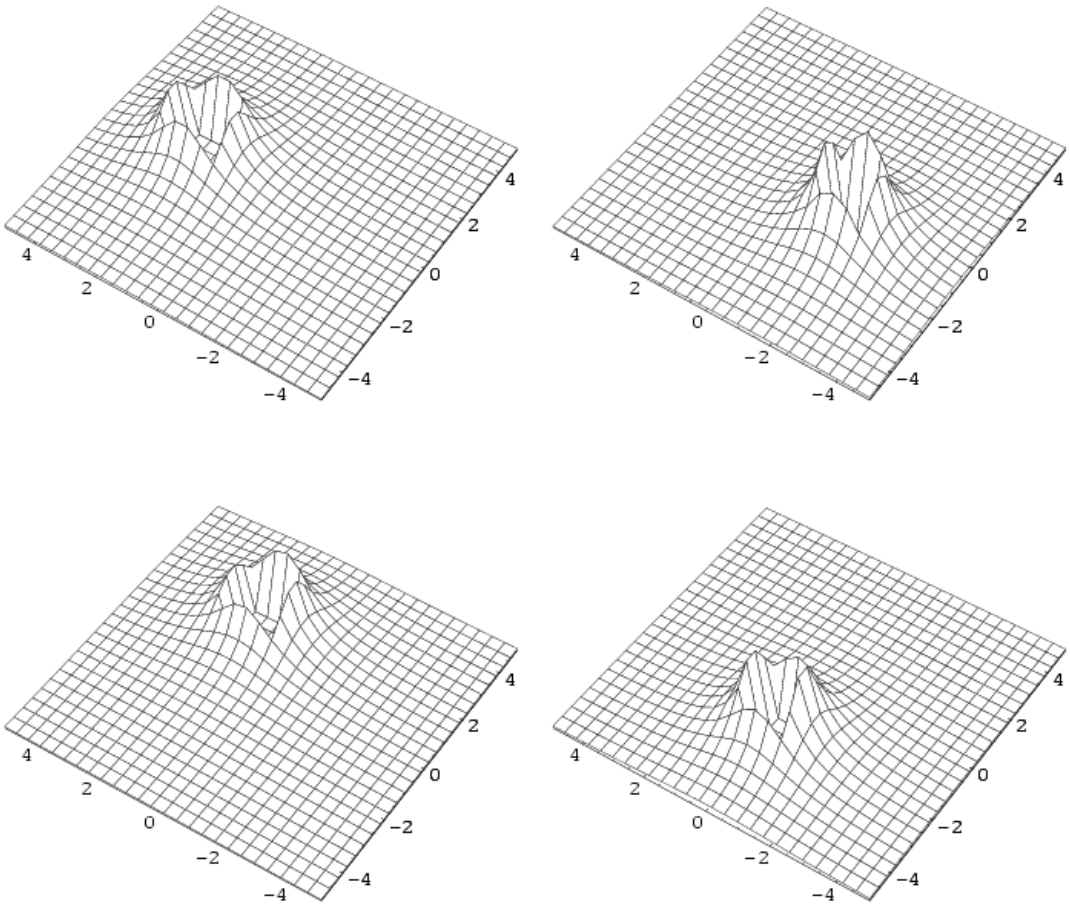


Figure 5.4: The four panels show snapshots of the evolution of K for a circling black hole in the xy-plane at times 0, $2000\,M$, $4000\,M$ and $6000\,M$.

invariant and its performance needs to be monitored by verifying that the excision region always remains inside the apparent horizon. We have verified this for all runs presented here.

By construction, our numerical code is second-order accurate in the spatial and temporal discretization of derivatives. The only exceptions are the advective shift derivatives and the excision extrapolations which are third-order accurate. An important check is thus investigating the convergence properties of the code. For this purpose, we have repeated the evolution of a circling black hole using finer resolutions ($0.2\,M$ and $0.125\,M$) and monitored the violation of the Hamiltonian constraint. The quantity depicted in Fig. 5.5 is the convergence order computed from the ℓ_2-norms according to

$$a = \frac{\ln\left[\,\big|u(h_1)\big|_2 \,/\, \big|u(h_2)\big|_2\,\right]}{\ln\left(h_1/h_2\right)}\,, \tag{5.12}$$

where h_1 and h_2 denote the two levels of refinement with $u(h_1)$ and $u(h_2)$ the corresponding violations of the Hamiltonian constraint. In Fig. 5.5, we plot the convergence order obtained using resolutions of $0.25\,M$ and $0.2\,M$ (upper panel) as well as $0.25\,M$ and $0.125\,M$ (lower panel). The coarse resolutions yield a mean convergence order 2.1 with standard deviation 0.12, consistent with the overall order of discretization used in the code. The convergence order using resolutions $M/4$ and $M/8$ has a mean of 1.66 and exhibits a larger amplitude spread, with a standard deviation of 0.48.

We have found that the deterioration of the convergence is restricted to the immediate vicinity of the excision boundary, where the code is more susceptible to outer boundary effects due to extrapolation. As a consequence, the violations of the Hamiltonian constraint are largest near the excision boundary; these values dominate the ℓ_2-norms and the associated convergence orders drop to lower order. No deterioration in the converge factor is observed if ℓ_1-norms are used instead. To demonstrate that the deterioration of the convergence order is restricted to the vicinity of the excision boundary, we show in Fig. 5.6 convergence plots obtained as before, but now using only points outside the apparent horizon. The fact that the convergence order is slightly larger than 2 is due to the third-order-accurate advective derivatives as well as the coarseness of the resolutions.

We have investigated the reasons behind this spread and found it to be due to a combination of outer boundary and trajectory turning points effects. A Fourier analysis of the $0.25M$ and $0.125M$ convergence order data reveals a period of $2\,\pi\,M$, one-fourth the orbital period. A closer look reveals that the drops take place when the black hole comes closest to the outer boundary. At these locations, the effects from outer boundary conditions on the excision boundary are strongest. These effects can be clearly observed from Fig. 5.7 where we plot the normalized Hamiltonian constraint in the xy-plane at times $0\,M$, $240\,M$, $480\,M$ and $720\,M$. This figure not only demonstrates that the constraint violation is strongest

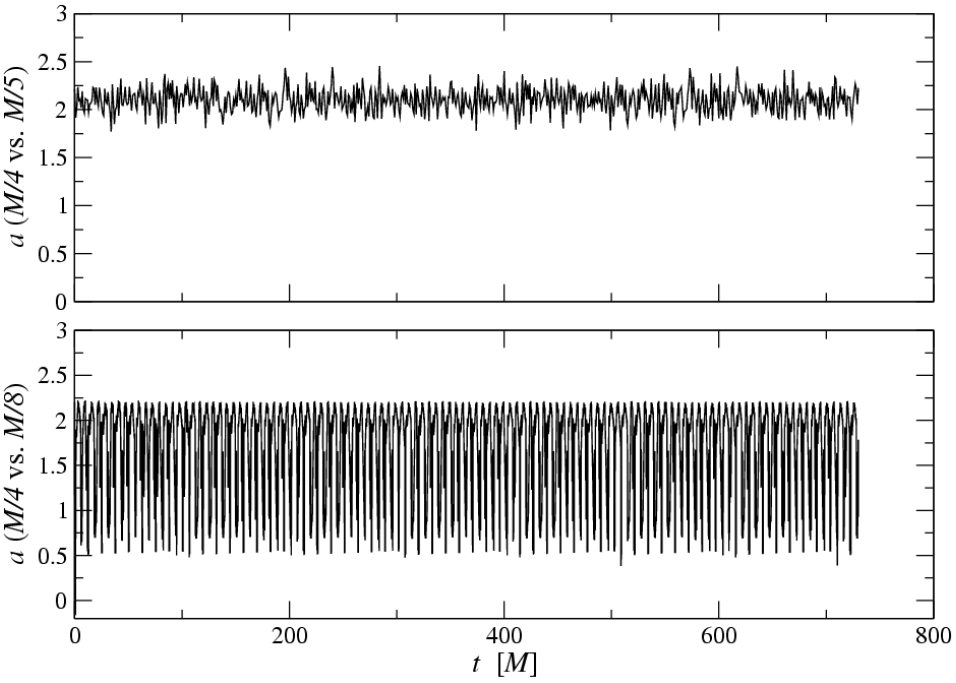


Figure 5.5: The order of convergence for the circling black hole as obtained from evolutions using resolutions $0.25M$ and $0.2M$ (upper panel) and $0.25M$ and $0.125M$ (lower panel). For the upper panel, the mean and standard deviation for a are 2.10 and 0.12, respectively. For the lower panel, the mean and standard deviation for a are 1.66 and 0.48, respectively.

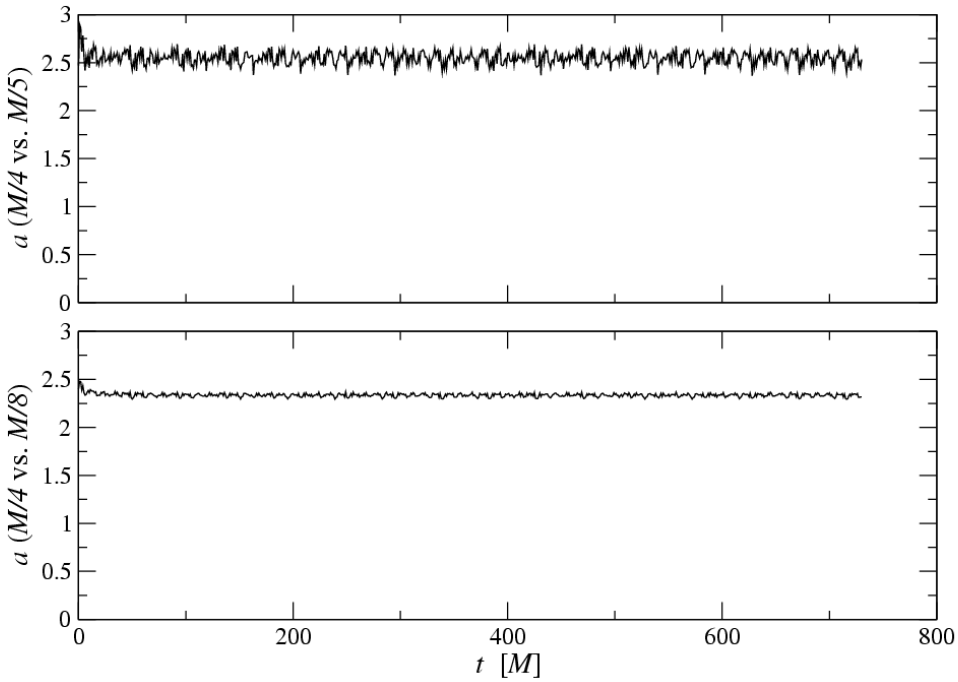


Figure 5.6: The order of convergence for the circling black hole as obtained from evolutions using resolutions $0.25M$ and $0.2M$ (upper panel) and $0.2M$ and $0.125M$ (lower panel). Only points outside the apparent horizon are used for this calculation.

near the excision boundary but also illustrates the correlation of the constraint violation with the corners of the excision and outer boundary.

In spite of the periodic drop in convergence, the lower panel of Fig. 5.5 indicates that no overall deterioration occurs over long integration times. We have checked this quantitatively by analyzing consecutive intervals of length $100\,M$ and calculating the average and root-mean-square deviation of the convergence factors in these intervals. The average remains constant to within three significant digits and the standard deviation to within two.

In order to further test the robustness of our excision infrastructure, we have performed evolutions using a different black hole trajectory. Having in mind the eventual target of simulating in-spiraling binary black holes, it will be of particular interest to see whether the code is able to evolve a black hole on an in-spiraling trajectory. We fake such a situation by introducing a new time-dependent path for the black hole location such that it mimics the motion of a particle in a central harmonic potential through a viscous medium.

Again, there is no new physics being simulated here; we are simply applying a "cute" coordinate transformation that emulates an in-spiraling motion via (5.7). The black hole is initially placed at a distance of $2.4\,M$ from the origin at $z = 0$ and given an initial purely tangential velocity $v = 0.3$. Because of the damping, the velocity and radius will decrease and the black hole would eventually approach a steady state at the origin. In order to keep the evolution dynamic for long times, we switch the sign of the damping term when the radius shrinks below $0.25\,M$ after which the hole starts spiraling outward. The damping constant is switched back to a positive value once a radius of $2.5\,M$ is reached and so on. The resulting trajectory in the xy-plane is displayed in Fig. 5.8. This path corresponds to an evolution lasting $3500\,M$ when we decided to terminate the evolution.

The apparent horizon finder had no difficulty calculating the outermost trapped surface every $20\,M$. The resulting horizon area is shown as a function of time in Fig. 5.9. The area remains constant with good accuracy and deviates by a few percent from the analytic value of $16\pi\,M^2$. Similarly we found the error in the normalized Hamiltonian constraint (*cf.* Eq. (5.11)) to be constant to within about 6%.

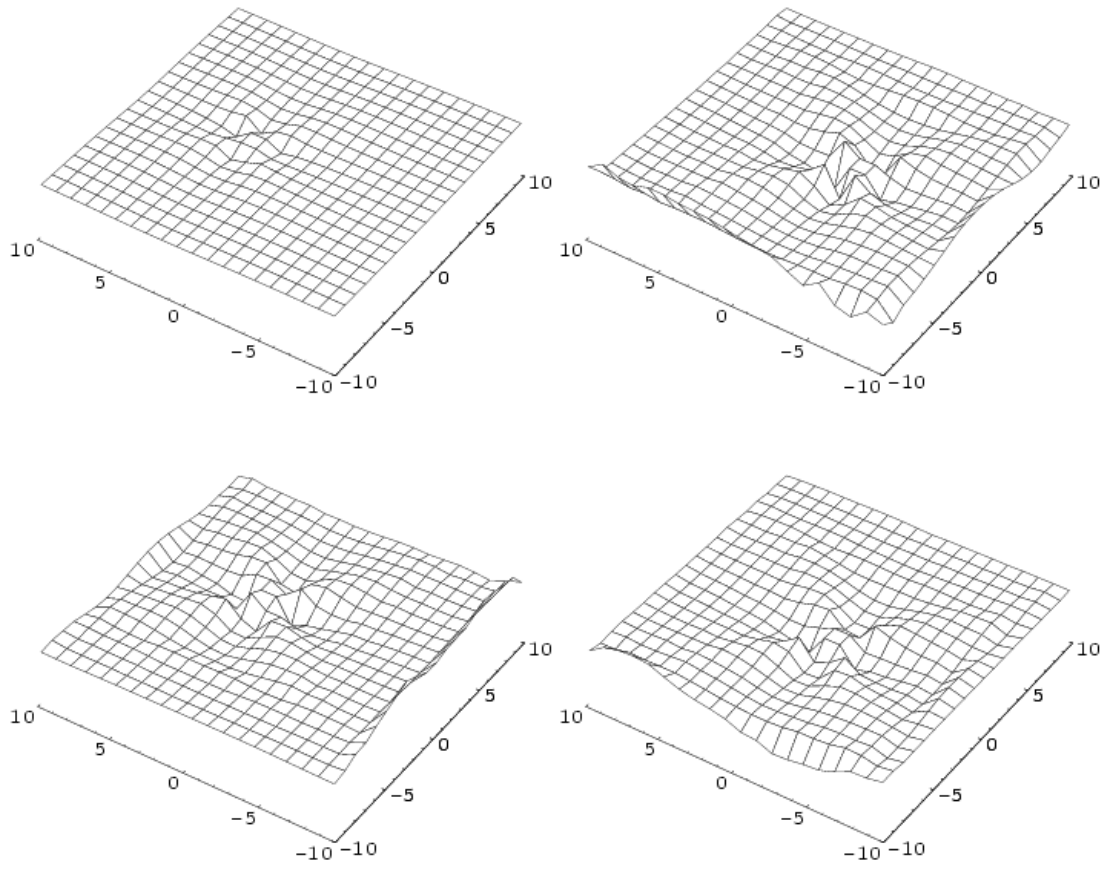


Figure 5.7: The four panels show snapshots of the evolution of the normalized Hamiltonian constraint for a circling black hole in the xy-plane at times 0, $240\,M$, $480\,M$ and $720\,M$.

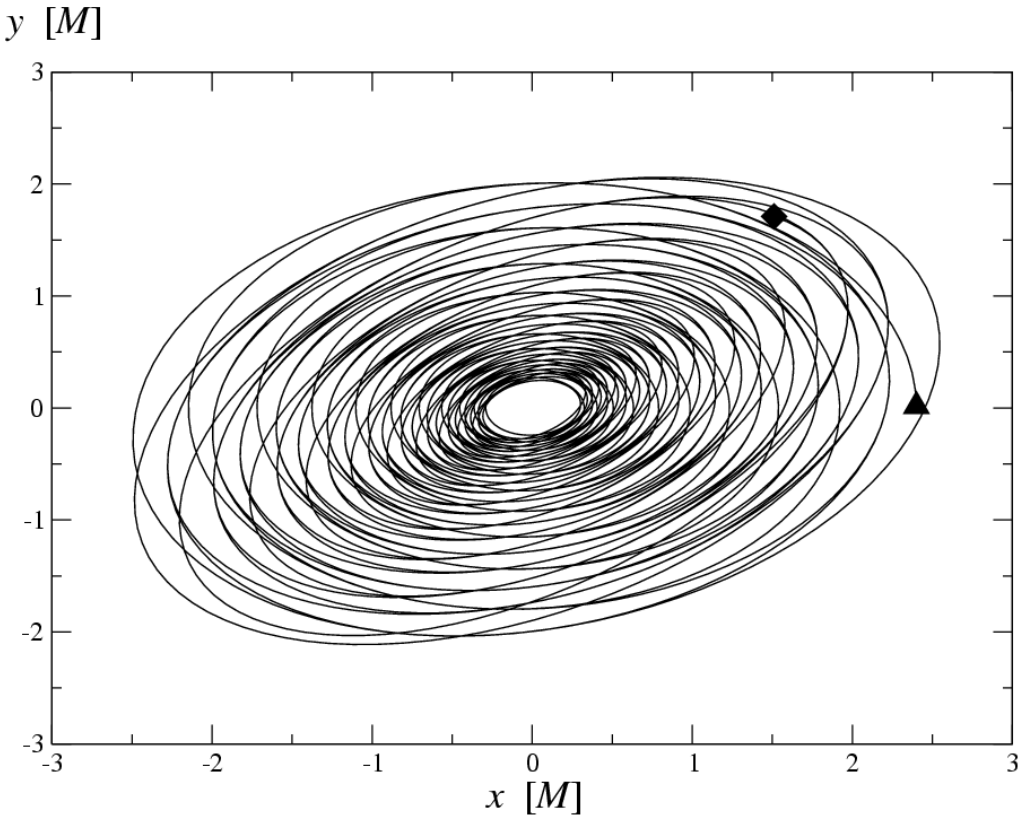


Figure 5.8: The trajectory of a black hole in-spiraling between radius 2.5 and 0.25. The initial position is marked by the filled triangle and the end position by the diamond. The z-position will remain zero throughout the evolution.

