Numerical Studies in Gravitational Collapse

by

Arman Akbarian Kaljahi

B.Sc., Sharif University of Technology, 2008
M.Sc., The University of British Columbia, 2010

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

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Physics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

December 2015

Abstract

In the first part of this thesis, we solve the coupled Einstein-Vlasov system in spherical symmetry using direct numerical integration of the Vlasov equation in phase space. Focusing on the case of massless particles we study critical phenomena in the model, finding strong evidence for generic type I behaviour at the black hole threshold that parallels what has previously been observed in the massive sector. For differing families of initial data we find distinct critical solutions, so there is no universality of the critical configuration itself. However we find indications of at least a weak universality in the lifetime scaling exponent, which is yet to be understood. Additionally, we clarify the role that angular momentum plays in the critical behaviour in the massless case.

The second part focuses on type II critical collapse. Using the critical collapse of a massless scalar field in spherical symmetry as a test case, we study a generalization of the BSSN formulation due to Brown that is suited for use with curvilinear coordinates. We adopt standard dynamical gauge choices, including 1+log slicing and a shift that is either zero or evolved by a Gamma-driver condition. With both choices of shift we are able to evolve sufficiently close to the black hole threshold to 1) unambiguously identify the discrete self-similarity of the critical solution, 2) determine an echoing exponent consistent with previous calculations, and 3) measure a mass scaling exponent, also in accord with prior computations. Our results can be viewed as an encouraging first step towards the use of hyperbolic formulations in more generic type II scenarios, including the as yet unresolved problem of critical collapse of axisymmetric gravitational waves.

In the last part, we present simulations of nonlinear evolutions of pure gravity waves. We describe a new G-BSSN code in axial symmetry that is capable of evolving a pure vacuum content in a strong gravity regime for both Teukolsky and Brill initial data. We provide strong evidence for the accuracy of the numerical solver. Our results suggest that the G-BSSN is promising for type II critical phenomena studies.

Preface

All of the work presented in this thesis, except the introduction chapter, are original work done by the Author and the research supervisor Matthew Choptuik. Chapter 2 and 3 of this thesis are identical to their published versions [1, 2] (Phys. Rev. D90, 104023, (2014) and Phys. Rev. D92, 084037, (2015)) with only minor changes to the typesetting to fit the thesis format. Some footnotes are added for further explanation of some of the concepts.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my research supervisor Matthew Choptuik for his guidance and tremendous support during the research work of my PhD program. I would also like to express my gratitude to other members of my PhD committee: William Unruh, Jeremy Heyl and Colin Gay for their insightful comments and thorough reading of a draft of this thesis. My colleagues and friends in numerical relativity group in UBC, Silvestre Aguilar, Graham Reid and Daoyan Wang have provided me with many helpful discussions and created a fun and stimulating work environment, and I am very thankful for that. Many thanks to my lovely friends, Joy Peng and Miya Gu, who have been my little family in Canada and brought lots of laughters and cats to my life during my PhD program. Lastly, I cannot thank my family enough for their support and encouragement. The majority of the simulations performed to obtain the results in this thesis are done on Westgrid cluster. Finally, I would like to acknowledge the financial support from UBC via FYF scholarship for my PhD program.

Chapter 1

Introduction

Einstein's theory of gravity relates the geometry of the spacetime to its matter content. The spacetime in General Relativity is modeled as a 4-dimensional Lorentzian manifold. On this manifold, gravity is identified as a characteristic of its 4 dimensional geometry (curvature), whereas in Newton's theory, gravity is a field defined on a 3 dimensional flat space and time is an independent coordinate (universal to all observers). In this geometric interpretation, the "gravitational interaction" of particles is solely by experiencing the curved spacetime as they move along the geodesics of the curved spacetime.

As one might expect, in the limit where the deviation of the spacetime metric from a flat metric is large, the general relativistic geometric equations can no longer be approximated with Newton's field equation. This limit is related to the concentration of matter, or equivalently one can define a characteristic "gravitational length", L_G, that depends on the total mass of the system[1] as:

$$L_G = \frac{G}{c^2} M \,. \tag{1.1}$$

Then the typical size of the system can be compared to L_G, and for a system with $L \approx L_G$ (a highly compact object relative to a typical star) the non-linear effects in General Relativity become prominent and the structure and dynamics of the spacetime is referred as a *strong gravity* scenario.[2] In this regime, one of the most

[1] This characteristic length for the Sun is 1.47 km.

[2] Of course, another parameter that distinguishes "classical" Newtonian system from a relativistic system is its particles typical speed compared to the speed of light: v/c.

remarkable predictions of General Relativity is the *black hole* solution—a region of spacetime that cannot causally affect the outside world.

Despite the early discovery (1916) of the black hole solution, only few other physically relevant analytic solutions have been found, since the proposal of the General Relativity theory by Einstein (1915). This is mainly due to the highly non-linear and complex nature of the geometric equations that describe the gravitational interaction. This complexity has spurred the development of numerical relativity, where large scale computing is used to model strong gravity scenarios. The main concentration of studies in numerical relativity have been on two aspects of strong gravity: 1) the astrophysically relevant scenarios, particularly in the context of compact objects dynamics and mergers, and 2) fundamental studies in the theory of General Relativity. Most of the research in this thesis is in the later spirit. We refer the reader to [3] for a recent and extensive review of the frontier of the numerical relativity field.

The fundamental studies of General Relativity in a "numerical lab" were pioneered by Choptuik [4] who numerically discovered an unexpected emerging phenomena at the threshold of black hole formation in the collapse of a massless scalar field. Following works in the collapse of various matter sources unveiled similar rich phenomenology, as the gravitational strength of the matter source is tuned to evolve precisely at the boundary of two possible classes of final states: black hole formation or dispersal (flat space). With features such as the apparent *universality* of the solution and *power-law scaling*, an analogy to phase transitions in statistical physics was established, thus the name: *critical phenomena* in gravitational collapse. The research work of this thesis is mainly focused on critical phenomena and is a contribution to fundamental studies in gravitational physics. References [5] and [6] provide extended reviews of studies in critical phenomena in gravitational collapse. and a more recent overview of the field can be found in [3]. A formal description of the threshold solution and its properties will follow shortly, and all relevant critical phenomena studies to the projects in this thesis will be reviewed in the introduction section of each chapter.

Besides the existence of black hole solutions, another prediction of General Relativity—that does not have a counterpart in Newtonian gravity—is the existence of gravitational waves. Gravitational waves may be viewed as ripples in the geometry of spacetime that propagate at a finite speed, namely that of light. Direct physical detection of gravitational waves is an ongoing global effort in physics. [7–11] The expected amplitude of a gravitational wave pulse that would reach the Earth is extremely low and can be studied in the linearized approximation. However, the source of the wave is believed to be in the regime of strong gravity dynamics, such as the merger of two black holes. The pioneering works in numerical relativity in this context are the very first successful long term evolution of binary black hole mergers by Pretorius (2005) [12] followed by Campanelli *et.al.* (2006) [13] and the extraction of the gravitational wave form by Baker *et.al.* (2006) [14]. For a review of the state of the art research in this topic see: [3].

Gravity waves are another topic of interest in this thesis, where again we adopt a fundamental perspective and focus on the numerical studies of the strong gravity regime where a gravitational wave packet can collapse and form a black hole. This part of the thesis is inspired by some unresolved questions in critical collapse of pure gravity waves, as we will discuss extensively in the next chapters.

We note that Chapter 2 and 3 of this thesis are published studies and therefore are written in a self-contained manner (to the extent that the standards of publication allow). Chapter 4 is also an independent project, but the theoretical formulation of it heavily depends on Chapter 3. Each chapter contains an introductory section where we introduce the research problem and review the literature on that topic. The rest of the current chapter aims to provide a general introduction to the formulations and techniques used in the rest of the thesis. In particular, we continue with: a formal presentation of Einstein's field equations, a description of gravity waves and black hole solutions, an overview of critical phenomena, a summary of two formulations of Einstein's equation used in numerical relativity as well as the choice of coordinates, and a quick overview of the numerical methods used in the thesis (which are more extensively discussed in the Appendix).

1.1 Notation

In this thesis we adopt units where Newton's constant G and the speed of light c are set to 1. The spacetime metric is chosen with the signature $-+++$ and all the sign conventions are similar to Misner et al.[15]. The Latin alphabets $\{a, b, c, \cdots, h\}$ are used for abstract indexing, introduced by Wald [16], of both 4-dimensional and spatial 3-dimensional tensors. We use Latin indices starting from i: $\{i, j, k, \cdots, n\}$ (that runs from 1 to 3) to specifically denote the purely spatial 3-dimensional tensors. The Greek indices (that run from 0 to 3) are used to denote the components of the 4-dimensional tensors in a specific coordinate choice. The Einstein's convention is adopted throughout the thesis, whereby the summation over repeated indices is assumed. ∇ denotes the 4-dimensional covariant derivative associated with the 4-metric, while D denotes the 3-dimensional covariant derivative associated with the induced 3-metric on the spatial hypersurfaces in a 3+1 decomposition. Parentheses enclosing the indices of a tensor denote symmetrization of the tensor, for example $A_{(ij)} = (A_{ij} + A_{ji})/2$.

1.2 Einstein's Gravitational Field Equations

The geometric description of the gravitational field is encoded in the 4-dimensional metric tensor, g_{ab}, where one dimension , identified as time, has the opposite sign to that of the rest of the 3 spatial dimensions. This metric measures the Lorentzian distance between two spacetime points:

$$ds^2 = g_{ab} dx^a dx^b \,, \tag{1.2}$$

where x^a labels the coordinates and dx^a is the difference vector between two nearby points. A special case of the metric is the flat metric (Minkowski spacetime) and in Cartesian coordinate where $x^0 = t$ and $x^i = (x, y, z)$ is given by:

$$\eta_{ab} = \mathrm{diag}(-1, 1, 1, 1) \,. \tag{1.3}$$

The structure of the spacetime (given by the metric g_{ab}) is governed by Einstein's equation:

$$G_{ab} = 8\pi T_{ab} \,, \tag{1.4}$$

where G_{ab} is the Einstein tensor and schematically depends on the metric g_{ab} and its first and second space and time derivatives. T_{ab} is the energy-momentum tensor and constitutes all of the non-gravitational energy and momentum contributions of matter sources that are present in the spacetime.

The Einstein tensor is given in terms of the 4-Ricci tensor, R_{ab} and its trace, $R = R^a_{\ a}$, as:

$$G_{ab} = R_{ab} - \frac{1}{2} g_{ab} R \,. \tag{1.5}$$

The Ricci tensor is constructed from the Riemann tensor by the contraction:

$$R_{ab} = R^c_{\ acb} \,. \tag{1.6}$$

The Riemann tensor measures the curvature of the spacetime and is defined by its action on a covariant vector as:

$$R^d_{\ abc} v_d = \nabla_b \nabla_c v_a - \nabla_c \nabla_b v_a \,. \tag{1.7}$$

It vanishes identically if and only if the spacetime is flat. The explicit form of the Riemann tensor in terms of the metric g_{ab} can be written using the definition of the Christoffel symbols, $\Gamma^a_{\ bc}$,

$$\Gamma^a_{\ bc} = \frac{1}{2} g^{ad} (\partial_c g_{db} + \partial_b g_{dc} - \partial_d g_{bc}) \, . \tag{1.8}$$

Specifically, the Riemann tensor components are:

$$R^a_{\ bcd} = \partial_c \Gamma^a_{\ bd} - \partial_d \Gamma^a_{\ bc} + \Gamma^a_{\ ec} \Gamma^e_{\ bd} - \Gamma^a_{\ ed} \Gamma^e_{\ bc} \, . \tag{1.9}$$

As one can see from (1.8,1.9) and the definition of Einstein tensor, the Einstein's equation becomes a set of 10 second order PDEs for the metric components g_{ab}. The properties of these equations will become clearer as we introduce the linearized approximation and the 3+1 decomposition of the equations. We note that there are several in-depth introductory textbooks in General Relativity such as Misner et al. [15], Weinberg [17], Wald [16] and Carroll [18]. The reader may refer to these textbooks for further explanations of the concepts introduced in this section.

1.3 Gravitational Waves

As mentioned before, one aspect of General Relativity that is absent from Newtonian gravity is the finite-speed propagation of disturbances in the gravitational field. The spacetime ripples—small changes in the metric g_{ab}—will carry information about their source and in a region far away from their origin, they can be modeled as perturbations of the Minkowski spacetime. Consider a small deviation from the flat metric in Cartesian coordinate:

$$g_{ab} = \eta_{ab} + h_{ab} \ , \ (h_{ab} \ll 1) \, , \tag{1.10}$$

and define the *trace-reversed* perturbation metric, $\bar{h}_{ab}$, as:

$$\bar{h}_{ab} = h_{ab} - \frac{1}{2}\eta_{ab}h\,, \tag{1.11}$$

where $h \equiv \eta^{cd}h_{cd}$ is the trace of the perturbed metric. Using the linearized coordinate freedom, one can further impose the condition:

$$\partial^a \bar{h}_{ab} = 0\,. \tag{1.12}$$

In this gauge choice, the linearized Einstein's equation in vacuum ($T_{ab} = 0$) is then given by:

$$G^{(1)}_{ab} = \partial^c \partial_c \bar{h}_{ab} = 0 \tag{1.13}$$

where superscript (1) denotes the approximation where we only keep the terms involving $\bar{h}_{ab}$ and its derivatives up to first order. This is a wave equation for the components of the perturbation metric, $\bar{h}_{ab}$, and illustrates the wave-like characteristics of the Einstein's equation. More discussions on linearized gravity waves can be found in standard texts on General Relativity such as [15, 16].

1.4 Gravitational Collapse: Black Hole Solution

A black hole is a region of spacetime that cannot signal information to the outside world. This region is a remnant of a *gravitational collapse* process where the strength of the self-gravitation of the matter is increasing as the matter is compressed to a rapidly shrinking region of the spacetime. Eventually, if the length scale of the system can reach to the order of its gravitational length, $L \approx L_G$, (as introduced in (1.1)) the system collapses to a black hole. From the geometrical point of view, all of the causal curves – including the null geodesics, the path of photons – become confined in the black hole region. The boundary of this region that causally disconnects the interior from the outside world is called the *event horizon*. It is known that black holes are indeed present in our universe: the super-massive black holes in the center of most galaxies [19, 20], and black holes as the final fate of massive stars [21].

One example of a spherically symmetric spacetime that contains a black hole is given by the Schwarzschild metric:

$$ds^2 = -\left(1 - \frac{2M}{r}\right) dt^2 + \left(1 - \frac{2M}{r}\right)^{-1} dr^2 + r^2 d\theta^2 + r^2 \sin^2\theta d\varphi^2 \,. \tag{1.14}$$

Here the event horizon is located at $r_s \equiv 2M$ and r_s is known as the Schwarzschild radius. [3] The interior of the black hole, $r < r_s$, cannot physically affect the outside world, $r > r_s$. The solution (1.14) is known as an eternal black hole, since it exists for all time $-\infty < t < \infty$.

This solution also demonstrates the coordinate dependence of the metric components, where here g_{rr} is divergent at the event horizon $r = r_s = 2M$. However, this pathology is purely due to the choice of coordinates and can be removed by a coordinate transformation. For example, the Schwarzschild solution can be written in an *isotropic* radial coordinate, [4] where it takes the form:

$$ds^2 = -\left(\frac{1 - M/2\tilde{r}}{1 + M/2\tilde{r}}\right)^2 dt^2 + \left(1 + \frac{M}{2\tilde{r}}\right)^4 (d\tilde{r}^2 + \tilde{r}^2 d\theta^2 + \tilde{r}^2 \sin^2\theta d\varphi^2) \,. \tag{1.15}$$

In this coordinate system, the horizon is located at $\tilde{r}_s = M/2$, and there is no divergence of the metric component g_{rr} at $\tilde{r} = \tilde{r}_s$.[5]

In addition to the existence of the horizon, another important property of the black hole solution is that it contains a spacetime *singularity.* This singularity is a true geometric pathology of the spacetime and is manifested as divergence of certain

[3] For a system with a mass equal to the Sun, the Schwarzschild radius is twice its gravitational length introduced in (1.1), i.e 2.94 km

[4] The coordinate transformation between the isotropic radial coordinate and the radial coordinate associated with (1.14)—known as the *areal coordinate*—is given by: $r = \tilde{r}(1 + M/2\tilde{r})^2$.

[5] Note that the g_{tt} is zero at the horizon, which *is* a coordinate pathology, however this form of the metric can be used for numerical simulations.

(coordinate independent) geometric invariants. [6] For example, for the Schwarzschild metric (1.14) the Kretschmann scalar is given by:

$$I \equiv R_{abcd}R^{abcd} = \frac{48M^2}{r^6} \tag{1.16}$$

and clearly diverges at $r \to 0$. Such singularities can also form from a gravitational collapse process, but whether they can form outside of event horizons in astrophysical scenarios is yet an open problem. The conjecture that an event horizon always forms to hide the singularity from the outside world is known as the *cosmic censorship hypothesis* [22].

The confinement of the singularity by an event horizon has a practical implication for numerical simulations of black hole dynamics or gravitational collapse. Of course, no numerical solver can perform evolutions on a domain that contains a singularity where the dynamical variables diverge. However, the event horizon isolates the observable numerical domain (outside of the black hole) from the singularity by disconnecting the physical characteristics of the equations inside from the outside of the black hole. In another word, no physical effect can be transferred to the outside since no causal curve can emerge from within the horizon. This effectively allows an arbitrary choice of the geometry inside of the black hole. [7] Therefore, assuming that no singularity forms outside of an event horizon, a numerical evolution of a collapse scenario can be carried out from a spacetime with no black hole to a one that contains a black hole which eventually settles to a stationary solution. In effect, cosmic censorship allows us to obtain the spacetime solution for the exterior of the black hole even if we do not have a theory to describe the physics of the singularity.

[6] Another example of a geometric singularity is a point of geodesic incompleteness where the geodesics of spacetime "end" at a finite affine parameter [16]. Spacetime singularity is been proven to always exist inside of a black hole [16].

[7] Under the assumption that the numerical method respects causality, for example the system is evolved via a set of hyperbolic equations.

1.5 Critical Phenomena in Gravitational Collapse

Critical phenomena arise in various systems of matter coupled to Einstein gravity (or pure gravity waves) through the *dynamical construction* of spacetime solutions. The construction starts with initialization of the matter content of the spacetime, some of whose properties are controlled by a parameter p. This parameter controls, for example, the self-gravitational strength of the system, and can be chosen arbitrarily. For instance, p can be the amplitude of a pulse of matter field, or the average velocity of a collection of particles. Naturally, this parameter also labels each of the dynamical spacetime solutions that are constructed by numerically evolving the system. A parameter survey over this family of spacetime solutions is then performed by numerically evolving initial configurations defined by various values of p. The value of p is assumed to vary in an interval with two end points: 1) "weak": corresponding to a sufficiently small value of p that the system's self-gravitation remains weak. The matter typically disperses to infinity leaving flat spacetime behind[8]; 2) "strong": corresponding to a sufficiently large value of p that gravity is strong enough to cause the system to collapse to a black hole. As one might speculate, these two regimes have a point of transition along p, namely the *critical value* $p^\star$. [9] For $p > p^\star$ the final state is a black hole with mass $M_{\rm BH}(p)$, while $p < p^\star$ does not result in black hole formation, i.e. $M_{\rm BH} = 0$. The solution associated with $p = p^\star$ is referred as the *black hole threshold solution* (or critical solution). Operationally, the value $p^\star$ can be found numerically using a binary search[10] and its accuracy is limited by the numerical resolution, and often (for sufficiently high resolution) can be pushed close to the machine precision, $\approx 10^{-16}$. The characteristics of the spacetime and matter configuration at and near the precise threshold value, $p^\star$, comprises

[8] Another scenario is that a bound, stable solution such as a star is formed.

[9] One can imagine another possibility where the corresponding values of p for weak and strong initial data spread over the interval with no defined single boundary. Such scenario will result in a "chaotic" behaviour, where the slightest change in the tuning parameter causes the final state to deviate drastically. The fact that the two regimes are disjoint by a single value is somewhat a non-trivial observation and suggests that gravitational collapse in General Relativity is not chaotic.

[10] We will formally describe this process in Chapter 2 and 3

the main subject of black hole critical phenomena.

In the numerical lab, the characteristics of the critical solution emerge only to a certain extent, i.e. as the *tuning process* (binary search) is performed to drive $p \to p^\star$ more closely, the features of the underlying critical solution can be observed only partially.[11] As will become clear shortly, fully capturing these features numerically (for a certain type of critical solution) requires infinite resolution and imposes severe challenges for a numerical solver. The extreme requirement for the resolution is due to the existence of fine structure in the critical solution. Historically, the first success in numerically exploring the critical regime by Choptuik [4] relied on the implementation of an Adaptive Mesh Refinement (AMR) algorithm that provided the needed resolution.

Black hole critical phenomena can be divided into two broad categories by considering the mass of the final black hole, $M_{\rm BH}$, as a function of the tuning parameter, p. In the weak or *sub-critical* regime, we simply have $M_{\rm BH}(p < p^\star) = 0$, while for $p > p^\star$, namely the *super-critical* regime, there are two possibilities: 1) the mass of the black hole increase continuously from zero as p crosses the value $p^\star$; or 2) the mass function has a finite gap. In analogy with phase transitions in thermodynamical systems, $M_{\rm BH}$ can be viewed as an order parameter and accordingly, the continuous transition is referred as *Type II* critical collapse while the critical collapse with a mass gap is named *Type I.* All of the type II critical solutions discovered so far, and some of the type I solutions, exhibit a common feature: *universality.* This universality is observed in the numerical tuning experiments, where there is unique final configuration (of the spacetime geometry as well as the matter distribution) that is independent of the choice of parameter p and, more generally, the initial configuration of the matter. This observation signals the existence of a unique (or at least isolated in function space) critical solution in the function-space of all so-

[11] For example, only a finite number of echoes (periodic behaviour in a logarithmic radial coordinate) can be observed rather than infinitly many of them that are present in the exact critical solution.

lutions. Moreover, the critical solution is an intermediate attractor point within the sub-function-space labeled by $p = p^\star$. The dynamical-system point of view of these critical phenomena is further discussed in [5, 23]. Beside universality, critical solutions tend to be 1-mode unstable and the unstable eigenvalue of the universal solution, in both types, is related to a measurable exponent in the critical solution as we discuss in the following.

Type II Critical Phenomena

In type II collapse, the mass of the black holes that form is often given by:

$$M_{\rm BH} = |p - p^\star|^\gamma \,, \tag{1.17}$$

The constant γ is known as the *mass-scaling exponent* and is universal for a given matter model, i.e. is independent of the details of the initial data. The mass scaling (1.17) indicates that one can create a black hole of arbitrarily small size by approaching $p \to p^\star$ from the super-critical regime.

Type II critical solutions also exhibit scale-invariance or *self-similarity*. A self-similar solution can be written as a function $Z^\star(\tau, x)$ in which $\tau = -\ln(t - t^\star)$—the logarithm of the proper time measured from the time, $t^\star$, that the critical solution forms, usually referred as the *accumulation* time—x is the scale invariant coordinate, $x = r/(t - t^\star)$, and Z denotes some function of the primary dynamical fields. In this notation, a continuous self-similar (CSS) solution is defined as:

$$Z^\star(\tau, x) = Z^\star(x) \,, \tag{1.18}$$

and a discrete self-similar (DSS) solution is:

$$Z^\star(\tau + \Delta, x) = Z^\star(\tau, x) \,. \tag{1.19}$$

An example of discrete self similarity (DDS) arose in the original work on a massless scalar field collapse by Choptuik. Another way to represent the discrete self similarity of the critical solution in time and spatial coordinates is:

$$Z^\star(\tau, r) = Z^\star(\tau + n\Delta, e^{n\Delta} r)\,, \tag{1.20}$$

in which the notion of the *echoing behaviour* of the solution is clearer. After each echo, $\tau \to \tau + \Delta$, the solution repeats itself on a scale that is e^Δ smaller. In addition, from (1.20), in every snapshot of the solution, $t = \text{const}$, there is a periodic behaviour in the profile of the matter field as a function of $\rho = \ln(r)$, with period Δ. A similar repetitive structure forms in the geometry of spacetime as well. The constant Δ is known as the *echoing exponent* and, as for the scaling exponent, is universal for a specific matter source. This echoing behaviour results in the formation of structure in the solution on ever smaller scales, and as we mentioned in the previous section, requires fine numerical resolution, i.e. adaptive distribution of grid points toward the central collapse region. Finally, we note that as the self-similar solution approaches the accumulation point $t \to t^\star$, or $\tau \to \infty$, in the continuum limit , i.e. for a precisely critical solution $p = p^\star$, the curvature diverges at the radial accumulation point (center of the collapse in symmetric cases) and a naked singularity forms.

Chapter 3 and 4 of this thesis are focused in type II critical collapse studies where we explore the applicability of a popular formulation of Einstein's equation in the context of type II DSS critical collapse scenarios. There, we will revisit type II critical phenomena and discuss the features of the massless scalar field threshold solution.

Type I Critical Phenomena

In critical collapse with a mass gap, $M_{\text{BH}}(p \to p^{\star +}) > 0$, the solution approaches an intermediate state (between collapse and dispersal) that is static (or periodic) and unstable. An initial configuration that is tuned to $p \approx p^\star$ exhibits a *time-scaling* behaviour: the dynamically evolving solution approaches the critical static solution and spends an increasingly large amount of time in the vicinity of it. This time scaling is given by:

$$\tau = -\sigma \ln |p - p^{\star}| \tag{1.21}$$

in which σ is known as *time-scaling exponent*, and similar to the exponents in Type II critical collapse, can be universal for certain matter types. This scaling behaviour suggests that the critical solution possesses a time translation symmetry.

In addition to the dynamical construction of type I/II critical solutions via a tuning process, critical solutions can be constructed directly from an ansatz to the coupled Einstein-matter system that reflects the symmetry of the solution (self-similar or static). In turn, the scaling laws (1.17,1.21) can be understood from the perspective of perturbation analysis [24–26]. The key observation here is that the existence of the universal exponent (σ for type I and γ for type II) can be explained by the existence of only one unstable mode in the critical solution with growth factor (Lyapanov exponent), λ. In fact, the exponent is directly related to the growing mode and is simply equal to its inverse, $1/\lambda$.

Chapter 2 of this thesis is concerned with type I critical collapse in the Einstein-Vlasov system, and answers some unresolved issues concerning this model. Extensive reviews of all the work in type I and II critical collapse can be found in [5, 23] and the more recent reference [3].

1.6 3+1 Formulations of Einstein's Equations

In General Relativity, the evolution of the spacetime geometry is given by a set of second order partial differential equations—as we observed in the approximate linearized form (1.13). Specifically, Einstein's equation can be posed as a standard initial value problem (Cauchy problem): loosely speaking, the solution is expected to be uniquely determined for given "position", i.e g_{ab}, and "velocity", i.e. $\partial_t g_{ab}$ at the initial time. Schematically, the Einstein's equation then determines the second time derivative of the metric, $\partial_t^2 g_{ab}$, which can be integrated forward for the given initial data. To pose the Einstein's equation as a Cauchy problem, one needs a foliation of the spacetime to 3 dimensional space-like surfaces labeled by a time coordinate t. Such process is usually referred as a *3+1 decomposition* of Einstein's equation. In this section we review the classic ADM [27] decomposition and also introduce a recasting of it, the BSSN formulation, that is particularly suitable for numerical computations and that has become the most popular 3+1 formulation in the numerical relativity community. Later in Chapter 3 and 4, a generalization of the BSSN formulation will be introduced and implemented to perform numerical simulations.

We note that the 3+1 casting of Einstein's equations further reveals the internal *constraints* in the field equations. They are not trivially apparent in the covariant form of the Einstein's equations. These equations are not of evolutionary type, but rather constrain the initial data: g_{ab} and $\partial_t g_{ab}$. Therefore the Cauchy problem cannot be posed for a arbitrary initial data; rather a set of elliptic-type constraint equations must be solved at the initial time.

1.6.1 ADM Decomposition

In the 3+1 ADM decomposition of the 4 dimensional spacetime manifold $(M, g_{\mu\nu})$, one assumes that there exist a family of disjoint 3 dimensional space-like hypersurfaces, Σ_t, which can be considered as level surfaces of a scalar function t. The variable t can be interpreted as the global time function and using the gradient of the function t one can define the normal vector, n^a, to the hypersurface Σ_t, as shown in Fig. 1.1:

$$n^a = -\alpha g^{ab} \nabla_b t \,. \tag{1.22}$$

We assume $\nabla_a t$ is non-zero everywhere and α is the normalization factor:

$$||\nabla_a t||^2 = g^{ab} \nabla_a t \nabla_b t \equiv -\frac{1}{\alpha^2} \,, \tag{1.23}$$

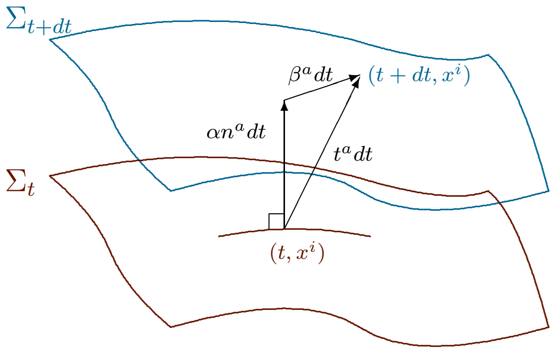


Figure 1.1: Coordinate system in the 3+1 decomposition of Einstein's equation. n^μ denotes the normal vector, while t^a represents the unit coordinate vector along the time coordinate t. These two vectors do not necessarily coincide and their deviation between two hypersurfaces Σ_t and Σ_{t+dt} defines the the shift vector, β^a, which is by construction purely spatial.

and is referred to as the *lapse* function. The space-like surfaces Σ_t, with a given spatial coordinate system x^i on them, constitute the 3+1 slicing of the 4-dimensional manifold M where each point in the spacetime is assigned coordinates (t, x^i). The time coordinate vector, $t^a = \partial/\partial t$, arises from the global time function and can be written as a linear combination of the normal vector, n^a, and a purely spatial vector: β^a:

$$t^a = \alpha n^a + \beta^a \,. \tag{1.24}$$

In this linear combination (also illustrated in Fig. 1.1), the coefficient of n^a is the lapse function, α. This can be easily seen by inserting the definition of n^a (1.22) in the identity: $t^a \nabla_a t = dt/dt = 1$, and using the fact that by definition we choose β^a to be a spatial (i.e. it lives on the hypersurface, Σ_t), therefore:

$$n_a \beta^a = 0 \,. \tag{1.25}$$

The vector β^a is referred as the *shift* vector and can be explicitly written as a 3 dimensional vector:

$$\beta^a = (0, \beta^i)\,, \tag{1.26}$$

in the coordinate system (t, x^i).

Using (1.24), one can compute the components of the normal vector, n^μ, in the coordinate system:

$$n^\mu = \left(\frac{1}{\alpha}, -\frac{1}{\alpha}\beta^i\right)\,. \tag{1.27}$$

Using the normal vector we can, in turn, can build the induced 3-metric on the the hypersurfaces, Σ_t:

$$\gamma_{ab} = g_{ab} + n_a n_b\,. \tag{1.28}$$

The tensor γ_{ab} is purely spatial, (as can be seen simply by verifying that $\gamma_{ab}n^a = 0$) and it measures the distances within the spatial surface Σ_t. In addition, the inner product of any two purely spatial vectors, (v^i, w^j), computed using γ_{ab} is identical to the inner product computed with the 4 dimensional metric, g_{ab}; therefore γ_{ab} is indeed the induced metric. The full spacetime metric, g_{ab}, can be reconstructed using the 3-metric and the lapse and shift:

$$g_{\mu\nu} = \begin{pmatrix} -\alpha^2 + \beta_l\beta^l & \beta_i \\ \beta_j & \gamma_{ij} \end{pmatrix}\,, \tag{1.29}$$

Using the normal vector n^a, one can define the projection tensor:

$$\gamma_a^{\ b} = g_a^{\ b} + n_a n^b = \delta_a^{\ b} + n_a n^b \tag{1.30}$$

which projects 4-vectors to their spatial component on the hypersurface Σ_t. The projection of higher rank tensors into the spatial hypersurfaces can be achieved by contracting each index with the projection operator. This operator is usually denoted by the symbol $\perp$. For example,

$$\perp A_{ab} \equiv \gamma_a^{\ c}\gamma_b^{\ d} A_{cd} \tag{1.31}$$

and $\perp A_{ab}$ is a purely spatial tensor.

One particular projection that characterizes the embedding of the hypersurfaces Σ_t in the spacetime is the projection of the gradient of the normal vector: $\nabla_c n_d$. Specifically, the extrinsic curvature of the surface Σ_t is defined as the projection of the negative gradient of its normal vector:

$$K_{ab} \equiv -\gamma_a^{\ c}\gamma_b^{\ d}\nabla_c n_d = -\frac{1}{2}\mathscr{L}_{\mathbf{n}}\gamma_{ab}\,, \tag{1.32}$$

where $\mathscr{L}$ denotes the Lie derivative. Here the last equality is due to the identity $\mathscr{L}_{\mathbf{n}} g_{ab} = \nabla_a n_b + \nabla_b n_a$ and relation (1.28) between the 3 metric and 4 metric. Using the last identity in (1.32) and the linearity of the Lie derivative:

$$\partial_t = \mathscr{L}_{\mathbf{t}} = \mathscr{L}_{\alpha\mathbf{n}+\beta} = \alpha\mathscr{L}_n + \mathscr{L}_\beta\,, \tag{1.33}$$

one can see that the definition of the extrinsic curvature can be written as:

$$(\partial_t - \mathscr{L}_\beta)\gamma_{ab} = -2\alpha K_{ab}\,. \tag{1.34}$$

Therefore the extrinsic curvature can be seen as generalization of the "time derivative" of the 3 metric γ_{ab}. Using this definition, Einstein's equations (1.4) can be reduced to a set of first order equations where the time evolution of the 3 metric is given by the extrinsic curvature (by definition) while the time evolution of the extrinsic curvature is given by the field equation projected to the spatial hypersurfaces[12].

The energy-momentum tensor on the left hand side of (1.4) can be projected

[12] Combined with constraint equations.

into 3 parts:

$$\rho = n_a n_b T^{ab} \,, \tag{1.35}$$

$$S^a = -\gamma^{ab} n^c T_{cb} \,, \tag{1.36}$$

$$S_{ab} = \gamma_{ac}\gamma_{bd} T^{cd} \,, \tag{1.37}$$

and accordingly, Einstein's equation relates these quantities to the dynamical quantities of 3+1 decomposition: the 3-metric and the extrinsic curvature. The equations can be found by relating the 4 dimensional Riemann tensor R_{abcd} to the 3 dimensional Riemann tensor and the extrinsic curvature of the spatial slice Σ_t. The process results in a set of identities known as Gauss-Codazzi equations. This rather lengthy but straightforward calculation is now part of standard textbooks in numerical relativity and we refer the reader to [28] for details of the derivation. Here, we only describe the results. The Einstein tensor contraction, $n^a n^b G_{ab}$ (1.35) results in scalar equation known as the *Hamiltonian constraint*,

$$\mathcal{H} = R + K^2 - K_{ij}K^{ij} - 16\pi\rho = 0 \,, \tag{1.38}$$

where R is the 3-Ricci scalar associated with the 3-metric γ_{ij} and K is the trace of the extrinsic curvature:

$$K \equiv \gamma_{ij} K^{ij} \,. \tag{1.39}$$

The contraction $\gamma^{ab} n^c G_{cb}$ (1.36) results in a spatial vector equation, known as the *momentum constraint*:

$$\mathcal{M}^i = D_j \left(K^{ij} - \gamma^{ij} K\right) - 8\pi S^i = 0 \,, \tag{1.40}$$

in which D_j denotes the covariant derivative associated with the 3-metric, γ_{ij}. The equations (1.38,1.40) are referred to as constraint equations since they do not contain any second time derivative and only relate the "positions" and "velocities", i.e. g_{ij} and K_{ij} on each spatial slice Σ_t.

Finally the contraction $\gamma_{ac}\gamma_{bd}G^{cd}$ (1.37) provides the equation for the generalized time derivative of the extrinsic curvature, $(\partial_t - \mathscr{L}_\beta)K_{ij}$. Combined with (1.34), the ADM evolution equations for the 3-metric and the extrinsic curvature are:

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i \beta_j + D_j \beta_i \,, \tag{1.41}$$

$$\begin{aligned} \partial_t K_{ij} = -D_i D_j \alpha \quad &+ \quad \alpha \left(R_{ij} - 8\pi S_{ij} + 4\pi (S - \rho)\gamma_{ij} \right) + \alpha \left(K K_{ij} - 2 K_{ik} K^k_j \right) \\ &+ \quad \beta^k \partial_k K_{ij} + K_{ik} \partial_j \beta^k + K_{kj} \partial_i \beta^k \,, \end{aligned} \tag{1.42}$$

in which we have explicitly expanded the Lie derivative terms. Again, D_i denote the covariant derivative associated with the 3-metric γ_{ij} and similarly R_{ij} is the 3-Ricci tensor associated with the 3-metric. This completes the process of casting Einstein's equations as a 3+1 Cauchy problem, where the initial data $\{g_{ij}, K_{ij}\}$ that satisfy the Hamiltonian and momentum constraints can be integrated forward in time using (1.41,1.42) to find the geometry of spacetime. We note that a formal in-depth derivation of 3+1 formulations of Eintein's equation can be found in numerical relativity textbooks such as Baumgarte et al. [28], Gourgoulhon [29] and Alcubierre [30].

Finally, note that as expected, the ADM formulation does *not* provide any equations for the lapse function, α, or the shift vector, β^i, as they represent the freedom of coordinate choice in General Relativity and can be set arbitrarily. However, as we will discuss further in Sec. 1.7, a stable and long term numerical integration of any 3+1 formulation of Einsteins' equation is highly sensitive to the choice of coordinates.

1.6.2 Recasting of ADM Equations: BSSN Formulation

In principle, the standard ADM system, (1.41,1.42), can be used to evolve the 3-metric γ_{ij} and extrinsic curvature K_{ij}. This method is known as a *free evolution*, or an *unconstrained evolution* scheme since the Hamiltonian and momentum constraints are only solved at the initial time. Another approach is to combine some or all of the constraint equations to determine some of the geometric variables at each time. Such methods are known as *partially or fully constrained evolution* schemes.

It turns out that most unconstrained simulations using the ADM equations, especially simulations of gravitational waves, are unstable. In particular, as we will discuss further in Chapter 4, in the case of gravitational waves it appears that the ADM formulation is not even capable of evolving spacetime dynamics in the weak field limit [28]. This can be traced back to the fact that the ADM equations are only weakly hyperbolic (for an extensive discussion on the notion of hyperbolicity and numerical relativity see: [28]) which makes them unsuitable for numerical simulations. This observation provided the original motivation for Shibata and Nakamura [31] to recast the ADM equations in a way that is more applicable for numerical calculations. This recasting was later revisited by Baumgarte and Shapiro [32], and is now known as the Baumgarte-Shapiro-Shibata-Nakamura (BSSN) formulation. The BSSN equations have become immensely popular in the numerical relativity community due to the fact that they are strongly hyperbolic (and are free evolution scheme). This property allows for successful long-time simulations of various strong gravity scenarios, most notably the simulations of compact binaries, including extraction of the gravitational wave-form [13, 14].

The BSSN formulation has 3 key features that differentiate it from the ADM formalism. In the following, we outline the corresponding changes that are made to the ADM formulation that make the resulting equations strongly hyperbolic. The manipulations of the equations in each step is straightforward, if sometimes lengthy, and we refer the reader to [28] for full details.

I) Conformal rescaling

In the BSSN formulation, we split the evolution equations (1.41,1.42) of the ADM variables $\{\gamma_{ij}, K_{ij}\}$ into the evolution of their overall scale and the evolution of their "scale free" part[13]. This is done by a conformal re-scaling of these variables. First one defines the conformal factor, e^{ϕ}, such that:

$$e^{12\phi} \equiv \gamma\,, \tag{1.43}$$

in which γ is the determinant of the 3-metric γ_{ij}. Then, the conformal metric $\tilde{\gamma}_{ij}$ defined as:

$$\gamma_{ij} \equiv e^{4\phi}\tilde{\gamma}_{ij} \;\Rightarrow\; \tilde{\gamma} = 1\,, \tag{1.44}$$

has determinant 1. In addition, the conformally re-scaled extrinsic curvature $\tilde{A}_{ij}$ defined through:

$$K_{ij} \equiv e^{4\phi}\tilde{A}_{ij} + \frac{1}{3}\gamma_{ij}K \;\Rightarrow \text{Trace}(\tilde{A}_{ij}) = \tilde{\gamma}^{ij}\tilde{A}_{ij} = 0\,, \tag{1.45}$$

is trace-free, i.e. scale free[14]. By fixing the determinant of $\tilde{\gamma}_{ij}$ and the trace of $\tilde{A}_{ij}$, the evolution equation for the 3-metric γ_{ij} (1.41) splits into two equations for the overall scale of the metric, ϕ, and the conformal metric $\tilde{\gamma}_{ij}$. Similarly the evolution equation for the extrinsic curvature K_{ij} (1.42) splits into two equations for the trace of the extrinsic curvature, K, (i.e the overall scale of the tensor) and the trace-free part $\tilde{A}_{ij}$. The equations for ϕ and K can be derived by contraction of the two evolution equations (1.41,1.42):

$$(\partial_t - \mathscr{L}_\beta)(6\phi) = (\partial_t - \mathscr{L}_\beta)\ln\gamma^{1/2} = -\alpha K\,, \tag{1.46}$$

[13] This is mainly motivated (heuristically) to separate the gravitational radiation part of the dynamics, encoded in the scale free part of the metric, and the overall gravitational field strength that is approximately determined by the overall scale of the metric, and its main contribution comes from the matter distribution.

[14] Note that the trace, K, is the "conjugate" scale to the determinant, γ. This will become clear shortly in the relation between the evolution of γ and K, i.e. Eq. 1.46.

$$(\partial_t - \mathscr{L}_\beta)K = -D^2\alpha + \alpha\left(R - 4\pi(3\rho - S)\right) \,, \tag{1.47}$$

where we have used the definition of the conformal factor e^{ϕ} (1.43). Here D^2 denotes the operator $\gamma^{ij}D_iD_j$.

The evolution equation for the scale free parts $\tilde{\gamma}_{ij}$ and $\tilde{A}_{ij}$ are then given by the scale free parts of the right hand side of equations (1.41,1.42):

$$(\partial_t - \mathscr{L}_\beta)\tilde{\gamma}_{ij} = -2\alpha\tilde{A}_{ij} \,, \tag{1.48}$$

$$\begin{aligned}(\partial_t - \mathscr{L}_\beta)\tilde{A}_{ij} &= e^{-4\phi}\left[-D_iD_j\alpha + \alpha(R_{ij} - 8\pi S_{ij})\right]^{\mathrm{TF}} \\ &+ \alpha(K\tilde{A}_{ij} - 2\tilde{A}_{il}\tilde{A}^l{}_j) \,, \end{aligned} \tag{1.49}$$

where the superscript $^{\mathrm{TF}}$ denotes the trace-free part with respect to the 3-metric. Specifically, for any rank-2 tensor X_{ij} we have:

$$X_{ij}^{\mathrm{T}F} \equiv X_{ij} - \frac{1}{3}(\gamma^{kl}X_{kl})\gamma_{ij} \tag{1.50}$$

II) Absorbing the Mixed Derivatives

The next step is to eliminate the mixed spatial derivatives appearing in the 3-Ricci tensor in (1.49) by defining a conformal connection function:

$$\tilde{\Gamma}^k = \tilde{\gamma}^{ij}\tilde{\Gamma}^k_{ij} \,, \tag{1.51}$$

as a new dynamical variable. That is, $\tilde{\Gamma}^k$, is to be evolved rather than being computed in terms of $\tilde{\gamma}_{ij}$. The evolution equation for the conformal connection can be derived from its definition and the time evolution of the conformal metric (1.48), yielding:

$$(\partial_t - \mathscr{L}_\beta)\tilde{\Gamma}^i = -2\tilde{A}^{ij}\partial_j\alpha - 2\alpha\partial_j\tilde{A}^{ij} + \frac{1}{3}\tilde{\gamma}^{li}\partial_l\partial_j\beta^j + \tilde{\gamma}^{lj}\partial_j\partial_l\beta^i \,. \tag{1.52}$$

Using the conformal re-scaling of the 3-metric, and the definition of the conformal connection, $\tilde{\Gamma}^i$, the 3-Ricci tensor is divided into two parts:

$$R_{ij} = R^{\phi}_{ij} + \tilde{R}_{ij} \,. \tag{1.53}$$

The first term, R^{ϕ}, is associated with the overall conformal factor and is given by

$$R^{\phi}_{ij} = -2\tilde{D}_i\tilde{D}_j\phi - 2\tilde{\gamma}_{ij}\tilde{D}^k\tilde{D}_k\phi + 4\tilde{D}_i\phi\tilde{D}_j\phi - 4\tilde{\gamma}_{ij}\tilde{D}^k\phi\tilde{D}_k\phi \,, \tag{1.54}$$

where $\tilde{D}_i$ is the covariant derivative associated with the conformal metric $\tilde{\gamma}_{ij}$. The second term in (1.53), $\tilde{R}_{ij}$, is the 3-Ricci tensor associated with the conformal metric $\tilde{\gamma}_{ij}$:

$$\tilde{R}_{ij} = -\frac{1}{2}\tilde{\gamma}^{lm}\partial_m\partial_l\tilde{\gamma}_{ij} + \tilde{\gamma}_{k(i}\partial_{j)}\tilde{\Gamma}^k + \tilde{\Gamma}^k\tilde{\Gamma}_{(ij)k} + \tilde{\gamma}^{lm}\left(2\tilde{\Gamma}^k_{l(i}\tilde{\Gamma}_{j)km} + \tilde{\Gamma}^k_{im}\tilde{\Gamma}_{klj}\right) \,. \tag{1.55}$$

where the conformal factor $\tilde{\Gamma}^k$ should be substituted by its *evolved* value via the evolution equation (1.52). As can be seen in (1.55), this process eliminated the mixed spatial derivative terms (which could spoil the hyperbolicity) in $\tilde{R}_{ij}$ By this substitution, the principle part of the conformal 3-Ricci tensor is $\tilde{\gamma}^{lm}\partial_m\partial_l\tilde{\gamma}_{ij}$, which is a wave-like spatial derivative operator.

III) Adding the Constraints to the Evolution Equations

Finally, in the BSSN formulation, both the Hamiltonian and momentum constraints are added to the evolution equations, creating a natural constraint damping feature to the free evolution equations. The Hamiltonian constraint is added to the evolution equation of the extrinsic curvature with coefficient $-\alpha$, (the lapse function):

$$(\partial_t - \mathscr{L}_\beta)K = -D^2\alpha + \alpha\left(R - 4\pi(3\rho - S)\right) - \alpha\mathcal{H}\,, \tag{1.56}$$

and the momentum constraint is added to the evolution of the conformal connection, with a coefficient 2α [15]:

$$(\partial_t - \mathscr{L}_\beta)\tilde{\Gamma}^i = -2\tilde{A}^{ij}\partial_j\alpha - 2\alpha\partial_j\tilde{A}^{ij} + \frac{1}{3}\tilde{\gamma}^{li}\partial_l\partial_j\beta^j + \tilde{\gamma}^{lj}\partial_j\partial_l\beta^i + 2\alpha\mathcal{M}^i\,. \tag{1.57}$$

Adding the constraints [16] and expanding the Lie derivatives [17] results in the final explicit form of the BSSN equations summarized as following:

Summary of BSSN equations:

$$\partial_t\phi = -\frac{1}{6}\alpha K + \beta^i\partial_i\phi + \frac{1}{6}\partial_i\beta^i\,, \tag{1.58}$$

$$\partial_t\tilde{\gamma}_{ij} = -2\alpha\tilde{A}_{ij} + \beta^k\partial_k\tilde{\gamma}_{ij} + \tilde{\gamma}_{ik}\partial_j\beta^k + \tilde{\gamma}_{kj}\partial_i\beta^k - \frac{2}{3}\tilde{\gamma}_{ij}\partial_k\beta^k\,, \tag{1.59}$$

$$\partial_t K = -\gamma^{ij}D_jD_i\alpha + \alpha(\tilde{A}^{ij}\tilde{A}_{ij} + \frac{1}{3}K^2) + 4\pi\alpha(\rho + S) + \beta^i\partial_i K\,, \tag{1.60}$$

$$\begin{aligned}\partial_t\tilde{A}_{ij} &= e^{-4\phi}\left[-D_iD_j\alpha + \alpha(R_{ij} - 8\pi S_{ij})\right]^{\mathrm{TF}}\\ &+ \alpha(K\tilde{A}_{ij} - 2\tilde{A}_{il}\tilde{A}^l{}_j)\\ &+ \beta^k\partial_k\tilde{A}_{ij} + \tilde{A}_{ik}\partial_j\beta^k + \tilde{A}_{kj}\partial_i\beta^k - \frac{2}{3}\tilde{A}_{ij}\partial_k\beta^k\,,\end{aligned} \tag{1.61}$$

[15] The choice can be $\xi\alpha$ where ξ must be greater than 1/2, but otherwise arbitrary. Choosing $\xi = 2$ leads to simpler equations.

[16] and some algebra to express constraints equations in terms of the BSSN variables.

[17] One should take into account that $\tilde{\gamma}_{ij}$ and $\tilde{A}_{ij}$ are tensor densities, ϕ is not a true scalar, but a scalar density related to the determinant of the metric, and similarly $\tilde{\Gamma}^i$ is a vector density. See [28].

$$\begin{aligned}\partial_t\tilde{\Gamma}^i &= -2\tilde{A}^{ij}\partial_j\alpha + 2\alpha\left(\tilde{\Gamma}^i_{jk}\tilde{A}^{kj} - \frac{2}{3}\tilde{\gamma}^{ij}\partial_j K + 6\tilde{A}^{ij}\partial_j\phi - 8\pi\tilde{\gamma}^{ij}S_j\right) \\ &+ \beta^j\partial_j\tilde{\Gamma}^i - \tilde{\Gamma}^j\partial_j\beta^i + \frac{2}{3}\tilde{\Gamma}^i\partial_j\beta^j + \frac{1}{3}\tilde{\gamma}^{li}\partial_l\partial_j\beta^j + \tilde{\gamma}^{lj}\partial_j\partial_l\beta^i\,. \qquad (1.62)\end{aligned}$$

In terms of the BSSN variables, the constraint equations are given by:

$$\mathcal{H} \equiv \tilde{\gamma}^{ij}\tilde{D}_i\tilde{D}_j e^{\phi} - \frac{e^{\phi}}{8}\tilde{R} + \frac{e^{5\phi}}{8}\tilde{A}^{ij}\tilde{A}_{ij} - \frac{e^{5\phi}}{12}K^2 + 2\pi e^{5\phi}\rho = 0\,, \qquad (1.63)$$

$$\mathcal{M}^i \equiv \tilde{D}_j\left(e^{6\phi}\tilde{A}^{ji}\right) - \frac{2}{3}e^{6\phi}\tilde{D}^i K - 8\pi e^{6\phi}S^i = 0\,. \qquad (1.64)$$

The above BSSN equations yield a well-posed Cauchy problem for Einstein's equations. If the initial data:

$$(\phi, \tilde{\gamma}_{ij}, K, \tilde{A}_{ij}, \rho, S^i, S_{ij})|_{t=0}\,, \qquad (1.65)$$

satisfies the Hamiltonian and momentum constraints[18] (1.63,1.64), and $\tilde{\Gamma}^i$ is initialized by (1.51), then experience has shown that the system can be evolved via the BSSN evolution equations (1.58-1.62) stably and for a long time. The BSSN formulation does not require use of the constraint equations (apart from determining the initial data), and therefore is a fully evolutionary formulation of the Einstein's equations.

[18] Numerically, the initial data satisfies these equations only approximately. To have a consistent set of initial data and evolution system, the error in the initial data is expected to be as small as the truncation error (the error due to the finite difference approximation of the continuum equations) of the evolution scheme.

1.7 Coordinate Choices

The coordinate freedom in General Relativity is encoded in the choice of the lapse function α and the shift vector β^i. Intuitively, the lapse function defines the shape of the embedding of the 3-dimensional hypersurfaces Σ_t within the spacetime, while the shift vector—that is defined as the deviation of the spatial coordinate from the normal direction to the hypersurfaces (Fig. 1.1)—determines the dynamics of the spatial coordinates relative to normal propagation. Choosing good forms for these functions has been a significant part of the research work in the field of numerical relativity. In fact, finding a *dynamical coordinate choice* (a notion that will be introduced shortly) that allows long-time and stable evolution of a strongly gravitating system is not a trivial task.

Two scenarios that pose particular challenges for the coordinate choices are the evolution of a spacetime that contains a black hole or the evolution of a spacetime containing matter collapsing to a black hole. As can be seen from the Schwarzschild solution (1.14), a specific coordinate choice can be singular (not necessarily only at the event horizon) where the singularity is not physical and can be removed by a coordinate transformation. A more challenging case is a physical singularity (point of infinite curvature) that can form, for example, at the center of symmetry in a collapse scenario. Obviously, a numerical code cannot continue execution if it encounters such a singular point during a simulation. A particular type of coordinate choice is necessary to appropriately deal with this scenario and is known as *singularity avoiding*. Intuitively, the lapse function is chosen so that it slows the evolution in the vicinity of a singularity.

Even if the spacetime does not contain a black hole or is not evolving toward the formation of one, the coordinate choice can still require special attention, since a (non-physical) coordinate singularity can form. This can be demonstrated using the simplest gauge choice:

$$\alpha = 1\,,\ \beta^i = 0\,, \tag{1.66}$$

which are known as *Gaussian-normal* coordinates, and which is a particular case of a coordinate system that incorporates geodesic slicing. Due to the vanishing shift vector, the coordinate time vector coincides with the normal vector, and due to the choice $\alpha = 1$, the observers moving along the direction normal to the hypersurfaces measure coordinate time which is equal to proper time. The acceleration of the normal observers is given by $a_b = D_b\alpha = 0$; hence $x^i =$ cons. trajectories are indeed geodesics. In this gauge choice, a coordinate pathology can develop quickly as can be seen from the evolution equations of the extrinsic curvature and the determinant of the 3-metric. In the Gaussian-normal system the equations (1.46) and (1.60) in vacuum are given by:

$$\partial_t \ln \gamma^{1/2} = -K \,, \tag{1.67}$$

$$\partial_t K = \tilde{A}^{ij}\tilde{A}_{ij} + \frac{1}{3}K^2 > 0 \,. \tag{1.68}$$

From the second equation (1.68) K grows monotonically in time, while from (1.67) the volume element of the 3-surfaces decays as the extrinsic curvature becomes unbounded (due to the K^2 term in (1.68)). Intuitively, this behaviour can be seen from the definition of the extrinsic curvature as the negative of the expansion of the normal direction—which in geodesic slicing is equal to the convergence (negative expansion) of the geodesics. The development of such coordinate singularities is not surprising as the geodesics can focus toward each other—for example from a gravitational radiation passing through a region of spacetime—and nothing prevents them from crossing.

Motivated by this geometric observation, one well-known coordinate choice is developed by imposing a condition that prevents the normals from converging. From the definition of the extrinsic curvature, this can be achieved by requiring

$$K = 0 \,. \tag{1.69}$$

This coordinate choice is known as the *maximal slicing* condition. Provided $K = 0$ at $t = 0$, the evolution equation of the extrinsic curvature (1.60) requires

$$0 = -\gamma^{ij} D_j D_i \alpha + \alpha(K_{ij} K^{ij} + 4\pi(\rho + s)) \,, \tag{1.70}$$

for all subsequent t in order for the maximal slicing condition to hold. This last equation can be solved as an elliptic equation for the lapse function.

Imposing a specific constraint, such as (1.69), on some of the variables of the ADM formulation, is one of the common approaches to define coordinates, and will typically create constraint-type equations for the gauge variables α and β^i. For the shift vector, one example of this type of conditions is the *quasi-isotropic* choice where one requires the 3-metric to remain diagonal. Assuming the 3-metric is diagonal at $t = 0$, this can be accomplished by requiring that the time derivative of the off-diagonal 3-metric components be 0 for all $t > 0$. This demand then yields

$$0 = -2\alpha K_{ij} + D_i \beta_j + D_j \beta_i \,, i \neq j \,, \tag{1.71}$$

which can be solved as a boundary value problem for the shift vector. We note that in general there are 3 degrees of freedom to choose the spatial coordinate on the 3-hypersurfaces, which can be used to fix the 3 free variables in the shift vector. However, for certain symmetric metrics, conditions such as (1.71) may not result in a sufficient number of equations, and extra algebraic condition may be needed in conjunction with them to completely fix the shift vector [28].

Another approach to specify the coordinate system, which is the main choice in Chapter 3 and 4 of this thesis, is by defining evolution equations for α and β^i. The constraint-type gauge choices, such as maximal slicing, usually result in a set of complex elliptic equations that needs to be solved on every time slice. Implementing an effective elliptic-solver for such equations is a challenging computational task. In contrast, if one can prescribe *evolution* equations for the coordinate system, then it is quite straight-forward to integrate them forward in time along with the evolution equations for the dynamical variables.

One particular dynamical slicing condition that has proven to be robust and which has the singularity avoidance property is the following

$$\partial_t \alpha = -2\alpha K \, . \tag{1.72}$$

This coordinate choice is known as the *1+log*[19] condition. It was implemented in the original BSSN work and now is considered its standard lapse choice. The 1+log condition can be viewed as a "K-damping" evolution equation, in the sense that the evolution of the lapse effectively damps the growth of K in the coupled system of PDEs for K and α. Overall, the 1+log gauge has been shown to mimic the maximal slicing condition (but requires much less computational effort to implement) and in particular has the singularity avoidance property[20].

We now want to consider conditions for the shift vector that are similarly dynamical. First, looking at the BSSN equations (1.58-1.62), we see that the system simplifies if the conformal connection $\tilde{\Gamma}^i$ can be set to zero. However such a condition, if implemented exactly, will lead to an elliptic type equation for the shift (known as *Gamma-freezing*) and may in fact spoil the hyperbolicity of the BSSN system. Taking a similar approach to 1+log slicing, which creates a natural K-damping system, as opposed to exactly enforcing $K = 0$, one can impose the following evolution equation for the shift vector:

$$\partial_t \beta^i = \mu \tilde{\Gamma}^i - \eta \beta^i \, . \tag{1.73}$$

This is known as the *Gamma-driver* shift condition. Here, μ and η are adjustable

[19] The name "1+log" is due to the fact that equation (1.72) combined with (1.58) and zero shift implies: $\partial_t \alpha - 12\partial_t \phi = 0$ which can be solved by assuming $\alpha = 1 + 12\phi$ and using the definition of ϕ (1.43) this condition is equivalent to: $\alpha = 1 + \ln(\gamma)$—hence the name: 1+log.

[20] We also note that the maximal slicing condition *cannot* be used in the BSSN formulation, as it spoils the hyperbolicity of the equations, making the resulting system ill-posed.

numerical parameters. μ is usually chosen to be 3/4 [21], while η has the unit of inverse of time, and since the time scale of the system is usually set by the total mass of the system, η is chosen to be of order $1/M$ where M is the total mass of the spacetime (ADM mass). Again, as with the 1+log choice, the Gamma-driver condition effectively creates a damping term and controls the value of the conformal factor $\tilde{\Gamma}^i$. Together, the 1+log and Gamma-drive conditions have proven to be successful in dealing with challenging problems in numerical relativity, such as binary black hole coalescence. These two coordinate choices are used in Chapter 3 and 4 of this thesis, where they are implemented to study type II critical collapse scenarios.

1.8 Overview of Numerical Techniques for Time Dependent Problems

When Einstein's field equations are decomposed through a 3+1 formulation, they become a set of time dependent PDEs. A standard approach to numerically solve PDEs is the *finite difference methods*, where continuum functions are discretized:

$$f(t,\vec{X}) \to f(t^n, x_i, y_j, \cdots) \equiv f^n_{ij\cdots} \equiv f^n_I \,, \tag{1.74}$$

and differential operators are replaced with difference operators. For example:

$$\frac{\partial}{\partial X} f(X) \to \left(\frac{E+E^{-1}}{2\Delta X}\right)(f_I) \equiv \frac{f_{I+1}-f_{I-1}}{2\Delta X} \,. \tag{1.75}$$

In (1.74) and (1.75) t^n denotes discrete values of time $t^n = t_0 + n\Delta t$, X_I symbolizes a mesh along one of the spatial coordinates: $X_I = X_{\min} + I\Delta X$ (for example: X_I can be $x_i = x_{\min} + i\Delta x$), ΔX and Δt are the step sizes of the discretization in t

[21] This particular value has been found from purely numerical experiments and is known to perform well.

and X and E denotes the shift operator along X:

$$E(f_I) \equiv f_{I+1}\,, \tag{1.76}$$

$$E^{-1}(f_I) = f_{I-1}\,, \tag{1.77}$$

and is the fundamental operator that creates all the difference operators. Operations similar to (1.75) can be defined to replace time derivatives with finite difference expressions. The key assumption here is the smoothness of the function that allows Taylor expansion. For instance, the Taylor expansion of the RHS of (1.75) yields

$$\frac{f_{i+1}+f_{i-1}}{2\Delta x} = \frac{df}{dx} + \frac{1}{6}\frac{d^3f}{dx^3}(\Delta x)^2 + \frac{1}{120}\frac{d^5f}{dx^5}(\Delta x)^4 + \cdots = \frac{df}{dx} + O(\Delta x^2) \approx \frac{df}{dx}\,, \tag{1.78}$$

where we are using big-O notation—$O(\Delta x^2)$ is a function that converges to zero as fast as Δx^2. Neglecting this term, the LHS of 1.78 becomes a *Finite Difference Approximation* (FDA) to the differential operator d/dx. The $O(\Delta x^2)$ term is usually referred as the *truncation error* of the FDA.

In a nutshell, the discretization process converts a PDE to a finite difference equation which is an algebraic equation that can be solved numerically on a computer. However, in practice, this process has several complications, including finding the FDA operators for derivatives with the correct accuracy, handling boundary points, initialization, developing testing facilities and generating solver routines. We developed a Maple based toolkit called FD that simplifies these steps while allowing full control over the entire process while helping the user to focus on the underlying physical/mathematical phenomena described by the PDE. This toolkit is a set of Maple procedures and definitions that provides a high level language to specify a PDE over a discretized numerical domain and solve it. It can compute the finite difference approximation (FDA) equivalent of a PDE and generate low level language (Fortran) routines and C wrappers that evaluate the FDA expression or solve it for the dynamical (unknown) field. FD also allows a rapid prototyping work-flow to create the diagnostic facilities used in finite difference methods, and generates routines that are parallel ready that can be used within a framework of a parallel computing infrastructure such as PAMR [33].

The Appendix of this thesis, in large part, is the user manual for this software. It also discusses the details and key concepts of the finite difference method as well as the mathematical notion of *convergence* and *independent residual evaluators* that are used throughout this thesis as diagnostic tools. At any point, the materials in the Appendix can be consulted as a pedagogical reference.

