

A Numerical Study of Boson Star Binaries

by

Bruno Coutinho Mundim

B.Sc., University of Brasilia, 1998

M.Sc., University of Brasilia, 2002

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

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Physics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

February 2010

© Bruno Coutinho Mundim 2010

Abstract

This thesis describes a numerical study of binary boson stars within the context of an approximation to general relativity. Boson stars, which are static, gravitationally bound configurations of a massive complex scalar field, can be made gravitationally compact. Astrophysically, the study of gravitationally compact binaries—in which each constituent is either a neutron star or a black hole—and especially the merger of the constituents that generically results from gravitational wave emission, continues to be of great interest. Such mergers are among the most energetic phenomena thought to occur in our universe. They typically emit copious amounts of gravitational radiation, and are thus excellent candidates for early detection by current and future gravitational wave experiments.

The approximation we adopt places certain restrictions on the dynamical variables of general relativity (conformal flatness of the 3-metric), and on the time-slicing of the spacetime (maximal slicing), and has been previously used in the simulation of neutron stars mergers. The resulting modeling problem requires the solution of a coupled nonlinear system of 4 hyperbolic, and 5 elliptic partial differential equations (PDEs) in three space dimensions and time. We approximately solve this system as an initial-boundary value problem, using finite difference techniques and well known, computationally efficient numerical algorithms such as the multigrid method in the case of the elliptic equations. Careful attention is paid to the issue of code validation, and a key part of the thesis is the demonstration that, as the basic scale of finite difference discretization is reduced, our numerical code generates results that converge to a solution of the continuum system of PDEs as desired.

The thesis concludes with a discussion of results from some initial explorations of the orbital dynamics of boson star binaries. In particular, we describe calculations in which motion of such a binary is followed for more than two orbital periods, which is a significant advance over previous studies. We also present results from computations in which the boson stars merge, and where there is evidence for black hole formation.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Tables	vi
List of Figures	vii
List of Abbreviations	ix
Acknowledgements	x
1 Introduction	1
1.1 Overview of Numerical Simulations of Compact Binaries	7
1.1.1 Black Hole Binaries	7
1.1.2 Neutron Star Binaries	10
1.2 An Overview of Boson Stars	10
1.3 Summary of Thesis	12
1.4 Conventions, Notation and Units	14
2 Formalism and Equations of Motion	16
2.1 The Field Equations of General Relativity	17
2.2 The 3 + 1 (ADM) Formalism	21
2.3 The Complex Scalar Field	29
2.3.1 Einstein-Klein-Gordon System	30
2.3.2 Noether Charge	34
2.4 The Conformally Flat Approximation (CFA)	38
2.4.1 Maximal Slicing	38

2.4.2	Conformal Decomposition of the Einstein Equations	39
2.4.3	The CFA Equations	51
2.4.4	Specialization to Cartesian Coordinates	56
2.4.5	Boundary Conditions	59
2.4.6	ADM/York Mass	70
2.5	Overview of the Equations of Motion	72
3	Initial Data	75
3.1	Spherically Symmetric Spacetime	76
3.2	Maximal-Isotropic Coordinates	78
3.3	Polar-Areal Coordinates	81
3.4	Constructing Boson Stars: The Static Ansatz	84
3.5	Boson Stars: Some Basic Properties	92
3.6	Boson Stars: Representation in Cartesian Coordinates	96
3.7	Boson Stars: Applying Approximate Lorentz Boosts	98
4	Numerical Techniques	107
4.1	Discretization of Partial Differential Equations	108
4.1.1	Basic Concepts Related to FDAs	112
4.1.2	Richardson Expansions	116
4.1.3	The Crank-Nicholson Discretization Scheme	117
4.2	Convergence Testing of Finite Difference Solutions	122
4.2.1	Convergence Factor	122
4.2.2	Independent Residual Evaluation	124
4.3	Multigrid Techniques	129
4.3.1	Introduction	129
4.3.2	The FAS Algorithm	132
5	Code Validation and Results	139
5.1	Summary of the Numerical Code	140
5.2	Generic Initial Data	146
5.3	Static Spherically Symmetric Initial Data - One Star	155
5.4	Single Boosted Boson Star	172

5.5	Head-on Collision of Two Boson Stars	179
5.6	Orbital Dynamics of Two Boson Stars	184
5.6.1	Case 1: $v_x = 0.09$ —Long Lived Orbital Motion	185
5.6.2	Case 2: $v_x = 0.07$ —Formation of Pulsating & Rotating Boson Star	186
5.6.3	Case 3: $v_x = 0.05$ —Formation of a Black Hole	186
5.7	Discussions and Further Developments	200
6	Conclusion	203
	Bibliography	205