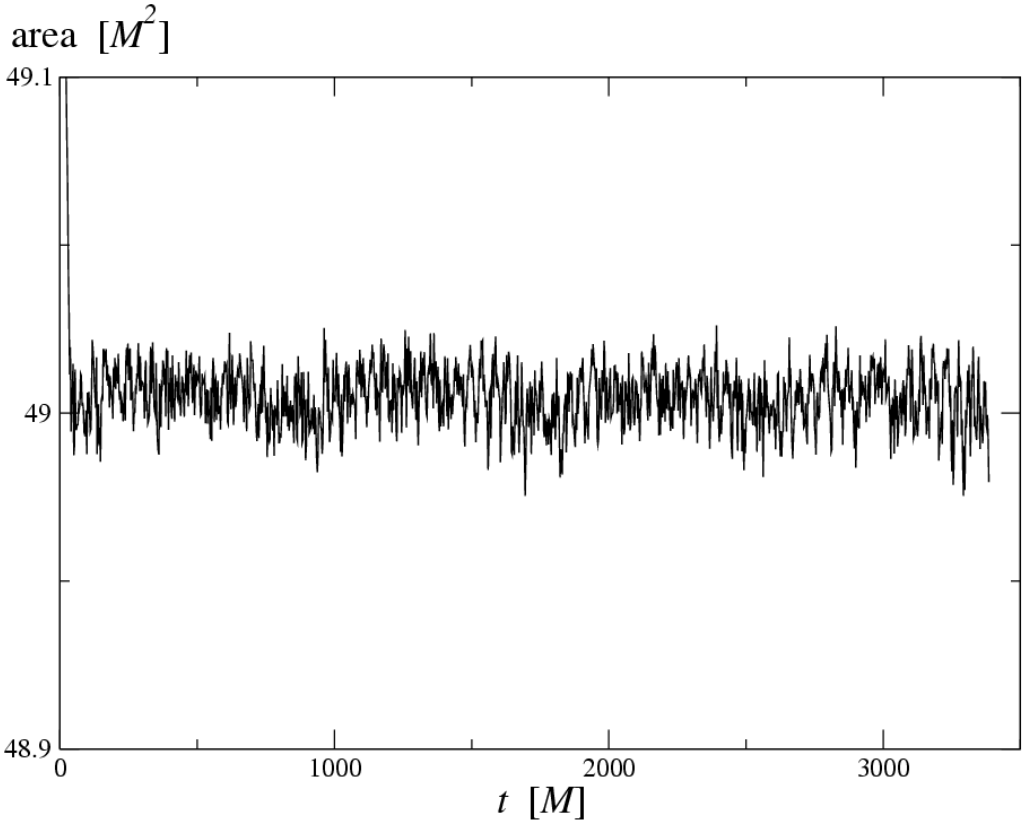


Figure 5.9: The apparent horizon area for the evolution of the in-spiraling black hole whose trajectory is illustrated in Fig. 5.8. The value predicted from the exact analytic solution is $16\,\pi\,M^2 \simeq 50.3\,M^2$.

Chapter 6

Conclusions

In this work, we have presented our implementation of the dynamic singularity excision algorithm. With an eye toward future simulations that will treat the binary-black-hole problem, evolving those dynamic spacetimes until late times, we have focused on developing a simple yet powerful method for allowing black holes (and their associated excision regions) to move through the computational domain. We feel that this is in some sense the most intuitive approach to handling dynamic single- and multiple- black-hole spacetimes. It speaks to the success of the method that the excision algorithms have been among the least modified parts of the Maya code since this author wrote them several years ago.

The results presented herein have demonstrated several key developments in the evolution of dynamic spacetimes using the technique of singularity excision. First and foremost, the impact of using a prescribed *densitized* lapse versus a prescribed lapse is amazingly evident in the comparison between Fig. 5.2 and Fig. 5.3. In our earlier work [66], we were aware that there was evidence that choosing a fixed lapse and shift might not allow for long-term stable evolutions. Again, faced with few alternatives and not savoring the prospect of spending months concocting and testing new gauge conditions, a fixed gauge condition seemed to be the best route to study our dynamic excision algorithm.

With an analytical explanation for the fixed gauge's failure coming in the form of the work by [48] and numerical experiments coming from [50] (in a different formulation) demonstrating that simulations using a densitized lapse showed improvement, we implemented the algebraic gauge condition Eq. (2.32) and found substantial improvements immediately.

As the use of a densitized lapse (or a generalized version thereof) is a necessary ingredient in the majority of hyperbolic formulations, it has naturally been used in those codes in place of the lapse function itself. We have bridged the gap between hyperbolic and BSSN-like formulations, by evolving a stationary single black hole spacetime using the BSSN formulation in combination with a densitized version of the lapse function. We have concretely demonstrated in a set of controlled experiments that by employing an analytically-prescribed *densitized* lapse as opposed to analytically specifying the lapse itself, one can drastically extend the lifetime of Kerr-Schild type black-hole simulations (at least in the zero-angular-momentum regimes we have tested).

Second, we have demonstrated that singularity excision need not be an overly com-

plicated technique to be useful in black-hole simulations. While the treatment here has attempted to be detailed and didactic in its presentation of the algorithm's details, the relative simplicity of the algorithm is one of its strongest points. At the end of all the technical details, the focus is clear: we wish to avoid computing derivatives numerically for some region inside a black hole. This creates a boundary on our computational domain. In order to provide data on the boundary, we simply extrapolate from the computational interior. By doing this in a clever way, we are able to make the method fast, while at the same time amazingly flexible, so that it can handle a broad range of problems, including the ability to handle excision regions that do not remain at fixed coordinate locations.

In many ways, the reason general relativity is such a difficult theory to treat numerically is that it gives us too much freedom. We are allowed to make gauge choices, to change the formulation by introducing new combinations of variables, to decide to enforce the constraints after every time-step or only once. Each new choice simply enlarges our already voluminous parameter space of possible implementations. It is the author's firm belief that the success of the Maya code can be attributed to its readability and simple, elegant design. The dynamic singularity excision algorithm was also designed with these virtues in mind. By employing a very simple technique (polynomial extrapolations in one dimension), we have been able to evolve a dynamic black-hole spacetime in which the black hole is constantly moving for run-times longer than anyone else in the field can report (compare with [44]).

In looking to the future, we must admit that the applications for the dynamic singularity excision algorithm presented in the previous chapter are a strong test of the method, but a very weak probe of interesting physics. With the method itself now well-established, we are in the process of using it to its full potential in the study of binary black-hole collision (to be described in [46]). Also, the details of this algorithm are currently being ported to the bam code which recently demonstrated a simulation of a binary black-hole orbit for just over the time corresponding to a full orbital period [27]. It is expected that the dynamic excision algorithm, when implemented in that code, may offer more flexibility (as bam currently requires that excision regions remain fixed for all times).

Of course, much work still remains to be done in terms of bettering the next generation of numerical relativity codes. Tedious as they may be, careful *empirical* investigation of boundary conditions must be performed at some point, as these seem to be the greatest single problem facing the community at large. The available literature for suggestions of plausible boundary conditions continues to expand, yet, as we have clearly shown with regards to the densitized lapse, seeing the evidence in practice makes all the difference.

We hope that this work, in the form of the dynamic excision algorithm itself, and the related testbeds, presented within and in our two publications, might similarly be found to be useful to other numerical relativity groups.

Appendix A

Complete derivation of the 3+1 decomposition

Here, separate from the main text, we present a fairly detailed presentation of the derivation of the 3+1 decomposition of Einstein's equations, with an intentional bias toward applications in numerical relativity. This presentation will thematically follow that of York [80], but will present the intermediate calculations for clarity and will slightly differ in notation and viewpoint. In constructing this presentation, the review article by Baumgarte & Shapiro [14], the notes by Choptuik [33], and a series of lectures given by Abhay Ashtekar in an advanced general relativity course at Penn State in the spring of 2004 were extremely helpful. Researchers exploring numerical relativity for the first time may find it helpful to refer to these notes as they follow the presentation in [80], but are encouraged to work through the derivations themselves to gain understanding of the material.

In what follows, it may be easy to get distracted by the many calculations and definitions that arise, but it should be kept in mind that our ultimate goal is simple: much as in Newtonian particle mechanics, if we know the configuration of a system at an instant of time and the rules which govern all interactions (equations of motion), then we can determine the evolution of the system in time into the future or the past as we see fit. Our goal with general relativity is then the same, to describe *space* at an "instant in time", how that space "evolves in time", and in doing so, construct a spacetime.

As an overview of the sections to follow, we will begin our derivation of the initial-value or 3+1 formulation of general relativity by assuming, at least initially, that we have the full spacetime, $(\mathcal{M}, g_{ab})$, along with the metric-compatible connection ∇, and see what geometry this induces in a one-parameter family of spatial slices. We will work for some time to collapse all of Einstein's equations into equations that are true within each slice (constraint equations) and equations that reach from one slice to the next (evolution equations). Once we have derived these relations, we change our focus. We imagine the situation in which we know the metric for a three-dimensional spatial slice, and this metric's "time derivative", (satisfying certain constraints) and we ask how we can evolve this space in time according to Einstein's equations so as to construct the full spacetime in which our initial spatial slice sits.

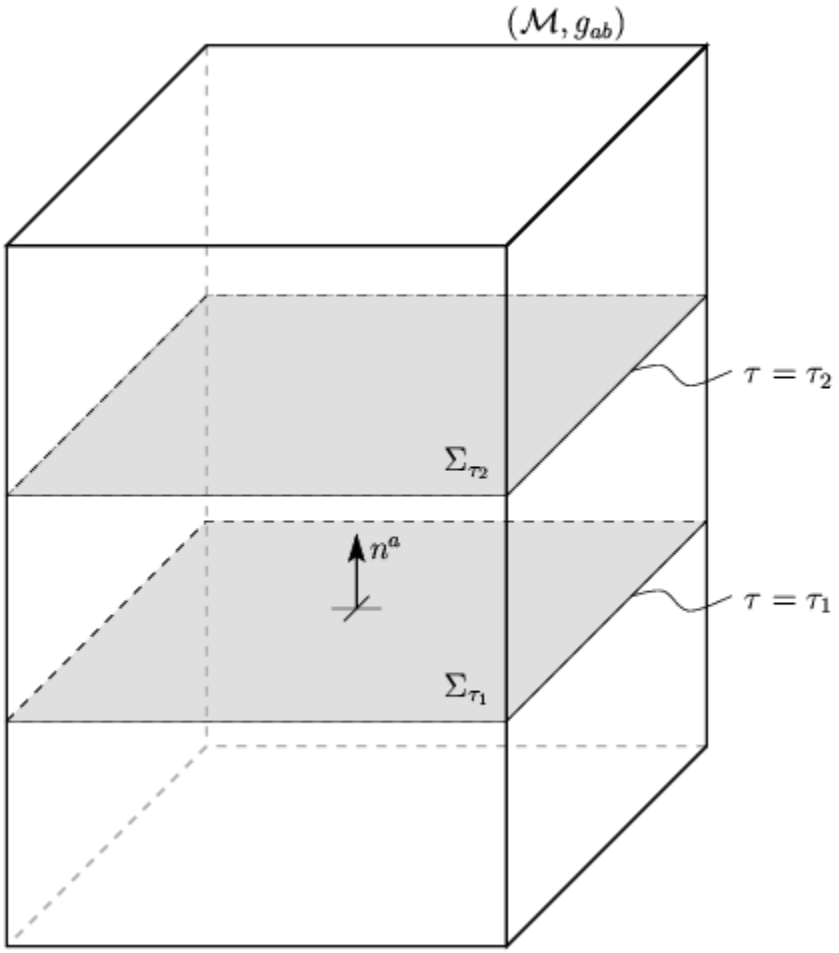


Figure A.1: Foliation of spacetime ($\mathcal{M}$, g_{ab}) as surfaces of constant time-parameter τ.

A.1 Slices of spacetime and their normals

From the very beginning of our discussion, we make the assumption that the manifold $\mathcal{M}$ of our spacetime has the topology of $\Sigma \times \mathbb{R}$, where each hypersurface, or slice, Σ will be a three-dimensional *spacelike* submanifold of $\mathcal{M}$ and each slice Σ will be labeled or identified by its own unique parameter $\tau \in \mathbb{R}$. Notationally, we will refer to a particular slice with parameter τ_0 as Σ_{τ_0}. Moreover, we see that this topology means that τ is a scalar field which exists everywhere in $\mathcal{M}$, *i.e.*, it is a function on spacetime: $\tau : \mathcal{M} \to \mathbb{R}$. This gives rise to the picture of spacetime depicted in Fig. A.1.

We could have begun more generally than this by not assuming the topology above, but in numerical relativity, we are only interested in constructing globally hyperbolic spacetimes, *i.e.*, spacetimes for which an initial-value formulation is applicable. Global hyperbolicity implies two features of direct importance to us. The first is that the topology of $\mathcal{M}$ is

$\Sigma \times \mathbb{R}$, as we have already assumed. The second is that there exists a global time function (what we're calling τ) such that each surface of constant τ is a **Cauchy surface**. Here we will adopt York's way of defining a Cauchy surface by saying that it is a submanifold of spacetime such that every causal curve (curves whose tangent vectors are timelike or null everywhere) intersects it exactly once.

What one is trying to hit upon with the descriptions above is a rigorous definition of the concept of predictability. In other words, we simply want to say that, given the conditions present at one "instant of time", represented by a Cauchy surface Σ, we should be able to determine the entire future (or past) history of spacetime. For the case of general relativity, many things can come along and ruin this seemingly elementary expectation, and so detailed analyses of global hyperbolicity and its associated consequences are quite necessary. We will not spend further time on this issue, comfortable in the assumptions we have made above, but for a more detailed look at these considerations, the interested reader is directed to chapter eight of Wald [77].

Given the arguments above and our global time function τ, each slice will arise as a surface of constant τ. Geometrically, we describe this by constructing a one-form field

$$\boldsymbol{\Omega} = \mathbf{d}\tau \text{ (equivalently, } \Omega_a = \nabla_a \tau). \tag{A.1}$$

This one-form defines our hypersurfaces in the sense that it singles out vector fields (and arbitrary tensor fields in general) tangent to Σ. Any vector field v^a tangent to Σ will satisfy

$$\boldsymbol{\Omega}(\boldsymbol{v}) = v^a \Omega_a = v^a \nabla_a \tau = 0, \tag{A.2}$$

because τ equals a constant within Σ, and thus all derivatives of τ *within* Σ will be zero.

We will now go about constructing a unit vector field normal to the surface, as this will turn out to be more useful to us in the work to follow. We wish our hypersurfaces to be spacelike, thus their normal must be timelike. (In fact, by assuming global hyperbolicity, we have guaranteed that there is such a universal time function whose gradient is everywhere timelike.) The norm of our one-form is $\|\boldsymbol{\Omega}\|^2 = g^{ab}\Omega_a\Omega_b < 0$. We then, for reasons which will hopefully become clear later, define a strictly positive scalar quantity, which we call the **lapse**, α, by

$$\alpha^2 \equiv \frac{-1}{\|\boldsymbol{\Omega}\|^2} = \frac{-1}{g^{ab}\Omega_a\Omega_b}. \tag{A.3}$$

We then scale our one-form $\boldsymbol{\Omega}$ by this value of α to construct a new unit-norm one-form

$\omega_a \equiv \alpha\Omega_a$. We confirm that

$$\|\boldsymbol{\omega}\|^2 = g^{ab}\omega_a\omega_b = \alpha^2 g^{ab}\Omega_a\Omega_b = -1. \tag{A.4}$$

Finally, then, we define our timelike unit vector normal to Σ by

$$n^a \equiv -g^{ab}\omega_b = -\alpha g^{ab}\Omega_b. \tag{A.5}$$

The convention of having a minus sign in this definition is to ensure that n^a is future-pointing (*i.e.*, points in the direction of increasing τ). We quickly verify then that this vector field has the requisite properties of being

1. timelike with unit norm: $n^a n_a = g^{ab}n_a n_b = g^{ab}(-\omega_a)(-\omega_b) = -1$, and
2. normal to Σ: For any v^a tangent to Σ, $g_{ab}v^a n^b = -v^a\omega_a = -\alpha v^a\nabla_a\tau = 0$.

Now that we have this vector field in hand, we are ready to use it to preferentially select or construct arbitrary tensor fields that are entirely tangent to Σ. Before we proceed, though, we pause briefly to consider a few properties of n^a that we will make extensive use of in later analyses. A result which we will use quite often is that

$$\begin{aligned} n_b\nabla_a n^b &= \nabla_a(n_b n^b) - n^b\nabla_a n_b \\ &= \nabla_a(-1) - n^b\nabla_a n_b \\ &= -n^b\nabla_a n_b \\ &= -n_b\nabla_a n^b, \end{aligned}$$

which must then imply that

$$n_b\nabla_a n^b = n^b\nabla_a n_b = 0. \tag{A.6}$$

We also note that we can interpret the vector field n^a as the four-velocity of a congruence of observers moving orthogonal to the hypersurfaces Σ. They are not necessarily geodesic observers, and will, in general, have a non-vanishing four-acceleration

$$a^b = n^a\nabla_a n^b. \tag{A.7}$$

With the use of (A.6), we can see that this four-acceleration is orthogonal to the four-velocity (and hence tangent to Σ) by observing that

$$n_b a^b = n_b n^c\nabla_c n^b = n^c\left(n_b\nabla_c n^b\right) = 0. \tag{A.8}$$

A.2 Projection and induced metric

Let us now focus our attention on some particular, but arbitrary, slice $\Sigma_{\tau_0} \in \mathcal{M}$ defined by the condition $\tau = \tau_0$, a constant. The points on this slice form a three-dimensional submanifold of $\mathcal{M}$. We would like to know what geometry is induced on Σ_{τ_0} by its embedding in $\mathcal{M}$. We know from the previous section that, at each point on this hypersurface, there exists a timelike vector field n^a that is everywhere orthogonal to it. We can use this to decompose four-dimensional spacetime tensors into hypersurface-tangential (spatial) and hypersurface-orthogonal (temporal) pieces. Just to be perfectly clear, a tensor is spatial if every free index, when contracted with n^a is identically zero, *i.e.*,

$$T^{abc}{}_{de}n_a = T^{abc}{}_{de}n_b = T^{abc}{}_{de}n_c = T^{abc}{}_{de}n^d = T^{abc}{}_{de}n^e = 0.$$

In particular, we will find it useful to project our spacetime tensors down into our three-dimensional spacelike surfaces, and once there, work entirely in terms of these spatial quantities, all defined at one "instant of time", or one value of the parameter τ.

To perform this feat of magic, we define the **projection tensor** by

$$\perp^a{}_b \equiv g^a{}_b + n^a n_b = \delta^a{}_b + n^a n_b. \tag{A.9}$$

For any spatial vector v^a tangent to Σ_{τ_0} and any temporal vector $\ell^a = fn^a$ orthogonal to it, this projection tensor has properties described below.

1. It is idempotent, *i.e.*, acting with the projection tensor multiple times is the same as acting once:
   $\perp^a{}_b \perp^b{}_c = (\delta^a{}_b + n^a n_b)(\delta^b{}_c + n^b n_c) = \delta^a{}_c + n^a n_c + n^a n_c + (n^b n_b) n^a n_c = \perp^a{}_c.$

2. It annihilates temporal vectors:
   $\perp^a{}_b \ell^b = f(\delta^a{}_b + n^a n_b) n^b = f(n^a + (n_b n^b) n^a) = 0.$

3. It acts as the identity on spatial vectors:
   $\perp^a{}_b v^b = (\delta^a{}_b + n^a n_b) v^b = v^a + (n_b v^b) n^a = v^a.$

This last point will be useful because it means that we can apply the projection tensor to any spatial tensor without effect (if it should prove helpful in shortening a calculation). In any case, to find the spatial tensor corresponding to a given spacetime tensor, we simply apply the projection tensor on every free index. For instance, we can define a spatial tensor $T_{ab}{}^c$ to be the completely spatial projection of a spacetime tensor $S_{ab}{}^c$ by

$$T_{ab}{}^c = \perp^p{}_a \perp^q{}_b \perp^c{}_r S_{pq}{}^r. \tag{A.10}$$

Note that this operation is sometimes denoted in the literature by $T_{ab}{}^{c} = \perp S_{ab}{}^{c}$, where the symbol $\perp$ without indices is a shorthand for applying the projection operator on every free index to its right with the appropriate dummy indices.

Projecting the spacetime metric itself gives the spatial metric on the hypersurface, given by

$$\begin{aligned} \gamma_{ab} &\equiv \perp^{c}{}_{a} \perp^{d}{}_{b} g_{cd} && \text{(A.11)} \\ &= (\delta^{c}{}_{a} + n^{c} n_{a})(\delta^{d}{}_{b} + n^{d} n_{b}) g_{cd} \\ &= (\delta^{c}{}_{a} + n^{c} n_{a})(g_{cb} + n_{c} n_{b}) \\ &= g_{ab} + n_{a} n_{b} + n_{b} n_{a} + (n^{c} n_{c}) n_{a} n_{b} \\ &= g_{ab} + n_{a} n_{b}. && \text{(A.12)} \end{aligned}$$

A similar calculation shows that the "inverse" spatial metric is given by

$$\gamma^{ab} = g^{ab} + n^{a} n^{b}. \tag{A.13}$$

We see that what we have been calling the projection tensor is just the mixed form of the spatial metric,

$$\gamma^{a}{}_{b} = \perp^{a}{}_{b} = \delta^{a}{}_{b} + n^{a} n_{b}. \tag{A.14}$$

Henceforth, we will dispense with the separate notation, and simply use the mixed form of the spatial metric $\gamma^{a}{}_{b}$ to perform all projections.