1.9 Outline of the Thesis

As mentioned above, this thesis is concerned with both Type I and Type II critical collapse. In Chapter 2, we study the Einstein-Vlasov model which describes a set of collision-less particles modeled as a phase-space distribution coupled to Einstein's equations. This study is to address some of the inconsistencies in the type I critical collapse studies of Einstein-Vlasov system, and in particular to focus on the massless[22] system and understand the role of the angular momentum. We directly integrate the Vlasov equation in phase-space, i.e. we evolve the distribution $f(t, r, p_r, l^2)$ where f is the density of the particles in phase-space, r is the radial coordinate, p_r is the radial momentum of the particles and l^2 is the angular momentum of the particles. The geometry and phase-space distribution are restricted to spherical symmetry, however the dynamics of f in the phase-space is indeed a 3 + 1 computation, 3 phase-space dimensions (r, p_r, l^2) and 1 time coordinate. The main finding in this section is the observation of type I critical behaviour and the universality of the time-scaling exponent in the massless system. In addition, we find a family of static solutions to the massless system and show that they all can play the role of type I critical solutions with similar time-scaling exponent when they are perturbed. This chapter also explains the numerical techniques and the

[22] Particles are moving along the null geodesics of the spacetime.

new finite-volume code we developed to solve the Vlasov part of the problem. The chapter is identical to its published version [1].

Chapter 3 and 4 are focused on type II critical collapse. In particular, we are interested in applying a modified version of the BSSN formulation, known as Generalized BSSN, to type II critical solutions and ultimately to develop an axisymmetric code that can be used in various type II critical phenomena studies. The ultimate goal is to extend the limited studies in the numerical relativity literature in type II critical collapse and in particular to find an appropriate formulation and coordinates choice that allow generic type II critical phenomena studies among which the critical collapse of pure gravitational waves is yet an unresolved problem.

In Chapter 3, we begin with adopting the BSSN formulation in spherical symmetry and apply the technique to the well-known problem of critical collapse of a massless scalar field. As will be discussed, the use of free evolution schemes and dynamical coordinate choices has not been successful in the past in resolving the discrete self similarity of type II threshold solutions. The main result of Chapter 3 is the first successful implementation of a hyperbolic formulation that is capable of evolving the spacetime dynamics sufficiently close to the critical solution to allow observation of the characteristics of the DSS spacetime. Our results establish a potential route to extend the type II critical phenomena studies in axial symmetry using the G-BSSN formulation. This chapter is also identical to its published version [2].

Chapter 4 extends techniques we implemented in spherical case to axial symmetry. We describe the implementation of a new generalized BSSN axisymmetric code that uses cylindrical coordinates and provide evidence confirming its robustness and accuracy. The code can in principal be coupled to any matter sources. However, we demonstrate the performance of the code in the strong pure gravitational waves content, which has been historically the most challenging case. We evolve highly non-linear gravitational waves in axial symmetry where the vacuum can collapse to a black hole. Our primary calculations suggest that, again, generalized BSSN appears to be promising, and an extension of the work can shed more insight, and perhaps solve the as yet unresolved problem of critical collapse in pure gravitational waves.

Chapter 2

Critical Collapse in the Spherically Symmetric Einstein - Vlasov Model

2.1 Introduction

In this paper[23] we report results from an investigation of critical collapse in the spherically symmetric Einstein-Vlasov system, which describes the interaction of collisionless matter with a general relativistic gravitational field. After more than two decades of study, the field of black hole critical phenomena has matured and although we present a brief overview below, we assume that the reader is at least somewhat familiar with the key concepts and results in the subject: those who are not can consult comprehensive review articles [5, 23].

We recall that critical phenomena can be identified in a given model by considering dynamical evolution of initial data that is characterized by a parameter, p, such that for sufficiently small p the gravitational interaction remains weak and the matter (or gravitational energy in the vacuum case) typically disperses, while for sufficiently large p a black hole forms. By tuning p between these limits we isolate a critical parameter value $p^\star$ that generates a solution representing the threshold of black hole formation for the particular family of initial data. The behaviour that

[23] This chapter is published in: Akbarian. A. and Choptuik M. W. "Critical collapse in the spherically-symmetric Einstein-Vlasov model. *Phys. Rev.* D90, 104023 (2014).

arises in the near-critical regime $p \to p^\star$ constitutes what is meant by black hole critical phenomena. Depending on the particulars of the model, these phenomena will comprise one or more of the following: 1) existence of a special solution at criticality with possible universality with respect to the parameterization of the initial data, 2) symmetry of the critical solution beyond any imposed in the model itself and 3) scaling of dimensionful physical quantities as a function of $|p - p^\star|$, with scaling exponents which may also be universal in the sense given above. These properties can largely be explained by observing that a critical solution has a single unstable mode in perturbation theory, whose associated eigenvalue (Lyapunov exponent) can be immediately related to the empirically measured scaling exponent.

For the most part, the critical transitions that have been observed to date fall into two classes that are dubbed type I and type II in analogy with first and second order phase transitions, respectively, in statistical mechanical systems, and where the behaviour of the black hole mass plays the role of an order parameter. A type I transition is characterized by a static or periodic critical solution, with a scaling law

$$\tau = -\sigma \ln |p - p^\star| \, . \tag{2.1}$$

Here, τ is the lifetime of the near-critical configuration—the amount of time that the dynamical configuration is closely approximated by the precisely critical solution—and the scaling exponent, σ, is the reciprocal of the Lyapunov exponent, λ, associated with the solution's single unstable mode. In this case the black hole mass is *finite* at threshold since when the marginally stable static or periodic solution collapses, most of its mass-energy will end up inside the horizon.

Previous studies [34–37] have strongly suggested that the critical behaviour in the Einstein-Vlasov model is generically type I and our current results bear this out. So far as we know, type II collapse, where the critical solution is self similar and the black hole mass is infinitesimal at threshold, is not relevant to the model and will not be considered here.

In the Einstein-Vlasov system the matter content of spacetime is specified by a density function $f(t, x^i, p_j)$ in phase space whose evolution is given by the Vlasov equation, while the geometry is governed by the Einstein equations. Numerical studies of the model have a long history, dating back to the work by Shapiro and Teukolsky, both in spherical symmetry [38–40] and axisymmetry [41, 42]. Investigation of critical collapse in the spherically symmetric sector was initiated by Rein et al [34] who observed finite black hole masses at threshold for all families considered. Subsequent work by Olabarrieta and Choptuik [35] corroborated these findings and additionally provided evidence that the threshold solutions were static with lifetime scaling of the form (2.1). Moreover, there were some indications in this latter study that there might be a universal critical solution and associated scaling exponent.

More recently, Andréasson and Rein have carried out a comprehensive study of precisely static solutions of the model, concentrating on their stability both generally and in the context of critical phenomena [37, 43]. Many of their observations and results are pertinent to our current investigation. First, they point out that static solutions can be constructed via a specific ansatz for the distribution function that is discussed in Sec. 2.3. Second, using this ansatz they construct parameterized sequences of static solutions, and, following astrophysical practice, characterize the solutions by their central redshifts and binding energies. Third, they present strong evidence that a maximum in the binding energy along a sequence signals an onset of instability and that at least some of the configurations that lie along an unstable branch can act as type I solutions in the critical collapse context. This immediately establishes that there can *not* be universality in the model. Fourth, and finally, they show that dispersal is not the only stable end state of sub-critical collapse, but that relaxation to a bound state is also possible, contingent on the sign of the binding energy. Overall, the picture of critical behaviour that emerges very much parallels that which is observed for type I transitions in the perfect-fluid and massive-scalar cases [44–51].

All of the work reviewed above used a non-zero particle mass. However, the massless case can also be considered and the current research is largely aimed at exploration of that sector. Additionally, we attempt to address some issues that remained open following Andréasson and Rein's work, including whether there is any explanation for the indications of universality seen in [35]. We note that for the massless model Martin-Garcia and Gundlach [52] considered the possibility of the existence of one-mode unstable self similar configurations that could serve as type II critical solutions. Interestingly, they concluded that since there are infinitely many matter configurations that give rise to any given static spacetime, any unstable solution must have an *infinite* number of unstable modes. Their argument also applies to the static case, which then suggests that there should be no type I behaviour in the model either.

In spherical symmetry the Vlasov equation is a PDE in time and three phase space dimensions.[24] Thus, direct numerical solution is costly and this fact motivated the use of particle-based algorithms in all previous studies excepting [36]. However, a key deficiency of particle approaches is that the results develop a stochastic character on a short time scale. This leads to poor convergence properties relative to a direct method, namely an error that is only $O(1/\sqrt{N})$, where N is the number of particles. With the substantial increase in computational resources over time, direct solution techniques have become feasible and about a decade ago Stevenson [36] implemented a finite-volume solver for the Vlasov PDE for the case that all particles have the same angular momentum. The code that we have developed is largely a continuation of his effort and produces results that have well-behaved convergence properties as a function of the mesh spacing.

Our numerical studies are based on two types of initial data. The first, which we term *generic*, is characterized by a relatively arbitrary functional form for $f(0, x^i, p_j)$. The second, which we call *near static*, is based on perturbations about some precisely

[24] The 3 phase space dimensions are: radial direction r, radial momentum p_r and the angular momentum l (see Sec. 2.2.2).

static solution that is constructed from the ansatz described in Sec. 2.3. We perform experiments using initial conditions of the first type for both massless and massive particles, but restrict attention to the massless sector for our near-static studies. Aiming to unearth as much phenomenology as possible, as well as to explore the issue of universality, we have attempted to broadly survey the possibilities for the specific form of the initial distribution function in all three sets of experiments.

The remainder of the paper is structured as follows. The next section describes the equations of motion for the model while Sec. 2.3 discusses the construction of static solutions from the ansatz mentioned previously. Sec. 2.4 details our numerical approach, including code validation. Sec. 2.5 is devoted to the main results from our study and we conclude with a summary and discussion in Sec. 2.6. We have adopted units in which $G = c = 1$.

2.2 Equations of Motion

A configuration of a system of particles can be described by the phase space density, $f(t, x^i, p_j)$, also known as the distribution function, where x^i and p_j are the particles' spatial positions and 3-momenta, respectively. In the Einstein-Vlasov system particles interact only through gravity. Consequently, the particles move on geodesics of the spacetime along which the density function is conserved[25]:

$$\frac{Df(t, x^j, p_j)}{d\tau} = 0\,. \tag{2.2}$$

Here, τ is the proper time of the particle and $D/d\tau$ is the Liouville operator:

$$\frac{D}{d\tau} \equiv \frac{dx^\mu}{d\tau}\frac{\partial}{\partial x^\mu} + \frac{dp_j}{d\tau}\frac{\partial}{\partial p_j}\,. \tag{2.3}$$

[25] This can be viewed as the conservation of the particles number in the volume element of the phase space (co-moving with the particles) since we assume no collision between the particles. An introduction to Einstein-Vlasov is given in [53] and an extensive textbook in relativistic Boltzmann equation is [54].

Using the geodesic equation

$$v^\mu \partial_\mu p_\nu - v^\mu \Gamma^\lambda_{\mu\nu} p_\lambda = 0\,, \tag{2.4}$$

where v^μ is the particle 4-velocity, the Vlasov equation can be written as

$$p^\mu \frac{\partial f}{\partial x^\mu} + p^\nu p_\lambda \Gamma^\lambda_{\nu j} \frac{\partial f}{\partial p_j} = 0\,. \tag{2.5}$$

The energy momentum tensor of the system is given by integrating over the momentum of the particles:

$$T_{\mu\nu}(t, x^i) = \int \frac{p_\mu p_\nu}{m} f(t, x^i, p_j) dV_{p_j}\,, \tag{2.6}$$

where m is the particle mass. Equations (2.5) and (2.6), together with Einstein's equations

$$G_{\mu\nu} = 8\pi T_{\mu\nu}\,, \tag{2.7}$$

govern the evolution of the Einstein-Vlasov system. These equations, restricted to spherical symmetry by requiring $f(t, x^i, p^j) = f(t, R(x^i), R(p^j))$, $R \in SO(3)$ is the system we study numerically.

2.2.1 Coordinate Choice and Equations for Metric Components

We adopt polar-areal coordinates (t, r) in which the spherically-symmetric metric takes the form

$$ds^2 = -\alpha(t,r)^2 dt^2 + a(t,r)^2 dr^2 + r^2 d\theta^2 + r^2 \sin^2\theta d\phi^2\,. \tag{2.8}$$

The radial metric function $a(t, r)$ can be determined from either the Hamiltonian constraint,

$$\frac{a'}{a} = \frac{1-a^2}{2r} - \frac{ra^2}{2} 8\pi T^t{}_t\,, \tag{2.9}$$

where $' \equiv \partial/\partial r$, or from the momentum constraint,

$$\frac{\dot{a}}{a} = \frac{ra^2}{2} 8\pi T^r{}_t \,, \tag{2.10}$$

with $\dot{} \equiv \partial/\partial t$. The lapse function $\alpha(t,r)$ is fixed by the polar slicing-condition

$$\frac{\alpha'}{\alpha} = \frac{a^2-1}{2r} + \frac{ra^2}{2} 8\pi T^r{}_r \,. \tag{2.11}$$

Equation (2.9) is solved subject to the boundary condition,

$$a(t,0) = 1 \,, \tag{2.12}$$

which follows from the demand of elementary flatness at the origin. For the lapse we set

$$\alpha(t, r_{\max}) = \frac{1}{a(t, r_{\max})} \,, \tag{2.13}$$

where $r_{\max}$ is the location of the outer boundary of the computational domain, so that coordinate and proper time coincide at infinity.

The $\theta\theta$ component of Einstein's equation yields an additional redundant equation, and we use the degree to which it is satisfied as a check of our numerical results.

2.2.2 The Energy Momentum Tensor

As noted above, for a given distribution function, $f(t, x^i, p_j)$, the stress tensor is computed from the momentum-space integral (2.6). With our choice of metric the volume element is given by

$$dV_{p_j} = \frac{m d^3 p_j}{p^0 \sqrt{|g|}} = \frac{m dp_r dp_\theta dp_\phi}{p^0 \alpha a r^2 \sin\theta} \,. \tag{2.14}$$

To impose spherical symmetry we require the distribution function to be uniform in all possible angular directions. This condition can be conveniently implemented by transforming to variables l^2 and ψ given by

$$l^2 \equiv p_\theta^2 + \frac{p_\phi^2}{\sin^2\theta}\,, \tag{2.15}$$

$$\psi \equiv \tan^{-1}\left(\frac{p_\theta \sin\theta}{p_\phi}\right)\,, \tag{2.16}$$

where l is the angular momentum of the particles. Spherical symmetry is then achieved by demanding that $f(t, x^i, p_r, l^2, \psi) \equiv f(t, r, \theta, \phi, p_r, l^2, \psi) = f(t, r, p_r, l^2)$. The volume element in the new variables is

$$dV_{p_j} = \frac{m dp_r dl^2 d\psi}{2a\bar{p}^t r^2}\,, \tag{2.17}$$

where

$$\bar{p}^t \equiv \alpha p^0 = \sqrt{m^2 + \frac{p_r^2}{a^2} + \frac{l^2}{r^2}}\,. \tag{2.18}$$

Integrating over ψ, the components of the energy momentum tensor are given by:

$$T^t{}_t = \frac{-\pi}{ar^2}\iint \bar{p}^t f dp_r dl^2\,, \tag{2.19}$$

$$T^r{}_r = \frac{\pi}{a^3r^2}\iint \frac{p_r^2}{\bar{p}^t} f dp_r dl^2\,, \tag{2.20}$$

$$T^r{}_t = \frac{-\pi\alpha}{a^3r^2}\iint p_r f dp_r dl^2\,, \tag{2.21}$$

$$T^\theta{}_\theta = \frac{-\pi}{2ar^4}\iint \frac{l^2 f}{\bar{p}^t} dp_r dl^2\,. \tag{2.22}$$

2.2.3 Evolution of the Distribution Function

Having imposed spherical symmetry the Vlasov equation (2.5) can be written as

$$p^t \frac{\partial f}{\partial t} + p^r \frac{\partial f}{\partial r} + \left(\frac{\alpha' p_t^2}{\alpha^3} + \frac{a' p_r^2}{a^3 p^t} + \frac{l^2}{r^3} \right) \frac{\partial f}{\partial p_r} = 0 \, . \tag{2.23}$$

By defining

$$\begin{aligned} g &\equiv \frac{\alpha p_r}{\alpha^2 \bar{p}^t} = \frac{\partial H}{\partial p_r} \, , & (2.24) \\ h &\equiv -\alpha' \bar{p}^t + \frac{\alpha a' p_r^2}{a^3 \bar{p}^t} + \frac{\alpha l^2}{r^3 \bar{p}^t} = -\frac{\partial H}{\partial r} \, , & (2.25) \end{aligned}$$

where H is the Hamiltonian,

$$H \equiv \alpha \sqrt{m^2 + (p_r/a)^2 + (l/r)^2} \, , \tag{2.26}$$

equation (2.23) can be cast as a conservation law:

$$\frac{\partial f}{\partial t} - \{H, f\} = \frac{\partial f}{\partial t} + \frac{\partial (gf)}{\partial r} + \frac{\partial (hf)}{\partial p_r} = 0 \, . \tag{2.27}$$

This form of the Vlasov equation facilitates the use of finite-volume techniques in our numerical treatment of the problem.

2.3 Static Solutions

Spherically symmetric static solutions of the Vlasov equation can be generated by simply requiring that the distribution function at the initial time take the form $f(0, r, p_r, l^2) = \Phi(E, l)$, where

$$E \equiv \alpha \sqrt{m^2 + (p_r/a)^2 + (l/r)^2} \tag{2.28}$$

is the energy of the particles and, again, l is the angular momentum parameter [55]. Indeed, since E and l are both conserved along particle geodesics in spherical symmetry, any distribution function of this form remains unchanged as the particles move and the Vlasov equation is automatically satisfied.

Explicit construction of the static spacetime resulting from a given choice of $\Phi(E,l)$ requires that the metric functions α and a be determined self-consistently. To that end we can write (2.9) and (2.11) as

$$\frac{-2r\partial_r \ln a + 1}{a^2} - 1 = 8\pi r^2 T^t{}_t(r;\alpha,\Phi)\,, \tag{2.29}$$

$$\frac{2r\partial_r \ln \alpha + 1}{a^2} - 1 = 8\pi r^2 T^r{}_r(r;\alpha,\Phi)\,, \tag{2.30}$$

where

$$T^t{}_t(r;\alpha,\Phi) = -\frac{\pi}{r^2}\iint \bar{p}^t \Phi(E(\alpha,r,w,l),l)dw\,dl^2\,, \tag{2.31}$$

$$T^r{}_r(r;\alpha,\Phi) = \frac{\pi}{r^2}\iint \frac{w^2}{\bar{p}^t}\Phi(E(\alpha,r,w,l),l)dw\,dl^2\,, \tag{2.32}$$

$$w = \frac{p_r}{a}\,, \tag{2.33}$$

$$\bar{p}^t = \sqrt{m^2 + w^2 + (l/r)^2}\,, \tag{2.34}$$

$$E = \alpha\sqrt{m^2 + w^2 + (l/r)^2}\,. \tag{2.35}$$

Given a functional form for $\Phi(E,l)$, we can integrate the equations for $\alpha(r)$ and $a(r)$ from $r=0$ outward, subject to the boundary conditions (2.12)-(2.13). Physically, we also want the particle distribution resulting from a given $\Phi(E,l)$ to have compact support in phase space and finite total mass. As shown in [56], these conditions can be satisfied by introducing a maximum (cut-off) energy, E_0, so that

$$\Phi(E, l) = \phi(E/E_0)\Theta(E_0 - E)F(l)\,, \tag{2.36}$$

where Θ is the unit step function. In Sec. 2.5.2 we construct static solutions based on this ansatz and then investigate their relationship to critical behaviour in the model.

2.4 Numerical Techniques

In this section we summarize our numerical approach for constructing approximate solutions of the equations of motion and the various tests we have performed to establish the correctness and accuracy of our implementation.

2.4.1 Evolution Scheme

As previously mentioned, we treat the matter evolution by a direct discretization of the multidimensional Vlasov equation. Relative to the particle methods adopted in most previous studies of the Einstein-Vlasov system, this has the advantage that our numerical solutions have superior convergence properties. In particular, in contrast to the particle approach, there is no stochastic component of the solution error. This in turn leads to improved confidence in our identification of key aspects of the critical phenomena exhibited in the model, including 1) evidence that the threshold solutions *are* static and 2) the scaling exponents associated with the critical configurations.

As also noted above, the Vlasov equation can be expressed in conservation form and is thus amenable to solution using finite-volume methods. These techniques, which are used extensively in fluid dynamics, for example, are well known for their ability to accurately resolve sharp features—including discontinuities—that often appear in the solution of conservation laws. In our case, evolutions of the distribution function generically exhibit significant mixing and steep gradients; moreover, some of our computations involve initial data which is not smooth in phase space.[26] The finite-volume strategy is thus natural for our purposes. We sketch our specific approach by considering the general form of a conservation equation for a quantity $q(t, x, y)$:

$$\frac{\partial q(t,x,y)}{\partial t} + \frac{\partial F_x(q)}{\partial x} + \frac{\partial F_y(q)}{\partial y} = 0\,, \tag{2.37}$$

where $F_x(q)$ and $F_y(q)$ are the fluxes in the x and y directions. We follow the usual finite volume approach (see [57] for example) by dividing the computational domain into $N_x \times N_y$ cells of uniform size $\Delta x \times \Delta y$ as shown in Fig. 2.1, and define the average value of the unknown q over the cell C_{ij} by

$$Q^n_{ij} = \frac{1}{\Delta x \Delta y} \iint_{C_{ij}} q(t^n, x, y) dx dy\,. \tag{2.38}$$

Here the superscript n labels the discrete time, $t^n \equiv n\Delta t$. We then rewrite (2.37) in integral form:

$$\begin{aligned}\frac{\partial Q}{\partial t} = &-\frac{1}{\Delta x \Delta y}\left(\int_{\mathrm{E}} F_x(q)dy - \int_{\mathrm{W}} F_x(q)dy\right) \\ &-\frac{1}{\Delta x \Delta y}\left(\int_{\mathrm{N}} F_y(q)dx - \int_{\mathrm{S}} F_y(q)dx\right)\,,\end{aligned} \tag{2.39}$$

where the subscripts E, W, N and S denote the east, west, north and south boundaries, respectively, of the cell C_{ij}. Applying a time-discretization to this last expression yields an equation that can be used to advance the cell average in time:

$$\begin{aligned}Q^{n+1}_{ij} = Q^n_{ij} &- \frac{\Delta t}{\Delta x}\left([F_x]^n_{i+1/2} - [F_x]^n_{i-1/2}\right) \\ &-\frac{\Delta t}{\Delta y}\left([F_y]^n_{j+1/2} - [F_y]^n_{j-1/2}\right)\,.\end{aligned} \tag{2.40}$$

Here the average fluxes at the boundaries, $[F_x]^n_{i+1/2}$ etc. are calculated using a Roe solver [57]. We note that our calculations are always performed on meshes that

[26] See for example the choice of $b = 1$ in Eq. (2.70). The radial density distribution of the particles in this choice is not smooth as can be seen in the right panel in Fig. 2.10 (labeled as $b = 1$).

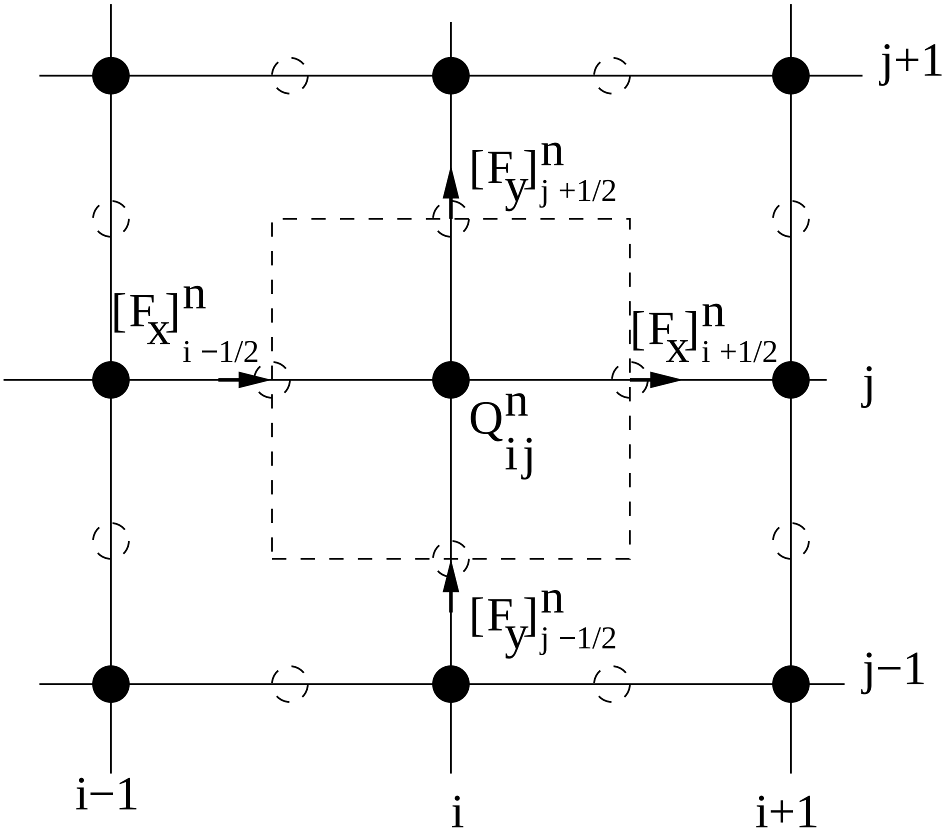


Figure 2.1: A portion of the discretized computation domain used in our finite volume code. The dashed lines delineate one finite volume cell. The cell-centred average value of the density, Q^n_{ij} is defined on the grid points marked with filled circles while the fluxes, $[F_x]^n_{i-1/2}$, $[F_x]^n_{i+1/2}$, etc. are computed at points denoted with dashed circles and which lie on cell boundaries. As described in more detail in the text, Q^n_{ij} is updated using the difference of the outgoing and ingoing fluxes through the cell boundaries.

are uniform in each coordinate direction, and that when we change resolution—to perform a convergence test for example—each mesh spacing is changed by the same factor. Thus, our discretization is fundamentally characterized by a single scale, h. Our specific finite volume approach is based on $O(h^2)$ approximations. However, the nature of the flux calculations—which are designed to inhibit the development of spurious oscillations—means that the scheme is only $O(h)$ in the vicinity of any local extrema in the solution.

The metric variables α and a, which need only be defined on a mesh in the r direction, are computed from $O(h^2)$ finite difference approximations of the Hamiltonian and slicing equations, (2.9) and (2.11). Since the equations for the matter and geometry are fully coupled—i.e. α and a appear in the flux computations, and f is needed for the calculation of the source terms for α and a—some care is needed to construct a scheme which is fully $O(h^2)$ accurate (modulo the degradation of convergence near extremal solution values just noted). In practice, we use an $O(\Delta t^2) = O(h^2)$ Runge-Kutta scheme for the time stepping, which necessitates computation of auxiliary quantities at the half time step $t^{n+1/2} = t^n + \Delta t/2$. Our overall scheme that advances the solution from t^n to t^{n+1}, and which *does* have $O(h^2)$ truncation error, is:

1. Compute $f^{n+1/2}$ from (2.40) using the fluxes F^n.

2. Compute $\tilde{a}^{n+1/2}$ from (2.10) with source $[T^r{}_t]^n$.

3. Compute $\left[T^t{}_t\right]^{n+1/2}$ and $[T^r{}_r]^{n+1/2}$ from (2.19)–(2.20) using $\tilde{a}^{n+1/2}$.

4. Compute $a^{n+1/2}$ and $\alpha^{n+1/2}$ from (2.9) and (2.11) with sources $\left[T^t{}_t\right]^{n+1/2}$ and $[T^r{}_r]^{n+1/2}$.

5. Compute $[T^r{}_t]^{n+1/2}$ from (2.21).

6. Compute fluxes $F_x^{n+1/2}$ and $F_y^{n+1/2}$ using $a^{n+1/2}$ and $\alpha^{n+1/2}$.

7. Compute f^{n+1} from (2.40) and the half-step fluxes $F^{n+1/2}$.

8. Compute $\tilde{a}^{n+1}$ from (2.10) with source $[T^r{}_t]^{n+1/2}$.

9. Compute $\left[T^t{}_t\right]^{n+1}$ and $[T^r{}_r]^{n+1}$ from (2.19) and (2.20) using $\tilde{a}^{n+1}$.

10. Compute a^{n+1} and α^{n+1} from (2.9) and (2.11) using sources $\left[T^t{}_t\right]^{n+1}$ and $[T^r{}_r]^{n+1}$.

11. Compute $[T^r{}_t]^{n+1}$ from (2.21).

12. Compute fluxes F_x^{n+1} and F_y^{n+1} using a^{n+1} and α^{n+1}.

13. One time step complete; start next time step.

To facilitate the use of large grid sizes, as well as to speed up the simulations, we parallelize the computations for the evolution of the distribution function and the calculation of the energy-momentum tensor components using the PAMR [27] package [33]. On the other hand, the calculation of the metric components, which has negligible cost relative to the updates of f and $T^\mu{}_\nu$, is performed on a single processor. The new values of the metric functions are then broadcast to the other CPUs.

2.4.2 Initial Data

In spherical symmetry the gravitational field has no dynamics beyond that generated by the matter content, so initial conditions for our model are completely fixed by the specification of the initial-time particle distribution function, $f(0, r, p_r, l^2)$. However, the Einstein equations (2.9)–(2.11) must also be satisfied at the initial time and, through the definition (2.18) for $\bar{p}^t$, a appears within the integrands for the stress tensor components. To determine all requisite initial values consistently we therefore use the following iterative scheme:

[27] Parallel Adaptive Mesh Refinement: a software developed by Fran Pretorius for parallelizing time dependent PDE solver codes and applying AMR algorithm.

1. Initialize the distribution function, $f(0, r, p_r, l^2)$, to a localized function on phase space.

2. Initialize the geometry to flat spacetime.

3. Calculate the energy momentum tensor using the current geometry.

4. Calculate the geometry using the current energy momentum tensor.

5. Iterate over the matter and geometry calculations until a certain tolerance is achieved.

In practice we find that this algorithm converges in a few iterations.

As discussed in Sec. 2.5.2, when we study static initial data we first specify $\Phi(E, l)$ and then integrate (2.29)–(2.30) outward. We note that the form of $\Phi(E, l)$ that we choose,

$$\Phi(E, l) = \phi(E/E_0)\Theta(E_0 - E)F(l)\,, \tag{2.41}$$

results in equations that are invariant under the transformation:

$$\begin{aligned} \alpha &\rightarrow k\alpha\,, && (2.42) \\ E_0 &\rightarrow kE_0\,. && (2.43) \end{aligned}$$

We can thus first integrate the slicing condition (2.30) subject to the boundary condition, $\alpha(0, 0) = \Lambda$, with $\Lambda < 1$ but otherwise arbitrary, and then linearly rescale $\alpha(0, r)$ so that $\alpha(0, r_{\max}) = 1/a(0, r_{\max})$. The central redshift of the configuration, Z_c, which we use in our analysis below, is then given by

$$Z_c \equiv \frac{1}{\alpha(0, 0)} - 1\,, \tag{2.44}$$

where $\alpha(0, 0)$ is now the rescaled value. It is important to emphasize that different choices for Λ result in distinct solutions, so that irrespective of any adjustable parameters that may appear in the specification of ϕ, equation (2.41) will always implicitly define an entire family of static configurations.

2.4.3 Diagnostic Quantities and Numerical Tests

We have validated our implementations of the algorithms described above using a standard convergence testing methodology that examines the behaviour of the numerical solutions as a function of the mesh spacing, h, keeping the initial data fixed. This section summarizes the tests we perform—which involve derived quantities that should be conserved in the continuum limit as well as the full solutions themselves—and presents results from their application to a representative initial data set using three scales of discretization, h, $h/2$ and $h/4$.

Conserved Quantities

The mass aspect function, $m(t, r)$, is given by

$$m(t,r) = \frac{r}{2}\left(1 - \frac{1}{a^2(t,r)}\right) , \tag{2.45}$$

and measures the amount of mass contained within radius r at time t. Its value at spatial infinity

$$M \equiv m(t, \infty) , \tag{2.46}$$

is the conserved ADM mass. Alternatively, M can be computed using

$$\begin{aligned} M &= \int_0^\infty \rho 4\pi r^2 dr , && (2.47) \\ \rho &= n^\mu n^\nu T_{\mu\nu} , && (2.48) \end{aligned}$$

where n^μ is the unit timelike vector normal to the spatial slices. In developing our code we computed mass estimates based on both of these expressions, but the results presented here and in the remainder of the paper use (2.46) exclusively. Fig. 2.2(c) graphs deviations of M relative to its time-averaged mean value $\langle M\rangle$ for the three computations performed with mesh scales h, $h/2$ and $h/4$. As noted in the caption, the values of $M-\langle M\rangle$ have been rescaled such that the near coincidence of the plots signals the expected $O(h^2)$ convergence to conservation.

The second conserved quantity that we monitor is the real-space particle flux, J_μ, given by

$$J_\mu(t,r) = g_{\mu\nu}\iint \frac{p^\nu}{m} f dV_{p_j}\,. \tag{2.49}$$

In spherical symmetry, the only nonzero components of J_μ are

$$J_t = -\frac{\alpha\pi}{ar^2}\iint f(t,r,p_r)dp_r\,, \tag{2.50}$$

$$J_r = \frac{\pi}{ar^2}\iint \frac{p_r}{\bar{p}^t} f(t,r,p_r)dp_r\,. \tag{2.51}$$

The divergence of the flux must remain zero as the system evolves—written explicitly we have

$$\begin{aligned}\nabla^\mu J_\mu &= \frac{1}{\alpha^3 a^3 r}\Big(- a^3 r\dot{J}_t\alpha + a^3 r J_t\dot{\alpha} + arJ_r\alpha^2\alpha' \\ + \alpha^3 r J_r' a &- \alpha r J_t a^2\dot{a} - \alpha^3 r J_r\alpha' + 2J_r\alpha^3 a \Big) = 0\,. \end{aligned} \tag{2.52}$$

Plots of the rescaled ℓ_2 spatial norm of $\nabla^\mu J_\mu$ as a function of time are shown in Fig. 2.2(d)—again $O(h^2)$ convergence is observed.

Independent Residual Test

As noted in Sec. 2.2.1, the $\theta\theta$ component of Einstein's equation is not used in our evolution scheme but must be satisfied in the continuum limit if our numerical results are valid. We thus define the residual

$$E^\theta{}_\theta \equiv G^\theta{}_\theta - 8\pi T^\theta{}_\theta\,, \tag{2.53}$$

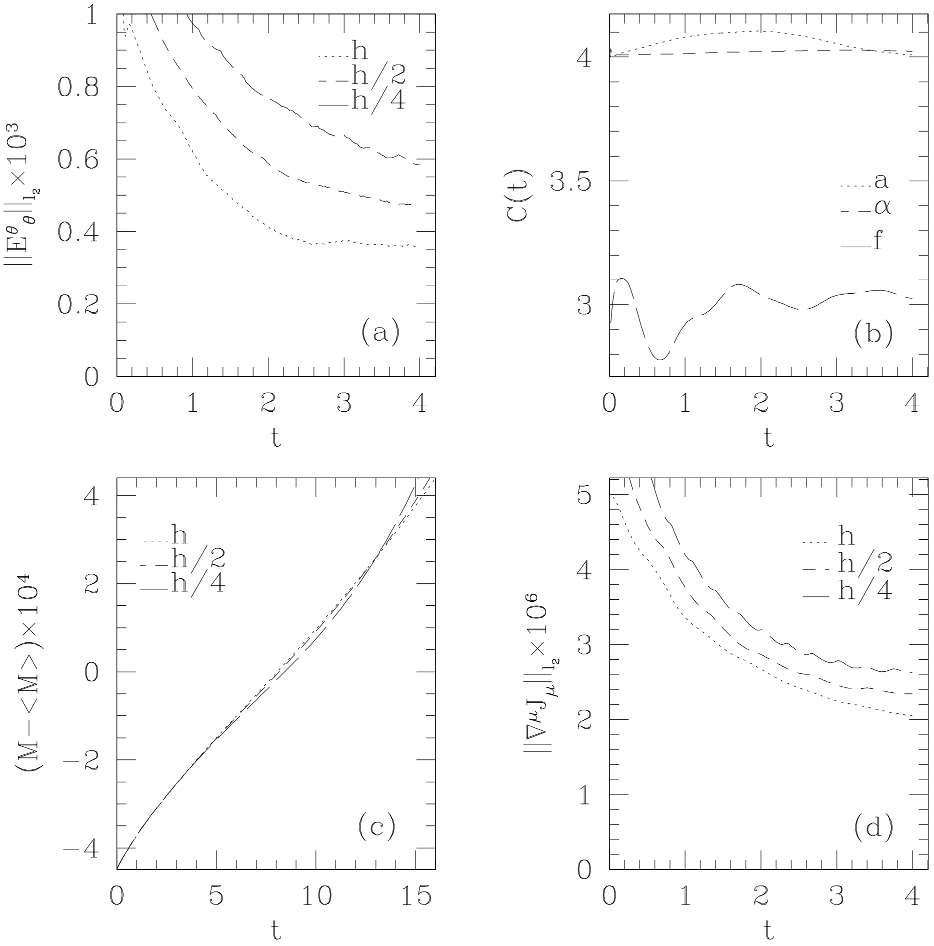


Figure 2.2: Results of various diagnostic tests used to test the numerical solver. The initial data and mesh resolutions used here are typical of any of the 2D calculations described in the paper. A standard convergence testing methodology, using three calculations with fixed initial data and mesh spacings h, $h/2$ and $h/4$, is employed. The coarsest mesh has $n_x \times n_y = n_r \times n_p = 128 \times 128$ grid points. Plots (a), (c) and (d) all display quantities that are residual in nature, i.e. which should tend to zero quadratically in the mesh spacing. Values from the $h/2$ and $h/4$ computations have been rescaled by factors of 4 and 16, respectively, and the near-coincidence of the rescaled values thus demonstrates that all three quantities are converging at the expected $O(h^2)$ rate. (a) Convergence of the l_2 norm of the independent residual, $\|E^\theta{}_\theta\|_2$, defined by (2.53). (b) Convergence factors (3.68) of the primary dynamical unknowns. Here, convergence of the metric functions, α and a, is clearly second order, while that for the distribution function is better than $O(h)$ but is not $O(h^2)$. This latter behaviour is to be expected since the finite volume method used to update f is only first order in the vicinity of local extrema. (c) Convergence of the deviation in computed total mass, calculated from (2.45) and (2.46). (d) Convergence of the particle flux divergence (2.52).

where

$$
\begin{aligned}
G^{\theta}{}_{\theta} &= G^{\phi}{}_{\phi} \\
&= -\frac{1}{r\alpha^3 a^3}\left(-\alpha^2 a\frac{\partial\alpha}{\partial r} + \alpha^3\frac{\partial a}{\partial r} + \alpha^2 r\frac{\partial\alpha}{\partial r}\frac{\partial a}{\partial r}\right. \\
&- \left. \alpha^2 ar\frac{\partial^2\alpha}{\partial r^2} + a^2\alpha r\frac{\partial^2 a}{\partial t^2} - a^2 r\frac{\partial\alpha}{\partial t}\frac{\partial a}{\partial t}\right) , \qquad (2.54)
\end{aligned}
$$

and $T^{\theta}{}_{\theta}$ is given by (2.22). Then, using second-order finite differences to approximate all derivatives, we monitor the ℓ_2 norm of $E^{\theta}{}_{\theta}$ during the calculations. We expect $\|E^{\theta}{}_{\theta}\|_2$ to be $O(h^2)$ and Fig 2.2(a) shows that this is the case.

Full-solution Convergence Test

The final check we perform is a basic convergence test of the primary dynamical variables, α, a and f. Denoting the values computed at resolution h for any of these by $q^h(t,X)$—where $X = r$ for α and a, and $X = (r, p_r)$ for f—we calculate convergence factors, $C(t;q)$, defined by

$$
C(t;q) = \frac{||q^h(t,X) - q^{h/2}(t,X)||_{l_2}}{||q^{h/2}(t,X) - q^{h/4}(t,X)||_{l_2}} . \qquad (2.55)
$$

If our scheme is $O(h^2)$ convergent then it is easy to argue that $C(t;q)$ should approach 4 in the continuum limit. Plots of $C(t;a)$, $C(t,\alpha)$ and $C(t;f)$ are shown in Fig. 2.2(b). Second order convergence of the geometric variables is apparent, while the behaviour of $C(t;f)$ reflects the fact that the finite volume method we use is only first-order accurate in the vicinity of extrema of f. Interestingly, at least at the resolutions used here, the deterioration of the convergence of f does not appear to significantly impact that of the geometric quantities.[28]

[28] This can be traced back to the fact that the geometric quantities are only related to f via the integral of the density function over the momentum direction in the phase space. Finite volume methods lose their point-wise second order accuracy in the vicinity of extrema points, but they are well preserving with respect to the quantities derived by integrating over the finite volume cells such as the energy momentum tensor.

Family	D	$f(0,r,p_r,l)$	p
G1	2	$\delta(l-l_0)\mathcal{G}(A,r_c,p_c)$	p_c
G2	2	$\delta(l-l_0)\mathcal{G}(A,r_c,p_c)$	l_0
G3	2	$\delta(l-l_0)\mathcal{G}(A,r_c,0)$	A
G4	2	$\delta(l-l_0)\left(\mathcal{G}(A,r_c,p_c)+\mathcal{G}(A,r_c+\Delta r,p_c+\Delta p)\right)$	p_c
G5	2	$\delta(l-l_0)\mathcal{E}(A,r_c,p_c)$	p_c
G6	2	$\delta(l-l_0)\mathcal{E}(A,r_c,0)$	A
G7	2	$\delta(l-l_1)\mathcal{G}(A,r_1,p_1)+\delta(l-l_2)\mathcal{G}(A,r_2,p_2)$	p_1
G8	3	$\exp(-(l-l_0)^2/\Delta l^2)\mathcal{G}(A,r_c,p_c)$	p_c
G9	3	$\exp(-(l-l_0)^2/\Delta l^2)\mathcal{G}(A,r_c,0)$	A
G10	3	$\Theta(l-5)\Theta(15-l)\mathcal{E}(A,r_c,0)$	A

Table 2.1: Families of generic initial data used in the studies described in text. The columns enumerate: (1) the label for the family, (2) the number, D, of phase-space dimensions on which the distribution function depends (and therefore whether the 2D or 3D code was used to generate the results), (3) the form of the initial data, $f(0,r,p_r,l)$ (see (2.57) and (2.58) for the definitions of $\mathcal{G}$ and $\mathcal{E}$), and (4) the control parameter, p, that was varied to study the critical behaviour. The quantities $l_0,l_1,l_2,r_c,r_1,r_2,p_c,p_1,p_2,\Delta r$ and Δp that appear in the various specifications of $f(0,r,p_r,l)$ are all parameters; i.e they have fixed scalar values in any given computation.

2.5 Results

In this section we describe the main results from our investigation of critical behaviour in the Einstein-Vlasov model. We have used many different families of initial data in our studies and what we report below is based on a representative sample of those. As mentioned in the introduction, the numerical experiments fall into three broad classes. The first uses massless particles and initial data which have some relatively arbitrary form in phase space. The second also uses massless particles but with initial conditions that represent perturbed static solutions. Finally, the third set is the same as the first but with massive particles. We will refer to these classes as generic massless, near-static massless, and generic massive, respectively. In addition, the calculations can be categorized according to whether l is a single fixed value, l_0, (2D) or if the distribution function has non-trivial l-dependence (3D). The functional form of the various families considered, along with the dimensionality of the corresponding PDEs and the parameter used for tuning to criticality are summarized in Table 2.1.

2.5.1 Generic Massless Case

Here we use initial distribution functions, $f_0 \equiv f(0, r, p_r, l)$, that describe configurations of particles localized in r, p_r and l, and that include various parameters which can be tuned to generate families of solutions that span the black hole threshold. Specifically, we set

$$f(0, r, p_r, l^2) = S(r, p_r)F(l)\,, \tag{2.56}$$

where $S(r, p_r)$ is given by either a gaussian function,

$$\mathcal{G}(r, p_r;\, A, r_c, p_c) \equiv A \exp\left(-\frac{(r - r_c)^2}{\Delta_r^2} - \frac{(p_r - p_c)^2}{\Delta_p^2}\right)\,, \tag{2.57}$$

or the truncated bi-quadratic form

$$\mathcal{E}(r, p_r;\, A, r_c, p_c) \equiv \begin{cases} A\bar{r}(1-\bar{r})\bar{p}(1-\bar{p}) & 0 < \bar{r} < 1\,, \\ & 0 < \bar{p} < 1\,, \\ 0 & \text{elsewhere,} \end{cases} \tag{2.58}$$

where $\bar{r} = (r - r_c + \Delta r)/2\Delta r$ and $\bar{p} = (p_r - p_c + \Delta p)/2\Delta p$. Note that the dependence of $\mathcal{G}$ and $\mathcal{E}$ on r and p_r is suppressed in the abbreviated notation used in Table 2.1. For the 3D calculations, we use two types of angular momentum distribution: the first is a gaussian,

$$F(l) = \exp\left(\frac{-(l - l_0)^2}{\Delta l^2}\right)\,, \tag{2.59}$$

while the second is uniform in l with cutoffs at some prescribed minimum and maximum values, $l_{\min}$ and $l_{\max}$, respectively,

$$F(l) = \Theta(l - l_{\min})\Theta(l_{\max} - l)\,. \tag{2.60}$$

It is important to point out that since the massless Einstein-Vlasov system is

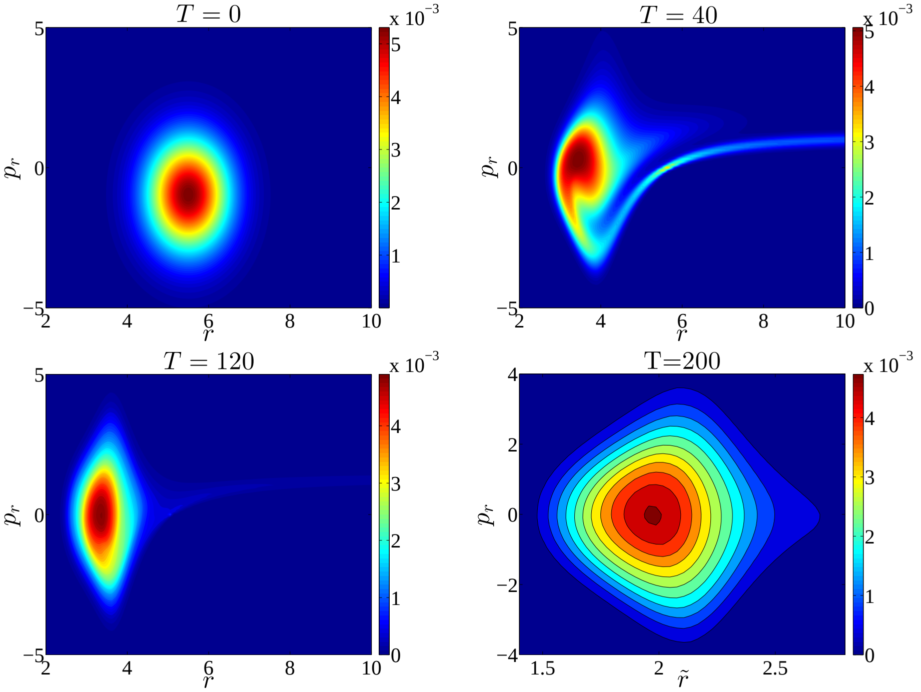


Figure 2.3: Snapshots of the distribution function from a typical near-critical calculation, with evolution proceeding left to right, top to bottom (note the reduction in the range of radial coordinate in the last frame). The displayed results are from family G8 (see Table 2.1) where p_c—which is loosely the average momentum of the initially imploding shell of particles—is the control parameter. As with all of the calculations discussed in the results section, the control parameter has been tuned to roughly machine precision. In the early stages of the evolution we observe phase space mixing and the ejection of some particles (the latter particularly visible as the "tail" in the second frame). At intermediate times the system approaches a static state which persists for a period that is long compared to the infall/dispersal timescale characterizing weak field dynamics. We note that this is a 3D calculation, with f non-trivial in the l direction: for visualization purposes we have integrated over l to produce a quantity depending only on r and p_r. Additionally, the first three frames are plotted using the computational coordinate, r, while for the purposes of direct comparison with Fig. 2.4, the fourth uses the rescaled coordinate, $\tilde{r}$, defined by (2.64). We emphasize that at criticality f retains non-trivial dependence on p_r; that is, although the geometry is static, the particle behaviour is still dynamic.

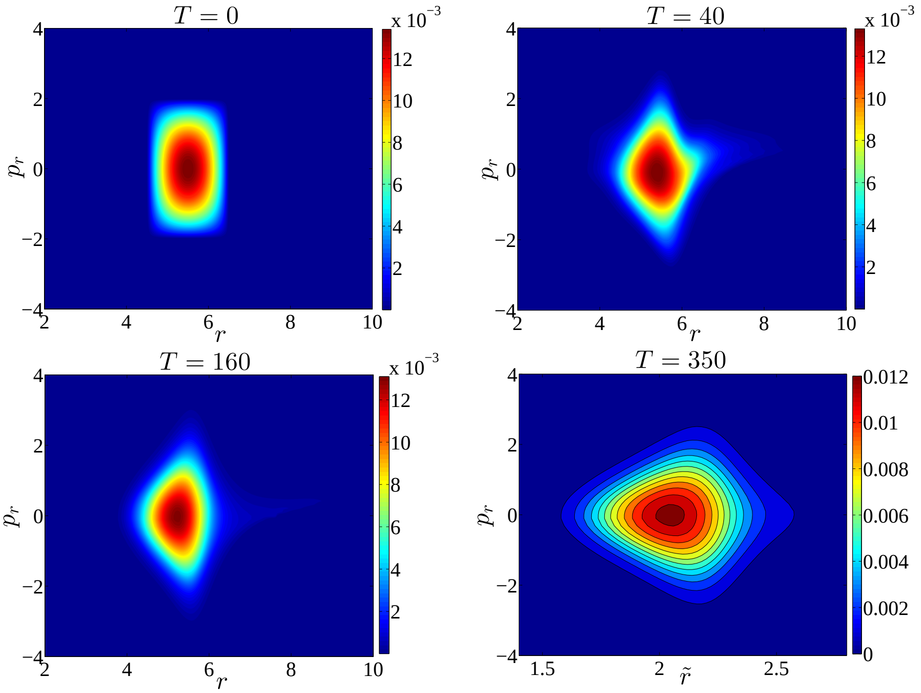


Figure 2.4: Snapshots of the distribution function for a near-critical calculation using family G10. Here the tuning parameter is the overall amplitude, A, of the initial particle distribution. As in the previous figure the sequence shows an approach to a static state, but it is evident that the form of the distribution function at criticality is significantly different in the two calculations. Due to the use of the rescaled radial coordinate, $\tilde{r}$, the fourth frames of the two figures can be meaningfully compared.

scale-free it has an additional symmetry relative to the massive case. Specifically, the equations of motion are invariant under the transformation

$$t \to kt\,, \tag{2.61}$$

$$r \to kr\,, \tag{2.62}$$

where k is an arbitrary positive constant. In order to meaningfully compare results from different initial data choices we must therefore adopt unitless coordinates in our analysis. We do this by rescaling t and r by the total mass, $M^\star$, of the putatively static solution which arises at criticality for any of the families that we have considered (that is, $M^\star$ includes only the mass associated with that portion of the overall matter distribution which appears to be static at criticality). Moreover, it is more natural and convenient to use central proper time, τ, rather than t itself in the analysis. Thus, the results below are described using rescaled coordinates, $\tilde{\tau}$ and $\tilde{r}$, defined by

$$\tilde{\tau} = \frac{\tau}{M^\star}\,, \tag{2.63}$$

$$\tilde{r} = \frac{r}{M^\star}\,. \tag{2.64}$$

We note that under the scaling (2.61)–(2.62) the angular momentum transforms as

$$l \to kl\,. \tag{2.65}$$

The process we use to generate near-critical solutions is completely standard for this type of work. All of the family definitions described above and summarized in Table 2.1 contain multiple parameters that can be used to tune to the black hole threshold and, consistent with what has been found in many other previous studies of black hole critical phenomena, we find that which particular parameter is actually varied is essentially irrelevant for the results. Having chosen *some* specific parameter, p, to vary, any critical search begins by determining an initial bracketing interval, $[p_l, p_h]$, in parameter space such that evolutions with p_l and p_h lead to dispersal and black hole formation, respectively. We then narrow the bracketing interval using a bisection search on p, predicating the update of p_l or p_h on whether or not a black hole forms. The search is continued until $(p_h - p_l)/p_h \sim 10^{-15}$, so that $p^\star$ is computed to about machine precision (8-byte floating point arithmetic). The value of p_l at the end of this process corresponds to what we dub the marginally sub-critical solution.

Quite generically, as we tune any family to a critical value $p^\star$, the phase space distribution function appears to settle down to a static solution which, as $p \to p^\star$, persists for a time that is long compared to the characteristic timescale for implosion and subsequent dispersal of the particles in the weakly-gravitating limit. Representative illustrations of this behaviour are shown for marginally sub-critical evolutions from two distinct initial data families in Fig. 2.3 (family G8 in Table 2.1) and Fig. 2.4 (family G10). Similarly, the spacetime geometry–encapsulated in the metric functions a and α—also becomes increasingly time-independent as criticality is approached. Fig. 2.5 displays the evolution of the ℓ_2-norm of the time derivative of a during marginally sub-critical evolution for family G1. We thus have strong evidence that the critical solutions that we are finding are static—characteristic of type I critical behaviour—and consistent with what has been observed previously for the case of the *massive* Einstein-Vlasov system.

Further evidence for generic type I transitions in the model is provided by observations of lifetime scaling of the form (2.1) near criticality, which is expected if the critical solutions are one-mode unstable. Typical results from calculations using families G1, G4, G8 and G10 are shown in Fig. 2.6: the linearity of the lifetime of the static critical configuration as a function of $\ln|p - p^\star|$ is apparent. We have observed such scaling for all of the families that we have studied (in both the 2D and 3D cases) and Table 2.2 provides a summary of the measured values of the scaling

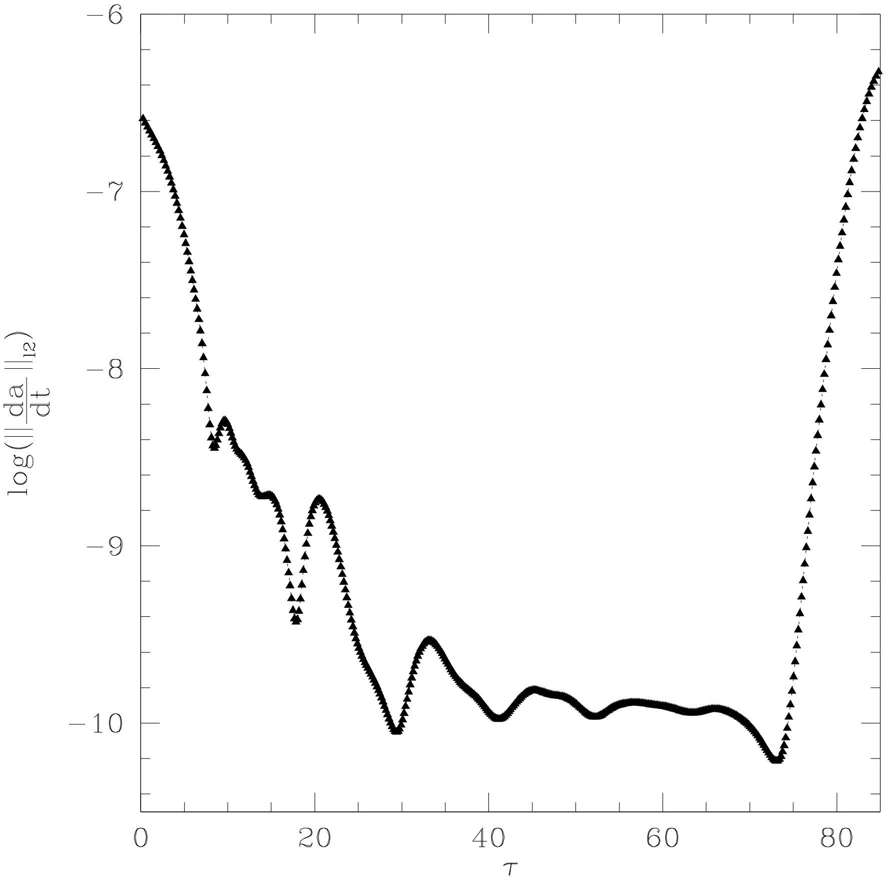


Figure 2.5: Time evolution of $\|\partial_t a(t,r)\|_2$ from a marginally sub-critical calculation using family G1. The plot provides strong evidence that the geometry of the threshold solution is static, a characteristic feature of type I behaviour.

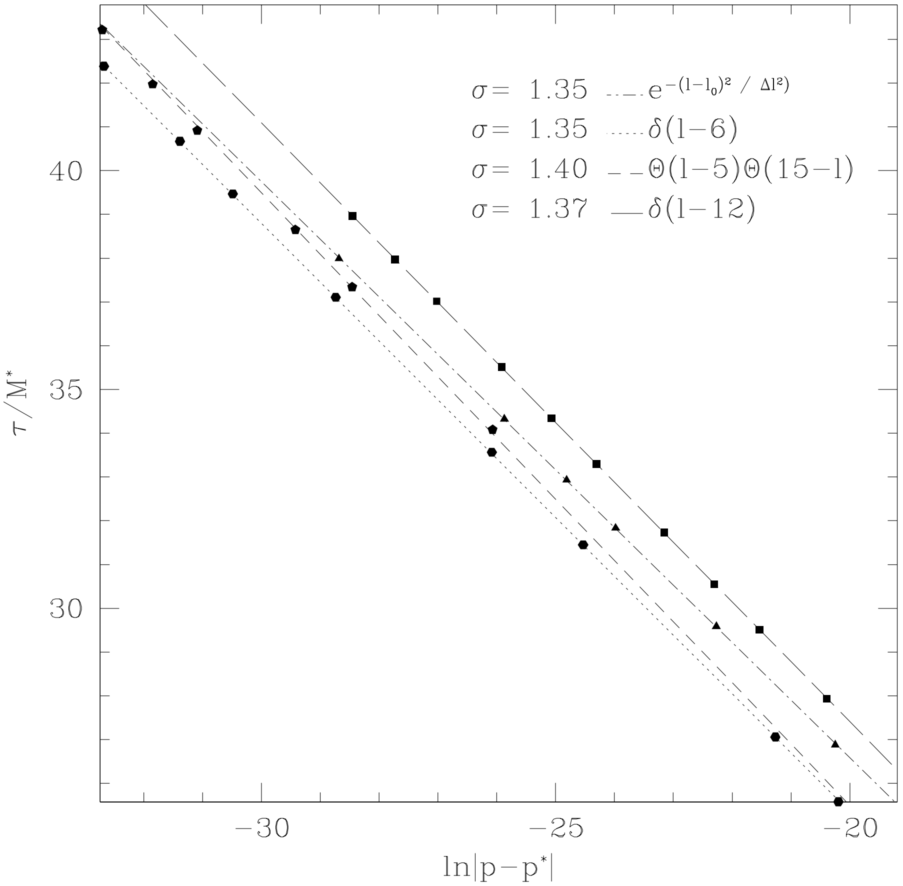


Figure 2.6: Lifetime scaling of near-critical configurations for families G8, G1, G10 and G4 (top to bottom and noting that G10 and G8 are 3D calculations while the others are 2D). Here the symbols plot estimates of the amount of time the state of the system is well approximated by the static critical solution—measured in units of the rescaled proper time defined by (2.63)—as a function of $\ln|p - p^\star|$. The lines are least squares fits to $\tau = -\sigma \ln|p - p^\star|$ where σ is the reciprocal of the eigenvalue (Lyapunov exponent) corresponding to the presumed single growing mode of the critical solution. To the estimated level of accuracy in our calculations the measured values of σ are the same for the four families. However, we cannot state with certainty that there is precise universality in this regard.

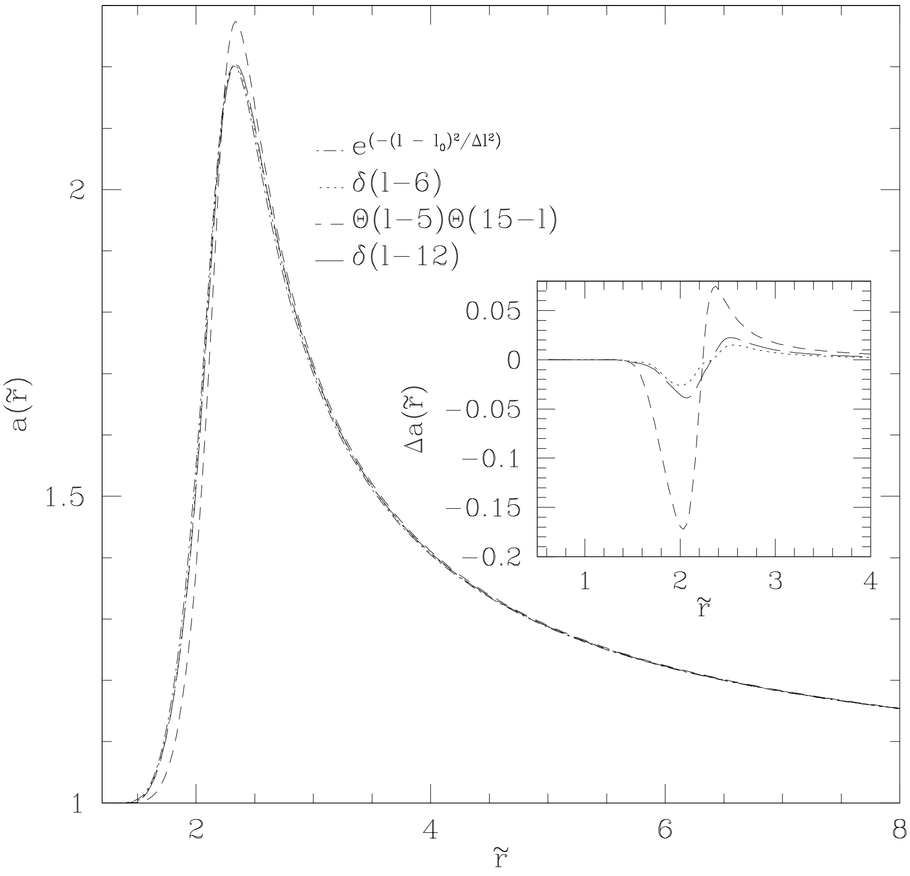


Figure 2.7: Radial metric function $a(\tilde{r})$ at criticality for families G8, G1, G10 and G4. The results plotted here, together with those displayed in Fig. 2.8, show that there is relatively little variation in the geometry of the static critical configuration as a function of the specifics of the initial data. The inset plots the deviation in a for families G1, G10 and G4 relative to G8.