Appendices

A	BSIDPA—Boson Star Initial Data Function	219
B	Finite Difference Approximations	225
B.1	Finite Difference Operators	225
B.2	FDAs for Operators of the Form $\partial/\partial(r^p)$	231
C	PDEFDAOFF: A MAPLE Package for FDAS	234
C.1	FDAs of n -dimensional Differential Operators	234
C.1.1	1-dimensional Examples	235
C.1.2	3-dimensional Example	238
C.2	Residual Evaluators for n -dimensional PDEs	238
D	Review of Relaxation Methods	241
D.1	The Linear Case: Jacobi and Gauss-Seidel Relaxation	241
D.2	The Nonlinear Case: Newton-Gauss-Seidel Relaxation	246

List of Tables

2.1	Equations of Motion.	55
A.1	Central Scalar Field Values and Eigenfrequencies ω for Mini Boson Stars.	223
A.2	Mini Boson Star Properties: Tail, Mass and Radius.	224
A.3	Mini Boson Star Properties: Extrema of Metric Components.	224
B.1	Centred FDAs of First Order Derivatives.	227
B.2	Time Averaging Operator.	227
B.3	Centred FDAs of Second Order Derivatives.	227
B.4	Centred FDAs of Mixed Second Order Derivatives.	228
B.5	Forward FDAs of First Order Derivatives.	228
B.6	Backward FDAs of First Order Derivatives.	228
B.7	Forward FDAs of Second Order Derivatives.	228
B.8	Backward FDAs of Second Order Derivatives.	229
B.9	Forward-Forward FDAs of Mixed Second Order Derivatives.	229
B.10	Forward-Backward FDAs of Mixed Second Order Derivatives.	229
B.11	Backward-Forward FDAs of Mixed Second Order Derivatives.	230
B.12	Backward-Backward FDAs of Mixed Second Order Derivatives.	230

List of Figures

3.1	Typical Boson Star Solutions.	94
3.2	ADM Mass M_{ADM} as a Function of Central Scalar Field Value $\phi_0(0)$ and the Star Radius R	95
4.1	3D Grid Sample.	111
4.2	Stencil for Second Spatial Derivative Operator	112
4.3	Crank-Nicholson Stencil.	119
4.4	A Pseudo-Code Representation of the FAS V -Cycle Multigrid Algorithm.	138
5.1	A Pseudo-Code Representation of the Numerical Code.	143
5.2	Time Evolution of a Generic Gaussian Profile.	150
5.3	Generic Initial Data: $Q^h(t)$ and $\ I(t)\ _2$ for α , ψ and β^x	151
5.4	Generic Initial Data: $Q^h(t)$ and $\ I(t)\ _2$ for β^y , β^z and ϕ_1	152
5.5	Generic Initial Data: $Q^h(t)$ and $\ I(t)\ _2$ for ϕ_2 , Π_1 and Π_2	153
5.6	ADM Mass $M_{\text{ADM}}(t)$ and Noether Charge $Q_N(t)$ for the Generic Initial Data.	154
5.7	Rescaled ADM Mass and Noether Charge Deviations for Generic Initial Data.	154
5.8	Time Evolution of the Static Boson Star, $S_{0.03}$	159
5.9	Time Evolution of the Static Boson Star, $S_{0.06}$	160
5.10	$S_{0.03}$ and $S_{0.06}$ Convergence Factors $Q^h(t)$ for α and ψ	161
5.11	$S_{0.03}$ and $S_{0.06}$ Convergence Factors $Q^h(t)$ for β^i	162
5.12	$S_{0.03}$ and $S_{0.06}$ Convergence Factors $Q^h(t)$ for ϕ_A and Π_A	163
5.13	$S_{0.06}$ and $S_{0.03}$ Independent Residual l_2 -Norms, $\ I(t)\ _2$, for α , ψ and β^x	164
5.14	$S_{0.06}$ and $S_{0.03}$ Independent Residual l_2 -Norms, $\ I(t)\ _2$, for β^y , β^z and ϕ_1	165
5.15	$S_{0.06}$ and $S_{0.03}$ Independent Residual l_2 -Norms, $\ I(t)\ _2$, for ϕ_2 , Π_1 and Π_2	166
5.16	$M_{\text{ADM}}(t)$ and $Q_N(t)$ for $S_{0.06}$ and $S_{0.03}$	167
5.17	Rescaled Deviations of ADM Mass and Noether Charge for $S_{0.06}$ and $S_{0.03}$	168

