Numerical Simulations of Gravitational Collapse

by

Frans Pretorius

B.Eng., The University of Victoria, 1996
M.Sc., The University of Victoria, 1999

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Department of Physics and Astronomy)

We accept this thesis as conforming
to the required standard

....

....

....

....

THE UNIVERSITY OF BRITISH COLUMBIA

August 23, 2002

ABSTRACT

In this thesis we present a numerical study of gravitational collapse, within the framework of Einstein's theory of general relativity. We restrict our attention to spacetimes possessing axial symmetry, and incorporate a massless scalar field as the matter source.

Our primary objectives are the study of critical phenomena at the threshold of black hole formation, and the stable simulation of black hole spacetimes. An integral part of the thesis is concerned with developing the necessary numerical tools and techniques to successfully solve these problems. To that end, we have implemented a variant of Berger and Oliger's adaptive mesh refinement (AMR) algorithm, with enhancements that allow us to incorporate elliptic equations into the AMR framework. Using this code, we simulate critical collapse of axisymmetric distributions of scalar field energy, which is the first time this has been done in the non-perturbative regime. For several classes of initial data, our results are consistent with a hypothesized universal critical solution. However, from the collapse of prolate initial data, we find indications that there may be an additional, non-spherical unstable mode. This putative instability eventually causes a near-critical echoing solution to bifurcate into *two*, causally disconnected solutions that each resemble the spherical critical solution. Furthermore, we speculate that this bifurcation process would continue indefinitely at threshold, resulting in an infinite cascade of near-spherical solutions. However, the evidence for this second unstable mode is not conclusive, and more work will be needed, possibly with an enhanced code, to answer this question.

To numerically study spacetimes containing black holes, one needs to avoid the singularities that occur inside of the holes. The technique that we have implemented to accomplish this is called *black hole excision*. This aspect of the code is still work-in-progress, for we have not yet incorporated excision into the AMR-based code, and the class of excision boundary conditions we currently employ are inconsistent with the complete set of field equations. However, we are able to obtain stable simulations using a constrained evolution scheme, and we present two preliminary examples, one showing black hole formation, the other a head-on black hole collision.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to express my gratitude to Matthew Choptuik, my research advisor, for his guidance and wisdom throughout this project. I would also like to thank: my collaborators in the research, Eric Hirschmann and Steve Liebling (in addition to Matthew Choptuik), who initiated this study of axisymmetric collapse and graciously allowed me to join the project; the additional members of my supervisory committee—Kristin Schleich, William Unruh and Jim Varah —for many helpful comments; my friends and colleagues in the UBC relativity group, who helped create a pleasant and stimulating research environment; and my family, for their support and encouragement. Additional thanks go to Inaki Olabarrieta for helping to check several chapters of this manuscript. The majority of simulations used to obtain results presented in this thesis were run on the **vn** computer cluster at the University of British Columbia and the **MACI** cluster at the University of Calgary. **vn** is funded by the Canadian Foundation for Innovation (CFI) and the BC Knowledge Development Fund. **MACI** is funded by the Universities of Alberta, Calgary, Lethbridge and Manitoba, and by C3.ca, the Netera Alliance, CANARIE, the Alberta Science and Research Authority, and the CFI. Finally, I would like to acknowledge the National Research Council of Canada and The Izaak Walton Killam Fund for their financial support.

CHAPTER 1

INTRODUCTION

Einstein's theory of general relativity is a theory about the nature of space and time. Space and time, or spacetime for short, is the arena in which we live, and is a structure of sufficient complexity to enable us to quantify notions of distance and time. General relativity describes this structure of spacetime as a four dimensional manifold, endowed with a Lorentzian geometry. A Lorentzian geometry is one with an indefinite metric, allowing one to classify the distance interval between two events in spacetime as timelike (negative distance-squared), spacelike (positive distance-squared) or lightlike (zero distance). Furthermore, general relativity predicts that the particular geometry in which we live is governed by the matter content of the universe, and conversely that matter, in effect, "experiences" the geometry as the force of gravity.

One of the remarkable predictions of general relativity is the phenomena of gravitational collapse to a black hole, whereby a local region of spacetime—the black hole—becomes causally disconnected from the rest of the universe. This means that observers interior to the black hole cease to be able to communicate with those outside of it. The inside of a black hole is intrinsically dynamical, and singularity theorems [1] prove that some sort of geometric pathology (singularity) must occur inside of the black hole. There is rather strong circumstantial evidence that black holes are common astrophysical objects—"supermassive" black holes are thought to exist in the centers of most galaxies [2], and it is believed that a certain fraction of stars with initial masses greater than approximately 10 solar masses collapse to black holes in the final stage of their evolution [3]. Furthermore, recent studies of the collapse of simple matter fields [1] have found a surprisingly rich set of phenomena, dubbed *critical phenomena*, at the threshold of black hole formation [4]. A distribution of energy near the threshold of black hole formation is one for which a small perturbation could either cause the energy to collapse and form a black hole, or evolve to a different, stable configuration not containing a black hole (for instance, the energy could disperse to infinity). The two characteristic features of critical phenomena are the apparent *universality* of the critical solution, where the *same* critical solution is observed regardless of the configuration of the initial distribution of energy, and *power-law scaling* of length scales that arise in near-critical solutions. In Sections 1.1 - 1.4 of this thesis we will give a more detailed introduction to general relativity, black holes, and critical phenomena.

The primary goal of the work described in this thesis is to develop a numerical code that can solve Einstein's field equations with axisymmetry, and apply it to a wide range of problems in black hole physics. The equations of general relativity are too complicated to hope for analytical solutions in many situations of interest, hence numerical solutions are *essential* to gain more insight into the nature of the theory. Aside from theoretical interest, improving our understanding of the predictions of general relativity is becoming increasingly important in describing many astrophysical processes in our universe. In the near future, gravitational wave "observatories", including LIGO, VIRGO, GEO, TAMA and AIGO [5, 6, 7, 8, 9] will begin to operate, and if there is any hope to gain information from signals that may be detected, one needs to have a better understanding of dynamical, strong-field gravity than what we currently have.

There are two main reasons why we are developing an axisymmetric code. First, the imposed symmetry is not too restrictive, in the sense that we can still simulate numerous phenomena of theoretical and astrophysical interest, including gravitational collapse and critical phenomena, black hole accretion disks and head-on black hole collisions. Second, todays supercomputers, let

[1] Such as scalar fields, perfect fluids, and pure gravitational waves—in other words not the "complicated" forms of matter that would be involved collapse in an astrophysical setting.

alone the computer systems that we have ready access to, are not powerful enough to solve the full equations in 3D (three spatial dimensions) with a reasonable level of accuracy. Therefore, by restricting our attention to a 2D system, we can solve a class of problems with relatively high accuracy, even on modest computers.

In addition, there are many unresolved issues in numerical relativity, including questions of the stability of various formulations of the field equations, coordinate choices and boundary conditions. Also, to-date there has not been enough exploration of numerical techniques and algorithms that are expected to be essential in obtaining accurate solutions. Most notable, perhaps, is the relative small number of finite-difference based codes incorporating adaptive mesh refinement (AMR) techniques—following the pioneering work of Choptuik [4], there have been few codes that have used AMR [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]. By focusing on the less complicated two dimensional case, we can hope to more rapidly address these issues.

The axisymmetric code that we are building is based upon a finite-difference solution of the field equations. We use the 2+1+1 formalism [21] to arrive at the set of differential equations that need to be solved. This formalism applies the common ADM [22] space-plus-time decomposition to a dimensionally reduced spacetime, obtained by dividing out the axial Killing vector[23]. In Section 1.5 we review the ADM formalism, and relevant aspects of finite-difference based solutions of partial differential equations, including the solution of elliptic equations via multigrid, and the solution of hyperbolic equations using AMR. In Chapter 2, we describe the 2+1+1 formalism, and details of the unigrid and AMR-capable codes that we have developed to solve the corresponding equations. In the current version of the code we have not incorporated the effects of angular momentum.

The first physical problem that we are applying the AMR code to is the critical collapse of a massless scalar field (Klein-Gordon field). To our knowledge, this is the first non-perturbative study of critical behavior of the Einstein-Klein-Gordon system away from spherical symmetry. Choptuik conjectured that critical behavior is universal, and should be visible at the threshold of black hole formation in a large class of generic families of initial data [4]. An essentially equivalent statement of this conjecture is that the critical solution is *one-mode unstable* [24, 25]; in other words, all perturbations of a critical solution decay with time, except for a single, exponentially growing mode, whose initial amplitude can be fine-tuned to zero by seeking the threshold solution. This idea has been successfully applied in several situations to predict scaling exponents, and pertubative studies of the Einstein-Klein-Gordon critical solution have found only a single growing mode [26, 27]. Our study of this system is presented in Chapter 3. For the most part, our results are consistent with the conjectured universality of the critical solution. However, we do see some indications that there may in fact be *an additional unstable mode* [2]. If so, the growth rate of the instability is small enough that it does not appreciably alter observed near-critical solutions at early times; though at late times, the effect of the putative unstable mode is such that, tuning closer to criticality, *two* black holes form along the axis instead of one (from slightly super-critical initial data). On the other hand, certain families of initial data, with significant asymmetries, form two black holes while still far enough away from the threshold that critical behavior is not observed about a single center of symmetry, rather, two distinct, spherical-like critical solutions develop. This can be thought of as arising from a "focusing" effect, whereby the initial distribution collapses to two local centers. Therefore, it may be that the apparent instability we see in some cases is actually arising from a small focusing effect.

Another class of axisymmetric spacetimes we would like study are those where black holes play a significant role in the dynamics, including head-on black hole collisions, and interactions of matter or gravitational waves with black holes. To this end, we need to deal with the singularities inside of black holes, and we have chosen to use *black hole excision* [28] to obtain long-term, stable evolution of these spacetimes. The details of excision are explained in Sections 1.5.3 and 2.2.6. At this stage,

[2]In terms of the spherical-harmonic decomposition used in [27], it appears as if an $\ell = 2$ mode is unstable; in the perturbative calculation such a mode was one with the slowest rate of decay.

we have not yet added excision to the AMR code, and the set of boundary conditions we apply are inconsistent with some of the evolution equations which are not used in our finite difference algorithm. Hence, in Chapter 4 we only present a few examples of spacetimes with excision, to demonstrate the feasibility of our method. Before attempting to extract meaningful physics from such scenarios, we will need to restrict the class of boundary conditions to achieve consistency. Furthermore, AMR will be essential to obtain accurate results, in particular to study black hole collisions and obtain good estimates of the emitted gravitational wave forms.

In Chapter 5, we conclude by mentioning a few of the extensions to the code that we hope to incorporate in the not-too-distant future. These include additional matter fields, and support for spacetimes with angular momentum.

In appendix A, we summarize the set of continuum equations we solve, and list the particular finite difference stencils we employ in the numerical code. In appendix B, we describe a piece of software that we have developed, called the *Data-Vault* (DV), that was specifically designed to help develop AMR-based finite difference codes. The DV is a data repository with visualization facilities, allowing us to analyze and view the kind of AMR grid-hierarchies our code produces. This tool has been extremely valuable in developing the code, for to efficiently deal with instabilities (and bugs) it is imperative to be able to "see" exactly what is happening at each stage in the algorithm.

1.0.1 Notation

We use a metric signature of $-+++$. For tensor indices, we use the Einstein summation convention, whereby repeated indices are summed over. Greek indices $(\alpha, \beta, \delta, ...)$ are used with four-dimensional (4D) tensors, lower-case Latin indices $(a, b, c, ...)$ with three-dimensional (3D) tensors, and upper-case Latin indices $(A, B, C, ...)$ are used with two-dimensional (2D) tensors. In equations where we project from an $(n+1)$ to an n-dimensional manifold, we often mix the Latin and Greek indices in the projection tensor, to denote the fact that the result is an intrinsically n-dimensional object. 4D covariant derivatives are denoted by a semi-colon (;) or ∇, and 3D covariant derivatives by either a vertical bar (|) or the operator D. Ordinary differentiation is often denoted by a comma (,). We use $\mathcal{L}_\xi$ to denote the Lie derivative [29] along a vector field ξ^α (or ξ^a). Square brackets enclosing tensor indices denote anti-symmetrization of the set of indices, for example $\xi_{[\alpha,\beta]} = \frac{1}{2}(\xi_{\alpha,\beta} - \xi_{\beta,\alpha})$. Throughout this thesis we use units where Newton's constant G and the speed of light c are set to 1.

1.1 The Field Equations of General Relativity

General relativity is Einstein's theory of the geometric structure of space and time. This geometry is conveniently encoded via the *metric tensor* $g_{\mu\nu}$, defined by

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu, \tag{1.1}$$

where ds is the infinitesimal distance measure between adjacent points in the spacetime, in a direction given by the vector dx^μ. The contravariant form $g^{\mu\nu}$ of the metric tensor is defined to be the inverse of the covariant form $g_{\mu\nu}$:

$$g_{\mu\gamma} g^{\gamma\nu} = \delta_\mu{}^\nu, \tag{1.2}$$

where $\delta_\mu{}^\nu$ is the Kronecker delta function. From this we can define the raising and lowering of tensor indices via contraction with the metric tensor; for example

$$T_\mu = T^\nu g_{\mu\nu}. \tag{1.3}$$

$g_{\mu\nu}$ is required to be a symmetric tensor with vanishing covariant derivative, i.e. $g_{\mu\nu;\gamma} = 0$. This gives the following form for the *metric connection* $\Gamma^\delta{}_{\mu\nu}$ (also called Christoffel symbols of the second kind):

$$\Gamma^{\delta}{}_{\mu\nu} = \frac{1}{2} g^{\delta\gamma} \left[g_{\nu\gamma,\mu} + g_{\gamma\mu,\nu} - g_{\mu\nu,\gamma} \right]. \tag{1.4}$$

Note that the $\Gamma^{\delta}{}_{\mu\nu}$ do not obey the usual tensor transformation properties, and thus Γ is strictly not a tensor.

A geometric object describing curvature in the spacetime, and of relevance to the Einstein equations, is the *Riemann tensor*, defined by

$$R_{\mu\nu\gamma\delta} V^{\delta} = V_{\gamma;\nu\mu} - V_{\gamma;\mu\nu} \ , \tag{1.5}$$

where V^{δ} is an *arbitrary* vector. Written out explicitly in terms of the connection, the Riemann tensor is

$$R_{\mu\nu\gamma}{}^{\delta} = \Gamma^{\delta}{}_{\mu\gamma,\nu} - \Gamma^{\delta}{}_{\nu\gamma,\mu} + \Gamma^{\rho}{}_{\mu\gamma}\Gamma^{\delta}{}_{\rho\nu} - \Gamma^{\rho}{}_{\nu\gamma}\Gamma^{\delta}{}_{\rho\mu} \tag{1.6}$$

A contraction over the second and fourth indices of the Riemann tensor gives the Ricci tensor

$$R_{\mu\nu} = R_{\mu\gamma\nu}{}^{\gamma}, \tag{1.7}$$

a further contraction of which gives the *Ricci scalar*

$$R = R_{\mu}{}^{\mu}. \tag{1.8}$$

The *Einstein tensor* is defined as

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu}. \tag{1.9}$$

The *Einstein field equations* describe the set of geometries that are believed to be relevant in nature, and are

$$G_{\mu\nu} = \frac{8\pi G}{c^5} T_{\mu\nu}, \tag{1.10}$$

where for the moment we have restored conventional units, so that G is Newton's constant and c is the speed of light in vacuum. $T_{\mu\nu}$ is the *stress-energy-momentum tensor* (or stress-energy tensor for short) of matter fields existing in the spacetime. Thus matter is responsible for the geometric structure of spacetime. In turn, the geometry of spacetime is "experienced" by matter as the force of gravity. The latter statement is in part quantified by the *geodesic hypothesis*[3], stating that point-like objects with infinitesimal mass travel along geodesics of the underlying geometry. A geodesic is a curve $x^{\alpha}(\lambda)$ of extremal length in the spacetime, with λ being the so called affine parameter, labelling distance along the curve. The requirement that $x^{\alpha}(\lambda)$ be a curve of extremal length results in the following set of differential equations:

$$u^{\alpha}{}_{;\beta} u^{\beta} = 0, \tag{1.11}$$

where $u^{\alpha}(\lambda) \equiv dx^{\alpha}(\lambda)/d\lambda$ is the vector tangent to the curve.

The particular value, $8\pi G/c^5$, of the constant appearing on the right hand side of (1.10) arises by demanding consistency between Einstein and Newton's theories of gravity in the appropriate limits. One such limit is the region far outside a central, compact body of matter, where certain geodesics of the spacetime will coincide with Keplarian orbits about the central object. Requiring the orbital period to match in the two theories fixes the value of the constant.

Note that the field equations (1.10) are compatible with conservation of stress energy—$T^{\mu\nu}{}_{;\nu} = G^{\mu\nu}{}_{;\nu} = 0$. This property of the Einstein tensor can be derived as a consequence of the *Bianchi identities* satisfied by the Riemann tensor: $\nabla_{[\mu} R_{\nu\gamma]\delta}{}^{\rho} = 0$.

[3]The geodesic hypothesis can be "derived" for simple forms of matter, such as pressureless dust, from the conservation of stress-energy.

1.2 Gravitational Collapse

Some of the more intriguing solutions of Einstein's equations are spacetimes containing *black holes*. A black hole is a region of spacetime that has undergone *gravitational collapse*, implying that the "force" of gravity has become so strong that no object can escape from it. From a geometric point of view, a black hole is a region of spacetime from which no causal (timelike or null) curves leave, though such curves can enter the black hole from the outside. The boundary of the region that cannot causally influence the exterior is called the *event horizon*. It is believed that black holes are rather common astrophysical objects— most galaxies are thought to contain supermassive black holes [2] (on the order of 10^6 to 10^9 solar masses), and a certain fraction of stars with initial masses on the order of 10 or more solar masses are expected to collapse to black holes in the final stages of their evolution [3].

The relevant length-scale in gravitational collapse is the *Schwarzschild radius*, R_S, which is the location of the event horizon in the *Schwarzschild metric*, the solution to the vacuum field equations describing a non-rotating black hole:

$$ds^2 = -\left(1 - \frac{2m}{r}\right) dt^2 + \left(1 - \frac{2m}{r}\right)^{-1} dr^2 + r^2 d\theta^2 + r^2 \sin^2\theta d\phi^2. \tag{1.12}$$

Here m is the mass of the black hole, and the Schwarzschild radius is at $r = 2m$, where the coordinate system of (1.12) becomes singular. In general, for a black hole to form from a distribution of matter, the matter must be compacted into a region on the order of its Schwarzschild radius. The Schwarzschild radius for an object with the mass of the sun is roughly $1km$.

Another important feature of black holes in classical general relativity is that, barring "unusual" forms of matter[4], spacetime *singularities* always form inside of black holes. A singularity is a pathology in the geometric structure of the spacetime—for instance a point of infinite curvature, or a point of geodesic incompleteness (where geodesics "terminate" in finite affine time). This latter form of pathology was proven to always exist inside of black holes (the well-known singularity theorems of Hawking and Penrose, see for example [29]), and analytic and numerical studies of gravitational collapse suggest that curvature singularities are generic [30, 31]. Whether singularities can form *outside* of event horizons in astrophysically relevant situations is still an open question. Penrose hypothesized that an event horizon will always form to hide the singularity from distant observers —the *cosmic censorship hypothesis* [32]. There are two versions of the hypothesis, *weak cosmic censorship* and *strong cosmic censorship*. Weak cosmic censorship states that generic singularities are always contained inside of black holes. Strong cosmic censorship further states than no generic *time-like* singularities ever occur, so that (for instance) an observer falling into a black hole will never see the singularity before reaching it. As of yet no convincing counter examples have been found to either form of cosmic censorship, though several "non-generic" examples of naked singularity formation are known [33, 34].

Cosmic censorship is also an important issue for numerical simulations of black hole spacetimes. The causal nature of the black hole allows us to "excise" the interior singularities in a simulation [28] (we will discuss this in more detail in Sections 1.5.3 and 2.2.6). For then, as long as either version of cosmic censorship holds and singularities do not form outside of the event horizon, the solution that one obtains there will not be affected by our lack of knowledge of what happens in the vicinity of the singularities[5].

[4]Matter violating the so-called *energy conditions* [1], for example matter with negative pressure, or matter that exerts pressure without having energy density, etc...

[5]Presumably an eventual theory of quantum gravity will provide a sensible description of singularities, but until then any other method one might devise to "resolve" singularities for numerical purposes would be purely ad-hoc, and would give physically meaningless answers to the causal future of the singularity.

1.3 Critical Phenomena

Interesting and unexpected behavior occurs in self-gravitating systems at the *threshold* of black hole formation. This behavior, discovered by Choptuik in 1993 [4], was dubbed *critical phenomena*, in analogy to similar laws governing certain properties of statistical mechanical systems in phase transformations. Before describing the particular set of phenomena that occurs in gravitational collapse, it will be helpful to clarify the notion of the "threshold of formation" with the following canonical experiment.

Consider a smooth, compact distribution of scalar field energy at $t = t_0$, with an adjustable amplitude labelled by a parameter p. If we evolve this system, there are two possible end states as $t \to \infty$: for small initial amplitudes the scalar field will disperse to infinity, while sufficiently large amplitudes will induce gravitational collapse, leaving behind a black hole (with some of the scalar field escaping to infinity). This implies that there is a critical amplitude, corresponding to $p = p^\star$, such that for $p > p^\star$ black holes do form, while for $p < p^\star$ all of the scalar field escapes to infinity. $p = p^\star$ is the threshold of black hole formation, and it is in this vicinity of phase space that critical phenomena—*universality*, *scale invariance* and *power law scaling*—is seen. We will describe each of these phenomena in more detail below. Note, however, that the specific kind of critical behavior observed depends upon the matter model. Here, we focus on the massless scalar field—see [35, 36] for review articles discussing the majority of systems studied to date, including in [35] a more extensive treatment of the massless scalar field than we are giving here.

One of the more remarkable aspects of a *critical solution*—the resulting geometry and matter field distribution in the limit as $p \to p^\star$ in the spacetime vicinity of the collapse— is its apparent universal nature. In other words, regardless of which particular one parameter (p) family of initial data is used to interpolate between dispersal and black hole formation[6], the resulting solution in the limit as $p \to p^\star$ is, up to certain trivial rescalings, *unique*. Moreover, in the vicinity of the critical solution in phase space ($|p - p^\star| \ll 1$), the deviation in the solution away from the critical one is uniquely specified by the number $p - p^\star$ (we explain this in more detail below).

Another of the features of the critical solution for the massless scalar field is that it is scale invariant. Specifically, the solution is discretely self-similar. Define a scale-invariant variable $x = r/(-t)$ and a time variable $\tau = -\ln(-t)$. Here r is a Schwarzschild-like radial coordinate (1.12) and $t < 0$ is proper time as measured by a central observer, translated so that $t = 0$ coincides with the so-called *accumulation point*, which is the time when all length scales x within the solution collapse to zero proper length at $r = 0$. Then the discretely self-similar critical solution for a given variable (the scalar field, for instance) is a function $Z^\star(x, \tau)$ that is periodic in τ: $Z^\star(x, \tau) = Z^\star(x, \tau + \Delta)$. In (r, t) coordinates, one would see $Z^\star$ as a function oscillating in time, but with frequency diverging as $1/t$, and after each cycle, which is shorter that the previous one by a factor of e^Δ, the shape of the function repeats on a scale that is smaller by a factor e^Δ. Δ is called the *echoing exponent.*

The critical solution is scale invariant, and thus has no intrinsic length scale. Near-critical solutions break this scale-invariance, introducing a length scale L into the solution that takes on the following form:

$$L \propto (p - p^\star)^\gamma. \tag{1.13}$$

For example, in a super-critical solution with $p > p^\star$ the mass M of the resulting black hole scales as

$$M \propto (p - p^\star)^\gamma. \tag{1.14}$$

Similarly, in a sub-critical solution with $p < p^\star$ the maximum value attained by the Ricci scalar (which has a dimension of $1/length^2$) during evolution is [112]

$$R_{max} \propto (p^\star - p)^{-2\gamma}. \tag{1.15}$$

This is the so called power law scaling behavior, and γ is called the *scaling exponent.* The scaling

[6]As long as the one parameter family is smooth in the vicinity of $p^\star$.

exponent is universal, although the constant of proportionality depends on the family of initial data, as well as the particular length scale considered.

From a dynamical systems point of view, the universal behavior can be explained if the critical solution is a *one-mode unstable* solution [24, 25]. This means that if one looks at linear perturbations about the critical solution $Z^\star$, and expands these perturbations as a sum of eigenfunctions with time dependence of the form $e^{\lambda\tau}$, then only a single eigenfunction will have $\lambda < 0$ and hence grow with time; the rest all decay with time. More specifically, suppose we have a near-critical solution with the following expansion:

$$Z(x,\tau;p) = Z^\star(x,\tau;p^\star) + \sum_i C_i(p-p^\star)e^{\lambda_i\tau}\xi_i(x,\tau), \tag{1.16}$$

where ξ_i is an eigenfunction periodic in τ, and each $C_i(p-p^\star)$ has the expansion $C_i(p-p^\star) = C_{i0}\cdot(p-p^\star) + O(p-p^\star)^2$ (so that $p = p^\star$ gives the critical solution). Let us label the growing mode λ_0, and suppose that at a time $\tau = \tau_0$ we specify initial data with some arbitrary perturbation, i.e., all C_i are in general non-zero. Then, as we evolve with time, all of the perturbations associated with decaying modes will eventually die out, leaving the following approximate solution at (say) $\tau = \tau_1$:

$$Z(x,\tau_1;p) - Z^\star(x,\tau_1;p^\star) \approx C_{00}\cdot(p-p^\star)e^{\lambda_0\tau_1}\xi_0(x,\tau_1), \tag{1.17}$$

where we have assumed that $p - p^\star$ is small. This explains the universality of the critical and near-critical solutions. Furthermore, we can obtain an explicit relationship between λ_0 and γ using the following argument. The perturbed critical solution posited in (1.16) will persist in time until the right hand side of (1.17) has grown to a value of order unity, at say $\tau = \tau_L$. After that, one would expect the non-linear terms to push the solution to either dispersal or black hole formation. However, at $\tau = \tau_L$, the only length-scale that has developed in the solution is τ_L, and hence this is the only scale that the subsequent solution could inherit. This gives

$$C_{00}(p-p^\star)e^{\lambda_0\tau_L}\xi_0(x,\tau_L) \approx 1 \quad \Rightarrow \tag{1.18}$$

$$\lambda_0\tau_L = \ln(p-p^\star) + \text{const}, \quad \Rightarrow \tag{1.19}$$

$$\tau_L \propto \ln(p-p^\star)^{1/\lambda_0}. \tag{1.20}$$

Comparing the last line above with (1.13), we get that $\gamma = 1/\lambda_0$. This relationship has been verified in perturbation theory in [26, 27]. Note however, that we have been a little bit sloppy in the above derivation; in particular, the fact that $\xi_0(x,\tau)$ is periodic in τ introduces a secondary scale into the problem, whose value depends upon when in the cycle of ξ_0's oscillation the amplitude of the mode grows above unity. This is a minor effect, but nevertheless introduces a small, periodic "wiggle" (with period $\Delta/(2\gamma)$) on top of the linear relationship $n\ln(L) = \gamma(p-p^\star)$, for a length scale of L^n [26, 37].

1.4 Black Hole Collisions and Gravitational Waves

Another of the intriguing predictions of general relativity is the existence of gravitational waves—geometric distortions of spacetime traveling at the speed of light. Currently a slew of gravitational wave observatories are starting to gather data or being constructed, including LIGO, VIRGO, GEO, TAMA and AIGO [5, 6, 7, 8, 9]. One of the most promising sources of gravitational radiation for the first generation of detectors is the inspiral and merger of compact objects, namely black holes and neutron stars. This thesis details work with an axisymmetric code, that can only model head-on collisions. Head-on collisions are not expected to occur frequently enough (if ever) in astrophysical situations to be of relevance to gravitational wave detection; however, due to the complexity of solving the field equations in 3D, it will still be of benefit to use a 2D code as a testing ground for issues such as adaptive mesh refinement, black hole excision, and code-parallelization. Also, some interesting new physics may emerge from the calculation. In the remainder of this section we give a brief overview of some properties of gravitational waves in general relativity.

In linearized theory, perturbing about Minkowski spacetime, one can identify two linearly independent polarizations of gravitational waves. These are the so called cross ($\times$) and plus ($+$) polarizations, and represent geometric distortions transverse to the direction of propagation. The $\times$ polarization is equivalent to the $+$ polarization after a 45° rotation about the propagation vector, and vice-versa. The distortion in the spacetime geometry can be visualized by considering the effect of the wave passing a circular ring of dust particles, initially at rest relative to each other, and positioned in the plane orthogonal to the propagation vector: the shape of the ring will cycle through the patterns *circle* $\rightarrow$ *oblate ellipse* $\rightarrow$ *circle* $\rightarrow$ *prolate ellipse* $\rightarrow$ *circle*, with the semi-major axis of the oblate and prolate ellipses oriented at 90° relative to one another. All gravitational wave detectors work by trying to measure this oscillatory change in distance induced by the passage of a wave.

In astrophysical settings, gravitational waves are produced by the motion of compact distributions of matter/energy. In terms of a multipole decomposition of the source, the leading order contribution to emitted waves is proportional to a time varying quadrupole moment [7]. Conservation of the energy and linear momentum of a stationary, isolated source implies that there cannot be any monopole or dipole radiation. The loss of energy due to the radiation of gravitational waves has implicitly been measured in the Hulse-Taylor binary pulsar [38], as the inferred energy loss due to changes in the orbital period of the system is consistent with the quadrupole formula of general relativity.

One useful characterization of the gravitational wave structure of spacetime comes from the Newman-Penrose (NP) formalism [39]. The NP formalism is a form of tetrad calculus, wherein the metric, Christoffel symbols and Riemann tensor are projected onto a complex, null tetrad. The null tetrad consists of 2 real null vectors, ℓ^μ and n^μ, a complex null vector m^μ, and its conjugate $\bar{m}^\mu$. The normalization of these vectors is chosen as follows: $\ell^\mu n_\mu = -1$ and $m^\mu \bar{m}_\mu = 1$. The particular choice of null tetrad is application-specific. In our case, we want to look at the radiation coming from an isolated source, at large distances r from it. Thus, a sensible choice of tetrad is to let l^μ and n^μ correspond to outgoing and ingoing radial null vectors respectively, while the two complex null vectors are created from two, orthogonal, unit spacelike vectors a^μ and b^μ that are tangent to the null vectors, via $m^\mu = (1/\sqrt{2})(a^\mu - ib^\mu)$.

Of particular interest to the study of gravitational waves is the decomposition of the Weyl tensor into tetrad components. The Weyl tensor $C_{\alpha\beta\gamma\delta}$ is defined as the trace free part of the Riemann tensor, and is given by

$$R_{\alpha\beta\gamma\delta} = C_{\alpha\beta\gamma\delta} - \frac{1}{2}(g_{\alpha\delta}R_{\beta\gamma} - g_{\alpha\gamma}R_{\beta\delta} + g_{\beta\gamma}R_{\alpha\delta} - g_{\beta\delta}R_{\alpha\gamma}) - (R/6)(g_{\alpha\gamma}g_{\beta\delta} - g_{\alpha\delta}g_{\beta\gamma}). \tag{1.21}$$

In other words, the Riemann tensor can be decomposed as the above combination of the Ricci tensor, Ricci scalar and Weyl tensor. In vacuum, where the Ricci tensor and scalar are both zero by Einstein's equations (1.10), the Weyl and Riemann tensors are equivalent. There are ten independent components of the Weyl tensor, which can be isolated via contraction with appropriate sets of null tetrads. The result is five complex scalars, the *Newman-Penrose scalars*, denoted by

[7]More specifically, the metric perturbations far from a slowly-moving source are proportional to the second time derivative of the trace-free quadrupole momentum tensor of the source, and the energy carried off by these waves is proportional to the square of the third time derivative of the same tensor—see for example Wald [29].

$\Psi_0, ..., \Psi_4$:

$$\begin{aligned}
\Psi_0 &= -C_{\alpha\beta\gamma\delta} n^\alpha m^\beta n^\gamma m^\delta, \\
\Psi_1 &= -C_{\alpha\beta\gamma\delta} n^\alpha \ell^\beta n^\gamma m^\delta, \\
\Psi_2 &= -C_{\alpha\beta\gamma\delta} \bar{m}^\alpha \ell^\beta n^\gamma m^\delta, \\
\Psi_3 &= -C_{\alpha\beta\gamma\delta} \bar{m}^\alpha \ell^\beta n^\gamma \ell^\delta, \\
\Psi_4 &= -C_{\alpha\beta\gamma\delta} \bar{m}^\alpha \ell^\beta \bar{m}^\gamma \ell^\delta \qquad (1.22)
\end{aligned}$$

One of the interesting properties of these NP scalars is the "peeling" property [40] in an asymptotically flat space: with no radiation coming in from infinity, the leading order r behavior of the NP scalars are

$$\begin{aligned}
\Psi_0 &= O(r^{-5}), \\
\Psi_1 &= O(r^{-4}), \\
\Psi_2 &= O(r^{-3}), \\
\Psi_3 &= O(r^{-2}) \quad \text{and} \\
\Psi_4 &= O(r^{-1}). \qquad (1.23)
\end{aligned}$$

Here r is the distance from some region of strong-field gravity, and the tetrad used is as described earlier: ℓ^μ and n^μ are tangent to radially outgoing and ingoing null vectors respectively. Equations (1.23) show that in the vicinity of the source, all of the scalars are relevant in describing the gravitational wave content of the spacetime; as one moves away from the source, one-by-one the scalars become irrelevant (they "peel" off). The only NP scalar that dies off sufficiently slowly to be relevant at infinity, and hence carry gravitational wave information, is Ψ_4. In axisymmetry Ψ_4 is real; a non-zero imaginary part would be associated with angular momentum carried by the waves, which requires ϕ (the azimuthal angle) dependence [41].

The energy flux carried away by the waves can be calculated at large distances from the source using [42]

$$\frac{dM}{dt} = \lim_{r\to\infty} \frac{1}{4\pi} \int_{r=\text{const}} \left[\int_0^t \Psi_4 dt \right]^2 r^2 d\Omega, \qquad (1.24)$$

where M is the ADM mass [43] of the spacetime, $d\Omega$ denotes integration over the unit sphere, and we assume the outgoing radiation is zero before $t = 0$.

1.5 Numerical Solution of the Field Equations

To numerically solve the Einstein field equations (1.10) requires, in general, that one solve a set of ten, second order, coupled, quasi-linear partial differential equations (PDEs). To arrive at equations amenable to numerical solution, one must first choose a particular coordinate system. Components of the metric tensor $g_{\mu\nu}$ are then the unknown functions that we wish to solve for. By looking at the definition of the Einstein tensor in terms of contractions of the Riemann tensor (1.9), which is defined in terms of products of Christoffel symbols and their first derivatives (1.6), which in turn are sums of products of metric components and their first derivatives(1.4), one can see how the Einstein equations reduce to a set of 2^{nd} order PDE's for the metric functions. Additional equations governing the evolution of the given set of matter fields forming the stress-energy tensor must be supplied.

One of the more common methods of specifying the set of metric functions is the ADM, or $3+1$ formalism [22], discussed in some detail in the next subsection. In this formalism, the coordinate system is chosen to have a timelike coordinate t, and three spatial coordinates (hence $3+1$). Here, initial data is provided on the (say) $t=0$ slice of the spacetime, and the rest of the solution is then obtained by evolving in t. The initial data at $t=0$, consisting of the values of the metric functions and their first time derivatives, cannot be specified arbitrarily. Four of the Einstein equations, when written in $3+1$ form, only involve the initial data. These are the *constraint equations*, while the remaining six Einstein equations are called *evolution equations*. In theory, we only need to solve the constraint equations at the initial time, as the evolution equations preserve the constraints, by virtue of the contracted Bianchi identities (however, nothing prevents us from solving some or all of the constraint equations on each time slice, in lieu of solving an equivalent number of evolution equations). This implies that we only have six independent PDEs, to be solved for six metric functions. The remaining four metric functions can therefore be specified arbitrarily (to within the requirement that t remain timelike) throughout the evolution. This encapsulates the coordinate freedom that one has in general relativity, and is discussed in more detail in Section 1.5.2.

We use finite difference methods to solve the set of PDEs numerically: the computational domain is divided into a grid (or mesh) of points, and the differential equations are converted into a set of finite difference equations at each point on the grid. An overview of the concepts and basic techniques used to solve finite difference equations is given in Section 1.5.4. In Section 1.5.5, we introduce adaptive mesh refinement (AMR), which is a technique used to give sufficient (but not excessive) grid resolution to adequately resolve functions everywhere in the computational domain. Section 1.5.6 describes the multigrid method, which is used to efficiently solve elliptic equations.

A problem, that to some extent is unique to numerical solutions of the field equations, is how to deal with singularities that may arise within the spacetime. Computers cannot represent infinities, and if they do occur in a calculation and are not dealt with, the code will "crash". The two broad classes of singularities that could occur in the numerical solution of Einstein's equations are coordinate "pathologies" and spacetime singularities. A coordinate pathology is some aspect of the chosen coordinate system that could be problematic for a numerical code, including (but not restricted to) an actual singularity of the coordinate system. A benign example of this is the axis in spherical-polar type coordinates, which can be dealt with by applying appropriate regularity conditions to functions in the vicinity of the axis. Other coordinate pathologies could develop during the evolution of the spacetime, and may not be easy to deal with numerically. In principle, one could change coordinates within the offending region of the computational domain, resulting in a framework similar to the multiple coordinate patches that are often used in analytic work to obtain sensible results in all regions of interest. However, in practice this would be quite difficult to implement numerically, and instead we opt to choose a single coordinate prescription at the outset that we hope will cover a sufficient region of the spacetime[8].

Spacetime, or physical singularities, cannot be removed via a coordinate transformation. Furthermore, as discussed in Section 1.2, they *do* occur (at the very least) inside of black holes, and must be dealt with in any code simulating black holes. As was also mentioned in Section 1.2, the causal structure of black hole spacetimes allows us to excise the region of the black hole containing the singularity from the numerical grid (as long as cosmic censorship holds). The technique is called black hole excision, and will be described in Section 1.5.3.

1.5.1 The ADM formalism

In this section we describe the ADM (Arnowitt-Deser-Misner) space-plus-time decomposition of the metric, and the resulting field equations.

We begin by slicing the spacetime into a family of $t=$ const., spacelike hypersurfaces—see Figure 1.1. The unit timelike one-form, dual to the hypersurface normal vector, is defined as

$$n_\alpha = -\alpha\nabla_\alpha t, \quad n^\alpha n_\alpha = -1. \tag{1.25}$$

[8]Thornburg is currently working on a code that incorporates multiple patches, to evolve a black hole spacetime [76].

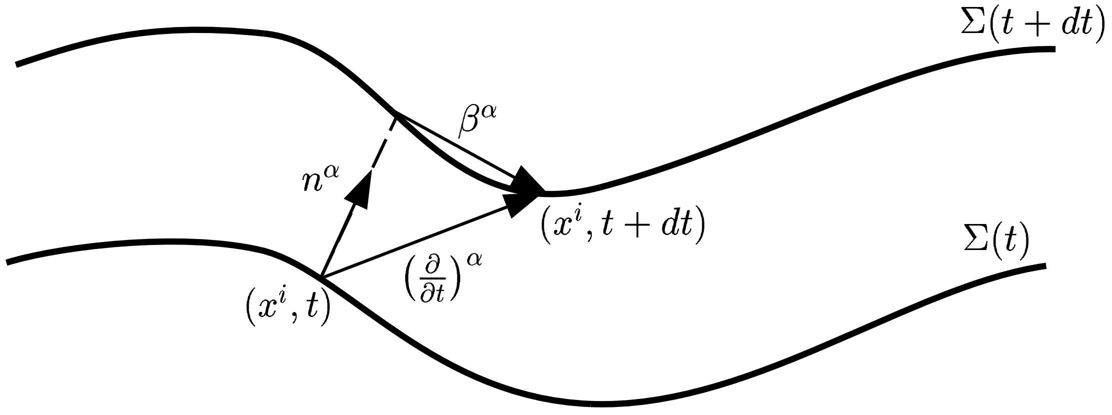


Figure 1.1: A schematic representation of the ADM space+time decomposition. $\Sigma(t)$ is a spacelike hypersurface with *unit* timelike normal n^α. The vector $\left(\frac{\partial}{\partial t}\right)^\alpha = \alpha n^\alpha + \beta^\alpha$ denotes the direction of constant coordinate position x^i within the sequence of hypersurfaces, and therefore represents the flow of coordinate time. α is called the *lapse function*, and β^α the *shift vector*.

α is called the *lapse function*[9], and embodies the coordinate freedom we have to rescale our time coordinate t. We will use coordinates $x^\alpha, \alpha \in 0..3$, with $x^0 = t$, and $x^a, a \in 1..3$ being the remaining spacelike coordinates within the $t =$ const. hypersurface. Using n^α, we can define a projection tensor $h_{\alpha\beta}$ as follows:

$$h_{\alpha\beta} = g_{\alpha\beta} + n_\alpha n_\beta. \tag{1.26}$$

$h_{\alpha\beta}$ is called a projection tensor because it "projects" tensors of the four dimensional manifold with metric $g_{\alpha\beta}$, into tensors that are tangent to the three dimensional manifold $t =$ const. with metric given by h_{ab}, where $a, b \in 1..3$. To see this, note from (1.25) and (1.26) that $h_{\alpha\beta}n^\alpha = h_{\alpha\beta}n^\beta = 0$, and $h_{\alpha\beta}(\partial/\partial x^a)^\alpha(\partial/\partial x^b)^\beta = g_{ab}$. Furthermore, with the projection of a four dimensional tensor defined as

$$\perp T_{ab...}{}^{cd...} \equiv h^\alpha{}_a h^\beta{}_b \; ... \; h_\gamma{}^c h_\delta{}^d \; ... \; T_{\alpha\beta...}{}^{\gamma\delta...} \tag{1.27}$$

it is straightforward to verify that the contraction of any index of $\perp T_{ab...}{}^{cd...}$ with n^α gives zero (a tensor satisfying this property is referred to as *spatial*), and hence from (1.26) one can raise and lower indices of $\perp T_{ab...}{}^{cd...}$ with either $h_{\alpha\beta}$ or $g_{\alpha\beta}$. The final piece of the decomposition of the 4-metric $g_{\alpha\beta}$, in addition to the lapse function α and 3-metric h_{ab}, is the *shift vector* β^a. The shift vector is a spatial vector (so $\beta^\alpha n_\alpha = 0$), and embodies the 3 degrees of coordinate freedom we have within the spacelike hypersurfaces, as shown in Figure 1.1. Explicitly, the decomposition is

$$\begin{aligned} ds^2 &= g_{\alpha\beta} dx^\alpha dx^\beta \\ &= -\alpha^2 dt^2 + h_{ab}\left(dx^a + \beta^a dt\right)\left(dx^b + \beta^b dt\right) \end{aligned} \tag{1.28}$$

The remainder of the ADM formalism involves writing the Einstein field equations (1.10) in terms of spatial tensors and their time derivatives. To this end, we use the extrinsic curvature

[9]Note that we the symbol α to denote the lapse function, as well as label tensor indices; it should be clear from the position of the symbol which α we are referring to.

tensor of the $t =$ const. hypersurface, defined as

$$\begin{aligned} K_{ab} &= -h_a{}^{\alpha} h_b{}^{\beta} n_{\alpha;\beta} \\ &= -\frac{1}{2} h_a{}^{\alpha} h_b{}^{\beta} \mathcal{L}_n g_{\alpha\beta}. \end{aligned} \tag{1.29}$$

Note that K_{ab} is a symmetric spatial tensor. Using the definition of the Lie derivative [29], we can (after some manipulation) write the above equation in the form of an evolution equation for h_{ab}:

$$\frac{\partial h_{ab}}{\partial t} = -2\alpha K_{ab} + \beta_{a|b} + \beta_{b|a}, \tag{1.30}$$

where the vertical bar | denotes the covariant derivative operator intrinsic to the spatial hypersurface:

$$[...]_{|a} \equiv h_a{}^{\alpha} [...]_{;\alpha}. \tag{1.31}$$

The projections of the stress-energy tensor $T_{\alpha\beta}$ into components tangent and normal to the hypersurface are defined as follows:

$$\rho = T_{\alpha\beta} n^{\alpha} n^{\beta}, \tag{1.32}$$

$$S_{ab} = T_{\alpha\beta} h^{\alpha}{}_a h^{\beta}{}_b, \tag{1.33}$$

$$J_a = -T_{\alpha\beta} h^{\alpha}{}_a n^{\beta}. \tag{1.34}$$

ρ is the energy density, J_a the momentum, and S_{ab} the spatial stresses of matter as seen by an observer traveling along n^{α}. The projection of the Einstein tensor into similar components can be obtained by using the Ricci identity (1.5) applied to n^{α}, and using the Gauss-Codazzi embedding equations for relating the projected four dimensional Riemann tensor $\perp R_{abcd}$ to the three dimensional Riemann tensor ${}^{(3)}R_{abcd}$ of the geometry intrinsic to the $t =$ const. hypersurface. This calculation is for the most part straight forward, but rather lengthy, and so we will refer the interested reader to [43, 44] for details. The result, substituted into the Einstein equations, and utilizing the definitions (1.32-1.34), are the Hamiltonian constraint

$${}^{(3)}R + K^2 - K_{ab}K^{ab} = 16\pi\rho, \tag{1.35}$$

the Momentum constraints

$$K_a{}^b{}_{|b} - K_{|a} = 8\pi J_a, \tag{1.36}$$

and evolution equations for the extrinsic curvature

$$\begin{aligned} \frac{\partial K_{ab}}{\partial t} = \alpha \left({}^{(3)}R_{ab} + K K_{ab} \right) - 2\alpha K_{ac} K^c{}_b - 8\pi\alpha \left(S_{ab} + \frac{1}{2} h_{ab} (\rho - S) \right) \\ -\alpha_{|ab} + \beta^c{}_{|a} K_{cb} + \beta^c{}_{|b} K_{ca} + \beta^c K_{ab|c}. \end{aligned} \tag{1.37}$$

In the above, K, S and ${}^{(3)}R$ are the traces of K_{ab}, S_{ab} and ${}^{(3)}R_{ab}$ respectively.

1.5.2 Coordinate and Slicing Conditions

In the ADM formalism, the four degrees of coordinate freedom we have in general relativity are most conveniently related to the lapse function α and three shift vector components β^a. Often, the choice of α is referred to as the *slicing condition*, and the choice of β^a as the *spatial coordinate condition*. In this section we briefly describe some of the more common conditions that have been used in numerical simulations in the past (see [44] and [48] for review articles describing these, and other conditions).

Perhaps the simplest coordinate choice is *geodesic* coordinates, where the lapse is set to one, and all of the shift vector components are set to zero. In that case, $x^a =$ const. observers follow geodesics of the spacetime. For long term numerical evolution this is not a very useful property to have in most cases, as gravity, or even non-trivial curvature on the initial slice, will cause geodesics to focus to caustics somewhere within the spacetime. Thus, the coordinate system will become singular there, as it is tied to the geodesics.

A slicing condition that is frequently used in $3+1$ numerical simulations is *maximal slicing* [44], where one enforces the conditions

$$K = 0, \quad \frac{\partial K}{\partial t} = 0. \tag{1.38}$$

From (1.37), these conditions result in the following elliptic equation for the lapse:

$$\nabla^2 \alpha = \alpha \left[K_{ab} K^{ab} + \frac{1}{3} (\rho + S) \right] \tag{1.39}$$

To obtain some understanding of the properties of this condition, notice from (1.29) that K measures (minus) the local expansion of the vector field n^α. Equivalently, $-K$ is equal to the fractional change of the spatial volume element normal to the hypersurface: $K = -(\partial \ln \sqrt{h} / \partial x^\alpha) n^\alpha$. This also implies that a spacelike slice with $K = 0$ is the slice with maximum total volume, within a region of spacetime inside a given, closed, spacelike boundary. Another property of maximal slicing is that it is *singularity avoiding.* The local volume element approaching a spacetime singularity often diverges, or goes to zero; forcing the volume element to remain fixed approaching such a singularity has the effect of "freezing" ($\alpha \to 0$) the slices in the vicinity of the singularity. A problem with this kind of singularity avoidance is that the $t =$ const. slices become severely distorted, resulting in steep gradients of metric functions that eventually will be as fatal to a numerical code as the curvature singularity.

The *minimal strain shift*, *minimal distortion shift* and *minimal strain* conditions are similar to maximal slicing in spirit, where some aspect of a congruence of timelike vectors is minimized over the spatial slices [45, 46]. The minimal strain/distortion shift conditions minimize the strain/stress (trace free part of the strain) of the vector field $(\partial/\partial t)^\alpha$, by variation over all possible choices of the shift vector β^α. This results in three elliptic equations, one for each component of the shift vector. The minimal strain shift conditions can be derived from the action $\sim h^{ac} h^{bd} \dot{h}_{ab} \dot{h}_{cd}$, where $\dot{h}_{cd}$ denotes differentiation with respect to t. The *minimal strain* conditions use the same action, though the variation is performed over all values of both the shift vector components and lapse function. This results in an algebraic condition for the lapse, in addition to the same elliptic conditions as before for the shift-vector components. The motivation behind these conditions is an attempt to reduce distortions of the coordinate system that may arise in highly dynamical spacetimes, for example in a binary black hole system. In the time of interest prior to merger, one would ideally like to follow the holes for several thousand orbits. A coordinate system that is co-rotating with the holes may be desirable (at least early on in the inspiral phase where the shape of the orbits are not expected to change rapidly), as this would eliminate a significant part of the non-radiative dynamics of the metric functions. The minimal strain condition may be able to provide such a co-rotating coordinate system[10].

In *harmonic coordinates*, the coordinate system is chosen so that one or more coordinates x^μ satisfy a curved-space wave equation, with source function H^μ:

$$\nabla^\alpha \nabla_\alpha x^\mu = H^\mu. \tag{1.40}$$

[10]Recent simulations of the merger of two relativistic clusters of particles by Shibata [47] with an *approximate minimal distortion (AMD)* condition showed promising results, in that relatively long-term, stable evolution was achieved in cases that did not result in black hole formation. The AMD condition is similar to the minimal distortion shift condition mentioned here, though the resulting elliptic equations are simplified at the expense of "not exactly" minimizing the strain of $(\partial/\partial t)^\alpha$.

Note that x^μ is considered a scalar in the above equation, despite the tensor-like index. If all coordinates are chosen to satisfy this condition, then the second-derivative terms of the metric in Einstein's equations take the form of scalar wave equations for each metric component [29]. This could be a desirable property to build a numerical code around, as solving a system of wave equations, particularly in a setting with adaptive mesh refinement, is in principle straight forward. Another benefit of the harmonic condition, applied to t, is that in some cases the resulting slices have singularity avoidance properties, similar to maximal slicing [49]. A possible disadvantage of using the harmonic conditions is that coordinate singularities tend to form when using them [50]; however, it may be possible to cure these problems which appropriate choices for the source functions H^μ [51]. In terms of the ADM formalism, one can write down expressions for the lapse and shift vector that would enforce the harmonic coordinate condition (see for instance [51]); however, directly using the variables of the ADM formalism might *not* be desirable if one wants to fully exploit the wave-like nature of the field equations.

Often, coordinates are chosen to simplify the form of the spatial metric h_{ab}, to simplify or alter the character of the resulting constraint or evolution equations, or to give metric variables certain properties. For example, if we write the spatial metric as a conformal metric $\hat{h}_{ab}$ times some conformal factor ψ^n, i.e. $h_{ab} = \psi^n \hat{h}_{ab}$, and at the initial time (or possibly for all time in a symmetry-reduced spacetime, such as spherical symmetry) we choose $\hat{h}_{ab}$ to be a flat metric, then one obtains an elliptic constraint equation for ψ [44]. Another example is to choose coordinates that are directly related to radiative variables in the asymptotically flat regions of the spacetime, to simplify the analysis of gravitational waves produces by sources in the interior [52].

1.5.3 Black Hole Excision

As mentioned in Section 1.2, black hole spacetimes contain curvature singularities somewhere inside of the event horizons, yet nowhere outside if cosmic censorship holds. These singularities need to be avoided in a numerical simulation, and the technique we have adopted in our code to do so is *black hole excision* (or simply excision).

The idea behind excision is as follows [28]. The defining property of a black hole is its event horizon, a one-way causal boundary that disconnects the exterior from any happenings in the interior. Therefore, we should be able to "remove", or excise, the interior from the computational domain, without affecting the physics of interest outside the black hole.

In terms of solving a set of differential equations, "removing" a piece of the domain requires setting appropriate boundary conditions on the surface of the region that is removed. For hyperbolic equations that have physical characteristic speeds, all of the characteristics of the equation will be directed into the excision boundary. Hence, boundary conditions are effectively not needed—one can simply employ "upwind" finite difference stencils in the corresponding difference equations at the excision surface (see Sections 1.5.4 and A.2 for more details on finite differencing and the particular difference stencils we use in our code, respectively). Various upwind-style difference techniques have been employed by several authors in the past [53, 54, 55, 56, 57, 58, 59, 60, 61, 62]. For elliptic or algebraic equations representing coordinate degrees of freedom (such as for the lapse and shift in general), one can in principle specify arbitrary conditions on the excision surface, as long as these conditions are consistent with the coordinate choices, regularity, etc. Desirable features for such conditions include being able to control the motion and/or coordinate size of the holes, and ensuring that the excision boundary does not move outside of the black hole during evolution.

Apparent Horizons

In practice, one excises a region within the *apparent horizon*, which is the outermost trapped surface on a given spacetime slice. A trapped surface is a two-sphere from which all families of inward and outward directed null curves, emanating from the surface, have negative expansion. If cosmic censorship holds, the apparent horizon is always inside the event horizon [1]. The reason one uses the apparent horizon, rather that the event horizon to excise, is that the position of the apparent horizon is a local property of a given slice[11]. An event horizon, in contrast, is globally defined, and in general cannot be found in an evolution until the entire spacetime exterior to the apparent horizon has been solved for (or at least until the spacetime has settled down to a stationary solution).

To find an equation that can be solved to give the position of the apparent horizon, using variables of the ADM formalism, we proceed as follows. Consider a family of hypersurfaces, $f(t, x^a) = R$. We want to calculate the null expansion $\theta_\pm$ normal to this surface, at a $t = t_0$ slice of the spacetime. θ_+ is the outward null expansion, θ_- the inward, and R is a constant labelling each member of the family. The unit spatial vector s^α normal to $f(t_0, x^a)$ is

$$s^\alpha = \frac{h^{\alpha\beta} f_{,\beta}}{\sqrt{h^{\gamma\delta} f_{,\gamma} f_{,\delta}}}. \tag{1.41}$$

Using s^α and the $t = t_0$ hypersurface normal vector n^α, we can construct future-pointing outgoing(+) and ingoing(−) null vectors normal to $f(t_0, x^a) = R$:

$$\ell^\alpha_\pm = n^\alpha \pm s^\alpha \tag{1.42}$$

The normalization of the null vectors is (arbitrarily) $\ell^\alpha_+ \ell_{-\alpha} = -2$. The expansion is then the divergence of $\ell^\alpha_\pm$ projected onto the hypersurface $f(t_0, x^a)$:

$$\theta_\pm = \left(h^{\alpha\beta} - s^\alpha s^\beta\right) \nabla_\beta \ell_{\pm\alpha}. \tag{1.43}$$

Using the definition of the extrinsic curvature K_{ab}, and substituting (1.42) into (1.43), we arrive at the standard form for the null expansion, in terms of ADM variables:

$$\theta_\pm = s^a s^b K_{ab} \pm s^a{}_{|a} - K. \tag{1.44}$$

Note that because the normalization of the null vectors is arbitrary, so (to some extent) is that of the expansion. The above normalization is chosen so that $\theta_\pm$ measures the fractional rate of change of area relative to a normal observer's (moving along n^α) time.

To find the apparent horizon on a given slice, we must find the outermost surface satisfying $\theta_+ = 0$ in (1.44). There are several ways of attempting to solve equation for $\theta_+ = 0$ (1.44) (for some examples see [65, 66, 67])—see Section 2.2.5 for a description of the particular method that we have implemented in our code.

1.5.4 Solution of Equations via Finite Difference Techniques

In this section we introduce some of the basic terminology and concepts in solving partial differential equations using finite difference techniques.

Discretization of a PDE

Denote a PDE by the differential operator L acting on a function u (or in general, a vector of functions), with possible "source" terms f, as

$$Lu = f \tag{1.45}$$

[11]Which can also be a disadvantage, as slices of a spacetime containing an event horizon may not always have an apparent horizon [63],[64].

To illustrate some of the concepts in this section, we will use the one dimensional wave equation as an example, for which L is

$$\frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial x^2}. \tag{1.46}$$

Boundary conditions must also be supplied to complete the specification of a system of PDEs. To solve (1.45), augmented with boundary conditions, using finite difference techniques, we first *discretize* the problem. This involves representing continuum functions, such as u, on a discrete *grid* (or *mesh*) of points, and replacing differential operators acting on continuum functions by *finite difference operators* that act on discrete grid functions. We will restrict attention here to uniform discretizations, where the coordinate spacing between gridpoints at a given time is h, and the spacing between time levels at a given grid location is λh, where both h and λ are constants. We can then denote the discretized differential operator by L^h, the discretized grid functions by u^h and f^h, and hence the resulting system of finite difference equations (FDEs) corresponding to (1.45) by

$$L^h u^h = f^h. \tag{1.47}$$

On the computer, we solve this set of FDEs for the grid function u^h, and, if implement "correctly", in the limit as $h \to 0$ the solution u^h obtained will tend to the continuum solution u. Before we discuss this desired feature of a solution of FDEs in more detail, as well as describe how to convert differential operators to finite difference (FD) operators, we introduce a few related definitions.

Basic Definitions and Concepts

The *solution error* e^h is defined as

$$e^h = u - u^h, \tag{1.48}$$

and is the difference between the continuum and finite difference solutions, evaluated at grid locations. Note that we assume $f = f^h$ at grid locations, i.e. there is no error in discretizing any source function. A related quantity is the *truncation error* τ^h:

$$\tau^h = L^h u - f, \tag{1.49}$$

which is a measure of the error in replacing the continuum operator L with the discrete operator L^h. A finite difference solution u^h is said to *converge* to the continuum solution u, if $\lim_{h\to 0} u^h = u$. Similarly, the system of FDEs is called *consistent* if $\lim_{h\to 0} \tau^h = 0$.

All of these concepts are encapsulated in the hypothesized *Richardson expansion* [68] of the continuum solution:

$$u = u^h + e_1 h + e_2 h^2 + e_3 h^3 + ... \tag{1.50}$$

Here, $e_1, e_2, e_3, ...$ are continuum functions that depend on t and x^i, but do not depend on the discretization scale h. For certain simple PDEs and discretizations one can prove that a Richardson expansion exists (for sufficiently smooth solutions), however, it is not possible to do so in general. In practice, if one obtains a consistent, convergent solution to the FDEs, then one effectively has a proof by construction that a Richardson expansion does exist for the given system.

The *order of convergence* n of a finite difference scheme is the lowest, non-vanishing power of h in (1.50). Both the truncation error and solution error scale as h^n in the small h limit. In fact, by substituting (1.50) into (1.48) and (1.49), we find that

$$\begin{aligned} \tau^h &= \left(L^h e_n\right) h^n + O(h^{n+1}) \\ &= L^h e^h + O(h^{n+1}). \end{aligned} \tag{1.51}$$

In other words, the truncation error is, to leading order, equal to the difference operator acting upon the solution error, and hence consistency implies convergence, and vice-versa, for systems possessing a smooth Richardson expansion.