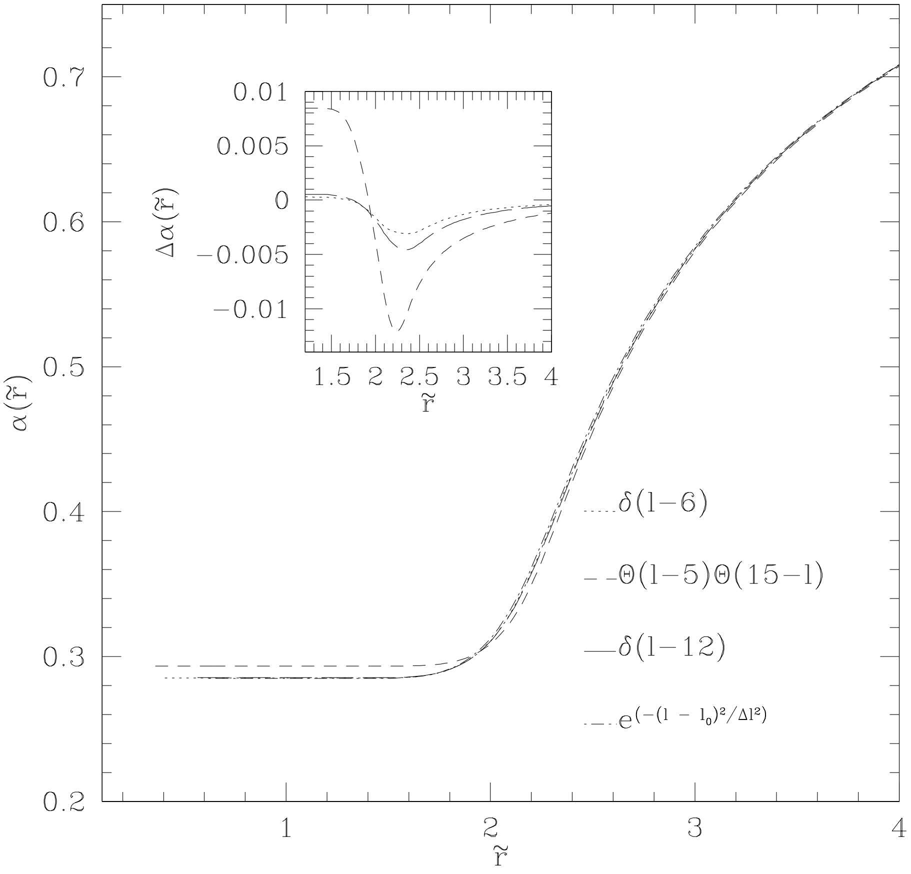


Figure 2.8: Lapse function $\alpha(\tilde{r})$ at criticality for families G8, G1, G10 and G4. The comments made in the caption of the previous figure apply here as well.

exponent, σ.

We note that the specific form of the matter configuration at criticality exhibits significant dependence on the family of initial data that is used to generate the critical solution. This can be seen, for example, by comparing the last frames of Figs. 2.3 and 2.4. On the other hand, as illustrated in Fig. 2.7 and Fig. 2.8, the *geometry* of the critical state is relatively insensitive to the initial conditions.

The spacetime geometry can be characterized by the central red shift, Z_c defined by (2.44), and the unitless compactness parameter, Γ, defined by

$$\Gamma = \max_r \frac{2m}{r} \,. \tag{2.66}$$

For the families considered in this section the values of Γ and Z_c fall in the ranges

$$0.79 \lesssim \Gamma \lesssim 0.81 \,, \tag{2.67}$$

$$2.4 \lesssim Z_c \lesssim 2.5 \,. \tag{2.68}$$

As discussed in the next section, these ranges are relatively small in comparison to those found in our investigation of critical behaviour using nearly-static initial data.

What is striking about the results assembled in Table 2.2 is that there appears to be a small variation, at most, in the time scaling exponent associated with the critical solutions produced from our generic initial conditions. Specifically, the data is consistent with

$$\sigma = 1.4 \pm 0.1 \,, \tag{2.69}$$

and we emphasize that this concordance arises despite the significant observed variation in the phase-space distribution of the particles among the various critical solutions.

Family	l_0	σ	Family	l_0	σ
G1	5	1.32 ± 0.08	G3	12	1.36 ± 0.06
G1	6	1.35 ± 0.07	G4	12	1.37 ± 0.05
G1	7	1.36 ± 0.06	G5	12	1.44 ± 0.06
G1	8	1.33 ± 0.06	G6	12	1.43 ± 0.04
G1	9	1.33 ± 0.06	G7	6 & 12	1.37 ± 0.07
G1	10	1.32 ± 0.06	G8	10	1.35 ± 0.05
G1	11	1.35 ± 0.05	G9	10	1.36 ± 0.05
G1	12	1.37 ± 0.05	G10	-	1.40 ± 0.05
G2	-	1.36 ± 0.07			

Table 2.2: Summary of measured lifetime scaling exponents for the massless Einstein-Vlasov model from experiments using the various initial data families enumerated in Table 2.1. In addition to the overall functional form of the initial distribution functions, a key parameter that varies among the sets of calculations is l_0, which is the angular momentum of any and all particles for families G1, G2–G6 (2D) and the center of the angular momentum distribution for families G8 and G9 (3D). (l_0 is the tuning parameter for G6, and family G7 is another special case where the initial data is comprised of a superposition of two shells of particles, each having a distinct angular momentum parameter. Since angular momentum is a conserved quantity there is no mixing of the two distributions during the evolution.) For simplicity of presentation we have not listed the other parameters defining the different initial configurations. Quoted uncertainties in the values of σ are based on variations in the total mass of the system during the evolutions and comparison with results computed at lower resolution. Typical grid sizes used for the listed results are $n_r \times n_p = 1024 \times 1024$ (2D) or $n_r \times n_p \times n_l = 256 \times 128 \times 64$ (3D). To the level of accuracy in our calculations we find consistency with a single value of the scaling exponent, $\sigma = 1.4 \pm 0.1$.

2.5.2 Near-static Massless Case

Our second approach to study critical solutions in the massless Einstein-Vlasov system starts with the construction of static initial data using the procedure described in Sec. 2.3. We specialize the general form (2.41) to

$$\Phi(E,l) = C(1 - E/E_0)^b \Theta(E_0 - E)\delta(l - l_0)\,, \tag{2.70}$$

where E_0 is a given cutoff energy and C, b and l_0 are additional adjustable parameters. Here we focus exclusively on the case of fixed angular momentum (2D calculations) since the results of the previous section suggest that the essential features of the critical solutions are not significantly dependent on whether or not f has non-trivial dependence on l. In addition, from the scale free symmetry in the system (see (2.61) and (2.65)), we can conclude that varying the value of angular momentum is equivalent to rescaling the radial coordinate. Therefore, without loss of generality we can set l to an arbitrary fixed value, eliminating one of the parameter-space dimensions in our surveys. Additionally, so that we can meaningfully compare results from different initial conditions, we again rescale the radial coordinate by the total mass of the system (2.64). Furthermore, by virtue of the transformation (2.43), the static profiles depend on E_0 only through the ratio E_0/α_0 and, since it simplifies the numerical analysis, we actually use this ratio as one of the control parameters.

For specified values of the free parameters C, b and E_0/α_0, we integrate equations (2.29)–(2.32) outward until we reach a radial location, r_X, where the particle density $\Phi(E,l)$ vanishes. We then extend the solution for a and α to the outer boundary of the computational domain by attaching a Schwarzschild geometry with the appropriate mass.

We note that not all choices of the three free parameters lead to distribution functions with compact support—that is, with $f(0,r,p_r) \equiv 0$ for r greater than some r_X—so that the configuration represents a single shell of particles. Indeed, by examining the expression for the particle energy in the massless case:

$$E(r, p_r) = \alpha(r)\sqrt{(p_r/a)^2 + (l/r)^2}\,, \tag{2.71}$$

we see that, for p_r sufficiently small, $E(r, p_r)$ can remain below the cutoff E_0 for large r. In practice this will yield solutions with multiple shells, where Φ vanishes at r_X, but then becomes non-zero on a infinite number of intervals in r (in general these intervals can be disjoint or contiguous, as has previously been seen in [43]). Although it might be interesting to consider the critical dynamics of multiple-shell solutions, we do not do so here. We also note that for given values of b and E_0/α_0 we find solutions with a distinct shell (i.e. where Φ *does* vanish at some radius) only for a certain range of C, but that range can span several orders of magnitude.

Fig. 2.9 shows the distribution function for four sample static configurations constructed as described above, with the associated geometrical variables plotted in Fig. 2.10. Relative to the apparently static solutions generated by tuning generic initial data, the family-dependence of both the distribution function and metric variables here is much more pronounced.

One interesting way of characterizing the static solutions is to plot the compactness parameter, Γ, defined by (2.66), as a function of the central redshift, Z_c. We do this for a large number of configurations in Fig. 2.11 where, as described in more detail in the caption, each set of points results from a two-dimensional parameter space survey wherein both E_0/α_0 and C are varied. The fact that the solutions from each of these surveys tend to "collapse" to one-dimensional curves in Z_c–Γ space is striking and we do not have any argument at this time for why this should be so.

All of the static solutions that we have found satisfy Buchdahl's inequality, $\Gamma < 8/9$, originally derived in the context of fluid matter [59], and the most compact configurations are quite close to that limit. Here it is crucial to note that Andréasson has proven rigorously that the Buchdahl inequality is satisfied by any static solution

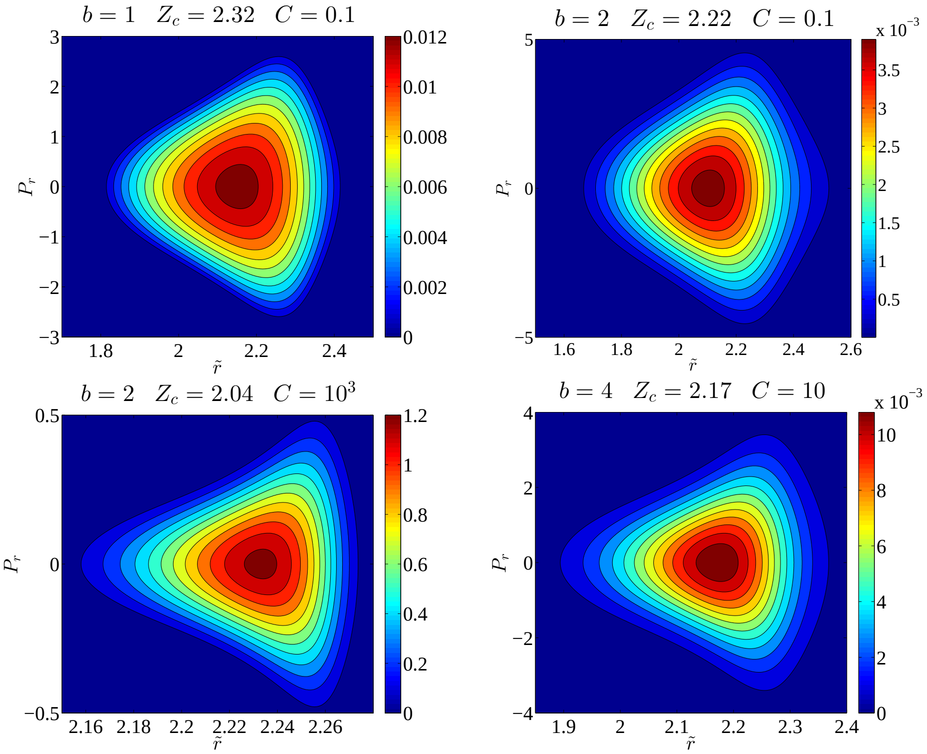


Figure 2.9: Sample static phase space configurations computed from the ansatz (2.70) using different choices of adjustable parameters. Note that although we use the rescaled radial coordinate $\tilde{r}$ in all of the plots, the ranges in $\tilde{r}$, p_r and f vary from frame to frame. Clearly, there is a strong dependence of f on the chosen parameter values. As described in more detail in the text, for any given values of b and Z_c there is a finite range of C for which we find static solutions where f has compact support.

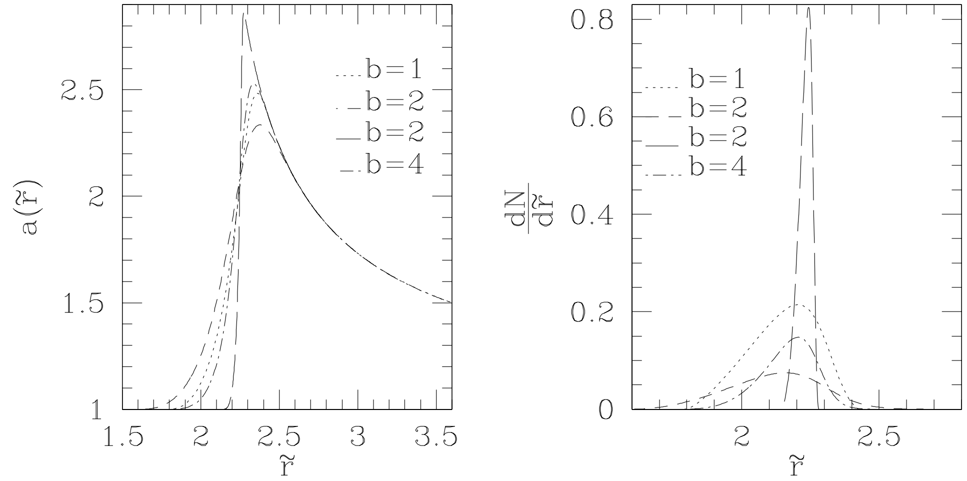


Figure 2.10: Plots of the radial metric function, $a(r)$, and differential particle number, $dN(r)/dr$, for the configurations shown in Fig. 2.9. The graphs of $dN(r)/dr$ highlight the fact that the critical solutions are shell-like, with a thicknesses and effective densities that are strongly dependent on the choice of parameters in (2.70).

of the spherically symmetric Einstein-Vlasov system [58]. Further, he has demonstrated that one can construct static shell-like configurations which, in the limit of infinitesimal thickness in r, can have Γ arbitrarily close to 8/9. Although not explicitly mentioned in [58], it is clear that his proof is valid for $m = 0$. Given the nature of Andréasson‘s result, the observation that our solutions satisfy the bound clearly amounts to little more than additional evidence that our calculations are faithful to the model under study. However it *is* interesting that the highest values of Γ seen in Fig. 2.11—and which plausibly *are* approaching 8/9—are associated with very thin shell-like solutions. Additionally, for the configurations we have studied (not all of which are represented in Fig. 2.11) there is apparently also a lower bound on the compactness, $\Gamma \sim 0.81$. Finally, the ranges of Γ and Z_c spanned by the explicitly static solutions

$$0.80 \lesssim \Gamma \lesssim 0.89\,, \qquad (2.72)$$

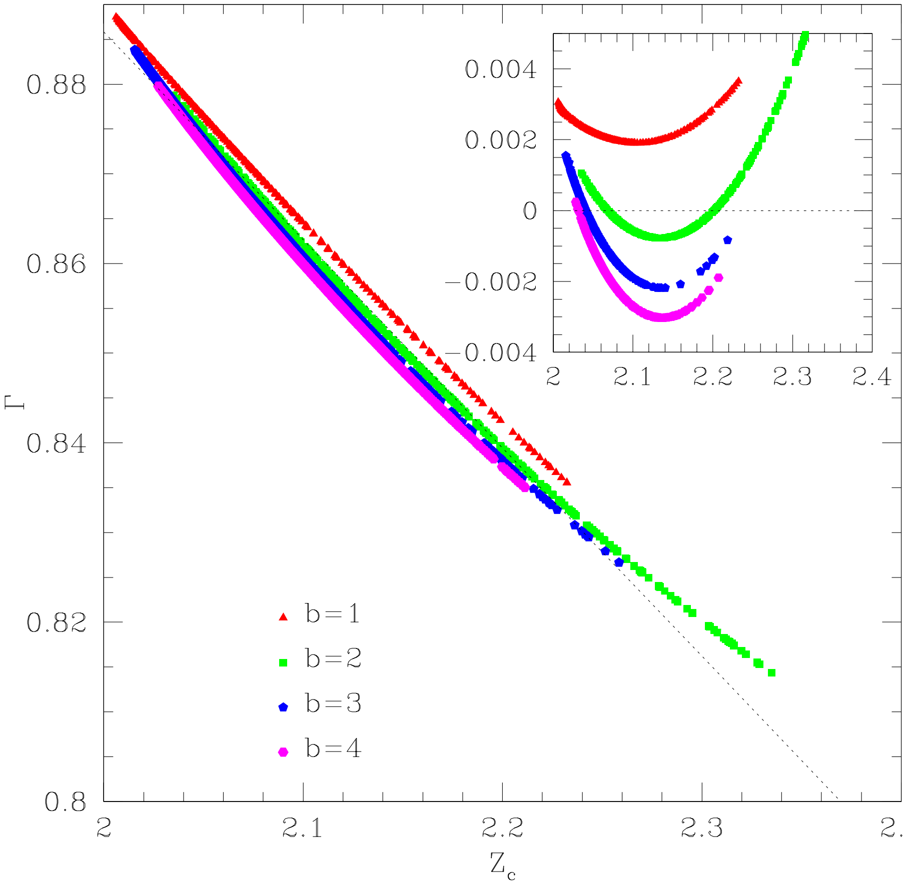


Figure 2.11: The value of $\Gamma = \max_r(2m/r)$ versus central redshift, Z_c, for various static solutions. Each set of points comprises several thousand distinct solutions and comes from a two-dimensional parameter space survey, in which both C and E_0/α_0 are varied. Although for given b and E_0/α_0 we can only find acceptable static solutions in certain ranges of C, those ranges can span several orders of magnitude. However, for fixed b the solutions tend to collapse to near-linear loci in Z_c–Γ space, and the inset graph, which plots the deviation of the data from a linear least squares fit, is intended to emphasize this behaviour. More detailed examination of the data suggests that the configurations do *not* lie precisely along one-dimensional curves, but additional study would be required to determine whether this is really the case. The solutions apparently satisfy the Buchdahl inequality $\Gamma < 8/9$ (also seen in the calculations reported in [43] for the massive case), as is expected from Andréasson's rigorous results [58]. Moreover, there also seems to be a *lower* bound on the compactness, $\Gamma \sim 0.81$.

b	Z_c	C	δf	σ
1	2.32	0.1	δf_1	1.45 ± 0.05
1	2.23	0.3	δf_1	1.45 ± 0.04
2	2.22	0.1	δf_1	1.43 ± 0.04
4	2.17	10	δf_1	1.43 ± 0.04
2	2.35	0.1	δf_1	1.40 ± 0.05
2	2.35	0.1	δf_2	1.40 ± 0.05
2	2.35	0.1	δf_3	1.40 ± 0.05

Table 2.3: Measured lifetime scaling exponent for explicitly static solutions constructed from ansatz (2.70) with various choices of the adjustable parameters b, E_0/α_0 and C (Z_c is effectively controlled by E_0/α_0, but is determined *a posteriori*), and the different types of perturbations, δf, enumerated in (2.75)–(2.77). Proceeding from the assumption that the static solutions *are* characterized by a single unstable mode, we anticipate that the computed value of σ associated with a specific configuration (i.e. for given b, Z_c and C) should be independent of the form of δf, and this is precisely what we observe (compare rows 1 and 2, and 5, 6 and 7). However, we also see once again that there is little, if any, variation in the scaling exponent with respect to the underlying critical solution: the results in the table are consistent with $\sigma = 1.43 \pm 0.07$

$$2.0 \lesssim Z_c \lesssim 2.4 \,, \tag{2.73}$$

are larger than those seen for the tuned generic data, consistent with the comment above concerning the relatively large variations in the metric variables as well as the distribution function.

Using our evolution code, we investigate the relation of the explicitly-static solutions to critical behaviour in the model as follows. For initial conditions we set

$$f(0, r, p_r, l^2) = f^0(r, p_r, l^2) + (A-1)\delta f(r, p_r, l^2) \,, \tag{2.74}$$

where f^0 is a static configuration, $\delta f(r, p_r, l^2)$ is some given perturbation function with at least roughly the same support as f^0, and A is a tunable parameter which controls the amplitude of the perturbation. Clearly, $A = 1$ results in initialization with the static solution itself. We have experimented with the following three choices for the perturbation function:

$$\delta f_1(r, p_r, l^2) = f^0(r, p_r, l^2)\,, \tag{2.75}$$

$$\delta f_2(r, p_r, l^2) = \sin\left(\frac{2\pi f^0(r, p_r, l^2)}{f_{\rm max}}\right)\,, \tag{2.76}$$

$$\delta f_3(r, p_r, l^2) = f^0(r, p_r, l^2)(f_{\rm max} - f^0(r, p_r, l^2))p_r\,, \tag{2.77}$$

where $f_{\rm max}$ is the maximum of f^0 over the computational domain. We then perform standard tuning experiments in which we vary A to isolate a threshold solution.

Interestingly, we find strong evidence that *all* of the static solutions based on (2.70) that we have found sit at the threshold of black hole formation, so that setting $A > 1$ results in black hole formation while taking $A < 1$ results in complete dispersal of the matter (or vice versa, dependent on the precise form of δf). As should be suspected then, and as is shown for four families in Fig. 2.12, the solutions generated by dynamically evolving the perturbed static configurations exhibit time scaling—this strongly suggests that the time-independent solutions are all one-mode unstable. Table 2.3 provides a summary of the time-scaling exponents we have measured for a set of experiments based on four distinct static solutions and the three different types of perturbation defined by (2.75)–(2.77).

As was the case for the generic families, the measurements here indicate that although the static solutions display significant variation in both the distribution function and geometric variables, there is little variation in the scaling exponent. Here we find

$$\sigma = 1.43 \pm 0.07\,. \tag{2.78}$$

Recalling (2.69), and given the estimated uncertainty in our calculations, we can not

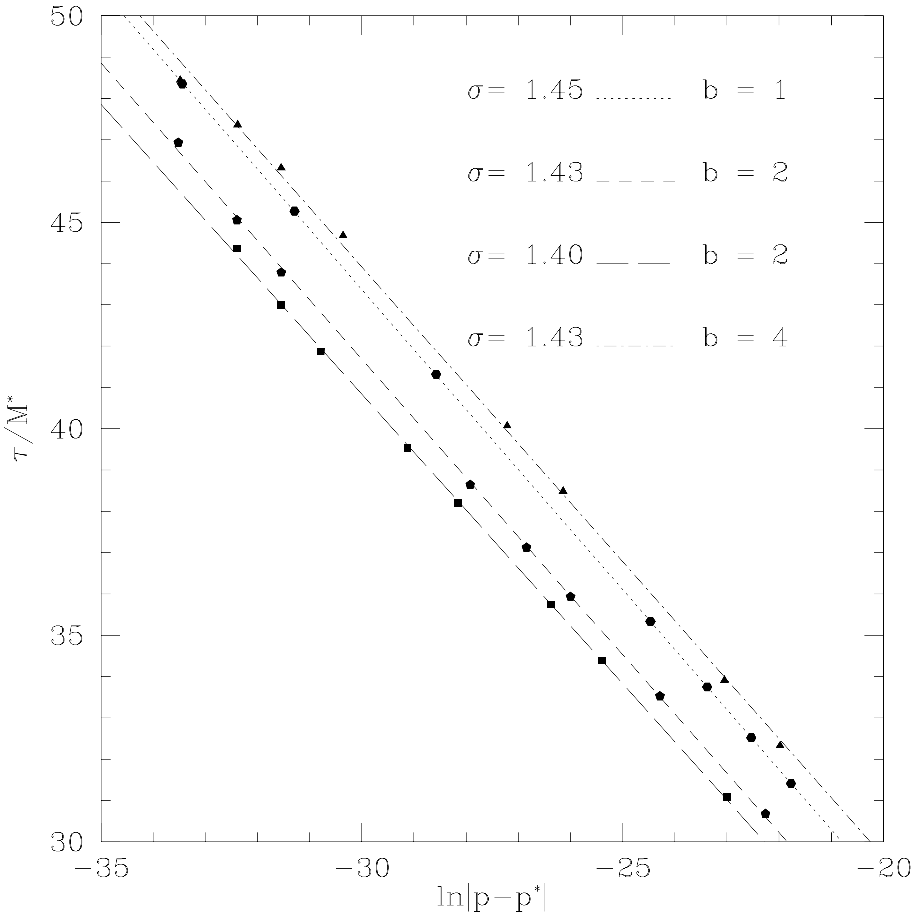


Figure 2.12: Lifetime scaling computed from families of initial data based on the static configurations plotted in Figs. 2.9 and 2.10. The tuning parameter in this instance controls the amplitude of a perturbation that is added to the base solution (here we used the form δf_1 (2.75)) and, in all cases, the sign of the perturbation determines whether the evolution leads to dispersal or black hole formation. The results shown here provide evidence that the static configurations calculated from the ansatz (2.70) act as type I critical solutions. Additionally, we see that there is very little variation in the measured scaling exponents, σ, which are again determined via least squares fits to (2.1).

exclude the possibility that σ is truly universal for the massless-sector critical solutions which we have constructed. Particularly given the variation in the spacetime geometries involved, constancy of the eigenvalue of the unstable mode associated with criticality would be truly remarkable. However, even if σ does span some finite range, the apparent tightness of that range is an aspect of critical behaviour in the massless system that begs understanding.

Finally, we note that the static critical solutions from the generic calculations are characterized by compactness, $\Gamma \sim 0.8$, which is at the low end of the range spanned by the explicitly static solutions. We do not yet know whether a more extensive parameter space survey of generic data could produce critical configurations with larger Γ, and it would be interesting to further investigate this issue.

2.5.3 Generic Massive Case

Following previous studies [34–36], we have also examined the case where the particles have rest mass and find results that are in general agreement with the earlier work, including strong evidence for the existence of static solutions at the black hole threshold that exhibit lifetime scaling. However, we note that in both [35] and [36] the initial data configurations were kinetic energy dominated. For example, a typical calculation in [35] used unit particle mass and $f(0, r, p_r, l)$ which was gaussian in the three coordinates with characteristic values $r \sim 3$, $p_r \sim 1$ and $l \sim 3$. From expression (2.35) for the particle energy we can thus infer that the initial data sets had kinetic energy about an order of magnitude larger than rest mass energy. Thus we expect that those previous results should be similar to what we see for massless particles. Indeed, taking into account the different time parameterization used (t normalized to coincide with property time at infinity), the scaling exponents quoted in [35] are consistent with our results.

Table 2.4 lists the values of the time scaling exponent we have determined in the massive case for the various types of initial data defined in Table 2.1. We note that the initial data families that are used include ones that are very similar to those adopted in [35] and [36]. We see that the time scaling exponents are in fact close to those measured in the massless calculations, although the spread in the values is noticeably larger here (as it was in [35] and [36]). This increased spread is almost certainly due to the particle mass—i.e. the evolutions are not *completely* kinetic energy dominated.

Paralleling what was done in Sec.2.5.2, as well as in [43], we can use perturbations of our explicitly static solutions in the massive sector to investigate critical behaviour. Here there is a larger function space of static configurations, especially since we can construct solutions with positive binding energy, E_b, defined by

$$E_b \equiv M_0 - M \, , \tag{2.79}$$

where M_0 is the total rest mass and M is the ADM mass. Moreover, we can build parameterized sequences of solutions that transition between positive and negative E_b, completely analogously to what can be done for perfect fluid models of general relativistic stars. As in the perfect fluid case, we anticipate that: 1) solutions with $E_b > 0$ will be perturbatively stable, 2) there will be a change of stability at $E_b = 0$, and 3) for at least some range of $E_b < 0$, the static configurations will be one-mode unstable, and thus should constitute type I critical solutions. We have performed additional calculations that confirm these expectations. In particular, we were able to build a static solution with E_b negative, but relatively close to 0, which *did* lie at the black hole threshold and which had an associated scaling exponent $\sigma = 3.0 \pm 0.1$. This value of σ is clearly distinct from those listed in Table 2.4. Thus, in contrast to the massless case where we can not conclusively state anything about possible variations in σ for type I critical solutions, we are confident that σ *is* is not universal in the massive case. In fact, were we able to construct static configurations with $E_b \to 0^-$, we assume that we would find $\sigma \to \infty$. Again, these observations and conjectures are entirely consistent with previous studies of the Einstein-Vlasov

Family	l_0	Z_c	σ	Family	l_0	Z_c	σ
G1	5	2.47	1.32 ± 0.14	G1	12	2.28	1.46 ± 0.07
G1	6	2.39	1.47 ± 0.13	G2	-	2.39	1.44 ± 0.09
G1	7	2.31	1.44 ± 0.08	G3	9	2.29	1.54 ± 0.07
G1	8	2.37	1.49 ± 0.08	G4	9	2.43	1.49 ± 0.08
G1	9	2.41	1.49 ± 0.08	G8	10	2.24	1.38 ± 0.14
G1	10	2.34	1.48 ± 0.07	G9	10	2.41	1.59 ± 0.15
G1	11	2.23	1.54 ± 0.07				

Table 2.4: Summary of measured lifetime scaling exponents for the massive Einstein-Vlasov model from experiments using the various initial data families enumerated in Table 2.1. The results quoted here derive from calculations that parallel those described in Table 2.2 for the massless system. In contrast to the massless case, the observed variation in σ is significant.

system, as well as work with gravitationally compact stars modelled with perfect fluids or bosonic matter.

2.6 Summary and Discussion

We have constructed a new numerical code to evolve the Einstein-Vlasov system in spherical symmetry using an algorithm where the distribution function $f(t, r, p_r, l^2)$ is directly integrated using finite volume methods. This approach eliminates the statistical uncertainty inherent in the particle-based techniques that have been used in previous studies. To reduce computational demands at a given discretization or, more importantly, to allow for higher resolution, we can also run the code in a 2D mode where l^2 is some fixed scalar constant so that f depends on only r and p_r.

We have used the code to perform extensive and detailed surveys of the critical behaviour in the model with a particular focus on the case where the particles are massless. We note that we are unaware of any previous dynamical numerical calculations pertaining to the massless sector.

Our results derive from two classes of initial configurations. In the first the initial states represent imploding shells of particles well removed from the origin, while the second involves perturbations of configurations that are precisely static by construction. Although time-independent solutions of the massive system have been constructed and analyzed previously, to our knowledge the static states we have found in the massless sector are the first of their kind. Within each class we have studied numerous specific forms for the initial data and, for the near-static calculations, the perturbations that are applied to generate the threshold behaviour. In all cases we find strong evidence for a Type I critical transition including: 1) a finite black hole mass at threshold and 2) lifetime scaling of the form (2.1). The observations are all consistent with the standard picture for Type I behaviour, namely a static critical solution with one unstable perturbative mode. Here we emphasize that—as is the case for any numerical study of critical behaviour—it is very difficult to preclude the existence of additional unstable modes. However, the degree to which the scaling laws are satisfied suggests that if such modes do exist they have growth rates significantly smaller than the dominant one.

For generic initial data with massless particles, we have found that there is a considerable variation in the morphology of f among the different critical solutions we have computed and, to a lesser extent, in the details of the spacetime geometries encoded in $a(t,r)$ and $\alpha(t,r)$. Interestingly though, there is relatively little variation in the time scaling exponents that we have measured: all seem to be in the range $1.3 \lesssim \sigma \lesssim 1.5$.

In the case of near-static initial conditions with $m = 0$ the key results are quite similar. Again, there is a large variation in the functional form of the distribution function at threshold. In this instance this can be seen as a direct reflection of the freedom inherent in the ansatz (2.70) which involves the specification of two essentially arbitrary functions. Not surprisingly, there is thus a more noticeable range in the geometries at criticality relative to the generic calculations, as can be clearly seen, for example, through examination of quantities such as the compactness and central redshift. Once again, however, we observe only a small dispersion in the measured scaling exponents. Specifically, across all near-static families that we have examined we find $\sigma = 1.43 \pm 0.07$.

Thus, considering *all* of the calculations that we have performed, we have indications of at least a weak form of universality of the time-scaling exponent in the massless Einstein-Vlasov model. Here we note that as mentioned in the introduction, the calculations reported in [35] were also suggestive of a universal value of σ and perhaps of the critical geometry. Those computations used a non-zero mass and, as also discussed previously, the work of [37, 43] established that the spacetime structure at criticality could *not* be universal in the massive model. However, as noted in Sec. 2.5.3 the initial data families used in [35] were kinetic energy dominated (effectively massless), and so there is no contradiction between what was seen there (and here) and [37, 43].

In all of our calculations, and in accord with Andréasson's proof of the Buchdahl inequality in the model [58], we observe that the gravitational compactness satisfies $\Gamma < 8/9$, with thin shell-like solutions coming closest to saturating the bound.

We also want to emphasize an additional feature of the massless model that is apparent from our calculations: the particle angular momentum does not have a significant impact on the features of the critical solution (apart from the obvious fact that the particles *do* have angular momentum in all of our computations). Heuristically, this can be at least partly ascribed to the scaling symmetry (2.61)–(2.62). The symmetry effectively reduces the number of free parameters—relative to a naive analysis—available for variation in the search for critical solutions. Specifically, given any distribution of the form $f(r, p_r)\delta(l - l_1)$, where l_1 is fixed, we can map to a distribution $f'(r, p_r)\delta(l - l_2)$, with $l_1 \neq l_2$, which has an associated geometry that is diffeomorphic to the original.

Given that there is clearly *no* universality of the fundamental dynamical variables at threshold, the fact that the variation in σ is, at most, small is a feature of the calculations for which we currently have no explanation. Additionally, as discussed in the introduction, the argument advanced in [52] suggests that there should be *no* type I behaviour in the Einstein-Vlasov system for either the massless or massive models. At this time, we do not understand how—if at all—this argument can be reconciled with our current results and those from previous numerical studies.

A direct analysis of the perturbations of the critical solutions—especially the precisely static ones—would be very helpful at this point. Starting with the perfect-fluid work of Koike *et al* [25], perturbation analyses of the critical configurations in many different models have been extremely effective in advancing our understanding of black-hole critical phenomena. In particular, relative to measurements made through direct solution of PDEs and tuning experiments, perturbative methods can provide highly accurate values for the eigenvalues of the unstable modes (or, equivalently, for the scaling exponents). However, in our case the task of explicitly constructing perturbations is significantly complicated by the fact that there is no one-to-one correspondence between the geometry and the phase-space distribution of the particles. So far we have been unable to formulate a well-defined approach to computation of the perturbations and will have to leave that for future work.

Finally, it would be interesting to extend this work to the Einstein-Boltzmann system, where the introduction of explicit interactions between particles would provide the means to investigate the connection between criticality in phase-space-based models and hydrodynamical systems. This in turn might lead to a more fundamental understanding of critical collapse in fluid models.

Chapter 3

Black Hole Critical Behaviour with the Generalized BSSN Formulation

3.1 Introduction

In this paper[29] we investigate the application of the Baumgarte-Shapiro-Shibata-Nakamura (BSSN) formulation of Einstein's equations [31, 32], as well as the dynamical coordinate choices typically associated with it, within the context of critical gravitational collapse. The BSSN formulation is a recasting of the standard 3+1 Arnowitt-Deser-Misner (ADM) [27] equations that is known to be strongly hyperbolic [60, 61] and suitable for numerical studies. It has been widely used in numerical relativity and provides a robust and stable evolution for the spacetime geometry. Most notably, various implementations of this formulation have allowed successful computation of dynamical spacetimes describing binaries of gravitationally-compact objects [3, 13, 14]. The standard gauge choices in BSSN—namely the 1+log slicing condition [62] and the Gamma-driver shift condition [63]—are partial differential equations (PDEs) of evolutionary type. Furthermore, the BSSN approach results in a set of so-called free evolution equations, meaning that the Hamiltonian and momentum constraints are only solved at the initial time. Thus, once initial data has

[29] This chapter is published in: Akbarian A. and Choptuik M. W. "Black hole critical behavior with the generalized BSSN formulation". *Phys. Rev.* D92, 084037 (2015).

been determined, one only has to solve time-dependent PDEs in order to compute the geometric variables in the BSSN scheme. In particular, during the evolution there is no need to solve any elliptic equations, which in general could arise either from the constraints or from coordinate conditions. This is advantageous since it can be quite challenging to implement efficient numerical elliptic solvers.

In addition to the BSSN approach, the numerical relativity community has adopted the generalized harmonic (GH) [64] formulation of Einstein's equations, which is also strongly hyperbolic and has performed very well in simulations of compact binaries [3, 12]. Like BSSN, the GH formulation is of evolutionary type so that all of the metric components satisfy time-dependent PDEs. It too uses dynamical coordinate choices: in this case one needs to provide a prescription for the evolution of the harmonic functions defined by $H^\mu \equiv \Box x^\mu$.

Despite the tremendous success of these hyperbolic formulations in evolving strongly gravitating spacetimes containing black holes and neutron stars, they have not seen widespread use in another area of strong gravity physics typically studied via numerical relativity, namely critical phenomena in gravitational collapse. First reported in [4] and briefly reviewed below, critical phenomena emerge at the threshold of black hole formation and present significant challenges for thorough and accurate computational treatment. The original observation of critical behaviour as well as many of the subsequent studies were restricted to spherical symmetry (for a review, see [5, 23]) and there is a clear need to extend the work to more generic cases. In this respect the BSSN and GH formulations would appear to be attractive frameworks. However, it is not yet clear if these hyperbolic formulations, in conjunction with the standard dynamical gauge choices that have been developed, will allow the critical regime to be probed without the development of coordinate pathologies. Particularly notable in this regard is an implementation of the GH formulation that was employed by Sorkin and Choptuik [65] to study the critical collapse of a massless scalar field in spherical symmetry. Despite extensive experimentation with a variety of coordinate conditions, the code that was developed was not able to calculate near-critical spacetimes: coordinate singularities invariably formed once the critical regime was approached. A natural question that then arises is whether the BSSN formulation (including the standard dynamical gauge choices used with it) is similarly problematic or if it provides an effective framework to study critical phenomena.

Here we begin the task of addressing this question by revisiting the model of spherically symmetric massless scalar collapse. We use a generalization of the BSSN formulation due to Brown [66] that is well suited for use with curvilinear coordinates. The choice of a massless scalar field as the matter source has the great advantage that the nature of the critical solution is very well known [67–72], making it straightforward for us to determine if and when our approach has been successful. We note that although the calculations described below *are* restricted to spherical symmetry our ultimate goal is to develop an evolutionary scheme—including gauge choices—that can be applied to a variety of critical phenomena studies in axial symmetry and ultimately generic cases.

We now briefly review the main concepts and features of black hole critical phenomena that are most pertinent to the work in this paper. Full details and pointers to the extensive literature on the subject may be found in review articles [5, 23].

Critical phenomena in gravitational collapse can be described as a phase transition, analogous to that in a thermodynamical system. Under certain assumptions, a matter source coupled to the Einstein gravitational field will evolve to one of two distinct final phases. On the one hand, weak initial data will eventually disperse to infinity leaving flat spacetime as the end state. On the other hand, sufficiently strong data will develop significant self gravitation and then collapse, resulting in a final phase which contains a black hole. Quite generically, remarkable behaviour emerges at and near the transition between these phases, and this behaviour is precisely what we mean by the critical phenomena in the system under consideration.

It transpires that there are two broad classes of critical phenomena that can be distinguished by the behaviour of the black hole mass at threshold. The class of interest here, known as type II, is characterized by infinitesimal mass at the transition. Further, the black hole mass, $M_{\mathrm{B}H}$, satisfies a scaling law:

$$M_{\mathrm{B}H} \sim |p - p^{\star}|^{\gamma}\,, \tag{3.1}$$

where p is an arbitrary parameter that controls the strength of the matter source at the initial time, $p^{\star}$ is the parameter value at threshold and the mass scaling exponent, γ, is a constant that is independent of the choice of the initial data. Type II behaviour is also characterized by the emergence of a unique solution at threshold which is generically self-similar. In some cases, including the massless scalar field, the self-similarity is discrete. Specifically, in spherically symmetric critical collapse with discrete self-similarity (DSS), as $p \to p^{\star}$ we find

$$Z^{\star}(\rho + \Delta, \tau + \Delta) \sim Z^{\star}(\rho, \tau)\,, \tag{3.2}$$

where $Z^{\star}$ represents some scale-invariant component (function) of the critical solution. Here $\rho \equiv \ln(r_{\mathrm{S}})$ and $\tau \equiv \ln(T_{\mathrm{S}} - T_{\mathrm{S}}^{\star})$ are logarithmically rescaled values of the areal radius, r_{S}, and polar time, T_{S}, respectively, and $T_{\mathrm{S}}^{\star}$ is the accumulation time at which the central singularity associated with the DSS solution forms. T_{S} has been normalized so that it measures proper time at the origin. As with γ, the echoing (rescaling) exponent, Δ, is a universal constant for a specific matter source; i.e. it is independent of the form of the initial data.

Another feature of type II collapse, intimately related to the self-similarity of the critical solution, is that the curvature can become arbitrarily large: in the limit of infinite fine-tuning, $p \to p^{\star}$, a naked singularity forms. Furthermore, the echoing behaviour (3.2) results in the development of fine structure in the solution around the center of the scaling symmetry. Observing this structure and measuring the echoing exponent Δ associated with it requires a code that can reliably evolve solutions very close to the critical spacetime and that provides sufficient numerical resolution in the vicinity of the accumulation point $(r_{\rm S}, T_{\rm S}) = (0, T_{\rm S}^{\star})$.

As mentioned above, most studies of critical phenomena have assumed spherical symmetry. This is particularly so for the case of type II behaviour where the resolution demands dictated by the self-similarity of the critical solutions makes multi-dimensional work extremely computationally intensive. As far as we know, the only work in spherical symmetry to have used a purely evolutionary approach based on the BSSN or GH forms of the Einstein equations is [65] which, as we have noted, was not successful in isolating the critical solution.[30] In axisymmetry there have been two investigations of type II collapse of massless scalar fields [74, 75], and several of type II collapse of pure gravitational waves (vacuum) [76–80]. Of these, only Alcubierre et al.'s [77] and Sorkin's [80] calculations of vacuum collapse adopted hyperbolic formalisms, and only the scalar field calculations—which employed a modified ADM formulation and partially constrained evolution—were able to completely resolve the critical behaviour, including the discrete self-similarity of the critical solutions. In the fully three-space dimension (3D) context there have also been a few studies of type II collapse to date. Perhaps most notable is the recent work of Healy and Laguna [81] which used a massless scalar field as a matter source and the BSSN formulation with standard dynamical gauge choices. The authors were able to observe the mass scaling (3.1) with a measured $\gamma \approx 0.37$ consistent with calculations in spherical symmetry. However, they were not able to conclusively see the discrete self-similarity of the critical solution; in particular they could not accurately measure the echoing exponent, Δ. This shortcoming was attributed to a lack of computational resources rather than a breakdown of the underlying methodology, including the coordinate conditions that were adopted. Finally, there

[30] However, see [73] for an investigation of type II behaviour in the collapse of a scalar field in 2+1 AdS spacetime that employs an ad hoc free evolution scheme.

have been two attempts to probe the black hole threshold for the collapse of pure gravitational waves in 3D [82, 83]. Both employed a BSSN approach with, for the most part, standard dynamical gauge choices. In both cases problems with the gauge apparently precluded calculation near the critical point (although resolution limitations may also have been an issue) and neither the mass scaling nor the echoing exponent could be be estimated in either study.

We can thus summarize the state of the art in the use of hyperbolic formulations for the study of type II critical collapse as follows: to our knowledge there has been no implementation of a fully evolutionary scheme, based on either BSSN or GH, that has allowed for evolution sufficiently close to a precisely critical solution to allow the unambiguous identification of discrete self-similarity (or continuous self-symmetry for that matter). Again, and particularly in light of the experience of [65], the key aim of this paper is to investigate the extent to which it *is* possible to use a BSSN scheme to fully resolve type II solutions. A major concern here is the appropriate choice of coordinate conditions, not least since dynamical gauge choices can be prone to the development of gauge shocks and other types of coordinate singularities [84, 85]. Such pathologies could, in principle, prevent a numerical solver from evolving the spacetime in or near the critical regime.

Now, as Garfinkle and Gundlach have discussed in detail [86], an ideal coordinate system for numerical studies of type II collapse is one which *adapts* itself to the self-similarity: for the DSS case this means that the metric coefficients and relevant matter variables are exactly periodic in the coordinates in the fashion given by (3.2). Clearly, if the coordinate system *is* adapted, then other than at the naked singularity—which is inaccessible via finite-precision calculations—it should remain non-singular during a numerical evolution. One can then argue that ensuring that the numerical scheme has adequate resolution will be the key to successful simulation of the critical behaviour. At the same time, it is also clear that there will be coordinate systems which do not necessarily adapt but which nonetheless remain non-singular during critical collapse, at least over some range of scales, and which are therefore potentially useful for numerical calculations. We will see below that there is strong evidence that the coordinate systems we have used belong to the latter class, and weaker evidence that they *do* adapt to the self-similarity.

Another potential source of problems, which is not specific to hyperbolic formulations, relates to our restriction to spherical symmetry. As is well known, the singular points of curvilinear coordinate systems, $r = 0$ in our case, can sometimes require special treatment to ensure that numerical solutions remain regular there. In critical collapse the highly dynamical nature of the solution near $r = 0$ might naturally be expected to exacerbate problems with regularity. In the work described below we have paid special attention to the ability of our approach to both fully resolve the near-critical configuration and maintain regularity of the solution at the origin.

The remainder of this paper is organized as follows: in Sec. 3.2 we review the generalized BSSN formulation and display the equations of motion for our model system. Sec. 3.3 expands the discussion of the issue of regularity at the coordinate singularity point, describes the numerical approach we have adopted, and provides details concerning the various tests and diagnostics we have used to validate our implementation. In Sec. 3.4 we present results computed using two distinct choices for the shift vector and provide conclusive evidence that the generalized BSSN formulation is capable of evolving in the critical regime in both cases. Sec. 3.5 contains some brief concluding remarks, and further details concerning the BSSN formalism in spherical symmetry and the scalar field equations of motion are included in App. 3.6 and App. 3.7, respectively. We adopt units where the gravitational constant and the speed of light are both unity: $G = c = 1$.

3.2 Equations of Motion

The dynamical system we intend to study in the critical collapse regime is a real, massless scalar field, Ψ, self gravitating via Einstein's equations,

$$G_{\mu\nu} = 8\pi T_{\mu\nu}\,. \tag{3.3}$$

Here, $T_{\mu\nu}$ is the energy-momentum tensor associated with the minimally coupled Ψ:

$$T_{\mu\nu} = \nabla_\mu \nabla_\nu \Psi - \frac{1}{2} g_{\mu\nu} \nabla^\eta \Psi \nabla_\eta \Psi\,, \tag{3.4}$$

and the evolution of the scalar field is given by

$$\nabla^\mu \nabla_\mu \Psi = 0\,. \tag{3.5}$$

The time-development of the geometry is then given by recasting Einstein's equations as an evolution system based on the usual 3+1 expression for the spacetime metric:

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt)\,. \tag{3.6}$$

Here, the 3-metric components, γ_{ij}, are viewed as the fundamental dynamical geometrical variables and the lapse function, α, and shift vector, β^i, which encode the coordinate freedom of general relativity, must in general be prescribed independently of the equations of motion.

3.2.1 Generalized BSSN

We now summarize the BSSN formulation of Einstein's equations and describe how it can be adapted to curvilinear coordinates. Readers interested in additional details are directed to [28] for a more pedagogical discussion.

In the standard ADM formulation [27, 87], the dynamical Einstein equations are rewritten as evolution equations for the 3-metric and the extrinsic curvature $\{\gamma_{ij}, K_{ij}\}$. The first difference between the BSSN formulation and the ADM decomposition is the conformal re-scaling of the ADM dynamical variables:

$$\gamma_{ij} = e^{4\phi}\tilde{\gamma}_{ij} \,, \tag{3.7}$$

$$K_{ij} = e^{4\phi}\tilde{A}_{ij} + \frac{1}{3}\gamma_{ij}K \,, \tag{3.8}$$

where e^{ϕ} is the conformal factor, $\tilde{\gamma}_{ij}$ is the conformal metric, $\tilde{A}_{ij}$ is the conformally rescaled trace-free part of the extrinsic curvature and $K = \gamma^{ij}K_{ij}$ is the trace of the extrinsic curvature. Here by fixing the trace of $\tilde{A}_{ij}$, and the determinant of the conformal metric, the set of primary ADM dynamical variables transforms to the new set:

$$\{\gamma_{ij}, K_{ij}\} \rightarrow \{\phi, \tilde{\gamma}_{ij}, K, \tilde{A}_{ij}\} \,, \tag{3.9}$$

in the BSSN formulation.

In the original BSSN approach, the conformal metric $\tilde{\gamma}_{ij}$ is taken to have determinant $\tilde{\gamma} = 1$. However this choice is only suitable when we adopt coordinates in which the determinant of the flat-space metric reduces to unity. This is the case, of course, for Cartesian coordinates but is not so for general curvilinear systems. For instance, the flat 3-metric in spherical coordinates:

$$ds^2 = dr^2 + r^2 d\theta^2 + r^2 \sin^2\theta d\phi^2 \,, \tag{3.10}$$

has determinant $\mathring{\gamma} = r^4 \sin^2\theta$. Recently, Brown [66] has resolved this issue by introducing a covariant version of the BSSN equations—the so-called *generalized* BSSN formulation, which we will hereafter refer to as G-BSSN—in which the primary dynamical variables are tensors so that the formulation can be adapted to non-Cartesian coordinate systems. In G-BSSN we no longer assume that the conformal

3-metric has determinant one. Rather, ϕ becomes a true scalar and for its dynamics to be determined a prescription for the time evolution of the determinant of $\tilde{\gamma}_{ij}$ must be given. In the following this will be done by requiring that the determinant be constant in time.

Another main difference between the ADM decomposition and BSSN is that the mixed spatial derivative terms occurring in the 3-Ricci tensor are eliminated through the definition of a new quantity, $\tilde{\Gamma}^k$:

$$\tilde{\Gamma}^k \equiv \tilde{\gamma}^{ij}\tilde{\Gamma}^k_{ij} \,, \tag{3.11}$$

which becomes an additional, independent dynamical variable. Note that $\tilde{\Gamma}^i$ is not a vector as it is coordinate dependent. To extend this redefinition so that it is well suited for all coordinate choices, in G-BSSN we define

$$\tilde{\Lambda}^k \equiv \tilde{\gamma}^{ij}(\tilde{\Gamma}^k_{ij} - \mathring{\Gamma}^k_{ij}) = \tilde{\Gamma}^k - \mathring{\Gamma}^k_{ij}\tilde{\gamma}^{ij} \,, \tag{3.12}$$

where $\mathring{\Gamma}^k_{ij}$ denotes the Christoffel symbols associated with the flat metric. This definition makes this so-called conformal connection, $\tilde{\Lambda}^i$, a true vector and it becomes a primary dynamical variable in G-BSSN.

We now summarize the G-BSSN equations, referring the reader to [88] for more details, including a full derivation. We begin by defining $\partial_\perp$, the time derivative operator acting normally to the $t =$ const. slices:

$$\partial_\perp \equiv \partial_t - \mathscr{L}_{\vec{\beta}} \,, \tag{3.13}$$

where $\mathscr{L}_{\vec{\beta}}$ denotes the Lie derivative along $\vec{\beta}$. We then have

$$\partial_\perp \phi = -\frac{1}{6}\alpha K + \sigma\frac{1}{6}\tilde{D}_k\beta^k \,, \tag{3.14}$$

$$\partial_\perp \tilde{\gamma}_{ij} = -2\alpha \tilde{A}_{ij} - \sigma \frac{2}{3} \tilde{A}_{ij} \tilde{D}_k \beta^k \,, \tag{3.15}$$

$$\partial_\perp K = -\gamma^{ij} D_j D_i \alpha + \alpha(\tilde{A}_{ij}\tilde{A}^{ij} + \frac{1}{3}K^2) + 4\pi(\rho + S) \,, \tag{3.16}$$

$$\begin{aligned} \partial_\perp \tilde{A}_{ij} &= e^{-4\phi} \left[-D_i D_j \alpha + \alpha(R_{ij} - 8\pi S_{ij}) \right]^{\mathrm{TF}} \\ &+ \alpha(K\tilde{A}_{ij} - 2\tilde{A}_{il}\tilde{A}^l{}_j) - \sigma \frac{2}{3} \tilde{A}_{ij} \tilde{D}_k \beta^k \,, \end{aligned} \tag{3.17}$$

$$\begin{aligned} \partial_\perp \tilde{\Gamma}^i &= -2\tilde{A}^{ij}\partial_j \alpha + \tilde{\gamma}^{lj} \partial_j \partial_l \beta^i \\ &+ 2\alpha \left(\tilde{\Gamma}^i_{jk} \tilde{A}^{kj} - \frac{2}{3} \tilde{\gamma}^{ij} \partial_j K + 6\tilde{A}^{ij}\partial_j \phi - 8\pi \tilde{\gamma}^{ij} S_j \right) \\ &+ \frac{\sigma}{3} \left[2\tilde{\Gamma}^i \tilde{D}_k \beta^k + \tilde{\gamma}^{li} \partial_l (\tilde{D}_k \beta^k) \right] \,. \end{aligned} \tag{3.18}$$

Here, a superscript TF denotes the trace-free part (with respect to the 3-metric γ_{ij}) of a tensor, and $\tilde{D}_i$ is the covariant derivative associated with the conformal metric $\tilde{\gamma}_{ij}$. Additionally, the quantity σ is an adjustable parameter that is discussed below and typically is either 0 or 1. Note that all the Lie derivatives in the G-BSSN equations operate on true tensors and vectors of weight 0. For instance,

$$\mathscr{L}_{\vec{\beta}} \tilde{A}_{ij} = \beta^k \partial_k \tilde{A}_{ij} + \tilde{A}_{ik} \partial_j \beta^k + \tilde{A}_{kj} \partial_i \beta^k \,. \tag{3.19}$$

Furthermore, in G-BSSN, rather than evolving (3.18), the redefined conformal connection, $\tilde{\Lambda}^i$, is evolved via

$$\partial_t \tilde{\Lambda}^k = \partial_t \tilde{\Gamma}^k - \mathring{\Gamma}^k_{ij} \partial_t \tilde{\gamma}^{ij} \,, \tag{3.20}$$

where the time derivative $\partial_t \tilde{\gamma}^{ij}$ is eliminated using (3.15). In equation (3.17), R_{ij}

denotes the 3-Ricci tensor associated with γ_{ij} and can be written as the sum

$$R_{ij} = R^{\phi}_{ij} + \tilde{R}_{ij} \,, \tag{3.21}$$

where R^{ϕ}_{ij} is given by

$$R^{\phi}_{ij} = -2\tilde{D}_i\tilde{D}_j\phi - 2\tilde{\gamma}_{ij}\tilde{D}^k\tilde{D}_k\phi + 4\tilde{D}_i\phi\tilde{D}_j\phi - 4\tilde{\gamma}_{ij}\tilde{D}^k\phi\tilde{D}_k\phi \,, \tag{3.22}$$

and $\tilde{R}_{ij}$ is the 3-Ricci tensor associated with the conformal metric:

$$\begin{aligned} \tilde{R}_{ij} = & - \frac{1}{2}\tilde{\gamma}^{lm}\partial_m\partial_l\tilde{\gamma}_{ij} + \tilde{\gamma}_{k(i}\partial_{j)}\tilde{\Gamma}^k + \tilde{\Gamma}^k\tilde{\Gamma}_{(ij)k} \\ & + \tilde{\gamma}^{lm}\left(2\tilde{\Gamma}^k_{l(i}\tilde{\Gamma}_{j)km} + \tilde{\Gamma}^k_{im}\tilde{\Gamma}_{klj}\right) \,. \end{aligned} \tag{3.23}$$

The matter fields ρ, S, S_i and S_{ij} are defined by

$$\rho = n_\mu n_\nu T^{\mu\nu} \,, \tag{3.24}$$

$$S = \gamma^{ij} S_{ij} \,, \tag{3.25}$$

$$S^i = -\gamma^{ij} n^\mu T_{\mu j} \,, \tag{3.26}$$

$$S_{ij} = \gamma_{i\mu}\gamma_{i\nu}T^{\mu\nu} \,, \tag{3.27}$$

where n^μ is the unit normal vector to the $t =$ const. slices.

As mentioned previously, we need to prescribe dynamics for the determinant of $\tilde{\gamma}_{ij}$ to have a complete set of equations of motion for the G-BSSN dynamical variables. One approach is to fix the determinant to its initial value by demanding that

$$\partial_t \tilde{\gamma} = 0 \,. \tag{3.28}$$

This is the so-called Lagrangian option and is associated with the choice $\sigma = 1$ in the equations. Another option is to define the determinant to be constant along the normal direction to the time slices, which can be implemented by requiring $\partial_\perp \tilde{\gamma} = 0$. This is usually referred to as the Lorentzian option, and is associated with the choice $\sigma = 0$. Here we choose (3.28), i. e. $\sigma = 1$.

Note that in the G-BSSN equations the divergence of the shift vector,

$$\tilde{D}_k \beta^k = \frac{1}{\sqrt{\tilde{\gamma}}} \partial_k (\sqrt{\tilde{\gamma}} \beta^k) \,, \tag{3.29}$$

no longer reduces to $\partial_k \beta^k$ since the determinant of the conformal metric $\tilde{\gamma}_{ij}$ is not necessarily 1, but by virtue of the choice (3.28) is equal to that of the initial background flat metric in the chosen curvilinear coordinates.

As usual, when setting initial data for any given evolution of the coupled Einstein-matter equations we must solve the Hamiltonian and momentum constraints. In terms of the G-BSSN variables these are

$$\begin{aligned} \mathscr{H} &\equiv \tilde{\gamma}^{ij} \tilde{D}_i \tilde{D}_j e^{\phi} - \frac{e^{\phi}}{8} \tilde{R} + \frac{e^{5\phi}}{8} \tilde{A}^{ij} \tilde{A}_{ij} \\ &- \frac{e^{5\phi}}{12} K^2 + 2\pi e^{5\phi} \rho = 0 \,, \end{aligned} \tag{3.30}$$

$$\mathscr{M}^i \equiv \tilde{D}_j \left(e^{6\phi} \tilde{A}^{ji} \right) - \frac{2}{3} e^{6\phi} \tilde{D}^i K - 8\pi e^{6\phi} S^i = 0 \,. \tag{3.31}$$

3.2.2 G-BSSN in Spherical Symmetry and Gauge Choices

In spherical symmetry a generic form of the conformal metric $\tilde{\gamma}_{ij}$ is given by

$$\tilde{\gamma}_{ij} = \begin{pmatrix} \tilde{\gamma}_{rr}(t,r) & 0 & 0 \\ 0 & r^2\tilde{\gamma}_{\theta\theta}(t,r) & 0 \\ 0 & 0 & r^2\tilde{\gamma}_{\theta\theta}(t,r)\sin^2\theta \end{pmatrix} . \tag{3.32}$$

Similarly, a suitable ansatz for the traceless extrinsic curvature is

$$\tilde{A}_{ij} = \begin{pmatrix} \tilde{A}_{rr}(t,r) & 0 & 0 \\ 0 & r^2\tilde{A}_{\theta\theta}(t,r) & 0 \\ 0 & 0 & r^2\tilde{A}_{\theta\theta}(t,r)\sin^2\theta \end{pmatrix} . \tag{3.33}$$

The shift vector and $\tilde{\Lambda}^i$ have only radial components:

$$\beta^i = [\beta(t,r), 0, 0] \ , \tag{3.34}$$

$$\tilde{\Lambda}^i = \left[\tilde{\Lambda}(t,r), 0, 0\right] \ . \tag{3.35}$$

Given (3.32-3.35), the G-BSSN equations become a set of first order evolution equations for the 7 primary variables

$$\Big\{\phi(t,r), \tilde{\gamma}_{rr}(t,r), \tilde{\gamma}_{\theta\theta}(t,r), K(t,r), \\ \tilde{A}_{rr}(t,r), \tilde{A}_{\theta\theta}(t,r), \tilde{\Lambda}(t,r)\Big\} \ .$$

These are coupled to the evolution equation (3.5) for the scalar field and constrained by the initial conditions (3.30–3.31). The explicit expressions for the full set of equations of motion are given in App. 3.6.

To fix the time slicing we implement a non-advective[31] version of the 1+log

[31] The terminology *non-advective* derives from the absence of an "advective" term, $\beta^j\partial_j$, on the

slicing condition:[32]

$$\partial_t \alpha = -2\alpha k\,. \tag{3.36}$$

for the spatial coordinates we either choose a zero shift:

$$\beta^i = 0\,, \tag{3.37}$$

or use what we will term the gamma-driver condition:

$$\partial_t \beta^i = \mu \tilde{\Lambda}^i - \eta \beta^i\,. \tag{3.38}$$

Here, μ and η are adjustable parameters which we set to $\mu = 3/4$ and $\eta \simeq 1/(2M_{\mathrm{ADM}})$, where M_{ADM} is the total mass of the system measured at infinity (see Sec. 3.3.4). We emphasize that (3.38) is *not* the usual Gamma-driver equation used in the standard BSSN approach:

$$\partial_t \beta^i = \mu \tilde{\Gamma}^i - \eta \beta^i\,, \tag{3.39}$$

but since it is a natural extension of the above to the G-BSSN case we have opted to use the same nomenclature. In the rest of this paper, we frequently refer to the shift vector evolved via (3.38) as β^G. Explicitly, in spherical symmetry β^G is defined by

$$\partial_t \beta^G(t,r) = \mu \tilde{\Lambda}(t,r) - \eta \beta^G(t,r)\,. \tag{3.40}$$

left hand side of equations (3.36,3.38). we note that we also experimented with the advective versions of the equations. the results were very similar to those for the non-advective case; in particular, near-critical solutions exhibiting echoing and scaling could also be obtained.

[32] The reader can easily check that in the case of zero shift, the lapse choice given by (3.36) combined with (3.14) implies $\partial_t(\alpha - 12\phi) = 0$. In Cartesian coordinates $12\phi = \ln\gamma$, so this last equation gives $\alpha - \ln\gamma = c(\vec{x})$, where the function $c(\vec{x})$ is time independent. the choice $c(\vec{x}) = 1$ then yields an algebraic expression for the lapse, $\alpha = 1 + \ln\gamma$, which is the origin of the terminology "1+log slicing".

3.3 Numerics

We use a second order finite differencing method to discretize equations (3.14-3.17) and (3.20). Further, the equations of motion are transformed to a compactified radial coordinate that we denote by $\tilde{r}$ and which is defined in terms of the original coordinate r by

$$r = e^{\tilde{r}} - e^{\delta} + \frac{R_\infty}{R_\infty - \tilde{r}} - \frac{R_\infty}{R_\infty - \delta}\,, \tag{3.41}$$

where δ and R_∞ are parameters with typical values $\delta \simeq -12$ and $R_\infty \simeq 3$. It is straightforward to verify the following:
1) the radial domain $r = (0, \infty)$ maps to the computational domain $\tilde{r} = (\delta, R_\infty)$,
2) the derivative $dr/d\tilde{r}$ decreases toward the origin ($\tilde{r} \simeq \delta$), so that a uniform grid on $\tilde{r}$ is a non-uniform grid on r with approximately 10^3 times more resolution close to the origin relative to the outer portion of the solution domain, $\tilde{r} \simeq 2$ ($r \simeq 10$), where the support of the scalar field is initially concentrated,
3) the parameter δ can be used to adjust the resolution near the origin; specifically, decreasing δ increases the resolution near $r = 0$. For notational simplicity, however, in the following we omit the explicit dependence of the fields on $\tilde{r}$ and denote the spacetime dependence of any dynamical variable X as previously: $X\left(t, r(\tilde{r})\right) \equiv X(t, r)$.