5.18	Magnified View of $ \phi(3.0, x, y, 0) $ — from $S_{0.03}$ Evolution.	169
5.19	Magnified View of $ \phi(3.0, x, y, 0) $ — from $S_{0.06}$ Evolution.	170
5.20	$\ \beta^x(t)\ _2$, $\ \beta^y(t)\ _2$ and $\ \beta^z(t)\ _2$ for $S_{0.06}$	171
5.21	Time Evolution of a Single Boosted $S_{0.06}$ Star.	174
5.22	Single Boosted $S_{0.06}$ Star: $Q^h(t)$ and $\ I(t)\ _2$ for α , ψ and β^x	175
5.23	Single Boosted $S_{0.06}$ Star: $Q^h(t)$ and $\ I(t)\ _2$ for β^y , β^z and ϕ_1	176
5.24	Single Boosted $S_{0.06}$ Star: $Q^h(t)$ and $\ I(t)\ _2$ for ϕ_2 , Π_1 and Π_2	177
5.25	ADM Mass $M_{\text{ADM}}(t)$ and Noether Charge $Q_N(t)$ for Boosted $S_{0.06}$ Star.	178
5.26	ADM Mass and Noether Charge Deviations for Single Boosted $S_{0.06}$ Star.	178
5.27	Head-on Collision of Boson Stars Solitonic Dynamics.	181
5.28	Head-on Collision of Boson Stars: $\ I_{\phi_1}(t)\ _2$	182
5.29	Head-on Collision of Boson Stars: $y \rightarrow -y$ Symmetry Assessment.	183
5.30	Long Lived Orbital Motion: Time Evolution of $ \phi(t, x, y, 0) $	188
5.31	Long Lived Orbital Motion: Trajectories.	189
5.32	Long Lived Orbital Motion: $\ I_{\phi_1}(t)\ _2$	190
5.33	Long Lived Orbital Motion: Rescaled $\ I_{\phi_1}(t)\ _2$	191
5.34	Long Lived Orbital Motion: $\ I_\alpha(t)\ _2$	192
5.35	Long Lived Orbital Motion: Rescaled $\ I_\alpha(t)\ _2$	193
5.36	Formation of Pulsating and Rotating Boson Star: Time Evolution Snapshots.	194
5.37	Formation of Pulsating and Rotating Boson Star: Trajectories.	195
5.38	Formation of Pulsating and Rotating Boson Star: $\ I_\alpha(t)\ _2$	196
5.39	Formation of a Black Hole: Time Evolution Snapshots.	197
5.40	Formation of a Black Hole: Trajectories.	198
5.41	Formation of a Black Hole: $\ I_\alpha(t)\ _2$	199

List of Abbreviations

ADM: refers to the authors of the formalism, R. Arnowitt, S. Deser and C. W. Misner.

AMR: Adaptive Mesh Refinement.

CFA: Conformally Flat Approximation.

EKG: Einstein-Klein-Gordon System of Equations.

FDA: Finite Difference Approximation.

FDE: Finite Difference Equation.

NGS: Newton-Gauss-Seidel Iterative Technique.

ODE: Ordinary Differential Equation.

PDE: Partial Differential Equation.

Acknowledgements

First and foremost I would like to express my profound gratitude for my supervisor, Matthew W. Choptuik. Thank you very much for all your support and wise guidance over the years this project took place. Second, I would like to thank all the members of my graduate committee: William G. Unruh, Jeremy S. Heyl and Ingrid Stairs for all their input on this project and for asking all those hard questions that made me think deeper about physics in general and general relativity in particular. Also I would like to thank Bill Unruh for fostering interesting discussions taking place in the gravity group meeting. I enjoyed being part of it and certainly will miss it a lot.

I should mention that this research was supported by NSERC and CIFAR as well as MPI-AEI. I would like to thank Gerhard Huisken and the Albert Einstein Institute for their hospitality and support during the time that I spent there.

It was great being a member of the numerical relativity group at UBC. I learned a lot over the years with the innumerable interactions among different members of the group. In particular, I would like to thank the members of the “old crew”: Iñaki Olabarrieta and Kevin Lai for their patience in answering all my questions when I first started. My thanks goes to Roland Stevenson as well for sharing his programming expertise. Special thanks for the members of the “new crew”, Jason Penner and Benjamin Gutierrez, for their promptness to help on the most diverse issues. Thank you guys for all your friendship.

I would like to thank my friends and colleagues from UBC for all their friendship and for making my stay at UBC so vibrant. Special thanks for my friend Sanaz Vafaei for all her support during the writing of this manuscript. I am deeply indebted to several of friends in Brazil that have actively encouraged me to pursue my dreams.