It is very important to realize that all tensor indices will continue to be raised and lowered by means of the spacetime metric g_{ab} and its inverse, but for purely spatial tensors, the spatial metric γ_{ab} can equivalently be used. Also, note that the spatial metric and the "inverse" spatial metric are not true inverses of one another, but again behave as such when acting on purely spatial tensors.

A.3 Spatial connection and Riemann tensor

Using the tools thus far, we have defined a hypersurface Σ_{τ_0}, which is a submanifold of $\mathcal{M}$, and have determined the spatial metric γ_{ab} induced on this hypersurface. In terms of differential geometry, then, the only additional structure that we might immediately feel is lacking is a metric-compatible connection.

Along these lines, we *define* the spatial covariant derivative operator D to act on spatial one-forms (ρ_a, where $\rho_a n^a = 0$) by

$$D_a \rho_b = \gamma^{c}{}_{a} \gamma^{d}{}_{b} \nabla_c \rho_d. \tag{A.15}$$

This is obviously a spatial tensor because all free indices are projected into Σ by the projection tensor.

We can then easily extend our definition in (A.15) to spatial tensors of any type by ensuring that the linearity and the Leibniz rules for derivative operators hold. For example, for some arbitrary, purely spatial tensor $T_{ab}{}^{cd}$, the spatial covariant derivative would take the form

$$D_e T_{ab}{}^{cd} = \gamma^t{}_e \gamma^p{}_a \gamma^q{}_b \gamma^c{}_r \gamma^d{}_s \nabla_t T_{pq}{}^{rs}. \tag{A.16}$$

Thus far, this connection is as good as any other, but we wish to have *the* metric-compatible connection. As it turns out, (A.15) is metric compatible, and is therefore the uniquely-defined connection we sought to find. We demonstrate this by applying the definition to our spatial metric (A.12) and see that

$$\begin{aligned} D_c \gamma_{ab} &= \gamma^r{}_c \gamma^p{}_a \gamma^q{}_b \nabla_r \gamma_{pq} \\ &= \gamma^r{}_c \gamma^p{}_a \gamma^q{}_b \nabla_r \left(g_{pq} + n_p n_q \right) \\ &= \gamma^r{}_c \gamma^p{}_a \gamma^q{}_b \nabla_r n_p n_q \\ &= \gamma^r{}_c \gamma^p{}_a \gamma^q{}_b \left(n_p \nabla_r n_q + n_q \nabla_r n_p \right) \\ &= 0. \end{aligned} \tag{A.17}$$

In the calculation above, the full metric vanishes on the third line because ∇ is spacetime-metric compatible, and on the second-to-last line, the term in parentheses vanishes because each of the un-differentiated n_a terms is annihilated by a projection tensor. By similar calculations one can show that the D operator is compatible with the contravariant spatial metric γ^{ab} and the mixed-form spatial metric (projection tensor) $\gamma^a{}_b$. Thus, we have singled out the unique, spatial, metric-compatible covariant derivative operator, and we now have on our submanifold all the structure we would typically ask for: a metric and its compatible connection.

It also bears mentioning, very briefly, that this derivative operator is, by construction, compatible with the spacetime metric, as seen from the fact that

$$D_c g_{ab} = \gamma^r{}_c \gamma^p{}_a \gamma^q{}_b \nabla_r g_{pq} = 0. \tag{A.18}$$

At this point, we are in a position to ask what the intrinsic curvature of Σ_{τ_0} must be. For a spatial one-form ρ_a ($\rho_a n^a = 0$), the three-dimensional curvature tensor is simply given by the usual formula

$$\mathsf{R}_{abc}{}^d \rho_d = 2 D_{[a} D_{b]} \rho_c. \tag{A.19}$$

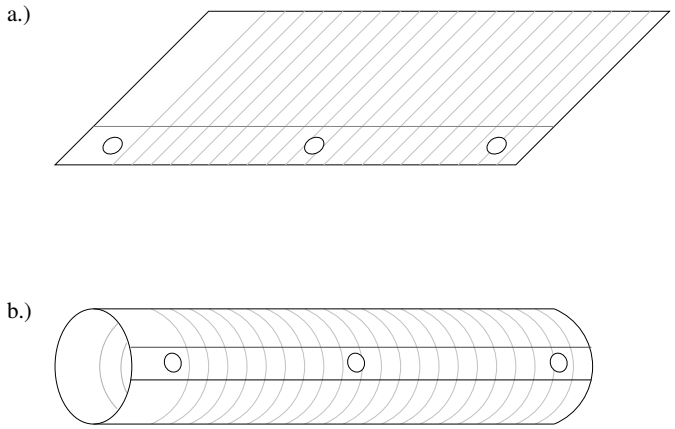


Figure A.2: Illustration of the role of extrinsic curvature. In a.) we see a piece of paper, or a plane $z = \text{const.}$ in $\mathbb{R}^3$. It has no curvature whatsoever. In b.) we see the same paper rolled around until its ends meet. The geometry intrinsic to the sheet has not changed but its new embedding in $\mathbb{R}^3$ has endowed it with an *extrinsic* curvature.

We have introduced the notation that the three-dimensional curvature tensor and its contractions will be denoted with a sans-serif 'R' to differentiate it from the four-dimensional object denoted with a regular italic 'R'. This Riemann tensor has all the normal symmetries, and the last index can be raised or lowered with either the full spacetime metric or the spatial metric (because this Riemann tensor, $\mathsf{R}_{abc}{}^{d}$, is a purely spatial tensor). The Ricci tensor and scalar are defined as usual by

$$\mathsf{R}_{ab} = \mathsf{R}_{acb}{}^{c}, \text{ and} \tag{A.20}$$

$$\mathsf{R} = \gamma^{ab}\mathsf{R}_{ab}. \tag{A.21}$$

A.4 Extrinsic curvature

With the addition of a metric-compatible connection in the previous section, we have a firm handle on the intrinsic geometry of our slices. Now we will address the issue of how our slice fits into the big picture, or more precisely, how it is embedded into the larger manifold of spacetime.

Before we do that, though, we would like to motivate the introduction of what is known as the extrinsic curvature tensor with a more pedestrian example. Imagine a sheet of paper, or a two-dimensional plane in $\mathbb{R}^3$. By rolling this sheet of paper into a cylinder, one has not changed the geometry intrinsic to the paper (see Fig. A.2). Parallel lines remain parallel, the angles in a triangle still add up to 180°, and so on. So, the intrinsic curvature (referring to the Riemann tensor, for example) must not have changed (still zero in our simple Euclidean example). Yet, we would obviously say that this cylinder is curved in some sense. This

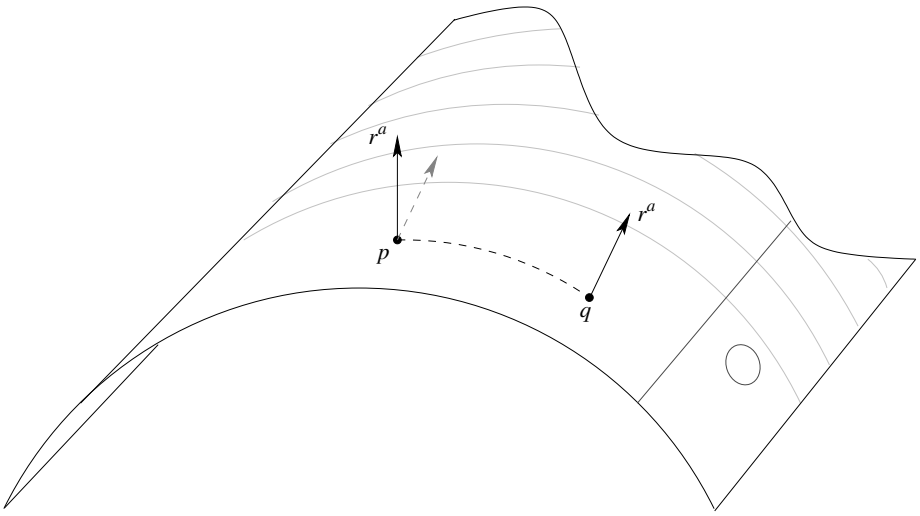


Figure A.3: Motivation for the definition of the extrinsic curvature tensor using the rolled paper cylinder example. The dashed vector is the parallel-transported copy of the normal (here a spacelike vector r^a) from point q to p along the dashed path. After parallel transport, this vector fails to coincide with the normal vector residing at p. The extrinsic curvature tensor then is a measure of how this normal vector changes along the surface Σ, with that difference then projected back into the surface.

refers to its **extrinsic curvature**, how the sheet of paper is embedded in $\mathbb{R}^3$. We quantify this curvature by looking at how the normal vectors to the surface change along the surface. In particular, we use the derivative operator of the larger space (here, $\mathbb{R}^3$) to measure the changes in the normal vector field, and then project back into the surface (see Fig. A.3). This is exactly how we will define our extrinsic curvature tensor by

$$K_{ab} \equiv -\gamma^p{}_a \gamma^q{}_b \nabla_p n_q. \tag{A.22}$$

The minus sign, strangely, is a convention common among numerical relativists, perhaps originating with York [80], and so we will adopt it here as well.

We will now proceed to examine this tensor and derive several useful formulae relating it to other quantities we have already discussed. First of all, note that it is a purely spatial tensor by construction. Next, we will see that it is symmetric. We do this by showing that its antisymmetric part vanishes, which is sufficient for a rank-two tensor. To do this, recall

that $n_a = -\omega_a = -\alpha\nabla_a\tau$, for then

$$\begin{aligned} K_{[ab]} &= -\gamma^p{}_{[a}\gamma^q{}_{b]}\nabla_p n_q \\ &= \gamma^{[p}{}_a\gamma^{q]}{}_b\nabla_p(\alpha\nabla_q\tau) \\ &= \gamma^p{}_a\gamma^q{}_b\nabla_{[p}(\alpha\nabla_{q]}\tau) \\ &= \gamma^p{}_a\gamma^q{}_b\left((\nabla_{[p}\alpha)(\nabla_{q]}\tau) + \alpha\nabla_{[p}\nabla_{q]}\tau\right) \\ &= 0, \end{aligned} \tag{A.23}$$

where in the second-to-last line, the first term in the sum vanishes because on a given slice, τ is constant and so the projection tensor annihilates $\nabla_a\tau$, while the second term in the sum vanishes because $\nabla_{[a}\nabla_{b]}f = 0$, for any scalar field f and a torsion-free connection (which we take for granted in general relativity). Thus we see that

$$K_{ab} = K_{(ab)} = -\gamma^p{}_a\gamma^q{}_b\nabla_{(p}n_{q)}. \tag{A.24}$$

A useful alternative expression can be found by expressing K_{ab} in terms of n_a, its derivative, and a_a, the four-acceleration defined in (A.7) by

$$\begin{aligned} K_{ab} &= -\gamma^p{}_a\gamma^q{}_b\nabla_p n_q \\ &= -(\delta^p{}_a + n^p n_a)(\delta^q{}_b + n^q n_b)\nabla_p n_q \\ &= -\left(\nabla_a n_b + n_b n^q\nabla_a n_q + n_a n^p\nabla_p n_b + n_a n_b n^p n^q\nabla_p n_q\right) \\ &= -\left(\nabla_a n_b + n_a n^p\nabla_p n_b\right) \\ &= -\left(\nabla_a n_b + n_a a_b\right), \end{aligned} \tag{A.25}$$

where we have used (A.6) on the fourth line. The symmetries of (A.24) can also be applied to this expression.

Finally, we present a much more powerful expression for the extrinsic curvature which will bring forward its status as the "time-derivative" of the spatial metric. We look at the Lie derivative of γ_{ab} along n^a, given by

$$\begin{aligned} \pounds_{\boldsymbol{n}}\gamma_{ab} &= n^c\nabla_c\gamma_{ab} + \gamma_{cb}\nabla_a n^c + \gamma_{ac}\nabla_b n^c \\ &= n^c\nabla_c(g_{ab} + n_a n_b) + (g_{cb} + n_c n_b)\nabla_a n^c + (g_{ac} + n_a n_c)\nabla_b n^c \\ &= n^c\nabla_c(n_a n_b) + \nabla_a n_b + n_b n_c\nabla_a n^c + \nabla_b n_a + n_a n_c\nabla_b n^c \\ &= n_a n^c\nabla_c n_b + n_b n^c\nabla_c n_a + \nabla_a n_b + \nabla_b n_a, \end{aligned}$$

where we have again used (A.6) on the last line. Recalling the definition of the four-

acceleration from (A.7), we have, then, that

$$\begin{aligned}\pounds_{\boldsymbol{n}}\gamma_{ab} &= n_a a_b + n_b a_a + \nabla_a n_b + \nabla_b n_a \\ &= 2\left(n_{(a}a_{b)} + \nabla_{(a}n_{b)}\right) = -2K_{(ab)},\end{aligned}$$

and so we find

$$K_{ab} = -\frac{1}{2}\pounds_{\boldsymbol{n}}\gamma_{ab}. \tag{A.26}$$

Beyond this, we can also interpret the extrinsic curvature as being the spatially-projected Lie-derivative along n^a of the spacetime metric,

$$\begin{aligned}\gamma^p{}_a\gamma^q{}_b\pounds_{\boldsymbol{n}}g_{pq} &= \gamma^p{}_a\gamma^q{}_b\left(n^r\nabla_r g_{pq} + g_{rq}\nabla_p n^r + g_{pr}\nabla_q n^r\right) \\ &= \gamma^p{}_a\gamma^q{}_b\left(\nabla_p n_q + \nabla_q n_p\right) \\ &= 2\gamma^p{}_a\gamma^q{}_b\left(\nabla_{(p}n_{q)}\right) \\ &= -2K_{ab},\end{aligned}$$

and so

$$K_{ab} = -\frac{1}{2}\gamma^p{}_a\gamma^q{}_b\pounds_{\boldsymbol{n}}g_{pq}. \tag{A.27}$$

While we originally motivated the definition of the extrinsic curvature tensor (A.22) as providing information about how our slice is embedded in the larger spacetime, seeing K_{ab} written in terms of this Lie derivative of the spatial metric as above (A.26) really drives home its interpretation as the time-derivative of γ_{ab}. The Lie derivative, of course, tells us how a tensor changes by an infinitesimal displacement along the integral curves of the vector field, which in this case we take to be our normal vector field n^a. In §A.1, we saw that n^a is a future-pointing timelike vector field normal to slices of constant time τ. Thus, we see why we refer to the extrinsic curvature as the time-derivative of the metric.

A.5 Projections of the Riemann tensor

With all of the groundwork laid, we are now in a position to ask how we can relate the four-dimensional spacetime curvature to quantities we can express at a given instant of time. One can imagine, from the arguments in the previous sections, that the curvature of spacetime will manifest itself in the intrinsic curvature of the spatial slices (the three-dimensional Riemann $\mathsf{R}_{abc}{}^d$ and its contractions) and the extrinsic curvature relating how the slices are embedded in $\mathcal{M}$. These relationships are precisely what we will derive here.

A.5.1 Gauss' equation

We start by looking at the spatial Riemann tensor acting on a spatial one-form, ρ_a, which is just

$$\mathsf{R}_{abc}{}^{d}\rho_d = 2D_{[a}D_{b]}\rho_c.$$

In order to connect this with the four-dimensional Riemann $R_{abc}{}^{d}$, we will begin to expand out the spatial covariant derivative D_a as projections of the spacetime covariant derivative ∇_a, using our definition (A.15). Plugging this into the above expression, we have

$$\begin{aligned}\mathsf{R}_{abc}{}^{d}\rho_d &= 2D_{[a}\big(\gamma^{p}{}_{b]}\gamma^{q}{}_{c}\nabla_p\rho_q\big) \\ &= 2\gamma^{r}{}_{[a}\gamma^{s}{}_{b]}\gamma^{t}{}_{c}\nabla_r\big(\underbrace{\gamma^{p}{}_{s}}_{\text{(I)}}\underbrace{\gamma^{q}{}_{t}}_{\text{(II)}}\underbrace{\nabla_p\rho_q}_{\text{(III)}}\big).\end{aligned} \tag{A.28}$$

Now we have the expression in the desired form. All that remains is to work through the arithmetic. We see that ∇_r acting on everything to the right will produce three terms. We write this symbolically as

$$\mathsf{R}_{abc}{}^{d}\rho_d = \text{(I)} + \text{(II)} + \text{(III)},$$

where (I), (II), and (III) as denoted in (A.28) will indicate which piece is differentiated while the rest come out of the derivative. We will tackle each term separately. We find that the first term vanishes by

$$\begin{aligned}\text{(I)} &= 2\gamma^{r}{}_{[a}\gamma^{s}{}_{b]}\gamma^{t}{}_{c}\gamma^{q}{}_{t}\big(\nabla_r\gamma^{p}{}_{s}\big)\big(\nabla_p\rho_q\big) \\ &= 2\gamma^{r}{}_{[a}\gamma^{s}{}_{b]}\gamma^{q}{}_{c}\big(\nabla_r(\delta^{p}{}_{s} + n^p n_s)\big)\big(\nabla_p\rho_q\big) \\ &= 2\gamma^{r}{}_{[a}\gamma^{s}{}_{b]}\gamma^{q}{}_{c}\big(\nabla_r n^p n_s\big)\big(\nabla_p\rho_q\big) \\ &= 2\gamma^{r}{}_{[a}\gamma^{s}{}_{b]}\gamma^{q}{}_{c}\big(n^p\nabla_r n_s + n_s\nabla_r n^p\big)\big(\nabla_p\rho_q\big) \\ &= 0.\end{aligned} \tag{A.29}$$

From the next-to-last line, the first term in the sum vanishes because $\gamma^{r}{}_{[a}\gamma^{s}{}_{b]}\nabla_r n_s = 0$ by (A.23), and the second term vanishes because a projection tensor out front annihilates the un-differentiated n_s term.

We continue with the second piece of (A.28), finding

$$\begin{aligned}
\text{(II)} &= 2\gamma^{r}{}_{[a}\gamma^{s}{}_{b]}\gamma^{t}{}_{c}\gamma^{p}{}_{s}\big(\nabla_r\gamma^{q}{}_{t}\big)\big(\nabla_p\rho_q\big)\\
&= 2\gamma^{r}{}_{[a}\gamma^{p}{}_{b]}\gamma^{t}{}_{c}\big(\nabla_r(\delta^{q}{}_{t}+n^q n_t)\big)\big(\nabla_p\rho_q\big)\\
&= 2\gamma^{r}{}_{[a}\gamma^{p}{}_{b]}\gamma^{t}{}_{c}\big(\nabla_r(n^q n_t)\big)\big(\nabla_p\rho_q\big)\\
&= 2\gamma^{r}{}_{[a}\gamma^{p}{}_{b]}\gamma^{t}{}_{c}\big(n^q\nabla_r n_t + n_t\nabla_r n^q\big)\big(\nabla_p\rho_q\big)\\
&= 2\gamma^{r}{}_{[a}\gamma^{p}{}_{b]}\gamma^{t}{}_{c}\big(\nabla_r n_t\big)n^q\big(\nabla_p\rho_q\big),
\end{aligned}$$

where we have been able to drop one term again thanks to the projection tensor annihilating the free n_t. We are now going to use the fact that $\rho_a n^a = 0$ to move the derivative acting on ρ_q onto n^q, picking up a minus sign,

$$\begin{aligned}
&= -2\gamma^{r}{}_{[a}\gamma^{p}{}_{b]}\gamma^{t}{}_{c}\big(\nabla_r n_t\big)\big(\nabla_p n^q\big)\rho_q\\
&= -2\gamma^{r}{}_{[a}\gamma^{p}{}_{b]}\gamma^{t}{}_{c}\big(\nabla_r n_t\big)\big(\nabla_p n_q\big)\rho^q\\
&= -2K_{c[a}K_{b]q}\rho^q\\
&= -\big(K_{ca}K_b{}^{d} - K_{cb}K_a{}^{d}\big)\rho_d. \qquad \text{(A.30)}
\end{aligned}$$

Finally, we now look at the last piece of (A.28), which will yield the spacetime Riemann tensor,

$$\begin{aligned}
\text{(III)} &= 2\gamma^{r}{}_{[a}\gamma^{s}{}_{b]}\gamma^{t}{}_{c}\gamma^{p}{}_{s}\gamma^{q}{}_{t}\big(\nabla_r\nabla_p\rho_q\big)\\
&= 2\gamma^{r}{}_{[a}\gamma^{p}{}_{b]}\gamma^{q}{}_{c}\big(\nabla_r\nabla_p\rho_q\big)\\
&= 2\gamma^{[r}{}_{a}\gamma^{p]}{}_{b}\gamma^{q}{}_{c}\big(\nabla_r\nabla_p\rho_q\big)\\
&= 2\gamma^{r}{}_{a}\gamma^{p}{}_{b}\gamma^{q}{}_{c}\big(\nabla_{[r}\nabla_{p]}\rho_q\big)\\
&= \gamma^{r}{}_{a}\gamma^{p}{}_{b}\gamma^{q}{}_{c}R_{rpq}{}^{s}\rho_s\\
&= \gamma^{r}{}_{a}\gamma^{p}{}_{b}\gamma^{q}{}_{c}\gamma^{s}{}_{d}R_{rpq}{}^{s}\rho_d. \qquad \text{(A.31)}
\end{aligned}$$

On the last line, we have used the fact that ρ_s is spatial, hence hitting it with the projection operator has no affect, and thus we can use it to change the index from an s to a d, which will be useful once we reassemble the terms.