We use a finite difference grid that is uniform in $\tilde{r}$ and analytically transform all r-derivative terms in the equations of motion to their $\tilde{r}$-coordinate counterparts prior to finite-differencing.

We also developed a Maple-based toolkit [89] that automates the process of discretizing an arbitrary derivative expression. This toolkit handles boundary conditions and generates a point-wise Newton-Gauss-Seidel solver in the form of Fortran routines for a given set of time dependent or elliptic PDEs . The calculations in this paper were all carried out using this infrastructure.

3.3.1 Initialization

The matter content is set by initializing the scalar field to a localized Gaussian shell:

$$\Psi(0,r) = p\exp\left(-\frac{(r-r_0)^2}{\sigma_r^2}\right)\,, \tag{3.42}$$

where p, r_0 and σ_r are parameters. Note that here r is the non-compactified radial coordinate which is related to the compactified coordinate $\tilde{r}$ via (3.41). A typical initial profile for the scalar field in our calculations has $\sigma_r \simeq 1$, $r_0 \simeq 10$, and p of order 10^{-1}. We use the overall amplitude factor p as the tuning parameter to find critical solutions. We initialize the conformal metric (3.32) to the flat metric in spherical symmetry,

$$\tilde{\gamma}_{rr}(0,r) = \tilde{\gamma}_{\theta\theta}(0,r) = 1\,, \tag{3.43}$$

and initialize the lapse function to unity,

$$\alpha(0,r) = 1\,. \tag{3.44}$$

We also demand that the initial data be time-symmetric,

$$\tilde{A}_{rr}(0,r) = \tilde{A}_{\theta\theta}(0,r) = K(0,r) = 0\,, \tag{3.45}$$

$$\beta(0,r) = \tilde{\Lambda}(0,r) = 0\,, \tag{3.46}$$

$$\partial_t\Psi(t,r)|_{t=0} = 0\,, \tag{3.47}$$

which means that the momentum constraint (3.31) is trivially satisfied. This leaves the Hamiltonian constraint (3.30) which is solved as a two-point boundary value problem for the conformal factor at the initial time,

$$\psi(r) \equiv e^{\phi(0,r)}\,. \tag{3.48}$$

The outer boundary condition for ψ,

$$\psi(r)|_{r=\infty} = 1\,, \tag{3.49}$$

follows from asymptotic flatness, while at $r = 0$ we have

$$\partial_r \psi(r)|_{r=0} = 0 \tag{3.50}$$

since $\psi(r)$ must be an even function in r for regularity at the origin.

3.3.2 Boundary Conditions

Due to the fact that the metric has to be conformally flat at the origin we have

$$\tilde{\gamma}_{rr}(t,0) = \tilde{\gamma}_{\theta\theta}(t,0)\,. \tag{3.51}$$

Further, since we are using the Lagrangian choice, $\sigma = 1$, the determinant of $\tilde{\gamma}_{ij}$ must at all times be equal to its value at the initial time, so

$$\tilde{\gamma}_{rr}\tilde{\gamma}_{\theta\theta}^2 = 1\,. \tag{3.52}$$

From these two results we have

$$\tilde{\gamma}_{rr}(t,0) = \tilde{\gamma}_{\theta\theta}(t,0) = 1\,. \tag{3.53}$$

Using (3.53) and (3.15) it is then easy to see that we must also have

$$\tilde{A}_{rr}(t,0) = \tilde{A}_{\theta\theta}(t,0) = 0\,. \tag{3.54}$$

As is usual when working in spherical coordinates, many of the boundary conditions that must be applied at $r = 0$ follow from the demand that the solution be regular there. Essentially, the various dynamical variables must have either even or odd "parity" with respect to expansion in r as $r \to 0$. Variables with even parity, typically scalars or diagonal components of rank-2 tensors, must have vanishing radial derivative at $r = 0$, while odd parity functions, typically radial components of vectors, will themselves vanish at the origin.

Applying these considerations to our set of unknowns we find

$$\partial_r \tilde{\gamma}_{rr}(t,r)|_{r=0} = \partial_r \tilde{\gamma}_{\theta\theta}(t,r)|_{r=0} = 0\,, \tag{3.55}$$

$$\partial_r \tilde{A}_{rr}(t,r)|_{r=0} = \partial_r \tilde{A}_{\theta\theta}(t,r)|_{r=0} = 0\,, \tag{3.56}$$

$$\beta(t,0) = \tilde{\Lambda}(t,0) = 0\,, \tag{3.57}$$

$$\partial_r K(t,r)|_{r=0} = \partial_r \phi(t,r)|_{r=0} = \partial_r \Psi(t,r)|_{r=0} = 0\,. \tag{3.58}$$

We use equations (3.53,3.54,3.57) to fix the values of the functions at the origin and a forward finite-differencing of (3.58) to update K, ϕ and Ψ at $r = 0$. Further, we apply a forward finite-differencing of (3.55,3.56) to update the values of the function at the grid point next to the origin. The 1+log condition (3.36) can be used directly at $r = 0$. Again, we emphasize that all of the r-derivative terms of the boundary conditions described above are analytically transformed to the numerical coordinate, $\tilde{r}$, before the equations are finite-differenced.

Since we are using compactified coordinates, all the variables are set to their flat spacetime values at the outer boundary $r = \infty$:

$$\tilde{\gamma}_{rr} = \tilde{\gamma}_{\theta\theta} = e^{\phi} = \alpha = 1 \quad \text{at} : (t, \infty)\,, \tag{3.59}$$

$$\tilde{A}_{rr} = \tilde{A}_{\theta\theta} = K = \tilde{\Lambda} = \beta = \Psi = 0 \quad \text{at} : (t, \infty)\,. \tag{3.60}$$

Here, we emphasize that spatial infinity, $r = \infty$, corresponds to the finite compactified (computational) coordinate point $\tilde{r} = R_\infty$.

3.3.3 Evolution Scheme and Regularity

We implemented a fully implicit, Crank-Nicolson [90] finite differencing scheme to evolve the system of G-BSSN equations. The precise form of the continuum equations used is given in App. 3.6 and all derivatives, both temporal and spatial, were approximated using second-order-accurate finite difference expressions.

During an evolution the correct limiting behaviour of the spatial metric components must be maintained near $r = 0$ to ensure a regular solution. For example, the limiting values of the conformal metric components $\tilde{\gamma}_{rr}$ and $\tilde{\gamma}_{\theta\theta}$ are given by

$$\tilde{\gamma}_{rr}(t, r) = 1 + O(r^2)\,, \tag{3.61}$$

$$\tilde{\gamma}_{\theta\theta}(t, r) = 1 + O(r^2)\,. \tag{3.62}$$

If the discrete approximations of the metric functions do not satisfy these conditions, then irregularity will manifest itself in the divergence of various expressions such as the Ricci tensor component (3.95)

$$R_{rr} = 2\frac{\tilde{\gamma}_{rr} - \tilde{\gamma}_{\theta\theta}}{r^2\tilde{\gamma}_{\theta\theta}} + \cdots\,, \tag{3.63}$$

which should converge to a finite value at the origin if conditions (3.61,3.62) hold.

One approach to resolve potential regularity issues is to regularize the equations [65, 88, 91], by redefining the primary evolution variables, so that the equations become manifestly regular at the origin. Another approach is to use implicit or partially implicit methods [92]. As recently shown by Montero and Cordeo-Carrion [93], such schemes can yield stable evolution without need for explicit regularization. Baumgarte et al. [94] also adopted a similar approach—using a partially implicit scheme without regularization—in an implementation of the G-BSSN formulation in spherical polar coordinates.

As mentioned, our implementation is fully implicit and we have also found that our generalized BSSN equations can be evolved without any need for regularization at the origin, even in strong gravity scenarios where the spacetime metric has significant deviations from flatness near the origin.

That said, we also experimented with other techniques aimed at improving regularity. For example, using the constraint equation (3.52) and the fact that $\tilde{A}_{ij}$ is trace-free,

$$\frac{\tilde{A}_{rr}}{\tilde{\gamma}_{rr}} + 2\frac{\tilde{A}_{\theta\theta}}{\tilde{\gamma}_{\theta\theta}} = 0\,, \tag{3.64}$$

we can compute $\tilde{\gamma}_{\theta\theta}$ and $\tilde{A}_{\theta\theta}$ in terms of $\tilde{\gamma}_{rr}$ and $\tilde{A}_{rr}$, respectively, rather than evolving them. However, when we did this we found no significant improvement in regularity relative to the original scheme.

Finally, to ensure our solutions remain smooth on the scale of the mesh we use fourth order Kreiss-Oliger dissipation [95] in the numerical solution updates.

3.3.4 Tests

This section documents various tests we have made to validate the correctness of our numerical solver as well as the consistency of the finite-differencing method used to evolve the system of G-BSSN equations. We use a variety of diagnostic tools, including monitoring of the constraint equations, convergence tests of the primary dynamical variables, and a direct computation to check if the metric and

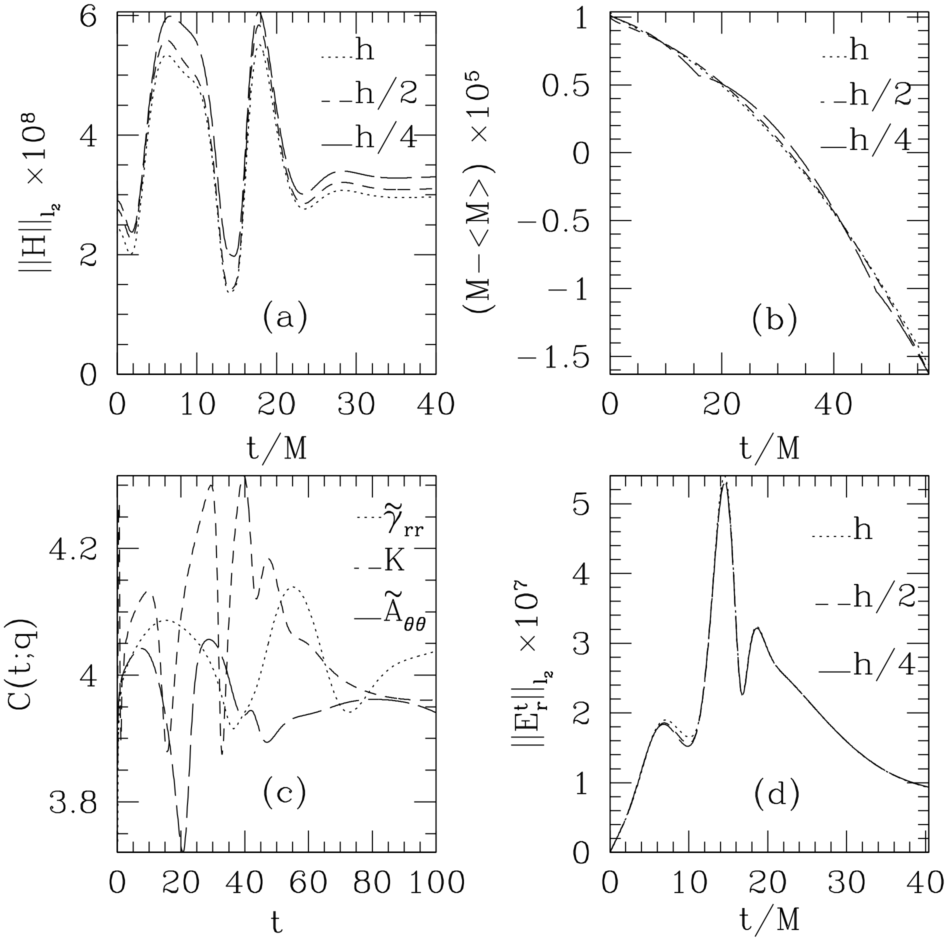


Figure 3.1: Results from various tests that verify the accuracy and consistency of our numerical solver and the finite differencing method used to integrate the equations. (a) The evolution of the l_2-norm (RMS value) of the Hamiltonian constraint. The norm is plotted for 3 different resolutions h, $h/2$ and $h/4$ corresponding to $N_r = 512$, 1024 and 2048, respectively. The data for the $N_r = 1024$ and $N_r = 2048$ curves have been rescaled by factors of 4 and 16, respectively, and the overlap of the three lines thus signals the expected second order convergence to zero of the constraint deviation. We observe similar convergence properties for the momentum constraint as well as the constraint equation (3.12) for $\tilde{\Lambda}^i$, and the constraint (3.64) for the trace of $\tilde{A}_{ij}$. Additionally, since the operator used to evaluate the residual of the Hamiltonian constraint is distinct from that used in the determination of the initial data, the test also validates the initial data solver. (b) Conservation of the ADM mass during the evolution of strong initial data. Here the deviation of the mass from its time average is plotted for 3 different resolutions. Higher resolution values have again been rescaled so that overlap of the curves demonstrates $O(h^2)$ convergence to 0 of the deviation of the total mass.

Figure 3.1: (c) The convergence factor defined in (3.68) for three of the primary BSSN variables: $\tilde{\gamma}_{rr}$, K, and $\tilde{A}_{\theta\theta}$. In the limit $h \to 0$ we expect all curves to tend to the constant 4. The plot thus provides evidence for second order convergence of all of the values throughout the evolution. All of the other primary BSSN variables as well as the dynamical scalar field quantities demonstrate the same convergence. (d) Direct verification that the metric found by numerically solving the BSSN equations satisfies Einstein's equations in their covariant form. Here the tr component of the residual E^{μ}_{ν} defined in (3.72) is plotted for 3 different resolutions. Once more, higher resolution values have been rescaled so that overlap of the curves signals the expected $O(h^2)$ convergence of the residuals to 0. All of the plots correspond to evolution of strong subcritical initial data with 1+log slicing. For (a) and (b) the shift vector was set to 0, while in (c) and (d) it was evolved using the Gamma-driver condition (i.e. $\beta = \beta^G$).

matter fields calculated via the G-BSSN formulation satisfy the covariant form of Einstein's equations. All of the calculations were performed using the 1+log slicing condition (3.36) and either $\beta = 0$ or $\beta = \beta^G$ where β^G satisfies the Gamma-driver condition (3.40).

Constraints and Conserved Quantities

We monitor the evolution of the constraint equations (3.30,3.31) during a strongly-gravitating evolution where the nonlinearities of the equations are most pronounced. As demonstrated in Fig. 3.1 (a), at resolutions typical of those used in our study, the Hamiltonian constraint is well preserved during such an evolution and, importantly, the deviations from conservation converge to zero at second order in the mesh spacing as expected.

The total mass-content of the spacetime seen at spatial infinity (the ADM mass) is a conserved quantity. Here, using the G-BSSN variables the Misner-Sharp mass function is given by

$$M(r) = \frac{r\tilde{\gamma}_{\theta\theta}^{1/2} e^{2\phi}}{2} \left[1 - \frac{\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{rr}} \left(1 + r\frac{\partial_r \tilde{\gamma}_{\theta\theta}}{2\tilde{\gamma}_{\theta\theta}} + 2r\frac{\partial_r e^{\phi}}{e^{\phi}} \right)^2 \right] . \qquad (3.65)$$

The total mass, M_{ADM}, can be evaluated at the outer boundary,

$$M_{\mathrm{ADM}} \equiv M(r = \infty)\,. \tag{3.66}$$

The deviation of the total mass from its time average is plotted in Fig. 3.1(b); as the resolution of the numerical grid increases the variations converge to zero in a second order fashion.

Convergence Test

As mentioned in Sec. 3.3.1 and Sec. 3.3.3, we implemented our code using second order finite differencing of all spatial and temporal derivatives. Denoting any continuum solution component by $q(t, X)$, where X is the spatial coordinate, and a discrete approximation to it computed at finite difference resolution, h, by $q^h(t, X)$, to leading order in h we expect

$$q^h(t, X) = q(t, X) + h^2 e_2[q](t, X) + \ldots\,. \tag{3.67}$$

Fixing initial data, we perform a sequence of calculations with resolutions h, $h/2$ and $h/4$ and then compute a convergence factor, $C(t; q)$, defined by

$$C(t; q) = \frac{||q^h(t, X) - q^{h/2}(t, X)||_2}{||q^{h/2}(t, X) - q^{h/4}(t, X)||_2}\,, \tag{3.68}$$

where $||\cdot||_2$ is the l_2 norm, i.e. the root mean square (RMS) value. It is straightforward to argue from (3.67) that, for sufficiently small h, $C(t, q)$ should approach 4 if the solution *is* converging at second order. The values of the convergence factor for a selection of dynamical variables are plotted for a strong-data evolution in Fig. 3.1(c) and provide clear evidence that the solution is second-order convergent throughout the time evolution.

Direct Validation via Einstein's Equations

A direct method to test the fidelity of our numerical solver involves the evaluation of a residual based on the *covariant* form of Einstein's equations. We start with a reconstruction of the four-dimensional metric in spherical symmetry,

$$ds^2 = (-\alpha^2 + \beta^2 a^2)dt^2 + 2a^2\beta dtdr + a^2dr^2 + r^2b^2d\Omega^2\,, \tag{3.69}$$

using the primary G-BSSN variables, $\tilde{\gamma}_{ij}$ and ϕ. In particular, a and b are simply given by

$$a(t,r) = e^{4\phi(t,r)}\tilde{\gamma}_{rr}(t,r)\,, \tag{3.70}$$

$$b(t,r) = e^{4\phi(t,r)}\tilde{\gamma}_{\theta\theta}(t,r)\,. \tag{3.71}$$

We then check to see if the metric (3.69) satisfies the covariant Einstein equations (3.3) to the expected level of truncation error. Specifically, defining the residual

$$E^{\mu}{}_{\nu} \equiv G^{\mu}{}_{\nu} - 8\pi T^{\mu}{}_{\nu}\,, \tag{3.72}$$

and replacing all derivatives in G^{μ}_{ν} with second order finite differences, we expect E^{μ}_{ν} to converge to zero as $O(h^2)$ as $h \to 0$.[33] Precisely this behaviour is shown in Fig. 3.1(d). This is a particularly robust test of our implementation since the non-trivial components of the covariant Einstein equations are quite complicated and, superficially at least, algebraically independent of the BSSN equations. For

[33] Although it is not crucial for the usefulness of this test, we discretize the E^{μ}_{ν} using a difference scheme that is distinct from the one used in the main code.

instance, the tr component of the residual (3.72) is given by

$$\begin{aligned}
E^t{}_r &= \frac{2\beta}{r\alpha^2}\left(\frac{\partial_r a}{a} - 2\frac{\partial_r b}{b} + \frac{\partial_r \alpha}{\alpha}\right) \\
&+ \frac{2\beta}{\alpha^2}\left(-\frac{\partial_r^2 b}{b} + \frac{\partial_r a \partial_r b}{ab} - \frac{(\partial_r \Psi)^2}{2} + \frac{\partial_r \alpha \partial_r b}{\alpha b}\right) \\
&+ \frac{2}{\alpha^2}\left(\frac{\partial_t \partial_r b}{b} + \frac{\partial_r \Psi \partial_t \Psi}{2} - \frac{\partial_t a \partial_r b}{ab} - \frac{\partial_t b \partial_r \alpha}{\alpha b}\right) \\
&+ \frac{2}{r\alpha^2}\left(-\frac{\partial_t a}{a} + \frac{\partial_t b}{b}\right) \qquad (3.73)
\end{aligned}$$

and depends on all of the dynamical variables of the system. The observed convergence of the residual is only plausible if 1) our G-BSSN equations (3.14-3.18) have been correctly derived from the covariant Einstein equations, 2) we have discretized the geometric and matter equations properly, and 3) we have solved the full set of discretized equations correctly.

3.3.5 Finding Black Hole Threshold Solutions

The strength of the initial data can be set by adjusting the amplitude of the scalar field, p, in (3.42). For weak enough initial data (small enough p), the matter shell will reach the origin and then disperse, with the final state being a flat spacetime geometry. Sufficiently strong initial data, on the other hand (large enough p), results in a matter concentration in the vicinity of the origin which is sufficiently self-gravitating that a black hole forms. Using a binary search, we can find the threshold initial data, defined by $p = p^\star$, for which $p < p^\star$ results in dispersal while $p > p^\star$ yields black hole formation. At any stage of the calculation, the binary search is defined by two “bracketing” values, p_l and p_h, such that evolutions with $p = p_l$ and $p = p_h$ result in dispersal and black hole formation, respectively. It is convenient to define the amount of parameter tuning that has occurred by the dimensionless quantity

$$\delta p \equiv \frac{p_h - p_l}{p_l}\,. \qquad (3.74)$$

The dispersal case can be detected easily as the scalar field leaves the vicinity of the origin and the geometry approaches flat spacetime. To detect black hole formation, we use an apparent horizon finder to locate a surface $r = \text{const.}$ on which the divergence of the outgoing null rays vanishes. We first define the divergence function

$$\Theta = q^{\mu\nu} \nabla_\mu k_\nu \,, \tag{3.75}$$

where $q^{\mu\nu}$ is the induced metric on the constant r surface. In spherical symmetry with metric (3.69) we have

$$q_{\mu\nu} = \text{diag}\left(0, 0, r^2 b^2, r^2 b^2 \sin^2\theta\right) \,, \tag{3.76}$$

where k^μ is the null outgoing vector given by

$$k_\mu = \frac{1}{\sqrt{2}} \left[a\beta - \alpha, a, 0, 0\right] \,. \tag{3.77}$$

Therefore, (3.75) becomes

$$\Theta = \frac{\sqrt{2}}{rb} \left(\frac{r}{\alpha} \partial_t(b) + \left(\frac{1}{a} - \frac{\beta}{\alpha} \right) \partial_r(rb) \right) \,. \tag{3.78}$$

The formation of an apparent horizon[34] is signaled by the value of the function Θ crossing zero at some radius implying that the spacetime contains a black hole. We note that since the focus of our work was on the critical (threshold) solution we made no effort to continue evolutions beyond the detection of trapped surfaces.

3.4 Results

In this section we describe results from two sets of numerical experiments to study the efficacy of the G-BSSN formulation in the context of critical collapse. Again,

[34] Technically a marginally trapped surface—the apparent horizon being the outermost of these.

our calculations use the standard 1+log slicing condition for the lapse, and a shift which is either zero or determined from the Gamma-driver condition.

3.4.1 Zero Shift

We first perform a collection of numerical experiments where the shift vector is set to zero. As described in Sec. 3.3.5, in principle we can find the black hole threshold solution $p \simeq p^\star$ using a binary search algorithm which at any stage is defined by two values p_l and p_h, with $p_l < p < p_h$, and where p_l corresponds to dispersal (weak data) while p_h corresponds to black hole formation (strong data).

As discussed in the introduction, the massless scalar collapse model has a very well-known critical solution, and we summarize the features most relevant to our study here. The threshold configuration is discretely self-similar with an echoing exponent measured from the first calculations to be $\Delta \approx 3.44$ [4]. Following the original studies, Gundlach [67] showed that the construction of the precisely discretely self-similar spacetime could be posed as an eigenvalue problem, the solution of which led to the more accurate value $\Delta = 3.4439 \pm 0.0004$. This estimate was subsequently improved by Martin-Garcia and Gundlach to $\Delta = 3.445452402(3)$ [72].

The original calculations determined a value $\gamma \approx 0.37$ for the mass-scaling exponent [4]; further work based on perturbation theory gave $\gamma \approx 0.374$ [68, 71]. Here it is important to note that, as pointed out independently by Gundlach [68] and Hod and Piran [69], the simple power law scaling (3.1) gets modified for discretely self-similar critical solutions to

$$\ln M = \gamma \ln |p - p^\star| + c + f\left(\gamma \ln |p - p^\star| + c\right) , \tag{3.79}$$

where f is a universal function with period Δ and c is a constant depending on the initial data. This results in the superposition of a periodic "wiggle" in the otherwise linear scaling of $\ln M$ as a function of $\ln |p - p^\star|$.

Finally, Garfinkle and Duncan [70] pointed out that near-critical scaling is seen

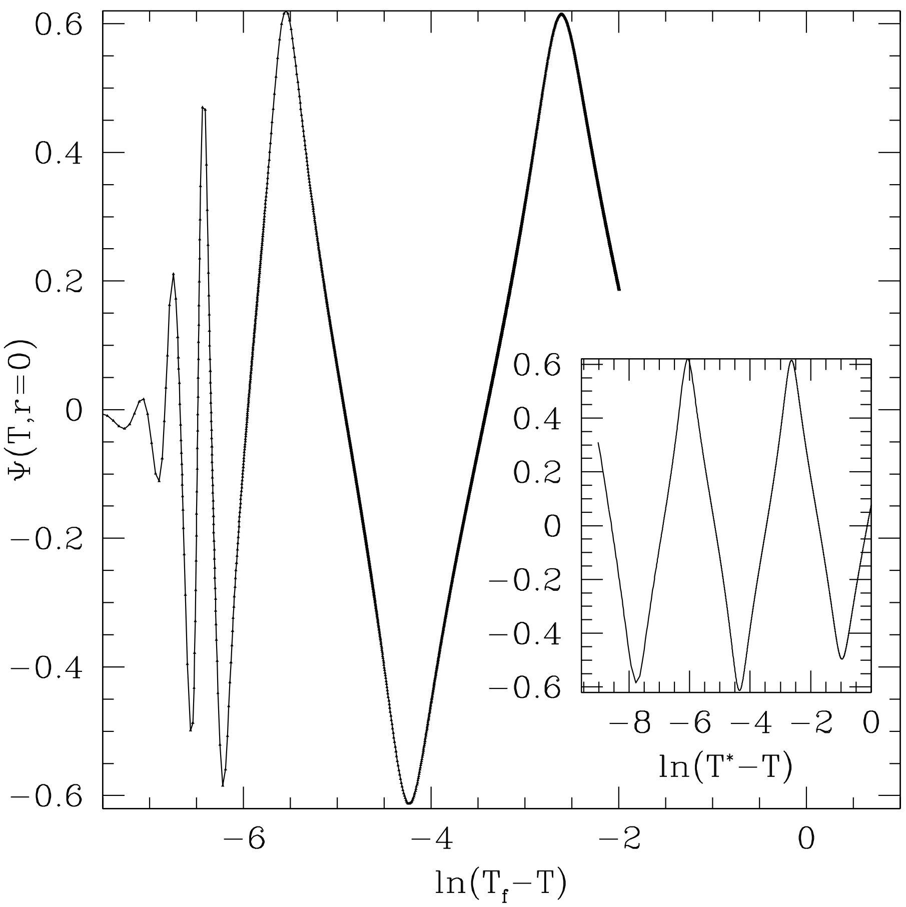


Figure 3.2: Echoing behaviour in the scalar field for a marginally subcritical evolution with $\delta p \approx 10^{-12}$. The main plot displays the central value of the scalar field versus a logarithmically scaled time parameter, $\ln(T_f - T)$, where T is central proper time and T_f is the approximate value of that time when near-critical evolution ceases and the total dispersal of the pulse to infinity begins. This particular scaling is chosen solely to more clearly demonstrate the evolution of the central value of Ψ during the critical phase through to dispersal. Note that our choice of abscissa means that evolution proceeds from right to left. The inset also plots Ψ at $r = 0$ but now in the "natural" logarithmic time coordinate $\tau \equiv \ln(T^\star - T)$ where $T^\star$ is the "accumulation time" at which the solution becomes singular and which has been estimated based on the positions of the extrema in Ψ. The amplitude of the scalar field at the origin oscillates between $(-0.61, 0.61)$, consistent with the calculations reported in [4]. The data yield an echoing exponent of $\Delta = 3.43 \pm 0.02$ which is in agreement with the value $\Delta = 3.445452402(3)$ Martin-Garcia and Gundlach have computed by treating the computation of the precisely-critical solution as an eigenvalue problem [72].

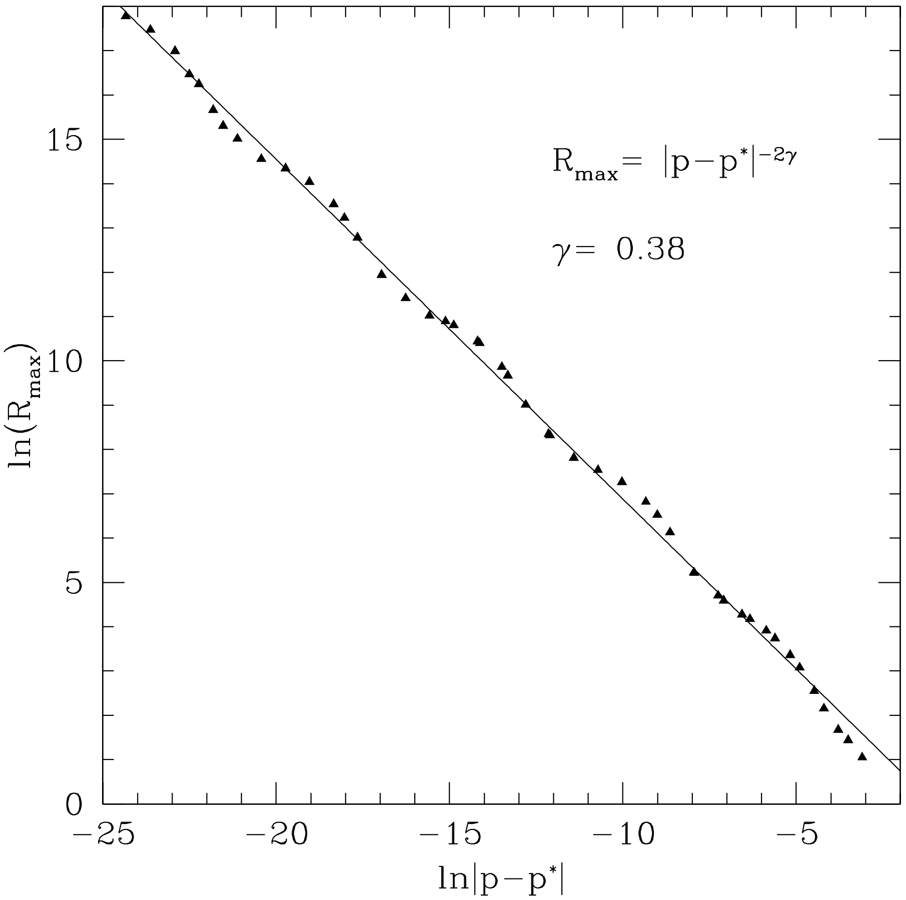


Figure 3.3: The maximum central value, $\mathcal{R}_{\rm max}$, of the four-dimensional Ricci scalar, $\mathcal{R}$, attained during subcritical evolution as a function of the logarithmic distance $\ln|p-p^\star|$ of the tuning parameter from the critical value. As first observed in [70] the Ricci scalar scales as $R \sim |p-p^\star|^{-2\gamma}$, where γ is the universal mass-scaling exponent in (3.1). The value $\gamma = 0.38 \pm 0.01$ computed via a least squares fit is in agreement with the original calculations [4] as well as many other subsequent computations. We note that the oscillations of the data about the linear fit are almost certainly genuine, at least in part. As discussed in the text, we *expect* a periodic wiggle in the data with period $\Delta/(2\gamma) \approx 4.61$. Performing a Fourier analysis of the residuals to the linear fit we find a peak at about 4 with a bandwidth of approximately 1, consistent with that expectation. As described in more detail in the text, although we have data from computations with $\ln|p-p^\star| < -25$, we do not include it in the fit. The naive method we use to estimate $p^\star$ means that the relative uncertainty in $p - p^\star$ grows substantially as $p \to p^\star$ so that inclusion of the data from the most nearly-critical calculations will corrupt the overall fit.

in physical quantities other than the mass and, dependent on the quantity, in the subcritical as well as supercritical regime. In particular they argued that in subcritical evolutions the maximum central value, $\mathcal{R}_{\rm max}$, of the four-dimensional Ricci scalar, $\mathcal{R}$, defined by

$$\mathcal{R}_{\rm max} \equiv \max_{t} \mathcal{R}(0,t)\,, \tag{3.80}$$

should satisfy the scaling

$$\mathcal{R}_{\rm max} \sim |p - p^{\star}|^{-2\gamma}\,, \tag{3.81}$$

where the factor -2 in the scaling exponent can be deduced from the fact that the curvature has units of length^{-2}. For the discretely self-similar case this scaling law is also modulated by a wiggle with period $\Delta/(2\gamma)$, which for the massless scalar field is about 4.61.

Using initial data given by (3.42) we tune p so that it is close to the critical value: typically this involves reducing the value of δp defined by (3.74) so that it is about 10^{-12}, which is a few orders of magnitude larger than machine precision. Our implementation includes code that actively monitors the dynamical variables for any indications of coordinate singularities or other pathologies which could cause the numerical solver to fail. Provided that such pathologies do not develop, we expect to observe features characteristic of critical collapse—discrete self-similarity and mass scaling in particular—to emerge as $p \to p^{\star}$.

One way the discrete self-similarity of the critical solution is manifested is as a sequence of "echoes"—oscillations of the scalar field near the origin such that after each oscillation the profile of the scalar field is repeated but on a scale $\exp(\Delta)$ smaller than that of the preceding echo (see Eq. (3.2)). The oscillations are similarly periodic in the logarithmic time scale $\ln(T^{\star} - T)$, where T is the proper time measured at the origin,

$$T(t) \equiv \int_0^t \alpha(\hat{t}, 0)\, d\hat{t}\,, \tag{3.82}$$

and $T^{\star}$ is the accumulation time at which the singularity forms (always at $r = 0$).

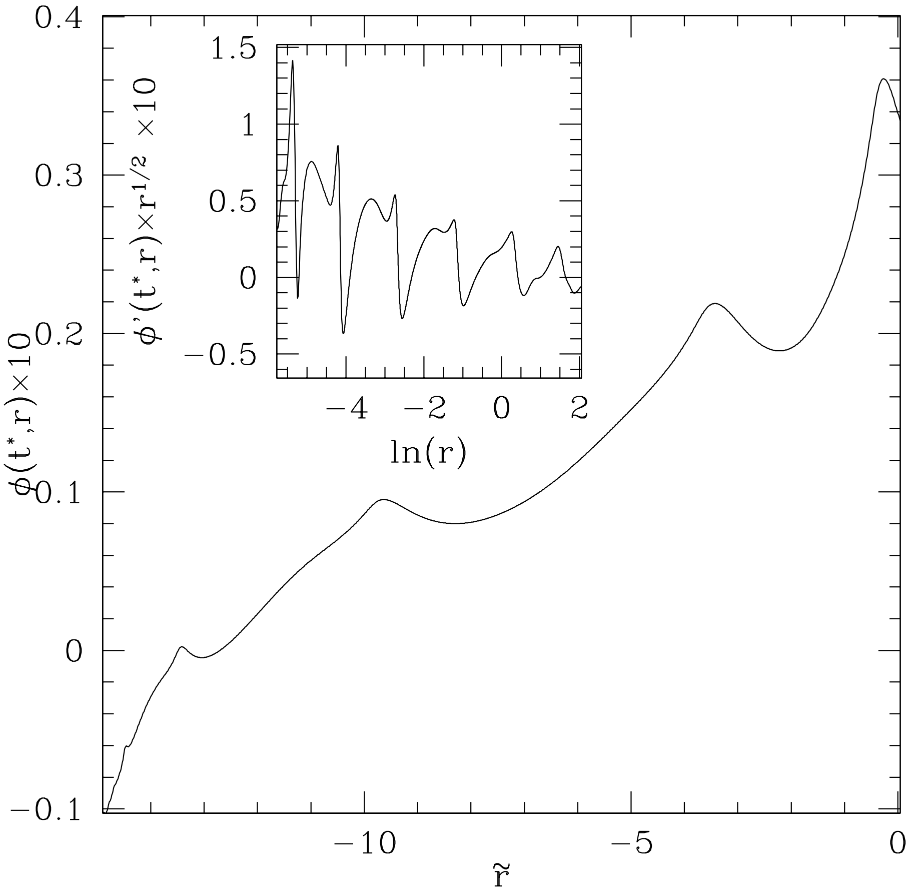


Figure 3.4: Discrete self-similarity of the geometry of spacetime in the black hole threshold evolution previously discussed in Fig. 3.2. Here, the G-BSSN variable ϕ is plotted as a function of the computational radial coordinate $\tilde{r}$ at the accumulation time $t^\star$. Note that from (3.83) ϕ measures the deviation of the determinant of the 3-metric from that of a flat metric. The inset graph is the radial derivative of ϕ scaled by $\sqrt{r}$ to highlight the formation of fine structure in the geometry of the critical solution. The approximate periodicity of $\sqrt{r}\phi'$ in $\ln(r)$ (modulo an overall varying scale) provides weak evidence that the coordinate system used in the calculation adapts to the self-similarity of the critical solution.

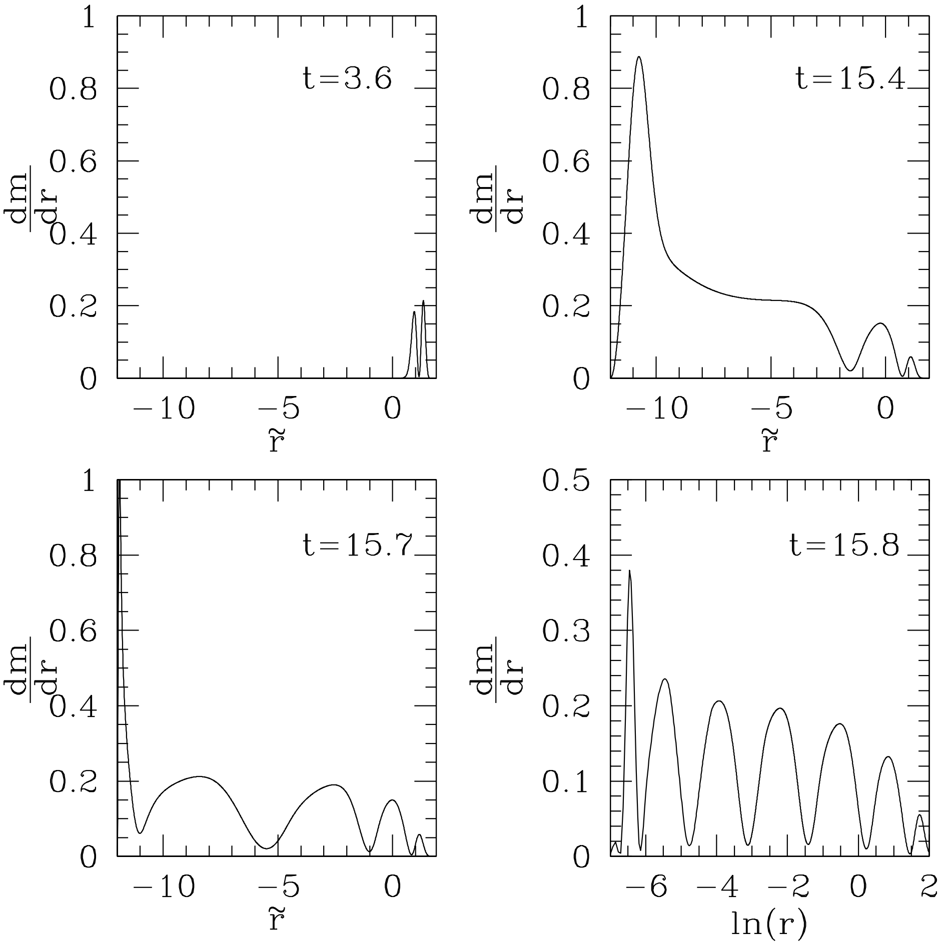


Figure 3.5: Snapshots of radial mass density for a marginally subcritical calculation ($\delta p \approx 10^{-12}$, $N_r = 2048$). Plotted is $dm/dr = r^2 \rho(t,r)$ where ρ is defined by (3.24). In this calculation $\beta = 0$ so we also have $dm/dr = T^t_{\ t}$. As the solution evolves, development of echos is clearly seen. In the final frame, which is at an instant $t = 15.8$ that is close to the accumulation time $t^\star$, we observe 4 echos. Note that we do not count the tall thin peak at the extreme left nor the first two peaks on the right as echos. The skinny peak will develop into an echo as p is tuned closer to $p^\star$. The two peaks on the right account for the bulk of the matter and represent the part of the initial pulse that implodes through the origin and then disperses "promptly", i.e. without participating in the strongly self-gravitating dynamics. A corresponding plot for an evolution far from criticality would contain *only* those two peaks. Note that the first three plots use the computational coordinate $\tilde{r}$ to provide a sense of the actual numerical calculation, while the last plot uses $\ln(r)$ in order to best highlight the discrete self-similarity of the threshold solution. As is the case for the data plotted in the inset of the previous figure, the approximate periodicity of the mass density in $\ln(r)$ suggests that the coordinates are adapting to the self-symmetry of the critical spacetime.

Furthermore, viewed at the origin, the oscillations of the scalar field occur at a fixed amplitude of about 0.61 (with our units and conventions for the Einstein's equations). As shown in Fig. 3.2, when we tune the initial data to the critical value, the central value of scalar field exhibits oscillatory behaviour and the amplitude is close to the expected value. The anticipated periodicity in logarithmic time is also apparent with a measured $\Delta = 3.43 \pm 0.02$, in agreement with previous results. We thus have strong evidence that the evolution has indeed approached the critical regime and that the measured oscillations are true echos rather than numerical artifacts.

Evidence that our code correctly captures the expected critical scaling behaviour (3.81) of $\mathcal{R}_{\max}$ is presented in Fig. 3.3. We find $\gamma = 0.38 \pm 0.01$, consistent with previous calculations. We note that we can measure scaling from our computations up to $\ln|p - p^\star| = -29$ (or $|p - p^\star| \approx 10^{-13}$). However, in Fig. 3.3 we have excluded the last few values closest to the critical point from both the plot and the linear fit: specifically, we truncate the fit at $\ln|p - p^\star| = -25$. The rationale for this is that we use the largest subcritical value of p as an approximation to the critical value $p^\star$ rather than, for example, implementing a multi-parameter fit that includes $p^\star$ as one of the parameters. Our estimate of $p^\star$ thus has an intrinsic error of $e^{-29} \approx 10^{-13}$ and by fitting to data with $\ln|p - p^\star| \geq -25$ we render the error in the $p^\star$ estimate essentially irrelevant. We note that consistent with the early observations of the robustness of mass scaling in the model [4], measuring the exponent γ can be achieved by moderate tuning, in this case $\ln|p - p^\star| \approx -9$, (i.e. $\delta p \approx 10^{-3}$). However, to be able to observe the echoing exponent (the oscillations around the fitting line, for example) we need to tune much closer to the critical value.

The echoing behaviour of the critical solution is also reflected in the geometry of spacetime and the matter variables other than the scalar field. Fig. 3.4 shows the radial profile of the G-BSSN variable ϕ at an instant close to the accumulation time $T^\star$. As seen in this plot, fine structure develops in the function in the near-critical regime. Observe that from the definition (3.7) and the choice (3.28), the scalar ϕ is the ratio of the determinant of the 3-metric, γ, to the determinant of the flat metric, $\mathring{\gamma}$:

$$\phi = \frac{1}{12} \ln(\gamma/\mathring{\gamma}) \, . \tag{3.83}$$

The radial matter density, $dm/dr = r^2 \rho$, is a convenient diagnostic quantity for viewing near-critical evolution. Snapshots of this function from a typical marginally subcritical calculation are shown in Fig. 3.5: the echoing behavior is clearly evident in the sequence. The number of echos is dependent on the degree to which the solution has been tuned to criticality. In this case, where $\delta p = 10^{-12}$, we expect and see about 4 echos (last frame of the figure). Here we note that each of the echos in dm/dr corresponds to half of one of the scalar field oscillations shown in Fig. 3.2 (where the inset shows about $2\frac{1}{2}$ full cycles).

Fig. 3.6(a) plots the central matter density $\rho(t, 0)$ for a marginally supercritical calculation. In accord with the self-similar nature of the near-critical solution, the central density grows exponentially with time. Fig. 3.6(b) is a snapshot of the extrinsic curvature at the critical time $t \approx t^\star$ while Fig. 3.6(c) shows the dynamics of the central value of the lapse function and compares it with α from the calculations performed with $\beta = \beta_G$ described in the next section. Fig. 3.6(d) displays the profile of the lapse at the critical time.

We note that we have not fully resolved the critical behaviour in the sense of having tuned p to the limit of machine precision, $\delta p \approx 10^{-16}$, which would capture roughly 2 additional echos in the threshold solution (one full echo in the scalar field). In principle, by setting N_r sufficiently large we could almost certainly do so since there are no indications that our method would break down at higher resolution and closer to criticality. However, we estimate that the required compute time for a complete critical search would increase from weeks to several months and we have thus not done this. Ultimately, a more effective approach to enhancing the resolution would be to incorporate a technique such as adaptive mesh refinement

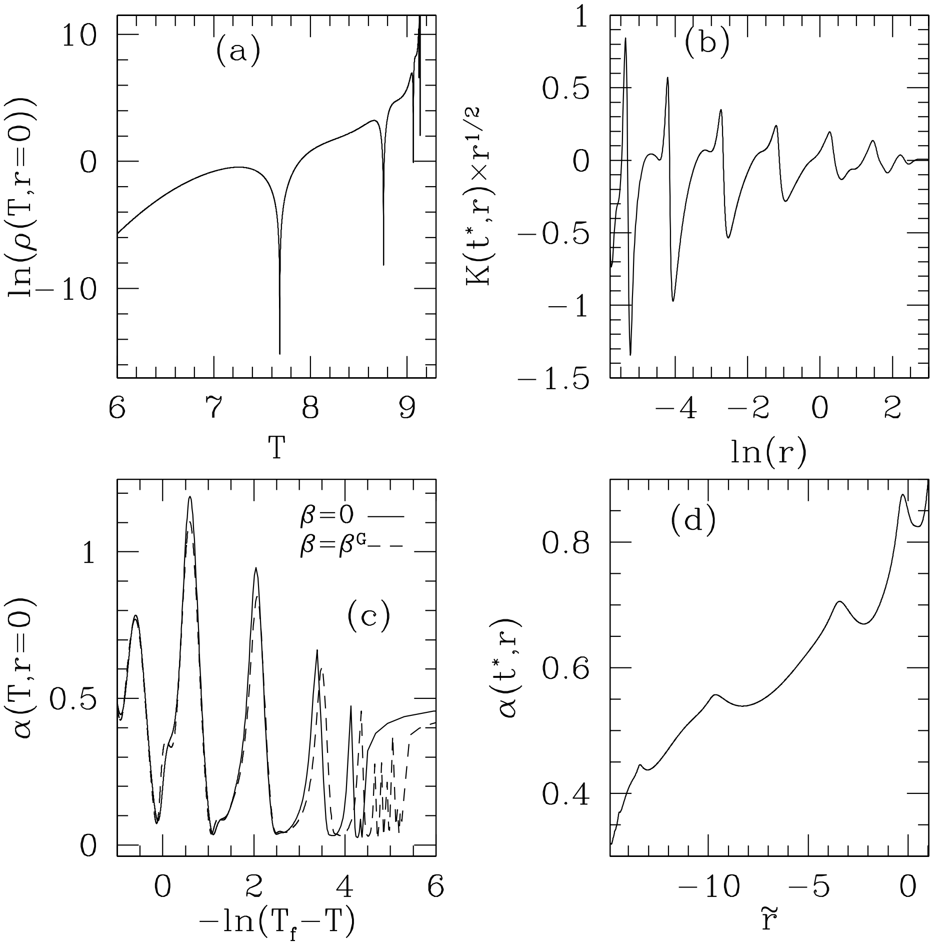


Figure 3.6: Profiles of matter and geometry variables from strongly gravitating, near-critical evolutions where the echoing behaviour emerges. Results were computed using 1+log slicing and zero shift, except for the dashed curve in (c) where $\beta = \beta^G$. (a) Central energy density, $\rho(T, 0)$, as a function of proper central time, T, and in logarithmic scale for a supercritical evolution. The density oscillates and grows exponentially as the system approaches the critical solution and then eventually collapses to form a black hole. (b) Profile of the extrinsic curvature, $K(t^\star, r)$—scaled by $r^{1/2}$ in order to make the echoing behaviour more visible—where $t^\star$ denotes a time very close to the accumulation time. The evolution is marginally subcritical in this case.

Figure 3.6: (c) Central value of the lapse function, α, during subcritical evolutions with $\beta = 0$ (solid) and $\beta = \beta_G$ (dashed). The plots use a logarithmically transformed proper time variable, $-\ln(T_f - T)$, where T_f is the approximate time at which the final dispersal of the pulse from the origin begins. In both cases α exhibits echoing and there is no evidence of pathological behaviour, such as the lapse collapsing or becoming negative. The close agreement of α for the two choices of β indicates that the time slicing varies little between the two coordinate systems. Note that there are three extra oscillations for the $\beta = \beta_G$ case, in the time interval $-\ln(T_f - T) \gtrsim 4.5$. These are spurious and due to a lack of finite-difference resolution; there are only 6 time steps in each oscillation. (d) Radial profile $\alpha(t^\star, r)$ at a time $t = t^\star$ which is close to the accumulation time and when the self-similarity and echoing in the spacetime geometry is apparent.

into our solver.

The results displayed in Figs. 3.2–3.6 provide strong evidence that the coordinate system consisting of 1+log lapse and zero shift remains non-singular in the critical regime, at least for the range of scales probed for $\delta p \approx 10^{-12}$. Additionally, the approximate periodicity in $\ln(r)$ that can be seen, for example, in $\sqrt{r}\phi'$ (Fig. 3.4) and dm/dr (Fig. 3.5) suggests that the coordinates may be adapting to the self-similarity. Whether or not this is actually the case is a matter requiring further study.

3.4.2 Gamma-driver Shift

We now briefly report on experiments similar to those of the previous section but where the shift was evolved with the Gamma-driver condition (3.38). A principal observation is that this gauge also facilitates near-critical evolutions with results similar to the $\beta = 0$ choice. In particular, we are again able to observe all of the characteristics of the black hole threshold solution.

The gauge condition (3.38) acts as a damping factor for the conformal connection, $\tilde{\Lambda}^i$, and we would therefore expect to observe a significant change in the profile of $\tilde{\Lambda}^i$ at threshold relative to the zero-shift case. This expectation is borne out by the comparison illustrated in Figs. 4.1 (a) and (b). When $\beta = 0$, $\tilde{\Lambda}^i$ diverges as

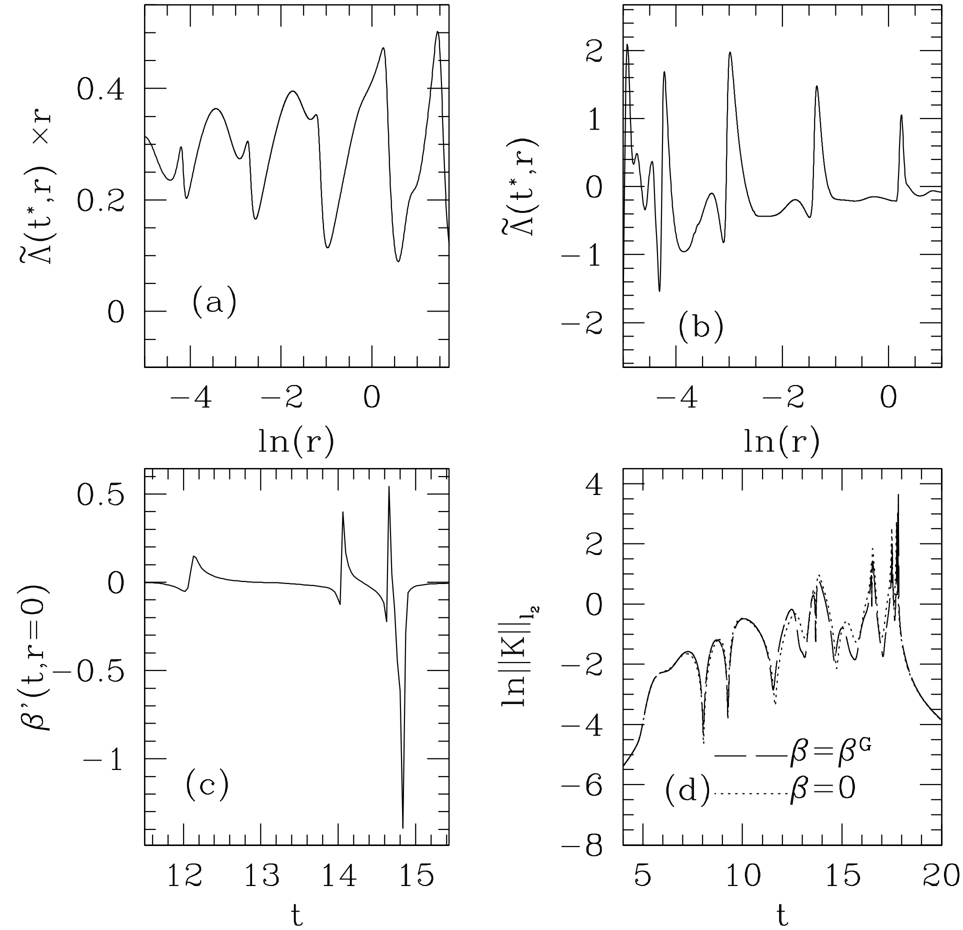


Figure 3.7: Profiles of various G-BSSN variables from marginally subcritical evolutions. (a) Profile of the conformal connection $\tilde{\Lambda}$ as computed with $\beta = 0$ and at a time $t^\star$ close to the accumulation time. Note that the function has been scaled by r and in fact diverges like $1/r$. (b) Profile of $\tilde{\Lambda}$ as computed with $\beta = \beta^G$, again at a moment close to the accumulation time. Here the $1/r$ growth seen when the condition $\beta = 0$ is adopted is absent. (c) Profile of the central spatial derivative of the shift vector, $\beta'(t, 0)$, as computed with $\beta = \beta^G$. As the echos develop closer to the origin, β' increases and presumably will diverge in the continuum, precisely-critical limit. (d) Time development of the l_2-norm of the extrinsic curvature during subcritical evolutions for both the $\beta = 0$ and $\beta = \beta^G$ calculations. In both cases the extrinsic curvature develops a divergent profile near $r = 0$ in the critical regime.

$1/r$ close to the origin while it appears to have finite amplitude for $\beta = \beta^G$. We find that the shift develops very sharp oscillations near the origin; some typical behaviour can be seen in the plot of $\beta'(t,0)$ shown in Fig. 4.1(c). We believe that these oscillations are genuine and our expectation is that $\beta'(t,0)$ will diverge in the precise critical limit. Further, we observe that the oscillations can create numerical artifacts and generally require higher resolution relative to the $\beta = 0$ case, as well as dissipation, to be controlled. Indeed, when using the Gamma-driver condition we find that Kreiss/Oliger dissipation is crucial to suppress unresolved high frequency oscillations close to the origin. Fig. 4.1 (d) shows the growth in the norm of the extrinsic curvature during a subcritical evolution. The norm of K does not exhibit any significant difference for the two choices of the shift.

As was the case for the $\beta = 0$ calculations, the results shown in Figs. 3.6 and 3.7 strongly suggest that the combination of 1+log slicing and Gamma driver shift provides a coordinate system which is adequate for computing the near-critical solution. In addition, the approximate periodicity seen in Figs. 3.6(b), 3.6(c), 3.7(a) and 3.7(b) suggest that this gauge may also be adapting to the self-symmetry.

3.5 Conclusion

We have described a numerical code that implements a generalized BSSN formulation adapted to spherical symmetry. Using standard dynamical coordinate choices, including 1+log slicing and a shift which either vanished or satisfied a Gamma-driver condition, we focused specifically on the applicability of the formulation and the gauge choices to studies of type II critical phenomena. As a test of the approach we revisited the model of massless scalar collapse, where the properties of the critical solution are very well known from previous work. For both choices of the shift, we found that our code was able to generate evolutions that were very close to criticality so that, in particular, we could observe the expected discrete self-similarity of the critical solution. To our knowledge, this is the first fully evolutionary implementation of a hyperbolic formulation of Einstein's equations that has been able to unequivocally resolve discrete self-similarity in type II collapse. Furthermore, measured properties from near-critical solutions, including the mass-scaling and echoing exponents, are in agreement with previous work. Our results strongly suggest that the G-BSSN formulation, in conjunction with standard dynamical coordinate conditions, is capable of evolving the spacetime near criticality without the development of coordinate pathologies. There is also some evidence that both gauges adapt to the self-similarity, but we have not yet studied this issue in any detail.

We found that certain of the primary G-BSSN variables diverge as the critical solution is approached: this is only to be expected since the precisely critical solution contains a naked singularity. Dealing with such solution features in a stable and accurate manner presents a challenge for any code and in our case we found that a combination of a non-uniform grid and Kreiss/Oliger dissipation was crucial. Our use of a time-implicit evolution scheme may have also been important although we did not experiment with that aspect of our implementation. However, we suspect that the implicit time-stepping helped maintain regularity of the solutions near $r = 0$, as other researchers have found.

Given the success of the G-BSSN approach, it is natural to consider its generalization and application to settings with less symmetry, but where curvilinear coordinates are still adopted. In particular, one axisymmetric problem that has yet to be resolved is the collapse of pure gravity waves. This scenario arguably provides the most fundamental critical phenomena in gravity as the behaviour must be intrinsic to the Einstein equations, rather than being dependent on some matter source. Critical collapse of gravitational waves–with mass scaling and echoing—was observed by Abrahams and Evans over 20 years ago [76]. However, their original results have proven very difficult to reproduce (or refute) [77–80, 83]. We refer the reader to the recent paper by Hilditch et al. [83] for detailed discussions concerning some apparent inconsistencies among the follow-up studies, as well as the challenges and complications involved in evolving various types of nonlinear gravitational waves. We are currently extending the methodology described above to the axisymmetric case with plans to use the resulting code to study vacuum collapse. Results from this undertaking will be reported in a future paper.

3.6 BSSN in Spherical Symmetry

In this appendix, we provide the explicit expressions of the G-BSSN evolution equations in spherical symmetry.

The evolution equations (3.14-3.15) for ϕ and the components of the conformal metric $\tilde{\gamma}_{ij}$ simplify to

$$\partial_t \phi = \frac{1}{6}\alpha K + \beta \partial_r \phi + \sigma \frac{1}{6}\mathscr{B}\,, \tag{3.84}$$

$$\partial_t \tilde{\gamma}_{rr} = -2\alpha \tilde{A}_{rr} + \beta \partial_r \tilde{\gamma}_{rr} + 2\tilde{\gamma}_{rr}\partial_r \beta - \sigma \frac{2}{3}\tilde{\gamma}_{rr}\mathscr{B}\,, \tag{3.85}$$

$$\partial_t \tilde{\gamma}_{\theta\theta} = -2\alpha \tilde{A}_{\theta\theta} + \beta \partial_r \tilde{\gamma}_{\theta\theta} + 2\frac{\beta}{r}\tilde{\gamma}_{\theta\theta} - \sigma \frac{2}{3}\tilde{\gamma}_{\theta\theta}\mathscr{B}\,, \tag{3.86}$$

where $\mathscr{B}$ is the divergence of the shift vector,

$$\mathscr{B}(t,r) = D_i \beta^i = \partial_r \beta + \frac{2\beta}{r} + \beta\left(\frac{\partial_r \tilde{\gamma}_{rr}}{2\tilde{\gamma}_{rr}} + \frac{\partial_r \tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{\theta\theta}}\right)\,. \tag{3.87}$$

To display the equation of motion for the trace of the extrinsic curvature K and $\tilde{A}_{ij}$ we first define

$$\mathscr{D}_{ij} \equiv D_i D_j \alpha\,, \tag{3.88}$$

which has 2 independent components,

$$\mathscr{D}_{rr} = \partial_r^2 \alpha - \partial_r \alpha \left(\frac{\partial_r \tilde{\gamma}_{rr}}{\tilde{\gamma}_{rr}} + 4\partial_r \phi\right)\,, \tag{3.89}$$

$$\mathscr{D}_{\theta\theta} = r\partial_r\alpha\frac{\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{rr}} + \frac{r^2}{2}\partial_r\alpha\left(\frac{\partial_r\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{rr}} + 4\partial_r\phi\frac{\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{rr}}\right) . \tag{3.90}$$

The trace of $\mathscr{D}_{ij}$ is

$$\mathscr{D} \equiv \gamma^{ij}\mathscr{D}_{ij} = e^{-4\phi}\left(\frac{\mathscr{D}_{rr}}{\tilde{\gamma}_{rr}} + 2\frac{\mathscr{D}_{\theta\theta}}{r^2\tilde{\gamma}_{\theta\theta}}\right) . \tag{3.91}$$

Then the evolution of K is given by

$$\begin{aligned}\partial_t K &= -\mathscr{D} + \alpha\left(\frac{1}{3}K^2 + \frac{\tilde{A}^2_{rr}}{\tilde{\gamma}^2_{rr}} + 2\frac{\tilde{A}^2_{\theta\theta}}{\tilde{\gamma}^2_{\theta\theta}}\right)\\ &+ \beta\partial_r K + 4\pi\alpha(\rho + S)\end{aligned} \tag{3.92}$$

and the evolution equations for the traceless part of the extrinsic curvature are

$$\begin{aligned}\partial_t\tilde{A}_{rr} &= e^{-4\phi}\left[-\mathscr{D}^{\mathrm{T}F}_{rr} + \alpha\left(R^{\mathrm{T}F}_{rr} + 8\pi S^{\mathrm{T}F}_{rr}\right)\right]\\ &+ \alpha\left(\tilde{A}_{rr}K - \frac{2\tilde{A}^2_{rr}}{\tilde{\gamma}_{rr}}\right)\\ &+ 2\tilde{A}_{rr}\partial_r\beta + \beta\partial_r\tilde{A}_{rr} - \sigma\frac{2}{3}\mathscr{B}\tilde{A}_{rr} ,\end{aligned} \tag{3.93}$$

$$\begin{aligned}\partial_t\tilde{A}_{\theta\theta} &= \frac{e^{-4\phi}}{r^2}\left[-\mathscr{D}^{\mathrm{T}F}_{\theta\theta} + \alpha\left(R^{\mathrm{T}F}_{\theta\theta} + 8\pi S^{\mathrm{T}F}_{\theta\theta}\right)\right]\\ &+ \alpha\left(\tilde{A}_{\theta\theta}K - 2\frac{\tilde{A}^2_{\theta\theta}}{\tilde{\gamma}_{\theta\theta}}\right)\\ &+ 2\frac{\beta}{r}\tilde{A}_{\theta\theta} + \beta\partial_r\tilde{A}_{\theta\theta} - \sigma\frac{2}{3}\tilde{A}_{\theta\theta}\mathscr{B} ,\end{aligned} \tag{3.94}$$

where R denotes the 3-Ricci tensor with non-vanishing components

$$\begin{aligned}
R_{rr} \;=\;& \frac{3(\partial_r\tilde{\gamma}_{rr})^2}{4\tilde{\gamma}_{rr}^2} - \frac{(\partial_r\tilde{\gamma}_{\theta\theta})^2}{2\tilde{\gamma}_{\theta\theta}^2} + \tilde{\gamma}_{rr}\partial_r\tilde{\Lambda} + \frac{1}{2}\partial_r\tilde{\gamma}_{rr}\tilde{\Lambda} \\
+\;& \frac{1}{r}\left(4\partial_r\phi - \frac{\partial_r\tilde{\gamma}_{rr} - 2\partial_r\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{\theta\theta}} - \frac{2\tilde{\gamma}_{rr}\partial_r\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{\theta\theta}^2}\right) \\
-\;& 4\partial_r^2\phi + 2\partial_r\phi\left(\frac{\partial_r\tilde{\gamma}_{rr}}{\tilde{\gamma}_{rr}} - \frac{\partial_r\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{\theta\theta}}\right) \\
-\;& \frac{\partial_r^2\tilde{\gamma}_{rr}}{2\tilde{\gamma}_{rr}} + \frac{2(\tilde{\gamma}_{rr} - \tilde{\gamma}_{\theta\theta})}{r^2\tilde{\gamma}_{\theta\theta}}\,,
\end{aligned} \tag{3.95}$$

$$\begin{aligned}
R_{\theta\theta} \;=\;& \frac{r^2\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{rr}}\left(\partial_r\phi\frac{\partial_r\tilde{\gamma}_{rr}}{\tilde{\gamma}_{rr}} - 2\partial_r^2\phi - 4(\partial_r\phi)^2\right) \\
+\;& \frac{r^2}{\tilde{\gamma}_{rr}}\left(\frac{(\partial_r\tilde{\gamma}_{\theta\theta})^2}{2\tilde{\gamma}_{\theta\theta}} - 3\partial_r\phi\partial_r\tilde{\gamma}_{\theta\theta} - \frac{1}{2}\partial_r^2\tilde{\gamma}_{\theta\theta}\right) \\
+\;& r\left(\Lambda\tilde{\gamma}_{\theta\theta} - \frac{\partial_r\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{\theta\theta}} - \frac{6\partial_r\phi\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{rr}}\right) \\
+\;& \frac{\tilde{\gamma}_{\theta\theta}}{\tilde{\gamma}_{rr}} - 1\,.
\end{aligned} \tag{3.96}$$

In the above expressions the superscript TF denotes application of the trace-free-part operator, whose action can be written explicitly as

$$X_{rr}^{\mathrm{T}F} = X_{rr} - \frac{1}{3}\gamma_{rr}X = \frac{2}{3}\left(X_{rr} - \frac{AX_{\theta\theta}}{Br^2}\right)\,, \tag{3.97}$$

$$X_{\theta\theta}^{\mathrm{T}F} = X_{\theta\theta} - \frac{1}{3}\gamma_{\theta\theta}X = \frac{1}{3}\left(X_{\theta\theta} - \frac{BX_{rr}}{A}\right)\,. \tag{3.98}$$

Here X represents any of the tensors $\mathscr{D}$, R or S.