I am grateful to my wife, Janine Kurtz, for all her dedication, love, companionship and patience during my final steps to conclude this project. Thanks for helping me to get through this difficult time.

I wish to thank my entire family for providing me a loving and nurturing environment. Without

their support and sacrifices during my education, I would not have made it so far. I would like to dedicate this thesis to my parents, Elson R. Mundim and Selmara C. Mundim, for all their love, support and encouragement during my entire life.

Chapter 1

Introduction

One of the most revolutionary predictions of general relativity—Einstein’s spectacularly successful theory of the gravitational interaction—is the existence of gravitational waves. Binary systems composed of gravitationally compact objects¹, such as two black holes, two neutron stars or one neutron star and one black hole, are among the most promising sources of these waves. According to Einstein’s theory, such binaries have orbits which decay due to the emission of gravitational radiation. This decay is thought to ultimately result in a violent “plunge and merger” of the two objects, characterized by an extremely strong and dynamical gravitational field, and the emission of an especially strong burst of gravitational radiation. An international network of laser interferometer detectors (e.g. LIGO [1], VIRGO [2], TAMA300 [3] and GEO600 [4]) has been constructed to detect these bursts (as well as other sources), where the detection sensitivity is best for compact objects having masses of the order 1 to 10 times that of the sun. However, even for such strong sources, the terrestrial signal strength from a typical event (not likely in our own galaxy, or even in our local group of galaxies, on the time scale of years) is expected to be much less than the inherent noise in the instruments. Faced with this situation, the most promising technique to extract the signals from the noise is matched filtering [5], which involves comparison of the measured signal against a known (precomputed) family of waveforms. Construction of such template waveforms is thus an urgent problem, and it requires *accurate* theoretical modelling of the process of compact binary inspiral.

Due to the complexity and nonlinearity of the Einstein field equations for the relativistic gravitational field—to which one must add the governing equations for any matter fields which are involved (such as a perfect fluid in the case of neutron stars)—accurate modelling of the late phases of binary inspiral and merger requires a numerical approach. Now, as we will discuss in more detail shortly (Sec. 1.1), the computer simulation of strongly interacting compact binaries

¹An object with mass, M , is gravitationally compact if its radius, R , is close to its Schwarzschild radius, R_S , defined by $R_S = 2GM/c^2$, where G is Newton’s gravitational constant, and c is the speed of light. For (spherical) black holes $R = R_S$, whereas for neutron stars R_S/R is typically in the range 0.04–0.27.

has seen tremendous advances over the past few years. This is especially true for the case of the inspiral and collision of two black holes, which although a formidable problem, is simplified relative to its neutron star counterpart by the fact that it does not involve matter fields. For neutron stars, the need to solve the equations of general relativistic hydrodynamics in concert with the Einstein equations leads to a host of other difficulties, including the need to deal with shocks, turbulence and uncertainties concerning the equation of state. Moreover, the parameter space describing the generic collision of two compact object has *many* dimensions, so that even with improved simulation techniques, identifying and extracting the key physics from the calculations will present a huge computational challenge for years to come. Here we should emphasize that calculations of interacting binaries must be done in three spatial dimensions and time, so that even a single calculation that is adequately resolved requires the use of high performance computing facilities.

Given this state of affairs there is considerable motivation to look for simplified “toy models”, with the hope and expectation that they can provide insight into aspects of the compact binary problem, especially for the neutron star case. We have already noted that the plunge and merger phase of neutron stars in inspiral is characterized by a strong and dynamical gravitational field. At least heuristically, the dynamics is dominated by the bulk motion of the two stars, so that localized features in the matter—such as individual shocks, or small-scale turbulence—should have relatively little impact on the overall dynamics, or on the gravitational radiation which is produced. We thus look for a matter model which can describe gravitationally compact objects, but where the equations of motion for the matter are easier to treat computationally than those for a relativistic fluid.

The studies in this thesis involve precisely such a matter model. Specifically, we adopt a single, massive, complex scalar field as a matter source. The model admits localized, time-independent, gravitationally-bound configurations known as *boson stars*, which, through an appropriate choice of parameters, *can* be made compact. We can also set up initial data representing two stars, with subsequent evolution describing a variety of different kinds of encounters. The equation of motion for the scalar field is simply the general relativistic wave equation, or Klein-Gordon equation, whose numerical solution using finite difference techniques is quite straightforward.