In most situations where we need to find a numerical solution, we do not know what the continuum solution u is (otherwise, why bother with the numerics). Thus we cannot directly compute the truncation or solution errors using expressions (1.49) and (1.48). However, the Richardson expansion gives us the means to compute these quantities to leading order in h, by comparing discrete solutions obtained with different discretization scales. Given two solutions u^{h_1} and u^{h_2} of the FDEs, computed with discretization scales h_1 and h_2 respectively, approximations to the solution error and truncation error can be obtained by calculating $u^{h_2} - u^{h_1}$:

$$e^{h_1} = \left(u^{h_2} - u^{h_1}\right)\left(1 - \frac{h_2^n}{h_1^n}\right)^{-1} + O(h_1^{n+1}, h_2^{n+1}), \tag{1.52}$$

$$\tau^{h_1} = L^{h_1} u^{h_2} \left(1 - \frac{h_2^n}{h_1^n}\right)^{-1} + O(h_1^{n+1}, h_2^{n+1}). \tag{1.53}$$

Often one uses a set of discretization scales differing by factors of two, $h_1 = h, h_2 = 2h, h_3 = 4h, ...$ Then a simple quantity to calculate, measuring the rate of convergence (or failure to converge in an incorrect scheme) is

$$\frac{u^{4h} - u^{2h}}{u^{2h} - u^h} = 2^n + O(h). \tag{1.54}$$

In practice, we compute the ℓ_2 norm of this quantity to obtain a single number, as follows:

$$Q_u \equiv \frac{||u^{4h} - u^{2h}||_2}{||u^{2h} - u^h||_2}. \tag{1.55}$$

The order n of convergence of a FD scheme is governed by the order to which the finite difference operators approximate the differential operators, as we now discuss.

Deriving Finite Difference Stencils

Finite difference operators can be derived via Taylor expansions. For simplicity we will restrict our attention to the one dimensional case. Also, in the remainder of this section we only consider a single discretization scale h, and so we will denote the discretized version of u by $\hat{u}$, and the value of $\hat{u}$ at grid location i by $\hat{u}_i$.

In general, a finite difference operator $\mathcal{D}$, acting upon $\hat{u}$ at grid location i, is a weighted sum of grid function values:

$$\mathcal{D}\hat{u}_i = ... + a_{i-2}\hat{u}_{i-2} + a_{i-1}\hat{u}_{i-1} + a_i\hat{u}_i + a_{i+1}\hat{u}_{i+1} + a_{i+2}\hat{u}_{i+2} + ..., \tag{1.56}$$

where the a_n are constants. To determine what the differential operator $\mathcal{D}$ in (1.56) represents for a given set of constants, or to derive new finite difference operators, we replace each $\hat{u}_j$ in (1.56) by the continuum function $u(x)$, Taylor expanded about $x = x_i$ (recalling that $x_j - x_i = (j-i)h$):

$$\begin{aligned}\mathcal{D}\hat{u}_i = ... &+ a_{i-2}\left(u + u'(-2h) + \frac{u''}{2!}(-2h)^2 + \frac{u'''}{3!}(-2h)^3 + ...\right)\\ &+ a_{i-1}\left(u + u'(-h) + \frac{u''}{2!}(-h)^2 + \frac{u'''}{3!}(-h)^3 + ...\right)\\ &+ a_i u\\ &+ a_{i+1}\left(u + u'(h) + \frac{u''}{2!}(h)^2 + \frac{u'''}{3!}(h)^3 + ...\right)\\ &+ a_{i+2}\left(u + u'(2h) + \frac{u''}{2!}(2h)^2 + \frac{u'''}{3!}(2h)^3 + ...\right) + ... \end{aligned} \tag{1.57}$$

In the above expression, $u \equiv u(x_i)$, $u' \equiv du(x)/dx|_{x=x_i}$, etc. We re-order the terms in (1.57) to

$$\begin{aligned}\mathcal{D}\hat{u}_i &= u\left(... + a_{i-2} + a_{i-1} + a_i + a_{i+1} + a_{i+2} + ...\right)\\ &+ u'h\left(... - 2a_{i-2} - a_{i-1} + a_{i+1} + 2a_{i+2} + ...\right)\\ &+ u''\frac{h^2}{2!}\left(... + 2^2a_{i-2} + a_{i-1} + a_{i+1} + 2^2a_{i+2} + ...\right)\\ &+ u'''\frac{h^3}{3!}\left(... - 2^3a_{i-2} - a_{i-1} + a_{i+1} + 2^3a_{i+2} + ...\right)\\ &+ ..., \end{aligned} \tag{1.58}$$

Defining the sum of coefficients in (1.58) multiplying the m^{th} derivative of u by S_m, we can write (1.58) as

$$\mathcal{D}\hat{u}_i = uS_0 + u'hS_1 + u''\frac{h^2}{2!}S_2 + u'''\frac{h^3}{3!}S_3 + ... \tag{1.59}$$

From (1.59), we can see that for $\mathcal{D}$ to be an n^{th} order accurate approximation to the m^{th} derivative of u (i.e. for $\mathcal{D}\hat{u}_i = d^m u/dx^m + O(h^n)$), the following must hold: $S_0 = S_1 = ... = S_{m-1} = 0$, $S_m = m!/h^m$, and $S_{m+1} = S_{m+2} = ... = S_{n+m} = 0$. The reason that sums of terms up to S_{n+m} must be zero for an order n approximation, is that to satisfy $S_m = m!/h^m$, the coefficients in S_m have to be of order h^{-m} in magnitude; since these coefficients appear in all of the sums, the order of sum S_k becomes h^{k-m}. Conversely, to derive an n^{th} order accurate approximation to the m^{th} derivative of u, we need to choose a set of a_k to satisfy these conditions, giving a system of $n+m$ linear equations for the coefficients. Thus, to obtain a unique solution, one must (in general) have exactly $n+m$ non-zero coefficients in the sum (1.56).

Here are several examples of common finite difference operators, including the leading order truncation error term:

$$
\begin{aligned}
\frac{\hat{u}_{i+1}-\hat{u}_{i-1}}{2h} &= u' + u'''\frac{h^2}{6} + ... && (1.60)\\
\frac{-\hat{u}_{i+2}+4\hat{u}_{i+1}-3\hat{u}_i}{2h} &= u' - u'''\frac{h^2}{3} + ... && (1.61)\\
\frac{3\hat{u}_i-4\hat{u}_{i-1}+\hat{u}_{i-2}}{2h} &= u' - u'''\frac{h^2}{3} + ... && (1.62)\\
\frac{\hat{u}_{i+3}-4\hat{u}_{i+2}+7\hat{u}_{i+1}-4\hat{u}_i}{2h} = &= u' + u'''\frac{h^2}{6} + ... && (1.63)\\
\frac{4\hat{u}_i-7\hat{u}_{i-1}+4\hat{u}_{i-2}-\hat{u}_{i-3}}{2h} = &= u' + u'''\frac{h^2}{6} + ... && (1.64)\\
\frac{\hat{u}_{i+1}-2\hat{u}_i+\hat{u}_{i-1}}{h^2} &= u'' + u''''\frac{h^2}{12} + ... && (1.65)
\end{aligned}
$$

Equations (1.60-1.64) are all second order accurate approximations to the first derivative of u; equation (1.65) is a second order accurate approximation to the second derivative of u. Equation(1.60) is often called the *leap frog*, or *centered* first difference operator, and equations (1.61) and (1.62) are called *forward* and *backward* first difference operators, respectively. Typically, forward and backward difference operators are applied at grid boundaries, so that undefined points outside of the grid are not referenced. Equations (1.63) and (1.64) are examples of *truncation matched* difference operators; notice that (1.63) (a forward operator) and (1.64) (a backward operator) both have the *same* truncation error, to leading order, as the leapfrog scheme (1.60). This is a desirable feature to have in some instances, as it will help give a smooth truncation error to a finite difference equation throughout the computational domain [60].

Given a set of finite difference approximations to derivatives, as in (1.60-1.65) above, it is a simple task to convert a PDE (1.45) to a system of FDEs (1.47): replace all continuum functions with grid functions, and all derivatives of continuum functions with finite difference operators acting upon grid functions. Then, based upon the Taylor expansion expressions for difference operators (1.59), it is not difficult to see that the order of convergence of a FD scheme will be equal to the order of the lowest order FD operator used to form the system of FDEs.

As an example, we will convert the one dimensional wave equation $Lu = 0$, with L given by (1.46), to finite difference form, using the operator (1.65) for both the space and time derivatives:

$$\frac{\hat{u}_i^{n+1}-2\hat{u}_i^n+\hat{u}_i^{n-1}}{\lambda^2h^2} - \frac{\hat{u}_{i+1}^n-2\hat{u}_i^n+\hat{u}_{i-1}^n}{h^2} = 0, \qquad (1.66)$$

and for simplicity we impose Dirichlet boundary conditions on u at the grid boundaries $i = 1$ and $i = N$:

$$\hat{u}_1 = 0, \quad \hat{u}_N = 0. \qquad (1.67)$$

In (1.66) we have introduced the notation $\hat{u}_j^n$ to denote the grid function $\hat{u}$ (with implied spatial discretization scale h and temporal discretization scale λh), evaluated at time level n and grid location i. Thus, three time levels are utilized in this particular discretization scheme, and we consider the values of $\hat{u}$ at time level $n+1$ to be the unknowns that must be solved for during evolution. This is an example of an *explicit* scheme, for we can solve for the unknowns $\hat{u}_{n+1,i}$ without reference to any of the other unknowns at time level $n+1$:

$$\hat{u}_i^{n+1} = 2\hat{u}_i^n - \hat{u}_i^{n-1} + \lambda^2(\hat{u}_{i+1}^n - 2\hat{u}_i^n + \hat{u}_{i-1}^n) \qquad (1.68)$$

An equivalent description of an explicit scheme is that, for the system of equations (1.68) in matrix form $\mathbf{A}\hat{\mathbf{u}}^{n+1} = \mathbf{b}$, $\mathbf{A}$ is diagonal. If $\mathbf{A}$ were not diagonal, the scheme would be called *implicit.*

Stability

A general requirement for an explicit update scheme is that λ cannot be chosen arbitrarily, but must satisfy the so-called *Courant-Friedrichs-Lewy* (CFL) condition, which for (1.68) is $\lambda \le 1$ [69] (we discuss this in more detail in the next paragraph). This ties in with the notion of *stability* of FD schemes (which applies to all schemes, implicit and explicit included). Roughly speaking, an unstable FD difference scheme is one that admits solutions that grow faster than an exponential of the form e^{kt}, where k is a constant [69]. Typically, such solutions do not satisfy the continuum PDEs, and hence stability implies convergence, and vice versa. For linear, time-dependent PDEs, a similar statement, known as the *Lax-Richtmyer equivalence theorem*, can be proven rigorously: a consistent finite difference scheme for a partial differential equation for which the initial value problem is well posed is convergent if and only if it is stable (see, for instance [69]). We cannot prove this theorem for the system of PDEs arising from Einstein's field equations, however, we will assume that a similar result *does* hold in our case. This implies that one of the keys to obtaining valid results from a numerical code is to eliminate the instabilities. There are numerous factors that could affect the stability of a simulation, including the particular finite difference stencils used, the set of variables that are solved for, applying the correct boundary and regularity conditions for these variables, etc. Therefore, the approach that we take to construct a stable code, is to begin by using techniques that have been proven to work for similar systems, and then attempt to solve any new problems that may arise.

We now return to the discussion of the CFL condition, which is of relevance to the stability of most hyperbolic systems. For the FD approximation (1.66) to the wave equation, the CFL condition can be derived by performing a Fourier analysis of the update scheme (1.68), and demanding that the single-step growth factor of an eigenmode $\propto e^{i(kx \pm \omega t)}$ be less than or equal to 1 for stability. Intuitively, one can see why the CFL condition should exist for an explicit scheme by the following argument (see Figure 1.2 below). The characteristic speed of propagation of signals in the 1D wave equation (1.46) is $v = 1$. Hence, in a numerical evolution, to correctly represent the flow of information to a grid point x_i at time t, grid points within a region of size *greater than* $[x_i - v\Delta t, x_i + v\Delta t] = [x_i - \Delta t, x_i + \Delta t]$ must be referenced at time $t - \Delta t$. The FD difference scheme in (1.68) references the points $[x_i - \Delta x, x_i, x_i + \Delta x]$; and therefore, to satisfy this causality requirement, we need $\Delta x >= \Delta t$, which is exactly the CFL condition.

Dissipation

A method that we often use to construct stable FD approximations of hyperbolic equations, is to explicitly add numerical *dissipation*. Dissipation is effectively a low-pass filter applied to the grid function, suppressing high-frequency components in the numerical solution. Here, high-frequency always refers to wavelengths on the order of the mesh spacing h. The reason why dissipation is effective, and usually necessary, is that most FD schemes cannot accurately propagate high-frequency components of the solution; moreover, it is the high-frequency components that tend to exhibit the fastest growth in an unstable scheme. The dissipation technique that we most often use is the Kreiss-Oliger method [70], whereby a term of the form

$$D_{ko}\hat{u}_i \equiv \frac{\epsilon}{16}\left(\hat{u}_{i-2} - 4\hat{u}_{i-1} + 6\hat{u}_i - 4\hat{u}_{i+1} + \hat{u}_{i+2}\right) \tag{1.69}$$

is added to a difference equation in an "appropriate" fashion. ϵ is a positive, adjustable parameter controlling the amount of dissipation added, and must be less that 1 for stability (as can be shown for the appropriately adjusted scheme (1.68) by Fourier analysis, or by a less rigorous argument that we will present shortly, applicable to a general FD scheme).

The Taylor expansion of (1.69) about x_i is

$$D_{ko}\hat{u}_i = \epsilon h^4 u'''' + O(h^5), \tag{1.70}$$

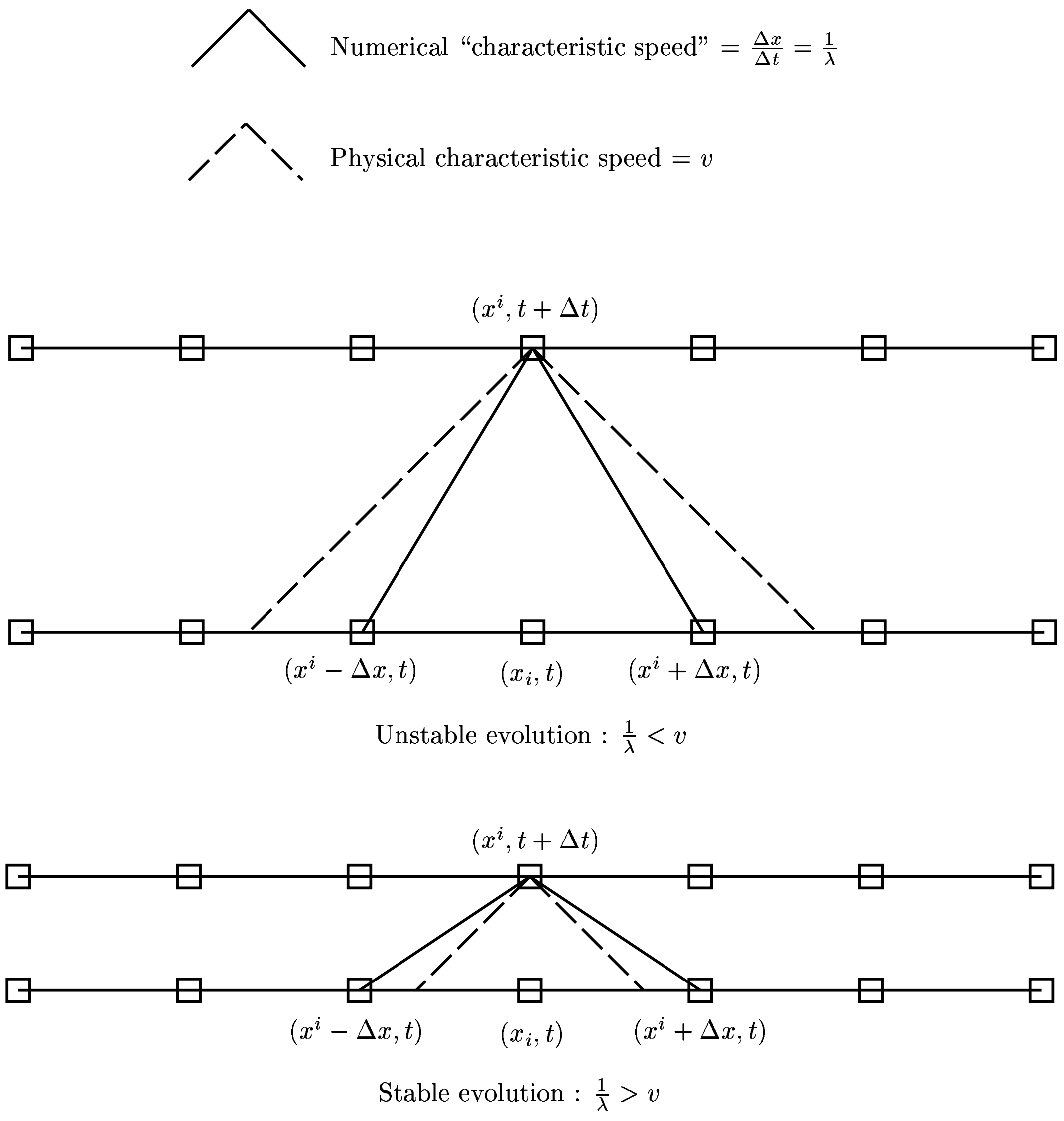

Numerical “characteristic speed” $= \frac{\Delta x}{\Delta t} = \frac{1}{\lambda}$
Physical characteristic speed $= v$
$(x^i, t + \Delta t)$
$(x^i - \Delta x, t)$
(x_i, t)
$(x^i + \Delta x, t)$
Unstable evolution : $\frac{1}{\lambda} < v$
$(x^i, t + \Delta t)$
$(x^i - \Delta x, t)$
(x_i, t)
$(x^i + \Delta x, t)$
Stable evolution : $\frac{1}{\lambda} > v$

Figure 1.2: A schematic demonstration of the CFL condition.

from which it is apparent that adding a term of the form (1.69) to a second-order-accurate FD approximation should not affect the convergence properties of the scheme. Considered as a filter, equation (1.69) is a convolution of the Kreiss-Oliger filter with the grid function $\hat{u}$. Fourier transforming to frequency space $k \equiv \xi/h$: $\hat{u}(x_i) \to \hat{u}(\xi)$, the convolution operator transforms to a multiplication, and (1.69) becomes

$$D_{ko}\hat{u}(\xi) = \epsilon \sin^4(\xi)\hat{u}(\xi). \tag{1.71}$$

ξ takes on values in the range $[-\pi, \pi]$, for the highest possible wave number k that can be represented on a mesh with spacing h is $\frac{\pi}{h}$ (the Nyquist limit). Written as is, equation (1.71) is a high-pass filter; i.e. the low-frequency components of $\hat{u}$ are almost completely suppressed after the convolution with D_{ko}. Hence, we must *subtract* $D_{ko}\hat{u}_i$ from $\hat{u}_i$ (with $\epsilon > 0$) in order to achieve the desired effect of reducing the high-frequency components of the solution. Furthermore, we can see that if $\epsilon > 1$, we will *over-compensate* for frequencies $\xi > \arcsin(\epsilon^{-4})$, i.e. we will just be reintroducing some of the high frequency components but with opposite sign, and possibly even amplifying them if ϵ is large enough.

In general, there are several possible ways to add Kreiss-Oliger dissipation to a given FD scheme. For example, one way to add it to (1.66) is to subtract the high-frequency part of the solution at the most retarded time level $\hat{u}_{n-1,i}$. Then (1.68) becomes

$$\begin{aligned} \hat{u}_i^{n+1} &= 2\hat{u}_i^n - \hat{u}_i^{n-1} + \lambda^2(\hat{u}_{i+1}^n - 2\hat{u}_i^n + \hat{u}_{i-1}^n) \\ &+ \frac{\epsilon}{16}\left(\hat{u}_{i-2}^{n-1} - 4\hat{u}_{i-1}^{n-1} + 6\hat{u}_i^{n-1} - 4\hat{u}_{i+1}^{n-1} + \hat{u}_{i+2}^{n-1}\right) \end{aligned} \tag{1.72}$$

Iterative Solution of Non-linear FDEs

The update scheme (1.72) for the wave equation is rather trivial, and can easily be solved exactly (i.e. to within machine precision) in an evolution code. However, this is not typical of the FDEs we will obtain from Einstein's equations; moreover, many of the FDEs we must solve are non-linear. The method we use to solve non-linear, hyperbolic type FDEs is *point-wise Newton-Gauss-Seidel (NGS) relaxation*, which we now describe (we use multigrid for elliptic equations—see Section 1.5.6).

Denote a system of finite difference equations via

$$F_i(x_j) = S_i. \tag{1.73}$$

Here, the indices label unknown *variables* x_i (not to be confused with a spatial coordinate), FD equations F_i and source functions S_i (so an index will run from 1 to $N_g N_v$, where N_g is the total number of grid points, and N_v is the number of unknown variables per grid point). The F_i are in general non-linear, and so we will solve for the corresponding x_i's via Newton's method. In Newton's method, we first write the solution in the form $x_j = x_j^{(0)} + \delta x_j^{(0)}$, and Taylor expand (1.73) about $x_j = x_j^{(0)}$ to first order:

$$F_i\left(x_j^{(0)}\right) + \sum_j \frac{\partial F_i}{\partial x_j}|_{x_j = x_j^{(0)}} \cdot \delta x_j^{(0)} = S_i + O\left(\left(\delta x_j^{(0)}\right)^2\right). \tag{1.74}$$

The *Jacobian* of the system is defined to be

$$J_{ij}\left(x_k^{(0)}\right) \equiv \frac{\partial F_i}{\partial x_j}|_{x_j = x_j^{(0)}}. \tag{1.75}$$

Using (1.75), we rewrite the leading order part of (1.74) in the following form:

$$\sum_j J_{ij}\left(x_k^{(0)}\right)\delta x_j^{(0)} = S_i - F_i\left(x_k^{(0)}\right). \tag{1.76}$$

Newton's method proceeds by first solving the system of linear equations (1.76) for the perturbations $\delta x_j^{(0)}$, given an initial guess $x_j^{(0)}$, and then iterating the procedure using the sequence of guesses $x_k^{(1)} = x_j^{(0)} + \delta x_j^{(0)}$, $x_j^{(2)} = x_j^{(1)} + \delta x_j^{(1)}$, ... The iteration stops once a solution $x_k^{(N)}$ has been obtain that satisfies the full non-linear FDEs (1.73) to within a specified tolerance. Newton's method is very efficient given a "good" initial guess: then a residual norm $||F_i\left(x_k^{(N)}\right) - S_i||$ will converge to zero quadratically as a function of the number of iterations N [71].

In our case, given the number of unknowns x_i we have in a typical problem, it would be too expensive (computationally) to solve each step of Newton's method (1.76) exactly. Instead, we apply a single *Gauss-Seidel relaxation sweep* to the linear system (1.76) per Newton iteration. A Gauss-Seidel relaxation sweep consists of using each equation F_i in turn to (approximately) solve for the corresponding unknown x_i, by assuming that all of the other variables x_j, $j \neq i$, are known. In addition, once a variable is solved for in this manner, it is immediately used in subsequent steps of the relaxation sweep (this distinguishes Gauss-Seidel from Jacobi relaxation). Thus the convergence properties of Gauss-Seidel relaxation depend on the order in which the variables are traversed in the sweep.

This combination of Newton iteration with Gauss-Seidel relaxation is what we refer to as Newton-Gauss-Seidel relaxation, and from experience it is quite efficient at solving hyperbolic-type equations (usually less than 10 iterations per time step were required for the results presented in this thesis).

1.5.5 Adaptive Mesh Refinement

A computational difficulty that often arises in the problems that we are interested in solving is that a wide range of relevant length scales are present within the computational domain. Furthermore, the smaller length scales tend to occupying smaller volumes of space. For example, in a black hole collision, the black holes represent the smallest length scales $\approx M$ (the black hole mass), but occupy a relatively small fraction of the grid, as the outer boundary needs to be on the order of $100M$ away from the collision to obtain good waveform estimates. A uniform grid then becomes quite inefficient at representing functions, because if given sufficient resolution to resolve the finest length scale, the grid will be over-dense in regions covering larger length scale features. Computational resources, including memory, disk-storage and CPU time, scale (at best) linearly with the number of grid-points processed during an evolution. This makes uniform grid solutions of FDEs, at the desired level of accuracy, impractical on extant computer systems. One obvious solution is to use non-uniform grids; however, this complicates the finite difference stencils, and, more significantly, a single non-uniform mesh cannot, in general, adapt to changing features of the solution. Another alternative, that we have chosen to implement, is called *adaptive mesh refinement* (AMR), whereby the computational domain is covered by a dynamical hierarchy of uniform grids of differing resolution. We use a version of the Berger and Oliger (B&O) algorithm [72], which we describe in the remainder of this section.

AMR Grid Hierarchy

In the Berger and Oliger AMR algorithm, the computational domain is decomposed into a hierarchy of uniform meshes (see Figure 1.3) with the following properties:

- The hierarchy contains ℓ_f *levels*. Each level ℓ contains grids of the same resolution—the coarsest grids are in level 1 (the *base grid*), the next-coarsest in level 2, and so on until level

ℓ_f, which contains the finest grids in the hierarchy.

- The ratio of discretization scales $h_\ell/h_{\ell+1}$ between levels ℓ and $\ell+1$ is called the *spatial refinement ratio* $\rho_{s,\ell}$. $\rho_{s,\ell}$ is typically an integer greater than or equal to 2. For simplicity, we will also assume that $\rho_{s,\ell}$ is the same for all levels (which is what we have used in practice so far), and therefore use the symbol ρ_s to denote the spatial refinement ratio.
- All grids at level $\ell+1$ (*child* grids) are *entirely* contained within grids at level ℓ (*parent* grids). Grids at the same level may overlap.
- In our simplified variant of the B&O algorithm, we require that all grids within the hierarchy share the same coordinate system. In particular, this implies that all grid boundaries run parallel to the corresponding boundaries of the computational domain. In addition, we require that a child grid must be aligned relative to its parent grid such that all points on the parent grid, within the common overlap region, are coincident with a point on the child level. The original B&O algorithm allowed for a child grid to be rotated relative to its parent.

As we will discuss in detail below under the heading "Dynamical Regridding via Truncation Error Estimation", the particular grid structure that exists at any given time is calculated so that at any point $\vec{x}$ within the computational domain, the finest grid covering that point has sufficient resolution to adequately resolve all features of the solution there. This is an important property of the grid hierarchy, not only for the obvious reason of providing the desired resolution everywhere, but it justifies the use of the B&O time-stepping algorithm to evolve the hierarchy, as we now explain.

The Berger and Oliger Time-Stepping Algorithm

An adaptive grid hierarchy is evolved in time by taking a particular sequence of unigrid time-steps on the individual grids within the hierarchy. This sequence of updates is determined by the following rule, applied recursively at any given level (see Figure 1.4 for a pseudo code description of the time-stepping procedure): all grids at level ℓ are evolved by a ***single*** time step of size Δt_ℓ *before* the grids at level $\ell+1$ are evolved by $\rho_{t,\ell}$ time steps of size $\Delta t_{\ell+1} = \Delta t_\ell/\rho_{t,\ell}$. $\rho_{t,\ell}$ is called the *temporal refinement ratio*, and is an integer greater than or equal to 2, though typically is set equal to $\rho_{s,\ell}$ or larger in order to satisfy the CFL condition. As with ρ_s, we have to-date only used a constant temporal refinement ratio ρ_t for all levels.

The reason why a time step is first taken on coarse level ℓ, is that the solution obtained there at time $t+\Delta t_\ell$ is then used to set boundary conditions, via linear interpolation in time, for the subsequent evolution on the finer level $\ell+1$ (unless some portion of the fine level abuts the boundary of the computational domain, where the given boundary conditions of the FDE can be applied). In other words, at interior boundaries we are, in a sense, not supplying "boundary conditions"; rather, we are evolving them using the FD equations from the coarser level. It is possible to do this because, as should become clear when we explain the dynamical regridding procedure, the solution obtained on the coarse level in the vicinity of the fine level boundary will be *as accurate*, to within the specified truncation error, as a putative solution obtained on a fine level encompassing the entire computational domain. The solution on the coarse level interior to this boundary will not be as accurate; however, the assumed hyperbolic nature of the FDEs will protect this inaccuracy from polluting the coarse/fine boundary region within a single coarse level time step.

After ρ_t time-steps on level $\ell+1$, when the solution on grids at levels ℓ and $\ell+1$ are again in synchrony, grid functions from level $\ell+1$ are *injected* into the coarse grids at level ℓ, in the region of overlap between the two levels. Thus, the most accurate solution available at a given point $\vec{x}$ is continuously propagated to all grids in the hierarchy that contain $\vec{x}$. Injection simply consists of copying values from level $\ell+1$ to level ℓ at common points (in the more general B&O algorithm, where finer levels can be rotated relative to coarser levels, the injection step requires some form of interpolation).

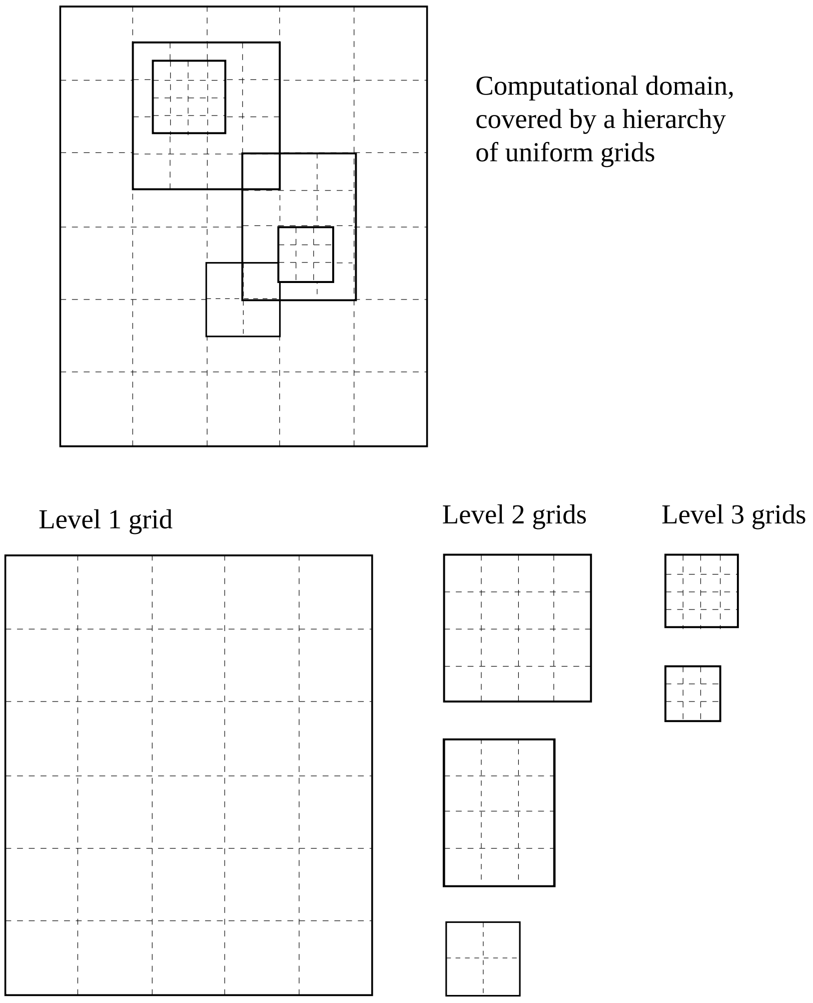


Figure 1.3: An example of a Berger and Oliger mesh hierarchy. The upper diagram shows the computational domain, covered by a three level deep hierarchy of uniform grids, where higher level numbers denote levels consisting of grids with higher spatial resolution. The plots below this demonstrate how the hierarchy is stored in memory, namely as a collection of individual grids. Thus a given point on the computational domain could be represented in multiple grids in the Berger and Oliger scheme.

```
for (requested number of coarse steps)
   call single_step(level 1)
   stop

subroutine single_step(level L)
   if regridding_time(L) then
      regrid from levels L to Lf
   end if

   if (L>1) then
      set boundary conditions along AMR boundaries at time
         t(L)+delta_t(L) via interpolation from level (L-1)
   end if

   perform 1 evolution step for all grids at level L

   t(L)=t(L)+delta_t(L)

   if (L<Lf) then
      1: repeat [rhot=delta_t(L)/delta_t(L+1)] times: call single_step(L+1)
      2: inject the solution from level L+1 to level L
         in the region of overlap between levels L and L+1
   end if

   return from subroutine
end of subroutine single_step
```

Figure 1.4: A pseudo-code representation of the Berger and Oliger time stepping algorithm.

Dynamical Regridding via Truncation Error Estimation

One of the key features of Berger and Oliger style AMR, is that the grid hierarchy is constructed *dynamically*, to give resolution where it is needed as a solution unfolds. The process of modifying an existing grid hierarchy, whether adding, removing, extending grids, etc., is called *regridding.* On each level, regridding is performed periodically, at a user specified rate of once every T_r time steps (a different rate could be chosen for each level). When it is time to regrid at level ℓ, all levels from ℓ to ℓ_f are involved. Ideally T_r should be set to the largest possible number that could track evolving features of the solution. For example, if the characteristic speeds of the FDEs are 1, and the Courant factor $\lambda = 0.25$, then one would expect a wave front to travel by at most 1 grid point every 4 time steps. In that case, T_r should be set to 4. However, T_r could be less than 4 without incurring much of a speed penalty (in numerical relativity, the evolution phase is usually the speed bottle-neck), and setting T_r to numbers several times larger than 4 would also not cause problems if sufficiently large *buffer zones* are added to each grid. We will explain buffer zones shortly.

Local truncation error estimates drive the regridding phase. To avoid confusion, notice that what we will now call the *truncation error estimate* (for historical reasons) is an approximation of what was defined as the *solution error* (1.48) in Section 1.5.4. However, by the assumed Richardson expansion, regridding based upon either form of error estimate should give equivalent results. At regridding time on level ℓ, the truncation error is estimated for each grid g on levels ℓ to ℓ_f via the following procedure (we use a slightly different mechanism in the actual code, which is explained in Section 2.3.1). Two copies of grid g, that has discretization scale h, are made: the first g^h is an identical copy of g, while the second g^{2h} is a $2:1$ coarsened version of the first. Then two unigrid evolution time-steps of size Δt_ℓ are taken on grid g^h, and one unigrid evolution time-step of size $2\Delta t_\ell$ is taken on grid g^{2h}. Dirichlet boundary conditions are used during the evolution on grid boundaries interior to the computational domain. Then, based on the Richardson expansion (1.50), an $O(h^n)$ estimate of the solution error of some evolved grid function u can be calculated by subtracting the solutions obtained at $t + 2\Delta t_\ell$ on grids g^{2h} and g^h (the grid function u^h is coarsened before the subtraction):

$$e^h(u) = u^h - u^{2h} + O(h^n), \tag{1.77}$$

where n is the order of the finite difference scheme. We define the truncation error estimate (TRE) τ_g for grid g at level ℓ to be some point-wise norm over solution error estimates of a specified set of grid functions:

$$\tau_g(\vec{x}) = \sum_u ||e^h(u(\vec{x}))|| \tag{1.78}$$

Once τ_g has been calculated for all grids on levels ℓ to ℓ_f, this information is passed to a *clustering algorithm*, which determines an appropriate set of grids to keep the truncation error below a global threshold $\tau_{\max}$ on the finest level covering *each* point within the computational domain. We will not go into the details of any specific clustering algorithm here, as there are many such algorithms (see [73] and the references therein), most rather *ad-hoc* in nature (except in 1D, where the problem is trivial); rather we will give a brief outline of the main steps in the clustering process. First, regions of high truncation error ($\tau_g(\vec{x}) > \tau_{\max}$) on grid g are flagged. Second, the flagged region is expanded in all directions by a user specified number of grid points—the *buffer zone.* This provides a "safety zone" between the region of high truncation and what will become the boundaries of child grids of g, so that we can self-consistently use evolution on grid g to set the boundary conditions for evolution on the child grids, via the B&O time-stepping algorithm. Also, as mentioned above, a larger buffer zone will allow us to regrid less frequently, if desired, as one would in general expect the region of high TRE to track fine features of the solution, which propagate with characteristic speeds. Third, the expanded region of high TRE on level g is covered by a set of child grids, or clusters, g_c. Ideally, one wants to choose a minimal set of clusters yet achieve a large *filling factor*, defined as the ratio of grid points with high TRE to the total number of grid points within the set of grids g_c. This is trivial to do in 1D; however, in 2D and higher, there is often no unique set of clusters that achieve a desired filling factor (which is why clustering algorithms are rather heuristic in nature). Specifying a high filling factor (close to 1) will produce many tiny clusters, which leads to larger computational overhead, and tends to complicate the process of dealing with high-frequency components of the solution of the FDEs that often develop at parent-child boundaries (we will refer to these unwanted high-frequency parts of the solution as *noise*). A small filling factor will give a few large clusters, which could result in a lot of unnecessary refinement.

Finally, once the set of child grids g_c at level ℓ has been determined, the grid functions on g_c are initialized, either by copying function data from the prior set of grids on level $\ell + 1$ that overlap the new set g_c, or via interpolation of data from parent grids at level ℓ to newly refined regions in g_c.

Advantages/Disadvantages to Using B&O AMR

To conclude this section, we list a few of the advantages and disadvantages of using Berger and Oliger style AMR, compared to using a non-uniform grid. The advantages of using B&O AMR include

- Dynamical regridding based on truncation error estimates allows one to "easily" probe unknown regions of solution space, whereas a non-uniform grid generally needs to be specified *a priori* to give resolution where it will be needed.
- The recursive nature of the B&O time stepping algorithm makes the most efficient use of resources possible, within limitations imposed by accuracy considerations (and the CFL condition). For in a typical hyperbolic system, temporal gradients are of the same order of magnitude as spatial gradients, and thus the time step should be proportional to the discretization scale in order to achieve the desired level of accuracy. Consequently, the time step one would need to use to evolve a non-uniform grid is set by the *smallest* cell size on the grid.
- Since the B&O algorithm operates by calling a sequence of unigrid evolution routines, the AMR code can be written in a generic fashion; in other words, it is a straight forward task to add an AMR framework to an existing unigrid code.

Some of the disadvantages compared to a non-uniform grid include:

- High-frequency solution components ("noise") often develop at parent/child boundaries when using standard interpolation methods there. This needs to be dealt with in some fashion, and often involves finding a set of interpolation and dissipation operators that work for the particular set of FDEs that are being solved.
- There is some extra overhead involved in storing and evolving the grid hierarchy, where a point with high truncation error could appear on grids in many levels. However, even with a small spatial refinement ratio of 2 : 1, this overhead is rather small.
- It is not a trivial task to incorporate non-hyperbolic equations into the B&O AMR framework—see Section 2.3.1 for a description of how we have incorporated elliptic equations into the algorithm.

1.5.6 Solution of Elliptic Equations via Multigrid

The Newton-Gauss-Seidel (NGS) relaxation method that we discussed in Section 1.5.4 above is not, by itself, an efficient method to solve elliptic-type finite difference equations. What we use instead is the *multigrid* (MG) method—one of the fastest algorithms currently available to solve elliptic FDEs [74]. In this section we give a brief overview of the *Full Approximation Storage* (FAS) multigrid algorithm [75], which is a variant of MG designed to directly solve non-linear PDEs.

Iterative Solution Methods and the Residual

Before we describe the FAS algorithm, it will be useful to discuss the operation of an iterative solution scheme from the point of view of the *residual.* The residual R^h is the truncation error of an approximate solution $\hat{u}^h$ to an FDE (1.47) (where again we use the superscript h to denote the discretization scale):

$$R^h(\hat{u}^h) = L^h\hat{u}^h - f \tag{1.79}$$

If we solved the FDE (1.47) exactly the residual would be zero. When solving (1.47) via an iterative technique, the goal is not, in general, to find a $\hat{u}^h$ such that $R^h(\hat{u}^h) = 0$ in (1.79); rather, starting from an initial guess $\hat{u}^h_0$, we want to generate, via iteration, a sequence of approximate solutions $\hat{u}^h_n, n = 1, 2, ...,$ until we obtain a solution $\hat{u}^h_N$ such that the norm of the corresponding residual is below some threshold. Desired properties of an iterator S^h_n, whose operation can be written in matrix form as $\hat{u}^h_{n+1} = S^h_n\hat{u}^h_n$, include that $||R^h(\hat{u}^h_{n+1})|| < ||R^h(\hat{u}^h_{n1})||$, and that each application of S^h_n can be performed relatively quickly. Note that the iterator S^h_n could change from one iteration to the next (hence the subscript n), as is the case with a NGS relaxation sweep of a non-linear FDE for instance, where S^h_n changes due to the Newton iteration.

The Concept of Multigrid

Multigrid is an iterative solution technique, employing a particular relaxation method to the PDE discretized on a sequence of grids, each with different spatial resolution. A general property of most relaxation methods is that they are efficient at reducing the *high-frequency components* of the residual (where, as in the discussion of dissipation above, high-frequency refers to Fourier components whose wavelengths are on the order of the mesh spacing h), while conversely, they are very ineffective at reducing the low-frequency components of the residual. In the context of MG, a relaxation scheme with this property is called a *smoothing operator.* It is this ability of relaxation to smooth the residual that is the motivation behind MG, the basic operation of which is as follows. We apply a few iterations of the smoothing operator. After this, the residual $R^h(\hat{u}^h)$ will be a smooth function relative to the discretization scale h, and thus we can accurately represent this residual on a coarser grid, with discretization scale $2h$ [12]. We therefore transfer the problem of reducing the residual to the coarser grid (the details of which will be presented shortly). Now, what was lower frequency (wavelength on the order of $2h$) on the fine grid becomes high-frequency on the coarse grid, and hence relaxation on the coarse grid will be effective at reducing these slightly longer wavelength components of the residual. This process of smoothing-plus-coarsening is repeated until we reach a small enough coarse grid where it is feasible to solve the resultant system of equations exactly, thus eliminating the lowest frequency components of the solution. The final stages of a MG iteration then consist of a series of operations that propagate the low-frequency corrections, calculated on the coarser grids, up through the grid hierarchy to the finest resolution grid. This particular form of MG iteration is called a *V-cycle*, because the smoothing-plus-coarsening phase proceeds all the way from the finest to coarsest levels (down the V), followed by the correction phase which goes all the way from the coarsest level back to the finest level (up the V). The V-cycle is repeated until the residual on the finest level mesh is driven below some threshold.

Pointwise Newton-Gauss-Seidel is one of the relaxation schemes that usually performs well as a smoothing operator [74]. We use this method in our code, and we simultaneously solve for all of the unknowns at a given grid point at each step of the relaxation sweep. Also, we visit points on the 2D grid in so-called *red-black* order. Imagine colouring points of the grid with red and black in a checkerboard fashion; then a red-black sweep updates all the unknowns at red grid points first (in lexicographic order), followed by unknowns at black grid points. For elliptic equations differenced with a 5-point stencil[13], red-black ordering decouples the FDEs between red and black

[12] A 2:1 coarsening ratio is typical, though not essential, for multigrid.

[13] An interior finite difference equation at location (i, j) then references variables at (i, j), $(i + 1, j)$, $(i, j + 1)$, $(i - 1, j)$ and $(i, j - 1)$.

points. Consequently, after a red-black relaxation sweep, the residual on all black points will be zero.

The FAS Multi-Grid Algorithm

In the FAS MG algorithm, we construct the set of finite difference operators that will be used to smooth the residual, on the sequence of grids with discretization scales $h_0, h_1, ...h_N$, where we use $h_{i+1}/h_i = 2$, as follows[14] (to simplify the notation a bit in the following discussion, we will only use a superscript $()^n$, rather than $()^{h_n}$, to denote discretization at level h_n). The fine-grid problem is

$$L^0 \hat{u}^0 = f + R^0, \tag{1.80}$$

where the goal of the MG iteration is to reduce $||R^0||$ to a specified small number. In the first step of the algorithm, we apply a certain number of relaxation sweeps to (1.80), until the residual $||R^0(\hat{u}^0)||$ is smooth. Next, we transfer all of the grid functions, including the residual, to a grid with $h = h_1$. This transfer operation is called *restriction*, and is denoted by the operator I_0^1. For example, the restricted residual will be $I_0^1 R^0$. We then define the system of FDEs on level h_1 as:

$$L^1 \hat{u}^1 = f^1 + R^1, \tag{1.81}$$

where L^1 is the operator describing the system of PDEs discretized on the mesh with spacing $h = h_1$, $\hat{u}^1$ are the new set of unknowns, R^1 is the new residual function for this system of FDEs with approximate solution $\hat{u}^1$, and the new "right hand side" vector f^1 is given by

$$f^1 = I_0^1[f + R^0] + \tau^0. \tag{1.82}$$

τ^0 is the truncation error introduced by approximating the desired system of FDEs with grid spacing h_0 on the coarser grid with $h = h_1$:

$$\tau^0 = L^1[I_0^1 \hat{u}^0] - I_0^1 f. \tag{1.83}$$

Adding the truncation error to the coarser grid FDEs is a *crucial* component of the algorithm, for we are *not* interested in the solution of the PDE $Lu = f$ on the mesh at level h_1; rather, we need a vehicle on the coarser mesh that will continue to smooth the restricted residual of the fine-mesh problem via adjustments of the unknowns, and we know that relaxation of the discretized operator $Lu = f$ on level h_1 gives us this vehicle. Therefore, by adding the truncation error, we are effectively adding a source term to the coarse grid problem that will drive the coarse grid solution to that of the fine grid, at common grid points. For notice in equations (1.81-1.83), that if $R^0 = 0$, i.e. the fine grid problem is solved, then the (presumed unique) solution of the coarse grid problem (1.81) is $\hat{u}^1 = I_0^1 \hat{u}^0$.

Now that we have the problem transfered to the coarser level h_1, we repeat the smoothing-plus-coarsening steps: the relaxation operator is applied to (1.81) until the residual R^1 is smooth, then the system of equations is transfered, as before, to a mesh with spacing $h = h_2$. This process is repeated until we reach the coarsest level, $h = h_N$. At this point, the coarse problem should be solved "exactly", preferably using a combination of Newton-iteration and direct solution of the resultant linear system. The reason why the rule-of-thumb is to use a direct solution, rather than continue to use the same relaxation method that was used for smoothing (which is feasible on the coarsest grid, as the corresponding system of equations is small enough that relaxation is efficient), is that studies of the convergence properties of some relaxation techniques on model problems (such as Gauss-Seidel relaxation for the Laplace operator) show that relaxation is marginally unstable for the longest wavelength components of the solution [75]. The growth rate of the unstable mode

[14]Notice that the level numbering scheme is opposite to that used in the description of AMR, where finer levels had a higher level number.

is often slow, i.e. it takes hundreds or thousands of iterations for it to manifest, and hence does not pose a problem in the stages of MG prior to the coarsest level, where comparatively few iterations are applied. In our case, however, we have not seen any adverse effects solving the coarse grid problem via relaxation [15], implying that relaxation is stable in this case. Furthermore, it is not necessary to solve the coarse grid problem exactly (i.e. to within machine precision); reducing the residual by two to three orders of magnitude is sufficient.

The first stage of the V-cycle is now complete. The second (and final) stage consists of propagating *coarse-grid-corrections (CGC)*, level-by-level, from the coarsest level h_N all the way up to the finest level h_0. The CGC of the solution on level h_i to that on level h_{i-1} is calculated as

$$\Delta\hat{u}^{i-1} = P_i^{i-1}[\hat{u}^i - I_{i-1}^i \hat{u}^{i-1}] \tag{1.84}$$

P_i^{i-1} is called a *prolongation* operator, and interpolates a grid function from the coarse to fine mesh. On the finer mesh, $\hat{u}^{i-1}$ is updated via $\hat{u}^{i-1} = \hat{u}^{i-1} + \Delta\hat{u}^{i-1}$. The CGC step from level h_{i-1} to h_i reduces low-frequency components of the residual on level h_i, but typically also introduces high frequency noise there. Therefore, one usually applies a few *post-CGC* sweeps of the relaxation operator on the corrected solution to eliminate this noise (the relaxation steps performed during the first stage of the V-cycle are similarly referred to as *pre-CGC* relaxation sweeps).

Figure 1.5 is a pseudo code summary of the FAS, V-cycle MG algorithm just described.

Restriction/Prolongation Operators

There are a large set of possible restriction and prolongation operators that one can use in MG. We will not go into any of the details describing these choices, or requirements that a restriction-prolongation pair must satisfy (see [74] for details); rather, we simply list the operators that we use, namely ***half-weight restriction*** and ***bilinear interpolation***.

Given a two dimensional fine-grid function $u^h_{i,j}$, and corresponding coarse grid function $u^{2h}_{I,J}$, where $(i,j) \in [1..N_i, 1..N_j]$ and $(I,J) \in [1..(N_i-1)/2, 1..(N_j-1)/2]$ are indices labeling identical points of the coordinate system within their respective meshes, the half-weight restriction operator is defined as

$$\begin{aligned} u^{2h}_{I,J} &= \frac{1}{2}u^h_{i,j} + \frac{1}{8}\left(u^h_{i+1,j} + u^h_{i-1,j} + u^h_{i,j+1} + u^h_{i,j-1}\right), \\ &\qquad i \in 2..N_i-1,\ j \in 2..N_j-1, \\ &= u^h_{i,j} \quad \text{otherwise.} \end{aligned} \tag{1.85}$$

The bilinear interpolation operator is

$$\begin{aligned} u^h_{i,j} &= u^{2h}_{I,J}, \\ u^h_{i\pm1,j} &= \frac{1}{2}\left(u^{2h}_{I,J} + u^{2h}_{I\pm1,J}\right), \quad i\pm1 \in 1..N_i, \\ u^h_{i,j\pm1} &= \frac{1}{2}\left(u^{2h}_{I,J} + u^{2h}_{I,J\pm1}\right), \quad j\pm1 \in 1..N_j, \\ u^h_{i\pm1,j\pm1} &= \frac{1}{4}\left(u^{2h}_{I,J} + u^{2h}_{I\pm1,J} + u^{2h}_{I,J\pm1} + u^{2h}_{I\pm1,J\pm1}\right), \\ &\qquad i\pm1 \in 1..N_i, j\pm1 \in 1..N_j. \end{aligned} \tag{1.86}$$

The combination of half-weight restriction and bilinear interpolation tends to work well with red-black NGS relaxation [74].

[15]Except in certain Brill wave dominated spacetimes, see Section 2.3.8, though it is still unclear whether the MG problems we have then *are* related to low-frequency instabilities of relaxation on the coarsest grid.

```
subroutine FAS_MG
   do while (residual > tolerance)
      call vcycle()
   end do
end of subroutine FAS_MG

subroutine vcycle()
   do i = 0 to N-1 (fine to coarse)
      repeat pre_CGC times:
         perform a relaxation sweep on level(h[i])

      compute the residual on level(h[i])

      restrict grid functions and residual to level(h[i+1])

      compute the truncation error of the solution on level(h[i+1])

      compute the new RHS vector for level(h[i+1]), by adding the
         restricted residual and RHS to the truncation error
   end do

   solve the system of FDEs on level(h[N]) exactly

   do i = N-1 to 0 (coarse to fine)
      compute the CGC from level(h[i+1]) to level(h[i])

      apply the CGC to unknown variables at level(h[i])

      repeat post_CGC times:
         perform a relaxation sweep on level(h[i])
   end do
end of subroutine vcycle
```

Figure 1.5: A pseudo-code representation of the FAS, V-cycle multigrid algorithm.

Boundary Conditions and Multigrid

We will not discuss in any detail how boundary conditions should be handled in MG; suffice it to say that the boundary conditions should be treated as an independent difference operator L_{b}^h within the MG framework. Hence, L_{b}^h should be applied so as to smooth the residual on the boundary.

CHAPTER 2

NUMERICAL EVOLUTION IN AXISYMMETRY

In this chapter we describe the formalism and corresponding numerical programs that we have written to study axisymmetric gravitational collapse. The unigrid version of the code was started about five years years ago by Choptuik, Hirschmann and Liebling. The AMR version of the code is an extension of a 1D AMR driver written by Choptuik. I began working on the axisymmetric project almost two years ago, and am predominantly responsible for the black hole excision and AMR aspects of it. In particular, I performed all of the simulations and data analysis of the scalar field collapse study presented in Chapter 3, and the excision results presented in Chapter 4.

Earlier numerical studies of axisymmetric spacetimes include: head-on black hole collisions [77, 78, 89, 79, 80, 81]; the collapse of rotating fluid stars [82, 67, 83, 85]; the evolution of collisionless particles applied to study the stability of star clusters [86], disk collapse [87], and the validity of cosmic censorship [88]; evolution of gravitational waves [94, 84]; black hole-matter-gravitational wave interactions [90, 91, 92, 93]; and the formation of black holes through gravitational wave collapse [95] and corresponding critical behavior at the threshold of formation [96].

As discussed in the introduction, there are two primary reasons to focus on a symmetry-reduced problem, in particular axisymmetry. First, axisymmetry still allows one to model phenomena such as black hole collisions, neutron stars, gravitational collapse, etc.; and second, axisymmetry reduces the complexity of the problem so that one can achieve reasonable accuracy via computer simulation, using current technology. There has not been a very strong effort over the past two decades to numerically explore the full breadth of axisymmetric, general relativistic physics (the studies listed in the previous paragraph constitute almost all of the related work done during this period). Some of the reasons for this include: inadequate computer power, concerns about axis regularity problems that can lead to instabilities, and, to some extent, a lack of interest, for recently most research effort in numerical relativity has been directed towards trying to solve the binary black hole problem in 3D. With regards to regularity, numerical dissipation (see Section 1.5.4) and the use of appropriate variables can eliminate this problem. Available computer power is also becoming less of an issue, to the extent that many interesting problems can be solved with a serial code (such as the current version of our code) in a reasonable amount of time, especially when incorporating AMR. However, to fully explore axisymmetric phenomena, parallel codes will be needed (for the next few years at least, even if the trend in the increase of computer power continues).

In Section 2.1 below we describe the 2+1+1 formalism, which is a particular decomposition of the Einstein field equations into a set of PDEs. In Section 2.2 we describe the unigrid code which solves these equations (without rotation at this stage), and in Section 2.3 we describe the AMR extension of the unigrid code (which, in addition to the angular momentum restriction, currently does not incorporate black hole excision). Finally, in Section 2.4, we describe how we solve the geodesic equations during a numerical evolution, to help probe the geometry of a solution.

2.1 The 2+1+1 Formalism

Here we describe the so-called *2+1+1 formalism* [21, 67] that we use to arrive at the specific differential equations which we solve. This formalism is the familiar ADM decomposition, as described in Section 1.5, applied to a dimensionally reduced spacetime. The dimensional reduction is performed by dividing out the axial Killing vector, following a method devised by Geroch [23]; the results are analogous to a Kaluza-Klein reduction [97], in that a 4D axisymmetric spacetime is described by 3D gravity with additional, effective "matter" fields. We now summarize the formalism developed by Maeda *et al* and Geroch, modifying the notation slightly to suite our needs.

We are concerned with axisymmetric spacetimes. This implies the existence of a rotational vector field ξ^α, satisfying Killing's equation:

$$\mathcal{L}_\xi g_{\alpha\beta} = \xi_{[\alpha,\beta]} = 0, \tag{2.1}$$

where $g_{\alpha\beta}$ is the four dimensional spacetime metric. We choose a coordinate system adapted to the Killing field, so that ξ^α takes the form

$$\xi^\alpha \equiv \left(\frac{\partial}{\partial\phi}\right)^\alpha. \tag{2.2}$$

In other words, ϕ is our azimuthal symmetry coordinate, and, as can be shown by using Killing's equation (2.1), the components of the metric are independent of ϕ. We introduce the scalar s and vector ω^α to be the magnitude and "twist" of ξ^α respectively:

$$s^2 \equiv \xi^\alpha \xi_\alpha, \tag{2.3}$$

$$\omega^\alpha \equiv \epsilon^\alpha{}_{\beta\gamma\delta}\xi^\beta\xi^{\delta;\gamma}, \tag{2.4}$$

and define

$$\omega^2 \equiv \omega^\alpha\omega_\alpha. \tag{2.5}$$

Notice that $\omega^\alpha\xi_\alpha = 0$. Next, we define the projection tensor $\mathcal{H}_{\alpha\beta}$ by

$$\mathcal{H}_{\alpha\beta} \equiv g_{\alpha\beta} - \frac{\xi_\alpha\xi_\beta}{s^2}. \tag{2.6}$$

$\mathcal{H}_{\alpha\beta}$ is used to project tensorial objects from the 4D manifold to the 3D one, via

$$\perp T_{ab...}{}^{cd...} \equiv T_{\alpha\beta...}{}^{\gamma\delta...}\mathcal{H}^\alpha{}_a\mathcal{H}^\beta{}_b \; ... \; \mathcal{H}_\gamma{}^c\mathcal{H}_\delta{}^d \; ... \tag{2.7}$$

$\perp T_{ab...}{}^{cd...}$ is an intrinsically three dimensional object, in that contraction of any of its indices with ξ^α gives zero. We have therefore (in slight abuse of notation) labeled the components of the projected tensor with Latin indices to signify this. In practice, the components of some three dimensional tensor are easily obtain from the corresponding components of the four dimensional tensor when both are expressed in contravariant form [1]. This is because the vector components of ξ^α for $\alpha \neq \phi$ are zero by (2.2), and hence there is a one-to-one correspondence between non-ϕ vector components of a 4D tensor and its projection, as can be seen by decomposing a general 4D tensor into tensors orthogonal to and tangent to ξ^α. As an example, if $x^4 = \phi$, and $x^1..x^3$ are the remaining three dimensional coordinates, then the contravariant components of the three dimensional metric tensor are, from (2.6)

$$\mathcal{H}^{ab} = g^{ab}, \quad a,b \in (1,2,3). \tag{2.8}$$

Defining the three dimensional covariant derivative, D_a, by

$$D_a = \mathcal{H}_a{}^\beta\nabla_\beta, \tag{2.9}$$

one can show (see [23]) that the Ricci tensor ${}^{(3)}R_{ab}$ intrinsic to the three dimensional geometry of

[1]This is in contrast to the ADM decomposition, where the translation is most easily performed with tensors in covariant form.

$\mathcal{H}_{ab}$ is related to the projected four dimensional Ricci tensor $\perp R_{ab}$ via

$$ {}^{(3)}R_{ab} = \frac{1}{2s^4}\left[\omega_a\omega_b - \mathcal{H}_{ab}\omega^2\right] + \frac{1}{s}D_aD_bs + \perp R_{ab}. \tag{2.10}$$

We use the symbol τ_{ab} to denote the projected four dimensional stress-energy tensor

$$\tau_{ab} \equiv T_{\gamma\delta}\mathcal{H}^{\gamma}{}_{a}\mathcal{H}^{\delta}{}_{b}, \tag{2.11}$$

from which it follows that

$$\tau = T - \frac{T_{\phi\phi}}{s^2}, \tag{2.12}$$

where τ is the trace of τ_{ab}, T is the trace of $T_{\alpha\beta}$, and $T_{\phi\phi}$ is the $\phi\phi$ component of $T_{\alpha\beta}$. Then, using the Einstein equations (1.10), and the definition of the Einstein tensor in 3D:

$$ {}^{(3)}G_{ab} = {}^{(3)}R_{ab} - \frac{1}{2}\mathcal{H}_{ab}\ {}^{(3)}R, \tag{2.13}$$

we can rewrite (2.10) as

$$ {}^{(3)}G_{ab} = 8\pi T^{\mathrm{E}}{}_{ab}, \tag{2.14}$$

where the effective 3-dimensional stress-energy tensor $T^{\mathrm{E}}{}_{ab}$ is

$$T^{\mathrm{E}}{}_{ab} = \tau_{ab} - \frac{1}{4}\mathcal{H}_{ab}\left(\tau - \frac{T_{\phi\phi}}{s^2}\right) + \frac{1}{8\pi}\left\{\frac{\omega_a\omega_b}{2s^4} + \frac{1}{s}\left[D_aD_bs - \frac{1}{2}\mathcal{H}_{ab}D^2s\right]\right\}. \tag{2.15}$$

Restating the above steps—the dimensional reduction of the Ricci tensor results in (2.10), which allows us to write a three dimensional Einstein equation (2.14), where all effects associated with the extra dimension along the direction of the Killing vector are now described by additional "matter" fields, namely the scalar s and twist vector ω^{α}.