Finally, the evolution of $\tilde{\Lambda}^i$ reduces to

$$\begin{aligned}
\partial_t\tilde{\Lambda} &= \beta\partial_r\tilde{\Lambda} - \partial_r\beta\tilde{\Lambda} + \frac{2\alpha}{\tilde{\gamma}_{rr}}\left(\frac{6\tilde{A}_{\theta\theta}\partial_r\phi}{\tilde{\gamma}_{rr}} - 8\pi S_r - \frac{2}{3}\partial_r K\right) \\
&+ \frac{\alpha}{\tilde{\gamma}_{rr}}\left(\frac{\partial_r\tilde{\gamma}_{rr}\tilde{A}_{rr}}{\tilde{\gamma}_{rr}^2} - \frac{2\partial_r\tilde{\gamma}_{\theta\theta}\tilde{A}_{\theta\theta}}{\tilde{\gamma}_{\theta\theta}^2} + 4\tilde{A}_{\theta\theta}\frac{\tilde{\gamma}_{rr} - \tilde{\gamma}_{\theta\theta}}{r\tilde{\gamma}_{\theta\theta}^2}\right) \\
&+ \sigma\left(\frac{2}{3}\tilde{\Lambda}\mathscr{B} + \frac{\partial_r\mathscr{B}}{3\tilde{\gamma}_{rr}}\right) + \frac{2}{r\tilde{\gamma}_{\theta\theta}}\left(\partial_r\beta - \frac{\beta}{r}\right) \\
&- 2\frac{\partial_r\alpha\tilde{A}_{rr}}{\tilde{\gamma}_{rr}^2}\,. \qquad (3.99)
\end{aligned}$$

3.7 Scalar Field Synamics and Energy-Momentum Tensor in Spherical Symmetry

Here we present the evolution equations of a complex scalar field, with an arbitrary potential V, minimally coupled to gravity. The governing equations for a massless real scalar field follow as a special case where the potential and the imaginary part of the field are both set to zero.

The geometry of spacetime is given by a generic metric in spherical symmetry:

$$ds^2 = (-\alpha^2 + \beta^2 a^2)dt^2 + 2a^2\beta dtdr + a^2dr^2 + r^2b^2d\Omega^2, \qquad (3.100)$$

where a, b, α and β are all functions of t and r and where a and b are related to the primary BSSN variables via $a = \tilde{\gamma}_{rr}\exp(4\phi)$ and $b = \tilde{\gamma}_{\theta\theta}\exp(4\phi)$.

The complex scalar field is given in terms of real and imaginary parts, Ψ_R and Ψ_I, respectively,

$$\Psi = \Psi_R(t, r) + i\Psi_I(t, r)\,, \qquad (3.101)$$

and has an associated energy-momentum tensor

$$\begin{aligned} T_{\mu\nu} &= \nabla_\mu \nabla_\nu \Psi_R - \frac{1}{2} g_{\mu\nu} \nabla^\eta \Psi_R \nabla_\eta \Psi_R \\ &+ \nabla_\mu \nabla_\nu \Psi_I - \frac{1}{2} g_{\mu\nu} \nabla^\eta \Psi_I \nabla_\eta \Psi_I \\ &- \frac{1}{2} g_{\mu\nu} V(|\Psi|) \, . \end{aligned} \tag{3.102}$$

The evolution of the real part of the scalar field can be reduced to a pair of first-order-in-time equations via the definition

$$\Xi_R \equiv \frac{b^2 a}{\alpha} \left(\partial_t \Psi_R - \beta \partial_r \Psi_R \right) \, . \tag{3.103}$$

We then find the following evolution equations for Ψ_R and Ξ_R:

$$\partial_t \Psi_R = \frac{\alpha}{b^2 a} \Xi_R + \beta \partial_r \Psi_R \, , \tag{3.104}$$

$$\begin{aligned} \partial_t \Xi_R &= \frac{\alpha b^2}{a} \left(\partial_r^2 \Psi_R + 2 \frac{\partial_r \Psi_R}{r} \right) + \partial_r \Psi_R \partial_r \left(\frac{\alpha b^2}{a} \right) \\ &+ \beta \partial_r \Xi_R + \Xi_R \partial_r \beta + \Xi_R \frac{2\beta}{r} \\ &- a \alpha b^2 \partial^2_{|\Psi|} V(|\Psi|) \, . \end{aligned} \tag{3.105}$$

The evolution equations for Ψ_I and Ξ_I follow from the index substitutions $R \leftrightarrow I$ in the right hand sides of (3.104) and (3.105), respectively.

The matter source terms in the G-BSSN equations, namely ρ, S, S^i, S_{ij}, can be simplified by defining Π and Φ as

$$\Pi \equiv \frac{a}{\alpha} \left(\partial_t \Psi - \beta \partial_r \Psi \right) \equiv \Pi_R(t, r) + i \Pi_I(t, r) \, , \tag{3.106}$$

$$\Pi_R = \frac{a}{\alpha} \left(\partial_t \Psi_R - \beta \partial_r \Psi_R \right) = \frac{\Xi_R}{b^2} \, , \tag{3.107}$$

$$\Pi_I = \frac{a}{\alpha}\left(\partial_t \Psi_I - \beta \partial_r \Psi_I\right) = \frac{\Xi_I}{b^2}\,, \tag{3.108}$$

$$\Phi \equiv \partial_r \Psi \equiv \Phi_R(t,r) + i\Phi_I(t,r)\,, \tag{3.109}$$

$$\Phi_R = \partial_r \Psi_R\,, \tag{3.110}$$

$$\Phi_I = \partial_r \Psi_I\,. \tag{3.111}$$

Using these definitions, the variables ρ and S are given by

$$\rho(t,r) = \frac{|\Pi|^2 + |\Phi|^2}{2a^2} + \frac{V(|\Psi|)}{2}\,, \tag{3.112}$$

$$S(t,r) = \frac{3|\Pi|^2 - |\Phi|^2}{2a^2} - \frac{3}{2}V(|\Psi|)\,. \tag{3.113}$$

In spherical symmetry, S^i has only a radial component,

$$S^i = \left[S^r(t,r), 0, 0\right]\,, \tag{3.114}$$

with

$$S^r = -\frac{\Pi_R\Phi_R + \Pi_I\Phi_I}{a}\,. \tag{3.115}$$

Similarly, the spatial stress tensor, S_{ij}, has only two independent components,

$$S_{ij} = \begin{pmatrix} S_{rr}(t,r) & 0 & 0 \\ 0 & r^2 S_{\theta\theta} & 0 \\ 0 & 0 & r^2\sin^2\theta S_{\theta\theta} \end{pmatrix}\,, \tag{3.116}$$

with

$$S_{rr} = \frac{|\Pi|^2 + |\Phi|^2}{2} - a^2 \frac{V(|\Psi|)}{2} \, , \tag{3.117}$$

$$S_{\theta\theta} = b^2 \left(\frac{|\Pi|^2 - |\Phi|^2}{2a^2} - \frac{V(|\Psi|)}{2} \right) \, . \tag{3.118}$$

Chapter 4

Non-linear Gravity Wave Evolutions with the G-BSSN Formulation

4.1 Introduction

As first discussed in the introduction, the General Relativistic theory of gravity has a radiative component to it, whereby the waves in the metric of pure vacuum carry information and energy. In Cartesian coordinates the linearized Einstein's equation (1.13), i.e. perturbation near the flat spacetime, has a simple planar wave solution (choosing z to be the direction of propagation):

$$\bar{h}_{ij} = H_{ij} e^{i(kz \pm \omega t)} \,, \tag{4.1}$$

where H_{ij} can be written as a linear combination of two basis tensors, e^{+} and $e^{\times}$:

$$H_{ij} = a e^{+}_{ij} + b e^{\times}_{ij} \,. \tag{4.2}$$

Here e^{+} is a tensor with only nonzero components: $e^{+}_{xx} = -e^{+}_{yy} = 1$ and $e^{\times}$ is a tensors for which the nonzero components are: $e^{\times}_{xy} = e^{\times}_{yx} = 1$.

The solutions to the linearized tensorial wave equation in spherical-polar coordinates with axial symmetry are known as Teukolsky waves [96]. Another vacuum axisymmetric ansatz to Einstein's equations was proposed by Brill [97]—which can be considered both in linear and non-linear regime. Both of these solutions are commonly used in gravitational waves evolutions, and we will introduce them in our discussion of the non-linear regime in Sec. 4.2.4.

The very first study of pure vacuum solutions of Einstein's equation using Brill data is due to Eppley [98] who demonstrated that a sufficiently strong pure vacuum Brill configuration contains a black hole. The first dynamical study of pure vacuum using a numerical approach was [99] and also adopted the Brill initial data (in axial symmetry). This study was the first simulation where pure vacuum dynamically collapses to a black hole. The first axisymmetric numerical evolutions of Teukolsky waves was due to Abrahams and Evans [100] who showed the formation of a black hole from an imploding Teukolsky wave packet.

The first full 3-dimensional simulations of gravity waves was by Shibata and Nakamura [31], who were the first to propose the rescaling of the BSSN formulation and perform successful long term evolutions of small amplitude waves. Follow up work by Baumgarte and Shapiro [32] completed the BSSN formulation by introducing the conformal connection functions and showed the much improved performance of BSSN formulation in comparison to the free ADM evolutions of small amplitude gravitational waves.

The 3D near-linear regime was first explored in [101, 102] with reports of numerical difficulties in achieving long-time dynamics. Strong gravity dynamics of 3D pure vacuum Brill data was first performed in [103] and [104] with the primary goal of finding black hole critical solutions. The evolution of Brill initial data in axial symmetry was revisited later in [105], again in search of critical behaviour. However as we discuss in the following, the critical phenomena in gravitational waves collapse is as yet an unsolved problem.

Critical Phenomena in Pure Gravity Waves Collapse

The follow up work to [100] by Abrahams and Evans [106] is the first report on the observation of type II critical behaviour in the collapse of Teukolsky waves. In addition, in [107], they found evidence for universality of the solution with a mass-scaling exponent $\gamma \approx 0.38$, surprisingly close to Choptuik's finding for the massless scalar field, and an echoing exponent $\Delta \approx 0.5$. However, there are 5 studies [77–80, 83] which further investigate the universality of the solution, all of which report unsuccessful attempts to reproduce the original results. Among these, Sorkin's work [80] stands out. He finds evidence for scaling in agreement with $\gamma \approx 0.38$ and hints of a DSS structure. However, the measured echoing exponent differs from [106] ($\Delta \approx 1.1$) for his choice of Brill initial data. In addition, Sorkin reports observation of a "ring of singularity" forming in the near-critical regime, which is somewhat unexpected and peculiar. These inconsistencies motivate a revisit of the problem, with a new axisymmetric code which implements the G-BSSN formulation and that appeared to be promising based on the spherical case we studied in the previous chapter.

In the rest of this chapter we discuss the development of a G-BSSN axisymmetric code and its application to non-linear gravity waves dynamics. The numerical techniques and the implementation of a new G-BSSN-based code in cylindrical coordinate is presented in Sec. 4.3 as is the initialization process using two types of Brill and Teukolsky-type initial data. In Sec. 4.4 we present the primary calculations performed using the code to evolve the pure vacuum in the strong gravity regime. Discussions of the future steps that are required to optimize and bring the code to production for type II critical phenomena studies are given in 4.5.

4.2 Equations of Motion for Strong Gravity Waves Dynamics

In this section, we summarize the G-BSSN equations and present their form in axial symmetry using cylindrical coordinates. As mentioned in the previous chapter, G-

BSSN is a generalization of the BSSN formulation to curvilinear coordinates where the flat 3-metric is not the unity matrix (see the discussion in Sec. 3.2.1). For example, in cylindrical coordinates, the flat 3-metric is given by:

$$ds^2 = d\rho^2 + dz^2 + \rho^2 d\varphi^2 \,, \tag{4.3}$$

which we denote as $\mathring{\gamma}_{ij}$ and has determinant $\mathring{\gamma} = \rho^2$. Here, we will not derive the G-BSSN equations in any detail, but rather refer the reader to [88] for a full derivation (also see the discussion in the previous chapter). In summary, then, the G-BSSN equations in vacuum are given by:

$$\partial_t \phi = -\frac{1}{6}\alpha K + \beta^i \partial_i \phi + \sigma \frac{1}{6} \tilde{D}_k \beta^k \,, \tag{4.4}$$

$$\partial_t \tilde{\gamma}_{ij} = -2\alpha \tilde{A}_{ij} + \beta^k \partial_k \tilde{\gamma}_{ij} + \tilde{\gamma}_{ik} \partial_j \beta^k + \tilde{\gamma}_{kj} \partial_i \beta^k - \sigma \frac{2}{3} \tilde{A}_{ij} \tilde{D}_k \beta^k \,, \tag{4.5}$$

$$\partial_t K = -\gamma^{ij} D_j D_i \alpha + \alpha(\tilde{A}_{ij} \tilde{A}^{ij} + \frac{1}{3} K^2) + \beta^i \partial_i K \,, \tag{4.6}$$

$$\begin{aligned} \partial_t \tilde{A}_{ij} &= e^{-4\phi} \left(-D_i D_j \alpha + \alpha R_{ij} \right)^{\mathrm{TF}} + \alpha (K \tilde{A}_{ij} - 2 \tilde{A}_{il} \tilde{A}^l{}_j) \\ &+ \beta^k \partial_k \tilde{A}_{ij} + \tilde{A}_{ik} \partial_j \beta^k + \tilde{A}_{kj} \partial_i \beta^k - \sigma \frac{2}{3} \tilde{A}_{ij} \tilde{D}_k \beta^k \,, \end{aligned} \tag{4.7}$$

$$\partial_t \tilde{\Lambda}^i = \partial_t \tilde{\Gamma}^i - \mathring{\Gamma}^i_{jk} \partial_t \tilde{\gamma}^{jk} \,, \tag{4.8}$$

in which we have set the matter sources to zero and written out the Lie derivatives explicitly. Here $\mathring{\Gamma}^i_{jk}$ denotes the Christoffel symbols associated with flat metric $\mathring{\gamma}_{ij}$. As in the previous chapter, σ is a parameter that determines the two standard choices of G-BSSN for the evolution of the determinant of the 3-conformal-metric,

$\tilde{\gamma}$:

$$\sigma = 1 \quad \Rightarrow \quad \partial_t \tilde{\gamma} = 0\,, \quad \text{(Lagrangian choice)}\,, \tag{4.9}$$

$$\sigma = 0 \quad \Rightarrow \quad (\partial_t - \mathscr{L}_{\vec{\beta}})\tilde{\gamma} = 0\,, \quad \text{(Lorentzian choice)}\,. \tag{4.10}$$

All of the simulations in this chapter use the Lagrangian option.

In (4.8) the first term, $\partial_t \tilde{\Gamma}^i$, is given by:

$$\begin{aligned}\partial_t \tilde{\Gamma}^i &= -2\tilde{A}^{ij}\partial_j \alpha + 2\alpha\left(\tilde{\Gamma}^i_{jk}\tilde{A}^{kj} - \frac{2}{3}\tilde{\gamma}^{ij}\partial_j K + 6\tilde{A}^{ij}\partial_j \phi\right) + \tilde{\gamma}^{lj}\partial_j \partial_l \beta^i \\ &+ \beta^j \partial_j \tilde{\Gamma}^i - \tilde{\Gamma}^j \partial_j \beta^i + \frac{\sigma}{3}\left[2\tilde{\Gamma}^i \tilde{D}_k \beta^k + \tilde{\gamma}^{li}\partial_l(\tilde{D}_k \beta^k)\right]\,, \end{aligned} \tag{4.11}$$

and in the second term, $\mathring{\Gamma}^i_{jk}\partial_t \tilde{\gamma}^{jk}$, the time evolution of the inverse of the conformal metric, $\partial_t \tilde{\gamma}^{jk}$, can be evaluated using the time evolution of $\tilde{\gamma}_{ij}$, (4.5) by the consideration:

$$\begin{aligned} \tilde{\gamma}^{ik}\tilde{\gamma}_{kj} = \delta^i_j \Rightarrow \partial_t(\tilde{\gamma}^{ik}\tilde{\gamma}_{kj}) &= 0 \\ \Rightarrow (\partial_t \tilde{\gamma}^{ik})\tilde{\gamma}_{kj} + \tilde{\gamma}^{ik}\partial_t \tilde{\gamma}_{kj} &= 0 \\ \Rightarrow (\partial_t \tilde{\gamma}^{ik})\tilde{\gamma}_{kj}\tilde{\gamma}^{jl} + \tilde{\gamma}^{ik}(\partial_t \tilde{\gamma}_{kj})\tilde{\gamma}^{jl} &= 0 \\ \Rightarrow \partial_t(\tilde{\gamma}^{ik})\delta^k_l + \tilde{\gamma}^{ik}(\partial_t \tilde{\gamma}_{kj})\tilde{\gamma}^{jl} &= 0 \\ \Rightarrow \partial_t(\tilde{\gamma}^{il}) = -\tilde{\gamma}^{ik}(\partial_t \tilde{\gamma}_{kj})\tilde{\gamma}^{jl}\,, \end{aligned} \tag{4.12}$$

in which $\partial_t \tilde{\gamma}_{kj}$ should be replaced by the right hand side (RHS) of (4.5). In addition, in (4.11), $\tilde{\Gamma}^i$ is substituted by the re-defined conformal connection, $\tilde{\Lambda}^i$, in G-BSSN via its definition (3.12):

$$\tilde{\Lambda}^k \equiv \tilde{\gamma}^{ij}\left(\tilde{\Gamma}^k_{ij} - \mathring{\Gamma}^k_{ij}\right) \equiv \tilde{\Gamma}^k - \mathring{\Gamma}^k_{ij}\tilde{\gamma}^{ij} \Rightarrow \tilde{\Gamma}^k = \tilde{\Lambda}^k + \tilde{\gamma}^{ij}\mathring{\Gamma}^k_{ij}\,. \tag{4.13}$$

We remind the reader that the 3-Ricci tensor, R_{ij}, in (4.7) is computed using the

G-BSSN primary variables as:

$$R_{ij} = R^{\phi}_{ij} + \tilde{R}_{ij}\,, \tag{4.14}$$

in which:

$$R^{\phi}_{ij} = -2\tilde{D}_i\tilde{D}_j\phi - 2\tilde{\gamma}_{ij}\tilde{D}^k\tilde{D}_k\phi + 4\tilde{D}_i\phi\tilde{D}_j\phi - 4\tilde{\gamma}_{ij}\tilde{D}^k\phi\tilde{D}_k\phi\,, \tag{4.15}$$

$$\tilde{R}_{ij} = -\frac{1}{2}\tilde{\gamma}^{lm}\partial_m\partial_l\tilde{\gamma}_{ij} + \tilde{\gamma}_{k(i}\partial_{j)}\tilde{\Gamma}^k + \tilde{\Gamma}^k\tilde{\Gamma}_{(ij)k} + \tilde{\gamma}^{lm}\left(2\tilde{\Gamma}^k_{l(i}\tilde{\Gamma}_{j)km} + \tilde{\Gamma}^k_{im}\tilde{\Gamma}_{klj}\right)\,. \tag{4.16}$$

In the expression for the conformal Ricci tensor, $\tilde{R}_{ij}$, we again use (4.13) to eliminate $\tilde{\Gamma}^k$ since in G-BSSN, $\tilde{\Lambda}^k$ is the primary dynamical variable, rather than $\tilde{\Gamma}^k$.

The Hamiltonian and momentum constraints for pure vacuum are given by:

$$\mathscr{H} \;\equiv\; \tilde{\gamma}^{ij}\tilde{D}_i\tilde{D}_j e^{\phi} - \frac{e^{\phi}}{8}\tilde{R} + \frac{e^{5\phi}}{8}\tilde{A}^{ij}\tilde{A}_{ij} - \frac{e^{5\phi}}{12}K^2 = 0\,, \tag{4.17}$$

$$\mathscr{M}^i \equiv \tilde{D}_j\left(e^{6\phi}\tilde{A}^{ji}\right) - \frac{2}{3}e^{6\phi}\tilde{D}^i K = 0\,, \tag{4.18}$$

and are only used at the initial time in the G-BSSN formulation. As we will discuss, we monitor the constraints as a diagnostic tool.

4.2.1 G-BSSN in Cylindrical Coordinate with Axial Symmetry

We now proceed by imposing the axial symmetry in cylindrical coordinates. Assuming that ∂_φ is a Killing vector, we consider the following form for the conformal 3-metric in cylindrical coordinates:

$$\tilde{\gamma}_{ij} = \begin{pmatrix} \tilde{\gamma}_{\rho\rho}(t,\rho,z) & \tilde{\gamma}_{\rho z}(t,\rho,z) & 0 \\ \tilde{\gamma}_{\rho z}(t,\rho,z) & \tilde{\gamma}_{zz}(t,\rho,z) & 0 \\ 0 & 0 & \rho^2\tilde{\gamma}_{\varphi\varphi}(t,\rho,z) \end{pmatrix}\,. \tag{4.19}$$

This reduces to the flat 3-metric (4.3) when the off-diagonal component, $\tilde{\gamma}_{\rho z}$ vanishes and the diagonal terms are unity. The traceless part of the extrinsic curvature has similar non-zero components:

$$\tilde{A}_{ij} = \begin{pmatrix} \tilde{A}_{\rho\rho}(t,\rho,z) & \tilde{A}_{\rho z}(t,\rho,z) & 0 \\ \tilde{A}_{\rho z}(t,\rho,z) & \tilde{A}_{zz}(t,\rho,z) & 0 \\ 0 & 0 & \rho^2 \tilde{A}_{\varphi\varphi}(t,\rho,z) \end{pmatrix} , \tag{4.20}$$

and all of the other terms on the right hand side of the G-BSSN equations (4.4,4.8) such as R_{ij}, $D_i D_j \alpha$ and the Lie derivative terms, have similar non-zero components, consistent with these ansatzes.

Furthermore, due to the axial symmetry, the shift vector and conformal connection function can only have non-zero components in the ρ and z directions:

$$\beta^i = [\beta^\rho(t,\rho,z), \beta^z(t,\rho,z), 0] \, , \tag{4.21}$$

$$\tilde{\Lambda}^i = [\tilde{\Lambda}^\rho(t,\rho,z), \tilde{\Lambda}^z(t,\rho,z), 0] \, . \tag{4.22}$$

4.2.2 Coordinate Choices

For this study, we use the standard 1+log and Gamma-driver[35] conditions to evolve the lapse and shift[36],

$$\partial_t \alpha(t,\rho,z) = -2\alpha K \, , \tag{4.23}$$

$$\partial_t \beta^i = \mu \tilde{\Lambda}^i - \eta \beta^i \, , \tag{4.24}$$

[35] The choice of $\beta = 0$, which worked fine in the spherical case in the previous chapter, is not an effective coordinate choice here. In particular, using $\beta = 0$ the off-diagonal component of the conformal 3-metric grows and causes the metric to become singular (non-invertable).

[36] We note that the code implements an advective version of these equations with the $\beta^i \partial_i$ term on the LHS and a more general expression on the RHS, $a\alpha K + b\alpha^2 K$, for experimental purposes. However the simulations shown in the results section use only the standard choice.

where μ and η are adjustable parameters. For most of the calculations presented here we use $\mu = 3/4$ and $\eta \approx 1/(10M)$, where M is the total mass of the system (ADM mass). As will be discussed in Sec. 4.3.2, we choose an initialization such that the mass is of order 1, therefore $\eta \approx 1/(10M)$ will be of order 10^{-1}.

4.2.3 Note on Complexity and Regularity of the Equations

We note that the G-BSSN equations (4.4,4.8), even limited to axial symmetry, contain tens of thousands of terms if the right hand side of the equations are expressed in terms of the conformal metric $\tilde{\gamma}_{ij}$ and the rest of the primary variables of G-BSSN. To reduce this complexity, and avoid potentially repetitive calculations, we introduce the inverse of the conformal metric $\tilde{\gamma}^{ij}$ as a new set of "work" variables in the numerical solver. Further, we compute the various covariant derivatives using some of the components of the conformal Christoffel symbols defined as additional work variables:

$$
\begin{aligned}
\tilde{\Gamma}^{\rho}_{\rho\rho}(t,\rho,z) &= \frac{1}{2}\tilde{\gamma}^{\rho\rho}\partial_\rho\tilde{\gamma}_{\rho\rho} + \tilde{\gamma}^{\rho z}\partial_\rho\tilde{\gamma}_{\rho z} - \frac{1}{2}\tilde{\gamma}^{\rho z}\partial_z\tilde{\gamma}_{\rho\rho}\,, && (4.25)\\
\tilde{\Gamma}^{\rho}_{z\rho}(t,\rho,z) &= \frac{1}{2}\tilde{\gamma}^{\rho z}\partial_\rho\tilde{\gamma}_{\rho\rho} + \tilde{\gamma}^{zz}\partial_\rho\tilde{\gamma}_{\rho z} - \frac{1}{2}\tilde{\gamma}^{zz}\partial_z\tilde{\gamma}_{\rho\rho}\,, && (4.26)\\
\tilde{\Gamma}^{\rho}_{zz}(t,\rho,z) &= \tilde{\gamma}^{\rho\rho}\partial_z\tilde{\gamma}_{\rho z} - \frac{1}{2}\tilde{\gamma}^{\rho\rho}\partial_\rho\tilde{\gamma}_{zz} + \frac{1}{2}\tilde{\gamma}^{\rho z}\partial_z\tilde{\gamma}_{zz}\,, && (4.27)\\
\tilde{\Gamma}^{z}_{\rho\rho}(t,\rho,z) &= \frac{1}{2}\tilde{\gamma}^{\rho z}\partial_\rho\tilde{\gamma}_{\rho\rho} + \tilde{\gamma}^{zz}\partial_\rho\tilde{\gamma}_{\rho z} - \frac{1}{2}\tilde{\gamma}^{zz}\partial_z\tilde{\gamma}_{\rho\rho}\,, && (4.28)\\
\tilde{\Gamma}^{z}_{z\rho}(t,\rho,z) &= \frac{1}{2}\tilde{\gamma}^{\rho z}\partial_z\tilde{\gamma}_{\rho\rho} + \frac{1}{2}\tilde{\gamma}^{zz}\partial_\rho\tilde{\gamma}_{zz}\,, && (4.29)\\
\tilde{\Gamma}^{z}_{zz}(t,\rho,z) &= \tilde{\gamma}^{\rho z}\partial_z\tilde{\gamma}_{\rho z} - \frac{1}{2}\tilde{\gamma}^{\rho z}\partial_\rho\tilde{\gamma}_{zz} + \frac{1}{2}\tilde{\gamma}^{zz}\partial_z\tilde{\gamma}_{zz}\,, && (4.30)\\
\tilde{\Gamma}^{\varphi}_{\varphi z}(t,\rho,z) &= \frac{1}{2}\frac{\partial_z\tilde{\gamma}_{\varphi\varphi}}{\tilde{\gamma}_{\varphi\varphi}}\,. && (4.31)
\end{aligned}
$$

Specifically, we use the left hand side (LHS) symbols in in the G-BSSN equations, while their values are given by the RHS and stored in separate work variables. For the remaining components of the conformal Christoffel symbols:

$$\begin{aligned}
\tilde{\Gamma}^{\varphi}_{\varphi\rho}(t,\rho,z) &= \frac{1}{\rho} + \frac{1}{2}\frac{\partial_\rho \tilde{\gamma}_{\varphi\varphi}}{\tilde{\gamma}_{\varphi\varphi}}\,, && (4.32)\\
\tilde{\Gamma}^{z}_{\varphi\varphi}(t,\rho,z) &= -\rho\tilde{\gamma}^{\rho z}\tilde{\gamma}_{\varphi\varphi} - \frac{1}{2}\rho^2\tilde{\gamma}^{\rho z}\partial_\rho\tilde{\gamma}_{\varphi\varphi} - \frac{1}{2}\rho^2\tilde{\gamma}^{zz}\partial_z\tilde{\gamma}_{\varphi\varphi}\,, && (4.33)\\
\tilde{\Gamma}^{\rho}_{\varphi\varphi}(t,\rho,z) &= -\rho\tilde{\gamma}^{\rho\rho}\tilde{\gamma}_{\varphi\varphi} - \frac{1}{2}\rho^2\tilde{\gamma}^{\rho\rho}\partial_\rho\tilde{\gamma}_{\varphi\varphi} - \frac{1}{2}\rho^2\tilde{\gamma}^{\rho z}\partial_z\tilde{\gamma}_{\varphi\varphi}\,, && (4.34)
\end{aligned}$$

we explicitly use the RHS expressions in the G-BSSN equations. This is important, as they contain powers of ρ and similarly our ansatz for the conformal metric (4.19) has explicit ρ^2 dependency and its inverse has explicit $1/\rho^2$ term. Therefore, at several places in the symbolic calculations, the powers of ρ appear and can cancel and simplify to regular terms. An example of such a cancellation is in the term $\tilde{\gamma}^{lm}\left(2\tilde{\Gamma}^{k}_{l(i}\tilde{\Gamma}_{j)km} + \tilde{\Gamma}^{k}_{im}\tilde{\Gamma}_{klj}\right)$ in the Ricci tensor (4.16). In addition, $\tilde{\Gamma}^{\varphi}_{\varphi\rho}$ contains a singular $1/\rho$ term, which will be appropriately eliminated by the subtraction of the flat background Christoffel symbols ($\mathring{\Gamma}^{\varphi}_{\varphi\rho} = 1/\rho$) in the re-definition of the conformal connection (4.13).

Even with the use of the Christoffel symbols as intermediate variables, the LHS of G-BSSN equations are rather lengthy and of course need to be derived using a tensor manipulation software. For example , the $\rho\rho$ component of the 3-Ricci tensor that appears in the RHS of the $\rho\rho$ component of Eq. (4.7) is given by:

$$\begin{aligned}
R_{\rho\rho} &= \tilde{R}_{\rho\rho} + R^{\phi}_{\rho\rho} = \\
&- \frac{1}{2}\tilde{\gamma}^{\rho\rho}\partial^2_\rho\tilde{\gamma}_{\rho\rho} - \frac{1}{2}\tilde{\gamma}^{zz}\partial^2_z\tilde{\gamma}_{\rho\rho} - \tilde{\gamma}^{\rho z}\partial_z\partial_\rho\tilde{\gamma}_{\rho\rho} + \tilde{\gamma}_{\rho z}\partial_\rho\tilde{\Lambda}^z + \tilde{\gamma}_{\rho\rho}\partial_\rho\tilde{\Lambda}^\rho \\
&+ \frac{1}{2}\tilde{\Lambda}^z\partial_z\tilde{\gamma}_{\rho\rho} + \frac{1}{2}\tilde{\Lambda}^\rho\partial_\rho\tilde{\gamma}_{\rho\rho} + \partial_z\tilde{\gamma}_{\rho\rho}\left(\frac{3}{2}\tilde{\gamma}^{\rho z}\tilde{\Gamma}^\rho_{\rho\rho} + \frac{3}{2}\tilde{\gamma}^{zz}\tilde{\Gamma}^\rho_{z\rho} + \frac{1}{2}\tilde{\gamma}^{\rho z}\tilde{\Gamma}^z_{z\rho} + \frac{1}{2}\tilde{\gamma}^{\rho\rho}\tilde{\Gamma}^z_{\rho\rho}\right) \\
&+ \partial_\rho\tilde{\gamma}_{\rho\rho}\left(\frac{3}{2}\tilde{\gamma}^{\rho\rho}\tilde{\Gamma}^\rho_{\rho\rho} + \frac{3}{2}\tilde{\gamma}^{\rho z}\tilde{\Gamma}^\rho_{z\rho}\right) + \partial_\rho\tilde{\gamma}_{zz}\left(-\frac{1}{2}\tilde{\gamma}^{zz}\tilde{\Gamma}^z_{z\rho} - \frac{1}{2}\tilde{\gamma}^{\rho z}\tilde{\Gamma}^z_{\rho\rho}\right) \\
&+ \partial_\rho\tilde{\gamma}_{\rho z}\left(\tilde{\gamma}^{\rho\rho}\tilde{\Gamma}^z_{\rho\rho} + \tilde{\gamma}^{\rho z}\tilde{\Gamma}^z_{z\rho}\right) + \partial_z\tilde{\gamma}_{\rho z}\left(2\tilde{\gamma}^{\rho z}\tilde{\Gamma}^z_{\rho\rho} + 2\tilde{\gamma}^{zz}\tilde{\Gamma}^z_{z\rho}\right) + \frac{1}{4}\frac{\partial_\rho\tilde{\gamma}^2_{\varphi\varphi}}{\tilde{\gamma}^2_{\varphi\varphi}} \\
&+ \frac{1}{\rho}\left(-\frac{1}{2}\frac{\partial_\rho\tilde{\gamma}_{\rho\rho}}{\tilde{\gamma}_{\varphi\varphi}} - \frac{\partial_\rho\tilde{\gamma}_{\varphi\varphi}}{\tilde{\gamma}_{\varphi\varphi}} + \tilde{\gamma}_{\rho\rho}\frac{\partial_\rho\tilde{\gamma}_{\varphi\varphi}}{\tilde{\gamma}^2_{\varphi\varphi}}\right) + \frac{1}{\rho^2}\left(\frac{\tilde{\gamma}_{\rho\rho}}{\tilde{\gamma}_{\varphi\varphi}} - 1\right) \quad \star \\
&+ \partial_\rho\phi\left(2\tilde{\Gamma}^\rho_{\rho\rho} + 2\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho\rho}\tilde{\Gamma}^\rho_{\rho\rho} + 4\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho z}\tilde{\Gamma}^\rho_{z\rho} + 2\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{zz}\tilde{\Gamma}^\rho_{zz}\right) \\
&+ \partial_z\phi\left(2\tilde{\Gamma}^z_{\rho\rho} + 2\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho\rho}\tilde{\Gamma}^z_{\rho\rho} + 4\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho z}\tilde{\Gamma}^z_{z\rho} + 2\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{zz}\tilde{\Gamma}^z_{zz}\right) \\
&+ \partial_\rho\phi\left(-\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho\rho}\frac{\partial_\rho\tilde{\gamma}_{\varphi\varphi}}{\tilde{\gamma}_{\varphi\varphi}} - \tilde{\gamma}^{\rho z}\tilde{\gamma}_{\rho\rho}\frac{\partial_z\tilde{\gamma}_{\varphi\varphi}}{\tilde{\gamma}_{\varphi\varphi}}\right) + \partial_z\phi\left(-\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho z}\frac{\partial_\rho\tilde{\gamma}_{\varphi\varphi}}{\tilde{\gamma}_{\varphi\varphi}} - \tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{zz}\frac{\partial_z\tilde{\gamma}_{\varphi\varphi}}{\tilde{\gamma}_{\varphi\varphi}}\right) \\
&+ (\partial_\rho\phi)^2\left(4 - 4\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho\rho}\right) - 4(\partial_z\phi)^2\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{zz} + 4\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho z}\partial_\rho\partial_z\phi \\
&- 2\partial^2_\rho\phi - 2\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho\rho}\partial^2_\rho\phi - 2\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{zz}\partial^2_z\phi \\
&+ \frac{1}{\rho}\left(-2\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho\rho}\partial_\rho\phi - 2\tilde{\gamma}_{\rho\rho}\tilde{\gamma}^{\rho z}\partial_z\phi\right) \quad \star \quad ,
\end{aligned} \tag{4.35}$$

in which we highlighted the potentially irregular terms containing negative powers of ρ. We note that these terms are indeed regular in the limit $\rho \to 0$ as we will discuss shortly, but it is nonetheless crucial to collect the powers of ρ and compute these terms in an appropriate way to prevent round-off errors. For instance, if the term containing $1/\rho^2$ is computed as two separate terms: $\tilde{\gamma}_{\rho\rho}/(\rho^2\tilde{\gamma}_{\varphi\varphi})$ and $-1/\rho^2$, both these terms are large (in fact diverging) floating point numbers at the vicinity of the origin, and if added to any other regular term, before being summed together, can create a significant round-off error. However, if the subtraction $\tilde{\gamma}_{\rho\rho}/\tilde{\gamma}_{\varphi\varphi} - 1$ is performed first, the value is a small floating point number, (behaving as ρ^2 in the limit $\rho \to 0$ as we will discuss in Sec. 4.3.4) and the division by ρ^2 creates a number that is of the same order as the rest of the terms in the expression.

4.2.4 Axisymmetric Initial Data

The initialization of a pure gravity wave is done by specifying a non-trivial form for the conformal 3-metric, $\tilde{\gamma}_{ij}$, and then solving the Hamiltonian constraint (4.17) for the conformal factor e^{ϕ}. The momentum constraint (4.18) is satisfied trivially by requiring that the initial data be time symmetric:

$$\tilde{A}_{ij} = K = 0\,. \tag{4.36}$$

Furthermore, without loss of generality, one can assume that at the initial time the ρ and z coordinates are orthogonal, i.e. the off-diagonal term in (4.19) is zero:

$$\tilde{\gamma}_{ij}(t=0,\rho,z) = \begin{pmatrix} A(\rho,z) & 0 & 0 \\ 0 & B(\rho,z) & 0 \\ 0 & 0 & \rho^2 C(\rho,z) \end{pmatrix}, \tag{4.37}$$

in which the functions A, B and C can be chosen arbitrarily. In addition it is easy to perform a coordinate transformation $z \to z'$ (at initial time slice) to make the zz and $\rho\rho$ components of $\tilde{\gamma}_{ij}$ equal (any two-metric can be written in a conformally flat form). Therefore, a suitably generic axisymmetric initial conformal 3-metric is given by:

$$\tilde{\gamma}_{ij}(t=0,\rho,z) = \begin{pmatrix} e^{V(\rho,z)} & 0 & 0 \\ 0 & e^{V(\rho,z)} & 0 \\ 0 & 0 & \rho^2 W(\rho,z) \end{pmatrix}. \tag{4.38}$$

Here we use e^V for the first two diagonal components of the conformal metric to be consistent with the literature. Note that this form for $\tilde{\gamma}_{ij}$ holds only at the initial time; during the evolution the metric evolves to a form given by (4.19). In this chapter, we particularly focus on two different types of initial data:

$$W(\rho, z) = 1\,, \tag{4.39}$$

and

$$W(\rho, z) = e^{-2V(\rho,z)}\,. \tag{4.40}$$

4.2.5 Brill Initial Data

The first choice (4.39), known as Brill initial data, yields the conformal metric:

$$\tilde{\gamma}^{(B)}_{ij}(t=0,\rho,z) \equiv \begin{pmatrix} e^{V(\rho,z)} & 0 & 0 \\ 0 & e^{V(\rho,z)} & 0 \\ 0 & 0 & \rho^2 \end{pmatrix}\,, \tag{4.41}$$

for which the Hamiltonian constraint simplifies to:

$$0 = \mathcal{H} = \left(\partial_\rho^2 + \partial_z^2 + \frac{1}{\rho}\partial_\rho\right)\psi(\rho,z) + \frac{1}{8}\psi(\rho,z)\left(\partial_\rho^2 + \partial_z^2\right)V(\rho,z)\,. \tag{4.42}$$

Here $\psi = e^{\phi}$ is the conformal factor, $V(\rho, z)$ will be chosen to a localized function, with proper behaviour in the limit $\rho \to 0$.

4.2.6 Teukolsky-type Initial Data

The second option (4.40) creates a conformal 3-metric with the following form:

$$\tilde{\gamma}^{(T)}_{ij}(t=0,\rho,z) \equiv \begin{pmatrix} e^{V(\rho,z)} & 0 & 0 \\ 0 & e^{V(\rho,z)} & 0 \\ 0 & 0 & \rho^2 e^{-2V(\rho,z)} \end{pmatrix}\,, \tag{4.43}$$

and the Hamiltonian constraint simplifies to:

$$\begin{aligned} 0 = \mathcal{H} &= \left(\partial_\rho^2 + \partial_z^2 + \frac{1}{\rho}\partial_\rho - \partial_\rho V \partial_\rho - \partial_z V \partial_z\right)\psi(\rho,z) \\ &- \frac{1}{8}\psi(\rho,z)\left[\left(\partial_\rho^2 + \partial_z^2 + \frac{4}{\rho}\partial_\rho\right)V(\rho,z) - 2(\partial_\rho V)^2 - 2(\partial_z V)^2\right] . \end{aligned} \tag{4.44}$$

We refer to this initialization as *Teukolsky-type* initial data as it somewhat mimics the Teukolsky wave [96] in the weak field limit, $V(\rho,z) \ll 1$. However, Teukolsky's solution is *not* time symmetric, while our choose (4.36) is time symmetric.

The main difference between the Brill initialization (4.41) and Teukolsky-type data is the fact that the Teukolsky-type initial data has determinant equal to that of the flat cylindrical metric, while that is not the case for Brill data. In the weak field limit, $V(\rho,z) << 1$, this translates to a difference in the trace of the two linearized metrics. The Teukolsky-type initial metric can be written as:

$$\tilde{\gamma}_{ij}^{(T)} \approx \mathring{\gamma}_{ij} + \eta_{ij}^{(T)} \,, \tag{4.45}$$

in which $\mathring{\gamma}_{ij}$ is the flat cylindrical metric, and $\eta_{ij}^{(T)}$ is the linearized deviation and is traceless:

$$\eta_{ij}^{(T)} = \begin{pmatrix} V(\rho,z) & 0 & 0 \\ 0 & V(\rho,z) & 0 \\ 0 & 0 & -2\rho^2 V(\rho,z) \end{pmatrix} \Rightarrow \mathrm{Tr}(\eta_{ij}^{(T)}) = \mathring{\gamma}^{ij}\eta_{ij}^{(T)} = 0 \,, \tag{4.46}$$

which parallels the traceless property of the linearized solution to the Einstein's equations in cylindrical coordinate. This solution, as will be seen, propagates similarly to a wave packet. However, in a similar linearization of the Brill initial data, the perturbation metric, $\eta_{ij}^{(B)}$, is *not* traceless:

$$\eta_{ij}^{(B)} = \begin{pmatrix} V(\rho,z) & 0 & 0 \\ 0 & V(\rho,z) & 0 \\ 0 & 0 & 0 \end{pmatrix} \Rightarrow \mathrm{Tr}(\eta_{ij}^{(B)}) = \mathring{\gamma}^{ij}\eta_{ij}^{(B)} = 2V(\rho,z)\,, \tag{4.47}$$

and as we will show in Sec. 4.4, the Brill data does not propagate like a wave packet, even in the weak field limit.

4.2.7 Computing the ADM Mass of the Gravitational Pulse

The ADM mass[15] is defined as:

$$M_{ADM} \equiv \frac{1}{16\pi}\int_{\partial\Sigma_\infty} \sqrt{\gamma}\gamma^{jn}\gamma^{im}\left(\partial_j\gamma_{mn} - \partial_m\gamma_{jn}\right) dS_i\,, \tag{4.48}$$

where $\partial\Sigma_\infty$ denotes the boundary surface at spatial infinity, $dS_i = N_i\sqrt{\gamma|_{\partial\Sigma_\infty}}dX^2$, is the surface element vector, $dX^2 = dX_1 dX_2$, where X_1 and X_2 are the coordinates on the boundary surface, $\gamma|_{\partial\Sigma_\infty}$ is the determinant of the reduced metric on $\partial\Sigma_\infty$, and finally, N_i is the unit normal vector to $\partial\Sigma_\infty$ ($N^iN_i = 1$). As described in [28], for a metric that is asymptotically conformally flat, (4.48) can be written as a volume integral:

$$M_{ADM} = -\frac{1}{2\pi}\int_\Sigma dx^3\sqrt{\tilde{\gamma}}\tilde{D}^2\psi\,. \tag{4.49}$$

Further, substituting $\tilde{D}^2\psi$ from the Hamiltonian constraint we have,

$$M_{ADM} = \frac{1}{16\pi}\int_\Sigma dx^3\sqrt{\tilde{\gamma}}\left(\psi^{-7}\tilde{A}_{ij}\tilde{A}^{ij} - \psi\tilde{R} - \frac{2}{3}\psi^5K^2\right)\,. \tag{4.50}$$

Finally, since $\tilde{A}_{ij} = K = 0$ at the initial time the ADM mass simplifies to:

$$M_{ADM} = -\frac{1}{16\pi}\int_\Sigma dx^3\sqrt{\tilde{\gamma}}\psi\tilde{R}\,, \tag{4.51}$$

which is a convenient expression for computing the ADM mass using BSSN variables.

4.3 Numerics

In this section, we summarize the numerical techniques used to initialize and evolve the G-BSSN system of equations. We also describe the diagnostic tools we implemented to validate the results and test the correctness of the numerical solver as well as the equations themselves.

4.3.1 Numerical Grid

Before discretizing the PDEs of the G-BSSN formulation, we first transform to non-uniform spatial coordinates denoted by $\tilde{\rho}$ and $\tilde{z}$ (their definition will follow shortly). This coordinate transformation has two purposes: 1) to compactify the coordinate by mapping the domain of the cylindrical coordinate to a numerical domain that contains spatial infinity:

$$D_{(\rho,z)} = [0,+\infty] \times [-\infty,+\infty] \to [\tilde{\rho}_{\mathrm{m}in}, \tilde{\rho}_{\mathrm{m}ax}] \times [\tilde{z}_{\mathrm{m}in}, \tilde{z}_{\mathrm{m}ax}] = D_{(\tilde{\rho},\tilde{z})}\,, \tag{4.52}$$

2) increasing resolution toward the origin, $(\rho, z) = (0, 0)$. Choosing a non-linear transformation, a uniform mesh in the compactified coordinate $(\tilde{\rho}, \tilde{z})$ is effectively a non-uniform grid in the cylindrical coordinate. The mesh step sizes in each direction of the coordinates are related by (approximately):

$$\Delta\rho = \frac{\partial\rho(\tilde{\rho})}{\partial\tilde{\rho}}\Delta\tilde{\rho}\,, \tag{4.53}$$

$$\Delta z = \frac{\partial z(\tilde{z})}{\partial\tilde{z}}\Delta\tilde{z}\,, \tag{4.54}$$

Therefore, by defining the functions $\tilde{\rho}(\rho)$ and $\tilde{z}(z)$ such that their derivatives decrease toward the origin we can achieve the desired non-uniform grid. Specifically, we choose the following two functions:

$$\rho = \exp(\tilde{\rho}) - \exp(\delta) + \frac{1}{1-\tilde{\rho}/R_\infty} - \frac{1}{1-\delta/R_\infty}\,, \tag{4.55}$$

$$z = \frac{1}{2}\left(\exp(\tilde{z}+\delta) - \exp(-\tilde{z}+\delta) + \frac{1}{1-(\tilde{z}+\delta)/R_\infty} - \frac{1}{1-(-\tilde{z}+\delta)/R_\infty}\right)\,. \tag{4.56}$$

The reader may easily check that the cylindrical coordinate domain, $[0,+\infty]_\rho \times [-\infty,+\infty]_z$, is mapped to the numerical domain $[\delta, R_\infty]_{\tilde{\rho}} \times [-(R_\infty-\delta), +(R_\infty-\delta)]_{\tilde{z}}$. We also define the notations $\partial\tilde{\Sigma}$:

$$\partial\tilde{\Sigma} \equiv \partial\tilde{\Sigma}^z_{\rho=\infty} \cup \partial\tilde{\Sigma}^{z=-\infty}_\rho \cup \partial\tilde{\Sigma}^{z=+\infty}_\rho \tag{4.57}$$

in which:

$$\partial\tilde{\Sigma}^z_{\rho=\infty} \equiv [R_\infty]_{\tilde{\rho}} \times [-(R_\infty-\delta), +(R_\infty-\delta)]_{\tilde{z}} \tag{4.58}$$

$$\partial\tilde{\Sigma}^{z=-\infty}_\rho \equiv [\delta, R_\infty]_{\tilde{\rho}} \times [-(R_\infty-\delta)]_{\tilde{z}} \tag{4.59}$$

$$\partial\tilde{\Sigma}^{z=+\infty}_\rho \equiv [\delta, R_\infty]_{\tilde{\rho}} \times [+(R_\infty-\delta)]_{\tilde{z}} \tag{4.60}$$

to denote the outer numerical boundaries that correspond to asymptotically flat spatial infinity. The inner boundary, which is the symmetric axis, is simply denoted by $(\rho = 0, z)$ and corresponds to $(\tilde{\rho} = -\delta, \tilde{z})$ on the numerical grid.

The parameters R_∞ and δ can be adjusted to change the numerical location of spatial infinity and the origin, therefore effectively changing the distribution of grid points across the spatial domain. For the choice (4.55,4.56) a uniform grid on the compactified coordinates creates an approximately uniform grid on a logarithmic scale, $\ln(\rho)$ and $\ln(z)$, at the vicinity of the origin. Such behaviour is presented in Fig. 4.1 in a rather coarse grid with 64 points (to better demonstrate the distribution

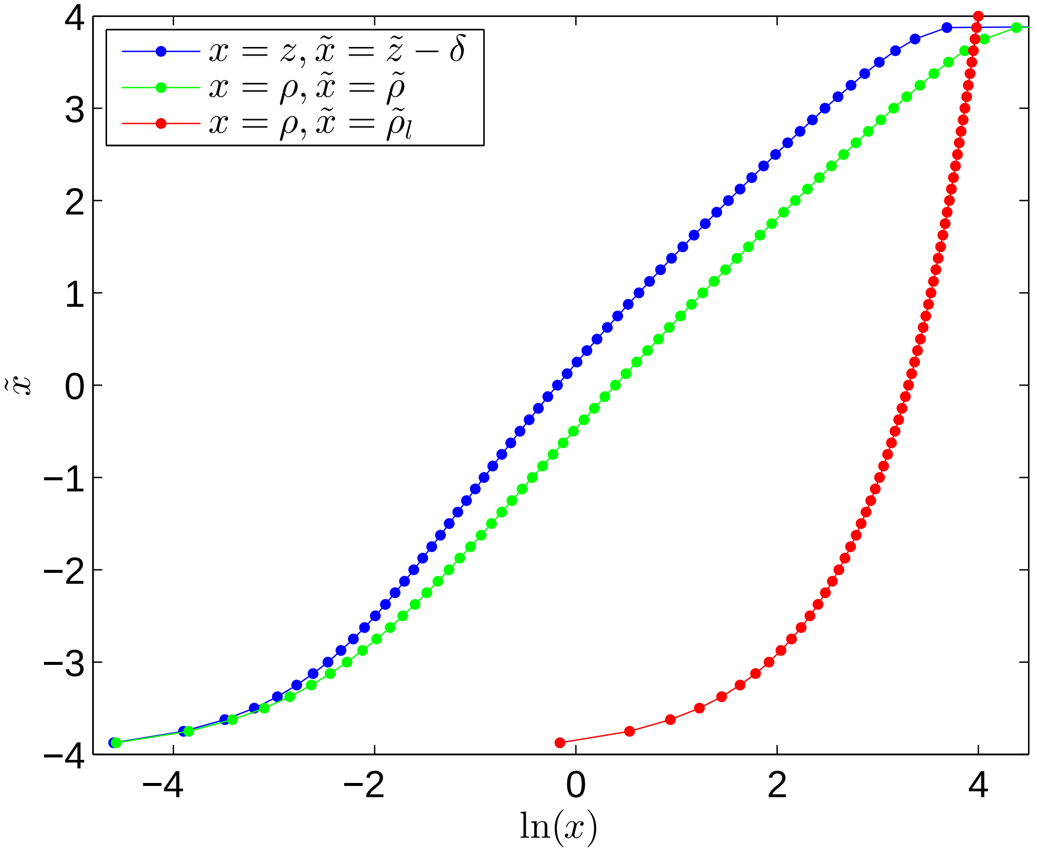


Figure 4.1: Distribution of grid points on a non-uniform grid with 64 points for the choice of (4.55,4.56) for ρ and z coordinates (green and blue curve respectively) using the parameters: $\delta = -4$ and $R_\infty = 4$. Points are distributed uniformly across the compactified coordinate (vertical axis, $\tilde{x}$). As it is clear from both curves, the points are also approximately distributed uniformly in the logarithm of the spatial coordinate (horizontal axis $\ln x$) for a range that expands up to $\ln(x) \approx -3$. Note that the blue curve that corresponds to the z coordinate is shifted by constant δ to have the same range as $\tilde{\rho}$. For comparison, we also plotted a uniform grid structure (the red curve), in which the 64 points are distributed uniformly in a *linear* coordinate $\rho = a\tilde{\rho}_l + b$ that maps $(-4, 4)$ to the range $\rho \in (0, 54)$ (to match the last points of the other two graphs: $\ln(54) \approx 4$). As is clear, the choices (4.55,4.56) provides much more resolution in the vicinity of the origin, $x = 0$, compared to the red curve. Approximately half of the grid points are in the region $\ln(x) < 0$ for our logarithmic coordinate choice, while a uniform grid has only 1 grid point next to the origin in that region, while the rest of the points are located in a region that is not of interest when the gravitational wave is focused in the vicinity of the origin.

of the points).

4.3.2 Initialization

We use three different types of profile to initialize the function V in (4.39,4.40) that defines the conformal metric. The standard choice of a localized function is often a Gaussian profile:

$$V(\rho, z) = A\rho^2 \exp\left(-\frac{(\rho-\rho_0)^2}{\Delta_\rho^2} - \frac{(z-z_0)^2}{\Delta_z^2}\right) . \tag{4.61}$$

We also use the following *seed functions* (choice of F is similar to [107]) to create wave packets:

$$F(r) := A\kappa\lambda^5 \left[1 - \left(\frac{r}{\lambda}\right)^2\right]^6 , \tag{4.62}$$

$$G(r) := A\kappa\lambda \left(1 - (\frac{r}{\lambda})^2\right)^4 . \tag{4.63}$$

These seed functions are used as following to initialize V:

$$V(\rho, z) = \left(\frac{F^{(4)}(r-r_0)}{r} - 2\frac{F^{(3)}(r-r_0)}{r^2}\right)\sin^2\theta\,, \tag{4.64}$$

$$V(\rho, z) = \left(\frac{G^{(1)}(r-r_0)}{r} - 2\frac{G(r-r_0)}{r^2}\right)\sin^2\theta\,. \tag{4.65}$$

where $F^{(n)}$ denotes the n-th derivative of F. Specifically, we use a Gaussian profile and G for Brill-type initial data, and F and G for a Teukolsky-type initial wave packet. Here κ is a normalization factor (different for the two functions) chosen such that when the amplitude A is set to a value of order 1—along with the typical choices of λ in this study—the ADM mass of the gravity wave pulse is also of order 1. In particular we have chosen $\kappa \approx 3.7 \times 10^{-4}$ for F, and for G, κ is approximately 1. The parameter λ determines the typical length of the wave packet. r is the radial

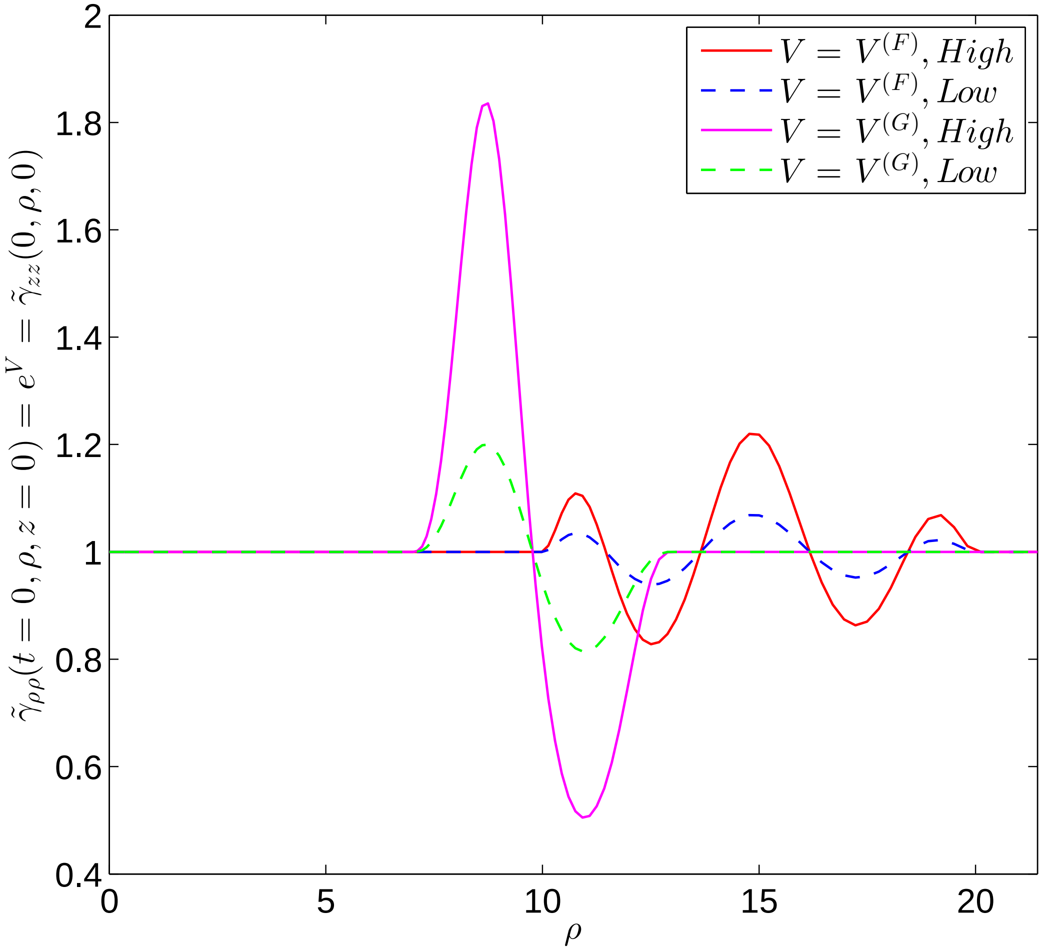


Figure 4.2: Initial profile of the conformal metric component $\tilde{\gamma}_{zz}$ or $\tilde{\gamma}_{\rho\rho}$ along the radial direction ($\theta = \pi/2$, or $z = 0$) in cylindrical coordinates for two choices of F and G in (4.64,4.65). Here the solid lines are the strong initial data that collapses and are associated with amplitudes $A = 3.0$ and $A = 2.0$ in Eq. (4.65) and Eq. (4.64) respectively. The dashed lines are still in a non-linear regime but will disperse after reflecting back from the center. They are associated with amplitudes $A = 1.0$ and $A = 0.6$ in (4.65) and (4.64) respectively. For the seed function F the rest of the parameters are set to: $r_0 = 15$ and $\lambda = 5.0$. For G as a seed function, the parameters are: $r_0 = 10$ and $\lambda = 3$. As one can see, F creates a wave packet with 5 extrema while G has a simpler form with 2 extrema.

distance from the center:

$$r^2 \equiv \rho^2 + z^2 \,, \tag{4.66}$$

and sin(θ) is the polar angle defined as:

$$\sin\theta \equiv \frac{\rho}{r} \,. \tag{4.67}$$

The typical shapes of the initial conformal metric constructed from the seed functions F and G are given in Fig. 4.2 for strong (black hole formation) and weak (dispersal) initial data.

The lapse function is set to unity at the initial time:

$$\alpha(t=0,\rho,z) = 1 \,, \tag{4.68}$$

and since we choose time symmetric initial data, we have:

$$\beta^\rho(t=0,\rho,z) = \beta^z(t=0,\rho,z) = 0 \,, \tag{4.69}$$

$$\tilde{A}_{ij}(t=0,\rho,z) = K(t=0,\rho,z) = 0 \,, \tag{4.70}$$

at the initial time. The conformal connection function, $\tilde{\Lambda}^k$, is initialized using its definition and the given initial conformal metric:

$$\tilde{\Lambda}^k(t=0,\rho,z) = \tilde{\gamma}^{ij}(t=0,\rho,z)\left(\tilde{\Gamma}^k_{ij}(t=0,\rho,z) - \mathring{\Gamma}^k_{ij}\right) \,. \tag{4.71}$$

Finally, the G-BSSN variable ϕ is initialized from its relation to the conformal factor:

$$\phi(t=0,\rho,z) = \ln(\psi(\rho,z))|_{\mathcal{H}(\psi)=0} \tag{4.72}$$

in which $|_{\mathcal{H}(\psi)=0}$ denotes that ψ solves the Hamiltonian constraint, (4.42) or (4.44) depending on the choice of Brill or Teukolsky-type initialization. In both case, ψ satisfies the boundary conditions:

$$\partial_\rho \psi(\rho, z)|_{(\rho=0,z)} = 0 \tag{4.73}$$

$$\psi|_{\partial\tilde{\Sigma}} = 1 \tag{4.74}$$

where $\partial\tilde{\Sigma}$ is defined in (4.57), and denotes the 3 outer boundaries of the numerical domain corresponding to spatial infinity.