Moreover, in this thesis we take the “toy model” approach one step further. Although it is certainly possible to simulate boson stars in a fully general relativistic setting [6, 7, 8, 9, 10], we opt to study them within the context of a relatively simple approximation to general relativity

which has previously been used to study strong-gravity effects in several scenarios of astrophysical interest, including the interaction of neutron stars.

More specifically, and as will be discussed in detail in Chap. 2, this approximation is based on the *conformally flat condition*, or CFC, for the spatial part of the metric (which is the fundamental dynamical variable that describes the gravitational field), along with the *maximal slicing condition*, which fixes the time coordinatization of the spacetime. Although the literature has traditionally used the acronym CFC for the resulting approximation to Einsteinian gravity, we prefer to use CFA, for *conformally flat approximation*. This stresses the fact that we *are* dealing with an approximation to general relativity. Also, we emphasize that although neither acronym makes explicit reference to the maximal slicing condition, that choice of time coordinatization has always been an essential ingredient of the approximation, and is so here.

The CFA is based on the heuristic assumption that the dynamical degrees of freedom of the gravitational field, i.e. those associated with gravitational radiation, play a small role in at least some phases of the strong field interaction of a merging binary, and on the related fact that the amount of energy contained in the waves, expressed as a fraction of the total mass-energy of the system, tends to be small. The CFA effectively eliminates the two dynamical degrees of freedom present in general relativity, but still allows for investigation of the same kinds of phenomena observed in the full general relativistic case. These include the description of compact objects, the dynamics of their interaction, and black hole formation. It is also worth mentioning that it is still possible to study gravitational wave generation within this approximation via a perturbative multipole expansion of the metric components. Briefly, the incorporation of radiation effects, although far from a trivial matter, can be realized through the introduction of a radiation reaction potential in the equations of motion for the matter model. We also note that for spherically symmetric systems the CFA is *not* an approximation, but can always be adopted through an appropriate choice of coordinates. Furthermore, use of the approximation for axially symmetric problems has indicated that the results obtained mimic those of general relativity quite accurately [11, 12, 13, 14]. There is thus considerable motivation to perform additional studies of strong-field gravity using this approach.

The CFA was first studied in a theoretical/mathematical context by Isenberg in the 1970's [15], and applied numerically (and independently) by Wilson *et al.* [16, 17, 18, 19, 20, 21, 22, 23] in the 1990's to the study of coalescing neutron-star binaries. In this prior work, Wilson and

his collaborators presented evidence that, for a realistic neutron-star equation of state, general relativistic effects might cause the stars to individually collapse to black holes prior to merging. Furthermore, they observed that at least for some set of initial orbital parameters, strong-field effects caused the last innermost stable circular orbit, or ISCO, to occur at a larger separation distance, and thus at lower frequency, than was previously estimated by post-Newtonian methods. This result had significance for the possible detection of gravitational waves, since it placed the frequency of radiation from coalescence closer to the maximum sensitivity range of current laser-interferometric detectors. However, the Wilson-Mathews compression effect was unexpected and controversial, and raised questions concerning the validity of the CFA.

Subsequently, Flanagan [24] identified an inconsistency in the derivation of some of the equations of motion used in this study, and suggested that use of the correct equations would reduce the crushing effect. A revised version of simulations was published shortly thereafter [25]: a key claim resulting from this work was that the crushing effect was still present, although the magnitude of the observed effect *was* reduced relative to the previous calculations. Further comparisons between a fully relativistic code and its CFA counterpart in the context of head-on collisions of neutron stars showed the presence of this effect and lent more credibility to the earlier calculations [26]. However, the most recent simulations [27] aimed at studying the possible crushing phenomenon actually indicate a *decompressing* effect on the neutron stars. Still, this result is not in direct contradiction to Wilson *et al's* results since the initial data used for the two sets of simulations differ. For a more complete review of the history of this controversy, as well as a possible explanation for the neutron star crushing effect, the reader should refer to the work by Favata [28] and references therein. We should also note at this point that since this thesis work was started, new “waveless” formalisms have been developed [29, 30], and have been used to improve the accuracy of certain compact binary calculations [31] in the inspiral phase, relative to fully relativistic computations.

An ultimate goal of the work started in this thesis is to determine to what extent the CFA *is* a good approximation for the modelling of *general* compact binaries. From this point of view, it is particularly interesting to study the CFA within the context of a simpler matter model than that previously adopted, and this provides an important motivation for our use of boson stars. Some of the main questions we wished to address are as follows:

- Would boson stars collapse individually before merger, or is this phenomena strongly dependent on how the matter is modelled? Favata [28] posited a mechanism that would tend to

compress the neutron stars given a particular set of conditions. Does that analysis apply to the binary boson star as well?