To complete the description of the effective three dimensional theory, we need equations of motion for s and ω^{α}. These are obtained by taking the curl and divergence of (2.4), and the Laplacian of s, and applying various definitions and identities, including the fact that ξ^{α} is a Killing vector. Here, we simply state the end results, again referring the interested reader to [23] for details of the tensor algebra used:

$$\begin{aligned} D_{[a}\omega_{b]} &= 8\pi s\epsilon_{abc}\tau^{c}, && (2.16)\\ D^{a}\left[\frac{\omega_a}{s^3}\right] &= 0, && (2.17)\\ \frac{1}{s}D^2s &= -\frac{\omega^2}{2s^4} - 4\pi\left(\frac{T_{\phi\phi}}{s^2} - \tau\right), && (2.18)\end{aligned}$$

where $\tau^{c} \equiv T^{\alpha}{}_{\phi}\mathcal{H}_{\alpha}{}^{c}$. All of equations (2.16), (2.17) and (2.18) are evolution-type equations, except for one component of equation (2.16) (the "t" component), which would be the equivalent of the angular momentum constraint equation in a 4D ADM decomposition.

In summary, equations (2.14-2.18) are equivalent to the four dimensional Einstein Equations in a spacetime with Killing vector ξ^{α}. The remainder of the $2+1+1$ formalism involves applying the ADM space plus time decomposition, as described in Section 1.5, verbatim, to the three dimensional spacetime with metric $\mathcal{H}_{ab}$, governed by the three dimensional Einstein equations of (2.14). Therefore, for brevity we will not repeat the decomposition here, rather, we only list a few definitions that will be used in subsequent sections. n^{a} is the unit time-like vector, normal to $t =$ const. hypersurfaces within the 3 dimensional manifold. This gives the decomposition

$$H_{ab} = \mathcal{H}_{ab} + n_an_b, \tag{2.19}$$

where H_{ab} is the 2 dimensional spatial metric tensor of the t = const. hypersurfaces. The corresponding extrinsic curvature tensor is defined to be

$$^{(2)}K_{AB} = -H_A{}^a H_B{}^b \frac{1}{2}\mathcal{L}_n H_{ab}. \tag{2.20}$$

2.1.1 A Scalar Field Matter Source

In the first version of our code, we include a real, massless scalar field Φ, as the matter source. The Lagrangian for the scalar field is

$$\mathcal{L}_\Phi = -\Phi_{;\alpha}\Phi_{;\beta}g^{\alpha\beta}. \tag{2.21}$$

Note that this differs by a factor of 2 from the convention in Hawking and Ellis [1], which amounts to rescaling Φ by a factor of $\sqrt{2}$. From the Lagrangian we can obtain the stress-energy tensor for Φ

$$T^\Phi{}_{\alpha\beta} = 2\Phi_{;\alpha}\Phi_{;\beta} - g_{\alpha\beta}\Phi_{;\delta}\Phi^{;\delta}, \tag{2.22}$$

and its equation of motion, namely the wave equation

$$\Box\Phi = 0. \tag{2.23}$$

2.1.2 Incorporating Angular Momentum using a Complex Scalar Field

An axisymmetric scalar field cannot carry any angular momentum[2]. To see this, look at the expression for the total, conserved angular momentum of all the matter fields within the spacetime:

$$J = -\int_\Sigma T_{\alpha\beta}\xi^\alpha n^\beta \sqrt{-h}\; d^3x, \tag{2.24}$$

where the integration is over a surface t = const., denoted here by Σ, and h is the determinant of the corresponding metric tensor. That this quantity is conserved can be verified by applying Gauss' theorem to convert the difference in J between two times, corresponding to hypersurfaces Σ_1 and Σ_2, into a volume integral (assuming that there is no contribution from spatial infinity). This volume integral will be zero, because $T_{\alpha\beta}$ is divergence free, and ξ^α is a Killing vector and thus satisfies (2.1); therefore J must the same on Σ_1 and Σ_2.

For the scalar field stress-energy tensor (2.22), the total angular momentum is given by

$$\begin{aligned} J_\Phi &= -2\int_\Sigma \left[\Phi_{;\alpha}\Phi_{;\beta} - g_{\alpha\beta}\Phi_{;\delta}\Phi^{;\delta}\right]\xi^\alpha n^\beta\sqrt{-h}\; d^3x & (2.25)\\ &= -2\int_\Sigma \Phi_{,\phi}\Phi_{;\beta}n^\beta\sqrt{-h}\; d^3x, & (2.26)\end{aligned}$$

where in the second line above we have substituted in (2.2), and utilized the fact that $n^\alpha\xi_\alpha = 0$. Thus, if Φ does not have any angular dependence, $J_\Phi = 0$.

One obvious way to add angular momentum to the problem is to use a fluid as the matter source (as originally done by Maeda *et al* [21]). We would like to do so eventually; however, for the moment we are most interested in gravitational phenomena, and would thus like to keep the matter relatively "simple" to deal with numerically. Therefore, we must add ϕ dependence to the scalar field in a manner that does not introduce ϕ dependence into the stress-energy tensor. One can do this with a combination of scalar fields, each given a ϕ dependence that is out of phase with that of the other fields, so as to "constructively interfere" in the resultant stress-energy tensor

[2]Neither, for that matter, can gravitational waves in axisymmetry—see for instance Wald [29] pg. 297.

(which is a sum of the stress-energy tensors of the individual fields)[3]. One easy way to implement this construction is to use a single, complex valued scalar field, as we describe in the remainder of this section.

The Lagrangian we use for the complex scalar field Ψ is

$$\mathcal{L}_\Psi = -\Psi_{;\alpha}\bar{\Psi}_{;\beta}g^{\alpha\beta} - m^2\Psi\bar{\Psi}, \tag{2.27}$$

where $\bar{x}$ denotes the complex conjugate of a quantity x, and we have given the field a mass m. The stress-energy tensor for Ψ is

$$T^{\Psi}{}_{\alpha\beta} = \left[\Psi_{;\alpha}\bar{\Psi}_{;\beta} + \bar{\Psi}_{;\alpha}\Psi_{;\beta}\right] - g_{\alpha\beta}\left[|\nabla\Psi|^2 + m^2|\Psi|^2\right], \tag{2.28}$$

where $|\nabla\Psi|^2 = \Psi_{;\alpha}\bar{\Psi}^{;\alpha}$ and $|\Psi|^2 = \Psi\bar{\Psi}$. Ψ satisfies the following wave equation

$$\Box\Psi = m^2\Psi. \tag{2.29}$$

Taking the complex conjugate of (2.29) gives the wave equation for $\bar{\Psi}$.

We give Ψ the following angular dependence:

$$\Psi(\phi,\rho,z,t) = \Psi_0(\rho,z,t)e^{i\omega_0\phi}, \tag{2.30}$$

where the constant ω_0 is an integer (for regularity of the field). With this form for Ψ it is not difficult to see that (2.28) is independent of ϕ. However, the total angular momentum of the field can be non-zero:

$$\begin{aligned} J_\Psi &= -\int_\Sigma \left[\Psi_{,\phi}\bar{\Psi}_{;\beta} + \bar{\Psi}_{,\phi}\Psi_{;\beta}\right] n^\beta\sqrt{-h}\; d^3x & (2.31)\\ &= 2\omega_0\int_\Sigma \mathcal{I}_m\left[\Psi_0\bar{\Psi}_{0;\beta}\right] n^\beta\sqrt{-h}\; d^3x, & (2.32)\end{aligned}$$

where $\mathcal{I}_m[x]$ denotes the imaginary part of x.

2.2 Structure of the Unigrid Numerical Code

This section describes a unigrid program implementing a version of the 2+1+1 formalism described in Section 2.1. This code only incorporates a real scalar field as the matter source, and thus excludes the effects of angular momentum. We describe the coordinate system and set of variables we have chosen to use in Section 2.2.1, with corresponding regularity and outer boundary conditions in Section 2.2.2. In Section 2.2.3 we state the initial data that we specify. The discretization scheme is described in Section 2.2.4. Section 2.2.5 discusses the apparent horizon finder we use in the code. And finally, in Section 2.2.6 we describe our implementation of black hole excision in the unigrid code.

2.2.1 Coordinate System and Variables

We have chosen to use a cylindrical (ρ, z) coordinate system, and coordinates where the 2D spatial metric H_{AB} (2.19) is conformally flat (all variables introduced here, as well as those utilized from Section 2.1, are functions of ρ,z and t):

$$ds^2 = \psi^4\left(dz^2 + d\rho^2\right). \tag{2.33}$$

[3]This is similar to the technique used to obtain static boson stars[98, 99], as well as that used to study angular momentum in spherical gravitational collapse [100].

To complete the specification of the 3-metric $\mathcal{H}_{ij}$ (2.6), we include a lapse function α, and the ρ and z components of the shift vector, β^ρ and β^z respectively. We define the magnitude of the Killing vector s^2 (i.e. $g_{\phi\phi}$) to be

$$s^2 = \psi^4 \rho^2 e^{2\rho\bar{\sigma}} \tag{2.34}$$

Without angular momentum, the twist vector (2.4) is zero. Incidentally, this implies that $g_{\phi x} = 0$, where $x \in [\rho, z, t]$, which is exactly the spacetime decomposition that the 4D ADM formalism would yield, in similar coordinates. In the numerical code we have decided to use $\bar{\sigma}$ as the fundamental variable that is evolved, rather than s. One of the reasons for using $\bar{\sigma}$ is that its leading order behavior in the limit as $\rho \to 0$ is $\bar{\sigma} \propto \rho + O(\rho^3)$, and to achieve stable evolution it appears to be easier to enforce such a regularity condition, rather than that of s, which involves higher powers of ρ: $s \propto \rho^2 + O(\rho^4)$.

We convert the resultant second-order-in-time evolution equation for $\bar{\sigma}$ to first-order-in-time form by defining a conjugate variable $\bar{\Omega}$ as follows

$$\begin{aligned} \rho\bar{\Omega} &\equiv -2^{(2)}K_\rho{}^\rho -^{(2)} K_z{}^z & (2.35)\\ &= -2\,\chi + \frac{\beta^z_{,z} - \beta^\rho_{,\rho}}{2\alpha}, & (2.36)\end{aligned}$$

where χ is given by

$$\chi = \frac{-1}{s} n^a \partial_a s. \tag{2.37}$$

The choice of this particular form for $\bar{\Omega}$ was motivated by regularity concerns, for certain terms of order $1/\rho$ in the vicinity of the axis cancel in (2.35), and, as with $\bar{\sigma}$, the leading order behavior of $\bar{\Omega}$ in the limit as $\rho \to 0$ is $\bar{\Omega} \propto \rho + O(\rho^3)$.

To convert the wave equation for the scalar field variable Φ to first-order-in-time form, we define

$$\Pi_\Phi = \psi^2 n^a \partial_a \Phi. \tag{2.38}$$

For a slicing condition, we have chosen to implement *maximal slicing*, where the trace of the three dimensional extrinsic curvature tensor of t = const. slices within the four dimensional manifold is zero:

$$^{(3)}K = 0. \tag{2.39}$$

The relationship between $^{(3)}K$ and $^{(2)}K$, the trace of the extrinsic curvature tensor of the two dimensional manifold with metric H_{ab} (2.19), is

$$^{(3)}K =^{(2)} K + \chi. \tag{2.40}$$

This equation can be derived by substituting H_{ab} from (2.19) into the definition of $^{(2)}K_{AB}$ (2.20), taking the trace, and using the definition of $^{(3)}K$ (the trace of (1.29)). Therefore, the maximal slicing condition in terms of variables of the $2+1+1$ formalism can be written as

$$^{(2)}K = -\chi. \tag{2.41}$$

To summarize, the fundamental variables for the unigrid code, with a single, real scalar field, and zero angular momentum are: $\psi, \alpha, \beta^\rho, \beta^z, \bar{\sigma}, \bar{\Omega}$, Φ and Π_Φ. All of these variables are functions of ρ, z and t. The resultant field equations for the metric variables, and wave equation for the scalar field, are written out explicitly in Appendix A.1, where the conditions $^{(3)}K = 0$ and $\partial^{(3)}K/\partial t = 0$ have been used to simplify the expressions where possible. We end up with four elliptic PDEs—the slicing condition for α (A.2), the Hamiltonian constraint which is used to solved for ψ (A.3), and the two non-trivial momentum constraints that we use to solve for β^ρ (A.4) and β^z (A.5). The remaining equations are hyperbolic in nature, one of which (A.8) we optionally used to solve for ψ

in place of the Hamiltonian constraint.

2.2.2 Regularity and Outer Boundary Conditions

To complete the description of our system of equations, we need to specify boundary conditions. In our cylindrical coordinate system, where $\rho \equiv r\cos(\theta)$ ranges from $\rho = 0$ to $\rho = \rho_{\rm max}$, and $z \equiv r\sin(\theta)$ ranges from $z_{\rm min}$ to $z_{\rm max}$, we have two distinct boundaries: the physical outer boundary at $\rho = \rho_{\rm max}$, $z = z_{\rm min}$, and $z = z_{\rm max}$; and the axis, at $\rho = 0$.

At the physical boundary, we assume that the "Coulombic" variables of our metric fall off sufficiently fast to approach an asymptotically flat form, and for the scalar field and "radiative" components of the metric we assume outgoing spherical waves (described in more detail below). These particular conditions were chosen for consistency, at least to within a reasonable approximation, with our goal of simulating an isolated region of strong-field gravity in an asymptotically flat universe.

For ψ, and optionally for α, β^ρ and β^z, we assume Coulombic behavior of the form $f = f_\infty + \mathrm{k}/r$, where f denotes one of these four variables with value f_∞ at $r = \infty$, and k is a constant. By differentiating and applying the chain rule, we can convert this condition into a form that is easier to finite-difference:

$$f - f_\infty + \rho f_{,\rho} + z f_{,z} = 0 \quad \text{at} \quad \rho = \rho_{\rm max}, z = z_{\rm min}, z = z_{\rm max}. \tag{2.42}$$

For asymptotic flatness $\psi_\infty = 1$, and we choose $\alpha_\infty = 1, \beta^\rho_\infty = 0$ and $\beta^z_\infty = 0$. The alternative boundary condition that we sometimes employ for α, β^ρ or β^z, is to enforce the required behavior at infinity as a Dirichlet condition at the outer boundary of the grid; i.e. $f = f_\infty$ at $\rho = \rho_{\rm max}, z = z_{\rm min}$ or $z = z_{\rm max}$.

For the remaining radiative variables $\bar{\sigma}, \bar{\Omega}, \Phi$ and Π_Φ, denoted by g, we assume that at the outer boundary they take on the form $g = h(t-r)/r$, where h is an arbitrary function. Again, we convert this condition to a form better suited for finite-differencing:

$$r g_{,t} + \rho g_{,\rho} + z g_{,z} + g = 0 \quad \text{at} \quad \rho = \rho_{\rm max}, z = z_{\rm min}, z = z_{\rm max}. \tag{2.43}$$

See Appendix A.1 for all of the outer boundary conditions written out explicitly for each variable.

The above classification of a metric variable as radiative or Coloumbic is somewhat arbitrary, in particular given the form that we have chosen for the metric[4]. The result is that we get a certain amount of reflection of waves that should propagate past the outer boundary, the effect being more pronounced the closer the outer boundaries are to the region of strong-field evolution, or the further the initial data and evolution deviate from spherical symmetry.

On the axis, $\rho = 0$, we enforce regularity of the metric and matter variables. The regularity conditions can be obtained by inspection of the equations in the limit $\rho \to 0$, or more formally, by transforming to Cartesian coordinates and demanding that components of the metric and matter fields be regular and single valued throughout [101]. Also, as discussed earlier, the particular choice

[4]See [52] for choices of metric variables that do, at large r, reduce to a form that can be identified as purely radiative, and furthermore, combinations of them that directly give components of the gravitational waveforms as given by linearized theory.

of $\bar{\sigma}$ and $\bar{\Omega}$ as fundamental variables was also motivated by regularity concerns. The results are:

$$\begin{aligned}
\alpha_{,\rho}(t,0,z) &= 0, && (2.44)\\
\psi_{,\rho}(t,0,z) &= 0, && (2.45)\\
\beta^z_{,\rho}(t,0,z) &= 0, && (2.46)\\
\beta^\rho(t,0,z) &= 0, && (2.47)\\
\bar{\sigma}(t,0,z) &= 0, && (2.48)\\
\bar{\Omega}(t,0,z) &= 0, && (2.49)\\
\Phi_{,\rho}(t,0,z) &= 0, && (2.50)\\
\Pi_{\Phi,\rho}(t,0,z) &= 0. && (2.51)
\end{aligned}$$

2.2.3 Initial Data

To obtain initial data that is consistent with the constraint equations and our maximal slicing condition, we freely specify the functions $\Phi, \Pi_\Phi, \bar{\sigma}$ and $\bar{\Omega}$ at $t = 0$, then solve the constraint equations (A.3-A.5) for ψ, β^ρ and β^z, and the slicing condition (A.2) for α.

In general, we use the following six parameter $(A, \epsilon, \rho_0, z_0, R_0, \Delta_0)$ function to specify the initial data for the free variables at $t = 0$:

$$G(\rho,z) = A \exp\left[-\left(\frac{\sqrt{(\rho-\rho_0)^2+\epsilon\,(z-z_0)^2}-R_0}{\Delta}\right)^2\right] \qquad (2.52)$$

Roughly speaking, G describes a ring of radius R_0, centered at (ρ_0, z_0), where each cross-section of the ring is Gaussian-shaped with amplitude A and average width Δ; a value $\epsilon \neq 1$ will produce a prolate or oblate distortion of the ring. A different set of 6 parameters is specified for each of $\Phi, \Pi_\Phi, \bar{\sigma}$ and $\bar{\Omega}$. After initialization with (2.52), both $\bar{\sigma}$ and $\bar{\Omega}$ are multiplied by ρ to satisfy the regularity conditions.

In some simulations, modified forms of (2.52) are used; these modifications will be discussed in the sections presenting the results of the corresponding simulations.

2.2.4 Discretization and Solution Scheme

In this section we describe the discretization scheme we use to convert the differential equations to finite difference form, and the methods used to solve the resulting finite difference equations.

We use a uniform grid of size N points in ρ by M points in z, with equal spacing $\Delta\rho = \Delta z$ in the ρ and z directions. The value of a function f at time level n and location (i,j) within the grid, corresponding to coordinate $(t,\rho,z) = (t,(i-1)\Delta\rho,(j-1)\Delta_z + z_{\min})$, is denoted by $f^n_{i,j}$. We employ a second-order accurate Crank-Nicholson type discretization of the evolution equations, whereby we define two time levels, t and $t + \Delta t$, and obtain our finite difference stencils by expanding in Taylor series about $t = t + \Delta t/2$. This gives the following second-order accurate approximation to a first-time derivative of f:

$$\frac{\partial f(t,\rho,z)}{\partial t}\Big|_{t=t+\Delta t/2} \Rightarrow \frac{f^{n+1}_{i,j} - f^n_{i,j}}{\Delta t} \qquad (2.53)$$

Second-order accurate approximations to functions and spatial derivative operators at $t = t+\Delta t/2$ are obtained by averaging the corresponding quantity Q in time:

$$Q(t+\Delta t/2,\rho,z) \Rightarrow \frac{Q^n_{i,j} + Q^{n+1}_{i,j}}{2} \qquad (2.54)$$

After discretization of the evolution equations using (2.53) and (2.54), function values are only referenced at times t and $t + \Delta t$, even though the stencils are centered at time $t + \Delta t/2$. Specific forms for all the finite difference stencils that we use can be found in Appendix A.2. Note that we also add Kreiss-Oliger style dissipation (1.69) to the evolution equations—this is essential to obtain stable (and thus convergent) evolution in our case.

The constraint and slicing equations (A.2-A.5) do not contain time derivatives, and hence for the multigrid (MG) relaxation scheme they are differenced at a single time. We employ the standard, second-order accurate, centered finite-difference stencils for all interior operators (1.60,1.65). The Neumann conditions on the axis (see (2.44-2.51)) are discretized using the following second-order accurate stencil:

$$\frac{\partial f(t,\rho,z)}{\partial \rho}|_{\rho=0} \Rightarrow \frac{-3f^n_{1,j} + 4f^n_{2,j} - f^n_{3,j}}{2\Delta\rho} \tag{2.55}$$

At the outer boundaries, we discretize boundary conditions involving spatial derivatives (2.42) using the following *symmetric scheme*[5], described here for the $\rho = \rho_{\max}$ boundary. We want to apply a boundary condition to solve for the variable f at grid location (N, j), corresponding to coordinate position $(\rho_{\max}, z)$. However, in the symmetric scheme, we apply the continuum conditions at coordinate position $(\rho_{\max} - \Delta\rho/2, z)$, similar to the Crank-Nicholson method we use to discretize in time. Therefore, ρ derivatives are replaced with the following second order accurate operator

$$\frac{\partial f(t,\rho,z)}{\partial \rho}|_{\rho=\rho_{\max}-\Delta\rho/2} \Rightarrow \frac{f^n_{N,j} - f^n_{N-1,j}}{\Delta\rho} \tag{2.56}$$

Z derivatives, and other grids functions that need to be evaluated at $(\rho_{\max} - \Delta\rho/2, z)$, are replaced by the average of the corresponding quantity discretized at $(\rho_{\max}, z)$ and $(\rho_{\max} - \Delta\rho, z)$. For example,

$$\frac{\partial f(t,\rho,z)}{\partial z}|_{\rho=\rho_{\max}-\Delta\rho/2} \Rightarrow \frac{1}{2}\left(\frac{f^n_{N,j+1} - f^n_{N,j-1}}{2\Delta z} + \frac{f^n_{N-1,j+1} - f^n_{N-1,j-1}}{2\Delta z}\right) \tag{2.57}$$

The $z_{\max}$ and $z_{\min}$ boundaries are handled in a similar fashion, as are the corner points at $(\rho_{\max}, z_{\max})$ and $(\rho_{\max}, z_{\min})$; however at the corner points the continuum equations are evaluated at $(\rho_{\max} - \Delta\rho/2, z_{\max} - \Delta z/2)$ and $(\rho_{\max} - \Delta\rho/2, z_{\min} + \Delta z/2)$ respectively.

Our multigrid routine is as described in Section 1.5.6. We utilize a pointwise Newton-Gauss-Seidel relaxation method, updating simultaneously the block of four unknowns $(\psi, \alpha, \beta^\rho, \beta^z)$ [6] at each interior grid point, to smooth the residual. The grid points are relaxed in red-black order. After each interior sweep, the boundary conditions are enforced via the same relaxation method (though in lexicographic ordering about the boundary).

With the above described discretization scheme, we use the following iterative time-stepping routine to solve for the set of unknown variables $(\psi, \alpha, \beta^\rho, \beta^z, \bar{\sigma}, \bar{\Omega}, \Phi, \Pi_\Phi)$ at time $t + \Delta t$, given known function values at time t:

```
As an initial guess to the solution at time t+dt, copy variables
   from t to t+dt

repeat until (residual norm < tolerance):
   1: perform 1 Newton-Gauss-Seidel relaxation sweep of the
```

[5]The name symmetric refers to the property this scheme often gives (at least to simple equations) to the resulting matrix representing the linearized system of equations. The reason for doing this is that symmetric matrices tend to have better convergence properties in an iterative solution method. However, we have experimented with an alternative, non-symmetric differencing scheme at the outer boundaries, without seeing a significant deterioration in the robustness of the MG.

[6]Or three unknowns, if ψ is freely evolved.

```
     evolution equations, solving for the unknowns at time t+dt
  2: perform 1 MG vcycle on the set of elliptic equations at
     time t+dt
end repeat
```

We set $\Delta t = \lambda \cdot \min(\Delta\rho, \Delta z)$, where $\lambda < 1$ to satisfy the CFL condition. The code implementing this scheme has been written in a combination of RNPL and Fortran 77. RNPL (Rapid Numerical Prototyping Language [106]) is used for the majority of the code, taking as input a set of operator definitions, residual statements and grid definitions, and producing as output a complete Fortran program. We use a custom Fortran routine to implement the MG solver.

2.2.5 Finding Apparent Horizons

In this section we describe the method that we use to search for apparent horizons (AH's). We restrict our search to isolated, simply connected AH's. In axisymmetry, such an AH can be described by a curve in the (ρ, z) plane, starting and ending on the axis at $\rho = 0$. Therefore, the equation for the apparent horizon (1.44, for $\theta_+ = 0$) can be reduced to an ordinary differential equation, by (for instance) defining the level surfaces in (1.41) as follows:

$$f_s = \bar{r} - R(\bar{\theta}), \tag{2.58}$$

where

$$\begin{aligned} \bar{r} &= \sqrt{\rho^2 + (z - z_0)^2}, & (2.59)\\ \rho &= \bar{r}\sin\bar{\theta}, & (2.60)\\ (z - z_0) &= \bar{r}\cos\bar{\theta}. & (2.61) \end{aligned}$$

Thus, $f_s =$ const. describes a family of spheroidal surfaces (when spun about the axis), centered at $(0, z_0)$. When (2.58) is substituted into (1.44) (for $\theta_+ = 0$), and defining $f_s = 0$ to be the surface with zero expansion, we obtain a second order, ordinary differential equation for $R(\bar{\theta})$:

$$R''(\bar{\theta}) + F(R'(\bar{\theta}), R(\bar{\theta}), \psi, \alpha, \beta^\rho, \beta^z, \bar{\sigma}, \bar{\Omega}, \psi_{,\rho}, \beta^\rho_{,\rho}, \beta^z_{,\rho}, \bar{\sigma}_{,\rho}, \psi_{,z}, \beta^\rho_{,z}, \beta^z_{,z}, \bar{\sigma}_{,z}, \rho, z) = 0. \tag{2.62}$$

Here F is a rather lengthy function of its arguments, and is non-linear in R and R' (the prime $'$ denotes differentiation with respect to $\bar{\theta}$). All the metric functions and their gradients appearing in (2.62) are evaluated along a given curve of integration, and hence are implicitly functions of $\bar{\theta}$. $\bar{\theta}$ ranges from 0 to π. For initial conditions, we specify $R(0)$, and regularity of the surface about the axis demands that $R'(0) = 0$,

To find the AH, we employ a so-called *shooting method.* If an AH exists, and assuming $f_s(0) > z_0$, there will be a locally unique[7] value of $R(0) = R_0$ such that integration of (2.62) will end at $R(\pi) = R_\pi$, with $R'(\pi) = 0$. For $R(0) > R_0$, R will either diverge at some value of $\bar{\theta} < \pi$, or $R'(\pi) > 0$; and for $R(0) < R_0$, $R'(\pi) < 0$. Therefore, if we can find a reasonable bracket about the unknown R_0, we can use a bisection search to "shoot" towards R_0. Currently, we find a bracket to search by testing a set of initial points, equally spaced in z at intervals of $3\Delta z$. This seems to work reasonably well in most situations.

We use a second-order Runge-Kutta method to integrate equation (2.62). The metric functions appearing in F are evaluated using bilinear interpolation along the curve. We did experiment integrating (2.62) using the 5th order adaptive Runge-Kutta routine found in LSODA [107]; however, this routine would slow down significantly in the rather common situation where $R \to \infty$ at some

[7]In a general collapse scenario, multiple inner horizons could be present, which would also satisfy (2.62). We want the outermost of these surfaces.

value of $\bar{\theta}$, and did not offer much improvement in the accuracy of the solution otherwise (we would probably need to use higher-order interpolation of the metric functions to gain improvements when using LSODA).

2.2.6 Black Hole Excision Technique

In this section we describe how we implement black hole excision in the code. In particular, we focus on the definition of the surface of excision, how we modify the finite difference equations on this surface, "repopulating" grid points when tracking moving black holes, and the smoothing operators we use to achieve stable excision.

The Excision Surface

From a mathematical point of view, "excising" a region of the grid means specifying appropriate boundary conditions on a surface enclosing that region. Therefore, to maintain a well-posed description of our system of equations when excising, we analytically define the surface of excision via a *smooth* function, and then specify the boundary conditions on the surface described by this function. Specifically, we define the surface of excision as a particular value of a *monotonic* level function $f_e(\rho, z)$, so that the surface defined by $f_e = c_0$ is entirely contained within the surface $f_e = c_1$ if and only if $c_0 < c_1$. The function f_e we currently use is

$$f_e = \left(a_1(z - z_{01})^2 + \rho^2\right)^{P_1} \left(a_2(z - z_{02})^2 + \rho^2\right)^{P_2}, \tag{2.63}$$

where $P_1 \equiv m_1/(2(m_1 + m_2))$, $P_2 \equiv m_2/(2(m_1 + m_2))$, and $z_{01}, z_{02}, a_1, a_2, m_1, m_2$ are parameters controlling the shape of the level surfaces. This function was designed to allow excision of either one ($m_2 = 0$) or two distinct black holes. For spacetimes containing two black holes, we have so far only experimented with equal mass black holes, for which $m_1 = m_2$. The values of the parameters are fixed during the Crank-Nicholson iteration at a given time, but may change with time (to track a moving black hole for instance). When an AH is detected, we search for a set of parameters so that $f_e =$ const. surfaces roughly match the shape of the AH. Then we choose a number f_{e0} so that the surface $f_e(\rho, z) = f_{e0}$ is some distance *inside* the AH. We call the region between $f_e(\rho, z) = f_{e0}$ and the AH the *excision buffer zone.*

Excision Boundary Conditions

After an excision level surface $f_e = f_{e0}$ is chosen, the region of the grid where $f_e(\rho, z) < f_{e0}$ is *excised.* As mentioned before, this means that we specify boundary conditions for all of the variables in the problem on the surface $f_e(\rho, z) = f_{e0}$, and only evolve grid functions at mesh points outside of this surface. For the variables that have hyperbolic-type evolution equations, namely $\psi, \bar{\sigma}, \bar{\Omega}, \Phi$ and Π_Φ, we simply apply the evolution equations at the boundary surface. We can do this in a stable fashion because the excision surface is inside the AH, and hence the physical characteristics are all directed into the excised region (so in a certain sense we are not enforcing any boundary conditions on hyperbolic equations there). In the numerical scheme, at *boundary points* we apply the same discretized evolution equations as at interior grid points, except that we replace centered difference operators with backward/forward difference operators as required to avoid referencing the grid in the excised region. A point $\vec{x}$ of the mesh, outside of f_{e0}, is defined as a boundary point if some point in a 3x3 stencil, centered at $\vec{x}$, is within the excised region. See Figure 2.1 below for an example. An exception to this rule is used for Kreiss-Oliger dissipation operators, which are simply turned off if they extend into the excised region (though we do smooth these points using alternative smoothing operators, as discussed below). For the remaining variables, β^ρ, β^z and α, solved for using elliptic equations, we specify Dirichlet boundary conditions (described below) on the surface of excision. If ψ is solved for using the Hamiltonian constraint, then, from the point of view of the multigrid (MG) solver, the excision boundary values for ψ are also Dirichlet; however,

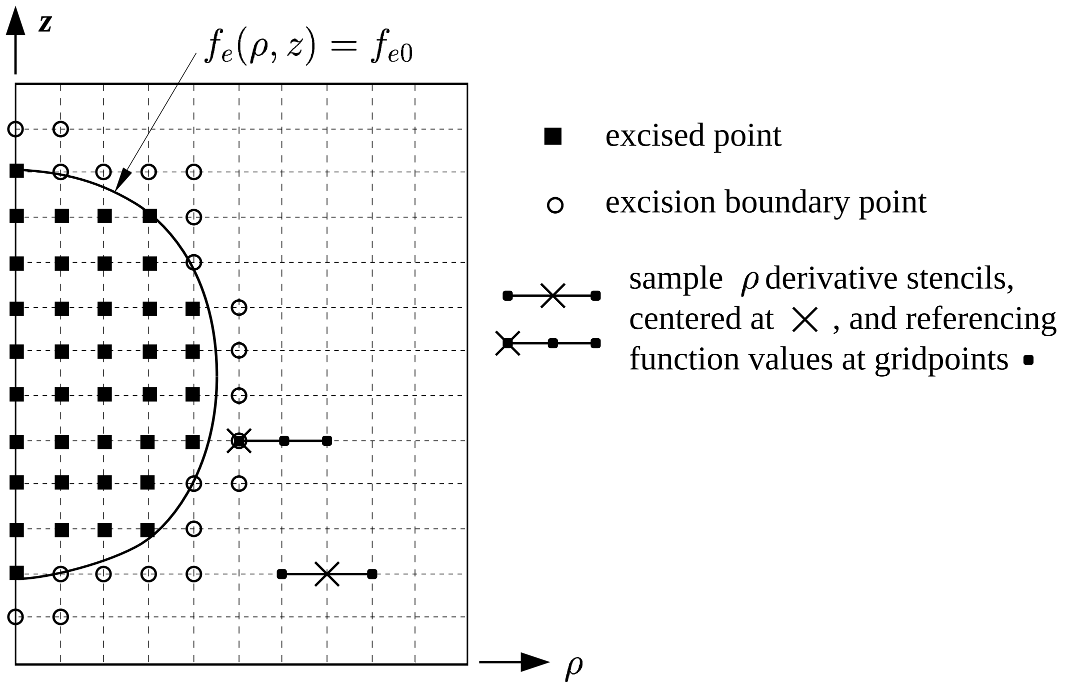

z
$f_e(\rho, z) = f_{e0}$
■ excised point
○ excision boundary point
sample ρ derivative stencils, centered at ×, and referencing function values at gridpoints ▪
ρ

Figure 2.1: The effect of an excision surface on the definition of grid points and FD stencils. In the figure, we show a sample grid, and excision surface $f_e(\rho, z) = f_{e0}$. All points on the grid within $f_e(\rho, z) < f_{e0}$ are excised. A boundary point is defined as a point at which one of the standard, interior FD stencils that we use (see Section A.2) needs to be modified in order to avoid referencing an excised point. As an example, in the figure we demonstrate how the first ρ derivative stencil changes from a centered-difference stencil to a forward-difference stencil, when applied to the right of an excised point.

the boundary values used in the MG on the advanced time level are obtained by evolving ψ from the retarded time level. Then, after the CN iteration is completed, we replace the advanced time-level evolved ψ grid function with the one computed using MG.

The MG algorithm is unaltered when excising, except in a situation where a fine-grid interior point becomes an excision boundary point after a coarsening operation. This can happen because we always define the excised region to be $f_e < f_{e0}$. When this *does* happen, Dirichlet boundary conditions are applied at the new coarse grid boundary point, with the function value set to that of the corresponding fine grid point.

Several of the boundary conditions that we have experimented with for β^ρ, β^z and α include:

- freezing the variables to the values they had at the time of excision
- linearly extrapolating the variables forward in time using the velocities measured at the time of excision (this is not a useful condition for long-term evolution, as the values of β^ρ, β^z will grow without bound, and α will go to zero in finite time if $d\alpha/dt < 0$)
- specifying some constant non-zero value for α on the surface, and choosing certain linear profiles for β^ρ and β^z to control the coordinate size and motion of the excised region. These conditions were specifically designed to treat black hole mergers—see Section 4.2 for more details.

Most of the boundary conditions appear to be reasonably stable for certain black hole spacetimes, and in a few situations where we have checked, we do seem to get convergence in the solution after excision (see Chapter 4 for examples). However, these boundary conditions do *not* appear to give results that are consistent with the remaining field equations not used during evolution. We surmise that the reason for this is as follows. The shift vectors in our coordinate system are not freely specifiable, for we have used up our "gauge-freedom" to maintain a conformally flat 2-metric H_{AB}. This is not *a-priori* a problem, as coordinate conditions result in *differential equations* for components of the shift vector, which could still be satisfied with a wide range of boundary conditions. The difficulty arises, at least in our case, because the effective evolution equations for the components of the shift vector do not permit the specification of arbitrary (regular) boundary conditions for both components of the shift vector. To see this, note the form of the $\rho\rho$ and ρz components of the extrinsic curvature in our coordinate system:

$$K_\rho{}^\rho = -\frac{1}{3\alpha}\left[\beta^z_{,z} - \beta^\rho_{,\rho} + \rho\alpha\bar{\Omega}\right] \tag{2.64}$$

$$K_\rho{}^z = \frac{1}{2\alpha}\left[\beta^\rho_{,z} + \beta^z_{,\rho}\right] \tag{2.65}$$

In a free evolution, we would evolve these two components of $K_A{}^B$, and then integrate equations (2.64) and (2.65) to obtain the shift vectors, at each time step. Notice though that these equations are *first order* coupled PDEs for β^z and β^ρ, and therefore one can only specify "initial conditions" when integrating them, which amounts to setting conditions on a limited portion of the boundary of the computational domain. This restriction holds even without an excision boundary present, and indeed we do see an inconsistency then, in the limit as $h \to 0$, with a fixed outer boundary location (see Section 2.2.7). However, consistent outer boundary conditions appear to satisfy the assumed $1/r$ fall-off behavior in β^z and β^ρ, and therefore we can make the inconsistency in an evolution without excision arbitrarily small, by moving the outer boundary far enough away (which is practical with the AMR code). Unfortunately, such a "solution" is not possible for the excision boundary, and work still needs to be done to limit the class of boundary conditions applied there to achieve fully consistent evolution.

Dissipation

To obtain stable, long-term evolution with excision, it is essential to control unphysical, high-frequency modes that inevitably develop tangential to the excision surface, in variables solved for with hyperbolic-type equations ($\psi, \Phi, \Pi_\Phi, \bar{\sigma}$ and $\bar{\Omega}$). We have experimented with several excision-boundary smoothing techniques, some more successful than others. One of the more effective techniques to control the noise is to apply a simple smoothing operation — namely, replace any point within a distance of 2 points from the excised region (i.e. those points that no longer have a Kreiss-Oliger filter applied to them) with the average value of the variable in a 3x3 sub-grid centered on the point (excluding excised points in the average). One problem with this technique though, is that it is too "aggressive", meaning that the averaging operation tends to reduce the gradient of the function normal to the surface to zero, over time. This is somewhat problematic for ψ, diverges like $1/r$ towards the black hole, and to conserve the mass of the spacetime, it is necessary to maintain this correct leading order behavior in ψ. The effective of the averaging operator does apparently converge away in the limit as $h \to 0$; however, at the typical grid resolutions that we run the unigrid code at, this smoothing can induce significant "mass loss".

At the root of the problem with the above described smoothing function, and what needs to be addressed for any high-frequency filter applied at the excision surface, is how to define exactly what one means by high-frequency, and then target it with an appropriate filter. For example, when smoothing ψ, we do not want to consider large gradients in ψ *normal* to the excision surface as high-frequency, whereas in the scalar field Φ it does no harm to smooth large normal gradients.

An early experiment we tried, to achieve a more effective filter for ψ, was to use a Kreiss-Oliger filter rotated in the coordinate grid (with a different angle at each point) to target only those high-frequency components tangent to the excision boundary. This, as implemented, was only marginally successful, for to apply the filter correctly, we needed to interpolate the grid-function values onto the rotated filter's coordinate system, and it appeared as if the interpolation scheme caused an instability in the long-run.

An alternative filtering operator that we have experimented with, and that seems to work quite well for certain evolved variables, is first to smoothly reconstruct the function within the excised region, and then to apply a (standard) Kreiss-Oliger filter that targets high-frequency components in the ρ and z directions, to the entire grid. The reconstruction function is rather *ad-hoc* in nature, designed solely to provide a smooth reference-background relative to which high-frequency modes can meaningfully be defined, and then removed. See Figure 2.2 below for a pseudo-code description of the algorithm, which operates as follows. The current reconstruction technique is (approximately) bi-linear extrapolation, applied iteratively, one layer of grid-points inwards from the excision surface per iteration. After an iteration, the set of newly extrapolated points are considered 'un-excised' for subsequent iterations[8]. The set of un-excised points used for extrapolation are spaced rather far apart (2-3 cell widths). This is done to effectively use only the lower-frequency part of the function for extrapolation, and is necessary to obtain a stable filter (which, incidentally, is why we did not try higher order extrapolation, as it tends to be more sensitive to higher-frequency irregularities in the function). Furthermore, after the first n iterations are complete, we force the extrapolated values on subsequent iterations to tend towards a local average, to prevent divergent behavior in the interior (especially for functions like ψ, which go as $1/r$). After the extrapolation, several passes of Kreiss-Oliger filters (some with their stencils expanded by factors of 2 or 3, to target lower frequency components) are applied to the excised region, to smooth away unwanted effects of the linear extrapolation. Then a single pass of a Kreiss-Oliger filter is applied to the entire grid, except in a region within 2 grid points of the outer boundary to prevent the 5-point stencil from extending off the grid. In a similar 2 point zone about the axis, we still apply the filter by "virtually" extending the function to negative values of ρ, as dictated by the even or odd character of the function in the limit $\rho \to 0$.

Grid Repopulation

Finally, we briefly comment on how grid "repopulation" is performed. When the black holes, and hence excision surfaces, are moving on the coordinate grid, or in situations where the shape of the apparent horizon changes significantly, it is sometimes necessary to 'un-excise', or repopulate portions of the grid. This is done by extrapolating the values of the functions from adjacent, pre-existing interior grid points to the newly uncovered points. The extrapolation method that has so far worked the best is to use the same reconstruction function that is used for smoothing. The other two methods that we have tried are (1) a straight-copy of function values from directly above or below the point (i.e. from the $\pm dz$ directions), and (2) bi-linear extrapolation from adjacent grid-points.

2.2.7 Convergence, Consistency and Mass Conservation Results

In this section we present convergence, consistency and mass conservation plots from two fully constrained evolutions, to demonstrate the level of "correctness" of the code. We examine three diagnostic quantities: the total ADM mass of the spacetime [43], the convergence factor Q_ψ (1.55) for ψ (other variables in the problem show similar convergence behavior as ψ, and for brevity we do not show them), and the ℓ_2 norm of the residual (1.79) of the $\rho\rho$ component of the evolution

[8]Though note that the reconstruction is a temporary operation, purely for the purpose of smoothing—we are not in any way 'un-excising' the black hole.

```
subroutine smooth_excised_function(grid function f[1..Nrho,1..Nz])
   call reconstruct(f)
   do i=1,ni
      do k=1,nk
         smooth f in the excised region only, with a Kreiss-Oliger
            filter whose stencil has been expanded by a factor of k
      end do
   end do

   smooth f over the entire grid, with the standard, un-expanded
      Kreiss-Oliger filter

   set f to 0 in the excised region

   return from subroutine
end of subroutine smooth_excised_function

subroutine reconstruct(grid function f[1..Nrho,1..Nz])
   for each (i,j) in the excised region
      avg_f(i,j) = local average of f over some portion of the
                   un-excised grid closest to (i,j)
   end for

   n=1
   repeat until all points (i,j) in the grid are unexcised:
     for each (i,j) in the excised region
        if (i,j) directly neighbors an un-excised point then
           f(i,j)=bi-linear extroplation of f from the 4 unexcised
                  points {(i0,j0),(i0+k,j0),(i0,j0+k),(i0+k,j0+k)}
                  that are nearest to (i,j) on the grid
           if (n>first_n) then
              f(i,j)=f(i,j)-epsilon*(f(i,j)-avg_f(i,j))
           end if
        end if
     end for

     mark all points (i,j) that were repopulated in the previous
        for-loop as un-excised

     n=n+1
   end repeat

   return from subroutine
end of subroutine reconstruct
```

Figure 2.2: A pseudo-code representation of the reconstruction function and smoothing filter used to smooth grid functions during excision. In smooth_excised_function(), the current value of ni = 4 and nk = 3. In reconstruct(), the current value of k = 3 and epsilon = 0.15; first_n = 0 when reconstruct() is used for smoothing, while first_n = 2 when reconstruct() is used for grid repopulation (see the next section).

equation for ${K_A}^B$, which we call $E(\dot{K_A}^B)$:

$$E(\dot{K_A^B}) \equiv \left\| \frac{\partial({K_A}^B)}{\partial t} - \text{rhs}\left(\frac{\partial({K_A}^B)}{\partial t}\right) \right\|_2 . \tag{2.66}$$

For brevity, we use the function rhs() in (2.66) to denote the right-hand-side of the evolution equation for ${K_A}^B$ (the 2-dimensional analog of (1.37)). The ADM mass is calculated via an integration over a closed surface, situated near the outer boundary of the computational domain [9]. In our coordinate system, the expression for the ADM mass is

$$\begin{aligned} M_{ADM} &= \frac{1}{2}\int_{z_{\max}} \rho\psi^4 \left(-\frac{\psi_{,z}}{\psi} - e^{2\rho\bar{\sigma}}\left[\frac{\psi_{,z}}{\psi} + \frac{\rho\bar{\sigma}_{,z}}{2}\right]\right) d\rho \\ &\quad -\frac{1}{2}\int_{z_{\min}} \rho\psi^4 \left(-\frac{\psi_{,z}}{\psi} - e^{2\rho\bar{\sigma}}\left[\frac{\psi_{,z}}{\psi} + \frac{\rho\bar{\sigma}_{,z}}{2}\right]\right) d\rho \\ &\quad +\frac{1}{2}\int_{\rho_{\max}} \rho\psi^4 \left(-\frac{\psi_{,\rho}}{\psi} - e^{2\rho\bar{\sigma}}\left[\frac{\psi_{,\rho}}{\psi} + \frac{(\rho\bar{\sigma})_{,\rho}}{2} + \frac{1}{4\rho}\right] + \frac{1}{4\rho}\right) dz. \end{aligned} \tag{2.67}$$

Figure 2.3 below shows these quantities for six Brill wave [10] evolutions with identical parameters, except for differing grid resolution and outer boundary location. Figure 2.4 shows similar plots for a scalar field evolution.

We have used second-order accurate finite difference stencils in our code, and hence, if the PDEs have been converted to FDEs correctly, we expect second-order convergence of the results. Specifically, for M_{ADM}, we expect second-order convergence to a constant prior to the time when energy reaches the outer boundary (the causal speed of propagation is ≈ 1), we expect Q_ψ to be ≈ 4, and similarly $E(\dot{K_\rho}^\rho)$ should decrease by a factor of roughly 4 when the grid resolution is doubled. We do see these trends in the data, with a couple of caveats. First, as discussed in some detail in Section 2.2.6, our outer boundary conditions for the constrained variables are not fully consistent with the evolution equations. This causes $E(\dot{K_\rho}^\rho)$ (and similarly for other components of $E(\dot{K_A}^B)$) to converge to some non-zero value as $\Delta\rho \to 0$, for a given $\rho_{\max}$; however, we do see a trend to consistent results in the limit as $\rho_{\max} \to \infty$. Second, the no-incoming-radiation boundary conditions used for evolved quantities are not perfect, and part of the wave-like components of these functions, upon reaching the outer boundary, are reflected back towards the interior. This leads to poor convergence results after most of the energy has left the computational domain.

2.3 Structure of the Adaptive Code

This section describes a version of the code incorporating an adaptive mesh refinement (AMR) driver. The algorithm used is a variant of Berger and Oliger's algorithm (see Section 1.5.5), and the code was started from a version of the 1D driver used in [4]. A modification of the basic algorithm is needed to deal with the elliptic equations that we solve for in our evolution scheme. This, and other minor changes, are described in Section 2.3.1. Section 2.3.2 details extensions to the multigrid (MG) algorithm required to solve elliptic equations on the grid hierarchies that are produced during AMR evolution. The clustering algorithm is described in Section 2.3.3, where we focus on how we satisfy the restrictions imposed on grid-sizes and positions so that the resultant hierarchy can be used directly by the MG routine. Section 2.3.4 describes how we calculate the truncation error estimate (TRE), and Section 2.3.5 shows the interpolation and restriction operators we use. The method that we use to initialize the grid hierarchy prior to evolution is discussed in Section 2.3.6.

[9] The ADM mass represents the total energy within the spacetime in the limit where the surface of integration goes to infinity, so the farther the outer boundary is from the region of strong-field gravity, the more accurate the expression for the ADM mass should become (modulo the usual convergence criteria).

[10] A Brill wave is an even parity gravitational wave in axisymmetry.

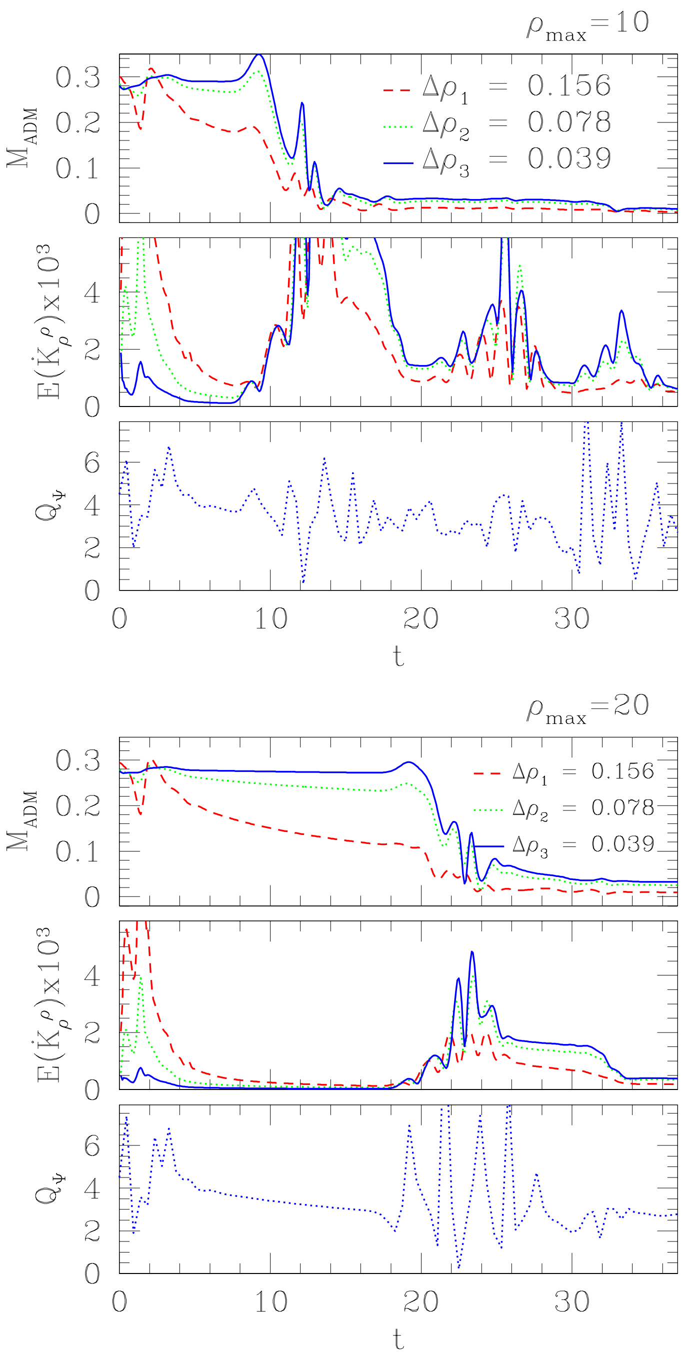
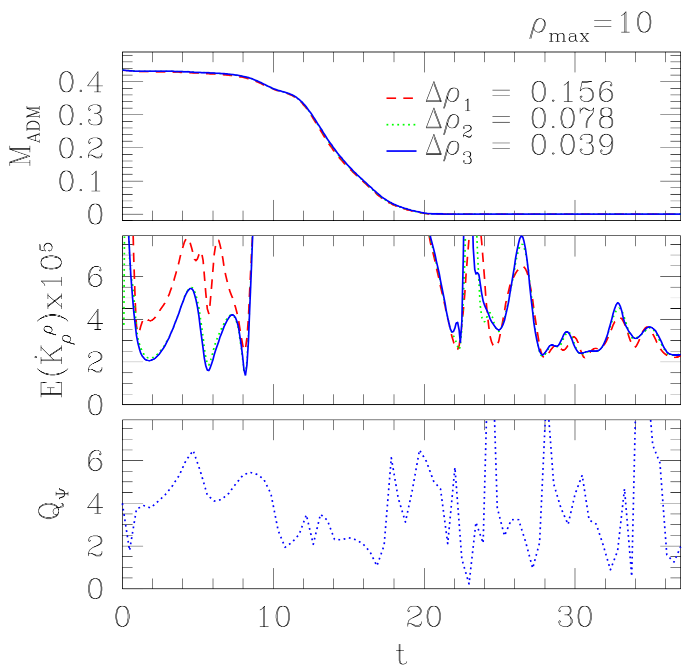


Figure 2.3: Test of the unigrid code for Brill wave evolution. Shown are the ADM mass M_{ADM} and $E(\dot{K}_\rho{}^\rho)$—the the ℓ_2 norm of the residual of the evolution equation for $K_\rho{}^\rho$ (2.66)—from simulations run with three different grid resolutions, and two different outer boundary locations (the figure to the left has $\rho_{\max} = z_{\max} = -z_{\min} = 10$, and the figure to the right has $\rho_{\max} = z_{\max} = -z_{\min} = 20$). Also plotted is the convergence factor Q_ψ for ψ (1.55), computed using the data from the three runs for each of the two $\rho_{\max}$ cases. The initial data for this example is: $\bar{\sigma}/\rho$ is initialized with the function (2.52), with $A = -3.0$, $R_0 = 0$, $\Delta = 1$, $\epsilon = 1$, and $(\rho_0, z_0) = (0, 0)$; $\bar{\Omega}$, Φ and Π_Φ are set to zero.



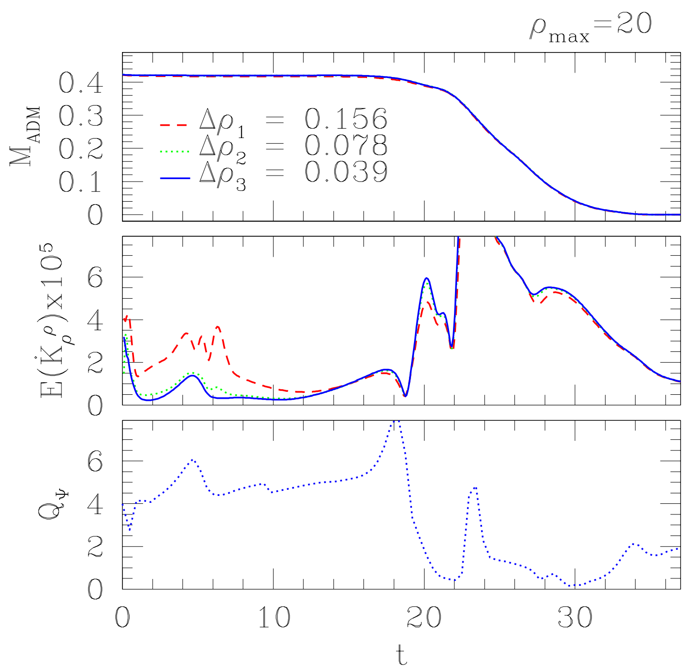


Figure 2.4: Test of the unigrid code for scalar field evolution. The same data is shown here as for the Brill wave example in Figure 2.3 above (though notice that the scale of the residual is about 2 orders of magnitude smaller than that in the Brill wave case). The initial data for this example is: Φ is initialized with the function (2.52), with $A = 0.15$, $R_0 = 0$, $\Delta = 3$, $\epsilon = 3$, and $(\rho_0, z_0) = (0, 0)$; $\bar{\Omega}, \Pi_\Phi$ and $\bar{\sigma}$ are set to zero.

In AMR evolution, unwanted high-frequency solution components of the FDEs ("noise") are often excited at grid boundaries. In Section 2.3.7 we describe the interpolation and smoothing operators we apply to certain variables at grid boundaries to control the noise. There are also situations where the MG solver fails to solve the constraint and slicing equations; in Section 2.3.8 we mention these problems and how (for the most part) we are able to overcome them. Finally, in Section 2.3.9 we present some convergence test results for the adaptive code.

2.3.1 Modifications to Berger and Oliger's Algorithm

The driver for the adaptive code follows Berger and Oliger's (B&O) algorithm quite closely. The most significant differences are that child grids are not rotated, we incorporate a *self-shadow hierarchy* to facilitate truncation error estimation, and the elliptic variables are handled via a combination of extrapolation and "delayed solution".

A Self-shadow Hierarchy for Computing TRE

A self-shadow hierarchy is a simplification of the idea of using a *shadow hierarchy* [108] to do truncation error estimation. A shadow hierarchy is a coarsened (usually with $\rho_s = 2:1$) version of the main hierarchy. Both hierarchies are evolved simultaneously, and the function values of a given grid in the shadow hierarchy are replaced with those of the corresponding grid in the main hierarchy whenever the two are in sync. For example, with $\rho_t = 2:1$, each time step of a shadow grid corresponds to two time steps of the main grid, and the shadow is updated every two main-grid time steps. A TRE can therefore readily be computed by comparing function values in the shadow with corresponding values in the main hierarchy just before the update step.

Notice however, that due to the recursive time-stepping procedure of the Berger and Oliger algorithm, information for computing a TRE is "naturally" available prior to the fine-to-coarse grid injection step (see Figure 2.5). The coarse level ℓ is evolved independently of the fine level $\ell+1$ from t_0 to $t_0 + \Delta t_c$, where Δt_c is the coarse level time step. Also, at $t = t_0$ the level ℓ grid functions are restricted copies of level $\ell+1$ grid functions in the region of overlap $O_\ell^{\ell+1}$. Therefore, prior to injection at time $t_0 + \Delta t_c$, the difference in an evolved variable f in levels ℓ and $\ell+1$, within the region $O_\ell^{\ell+1}$, can serve as an approximation to the truncation error $\tau_s(f_{\ell+1})$ for f at level $\ell+1$:[11]

$$\tau_s(f_{\ell+1}) \equiv f_{\ell+1} - f_\ell. \tag{2.68}$$

Therefore, for levels $\ell > 1$, we can use (2.68) as the basis for computing truncation error estimates, without the need to refer to a shadow hierarchy (i.e., the main hierarchy "casts its own shadow", hence the name self-shadow hierarchy). This method cannot give a TRE for the coarsest level (1) in the hierarchy, and so we require that the coarsest level always be fully refined. Thus, the resolution of level 2 should be chosen to match the desired coarsest resolution for a given problem.

Incorporating MG into the AMR Algorithm

The B&O algorithm is tailored to the solution of hyperbolic PDEs. A time step is taken on a coarse level ℓ *before* steps are taken on finer levels, so that the level ℓ solution can be used to set boundary conditions on level $\ell+1$. The boundary conditions used on level $\ell+1$ thus come from a solution of the finite difference equations on level ℓ, and if a sufficient buffer zone about the region of high truncation error is used during regridding, then the solution obtained on level ℓ in the vicinity of the boundary of level $\ell+1$ will differ by an amount less than $\tau_{\max}$ from a putative FD solution obtained there at the resolution of level $\ell+1$.

[11]In fact, if there are only two levels in the hierarchy, then this estimate is exactly the truncation error estimate defined in (1.49). If levels finer than $\ell+1$ exist, then in the overlap $O_{\ell+1}^{\ell+2}$ the estimate (2.68) will be modified by an amount of order $(\Delta t_c/\rho_s)^2$.

The preceding statement is only true for hyperbolic equations, where the finite speed of propagation will prevent contamination of the solution in the boundary region of level ℓ from a poorly resolved solution in the interior, where $\tau_\ell > \tau_{\max}$. Solutions to elliptic equations do not share this property, and therefore it is not feasible to solve for such equations on the coarse grid alone, with the intention of supplying boundary conditions for subsequent fine grid time steps. One way to circumvent this problem is to abandon the B&O recursive time-stepping procedure. In other words, we could evolve the entire hierarchy forwards in time with a global time step, by performing a Crank-Nicholson style iteration as in the unigrid code. The elliptic equations would then be solved over the entire hierarchy using the scheme presented in Section 2.3.2. A major drawback to this method is that, to satisfy the CFL condition, the global time step will need to be set to $\lambda\Delta\rho(\ell_f)$, where $\Delta\rho(\ell_f)$ is the cell size on finest level ℓ_f in the hierarchy.

An alternative option to incorporating elliptic equations into the standard B&O time-stepping framework, that we have decided to use, is to employ a combination of extrapolation and delayed solution of the elliptic variables (see Figure 2.5). Simply stated, on coarse levels we do *not* solve the elliptic equations during the evolution step of the algorithm; rather, we extrapolate the corresponding variables to the advanced time from the solution obtained at earlier times. The solution of the elliptic equations is delayed until *after* the fine to coarse grid injection step from level $\ell+1$ to ℓ. At that stage, all levels from ℓ to ℓ_f are in sync, and we solve the elliptic variables over this subset of the hierarchy, thus including all of the fully resolved details from finer levels interior to level ℓ in the solution. Extrapolated boundary conditions are used at AMR boundaries on level ℓ.