Putting together (A.28), (A.29), (A.30), and (A.31), we have

$$\begin{aligned}
\mathsf{R}_{abc}{}^{d}\rho_d &= \Big(0\Big) - \Big((K_{ca}K_b{}^{d} - K_{cb}K_a{}^{d})\rho_d\Big) + \Big(\gamma^{p}{}_{a}\gamma^{q}{}_{b}\gamma^{r}{}_{c}\gamma^{s}{}_{d}R_{pqr}{}^{s}\rho_d\Big)\\
&= \Big(-\big(K_{ca}K_b{}^{d} - K_{cb}K_a{}^{d}\big) + \gamma^{p}{}_{a}\gamma^{q}{}_{b}\gamma^{r}{}_{c}\gamma^{s}{}_{d}R_{pqr}{}^{s}\Big)\rho_d.
\end{aligned}$$

This yields the relation

$$\mathsf{R}_{abc}{}^{d} + K_{ca}K_b{}^{d} - K_{cb}K_a{}^{d} = \gamma^{p}{}_{a}\gamma^{q}{}_{b}\gamma^{r}{}_{c}\gamma^{d}{}_{s}R_{pqr}{}^{s}, \qquad \text{(A.32)}$$

which goes by the name of **Gauss' equation**.

We can see that what we have just done, effectively, is to completely project the spacetime Riemann tensor into our hypersurface Σ, *i.e.*, we've projected *every* index down. Our aim is still to determine what relations the four-dimensional geometry imposes on the spatial geometry, especially in terms of the curvature. In addition to the purely spatial projection of the spacetime Riemann we have just found, we can also ask how the contractions of the curvature tensor along n^a project down into Σ. Because of the symmetries associated with R_{abcd}, there are only two contractions that we can make, namely, $R_{abcd}n^d$ and $R_{abcd}n^b n^d$. Any attempt to contract with one more n^a will vanish. So, we will now see what relations result from projecting these contracted objects.

A.5.2 Codazzi's equation

Projecting the once-contracted Riemann gives us

$$\begin{aligned}
\gamma^p{}_a\gamma^q{}_b\gamma^r{}_c R_{pqrd}n^d &= \gamma^p{}_a\gamma^q{}_b\gamma^r{}_c R_{pqr}{}^d n_d \\
&= 2\gamma^p{}_a\gamma^q{}_b\gamma^r{}_c \nabla_{[p}\nabla_{q]}n_r \\
&= 2\gamma^p{}_{[a}\gamma^q{}_{b]}\gamma^r{}_c \nabla_p\nabla_q n_r \\
&= 2\gamma^p{}_{[a}\gamma^q{}_{b]}\gamma^r{}_c \nabla_p(-K_{qr} - n_q a_r).
\end{aligned}$$

We have used (A.25) in the last line to simplify the calculation. We see that the projection operators acting on the first term just yield a spatially covariant derivative, *i.e.*,

$$\begin{aligned}
\gamma^p{}_a\gamma^q{}_b\gamma^r{}_c R_{pqrd}n^d &= -2\left(D_{[a}K_{b]c} + \gamma^p{}_{[a}\gamma^q{}_{b]}\gamma^r{}_c \nabla_p(n_q a_r)\right) \\
&= -2\left(D_{[a}K_{b]c} + \gamma^p{}_{[a}\gamma^q{}_{b]}\gamma^r{}_c (n_q\nabla_p a_r + a_r\nabla_p n_q)\right) \\
&= -2\left(D_{[a}K_{b]c} - 0 - a_c K_{[ab]}\right) \\
&= -2D_{[a}K_{b]c},
\end{aligned}$$

where on the third line, we have used the fact that the projection tensor will annihilate the un-differentiated n_q, and on the fourth line, we have used the fact that there is no antisymmetric part of K_{ab} (A.23). This projection then leads to the relation known as **Codazzi's equation**.

$$D_b K_{ac} - D_a K_{ac} = \gamma^p{}_a\gamma^q{}_b\gamma^r{}_c R_{pqrd}n^d. \tag{A.33}$$

A.5.3 Ricci's equation

Continuing with the main theme, we will project the twice-contracted Riemann tensor, but before we do that, we will need a couple of secondary results. We begin the first of two detours, aiming to show that for any spatial, completely covariant tensor $T_{a_1\ldots a_s}$, the Lie

derivative along n^a is also a spatial tensor. This means, namely, that

$$\gamma^{p_1}{}_{a_1}\gamma^{p_2}{}_{a_2}\cdots\gamma^{p_s}{}_{a_s}\pounds_{\boldsymbol{n}}T_{p_1\ldots p_s} = \pounds_{\boldsymbol{n}}T_{a_1\ldots a_s}. \tag{A.34}$$

In the next section, we will construct a vector field for which the Lie derivative along it will preserve spatiality for all spatial tensors, but for now, this result will have to suffice.

The argument will be quite simple: if the Lie derivative of this class of spatial tensors is itself spatial, then any and every index contracted with n^{a_t} will vanish for any $t \in 1,\ldots,s$. We perform this calculation, which yields

$$n^{a_t}\pounds_{\boldsymbol{n}}T_{a_1\ldots a_s} = n^{a_t}\left(n^c\nabla_c T_{a_1\ldots a_s} + T_{ca_2\ldots a_s}\nabla_{a_1}n^c + \cdots T_{a_1\ldots a_{s-1}c}\nabla_{a_s}n^c\right).$$

The tensor $T_{a_1\ldots a_s}$ is completely spatial by assumption, thus when n^{a_t} contracts with a matching index on any of the un-differentiated $T_{a_1\ldots a_s}$'s, the result will vanish. Obviously the only term which will not vanish is the one in which the t^{th} index has been replaced with c. Thus all terms but two vanish, namely

$$n^{a_t}\pounds_{\boldsymbol{n}}T_{a_1\ldots a_s} = n^{a_t}n^c\nabla_c T_{a_1\ldots a_s} + n^{a_t}T_{a_1\ldots c\ldots a_s}\nabla_{a_t}n^c.$$

Now, we use the fact that $n^{a_t}T_{a_1\ldots a_t\ldots a_s} = 0$ to swap the derivative from T onto n^a in the first term,

$$n^{a_t}\pounds_{\boldsymbol{n}}T_{a_1\ldots a_s} = -n^cT_{a_1\ldots a_j\ldots a_s}\nabla_c n^{a_j} + n^{a_t}T_{a_1\ldots c\ldots a_s}\nabla_{a_t}n^c.$$

But we can see, relabeling dummy indices, that the first and second terms are the same, and thus we see that the expression above vanishes. This is true for every $t \in 1,\ldots,s$ and thus the Lie derivative of a spatial, completely covariant tensor is also spatial, and so (A.34) is valid.

The second detour before we perform our last Riemann projection concerns the four-acceleration of our spatial "observers". Recall that the four-acceleration a^a is orthogonal to n^a ($a^bn_b = 0$), and thus the four-acceleration is a spatial tensor. What we show below is that we can relate the four-acceleration to the spatially-covariant derivative of the lapse. To simplify the calculation, we use the fact that applying the projection tensor to a spatial tensor is the same as applying the identity map,

$$\begin{aligned} a_b &= \gamma^c{}_b a_c = \gamma^c{}_b n^a\nabla_a n_c \\ &= -\gamma^c{}_b n^a\nabla_a\left(\alpha\nabla_c\tau\right), \end{aligned}$$

where we have recalled our earlier definitions (A.1), (A.5). Expanding, we see that

$$\begin{aligned} &= -\gamma^c{}_b n^a \Big(\alpha \nabla_a \nabla_c \tau + (\nabla_a \alpha)(\nabla_c \tau)\Big) \\ &= -\gamma^c{}_b n^a (\alpha \nabla_c \nabla_a \tau), \end{aligned}$$

where we have used used the fact that ∇ is torsion-free and thus derivatives of functions commute as well as the fact that the projection operator will annihilate $\nabla_c \tau$ because τ is constant in the slice. Using the Leibniz rule again, we find

$$\begin{aligned} &= -\gamma^c{}_b n^a \Big(\nabla_c (\alpha \nabla_a \tau) - (\nabla_c \alpha)(\nabla_a \tau)\Big) \\ &= -\gamma^c{}_b n^a \Big(\nabla_c n_a - (\nabla_c \alpha)\Omega_a\Big) \\ &= \gamma^c{}_b n^a (\nabla_c \alpha)\Omega_a \\ &= \gamma^c{}_b n^a (\nabla_c \alpha)(-\alpha^{-1} n_a) \\ &= -\gamma^c{}_b (n^a n_a)(\nabla_c \ln \alpha) \\ &= D_b (\ln \alpha), \end{aligned} \tag{A.35}$$

using (A.6) between the second and third lines to eliminate the first term in the difference. This expression nicely characterizes the four-acceleration as the spatial covariant derivative of the lapse.

Now, detours finally aside, we can perform the desired calculations and find what results from the projection of the twice-temporally-contracted Riemann tensor,

$$\begin{aligned} \gamma^p{}_a \gamma^q{}_b R_{pcqd} n^c n^d &= \gamma^p{}_a \gamma^q{}_b n^c R_{pcq}{}^d n_d \\ &= \gamma^p{}_a \gamma^q{}_b n^c (\nabla_p \nabla_c n_q - \nabla_c \nabla_p n_q) \\ &= \gamma^p{}_a \gamma^q{}_b n^c (\nabla_p (-K_{cq} - n_c a_q) - \nabla_c (-K_{pq} - n_p a_q), \end{aligned}$$

where we have again used (A.25) to replace the inner-most derivatives with the extrinsic curvature, normal vector, and its acceleration. Continuing, we find

$$\begin{aligned} &= \gamma^p{}_a \gamma^q{}_b (n^c \nabla_c K_{pq} + n^c \nabla_c (n_p a_q) - n^c \nabla_p K_{cq} - n^c \nabla_p (n_c a_q)) \\ &= \gamma^p{}_a \gamma^q{}_b (n^c \nabla_c K_{pq} + n^c \nabla_c (n_p a_q) + K_{cq} \nabla_p n^c - n^c \nabla_p (n_c a_q)), \end{aligned}$$

where we used the fact that $K_{cq} n^c = 0$ to move the derivative onto the n^c in the third term. Expanding out the compound terms using the Leibniz rule, we see that

$$\begin{aligned} \gamma^p{}_a \gamma^q{}_b R_{pcqd} n^c n^d &= \gamma^p{}_a \gamma^q{}_b \big(n^c \nabla_c K_{pq} + n^c n_p \nabla_c a_q + n^c a_q \nabla_c n_p \\ &\qquad + K_{cq} \nabla_p n^c - n^c n_c \nabla_p a_q - n^c a_q \nabla_p n_c\big) \\ &= \gamma^p{}_a \gamma^q{}_b (n^c \nabla_c K_{pq} + 0 + a_q a_p + K_{cq} \nabla_p n^c + \nabla_p a_q - 0). \end{aligned}$$

In the above, the first zero term comes about because the un-differentiated n_p is annihilated by a projection operator, and the final zero appears because of (A.6). Also, we used (A.7) to simplify the third term. Looking at this expression, we can see that we have two of the three pieces of $\pounds_{\boldsymbol{n}} K_{pq}$ when expanded out. To complete this expression, we simply add and subtract the missing piece of the Lie derivative simultaneously, yielding

$$\begin{aligned}
\gamma^p{}_a\gamma^q{}_b R_{pcqd}n^c n^d &= \gamma^p{}_a\gamma^q{}_b\big(n^c\nabla_c K_{pq} + K_{cq}\nabla_p n^c + (K_{pc}\nabla_q n^c - K_{pc}\nabla_q n^c) \\
&\qquad + a_q a_p + \nabla_p a_q\big) \\
&= \gamma^p{}_a\gamma^q{}_b\big(\pounds_{\boldsymbol{n}} K_{pq} - K_{pc}\nabla_q n^c + a_q a_p + \nabla_p a_q\big) \\
&= \gamma^p{}_a\gamma^q{}_b\big(\pounds_{\boldsymbol{n}} K_{pq} - K_{pr}\gamma^r{}_c\nabla_q n^c + a_q a_p + \nabla_p a_q\big).
\end{aligned}$$

In this last expression, we have again called upon the fact that the projection tensor acts as the identity map on spatial tensors (such as the extrinsic curvature), and so we can use it whenever it is useful to do so. Having this explicit, we now can let the projection tensors act on everything, and using the result of our first detour (A.34), and our definition of the extrinsic curvature (A.22) we get

$$\gamma^p{}_a\gamma^q{}_b R_{pcqd}n^c n^d = \pounds_{\boldsymbol{n}} K_{ab} + K_{ar}K_b{}^r + a_a a_b + D_a a_b. \tag{A.36}$$

We will use the result of our second detour (A.35) to simplify this even further. Looking at just the a_a pieces of (A.36), we have

$$\begin{aligned}
a_a a_b + D_a a_b &= D_a(\ln\alpha)D_b(\ln\alpha) + D_a\big(D_b(\ln\alpha)\big) \\
&= \alpha^{-2}(D_a\alpha)(D_b\alpha) + D_a\big(\alpha^{-1}(D_b\alpha)\big) \\
&= \alpha^{-2}(D_a\alpha)(D_b\alpha) + \alpha^{-1}D_aD_b\alpha - \alpha^{-2}(D_a\alpha)(D_b\alpha) \\
&= \alpha^{-1}D_aD_b\alpha.
\end{aligned} \tag{A.37}$$

Finally, combining this with our result (A.36), we arrive at

$$\gamma^p{}_a\gamma^q{}_b R_{pcqd}n^c n^d = \pounds_{\boldsymbol{n}} K_{ab} + K_{ac}K_b{}^c + \alpha^{-1}D_aD_b\alpha, \tag{A.38}$$

which is known as **Ricci's equation**.

A.6 From kinematics to dynamics

In the previous section, we derived some of the more computationally-intensive expressions which will be of immediate use to us in our attempts to construct an appropriate formalism for numerical relativity. We pause now to reflect upon what we have accomplished and what laws of physics we have used in our derivations.

The answer is, somewhat surprisingly, that no laws of physics have been used thus far. Everything to this point has been entirely derived without reference to the dynamical equations of the theory, namely Einstein's equations. The situation is analogous to that of an introductory mechanics course wherein one learns the relationship between the position, velocity, and acceleration of a particle, all without reference to, for example, what causes a body to accelerate. In this sense, everything to this point has been a study in **kinematics**, the relationships between our dynamical variables in the absence of any field equations.

As we discussed in §A.4, we can think of the Lie derivative along n^a as behaving like a time-derivative operator. Thinking of the spatial metric like a configuration variable (such as the position of a particle in Newtonian mechanics), we would then interpret (A.26) as telling us that the extrinsic curvature is the time-derivative of the spatial metric. It is, in a sense, our "velocity" variable. Ricci's equation (A.38) from the previous section then gives us an expression for the Lie derivative of the extrinsic curvature. Thus, pulling the Lie derivative to one side of the equation by itself, we effectively then have an expression for the "acceleration" of the spatial metric.

More than that, the relations derived in §A.1 – §A.5 are really just consequences of differential geometry, a particular choice of slices of our manifold, and the signature of our manifold's metric. To be fair, we have adopted a few application-specific conventions, such as the minus sign in our definition of the extrinsic curvature (A.22). In general, though, the Gauss-Codazzi-Ricci equations (A.32), (A.33), (A.38) are, up to a few sign flips, the general equations relating the curvature of a manifold to the intrinsic and extrinsic curvature of a submanifold with appropriately defined normal vector field and projection operator. These are relations that are true "no matter what".

We turn our attention now to the **dynamics** of the theory, namely the Einstein equations. Whereas the kinematics of the problem defines basic operating relationships between our "position", "velocity" and "acceleration" variables, one must provide an additional structure to explain what in nature "causes" the acceleration, or rules out certain positions, creating classes of allowed and un-allowed solutions. This is where one supplements the mathematical structure laid down with a "law of physics". In the case of elementary mechanics, this comes in the form of Newton's second law, $\mathbf{F} = m\mathbf{a}$. For the case under consideration here, the dynamics is given by $G_{ab} = R_{ab} - \frac{1}{2}Rg_{ab} = \kappa T_{ab}$, *i.e.*, Einstein's equations.

One should be mindful of the fact that it could be ultimately discovered one day that a given "law" is in fact incorrect, or as is more often the case, merely a limit of a more general law. These laws are postulated by scientists to explain existing phenomena and predict new features of nature. The only real constraint on a physical theory is that it not contradict experimental evidence (although sometimes the notion that a physical theory should possess "beauty" can act as a filter, favoring some theories over others). In any

case, we are perfectly content accepting Einstein's equations as the well-established theory of gravity, and so now we focus our attention on what these equations will mean in terms of the framework we have built in the previous sections.

A.7 The constraint equations

Einstein's equations tell us how matter and energy are coupled to the spacetime metric. Given this coupling, and all of our work thus far in projecting the geometry of spacetime into an "instant of time", we can expect that by combining the two, we are going to find a set of relations between spatial geometry and matter/stress-energy for a fixed instant in time.

In order to work out what these relations are, it stands to reason that we will need to try various contractions of the Einstein tensor with the normal vector field and/or the spatial metric. We will proceed along a slightly different path, working from our earlier results in terms of the Gauss-Codazzi-Ricci relations, and manipulating them until we get the desired factors of R_{ab} and R.

A.7.1 Hamiltonian constraint

To begin, let us contract Gauss' equation (A.32) twice with the spatial metric. The first contraction will yield an expression for the three-dimensional Ricci tensor,

$$\begin{aligned} \mathsf{R}_{abc}{}^{b} &= -K_{ca}K_{b}{}^{b} + K_{cb}K_{a}{}^{b} + \gamma^{p}{}_{a}\gamma^{q}{}_{b}\gamma^{r}{}_{c}\gamma^{b}{}_{s}R_{pqr}{}^{s} \\ \mathsf{R}_{ac} &= -K_{ca}K + K_{cb}K_{a}{}^{b} + \gamma^{p}{}_{a}\gamma^{q}{}_{s}\gamma^{r}{}_{c}R_{pqr}{}^{s} \end{aligned} \tag{A.39}$$

where we have used the usual notation of denoting the trace of the extrinsic curvature (and, in general, any tensor but the metric) as $K \equiv K_{a}{}^{a} = g^{ab}K_{ab}$.