We note that the multigrid solver that we use (PAMR's default MG solver) is sensitive to imposing inner boundary conditions such as (4.73). Therefore we use a regularized version of the Hamiltonian constraint equations (4.42,4.44) where we use L'Hospital's rule to replace the irregular term on the axis by:

$$\frac{1}{\rho}\partial_\rho \psi|_{\rho\to 0} = \frac{\partial^2 \psi}{\partial \rho^2} \tag{4.75}$$

In addition, to evaluate the finite difference equivalent of the term $\partial^2_\rho \psi$ on the axis, we use the fact that ψ is an even function in ρ due to the symmetry. The detailed description of dealing with such boundaries and the ghost-cell-equivalent implementation of boundary conditions in our finite differencing toolkit (FD) is described in the Appendix.

4.3.3 Boundary Conditions

Axial symmetry demands that the diagonal metric components be even functions in ρ while the off-diagonal term is an odd function:

$$\partial_\rho \tilde{\gamma}_{\rho\rho}(t, \rho, z)|_{(\rho=0,z)} = \partial_\rho \tilde{\gamma}_{zz}(t, \rho, z)|_{(\rho=0,z)} = \partial_\rho \tilde{\gamma}_{\varphi\varphi}(t, \rho, z)|_{(\rho=0,z)} = 0\,, \tag{4.76}$$

$$\tilde{\gamma}_{\rho z}(t, \rho = 0, z) = 0\,. \tag{4.77}$$

The trace-free extrinsic curvature obeys analogous conditions to the conformal 3-metric:

$$\partial_\rho \tilde{A}_{\rho\rho}(t, \rho, z)|_{(\rho=0,z)} = \partial_\rho \tilde{A}_{zz}(t, \rho, z)|_{(\rho=0,z)} = \partial_\rho \tilde{A}_{\varphi\varphi}(t, \rho, z)|_{(\rho=0,z)} = 0\,, \tag{4.78}$$

$$\tilde{A}_{\rho z}(t, \rho = 0, z) = 0\,. \tag{4.79}$$

The extrinsic curvature and ϕ are scalars, and are therefore even functions in ρ (invariant under $\rho \to -\rho$ symmetry):

$$\partial_\rho K(t, \rho, z)|_{(\rho=0,z)} = \partial_\rho \phi(t, \rho, z)|_{(\rho=0,z)} = 0\,. \tag{4.80}$$

The reflection of the ρ components of the conformal connection vector, $\tilde{\Lambda}^\rho$, and the shift vector, β^ρ, under $\rho \to -\rho$ symmetry result in a negative sign. Therefore they are odd functions in ρ:

$$\tilde{\Lambda}^\rho(t, \rho = 0, z) = \beta^\rho(t, \rho = 0, z) = 0\,. \tag{4.81}$$

Their z components, on the other hand, remain unchanged under $\rho \to -\rho$ symmetry, therefore they are even functions:

$$\partial_\rho \tilde{\Lambda}^z(t, \rho, z)|_{(\rho=0,z)} = \partial_\rho \beta^z(t, \rho, z)|_{(\rho=0,z)} = 0\,. \tag{4.82}$$

Forward (one-sided stencil) finite difference approximations of these boundary conditions are used to update the value of the functions on the axis.

Note that for the odd functions described above, beside the value of the function, the second derivative along ρ also vanishes on the axis:

$$\partial^2_\rho \tilde{\Lambda}^\rho(t,\rho,z)|_{(\rho=0,z)} = \partial^2_\rho \beta^\rho(t,\rho,z)|_{(\rho=0,z)} = 0\,. \tag{4.83}$$

$$\partial^2_\rho \tilde{\gamma}_{\rho z}(t,\rho,z)|_{(\rho=0,z)} = \partial^2_\rho \tilde{A}_{\rho z}(t,\rho,z)|_{(\rho=0,z)} = 0\,. \tag{4.84}$$

We impose a forward (one-sided) finite difference equivalent of these conditions to update values at the point next to the axis to improve the smoothness of the functions.

At infinity, the boundary conditions are rather simple, since we are working in a compactified coordinate, and are given by the asymptotically flat values:

$$\tilde{\gamma}_{ij}(t,\rho,z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ at} : (\rho, z) \in \partial\tilde{\Sigma}\,, \tag{4.85}$$

$$\tilde{A}_{ij} = K = \phi = \beta^i = \tilde{\Lambda}^i = 0 \text{ at} : (\rho, z) \in \partial\tilde{\Sigma}\,, \tag{4.86}$$

$$\alpha = 1 \text{ at} : (\rho, z) \in \partial\tilde{\Sigma}\,. \tag{4.87}$$

4.3.4 Evolution Scheme

To evolve the G-BSSN system we implemented a Crank-Nicholson implicit scheme. As discussed in the previous chapter, this implicit scheme appears to be essential to deal with the coordinate singularity $\rho = 0$. There is an extra boundary condition that is implicit in the system (due to the symmetry)—the $\rho \to 0$ limiting behavior of the two components:

$$\tilde{\gamma}_{\rho\rho} = C + O(\rho^2)\,, \tag{4.88}$$

$$\tilde{\gamma}_{\varphi\varphi} = C + O(\rho^2)\,, \tag{4.89}$$

where the constant C is the same for both. Equivalently, their difference should behave as:

$$\tilde{\gamma}_{\rho\rho} - \tilde{\gamma}_{\varphi\varphi} = O(\rho^2)\,, \tag{4.90}$$

and a violation of this condition will manifest itself, for example, in a diverging term in the Ricci component previously shown in (4.35):

$$R_{\rho\rho} = \frac{1}{\tilde{\gamma}_{\varphi\varphi}} \left(\frac{\tilde{\gamma}_{\rho\rho} - \tilde{\gamma}_{\varphi\varphi}}{\rho^2} \right) + \cdots \tag{4.91}$$

Our implementation of a implicit scheme appears to be sufficient to keep the system of equations regular on the axis with no need to explicitly regularize the equations at the analytic level. However, similar to the spherical case of the previous chapter, Kreiss-Oliger dissipation is crucial to suppress numerical noise which can particularly affect the near-origin evolution.

4.3.5 Note on G-BSSN's Additional Constraints

We emphasize that we were only able to achieve stable evolution using the so called Lagrangian choice (4.10) for the evolution of the determinant of the conformal metric. In this choice, the determinant of the conformal metric is given by:

$$\tilde{\gamma} = \rho^2(\tilde{\gamma}_{\rho\rho}\tilde{\gamma}_{zz} - \tilde{\gamma}_{\rho z}^2)\tilde{\gamma}_{\varphi\varphi} = \begin{cases} \rho^2 & \text{for Teukolsky-type} \\ \rho^2 e^{2V(\rho,z)} & \text{for Brill data} \end{cases} \tag{4.92}$$

and is equal to its initial time value as given above for both the Teukolsky-type and Brill data. This constraint can be monitored or can be imposed to fix one of the components of the $\tilde{\gamma}_{ij}$ metric. Our numerical experiments suggest that imposing this equation improves the performance of the code. In particular, we have applied this constraint to compute the $\tilde{\gamma}_{zz}$ component of the 3-conformal-metric. Similarly, the tensor $\tilde{A}_{ij}$ is expected to remain traceless during the evolution—which we have monitored as an accuracy gauge. However, this equation can also be imposed to improve the accuracy, and we use it to fix the same component as the component fixed in 3-conformal metric, i.e. $\tilde{A}_{zz}$.

4.3.6 Tests: Convergence of Primary Variables

As discussed in previous chapters, the first diagnostic test is to perform a convergence test. This test validates the consistency of the finite difference approximation and determines whether the finite difference numerical solution is converging to an underlying continuum function well-resolved in the discretized mesh. The convergence factor,

$$Q(t;q) = \frac{||q^h(t,X) - q^{h/2}(t,X)||_2}{||q^{h/2}(t,X) - q^{h/4}(t,X)||_2}\,, \tag{4.93}$$

has been plotted for three of the primary variables of the G-BSSN equations, $\tilde{g}_{rr}, K, \tilde{\Lambda}^z$ and the RHS of the evolution equation for $\tilde{A}_{\rho z}$ for a strongly gravitating evolution in Fig. 4.3. As can be seen from this plot, all variables exhibit at least first order convergence and some second order convergence—values of $\log_2(Q)$ between 2 and 1. We particularly chose to measure the convergence factor for the RHS of $\partial_t \tilde{A}_{\rho z}$, since it is one of the computationally most complex expressions to discretize and evaluate, and contains several “irregular”, $1/\rho$ and $1/\rho^2$ terms, which can potentially be sources of further numerical error. This function appears to have the smallest convergence factor, yet it is at least first order convergent and at best second order convergent as demonstrated in the figure. Note that the convergence curves are plotted for three resolutions: 128×256, 256×512 and 512×1024 which are not particularly high resolutions and one might expect to observe better convergence at higher resolutions. We note that we performed a similar test for all of the other primary G-BSSN variables, and they are all about second order convergent.

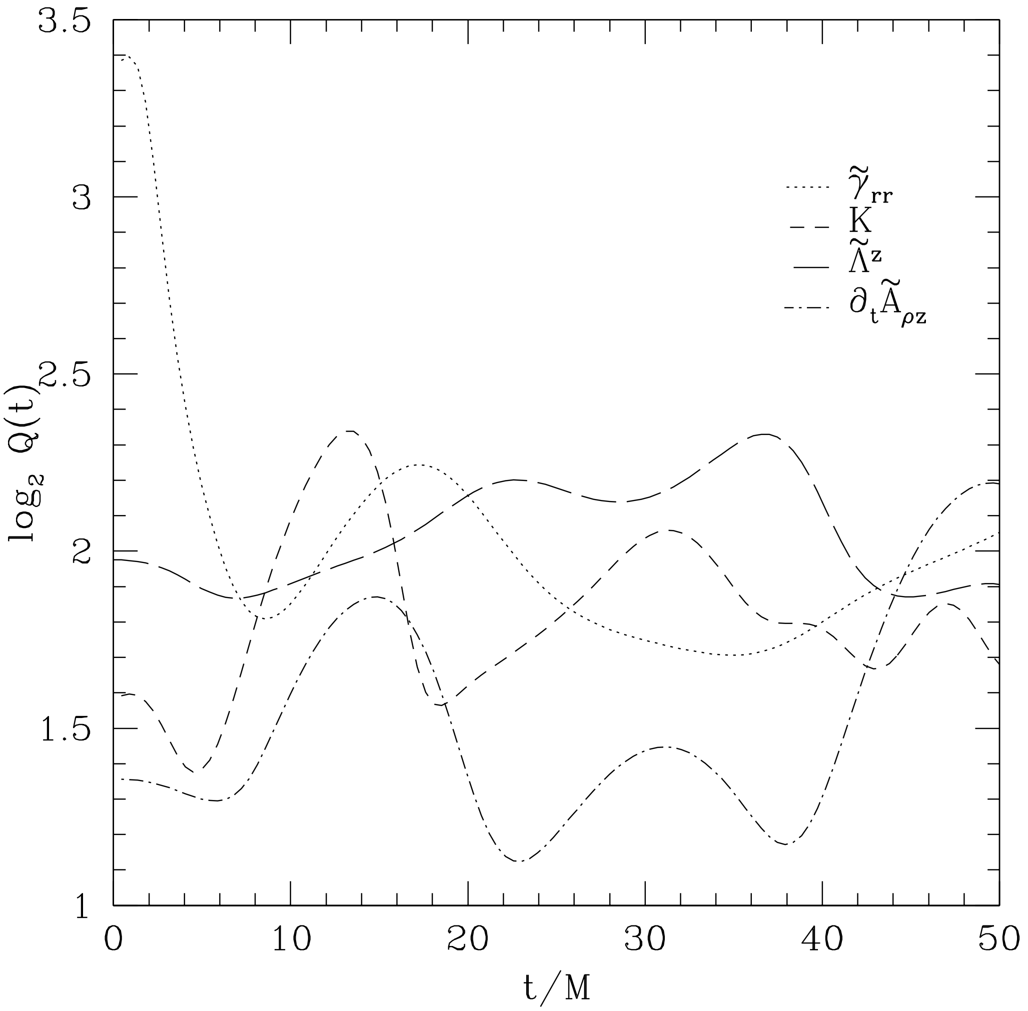


Figure 4.3: Convergence factor for the G-BSSN variables: $\tilde{g}_{rr}, K$ and $\tilde{\Lambda}^z$. The convergence factor (4.93) for these functions is plotted during the evolution of a strong pure gravity wave. As presented in the plot, they all converge in a second order to first order fashion corresponding to the values 2 and 1 in $\log_2(Q)$ respectively. The last curve is the convergence factor for the RHS of the evolution equation for $\tilde{A}_{\rho z}$ which is a rather complicated function of 3-metric and the rest of the G-BSSN variables and contains several "irregular" terms (containing $1/\rho$ and $1/\rho^2$). Again, we observe at least first order convergence for this function.

4.3.7 Tests: Conservation of Constraints During Evolution

We also monitor the momentum constraint (4.18), which has only ρ and z components, and the Hamiltonian constraint (4.17). Since the G-BSSN formulation is a free evolution system, these constraints are not imposed during the evolution of the geometry and can be used as an effective diagnostic tool to gauge the accuracy of the numerical time integration of the equations.

Fig 4.4(a) and Fig 4.4(b) demonstrate the second order convergence to zero of the two components of the momentum constraint as the resolution improves. Results for the Hamiltonian constraint are presented in Fig 4.4(c) for 4 consecutively decreasing resolutions and similarly show second order convergence to zero. Furthermore, since the evaluator of the Hamiltonian constraint in this plot uses a different form of the equation and a different finite differencing scheme, the convergence of the its value at $t = 0$ is also an independent test for the multi-grid initial value solver that we have used to solve the Hamiltonian constraint at the initial time.

Beside the Hamiltonian and momentum constraints, here we also present another constraint that naturally arises from the definition of the conformal connection $\tilde{\Lambda}^i$ (4.13). Since $\tilde{\Lambda}^i$ is evolved via a time dependent equation (4.8), the difference between its evolved value and its computed value, $\tilde{\Lambda}^k_c$ from the conformal metric,

$$\tilde{\Lambda}^k_c = \tilde{\gamma}^{ij}\left(\tilde{\Gamma}^k_{ij} - \mathring{\Gamma}^k_{ij}\right) , \tag{4.94}$$

forms a residual: $\tilde{\Lambda}^i - \tilde{\Lambda}^i_c$, that should converge to zero as $h \to 0$. Fig 4.4(d) presents the value of the ρ component of this residual, $||\tilde{\Lambda}^\rho - \tilde{\Lambda}^\rho_c||_{l2}$, in a $\log_2$ scale, where a factor of 2 decrease for the resolution refinement $h \to h/2$ signals a second order convergence to zero. Such behaviour is clearly present in the plot.

Note that we also monitored all of the other constraints that occur in the G-BSSN formulation, including: the trace of $\tilde{A}_{ij}$, which by definition should remain zero and the determinant of the conformal 3-metric, $\tilde{\gamma}$, which should stay equal to

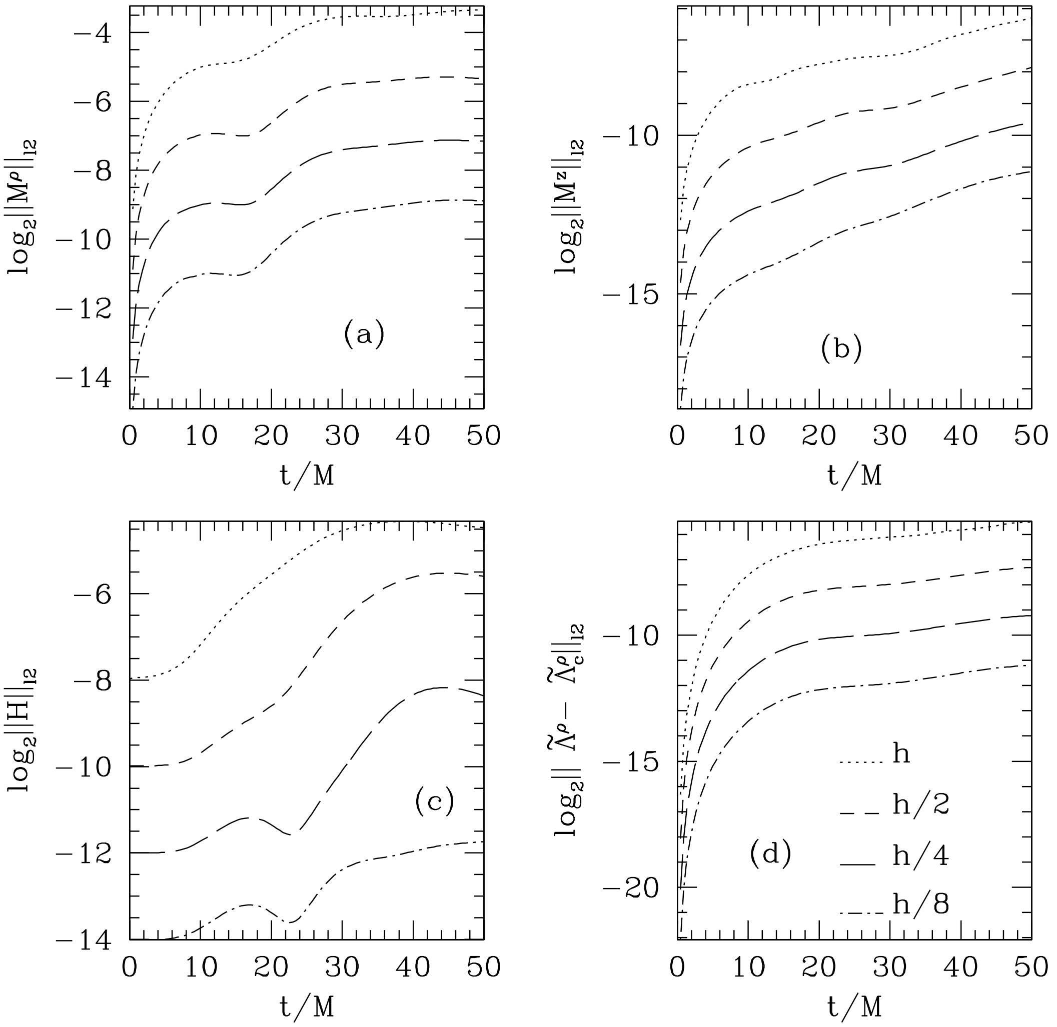


Figure 4.4: Evolution of the conserved variables: (a): The time evolution of the norm of the ρ component of the momentum constraint (4.18), M^ρ, for 4 different resolutions $h, h/2, h/4, h/8$, plotted in a $\log_2$ scale. Each step of resolution improvement by a factor of 2 results in the decrease of the value of the function by a factor of about 4, (2 in the scale of the plot) which demonstrates second order convergence to zero. (b): Similar to (a) for the z component of the momentum constraint. (c): The norm of the Hamiltonian constraint (4.17) at 4 different resolutions similar to (a). Again, we observe second order convergence to zero. The convergence of the value of the curve at $t = 0$ is an independent test for the validity of the initial value solver implemented to solve the Hamiltonian constraint. (d): The norm of the ρ component of the residual, $\tilde{\Lambda}^i - \tilde{\Lambda}^i_c$ which exhibits second order convergence to zero as the resolution improves. This suggests that the free evolution of the conformal connection, $\tilde{\Lambda}^i$ via (4.8) in G-BSSN formulation is consistent with its definition in (4.13).

its value at initial time for the Lagrangian choice (4.10). Of course, these tests are only meaningful if we do not enforce these constraints during the test runs. As mentioned before, after the test runs, we do impose these constraints during the evolution of the results presented below.

4.3.8 Tests: Direct Validation via Einstein's Equations

Finally, the most robust test we developed involves a direct evaluation of the components of the Einstein's equations for a given 4-metric, $g_{\mu\nu}$, reconstructed from the 3-metric, γ_{ij}, and the coordinate variables:

$$g_{\mu\nu} = \begin{pmatrix} -\alpha^2 + \beta_l \beta^l & \beta_i \\ \beta_j & \gamma_{ij} \end{pmatrix} \tag{4.95}$$

where the 3-metric and coordinate functions are given by the solutions the G-BSSN solver produces. We define the residual for the Einstein's equation:

$$E^{\mu}_{\ \nu} \equiv G^{\mu}_{\ \nu} - 8\pi T^{\mu}_{\ \nu} \,. \tag{4.96}$$

For the case of pure gravity waves, since the energy-momentum and the Ricci scalar are zero[37], the residual can be defined equal to the Ricci tensor:

$$E^{\mu}_{\ \nu} \equiv R^{\mu}_{\ \nu} \,, \tag{4.97}$$

and is zero if and only if the Einstein's equation is solved correctly in vacuum.

The norm of various components of the residual (4.97) are plotted in Fig. 4.5, and the plots suggest that the residual converges to zero in a second order fashion, providing strong evidence that the computed metric does indeed satisfy the Einstein's equations. We also note that this is not only a validation of the correctness

[37] Taking the trace of the Einstein's equation: $R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} = 8\pi T_{\mu\nu}$ we get: $-R = 8\pi T$ where T is the trace of the energy-momentum tensor. Therefore, in vacuum, $T_{\mu\nu} = 0$, the Ricci scalar is zero.

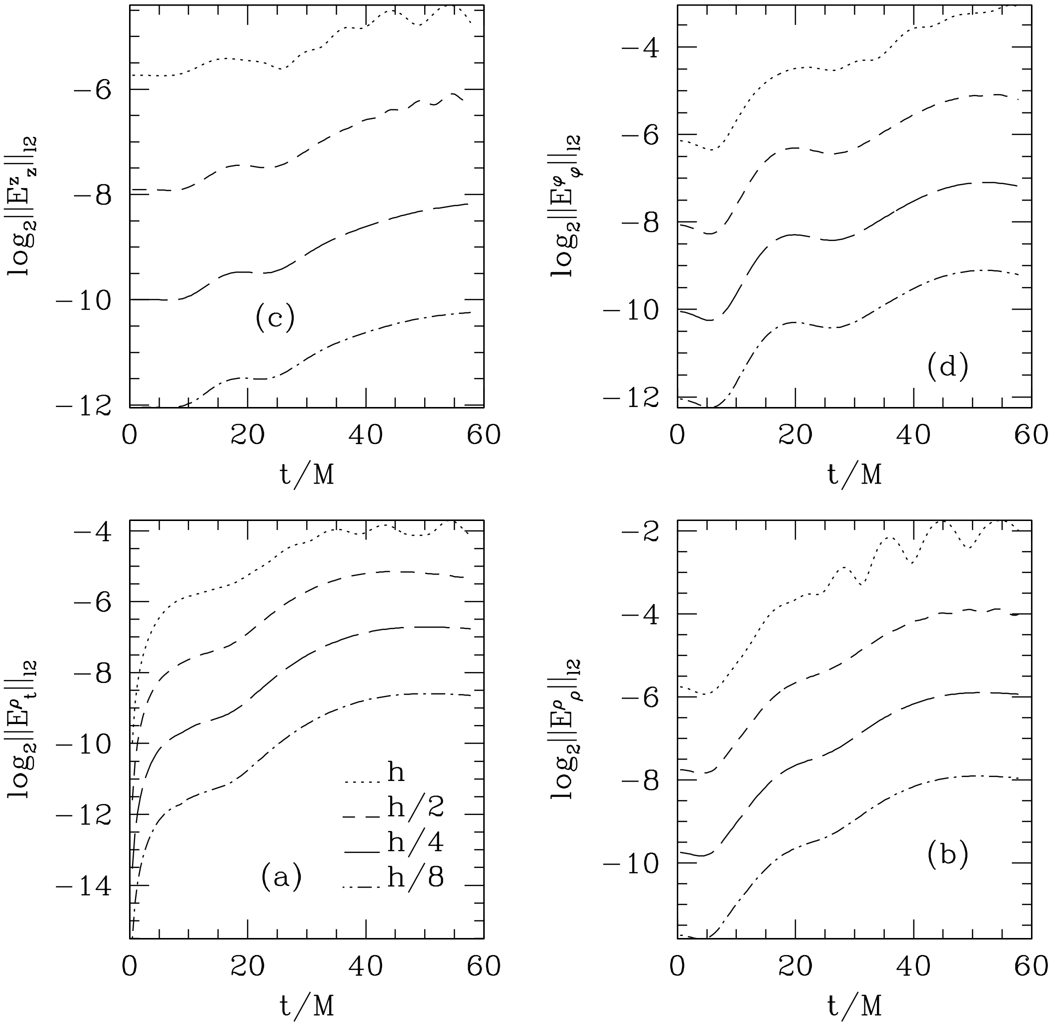


Figure 4.5: Convergence of the Einstein equations residuals defined in (4.97) for a strong gravity evolution. The curves present 4 different consecutively improving (by factor of 2) mesh sizes, and various components of the residual are plotted here. As it is clear on the $\log_2$ scale of the graph, the value of the residual decreases by a factor of about 4 at each step of resolution refinement. This convergence to zero in second order fashion suggests that the numerical solver that provides the 4-metric to the independent residual evaluator code is indeed computing a metric that satisfies the Einstein's equation. This direct validation further suggests that the set of equations (G-BSSN) used in the numerical solver are equivalent to the Einstein's equations.

of our numerical solver, but also proves that the derivation of G-BSSN equations was done correctly, and that the equations are equivalent to Einstein's equations. We note that we developed the testing facilities independently of the numerical solver. In particular, we derived the residual (4.97) using a different tensor manipulation package than the one used to derive the G-BSSN equations. We also used a different finite difference scheme than the one adopted to discretize the G-BSSN evolution equations.

4.4 Results

In this section we provide examples of the performance of the code for evolving pure gravity initial data that is set to be in the non-linear regime, slightly above or below the threshold of black hole formation. We note that the code is developed for a generic axisymmetric system *with* matter content and can be used with other numerical solvers to evolve the coupled matter-gravity systems.

4.4.1 Evolution of Teukolsky-type Initial Data

First we present a typical evolution of the Teukolsky-type initial data using the seed function F (4.62,4.64) that creates the initial wave packet plotted in Fig. 4.2. The zz component of the conformal metric $\tilde{\gamma}_{ij}$ is plotted in Fig. 4.6. The amplitude is set to about 10% less than the critical value. Even though the waves eventually completely disperse, this is still a strong field evolution. As can be seen from the plots, the system exhibits an intermediate confined state where the gravitational wave packet is held together by its own gravity. The system at the initial time is only slightly away from the linear regime—the amplitude of the conformal metric differs from 1 by only ≈ 0.05—however as the wave focuses toward the origin, the self-gravitation amplifies and the system exhibits non-linear dynamics.

An evolution of a similar configuration but where the amplitude is now about 10% higher than the critical value is presented in Fig. 4.7. Here, the wave packet gets trapped by its own self gravitation and collapses to form a black hole. As demonstrated in the last snapshots of the evolution in Fig. 4.7 the conformal metric shows large deviations from the flat metric. The wave packet is confined in the vicinity of the origin where the lapse function starts to collapse as a black hole forms. Comparing Fig. 4.7 and Fig. 4.6 one can observe the two distinct end states of the evolution of the system: black hole and dispersal.

Another initialization that we implemented uses the seed function G, which creates a wave packet with 2 extrema and a simpler overall structure. Fig. 4.8 and 4.9 demonstrate the two distinct end state of the evolution, dispersal and black hole formation. Again, the dispersal data is set to a non-linear regime and is close to the critical value, therefore as the wave packet experiences a non-linear evolution near the origin, the reflected wave develops a different wave front structure as shown in the last snapshot in Fig. 4.8.

Finally, in Fig. 4.10 we plot the central value of the lapse function for the collapsing and dispersing Teukolsky-type wave shown in Fig. 4.8 and 4.9. The two distinct end states are apparent from the fact that the dispersal data has an intermediate time where the lapse function decreases to small value, while for the large initial amplitude, the lapse collapses and a black hole forms.

4.4.2 Evolution of Brill Initial Data

In this section, we demonstrate some of the simulations that we performed using Brill initial data. As discussed, one important difference between Teukolsky-type and Brill initial data is in the trace of the conformal metric in the linearized regime where a Teukolsky-type wave packet satisfies the traceless condition, while the Brill data does not. We have experimented with the Gaussian initialization (4.61), that is commonly used for Brill initial data in the literature, but for comparison here we demonstrate the simulations of Brill initial data that are initialized using a seed function G similar to Fig. 4.8-4.9.

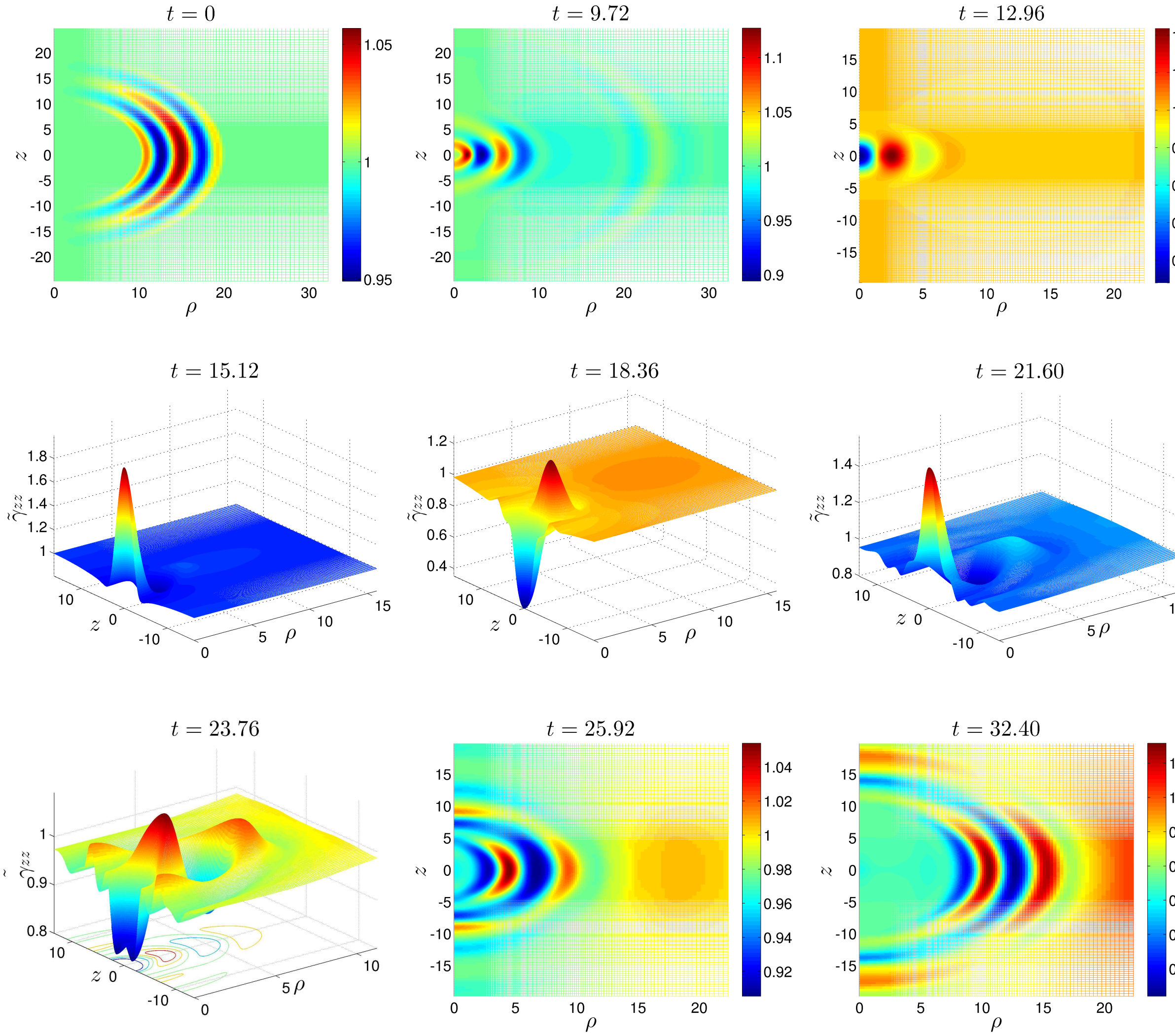


Figure 4.6: Evolution of a non-linear Teukolsky-type wave packet: Here the gravitational wave is initialized by the seed function F, and as can be seen from the snapshot $t = 0$, has 5 extrema. All of the snapshots plot the $\tilde{\gamma}_{zz}$ component of the conformal metric. The initial data is time symmetric, therefore part of the wave packet moves inward while the other part disperses toward infinity as seen in th thee $t = 9.72$ snapshot. During the intermediate time, $12.96 < t < 21.60$, the wave packet forms a gravitational *geon* confined to the vicinity of the origin. Eventually, the wave starts to disperse to infinity at $t > 23.76$. The system is in a strong gravity regime and as can be seen from the last snapshot at $t = 32.40$ the non-linear effects changes the outgoing shape of the wave packet relative to $t = 0$. This calculation, as well as the other simulations presented in this chapter, are performed on a non-uniform grid with resolution 512×1024. Here, we have transformed the functions back to the spatial (ρ, z) coordinates for demonstration purposes.

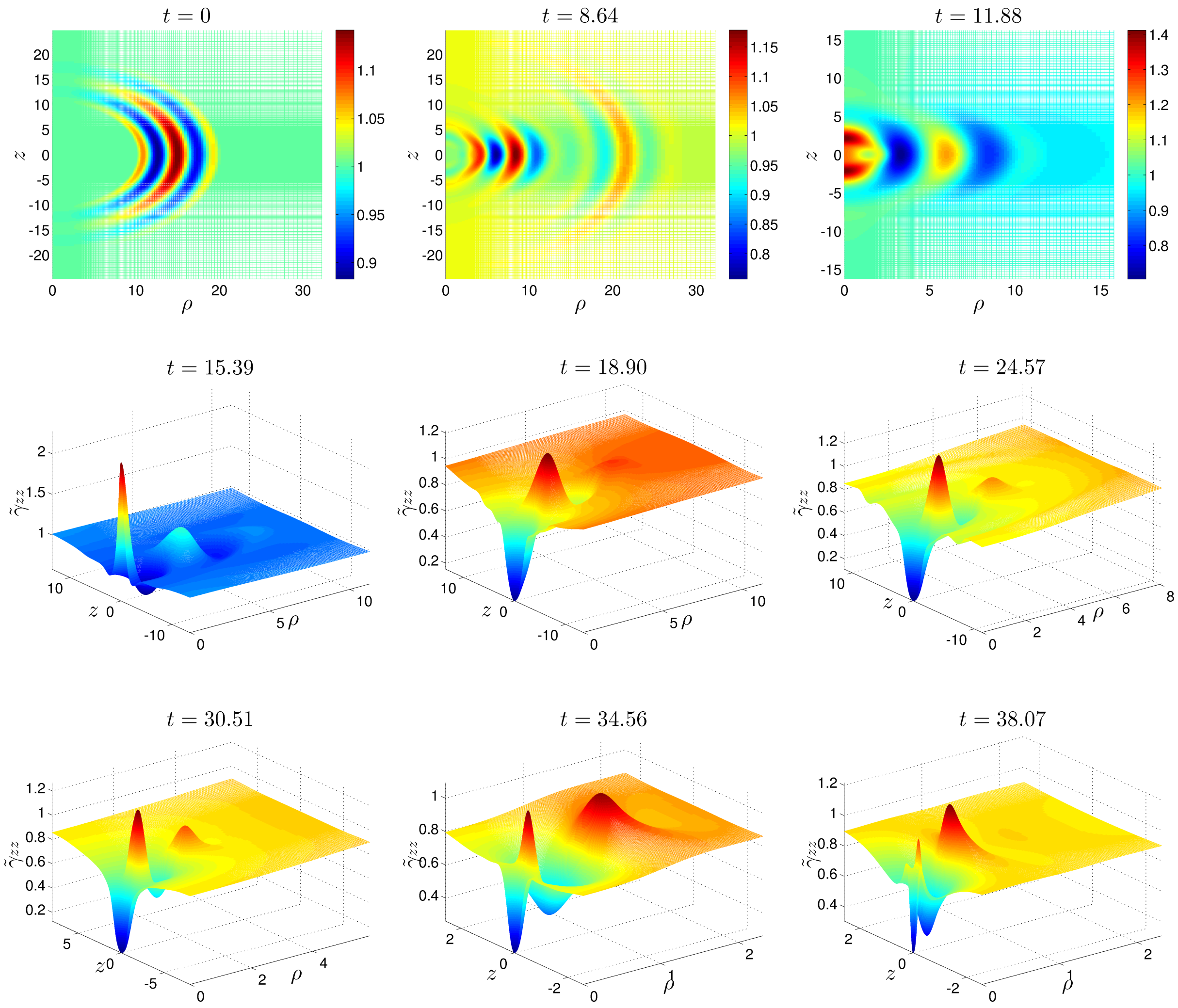


Figure 4.7: Collapse of Teukolsky-type wave initial data: Here the amplitude of the initial pulse is large enough to cause the system to collapse. The wave packet is initialized using the seed function F and similar to Fig. 4.6 we are plotting the $\tilde{\gamma}_{zz}$ component of the conformal metric. As one can observe, in the later snapshots, the pulses accumulate and become confined within a very small region close to the origin and eventually system collapses to a black hole. The simulations take about 100 hours using 32 CPUs. The non-uniform structure of the numerical grid is observable in the first snapshot.

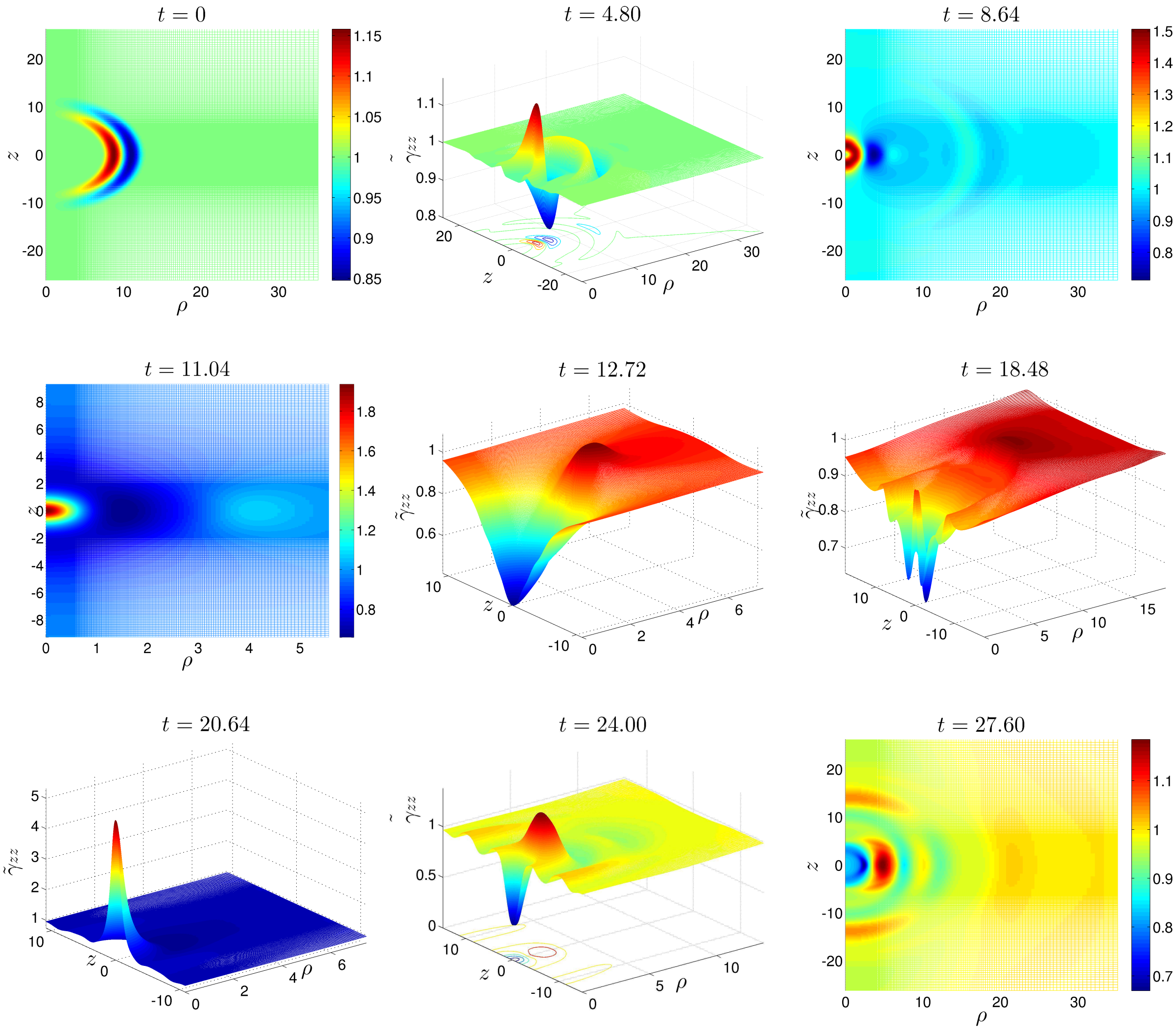


Figure 4.8: Non-linear evolution of a Teukolsky-type wave packet: Various snapshots of the zz component of the conformal metric for a Teukolsky-type data initialized using the seed function G (4.63) are shown. Compared to Fig. 4.6, the wave packet has a simpler structure. Again, the time symmetric initial data has both an ingoing and an outgoing part (seen at $t = 4.80$ snapshot). After an intermediate non-linear phase where the metric components deviate strongly from 1, the wave packet disperses as seen in $t = 27.60$, where the outgoing form of the wave is somewhat different than the initial time. The central value of the lapse function for this evolution is shown in Fig. 4.10.

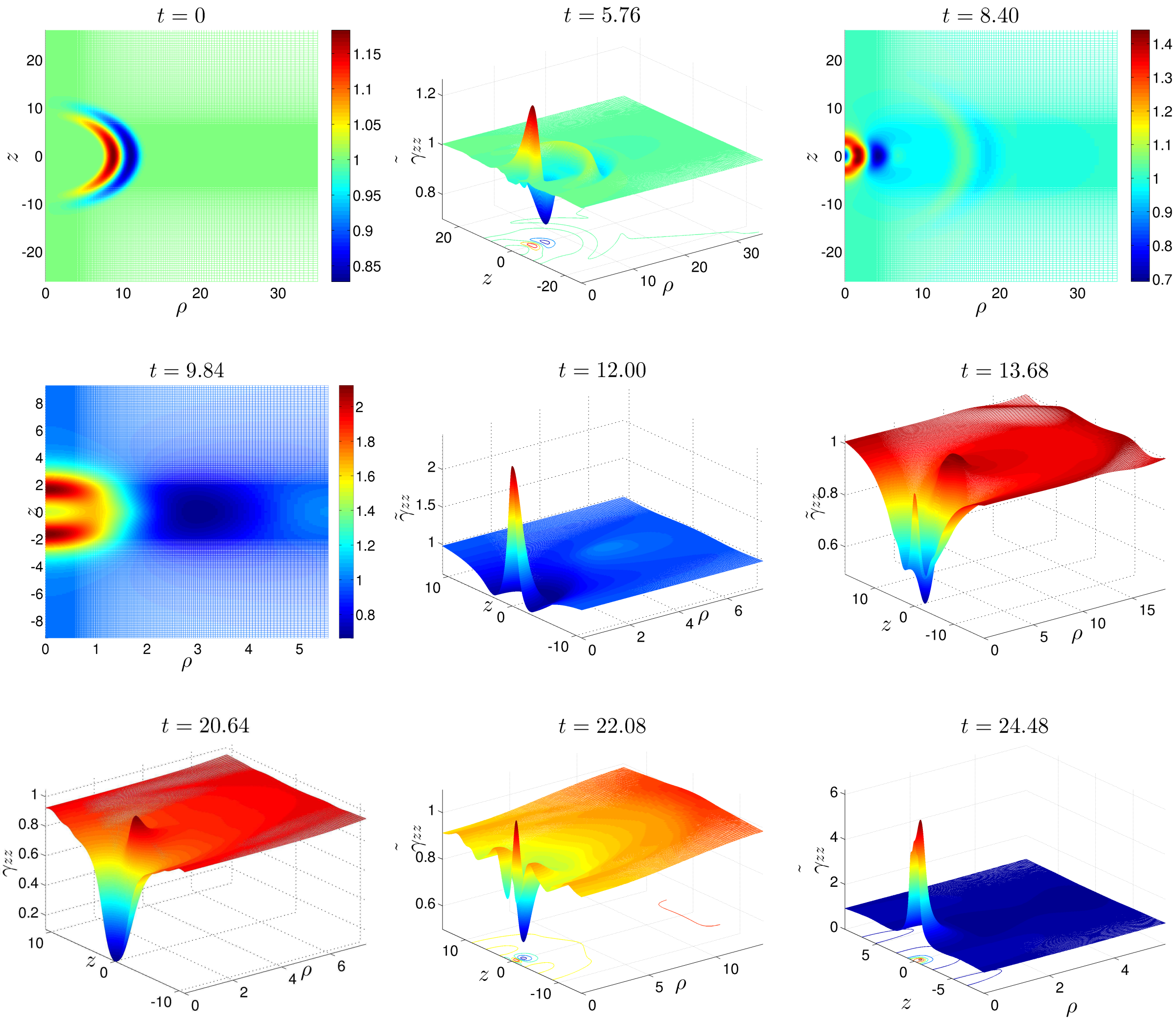


Figure 4.9: Typical evolution of collapsing Teukolsky-type data using the seed function G. Here, similar to Fig. 4.7, we the $\tilde{\gamma}_{zz}$ component of the conformal metric. The wave packet focuses toward the origin and eventually the system collapses to a black hole. The collapse of the lapse function (singularity avoidance property) at the vicinity of the origin slows down the evolution. The central value of the lapse function for this evolution is plotted in Fig. 4.10.

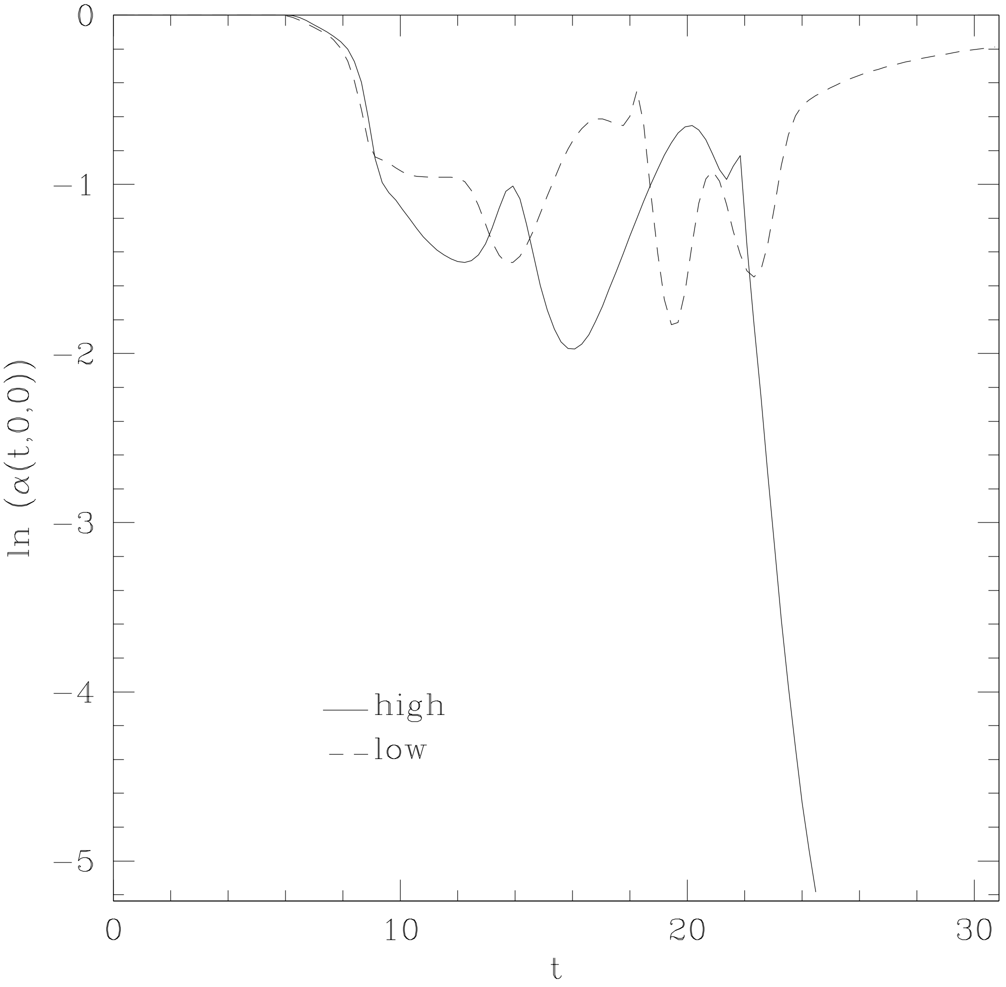


Figure 4.10: Time evolution of central lapse: The logarithm of the lapse function at the origin $\alpha(t, \rho = 0, z = 0)$ is plotted versus the time coordinate. Here "high" denotes the collapse scenario and "low" is the dispersal case that are illustrated in Fig. 4.9 and 4.8 respectively. As one can see, the central value of the lapse for dispersal data exhibits an intermediate time with significantly large central red-shift, but eventually rises back to 1 as the wave packet leaves the origin. However, in a collapse scenario, the central lapse collapses, as shown by the solid curve.

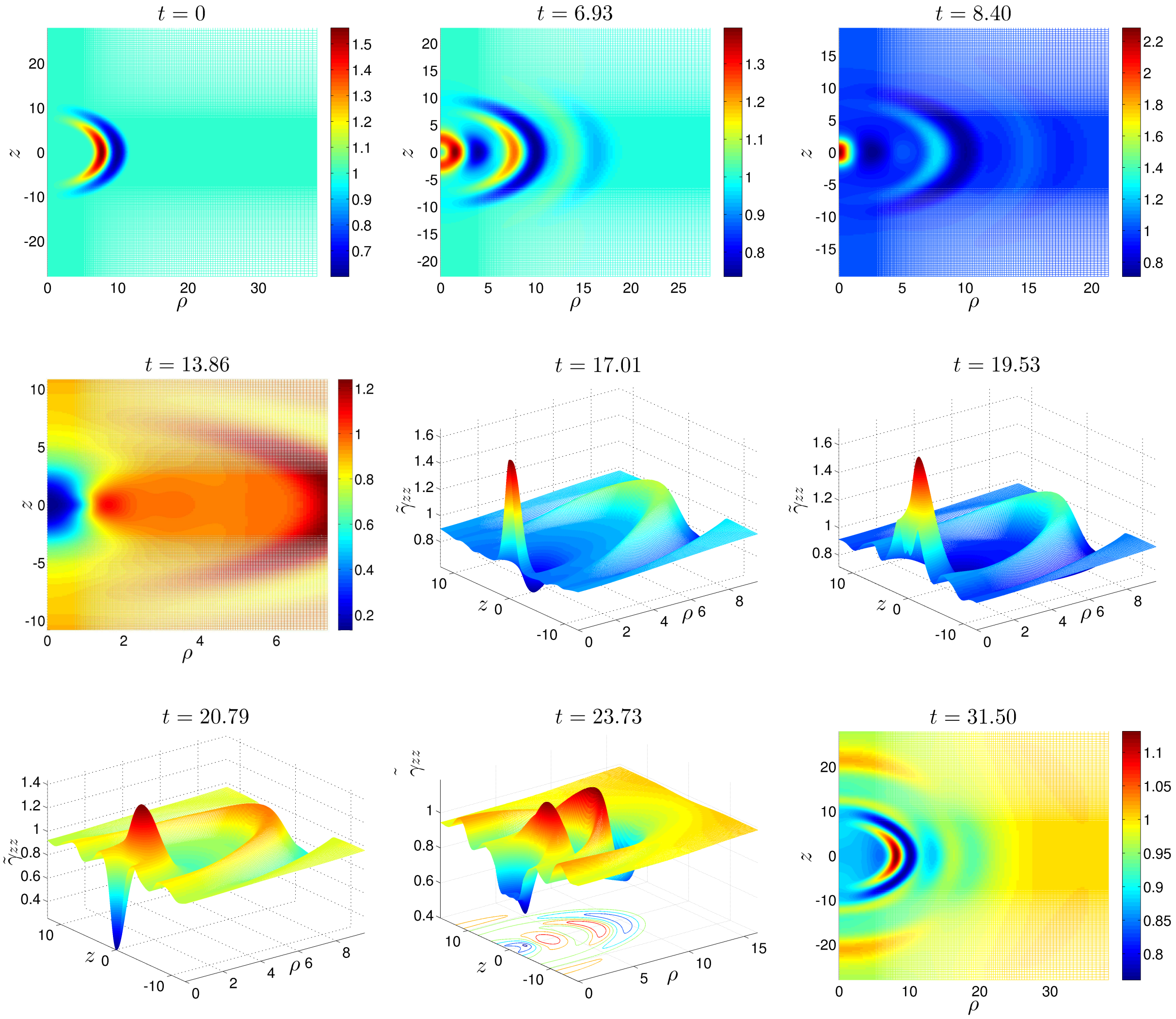


Figure 4.11: Dispersal evolution of Brill initial data: Plotted is the $\tilde{\gamma}_{zz}$ component of the conformal metric at different times during a dispersal Brill data initialized using the seed function G, (4.63,4.65). Note the large difference between the inital amplitude here and the initial amplitude of Teukolsky data in Fig. 4.8. As is apparent in the second snapshot ($t = 6.93$) the Brill "wave" has pure gauge content, that does not propagate. The ingoing part of the wave in the second snapshot ($t = 6.93$) is located at $\rho \approx 2.5$ while the outgoing part is at $\rho = 13$, however in between there is a third pulse at $\rho = 8$ (same radius as the initial data) that does not propagate. The ingoing wave eventually reflects back from the center and in snapshots $t = 20.79$ and $t = 23.73$ starts to move outward. Note that in most of the diagrams the gauge content is clearly observable and is located at about $\rho \approx 8$ without moving. In fact, in the last snapshot the reflected wave has passed through the gauge pulse and is located at $\rho \approx 15$.

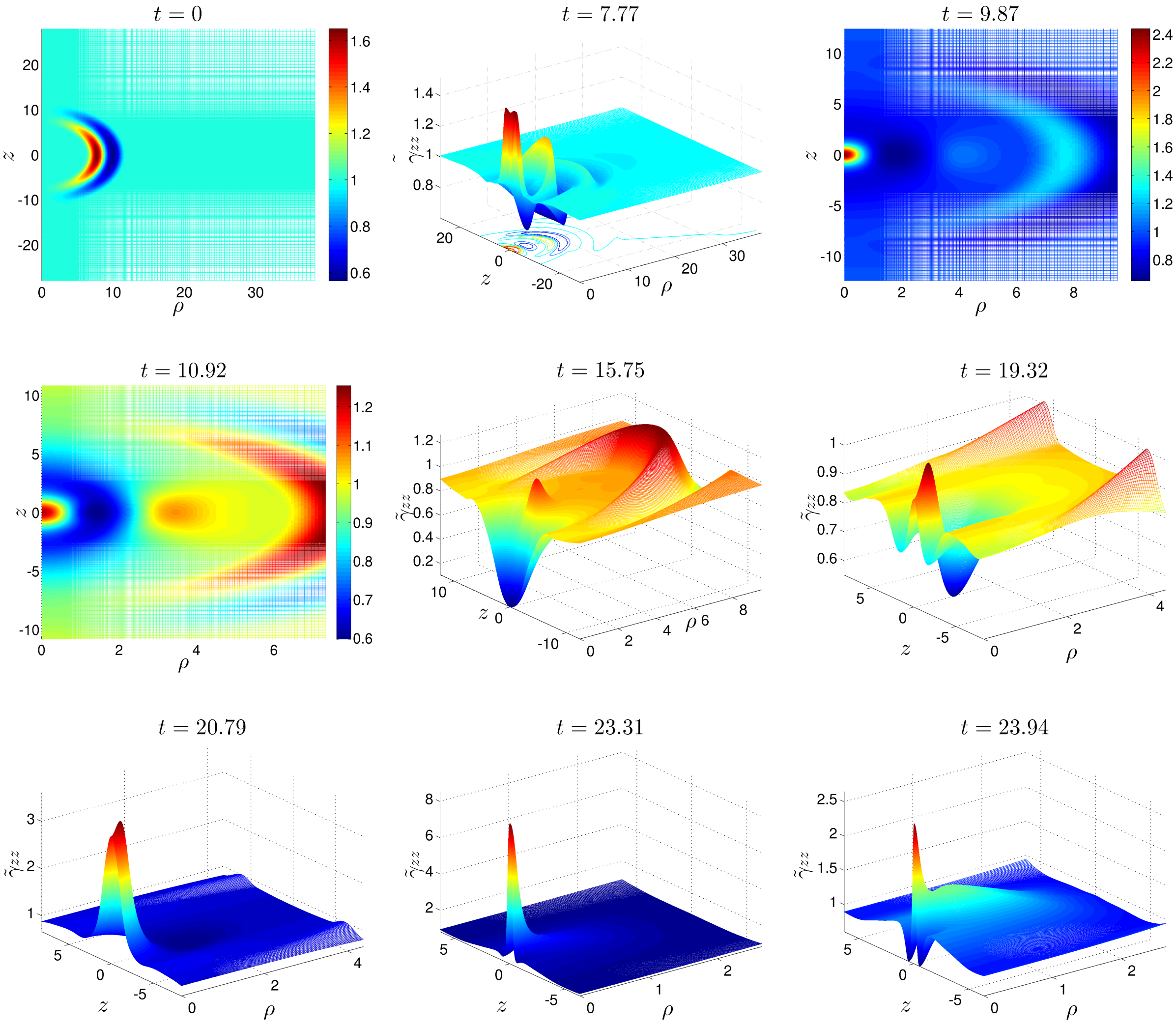


Figure 4.12: Collapse of Brill data: Evolution of strong initial data that eventually collapse to a black hole. Again, similar to Fig. 4.11, the wave packet contains a pure gauge component, which does not propagate. The ingoing pulse eventually focuses to a very small region and collapses to a black hole. The central value of the lapse for this evolution is given in Fig. 4.13.

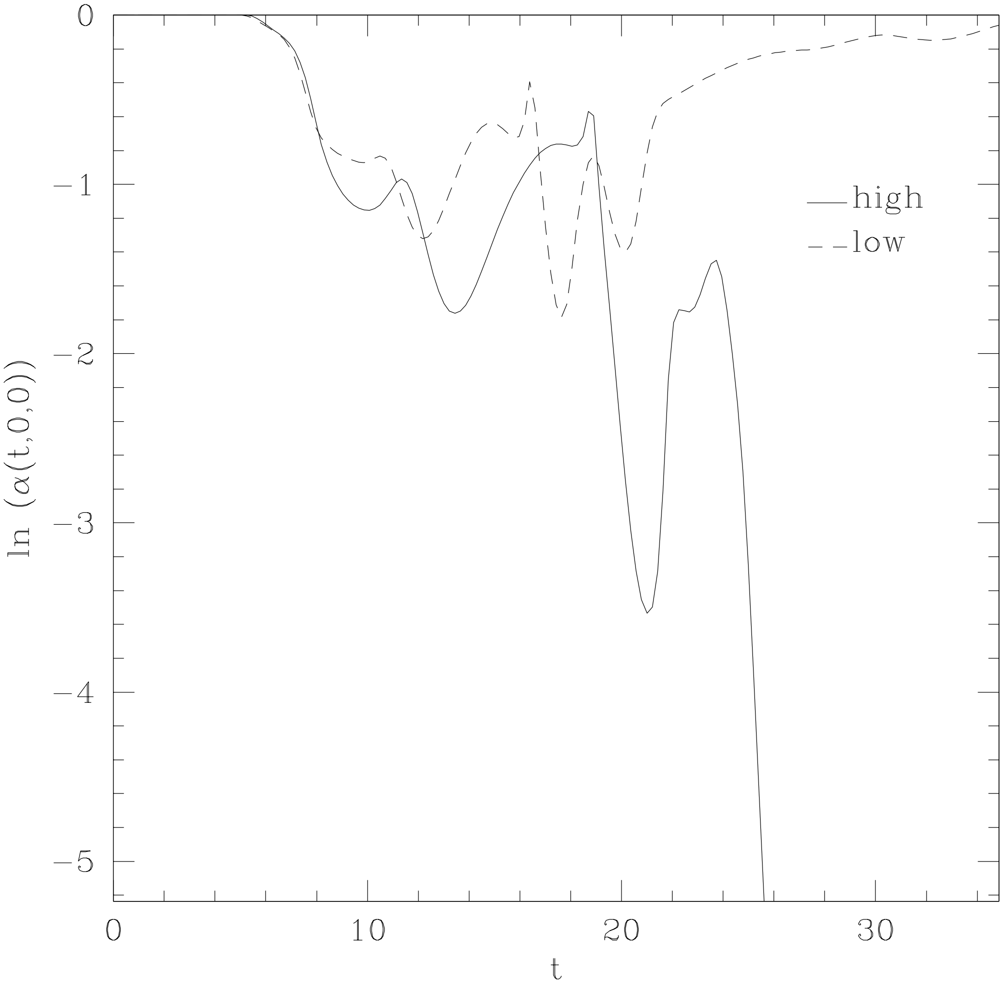


Figure 4.13: Central lapse for the Brill data evolution: $\alpha(t, \rho = 0, z = 0)$ as a function of time in a logarithmic scale. Similar to the Teukolsky-type data (Fig. 4.10) the dispersal case exhibits an intermediate high red-shift while eventually rising back to the unity, while during a collapse process the lapse function at the center collapses and slows down the evolution which is expected from the singularity avoidance property of the 1+log slicing.

The dispersal scenario is shown in Fig. 4.11 where the initial wave packet has a simpler shape to Teukolsky-type evolution shown in Fig. 4.8. However, first note the large difference in the amplitude in the first snapshots of the two figures. Secondly, as described in the caption of Fig. 4.11 the wave packet has a pure gauge content that simply remains at a fixed point as the radiative part of the wave moves inward and outward. This evolution eventually disperses and the gravitational radiation propagates toward infinity. However, the pure gauge content remains in the numerical domain and the final state of the conformal metric is not unity. A collapse scenario is plotted in Fig. 4.12. As one can see, similar to the dispersal case, the wave packet contains a pure gauge component that remains at a fixed radius. The in-going part of the time symmetric data eventually collapses and a black hole forms.

For both dispersal and collapse cases, the central value of the lapse is plotted in Fig. 4.13. As seen from the dashed curve, the dispersal data experiences a large central red-shift while eventually the wave packet disperses. However, the collapse is distinct and the lapse function collapses as the black hole forms. One can compare this to Fig. 4.10.

4.5 Further Remarks and Conclusion

In this chapter, we described a new G-BSSN axisymmetric code. The code is implemented such that it can be coupled to arbitrary matter content. However, we measured the performance of the code in the most challenging scenario: pure nonlinear gravity waves evolution. We demonstrated both Teukolsky-type and Brill initial data evolution and discussed their difference. In particular, the Teukolsky-type wave packet appears to be a better choice since it mimics the linearized regime wave-type propagation better than Brill initial data. Our results suggest that G-BSSN is a promising formulation to evolve pure gravity waves and further extension of this work can shed more insight into the expected type II critical solution in pure gravity waves.