The current extrapolation scheme used for the elliptic variables is as follows. We use linear (2nd order) extrapolation in time, with periodic "corrections" to try to account for changes that occur upon global MG re-solves. For level ℓ, the two past-times used in the extrapolation are the two most recent times when levels ℓ and $\ell-1$ were in sync (thus, at times when a solution of the MG variables involving at least levels $\ell-1$ to ℓ_f was obtained). In other words, every ρ_t steps we save the MG variables for use in extrapolation. The correction is calculated as follows. Whenever level ℓ is in sync with level ℓ_c, where $\ell_c < \ell$, a MG solve takes place over levels $\ell_c..\ell_f$. Denote by $\hat{f}_\ell(t)$ the value of a variable f at level ℓ, calculated via extrapolation from $f(t-\rho_t\Delta t_\ell)_\ell$ and $f(t-2\rho_t\Delta t_\ell)_\ell$, and let $f_\ell(t)$ denote the value of the same variable *after* the MG re-solve at time t. The correction is defined to be

$$f_{c\ell}(t) \equiv \frac{f_\ell(t)-\hat{f}_\ell(t)}{\rho_t^{\ell-\ell_c}}, \tag{2.69}$$

and is used to change $f(t-\rho_t\Delta t_\ell)_\ell$ as follows (see Figure 2.6):

$$f_\ell(t-\rho_t\Delta t_\ell) \longrightarrow f_\ell(t) - \left(\hat{f}_\ell(t) - f_\ell(t-\rho_t\Delta t_\ell)\right) - f_{c\ell}(t) \tag{2.70}$$

The logic behind this form of correction stems from an observation that in general $\hat{f}_\ell(t) - f_\ell(t)$ is (in a loose sense) proportional to $\ell_c - \ell$; i.e. the more levels over which the elliptic equations are re-solved, the larger the change in the interior, finer level ℓ, whose boundary conditions have essentially been evolved via extrapolation in the interim between re-solves over the larger domain of level ℓ_c. Thus, if in hindsight, the quantity (2.69) had been added to f at each one of the $\rho_t^{\ell-\ell_c}$ intermediate time steps between re-solves over levels $\ell_c..\ell_f$, then (2.69) would be zero at time t (ignoring, of course, the effect that this putative correction would have had on the solution).

This extrapolation technique is rather ad-hoc, though both the choice of which past times to extrapolate from, and the use of corrections, play a significant role in the stability of the adaptive evolution code. It would be interesting in the future to investigate higher order extrapolation schemes (earlier experiments with fourth order extrapolation, involving the four most recent time steps, was unstable—possibly synchronizing the times used for extrapolation with coarser level times would help here too, as it did with linear extrapolation).

Lastly, with regards to the extrapolation, note that on the finest level we *do* solve the elliptic equations within the Crank-Nicholson iteration, as in the unigrid case. Here the extrapolation

```
for (requested number of coarse steps)
   call single_step(level 1)
   stop
end

subroutine single_step(level L)
   if regridding_time(L) then
      regrid from levels L to Lf
   end if

   for evolved variables: if (L>1) then set boundary conditions along
      AMR boundaries at time t(L)+delta_t(L) via interpolation from
      level (L-1)

   for MG variables: extrapolate the entire grid function to time
      t(L)+delta_t(L) from earlier-time solutions at level L

   repeat until (residual norm < tolerance):
      1: if (L=Lf) then perform 1 MG vcycle on elliptic variables
         at time t(L)+delta_t(L)
      2: perform 1 iteration of the Crank-Nicholson sweep for all evolved
         variables
   end repeat

   t(L)=t(L)+delta_t(L)

   if (L<Lf) then
      1: repeat [rhot=delta_t(L)/delta_t(L+1)] times: call single_step(L+1)
      2: compute the TRE for level (L+1) by comparing the solutions
         in the region of overlap between levels L and L+1
      3: inject the solution from level L+1 to level L
         in the region of overlap between levels L and L+1
   end if

   if ( L=1 or (L<Lf and t(L)!=t(L-1)) ) then
      1: re-solve the elliptic equations over the sub-hierarchy from
         levels L to Lf
      2: compute the correction to the MG-variable extrapolation functions
   end if

   return from subroutine
end of subroutine single_step
```

Figure 2.5: A pseudo-code representation of the Berger and Oliger time stepping algorithm, as modified and implemented in our code. Note however that the actual code is written in Fortran 77, hence the recursion is manually implemented using stacks.

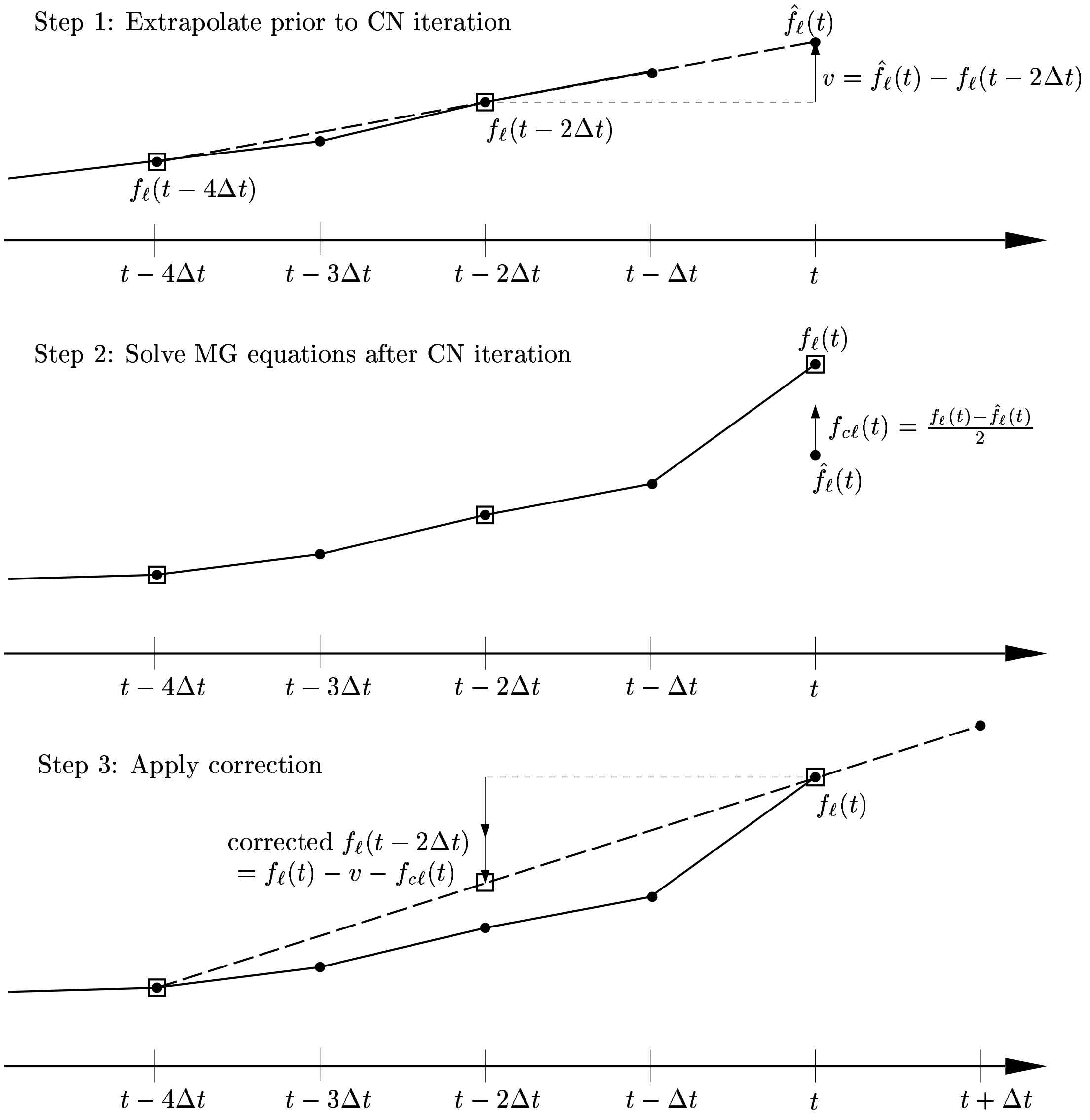

Step 1: Extrapolate prior to CN iteration
$\hat{f}_\ell(t)$
$v = \hat{f}_\ell(t) - f_\ell(t-2\Delta t)$
$f_\ell(t-2\Delta t)$
$f_\ell(t-4\Delta t)$
$t-4\Delta t$
$t-3\Delta t$
$t-2\Delta t$
$t-\Delta t$
t
Step 2: Solve MG equations after CN iteration
$f_\ell(t)$
$f_{c\ell}(t) = \frac{f_\ell(t)-\hat{f}_\ell(t)}{2}$
$\hat{f}_\ell(t)$
Step 3: Apply correction
corrected $f_\ell(t-2\Delta t)$
$= f_\ell(t) - v - f_{c\ell}(t)$
$f_\ell(t)$
$t+\Delta t$

Figure 2.6: An illustration of the technique we use to extrapolate and solve for elliptic variables within the AMR framework. In this example, we show the evolution of an elliptic variable f_ℓ from $t-\Delta t$ to t, assuming that $\rho_t = 2$ and $\ell < \ell_f$. Initially (step 1), f_ℓ at t is calculated via linear extrapolation from data at past times $t-2\Delta t$ and $t-4\Delta t$ (quantities used for extrapolation are depicted by boxes in the diagram). This value, labeled $\hat{f}_\ell(t)$, is used during the Crank-Nicholson (CN) iteration of the evolved variables. At time t, we assume levels ℓ and $\ell_c = \ell - 1$ are in sync, and so after the CN iteration (*and* after the equations on finer levels $\ell+1..\ell_f$ have been evolved to t) we re-solve the elliptic equations over levels $\ell_c..\ell_f$ (step 2). This results in a change in the value of f from $\hat{f}_\ell(t)$ to $f_\ell(t)$ (for simplicity we assume that a similar change that occurred at time $t-2\Delta t$ is zero). This change is propagated back to $t-2\Delta t$, so that the same velocity v, *modulo a correction* $f_{c\ell}(t)$, will be used to extrapolate f to $t+\Delta t$ (step 3). Here, since $\ell - \ell_c = 1$, the correction is such that we are effectively extrapolating from $t-4\Delta t$ and t to time $t+\Delta t$; this would not be the case otherwise—see (2.69-2.70). (If $\ell = \ell_f$, then the elliptic variables are solved within the CN iteration, and the value obtained afterwards is used for $\hat{f}_\ell(t)$.)

simply serves to set the boundary conditions.

2.3.2 Multigrid on an Adaptive Hierarchy

The extension of the basic MG algorithm (see Section 1.5.6) to a grid hierarchy is rather straightforward in concept. Figure 1.3 shows a typical AMR hierarchy over which we want to solve a set of elliptic equations using MG. To simplify the process, we require that ρ_s be an integer power of two; then the AMR hierarchy can easily be extended to incorporate the MG levels, which have a refinement ratio ρ_{mg} of 2:1. When building the MG hierarchy, each AMR grid is individually coarsened by factors ρ_{mg} until either a) one dimension of the grid is smaller that the minimum allowed, or b) the coarsened grid can be absorbed into a larger grid at that level in the MG hierarchy.

The V-cycle proceeds in essentially the same manner as before—the pseudo code shown in Figure 1.5 can be read verbatim, the only "modifications" being the manner in which several of the statements in the listing are applied. First, inter-level operations, such as restriction, computing coarse-grid corrections, etc., are only performed in the region of overlap between the two levels (which is *always* the region of the fine level, given the kind of hierarchies that are produced by B&O AMR). Second, the manner in which the relaxation sweep proceeds over a level is modified, to account for possible grid overlap and AMR boundaries, as follows. During the relaxation sweep, the variables at a given coordinate location and level are relaxed only once, regardless of the number of grids encompassing that location. This is necessary to preserve the smoothing properties of the relaxation scheme. We use a mask function to enforce this single update per grid point. The mask is initialized to zero on all grids in the level prior to the relaxation sweep. Then, a sweep is applied, in turn, to each grid in the level, though only variables at points where the mask is equal to zero are modified. After the sweep is done on a given grid, the mask is set to one throughout that grid, and the mask and other grid functions are copied to overlapping grids at the same level. Therefore, subsequent relaxation sweeps on adjacent grids skip over points that have already been relaxed. This communication step, in addition to enforcing a single update per point, ensures that grid functions are numerically unique at all physical grid locations[12], which is important for preserving the V-cycle's convergence properties. Also, though we use Dirichlet boundary conditions at AMR boundaries, the communication ensures that points interior to a union of grids are always treated as interior by the end of the V-cycle, even if they lie on the boundary of some grid in the overlap region.

With regards to the performance of this MG scheme on a general adaptive hierarchy, there are two situations of relevance where performance could suffer, compared to the single grid MG algorithm. The first occurs when, at some level down (coarser) in the MG hierarchy, one or more grids in a connected union of grids is a "coarsest grid", and hence needs to be solved "exactly" — see Figure 2.7. Experimentation showed that the entire union needed to be solved exactly in that situation; i.e. it was not sufficient to solve the equations exactly on the coarsest grids, then proceed down the V-cycle on the remaining grids. If the union of grids consists of a relatively small number of grid points, then such a situation will not be a problem; otherwise, there will be a significant slow-down of the algorithm, for the speed of relaxation suffers dramatically as the number of unknowns increase. For the physical scenarios discussed in this thesis, we found that single, isolated clusters (see Section2.3.3) were efficient in covering the regions of high truncation error, and so avoided this problem.

The second situation where performance suffers is when the TRE requires long, skinny rectangular regions to be refined. This does occur with the more prolate initial data configurations that we have studied. What happens in this case, is that such an elongated grid can not be coarsened very much along the larger grid dimension before the smaller dimension has reached the smallest

[12]Note that we also perform such a communication step after relaxation of evolved variables, during the Crank-Nicholson iteration.

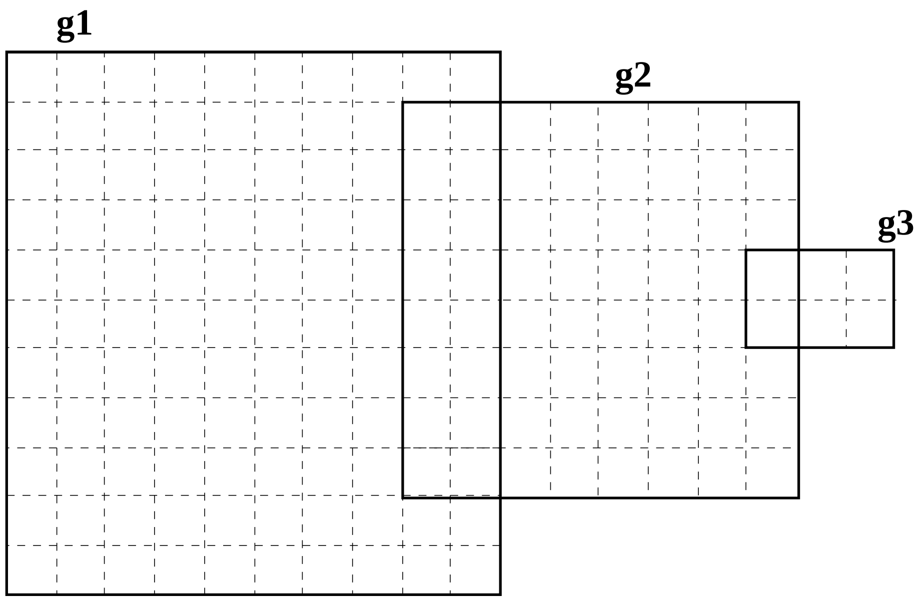


Figure 2.7: A hypothetical MG grid configuration that will adversely affect execution speed. In the figure we depict three grids, **g1,g2** and **g3**, several levels down in the grid hierarchy (i.e., after several coarsening steps have already been performed). Grid **g3** cannot be coarsened any further, while grids **g1** and **g2** can, and ideally should be coarsened further, to maintain the speed of the algorithm. However, because **g3** overlaps the other two grids, this entire level must be considered a coarsest level, and solved "exactly".

allowed size. Again, this results in a relatively large number of grid points where the solution needs to be obtained exactly. As of yet we have not addressed this problem, accepting the slowdown that occurs. A possible solution may be to use line relaxation; however, because we simultaneously solve for several variables per grid point, the resulting linear system would be represented by a banded matrix, the solution of which may not be much faster than the current point-wise relaxation.

2.3.3 Clustering

We have incorporated two clustering routines into the code. The first, written by Reid Guenther, Mijan Huq and Dale Choi [109], is based upon the signature-line method of Berger and Rigoutsos [73]. The second is a simple routine that produces single, isolated clusters—each isolated region of high TRE is surrounded by a single cluster, and then all clusters within a certain distance of each other are merged together into an encompassing cluster. For the specific problems that we have studied so far, the isolated cluster method turns out to be almost as efficient as the signature-line method. Therefore, since efficiency is not an issue, the isolated cluster method is preferable, because it avoids one of the potential speed-bottlenecks of the MG algorithm discussed in Section 2.3.2; furthermore, as we mention in Section 2.3.7, minimizing cluster overlap helps reduce high-frequency noise problems.

A clustering issue that needs to be dealt with in our code is that the resultant grid hierarchy must be acceptable by the MG solver. This places two restrictions on cluster sizes and positions. First, an individual grid must have dimensions that can be factored into $x_{\min}2^n$, where $x_{\min}$ is one of the smallest, allowed grid dimensions, and n is a non-negative integer. Second, if several grids overlap, then their relative positions must be such that the common grid points align on all possible levels of a MG hierarchy. Specifically, if a union of overlapping grids can collectively be coarsened m times in the MG hierarchy, then the relative offsets of grid origins on the finest level must be multiples of 2^m grid points. We have decided to enforce these requirements after the initial clustering algorithm is applied, by modifying the returned cluster list accordingly. This gives us the flexibility to experiment with different clustering routines, whether or not they produces clusters compliant with the MG solver.

We have also decided to implement several additional options in the post-clustering routine. Two of these are adding "ghost zones" between adjacent clusters, so that both the MG and evolution relaxation sweeps correctly solve the system of equations in a domain given by the union of grids at a given level; and optionally moving or shrinking clusters, if necessary, to prevent them from touching parent boundaries [13], which helps to avoid instabilities that occasionally occur in such situations.

2.3.4 Truncation Error Estimation

The truncation error estimate is computed as described in Sections 1.5.5 and 2.3.1. Here we list the specific calculation and set of variables used. Depending upon the problem, one or more of ψ, Φ, Π_Φ, $\bar{\Omega}$, and $\bar{\sigma}$ are used in the calculation (i.e., all evolved quantities—ψ is not used when the Hamiltonian constraint is used to solved for ψ). Optionally, the truncation error estimate is scaled by the norm of the function if the norm is larger than some constant k ($k = 1$ currently); we also multiply the corresponding TRE by a power of ρ if necessary, chosen heuristically to counter either the asymptotic or near-axis behavior of the particular function. The TRE for a function f_ℓ at level ℓ is thus defined to be

$$\tau_s(f_\ell) = \frac{(f_\ell - f_{\ell-1})\,\rho^p}{\max(k, ||f_\ell||_2)}, \tag{2.71}$$

[13]In principle this should never occur if one adds a buffer zone about the region of high truncation error. However, because of the grid shuffling performed to obtain a hierarchy acceptable to MG, a grid could be extended to touch a parent boundary. With the option enabled to prevent this, the grid will be reduced rather than extended to fit into the MG scheme. This comes at the expense of not obtaining "optimal" zones about the region of high truncation error.

where it is implied that f_ℓ is only defined in the overlap between levels ℓ and $\ell-1$, f_ℓ is restricted to the resolution of level $\ell-1$ prior to subtraction, and the result is then interpolated back to the resolution of level ℓ. Typically, we choose $p=2$ for $\bar{\Omega}$, $p=1$ for $\bar{\sigma}$, and $p=0$ for the other variables. The TRE for the level is defined to be

$$\tau_s(\ell) = \sqrt{\sum \tau_s(f_\ell)^2}, \tag{2.72}$$

where the sum is taken over the desired subset of variables listed above.

Optionally, the TRE calculated in (2.72) is further smoothed (using simple averaging over a 5-by-5 square cell of points), and/or scaled by a quantity $H_\ell \geq 1$ in the region of overlap between levels ℓ and $\ell+1$. H_ℓ therefore provides a kind of "hysteresis" to the truncation error estimation process: when the TRE in a region of level ℓ grows above $\tau_{\max}$, that region is refined; however, for the region to be unrefined at a later time, the TRE needs to drop below $\tau_{\max}/H_\ell$ there. In most of the simulations we keep $H_\ell = 1$, though occasionally it has proven useful to set it to around $5-10$. In one situation, this prevented apparent erroneous unrefinement in a region where it appeared as if the lapse function α may have been the dominant source of solution error, yet for a short period of time this fact was not reflected in the TRE computed from the evolved variables.

2.3.5 Interpolation and Restriction Operators

Here we state the restriction and interpolation operators used in the AMR code. Straight-injection is used to restrict a fine grid to a coarse grid during the injection phase of the AMR algorithm, and when computing the TRE. A fourth order interpolation scheme is used to initialize newly refined fine grids (or regions thereof) from the encompassing coarser grid. The scheme proceeds by first interpolating every row of the coarse grid to the fine grid (i.e. every ρ_sth row of the fine grid is filled in), then all the remaining points on the fine grid are computed via interpolation, column-by-column. The adaptive MG routine uses the same set of operators as the unigrid MG routine, namely half-weight restriction when transferring from a fine to coarse grid, and linear interpolation for the opposite.

2.3.6 Initializing the Grid Hierarchy

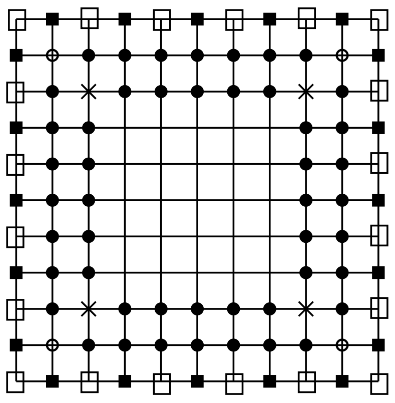
Here we list the steps currently used to initialize the grid hierarchy.

1) The first two levels (the "coarsest" desired level 2, and its shadow level 1) are initialized with the freely specified data; then the constraints and slicing condition are solved on these two levels.

2) One step of the AMR evolution algorithm is taken, thus evolving the hierarchy to $t = \Delta t_1$. For the first couple of time steps per level, only first order extrapolation of the MG variables is performed, as earlier time information is not yet available.

3) A regridding step is performed, based upon the truncation error estimates calculated during step 2). t is then reset to 0, the new hierarchy is reinitialized with the free data, and the constraints and slicing condition are re-solved. Steps 2) and 3) are repeated until the number of levels in the hierarchy stops increasing after the regridding step, or the maximum allowed number of levels is reached.

4) Now that the initial grid hierarchy has been determined, we initialize the grid functions for elliptic variables at retarded time levels $t_\ell = -\rho_t \Delta t_\ell$, required for extrapolation (see Section 2.3.1), as follows. We evolve the hierarchy one coarse step Δt_1 forwards in time, then one coarse step backwards in time, returning to $t=0$. After this, the grid functions at times $t = \rho_t \Delta t_\ell$ (stored as "past" times because of the backwards evolution) and $t=0$ are used to linearly extrapolate to $t = -\rho_t \Delta t_\ell$. (Alternatively one could first evolve backwards then forwards in time by one coarse step, however the results would essentially be the same due to the fact that we are using linear extrapolation).



Interpolation Method

□ (1) linear in time from parent grids

■ (2) 4th order in space using points from (1)

● (3) 4th order in space using a point from (1) or (2) and 3 interior points normal to the boundary

× (4) as (3), though utilizing points in (3) instead of interior points

○ (5) bilinear in space using points from (1) and (4)

Figure 2.8: An illustration of the interpolation method used for $\bar{\sigma}$ and $\bar{\Omega}$ during AMR evolution. In the figure we assume that the spatial refinement ratio is 2 : 1, and that all four grid boundaries are AMR boundaries. Points labeled by (1) and (2) are set once prior to the Crank-Nicholson (CN) iteration, while points labeled by (3), (4) and (5) are reset after every CN step (see the pseudo-code in Figure 2.9). Points not explicitly labeled are "interior" points, and are evolved.

2.3.7 Controlling High-Frequency Grid-Boundary Noise

An issue that needs to be dealt with in a Berger & Oliger style AMR scheme is controlling high-frequency noise that may occur at parent-child grid boundaries. For a second order accurate finite-difference scheme, the second derivatives of grid functions are typically not continuous across the boundaries after child to parent injection. This potential source of high-frequency noise on the parent level is rather efficiently eliminated by the Kreiss-Oliger dissipation filters that we use during evolution.

In certain situations we find that high-frequency noise also develops on child grids, within a grid point or two of the AMR boundary (in particular near the corners of the grid, or places where two grids overlap). This noise is not so easily dealt with, as the Kreiss-Oliger filter acting normal to the boundary is only applied a distance three points and further away from boundary. The source of this noise appears to be the parent-child interpolation scheme used to set the boundary values, and in general the interpolation method must be tailored to each variable in order to reduce the noise to an acceptable level. We use the following interpolation method. For all evolved variables ($\bar{\sigma}, \bar{\Omega}, \Phi, \Pi_\Phi$ and ψ) we use linear interpolation in time from the parent level to set boundary values on the child grid at points coincident with parent grid points. This is followed by fourth-order interpolation in space (as described in Section 2.3.5) for the remaining boundary points. Furthermore (see Figures 2.8 and 2.9), after each step of the Crank-Nicholson iteration we reset $\bar{\Omega}$ and $\bar{\sigma}$ in a zone two grid points in from AMR boundaries with values obtained either 1) by fourth order interpolation using function values from the boundary and three additional points inward from this zone, or 2) via bilinear interpolation at "corner" points, i.e. those points that are a single cell width away from two boundaries. This technique for $\bar{\sigma}$ and $\bar{\Omega}$ was discovered after quite a bit of experimentation with different interpolation schemes, and is quite effective in reducing the level of noise at the grid boundaries.

```
subroutine interp_interior_AMR_bnd(grid function f[1..Nrho,1..Nz])
   for i=3 to Nrho-2 do
      f(i,2)=0.4*f(i,1)+2.0*f(i,4)-2.0*f(i,5)+0.6*f(i,6)
      f(i,3)=0.1*f(i,1)+2.0*f(i,4)-1.5*f(i,5)+0.4*f(i,6)
      f(i,Nz-1)=0.4*f(i,Nz)+2.0*f(i,Nz-3)-2.0*f(i,Nz-4)+0.6*f(i,Nz-5)
      f(i,Nz-2)=0.1*f(i,Nz)+2.0*f(i,Nz-3)-1.5*f(i,Nz-4)+0.4*f(i,Nz-5)
   end do

   for j=3 to Nz-2 do
      f(2,j)=0.4*f(1,j)+2.0*f(4,j)-2.0*f(5,j)+0.6*f(6,j)
      f(3,j)=0.1*f(1,j)+2.0*f(4,j)-1.5*f(5,j)+0.4*f(6,j)
      f(Nrho-1,j)=0.4*f(Nrho,j)+2.0*f(Nrho-3,j)-2.0*f(Nrho-4,j)+0.6*f(Nrho-5,j)
      f(Nrho-2,j)=0.1*f(Nrho,j)+2.0*f(Nrho-3,j)-1.5*f(Nrho-4,j)+0.4*f(Nrho-5,j)
   end do

   f(2,2)=(f(1,1)+f(3,3)+f(1,3)+f(3,1))/4
   f(Nrho-1,2)=(f(Nrho,1)+f(Nrho,3)+f(Nrho-2,1)+f(Nrho-2,3))/4
   f(2,Nz-1)=(f(1,Nz)+f(3,Nz)+f(1,Nz-2)+f(3,Nz-2))/4
   f(Nrho-1,Nz-1)=(f(Nrho,Nz)+f(Nrho,Nz-2)+f(Nrho-2,Nz)+f(Nrho-2,Nz-2))/4

   return from subroutine
end of subroutine interp_interior_AMR_bnd
```

Figure 2.9: A pseudo-code description of part of the interpolation method used for $\bar{\sigma}$ and $\bar{\Omega}$ during AMR evolution. For simplicity, in this subroutine we assume that all four boundaries are interior to the computational domain boundaries. The set of points altered here correspond to (3), (4) and (5) in Figure 2.8, and the interpolation operators used are independent of the spatial refinement ratio (as opposed to points (1) and (2) in Figure 2.8).

For the MG variables (α, β^ρ and β^z), prior to a Crank-Nicholson evolution cycle we reset these variables on AMR boundaries at points unique to the grid (in between points coincident with parent level points—those labeled by (2) in Figure 2.8) using fourth order interpolation from the remaining points on the boundary. Thus we overwrite the values set by coarse-grid corrections (CGCs) during the most recent MG solve that involved coarser levels. The reason we do this is as follows. In MG, CGCs typically introduce high-frequency noise on the finer level, while the subsequent post-CGC relaxation sweeps smooth out this noise. However, since no relaxation is applied on AMR boundaries, some form of explicit smoothing is required — the above described fourth order interpolation provides this smoothing mechanism.

Another stage in the algorithm where high-frequency noise can creep into the solution is during the regridding phase, if the refined region on a given level expands. Then, within the part of a new grid overlapping the old refined region, grid functions will be initialized by copying data from an old fine grid, while on the rest of the new grid the grid functions will be initialized via interpolation from a parent grid. Sometimes, at the interface between the copied/interpolated data, tiny discontinuities are introduced. Fortunately, the grid functions are easily smoothed by applying a Kreiss-Oliger filter to them (at all time levels) after regridding.

2.3.8 Dealing with Multigrid Failures

Occasionally during an evolution, predominantly of strong-field gravitational wave spacetimes, the MG routine "fails" in some fashion. A failure is either an instability, where a grid function diverges at some point, or MG fails to converge to within the desired tolerance after a specified number of time steps. Often, these failures can be cured, by first restoring the grid functions to the values they had prior to the failure, and then adjusting some parameter within the algorithm before attempting to re-solve the elliptic equations. Here, we briefly mention some of the specific problems, and adjustments that relieve them. However, even with these fixes, the MG routine is still not robust enough to study critical collapse of gravitational waves.

The residual is often large and near-discontinuous in the vicinity of a child-grid boundary on the parent level. This rarely causes problems, though when it does, simply increasing the number of pre-CGC and post-CGC relaxation sweeps fixes the problem.

Occasionally, a low-frequency oscillation develops in the residual, where, even though the magnitude decreases somewhat, the overall sign of the residual flips from one V-cycle to the next. This adversely affects the rate of convergence, and in extreme cases can cause the residual to diverge. In most cases, an effective cure is to weight the restricted residual by a factor $w < 1$ when proceeding down the V-cycle, and then weight the coarse grid correction by a factor of $1/w$ when proceeding up the V-cycle[14]. We find that values of w between 0.9 and 0.95 work well.

Sometimes the relaxation becomes unstable (usually when one of the source functions, such as $\bar{\Omega}$, becomes very large in magnitude). If this occurs on a coarser level in the MG hierarchy (coarser than the coarsest level in the AMR hierarchy), we simply truncate the V-cycle prior to the offending level. This can result in a significant slow-down of the code if the truncation needs to be maintained for many time steps. An alternative that sometimes works for this problem, without slowing down the code, is to smooth the source functions on the coarser levels. We have not fully explored this second alternative yet, for in principle, the source functions can be drastically altered on coarser levels, without affecting the desired solution of the problem on the fine level. This is because the truncation error added to the right-hand-sides of the differential operator on coarser levels will compensate for such changes (1.82). One might at first think that altering source functions would only be transferring any irregularities in them to the right-hand-sides. However, the source functions *are* smooth on the finest levels, and they only become irregular after possibly dozens of coarsening operations during a V-cycle. Early positive results obtained when smoothing source functions suggest that "manipulating" them further, in some fashion, may be a promising

[14]This parameter first appeared in Hackbusch's multigrid method [110].

	$\alpha_{\mathbf{4h}} - \alpha_{\mathbf{h}}$	$\alpha_{\mathbf{2h}} - \alpha_{\mathbf{h}}$	$\alpha_{\mathbf{AMR}} - \alpha_{\mathbf{h}}$	$\mathbf{\Phi_{4h} - \Phi_h}$	$\mathbf{\Phi_{2h} - \Phi_h}$	$\mathbf{\Phi_{AMR} - \Phi_h}$
$\|\|\cdot\|\|_{\ell_2}$	2.24e-4	3.20e-5	3.55e-5	5.83e-5	1.06e-5	9.83e-6
$\|\|\cdot\|\|_{\ell_\infty}$	8.34e-3	1.63e-3	3.34e-4	3.13e-3	1.24e-4	2.04e-4

Table 2.1: AMR code test: ℓ_2 and ℓ_∞ norms of the data shown in Figures 2.11 and 2.12.

route to achieving robust relaxation on coarser levels.

2.3.9 Convergence, Consistency and Mass Conservation Results

In this section we present two tests of the AMR code, similar to those presented in Section 2.2.7 for the unigrid code. In the first example, we compare the results of a strong-field AMR evolution to that of a similar unigrid evolution. In the second example, we do a convergence test of an AMR evolution, by recording the grid hierarchy at the lowest resolution, then using the exact same hierarchy for runs at double and quadruple the resolution.

Comparison with a Unigrid Evolution

First, we compare an evolution obtained with the AMR code to a unigrid evolution, where the entire unigrid mesh is given the resolution h of the finest level in the AMR hierarchy. To gauge how well the AMR solution approximates the unigrid one, we also compare the unigrid solution to that obtained with two additional unigrid runs, with resolutions of $2h$ and $4h$. In principle, we could force the AMR solution to agree with the h-unigrid solution by setting the maximum allowed TRE to 0; however, that defeats the purpose of AMR.

For this example, we evolve a spherically symmetric, time symmetric scalar field spacetime, in the strong-field regime (the lapse α attains a minimum of around 0.05 during evolution; in the second comparison below, we look at less symmetric, though weaker initial data, and also demonstrate what happens when the energy disperses). We initialize Φ using (2.52), with $A = 0.23$, $R_0 = 0$, $\Delta = 1.0$, $\epsilon = 1$ and $(\rho_0, z_0) = (0,0)$; and set Π_Φ, $\bar{\Omega}$ and $\bar{\sigma}$ to zero initially. The outer boundary is at $\rho_{\max} = z_{\max} = -z_{\min} = 10$. The initial ADM mass of the spacetime is around 0.37. We set the maximum value for the TRE at 10^{-4} (using only Φ and Π_Φ in the calculation). The resolution of the base grid (level 2) is set to 65×129, and we use $\rho_s = 2$. The evolution of the AMR code was stopped at $t = 2.34$, by which time the hierarchy was 5 levels deep—see Figure 2.10 below for a plot of α then. Therefore, for the unigrid comparison runs, we used resolutions of 513×1025 (h), 257×513 ($2h$) and 129×257 ($4h$).

Figure 2.11 contains plots of the differences in α, between the $4h$ and h, the $2h$ and h, and the AMR and h solutions at $t = 2.34$. Figure 2.12 is a comparison plot of Φ (similar results are obtained when examining other variables in the problem, and for brevity we do not show them all). The essential information on these plots is summarized in Table 2.1. One can see that the AMR evolution, in general, gives comparable or better performance than the $2h$ unigrid solution, compared to the h solution.

AMR Convergence Test

To examine the bevahior of the AMR code in a more dynamical situation than the previous example, and at resolutions unattainable by a unigrid code, we do the following convergence test of an AMR evolution. We run a simulation with a given set of parameters (including a modest level for the maximum allowed TRE), and during the run, after each regridding phase, we record the grid hierarchy produced. The resultant data, which we label with h, becomes the lowest resolution data for the subsequent convergence test. For the higher resolution runs—$h/2$ and $h/4$—we re-run the code using the previously recorded grid hierarchy, except that each grid in the hierarchy is

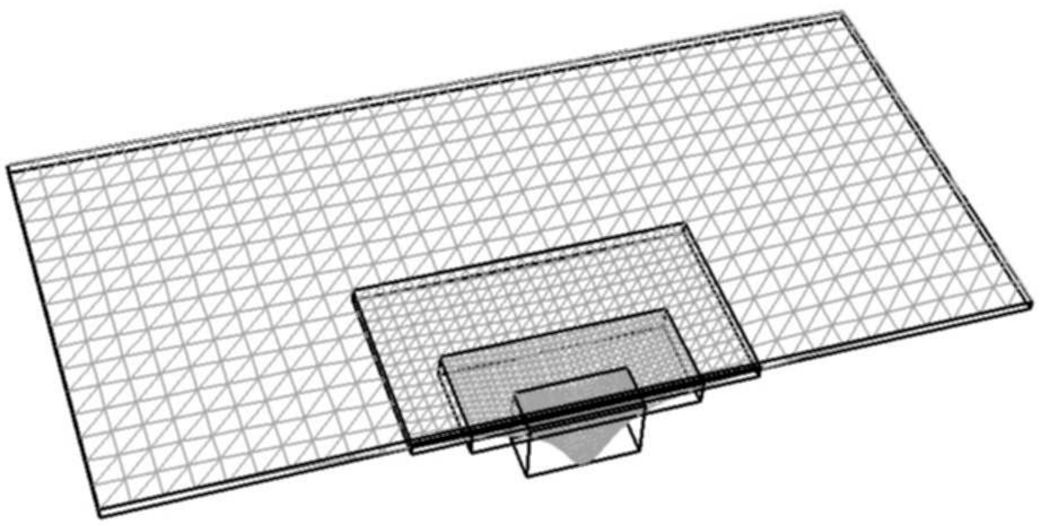


Figure 2.10: AMR code test: α at $t = 2.34$, showing the AMR hierarchy and mesh structure. The data in the figure has been coarsened by a ratio of 4:1, to give a clearer view of the meshes. The height of the surface represents the magnitude of α, and the origin $\rho = z = 0$ is coincident with the minimum in α (of ≈ 0.14).

given double the resolution for the $h/2$ run, and quadruple the resolution for the $h/4$ run. The higher resolution runs will not have an "ideal" grid structure as computed by local truncation error estimates, however, as the grid hierarchies are identical for the three runs, it becomes feasible to do a direct, point-wise comparison of the results.

In this example, we consider non-trivial initial data for all of the four, freely specifiable variables: $\Phi, \Pi_\Phi, \bar{\Omega}$ and $\bar{\sigma}$. We initialize Φ using (2.52), with $A = 0.025$, $R_0 = 3$, $\Delta = 1.0$, $\epsilon = 1$ and $(\rho_0, z_0) = (0, 0)$. Π_Φ is initialized as described in Section 2.2.3 to give approximately ingoing waves at $t = 0$. $\bar{\sigma}/\rho$ is also initialized using (2.52), with $A = -0.01$, $R_0 = 3$, $\Delta = 1.0$, $\epsilon = 1$ and $(\rho_0, z_0) = (0, 0.3)$; as is $\bar{\Omega}/\rho$, with $A = 0.01$, $R_0 = 3$, $\Delta = 1.0$, $\epsilon = 1$ and $(\rho_0, z_0) = (0, -0.3)$. The outer boundary is set to $\rho_{\max} = z_{\max} = -z_{\min} = 48$, and the base level (2) has a resolution of 49×97 grid points. The Courant factor λ is set to 0.15, and we only present results of the evolution up to $t = 24$ (as we wish to focus upon AMR-specific aspects of the code—the AMR code uses the same boundary conditions as the unigrid code, and we would see the same rather poor behavior after waves reach the outer boundary at $t \approx 48$). We use free evolution for ψ. The maximum TRE is set to 10^{-3}, which results in a set of levels versus time as shown in Figure 2.13 below. The finest level (6) of the $h/4$ run has an effective unigrid resolution of 3073×6145, making it impractical to do a comparison as in the first example above. The smallest features in the solution develop at around $t = 3 - 6$, in the vicinity of the origin, and consequently the AMR hierarchy is deepest then. After $t \approx 10$, essentially all of the radiative fields have left the vicinity of the origin, and are traveling towards the outer boundary.

Figure 2.14 shows the total ADM mass of the spacetime (we integrate (2.67) on the surface given by $\rho_{\max} = z_{\max} = -z_{\min} = 43$ for this simulation). Figure 2.15 contains plots of $E(\dot{K}_\rho{}^\rho)$ and $E(\dot{K}_\rho{}^z)$—the ℓ_2 norms of the residual of the evolution equations for $K_\rho{}^\rho$ and $K_\rho{}^z$ respectively(2.66). Figure 2.16 shows plots of the convergence factor (1.55) for three representative variables: $\bar{\Omega}$, ψ and α (to demonstrate the effectiveness of our delayed-solution scheme for elliptic variables). When

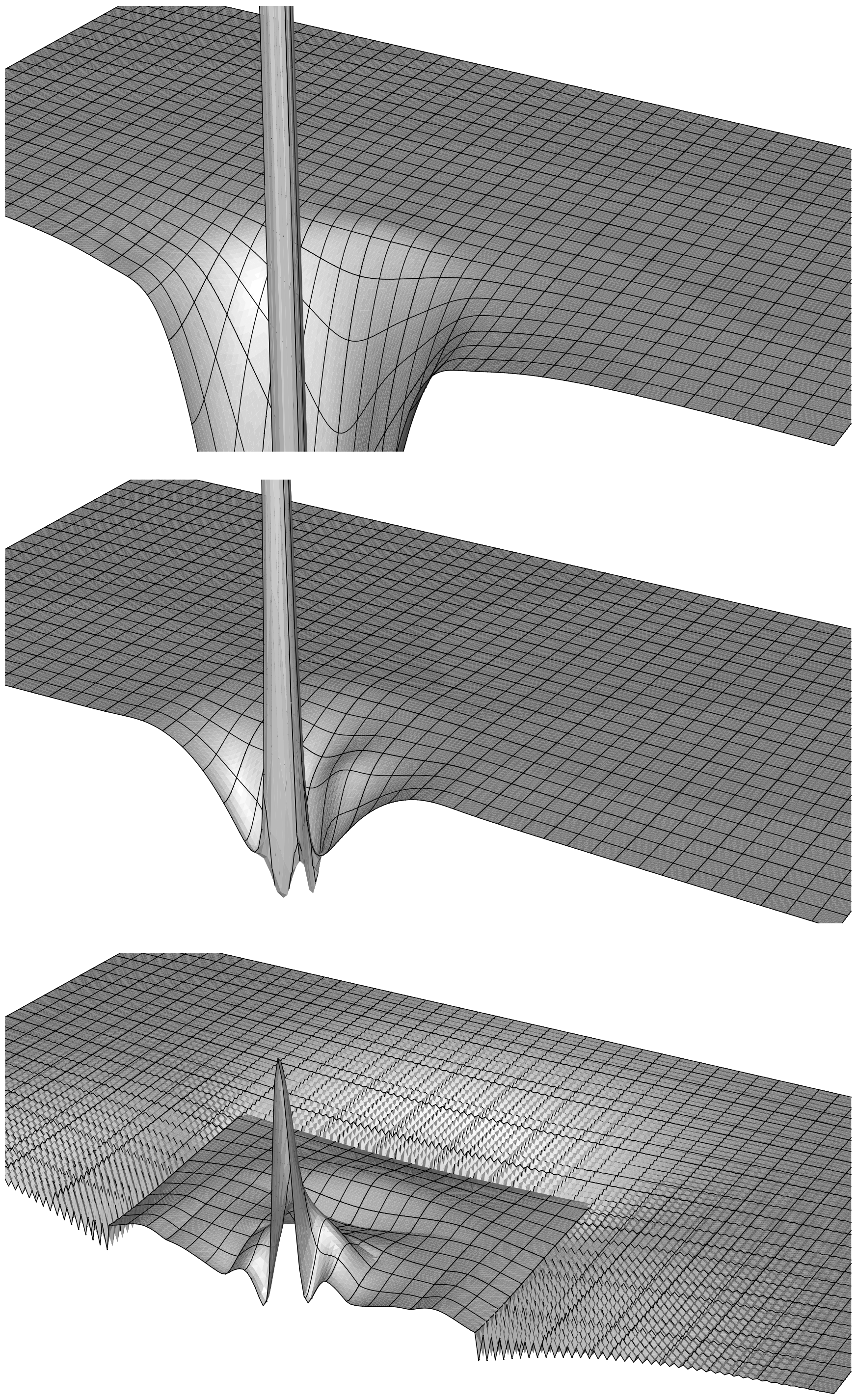


Figure 2.11: AMR code test: comparison of α to a unigrid evolution, at $t = 2.34$. The top figure shows the difference between the h and $4h$ unigrid solutions, the middle figure shows the difference between the h and $2h$ solution, and the bottom figure is the difference between the h unigrid and the AMR solution. The height of each surface represents magnitude, and the same vertical scale is used for each frame. The mesh-lines shown are for visual aid only, and do not reflect any computational mesh structure. The largest differences are in the vicinity of the origin, $\rho = z = 0$.

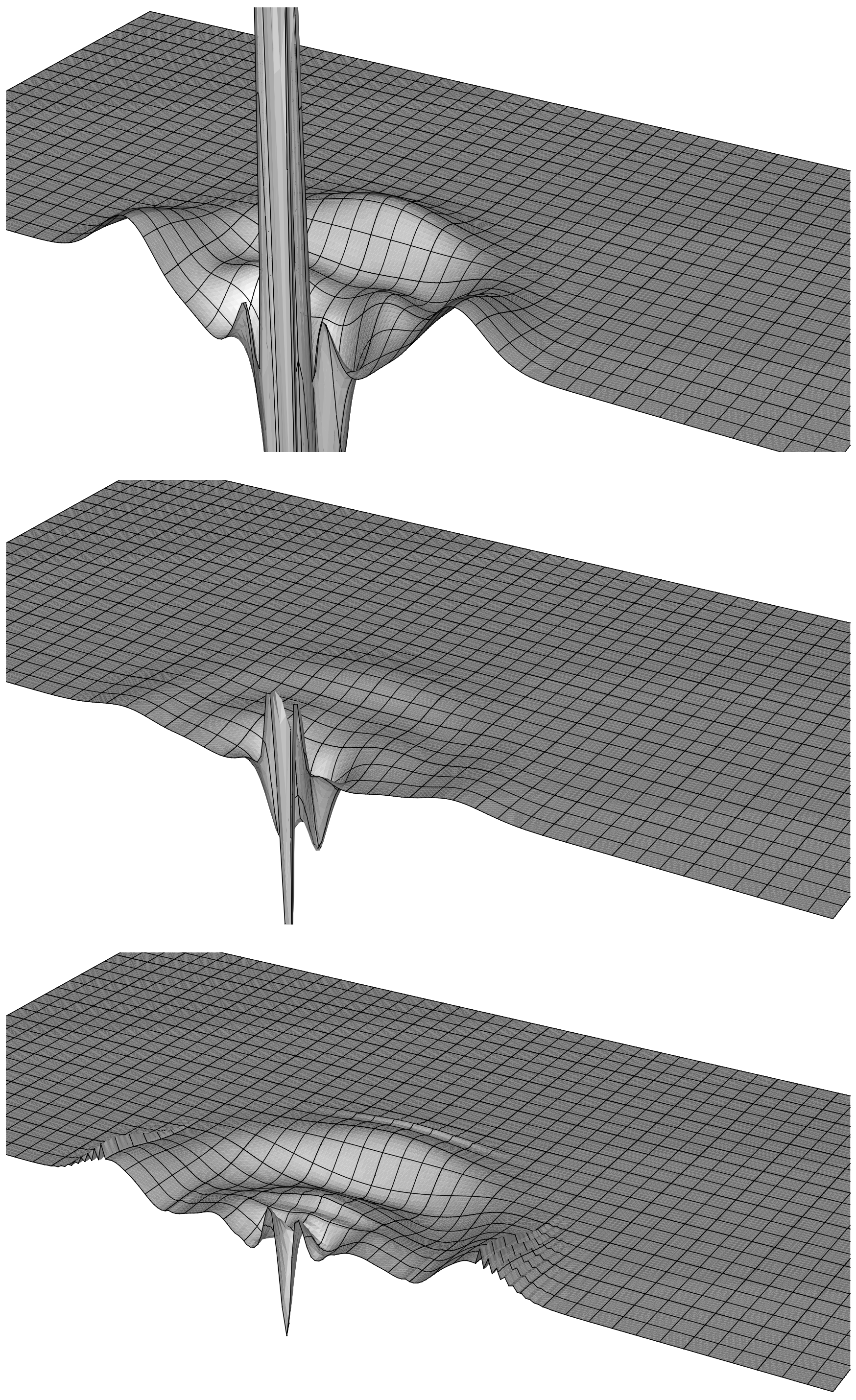


Figure 2.12: AMR code test: comparison of Φ to a unigrid evolution, at $t = 2.34$. The top figure shows the difference between the h and $4h$ unigrid solutions, the middle figure shows the difference between the h and $2h$ solution, and the bottom figure is the difference between the h unigrid and the AMR solution. The height of each surface represents magnitude, and the same vertical scale is used for each frame. The mesh-lines shown are for visual aid only, and do not reflect any computational mesh structure. The largest differences are in the vicinity of the origin, $\rho = z = 0$.

calculating the ℓ_2 norms for $Q_{\bar{\Omega}}$, $E(\dot{K}_\rho{}^\rho)$ and $E(\dot{K}_\rho{}^z)$, we restricted the domain of the calculation to $[0, 15, -15, 15]$, for the corresponding variables are very close to zero outside of this region during the time of the simulation. ψ and α are more global in nature, and hence we included the entire domain in their convergence test calculations.

As demonstrated in these figures, at early times, while the dynamics are unfolding within the vicinity of the origin, we get very good convergence results. If fact, for the convergence factors Q we obtain numbers higher that the expected value of 4. The convergence factors for $E(\dot{K}_\rho{}^\rho)$ and $E(\dot{K}_\rho{}^z)$ going from runs $h/2$ to $h/4$ are closer to 2 then; the reason for this (compared to the relatively high reduction in error going from h to $h/2$) is apparently due to the inconsistency of the outer boundary conditions (see the discussion in Sections 2.2.6 and 2.2.7), for, as the resolution is increased, the residual norm starts to be dominated by a lower-frequency component that encompasses most of the computational domain.

At later times, after most of the energy has left the origin, the convergence figures look quite bad. The reason appears to be that then, in the vicinity of the origin, the solution for the radiative variables (such as $\bar{\Omega}$ and Φ) is dominated by a low magnitude ($\approx 10^{-4}$ to 10^{-5}) residual "noise". The noise is predominantly sourced at parent-child boundaries, earlier in the evolution. Dissipation eliminates the highest frequency components of the noise, though some lower-frequency components remain, primarily in $\bar{\Omega}$. There are two factors that appear to be most responsible for the noise produced in $\bar{\Omega}$ at parent-child boundaries. First, the noise decreases as λ decreases; this may simply be because the linear interpolation used to set the boundary conditions becomes more accurate as λ decreases, though it may also have something to do with the fact that decreasing λ improves the extrapolation of the elliptic variables. In particular, α and $\bar{\Omega}$ seem to be rather strongly coupled, in that even slight irregularities in the second ρ derivative of α near outer parent-child boundaries excites noise in $\bar{\Omega}$ there. This noise is greatly suppressed by the particular form of smoothing we apply to $\bar{\Omega}$, as discussed in Section 2.3.7 (though, of course, there may still be better smoothing methods that could be applied). Second, the larger the gradient in $\bar{\Omega}$ across a parent-child boundary, the more likely it is that noise will be generated. This source of noise could be alleviated somewhat by tuning the truncation error estimation scheme (for instance by weighting $\bar{\Omega}$ more heavily when calculating the TRE), so that the resultant grids more completely enclose the regions where the gradients of $\bar{\Omega}$ are larger.

2.4 Probing the Geometry of a Numerical Solution

Due to the coordinate freedom one has in general relativity, it is often not an easy task to extract physically relevant information about the geometry of a particular solution. This difficulty is somewhat compounded in a numerical solution, where the 4D geometry is obtained as a sequence of 3D spatial slices, and it is a cumbersome task to analyze this data relative to a time coordinate other than the one used in the simulation. The simplest quantities to analyze are therefore invariant quantities that do not depend upon a particular slicing. For example, in scalar field critical collapse, we calculate the maximum value attained by the Ricci scalar during sub-critical evolution, or the local maxima and minima that the scalar field attains within the oscillation of each self-similar echo. During gravitational wave collapse, where the Ricci scalar is zero, the extreme values of other invariants could be monitored, for instance the square of the Riemann tensor:

$$R_{abdc}R^{abcd}. \tag{2.73}$$

Of course, observing a curvature invariant on some "time" $t = \text{const.}$ slice, and its evolution with respect to t, is a valid probe of the geometry of the solution, and potentially just as useful as that of any other time slicing. However, when we want to compare information between different solutions (if it makes sense to do so), we need to fix some aspect of the slice on which we do the comparison. In some situations, for instance when measuring waveforms produced in black hole

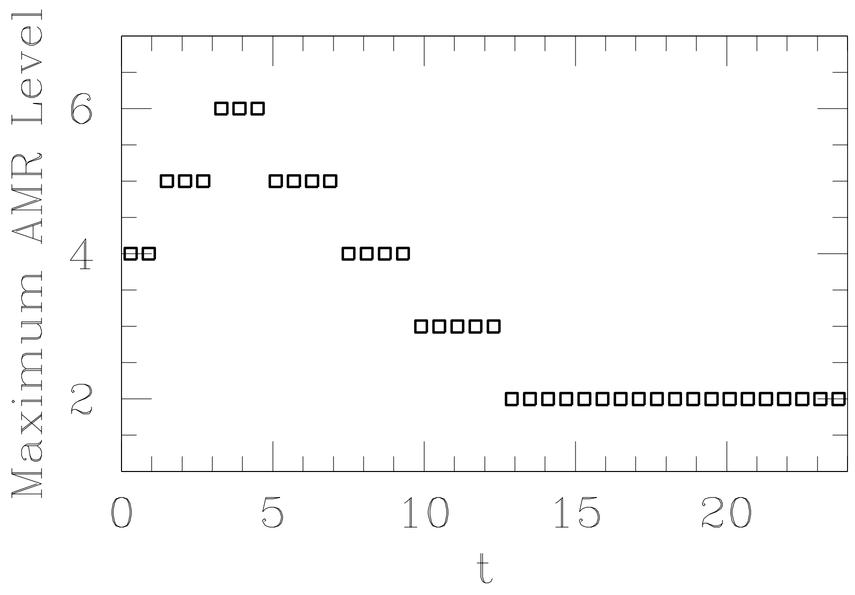


Figure 2.13: AMR code test: depth of the hierarchy as a function of time.

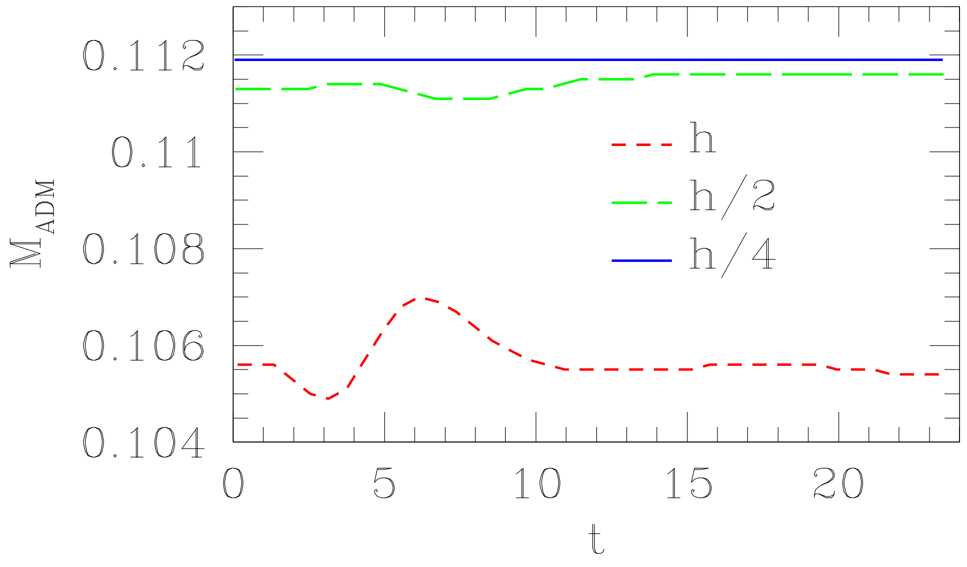


Figure 2.14: AMR code test: M_{ADM}.

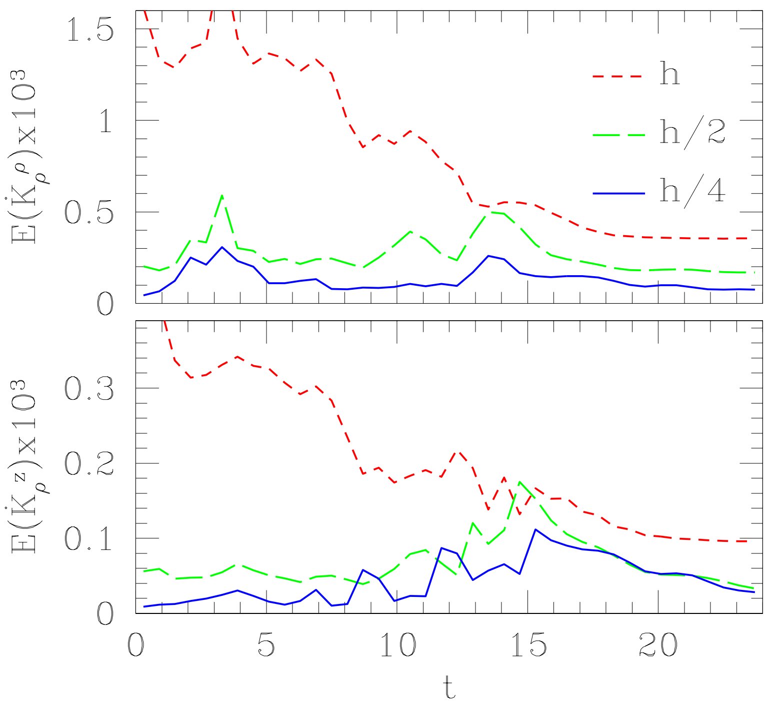


Figure 2.15: AMR code test: consistency.

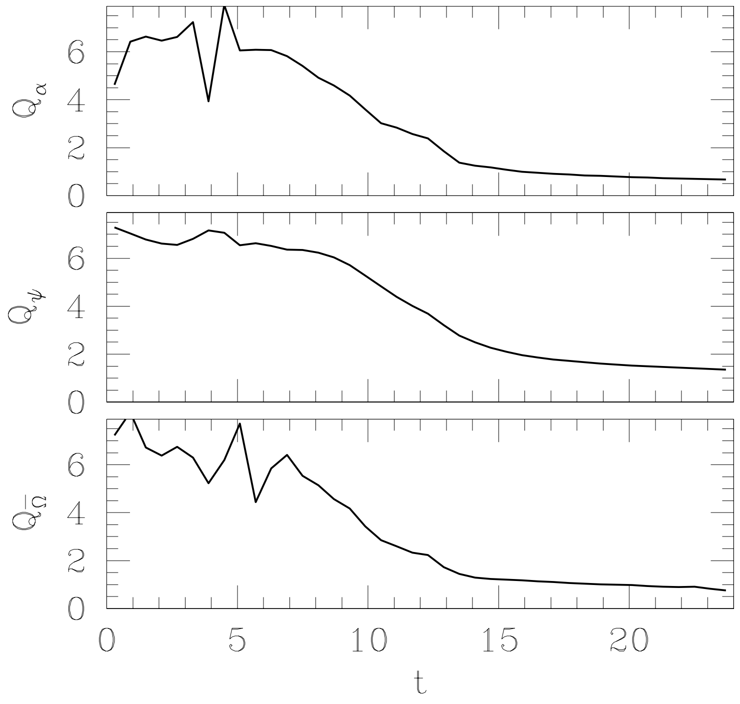


Figure 2.16: AMR code test: convergence factors.

collisions, it is sufficient to compare the solutions near the outer boundary of the computational domain, where one is approaching an unambiguous, asymptotically flat coordinate system. In other situations, we want to compare quantities in the strong field region of the solution. Specifically, in the study of scalar field critical collapse in Chapter 3, we would like to measure how close a given near-critical solution is to the putative universal spherically symmetric solution. As mentioned above, observing the extreme values of Φ and R at the center of self-similarity provides some information; though this will not tell us how rapidly the asymmetric modes decay, or perhaps if there are some less "local" features of the solution that do not match the spherically symmetric one. Therefore, we want to compare entire solutions on a "common" slice. One way to generate such a slice is to use null geodesics in the following manner. We wait for the evolution to proceed to some landmark spacetime event within the solution; for instance one of the local extrema attained by the scalar field during its self-similar echoing behavior. Then, from that point, we send out a family of null geodesics in all directions (i.e., a lightcone), parameterized by an affine parameter λ. The geodesics will then probe a particular slice of the spacetime, and using the angle θ to label the initial direction of each geodesic (the azimuthal angle ϕ is redundant in an axisymmetric solution), we can unambiguously compare invariants of the solutions at common (θ, λ) coordinates along the null slice.