Contracting this expression again will give us the three-Ricci scalar,

$$\begin{aligned} \gamma^{ac}\mathsf{R}_{ac} &= \gamma^{ac}\left(-K_{ca}K + K_{cb}K_{a}{}^{b} + \gamma^{p}{}_{a}\gamma^{q}{}_{s}\gamma^{r}{}_{c}R_{pqr}{}^{s}\right) \\ \mathsf{R} &= -K^{2} + K_{cb}K^{cb} + \gamma^{pc}\gamma^{q}{}_{s}\gamma^{r}{}_{c}R_{pqr}{}^{s} \\ &= -K^{2} + K_{ab}K^{ab} + \gamma^{pr}\gamma^{qs}R_{pqrs}. \end{aligned} \tag{A.40}$$

Looking at this expression above, we can see that the four-dimensional Riemann term on the right-hand side is contracted in just such a way as to look like a four-dimensional Ricci tensor, but the spatial-metric is not quite what we need to properly contract the indices. This is simply remedied by expanding the spatial metric in terms of the spacetime-metric

and the normal vector,

$$\begin{aligned}
\mathsf{R} + K^2 - K_{ab}K^{ab} &= \gamma^{pr}\gamma^{qs}R_{pqrs} \\
&= \gamma^{pr}(g^{qs} + n^q n^s)R_{pqrs} \\
&= \gamma^{pr}(R_{pr} + n^q n^s R_{pqrs}) \\
&= (g^{pr} + n^p n^r)(R_{pr} + n^q n^s R_{pqrs}) \\
&= R + n^q n^s R_{qs} + n^p n^r R_{pr} + n^p n^q n^r n^s R_{pqrs} \\
&= R + 2n^a n^b R_{ab},
\end{aligned}$$

where the contraction of the Riemann with the four copies of n^a vanishes because of the symmetries of the Riemann. We can massage this into a clearer form by multiplying the Ricci scalar by a curious way of expressing unity,

$$\begin{aligned}
\mathsf{R} + K^2 - K_{ab}K^{ab} &= (-1)(-1)R + 2n^a n^b R_{ab} \\
&= (-1)(g_{ab}n^a n^b)R + 2n^a n^b R_{ab} \\
&= 2\left(R_{ab} - \tfrac{1}{2}Rg_{ab}\right)n^a n^b,
\end{aligned}$$

from which we see we have

$$\mathsf{R} + K^2 - K_{ab}K^{ab} = 2G_{ab}n^a n^b. \tag{A.41}$$

Now, we use Einstein's equation to replace the Einstein tensor with the stress-energy tensor,

$$\begin{aligned}
\mathsf{R} + K^2 - K_{ab}K^{ab} &= 2\kappa T_{ab}n^a n^b \\
&= 2\kappa\rho
\end{aligned} \tag{A.42}$$

where we have defined

$$\rho \equiv T_{ab}n^a n^b \tag{A.43}$$

which is just the energy density that an observer with four-velocity n^a would measure.

Equation (A.42) is known as the **Hamiltonian constraint**, or sometimes, the scalar constraint. One can think of it as a "time-time" projection of Einstein's equations. Note that no explicit time-derivatives (Lie derivatives along n^a) exist anywhere in it, and so it must be satisfied at every point in time. This explains the use of term *constraint*. One can draw an analogy between the constraint equations here and the constraints of electromagnetism: $\boldsymbol{\nabla}\cdot\mathbf{E} = \rho_e$ and $\boldsymbol{\nabla}\cdot\mathbf{B} = 0$. This will also be true of the momentum constraint which we examine next.

A.7.2 Momentum constraint

To construct the momentum constraint, we will now contract Codazzi's equation (A.33) once with the spacetime metric, finding

$$\begin{aligned} g^{ac}\big(D_b K_{ac} - D_a K_{bc}\big) &= g^{ac}\big(\gamma^p{}_a \gamma^q{}_b \gamma^r{}_c n^d R_{pqrd}\big) \\ D_b K - D_a K_b{}^a &= \gamma^{pr}\gamma^q{}_b n^d R_{pqrd} \\ &= \gamma^q{}_b n^d (g^{pr} + n^p n^r) R_{pqrd} \\ &= \gamma^q{}_b (n^d R_{qd} + n^p n^r n^d R_{pqrd}) \\ &= \gamma^q{}_b n^d R_{qd} \\ &= \gamma^q{}_b n^d G_{qd}, \end{aligned} \tag{A.44}$$

where, as before, the symmetries of the Riemann tensor ensure that contraction of Riemann with $n^r n^d$ will cause it to vanish after the second-to-last line. The identity on the last step might not be obvious, but we can easily demonstrate that this contraction of the Ricci tensor is the same as the contraction of the Einstein tensor by

$$\begin{aligned} \gamma^q{}_b n^d G_{qd} &= \gamma^q{}_b n^d \left(R_{qd} - \tfrac{1}{2} g_{qd} R\right) \\ &= \gamma^q{}_b n^d R_{qd} - \tfrac{1}{2}\gamma^q{}_b n_q R \\ &= \gamma^q{}_b n^d R_{qd}. \end{aligned}$$

So, we see that our contraction of Codazzi's equation has then resulted in

$$\begin{aligned} D_a K - D_b K_a{}^b &= \gamma^c{}_a n^b G_{cb} \\ &= \kappa \gamma^c{}_a n^b T_{cb}. \end{aligned} \tag{A.45}$$

We define this projection of the stress-energy tensor as the momentum density,

$$j_a \equiv -\gamma^c{}_a n^b T_{cb}. \tag{A.46}$$

where the minus sign is a common convention in numerical relativity. We can again interpret this in terms of observers with four-velocity n^a as the mass/energy current density that these observers would measure.

Putting this all together, we then have the **momentum constraint**,

$$D_b K_a{}^b - D_a K = \kappa j_a. \tag{A.47}$$

Given that this equation, like the Hamiltonian constraint, does not have any time derivatives, it must be satisfied on each spatial slice. Satisfaction of (A.42) and (A.47) by (Σ, γ_{ab}, K_{ab}) is a necessary condition for Σ to be an embedded submanifold of spacetime ($\mathcal{M}$, g_{ab})

such that g_{ab} is a solution of Einstein's equations.

A.8 Time evolution

In the last section, we effectively examined how two different projections of the Einstein tensor are related to quantities defined in our spatial slices at an instant of time. These were the "time-time" and "time-space" projections of G_{ab}. Obviously, the only projection left is the "space-space" projection, or in other words, the completely spatial projection. We will see that this projection is what gives rise to our evolution equations, telling us how to take the spatial metric and extrinsic curvature from one moment in time to the next.

Before we proceed, though, we stop to note that our workhorse vector field n^a is not really the best suited for this task of time-differentiation. The issue lies in the fact that we engineered n^a to be a unit-norm vector field. This fact certainly came in quite handy throughout the calculations above because it caused many unwanted terms to vanish. But what we want is a vector field that will take us from one slice of constant time parameter τ_1 to another slice of constant time parameter τ_2, or better yet, in an infinitesimal sense, from τ_0 to $\tau_0 + \delta\tau$, and, as we can verify, n^a does not serve this role.

In other words, we are looking for a vector field that is *dual* to the one forms $\mathbf{\Omega} = \mathbf{d}\tau$. That is to say, we want to find a vector field t^a such that

$$\mathbf{d}\tau(\boldsymbol{t}) = t^a \nabla_a \tau = 1.$$

Then we can interpret t^a as the tangent vector $(\partial/\partial\tau)$. Now we see where our unit-vector field n^a fails, for its contraction with $\mathbf{d}\tau$ yields

$$n^a \nabla_a \tau = n^a \Omega_a = -\alpha \Omega^a \Omega_a = \alpha^{-1}.$$

From this we can see that the vector αn^a *would* be dual to $\nabla_a \tau$, but we can be even more general than this. If we took an arbitrary spatial vector β^a, the vector field t^a defined by

$$t^a \equiv \alpha n^a + \beta^a \tag{A.48}$$

will also be dual to $\nabla_a \tau$ (again, because $v^a \nabla_a \tau = 0$ for any spatial vector v^a). The spatial vector β^a defined in this way is known as the **shift** vector.

For the time being, we assume that the shift vector is zero (*i.e.*, $t^a = \alpha n^a$). We will use this simplification to examine the effect of Lie-differentiation along purely hypersurface-orthogonal directions. We will show that the Lie derivative along t^a of any spatial tensor is

spatial. Thus, we want to show, for any spatial tensor $T^{a\cdots b}{}_{c\cdots d}$, that

$$n_e \pounds_{\boldsymbol{t}} T^{a\cdots e\cdots b}{}_{c\cdots d} = 0.$$

This follows from

$$\begin{aligned} n_e \pounds_{\boldsymbol{t}} T^{a\cdots e\cdots b}{}_{c\cdots d} &= -\alpha\Omega_e \pounds_{\boldsymbol{t}} T^{a\cdots e\cdots b}{}_{c\cdots d} \\ &= -\alpha\Big(\pounds_{\boldsymbol{t}}\big(\Omega_e T^{a\cdots e\cdots b}{}_{c\cdots d}\big) - T^{a\cdots e\cdots b}{}_{c\cdots d}\pounds_{\boldsymbol{t}}\Omega_e\Big) \\ &= \alpha T^{a\cdots e\cdots b}{}_{c\cdots d}\pounds_{\boldsymbol{t}}\nabla_e\tau \\ &= \alpha T^{a\cdots e\cdots b}{}_{c\cdots d}\nabla_e\left(\pounds_{\boldsymbol{t}}\tau\right), \end{aligned}$$

where we have used the fact that, on scalars, $\pounds_{\boldsymbol{v}}\nabla_a f = \nabla_a\left(\pounds_{\boldsymbol{v}} f\right)$. Continuing,

$$\begin{aligned} n_e \pounds_{\boldsymbol{t}} T^{a\cdots e\cdots b}{}_{c\cdots d} &= \alpha T^{a\cdots e\cdots b}{}_{c\cdots d}\nabla_e\left(t^b\nabla_b\tau\right) \\ &= \alpha T^{a\cdots e\cdots b}{}_{c\cdots d}\nabla_e(1) \\ &= 0. \end{aligned} \tag{A.49}$$

It is the fact that t^a is dual to $\mathbf{d}\tau$ that allows this cancellation to happen. One could also have performed the calculation with a contraction of n^a on any covariant index of $\pounds_{\boldsymbol{t}} T^{a\cdots b}{}_{c\cdots d}$, and the result would similarly have vanished, albeit after a longer calculation.

We can use this result to show that $\pounds_{\boldsymbol{t}}$ preserves "spatiality" by showing that $\pounds_{\boldsymbol{t}}\gamma^a{}_b = 0$. Given a spatial vector v^a, we have from the result above that the Lie derivative along t is also spatial. This means that we can apply the projection tensor without changing anything, *i.e.*,

$$\begin{aligned} \pounds_{\boldsymbol{t}} v^a &= \gamma^a{}_b \pounds_{\boldsymbol{t}} v^b \\ &= \pounds_{\boldsymbol{t}}\big(\gamma^a{}_b v^b\big) - v^b \pounds_{\boldsymbol{t}}\gamma^a{}_b \\ &= \pounds_{\boldsymbol{t}} v^a - v^b \pounds_{\boldsymbol{t}}\gamma^a{}_b \\ 0 &= v^b \pounds_{\boldsymbol{t}}\gamma^a{}_b. \end{aligned}$$

This must be true for every spatial vector v^a, so $\pounds_{\boldsymbol{t}}\gamma^a{}_b = 0$. We could also show this by working out the expression for a Lie derivative of a mixed-type tensor, but we would see that, after a long calculation, everything cancels and we are left with zero.

Given this result, we see why t^a is our preferred time-evolution vector field. It takes spatial tensors from one time slice to spatial tensors in the next time slice. One can then ask, for an infinitesimal change in parameter time $\delta\tau$, how much proper time $\delta\bar{\tau}$ elapses for observers whose tangent vector is t^a. The answer is simply

$$(\delta\bar{\tau})^2 = -g_{ab}t^a t^b(\delta\tau)^2 = -g_{ab}(\alpha n^a)(\alpha n^b)(\delta\tau)^2 = \alpha^2(\delta\tau)^2. \tag{A.50}$$

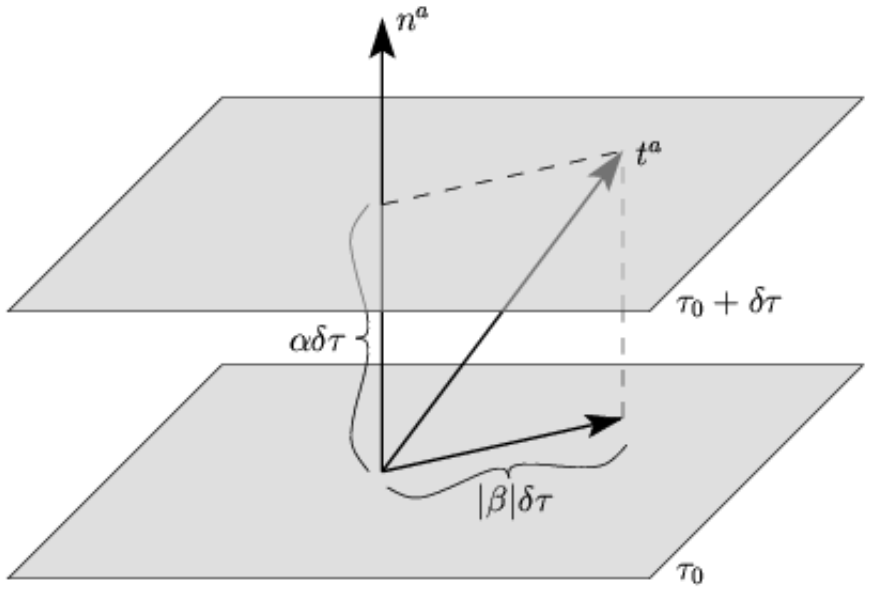


Figure A.4: Preferred time-evolution vector field t^a. This vector field is composed of a hypersurface-orthogonal piece (the normal vector n^a scaled by the lapse, α) and a hypersurface-tangential piece (the spatial shift vector, β^a).

Thus, the lapse just determines how much proper time elapses between time-slices along the normal vector n^a, via $\delta\bar{\tau} = \alpha\delta\tau$.

Now we relax our requirement that β^a equal zero. The shift vector then just provides a spatial translation relative to a completely orthogonal time-translation. It represents the remaining kinematical degree of freedom we have once our slicing is prescribed. These notions give rise to the situation depicted in Fig. A.4. Due to the additive property of Lie derivatives ($\pounds_{\boldsymbol{v}+\boldsymbol{w}} = \pounds_{\boldsymbol{v}} + \pounds_{\boldsymbol{w}}$), having a non-zero shift will simply mean that $\pounds_{\boldsymbol{t}} = \pounds_{\alpha\boldsymbol{n}} + \pounds_{\boldsymbol{\beta}}$. We will use these relations in the next section to finally construct our time-evolution equations.

A.9 Evolution equations

At last, we come to the evolution equations of the 3+1 formulation of general relativity. Recall that we have used the relations we derived regarding projections of the four-dimensional Riemann tensor, namely the Gauss and Codazzi equations, along with the Einstein equations, to derive the constraint equations of §A.7. We saw that by looking at the "time-time" projection of G_{ab} we found the Hamiltonian constraint and the "time-space" projection led us to the momentum constraint.

All that remains then is to look at the "space-space" projections. Much as in Newtonian particle mechanics, the physical theory providing the dynamics will tell us how to calculate

the "acceleration", which is related to the "velocity" and "position" then by kinematical relations. Considering our spatial metric as the configuration or "position" variable, we recall that we can express its normal-direction Lie derivative as $\pounds_{\boldsymbol{n}}\gamma_{ab} = -2K_{ab}$ from (A.26).

Given our discussion in the previous section, we realize that what we want, in fact, is the Lie derivative along t^a,

$$\pounds_{\boldsymbol{t}}\gamma_{ab} = \pounds_{\alpha\boldsymbol{n}}\gamma_{ab} + \pounds_{\boldsymbol{\beta}}\gamma_{ab}.$$

We can simplify the expression for $\pounds_{\alpha\boldsymbol{n}}\gamma_{ab}$ in terms of quantities we already know, finding that

$$\begin{aligned}
\pounds_{\alpha\boldsymbol{n}}\gamma_{ab} &= \alpha n^c\nabla_c\gamma_{ab} + \gamma_{cb}\nabla_a\left(\alpha n^c\right) + \gamma_{ac}\nabla_b\left(\alpha n^c\right) \\
&= \alpha n^c\nabla_c\gamma_{ab} + \gamma_{cb}n^c\nabla_a\alpha + \alpha\gamma_{cb}\nabla_a n^c + \gamma_{ac}n^c\nabla_b\alpha + \alpha\gamma_{ac}\nabla_b n^c \\
&= \alpha\left(n^c\nabla_c\gamma_{ab} + \gamma_{cb}\nabla_a n^c + \gamma_{ac}\nabla_b n^c\right) \\
&= \alpha\pounds_{\boldsymbol{n}}\gamma_{ab}.
\end{aligned}$$

Note that this result would have applied to any covariant spatial tensor, not just the spatial metric in particular.

We use this, as well as the extrinsic curvature relationship (A.26), to express the spatial metric evolution equation in the desired form,

$$\begin{aligned}
\pounds_{\boldsymbol{t}}\gamma_{ab} &= \alpha\pounds_{\boldsymbol{n}}\gamma_{ab} + \pounds_{\boldsymbol{\beta}}\gamma_{ab} = \alpha(-2K_{ab}) + \pounds_{\boldsymbol{\beta}}\gamma_{ab} \\
&= -2\alpha K_{ab} + \pounds_{\boldsymbol{\beta}}\gamma_{ab}. \qquad \text{(A.51)}
\end{aligned}$$

This last equation will be the evolution equation used to evolve the spatial metric from one constant-τ slice to another. One can think of this as the equivalent of $\dot{\mathbf{x}} = \mathbf{v}$, where again we use the analogy of the spatial metric as the position variable and the extrinsic curvature as a velocity.

Continuing this analogy further, we would have $\dot{\mathbf{v}} = \mathbf{a}$, and $\mathbf{a}$ in turn would be determined by Newton's second law and the respective forces acting on the particle in question. Here, Ricci's equation (A.38) will provide the kinematical role, and we will have to work to re-express it in terms of a projection of the Einstein tensor in order to work in the correct dynamics. We first get the equation into the appropriate time-derivative form we seek,

$$\begin{aligned}
\pounds_{\alpha\boldsymbol{n}}K_{ab} &= \alpha\pounds_{\boldsymbol{n}}K_{ab} \\
&= \alpha\left(\gamma^p{}_a\gamma^q{}_b R_{pcqd}n^c n^d - K_{ac}K_b{}^c - \alpha^{-1}D_aD_b\alpha\right) \\
&= \alpha\gamma^p{}_a\gamma^q{}_b R_{pcqd}n^c n^d - \alpha K_{ac}K_b{}^c - D_aD_b\alpha.
\end{aligned}$$

Then we will have that

$$\begin{aligned}\pounds_{\boldsymbol{t}} K_{ab} &= \pounds_{\alpha \boldsymbol{n}} K_{ab} + \pounds_{\beta} K_{ab} \\ &= \alpha \gamma^{p}{}_{a} \gamma^{q}{}_{b} R_{pcqd} n^{c} n^{d} - \alpha K_{ac} K_{b}{}^{c} - D_{a} D_{b} \alpha + \pounds_{\beta} K_{ab}. \end{aligned} \tag{A.52}$$

We will now work on re-describing the particular projection of the spacetime Riemann tensor appearing above as a projection of Einstein's equation. We find that

$$\begin{aligned}\gamma^{p}{}_{a} \gamma^{r}{}_{c} R_{pqrs} n^{q} n^{s} &= \gamma^{p}{}_{a} \gamma^{r}{}_{c} R_{pqrs} \left(\gamma^{qs} - g^{qs}\right) \\ &= \gamma^{p}{}_{a} \gamma^{r}{}_{c} \gamma^{qs} R_{pqrs} - \gamma^{p}{}_{a} \gamma^{r}{}_{c} R_{pr} \\ &= \gamma^{p}{}_{a} \gamma^{r}{}_{c} \gamma^{q}{}_{b} \gamma^{bs} R_{pqrs} - \gamma^{p}{}_{a} \gamma^{r}{}_{c} R_{pr} \\ &= g^{bd} \gamma^{p}{}_{a} \gamma^{r}{}_{c} \gamma^{q}{}_{b} \gamma^{s}{}_{d} R_{pqrs} - \gamma^{p}{}_{a} \gamma^{r}{}_{c} R_{pr} \\ &= g^{bd} \left(\mathsf{R}_{abcd} + K_{ca} K_{bd} - K_{cb} K_{ad}\right) - \gamma^{p}{}_{a} \gamma^{r}{}_{c} R_{pr} \\ &= \mathsf{R}_{ac} + K_{ca} K - K_{cb} K_{a}{}^{b} - \gamma^{p}{}_{a} \gamma^{r}{}_{c} R_{pr}. \end{aligned} \tag{A.53}$$

We have used Gauss' equation again (A.32) in order to replace the fully-spatially projected Riemann tensor with the extrinsic curvature and three-dimensional Riemann terms. With this expression, it is clear how we might see Einstein's equations enter. Rewriting Einstein's equations as

$$R_{ab} = \kappa \left(T_{ab} - \tfrac{1}{2} T g_{ab}\right),$$

we simply need to contract twice with the projection tensor,

$$\begin{aligned}\gamma^{p}{}_{a} \gamma^{q}{}_{b} R_{pq} &= \kappa \gamma^{p}{}_{a} \gamma^{q}{}_{b} T_{pq} - \tfrac{1}{2} \kappa T \gamma^{p}{}_{a} \gamma^{q}{}_{b} g_{pq} \\ &= \kappa S_{ab} - \tfrac{1}{2} \kappa T \gamma_{ab}. \end{aligned} \tag{A.54}$$

In doing so, we have defined the spatial stress tensor,

$$S_{ab} \equiv \gamma^{p}{}_{a} \gamma^{q}{}_{b} T_{ab}. \tag{A.55}$$

We can work out an expression for the trace of the stress-energy tensor in terms of three-dimensional quantities, namely

$$\begin{aligned} T &= g^{ab} T_{ab} = (\gamma^{ab} - n^{a} n^{b}) T_{ab} \\ &= \gamma^{cd} \gamma^{a}{}_{c} \gamma^{b}{}_{d} T_{ab} - \rho \\ &= \gamma^{cd} S_{cd} - \rho = S - \rho, \end{aligned} \tag{A.56}$$

where we have used the definition of the energy density (A.43).