However, there are several other steps required to bring the code to production. Specifically, resolution higher than 512×1024 is needed to capture discrete self similarity. This can be achieved by further optimizing the code, as well as by using larger number of CPUs. At the moment, our numerical Hamiltonian constraint solver for the initialization is rather slow and has limited our work to this resolution. In addition, the number of grid functions can be decreased by carefully examining the dependencies of fields which in turn will reduce the CPU communication time and improve the scaling of the code to higher number of CPUs and higher resolution.

Chapter 5

Conclusion

In this thesis we presented three projects in critical phenomena studies in gravitational collapse. Chapter 2 focused on the Einstein-Vlasov system and the observation of type I behaviour and weak universality in the system. Chapter 3 demonstrated a possibility for using G-BSSN formulation in type II critical phenomena studies where, for the first time, we presented an implementation of a fully evolutionary system that can successfully derive the spacetime evolution close to the critical regime and find a type II threshold solution. Chapter 4 described the G-BSSN formulation in axial symmetry and outlined the new numerical code we developed and used to evolve pure gravity wave content. The primary results seem very promising to be extended to find the expected, but as yet unresolved, type II critical behaviour in pure gravity wave collapse.

Bibliography

[1] Arman Akbarian and Matthew W. Choptuik. Critical collapse in the spherically-symmetric Einstein-Vlasov model. *Phys. Rev.*, D90(10):104023, 2014.

[2] Arman Akbarian and Matthew W. Choptuik. Black hole critical behavior with the generalized bssn formulation. *Phys. Rev. D*, 92:084037, Oct 2015.

[3] M.W. Choptuik, L. Lehner, and F. Pretorius. Probing strong-field gravity through numerical simulations. In A. Ashtekar, B.K. Berger, J. Isenberg, and M. MacCallum, editors, *General Relativity and Gravitation: A Centennial Perspective*. Cambridge, Cambridge University Press, (2015).

[4] Matthew W. Choptuik. Universality and scaling in gravitational collapse of a massless scalar field. *Phys. Rev. Lett.*, 70:9–12, 1993.

[5] Carsten Gundlach and Jose M. Martin-Garcia. Critical phenomena in gravitational collapse. *Living Rev. Relativ.*, 10(5), 2007.

[6] Carsten Gundlach. Critical phenomena in gravitational collapse. *Phys. Rept.*, 376:339–405, 2003.

[7] Emanuele Berti, Vitor Cardoso, and Clifford M. Will. On gravitational-wave spectroscopy of massive black holes with the space interferometer LISA. *Phys. Rev.*, D73:064030, 2006.

[8] A. Giazotto. The Virgo Project: A Wide Band Antenna for Gravitational Wave Detection. *Nucl. Instrum. Meth.*, A289:518–525, 1990.

[9] B. P. Abbott et al. LIGO: The Laser interferometer gravitational-wave observatory. *Rept. Prog. Phys.*, 72:076901, 2009.

[10] Gregory M. Harry. Advanced LIGO: The next generation of gravitational wave detectors. *Class. Quant. Grav.*, 27:084006, 2010.

[11] B. Willke et al. The GEO 600 gravitational wave detector. *Class. Quant. Grav.*, 19:1377–1387, 2002.

[12] Frans Pretorius. Evolution of binary black hole spacetimes. *Phys.Rev.Lett.*, 95:121101, 2005.

[13] Manuela Campanelli, C.O. Lousto, P. Marronetti, and Y. Zlochower. Accurate evolutions of orbiting black-hole binaries without excision. *Phys.Rev.Lett.*, 96:111101, 2006.

[14] John G. Baker, Joan Centrella, Dae-Il Choi, Michael Koppitz, and James van Meter. Gravitational wave extraction from an inspiraling configuration of merging black holes. *Phys. Rev. Lett.*, 96:111102, 2006.

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

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

[17] Steven Weinberg. *Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity.* New York,Wiley, 1972.

[18] Sean M. Carroll. Lecture notes on general relativity. 1997.

[19] John Kormendy and Douglas Richstone. Inward bound: The Search for supermassive black holes in galactic nuclei. *Ann. Rev. Astron. Astrophys.*, 33:581, 1995.

[20] John Kormendy and Karl Gebhardt. Supermassive black holes in nuclei of galaxies. *AIP Conf. Proc.*, 586:363–381, 2001.

[21] D. Arnett. *Supernovae and nucleosynthesis: An investigation of the history of matter, from the big bang to the present.* 1996.

[22] R. Penrose. Gravitational collapse: The role of general relativity. *Riv. Nuovo Cim.*, 1:252–276, 1969. [Gen. Rel. Grav.34,1141(2002)].

[23] Carsten Gundlach. Critical phenomena in gravitational collapse. *Adv. Theor. Math. Phys*, 2:1–49, 1998.

[24] Charles R. Evans and Jason S. Coleman. Observation of critical phenomena and selfsimilarity in the gravitational collapse of radiation fluid. *Phys. Rev. Lett.*, 72:1782–1785, 1994.

[25] Tatsuhiko Koike, Takashi Hara, and Satoshi Adachi. Critical behavior in gravitational collapse of radiation fluid: A Renormalization group (linear perturbation) analysis. *Phys. Rev. Lett.*, 74:5170–5173, 1995.

[26] Dieter Maison. Nonuniversality of critical behavior in spherically symmetric gravitational collapse. *Phys. Lett.*, B366:82–84, 1996.

[27] R. Arnowitt, S. Deser, and C. W. Misner. The dynamics of general relativity. In L. Witten, editor, *Gravitation: An Introduction to Current Research.* New York, Wiley, (1962).

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

[29] Eric Gourgoulhon. 3+1 formalism and bases of numerical relativity. 2007.

[30] Miguel Alcubierre. *Introduction to 3+1 Numerical Relativity.* Usa, Oxfort University Press, 2012.

[31] Masaru Shibata and Takashi Nakamura. Evolution of three-dimensional gravitational waves: Harmonic slicing case. *Phys.Rev.*, D52:5428–5444, 1995.

[32] Thomas W. Baumgarte and Stuart L. Shapiro. On the numerical integration of Einstein's field equations. *Phys.Rev.*, D59:024007, 1999.

[33] Frans Pretorius. *PAMR Reference Manual,*
http://bh0.phas.ubc.ca/Doc/PAMR_ref.pdf, 2002.

[34] Gerhard Rein, Alan D. Rendall, and Jack Schaeffer. Critical collapse of collisionless matter: A numerical investigation. *Phys. Rev. D*, 58:044007, Jul 1998.

[35] Ignacio Olabarrieta and Matthew W. Choptuik. Critical phenomena at the threshold of black hole formation for collisionless matter in spherical symmetry. *Phys. Rev. D*, 65:024007, 2001.

[36] Roland Stevenson. The spherically symmetric collapse of collisionless matter. *M. Sc. thesis, University of British Columbia,*
http://bh0.phas.ubc.ca/Theses/stevenson.pdf, 2005.

[37] Hakan Andréasson and Gerhard Rein. A numerical investigation of the stability of steady states and critical phenomena for the spherically symmetric einstein-vlasov system. *Class. Quant. Grav.*, 23(11):3659, 2006.

[38] S. L. Shapiro and S. A. Teukolsky. Relativistic stellar dynamics on the computer. I - Motivation and numerical method. *Astrophys. J.*, 298:34–79, November 1985.

[39] S. L. Shapiro and S. A. Teukolsky. Relativistic Stellar Dynamics on the Computer - Part Two - Physical Applications. *Astrophys. J.*, 298:58, November 1985.

[40] S. L. Shapiro and S. A. Teukolsky. Relativistic stellar dynamics on the computer. IV - Collapse of a star cluster to a black hole. *Astrophys. J.*, 307:575–592, August 1986.

[41] Stuart L. Shapiro and Saul A. Teukolsky. Formation of naked singularities: The violation of cosmic censorship. 66:994–997, 1991.

[42] Stuart L. Shapiro and Saul A. Teukolsky. Collision of relativistic clusters and the formation of black holes. 45:2739–2750, 1992.

[43] Hakan Andréasson and Gerhard Rein. On the steady states of the spherically symmetric einstein-vlasov system. *Class. Quant. Grav.*, 24(7):1809, 2007.

[44] Patrick R. Brady, Chris M. Chambers, and Sergio M.C.V. Goncalves. Phases of massive scalar field collapse. *Phys. Rev.*, D 56:6057–6061, 1997.

[45] Scott H. Hawley and Matthew W. Choptuik. Boson stars driven to the brink of black hole formation. 62:104024, 2000.

[46] Scott Charles Noble. *Ph. D. thesis, University of Texas at Austin, http://bh0.phas.ubc.ca/Theses/noble.pdf*, 2003.

[47] Ke-Jian Jin and Wai-Mo Suen. Critical phenomena in head-on collisions of neutron stars. *Phys. Rev. Lett.*, 98:131101, 2007.

[48] Thorsten Kellermann, Luciano Rezzolla, and David Radice. Critical Phenomena in Neutron Stars II: Head-on Collisions. *Class. Quant. Grav.*, 27:235016, 2010.

[49] David Radice, Luciano Rezzolla, and Thorsten Kellermann. Critical Phenomena in Neutron Stars I: Linearly Unstable Nonrotating Models. *Class. Quant. Grav.*, 27:235015, 2010.

[50] Mew-Bing Wan. Universality and properties of neutron star type I critical collapses. *Class. Quant. Grav.*, 28:155002, 2011.

[51] Steven L. Liebling et al. Evolutions of magnetized and rotating neutron stars. 81:124023, 2010.

[52] Jose M. Martin-Garcia and Carsten Gundlach. Self-similar spherically symmetric solutions of the massless einstein-vlasov system. *Phys. Rev. D*, 65:084026, 2002.

[53] Alan Rendall. An introduction to the einstein-vlasov system. *Banach Center Publications*, 41(1):35–68, 1997.

[54] C. Cercignani and G.M. Kremer. *The Relativistic Boltzmann Equation: Theory and Applications*. Progress in Mathematical Physics. Birkhäuser Basel, 2002.

[55] Gerhard Rein. Static solutions of the spherically symmetric vlasov-einstein system. *Math. Proc. Cambridge*, 115:559–570, 5 1994.

[56] G. Rein and A.D. Rendall. Global existence of solutions of the spherically symmetric vlasov-einstein system with small initial data. *Commun. Math. Phys.*, 150(3):561–583, 1992.

[57] R. J. Leveque. Finite volume methods for hyperbolic problems. *Cambridge University Press*, 2002.

[58] Hakan Andréasson. An investigation of the buchdahl inequality for spherically symmetric static shells. *J. Phys. Conf. Ser.*, 66(1):012008, 2007.

[59] H. A. Buchdahl. General relativistic fluid spheres. *Phys. Rev.*, 116:1027–1034, Nov 1959.

[60] Olivier Sarbach, Gioel Calabrese, Jorge Pullin, and Manuel Tiglio. Hyperbolicity of the BSSN system of Einstein evolution equations. *Phys.Rev.*, D66:064002, 2002.

[61] Gen Yoneda and Hisaaki Shinkai. Advantages of modified ADM formulation: Constraint propagation analysis of Baumgarte-Shapiro-Shibata-Nakamura system. *Phys.Rev.*, D66:124003, 2002.

[62] C. Bona, J. Massó, E. Seidel, and J. Stela. First order hyperbolic formalism for numerical relativity. *Phys.Rev.*, D56:3405–3415, 1997.

[63] Miguel Alcubierre, Bernd Brügmann, Peter Diener, Michael Koppitz, Denis Pollney, et al. Gauge conditions for long term numerical black hole evolutions without excision. *Phys.Rev.*, D67:084023, 2003.

[64] Frans Pretorius. Numerical relativity using a generalized harmonic decomposition. *Class.Quant.Grav.*, 22:425–452, 2005.

[65] Evgeny Sorkin and Matthew W. Choptuik. Generalized harmonic formulation in spherical symmetry. *Gen.Rel.Grav.*, 42:1239–1286, 2010.

[66] J. David Brown. Covariant formulations of BSSN and the standard gauge. *Phys.Rev.*, D79:104029, 2009.

[67] Carsten Gundlach. The Choptuik space-time as an eigenvalue problem. *Phys.Rev.Lett.*, 75:3214–3217, 1995.

[68] Carsten Gundlach. Understanding critical collapse of a scalar field. *Phys. Rev. D*, 55:695–713, Jan 1997.

[69] Shahar Hod and Tsvi Piran. Fine structure of choptuik's mass-scaling relation. *Phys. Rev. D*, 55:R440–R442, Jan 1997.

[70] David Garfinkle and G. Comer Duncan. Scaling of curvature in subcritical gravitational collapse. *Phys.Rev.*, D58:064024, 1998.

[71] José M. Martín-García and Carsten Gundlach. All nonspherical perturbations of the choptuik spacetime decay. *Phys. Rev. D*, 59:064031, Feb 1999.

[72] José M. Martín-García and Carsten Gundlach. Global structure of Choptuik's critical solution in scalar field collapse. *Phys. Rev. D*, 68:024011, 2003.

[73] Frans Pretorius and Matthew W. Choptuik. Gravitational collapse in (2+1)-dimensional AdS space-time. *Phys. Rev.*, D62:124012, 2000.

[74] Matthew W. Choptuik, Eric W. Hirschmann, Steven L. Liebling, and Frans Pretorius. Critical collapse of the massless scalar field in axisymmetry. *Phys.Rev.*, D68:044007, 2003.

[75] Matthew W. Choptuik, Eric W. Hirschmann, Steven L. Liebling, and Frans Pretorius. Critical collapse of a complex scalar field with angular momentum. *Phys.Rev.Lett.*, 93:131101, 2004.

[76] A.M. Abrahams and C.R. Evans. Critical behavior and scaling in vacuum axisymmetric gravitational collapse. *Phys.Rev.Lett.*, 70:2980–2983, 1993.

[77] Miguel Alcubierre, Gabrielle Allen, Bernd Brügmann, Gerd Lanfermann, Edward Seidel, et al. Gravitational collapse of gravitational waves in 3-D numerical relativity. *Phys.Rev.*, D61:041501, 2000.

[78] David Garfinkle and G. Comer Duncan. Numerical evolution of Brill waves. *Phys.Rev.*, D63:044011, 2001.

[79] Oliver Rinne. Constrained evolution in axisymmetry and the gravitational collapse of prolate Brill waves. *Class.Quant.Grav.*, 25:135009, 2008.

[80] Evgeny Sorkin. On critical collapse of gravitational waves. *Class.Quant.Grav.*, 28:025011, 2011.

[81] James Healy and Pablo Laguna. Critical Collapse of Scalar Fields Beyond Axisymmetry. *Gen.Rel.Grav.*, 46:1722, 2014.

[82] Lucía Santamaría Lara. Nonlinear 3d evolutions of brillwave spacetimes and critical phenomena. *M. Sc. thesis, Friedrich-Schiller-Universität Jena*, 2006.

[83] David Hilditch, Thomas W. Baumgarte, Andreas Weyhausen, Tim Dietrich, Bernd Brügmann, et al. Collapse of Nonlinear Gravitational Waves in Moving-Puncture Coordinates. *Phys.Rev.*, D88(10):103009, 2013.

[84] Miguel Alcubierre. The Appearance of coordinate shocks in hyperbolic formalisms of general relativity. *Phys.Rev.*, D55:5981–5991, 1997.

[85] Miguel Alcubierre. Hyperbolic slicings of space-time: Singularity avoidance and gauge shocks. *Class.Quant.Grav.*, 20:607–624, 2003.

[86] David Garfinkle and Carsten Gundlach. Symmetry seeking space-time coordinates. *Class. Quant. Grav.*, 16:4111–4123, 1999.

[87] J. W. York, Jr. Kinematics and dynamics of general relativity. In L. Smarr, editor, *Sources of Gravitational Radiation.* Seattle, Cambridge University Press, (1979).

[88] Miguel Alcubierre and Martha D. Mendez. Formulations of the 3+1 evolution equations in curvilinear coordinates. *Gen.Rel.Grav.*, 43:2769–2806, 2011.

[89] Arman Akbarian. *FD: finite differencing toolkit in Maple, http://laplace.phas.ubc.ca/People/arman/FD*, 2014.

[90] J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Advances in Computational Mathematics*, 6(1):207–226, 1996.

[91] Milton Ruiz, Miguel Alcubierre, and Dario Nunez. Regularization of spherical and axisymmetric evolution codes in numerical relativity. *Gen.Rel.Grav.*, 40:159–182, 2008.

[92] Isabel Cordero-Carrión and Pablo Cerda-Duran. Partially implicit Runge-Kutta methods for wave-like equations in spherical-type coordinates. 2012.

[93] Pedro J. Montero and Isabel Cordero-Carrión. BSSN equations in spherical coordinates without regularization: vacuum and non-vacuum spherically symmetric spacetimes. *Phys.Rev.*, D85:124037, 2012.

[94] Thomas W. Baumgarte, Pedro J. Montero, Isabel Cordero-Carrión, and Ewald Müller. Numerical Relativity in Spherical Polar Coordinates: Evolution Calculations with the BSSN Formulation. *Phys.Rev.*, D87(4):044026, 2013.

[95] H.-O. Kreiss and J. Oliger. *Methods for the approximate solution of time dependent problems.* GARP Publications Series No. 10, (1973).

[96] S. A. Teukolsky. LINEARIZED QUADRUPOLE WAVES IN GENERAL RELATIVITY AND THE MOTION OF TEST PARTICLES. *Phys. Rev.*, D26:745–750, 1982.

[97] Dieter R. Brill. On the positive definite mass of the Bondi-Weber-Wheeler time-symmetric gravitational waves. *Annals Phys.*, 7:466–483, 1959.

[98] K. Eppley. Evolution of Time Symmetric Gravitational Waves: Initial Data and Apparent Horizons. *Phys. Rev.*, D16:1609–1614, 1977.

[99] Shoken M. Miyama. Time Evolution of Pure Gravitational Waves. *Prog. Theor. Phys.*, 65:894, 1981.

[100] A. M. Abrahams and C. R. Evans. Trapping a geon: Black hole formation by an imploding gravitational wave. *Phys. Rev.*, D46:4117–4121, 1992.

[101] Peter Anninos, Joan Masso, Edward Seidel, Wai-mo Suen, and Malcolm Tobias. The Near linear regime of gravitational waves in numerical relativity. *Phys. Rev.*, D54:6544–6547, 1996.

[102] Peter Anninos, Joan Masso, Edward Seidel, Wai-Mo Suen, and Malcolm Tobias. Dynamics of gravitational waves in 3-D: Formulations, methods, and tests. *Phys. Rev.*, D56:842–858, 1997.

[103] Miguel Alcubierre, Gabrielle Allen, Bernd Bruegmann, Gerd Lanfermann, Edward Seidel, Wai-Mo Suen, and Malcolm Tobias. Gravitational collapse of gravitational waves in 3-D numerical relativity. *Phys. Rev.*, D61:041501, 2000.

[104] David Garfinkle and G. Comer Duncan. Numerical evolution of Brill waves. *Phys. Rev.*, D63:044011, 2001.

[105] Oliver Rinne. Constrained evolution in axisymmetry and the gravitational collapse of prolate Brill waves. *Class. Quant. Grav.*, 25:135009, 2008.

[106] A. M. Abrahams and C. R. Evans. Critical behavior and scaling in vacuum axisymmetric gravitational collapse. *Phys. Rev. Lett.*, 70:2980–2983, 1993.

[107] Andrew M. Abrahams and Charles R. Evans. Universality in axisymmetric vacuum collapse. *Phys. Rev. D*, 49:3998–4003, Apr 1994.

[108] D. E. Holz, W. A. Miller, M. Wakano, and J. A. Wheeler. Coalescence of primal gravity waves to make cosmological mass without matter. In *Directions in General Relativity: An International Symposium in Honor of the 60th Birthdays of Dieter Brill and Charles Misner College Park, Maryland, May 27-29, 1993*, 1993.

[109] M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahn, S. M. Vorkoetter, J. McCarron, and P. DeMarco. *Maple Introductory Programming Guide.* Maplesoft, (2005).

[110] L. Bernardin, P. Chin, P. DeMarco, K. O. Geddes, D. E. G. Hare, K. M. Heal, G. Labahn, J. P. May, J. McCarron, M. B. Monagan, D. Ohashi, and S. M. Vorkoetter. *Maple Programming Guide.* Maplesoft, (2011).

[111] P. Musgrave, D. Pollney, and K. Lake. *GRTensor II, http://grtensor.phy.queensu.ca/*, (1994).

[112] A. R. Mitchell and D. F. Griffiths. *The Finite Difference Method in Partial Differential Equations*. New York: Wiley, (1980).

[113] H. Kreiss Gustatsson, B. and J. Oliger. *Time-Dependent Problems and Difference Methods*. New York: Wiley, (1995).

[114] R. L. Marsa and M. Choptuik. *RNPL Reference Manual, http://laplace.physics.ubc.ca/People/matt/Rnpl/index.html*, (1995).

Appendix A

Appendix: FD, Finite Difference Toolkit

A.1 Introduction

FD is a set of Maple [109, 110] routines and definitions designed to handle various tasks in applying finite difference techniques in solving partial differential equations (PDEs). Particularly, it is developed to provide a methodology and a syntactic language to solve time dependent or boundary value PDEs arising in physics. Solving a PDE involves various complications, including finding the correct finite difference approximation (FDA) to a specific accuracy, dealing with boundary points on the discretized numerical domain, initialization, developing testing facilities for insuring accuracy, and finally creating routines to solve the FDA equations over the numerical domain. FD is designed to simplify these steps while providing full control over the entire process, allowing the user to focus on the underlying physical phenomena. Specifically, FD is not created to be a "blackbox" PDE solver, rather it provides a mixed level of automation and user controlled definitions.

FD is still under development and was originally designed to be used in the numerical relativity research where the computational task to numerically solve the Einstein's equations [38], is rather challenging. Keeping that in mind, FD was developed to deal with PDEs and differential expressions that are lengthy (in some case

[38] A set of 10 highly complex and non-linear coupled PDEs that govern the dynamics of the curved spacetime in strongly gravitating objects like black holes or neutron stars.

thousands or tens of thousands of expressions) and are usually machine generated to avoid human error. Therefore, FD is written in the Maple language, which provides a powerful symbolic manipulation environment and unifies the process of deriving the continuum form of the PDEs, and applying finite difference methods to create a discretized form. Furthermore, FD is built to directly parse a given differential expression[39] in its canonical continuous form [40] in Maple. This eliminates the need for having another high-level specification to define a PDE which can be a cumbersome task for the user, especially if the PDEs are derived from tensorial equations – such as PDEs arising in general relativity. This prevents potential human errors in transferring the equations from the symbolic calculation environment to the target "PDE solver" environment. In addition, FD inherits all of the capabilities of Maple language to deal with PDEs and algebraic expressions. In particular, the user can manage their working environment using Maple's built-in data and control structures and use *PDEtools* package to implement various other tasks such as coordinate transformation and checking for the consistency of the equations.[41]

After posing a PDE as a set of FDA equations over a discretized domain, these equations can be solved using FD's default *point-wise Newton-Gauss-Sidel* relaxation algorithm (see Sec. A.2.2) – which is a common method in solving nonlinear time dependent PDEs. FD generates Fortran subroutines (and C wrappers) to perform the relaxation and may be used as a rapid prototyping tool to implement various finite difference schemes to solve a PDE. It also provides a rapid development work-flow to create routines to evaluate the residual of the given FDA expression as a diagnostic tool.

FD is capable of dealing with the boundaries of the numerical domain by providing a syntax to specify the PDE or boundary conditions differently at different parts

[39] PDE, written in the from: $D(f) = 0$, where D is a differential operator and f is the unknown function, would be a special case of a differential expression that is equal to zero.

[40] An expression in which derivatives are presented using Maple's `diff` operator. An example of such expression is: `diff(f(t,x),t,t) - diff(f(t,x),x,x)`

[41] We note that GRTensor [111] Maple package is available for dealing with tensorial partial differential equations and tensor manipulation.

of the discretized domain. This allows the user to impose various boundary conditions such periodic boundary conditions, asymptotic behaviour boundary conditions or inner boundary conditions. This, particularly, is achieved in FD by implementing an equivalent method to the ghost cell technique used in finite difference methods, and can be used to create inner boundary conditions that arise from the symmetries in the system – such as requesting particular functions to be even or odd in specific coordinate direction.

In FD's environment, specifying the finite difference scheme by the user is as simple as merely providing the order of accuracy and limitation on the allowed grid points in the Finite Difference Molecule (FDM). FD has a simple internal algorithm to determine the number of points required to do "forward", "backward" and "centered" finite differencing of a given partial differential expression with the given accuracy. It ensures that the generated stencil expression has accuracy that is equal to the user specified value or better. The computed stencils are all stored in an internal table and are user accessible to be monitored for their order of accuracy and form.

Finally, FD produces Fortran routines (and C wrappers) that are parallel-ready and can be used in the framework of a high performance computing infrastructure. This is achieved by passing boundary flags to the routines which specify if the boundaries of the grid are between CPUs or are real physical boundaries. FD adopts PAMR's [33] standard in this matter, but any other parallelization framework should also have a similar method to deal with the inner CPU boundaries. We note that the Fortran routines generated by FD use only the basic data types of Fortran language and creating wrappers to communicate with them from a different language should be a straightforward task. By default, FD generates the C language wrappers which is one of the most common languages in high performance computing.

This user manual describes all of the features mentioned above and introduces the syntax of FD for posing a PDE as a finite difference equation with the given boundary conditions. First, two algebraic types are defined which are the fundamental objects that FD uses to identify a finite difference expression. These types are the building blocks that FD uses to directly translate a PDE to a discretized equation and eventually to Fortran routines. Then, a derived Maple table is introduced that specifies the PDE and the boundary conditions over the discretized numerical domain. Finally, we present the utilities FD provides to choose a finite difference scheme, compute the FDA equivalent of a given PDE and create Fortran codes to solve it. We assume that the reader has a working knowledge of Maple programming and is familiar with the basic concepts of finite difference methods. Some of these concepts are reviewed in Sec. A.2. An experienced user may skip this section, while those who are not are encouraged to consult the references [109, 110, 112, 113].

A.2 Overview of Finite Difference Method

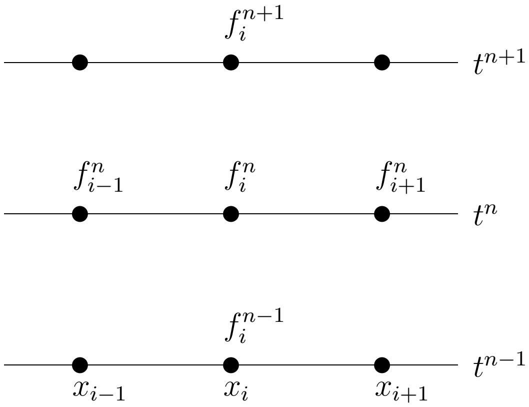
Finite difference methods are numerical techniques to express continuum differential expressions/equations as (approximate) algebraic expressions/equations. The resulting expression is known as the *Finite Difference Approximation* (FDA). An FDA for a derivative term, such as $df(x)/dx$, at a given point x, is a combination of the values of the function at certain points in the vicinity of x. For instance, values at the points $\{f(x), f(x+\Delta x), f(x+2\Delta x)\}$ (discretized values) can be used to approximate the first derivative of the function as:

$$\frac{df(x)}{dx} \approx \frac{-3f(x) + 4f(x+\Delta x) - f(x+2\Delta x)}{2\Delta x}\,, \tag{A.1}$$

where Δx is the *step size* of the discretization. This “scheme” is called *forward finite differencing*, as the discrete values are extended in positive(forward) x direction. Similarly, one can use the points $\{f(x), f(x-\Delta x), f(x+\Delta x)\}$ to compute the second derivative of the function,

$$\frac{d^2 f(x)}{dx^2} \approx \frac{f(x-\Delta x) - 2f(x) + f(x+\Delta x)}{\Delta x^2} \,. \tag{A.2}$$

Here the point x is at the center, and thus the scheme is named *centered finite differencing.* The discretized points, $(\cdots, x-\Delta x, x, x+\Delta x, \cdots)$, construct a domain for an Ordinary Differential Equation (ODE) or a Partial Differential Equation (PDE). The following diagram illustrates this concept of *discretized numerical domain* for a 1+1 (1 spatial, 1 time) dimensional spacetime:



A discretization method transforms a function from a continuum form to a discrete form symbolized as:

$$f(t,x) \rightarrow f(t_n, x_i) \equiv f_i^n \,. \tag{A.3}$$

Here, we denote the time indexing with the superscript n and the spatial indexing using the subscript symbols (i,j,k). The grid structure, $\cup x_i \times \cup t^n$, (and similarly in higher dimensions add y_j and z_k) is usually considered to be uniform:

$$t^n = t^0 + n\Delta t \equiv t^0 + n h_t \,, \tag{A.4}$$

$$x_i = x_{\min} + i\Delta x \equiv x_{\min} + ih_x \,, \tag{A.5}$$

$$y_j = y_{\min} + j\Delta y \equiv y_{\min} + jh_y \,. \tag{A.6}$$

Using these symbols, a partial differential expression such as $\partial_x f(t,x)$ can be written as:

$$\frac{\partial f(t,x)}{\partial x} = \frac{f(t,x+h_x) - f(t,x-h_x)}{2h_x} + O(h_x^2) = \frac{f_{i+1}^n - f_{i-1}^n}{2h_x} + O(h_x^2) \,, \tag{A.7}$$

and the wording "approximation" is due to the neglecting of the $O(h_x^2)$ term. Here the function $O(h_x^2)$ has explicit dependency of the from h_x^2 on the step size, and represents the error of the approximation (or equivalently can be interpreted as the "accuracy" of the FDA). Replacing all of the derivatives with FDA expressions, a PDE becomes an algebraic equation for the discrete values of the function. For example, consider performing the following FDA on the heat equation,

$$\frac{\partial f(t,x)}{\partial t} + \alpha \frac{\partial^2 f(t,x)}{\partial x^2} = 0 \quad \rightarrow \quad \frac{f_i^{n+1} - f_i^n}{h_t} + \alpha \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{h_x^2} = 0 \,, \tag{A.8}$$

where in the discretized version of the equation, the unknown is the vector:

$$\mathbf{F}^{n+1} = f_i^{n+1} \,, \tag{A.9}$$

and is to be solved numerically for a given $\mathbf{F}^n$. Obviously knowing the values $\mathbf{F}^1$, i.e the initial time profile of the function f, the process of solving $\mathbf{F}^{n+1}$ in terms of $\mathbf{F}^n$ means, by induction, finding the entire solution on the time domain indexed by n.

A.2.1 Computing the FDA Expression

There is a systematic method to find the FDA of the l'th derivative of a function, $d^l f(x)/dx^l$. Consider L points, in the vicinity of x as:

$$\{x + q_1\Delta x, x + q_2\Delta x, \cdots , x + q_L\Delta x\} \,, \tag{A.10}$$

where q_i's are L distinct integers usually chosen in a minimalistic fashion such that $x + q_i\Delta x$ is close to x. For example, the forward and centered finite differencing in Eq. (A.1) and Eq. (A.2) are associated with:

$$\{q_1, q_2, q_3\}_{forward} = \{0, 1, 2\} \;\; , \;\; \{q_1, q_2, q_3\}_{center} = \{-1, 0, 1\} \,. \tag{A.11}$$

Using these L points, and L unknown coefficients $\{\beta_1, \beta_2, \beta_3, \cdots , \beta_L\}$ one can create L Taylor expansions upto truncation error $O(\Delta x^L)$,

$$\beta_1 f(x+q_1\Delta x) = \beta_1 F^{(0)}+\beta_1 q_1 F^{(1)}+\beta_1 q_1^2 F^{(2)}+\cdots+\beta_1 q_1^l F^{(l)}+\cdots+\beta_1 q_1^{(L-1)} F^{(L-1)} \,, \tag{A.12}$$

$$\beta_2 f(x+q_2\Delta x) = \beta_2 F^{(0)}+\beta_2 q_2 F^{(1)}+\beta_2 q_2^2 F^{(2)}+\cdots+\beta_2 q_2^l F^{(l)}+\cdots+\beta_2 q_2^{(L-1)} F^{(L-1)} \,, \tag{A.13}$$

$$\vdots$$

$$\beta_L f(x+q_L\Delta x) = \beta_L F^{(0)}+\beta_L q_L F^{(1)}+\beta_L q_L^2 F^{(2)}+\cdots+\beta_L q_L^l F^{(l)}+\cdots+\beta_L q_L^{(L-1)} F^{(L-1)} \,, \tag{A.14}$$

where we defined:

$$F^{(r)} = \frac{d^r f(x)}{dx^r}\frac{(\Delta x)^r}{r!} \,, \tag{A.15}$$

and $F^{(0)} = f(x)$. Then we can find the coefficients $\{\beta_1, \beta_2, \beta_3, \cdots, \beta_L\}$ such that summing over the entire right hand sides of the equations, all of the $F^{(r)}$ terms have coefficients zero, except $F^{(l)}$ which can be set to have coefficient 1. This process leads to the following set of L linear equations for β_i's:

$$\begin{aligned}\sum_{m=1}^{L} \beta_m f(x + q_m \Delta x) &= F^{(0)} \sum_m \beta_m + F^{(1)} \sum_m q_m \beta_m + F^{(2)} \sum_m q_m^2 \beta_m \\ &+ \cdots + F^{(l)} \sum_m q_m^l \beta_m + \cdots + F^{(L-1)} \sum_m q_m^{(L-1)} \beta_m = F^{(l)}\end{aligned}$$

$$\Rightarrow$$

$$\begin{aligned}\sum_m \beta_m &= 0 \\ \sum_m q_m^1 \beta_m &= 0 \\ \sum_m q_m^2 \beta_m &= 0 \\ &\vdots \\ \sum_m q_m^{l-1} \beta_m &= 0 \\ \sum_m q_m^{l} \beta_m &= 1 \\ \sum_m q_m^{l+1} \beta_m &= 0 \\ &\vdots \\ \sum_m q_m^{(L-1)} \beta_m &= 0\end{aligned}$$

For L distinct given q_i's, this linear system has a unique solution vector which we denote by $\beta_i^\star$. Note that the left hand side of the summation is a finite difference expression:

$$\sum_{m=1}^{L} \beta^{\star}_{m} f(x+q_m \Delta x) = F^{(l)} = \frac{d^l f(x)}{dx^l} \frac{(\Delta x)^l}{l!} \Rightarrow \frac{d^l f(x)}{dx^l} = \frac{l!}{(\Delta x)^l} \sum_{m=1}^{L} \beta^{\star}_{m} f(x+q_m \Delta x)\,, \tag{A.16}$$

and therefore we find the desired FDA expression for the l'th derivative using L neighboring points. In this calculation, clearly one should assume,

$$L \geq l+1\,, \tag{A.17}$$

which simply indicates that finding the FDA of a l'th derivative term requires at least $l+1$ points. The truncation error in the Taylor expansions is $O(\Delta x^L)$ and since the finite difference sum is divided by Δx^l in Eq. (A.16) the accuracy of the final finite difference expression is at least $O\left(\Delta x^{(L-l)}\right)$. However in certain cases (for example in centered scheme) the finite difference expression can have higher accuracy as the coefficient in the next leading $O(\Delta x^L)$ term in the summation happens to simplify to zero. The reader may verify this for the FDA given in Eq. (A.2)

This calculation is internally performed by FD as it encounters derivative terms in a PDE and returns the FDA equivalent of them.[42] There is a simple front-end function (mostly for demonstration purposes) in FD:

```
Sten(diffexpr,[points])
```

which calls the internal FDA operator on the given differential expression, `diffexpr`, and computes the stencil using the points, `[points]`, (denoted by $\{q_i\}$ in the systematic derivation above). For example in the following we demonstrate the computation of the forward and centered FDA in Eq. (A.1) and Eq. (A.2) for the first and second derivatives respectively:

[42] We note that FD does *not* use any of Maple's substitution/replacement procedures, rather it performs recursively to parse a PDE and *return* FDA equivalents of its differential expressions.

```
> Sten(diff(f(x),x),[0,1,2]);
                            -3 f(x) + 4 f(x + h) - f(x + 2 h)
                        1/2 ---------------------------------
                                            h
> Sten(diff(f(x),x,x),[-1,0,1,2,3]);
              11 f(x - h) - 20 f(x) + 4 f(x + 2 h) + 6 f(x + h) - f(x + 3 h)
         1/12 --------------------------------------------------------------
                                           2
                                          h
```

Example 1: Simple FDA of derivatives using FD

We emphasize that this procedure is solely for demonstration purposes, and acts only on a single derivative term. In practice, FD uses a different procedure, `Gen_Sten`, that performs the FDA operation according to an FDA scheme specification provided by the user, and it performs on arbitrary length PDEs.

A.2.2 Iterative Schemes for Non-Linear PDEs

Solving a time dependent PDE for a function $f(t,\vec{x})$ involves integrating the equation forward in time, given the initial value $f(0,\vec{x})$. In the discrete language of finite differencing, this process reduces to finding the *advanced time level* value of the function, f^{n+1}_{ijk}, for the given *current value*, f^{n}_{ijk}. Starting with the "initial data", f^{0}_{ijk}, the time integration can be performed by applying this process consecutively for N_t time levels:

$$\text{Initial Data}\quad f^{n=0}_{i,j,k} \rightarrow f^{n=1}_{i,j,k} \rightarrow \quad \ldots \quad \rightarrow f^{n=N_t}_{i,j,k} \quad \text{Final State} \tag{A.18}$$

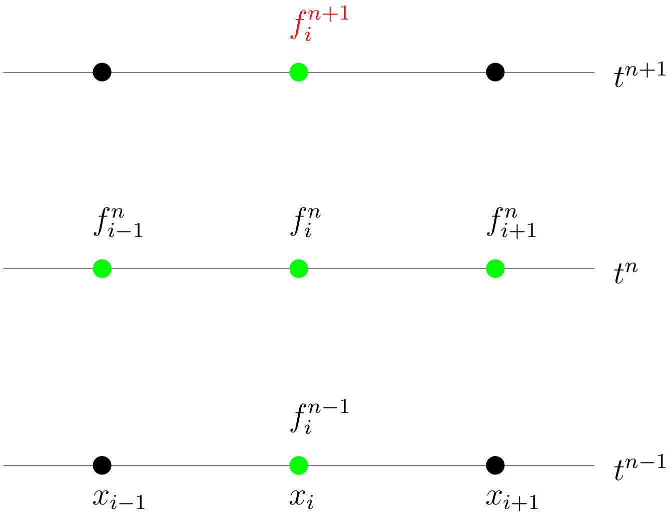
To demonstrate this update process, let's revisit the 1-D heat equation, with a different discretization scheme (known as leap-frog):

$$\frac{\partial f(t,x)}{\partial t} + \alpha \frac{\partial^2 f(t,x)}{\partial x^2} = 0 \quad \rightarrow \quad \frac{f^{n+1}_{i} - f^{n-1}_{i}}{2h_t} + \alpha \frac{f^{n}_{i+1} - 2f^{n}_{i} + f^{n}_{i-1}}{h_x^2} = 0\,. \tag{A.19}$$

This finite difference equation (FDE), is a second order approximation to the PDE at the point (t^n, x_i), and it involves values of the function at that point, and the points in the vicinity of it. The FDE includes the following points:

$$\{(n+1,i),(n-1,i),(n,i-1),(n,i),(n,i-1)\}\,. \tag{A.20}$$

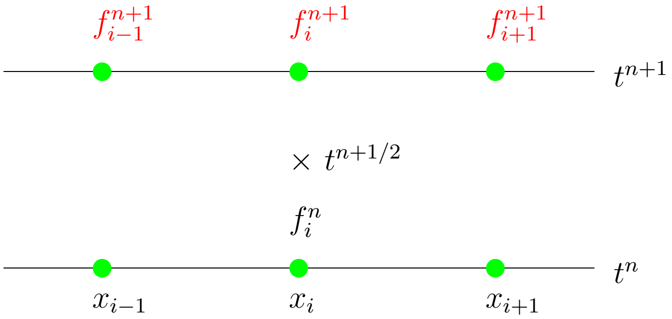
and the"unknown" in this set, as highlighted in (A.19), is f_i^{n+1}. This set of points is called the *Finite Difference Molecule* (FDM) and is illustrated in the following diagram for the FDA of heat equation (A.19):



FDM depends on the finite difference scheme. For example, consider a different (also second order accurate) FDA of the heat equation at the point $(t^{n+1/2}, x_i)$, where $t^{n+1/2}$ denotes the point $t_n + h_t/2$:

$$\frac{\partial f(t,x)}{\partial t}+\alpha\frac{\partial^2 f(t,x)}{\partial x^2}=0 \quad\rightarrow\quad \frac{f_i^{n+1}-f_i^n}{h_t}+\frac{1}{2}\alpha\left(\frac{f_{i+1}^n-2f_i^n+f_{i-1}^n}{h_x^2}+\frac{f_{i+1}^{n+1}-2f_i^{n+1}+f_{i-1}^{n+1}}{h_x^2}\right)=0\,. \tag{A.21}$$

The FDM of this equation is illustrated in the following diagram:



and again the unknown is highlighted both in the diagram and the equation. The main difference between this discretization and the previous one is in the fact that, this FDM requires 2 time level, whereas FDE (A.19) has 3 time levels. More importantly, in this scheme there are 3 unknowns in the FDA: $\{f_{i-1}^{n+1}, f_i^{n+1}, f_{i+1}^{n+1}\}$ and therefore there is an implicit dependency of advanced time level unknowns. This type of FD schemes are known as *implicit schemes*. The FD schemes such as the leap-frog scheme used in (A.19) – where the dependency of the FDM on the advanced time level is explicitly a single point – are known as *explicit schemes*.

After converting a PDE to a FDE, the next step is solving this algebraic equation. We can write an FDE in a compact form:

$$L_i^h\left(f_i^{n+1}, f_i^n, \cdots\right) \equiv \mathbf{L}^h\left(\mathbf{F}^{n+1}, \mathbf{F}^n, \cdots\right) = \vec{0}\,, \tag{A.22}$$

where $\mathbf{L}^h$ is the FDA operator, the most advanced time level values, f_i^{n+1}, is considered as the unknown, and the superscript h denotes the typical step size of the discretization. Here we defined the vector:

$$\mathbf{F}^{n+1} \equiv [f_i^{n+1}]\,. \tag{A.23}$$

Depending on the PDE and the chosen FDA scheme, this equation can be solved numerically using various methods. For a linear PDE and an explicit scheme, Eq. (A.22) is indeed a linear equation:

$$\mathbf{A}\mathbf{F}^{n+1} = \mathbf{b} \tag{A.24}$$

where $\mathbf{A}$ is a diagonal matrix and $\mathbf{b}$ is a vector that depends on previous time level values of the function, $\mathbf{F}^{n}, \mathbf{F}^{n-1}, \cdots$. In this case, solving the FDE simply reduces to inverting a diagonal matrix, i.e. inverting the diagonal terms – which can be done in a single (trivial) matrix operation. But in general, if the PDE is linear and FD scheme is implicit, the FDE reduces to the same linear equation as (A.24), but the matrix $\mathbf{A}$ is no longer diagonal. In even more general case, where the PDE is non-linear, and the FD scheme is implicit, one needs to solve a non-linear algebraic equation for a vector of unknowns. Such systems are perhaps the most interesting and are the subject of study with the FD toolkit.

In this scenario, one can solve the non-linear FDE using the multivariable iterative Newton method:

$$\mathbf{F}^{n+1}_{l+1} = \mathbf{F}^{n+1}_{l} - \mathbf{J}^{-1}(\mathbf{R}_l) \tag{A.25}$$

in which the sub-subscript l index's the number of Newton method iterations, i.e. $\mathbf{F}^{n+1}_{l+1}$ is the new approximate solution after a single iteration, and $\mathbf{F}^{n+1}_{l}$ is the old solution. In recursive Eq. (A.25), $\mathbf{J}^{-1}$ is the inverse of the Jacobian matrix of the FD operator $\mathbf{L}^h$ as a function of $\mathbf{F}^{n+1}$. More explicitly, it is the multivariable derivative of the nonlinear FDA operator $\mathbf{L}$:

$$\mathbf{J}_{ji} \equiv \frac{\partial L_j}{\partial f^{n+1}_i} \,. \tag{A.26}$$

Finally, in Eq. (A.25), ${}_l\mathbf{R}$ denotes the "residual" of the FDE for the previous approximate solution generated from the Newton iteration:

$$\mathbf{R}_l \equiv \mathbf{L}\left(\mathbf{F}^{n+1}_{l}\right) \,. \tag{A.27}$$

Note that this iterative method requires an initial guess that is usually taken to be the previous time step solution:

$$\mathbf{F}_0^{n+1} = \mathbf{F}^n \,. \tag{A.28}$$

Here the logic is simple: if the PDE evolves the function slowly in time, $\mathbf{F}^{n+1}$ is close to $\mathbf{F}^n$ and thus $\mathbf{F}^n$ should be a good initial guess for it. Note that in this method, each time level update demonstrated in (A.18) has another layer of Newton iteration presented in (A.25). This internal iteration usually converges very quickly (in few steps).

So far, we have only provided a formal description of solving a non-linear FDE. Practically, the numerical inversion of $\mathbf{J}$ is a non-trivial task. One can use the Gauss-Seidel or Jacobi methods to find the inverse matrix iteratively, however, since this Jacobian is going to be used in the Newton iteration (A.25) rather than performing two independent iterative schemes, one can simply find an approximate inverse Jacobian by only taking the diagonal part of this matrix and use that in the Newton solver. [43] This approach is called *point-wise Newton Gauss-Seidel* method and is equivalent to assuming that the only unknown in FDE is f_i^{n+1} (fixing the rest of advanced time level values that occur in an implicit FDA scheme) and solve for it using a single variable Newton method:

$$[f_i^{n+1}]_{l+1} = [f_i^{n+1}]_l - [R_{ii}]_l / J_{ii} \tag{A.29}$$

where:

$$[R_{ii}]_l = L_i\left([f_i^{n+1}]_l\right) \tag{A.30}$$

[43] The convergence of such method is guaranteed if the Jacobi matrix is diagonally dominant, i.e. $\sum_{i=j} |A_{ij}| > \sum_{i\neq j} |A_{ij}|$

is the residual of the FDA equation at the point i and:

$$J_{ii} = \frac{\partial L_i \left(f_i^{n+1}, f_i^n, \cdots \right)}{\partial f_i^{n+1}} \tag{A.31}$$

is the diagonal element of the Jacobian matrix. Note that there are two iterations involved here, one over index i, the numerical grid, and one on the Newton iteration index l. It is ineffective to perform the l iteration first, since a highly accurate solution to the point-wise Newton problem will become completely disrupted as soon as the value of the next neighboring point f_{i+1}^{n+1} is changed via the next Newton iteration. Therefore, it is much more effective to perform the iterating over the numerical grid first. This is known as a single point-wise Newton Gauss-Seidel *relaxation sweep* and if it converges, it usually only takes few iteration. Performing this relaxation, for few times, a single time step evolution is complete and the algorithm (A.18) can proceed to the next step.

This algorithm is the first approach to solve a non-linear PDE and is the default (and at the moment only) method that is built into FD toolkit for solving the PDEs. As we will discuss in detail, invoking the procedure:

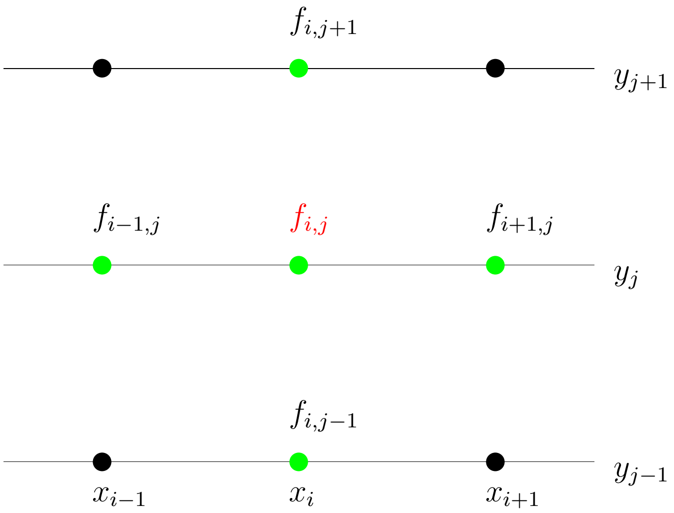
```
A_Gen_Solve_Code(DDS,{solve_for_var},input="d/c*",proc_name="my_proc")
```

will create a low level (Fortran) routine that performs the relaxation sweep. Having this routine, a PDE can be solved by a driver routine that applies the relaxation as needed (depending on some stopping criteria). Of course, solving a PDE involves several other steps, such as dealing with boundary points where rather than FDA equivalent of PDE, a boundary condition needs to be imposed. This is done by defining and passing the `DDS` variable which is a Maple data type to specify a PDE and its boundary conditions over a discretized domain. It is the description of the `DDS` and other tools and objects that are needed before applying this procedure that constitutes the majority of this documentation.

We note that a similar discussion to what we described about the time dependent PDEs also applies to the boundary value problem PDE's (elliptic PDEs). For example, consider the following second order discretization of the Laplace equation:

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = 0 \quad \rightarrow \quad \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h_x^2} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{h_y^2} = 0\,. \tag{A.32}$$

The finite difference molecule for this FDA is illustrated in the following diagram:



In this case, one needs to provide the discrete values of the function at the boundary points, and the unknowns are all of the values in the interior points f_{ij}:

$$\text{(BVP)} \quad \{f_{1,j} \;\; f_{N_x,j} \;\; f_{i,1} \;\; f_{i,N_y}\} \;\rightarrow\; f_{i,j} \quad \text{(unknown)} \tag{A.33}$$

Again, a simple approach to solve this PDE is to use iterative schemes. For example, one can solve the FDA equation (A.32) for the mid-point value f_{ij}, assuming the values at the neighboring points are fixed. Then performing this point-wise solver process over all of the interior points (a relaxation sweep) iteratively will decrease the residual to the desired tolerance (if it converges). However we note that relaxation schemes for boundary value problems (BVP) converge very slowly and other algorithms such as multigrid [33] are essential to efficiently solve elliptic-type PDEs.

A.2.3 Testing Facilities: Convergence and IRE

Finding a solution to a PDE or an ODE can be a complex task. However, if the solution is given as a discrete function, checking that it satisfies the equation is somewhat a straightforward process. Consider the equation:

$$L(f) = 0\,, \tag{A.34}$$

where L is a differential operator and f is the unknown function. One can use an FDA scheme to discretize the differential operator:

$$L \to L^h \tag{A.35}$$

where h denotes the typical "size" of the discretization. Then for a given solution function, $\tilde{f}$, one can evaluate the residual:

$$R^h = L^h(\tilde{f}) \tag{A.36}$$

to confirm if the function $\tilde{f}$ satisfies the discretized version of the equation. A solid testing facility for a numerical solver is to *independently* develop this residual evaluator, which we refer as *Independent Residual Evaluator* (IRE). Of course, the residual (A.36) will not be exactly zero since L^h is an approximation to L and perhaps $\tilde{f}$ is also a numerical solution to (A.34) that differs from the exact solution f. However, one would expect if the solution $\tilde{f}$ is well resolved, is "close enough" to the exact solution, and FDA operator L^h is a "good" approximation of L, then the norm of this residual should be orders of magnitude smaller than the actual norm of the function $\tilde{f}$. A more rigorous definition of all these concepts and how to validate the numerical solution using an IRE test will follow. However, before that we momentarily dive into FD toolkit and how it provides a rapid work-flow to creating IRE routines.

Consider the following ODE for $a(x)$ on a given time t:

$$\frac{da(x)}{dx} - \frac{1-a(x)^2}{2x} - \frac{1}{2}x\left[\left(\frac{\partial\phi(t,x)}{\partial x}\right)^2 + \left(\frac{\partial\phi(t,x)}{\partial t}\right)^2\right] = 0 \tag{A.37}$$

where $\phi(t,x)$ is a time dependent field which can have its own dynamical PDE. Here we want to evaluate the left hand side of the equation for the given discrete solutions a_i and ϕ_i^n and verify that it is zero (numerically). The process involves creating an FDA of this ODE, evaluating the residual over the numerical domain, summing up the point-wise residuals and returning a norm of it. FD toolkit provides an almost fully automated mechanism to do so. For example, if we use FD's default FDA scheme (second order accurate and centered), the Maple code to generate the IRE Fortran routine in this case is:

```
> read "FD.mpl": Make_FD():
> grid_functions:={a,phi}:
> res_a := diff(a(x),x)/a(x) - (1 - a(x)^2)/(2*x) -
    1/2*x*(diff(phi(t,x),x)^2+diff(phi(t,x),t)^2):
> Gen_Res_Code(res_a,input="c",proc_name="ire_a");
Fortran Code is written to ire_a.f
C header is written to ire_a.h
C call is written to ire_a_call
```

Example 2: Creating testing (IRE) routines with FD is fully automated.

The steps in this examples are: loading the FD package, initializing the internal variables of FD, defining symbols 'a' and 'phi' as grid functions, writing down the ODE, and passing the equation in its continuum form to the procedure:

```
Gen_Res_Code(expr,input="c*/d",proc_name="myproc");
```

This call creates 3 source code files:

- `ire_a.f`: is the Fortran subroutine that evaluates the residual (A.37). This subroutine has the following header:

```
subroutine ire_a(a,n_phi,nm1_phi,np1_phi,x,Nx,ht,hx,res)
```

and as you can see, it requires passing in the function a and 3 time levels of function ϕ, denoted by `n_phi` (current time), `nm1_phi` (retarded time), and `np1_phi` (advanced time) since these values are required to compute the time derivative expression in the residual (in centered scheme). The last parameter `res` is a generic name, that always stores and returns the result of the computation (it will correspond to the updated value of the dynamical function when solver routines are generated).

- `ire_a.h` is the C header (wrapper) file that needs to be included in a C driver routine to use the subroutine, the content of this file is:

```
void ire_a_(double *a,double *n_phi,double *nm1_phi, double *np1_phi,
        double *x,int *Nx,double *ht,double *hx,double *res);
```

- `ire_a_call`: is a plain text file containing a typical C call of the routine. `_call` files can be copied to a C driver code. For example, here the content of the file is:

```
ire_a_(a,n_phi,nm1_phi,np1_phi,x,&Nx,&ht,&hx,res);
```

which as you can see, is a C call with the last parameter, again, labeled as `res`. After copying the content of `_call` to the driver code, the user needs to appropriately change the name of the last parameter to the allocated vector (pointer) or the single variable defined in the C driver to store the result. [44] In this example, the result, `res`, is a number (a double precision floating point number) containing the norm of the residual. FD also assumes that in the C driver, the user will define the name of the allocated vectors and parameters for the PDE similar to what they are defined in the Maple expression.

[44] Of course, a good strategy is to avoid naming any variables in the C driver code as `res`. The name `res` does not need any modification in the Fortran routine or C header file.

We will discuss this procedure and similar other code generator procedures in more details through Sec. A.3 to Sec. A.6. Following note is a mathematical discussion on the notion of convergence and independent residual evaluators. Even though, these concepts are crucial to validate the consistency and accuracy of the numerical solver, the following is somewhat independent of the FD toolkit and applies to any finite difference method. This manual should be accessible without expertise in the mathematical discussion in the following note.

...

Note on convergence and IRE tests

Consider that the solution in Eq.(A.34) is produced by solving a finite difference approximation for the PDE. To preface this section, we first review our notation:

$$L(f) = 0 \tag{A.38}$$

$$S^h(f^h) = 0 \tag{A.39}$$

$$L^h(f^h) = R^h \tag{A.40}$$

i.e. L is the PDE operator in continuum form, and f is the continuum solution, f^h is the numerical solution and S^h is the solver FDA (the FDA of the original PDE that is used in the numerical solver). Finally, L^h is another FDA to L *that is different* than S^h, and due to this difference the RHS is nonzero and symbolized by the residual R^h. Note that previously we used L^h to denote the FDA used in the numerical solver, but here we are mostly interested in testing the solver using a different FDA operator which is the main focus of this section and thus denoted by L^h.

If the numerical solution f^h is convergent at the continuum limit – where the discretization size h approaches zero– we denote the continuum limit by u:

$$\exists u = \lim_{h \to 0} f^h \tag{A.41}$$

therefore one can assume the following Richardson expansion:

$$f^h = u + e_f^h = u + e_1 h + e_2 h^2 + \cdots \tag{A.42}$$

where the coefficients e_1, e_2 are functions independent of h. As one might expect, the error in the solution e_f^h depends on the accuracy of FDA S^h that is used in the numerical solver. The first non-zero coefficient e_p that appears in the expansion defines the accuracy of the solution, and is the dominant part of the error in the limit $h \to 0$. For example, a second order convergent solution has the form:

$$f^h = u + e_2 h^2 + \cdots \tag{A.43}$$

and using this expansion it is easy to show that for the 3 consecutively refined convergent solutions: f^h, $f^{h/2}$ and $f^{h/4}$ the limit of the following ratio:

$$\lim_{h \to 0} Q = \frac{||f^h - f^{h/2}||}{||f^{h/2} - f^{h/4}||} = 4\,, \tag{A.44}$$

is 4. Here $||.||$ is some norm of a discretized functions. Measuring the factor Q is referred as standard *convergence test* in the literature.

For a convergent numerical solution f^h, it is not clear that the limiting continuum function u (A.41) is indeed the solution to the continuum problem $L(f) = 0$, i.e. we want to know if:

$$u \overset{?}{=} f\,. \tag{A.45}$$

To further emphasize this: the numerical solution f^h might be convergent but we need some sort of proof to show that it is in fact converging to the correct solution.

One might speculate that this should be the case if

1) S^h approximates L correctly, or more rigorously:

$$\lim_{h\to 0} S^h = L \tag{A.46}$$

known as *consistency condition* condition for the finite difference scheme.

2) The method used to solve the finite difference equation is *stable*. We refer the reader to [95] for mathematical definition and discussion on the notion of stability. In certain cases (for linear PDEs) it can be proven that stability and consistency are sufficient conditions for convergence. However, to our knowledge, there is no such proof for non-linear cases which most of the interesting physical systems exhibit. We also note that from a practical point of view there is no simple prescription or condition that can be checked off to ensure the stability of the method for non-linear systems.

Here we rather take a practical approach: the independent residual evaluation test. The IRE test provides a stronger test than the standard convergence test, and validates (or rejects) the equality A.45. Suppose that f^h is $O(h^p)$ convergent, meaning:

$$\begin{aligned} f^h &= u + e_f^h \\ e_f^h &= e_p h^p + o(h^p) \end{aligned} \tag{A.47}$$

where e_p is a function, independent of h and $o(h^p)$ is an h dependent function that converges to zero faster than h^p:

$$\lim_{h\to 0} \frac{||o(h^p)||}{h^p} = 0 \tag{A.48}$$

Now suppose, as defined in the beginning of this discussion in Eq. A.40, L^h is another FDA of the original continuum operator L (created with a different FD scheme than S^h and is also created independently). L^h is what we refer as independent residual evaluator. We assume that this operator is *consistent* with the continuum operator L upto accuracy $O(h^q)$, meaning:

$$\begin{aligned} L^h(g) &= L(g) + e^h_L(g) \\ e^h_L(g) &= h^q E_L(g) + o_L(g; h^q) \end{aligned} \tag{A.49}$$

where E_L is an h independent operator, and $o_L(.; h^q)$ is an h dependent operator with a norm that converges to zero faster than h^q:

$$\lim_{h \to 0} \frac{||o_L(g, h^q)||}{h^q} = 0 \tag{A.50}$$

Note that here we are assuming that the operator expansion (A.49) is possible for the function g. Intuitively, one would expect this assumption to hold for functions that are well resolved over the discretized domain. Particularly in the case of $g = f^h$, this is a plausible assumption, as we expect the numerical solver to produce a well-resolved discrete solution.

Now the claim is that if the conditions (A.47) and (A.49) hold then the residual defined as:

$$R^h \equiv L^h(f^h) \tag{A.51}$$

converges to zero if and only if f^h is indeed converging to f, the continuum solution, i.e.:

$$u = f \tag{A.52}$$

Furthermore the convergence behaviour of the residual is dominated by the two errors: the solution f^h error, which we assumed to be $O(h^p)$ and the error of the L^h operator which we assumed to be $O(h^q)$ and is explicitly of the form:

$$||R^h|| = O(h^p) + O(h^q) = O(h^{\min(p,q)}) \tag{A.53}$$

Therefore, for example if both the solution and the IRE are second order convergent then, one would expect to observe a second order convergence in the residual R^h as well.

Linear case:

We first prove the claim for the linear operators L and L^h:

$$\begin{aligned} L^h(f^h) = L^h(u + e_f^h) = L^h(u) + L^h(e_f^h) = L(u) + e_L^h(u) + L(e_f^h) + e_L^h(e_f^h) \\ = L(u) + h^q E_L(u) + h^p L(e_p) + h^q h^p E_L(e_q) + \cdots = L(u) + O(h^q) + O(h^p) + \cdots \end{aligned} \tag{A.54}$$

where $\cdots$ are higher order terms. We used the fact that L, L^h and e_L^h are linear operators (the linearity of e_L^h follows from its definition (A.49)). Note that in the expansion of the term $L^h(e_f^h)$, we are assuming that the error function e_f^h is also well resolved function on the mesh such that the expansion (A.49) is meaningful.

Nonlinear case:

In the non-liner case, a similar analysis can be performed by linearizing the FDA operator L^h. We assume that L^h is differentiable around g, meaning there exist a linear operator $\mathcal{D}_L^h[g]$ such that:

$$L^h(g + q) = L^h(g) + \mathcal{D}_L^h[g](q) + o_L^h[g](q) \tag{A.55}$$

and $o_L^h[g]$ is an operator with a norm converging to zero faster than $||q||$:

$$\lim_{||q|| \to 0} \frac{||o_L^h[g](q)||}{||q||} = 0 \tag{A.56}$$

The differential operator $\mathcal{D}_L^h[g]$ can be naively defined as the limit:

$$\mathcal{D}_L^h[g](q) \equiv \lim_{\epsilon \to 0} \frac{L^h(g + \epsilon q) - L^h(g)}{\epsilon} \tag{A.57}$$

Note that the differentiability of L^h is simply guaranteed if all of the partial derivatives $\partial L_i(g)/\partial g^{\tilde{i}}$ exist where L_i is the FDA equation at the point indexed by i and $g^{\tilde{i}}$

is the discrete value of the function at the point indexed by $\tilde{i}$. [45] These derivatives obviously exist for normal FDA operators used in finite difference methods. We also note that the abstract $\mathcal{D}^h_L[g]$ operator in a matrix representation is simply the $\partial L_i(g)/\partial g^j$ matrix. Furthermore, not surprisingly, it is equal to the FDA operator L^h itself, when L^h is linear:

$$\begin{aligned} \mathcal{D}^h_L[g](q) = \frac{L^h(g+\epsilon\ q) - L^h(g)}{\epsilon} = \frac{\epsilon L^h(q)}{\epsilon} = L^h(q) \\ \Rightarrow \mathcal{D}^h_L[g] = \mathcal{D}^h_L = L^h \end{aligned} \tag{A.58}$$

Note that in linear case, $\mathcal{D}^h_L$ indeed does not depend on g anymore, as the operator L^h. Assuming the differentiability of L^h around u, we have:

$$\begin{aligned} L^h(f^h) &= L^h(u+e^h_f) = L^h(u) + \mathcal{D}^h_L[u](e^h_f) + o^h_L[u](e^h_f) \\ &= L(u) + e^h_L(u) + \mathcal{D}^h_L[u](e^h_f) + o^h_L[u](e^h_f) \\ &= L(u) + h^q E_L(u) + o_L(u;h^q) + \mathcal{D}^h_L[u](e_p h^p + o(h^p)) + o^h_L[u](e_p h^p + o(h^p)) \\ &= L(u) + h^q E_L(u) + h^p \mathcal{D}^h_L[u](e_p) + o(h^p) + o(h^p) \end{aligned} \tag{A.59}$$

where in the last step we used the linearity of $\mathcal{D}^h_L[u]$ and the property of $o^h_L[u]$ operator (A.56). This result again translates to:

$$L^h(f^h) = L(u) + O(h^p) + O(h^q) = L(u) + O(h^{min(p,q)}) \tag{A.60}$$

and the residual $L^h(f^h)$ will converge to zero, if and only if $L(u) = 0$, or u the continuum function that the numerical solution f^h is converging to, is indeed the underlying continuum solution f.