- How well does the CFA approximate the full general relativistic equations? Can we shed some light on the nature of the radiative degrees of freedom in the full theory from a detailed study of the differences between results obtained using the CFA and the full Einstein equations?
- What is the final result of the merger? Can we compare the results to those obtained from other techniques, including fully general relativistic calculations?
- Where is the ISCO? Do the results obtained match those seen in collisions of fluid stars, or can they at least be compared qualitatively?

In order to answer these and related questions, the Einstein equations (and ultimately the equations that result from adopting the CFA) have to be cast in a form suitable for numerical computation. What follows is a brief description of our modelling procedure; more details will be provided in subsequent chapters.

The current work makes use of the $3 + 1$ or ADM (Arnowitt-Deser-Misner) [32] decomposition of the Einstein equations. One of the main features of the $3 + 1$ approach, which underlies the majority of the work in numerical relativity², is that it provides a prescription for disentangling the dynamical field variables from those associated with the coordinate invariance of general relativity. Specifically, the spacetime metric components are grouped into 4 kinematical components which encode the coordinate freedom of the theory (the *lapse function* and three components of the *shift vector*), and 6 dynamical ones (the components of a 3-dimensional metric induced in a constant-time hypersurface). Now, for any specific calculation, the coordinate system must be completely fixed by giving prescriptions for the lapse and shift. In particular, the time coordinate is fixed by a choice of the lapse function, and, as already mentioned, in this work we follow previous studies using the CFA and adopt so-called maximal slicing (a very commonly adopted slicing condition in numerical relativity that was originally proposed by Lichnerowicz [33]). A key property of maximal slicing is that its use inhibits the focusing of the world lines of observers that move orthogonally to the hypersurfaces: such focusing can result in the development of coordinate singularities, and is also associated with the formation of physical singularities. Another important point is that this coordinate choice generally results in a well-posed elliptic equation for the lapse function.

²Numerical relativity can be defined as the subfield of general relativity that is concerned with the solution of the Einstein equations, as well as the equations of motion for any matter sources, using computational methods.

As the name suggests, the CFA requires the spatial 3-geometry to be conformally flat. This reduces the number of independent components of the spatial metric from 6 to 1, and the single, non-trivial function that then defines the spatial metric is called the *conformal factor*. Given the maximal slicing condition, it transpires that conformal flatness implies that all non-trivial components of the 4-metric are governed by elliptic equations.

Within the CFA, the dynamics of the gravitational field is completely determined by the dynamics of the matter source(s), which in our case is a complex scalar field that satisfies the Klein-Gordon equation. Overall then, the model considered here—which we will hereafter frequently refer to as “our model”—is governed by a mixed system of 4 first-order hyperbolic partial differential equations (PDEs), coming from the complex Klein-Gordon equation, and 5 elliptic PDEs. We treat this system as an initial-boundary value problem and the thesis focuses on the development, testing and preliminary use of a code for its numerical solution. Our computational approach is based on finite differencing [34], and we use approximations that are second-order accurate in the grid spacing.

A numerical evolution of our model starts with the specification of initial conditions for the complex scalar field. For example, for the relatively simple case of two identical boson stars that are initially at rest, the solution of the coupled Einstein/scalar field equations in spherical symmetry, and with a time-periodic ansatz for the scalar field, provides a static star profile that can be duplicated and interpolated onto a three dimensional domain. This sets initial values representing a binary system. Discrete versions of the elliptic equations governing the geometric quantities are then solved at the initial time using the multigrid method [35]. The matter field values are then time advanced using a second order discretization of the Klein-Gordon equation, which, along with the elliptic equations for the advanced-time metric unknowns, is solved iteratively.

Our code has been subjected to a thorough series of tests that assess convergence of the numerical results with respect to the basic discretization scale. Additional evidence for code validity involves the use of conserved quantities, as well as the technique of independent residual evaluation (which is defined and discussed in Chap. 4). Overall, results from these tests indicate that the code is correctly solving the equations of motion that define the model. Specific results calculated using the code include long-term evolution of an orbiting binary system, (more than two orbital periods), as well as high speed head-on collisions. These results are promising and suggest that, especially with enhancements such as the incorporation of adaptive mesh refinement and capabili-

ties for parallel execution, the code will be a powerful tool for investigating strong-gravity effects in the interaction of boson stars.

We now proceed to two short overviews of subjects that are germane to this work. First, in Sec. 1.1, the current status of the numerical modelling of gravitationally-compact binary systems is summarized. Second, some of the key previous studies of boson stars are reviewed in Sec. 1.2.