Then, combining (A.53), (A.54), and (A.56), we have

$$\gamma^p{}_a\gamma^q{}_b R_{pq} = \kappa S_{ab} - \tfrac{1}{2}\kappa(S-\rho)\gamma_{ab}$$

and

$$\gamma^p{}_a\gamma^r{}_c R_{pqrs}n^q n^s \;=\; \mathsf{R}_{ac} + K_{ca}K - K_{cb}K_a{}^b - \kappa\left(S_{ac} - \tfrac{1}{2}(S-\rho)\gamma_{ac}\right).$$

We then replace this expression for a particular contraction of the spacetime Riemann tensor, into its corresponding place in the evolution equation for the extrinsic curvature (A.52),

$$\begin{aligned}\pounds_{\boldsymbol{t}} K_{ab} \;=\; & \alpha\Big[\mathsf{R}_{ab} + K_{ba}K - K_{bc}K_a{}^c - \kappa\left(S_{ab} - \tfrac{1}{2}(S-\rho)\gamma_{ab}\right)\Big] \\ & -\alpha K_{ac}K_b{}^c - D_aD_b\alpha + \pounds_\beta K_{ab}.\end{aligned}$$

Cleaning this up a bit, we finally have the evolution equation for the extrinsic curvature,

$$\begin{aligned}\pounds_{\boldsymbol{t}} K_{ab} \;=\; & -D_aD_b\alpha + \alpha\Big[\mathsf{R}_{ab} - 2K_{ca}K_b{}^c + K_{ba}K\Big] \\ & -\alpha\kappa\Big[S_{ab} - \tfrac{1}{2}(S-\rho)\gamma_{ab})\Big] + \pounds_\beta K_{ab}.\end{aligned} \tag{A.57}$$

Equations (A.51) and (A.57) together form the evolution equations, which, given an initial data set consisting of (Σ, γ_{ab}, K_{ab}) satisfying both constraint equations (A.42) and (A.47), will evolve the spatial metric into the future (or past). With the spatial metric, the lapse, and the shift on all of these different time-slices, one can reconstruct the spacetime metric, thereby finding the solution to Einstein's equations that one desired.

It should be noted that, by virtue of the Bianchi identities, if the constraint equations are identically satisfied on an initial time slice, the evolution equations will preserve this property (*i.e.*, the constraints will be satisfied for all times).

We could end at this point, having fully derived the equations necessary to treat general relativity in an initial-value formulation, but we will consider two more details, both of relevance for numerical relativity. The first concerns constructing an appropriate coordinate system for our evolutions and the second concerns reconstructing a spacetime from a numerical simulation.

A.10 3+1 computational frames

Every calculation thus far has been performed without reference to a coordinate system — all tensor equations have been written in the abstract index notation (*cf.* [77]). When it comes time to write a computer code to implement the constraint and evolution equations, it will often (although not always) be helpful to work in terms of component relations. What

we will aim to do here is to introduce a basis that is adapted to the formulation presented thus far. In this section, we will closely follow the corresponding development in Baumgarte and Shapiro's review article [14].

A.10.1 Introducing the frames

We already have a natural basis vector for the time-like direction, namely our time-evolution vector field t^a. We choose this to be our temporal basis vector, $(\mathrm{e}_0)^a \equiv t^a$. Equivalently, one could say, in this basis, t^a has components $(1,0,0,0)$. Even more than this, we can make this a *coordinate* basis, where τ plays the role of the temporal coordinate. Specifically, we will have the basis vector $\mathbf{e}_0 = \boldsymbol{\partial/\partial\tau}$ and the dual basis one-form $\mathbf{e}^0 = \mathbf{d}\tau$. Choosing this basis has the particularly nice feature that suddenly, all of our Lie derivatives along t^a become partial derivatives with respect to τ: $\pounds_{\boldsymbol{t}} \rightarrow \partial/\partial\tau$.

Then, on any given spatial slice Σ, we will introduce a basis of spatial vectors $\{\mathbf{e}_i\}$, where the index i will take on values 1, 2, 3 and will denote *which* basis vector one is discussing. This basis will be, by construction, tangent to Σ, and therefore normal to Ω_a,

$$\boldsymbol{\Omega}(\mathbf{e}_i) = \Omega_a(\mathrm{e}_i)^a = 0. \tag{A.58}$$

Defined on one slice, we will then insist that this basis be Lie-dragged along t^a to every other slice. We do this because we want our spatial basis, tangent to Σ_τ, to also be the spatial basis tangent to $\Sigma_{\tau+\delta\tau}$, and the next slice, and so on. We can see then that what we want is just

$$\pounds_{\boldsymbol{t}}\big(\boldsymbol{\Omega}(\mathbf{e}_i)\big) = \pounds_{\boldsymbol{t}}\big(\Omega_a(\mathrm{e}_i)^a\big) = 0. \tag{A.59}$$

But, by the Leibniz rule, we know that

$$\begin{aligned}\pounds_{\boldsymbol{t}}\big(\Omega_a(\mathrm{e}_i)^a\big) &= \Omega_a \pounds_{\boldsymbol{t}}(\mathrm{e}_i)^a + (\mathrm{e}_i)^a \pounds_{\boldsymbol{t}}\Omega_a \\ &= \Omega_a \pounds_{\boldsymbol{t}}(\mathrm{e}_i)^a,\end{aligned} \tag{A.60}$$

where we have used the fact that $\pounds_{\boldsymbol{t}}\Omega_a = 0$ coming from the duality $t^a\Omega_a = 0$, *i.e.*,

$$\pounds_{\boldsymbol{t}}\Omega_a = \pounds_{\boldsymbol{t}}\big(\nabla_a\tau\big) = \nabla_a\big(\pounds_{\boldsymbol{t}}\tau\big) = \nabla_a\big(1\big) = 0.$$

So, the condition that the spatial basis remain always tangent to whatever slice it is carried (A.59), along with the relation (A.60) tells us that

$$\pounds_{\boldsymbol{t}}\mathbf{e}_i = 0,$$

which, as we stated, simply means that the spatial basis must be Lie-dragged along t^a to

each slice.

Thus, we have our full set of basis vectors $\{\mathbf{e}_\mu\}$, where any Greek index will take on the component values $0, 1, 2, 3$. We define the dual basis of one-forms $\{\mathbf{e}^\mu\}$ by requiring that the bases be appropriately dual: $\mathbf{e}^\mu(\mathbf{e}_\nu) = \delta^\mu{}_\nu$ (for $\mu, \nu = 0, 1, 2, 3$). We can then find the relationships

$$(\mathrm{e}^\mu)_a(\mathrm{e}_\nu)^b\delta^a{}_b = \delta^\mu{}_\nu \text{ and} \tag{A.61}$$

$$(\mathrm{e}^\mu)_b(\mathrm{e}_\nu)^a\delta^\nu{}_\mu = \delta^a{}_b. \tag{A.62}$$

The bases introduced in this way are sometimes called **computational frames** (or 3+1 frames).

A.10.2 Spacetime and spatial metric components

We now wish to re-express our earlier abstract-index expressions in this particular basis. We will start by building up covariant and contravariant expressions for the normal vector n^a and the spatial metric γ_{ab} so that we can reconstruct the full spacetime metric in this basis. We note that all of the spatial, covariant components of n^a must vanish in this basis, *i.e.*,

$$n_i = n_a(\mathrm{e}_i)^a = -\alpha\Omega_a(\mathrm{e}_i)^a = 0,$$

using our construction (A.58). Thus, in components, $[n_\mu] = (n_0, 0, 0, 0)$. From this, we can see that any spatial tensor with contravariant indices must have vanishing (0)-components. For a spatial, contravariant vector, in particular, we find that

$$v^\mu n_\mu = 0 = v^0 n_0 + v^i n_i = v^0 n_0 \quad \Rightarrow \quad v^0 = 0. \tag{A.63}$$

In general then, a spatial, contravariant vector will have components: $[v^\mu] = (0, v^i)$. We use this fact to find the components of the contravariant normal-vector field, n^μ. Using (A.48), we know $n^a = \alpha^{-1}(t^a - \beta^a)$, so in components, we can use the knowledge that $[t^\mu] = (1, \mathbf{0})$ and $[\beta^\mu] = (0, \beta^i)$ to write

$$[n^\mu] = \alpha^{-1}(1, -\beta^i). \tag{A.64}$$

Using this result and the normalization of n^a, (*i.e.*, $n^a n_a = -1$), we can work out the covariant components,

$$n^\mu n_\nu = -1 = n^0 n_0 + n^i n_i = n^0 n_0 + 0 = \alpha^{-1} n_0.$$

We then find that

$$[n_\mu] = (-\alpha, \mathbf{0}). \tag{A.65}$$

Having component-expressions for n^a, we just have to work out component expressions for the spatial metric, and then we will have the components of the full spacetime metric as well. Recall from above that any spatial, contravariant tensor, must have vanishing (0)-components. This means that the spatial metric's components in this basis are

$$[\gamma^{\mu\nu}] = \left(\begin{array}{c|c} 0 & \mathbf{0} \\ \hline \mathbf{0} & \gamma^{ij} \end{array}\right), \tag{A.66}$$

and we have from (A.64) the expression for $n^\mu n^\nu$,

$$[n^\mu n^\nu] = \alpha^{-2} \left(\begin{array}{c|c} 1 & -\beta^j \\ \hline -\beta^i & \beta^i\beta^j \end{array}\right). \tag{A.67}$$

Combining these two matrices, according to the definition of the contravariant spatial metric, $g^{ab} = \gamma^{ab} - n^a n^b$, we have

$$[g^{\mu\nu}] = \left(\begin{array}{c|c} -\alpha^{-2} & \alpha^{-2}\beta^j \\ \hline \alpha^{-2}\beta^i & \gamma^{ij} - \alpha^{-2}\beta^i\beta^j \end{array}\right). \tag{A.68}$$

We now set out to find the covariant form of the spacetime metric in this basis. Given our expression for the covariant components of n_a (A.65), we can easily see that the tensor $n_a n_b$ will have the component expression

$$[n_\mu n_\nu] = \left(\begin{array}{c|c} \alpha^2 & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right). \tag{A.69}$$

Given the above component representation and $g_{ab} = \gamma_{ab} + n_a n_b$, we can see right away that $g_{ij} = \gamma_{ij}$, and for that matter, that $g_{0i} = \gamma_{0i}$. With this information in hand, we will now go about finding the covariant components of the spacetime metric. We will do

this by constructing the inverse of $g^{\mu\nu}$. In other words, we seek the tensor $g_{\nu\rho}$ such that $g^{\mu\nu}g_{\nu\rho} = \delta^{\mu}{}_{\rho}$. This leads to the relations:

$$g^{0\nu}g_{\nu 0} = \delta^{0}{}_{0} = \quad 1 = \; g^{00}g_{00} + g^{0i}g_{i0} \tag{A.70a}$$

$$g^{0\nu}g_{\nu i} = \delta^{0}{}_{i} = \quad 0 = \; g^{00}g_{0i} + g^{0j}g_{ji} \tag{A.70b}$$

$$g^{i\nu}g_{\nu j} = \; \delta^{i}{}_{j} = \delta^{i}{}_{j} = g^{i0}g_{0j} + g^{ik}g_{kj}. \tag{A.70c}$$

We know the contravariant components of the spacetime metric from (A.68), so these relations give us a simple system of equations to solve in order to construct the covariant components. We begin with the second relation (A.70b),

$$\begin{aligned} 0 &= g^{00}g_{0i} + g^{0j}g_{ji} \\ 0 &= (-\alpha^{-2})\gamma_{0i} + (\alpha^{-2}\beta^{j})\gamma_{ji} \\ \gamma_{0i} &= \beta^{j}\gamma_{ij}. \end{aligned}$$

We will use this result in computing the g_{00} component from the first equation (A.70a),

$$\begin{aligned} 1 &= g^{00}g_{00} + g^{0i}g_{i0} \\ 1 &= (-\alpha^{-2})g_{00} + (\alpha^{-2}\beta^{i})\gamma_{i0} \\ -\alpha^{2} &= g_{00} - \beta^{i}\gamma_{i0} \\ g_{00} &= -\alpha^{2} + \beta^{i}\gamma_{i0} \\ g_{00} &= -\alpha^{2} + \beta^{i}\beta^{j}\gamma_{ij}. \end{aligned}$$

Finally, we will examine the last relation (A.70c),

$$\begin{aligned} \delta^{i}{}_{j} &= g^{i0}g_{0j} + g^{ik}g_{kj} \\ \delta^{i}{}_{j} &= (\alpha^{-2}\beta^{i})\gamma_{0j} + (\gamma^{ik} - \alpha^{-2}\beta^{i}\beta^{k})\gamma_{kj} \\ \delta^{i}{}_{j} &= \alpha^{-2}\beta^{i}\beta^{k}\gamma_{kj} + \gamma^{ik}\gamma_{kj} - \alpha^{-2}\beta^{i}\beta^{k}\gamma_{kj} \\ \delta^{i}{}_{j} &= \gamma^{ik}\gamma_{kj}. \end{aligned}$$

In constructing the covariant spacetime metric as the inverse of the contravariant spacetime metric, we get for free that the covariant and contravariant spatial metrics are the inverses of each other, when contracted over indices $i, j = 1, 2, 3$. Using the components we have just derived, the covariant spacetime metric takes on the form

$$[g_{\mu\nu}] = \left(\begin{array}{c|c} -\alpha^{2} + \beta^{i}\beta^{j}\gamma_{ij} & \beta^{i}\gamma_{ij} \\ \hline \beta^{j}\gamma_{ji} & \gamma_{ij} \end{array}\right), \tag{A.71}$$

and the components of the spatial metric are

$$[\gamma_{\mu\nu}] = \left(\begin{array}{c|c} \beta^i\beta^j\gamma_{ij} & \beta^i\gamma_{ij} \\ \hline \beta^j\gamma_{ji} & \gamma_{ij} \end{array}\right). \tag{A.72}$$

Simply for completeness, we also mention that we can now work out what the projection tensor will look like in this basis, namely

$$[\gamma^\mu{}_\nu] = \left(\begin{array}{c|c} 0 & \mathbf{0} \\ \hline \beta^j & \delta^i_j \end{array}\right) = [\gamma_\nu{}^\mu]^{\mathrm{T}}, \tag{A.73}$$

where $[\,\cdot\,]^{\mathrm{T}}$ denotes the matrix transpose.

A.10.3 Utility of computational frames

The real benefit of this basis is that, for spatial tensors, we need only concern ourselves with the components running over index values $1, 2, 3$. The reason for this can be seen by the matrix expression for the covariant spatial metric above. We see that in the (i, j)-sub-block of the matrix, we just have γ_{ij}. In the other components of the matrix, we simply have this object contracted with the shift, which is, by construction, also a spatial vector. In other words, the only independent components of the four-dimensional tensor $\gamma_{\mu\nu}$ come from the spatial $(1, 2, 3)$ components γ_{ij}. The (00), $(0j)$, and $(i0)$ components just carry redundant information and can all be determined from the (ij) components (given a shift vector β^i). This is true for any spatial tensor. We will concern ourselves shortly with how to reconstruct the four-dimensional components of spatial tensors when only the three-dimensional components are provided.

In fact, there is an issue underlying this treatment which is almost entirely glossed-over in many treatments of the 3+1 decomposition. When we perform calculations with spatial tensors in numerical relativity, we truly are working only with three-dimensional objects, often written out in terms of components on a three-dimensional vector (and dual vector) basis. The subtlety then, is that one is working with a three-dimensional manifold, say, denoted by M, that is *diffeomorphic* to the slice Σ_τ, a submanifold of $\mathcal{M}$. In practice, this is mostly just an argument in semantics. As long as the space in which our calculations are preformed *is* diffeomorphic to the slice Σ_τ in which we want our tensors to actually live, we will not need to make this distinction.

But the real advantage of the computational, or 3+1 frames, is that the diffeomorphism

map is quite simple. Within our computational manifold M, the components ${S_{ijk}}^{lmn}$ of any arbitrary tensor, and the three basis vectors $\{\mathbf{e}_i\}$ suffice to completely determine the tensor. On Σ_τ, the same is also true, but as a part of the larger spacetime, we will want to express the tensor in terms of the fourth basis vector $\mathbf{e}_0$. In this four-dimensional basis, any spatial tensor can have non-zero non-spatial components, but because the tensor *is* uniquely determined by its spatial components on the spatial basis, its non-spatial components *must* be redundant in the sense that the non-spatial components can only be algebraic combinations of the spatial components.

From this point forward, we will not make the distinction between M and Σ_τ, and will think of all spatial tensors as objects with indices i, j, k running over values $1, 2, 3$, until we wish to construct their four-dimensional components over the four-dimensional set of basis vectors.

We note that this point of view is to easy to adopt because, while the four-dimensional representations of the spatial metric $\gamma^{\mu\nu}$ and $\gamma_{\mu\nu}$ are *not* inverses of each other, within the (i, j) sub-block, γ^{ij} and γ_{ij} are inverses, and so we can raise indices on spatial tensors with γ^{ij} and lower them with γ_{ij}. In fact, for all relations holding between spatial tensors, we will drop the other components altogether. To drive this home, we now restate the constraint and evolution equations of the 3+1 formalism in the computational frame. The constraint equations will become

$$\mathsf{R} + K^2 - K_{ij}K^{ij} = 2\kappa\rho \tag{A.74}$$

$$D_j {K_i}^j - D_i K = \kappa j_i, \tag{A.75}$$

while the evolution equations will become

$$\frac{\partial}{\partial\tau}\gamma_{ij} = -2\alpha K_{ij} + \pounds_\beta \gamma_{ij} \tag{A.76}$$

$$\begin{aligned}\frac{\partial}{\partial\tau}K_{ij} = &- D_i D_j \alpha + \alpha\Big[\mathsf{R}_{ij} - 2K_{ki}{K_j}^k + K_{ij}K\Big] \\ &- \alpha\kappa\Big[S_{ij} - \tfrac{1}{2}(S-\rho)\gamma_{ij})\Big] + \pounds_\beta K_{ij}.\end{aligned} \tag{A.77}$$

These equations, (A.74), (A.75), (A.76), and (A.77) collectively, typically are known as the **ADM equations**, although, one should point out that the *true* ADM equations, as written by Arnowitt, Deser, and Misner, were expressed in terms of the spatial metric and its conjugate momentum π^{ij}, not the extrinsic curvature [12].

A few comments are necessary regarding this organization.

1. Again, the labels i, j, k *are* component labels, and they will take on values $1, 2, 3$.

2. We have not made any restriction on the spatial basis. One can choose it to be a coordinate basis or an orthonormal one as the case presents itself. For the purposes

of computational simplicity, we will almost invariably choose a coordinate basis.

3. Because of the freedom in choosing one's basis, one must take care regarding the spatial covariant derivative operator D. Specifically, it is true that it is the unique torsion-free metric-compatible derivative operator (compatible with γ_{ij}), but one's basis will affect component expressions of the connection coefficients.