Now using this result we have a stronger test: The convergence of the IRE $L^h(f^h)$ is only possible if the solution is convergent *and* is converging to the correct

[45] Note that here i and $\tilde{i}$ can be any of the discrete domain indices, here we are simply using i as a symbol of discretization

solution. Therefore if one can create a solid IRE operator L^h that is consistent with L, checking the convergence of the IRE will guarantee the accuracy of the solution. Of course, one can ask: what if L^h also has an error in its implementation ? Here the keyword *independent* development becomes crucial. If the independent residual is converging, it is extremely unlikely that S^h and L^h that should be developed completely independently will both have an internal error, and both of the errors agree, i.e both S^h and L^h happen to be identical to an FDA for another PDE that is not the original PDE. Often it is best to create the IRE operator L^h using an automated process to reduce possible human errors. This, in part, was the original motivation to develop FD and as it will be discussed further, generating IRE routines is been fully automated in FD toolkit.

A.3 Semantics of FD

In this section, we describe some of the internal variables of FD and two derived algebraic data types that FD uses to work with finite difference expressions.

A.3.1 Parsing a PDE: Fundamental Data Type

As mentioned in the introduction, FD is developed with the philosophy that user's involvement in the straightforward tasks should remain minimal. Consider the following PDE for f:

$$\begin{aligned} &\partial_t f(t,x,z) + \beta(t,x,z)\partial_x f(t,x,z) \\ +\gamma(x)\partial_z f(t,x,z) + a\partial_x^2 f(t,x,z) + b\partial_z^2 f(t,x,z) + g(x,z) &= 0 \end{aligned} \tag{A.61}$$

The LHS written in canonical Maple form (without use of aliases) is:

```
PDE:=diff(f(t,x,z),t) + beta(t,x,z)*diff(f(t,x,z),x)+
gamma(x)*diff(f(t,x,z),z) + a*diff(f(t,x,z),x,x)+b*diff(f(t,x,z),z,z)
+ g(x,z);
```

One can easily observe that this expression, by itself, contains enough information regarding the dimensionality of the problem, functions and their dependencies, parameters, and of course derivatives. By looking at the expression, we can conclude that:

- f is a time dependent function, defined on a 2 dimensional spatial domain labeled by (x, z).
- β is also time dependent with same spatial domain as f.
- g is a time independent function only defined on the (x, z) domain.
- γ has only 1 dimensional dependency on x coordinate.
- a and b are parameters (assuming that all dependencies are explicitly presented)
- the order and direction of derivatives of f are clear.

There is no need for further specification to implement this PDE on a computer. The first step to reducing potential human errors is to eliminate another intermediate syntactic language to *write* a PDE. Rather, FD uses Maple's powerful symbolic manipulation capabilities and has a built-in parser which allows directly passing a PDE to its routines. This puts the entire complexity of the fundamental data type on the expression, and frees the user from providing any further specification. As soon as an error-proof PDE is written down, (which is easily possible as the working environment of FD is Maple with all its symbolic tools) the task of identifying the parameters, functions, dimensionalities, derivatives, and required time levels to perform FDA in time dimension is left to the software. This is one of the advantages of FD, over previously developed software such as RNPL [114]. This also makes FD an efficient prototyping language, particularly for developing testing facilities as we demonstrated in Example 2.

A.3.2 Coordinates

FD reserves the variables `(t,x,y,z)` for the name of the time and spatial coordinates that define the domain of a PDE. They are protected variables after FD is loaded. Similarly, FD reserves the symbols `(n,i,j,k)` for indexing the corresponding coordinate points `(t(n),x(i),y(j),z(k))`. It uses `(ht,hx,hy,hz)` as the name for the step-size of the discretization along these coordinates, respectively. The names `(Nt,Nx,Ny,Nz)` are reserved for the size of the discretized domain, and `(xmin,xmax)`, `(ymin,ymax)`, `(zmin,zmax)` are reserved for flags to specify the inner CPU boundary points of the coordinates (their applicable is in the context of parallelization).

This association can be demonstrated as:

$$\begin{aligned} t \leftrightarrow n \leftrightarrow h_t \leftrightarrow N_t \\ x \leftrightarrow i \leftrightarrow h_x \leftrightarrow N_x \leftrightarrow (xmin, xmax) \\ y \leftrightarrow j \leftrightarrow h_y \leftrightarrow N_y \leftrightarrow (ymin, ymax) \\ z \leftrightarrow k \leftrightarrow h_z \leftrightarrow N_z \leftrightarrow (zmin, zmax) \end{aligned} \tag{A.62}$$

and is built into FD. The coordinate names, and this association table are necessary to identify functions, differential expressions, and perform finite differencing. For example, FD recognizes that an expression such as `f(x+hx,y-2*hy)` should be discretized as `f(i+1,j-2)`, or an expression such as `f(x+hy)` is invalid and cannot be discretized, since `hy` is not an stepping size in `x` direction. Ultimately, this association table allows FD to discretize a differential expression such as $\partial_x f(x, y)$ (in Maple notation: `diff(f(x,y),x)`), directly to `(f(i+1,j)-f(i-1,j))/(2*hx)` without any need for further specification. (See the example in Sec. A.3.4).

A.3.3 Initializing FD, `Make_FD`, `Clean_FD`

As the reader may have noticed from the previous examples, FD is in a Maple script format, and can be imported to a Maple worksheet/script by executing:

```
read("/your/fd/directory/FD.mpl");
```

FD's internal variables are initialized by calling the procedure:

```
Make_FD();
```

which has a short alias, `MFD()`, and creates the table for the coordinate association described in Eq. A.62) and initializes the default finite difference table that specifies the finite difference scheme. We will further discuss this table in Sec A.4.2. To clean the initialized variables, user can execute:

```
Clean_FD();
```

or use the alias `CFD()`.

A.3.4 Grid Functions Set: `grid_functions`

FD uses a global variable named `grid_functions` (of type `set` in Maple) as its reference for all of the function names that are expected to be discretized as:

$$f(t,x,y,z) \to f(t^n, x_i, y_j, z_k) \equiv f^n_{i,j,k}\,. \tag{A.63}$$

In Maple language, if symbol `f` is in the `grid_functions`, then the function `f(t,x,y,z)` (in its most generic 1+3 dimensional case) will be converted to `f(n,i,j,k)` during the process of discretization. The following example demonstrates how FD uses the coordinate names, the coordinate association table, and the symbols defined in grid functions to produce FDA expressions:

```
> read "FD.mpl": MFD():
> grid_functions:={f}:
> Gen_Sten(f(t,x,y,z));
                                f(n, i, j, k)
> Gen_Sten(diff(f(x,y),x));
                            f(i - 1, j) - f(i + 1, j)
                       -1/2 -------------------------
                                       hx
> Gen_Sten(x+g(y,z));
                            x(i) + g(y(j), z(k))
```

Example 3: Discretization of grid functions vs non-grid functions

Here, `Gen_Sten` is the main routine that performs the finite differencing and will be discussed extensively. However, user can easily guess its functionality from the example. As it can be seen, beside the names that are included in the grid function set, the coordinate variables `(t,x,y,z)` are by definition grid functions and are discretized as `(t(n),x(i),y(j),z(k))`. Furthermore, if a symbol with coordinate dependency (such as `g(y,z)` above) is *not* included in the `grid_functions` set, it will be considered as an external function that user will provide to the Fortran routines. FD discretizes its coordinate functions rather than the function. For example, here it is discretized as: `g(y(j),z(k))` rather than `g(j,k)`.

Time Level Reduction:

We shall emphasize that the discrete expression `f(n,i,j,k)` will be eventually (at the point of code generation) replaced by: `n_f(i,j,k)`. This process is done internally, and is referred as *time level reduction* (See Sec. ??). The time level `n` is usually referred as *current* time level, `n-1` is referred as *retarded* time level and `n+1` is called *advanced* time level. When FD performs the time level reduction, it uses the prefix `np1_` and `nm1_` in the names of the advance and retarded time levels functions respectively:

$$\texttt{f(n,i,j,k)} \rightarrow \texttt{n\_f(i,j,k)}$$

$$\texttt{f(n+1,i,j,k)} \rightarrow \texttt{np1\_f(i,j,k)}$$

`f(n-1,i,j,k)` → `nm1_f(i,j,k)`

The higher time level `f(n+2,i,...)` will be renamed to `np2_f(i,...)` and the syntax for the other cases should be clear. This replacement is simply because in time dependent finite difference algorithms only a finite number of time levels are needed and stored in the memory during the time evolution. The user can define the time levels in the C driver code according to this standard, or can define "alias" pointers (that adopts these names) to the underlying data structure to be able to use the FD generated routines.

A.3.5 Known Functions

FD has a set of "known" functions, which is basically a set of floating point functions that are known to the low level language (Fortran here). These functions in FD are:

```
{ln,log,exp,sin,cos,tan,cot,tanh,coth,sinh,cosh,exp,sqrt,‘^‘,‘*‘,‘+‘,‘-‘,‘/‘}
```

and during a discretization process, FD does *not* convert their arguments to a discrete version, rather it discretizes the arguments accordingly. For example, `sin(z)*f(x)+exp(y)` will be discretized as:

```
> Gen_Sten(sin(z)*f(x)+exp(y));
                             sin(z(k)) f(i) + exp(y(j))
```

assuming that `f` is in `grid_functions`.

A.3.6 Valid Continuous Expression, VCE

Valid Continuous Expression (VCE) is an algebraic function of the continuous coordinate variables, (t, x, y, z), in which the dependencies of grid functions on the coordinates are only of the form:

$$f(t + lh_t, x + mh_x, y + qh_y, z + ph_z)\,, \tag{A.64}$$

where (l, m, q, p) are known integers (not variable), and (h_t, h_x, h_y, h_z) are the associated stepsize variables. Furthermore, a VCE does *not* have explicit dependency on the discretization indices (n, i, j, k). For example, if functions `f` and `g` are grid functions, then all of the expressions:

```
f(t,x,y) + (g(x+hx)-g(x-hx))/(2*hx)
r(x*y)
u(sin(x*y),g(z))
f(x+2*hx,y-3*hy)/hz + x*z^2 + g(z,x,t,y)
```

are VCE, and

```
f(t,x+hy)
g(x,y+2)
f(x(i),y(j))
cos(j)
f(u(x),y)
g(x*y)
diff(f(x),x)
```

are all *invalid* continuous expressions. Particularly, compare `g(x*y)` and `r(x*y)`, former is not VCE, since `g` is defined as a grid function, while later is a VCE as r is considered an external function. Note that FD does not check for the consistency in the order of the variables, i. e. `f(x,y) + f(y,x)` is considered a VCE.

A.3.7 Valid Discrete Expression, VDE

Valid Discrete Expression (VDE) is an expression in which the explicit dependencies of functions on the discretization indices `(n,i,j,k)` is only via the grid functions or coordinates. Furthermore, this dependency is of the form: $f(n+q, i+m, j+p, \dots)$, where $q, m, p, \cdots$ are known integers, and f is either a grid function or is one of the coordinates (t, x, y, z). In the case of coordinate, indexing must be done according to the coordinate-index association (A.62). For example, for `grid_functions:={f,g}`:

```
g(i+1,j-2)

x(i)

u(x(i),f(j,k),a)

f(j,k+2,i)
```

are all VDE and,

```
y(i)

x(i)+k

sin(i)

f(i*j)

u(i)

f(i,y(j))
```

are *invalid* discrete expression.

A.3.8 Conversion Between VDE and VCE

The definition of VDE and VCE allows a one-to-one mapping between these two types. FD provides two functions for the conversion:

```
A:=DtoC(B::VDE);
```

```
B:=CtoD(A::VCE);
```

Even though VCE's are not practically useful for numerical implementations, the conversion of a VDE to VCE can be used for demonstration and testing purposes. For example, a finite difference expression in VDE form, can be converted to a VCE, and then a Taylor expansion of it can reveal its equivalent continuum differential operator. The following demonstrates the process for Kreiss-Oliger dissipation operator [95] that is commonly used in finite difference methods:

```
> read "FD.mpl": Make_FD():
> grid_functions:={f}:
> A:= -epsdis/(16*ht)*( 6*f(n,i) + f(n,i+2) + f(n,i-2)
                     -4*(f(n,i+1) + f(n,i-1))        ):
> B:=DtoC(A):
> E:=convert(series(B,hx),polynom);
                                                          4
                         epsdis D[2, 2, 2, 2](f)(t, x) hx
               E := -1/16 ---------------------------------
                                       ht
```

Example 4: Conversion between VDE and VCE

which gives:

$$E = \frac{-\epsilon}{16}(\frac{\partial^4 f(t,x)}{\partial x^4})\frac{h_x^4}{h_t} \tag{A.65}$$

A.4 Discretizing a PDE

In this section we discuss how to perform a finite differencing on a PDE using the facilities of FD, how to choose a specific discretization scheme and and how to access the results of a lengthy finite difference operation.

A.4.1 Performing the Finite Differencing, `Gen_Sten`

The main routine that performs FDA is:

```
VDE/VCE::Gen_Sten(expr)
```

(with an alias: `GS`) where the `expr` is an arbitrary mixed differential/algebraic Maple expression. As mentioned before, this routine performs the discretization on the grid functions and coordinates, leaving parameters and other functions unchanged (the coordinate of the functions however will be discretized). The result is by default a VDE type. To return a the finite difference expression in VCE form, the optional input `discretized` should be disabled:

`VCE::Gen_Sten(expr,discretized=false)`

Note: In the examples in the rest of this manual, we assume that the FD initialization is invoked and `f` and `g` are grid functions:

```
> read "FD.mpl": MFD():
Warning, grid_functions is not assigned
FD table updated, see the content using SFDT() command
> grid_functions:={f,g}:
```

Here is an example of discretizing differential expressions:

```
> A:=diff(f(x,y),x,y):
> B:=Gen_Sten(A);
          -f(i - 1, j - 1) + f(i - 1, j + 1) + f(i + 1, j - 1) - f(i + 1, j + 1)
 B := -1/4 ----------------------------------------------------------------------
                                        hy hx
 > Gen_Sten(diff(f(x),x)+g(y)+cos(f(x))+r(x)+z);
     f(i - 1) - f(i + 1)
-1/2 ------------------- + g(j) + cos(f(i)) + r(x(i)) + z(k)
              hx
 > Gen_Sten(A,discretized=false);
    -f(x - hx, y - hy) + f(x - hx, y + hy) + f(x + hx, y - hy) - f(x + hx, y + hy)
 -1/4 ------------------------------------------------------------------------------
                                     hy hx
```

Example 5: Discretizing a PDE

As one can see, the default discretization scheme in FD is centered (and second order accurate). In the next section, we describe how to change the finite difference scheme.

A.4.2 Discretization Scheme, `FD_table`

FD uses an internal table, `FD_table`, to perform the finite difference operations such as ones in Example 5. This table, simply is a list of the points that can be used for the n'th derivative computation for each of the coordinates (t,x,y,z). For example, the x component of this table is:

```
> FD_table[x];
[[0], [-1, 0, 1], [-1, 0, 1], [-2, -1, 0, 1, 2], [-2, -1, 0, 1, 2], ...
```

where n'th element (counting from zero), is a list of points specifying the finite differencing scheme for the n'th derivative along x. The numbers present the list of neighboring points to x(i) that are allowed to be used for FDA. For instance, the third element, [-2,-1,0,1,2], presents the 5 points: central point x(i), 2 to left and 2 to right, that are allowed for FDA of the third derivatives in x coordinate. This is demonstrated in the following diagram.

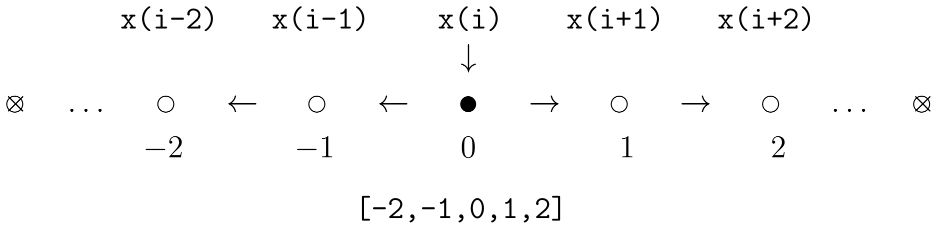


Figure A.1: Five points specifying the FDA scheme for the third derivatives in FD_table in x direction.

FD initializes the finite difference table when Make_FD() is invoked. By default, the table is upto the 5'th derivatives (adjustable by the global variable MAX_DERIVATIVE_NUMBER=5) in all dimensions, and the points expand symmetrically around the central point (centered finite differencing).

A.4.3 Changing the FDA Scheme: FDS, Update_FD_Table

The FD scheme can be chosen by adjusting the content of FD_table. FD provides a convenient routine for this purpose:

```
Update_FD_Table(order::integer,fds::FDS);
```

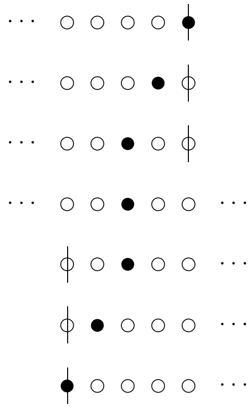
in which the user specifies the desired order of accuracy, `order`, and the scheme via the second argument fds. This argument is a table with a particular format which we refer as a *Finite Difference Specifier* (FDS). A FDS is a table for the 4 coordinates, `(t,x,y,z)`,

```
fds:=table( [ t=... , x=..., y=... , z=...] );
```

and each element has the following format:

```
X = [p_left,-1] or [-1,-1] or [-1,p_right]
```

in which `X` denotes one of the coordinates, and the values `p_left` and `p_right` are known integers. `p_left` specifies how many points to left of the central point is allowed, and similarly `p_right` specifies the number of points to the right that can be used in an FDA for coordinate `X`. If these values are set to -1 it allows FD to expand in that direction to as many point as needed to achieve the desired accuracy. At least one of the `p`-values must be set to -1. Particularly, the `p_left` and `p_right` need to be adjusted for creating FDAs that can be applied in the vicinity of the boundaries of the numerical grid. This is demonstrated in the following diagram:



```
fds:=table([ t=[-1,-1],x=[-1,0],y=[-1,-1],z=[-1,-1] ]):
fds:=table([ t=[-1,-1],x=[-1,1],y=[-1,-1],z=[-1,-1] ]):
fds:=table([ t=[-1,-1],x=[-1,2],y=[-1,-1],z=[-1,-1] ]):
fds:=table([ t=[-1,-1],x=[-1,-1],y=[-1,-1],z=[-1,-1] ]):
fds:=table([ t=[-1,-1],x=[2,-1],y=[-1,-1],z=[-1,-1] ]):
fds:=table([ t=[-1,-1],x=[1,-1],y=[-1,-1],z=[-1,-1] ]):
fds:=table([ t=[-1,-1],x=[0,-1],y=[-1,-1],z=[-1,-1] ]):
```

Figure A.2: Specifying different types of FD schemes: note the values in the highlighted color and how it associates with each case at the vicinity of the boundary in x direction.

We remind the reader that higher derivatives require more points. In addition, increasing the accuracy `order` also adds to the number of the points used in FDAs. The routine `Update_FD_Table` has a built-in function $P(n,m)$

$$\frac{\partial^m}{\partial X^m} \text{ with } O(h^n) \text{ accuracy} \rightarrow P(n,m) \tag{A.66}$$

for each of the forward, backward and centered schemes that estimates the minimum number of points required to achieve the desired accuracy (or better).

For example the following code updates the FD table use FD scheme forward in time, centered in `x`, backward in `y` and asymmetric backward in `z`, with 4'th order accuracy. The resulting `FD_table` is demonstrated by inspecting each element of it:

```
> fds:=table([t=[0,-1],x=[-1,-1],y=[-1,0],z=[-1,2]]):
> Update_FD_Table(4,fds):
FD table updated, see the content using SFDT() command
> FD_table[t];
[[0], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4, 5, 6],...]
> FD_table[y];
[[0], [-4, -3, -2, -1, 0], [-6, -5, -4, -3, -2, -1, 0], ...]
> FD_table[z];
[[0], [-2, -1, 0, 1, 2], [-4, -3, -2, -1, 0, 1, 2], [-4, -3, -2, -1, 0, 1, 2],...]
```

Example 6: Changing the Finite Difference Scheme

We note that FD table can be updated manually by overwriting the elements, however this method is error-prone, and higher derivatives in particular might not have a sufficient number of points to be evaluated as a FDA. For example, in the following, we specify only 2 points for the second derivative in time, and the `Gen_Sten` procedure outputs an error as it is impossible to compute the FDA equivalent of the input according to the FD table:

```
> FD_table[t]:=[[0],[0,1,2],[0,1]]:
> Gen_Sten(diff(f(t,x),t));
           -f(n, i) + f(n + 1, i)
           ----------------------
                    ht
> Gen_Sten(diff(f(t,x),t,t));
Error, (in Calc_Stencil_L) Failed to find FDA coefficients, check FD_table content!
```

Finally, we note that the entire content of FD_table (rather lengthy sequence of integers!) can be viewed using the procedure:

```
Show_FD_Table();
```

A.4.4 Accessing the FD Results: Show_FD

If the Gen_Sten procedure is used to perform finite differencing on a lengthy differential expression, the resulting FDA is not human readable. To better present what Gen_Sten has performed, the routine stores the differential expressions it finds in the input and their FDA equivalent that it replaces them with, in an internal table named FD_results. The content of this table can be accessed using the procedure:

```
Show_FD();
```

For example, consider the following finite differencing operation:

```
> A:=diff(y*f(x,y)*diff(sin(x*y)*g(x),x),x,y):
> B:=Gen_Sten(A):
memory used=11.4MB, alloc=5.4MB, time=0.59
> lprint(B);

1/2*(-f(i-1,j)+f(i+1,j))/hx*(cos(x(i)*y(j))*y(j)*g(i)-1/2*sin(x(i)*y(j))*(g(i-1)-g(i+1))
/hx)+1/4*y(j)*(f(i-1,j-1)-f(i-1,j+1)-f(i+1,j-1)+f(i+1,j+1))/hy/hx*(cos(x(i)*y(j))*y(j)*g
(i)-1/2*sin(x(i)*y(j))*(g(i-1)-g(i+1))/hx)+1/2*y(j)*(-f(i-1,j)+f(i+1,j))/hx*(-sin(x(i)*y
(j))*x(i)*y(j)*g(i)+cos(x(i)*y(j))*g(i)-1/2*cos(x(i)*y(j))*x(i)*(g(i-1)-g(i+1))/hx)+f(i,
j)*(-sin(x(i)*y(j))*y(j)^2*g(i)-cos(x(i)*y(j))*y(j)*(g(i-1)-g(i+1))/hx+sin(x(i)*y(j))*(g
(i-1)-2*g(i)+g(i+1))/hx^2)+1/2*y(j)*(-f(i,j-1)+f(i,j+1))/hy*(-sin(x(i)*y(j))*y(j)^2*g(i)
-cos(x(i)*y(j))*y(j)*(g(i-1)-g(i+1))/hx+sin(x(i)*y(j))*(g(i-1)-2*g(i)+g(i+1))/hx^2)+y(j)
```

```
*f(i,j)*(-cos(x(i)*y(j))*x(i)*y(j)^2*g(i)-2*sin(x(i)*y(j))*y(j)*g(i)-sin(x(i)*y(j))*x(i)
*y(j)*(-g(i-1)+g(i+1))/hx+cos(x(i)*y(j))*(-g(i-1)+g(i+1))/hx+cos(x(i)*y(j))*x(i)*(g(i-1)
-2*g(i)+g(i+1))/hx^2)

# Checking if B is indeed an FDA for A:
> E:=DtoC(B):
> E:=convert(series(E,hx,4),polynom):
> E:=convert(series(E,hy,4),polynom):
> residual:=simplify(eval(A-E,hx=0,hy=0));
                residual := 0
```

Expression A has several derivatives of the functions f and g that are replaced with FDA expressions. Now by invoking Show_FD() we can see what replacements have been done:

```
 > Show_FD();

  d                 -f(i - 1, j) + f(i + 1, j)
{-- f(x, y) = [1/2 --------------------------, [[x, 2], [y, -1]]],
  dx                            hx

 d                  f(i, j - 1) - f(i, j + 1)
 -- f(x, y) = [-1/2 -------------------------, [[y, 2], [x, -1]]],
 dy                            hy

 d              -g(i - 1) + g(i + 1)
 -- g(x) = [1/2 --------------------, [[x, 2]]],
 dx                     hx

  2
 d           g(i - 1) - 2 g(i) + g(i + 1)
 --- g(x) = [----------------------------, [[x, 2]]],
   2                       2
 dx                     hx

 d               g(j - 1) - g(j + 1)               d
 -- g(y) = [-1/2 -------------------, [[y, 2]]], ----- f(x, y) = [
 dy                     hy                       dy dx
```

```
     -f(i - 1, j - 1) + f(i - 1, j + 1) + f(i + 1, j - 1) - f(i + 1, j + 1)
-1/4 ----------------------------------------------------------------------,
                                hy hx
[[x, 2], [y, 2]]] }
```

Here the numbers next to the coordinate variables `x,y` denotes the order of accuracy of the replacement, and as expected they are all second order accurate. -1 represents exact FDA, i.e. there is no differentiation with respect to that coordinate. Note that this accuracy is *not* what user specifies when updating FD scheme (in previous section). It is indeed the computed value of the actual accuracy of FDA which should be equal or higher to the user specified value.

A.4.5 Defining Manual Finite Difference Operators: `FD`

FD provides a way to define an arbitrary FDA operator. In principal, any finite difference operator can be created from the shifting operator (See [95]) defined (in 1 dimension) as:

$$E(f_i) = f_{i+1} \tag{A.67}$$

and its inverse is simply: $E^{-1}(f_i) = f_{i-1}$. The generalization of this operator is defined in the FD toolkit, and is named `FD` with the following format:

```
VDE::FD(dexpr::VDE, [ [t_shift] ,[x_shift,y_shift,z_shift] ])
```

in which `FD` takes an input `dexpr` of type VDE, and returns a VDE that is shifted by the given integers `(t_shift, x_shift,y_shift,z_shift)`. If there is no time index dependency in the expression, the first argument, `[t_shift]`, can be dropped and the routine accepts a shorter format:

```
FD(VDE,[x_shift,y_shift,z_shift])
```

Similarly if `z` index `k` does not occur in the VDE, the routine accepts shorter list `[x_shift,y_shift]` and so on. For example, the following demonstrates the definition of 3 manual FDA operators: 1) a forward time derivative FDA (`DT`) that is equivalent to ∂_t upto first order accuracy, 2) centered in x derivative FDA (DXC), which is equivalent to ∂_x upto second order accuracy, and 3) the time averaging operator AVGT that is *not* an FDA. This operator is usually used in Crank-Nicolson method to create a implicit FD scheme.

```
> DT := f -> ( FD(f,[[1],[0]]) - FD(f,[[0],[0]]) )/ht:
> df:= DT(f(n,i));
                      f(n + 1, i) - f(n, i)
                df := ---------------------
                               ht
> DXC:= f -> ( FD(f,[1,0]) - FD(f,[-1,0]) ) /(2*hx):
> DXC(f(i)*x(i)^2*g(j)+y(j));
                                   2                         2
                  f(i + 1) x(i + 1)  g(j) - f(i - 1) x(i - 1)  g(j)
              1/2 -------------------------------------------------
                                         hx
> AVGT := f -> ( FD(f,[[1],[0]]) + FD(f,[[0],[0]]) )/2:
> AVGT(Gen_Sten(diff(f(t,x),x)));
          f(n + 1, i - 1) - f(n + 1, i + 1)       f(n, i - 1) - f(n, i + 1)
     -1/4 --------------------------------- - 1/4 -------------------------
                         hx                                  hx
```

Example 7: Defining manual difference operators

A.5 Posing a PDE & Boundary Conditions Over a Discrete Domain

In solving PDEs, it often occurs that some part of the discretized domain needs special treatment. By its nature, boundary points require different equations than the original PDE. In addition, if the discretization scheme results in large finite difference molecules, the points next to the boundaries also require special handling. For example consider 4'th order accurate FDA of the derivative of a function, $\partial_x f(x)$:

```
            f(i - 2) - 8 f(i - 1) + 8 f(i + 1) - f(i + 2)
     1/12 --------------------------------------------
                                  hx
```

This expression cannot be evaluated where $i < 3$ or $i > N_x - 2$, as the finite difference molecule `(-2,-1,0,1,2)` require points that do not exist in the discretized domain at these limits. In this section, we describe the methodology to create different equations for each part of the numerical domain, and the facilities FD provides to impose boundary conditions and implement techniques such as ghost cells.

A.5.1 Discrete Domain Specifier: `DDS`

To specify each portion of the discrete domain, $\{i \in (1, N_x)\} \times \{j \in (1, N_y)\} \cdots$, FD uses a syntax similar to RNPL [114], via a derived data type that we refer as Discrete Domain Specifier (DDS). A DDS is a list of equations:

```
DDS = [ equation1, equation2, ...  ]
```

where each equation specifies part of the discrete domain and has the following LHS and RHS:

```
each equation:  { indexeq1, indexeq2, ...  } = expression
```

in which each `expression` can be a VDE, or a continuous PDE, and each `indexeq` describes the indexing for one of the spatial dimensions:

```
each indexeq:  I = [start,NI-stop,step]
```

Here, the variable `I` denotes one of the indexing labels, `(i,j,k)`, `NI` is the associated domain size `Nx,Ny,Nz`, and `step` is a known integer that determines the stepping size. The `indexeq` symbolizes a portion of the domain in which index `I` takes the values: `(start,start+step,start+2*step,...)` and ends at value smaller or equal to `NI-stop`. The reader may notice that this is exactly equivalent to a *for loop* structure. For example

```
{ i = [1,Nx,1] , j =[2,Ny-1,2] } = ...
```

is equivalent to (in Fortran syntax):

```
DO i=1,Nx,1
  DO j=2,Ny-1,2
    ...
  ENDDO
ENDDO
```

The following example clarifies this syntax, and demonstrates a DDS for heat equation where the boundary points are fixed to values `T0` and `T1` and interior points are specified by the heat equation.

```
HeatEq:= diff(f(t,x),t) - diff(f(t,x),x,x);
HeatDDS := [
  { i=[1,1,1]     } = f(n+1,i) - T0 + myzero*x(i)   ,
  { i=[2,Nx-1,1]  } = Gen_Sten(HeatEq)              ,
  { i=[Nx,Nx,1]   } = f(n+1,i) - T1 +myzero*x(i)
];
```

Example 8: 1-D discrete domain specifier for the heat equation

The necessity of `myzero*x(i)` expression will become clear later when we use this DDS as an input to FD's solver routine generator.

Note that heat equation and its boundary conditions are simple and compact enough to be discretized inside the DDS. For a more complex case, it is better to create the discrete version of the equations for the boundaries separately, and pass them into the DDS using human readable names. For example, the following demonstrates a 2 dimensional DDS where each boundary uses a specific discrete equation priorly created by the user:

```
mydds2d := [
 # Interior points:
   { i=[2,Nx-1,1]  , j = [2,Ny-1,1] } =  EQD_interior ,
 # Boundaries:
   { i=[1,1,1]     , j = [1,Ny,1]   } =  EQD_left     ,
   { i=[Nx,Nx,1]   , j=[1,Ny,1]     } =  EQD_right    ,
   { i=[1,Nx,1]    , j=[1,1,1]      } =  EQD_bottom   ,
   { i=[1,Nx,1]    , j=[Ny,Ny,1]    } =  EQD_top
];
```

Example 9: Two dimensional DDS

For a set of coupled PDEs, the user can create the FDAs and DDS's using a Maple for loop. Note that FD will check for consistency of the LHS and RHS of each element of DDS as well as the consistency between all the elements. It will raise errors if the finite difference expression on the RHS does not have the same dimensionality as the LHS. However at the moment FD does *not* check if the finite difference molecule on the RHS fits into the domain specified on the LHS. The user need to be careful with the manual discretization of the equations, and to avoid out of range errors, it is best to use finite difference specifiers (FDS) discussed in Sec A.4.3.

A.5.2 Imposing Outer Boundary Conditions

The next step is to create the appropriate FDA expressions (the RHS expressions in the DDS) that are compatible with specific boundary conditions and also are created under consideration that there are limitations on the allowed points in the vicinity of the boundary points. The case of fixing the value of the function or *Dirichlet boundary condition* is quite simple and demonstrated in the example for heat equation. Often other types of boundary conditions appear in physical systems. One of particular interest is the "out-going" type or *Neumann boundary condition.* For a 1-D wave equation the out-going boundary condition is given by:

$$\partial_t f(t,x) = -\partial_x f(t,x) \tag{A.68}$$

for the right side boundary i=Nx that corresponds to the approximate positive "infinity" of the numerical domain $x = +L_\infty$. The out-going boundary condition at the the left side of the numerical domain is given by:

$$\partial_t f(t,x) = +\partial_x f(t,x) \tag{A.69}$$

for the point i=1 or $x = -L_\infty$. To implement such boundary conditions, one needs to use FDA expressions that can be evaluated at the point of boundary that does not allow symmetric FD scheme. As described in Sec. A.4.3, this is achieved by changing the FDA scheme in FD using finite difference specifiers (FDS). For example, the following demonstrates an implementation of a mixed boundary for wave equation in which left boundary is fixed while the right boundary is outgoing. Note the change of FDA scheme using the FDS: fds_backwardx.

```
WaveEq := diff(f(t,x),t,t) - diff(f(t,x),x,x):
WaveEqBdy := diff(f(t,x),t) + diff(f(t,x),x):

WaveEqD := Gen_Sten(WaveEq):

fds_backwardx:=table([ t=[-1,-1],x=[-1,0],y=[-1,-1],z=[-1,-1] ]):
Update_FD_Table(2,fds_backwardx):

WaveEqBdyD := Gen_Sten(WaveEqBdy):

ddsWAVE:= [
   i=[1,1,1]      = f(n+1,i) - myzero*x(i)   ,
   i=[2,Nx-1,1]   = WaveEqD,
   i=[Nx,Nx,1]    = WaveEqBdyD
];
```

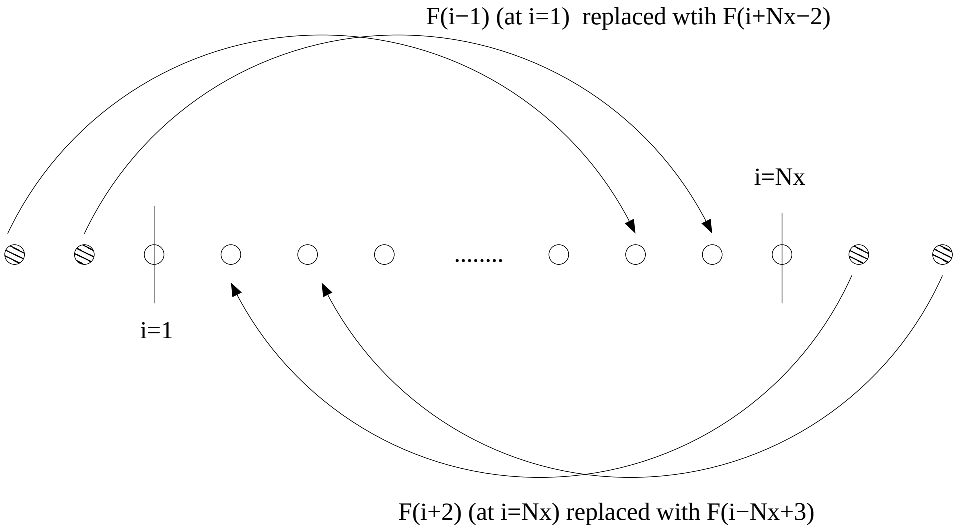
Example 10: Specifying outgoing/mixed boundary condition for wave equation

A.5.3 Periodic Boundary Condition: FD_Periodic

Another common boundary specification is periodic boundary condition (PBC). FD provides a facility to implement PBCs by making a VDE periodic. The procedure:

```
FD_Periodic(exprd::VDE,{I=1/NI})
```

takes an input, exprd, of type VDE and creates a periodic version of it. This location is specified by the second argument, in which I is one of the indices (i,j,k) and the RHS is either 1 or NI where NI is one of the associated grid size (Nx,Ny,Nz). For example i=1 denotes that a periodic version of VDE is needed at the left boundary and i=Nx denotes the same for the right boundary point. The replacements done on FDA is illustrated in the following graph:


F(i−1) (at i=1) replaced wtih F(i+Nx−2)
i=Nx
....
i=1
F(i+2) (at i=Nx) replaced with F(i−Nx+3)

The following example, demonstrates the effect of the FD_Periodic procedure on a VDE:

```
> FD_table[x]:=[ [0], [-1,0,1] ,[-2,-1,0,1,2] ]:
> A:= Gen_Sten(diff(f(x),x,x));
                  f(i - 2) - 16 f(i - 1) + 30 f(i) - 16 f(i + 1) + f(i + 2)
       A := -1/12 ---------------------------------------------------------
```

```
                                                   2
                                                 hx
> FD_Periodic(A,{i=1});
          f(i - 3 + Nx) - 16 f(i - 2 + Nx) + 30 f(i) - 16 f(i + 1) + f(i + 2)
    -1/12 -------------------------------------------------------------------
                                                 2
                                               hx
> FD_Periodic(A,{i=Nx});
          f(i - 2) - 16 f(i - 1) + 30 f(i) - 16 f(i + 2 - Nx) + f(i + 3 - Nx)
    -1/12 -------------------------------------------------------------------
                                                 2
                                               hx
```

Finally, using this procedure, the implementation of a periodic DDS for wave equation can be achieved as following:

```
ddsWAVE_Periodic:= [
  { i=[1,1,1]     } = FD_Periodic(WaveEqD,{i=1})   ,
  { i=[2,Nx-1,1]  } = WaveEqD,
  { i=[Nx,Nx,1]   } = FD_Periodic(WaveEqD,{i=Nx})
];
```

Example 11: Implementation of a periodic boundary condition

in which WaveEqD is the same as Example 10.

A.5.4 Implementing Ghost Cells for Odd and Even Functions: A_FD_Odd, A_FD_Even

The boundaries of the numerical domain often correspond to the spatial infinity. However, a different coordinate system than Cartesian coordinate can be chosen, particularly to impose a certain symmetry. For example, one can work in a spherical coordinate and assume that the function's spatial dependency is only of the form:

$$f(t,x,y,z) = f(t,r) \ \ , \ \ r = \sqrt{x^2+y^2+z^2}\,. \tag{A.70}$$

In this case, the domain of the PDE (and the function f) is $r \in (0, \infty)$. The point $r = 0$ is superficially a boundary of the numerical domain in this coordinate system, while in fact there is no physical boundary. These types of boundaries are often referred as *inner boundaries* and usually are treated by imposing a specific behaviour for the functions derived from the underlying symmetry.

One common scenario that occurs in the point (or axis) of symmetry is that functions (depending on what they represent: scalar, vector, component of a tensor etc) become even or odd. For example, a scalar function with spherical symmetry, $\psi(t, r)$, is an even function at $r = 0$, i.e:

$$\psi(t, -r) = \psi(t, r) \tag{A.71}$$

Note that here, $-r$ is neither a physical location, nor is part of the numerical domain. However, one can simply consider the function along the x axis (where $r = |x|$) and the refection symmetry $x \to -x$ implies that the function is even in x. This means, equivalently, function is even in r if we consider an extension of it to $r < 0$ that represents the value of the function at $-x$. More rigorously, the functions at the limit of $r \to 0$ take one of the two forms:

$$f(t, r) = C_0(t) + C_2(t)r^2 + C_4(t)r^4 + \dots \quad \to \quad \text{function is even} \tag{A.72}$$

$$f(t, r) = C_1(t)r + C_3(t)r^3 + C_5(t)r^5 + \dots \quad \to \quad \text{function is odd} \tag{A.73}$$

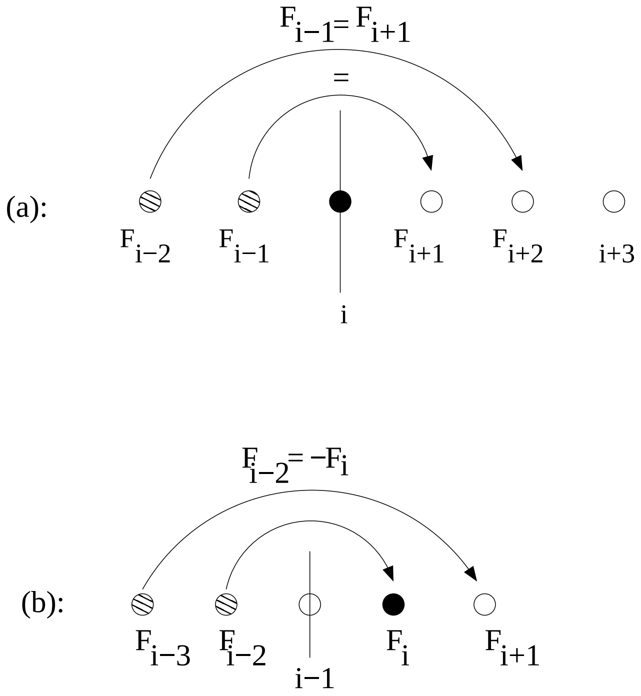
where the first case is for “scalar” functions and the second case is for “vector” functions.

This property of the functions at inner boundaries allows a discretization technique known as “ghost cells” in finite difference method. For example, consider the second and first derivative of the function with 4’th order accuracy:

```
# FD Updated to 4'th order before...
> A:=Gen_Sten(diff(f(x),x,x));
                   f(i - 2) - 16 f(i - 1) + 30 f(i) - 16 f(i + 1) + f(i + 2)
        A := -1/12 ---------------------------------------------------------
                                               2
                                             hx
> B:=Gen_Sten(diff(f(x),x));
                            f(i - 2) - 8 f(i - 1) + 8 f(i + 1) - f(i + 2)
                  B := 1/12 ---------------------------------------------
                                                hx
```

Obviously, these terms cannot be used at $i = 1$, the left boundary point, and also the point next to it $i = 2$. However, if we impose the even or odd behaviour on the function, the values `f(i-1)` and `f(i-2)` are known from the symmetry. For example, assuming that `i` is the point of symmetry, i.e the FDA will be used at `i=1`, then for an even function: `f(i-1) = f(i+1)`. This condition is illustrated in the following diagram (a). Similarly, for an odd function we have: `f(i-2) = -f(i+2)`.

Consider another example where `f` is odd, and the FDA will be used at `i=2`, the point next to the inner boundary point. The out-of-bound term in the FDA in this case is only `f(i-2)` and from the symmetry we must have: `f(i-2) = -f(i)`. The following diagram (b) clarifies this condition:



One standard method to implement this symmetry is to actually extend the numerical domain to have extra points outside the physical domain. These points, namely *ghost cells*, are updated via the symmetry, and allow FDA operations at the boundary point.

FD provides a tool equivalent to the ghost cell technique. One can directly manipulate the FDA expression according to the symmetry such that the out of bound terms are replaced appropriately and the FDA can be used at the boundary points. FD provides two procedures to perform this task:

```
A_FD_Even(exprd::VDE,coord,set_of_even_funcs,symm_loc,"forward/backward")
```

```
A_FD_Odd(exprd::VDE,coord,set_of_odd_funcs,symm_loc,"forward/backward")
```

where `exprd` is an FDA expression of type VDE, `coord` is the name of the coordinate which we are imposing the symmetry on (one of the `(x,y,z)`), the two variables `set_of_even_funcs` and `set_of_odd_funcs` are of type set and include the name of the functions that are even and odd respectively. `symm_loc` is an integer that determines the location of the inner boundary relative to the point where FDA will be evaluated. For example the diagram (a) above corresponds to: `symm_loc = 0` and the diagram (b) can be imposed by setting: `symm_loc = -1`. The last argument is of type string, and determines if the replacement should be "forward" – when the smaller index values are the out-of-bound ones and must be replaced – or "backward", i.e the larger index values are the out-of-bound and require replacement with indexed terms inside the physical domain. Normally, if the inner boundary is chosen to be the index $i = 1$ or ($j = 1, k = 1$ in higher dimensions), these procedures will only be use in "forward" mode.

The following demonstrates the usage of these two procedures and their output:

```
> A:=Gen_Sten(diff(f(x),x,x));
                   f(i - 2) - 16 f(i - 1) + 30 f(i) - 16 f(i + 1) + f(i + 2)
        A := -1/12 ---------------------------------------------------------
                                                2
                                              hx
# Diagram (a) above:
> A_FD_Even(A,x,{f},0,"forward");
                                  2 f(i + 2) - 32 f(i + 1) + 30 f(i)
                            -1/12 ----------------------------------
                                                   2
                                                 hx
# Diagram (b) above:
> A_FD_Even(A,x,{f},-1,"forward");
                            31 f(i) - 16 f(i - 1) - 16 f(i + 1) + f(i + 2)
                      -1/12 ----------------------------------------------
                                                   2
                                                 hx
> B:= Gen_Sten(diff(f(x),x));
                       f(i - 2) - 8 f(i - 1) + 8 f(i + 1) - f(i + 2)
             B := 1/12 ---------------------------------------------
                                              hx
```

```
> A_FD_Odd(B,x,{f},0,"forward");
                                 -2 f(i + 2) + 16 f(i + 1)
                            1/12 -------------------------
                                            hx
> A_FD_Even(B,x,{f},0,"forward");
                                         0
```

Example 12: Imposing even and odd symmetry at inner boundary point

Note that in the last execution, the result is identical to zero since the first derivative of an even function is zero at the point of symmetry. We also note that if the FDA involves several functions of mixed even and odd type, both of the routines need to be applied consecutively to the FDA to achieve a proper discretized version, usable at the inner boundary point.

A.6 Solving a PDEs

This section demonstrates how to incorporate all of FD's procedures and structures to solve a PDE.

A.6.1 Creating Initializer Routines: `Gen_Eval_Code`

The first step is to create routines that initialize the function $f(t = 0, x, y, z)$. If this initialization has an explicit function form depending on the coordinate and can be evaluated on every point of the numerical grid `(x(i),y(j),z(k))` then it can be simply created using the procedure:

```
Gen_Eval_Code(expr,input="c*/d",proc_name="my_init_proc");
```

where `expr` is either a continuous expression (setting `input="c"`, this is the default setting) or it is a VDE (by setting `input="d"`). The next option `proc_name` is the name of the Fortran procedure we want to create and it denotes both the name of the file (without the suffix `.f`) and the name of the procedure.

For example, consider the case where we want to set the initial profile of the wave package to a Gaussian function:

$$f(t=0,x,y) = A\exp\left(-\frac{(x-x_c)^2}{\delta_x^2} - \frac{(y-y_c)^2}{\delta_y^2}\right) \tag{A.74}$$

the following FD code performs the desired task:

```
> read "../FD.mpl": MFD():
Warning, grid_functions is not assigned
FD table updated, see the content using SFDT() command
> grid_functions:={f}:
> init_f:=A*exp( -(x-xc)^2/delx^2 - (y-yc)^2/dely^2 ):
> Gen_Eval_Code(init_f,input="c",proc_name="init_to_gauss");
Fortran Code is written to init_to_gauss.f
C header is written to init_to_gauss.h
C call is written to init_to_gauss_call
```

Example 13: Creating Initializer Fortran routines

Similar to the very first example of creating IRE routines, all of FD's code generator routines create 3 files, `X.f`, `X.h` and `X_call`, where the Fortran file `X.f` is the body of the Fortran procedure that performs the desired task. All of the procedures generated by FD have a last argument named **res**. For example the routine created by the execution above, `init_to_gauss.f` has the following header:

```
subroutine init_to_gauss(x,y,Nx,Ny,A,delx,dely,xc,yc, res)
```

in which the highlighted `res` is the pointer to the vector that stores the returned value of the procedure. In this case it is the function. Therefore user should pass in the pointer that stores `f` at initial time in the driver code. The header file `.h` is a wrapper that can be included in a C driver program to use this routine. In the example above, the content of the file, `init_to_gauss.h` is:

```
void init_to_gauss_(double *x,double *y,int *Nx,int *Ny,double *A,double *delx,
double *dely,double *xc,double *yc,double *res);
```

and finally the `X_call` files are typical C calls that can be copied to the C driver and after changing the last argument **res** to the appropriate pointer, can be used to call the Fortran routine. In the example above, the content of `init_to_gauss_call` is:

```
init_to_gauss_(x,y,&Nx,&Ny,&A,&delx,&dely,&xc,&yc,res);
```

We note that if the expression that is passed to the procedure contains derivatives, (or FDA expressions in discrete form) then this procedure only evaluates/initializes the expression at the points where the evaluation is possible i. e. allowed by the size of the finite difference molecule (FDM). This usually results in a Fortran routine that ignores the evaluation of the function on the boundary points (and perhaps in its vicinity depending on how large the resulting FDM is). If the evaluation is required at the boundary points, then the procedure described in the next section should be used.

A.6.2 Point-wise Evaluator Routines with `DDS`: `A_Gen_Eval_Code`

If the initialization is needed to vary at different portions of the discrete domain, the Fortran routine can be generated using the "evaluator" routine generator:

```
A_Gen_Eval_Code(dds:DDS,input="c*/d",proc_name="my_eval_proc");
```

where the only difference between this procedure and `Gen_Eval_Code` is in the first argument, where this procedure accepts `DDS` type to allow specific calculations at different parts of the domain. This procedure can also be used to evaluate a specific function that depends on the primary dynamical fields and their derivatives. It can also be used to evaluate the point-wise residuals of PDEs if needed.

For example, consider the problem where we want to evaluate the function f that is the Laplacian of the function ϕ in cylindrical coordinate with axial symmetry:

$$f(\rho, z) = \nabla^2\phi = \frac{1}{\rho}\partial_\rho(\rho\partial_\rho\phi(\rho, z)) + \partial_z^2\phi(\rho, z) = \partial_\rho^2\phi + \frac{\partial_\rho\phi}{\rho} + \partial_z^2\phi \tag{A.75}$$

Note that we would like to evaluate this Laplacian value on the axis $\rho = 0$ (which is an inner boundary) as well as the interior points. To do so, we need to deal with the irregular term $\frac{1}{\rho}$ and also impose an inner boundary condition at $\rho = 0$. As discussed previously, these types of inner boundary conditions are dealt by looking into the behaviour of the function at the limit of approaching the boundary, here: $\rho \to 0$. We know that the function ϕ is a scalar and its first derivative $\partial_\rho \phi$ approaches zero on the axis as $O(\rho)$. Using the L'Hospital's rule:

$$\lim_{\rho \to 0} \frac{\partial_\rho \phi}{\rho} = \frac{\partial^2 \phi}{\partial \rho^2} \tag{A.76}$$

Therefore, two versions of the expression are used to evaluate f:

$$f = \nabla^2 \phi = \begin{cases} \partial_\rho^2 \phi + \frac{\partial_\rho \phi}{\rho} + \partial_z^2 \phi & \text{if } \rho \neq 0 \\ 2\partial_\rho^2 \phi + \partial_z^2 \phi & \text{if } \rho = 0 \end{cases} \tag{A.77}$$

In addition, the evaluation of the derivative as an FDA at $\rho =$ requires implementation of boundary condition as described in Sec. A.5.4. Here the inner boundary condition is created using the fact that ϕ is an even function in ρ. The following example demonstrates all of the steps described to achieve this evaluation:

```
read "../FD.mpl": CFD(): MFD():
grid_functions:={phi}:

Laplace_Interiour:= diff(phi(x,z),x,x) + diff(phi(x,z),x)/x + diff(phi(x,z),z,z):
Laplace_Boundary_D:= Gen_Sten(2*diff(phi(x,z),x,x) + diff(phi(x,z),z,z)):

dds_2Dlaplace:= [
{ i=[2,Nx-1,1] , k=[2,Nz-1,1] } = Gen_Sten(Laplace_Interiour),
{ i=[1,1,1], k = [2,Nz-1,1] } = A_FD_Even(Laplace_Boundary_D,x,{phi},0,"forward"),
{ i=[Nx,Nx,1] , k=[1,Nz,1] } = myzero*x(i)*z(k),
{ i=[1,Nx,1] , k =[1,1,1] } = myzero*x(i)*z(k),
{ i=[1,Nx,1] , k =[Nz,Nz,1] } = myzero*x(i)*z(k)
]:
```

```
A_Gen_Eval_Code(dds_2Dlaplace,input="c",proc_name="eval_laplace");
Fortran Code is written to eval_laplace.f
C header is written to eval_laplace.h
C call is written to eval_laplace_call
```

Example 14: Point-wise Evaluator Routine Generator Using a DDS

A.6.3 Creating IRE Testing Routines: Gen_Res_Code

If the function that needs to be evaluated is indeed a residual, i.e. expected to be zero in the continuum limit, then often the user is interested in monitoring the l_2-norm of this residual. FD provides a procedure that creates a Fortran routine for such an evaluation:

```
Gen_Res_Code(expr,input="c*/d",proc_name="my_res_proc");
```

where `expr` can be a PDE residual in a continuous form or a VDE. The only difference between this procedure and `Gen_Eval_Code` is that the Fortran routine generated here will perform a l_2-norm (root mean square to be specific) on the function and returns a single real number. This routine can be used as a fast prototyping tool to create Independent Residual Evaluator routines. The following demonstrates an example of creating IRE for wave equation:

```
read "../FD.mpl": Clean_FD(): Make_FD():

grid_functions := {f}:

WaveEq := diff(f(t,x),t,t) = diff(f(t,x),x,x):

Gen_Res_Code(lhs(WaveEq)-rhs(WaveEq),input="c",proc_name="ire_wave");
```

Example 15: Fast Prototyping IRE Routines

A.6.4 Creating Piece-wise Residual Evaluator Routines

Similar to the generalization of `Gen_Eval_Code` to `A_Gen_Eval_Code` such that the procedure accepts a DDS such that the function can be evaluated on each portion of the discrete domain, here `A_Gen_Res_Code` extends the capability of previous procedure `Gen_Res_Code` to evaluate the l_2-norm of the residual that is specified by a DDS:

```
A_Gen_Res_Code(dds:DDS,input="d/c*",proc_name="my_res_proc");
```

This procedure is perhaps most useful to evaluate the norm of the residual of the PDE under study. The returned norm of the residual can be compared to a *threshold value* to determine if the PDE is numerically solved after applying the solver routine (or after certain number of iterations of the solver routines are applied). Note that this routine can also be used as an IRE generator. Example 14 can be used to demonstrate the use of this procedure, the difference is that the Fortran routine created by this procedure will return the l_2-norm of the laplace equation, and therefore can be useful if we are monitoring the norm of the residual and convergence of our numerical solver.

A.6.5 Creating Solver Routine: `A_Gen_Solve_Code`

As we discussed in Sec. A.2.2, for a given FDA of a PDE written in canonical form:

$$PDE = L(f) = 0 = L^h(f^h) \tag{A.78}$$

the solving process involves finding the unknowns f_{ijk} (for a boundary value problem) or f^{n+1}_{ijk} for initial value problem using f^n_{ijk}. As introduced in Sec. A.2.2, a standard approach for a nonlinear system is to use Newton-Gauss-Seidel iterative method. FD provides a procedure that generates routines which implement single iteration of this method:

```
A_Gen_Solve_Code(dds:DDS,{solve_for_var},input="d/c*",proc_name="my_solver_proc");
```

where the first argument is of type `DDS`, and the second argument is a set of unknowns for which the FDA must be solved. At the moment this set must contain only a single term, such as f(n+1,i,j,k) as the unknown. The created Fortran routine performs a single iteration of Newton-Gauss-Seidel and returns the "updated" function in the last argument, namely res which shall be adjusted by the user. This completes all the necessary tools to create a set of solver routines for a PDE, and in the next section we put together all of the features of FD discussed to demonstrate an implementation of a solver system for 1-D wave equation using an implicit scheme. This example also demonstrates the use of this solver procedure.

Note on myzero Expression

As it has been seen at several points in this document, the user needs to implement constant functions or residual equations by adding a trivial VDE such as myzero*x(i)*y(j). This is due to the fact that FD uses VDE's to figure out the dimensionality and dependencies of the PDEs, therefore if a single expression such as a constant number is given to FD's discretization routines, it has no way of finding the dimensionality of the problem. In particular, the common scenario that the use of myzero is essential is when in the equation that needs to be solved the solution simplifies to a single constant or zero. For instance, in Example 10 we are imposing fixed boundary condition $f = 0$ at $x = 0$, therefore the residual of the equation (LHS of equation in canonical form: $L(f) = 0$) is simply: f. However, the implementation of this residual has to be $f - myzero * x$, since if f is passed in as the residual, the solver VDE simplifies to 0, which has no valid dependency on any discrete index, {i,j,k,n}, to be understood by FD.

A.6.6 Communicating with Parallel Computing Infrastructure

Here we present a simple communication method with a parallelization infrastructure (FD adopts PAMR's [33] standard). To achieve this goal a vector of integer flags, phys_bdy is passed to the solver/evaluator routines in which the value 1 denotes that the boundary is a real physical boundary, therefore the boundary condition should be imposed, and the value 0 denotes that it is a boundary between CPUs and usually no calculation is required as the parallel frameworks often implement between CPU ghost cells for the distributed sub-domains. These flags are invoked by setting the variable `b` to their associated names `xmin,xmax,...` as noted in table .(A.62) The following example demonstrates a DDS that implements boundary flags:

```
ddsfWave := [
  { i=[2,Nx-1,1] , j=[2,Ny-1,1]          } =  PDEWave_D,
  { i=[1,1,1]    , j=[1,Ny,1]   , b=xmin } =  f(n+1,i,j) - myzero*x(i)*y(j),
  { i=[Nx,Nx,1]  , j=[1,Ny,1]   , b=xmax } =  f(n+1,i,j) - myzero*x(i)*y(j),
  { i=[1,Nx,1]   , j=[1,1,1]    , b=ymin } =  f(n+1,i,j) - myzero*x(i)*y(j),
  { i=[1,Nx,1]   , j=[Ny,Ny,1]  , b=ymax } =  f(n+1,i,j) - myzero*x(i)*y(j)
];
```

We encourage the reader to look into the Fortran files that are created using this type of DDS to inspect how the `phys_bdy` flags are positioned in the file.

The tutorial: `FD/tutorials/wave2d_pamr_fixed_boundary` in the distribution package is an implementation of a parallel 2 dimensional wave equation solver.

A.6.7 Example: Crank-Nicolson Implementation of Wave Equation

We complete this section by combining all of the tools we discussed to a single Maple script that creates a solver routine, residual evaluator and an independent residual evaluator as well as an initializer routine for the 1 dimensional wave equation. The wave equation is given by:

$$\partial_t^2 f(t,x) = \partial_x^2 f(t,x)\,, \tag{A.79}$$

and can be reduced to a first order system by defining f_t as:

$$f_t(t,x) \equiv \partial_t f(t,x) \tag{A.80}$$

$$\Rightarrow$$

$$\partial_t f(t,x) = f_t(t,x) \tag{A.81}$$

$$\partial_t f_t(t,x) = \partial_x^2 f(t,x) \tag{A.82}$$

Here we assume periodic boundary conditions Note that the example, first implements an IRE for the system using the original form of the wave equation and FD's default second order leap-frog scheme. After that, the FD scheme is updated to forwards in time, and by virtue of time averaging we achieve second order accuracy. It also demonstrate how to create initializer routines as well as residual evaluator routines to measure how accurate the PDE is solved.

```
read "../FD.mpl": Clean_FD(); Make_FD();
grid_functions := {f,f_t};

eq1 := diff(f(t,x),t) = f_t(t,x);
eq2 := diff(f_t(t,x),t) = diff(f(t,x),x,x);
eq3 := diff(f(t,x),t,t) = diff(f(t,x),x,x);

Gen_Res_Code(lhs(eq3)-rhs(eq3),input="c",proc_name="ire_f");

FD_table[t] := [[0],[0,1]];

AVGT := a -> (   FD( a,[ [1],[0] ]) + FD( a,[ [0],[0] ]) )/2;

eq1_D :=   Gen_Sten(lhs(eq1)) - AVGT(Gen_Sten(rhs(eq1)));
eq2_D :=   Gen_Sten(lhs(eq2)) - AVGT(Gen_Sten(rhs(eq2)));

init_f:=A*exp(-(x-x0)^2/delx^2);
init_f_t:=idsignum*diff(init_f,x);

Gen_Eval_Code(init_f,input="c",proc_name="init_f");
Gen_Eval_Code(init_f_t,input="c",proc_name="init_f_t");
```

```
dss_f:= [
  { i=[1,1,1]     } = FD_Periodic(eq1_D,{i=1}) ,
  { i=[2,Nx-1,1]  } = eq1_D,
  { i=[Nx,Nx,1]   } = FD_Periodic(eq1_D,{i=Nx})
  ];

dss_f_t:= [
  { i=[1,1,1]     } = FD_Periodic(eq2_D,{i=1}) ,
  { i=[2,Nx-1,1]  } = eq2_D,
  { i=[Nx,Nx,1]   } = FD_Periodic(eq2_D,{i=Nx})
];

A_Gen_Res_Code(dss_f,input="d",proc_name="res_f",is_periodic=true);
A_Gen_Res_Code(dss_f_t,input="d",proc_name="res_f_t",is_periodic=true);

A_Gen_Solve_Code(dss_f,{f(n+1,i)},input="d",proc_name="u_f",is_periodic=true);
A_Gen_Solve_Code(dss_f_t,{f_t(n+1,i)},input="d",proc_name="u_f_t",is_periodic=true);
```

Example 15: Implementation of Crank-Nicolson Scheme to Solve 1D Wave Eq. (A.79)

Note that several other complete examples are included in FD's distribution package in the directory `tutorials`, including: 2D wave equation in parallel, non-linear mixed boundary 1D wave equation, heat equation, and 2D wave equation in cylindrical coordinate with axial symmetry. All of the examples in this manual are also included in the distribution under `examples` directory.

Also see: `http://laplace.phas.ubc.ca/People/arman/FD_doc/tutorials.html` for detailed tutorials on how to use FD.

A.7 List of Abbreviations

BVE: Boundary Value Problem

DDS: Discrete Domain Specifier

FD: Finite Difference, also the name of the toolkit

FDA: Finite Difference Approximation

FDE: Finite Difference Equation

FDM: Finite Difference Molecule

FDS: Finite Difference Specifier

IVE: Initial Value Problem

LHS: Left Hand Side

ODE: Ordinary Differential Equation

PBC: Periodic Boundary Condition **PDE**: Partial Differential Equation

RHS: Right Hand Side

VCE: Valid Continuous Expression

VDE: Valid Discrete Expression