In the following section we describe how we integrate null (and timelike) geodesics in the code.

2.4.1 Geodesics

We use the Lagrangian formulation of geodesic motion to arrive at the set of equations we integrate in the code [111]. We restrict attention to geodesics with zero "orbital" angular momentum, thus $d\phi/d\lambda \equiv \dot{\phi} = 0$, where λ is the affine parameter along the geodesic, and we define the over-dot to mean differentiation with respect to λ. Hence, in our metric, the Lagrangian takes the form

$$\mathcal{L} = g_{tt}\dot{t}^2 + 2g_{\rho t}\dot{\rho}\dot{t} + 2g_{zt}\dot{z}\dot{t} + \psi^4\left(\dot{\rho}^2 + \dot{z}^2\right), \tag{2.74}$$

where, to keep the expressions simple, we have not expanded the $g_{tt}, g_{\rho t}$ or g_{zt} components of the metric in terms of $\alpha, \beta^\rho, \beta^z$ and ψ. The Euler-Lagrange equations of motion, with λ an affine parameter, are

$$\frac{d}{d\lambda}\left(\frac{\partial\mathcal{L}}{\partial\dot{x}}\right) - \frac{\partial\mathcal{L}}{\partial x} = 0, \tag{2.75}$$

where $x = x(\lambda)$ is one of $\rho(\lambda), z(\lambda)$ or $t(\lambda)$. Note that when computing the functional derivatives in (2.75), one considers x and $\dot{x}$ to be independent variables, yet functions of λ.

We are only going to integrate time-like or null geodesics, for which $\mathcal{L}$ is normalized to the constants -1 and 0 respectively. One can use the normalization condition in lieu of one of the equations of motion, and we will do so, replacing the t equation from (2.75). We convert the remaining ρ and z equations in (2.75) to first order form by defining the following conjugate variables

$$\Pi_\rho = \frac{\partial\mathcal{L}}{\partial\dot{\rho}}, \quad \text{and} \tag{2.76}$$

$$\Pi_z = \frac{\partial\mathcal{L}}{\partial\dot{z}}. \tag{2.77}$$

The reason why we have chosen this particular set of conjugate variables, and eliminated the t equation of motion in favor of the normalization condition, is that consequently no t derivatives of metric functions appear in the resultant set of first order equations. This is a useful property to have for integration within the AMR code, because, as a result of the method we use to extrapolate the elliptic variables in the Berger and Oliger algorithm, single time-step finite difference approximations to the time derivatives of these variables have a fair amount of high frequency noise (in time).

Finally, to facilitate integration within the code, we change independent variables from λ to t. Representing the derivative with respect to t with a prime, the set of equations we integrate are:

$$\begin{aligned}
\lambda' &= \frac{2\alpha\psi^2}{\sqrt{\Pi_\rho^2 + \Pi_z^2 - 4\psi^4\mathcal{L}}}, && (2.78)\\
\rho' &= \frac{\Pi_\rho \lambda'}{2\psi^4} - \beta^\rho, && (2.79)\\
z' &= \frac{\Pi_z \lambda'}{2\psi^4} - \beta^z, && (2.80)\\
\Pi_\rho' &= \frac{1}{\lambda'}\left[g_{tt,\rho} + 2g_{\rho t,\rho}\rho' + 2g_{zt,\rho}z' + 4\psi^4\psi_{,\rho}\left({\rho'}^2 + {z'}^2\right)\right] && (2.81)
\end{aligned}$$

and

$$\Pi_z' = \frac{1}{\lambda'}\left[g_{tt,z} + 2g_{\rho t,z}\rho' + 2g_{zt,z}z' + 4\psi^4\psi_{,z}\left({\rho'}^2 + {z'}^2\right)\right] \tag{2.82}$$

CHAPTER 3

SCALAR FIELD CRITICAL COLLAPSE

In this chapter we explore critical gravitational collapse of the massless scalar field in axisymmetry. The results were obtained using our numerical code that incorporates adaptive mesh refinement.

Based upon studies of critical collapse in spherical symmetry, Choptuik conjectured that critical behavior is universal, and should be observed away from spherical symmetry [4]. This is consistent with the idea that the critical solution is a one-mode unstable solution (see the discussion in Section 1.3), and perturbative calculations support this claim [26, 27]. To date, however, the only non-perturbative study of critical phenomena away from spherical symmetry was carried out by Abrahams and Evans [96], who simulated the collapse of pure gravitational waves in axisymmetry. They found evidence for a discretely self-similar critical solution, with a scaling exponent $\gamma \approx 0.37$, and echoing exponent $\Delta \approx 0.6$. Surprisingly, to within numerical error, the scaling exponent is the same as that of the self-gravitating scalar field, however the echoing exponents differ significantly, with $\Delta \approx 3.44$ for the scalar field. The corresponding critical solutions are markedly different though—the scalar field critical solution is spherically symmetric, while the gravitational wave critical solution is intrinsically asymmetric, as gravitational waves do not exist in spherical symmetry (which is why it is rather surprising that the scaling exponents are nearly the same).

The main questions that we would like to answer in this chapter are: does one observe critical phenomena at the threshold of black hole formation in the collapse of asymmetric distributions of scalar field energy, and if so, what is the nature of the critical solution? To this end, in the remaining sections of this chapter, we present results from the study of threshold collapse of several distinct initial configurations of the scalar field. First, in Section 3.1, we show results from the collapse of spherically symmetric initial data. The results obtained are consistent with those observed in previous critical collapse studies in spherical symmetry (for a comprehensive list of these see [35, 36]) Furthermore, due to the rectangular computational domain, our 2D code can never exactly compute spherically symmetric solutions; hence the fact that we *can* reproduce the 1D results to a reasonable degree of accuracy is already some confirmation that the critical solution persists with very small asymmetric perturbations. In Section 3.2 we show the results from slightly prolate and oblate families of initial data. Again, as close to criticality as we can tune these two families, we find consistency with the spherically symmetric solution In Section 3.3 we look at initial data deviating significantly from being spherical symmetry—namely a solution that is antisymmetric about $z = 0$, and several highly prolate families of initial data. For the antisymmetric family (discussed in 3.3.1), we find that *two* self-similar solutions develop at threshold, separated by some distance along the axis. Each of the two solutions locally resemble the spherically symmetric solution. However, our examination of the collapse from highly prolate distributions in Section 3.3.2 suggests that there may be an additional unstable mode in axisymmetry. As mentioned in the introduction, at late times, this mode (if not merely an artifact of the initial data) causes the critical solution to bifurcate into two new, discretely self similar solutions.

Unless otherwise stated, we do not show the grid structure in any of the figures of grid functions—grid-lines drawn on the plots are for visual aid only. Also, in all of the surface plots, the height of the surface represents the magnitude of the function.

All of the simulations were run on single nodes of either UBC's **vn** cluster, consisting of 850Mhz PIII processors with 512MB of memory, or on nodes of the **MACI Alpha Cluster** at the University of Calgary, which contain DEC Alpha processors ranging in speed from 500-833Mhz, and having up to 4GB of memory. The most demanding simulations we ran utilized up to 1GB of memory, and took a few days to complete. We estimate that we have used roughly 150,000 CPU hours over the past year to obtain the results presented here (a fair portion of which was used during the

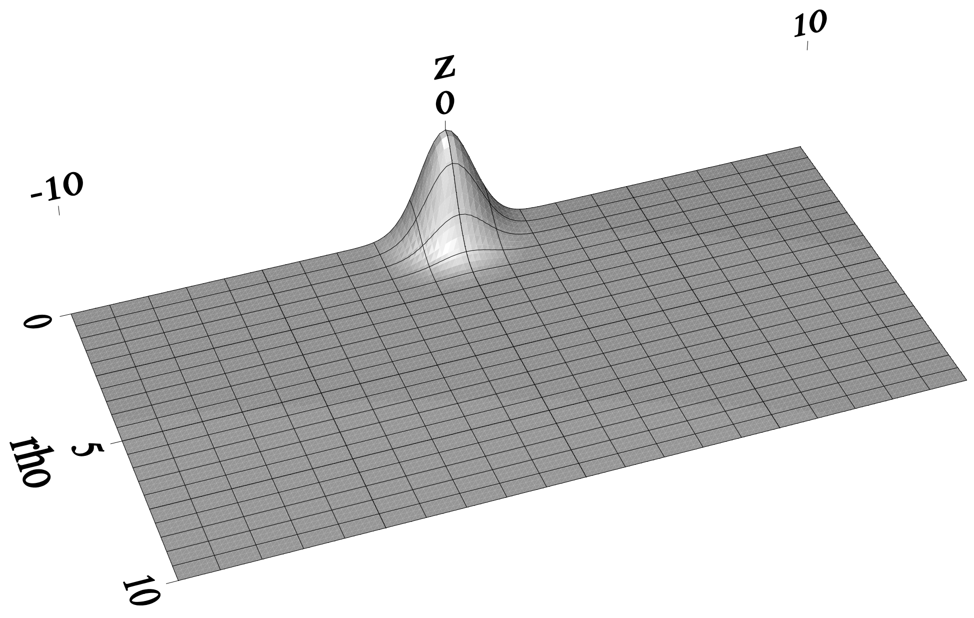


Figure 3.1: Initial profile of Φ, for spherically symmetric critical collapse.

developmental stages of the code).

3.1 Spherically Symmetric Initial Data

In this section we present results from the simulation of critical collapse of spherically-symmetric, time-symmetric Gaussian pulses of the scalar field Φ. Specifically, we initialize Φ using (2.52) with $\epsilon = 1$, $\Delta = 1$ and $R_0 = 0$, and then vary the amplitude A to find the critical amplitude $A^\star$. Figure 3.1 is a plot of one member of this family of initial data. The remaining freely specifiable variables, Π_Φ, $\bar{\sigma}$ and $\bar{\Omega}$, are set to zero at the initial time. The outer boundary is at $\rho_{\rm max} = z_{\rm max} = -z_{\rm min} = 10$. For α, β^ρ and β^z, we use Dirichlet conditions at the outer boundary, while for ψ we use the $1/r$ fall-off condition (A.12). The base grid (level 2) within the AMR hierarchy has resolution 65×129, and we use a spatial refinement ratio of 2:1 (hence the shadow grid at level 1 has resolution 32×64). Up to 22 levels of refinement were used for evolutions closest to criticality (for an effective unigrid resolution of $\sim 6,000,000 \times 12,000,000$). We use a temporal refinement ratio of $2:1$. A few of the other important parameters used for the evolution are: the Courant factor $\lambda = 0.3$, the regridding interval is 8, the minimum buffer width is 4, and we use Φ and Π_Φ to calculate truncation error estimates. We have performed two sets of runs, one with the maximum truncation error, which we call ϵ_τ here, at 10^{-3}, the other with $\epsilon_\tau = 10^{-4}$.

To measure the scaling exponent γ and echoing parameter Δ, we look at the maximum absolute value attained by the Ricci scalar on axis—$\max_{z,t}|R(0,z,t)|$, or $R_{\rm max}$ for short—in subcritical evolutions, as a function of distance from the critical solution in parameter space (as first suggested by Garfinkle and Duncan [112]). If the near-critical solutions are approaching the spherically

$-\ln(\tau^\star-\tau)$	**Central** Φ/Φ_0	Δ/Δ_0
1.191	-0.82	1.01
2.925	1.02	1.00
4.610	-1.00	0.93
6.325	1.01	0.75
7.756	-0.96	—
8.859	1.12	—

Table 3.1: Extremes of the central value of Φ, normalized to the expected value $\Phi_0 = 0.426$, in near-critical spherically symmetric collapse ($\epsilon_\tau = 10^{-3}$ case). We also list the normalized values of the echoing exponent Δ, calculated using the difference in central proper time between the value listed at a given line in the table and one two lines further down (note that $\tau^\star$ was chosen to give $\Delta/\Delta_0 \approx 1$ in the second line of the table). We use $\Delta_0 = 3.4$. See Figure 3.4 for the corresponding curve of the central value of Φ versus $-\ln(\tau^\star-\tau)$.

symmetric one, then we expect the curve of $\ln(R_{\max})$ versus $\ln(A^\star - A)$ to approximate a straight line with slope $-2\gamma \approx -0.74$, modulated by a small oscillation with period $\Delta/2\gamma \approx 4.6$. We estimate the critical amplitude $A^\star$ by taking the average of the nearest-to-critical super and sub-critical amplitudes that we have been able to determine. See Figures 3.2 and 3.3 below for such plots from the $\epsilon_\tau = 10^{-3}$ and $\epsilon_\tau = 10^{-4}$ evolutions, respectively. We were able to tune the $\epsilon_\tau = 10^{-3}$ runs closer to criticality, as a larger TRE results in smaller grids in the AMR hierarchy, and hence consumes less computer resources[1].

A second method to estimate Δ, and to verify the self-similar nature of the solution, is to plot the central value of Φ versus logarithmic proper time $-\ln(\tau^\star-\tau)$, where τ (>0) is the proper time of a central observer, and $\tau^\star$ is the accumulation point of the critical solution (see Section 1.3). In these coordinates, the central value of Φ, when in the self-similar regime, is a periodic function with period Δ, oscillating between extremes of $\pm\Phi_0$. Choptuik [4] found a value $\Phi_0 \approx 0.603$; in our case, as we use a different overall scale in the Lagrangian for the scalar field, we would expect this value divided by $\sqrt{2}$, or $\Phi_0 \approx 0.426$. One difficulty that we have in obtaining this data in a trivial fashion is that, in an evolution tuned close to criticality, accumulated numerical error causes the center of the critical solution to drift along the axis, away from the coordinate origin $\rho = z = 0$. This drift is very small (around 10^{-5} in z) in absolute terms, however, due to the rapidly decreasing length-scales that unfold during evolution, the drift eventually becomes large enough that simply measuring Φ at $\rho = z = 0$ ceases to give accurate results. Where we measure Φ instead then, is along a timelike geodesic, placed at the origin at $t = 0$ with zero initial velocity (see Section 2.4.1 for a description of how we integrate geodesics within the code). It turns out that this geodesic tracks the center of the near-critical solution quite well. Figure 3.4 below shows the results for the nearest-to-critical solution we obtained with the $\epsilon_\tau = 10^{-3}$ case. The largest uncertainty in the plot comes from guessing what $\tau^\star$ is. We have chosen $\tau^\star$ such that the period between the first two maxima in Φ is roughly equal to the expected value of Δ, namely 3.4. Then periods between remaining features of the plot can be checked for consistency with this guess for $\tau^\star$—see Table 3.1 for a list of the extremes in the plot, which show reasonable agreement (to within about 1% at intermediate times) with expected results. Certainly, as we do not have a large number of echos in the solution, we cannot claim a high degree of accuracy in measuring Δ using this method. However, here we *are* measuring Δ *directly*, unlike in Figures 3.2 and 3.3, where we use the hypothesized one-mode unstable nature of the critical solution to obtain a relationship between the echoing period of the actual solution and that of the "wiggle" in parameter space.

[1]Recall that the critical solution is self-similar, and so the closer to criticality one tunes, the larger the range of lengths scales that appear in the solution, which all need to be resolved with a deeper grid hierarchy.

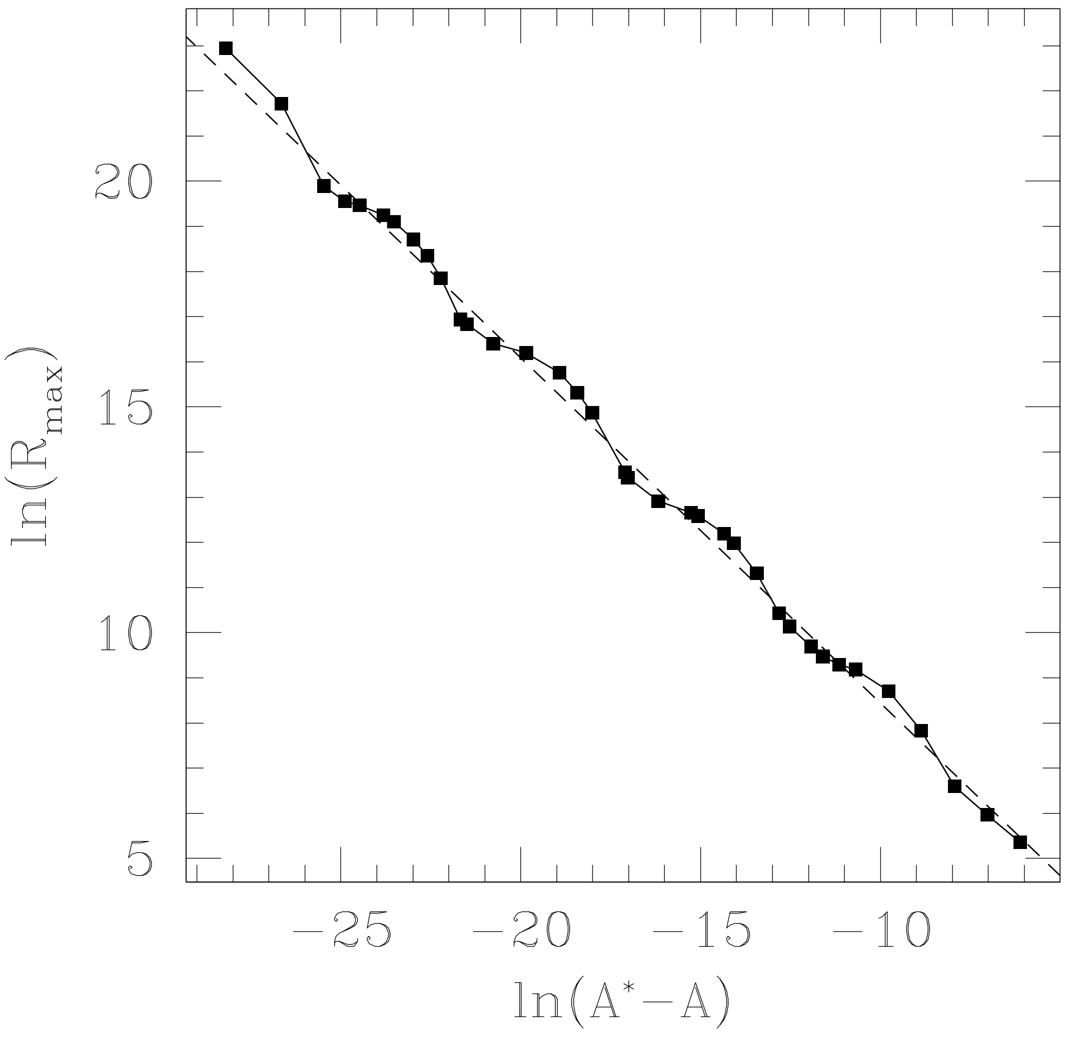


Figure 3.2: $\ln(R_{\mathrm{max}})$ vs. $\ln(A^\star - A)$ from sub-critical evolution of spherically symmetric initial data, with $\epsilon_\tau = 10^{-3}$. Each point on the curve represents a single evolution. A linear-regression line fit to the data is also shown, with slope $-2\gamma \approx -.76$. We estimate that the period $\Delta/2\gamma$ of oscillation of the plotted function about this line is ≈ 4.3. Hence, $\gamma \approx 0.38$, and $\Delta \approx 3.3$.

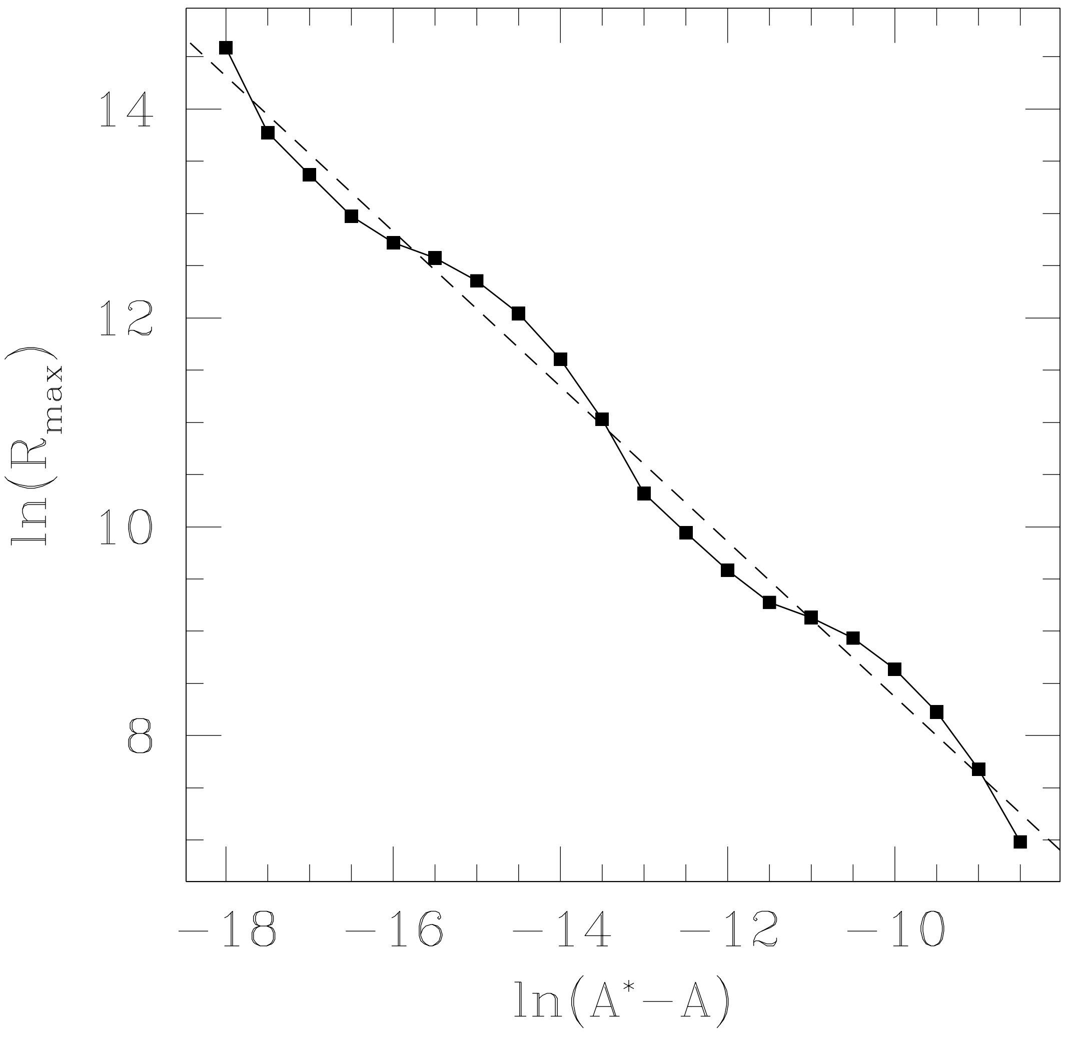


Figure 3.3: $\ln(R_{\mathrm{max}})$ vs. $\ln(A^{\star} - A)$ from sub-critical evolution of spherically symmetric initial data, with $\epsilon_\tau = 10^{-4}$. In this case, we estimate that $\gamma \approx 0.37$, and $\Delta \approx 3.3$.

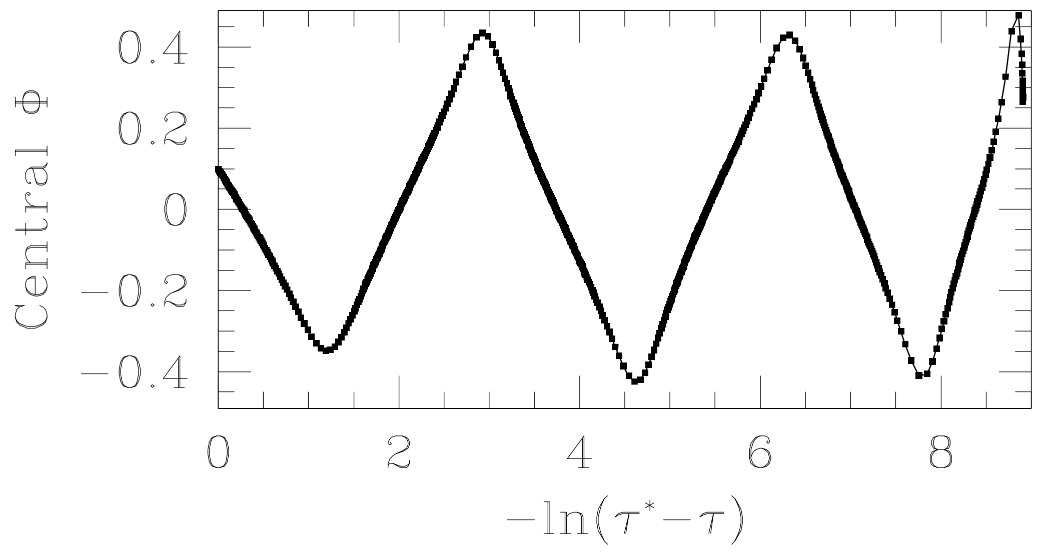


Figure 3.4: Central value of Φ as a function of $-\ln(\tau^\star - \tau)$, from spherically symmetric critical collapse (from the $\epsilon_\tau = 10^{-3}$ case). See Table 3.1 for a list of the minima and maxima of this curve.

Finally, in Figure 3.5 we show several snapshots of Φ from the near-critical, $\epsilon_\tau = 10^{-3}$ evolution, near $t = 0$ and at times where Φ attains a local extreme value (see also Figure 3.17 in Section 3.3.2 for a similar sequence of images). In the plot, we have changed to logarithmic coordinates in r to better demonstrating the self-similar nature of the solution.

All of the indicators studied above show that we obtain fairly good agreement with prior studies of spherically symmetric critical collapse—within $\approx 1\%$ measuring the extreme values of Φ, within $\approx 3\%$ the scaling exponent γ, and within $\approx 5\%$ the echoing exponent Δ. In subsequent sections, when we state that values of Φ_0, γ or Δ are "consistent" with the spherically symmetric solution, we mean that the deviations from the assumed spherically symmetric values are smaller than the uncertainties just quoted from our simulation of the spherically symmetric near-critical solution.

3.2 Small Deviations from Spherical Symmetry

In this section we present results of critical collapse from initial data with slight asymmetric deviations. The conclusions are that, as close as we are able to tune to the critical solution, we still obtain results consistent with a universal spherically symmetric solution. Hence, we only briefly show a few figures and tables, and spend more time in subsequent sections discussing the simulations with more significant asymmetries.

The two asymmetric families of initial data we consider here are a slightly prolate distribution—(2.52) with $\epsilon = 2/3$, and a slightly oblate distribution—(2.52) with $\epsilon = 3/2$; all other parameters are identical to those used for the spherically symmetric results presented in the previous section. Figures 3.6 and 3.7 below are plots of $\ln(R_{\text{max}})$ versus $\ln(A^\star - A)$ for the two families, with the spherically symmetric family overlaid for comparison, for runs with $\epsilon_\tau = 10^{-3}$ and $\epsilon_\tau = 10^{-4}$ respectively. For the $\epsilon_\tau = 10^{-3}$ nearest-to-critical solutions, Figure 3.8 below shows plots of the central value of Φ versus $-\ln(\tau^\star - \tau)$ for the three families, and Table 3.2 shows the corresponding values and times of the extremes in Φ for the $\epsilon = 2/3$ and $\epsilon = 3/2$ cases.

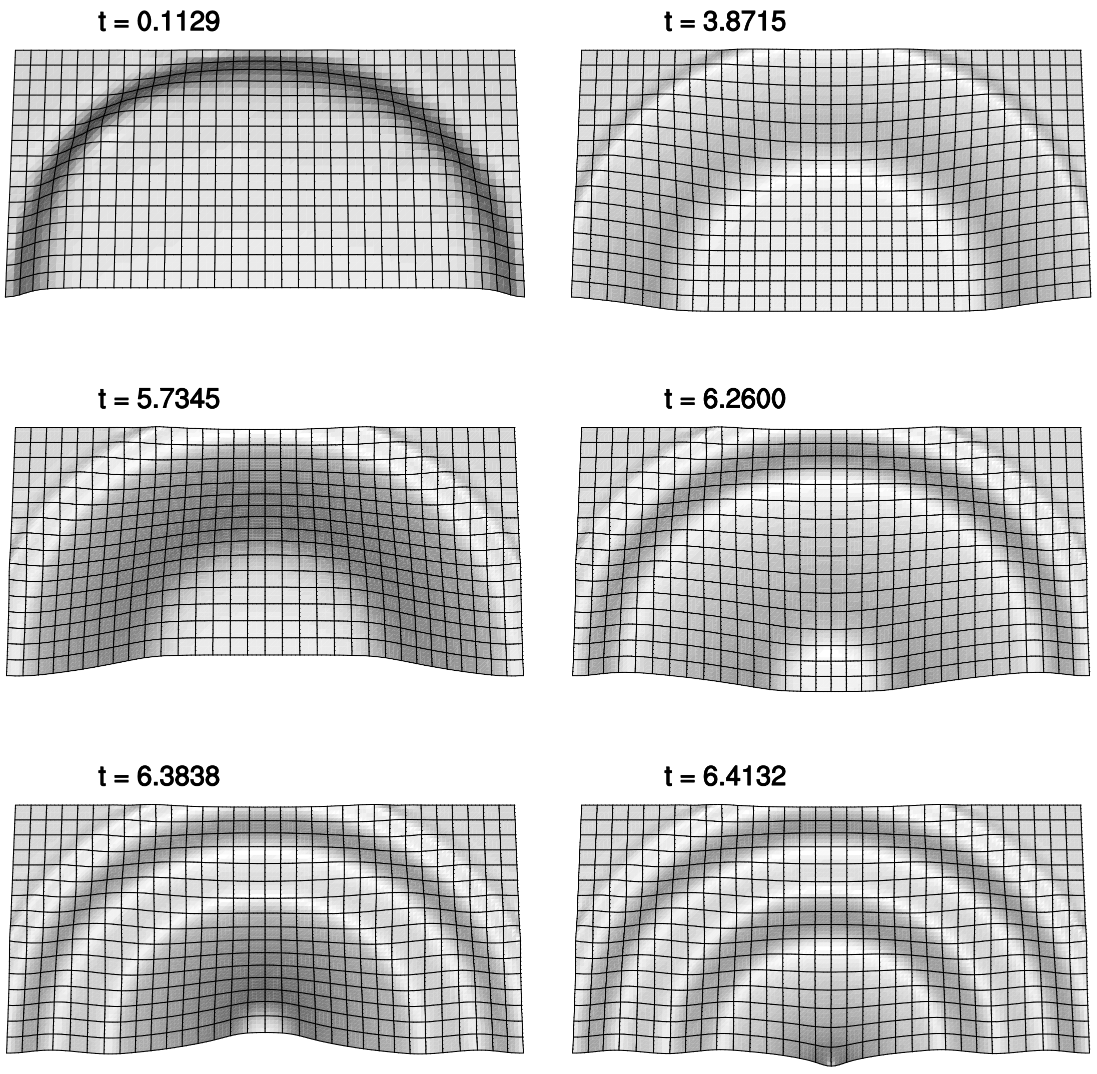


Figure 3.5: Evolution of Φ from spherically symmetric critical collapse. In this plot we have transformed the grid to logarithm coordinates in r, via $(r,\theta) \rightarrow (\ln(r+\epsilon_r),\theta)$, where $r = \sqrt{\rho^2+z^2}$, $\tan\theta = \rho/z$, and ϵ_r was set to 10^{-4} to give a reasonable scale for the figure. Thus the origin of each frame is at $r = 10^{-4}$, and roughly five orders of magnitude in r are spanned moving radially outward from the origin.

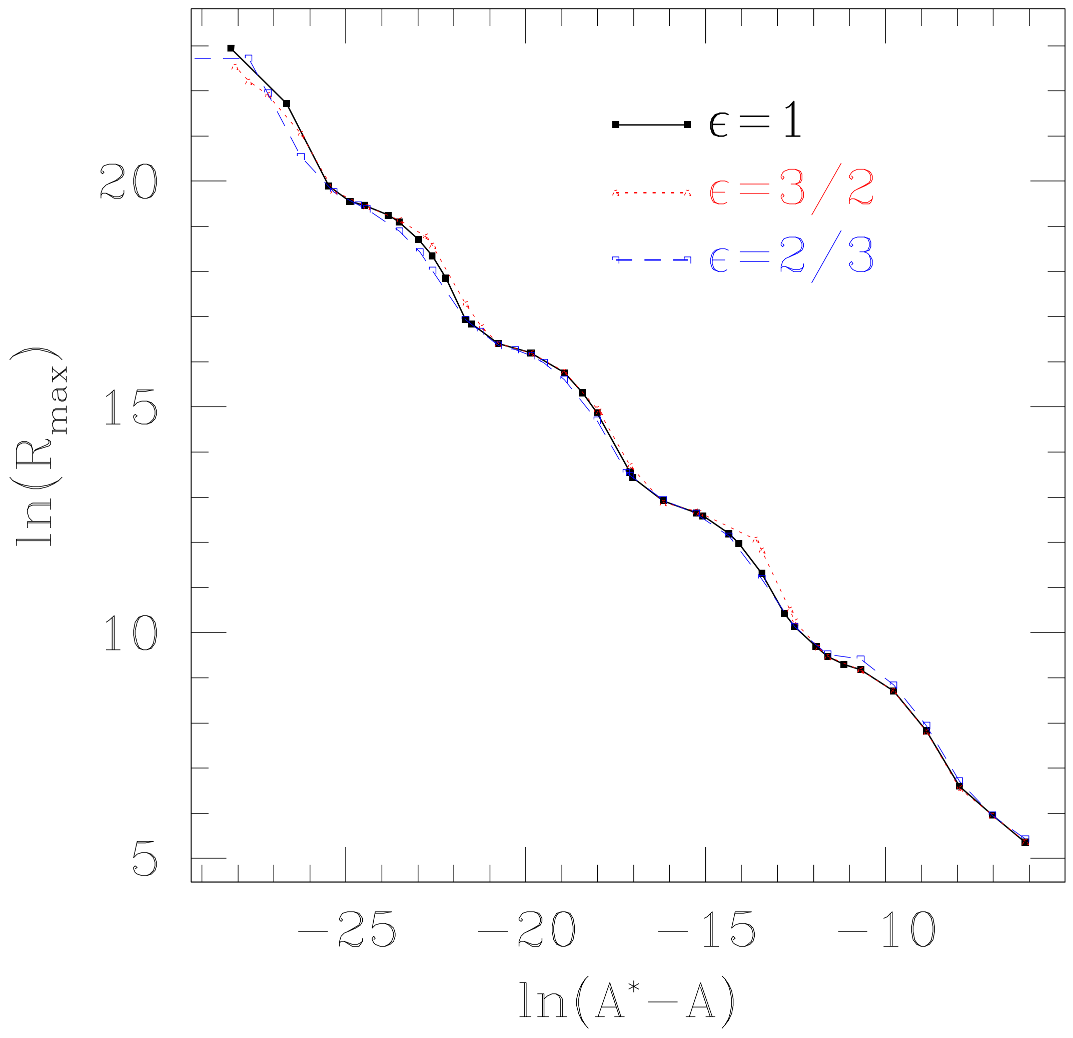


Figure 3.6: $\ln(R_{\rm max})$ vs. $\ln(A^\star - A)$ from sub-critical evolution of near-spherically symmetric initial data, with $\epsilon_\tau = 10^{-3}$. The curves from the two asymmetric families (see (2.52) for the definition of ϵ) of initial data have been shifted by constant amounts along the vertical axis to overlap as much as possible with the spherical data, for easier comparison.

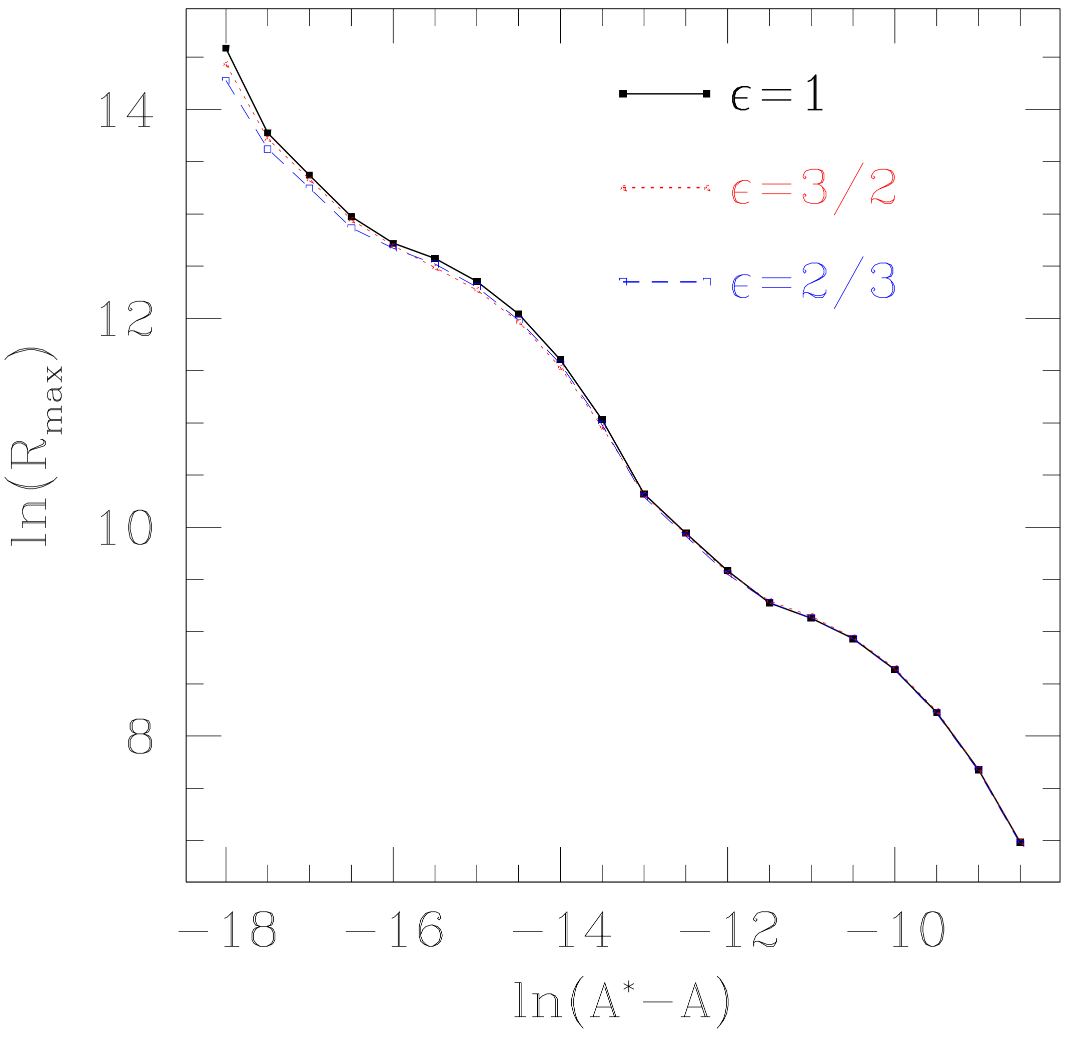


Figure 3.7: $\ln(R_{\max})$ vs. $\ln(A^\star - A)$ from sub-critical evolution of near-spherically symmetric initial data, with $\epsilon_\tau = 10^{-4}$. The curves from the two asymmetric families (see (2.52 for the definition of ϵ)) of initial data have been shifted by constant amounts along the vertical axis to overlap as much as possible with the spherical data, for easier comparison.

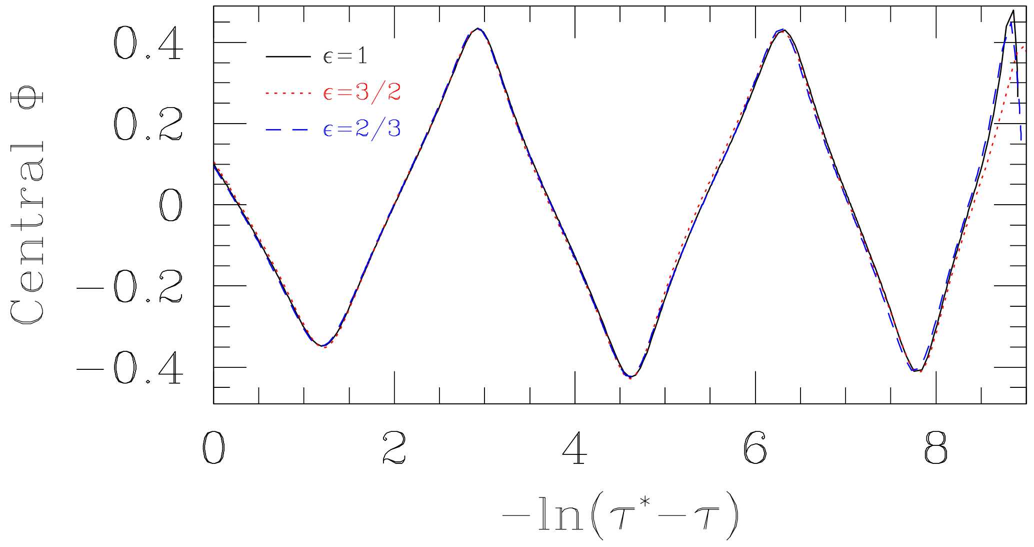


Figure 3.8: Comparison of the central value of Φ versus $-\ln(\tau^\star - \tau)$, for near-spherically symmetric initial data ($\epsilon_\tau = 10^{-3}$ case). The curves have been shifted by constant amounts along the time axis so that the first maxima of Φ overlap in the three cases. Table 3.2 contains the times and values of the extremes in Φ, for each of these curves.

	$\epsilon = 2/3$			$\epsilon = 3/2$	
$-\ln(\tau^\star - \tau)$	**Central** Φ/Φ_0	Δ/Δ_0	$-\ln(\tau^\star - \tau)$	**Central** Φ/Φ_0	Δ/Δ_0
1.127	-0.82	1.02	1.291	-0.82	0.99
2.844	1.02	1.00	2.974	1.02	1.00
4.582	-0.99	0.91	4.664	-1.00	0.95
6.254	1.02	0.74	6.371	1.00	0.78
7.682	-0.96	—	7.883	-0.96	—
8.783	1.05	—	9.027	0.92	—

Table 3.2: Extremes of the central value of Φ, normalized to the expected value $\Phi_0 = 0.426$, in near-critical, near-spherically symmetric collapse ($\epsilon_\tau = 10^{-3}$ case). We also show the normalized echoing exponent Δ/Δ_0, calculated as explained in the caption of Table 3.1. Note that as opposed to the data shown in Figure 3.8, we have not shifted the time values here.

3.3 Large Deviations from Spherical Symmetry

In this section, we present results of critical collapse, starting from initial data that deviates more drastically from spherical symmetry than that shown in the previous section.

3.3.1 Equatorial-Plane Antisymmetric Scalar Field Collapse

The first significantly aspherical family of initial data that we are going to look at is a configuration that is antisymmetric about the equatorial plain (or antisymmetric for short)—i.e. $\Phi(\rho, z) = -\Phi(\rho, -z)$. Figure 3.10 below is a sample of the initial data that we use for Φ. The specific function is $zG(\rho, z)$, where $G(\rho, z)$ is given (as before) by (2.52), with $\epsilon = 1$, $R_0 = 3$, $\Delta = 1$, $(\rho_0, z_0) = (0, 0)$ and A is our tunable parameter. We choose Π_Φ such that the scalar field is initially ingoing (see Section 2.2.3); all the other initial data functions and AMR parameters are the same as those used for the $\epsilon_\tau = 10^{-3}$ spherically symmetric evolution, discussed in Section 3.1.

Recall that one of the characteristic features of the discretely self-similar critical solution is that the central value of Φ oscillates back and forth between some fixed value Φ_0. Therefore, one reason to look at antisymmetric collapse is that one may expect that a qualitatively different critical solution would develop at threshold, for the antisymmetric property of the scalar field is preserved during evolution, and hence no $z = 0$ echoing behavior can develop. Surprisingly though, it turns out that the local solution that unfolds in the antisymmetric case, when one tunes to the threshold of black hole formation, *is the same critical solution as for spherically symmetric initial data*. What happens is the following (see Figure 3.9). For very large initial amplitudes, a single, encompassing apparent horizon, and hence black hole, forms during evolution. As the amplitude is reduced, at some intermediate value of A, *two distinct* black holes form, equidistant from $z = 0$ (and in fact moving away from one another). As we continue to reduce A towards $A^\star$, two local critical solutions develop *away* from $z = 0$. This is how the solution avoids the apparent paradox of a "central" value of Φ that must be zero, while the scalar field still exhibits the spherically symmetric critical solution.

Figure 3.11 below shows four snapshots of a near-critical evolution from antisymmetric initial data, demonstrating this behavior. The two echoing solutions that develop equidistant from $z = 0$ are, to within numerical errors, exactly out of phase; however this is a function of the initial conditions, and not because of any dynamical link between the two solutions (the critical behavior unfolds too rapidly for there to be any causal contact). Numerical errors actually prevent us from maintaining exact antisymmetry during evolution, and thus we can only simultaneously fine-tune both local solutions to within $\ln(|A - A^\star|) \approx -12$; beyond that, the black hole/dispersal bracketing process will "select" one of the two solutions to tune closer to critically, and in this case it was the one on the positive z axis[2]. Figure 3.12 is a plot of the late time behavior of the nearest-to-critical solution that we have obtained, for the $z > 0$ portion of the scalar field, in logarithm coordinates.

To present more quantitative evidence that the local near-critical solutions are the same as before, in Figure 3.13 below we plot $\ln(R_{\max})$ vs. $\ln(A^\star - A)$ from sub-critical evolutions, and overlay the results from the spherically symmetric $\epsilon_\tau = 10^{-3}$ case for comparison. Also, in Table 3.3 we show the local extremes attained by Φ during evolution. In the antisymmetric case, we have not yet been able to trace the local centers of symmetry with geodesics accurately enough to measure Φ as a function of central proper time[3]; we thus list the extreme values as functions of coordinate time t. In Table 3.3, the values of the extremes of Φ do not match the spherically symmetric values shown in Table 3.1—however, the *relative difference* between the minimum and

[2]We can rephrase this more correctly from a numerical standpoint as follows: for a given maximum truncation error, the antisymmetric family of initial data has two critical parameters, $A^\star_+$ and $A^\star_-$, corresponding to the local solutions that develop above and below $z = 0$ respectively. With $\epsilon_\tau = 10^{-3}$, $\ln(|A^\star_+ - A^\star_-|) \approx -12$, so while $\ln(|A - A^\star_\pm|) >\approx -12$, we can effectively fine tune both solutions simultaneously. Presumably, in the limit as $\epsilon_\tau \to 0$, $A^\star_+$ will coincide with $A^\star_-$.

[3]We have not been able to calculate correct initial conditions for the geodesic integration to accurately track the rapidly moving centers.

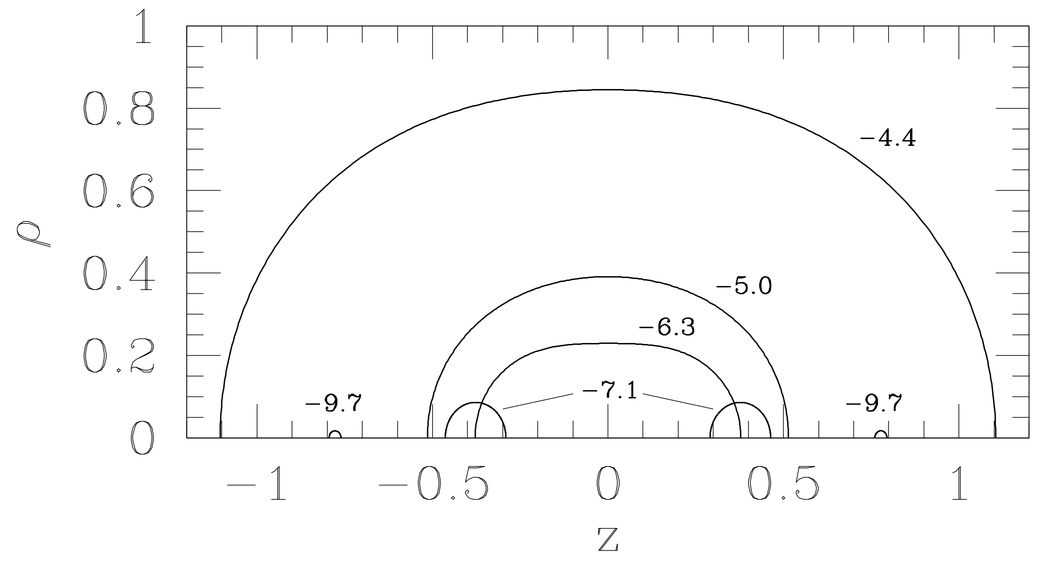


Figure 3.9: Apparent horizon shapes from super-critical, antisymmetric collapse. The shape of the apparent horizon(s) first detected during collapse are shown, from five simulations, each with a different initial amplitude A. Each curve is labeled by $\ln(A - A^{\star})$. Notice that for large initial amplitudes, a single AH, and consequently single black hole, forms. For smaller initial amplitudes, two distinct AHs form; the 2 horizons move away from each other during evolution and hence do not merge, indicating that the final state for small amplitude, super-critical collapse contains two black holes.

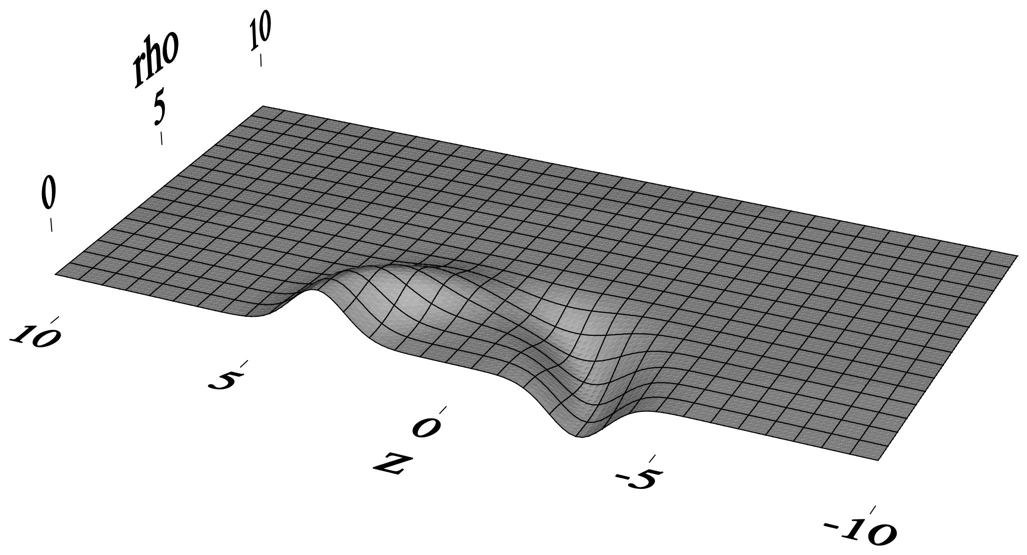


Figure 3.10: Initial profile of Φ for antisymmetric critical collapse.

maximum values attained by Φ does approach, at intermediate times, a number consistent with the spherically symmetric solution. For a massless scalar field, results that differ by a constant are effectively the same, for only the gradients of Φ gravitate (see e.g. 2.21).

3.3.2 Highly Prolate Initial Scalar Field Profiles

In this section, we look at the critical collapse of 4 families of initial data with increasing prolateness: $\epsilon = 1/2, 1/4, 1/6$ and $1/8$ (2.52). The motivation for studying this sequence stemmed from an early examination of collapse of a family with $\epsilon = 1/20$. There, as in the antisymmetric case, two black holes formed while still far from criticality (in the super-critical regime), while recall that for the $\epsilon = 2/3$ case presented in Section 3.2, a spherical-like critical solution was observed. This suggested that interesting behavior might be seen at some intermediate value of ϵ, where the transition between one and two black hole solutions occurred. However, as we will show in this section, it may be that there is an additional, "mildly" unstable mode in scalar field critical collapse, that will always cause two black holes to form eventually, if some asymmetry is present in the initial data. In terms of spherical harmonics $Y_{\ell m}$, this unstable mode has a $\theta = \tan^{-1}(\rho/z)$ dependence that is closest to an $\ell = 2$ mode (and, of course, $m = 0$).

All of the simulations results we show in this section were run with parameters identical to the $\epsilon_\tau = 10^{-3}$ spherically-symmetric case discussed in Section 3.1, except with differing values of ϵ for the initial distribution of Φ.

Figure 3.14 shows the $t = 0$ distribution of Φ for the four cases, in addition to the spherically symmetric case $\epsilon = 1$ for reference. Figure 3.15 shows the scaling of $\ln(R_{\max})$ vs. $\ln(A^\star - A)$ for these families. Except for the $\epsilon = 1/4$ case, and to a lesser extent the $\epsilon = 1/8$ case, the slopes are consistent with a scaling exponent γ of 0.37. However for $\epsilon = 1/4$ and lower, there is no well defined oscillation in this curve, and thus we cannot conclude much about the echoing exponent (other than that in this region of parameter space, this aspect of discrete self-similar behavior is

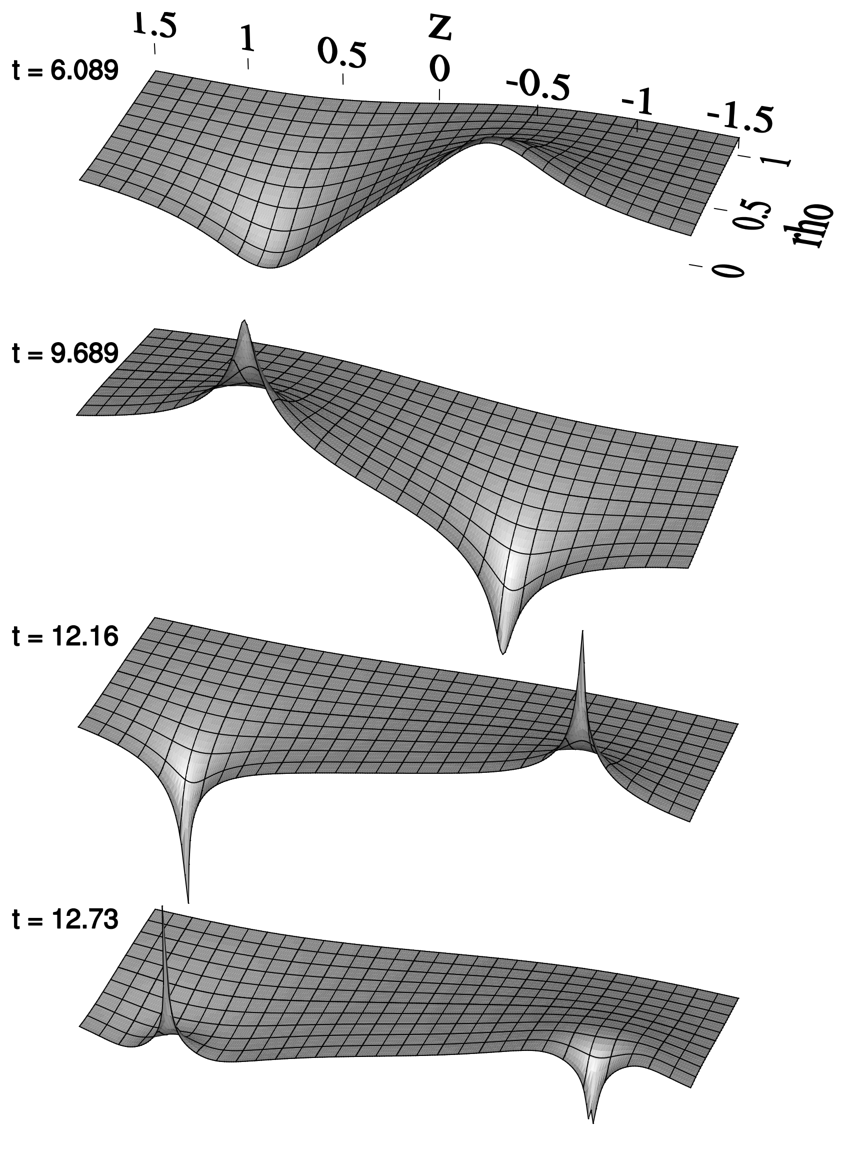


Figure 3.11: Evolution of Φ from antisymmetric initial data. Only a portion of the computational domain about the axis is shown, for clarity.

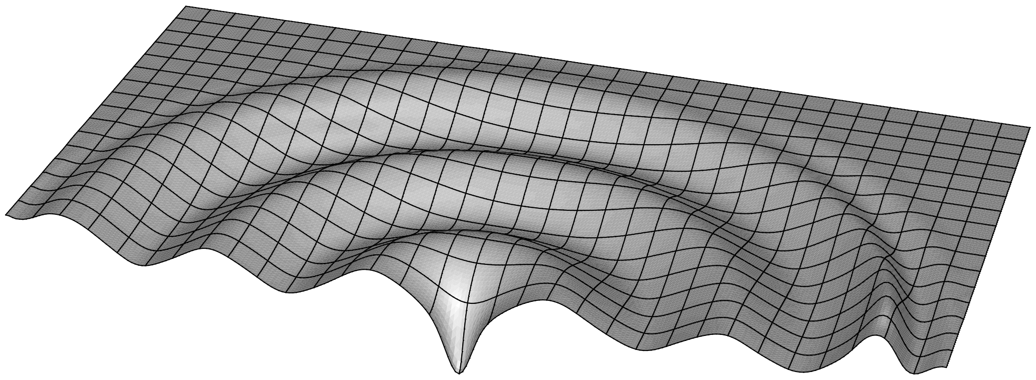


Figure 3.12: Late-time profile of Φ, evolved from near-critical antisymmetric initial data. In this plot we have transformed to logarithm coordinates in r, via $(r,\theta) \rightarrow (\ln(r+\epsilon_r),\theta)$, where $r=\sqrt{\rho^2+(z-z_0)^2}$, $\tan\theta=\rho/(z-z_0)$, and $\epsilon_r=10^{-4}$ (thus roughly five orders of magnitude in r are spanned moving radially outward from the origin $(0,z_0)$). The shift in z by z_0 was chosen so that the origin of the figure is coincident with the center of symmetry of the $z>0$ near-critical solution. The barely noticeable "feature" near the right side of the figure, along the axis, is the $z<0$ near-critical solution, rendered near invisible by the logarithmic transformation. (This figure was produced from data at $t=12.903$, three half-echoes in time later than that of the last plot shown in Figure 3.11).

t	**Central Φ, $z>0$**	**$\Delta\Phi/\Delta\Phi_0$, $z>0$**	**Central Φ, $z<0$**	**$\Delta\Phi/\Delta\Phi_0$, $z<0$**
6.093	-0.225	0.74	0.225	-0.74
9.673	0.405	-1.03	-0.405	1.03
12.164	-0.476	1.01	0.472	-0.81
12.734	0.382	-1.00	-0.222	
12.866	-0.470	0.98	—	
12.897	0.366	-0.77	—	
12.903	-0.292	—	—	

Table 3.3: Extremes of the local central values of Φ in near-critical antisymmetric collapse. We also show a normalized central value—$\Delta\Phi/\Delta\Phi_0$—calculated by taking the difference in Φ between two subsequent rows, and dividing by the expected difference $\Delta\Phi_0=0.852$. We do this to eliminate an (irrelevant) overall displacement in Φ. The $z>0$ part of the solution has been tuned closest to criticality, hence we see more self-similar echoes there.