4. In the evolution equations, we have made explicit that our four-dimensional basis is partially holonomic: we have selected a coordinate basis for the time-direction, and so the Lie derivatives with respect to t^a have been legitimately turned into partial derivatives with respect to time τ.

We consider one further aspect of this framework. If one *does* choose the spatial basis to be a coordinate basis, (*i.e.*, $\mathbf{e}_i = \partial/\partial x^i$ and $\mathbf{e}^i = \mathbf{d}x^i$), then we see from the covariant components of the spacetime metric that the line-element will take on the form

$$ds^2 = -\alpha^2 \mathbf{d}\tau^2 + \gamma_{ij}(\mathbf{d}x^i + \beta^i \mathbf{d}\tau)(\mathbf{d}x^j + \beta^j \mathbf{d}\tau). \tag{A.78}$$

It is this form of the metric that lets one translate known solutions of the Einstein's equations into the 3+1 formalism. Generally, one can read off the spatial metric, and from this work out what the lapse and shift must be. Finding the extrinsic curvature for a known solution of Einstein's equations, however, will require knowledge of time derivatives and can sometimes be computationally expensive and symbolic manipulation packages such as *Mathematica* or *Maple* can be quite handy.

A.10.4 There and back again

An aspect that is rarely touched upon in numerical relativity, strangely, is the reconstruction of four-dimensional spacetime quantities from the results of a numerical evolution involving the spatial metric, extrinsic curvature, lapse, and shift. In our computational frames, we will have a spatial basis (most likely a coordinate basis) prescribed on the initial time slice. This basis will be Lie-dragged to each subsequent time slice as we integrate forward (or backward) in time. One would then like to know, at some time τ_0, what are the components of, say, the four-dimensional spacetime metric? Or, even better, what are the components of the four-dimensional Einstein tensor? Do they all equal the corresponding reconstructed four-dimensional components of the stress-energy tensor as they should?

There are a few vectors and tensors for which it is a trivial task to reconstruct the full spacetime components. Of course, we already know that the lapse and shift together contain all the information necessary to form the four-dimensional components of the n^μ vectors and n_μ one-forms via equations (A.64) and (A.65). And also given the lapse and shift, the

spatial metric γ_{ij} contains all the information necessary to construct the spacetime metric via (A.71) and its inverse via (A.68).

For any arbitrary spatial vector, v^i, we recall from the discussion surrounding (A.63) that the contravariant (0)-components of spatial tensors will all be zero. Thus we know that the contravariant components on the full four-dimensional basis will then simply be

$$[v^\mu] = (0, v^i).$$

The covariant form of any spatial vector can be seen by contracting the covariant spacetime metric with the expression above. The result is just

$$[v_\mu] = (\beta^j v_j, v_i) = (\gamma_{ij}\beta^i v^j, \gamma_{ij} v^j).$$

For more complicated tensors, the situation is a bit different. In general, one will need to have *all* of the spatio-temporal projections available on a time slice in order to reconstruct the full tensor. For instance, consider the stress-energy tensor T_{ab}. We defined all of the projections possible:

$$\begin{aligned} S_{ab} &= \gamma^c{}_a \gamma^d{}_b T_{cd} \\ j_a &= -\gamma^b{}_a n^c T_{bc} \\ \rho &= n^a n^b T_{ab}. \end{aligned}$$

The number of independent projections/contractions with $\gamma^a{}_b$ and n^a will depend upon the rank and symmetries of the tensor under question.

For any tensor though, we can discover what pieces are necessary simply by applying the identity map on every index and expanding it in terms of $\gamma^a{}_b$ and n^a. In particular, for the stress-energy tensor (or any symmetric, second-rank covariant tensor with similarly defined projections), we see that

$$\begin{aligned} T_{\mu\nu} &= \delta^\rho{}_\mu \delta^\sigma{}_\nu T_{\rho\sigma} \\ &= (\gamma^\rho{}_\mu - n^\rho n_\mu)(\gamma^\sigma{}_\nu - n^\sigma n_\nu) T_{\rho\sigma} \\ &= (\gamma^\rho{}_\mu \gamma^\sigma{}_\nu - \gamma^\rho{}_\mu n^\sigma n_\nu - n^\rho n_\mu \gamma^\sigma{}_\nu + n^\rho n_\mu n^\sigma n_\nu) T_{\rho\sigma} \\ &= S_{\mu\nu} + j_\mu n_\nu + j_\nu n_\mu + n_\mu n_\nu \rho \\ &= S_{\mu\nu} + 2 j_{(\mu} n_{\nu)} + n_\mu n_\nu \rho. \end{aligned}$$

Now we see a potential problem: we do not have the four-dimensional components of higher-rank tensors like $S_{\mu\nu}$. We only have the components of the three-dimensional sub-block, S_{ij}. As we will see, though, this is not a reason for concern.

The same reasoning which allowed us to determine the four-dimensional components of any spatial vector also allow us to reconstruct higher rank tensors as well. Again, because of the components of n_a in this basis (A.65), all purely contravariant spatial tensors will have vanishing (0)-components. Thus, we could use our spatial, three-dimensional metric γ^{ij} to raise all the spatial indices so that we would have a purely contravariant spatial tensor, say, $S^{i\ldots j}$. Then, the four-dimensional tensor is just $S^{\mu\ldots\nu}$, where the three-dimensional sub-block is given by $S^{i\ldots j}$, and all other components are zero. Then, we can use the spacetime metric to lower the indices until we get the tensor back in the form we want.

We illustrate this procedure with a rank-two example, only to make it easy to interpret the components as matrices. Suppose we have the components S_{ij}, the three-dimensional projection of the stress-energy tensor. We wish to have the four-dimensional components $S_{\mu\nu}$. Via the method proposed above, we construct the contravariant tensor $S^{ij} = \gamma^{ik}\gamma^{jl}S_{kl}$. Then, the four-dimensional tensor will have components

$$[S^{\mu\nu}] = \left(\begin{array}{c|c} 0 & \mathbf{0} \\ \hline \mathbf{0} & \gamma^{ik}\gamma^{jl}S_{kl} \end{array}\right).$$

We get the desired tensor by lowering the indices with the spacetime metric (or spatial metric; it doesn't matter). So, we have $S_{\mu\nu} = g_{\mu\sigma}g_{\nu\rho}S^{\sigma\rho}$. Using the components of the metric given in (A.71), we find

$$\begin{aligned}[S_{\mu\nu}] &= \left(\begin{array}{c|c} -\alpha^2 + \beta^m\beta_m & \beta_j \\ \hline \beta_i & \gamma_{ij} \end{array}\right)\left(\begin{array}{c|c} 0 & \mathbf{0} \\ \hline \mathbf{0} & S^{jk} \end{array}\right)\left(\begin{array}{c|c} -\alpha^2 + \beta^m\beta_m & \beta_l \\ \hline \beta_k & \gamma_{kl} \end{array}\right) \\ &= \left(\begin{array}{c|c} S^{jk}\beta_j\beta_k & S^j{}_l\beta_j \\ \hline S_i{}^k\beta_k & S_{il} \end{array}\right),\end{aligned}$$

where the results follow from simple matrix multiplication. Simplifying this expression by juggling a few indices, we see that

$$[S_{\mu\nu}] = \left(\begin{array}{c|c} S_{jk}\beta^j\beta^k & S_{jl}\beta^j \\ \hline S_{ik}\beta^k & S_{il} \end{array}\right). \tag{A.79}$$

One can imagine that if we had to follow this procedure for every tensor, first raising the three-dimensional indices with the three-dimensional metric γ^{ij} and then lowering the four-dimensional indices with the four-dimensional metric, this procedure would become quite computationally expensive for higher-rank tensors.

There is a shortcut, thankfully, to this procedure that produces the same effect. One can construct an *ersatz* four-dimensional tensor which contains the correct three-dimensional components, but has zeroes in all other components, even if all the indices are not contravariant. Then one applies the projection operator $\gamma^{\mu}{}_{\nu}$ to every index. The resulting tensor will have the correct components by virtue of the projection tensor being the identity map on spatial tensors.

We repeat the example above with this procedure and see how it works. We trivially construct our false four-dimensional tensor $\tilde{S}_{\mu\nu}$ from S_{ij}, with zeroes in the non-spatial components,

$$[\tilde{S}_{\mu\nu}] = \left(\begin{array}{c|c} 0 & \mathbf{0} \\ \hline \mathbf{0} & S_{ij} \end{array}\right).$$

Then we apply the four-dimensional projection tensor to each index of this false-tensor and the result will be the correct $S_{\mu\nu}$. So, with the aid of (A.73), we find that $S_{\mu\nu} = \gamma_{\mu}{}^{\sigma}\tilde{S}_{\sigma\rho}\gamma^{\rho}{}_{\nu}$ provides us with

$$\begin{aligned}[S_{\mu\nu}] &= \left(\begin{array}{c|c} 0 & \beta^j \\ \hline \mathbf{0} & \delta_i{}^j \end{array}\right)\left(\begin{array}{c|c} 0 & \mathbf{0} \\ \hline \mathbf{0} & S_{jk} \end{array}\right)\left(\begin{array}{c|c} 0 & \mathbf{0} \\ \hline \beta^k & \delta^k{}_l \end{array}\right) \\ &= \left(\begin{array}{c|c} S_{jk}\beta^j\beta^k & S_{jl}\beta^j \\ \hline S_{ik}\beta^k & S_{il} \end{array}\right).\end{aligned}$$

A quick comparison with (A.79) shows that this procedure has produced exactly the same components as the first method. The fact that this works at all comes from the fact that the non-spatial four-dimensional components of a spatial tensor carry nothing but redundant information. In fact, we need not have defined our *ersatz* tensor as having zeroes in the non-spatial components. Anything at all could go in those components. The projection tensor will "clean up" in a manner of speaking, whatever is placed by hand into those components and the result will be the correct combinations of the shift and the spatial

tensor in question.

The overall program is then quite simple. Given a spatial tensor, we can reconstruct its four-dimensional component representation. When one assembles together all of the spatial projections of a tensor, as we did above for the stress-energy tensor, then one can work out how to put these projections together with various factors of $\gamma^{\mu}{}_{\nu}$ and n^{μ} in order to recover the full four-dimensional components of the tensor. In this way, one can recover spacetime tensors, living in a given time-slice, from the output of a three-dimensional simulation.

A.11 Summary

What we have attempted to provide in this appendix is a detailed work-through of the 3+1 decomposition as it is typically used in numerical relativity. Once one arrives at the so-called ADM equations, (A.74) – (A.77), the analysis is essentially complete, and one can proceed to implement these equations in a computational code, or choose to introduce alternate variables such that the resulting system of equations might have more desirable properties in a numerical implementation.

The motivation, then, behind repeating this derivation here is to point out some of the subtleties that arise along the way, as well as to offer guidance in understanding from what principles the equations have arisen, and how one should interpret them.

Appendix B

Kerr-Schild coordinates

Historically, the spatially isotropic Schwarzschild coordinates were the preferred coordinates for constructing initial data that represented as single, non-rotating black hole. Those coordinates, as discussed in more detail in §3.1, have the advantage of not intersecting the physical singularity of the Schwarzschild spacetime.

For many of the more recent evolutions of black hole spacetimes, we are motivated by the development of singularity excision techniques to have coordinates that are horizon-penetrating. (Note that the constant-time slices of the isotropic Schwarzschild coordinates do penetrate the horizon, but do not enter the black-hole region, *i.e.*, the upper quadrant of a Kruskal-Szekeres diagram. In this sense, these coordinates "skim" the surface of the black hole. Evolving from the $t = 0$ slice with a different lapse prescription, such as the symmetric maximal slicing condtions, will lead to time-slices that do penetrate the horizon and enter the black-hole region.) From these considerations, we briefly present the Kerr-Schild form of solutions, and their specialization to non-rotating black holes.

B.1 Kerr-Schild metric

In [47], Kerr and Schild present a spacetime metric that includes as special cases solutions representing non-spinning and spinning black holes. This metric takes on the surprisingly simple form,

$$g_{\mu\nu} = \eta_{\mu\nu} + 2H\ell_\mu\ell_\nu, \tag{B.1}$$

where $\eta_{\mu\nu}$ is the standard four-dimensional Minkowskian metric in Cartesian coordinates, *i.e.*, $\eta_{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1)$, $H = H(x^\mu)$ is a scalar function, and ℓ_μ is a null vector (with respect to either $g_{\mu\nu}$ or $\eta_{\mu\nu}$),

$$g_{\mu\nu}\ell^\mu\ell^\nu = \eta_{\mu\nu}\ell^\mu\ell^\nu = 0. \tag{B.2}$$

Note that this equation equivalently implies that $(\ell_0)^2 = \ell_i\ell^i = \delta^{ij}\ell_i\ell_j$. Also, we note that the inverse metric is given by $g^{\mu\nu} = \eta^{\mu\nu} - 2H\ell^\mu\ell^\nu$.

We can see that this metric does indeed correspond to the Kerr solution (in Cartesian Kerr coordinates) when one introduces the choices

$$H = \frac{Mr^3}{r^4 + a^2 z^2}, \tag{B.3}$$

where $r = r(x, y, z, a)$ is given by the solutions to

$$r^4 - r^2(\rho^2 - a^2) - a^2 z^2 = 0 \quad \text{with} \quad \rho^2 = x^2 + y^2 + z^2, \tag{B.4}$$

and

$$[\ell_\mu] = \left(1, \frac{rx + ay}{r^2 + a^2}, \frac{ry - ax}{r^2 + a^2}, \frac{z}{r}\right). \tag{B.5}$$

Here, M is the mass of the black hole and a is the spin parameter, $a = J/M$ (where the spin is aligned with the z-axis). In the case of $a = 0$, we recover the Schwarzschild solution in the Eddington-Finkelstein form, to be discussed in greater detail below.

B.2 3+1 decomposition of Kerr-Schild

Because the Kerr-Schild metric takes on such a simple form, it is very easy to derive the three-dimensional fields that we would use as initial data for a 3+1 simulation. Comparing the Kerr-Schild metric Eq. (B.1) with the 3+1 decomposition provided in Eq. (A.71), we see that the ADM quantities are simply:

$$\alpha = \left(1 + 2H(\ell_0)^2\right)^{-1/2} \tag{B.6}$$

$$\beta_i = 2H\ell_0\ell_i \tag{B.7}$$

$$\gamma_{ij} = \delta_{ij} + 2H\ell_i\ell_j \tag{B.8}$$

The formula for the extrinsic curvature for arbitary spin parameter a is a bit more involved, and will not be presented here. At this point, we will concentrate on the $a = 0$ solutions, as these will be used extensively in our studies of excision techniques in black-hole spacetimes.

B.3 Eddington-Finkelstein coordinates

For vanishing spin-paramater a, the Kerr-Schild metric as described above reduces to what is commonly known as the ingoing (or outgoing) Eddington-Finkelstein coordinates. Eddington [38] and Finkelstein [42] separately (and separated by ∼35 years) noticed a modification of the Schwarzschild solution which avoids the problem of the Schwarzschild coordinates, namely the coordinate singularity at $r = 2M$. In essence, these "new" coordinates just in-

volve a rescaling of the Schwarzschild time coordinate so that null geodesics pass smoothly through the horizon. One can construct one set of coordinates such that ingoing (decreasing radial coordinate with increasing coordinate time) null geodesics pass smoothly through the horizon or another set such that the outgoing geodesics are smooth. Both Eddington's and Finkelstein's original papers consider only the outgoing set. In our work, we will find it useful to concentrate on the ingoing set of coordinates, which we refer to as 'IEF' for short.

As an aside, it should be noted that both authors constructed their coordinates in terms of a temporal and spatial coordinate system, rather than an explictly null system. There has been a slight amount of confusion on this issue within the numerical relativity community, possibly because of the ubiquity of the text by Misner, Thorne, and Wheeler [53], in which they define the Eddington-Finklestein coordinates using null coordinates. We merely point out here that the two descriptions, in terms of "t" and "r" coordinates or "u" and "v", are completely equivalent, and as we will see shortly, the ingoing or outgoing Eddington-Finkelstein solutions *are* equivalent to the Schwarzschild solution in Kerr-Schild coordinates.

Starting from the Schwarzschild metric in Schwarzschild coordinates,

$$ds^2 = -\left(1 - \frac{2M}{r}\right)\mathbf{d}t^2 + \left(1 - \frac{2M}{r}\right)^{-1}\mathbf{d}r^2 + r^2 d\Omega^2, \tag{B.9}$$

they both (in slightly different notation), introduce a coordinate transformation from the Schwarzschild time coordinate t, to a new coordinate $\bar{t}$, defined by

$$\bar{t} = t \pm 2M \ln\left|\frac{r - 2M}{2M}\right|, \tag{B.10}$$

where the positive sign above will lead to the ingoing form (IEF), and the negative sign will lead to the outgoing form (OEF). In Fig. B.1, we see what effect this transformation has on radial null geodesics. For the OEF solution in panel c., outgoing null geodesics follow 45° lines and for the IEF solution in panel b., ingoing null geodesics lie on -45° lines.

Under this transformation, we see that the Schwarzschild metric in IEF coordinates takes the form

$$ds^2 = -\left(1 - \frac{2M}{r}\right)\mathbf{d}\bar{t}^2 + \frac{4M}{r}\mathbf{d}\bar{t}\,\mathbf{d}r + \left(1 + \frac{2M}{r}\right)\mathbf{d}r^2 + r^2 d\Omega^2, \tag{B.11}$$

and, in a Cartesian basis, we have

$$ds^2 = -\left(1 - \frac{2M}{r}\right)\mathbf{d}\bar{t}^2 + \frac{4M}{r}\frac{x_i}{r}\mathbf{d}\bar{t}\,\mathbf{d}x^i + \left(\delta_{ij} + \frac{2M}{r}\frac{x_i}{r}\frac{x_j}{r}\right)\mathbf{d}x^i\mathbf{d}x^j. \tag{B.12}$$

Comparing with the Kerr-Schild metric, Eq. B.1, it is very easy to see that the IEF