1.1 Overview of Numerical Simulations of Compact Binaries

1.1.1 Black Hole Binaries

As mentioned previously, accurate modelling of compact star binaries is needed to help extract the gravitational wave signal from the noise in current and planned terrestrial gravitational wave interferometers. This section briefly describes the status of numerical relativistic modelling of these binaries by means of a quick survey of the pertinent literature. We first note that there are several good review papers on the general subject, including the following:

- Rasio & Shapiro’s review of coalescing binary neutron stars [36]. This covers work up to 1999 on coalescing binary neutron stars—using Newtonian, post-Newtonian and semi-relativistic approximations—and is highly recommended for familiarizing the interested reader with earlier research in the field.
- Baumgarte & Shapiro’s extensive review of the use of numerical relativity to model compact binaries [37].
- Pretorius’ more recent review on black hole collisions [38].

Efforts to accurately model the collision of compact objects using fully general relativistic calculations began with the work on black hole head-on collisions by Smarr and collaborators in the 1970’s [39]. These calculations established numerical relativity as a sub-field of general relativity in its own right, and indicated that one could expect perhaps a few percent of the total mass-energy of the system to be emitted in gravitational waves. In the early 1990’s, Anninos *et al.* [40] revisited and extended the head-on calculations. A major result from this effort was the demonstration that the extracted gravitational waveform could be well-matched to (perturbative)

black hole normal modes. In 1993 the Binary Black Hole Grand Challenge Alliance, involving many investigators from several different institutions in the US, was founded with the mission to provide stable, convergent algorithms to compute the gravitational waveforms from black-hole collisions. Although this research spurred the development of computational infrastructure that is still being used by the numerical relativity community, the project ended well short of its stated physics goals. The most important simulations that resulted from the 5-year effort described the propagation of a single black hole through a three dimensional computational domain [41]. Nonetheless, this result constituted an important step towards the solution of the binary black hole problem, especially when contrasted with previous computations where the black-holes were kept fixed in the domain by specific choices of coordinates. It was not until 2004 that a significant amount of orbital motion for a black hole binary was successfully simulated, when Brügmann and collaborators published results [42] from computations of a full orbital period for a widely-separated black hole pair.

2005 then saw the publication of breakthrough results by Pretorius [43, 44], and the beginning of the most productive period of binary black hole research. Pretorius presented calculations that tracked the evolution of a binary black hole spacetime through several orbits, the plunge and merger phase, and into the late-time stage where the final black hole quickly rings down to a near-stationary (Kerr) configuration. Furthermore, Pretorius was able to extract a reasonably accurate gravitational wave signal from the whole evolution. Pretorius’ work had an electrifying effect on research in the field, and other groups, most notably the University of Texas at Brownsville (UTB) [45, 46, 47] and NASA/Goddard [48, 49] teams, were quickly able to produce comparable results using similar [50] or different techniques (moving punctures) [48, 49, 45, 46, 47, 51, 52]. Moreover, it was possible to compare the various simulations and to show that the results were generally consistent to within the level of numerical error [53].

Since that time, a large number of distinct simulations have been performed, and some interesting phenomenology has been unearthed. Particularly noteworthy are the “kicks” that the final black holes formed from mergers can experience due to asymmetric emission of gravitational radiation during the coalescence [54, 55, 56, 57]. When the merger process is itself sufficiently asymmetric—as is the case for the inspiralling collision of two holes with unequal masses—the gravitational waves produced carry away net linear momentum. This imparts an equal and opposite momentum, or kick, to the final black hole, and the magnitude of the resulting recoil can have astrophysical implications. For example, depending on the mass of the stellar cluster in which

the binary is embedded, the merged black hole may actually be ejected from the cluster or even the host galaxy. Early calculations concerning these kicks—which used non-spinning black holes (no intrinsic angular momentum)—have now been extended to investigate the effects of the black hole spins on the recoil velocity of the final black hole [58, 59, 60, 61, 62, 63]. These studies have indicated that impressively large kicks ($\sim 4000\text{km/s}$) can result from misalignment of the black holes’ intrinsic spins and the orbital angular momentum, as well as from large initial spins. In addition, pioneering studies by the UTB/Rochester Institute of Technology group [64, 65, 66, 67] have shown that the black holes’ initial spins can have a significant impact on the orbital dynamics, as well as on the emitted gravitational waveform at late times.