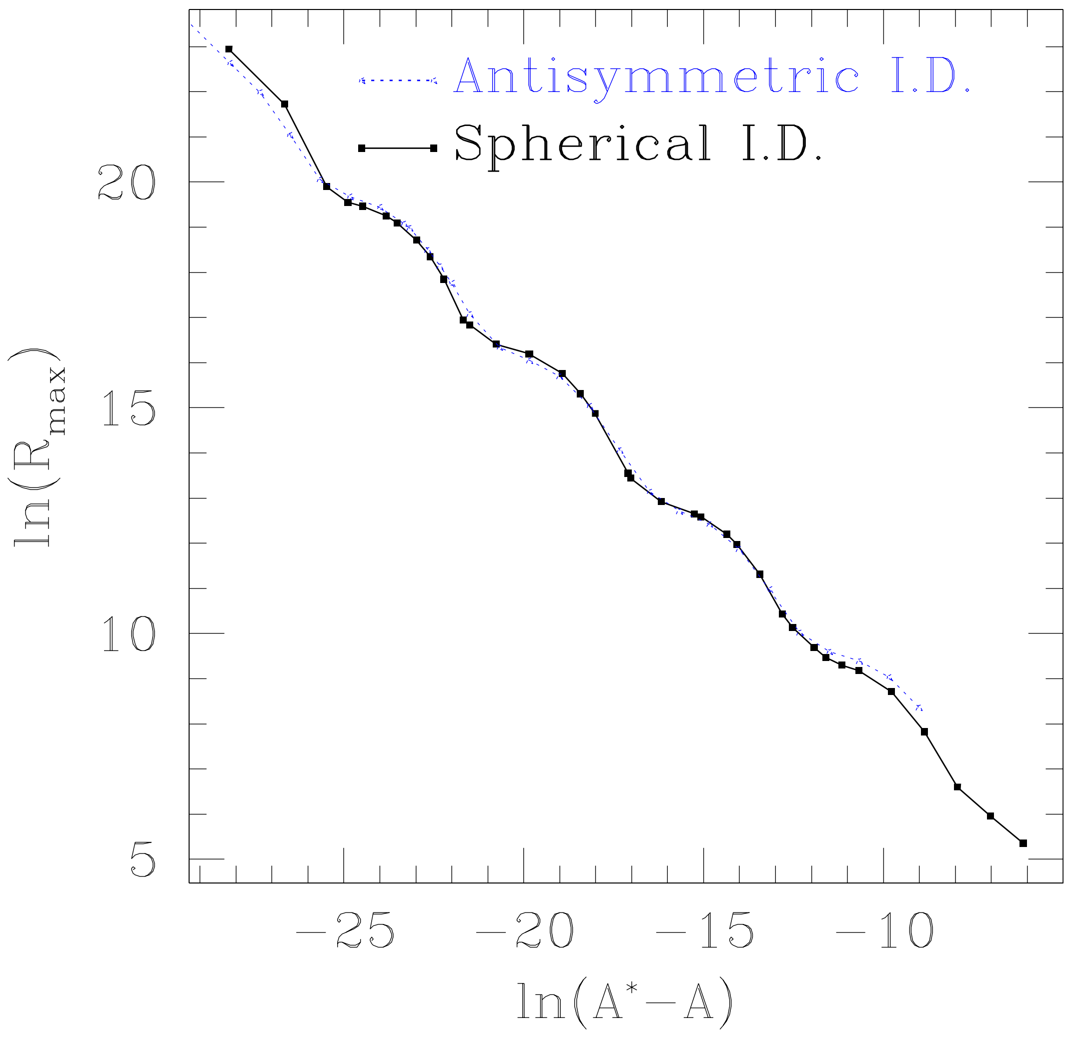


Figure 3.13: $\ln(R_{\mathrm{max}})$ vs. $\ln(A^{\star} - A)$ from sub-critical evolution of the antisymmetric initial data. The curve from the spherically symmetric, $\epsilon_\tau = 10^{-3}$ case in 3.2 is also shown for comparison (with the axis of one shifted so that the two curves overlap as closely as possible).

not too evident yet). As ϵ decreases, the maximum value of $|\ln(A^\star - A)|$ that we have been able to probe in our simulations decrease, due to increasing computer resource requirements. This is because the more prolate the initial data, the larger, more elongated set of clusters are needed in the AMR grid hierarchy to cover the region of high TRE.

Figure 3.16 plots the central value of the scalar field versus logarithmic proper time of a local geodesic observer (see the discussion in Section 3.1), for the nearest-to-critical solutions that we have been able to obtain in each case. Here, the deviation from the spherically symmetric solution, as ϵ decreases, is quite apparent (note that the deviation for smaller values of ϵ is *not* due to the fact that we are further from the critical solution in $|\ln(A^\star - A)|$ space—additional fine-tuning will not change the lower-ϵ curves appreciably). What appears to be happening with the smaller ϵ solutions is that two local centers of "symmetry" are developing. Thus, as we tune closer to criticality, the echoing behavior begins to unfold away from the origin (and thus away from the central observer). In Table 3.4, and Figures 3.17-3.22, we present some evidence for this. Table 3.4 shows the local extremes attained by the scalar field on axis, and the corresponding times and z-coordinates where the maxima or minima are observed. Notice that for $\epsilon = 1/6$ and $\epsilon = 1/8$, Φ still oscillates between values not too different from the spherical case, however, at some point the position where the local extreme values occur "bifurcates" into two local extremes, separated by some distance in z. For the $\epsilon = 1/6$ case, in Figure 3.22 we show plots of the shape of the apparent horizon that is first detected, from several super-critical evolutions. This also shows the single-to-double bifurcate behavior as we tune closer to criticality. Figures 3.17-3.21 are surface plots of the evolution of Φ in logarithmic coordinates in r, for all five cases of $\epsilon = 1$ to $\epsilon = 1/8$. The times of the plots were chosen to coincide with a minimum, maximum, or zero-crossing of the central value of Φ. Note that these plots are only meant to give a qualitative depiction of the supposed instability; in particular, we cannot dis-entangle slicing or coordinate effects between the solutions, which would render any quantitative comparison of the profiles of Φ somewhat meaningless. However, one invariant statement that we can make is this: for $\epsilon = 1, 1/2$ and $1/4$, we observe a sequence of *timelike separated, single* global minima/maxima in Φ; for $\epsilon = 1/6$ and $\epsilon = 1/8$, at early times we see the same behavior as with the larger ϵ, however at late times (later for $\epsilon = 1/6$ than $\epsilon = 1/8$) we see a situation develop where Φ attains a sequence of global minima/maxima in *two, spacelike separated* locations.

An intriguing aspect of this behavior seen in the $\epsilon = 1/6$ and $\epsilon = 1/8$ cases is that echoing behavior is observed about a single center *before* it develops about two local centers. For one possible explanation of multiple, local critical solutions is that they are due to a "focusing" artifact of the initial data. In other words, it may simply be that the prolateness causes the initial distribution, when evolved, to focus to two local centers—the closer to spherical the initial data, the closer the two foci are. However, by the supposition that the critical solution is a one-mode unstable solution, if one enters a regime where the solution is accurately described as a critical solution plus perturbations, then any asymmetries in the solution should decay with time, and this includes a "focusing asymmetry", where the energy density is slightly peaked in two locations about the center of the solution (and presumably, early on in the four prolate cases we have looked at we *are* in this regime, for we do see at least one cycle of a discretely self-similar collapse). Therefore, one would expect focusing of the scalar field to local centers, that then undergo gravitational collapse, to occur *before* critical behavior is observed there.

If the spherically symmetric critical solution does have a second, aspherical unstable mode, then eventually we should see a bifurcation develop in all simulations as we tune closer to threshold[4]. A further intriguing scenario may then develop, if the nature of the unstable mode is such that the bifurcation results in two, nearly spherical distributions of scalar field energy. For then, tuning closer to threshold, one should again see critical behavior develop about one of the local

[4]Including about the $z > 0$ near-critical solution observed in the antisymmetric collapse discussed in the previous section; though note that the bifurcation discussed here is qualitatively different from the two-to-one black hole transition shown for the antisymmetric collapse family in Figure 3.9—there, by construction, we never see any critical behavior about $r = 0$.

$\epsilon = 1/2$		$\epsilon = 1/4$		$\epsilon = 1/6$		$\epsilon = 1/8$	
t	Φ_{ext}/Φ_0	t	Φ_{ext}/Φ_0	t	Φ_{ext}/Φ_0	t	Φ_{ext}/Φ_0
4.488	-0.82	5.555	-0.83	6.504	-0.83	7.383	-0.83
6.634	1.03	8.270	1.01	9.803	0.93	11.314	0.83
7.249	-1.00	9.093	-0.94	10.902	-0.77	12.585	-0.92($2.7\text{x}10^{-2}$)
							-0.92($-2.7\text{x}10^{-2}$)
7.394	1.01	9.325	0.94	11.249	0.98($6.4\text{x}10^{-3}$)	13.145	0.89($1.2\text{x}10^{-2}$)
					0.98($-6.8\text{x}10^{-3}$)		0.89($-1.2\text{x}10^{-2}$)
7.429	-0.96	9.393	-0.83	11.417	-0.74($2.5\text{x}10^{-3}$)		
					-0.74($-2.9\text{x}10^{-3}$)		
7.437	0.81	9.416	0.68	11.441	0.45($3.9\text{x}10^{-3}$)		
					0.41($-4.3\text{x}10^{-3}$)		

Table 3.4: Extremes of the central values of Φ, normalized to the expected value $\Phi_0 = 0.426$, in near-critical prolate collapse. For the $\epsilon = 1/6$ and $\epsilon = 1/8$ cases, two local minima/maxima of the scalar field develop off-center—in those instances we list both values of the scalar field, followed by the corresponding z location in parenthesis.

centers. This would eventually result in a second bifurcation, as some of the asymmetric unstable mode would, in general, be present about the new local center. This process could be continued indefinitely, resulting in a fractal-like cascade of near-spherical critical solutions that unfold at the threshold of generic, axisymmetric collapse.

On the other hand, if the two centers that form in prolate collapse are due to focusing of the initial data, and the threshold solution is in fact universal, then further fine-tuning of the prolate cases should reveal a spherically symmetric critical solution forming about one of the local centers (again, as with the antisymmetric example, numerical errors will prevent us from fine-tuning both centers to threshold). Additional resolution will be needed to answer these questions, which may require that we parallelize the code, or find initial data that demonstrates this bifurcate behavior without excessive elongation of the resulting grids (which is the primary speed bottle-neck at this stage with the prolate collapse simulations).

3.4 Sample AMR Mesh Structure

We conclude this chapter by showing a few pictures of the typical AMR mesh structure that develops at late times in a critical collapse simulation. We use data from the antisymmetric evolution discussed in Section 3.3.1. Figure 3.23 is a view of the mesh structure at $t = 0$, and Figures 3.24 to 3.27 show the structure at $t = 12.903$ (i.e. at the same time as the plot shown in Figure 3.12).

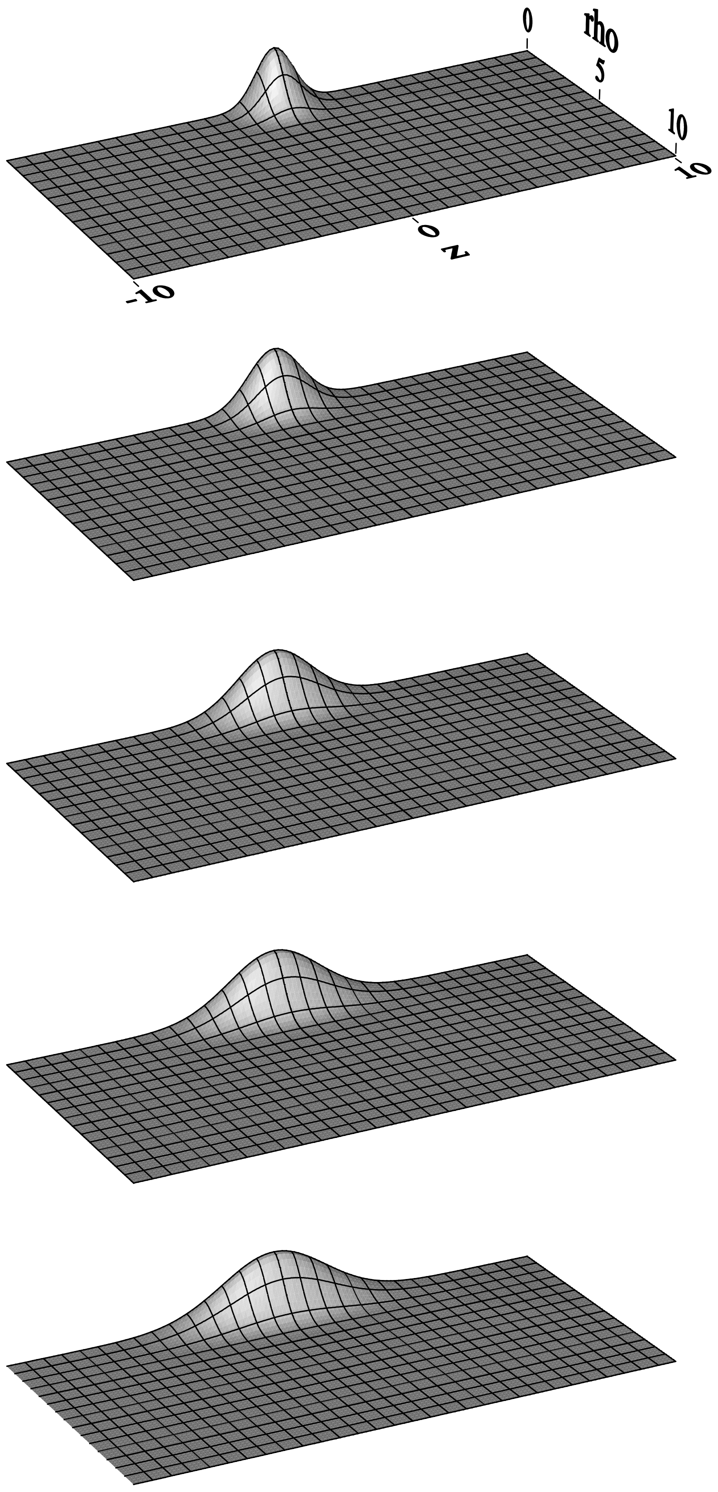


Figure 3.14: Initial profile of Φ, for prolate critical collapse. The top-most figure is the spherically symmetric initial data, with $\epsilon = 1$, for comparison; the next four figures, moving down the page, have $\epsilon = 1/2, 1/4, 1/6$ and $1/8$.

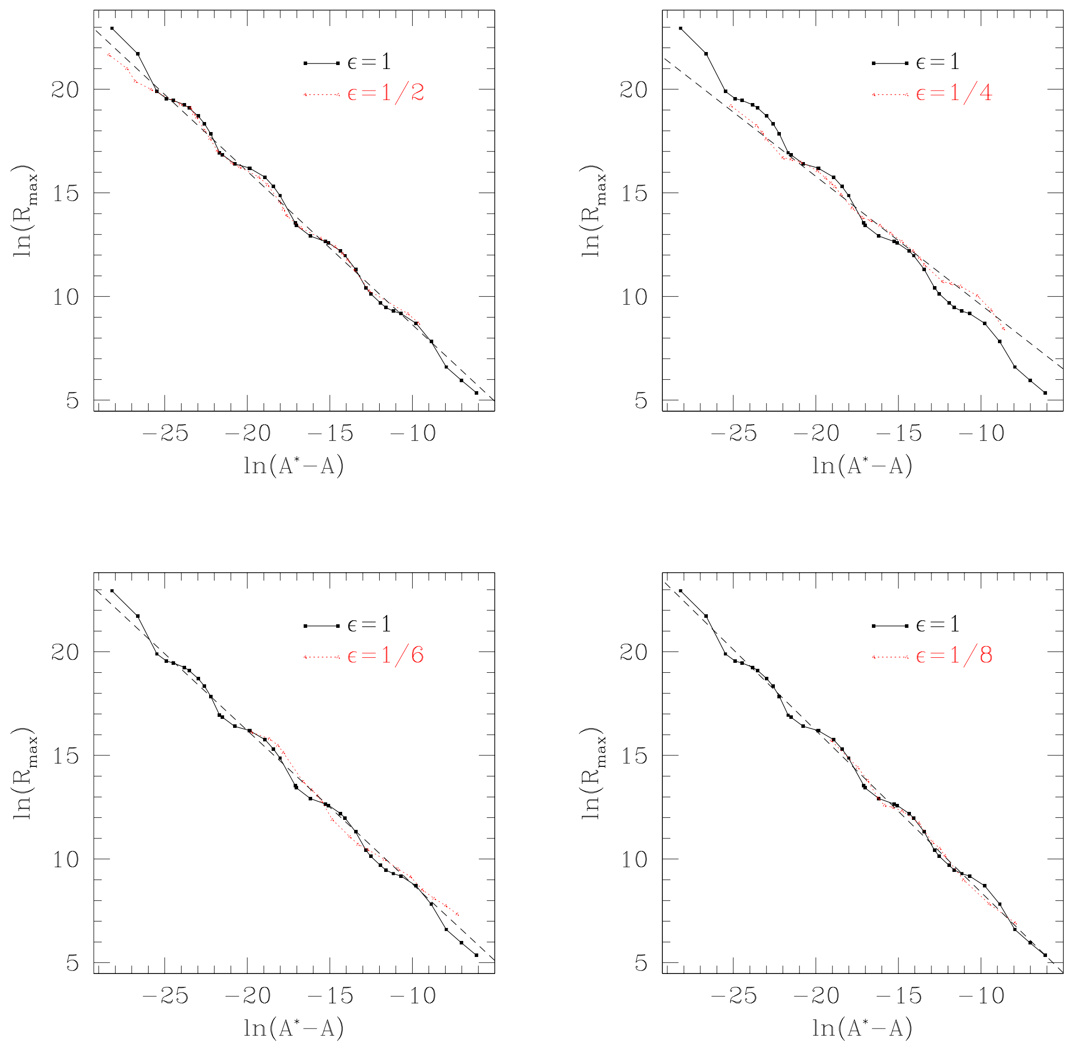


Figure 3.15: $\ln(R_{\rm max})$ vs. $\ln(A^\star - A)$ for $\epsilon = 1/2, 1/4, 1/6$ and $1/8$ prolate, sub-critical evolution. The $\epsilon = 1/2$ matches the spherically symmetric data closely, giving $\gamma \approx 0.37$ and $\Delta \approx 3.3$. For the $\epsilon = 1/4, 1/6$ and $1/8$ cases, the straight dashed-lines fitted to the corresponding curves give $\gamma \approx 0.31, 0.37$ and 0.39 respectively; however the periodic "wiggle" is not very evident in these cases, and so it would be difficult to estimate an echoing exponent.

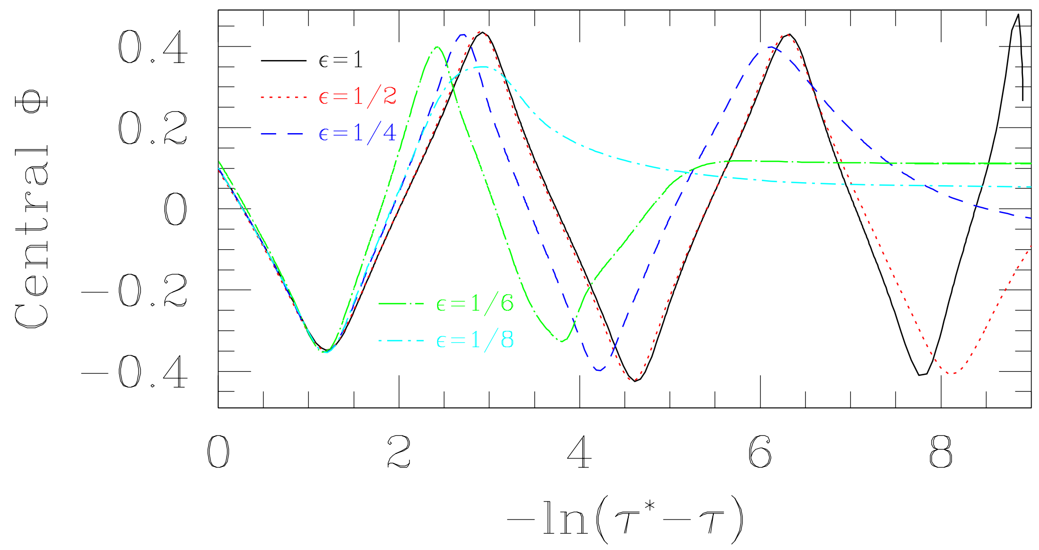


Figure 3.16: Central value of Φ as a function of $-\ln(\tau^{\star} - \tau)$, from prolate critical collapse. The data shown here was gathered from simulations with $\ln|A^{\star} - A| \approx -28, -28, -25, -20$ and -19 for the $\epsilon = 1, 1/2, 1/4, 1/6$ and $1/8$ cases, respectively.

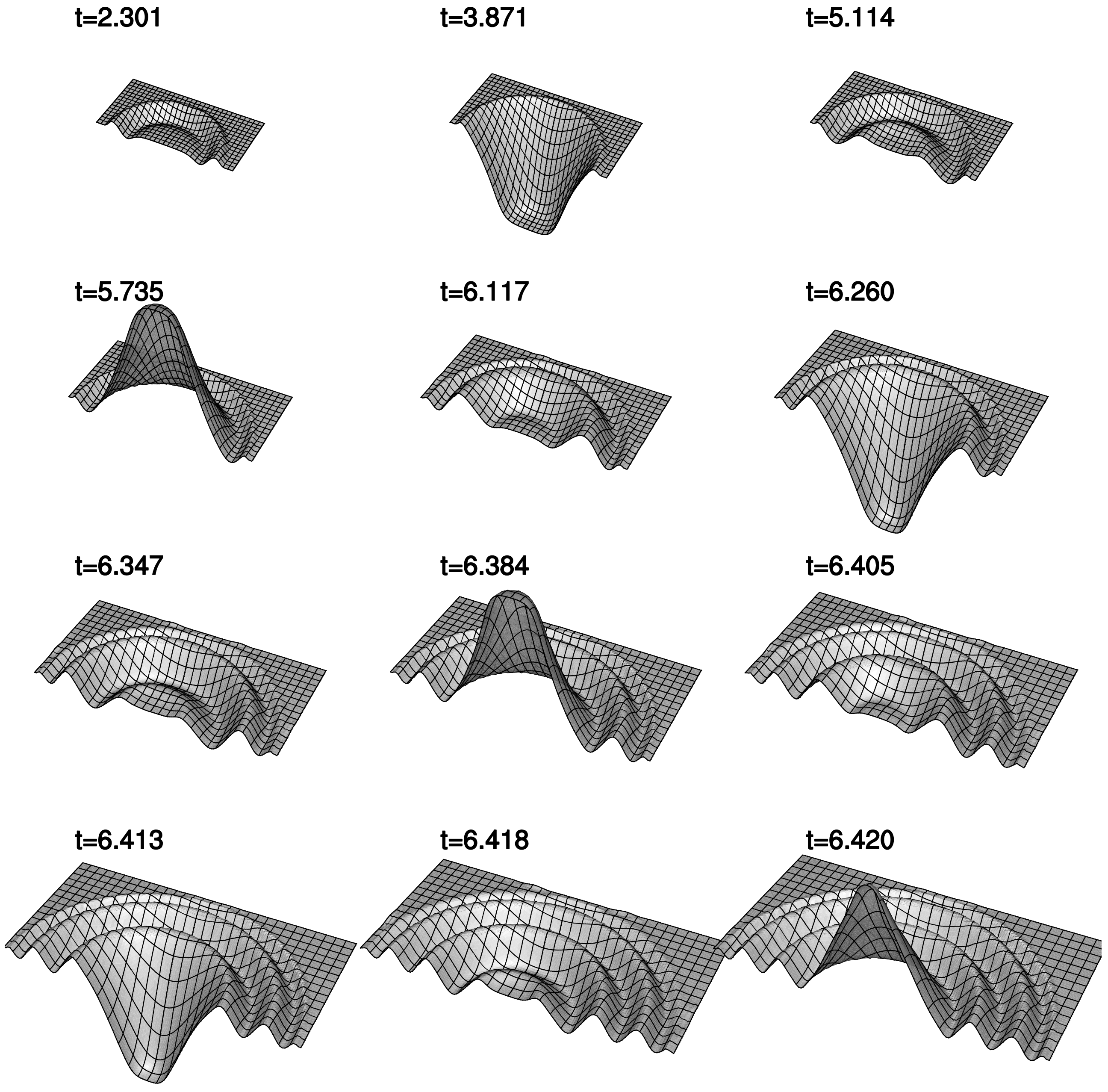


Figure 3.17: Near-critical evolution of Φ from $\epsilon = 1$ initial data. In each frame we have transformed to logarithmic coordinates in r, via $(r, \theta) \to (\ln(r + \epsilon_r), \theta)$, where $r = \sqrt{\rho^2 + (z - z_0)^2}$ and $\tan\theta = \rho/(z - z_0)$. ϵ_r ranges from 4.5×10^{-3} at the earliest time, to 8.0×10^{-7} at the latest time, so that the coordinate range in r covering the self-similar echoing part of the solution is *roughly* the same in all frames. z_0 was chosen to offset the slight drift in the center of symmetry that occurs with time (see Section 3.1). The particular times where chosen to coincide with the minima, maxima and zero-crossings of the local central value of Φ.

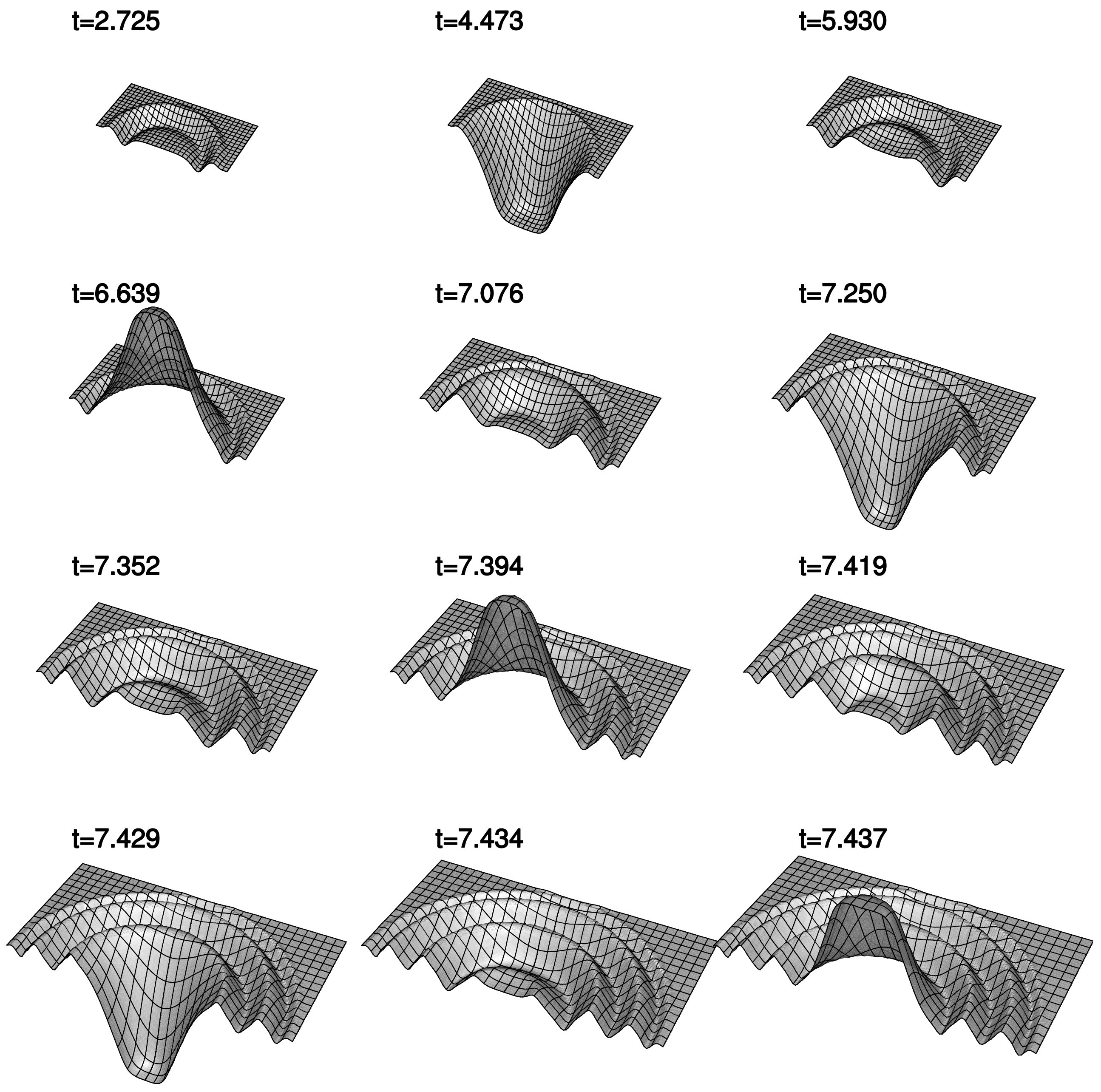


Figure 3.18: Near-critical evolution of Φ from $\epsilon = 1/2$ initial data. The same type of coordinate transformation has been applied here as with the $\epsilon = 1$ case in Figure 3.17 for comparison, and the same criteria was used to choose the set of times shown. Qualitatively, there is little to distinguish this sequence from the spherically symmetric case in Figure 3.17.

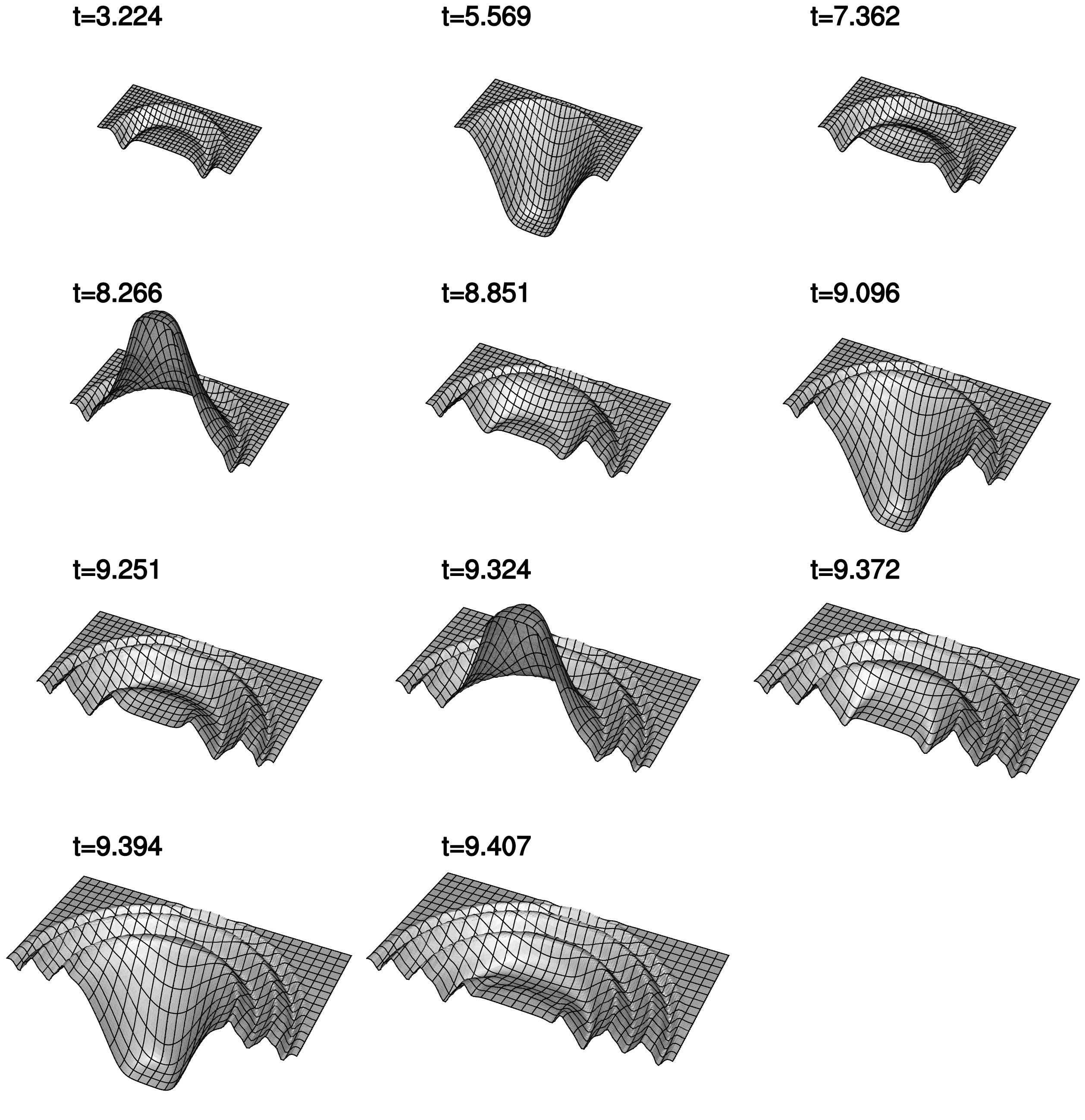


Figure 3.19: Near-critical evolution of Φ from $\epsilon = 1/4$ initial data. The same type of coordinate transformation has been applied here as with the $\epsilon = 1$ case in Figure 3.17 for comparison, and the same criteria was used to choose the set of times shown. In this case, the echoing behavior looks similar to that of the $\epsilon = 1$ and $\epsilon = 1/2$ cases in Figures 3.17 and 3.18, however notice (particularly at the the zero-crossing times of Φ, looking at the first oscillation away from the center) that there is a slight growth in the asymmetry with time.

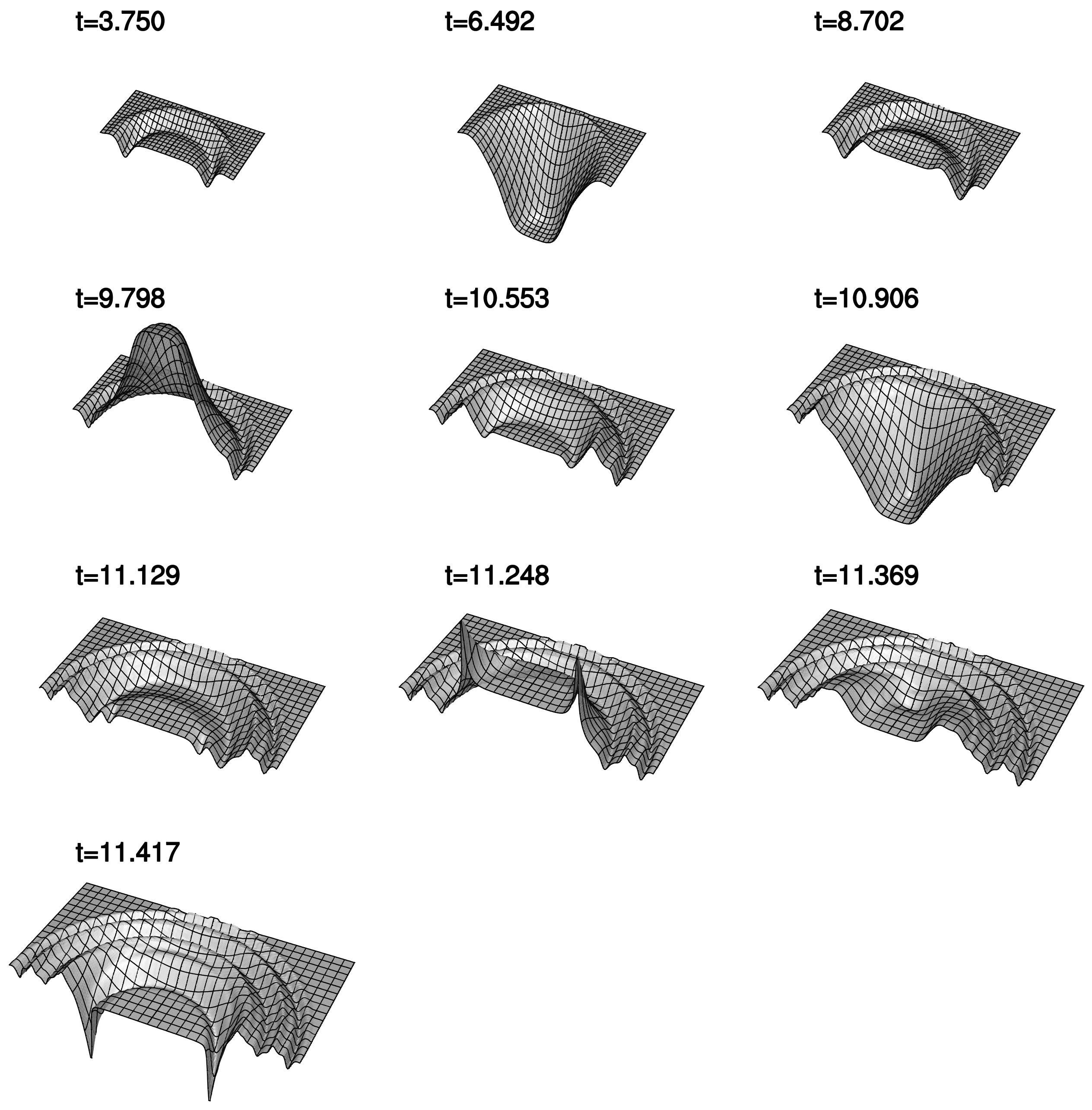


Figure 3.20: Near-critical evolution of Φ from $\epsilon = 1/6$ initial data. The same type of coordinate transformation has been applied here as with the $\epsilon = 1$ case in Figure 3.17 for comparison, and the same criteria was used to choose the set of times shown. Here, as in Figure 3.21 for the $\epsilon = 1/8$ case, the echoing behavior that develops off-center is quite apparent.

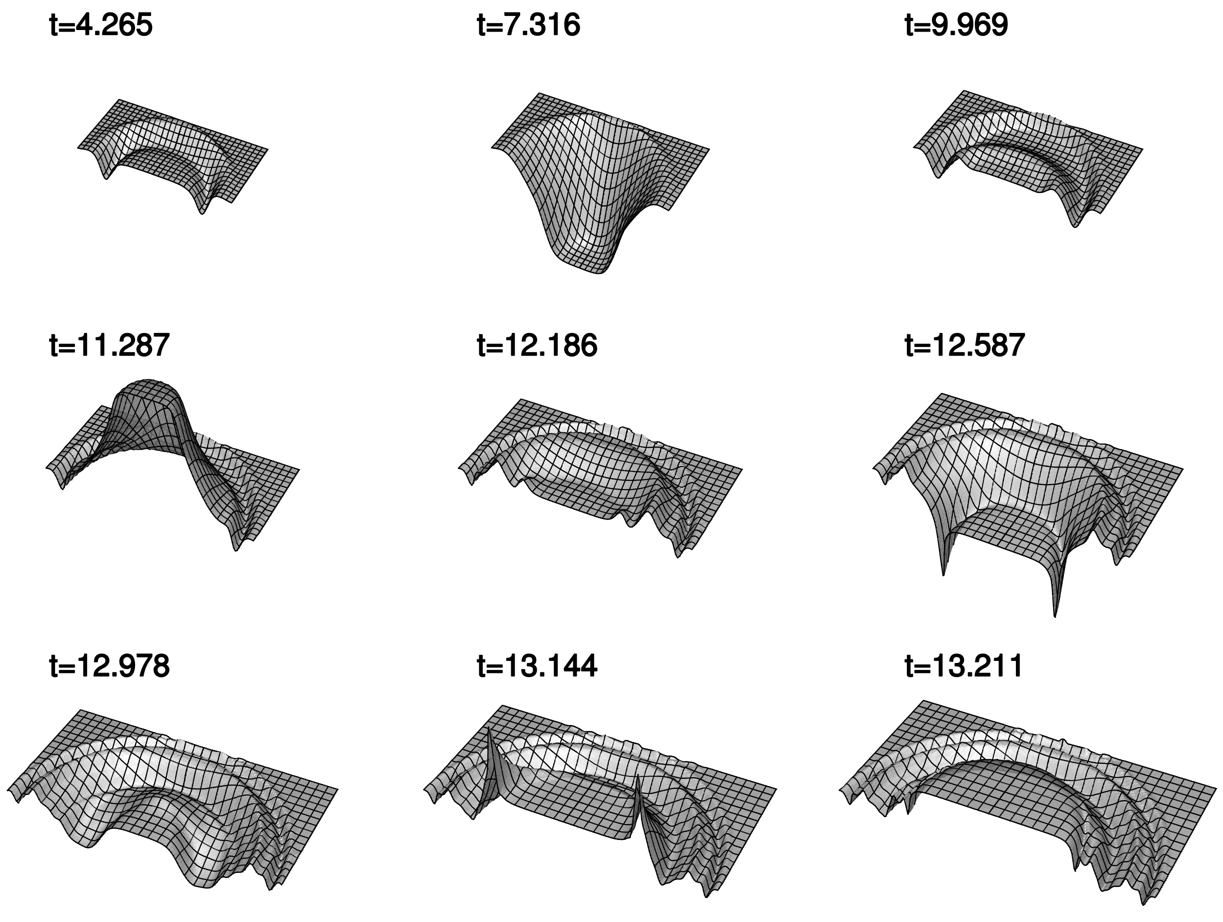


Figure 3.21: Near-critical evolution of Φ from $\epsilon = 1/8$ initial data. The same type of coordinate transformation has been applied here as with the $\epsilon = 1$ case in Figure 3.17 for comparison, and the same criteria was used to choose the set of times shown. Here, as in Figure 3.20 for the $\epsilon = 1/6$ case, the echoing behavior that develops off-center is quite apparent.

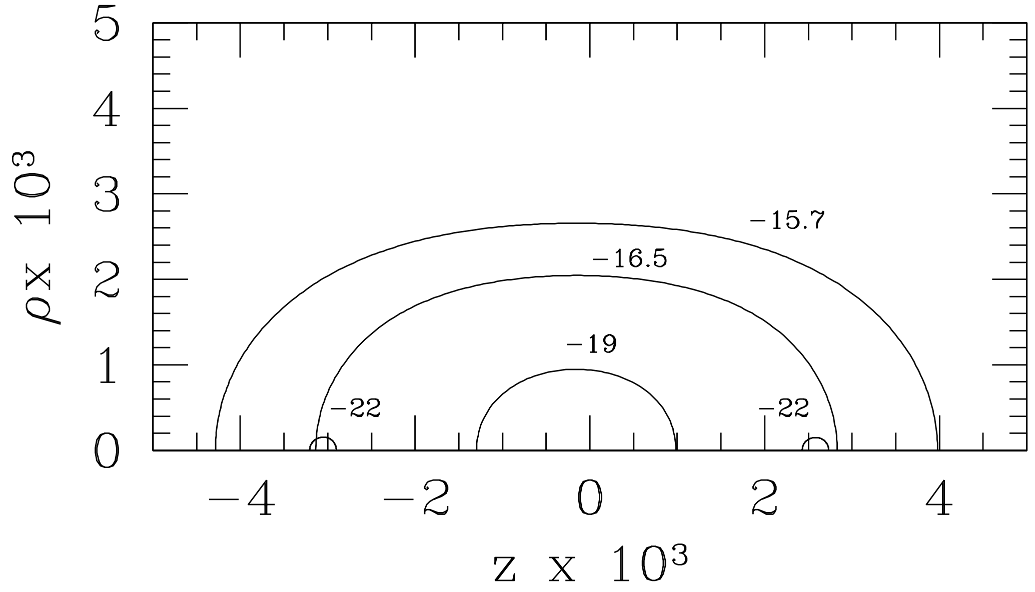


Figure 3.22: Apparent horizon shapes, as first detected, from super-critical, $\epsilon = 1/6$ prolate collapse. The labels are $\ln(A - A^{\star})$. Notice that for the closest-to-critical case shown here ($\ln(A - A^{\star}) \approx -22$), *two* apparent horizons are detected.

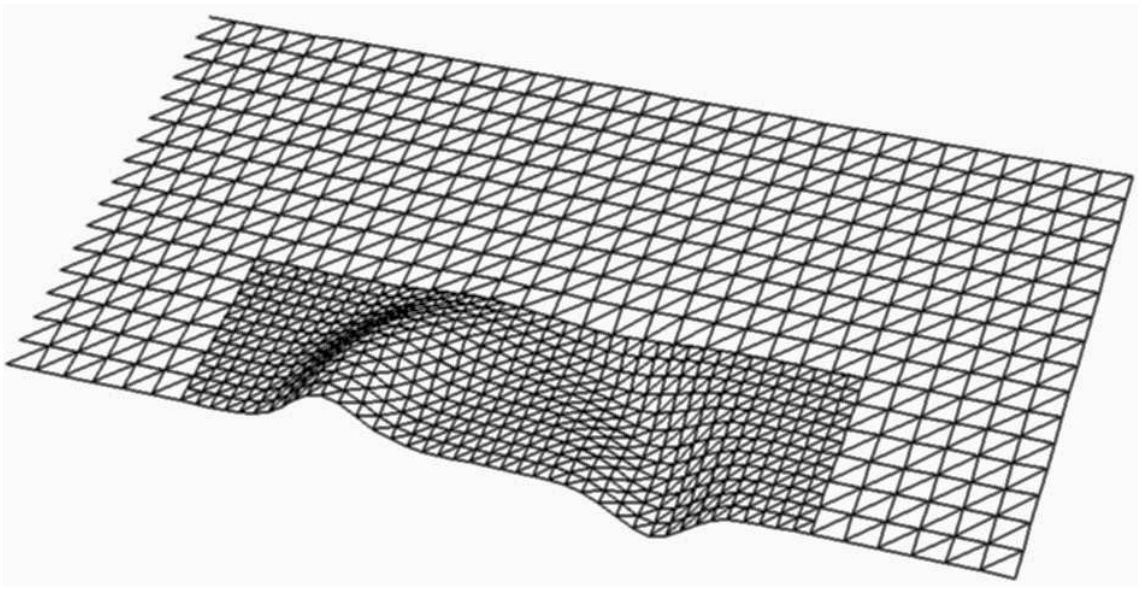


Figure 3.23: Φ at $t = 0$ from the antisymmetric scalar field collapse simulation (the same data as shown in Figure 3.10), drawn with a 4 : 1 coarsened wireframe mesh depicting the AMR grid hierarchy.

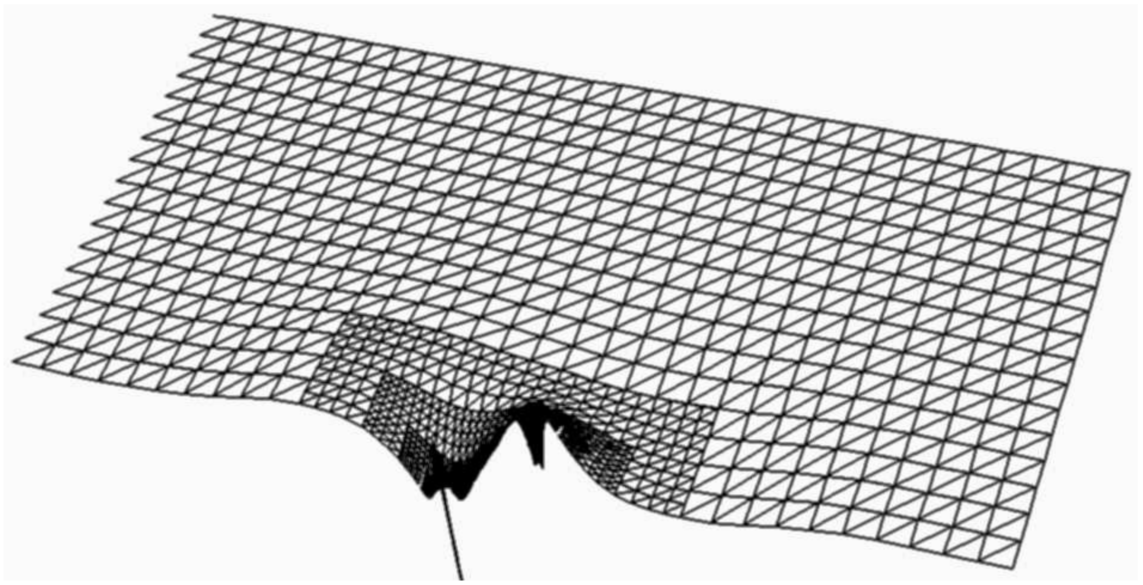


Figure 3.24: Φ at $t = 12.903436$ from the antisymmetric scalar field collapse simulation (the same time as in Figure 3.11), with a $2:1$ coarsened wireframe mesh depicting the AMR grid hierarchy. z ranges from -10 to 10 here—see Figures 3.25-3.27 for a series of close-up views of this mesh structure.

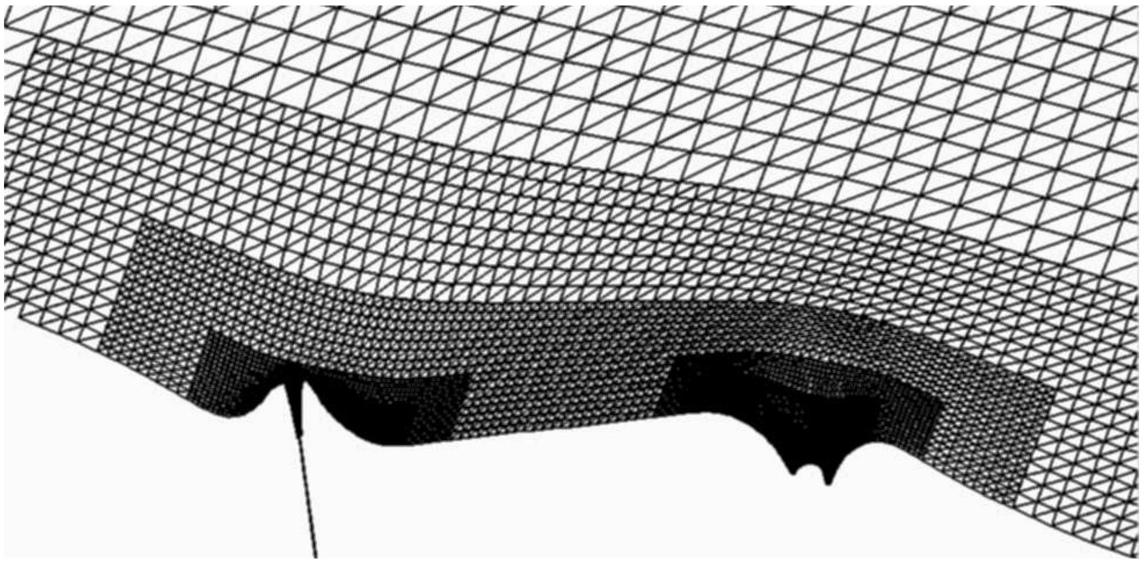


Figure 3.25: The same data is shown here as in Figure 3.24 above, though we have zoomed in by a factor of ≈ 4, so that the visible part of the axis ranges from $z \approx -2.5...2.5$ (coincident with the z-bounds of level 4 in the hierarchy).

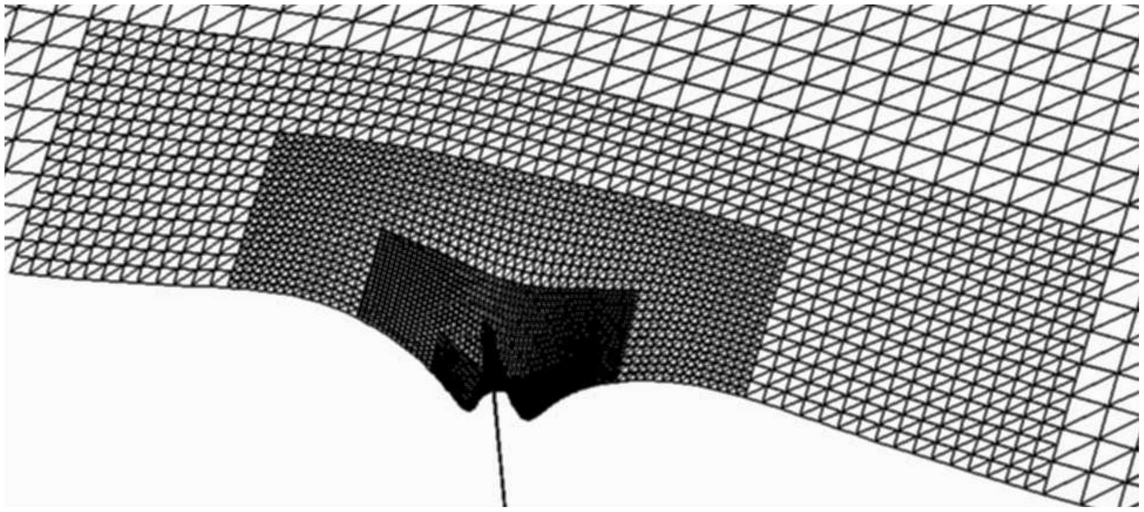


Figure 3.26: The same data is shown here as in Figure 3.24 above, though we have zoomed in by a factor of ≈ 70, so that the visible part of the axis ranges from $z \approx 0.91...1.19$ (coincident with the z-bounds of the level 8 grid surrounding the positive-z echoing solution).

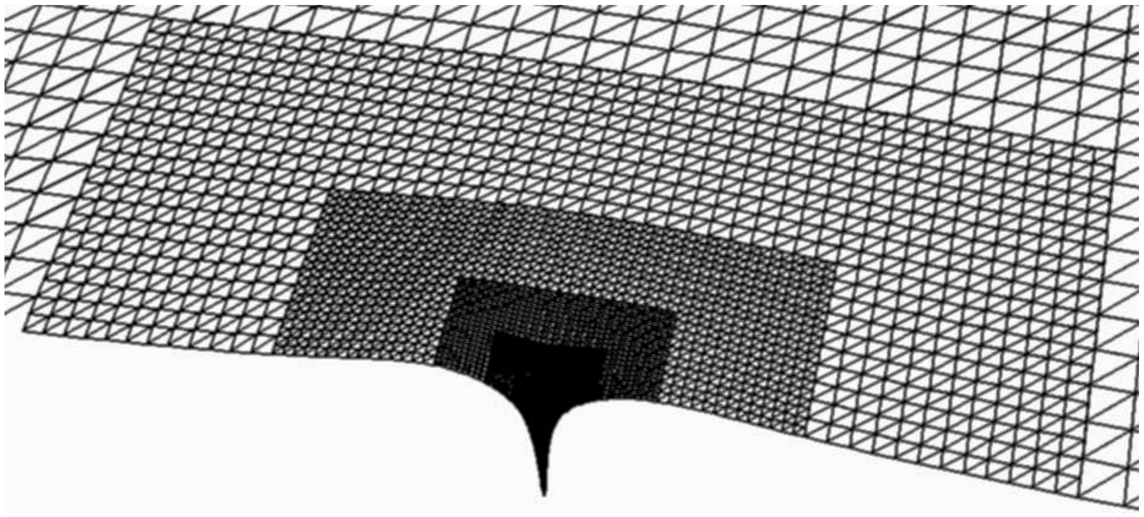


Figure 3.27: The same data is shown here as in Figure 3.24 above, though we have zoomed in by a factor of ≈ 2800, so that the visible part of the axis ranges from $z \approx 1.0599...1.0672$ (coincident with the z-bounds of the level 13 grid surrounding the positive-z echoing solution).

CHAPTER 4

BLACK HOLE EXCISION

In this chapter, we briefly include some of the results from unigrid simulations of black hole formation, showing the excision aspects of the code. In Section 4.1 we show simulations of an initially prolate scalar field collapsing to form a black hole, and in Section 4.2 we present a simulation of a black hole collision.

Our main purpose in this chapter is to demonstrate the feasibility of the excision mechanism in our code, namely, that we can achieve relatively long term, stable evolution of black holes spacetimes by applying an excision technique. With regards to obtaining new physical results, the code is not yet adequate, for two main reasons. First, as discussed in Section 2.2.6, the excision boundary conditions used for β^ρ and β^z are not fully consistent with the equations of free evolution. As shown in the examples below, the inconsistency is relatively small, producing errors on the order of the truncation error of the lower resolutions that the code was run at. However, such an inconsistency still needs to be removed if we want to explore physics with the code. Second, for the process of perhaps the most interest at this time, namely black hole collisions, the resolution demands for a unigrid code are excessive. The combined requirements of adequately resolving each black hole, and placing the outer boundary far enough away to obtain reasonable measures of the gravitational waves emitted during the collision process, result in desired resolutions several times higher than what we can obtain with the unigrid code. Consequently, at the current highest practical unigrid resolution of roughly 300^2, the waveform measurements we obtain are significantly worse (primarily due to reflections off the too-close outer boundary) than those originally obtained by Smarr [77] more than 20 years ago (and other authors since [78, 79, 80, 81, 89]), using less powerful computers. However, the eventual goal of this code is to study a wide range of axisymmetric gravitational phenomena, necessitating a more general coordinate system than the "body-fitting" type of coordinates used by Smarr to attack the specific problem of equal mass, head-on collisions. Using a more general coordinate system then forces us to implement an AMR scheme to achieve the necessary resolution for a range of problems (and perhaps more importantly, AMR will allow us to search for new physical phenomena, where we cannot foresee what the best coordinate system may be). Therefore, we will wait until we have incorporated excision into our AMR code, and have fixed the boundary conditions, before tackling the physics of black hole spacetimes.

4.1 Scalar Field Collapse

In this section, we consider the collapse of a time symmetric, prolate distribution of scalar field energy. The initial conditions are: Φ takes on the form (2.52), with $A = 0.32$, $\Delta = 2$, $(\rho_0, z_0) = (0,0)$ and $\epsilon = 1/20$, and $\Pi_\Phi = \bar{\sigma} = \bar{\Omega} = 0$. We consider two sets of runs, one with the outer boundary at $\rho_{\max} = z_{\max} = -z_{\min} = 40$, the other at $\rho_{\max} = z_{\max} = -z_{\min} = 80$. We use free evolution for the conformal factor ψ. For the $\rho_{\max} = 40$ outer boundary case, we performed three evolutions, with grid sizes of 65x129, 129x257 and 257x513 that correspond to mesh spacings of h, $h/2$ and $h/4$, with $h = 40/64$. For the $\rho_{\max} = 80$ case, we only have two evolutions, with grid sizes of 129x257 and 257x513, i.e. with resolutions corresponding to the h and $h/2$ runs of the $\rho_{\max} = 40$ case; an $h/4$ run for the $\rho_{\max} = 80$ case would have been too computer-resource intensive. For excision parameters (see Section 2.2.6), we use "frozen" boundary conditions, the minimum coordinate radius of the excised region is set to 2, and the maximum buffer between the apparent horizon and excision surface is set to 4, for all cases. We only start excising once the apparent horizon has been detected; however, if at a later time we "lose" the AH (either the AH moves inside the surface of excision, or the resolution is too poor and the AH-finder fails), the simulation still continues.

Figures 4.1 and 4.2 show several frames of the evolution of Φ ($r\Phi$ is plotted, where $r = \sqrt{\rho^2+z^2}$), from the 257x513, $\rho_{\max} = 40$ case. Figure 4.3 shows the shapes of the apparent horizon, at the same times as the plots in Figures 4.1 and 4.2, for comparison. Figure 4.4 is a plot of the ADM mass as a function of time for the different runs, while Figure 4.5 gives an estimate of the mass based on the apparent horizon area A, from

$$M_{area} = \sqrt{\frac{A}{16\pi}}. \tag{4.1}$$

An interesting aspect of highly prolate collapse, demonstrated in Figure 4.4, is that the initial shape of the AH is nearly spherical. We have noticed this in simulations with the prolateness factor ϵ as small as 1/200 (2.52). This is consistent with the spirit of Thorne's *hoop conjecture*, stating that black holes with horizons form when and only when a mass M gets compacted into a region whose circumference in every direction is $\leq 4\pi M$ [102].