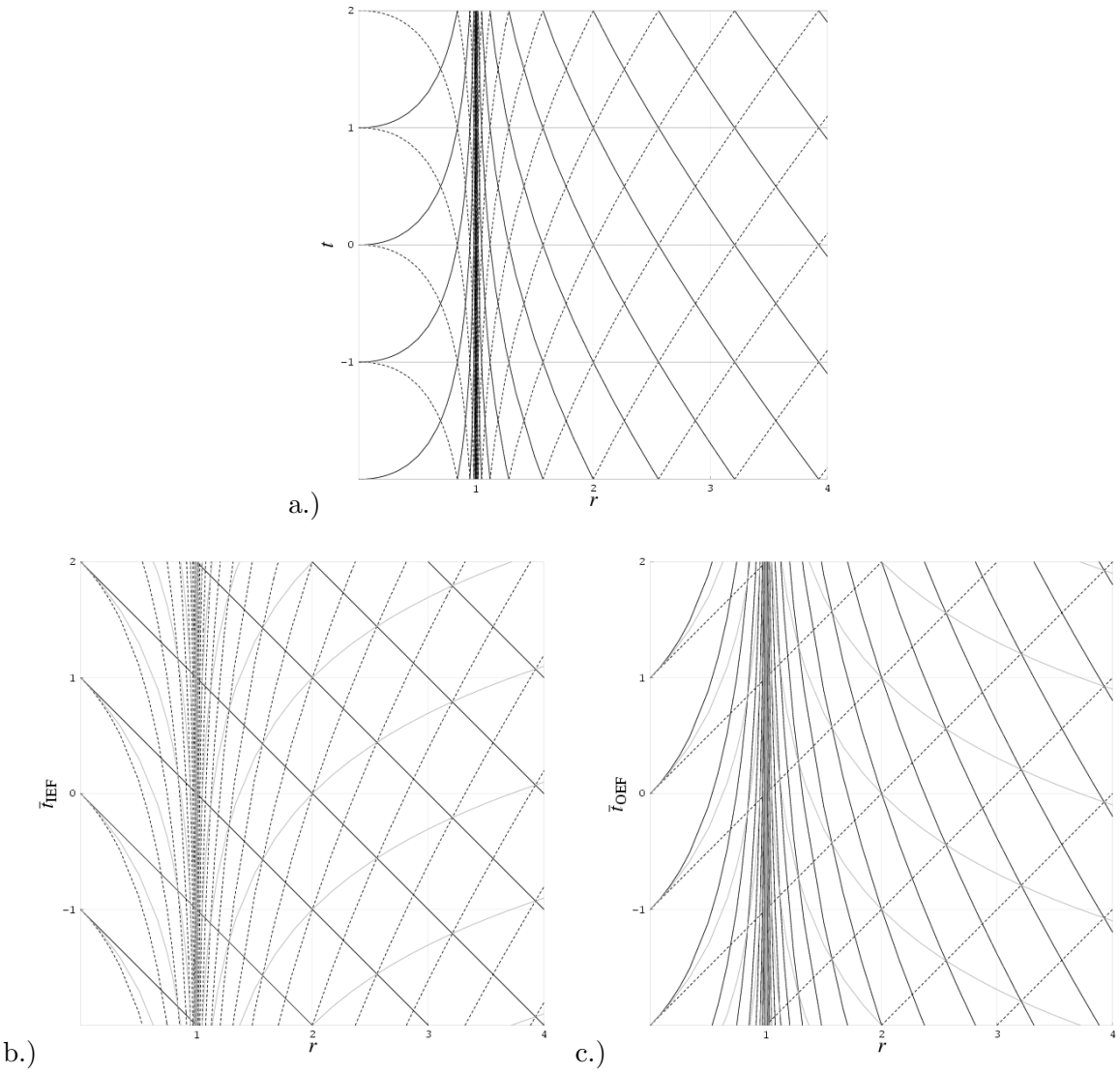
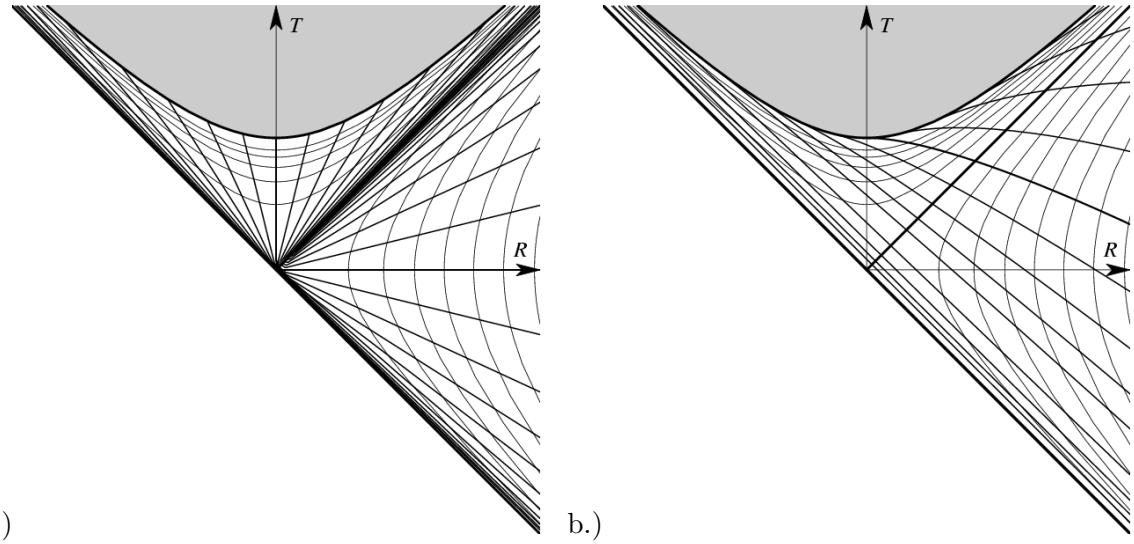


Figure B.1: Comparison of radial null geodesics in a.) Schwarzschild coordinates, b.) Ingoing Eddington-Finkelstein (IEF), and c.) Outgoing Eddington-Finkelstein (OEF). The dark solid and dashed lines represent, respectively, ingoing and outgoing null geodesics. The light, solid lines represent lines of constant Schwarzschild time and radial coordinates t and r. In these plots, a mass of $M = 0.5$ has been assumed, such that the Schwarzschild radius $r = 2M$ lies at $r = 1$ in all three figures.



a.) b.)

Figure B.2: Illustration of constant-time-coordinate foliation of non-maximally extended Schwarzschild spacetime in a.) Schwarzschild coordinates and b.) Ingoing Eddington-Finkelstein (IEF). The lines of constant radial coordinates (hyperbolae $R^2 - T^2 = f(r)$) are the same in both figures, as well as the horizons (bold, diagonal lines). In a.) lines of constant time all meet at the center of the figure, and bend discontinuously across the horizon, such that the timelike and spacelike translation vectors trade roles. In b.) lines of constant time smoothly penetrate the horizon.

metric is, in fact, written in the Kerr-Schild form, where we read off

$$H = \frac{M}{r} \quad \text{with} \quad r^2 = x^2 + y^2 + z^2, \quad \text{and} \tag{B.13}$$

$$[\ell_\mu] = \left(1, \frac{x}{r}, \frac{y}{r}, \frac{z}{r}\right). \tag{B.14}$$

The major advantage that these coordinates have over many of the others used to describe the Schwarzschild solution is all of the "trouble" is concentrated at the origin, $r = 0$. This solution has no coordinate singularities except for the one at the origin, which is actually the location of the *physical* singularity. Of course, this set of coordinates is not without its "quirks". For one, outgoing null geodesics pile up at $\bar{t} = -\infty$, as seen in Fig. B.1. Also, the slices defined by using the Kerr-Schild lapse and shift could hardly be worse at avoiding the singularity — every spatial slice hits the singularity at $\bar{r} = 0$, as seen in Fig. B.2, where the IEF coordinates are mapped to a Kruskal-Szekeres diagram. Of course, we are largely interested in this system in the context of singularity excision. As can be seen from this figure, singularity excision will be necessary for *every* time-slice in the IEF foliation.

For the sake of completeness, we note that [53] records the following as the IEF metric,

$$ds^2 = -\left(1 - \frac{2M}{r}\right)\mathbf{d}\tilde{V}^2 + 2\mathbf{d}\tilde{V}\mathbf{d}r + r^2 d\Omega^2, \tag{B.15}$$

which can be derived from Eq. (B.11) by introducing the new null coordinate $\tilde{V} \equiv t + r$.

B.4 3+1 Decomposition of IEF

To conclude, we record here the 3+1 decomposition of the IEF solution in Cartesian coordinates:

$$\alpha = \left(1 + \frac{2M}{r}\right)^{-1/2} \tag{B.16}$$

$$\beta^i = \left(1 + \frac{2M}{r}\right)^{-1} \frac{2Mx^i}{r^2} \tag{B.17}$$

$$\gamma_{ij} = \delta_{ij} + \frac{2Mx_i x_j}{r^3} \tag{B.18}$$

$$K_{ij} = \left(\frac{2M}{r^2}\right)\left(1 + \frac{2M}{r}\right)^{-1/2}\left[\delta_{ij} - \frac{x_i x_j}{r^3}(2r + M)\right] \tag{B.19}$$

References

[1] Abrahams, A., D. Bernstein, D. Hobill, E. Seidel, and L. Smarr, Phys. Rev. D, **45**, 3544 (1992).

[2] Alcubierre, M., Class. Quant. Grav., **20**, 607 (2003), gr-qc/0210050.

[3] Alcubierre, M., G. Allen, B. Brügmann, E. Seidel, and W.-M. Suen, Phys. Rev. D, **62**, 124011 (2000), gr-qc/9908079.

[4] Alcubierre, M. and B. Brügmann, Phys. Rev. D, **63**, 104006 (2001), gr-qc/0008067.

[5] Alcubierre, M., B. Brügmann, P. Diener, M. Koppitz, D. Pollney, E. Seidel, and R. Takahashi, Phys. Rev. D, **67**, 084023 (2003), gr-qc/0206072.

[6] Alcubierre, M., B. Brügmann, D. Pollney, E. Seidel, and R. Takahashi, Phys. Rev. D, **64**, 61501 (2001), gr-qc/0104020.

[7] Alcubierre, M. and B. Schutz, J. Comput. Phys., **112**, 44 (1994), physics/0009029.

[8] Anninos, P., G. Daues, J. Massó, E. Seidel, and W. Suen, Phys. Rev. D, **51**, 5562 (1995).

[9] Anninos, P., D. Hobill, E. Seidel, L. Smarr, and W. Suen, Phys. Rev. Lett., **71**, 2851 (1993).

[10] Anninos, P., D. Hobill, E. Seidel, L. Smarr, and W. Suen, Phys. Rev. D, **52**, 2044 (1995).

[11] Anninos, P., J. Massó, E. Seidel, W.-M. Suen, and J. Towns, Phys. Rev. D, **52**, 2059 (1995).

[12] Arnowitt, R., S. Deser, and C. W. Misner, in *Gravitation: An Introduction to Current Research*, edited by L. Witten (Wiley, New York, 1962), pp. 227–265.

[13] Baumgarte, T. W. and S. L. Shapiro, Phys. Rev. D, **59**, 024007 (1999), gr-qc/9810065.

[14] Baumgarte, T. W. and S. L. Shapiro, Phys. Rept., **376**, 41 (2003), gr-qc/0211028.

[15] Beig, R. and N. Ó Murchadha, Phys. Rev. D, **57**, 4278 (1998).

[16] Berger, B. K., Living Rev. Relativity, **5**, 1 (2002), [Article in Online Journal] cited on 16 Feb 2004, http://www.livingreviews.org/Articles/lrr-2002-1/.

[17] Bernstein, D., D. Hobill, E. Seidel, L. Smarr, and J. Towns, Phys. Rev. D, **50**, 5000 (1994).

[18] Bernstein, D. H., D. W. Hobill, and L. L. Smarr, in *Frontiers in Numerical Relativity*, edited by C. R. Evans, L. S. Finn, and D. W. Hobill (Cambridge University Press, Cambridge, 1989), pp. 57–73.

[19] Bona, C. and J. Massó, Phys. Rev. D, **38**, 2419 (1988).

[20] Bona, C., J. Massó, E. Seidel, and J. Stela, Phys. Rev. Lett., **75**, 600 (1995), gr-qc/9412071.

[21] Bowen, J. M. and J. James W. York, Phys. Rev. D, **21**, 2047 (1980).

[22] Brandt, S. and B. Brügmann, Phys. Rev. Lett., **78**, 3606 (1997), gr-qc/9703066.

[23] Brandt, S. *et al.*, Phys. Rev. Lett., **85**, 5496 (2000), gr-qc/0009047.

[24] Brill, D. R. and R. W. Lindquist, Phys. Rev., **131**, 471 (1963).

[25] Brügmann, B., Phys. Rev. D, **54**, 7361 (1996), gr-qc/9608050.

[26] Brügmann, B., Int. J. Mod. Phys. D, **8**, 85 (1999), gr-qc/9708035.

[27] Brügmann, B., W. Tichy, and N. Jansen, *Numerical simulation of orbiting black holes* (2003), gr-qc/0312112.

[28] Burden, R. L. and J. D. Faires, *Numerical Analysis* (Brooks/Cole Publishing Co., Pacific Grove, CA, 1997), sixth edn.

[29] Cactus, http://www.cactuscode.org.

[30] Čadež, A., *Colliding Black Holes*, Ph.D. thesis, University of North Carolina at Chapel Hill (1971), (unpublished).

[31] Calabrese, G., L. Lehner, D. Neilsen, J. Pullin, O. Reula, O. Sarbach, and M. Tiglio, Class. Quant. Grav., **20**, L245 (2003), gr-qc/0302072.

[32] Calabrese, G. and D. Neilsen, *Spherical excision for moving black holes and summation by parts for axisymmetric systems* (2003), [preprint] cited on 23 Feb 2004, gr-qc/0308008.

[33] Choptuik, M. W. (1998), lecture notes written by M. W. Choptuik for an advanced GR course taught at U.T. Austin in spring of 1998. Notes available online at http://laplace.physics.ubc.ca/People/matt/Teaching/98Spring/Phy387N/Doc/3+1.ps.

[34] Cook, G. *et al.*, Phys. Rev. Lett., **80**, 2512 (1998), gr-qc/9711078.

[35] Cook, G. B., Living Rev. Relativity, **3**, 5 (2000), [Article in Online Journal] cited on 16 Feb 2004, http://www.livingreviews.org/Articles/lrr-2000-5/.

[36] Cook, G. B., M. W. Choptuik, M. R. Dubal, S. Klasky, R. A. Matzner, and S. R. Oliveira, Phys. Rev. D, **47**, 1471 (1993).

[37] Eardley, D. M. and L. Smarr, Phys. Rev. D, **19**, 2239 (1979).

[38] Eddington, A. S., Nature, **113**, 192 (1924).

[39] Einstein, A. and N. Rosen, Phys. Rev., **48**, 73 (1935).

[40] Eppley, K. R., *The numerical evolution of the collision of two black holes*, Ph.D. thesis, Princeton University (1975), (unpublished).

[41] Estabrook, F., H. Wahlquist, S. Christensen, B. DeWitt, L. Smarr, and E. Tsiang, Phys. Rev. D, **7**, 2814 (1973).

[42] Finkelstein, D., Phys. Rev., **110**, 965 (1958).

[43] Gentle, A. P., N. D. George, A. Kheyfets, and W. A. Miller, Class. Quant. Grav., **21**, 83 (2004), gr-qc/0307007.

[44] Gómez, R., L. Lehner, R. L. Marsa, and J. Winicour, Phys. Rev. D, **57**, 4778 (1998).

[45] Hawking, S. and G. Ellis, *The Large Scale Structure of Space-time* (Cambridge University Press, Cambridge, 1973).

[46] Kelly, B. J., *The Next Generation of Binary Black Hole Head-On Collisions and their Aftermath*, Ph.D. thesis, The Pennsylvania State University (2004), (unpublished).

[47] Kerr, R. P. and A. Schild, in *Applications of Nonlinear Partial Differential Equations in Mathematical Physics, Proc. of Symposia B Applied Math.* (American Mathematical Society, 1965), vol. 17.

[48] Khokhlov, A. M. and I. D. Novikov, Class. Quant. Grav., **19**, 827 (2002), gr-qc/0111023.

[49] Knapp, A. M., E. J. Walker, and T. W. Baumgarte, Phys. Rev. D, **65**, 06403 (2002), gr-qc/0201051.

[50] Laguna, P. and D. Shoemaker, Class. Quant. Grav., **9**, 3679 (2002), gr-qc/0202105.

[51] Lehner, L., M. Huq, and D. Garrison, Phys. Rev. D, **62**, 084016 (2000).

[52] Lindblom, L. and M. A. Scheel, Phys. Rev. D, **67**, 124005 (2003), gr-qc/0301120.

[53] Misner, C., K. Thorne, and J. A. Wheeler, *Gravitation* (W. H. Freeman, New York, 1973).

[54] Misner, C. W., Phys. Rev., **118**, 1110 (1960).

[55] Press, W. H., S. A. Teukolsky, V. W. T., and F. B. P., *Numerical Recipes in Fortran* (Cambridge University Press, Cambridge, 1992), second edn.

[56] Reimann, B. and B. Brügmann, Phys. Rev. D, **69**, 044006 (2004), gr-qc/0307036.

[57] Reula, O. A., Living Rev. Relativity, **1**, 3 (1998), [Article in Online Journal] cited on 16 Feb 2004, http://www.livingreviews.org/Articles/lrr-1998-3/.

[58] Sarbach, O., G. Calabrese, J. Pullin, and M. Tiglio, Phys. Rev. D, **66**, 064002 (2002), gr-qc/0205064.

[59] Scheel, M. A., T. W. Baumgarte, G. B. Cook, S. L. Shapiro, and S. A. Teukolsky, Phys. Rev. D, **56**, 6320 (1997), gr-qc/9708067.

[60] Schiesser, W., *The Numerical Method of Lines* (Academic Press, Inc., San Diego, 1991).

[61] Schnetter, E., Class. Quant. Grav., **20**, 4719 (2003), gr-qc/0306006.

[62] Schnetter, E., S. H. Hawley, and I. Hawke, *Evolutions in 3d numerical relativity using fixed mesh refinement* (2003), [preprint] cited on 24 Feb 2004, gr-qc/0310042.

[63] Seidel, E. and W.-M. Suen, Phys. Rev. Lett., **69**, 1845 (1992).

[64] Shibata, M., Progress of Theoretical Physics, **101**, 251 (1999).

[65] Shibata, M. and T. Nakamura, Phys. Rev. D, **52**, 5428 (1995).

[66] Shoemaker, D., K. Smith, U. Sperhake, P. Laguna, E. Schnetter, and D. Fiske, Class. Quant. Grav., **20**, 3729 (2003), gr-qc/0301111.

[67] Shoemaker, D. M., *Apparent Horizons in Binary Black Hole Spacetimes*, Ph.D. thesis, University of Texas at Austin (1999), (unpublished).

[68] Smarr, L., A. Čadež, B. DeWitt, and K. Eppley, Phys. Rev. D, **14**, 2443 (1976).

[69] Smarr, L. L., *The structure of general relativity with a numerical illustration: The collision of two black holes*, Ph.D. thesis, University of Texas at Austin (1975), (unpublished).

[70] Smarr, L. L., in *Sources of Gravitational Radiation*, edited by L. L. Smarr (Cambridge University Press, Cambridge, 1979), pp. 245–274.

[71] Sperhake, U., K. L. Smith, P. Laguna, B. Kelly, and D. Shoemaker, Phys. Rev. D, **69**, 024012 (2004), gr-qc/0307015.

[72] Teukolsky, S. A., Phys. Rev. D, **61**, 087501 (2000).

[73] Thornburg, J., Class. Quant. Grav., **4**, 1119 (1987).

[74] Thornburg, J., *A 3+1 computational scheme for dynamic spherically symmetric black hole spacetimes — ii: Time evolution* (1999), [preprint] cited on 23 Feb 2004, gr-qc/9906022.

[75] Thornburg, J., Class. Quant. Grav., **21**, 743 (2004), gr-qc/0306056.

[76] Tiglio, M., L. Lehner, and D. Neilsen, *3d simulations of einstein's equations: symmetric hyperbolicity, live gauges and dynamic control of the constraints* (2003), [preprint] cited on 24 Feb 2004, gr-qc/0312001.

[77] Wald, R. M., *General Relativity* (The University of Chicago Press, Chicago, 1984).

[78] Yo, H. J., T. W. Baumgarte, and S. L. Shapiro, Phys. Rev. D, **64**, 124011 (2001), gr-qc/0109032.

[79] Yo, H.-J., T. W. Baumgarte, and S. L. Shapiro, Phys. Rev. D, **66**, 084026 (2002), gr-qc/0209066.

[80] York, J., J.W., in *Sources of Gravitational Radiation*, edited by L. Smarr (Cambridge University Press, Cambridge, 1979), pp. 83–126.

Vita

Kenneth L. Smith was born in Aurora, IL on November 27, 1978. He is the eldest of two sons of Terrell and Charlene Smith. In May of 2000, he received his B.S. degree, *summa cum laude*, from Bradley University. A mere matter of weeks later, he moved to State College, Pennsylvania to begin working with the Numerical Relativity group there to get a head-start on research. In the fall of 2000, he officially began his studies in the Ph.D. program in Physics at the Pennsylvania State University, University Park campus. He was employed by the Physics Department his first year as a graduate teaching assistant, and received the Department's Excellent in Teaching TA award in 2001. In 2003, the Department nominated him for the American Association of Physics Teachers' Outstanding TA award. In March of 2003, he was married to Amanda Callision, also a Bradley alum, in a small, private ceremony in Sheboygan, Wisconsin.

Kenneth L. Smith is a member of the American Association of Physics Teachers and the American Physical Society. His publications include:

M. Fitzpatrick, K. Smith, D.W. Belousek, A. Delgado, K.R. Roos and J.P. Kenny, The Quantum Cellular Automaton as a Markov Process, *Chaos, Solitons & Fractals* **8** 1375–1386 (1999).

D. Shoemaker, K. Smith, U. Sperhake, P. Laguna, E. Schnetter, and D. Fiske, Moving black holes via singularity excision, *Class. Quantum Grav.* **20** 3729–3743 (2003).

U. Sperhake, K.L. Smith, B. Kelly, P. Laguna, and D. Shoemaker, Impact of densitized lapse slicings on evolutions of a wobbling black hole, *Phys. Rev. D* **69**, 024012 (2004).