Figure 4.6 has plots of $E(\dot{K}_\rho{}^\rho)$ for the five simulations. The inconsistency discussed in Section 2.2.6 is noticeable in the $\rho_{\max} = 40$ plot, where the norm of $E(\dot{K}_\rho{}^\rho)$ for the $h/4$ run almost doubles after excision starts (at $t = 23.5$). A part of the increase can be attributed to the relative closeness of the outer boundary, however, in the vicinity of the excision surface, the residual does not converge to zero as the outer boundary is moved further away. Figure 4.7 shows the convergence factor Q_ψ for ψ, obtained from the $\rho_{\max} = 40$ runs (it is unusually high in this case—typically we notice convergence factors between 2 and 4 in most functions, after excision begins).

Finally, in Figure 4.8, we plot the ADM mass and $E(\dot{K}_\rho{}^\rho)$ for the h resolution, $\rho_{\max} = 80$ run, till $t = 6000$ (or roughly $t = 1000M$), to demonstrate that we can obtain relatively long term evolution with excision. We only show a plot for the low-resolution run, as the $h/2$ run would have taken too long to complete (we estimate about a month and a half, compared to several days for the h run). Simulations with excision will not be able to run "forever" though, for, even with ideal parameters[1], the mass of the spacetime slowly decreases due to the dissipation, and eventually the excision surface will move outside of the apparent horizon, causing a CFL-type instability (though presumably a set of excision boundary conditions could be devised that will keep the excision surface within the AH for a longer period of time).

4.2 Black Hole Collisions

In this section we present results from a black hole collision simulation. In contrast to other simulations of black hole mergers performed to date [77, 78, 79, 80, 81, 103, 104, 105, 59], with the exception of [89], our initial spatial slice does not contain any black holes; rather, we start with highly concentrated local distributions of scalar field energy, which, upon evolution, promptly collapse to black holes that subsequently merge. To obtain initial conditions that mimic black holes moving towards each other with a given initial velocity, we use Lorentz-boosted scalar field profiles, as described in Section 4.2.1 below (however, for the sample collision presented here we use time-symmetric initial data). The reason we construct initial conditions for mergers via scalar field collapse, is simply that to do otherwise would require modification of the constraint solver at the initial time (namely, specifying an initial excised region with appropriate boundary conditions). The relatively minor drawback to this approach (if our goal is to study vacuum collisions) is that some spurious scalar field energy will permeate the spacetime after black hole formation; however,

[1]The two most important parameters affecting the run-time with excision are having adequate resolution about the surface of excision, and placing the outer boundaries far enough away (if they are too close, ψ tends to drift towards values < 1 near the $z_{\max}, z_{\min}$ boundaries, eventually causing a crash). The boundary conditions on the excision surface, as well as the smoothing operators used there, can also affect the run-time.

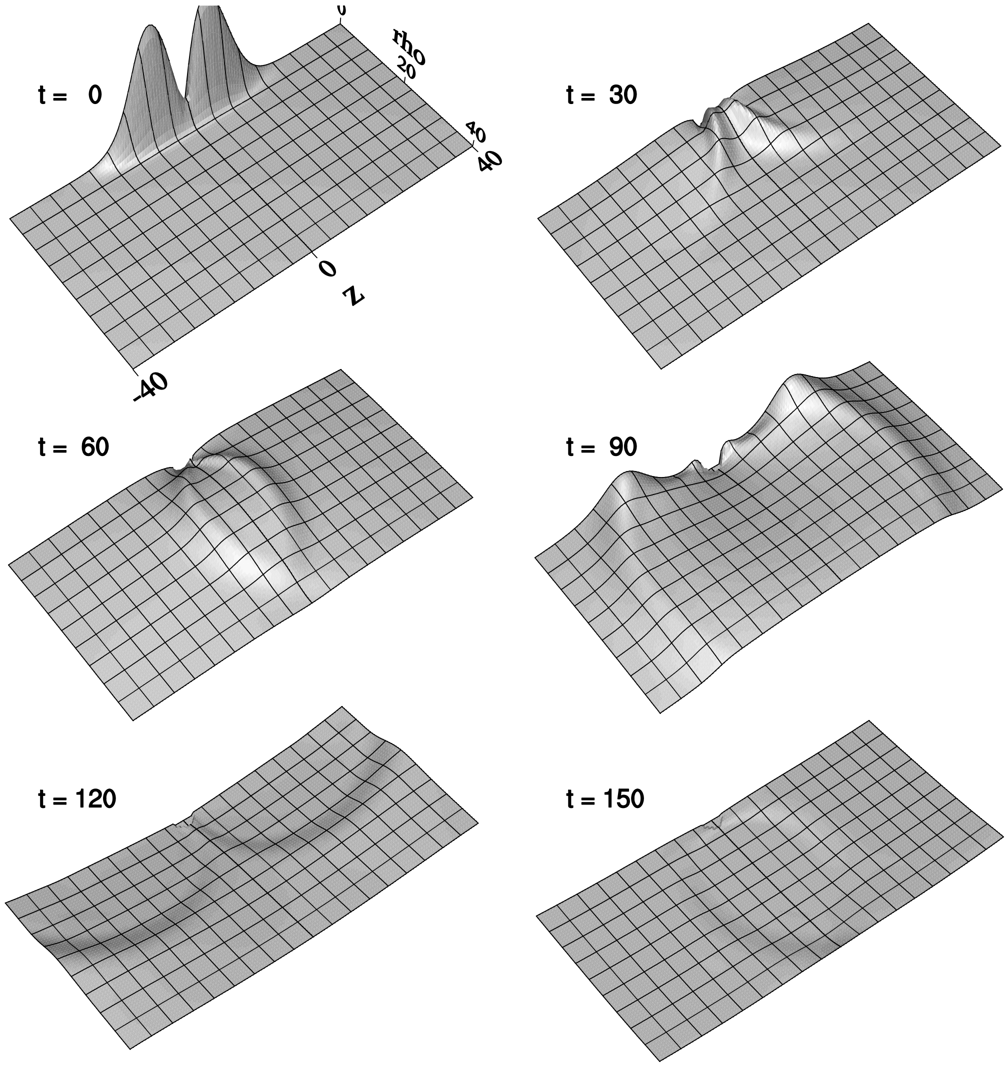


Figure 4.1: Evolution of Φ during prolate scalar field collapse. Shown in the figure is $r\Phi$, where $r = \sqrt{\rho^2 + z^2}$, for several times early on during the evolution of the 257x513, $\rho_{\max} = 40$ case. The height of the surface plot represents the magnitude of the function, and the gridlines are drawn for reference only, and do not reflect the actual mesh structure.

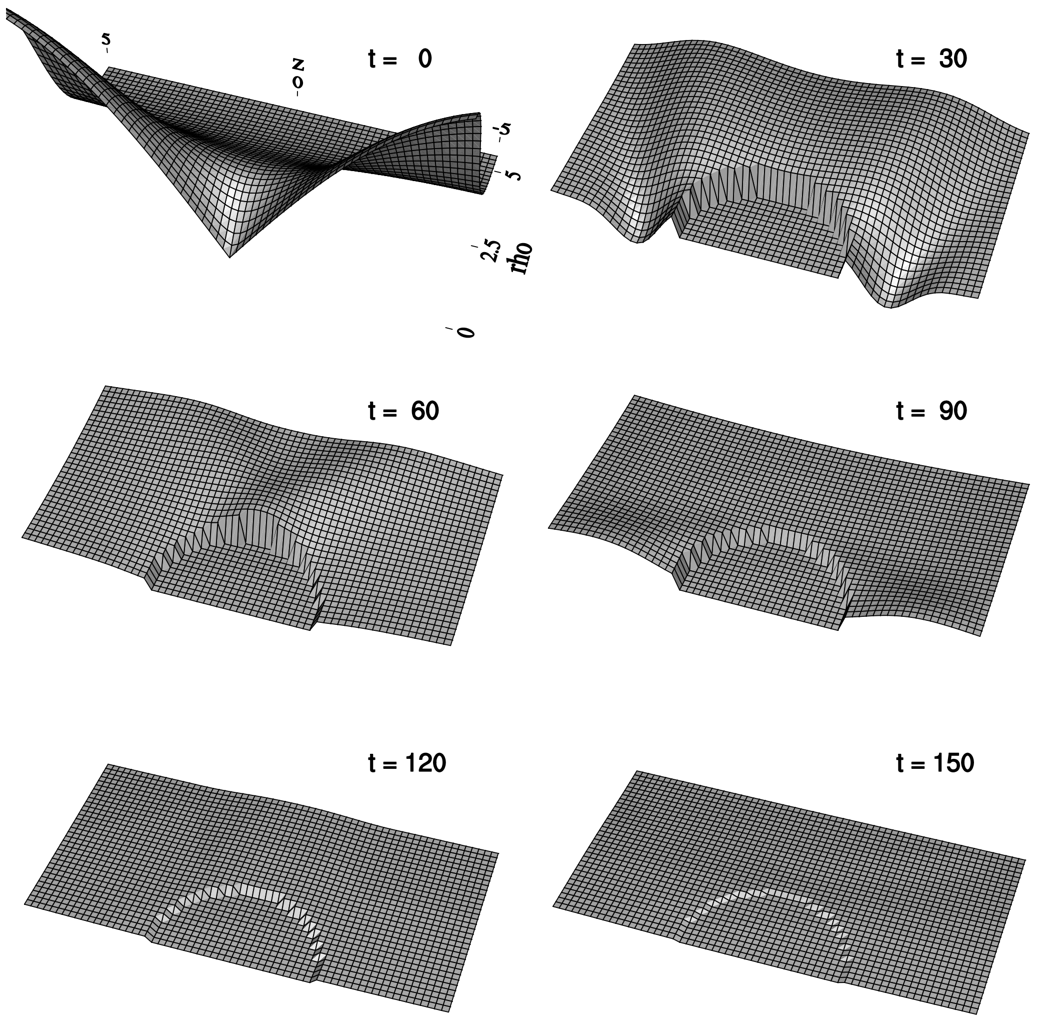


Figure 4.2: The same information is shown here as in Figure 4.1, except we have truncated the grid outside of the coordinate range $\rho = 0, .., 5$ and $z = -5..5$, to better show the excised region. Here, the mesh lines *do* correspond to the actual grid lines. Φ in the excised region is set to zero.

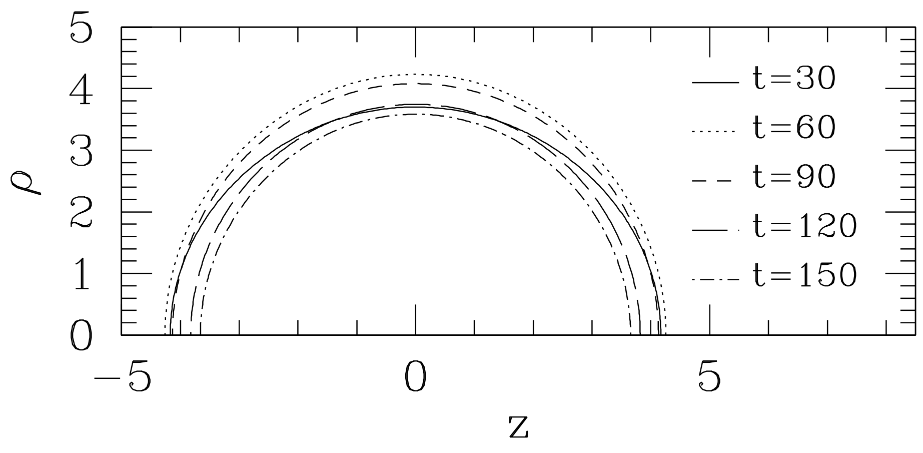


Figure 4.3: The shapes of the apparent horizon, from the same simulation and at the same times as the plots in Figures 4.2 and 4.1.

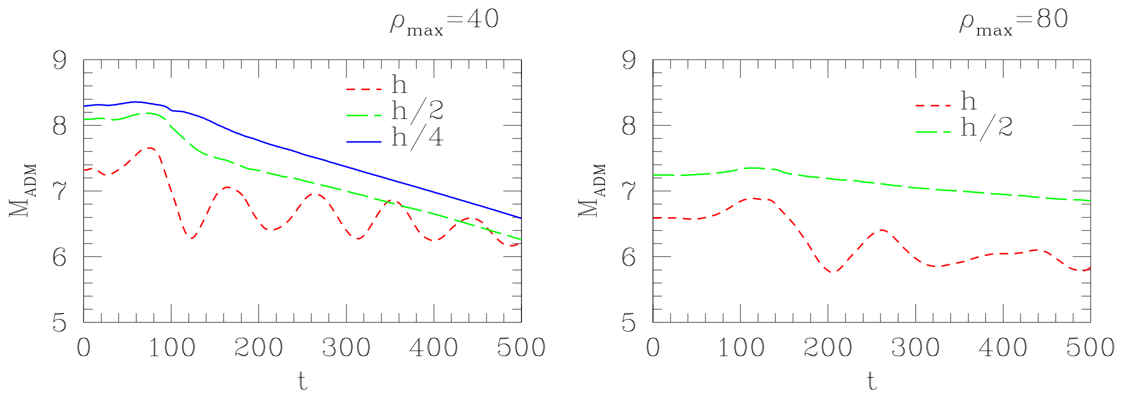


Figure 4.4: The ADM mass (2.67) as a function of time, from the prolate scalar field collapse simulations.

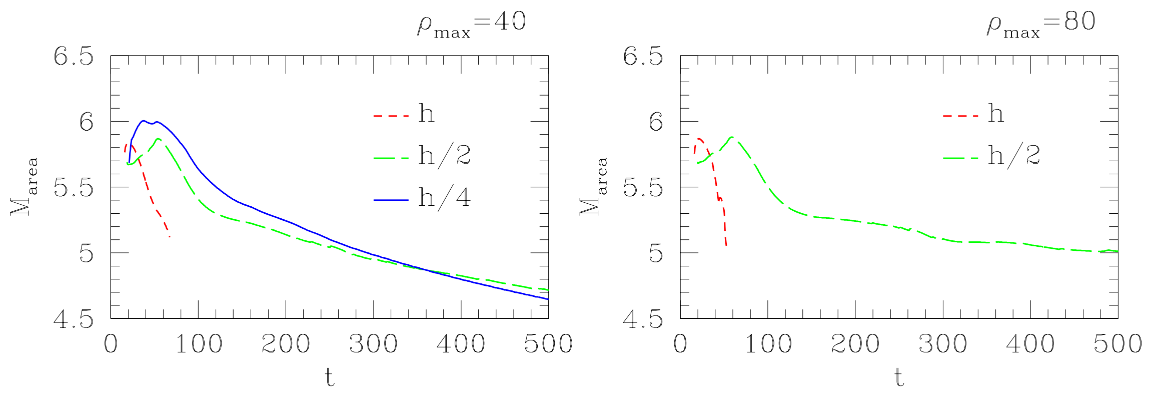


Figure 4.5: The area mass estimate (4.1) as a function of time, from for the prolate scalar field collapse simulations. The area mass can only be calculated if an apparent horizon is found; at early times there is no apparent horizon, and we fail to find an apparent horizon at late times in the lowest resolution simulation.

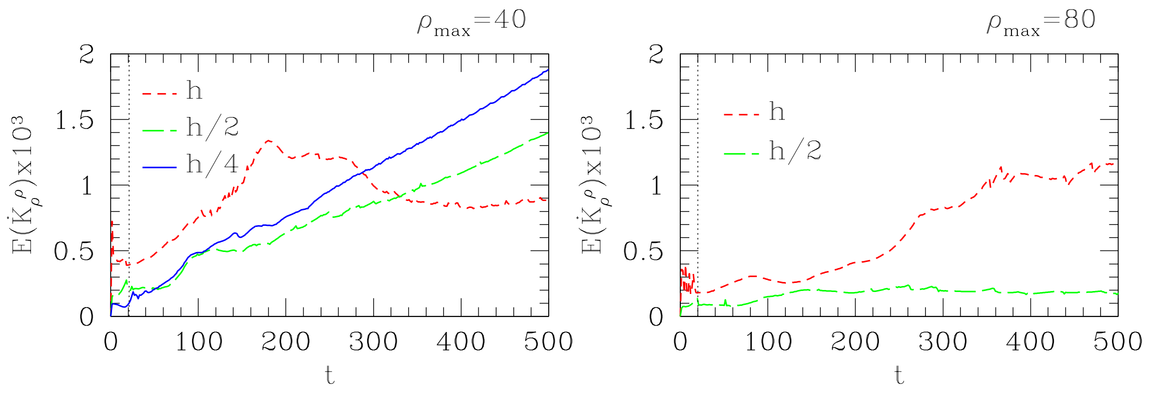


Figure 4.6: $E(\dot{K}_\rho{}^\rho)$—the ℓ_2 norm of the residual of the evolution equation for $K_\rho{}^\rho$ (2.66)—from the prolate scalar field collapse simulations. The time when excision begins is denoted by the vertical dashed line for the $h/4$ and $h/2$ runs in the $\rho_{\max} = 40$ and $\rho_{\max} = 80$ cases respectively (the corresponding times for the other resolution runs are very similar, and so are not shown to avoid clutter in the figure).

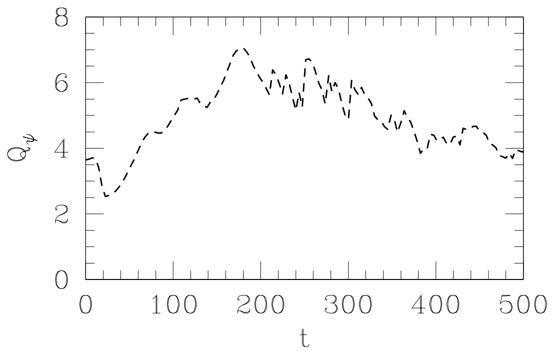


Figure 4.7: The convergence factor Q_ψ (1.55) for ψ, from the $\rho_{\max} = 40$ simulations.

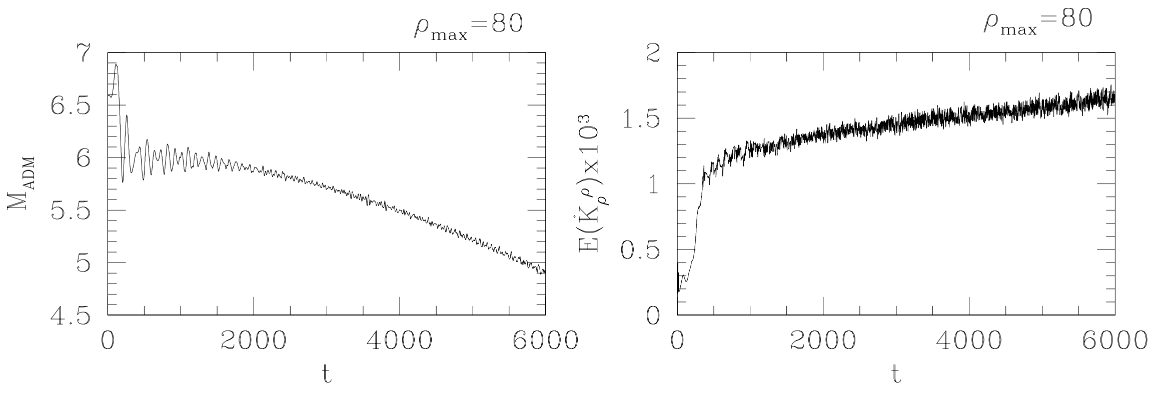


Figure 4.8: These plots demonstrates the "long term" stability of our excision scheme. The figure on the left shows the ADM mass, while the figure on the right shows the error norm $E(\dot{K}_\rho{}^\rho)$, for the 129x257, $\rho_{\max} = 80$ case. The code will not run forever with excision, however, with a judicious choice of excision parameters, sufficient resolution, and the outer boundaries far enough away, we should be able to evolve for a sufficient length of time to study most axisymmetric phenomena.

by using compact initial distributions of energy, most of the scalar field will fall into the black holes, and the remnant field will have little gravitational influence on the evolution.

One of the potentially useful results from our study of black hole collisions is the use of boundary conditions on the excision surface to manipulate the spacetime slices, and hence achieve desired coordinate velocities of the holes during evolution. This is useful for several reasons, perhaps the most significant being that by forcing the holes to move towards each other rapidly in coordinate space, the shape of the encompassing apparent horizon (and hence excision surface) when the holes do merge turns out to be less distorted than otherwise, improving the stability of the code and reducing the resolution required to resolve the merged black hole. As will be discussed in more detail in Section 4.2.2 below, we control the motion of a black hole by displacing β^z on the excision surface by some amount. The particular form that β^z takes there is not important, and therefore we believe that we will still be able to control black hole motion in our coordinate system once we restrict the class of boundary conditions to obtain consistent evolution.

4.2.1 Boosted Scalar Field Initial Conditions

To obtain initial conditions for a black hole collision from scalar field collapse that mimic black holes moving along the z-axis with given velocities, we specify the initial scalar field profile to be that of a Lorentz boosted, *flat space* evolution. A Lorentz transformation along the z-axis takes the form

$$\begin{aligned} z &= \gamma[z' - vt'] \\ t &= \gamma[t' - vz'], \end{aligned} \tag{4.2}$$

where $\gamma = 1/\sqrt{1-v^2}$, (ρ', z', t') are the coordinates relative to the "rest-frame" of the initial pulse (i.e., the frame where the center of mass of the pulse is stationary), (ρ, z, t) are the coordinates of the simulation, and v is the velocity of the boost. Therefore, we specify $\Phi(\rho', z', t' = 0)$ via (2.52), and $\Pi_\Phi(\rho', z', t' = 0)$ as described in Section 2.2.3 to obtained ingoing, out-going or time-symmetric initial conditions from the perspective of (ρ', z', t'). Then, we evolve the scalar field on a Minkowski background, as far forwards and backwards in time t' as necessary to use (4.2) to obtain the desired initial conditions for $\Phi(\rho, z, t = 0)$ and $\Pi_\Phi(\rho, z, t = 0)$ on the entire (ρ, z) computational grid. This procedure is repeated separately, with different values of v, for each of the two pulses that will collapse to form the two black holes in the full evolution. We then translate each of the resultant profiles by a given distance along the z-axis, to achieve a desired initial separation of the pulses, before adding them to give the final $t = 0$ shapes for Φ and Π_Φ.

We use a flat-space evolution in the above described procedure for two reasons. First, we would not be able to apply the transformation (4.2) to a solution obtained with our code in the general relativistic case, for (4.2) will not give initial conditions consistent with our coordinate and slicing conditions. Second, we would not be able to add boosted solutions of the full evolution of single pulses, as GR is not a linear theory (though, we could add the resultant scalar field solutions, and use those as initial conditions, as is done with the flat-space generated data). Regardless, at this stage in the development of the excision code we are not too concerned about the precise form of the initial data, as long as there is some linear momentum (if desired) in each pulse, relative to a distant observer.

4.2.2 Excision Boundary Conditions

The idea behind applying appropriate excision boundary conditions to β^z, to modify the coordinate velocity of a given black hole, is rather simple, and is illustrated in Figure 4.9 below. In the figure, we show schematically what the axis of a maximal slice of a single, spherically symmetric, origin centered black hole spacetime might look like. For recall from the discussion in Section 1.5.2, that the singularity avoidance property of maximal slicing manifests as a "freezing" of the slice as the singularity is approach, i.e. $\alpha \to 0$ there, hence the evolution of the spacetime slice slows down (as measured by t) in the vicinity of the hole. Consequently, the hypersurface normal vector n^a on the excision surface will point towards the black hole.

If the black hole were to remain at the same coordinate location during evolution, β^z would need to be an odd function in z, approaching some positive number on the "upper" (towards $z_{\max}$) side of the excision surface, and minus the same number on the lower side of the excision surface. Then, to cause the black hole to move on the coordinate grid during evolution, we can simply displace the profile of β^z by some constant amount along the excision boundary. As illustrated in the figure, adding a positive number to β^z their will cause the black hole to move in the negative z direction, and vice-versa.

During a merger simulation, we can therefore speed-up the merger in coordinate time by shifting the profile of β^z about the upper excision surface by a positive number, and that of β^z about the lower surface by a negative number. We determine the amounts to shift each profile by, in a given simulation, by trial-and-error. Note that we do not instantaneously modify β^z to the desired shape. Instead, at the time of excision, we replace the profile along the excision boundary with a linear function in z, that has the same values on the maximum and minimum z points of the boundary as before excision. Then, we slowly displace this line with time until the desired shape is reached. As the two excised regions get close to one another, the linear profiles on both boundaries are adjusted so that β^z approaches a common value at the place where the boundaries will come into contact (though often a single, encompassing apparent horizon is found before then, and the two excised zones are replaced by a single one). An example of this is given in the following section (see Figure 4.14).

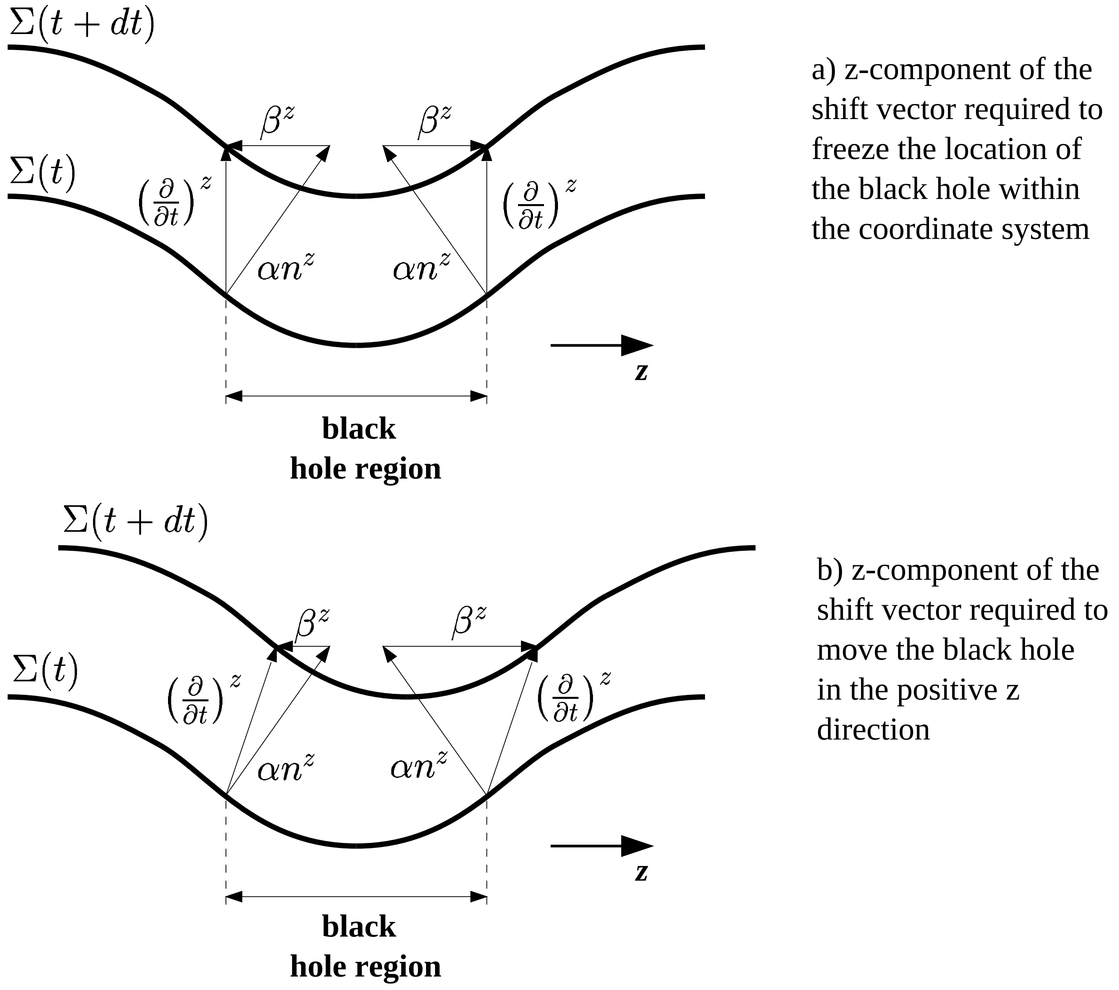
4.2.3 A Sample Merger

Here, we show several figures from a time-symmetric, head-on, equal mass black hole collision simulation. At t=0, Φ is initialized as the sum of two profiles of the form (2.52), with $A = 0.45$, $\Delta = 3$, $(\rho_0, z_0) = (0, \pm 20)$ and $\epsilon = 1$; $\Pi_\Phi = \bar{\Omega} = \bar{\sigma} = 0$ then. The outer boundary is at $\rho_{\max} = z_{\max} = -z_{\min} = 80$, and the size of the mesh is 257x513. We use frozen boundary conditions for α and β^ρ after excision, and the "shifted-β^z" boundary conditions for β^z, as described in the previous section.

Figure 4.10 contains several snapshots of ψ during the evolution (which was stopped at $t \approx 1200$ to prevent the local hard-disk from filling—by then over 600MB of data had been saved per grid function). Figure 4.11 contains plots of the ADM and area mass estimates. The area mass can only be computed when an apparent horizon is found—at intermediate times we lose track of the AH (due to poor resolution), so then we extrapolate the motion of each hole based on prior velocities. Even after the merger, the AH-finder fails to find the AH on occasion, hence the "jittery" behavior of the area mass function then. We suspect that the increased jaggedness of the ADM mass after a merger is due, in part, to reflections of gravitational waves off the outer boundary.

Figure 4.12 shows $E(\dot{K}_\rho{}^\rho)$ (2.66) for the merger simulation, plus, for comparison, the same function from a simulation started with identical initial data, though without excision (this simulation crashed at around $t = 500$). Ideally, we would like to run the same simulations with excision at different resolutions, to be able to do convergence testing; however, we do not have the computer resources to run at higher resolutions, and at lower resolutions the AH-finder has trouble finding the apparent horizons.

Figure 4.13 shows the gravitational waves emitted during the merger, as measured by the Newman-Penrose scalar Ψ_4 (1.22) at different locations along the equatorial plane. The initial burst has features that *roughly* agree with similar results obtained in other simulations (see for instance [79]), in particular the amplitude of the pulse, and its wavelength, which is expected to be on the order of $32M$ [42]; though, as can be seen from Figure 4.11, we have a tremendous uncertainty in what to use for M, and our outer boundary is about 5-10 times closer than that used in [79]. After the initial burst, our result is rather "noisy". A significant part of this noise is



$\Sigma(t+dt)$
$\Sigma(t)$
β^z
β^z
$\left(\frac{\partial}{\partial t}\right)^z$
$\left(\frac{\partial}{\partial t}\right)^z$
αn^z
αn^z
z
black hole region
a) z-component of the shift vector required to freeze the location of the black hole within the coordinate system
$\Sigma(t+dt)$
$\Sigma(t)$
β^z
β^z
$\left(\frac{\partial}{\partial t}\right)^z$
$\left(\frac{\partial}{\partial t}\right)^z$
αn^z
αn^z
z
black hole region
b) z-component of the shift vector required to move the black hole in the positive z direction

Figure 4.9: A schematic representation of a maximal slicing of a black hole spacetime. In the upper figure, we show the values that β^z would need to have, on either side of the excised region, in order for the black hole to remain at a fixed coordinate location in the grid as we evolve. Similarly, the lower figure shows the values that β^z would need to have for the black hole to move to the right, within the coordinate system.

due to reflections from the outer boundary (we can directly see the reflections by looking at the evolution of Ψ_4 over the entire domain).

Finally, to demonstrate the effectiveness of the shifted-β^z boundary conditions we used in this simulation, in Figure 4.14 we show β^z along the axis for this case, and for comparison, β^z obtained in a simulation with identical initial conditions, though using frozen-β^z boundary conditions. In Figure 4.15 we show the AH shapes at different times for these two simulations.

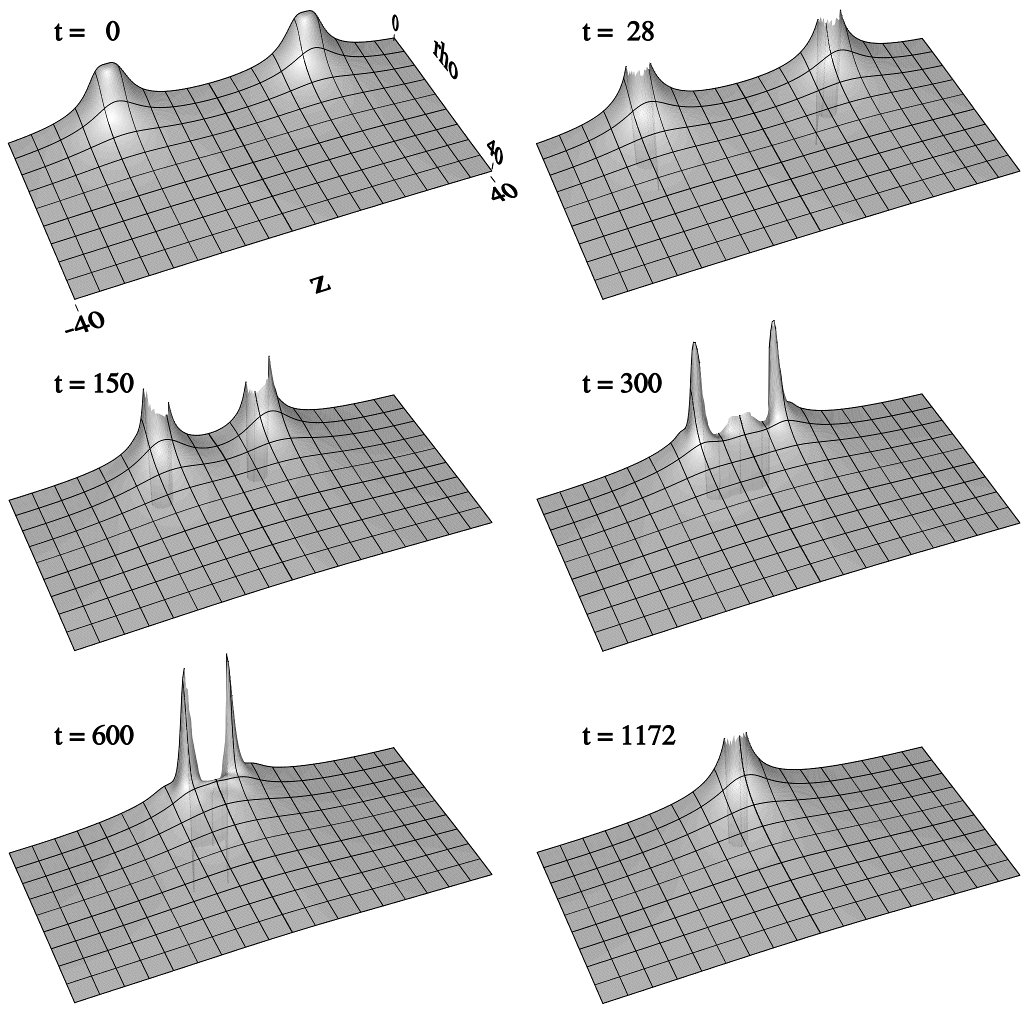
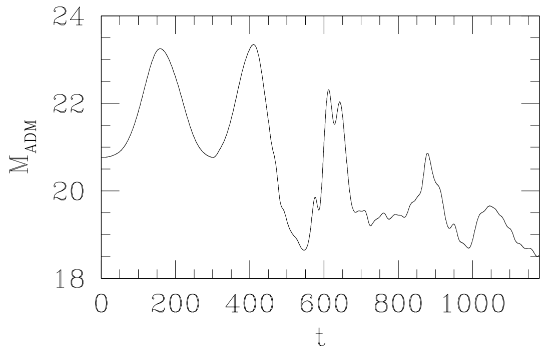


Figure 4.10: ψ at several times during the merger simulation. Note that in the figure we have restricted the domain to a [0,40,-40,40] region, to better show the excised region. Also, the mesh lines are for visualization purposes only, and do not reflect the underlying computational grid.



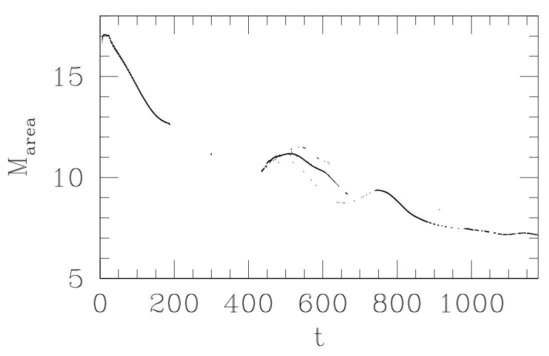


Figure 4.11: The figure to the left shows the ADM mass estimate (2.67) of the merger spacetime, while the figure to the right shows the area mass estimate (4.1). We did not draw the area mass estimate with a connected line, for at times we lose track of the apparent horizon, due to poor resolution.

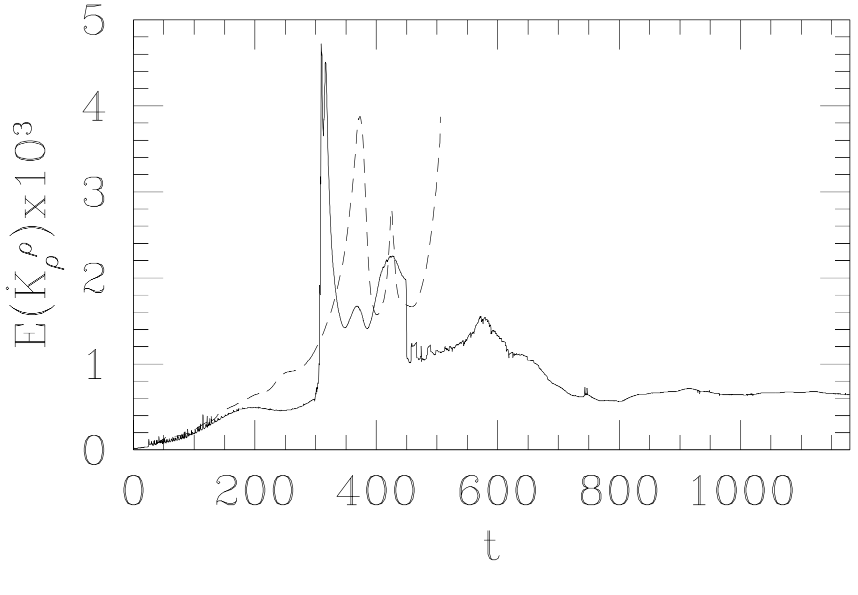


Figure 4.12: $E(\dot{K}_\rho{}^\rho)$—the ℓ_2 norm of the residual of the evolution equation for $K_\rho{}^\rho$ (2.66)—from the black hole merger simulation (solid line). Without additional runs at different resolutions, such a plot is not too meaningful; however, to offer some comparison, we plot the same function, obtained from a simulation with the exact same initial data, though *without excision* (dashed-line). This second simulation crashed at around $t = 500$ (the multigrid solver failed, probably because gradients in ψ were becoming too steep), hence the data ends there.

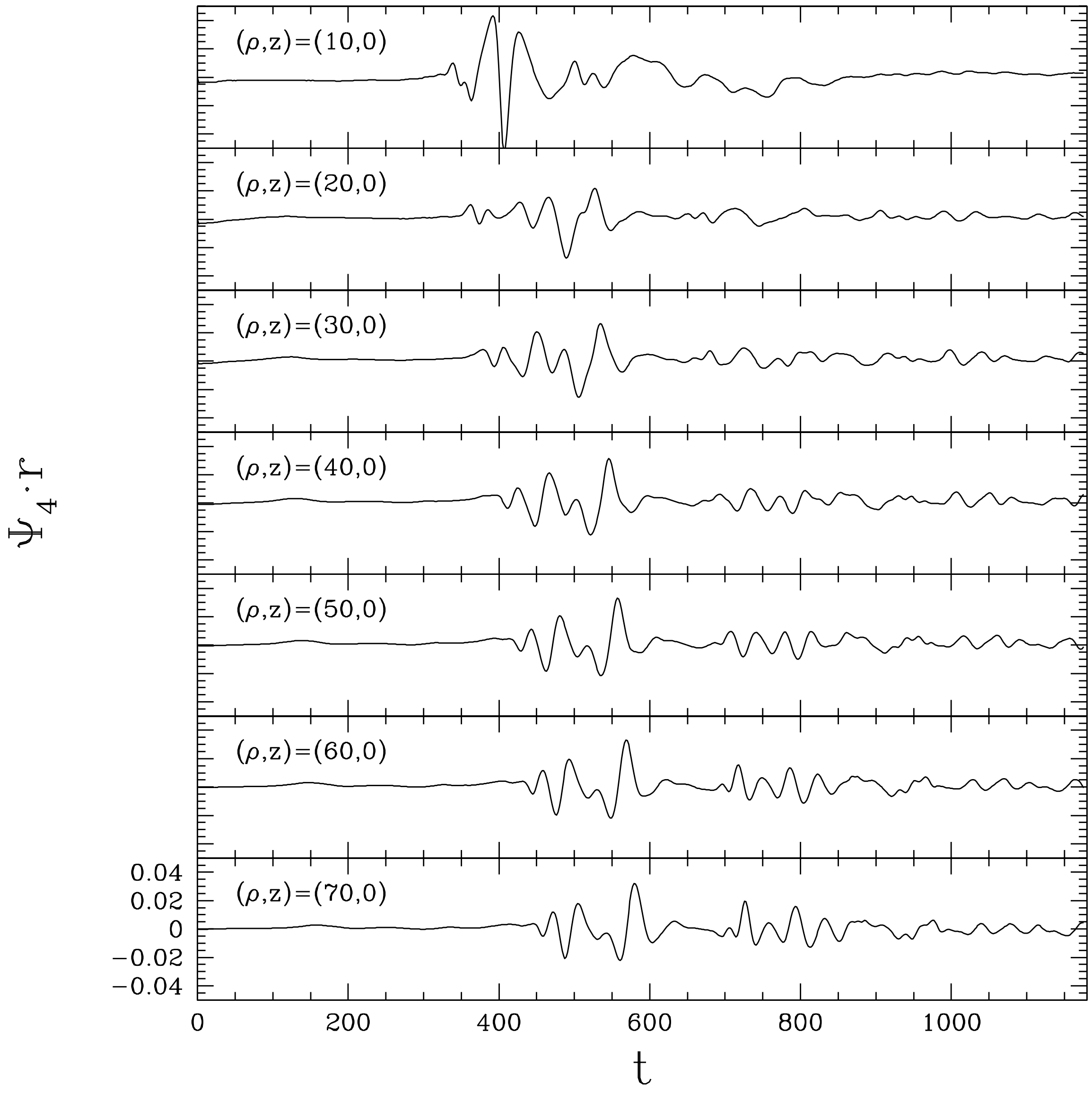


Figure 4.13: The Newman-Penrose scalar Ψ_4 (1.22) multiplied by $r = \sqrt{\rho^2 + z^2}$, from the black hole merger simulation, measured at several different locations along the equatorial plane ($z = 0$). The initial burst of energy coming from the merger does *approximately* have the expected wave-length and magnitude; however, it is difficult to judge which of the features are "real", and which come from outer-boundary reflections (the dominant contribution to the "noise", we suspect), solution error, or inconsistent boundary conditions.

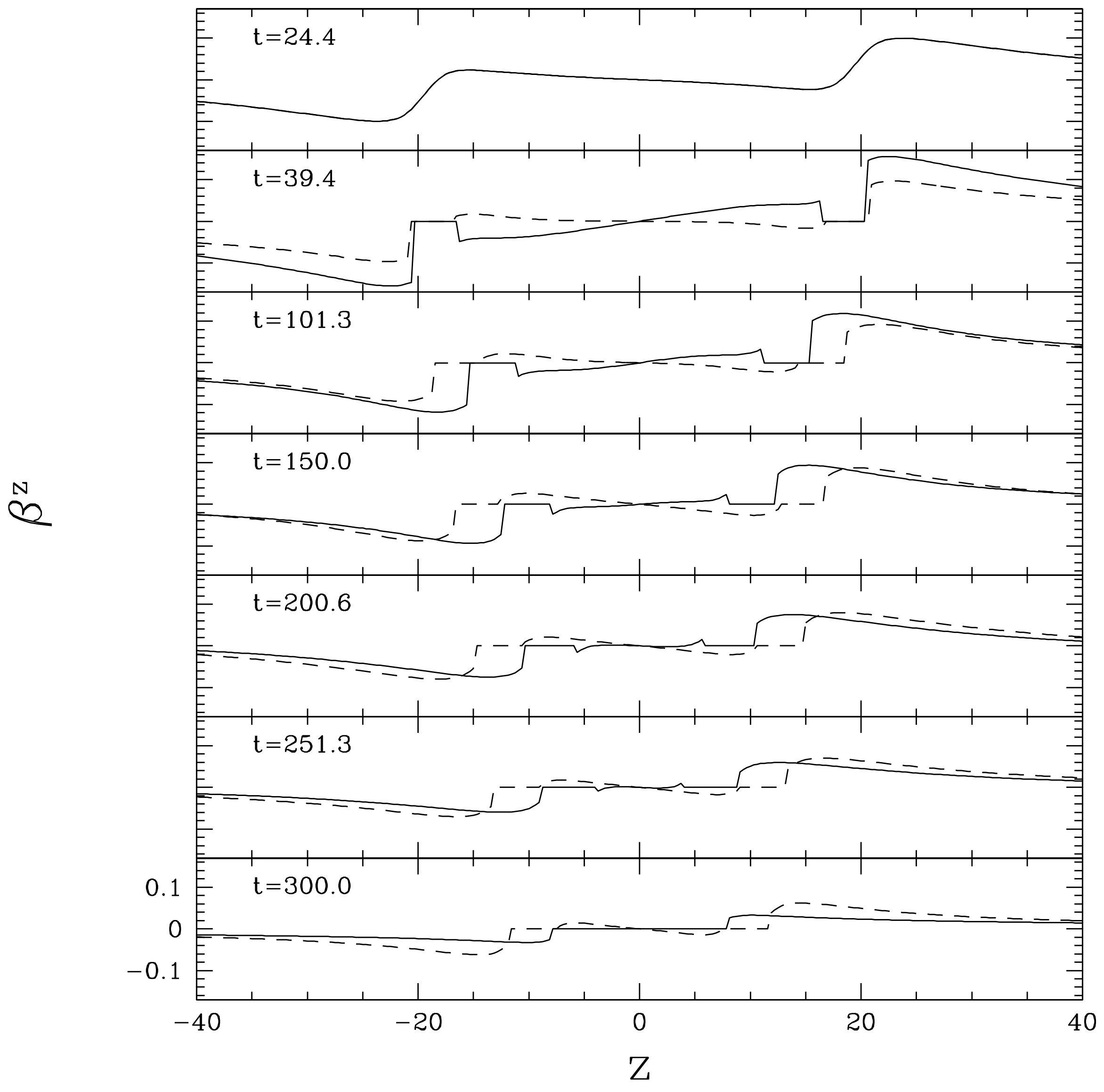


Figure 4.14: This sequence of images demonstrates the boundary conditions we place on β^z, to force the black holes to move together relatively rapidly in coordinate space. The solid curves show β^z along the axis, in the z range $[-40, 40]$, from the merger simulation using the shifted-β^z coordinate condition (see Section 4.2.2). For comparison, the dashed curves show β^z from a simulation with identical initial data, though using a boundary condition where the profile of β^z is frozen-in at the time of first excision (just after $t = 24.4$, shown in the upper-most plot). β^z is set to zero in the excised region. For the shifted-β^z condition, we slowly displace β^z until a specified maximum displacement is reached, at $t = 39.4$. This causes the two holes to move together more rapidly in coordinate space, until we obtain a merger (at $t = 300$ in the shifted-β^z case). See Figure 4.15 for the observed AH shapes, from these two simulations.

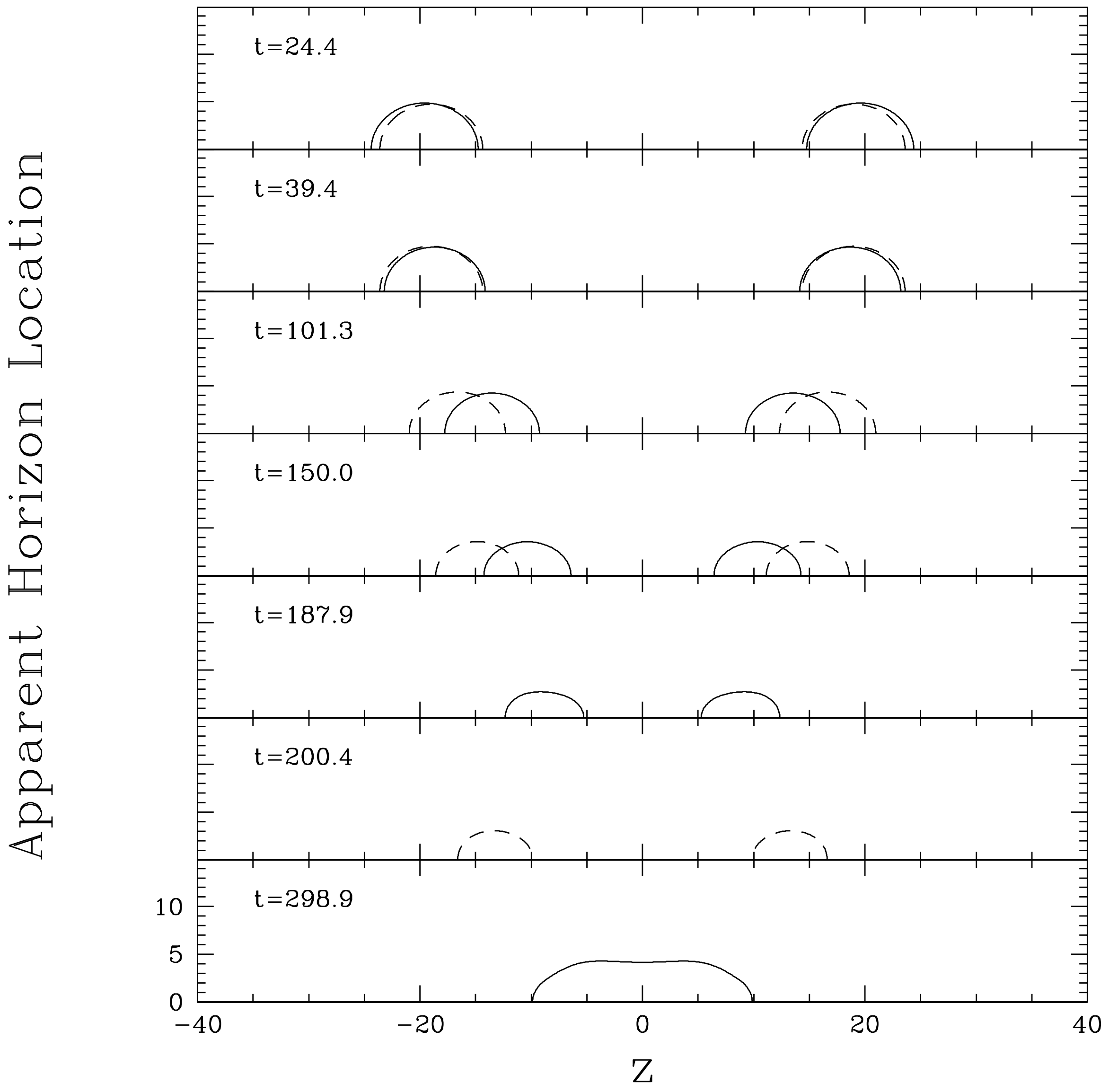


Figure 4.15: Apparent horizon shapes from the two simulations discussed in the caption of Figure 4.14. The AHs drawn with a solid curve correspond to the simulation with the shifted-β^z boundary conditions, while those drawn with the dashed curves correspond to the frozen-β^z boundary conditions. In the shifted-β^z simulation, we lose track of the AH at $t = 187.9$, before finding it again at $t = 298.9$; while in the other simulation we lose track of the AH shortly after $t = 200$.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

The principal new results presented in thesis stem from a numerical study of gravitational collapse of axisymmetric distributions of scalar field energy. We find that critical behavior at the threshold of black hole formation is still evident in axisymmetry. However, in the presence of certain asymmetries, we find some evidence that the nature of the critical solution *may* differs from the spherically symmetric one in the following manner. Near threshold, the scalar field initially approaches a solution that approximates the discretely self-similar, spherically symmetric solution. At later times during the evolution, small asymmetries in the initial data eventually grow to the point where the self-similar region *bifurcates* into two new, locally self-similar solutions, separated by some distance along the axis of symmetry. If this is an instability of the spherical critical solution, then we hypothesize that the bifurcation process would repeat indefinitely, as one continues to tune to the threshold of black hole formation. However, we have not been able to rule out the possibility that the bifurcation is due to a focusing effect: as the initial, prolate distribution collapses to the origin, asymmetries cause the scalar field energy to focus to two distinct centers (in principle, one could construct initial data where there are an arbitrary number of these foci, though we have not yet studied such cases). Additional work will be needed to confirm these speculations. In particular, we will need to tune closer to criticality than what we have done so far. This may require that we parallelize the code, to have access to more memory and shorter run times; or perhaps we could construct a family of initial data that demonstrates bifurcation behavior without forcing excessively elongated grids in the z direction, which would speed up the run time considerably.

Adaptive Mesh Refinement was *essential* to study critical behavior, and as mentioned in Chapter 4, AMR will be necessary to explore black hole collisions, and the interactions of black holes with matter and gravitational waves. To this end, we need to incorporate black hole excision into the AMR-based code. As shown in Chapter 4, our experiments with excision in the unigrid code were reasonably successful, although more work needs to be done vis-a-vis inner boundary conditions that are fully consistent with all of Einstein's field equations.

Finally, we mention some of the additional directions in which we want to extend the code in the near future. We need to improve the robustness of the multigrid solver to the point where it is feasible to simulate critical collapse of gravitational waves. As mentioned in Section 2.3.8, we may be able to cure the instability problems by judicious manipulation of the source functions within the differential operators. However, it may be that a new system of variables will be needed—early experiments performed by Hirschmann with alternative variables have given encouraging results [115]. We would also like to incorporate additional matter fields, including fluids, electromagnetism, and the complex scalar field discussed in Section 2.1.2, in particular to be able to include the effects of angular momentum in our studies. This will require that we implement the full 2+1+1 formalism in the code.

BIBLIOGRAPHY

[1] S.W. Hawking and G.F.R. Ellis, *The Large Scale Structure of Space-Time*, Cambridge, Cambridge University Press 1973

[2] J. Kormendy and K. Gebhardt, *Supermassive Black Holes in Nuclei of Galaxies* , astro-ph/0105230

[3] D. Arnett *Supernova and Nucleosynthesis*, Princeton NJ, Princeton University Press 1996

[4] M.W. Choptuik, "Universality And Scaling in Gravitational Collapse Of a Massless Scalar Field", *Phys. Rev. Lett.* **70**, 9 (1993)

[5] A. Abramovici et al., "LIGO: The Laser Interferometer Gravitational Wave Observatory", *Science* **256**, 325 (1992)

[6] C. Bradaschia et al., "First Results on the Electronic Cooling of the Pisa Seismic Noise Super-attenuator For Gravitational Wave Detection", *Phys. Lett.* **A137**, 329 (1989)

[7] K. Danzmann and the GEO Team, *Lecture Notes in Physics* **410**, 184 (1992)

[8] K. Tsubomo, M.K. Fujimoto and K. Kuroda, in *Proceedings of the TAMA International Workshop on Gravitational Wave Detection*, Tokio, Universal Academic Press (1996)

[9] see http://www.gravity.pd.uwa.edu.au/AIGO/AIGO.html

[10] R.S. Hamade and J.M. Stewart, "The Spherically Symmetric Collapse of a Massless Scalar Field", *Class. Quant. Grav.* **13**, 497 (1996)

[11] R.S. Hamade, J.H. Horne and J.M. Stewart, "Continuous Self-Similarity and S-Duality", *Class. Quant. Grav.* **13**, 2241 (1996)

[12] B. Brügmann, "Adaptive Mesh and Geodesically Sliced Schwarzschild Spacetime in 3+1 Dimensions", *Phys. Rev.* **D 54**, 7361 (1996)

[13] M.W. Choptuik, T. Chmaj and P. Bizon, "Critical Behaviour in Gravitational Collapse of a Yang-Mills Field", *Phys. Rev. Lett* **77**, 424 (1996)

[14] S.L. Liebling and M.W. Choptuik "Black Hole Criticality in the Brans-Dicke Model", *Phys. Rev. Lett* **77**, 1424 (1996)

[15] M.W. Choptuik, E.W. Hirschmann and S.L. Liebling, "Instability of an 'Approximate Black Hole' ", *Phys. Rev.* **D 55**, 6014 (1997)

[16] P. Papadopoulos, E. Seidel and L. Wild, "Adaptive Computation of Gravitational Waves from Black Hole Interactions", *Phys. Rev.* **D 58**, 084002 (1998)

[17] P. Diener, N. Jansen, A. Khokhlov and I. Novikov, "Adaptive Mesh Refinement Approach to Construction of Initial Data for Black Hole Collisions", *Class. Quant. Grav.* **17**, 435 (2000)

[18] K.C.B. New, D. Choi, J.M. Centrella, P. MacNeice, M.F. Huq and K. Olson, "Three-dimensional Adaptive Evolution of Gravitational Waves in Numerical Relativity", *Phys. Rev.* **D62**, 084039 (2000)

[19] S. Hern, “Numerical Relativity and Inhomogeneous Cosmologies”, PhD dissertation, Cambridge (1999), gr-qc/0004036

[20] S.L. Liebling, “The Singularity Threshold of the Nonlinear Sigma Model Using 3D Adaptive Mesh Refinement”, gr-qc/0202093

[21] K.Maeda, M.Sasaki, T.Nakamura and S.Miyama, “A New Formalism of the Einstein Equations for Relativistic Rotating Systems”, *Prog. Theor. Phys.* **63**, 719 (1980)

[22] R. Arnowitt, S. Deser and C.W. Misner, in *Gravitation: An Introduction to Current Research*, ed. L. Witten, New York, Wiley (1962)

[23] R. Geroch, “A Method for Generating Solutions of Einstein’s Equations”, *Jour. Math. Phys.* **Vol. 12**, **No. 6**, 918 (1971)

[24] P.C. Argyres, in *Contributions from the G1 Working Group at the APS Summer Study on Particle and Nuclear Astrophysics and Cosmology in the Next Millennium*, Snowmass, Colorado, June 29 - July 14 (1994), astro-ph/9412046

[25] T. Koike, T. Hara, and S. Adachi, “Critical Behavior in Gravitational Collapse of Radiation Fluid: A Renormalization Group (Linear Perturbation) Analysis”, *Phys. Rev. Lett.* **74**, 5170 (1995)

[26] C. Gundlach, “Understanding Critical Collapse of a Scalar Field”, *Phys.Rev.* **D55**, 695 (1997)

[27] J. M. Martin-Garcia and C. Gundlach, “All Nonspherical Perturbations of the Choptuik Space-Time Decay”, *Phys.Rev.* **D59**, 064031 (1999)

[28] The notion of black hole excision was first expounded by Unruh (1984)—see J. Thornburg, “Coordinates and boundary conditions for the general relativistic initial data problem”, *Class. Quantum Grav.* **4**, 1119 (1987)

[29] R.M. Wald, *General Relativity*, Chicago IL, University of Chicago Press 1984

[30] B.K. Berger, “Numerical Approaches to Spacetime Singularities”, *Living Rev. Rel. 2002-1*, gr-qc/0201056

[31] V.A. Belinskii, J.M. Khalatnikov and E.M. Lifshitz, “Oscillatory Approach to a Singular Point in the Relativistic Cosmology”, *Advan. in Phys.* **19**, 525 (1970)

[32] R. Penrose, “Gravitational Collapse: The Role of General Relativity”, *Revistas del Nuovo Cimento* **1** 252 (1969)

[33] R.M. Wald, *Gravitational Collapse and Cosmic Censorship*, to appear in “The Black Hole Trail”, ed. by B. Iyer., gr-qc/9710068

[34] T. Harada, H. Iguchi and K. Nakao, “Physical Processes in Naked Singularity Formation”, *Prog. Theor. Phys.* **107** 449 (2002)

[35] C. Gundlach, “Critical phenomena in gravitational collapse”, *Living Rev.Rel. 2 1999-4*, gr-qc/0001046

[36] A. Wang, “Critical Phenomena in Gravitational Collapse: The Studies So Far”, *Braz.J.Phys.* **31**, 188 (2001)

[37] S. Hod and T. Piran, “Fine Structure of Choptuik’s Mass Scaling Relation”, *Phys. Rev.* **D 55**, 440 (1997)

[38] R.A. Hulse and J.H. Taylor "Discovery of a Pulsar in a Binary System", *Astrophys. J.* **195**, L51 (1975)

[39] E. Newman and R. Penrose, "An Approach to Gravitational Radiation by a Method of Spin Coefficients", *J. Math. Phys* **3**, 566 (1962); **erratum 4**, 998

[40] R. Penrose *Proc. Roy. Soc. Lond.* "Zero Rest-Mass Fields Including Gravitational—Asymptotic Behavior", **A284**, 159 (1965)

[41] S.A. Teukolsky, "Perturbations of a Rotating Black Hole. 1. Fundamental Equations for Gravitational Electromagnetic, and Neutrino Field Perturbations", *Astrophys. J.* **185**, 635 (1973)

[42] L. Smarr, in *Sources of Gravitational Radiation*, ed. L. Smarr, Seattle, Cambridge University Press (1978)

[43] C.W. Misner, K.S. Thorne and J.A. Wheeler, *Gravitation*, New York, W.H. Freeman and Company (1973)

[44] J.W. York, Jr., in *Sources of Gravitational Radiation*, ed. L. Smarr, Seattle, Cambridge University Press (1979)

[45] L. Smarr and J.W. York, Jr., "Kinematical Conditions in the Construction of Spacetime", *Phys. Rev.* **D17**, 2529 (1978)

[46] P.R. Brady, J.D.E. Creighton and K.S. Thorne, "Computing the Merger of Black-Hole Binaries: the IBBH Problem", *Phys. Rev.* **D58**, 061501 (1988)

[47] M. Shibata, "Fully General Relativistic Simulation of Merging Binary Clusters – Spatial Gauge Condition –", *Prog. Theor. Phys.* **101**, 1199 (1999)

[48] L. Lehner, "Numerical Relativity: A Review", *Class. Quant. Grav.* **18**, R25 (2001)

[49] C. Bona and J. Masso, "Harmonic Synchronizations of Space-Time", *Phys.Rev.* **D38**, 2419 (1988)

[50] M. Alcubierre, "The Appearance of Coordinate Shocks in Hyperbolic Formalisms of General Relativity", *Phys.Rev.* **D55**, 5981 (1997)

[51] D. Garfinkle, "Harmonic Coordinate Method for Simulating Generic Singularities", *Phys.Rev.* **D65**, 044029 (2002)

[52] A.M. Abrahams and C.R. Evans, "Gauge Invariant Treatment of Gravitational Radiation Near the Source: Analysis and Numerical Simulations", *Phys. Rev.* **D42**, 2585 (1990).

[53] E. Seidel and W. Suen, "Towards a singularity-proof scheme in numerical relativity", *Phys. Rev. Lett.* **69**,1845 (1992)

[54] P. Anninos, J. Massó, E. Seidel, W. Suen and J. Towns, "Three-dimensional Numerical Relativity: The Evolution of Black Holes", *Phys. Rev.* **D52**, 2059 (1995)

[55] R.L. Marsa and M.W. Choptuik, "Black Hole–Scalar Field Interactions in Spherical Symmetry", *Phys. Rev.* **D54** 4929 (1996)

[56] R. Gómez, "Stable Characteristic Evolution of Generic Three-Dimensional Single-Black-Hole Spacetimes", *Phys. Rev. Lett.* **80**, 3915 (1998).

[57] G.B. Cook et al, "Boosted Three-Dimensional Black-Hole Evolutions with Singularity Excision", *Phys. Rev. Lett.* **80**, 2512 (1998).

[58] M.W. Choptuik, E.W. Hirschmann and R.L. Marsa, "New Critical Behavior in Einstein-Yang-Mills Collapse", *Phys. Rev.* **D60** 124011 (1999)

[59] S. Brandt et al, "Grazing Collisions of Black Holes via the Excision of Singularities", *Phys. Rev. Lett.* **85**, 5496 (2000).

[60] F. Pretorius and M.W. Choptuik, "Gravitational collapse in 2+1 dimensional AdS spacetime", *Phys. Rev.* **D62**, 124012 (2000)

[61] M. Alcubierre and B. Brugmann, "Simple Excision of a Black Hole in 3+1 Numerical Relativity", *Phys. Rev.* **D63**, 104006 (2001)

[62] M. Alcubierre, B. Brugmann, D. Pollney, E. Seidel and R. Takahashi "Black Hole Excision for Dynamic Black Holes", *Phys. Rev.* **D64**, 061501 (2001).

[63] R.M. Wald and V. Iyer, "Trapped Surfaces in the Schwarzschild Geometry and Cosmic Censorship", *Phys. Rev.* **D44**, R3719 (1991).

[64] M.A. Pelath, K.P. Tod and R.M. Wald., "Trapped Surfaces in Prolate Collapse in the Gibbons-Penrose Construction", *Class. Quant. Grav.* **15**, 3917 (1998)

[65] J. Thornburg, "Finding Apparent Horizons in Numerical Relativity", *Phys. Rev.* **D54**, 4899 (1996)

[66] M. Alcubierre, S. Brandt, B. Brugmann, C. Gundlach, J. Masso, E. Seidel and P. Walker, "Test Beds and Applications for Apparent Horizon Finders in Numerical Relativity", *Class. Quant. Grav.* **17** 2159 (2000)

[67] T. Nakamura, K. Oohara and Y. Kojima, "General Relativistic Collapse of Axially Symmetric Stars", *Prog. Theor. Phys. Suppl.* **90**, 13 (1987)

[68] L.F. Richardson, "The Approximate Arithmetical Solution by Finite Differences of Physical Problems involving Differential Equations, with an Application to the Stresses in a Masonry Dam.", *Phil. Trans. Roy. Soc.* **210**, 307 (1910)

[69] B. Gustafsson, H. Kreiss and J. Oliger, *Time Dependent Problems and Difference Methods*, New York, John-Wiley & Sons, Inc. 1995

[70] H. Kreiss and J. Oliger, "Methods for the Approximate Solution of Time Dependent Problems", *Global Atmospheric Research Programme, Publications Series No. 10.* (1973)

[71] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, Cambridge, Cambridge University Press 1997

[72] M.J. Berger and J. Oliger, "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations", *J. Comp. Phys.* **53**, 484 (1984)

[73] M.J. Berger and I.Rigoutsos, "An Algorithm for Point Clustering and Grid Generation", *IEEE Trans. Vol. 21* **No. 5** (1991)

[74] U. Trottenberg, C. Oosterlee and A. Schuller, *Multigrid*, London, Academic Press 2001

[75] A. Brandt, "Multi-level Adaptive Solutions to Boundary-value Problems", *Math. Comput.* **31**, 330 (1977)

[76] J. Thornburg, "A Multiple-Grid-Patch Evolution Scheme for 3-D Black Hole Excision", to appear in *Proceedings of the 9th Marcel Grossman Meeting*, qr-qc/0012012

[77] L. Smarr, "The Structure of General Relativity with a Numerical Illustration: The Collision of Two Black Holes", *Univ. of Texas at Austin Ph.D. Thesis* (1975)

[78] K.R. Eppley, *Princeton Ph.D. Thesis* (1977)

[79] P. Anninos, D. Hobill, E. Seidel, L. Smarr and W. Suen, "Collision of Two Black Holes", *Phys. Rev. Lett.* **71**, 2851 (1993)

[80] J. Baker, A. Abrahams, P. Anninos, S. Brandt, R. Price, J. Pullin and E. Seidel, "The Collision of Boosted Black Holes", *Phys.Rev.* **D55**, 829 (1997)

[81] P. Anninos and S. Brandt, "Head-On Collision of Two Unequal Mass Black Holes", *Phys. Rev. Lett.* **81**, 508 (1998)

[82] T. Nakamura, "General Relativistic Collapse of Axially Symmetric Stars Leading to the Formation of Black Holes", *Prog. Theor. Phys.* **65**, 1876 (1981)

[83] R.F. Stark and T. Piran, "Gravitational-Wave Emission from Rotating Gravitational Collapse", *Phys. Rev. Lett.* **55**, 891 (1985),
erratum *Phys. Rev. Lett.* **56**, 97 (1986).

[84] M. Alcubierre, S. Brandt, B. Brugmann, D. Holz, E. Seidel, R.Takahashi and J. Thornburg, "Symmetry without Symmetry: Numerical Simulation of Axisymmetric Systems using Cartesian Grids", *Int. J. Mod. Phys.* **D10** 273, (2001)

[85] M. Shibata, "Axisymmetric Simulations of Rotating Stellar Collapse in Full General Relativity — Criteria for Prompt Collapse to Black Holes", *Prog. Theor. Phys.* **104** 325 (2000)

[86] A.M. Abrahams, G.B. Cook, S.L. Shapiro and S.A. Teukolsky, "Solving Einstein's Equations for Rotating Spacetimes: Evolution of Relativistic Star Clusters", *Phys. Rev.* **D49**, 5153 (1994)

[87] A.M. Abrahams, S.L. Shapiro and S.A. Teukolsky, "Disk Collapse in General Relativity", *Phys. Rev.* **D 50**, 7282 (1994)

[88] S.L. Shapiro and S.A. Teukolsky, "Formation of Naked Singularities: The Violation of Cosmic Censorship", *Phys. Rev. Lett.* **66**, 994 (1991)

[89] S.L. Shapiro and S.A. Teukolsky, "Collisions of Relativistic Clusters and the Formation of Black Holes", *Phys. Rev.* **D45**, 2739 (1992)

[90] S. Brandt, J.A. Font, J.M. Ibanez, J. Masso and E. Seidel, "Numerical Evolution of Matter in Dynamical Axisymmetric Black Hole Spacetimes", *Comput. Phys. Commun.* **124** 169 (2000)

[91] S. Brandt and E. Seidel, "The Evolution of Distorted Rotating Black Holes I: Methods and Tests", *Phys. Rev.* **D52**, 856 (1995)

[92] S. Brandt and E. Seidel, "The Evolution of Distorted Rotating Black Holes II: Dynamics and Analysis", *Phys. Rev.* **D52**, 870 (1995)

[93] S. Brandt and E. Seidel, "The Evolution of Distorted Rotating Black Holes III: Initial Data", *Phys. Rev.* **D54**, 1403 (1996)

[94] D. Garfinkle and G.C. Duncan, "Numerical Evolution of Brill Waves", *Phys.Rev. D63*, 044011 (2001)

[95] A.M. Abrahams and C.R. Evans, "Trapping a Geon: Black Hole Formation by an Imploding Gravitational Wave", *Phys. Rev.* **D46**, R4117 (1992).

[96] A.M. Abrahams and C.R. Evans, "Critical Behavior and Scaling in Vacuum Axisymmetric Gravitational Collapse", *Phys. Rev. Lett.* **70**, 2980 (1993).

[97] J.M. Overduin and P.S. Wesson, "Kaluza-Klein Gravity", *Phys. Rept. 283*, 303 (1997)

[98] D.J. Kaup, "Klein-Gordon Geon", *Phys. Rev.* **172**, 1331 (1968)

[99] R. Ruffini and S. Bonazzola, "Systems of Selfgravitating Particles in General Relativity and the Concept of an Equation of State", *Phys. Rev.* **187** 1767 (1969)

[100] M.W. Choptuik, I. Olabarrieta, W.G. Unruh and J. Ventrella, *in preparation*

[101] J.M. Bardeen and T. Piran, "General Relativistic Axisymmetric Rotating Systems: Coordinates and Equations", *Phys. Rep.* **96** 205 (1983)

[102] K.S. Thorne, in *Magic Without Magic; John Archibald Wheeler*, edited by J. Klauder, Frieman, San Francisco, 1972

[103] E. Seidel, "Black Hole Coalescence and Mergers: Review, Status, And 'Where are we Heading?' ", *Prog. Theor. Phys. Suppl.* **136**, 87 (1999).

[104] B. Brugmann, "Numerical Relativity in (3+1)-Dimensions", *Annalen Phys.* **9**, 227 (2000).

[105] M. Alcubierre et al, "The 3D Grazing Collision of Two Black Holes", *Phys. Rev. Lett.* **87**, 271103 (2001)

[106] R.L. Marsa and M.W. Choptuik, "The RNPL User's Guide", http://laplace.physics.ubc.ca/Members/marsa/rnpl/users_guide/users_guide.html (1995)

[107] L.R. Petzold and A.C. Hindmarch, "LSODA—Livermore Solver for Ordinary Differential equations, Automatic method switching for stiff and nonstiff problems", Lawrence Livermore National Laboratory (1987)

[108] M.W. Choptuik, Private Communication.

[109] D. Choi, Private Communication.

[110] W. Hackbusch, *Theory and Numerical Treatment of Elliptic Differential Equations*, Springer, New York, 1992

[111] R. D'Inverno, *Introducing Einstein's Relativity*, New York, Oxford University Press 1996

[112] D. Garfinkle and G.C. Duncan, "Scaling of Curvature in Sub-critical Gravitational Collapse", *Phys.Rev. D58*, 064024 (1998)

[113] T.C. Zhao and M. Overmars, *Forms Library: A Graphical User Interface Toolkit for X* 1998

[114] J. Neider, T. Davis and M. Woo, *The Official Guide to Learning OpenGL, Release 1* Addison-Wesley, 1994

[115] E. Hirschmann, Private Communication.

APPENDIX A

EQUATIONS AND FINITE DIFFERENCE OPERATORS

In this Appendix, we list the equations that we have obtained by applying the $2+1+1$ decomposition, described in Section 2.1, in our chosen coordinate system (Section 2.2.1), with a real, massless scalar field (Section 2.1.1). We also list the set of finite difference operators that we have used to convert the continuum equations to finite difference form.

A.1 Analytic Equations

Summarizing Section 2.2.1, the four-metric is

$$\begin{aligned} ds^2 = &\left(-\alpha^2 + \psi^4\left[(\beta^\rho)^2 + (\beta^z)^2\right]\right)dt^2 + 2\psi^4\left(\beta^\rho d\rho + \beta^z dz\right)dt \\ &+\psi^4\left(d\rho^2 + dz^2 + \rho^2 e^{2\rho\bar\sigma}d\phi^2\right) \end{aligned} \tag{A.1}$$

The conjugate variable to the scalar field Φ is Π_Φ (2.38), and the conjugate to $\bar\sigma$ is $\bar\Omega$ (2.35). All of these variables are functions of ρ, z and t.

The maximal slicing condition results in the following elliptic equation for α:

$$\begin{aligned} &2\left(\rho\alpha_{,\rho}\right)_{,\rho^2} + \alpha_{,zz} + \alpha_{,\rho}\left(2\frac{\psi_{,\rho}}{\psi} + (\rho\bar\sigma)_{,\rho}\right) + \alpha_{,z}\left(2\frac{\psi_{,z}}{\psi} + (\rho\bar\sigma)_{,z}\right) \\ &-\frac{\psi^4}{2\alpha}\left[(\beta^\rho{}_{,\rho} - \beta^z{}_{,z})^2 + (\beta^\rho{}_{,z} + \beta^z{}_{,\rho})^2\right] - \frac{\psi^4}{6\alpha}\left[2\alpha\rho\bar\Omega + \beta^\rho{}_{,\rho} - \beta^z{}_{,z}\right]^2 - 16\pi\alpha\Pi_\Phi^2 = 0 \end{aligned} \tag{A.2}$$

The Hamiltonian constraint is

$$\begin{aligned} &8\frac{\psi_{,\rho\rho}}{\psi} + 8\frac{\psi_{,zz}}{\psi} + 16\frac{\psi_{,\rho^2}}{\psi} + 8\left(\rho\bar\sigma\right)_{,\rho}\frac{\psi_{,\rho}}{\psi} + 8\left(\rho\bar\sigma\right)_{,z}\frac{\psi_{,z}}{\psi} \\ &+\frac{\psi^4}{2\alpha^2}\left[(\beta^\rho{}_{,\rho} - \beta^z{}_{,z})^2 + (\beta^z{}_{,\rho} + \beta^\rho{}_{,z})^2\right] + \frac{\psi^4}{6\alpha^2}\left[2\alpha\rho\bar\Omega + \beta^\rho{}_{,\rho} - \beta^z{}_{,z}\right]^2 \\ &= -16\pi\left(\Pi_\Phi^2 + \Phi_{,\rho}{}^2 + \Phi_{,z}{}^2\right) - 6\left(\rho^2\left(\rho\bar\sigma\right)_{,\rho}\right)_{,\rho^3} \\ &-2\left(\left(\rho\bar\sigma\right)_{,\rho}\right)^2 - 2\left(\rho\bar\sigma\right)_{,zz} - 2\left(\left(\rho\bar\sigma\right)_{,z}\right)^2 \end{aligned} \tag{A.3}$$

The ρ and z momentum constraints are

$$\begin{aligned} &\frac{2}{3}\beta^\rho{}_{,\rho\rho} + \beta^\rho{}_{,zz} + \frac{1}{3}\beta^z{}_{,z\rho} + 32\pi\frac{\alpha}{\psi^2}\Pi_{\Phi,\rho} - \left(\left(\frac{\alpha_{,z}}{\alpha} - 6\frac{\psi_{,z}}{\psi}\right) - \left(\rho\bar\sigma\right)_{,z}\right)\left(\beta^z{}_{,\rho} + \beta^\rho{}_{,z}\right) \\ &-\frac{2}{3}\left(\frac{\alpha_{,\rho}}{\alpha} - 6\frac{\psi_{,\rho}}{\psi}\right)\left[\beta^\rho{}_{,\rho} - \beta^z{}_{,z}\right] - \frac{8}{3}\alpha\bar\Omega - \frac{2\alpha\rho}{3}\left[6\bar\Omega\frac{\psi_{,\rho}}{\psi} + \bar\Omega_{,\rho} + 3\bar\Omega\left(\rho\bar\sigma\right)_{,\rho}\right] = 0, \end{aligned} \tag{A.4}$$

and

$$\beta^z{}_{,\rho\rho} + \frac{4}{3}\beta^z{}_{,zz} - \frac{1}{3}\beta^\rho{}_{,z\rho} + \left[(\rho\bar{\sigma})_{,\rho} + \frac{2\alpha}{\psi^6}\left(\frac{\rho\psi^6}{\alpha}\right)_{,\rho^2}\right](\beta^\rho{}_{,z} + \beta^z{}_{,\rho})$$
$$+\left[2(\rho\bar{\sigma})_{,z} - \frac{4}{3}\left(\frac{\alpha_{,z}}{\alpha} - 6\frac{\psi_{,z}}{\psi}\right)\right](\beta^z{}_{,z} - \beta^\rho{}_{,\rho}) - \frac{2\alpha\rho}{3}\left(6\bar{\Omega}\frac{\psi_{,z}}{\psi} + \bar{\Omega}_{,z}\right)$$
$$+32\pi\frac{\alpha}{\psi^2}\Pi_{\Phi,z} - 2\alpha(\bar{\sigma}_{,z})\rho^2\bar{\Omega} = 0 \qquad \text{(A.5)}$$

The definition of $\bar{\Omega}$ (2.35) gives an evolution equation for $\bar{\sigma}$ (where the over-dot $\dot{}$ denotes a time derivative):

$$\dot{\bar{\sigma}} = 2\beta^\rho(\rho\bar{\sigma})_{,\rho^2} + \beta^z\bar{\sigma}_{,z} - \alpha\bar{\Omega} - \left[\frac{\beta^\rho}{\rho}\right]_{,\rho} \qquad \text{(A.6)}$$

The evolution equation for $\bar{\Omega}$ is

$$\dot{\bar{\Omega}} = 2\beta^\rho\left(\rho\bar{\Omega}\right)_{,\rho^2} + \beta^z\bar{\Omega}_{,z} - \frac{1}{2\alpha\rho}\left(\beta^z{}_{,\rho}{}^2 - \beta^\rho{}_{,z}{}^2\right) + \frac{1}{\psi^4}\left(\frac{\alpha_{,\rho}}{\rho}\right)_{,\rho}$$
$$+\frac{\alpha}{\psi^6}\left(\frac{(\psi^2)_{,\rho}}{\rho}\right)_{,\rho} - \frac{2\alpha}{\psi^4}\left(4\frac{\psi_{,\rho^2}}{\psi} + (\rho\bar{\sigma})_{,\rho^2}\right)\left(\frac{\alpha_{,\rho}}{\alpha} + \frac{2\psi_{,\rho}}{\psi}\right)$$
$$-\frac{\alpha}{\psi^4}\left[\bar{\sigma}_{,z}\left(\frac{\alpha_{,z}}{\alpha} + \frac{2\psi_{,z}}{\psi}\right) + \rho\bar{\sigma}_{,z}{}^2 + \bar{\sigma}_{,zz}\right] + 64\pi\frac{\alpha}{\psi^4}\rho(\Phi_{,\rho^2})^2 \qquad \text{(A.7)}$$

Using ${}^{(3)}K = 0$, we obtain an evolution equation for ψ, which can be used instead of the Hamiltonian constraint (A.3) to solve for ψ:

$$\dot{\psi} = \psi_{,z}\beta^z + \psi_{,\rho}\beta^\rho + \frac{\psi}{6}\left(2\beta^\rho_{,\rho} + \beta^z_{,z} + \rho\alpha\bar{\Omega}\right). \qquad \text{(A.8)}$$

The definition of Π_Φ and the wave equation for Φ give

$$\dot{\Phi} = \beta^\rho\Phi_\rho + \beta^z\Phi_z + \frac{\alpha}{\psi^2}\Pi_\Phi, \qquad \text{(A.9)}$$

and

$$\dot{\Pi}_\Phi = \beta^\rho\Pi_{\Phi,\rho} + \beta^z\Pi_{\Phi,z} + \frac{1}{3}\Pi_\Phi\left(\alpha\rho\bar{\Omega} + 2\beta^\rho_{,\rho} + \beta^z_{,z}\right)$$
$$+\frac{1}{\psi^4}\left[2\left(\rho\alpha\psi^2\Phi_\rho\right)_{,\rho^2} + \left(\alpha\psi^2\Phi_z\right)_{,z}\right] + \frac{\alpha}{\psi^2}\left[(\rho\bar{\sigma})_{,\rho}\Phi_\rho + (\rho\bar{\sigma})_{,z}\Phi_z\right] \qquad \text{(A.10)}$$

The set of axis regularity conditions, applied at $\rho = 0$ are:

$$\begin{aligned}
\alpha_{,\rho} &= 0, \\
\psi_{,\rho} &= 0, \\
\beta^z_{,\rho} &= 0, \\
\beta^\rho &= 0, \\
\bar{\sigma} &= 0, \\
\bar{\Omega} &= 0, \\
\Phi_{,\rho} &= 0, \\
\Pi_{\Phi,\rho} &= 0.
\end{aligned} \tag{A.11}$$

The outer boundary conditions we use, applied at $\rho = \rho_{max}$, $z = z_{max}$ and $z = z_{min}$, are:

$$\begin{aligned}
\alpha = 1, \quad \text{or} \quad \alpha - 1 + \rho\alpha_{,\rho} + z\alpha_{,z} &= 0, \\
\psi = 1, \quad \text{or} \quad \psi - 1 + \rho\psi_{,\rho} + z\psi_{,z} &= 0, \\
\beta^z = 0, \quad \text{or} \quad \beta^z + \rho\beta^z_{,\rho} + z\beta^z_{,z} &= 0, \\
\beta^\rho = 0, \quad \text{or} \quad \beta^\rho + \rho\beta^\rho_{,\rho} + z\beta^\rho_{,z} &= 0, \\
r\bar{\sigma}_{,t} + \rho\bar{\sigma}_{,\rho} + z\bar{\sigma}_{,z} + \bar{\sigma} &= 0, \\
r\bar{\Omega}_{,t} + \rho\bar{\Omega}_{,\rho} + z\bar{\Omega}_{,z} + \bar{\Omega} &= 0, \\
r\Phi_{,t} + \rho\Phi_{,\rho} + z\Phi_{,z} + \Phi &= 0, \\
r\Pi_{\Phi,t} + \rho\Pi_{\Phi,\rho} + z\Pi_{\Phi,z} + \Pi_\Phi &= 0.
\end{aligned} \tag{A.12}$$

See Section 2.2.6 for a description of the excision boundary conditions.

A.2 Finite Difference Operators

In this section, we write out all of the difference operators we have used to convert the differential equations in the previous section to finite difference equations. At all interior points of the mesh, the centered forms of the derivate operators are used, and along boundaries (including the excision boundary), backward and forward operators are used as appropriate. Kreiss-Oliger style dissipation (see Section 1.5.4) is applied to evolution equations, at interior points at least two grid points inward, in the direction of the stencil, from any boundary. For $\bar{\sigma}$ and $\bar{\Omega}$, we linearly interpolate in ρ at location $\Delta\rho$ (and optionally at $2\Delta\rho$ as well), using the values of these variables at $\rho = 0$ and $\rho = 2\Delta\rho$ (or $\rho = 3\Delta\rho$). Below, we use the notation $u_{i,j}$ to label a point in the mesh corresponding to coordinate location $((i-1)\Delta\rho, (j-1)\Delta z + z_{min})$ (except for the coordinate variable ρ, where it is sufficient to use ρ_i). For time derivatives, we use $u^n_{i,j}$ to denote the retarded time level, and $u^{n+1}_{i,j}$ the advanced time level. All of the finite-difference operators are 2nd order accurate.

Centered Difference Operators

$$u_{,\rho} \rightarrow \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta\rho} \tag{A.13}$$

$$u_{,z} \rightarrow \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta z} \tag{A.14}$$

$$u_{,\rho\rho} \rightarrow \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta\rho)^2} \tag{A.15}$$

$$u_{,zz} \rightarrow \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta z)^2} \tag{A.16}$$

$$u_{,\rho^2} \rightarrow \frac{u_{i+1,j} - u_{i-1,j}}{\rho_{i+1}^2 - \rho_{i-1}^2} \tag{A.17}$$

$$(u/\rho)_{,\rho} \rightarrow \frac{u_{i+1,j} + u_{i,j}}{2\Delta\rho(\rho_i + \Delta\rho/2)} - \frac{u_{i,j} + u_{i-1,j}}{2\Delta\rho(\rho_i - \Delta\rho/2)} \tag{A.18}$$

$$\left(u_{,\rho}/\rho\right)_{,\rho} \rightarrow \frac{u_{i+1,j} - u_{i,j}}{(\Delta\rho)^2(\rho_i + \Delta\rho/2)} - \frac{u_{i,j} - u_{i-1,j}}{(\Delta\rho)^2(\rho_i - \Delta\rho/2)} \tag{A.19}$$

$$u_{,t} \rightarrow \frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} \tag{A.20}$$

Forward-Difference Operators

$$u_{,\rho} \rightarrow \frac{-u_{i+2,j} + 4u_{i+1,j} - 3u_{i,j}}{2\Delta\rho} \tag{A.21}$$

$$u_{,z} \rightarrow \frac{-u_{i,j+2} + 4u_{i,j+1} - 3u_{i,j}}{2\Delta z} \tag{A.22}$$

$$u_{,\rho^2} \rightarrow \frac{-u_{i+2,j} + 4u_{i+1,j} - 3u_{i,j}}{4\rho_i\Delta\rho} \tag{A.23}$$

$$(u/\rho)_{,\rho} \rightarrow \frac{-u_{i+2,j}/\rho_{i+2} + 4u_{i+1,j}/\rho_{i+1} - 3u_{i,j}/\rho_i}{2\Delta\rho} \tag{A.24}$$

$$\begin{aligned}\left(u_{,\rho}/\rho\right)_{,\rho} &\rightarrow \frac{u_{i,j}(12\rho_i + 11\Delta\rho)/(6\rho_i^2) - u_{i+1,j}(5\rho_i + 3\Delta\rho)/(\rho_i^2)}{(\Delta\rho)^2} \\ &+ \frac{u_{i+2,j}(8\rho_i + 3\Delta\rho)/(2\rho_i^2) - u_{i+3,j}(3\rho_i + \Delta\rho)/(3\rho_i^2)}{(\Delta\rho)^2}\end{aligned} \tag{A.25}$$

Backward-Difference Operators

$$u_{,\rho} \rightarrow \frac{u_{i-2,j} - 4u_{i-1,j} + 3u_{i,j}}{2\Delta\rho} \tag{A.26}$$

$$u_{,z} \rightarrow \frac{u_{i,j-2} - 4u_{i,j-1} + 3u_{i,j}}{2\Delta z} \tag{A.27}$$

Dissipation Operators

The following dissipation operator is applied, as discussed in Section 1.5.4, in the ρ direction:

$$\frac{\epsilon_d}{16}\left(u_{i-2,j} - 4u_{i-1,j} + 6u_{i,j} - 4u_{i+1,j} + u_{i+2,j}\right) \tag{A.28}$$

and in the z direction:

$$\frac{\epsilon_d}{16}\left(u_{i,j-2} - 4u_{i,j-1} + 6u_{i,j} - 4u_{i,j+1} + u_{i,j+2}\right), \tag{A.29}$$

where ϵ_d ranges from 0 to 1.

APPENDIX B

DATA ANALYSIS AND VISUALIZATION

In this appendix, we describe aspects of a data analysis and visualization program, called *the Data-Vault* (DV for short), that we have written to aid in the development of AMR based finite difference codes. The original idea and design philosophy behind the DV is due to Choptuik; I was responsible for some additional details in this first specification, and wrote the corresponding programs.

The two principle goals of the DV are: 1) to serve as a central data repository, receiving and collating information produced by programs running across a network (the *core* functionality of the DV), and 2) to provide a graphic-user-interface (GUI) based front-end for a user to efficiently visualize and manipulate the data in the repository.

From the outset, the design philosophy behind the DV was that the core functionality should be implemented as some realization of a *virtual machine*. In other words, we think of the core functions of the DV as a CPU (central-processing-unit), that has internal registers, an instruction set, and the ability to execute user programs (an example of which would be the GUI front-end). There are several advantages to designing the program in this fashion. First of all, we cannot, and should not, try to foresee all of the data analysis functions that will be required to interpret results from finite-difference based numerical codes. Instead, we want to try to identify the basic building-blocks of these data analysis functions, and then provide a programmable environment so that the user can, with relative ease, build the necessary analysis tools. A virtual machine is a natural concept to achieve this environment. Second, we know that we will not be able to exactly identify what all the needed building-blocks are, until the applications exist that produce the data. For example, the current virtual machine specification, described in Section B.1 below, was driven by the needs of our 2D AMR code, and it may not be entirely sufficient for codes using non-uniform grids, staggered meshes, vector fields, etc. Therefore, we expect that the DV will change as our needs expand, and it is important to have a design philosophy that is flexible enough to accommodate changes in a manner that will not degrade the performance of the software. One way to achieve this is to *define* the DV as an implementation of the virtual machine, and hence fundamental modifications to the program should be made starting from the level of the virtual machine specification.

In the remainder of this appendix, we describe the virtual machine specification of the DV, and give some details of the current implementation and GUI visualization front-end.

B.1 The Data-Vault Virtual Machine Specification

We present the DV virtual machine specification in three parts—the definition of a register (Section B.1.1), the basic instruction set (Section B.1.2), and the program execution environment (Section B.1.3).

B.1.1 Register Structure

A *register* contains the fundamental unit of data produced by our AMR-based finite difference codes, namely a *grid function*. A grid function is the discretized representation of a continuum function, in both space *and* time. This leads to the following basic definition of a register, summarized in Figure B.1 below: a register is a linked list of *time structures*, a time structure is a linked list of *level structures*, and a level structure is a linked list of single grid structures. In other words,

Name	Description
name	a unique character string labelling the register
next	pointer to the next register within the global list of registers
prev	pointer to the previous register within the global list of registers
ts	pointer to the list of time structures within this register
coord_names[4]	optional strings for the names of the coordinates (time plus 3 spatial)
fuzzy[4]	a real number per coordinate, defining a threshold for "fuzzy logic"
...	optional, user-definable attributes

Table B.1: DV register structure definition

Name	Description
time	the coordinate time of all grids within this time structure
next	pointer to the next time structure within the list
prev	pointer to the previous time structure within the list
levels	pointer to the list of level structures at this time
...	optional, user-definable attributes

Table B.2: DV time structure definition

all grids linked to a register structure (via level and time links) belong to the same grid function; all grids linked to a time structure (via level links) share the same coordinate time; and all grids linked to the same level structure share the same spatial discretization scale. Notice then, that by "fundamental unit of data", we do not mean the smallest, indivisible object produced by the code (which would be a single real number); rather, we want a register to represent a single variable of our solution.

In addition to linked-list information, each data structure within a register has several named attributes, describing the structure. The DV also provides a mechanism for user-defined attributes to be added, modified and removed at any structure within a register. Tables B.1, B.2, B.3 and B.4 below describe all of the attributes currently defined for the register, time, level and grid structures respectively.

An important aspect of a register, is that the sequence of levels, times and grids within a given linked list is *ordered* and *unique*. By unique, we mean the following: a register is built by adding a set of grids, one by one, to the register (see the instruction **add_grid_str** in Section B.1.2); the register structure obtained after all the grids have been added is required to be unique, regardless of the sequence in which the individual grids were added. The uniqueness, and resultant ordering of grids, is defined by the instruction **gridcmp(A,B)** (see Section B.1.2), that determines whether grid A is "less than", "greater than" or "equal" to grid B. The particular criteria used by **gridcmp** is up to the implementation, as is the course of action to be taken if two grids cannot be differentiated (if they are "equal").

B.1.2 Instruction Set

In this section we describe the "minimal" set of primitive instructions that should be provided by any implementation of the DV. In practice, depending upon the implementation, a larger set of instructions may be necessary. For example, as described in B.2, in the current version of DV, all of the data structures are implemented as **C** structures, and we allow programs to directly access the structure elements, bypassing the need to define instructions to read/write individual attributes of

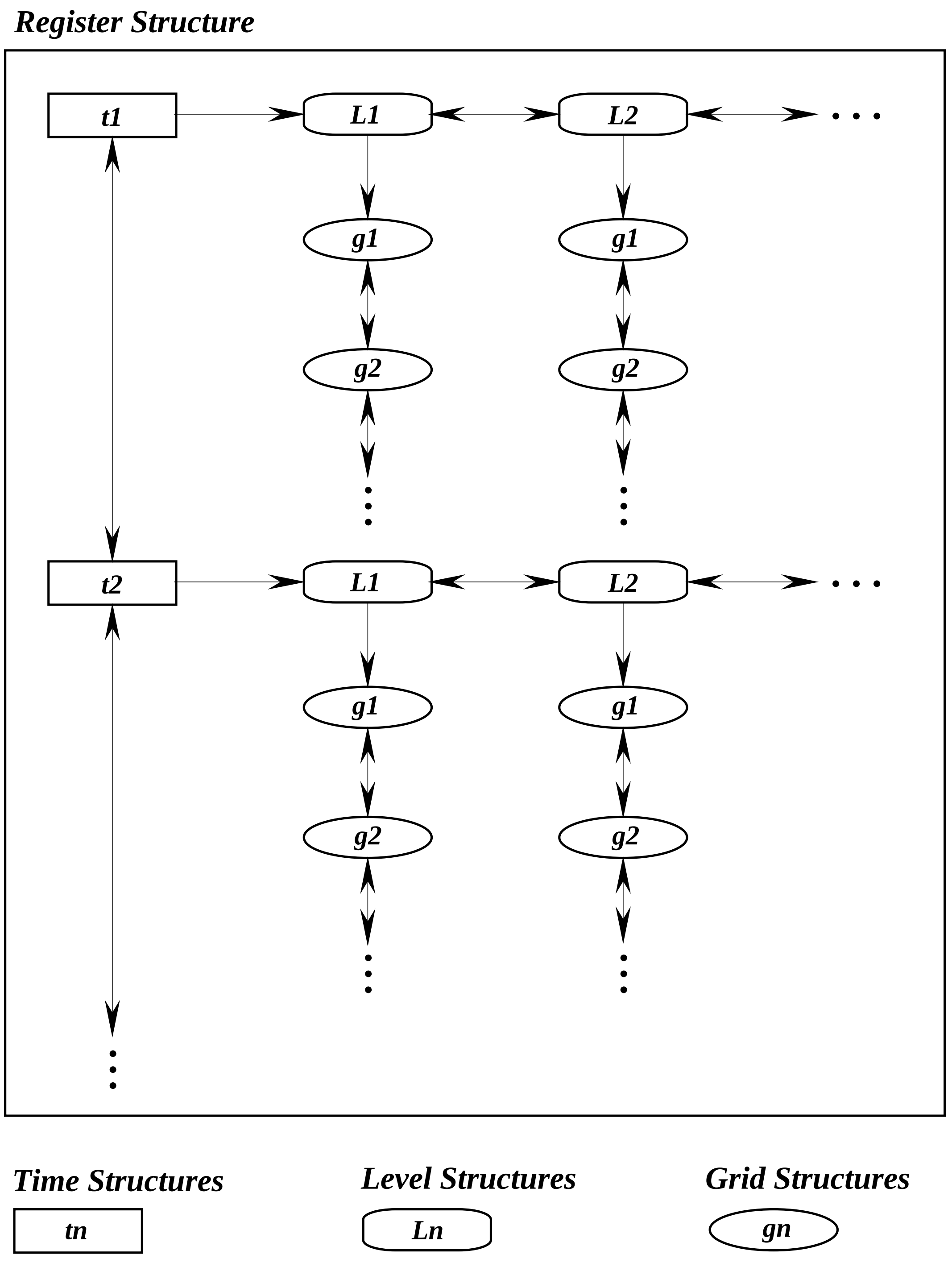


Figure B.1: A diagrammatic description of the Data-Vault register structure.

Name	Description
dx	the mesh spacing defining this level
next	pointer to the next level structure within the list
prev	pointer to the previous level structure within the list
grids	pointer to the list of grid structures at this level
...	optional, user-definable attributes

Table B.3: DV level structure definition

Name	Description
next	pointer to the next grid structure within the list
prev	pointer to the previous grid structure within the list
dim	spatial dimension of the grid
shape[**dim**]	size of the grid
coord_type	a flag, indicating the grid type (uniform, curvilinear, etc.)
data	pointer to the data array of the grid
coords	pointer to the coordinate array(s) of the grid
...	optional, user-definable attributes

Table B.4: DV grid structure definition

a register[1] (one exception is the **rename_reg** instruction to rename a register).

add_grid_str

The **add_grid_str(A,g)** instruction adds a grid structure **g** to register **A**. If register **A** does not yet exist, then **add_grid_str** will create it before adding the grid. The position within **A** where **g** is inserted is governed by the operation of the instruction **gridcmp**.

gridcmp

The **gridcmp(g1,g2)** instruction compares grid **g1** to **g2**, and indicates (by whatever mechanism the implementation provides) whether $g1 > g2$, $g1 < g2$ or $g1 = g2$. See Section B.2 for the particular sequence of comparisons used in the current implementation.

delete_reg

delete_reg(A) removes register **A** from the DV.

delete_grids

delete_grids(A,g1,g2,...) removes the set of grids **g1,g2,...** from register **A**. If all grids are removed by this operation, then register **A** is deleted. In other words, *empty registers are not allowed.*

rename_reg

rename_reg(A,B) changes the name of register **A** to **B**, if **B** does not exist. This operation must always be performed as an instruction, to enforce the required uniqueness of a register name

[1]Of course, this is somewhat dangerous in the sense that we cannot enforce the read-only property that many attributes are implied to have.

(thus in the current implementation, users must refrain from simply altering the **name** field of the register structure).

gclip

The instruction **g1=gclip(g2,ibbox)** returns in **g1** a clipped version of grid **g2**, as specified by the index bounding box **ibbox**. See Section B.2.1 for a description of an index bounding box.

Sequential Iterator Functions

At this stage, the only defined mechanism by which a user can access individual data structures within a register is via sequential iteration, with limited back-tracking support, and optional filtering of the data. In the future it may be useful to define a random access mechanism, however we have not yet had the need for it. The set of sequential iterator instructions are: **init_s_iter,next_g,next_ts,next_l, save_s_iter** and **restore_s_iter**. Each one of the **next_...** instructions, when invoked repetitively after initialization with **init_s_iter**, implements a depth-first traversal of the register structure up to a given level. However, nothing prevents the user from issuing an arbitrary sequence of different **next_...** instructions, resulting in a more complicated traversal of the register structure. The following paragraphs describe each of these functions in detail.

init_s_iter

The instruction **g=init_s_iter(S,A,givec)** initializes an iterator **S** for register **A**, with an optional "generalized index vector" **givec** allowing the user to select a subset of grids within the register to loop through. The first grid **g** in the register is returned. The generalized index vector contains information selecting a set of times, levels, and/or some portion of the computational domain. When a **givec** is specified, the iteration instructions **next_...** do *not* actually skip over or clip grids; rather a flag within the iterator is set to one of **GIV_ON** (the current grid is included within the selection specified by **givec**), **GIV_OFF** (the current grid is excluded), or **GIV_CLIP** (the current grid is included, but must be clipped — see the **gclip** function above, and the iterator supplies the relevant index bounding box for the clip function). The user can then skip/clip as desired. It is up to the implementation how complicated a selection is allowed by **givec**—see Section B.2.1 for that of the current version of the DV.

next_g

The instruction **g=next_g(S)** returns the next grid **g**, in sequence, within the register referenced by the iterator **S**. The sequence of grids is as defined by the **gridcmp** function, where grid **g1** will be returned in a set of **next_g** calls before **g2** if $g1 < g2$. Equivalently, the register tree-structure depicted in Figure B.1 is traversed in depth first order by **next_g**.

next_l

The instruction **g=next_l(S)** returns the first grid **g** at the next level within the register referenced by the iterator **S**, in the sequence defined by the **gridcmp** function. Equivalently, a sequence of **next_l** calls implements a depth first traversal of the tree-structure depicted in Figure B.1, up to the depth of the level nodes.

next_ts

The instruction **g=next_ts(S)** returns the first grid **g** at the next time within the register referenced by the iterator **S**, in the sequence defined by the **gridcmp** function. Equivalently, a sequence of **next_ts** calls implements a depth first traversal of the tree-structure depicted in Figure B.1, up to the depth of the time nodes.

save_s_iter, restore_s_iter

The instruction **save_s_iter(S)** records (internally) the current position of the iterator **S** within the corresponding register. The instruction **g=restore_s_iter(S)** resets the position of the iterator **S** to that saved by the most recent **save_s_iter** instruction, and returns the corresponding grid—if no prior **save_s_iter** had been issued, the iterator is reset to the initial state.

apply_unary_gf

The instruction **A=apply_unary_gf(f,B,MASK,MASK_VAL,givec)** applies a function **f**, that operates on a single grid, sequentially to all grids in the source register **B**, and stores the result in a *new* register **A**. The source register **B** can optionally be filtered with a generalized index vector **givec**, and an optional mask register **MASK** with corresponding real number value **MASK_VAL** can be specified. Registers **B** and **MASK** are required to have identical structure. It is up to the function **f** to interpret the mask; the DV merely synchronizes and optionally filters the source and mask registers prior to the call to **f**. The intended purpose of the mask is to specify a non-trivial region (i.e. more complicated that what can be specified via **givec**) within the register where the given function is to be applied.

The function **f** is required *not* to alter the contents of any source or mask grids; the result of the function must be returned via a new grid. This is to prevent the operation of **f** from causing the structure of a register to be altered mid-stream during **A=apply_unary_gf**, which could happen if **f** were allowed to alter a grid in a manner that changed its relative position within the register (for instance in a coarsening operation). This also allows **f** to be completely ignorant of the register structure (in other words **f** only needs to know what a *grid* is).

apply_binary_gf

The instruction **A=apply_binary_gf(f,B,C,MASK,MASK_VAL,givec)** operates identically to **apply_unary_gf** described above, except here **f** is a binary grid function, accepting two (identical) grids as input, and returning a single grid as output.

B.1.3 Execution Model

Here we briefly describe the execution model of the DV, which is the mechanism by which the DV executes a "program". In some sense the execution model is more akin to the operation of a client/server application, rather than a traditional microprocessor, as the current DV instruction set does not offer program control statements, such as branching statements, logical operations, etc. Stated simply, the execution model of the DV is an *instruction queue*. Users specify the program to be executed in the DV by placing instructions on the queue. No user can obtain exclusive access to the queue, and instructions are executed on a first-come-first-serve basis. Hence, there are no guarantees that a given program will produce the same results, if multiple users are queuing instructions simultaneously.

Depending upon the implementation of the DV, multiple instructions within the queue could be executed simultaneously, as long as the implementation can ensure that there are no conflicts in doing so (in other words, the end results of parallel execution should be indistinguishable from sequential execution). For example, an instruction performing the arithmetic operation **A=B+C**, thus utilizing registers **A,B** and **C**, can execute concurrently with an **add_grid_str** instruction that adds grids to register **D**.

B.2 Current Implementation

In this section we briefly describe some of the details of our current implementation of the DV.

The majority of the code for the DV is written in **C**. The register structure, including all substructures discussed in Section B.1.1, are implemented as **C** data structures. In particular, all current user-definable attributes are hard-coded within the **C** data structures [2]. Thus, as **C** does not support private data structures, there is no mechanism to enforce the integrity of the internal state of the DV — it is up to the user to respect the implied "read-only" status of the register, and only modify their contents via the supplied instruction set.

The instructions described in Section B.1.2 are implemented as **C** functions, and hence a "program" running on the DV is a **C** subroutine that calls instruction functions. Some of the instructions supply an additional return value above that specified in Section B.1.2; this return value denotes success or failure of a particular instruction (for instance, if there is insufficient memory for an **add_grid_str** instruction to add the grid to the register, then an error will be returned). The DV specification could be extended to allow for such error flags, however, some of the errors are implementation specific, and so we have not added these flags to the specification yet. The sequence of keys currently used in **gridcmp** are (in order): time, grid dimension, x_1, x_2, x_3, N_{x1}, N_{x2} and N_{x3}, where x_1, x_2, x_3 denote spatial coordinates, and N_{x1}, N_{x2}, N_{x3} denote array sizes of the corresponding grid dimensions.

There are currently two programs running on the DV. The first is a TCP/IP server that receives grid function data (in **sdf** format —see [106] for a description of this format). The only instruction it ever issues is **add_grid_str**, after receiving **sdf** data. The second program is a GUI-based interface to the DV, allowing the user, via "point-and-click" operations, to view the current register structures in the DV, and to issue various instructions to manipulate the registers. In addition, the GUI offers a visualization front-end, providing users with the ability to view 2D grid functions as surface plots. The GUI was written with the aid of Xforms [113], and the visualization routines with OpenGL [114]. See Figures B.2-B.4 below for sample screen shots of the DV.

At this stage there is no explicit instruction queue. An instruction is executed whenever the corresponding function implementing the instruction is executed. The two programs (the TCP/IP server and GUI interface) do run concurrently, and access the various data structures using the semaphore mechanism provided by the Linux operating system, to avoid conflicts.

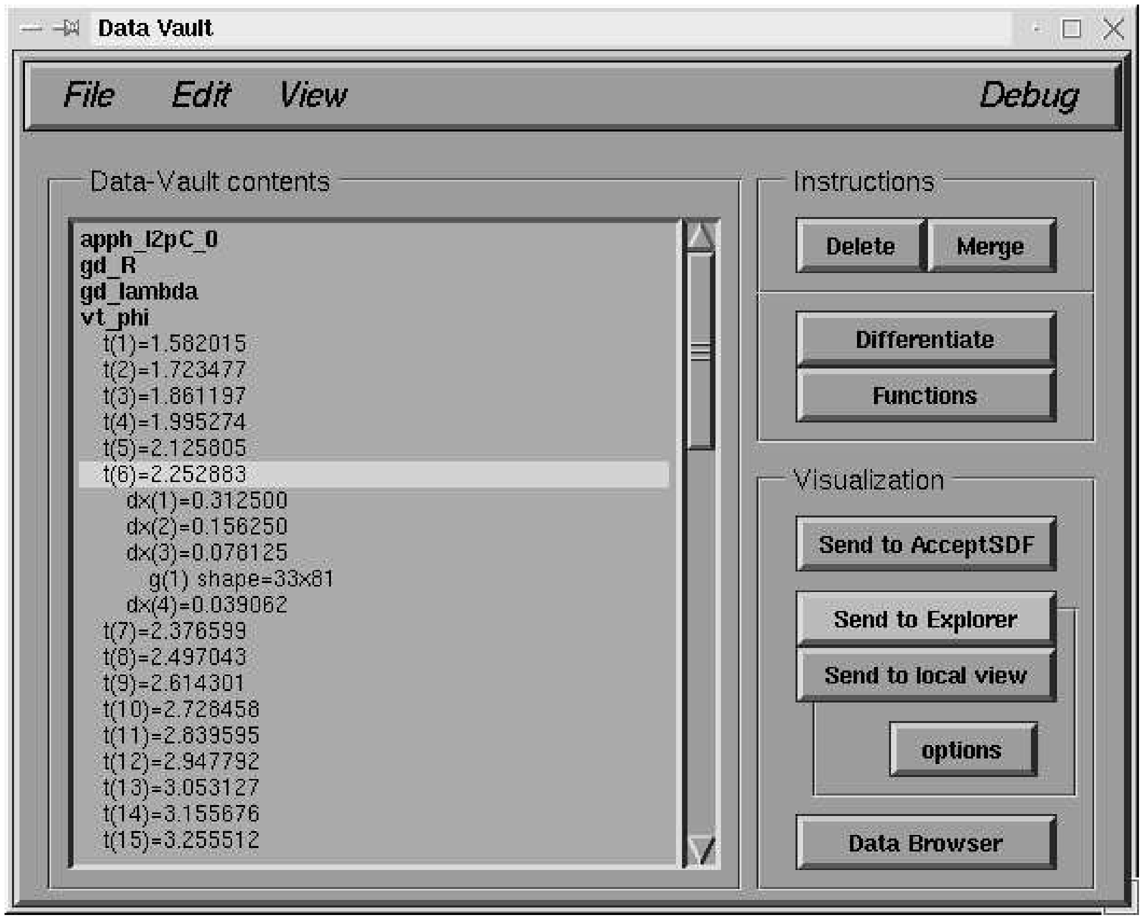
B.2.1 Generalized Index Vector format

In the present version of the DV, the generalized index vector option of the sequential iterator (see Section B.1.2) is specified via a character string, and has the format

```
t=<ivec>;l=<ivec>; [cb=x1,x2,y2,y2,... | ib=i1,i2,j1,j2,...]
```

$\langle ivec \rangle$ is an *index vector* as implemented in the **sdf** library [106], and offers a convenient notation for specifying sequences of integers. An example of an index vector is "1,3,5-10/2,15-*"; single numbers are interpreted verbatim, a set $n_1 - n_2/s$ denotes the sequence $n_1, n_1 + s, n_1 + 2s, ..., n_2$, and an asterisk denotes the last valid index of the relevant data structure. The $t = \langle ivec \rangle$ statement selects a sequence of times (1 to the number of time structures within a register, with 1 being the earliest time), and the $l = \langle ivec \rangle$ statement selects a sequence of levels (1 to the number of level structures within a register, 1 being the coarsest level). The $cb = x_1, x_2, y_2, y_2, ...$ term specifies a coordinate bounding box, whereby all data outside of the rectangle $[x_1..x_2, y_1..y_2, ...]$ will be clipped. Similarly $ib = i_1, i_2, j_1, j_2, ...$ specifies an *index bounding box*, whereby each grid is clipped relative to its array shape $[1..N_x, 1..N_y, ...]$. Only one of the coordinate or index bounding box statements may be specified within a generalized index vector; any combination of a time index

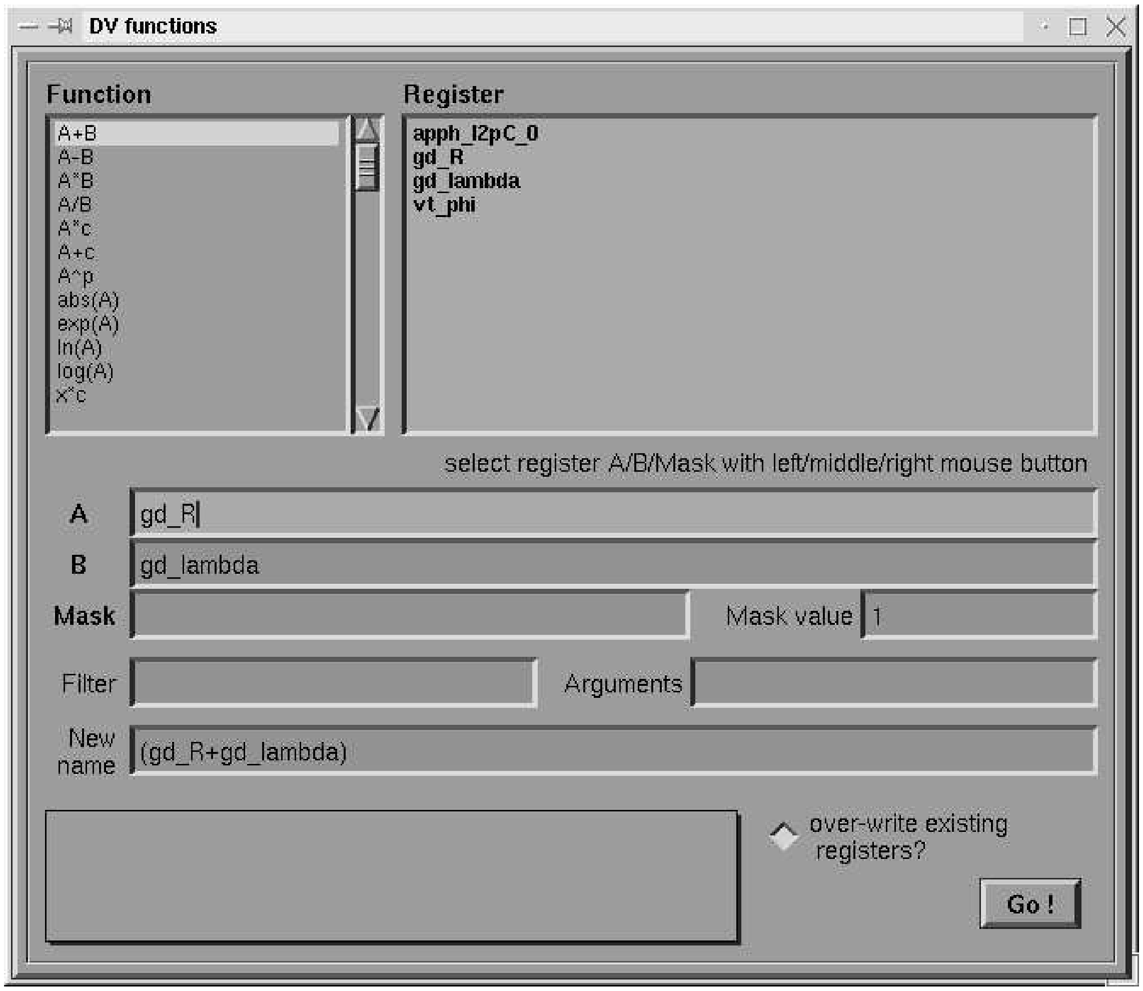
[2] In the future it may be useful to solidify this aspect of the specification, i.e. provide instructions to *dynamically* define and access user-definable attributes.



```
apph_l2pC_0
gd_R
gd_lambda
vt_phi
  t(1)=1.582015
  t(2)=1.723477
  t(3)=1.861197
  t(4)=1.995274
  t(5)=2.125805
  t(6)=2.252883
    dx(1)=0.312500
    dx(2)=0.156250
    dx(3)=0.078125
      g(1) shape=33x81
    dx(4)=0.039062
  t(7)=2.376599
  t(8)=2.497043
  t(9)=2.614301
  t(10)=2.728458
  t(11)=2.839595
  t(12)=2.947792
  t(13)=3.053127
  t(14)=3.155676
  t(15)=3.255512
```

Figure B.2: The DV GUI main window

vector, a level index vector, and a bounding box statement may be specified — multiple selections are interpreted with a logical *and*.



apph_l2pC_0
gd_R
gd_lambda
vt_phi

select register A/B/Mask with left/middle/right mouse button

A	gd_R
B	gd_lambda
Mask	
Mask value	1
Filter	
Arguments	
New name	(gd_R+gd_lambda)

Figure B.3: The DV GUI function window

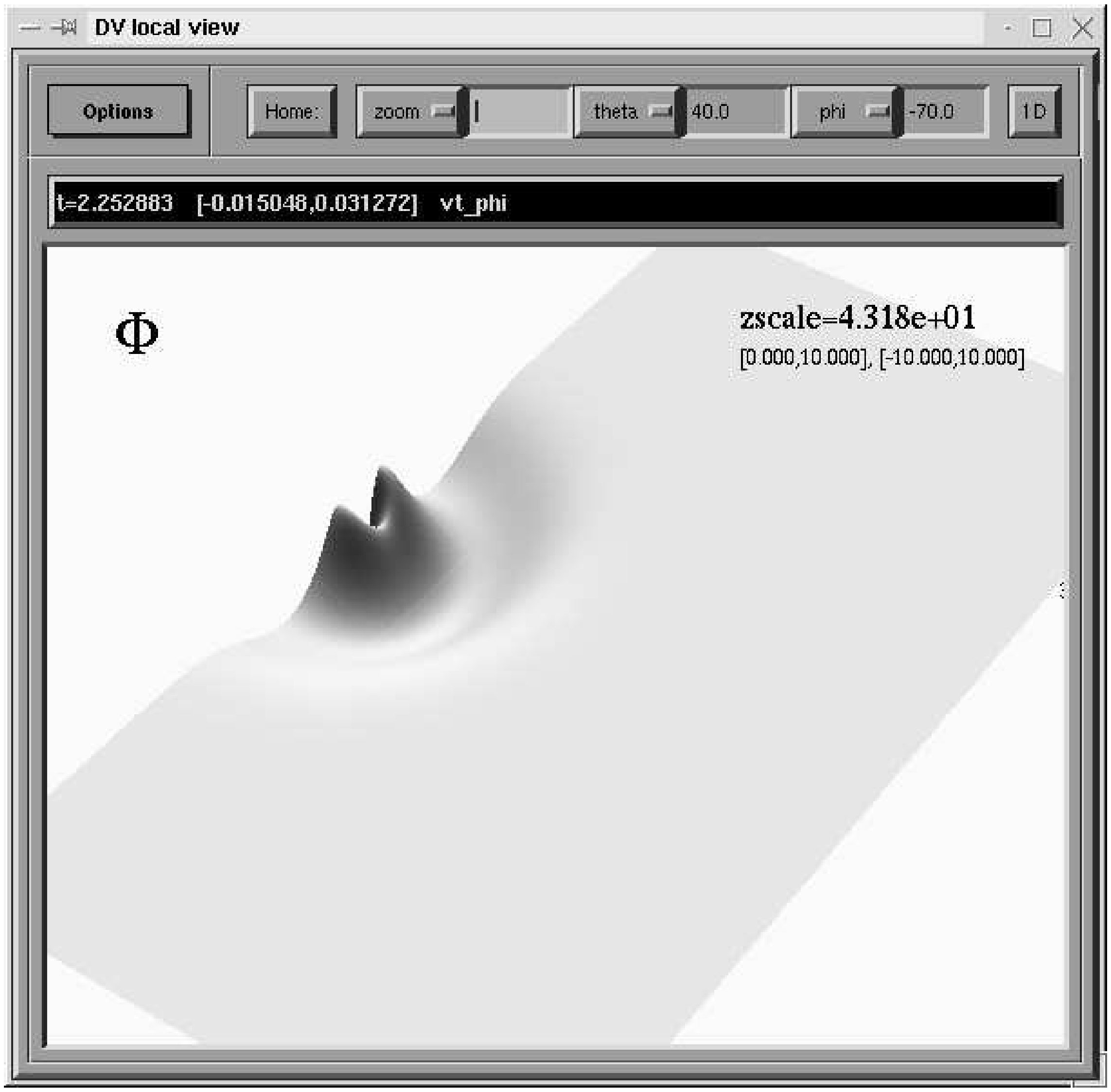


Figure B.4: The DV visualization window