Another interesting effect that is still being studied concerns the effect of orbital eccentricity on the gravitational radiation waveforms; summaries of recent progress on this problem are given in [54, 68, 69, 70, 71]. Moreover, comparison of the waveforms resulting from numerical simulations on the one hand, and post-Newtonian calculations on the other, is also a rapidly developing industry [72, 73, 74, 75, 76]. Especially notable in this regard are the very high accuracy simulations, recently reported by the Cornell-Caltech group, that use pseudo-spectral techniques [77, 78]. Gravitational waveforms from some of these studies have now been compared to those calculated from post-Newtonian techniques [79]. These comparisons provide conclusive evidence that the best numerical results can now provide more accurate waveforms for the late phases of inspiral than those computed using post-Newtonian methods.

Finally, it is worth mentioning two studies of relativistic (high speed) black hole collisions [80, 81]. This type of interaction is of particular current interest since it ties in with the phenomenology of possible mini-black hole formation in new generations of particle accelerators such as the Large Hadron Collider at CERN. Moreover, ultrarelativistic black-hole collisions represent largely unexplored territory where fundamental theoretical issues might be addressed. These include the possibilities of naked singularity formation, as well as the production of an end state containing three or more black holes, starting from a two black hole initial state. Although the preliminary results show no evidence for naked singularities, they *do* provide explicit demonstration of the copious amount of gravitational radiation that can be emitted from such encounters. Specifically, in some of the cases simulated, as much as 25% of the initial rest-mass energy is emitted as gravitational waves. This approaches the upper bound of 29% estimated by Penrose [82, 83].

1.1.2 Neutron Star Binaries

The first fully relativistic simulation of binary neutron star inspiral and merger was performed by Shibata and Uryū in 2000 [84]. Adopting a simple equation of state, they examined the issue of the nature of the end state from the coalescence of two equal-mass stars. Perhaps not surprisingly, their work indicated that the final configuration depended on the compactness of the initial neutron stars: more compact stars would lead to black hole formation, while those less condensed would merge to form a high-mass, differentially rotating neutron star. In 2002, subsequent work by the same authors, using increased resolution in the simulations, produced gravitational waveform estimates for the mergers [85]. Additional research by Shibata and his collaborators [86] has focused on unequal mass binaries, as well as on the use of several different approaches to mimic realistic equations of state [87, 88].

Work by other groups has followed, and there have now been several reports of stable long term simulations that extend over a few inspiraling orbits through to merger (the end state is typically a black hole) [89, 90, 91, 92]. It is worth noting that in all of the work mentioned this far, effects due to magnetic fields were neglected. However, within the past year or so, preliminary results involving magnetized neutron star binaries have started to appear [93, 94]. Finally, three dimensional general relativistic simulations of mixed binaries—where one constituent is a black hole and the other is a neutron-star—are also underway; see [95, 96, 97, 98, 99] and references therein.

1.2 An Overview of Boson Stars

The concept of a boson star can be traced back to work by Wheeler and collaborators [100, 101] who considered self-gravitating “lumps” of massless fields (electromagnetic or gravitational), which they dubbed *geons*. Unfortunately such configurations were soon found to be unstable and research on them quickly subsided. However, the geon idea provided impetus for the investigation of scalar fields in the context of general relativity. The earliest published work of direct relevance to our study is due to Kaup [102], who pioneered the study of self-gravitating configurations of a massive complex scalar field, governed by the coupled system of Einstein and Klein-Gordon (EKG) equations. The main result of his work, which assumed spherical symmetry and a harmonic ansatz for the time dependence of the scalar field, was to determine the equilibrium states of the system. These are precisely the states that we now know as boson stars (Kaup referred to them as “scalar geons”). He

emphasized the similarities with neutron (fluid) stars, and established the conditions under which the stars would be stable (or unstable) to nonadiabatic radial perturbations. Indeed, Kaup was primarily interested in the issue of stability, and how resistant to gravitational collapse the massive complex scalar field was.

At about the same time, Ruffini and Bonazzola [103] investigated the same spherically symmetric system, and were also able to determine the equilibrium boson star solutions. However, in contrast to Kaup, these authors viewed the model as representing a semi-classical approach in which the Klein-Gordon field was fundamentally quantum mechanical. From considerations that parallel the calculations performed by Chandrasekhar for fluid stars [104], one of their main results was the notion of a critical mass for boson stars; that is a mass above which the pressure support from the star (supposedly arising from the Heisenberg uncertainty principle) would be insufficient to resist gravitational collapse. They showed that the critical mass for bosonic matter scaled differently with the particle mass, m , than it did for fermionic (neutron star) matter, and that for any reasonable values of m , the resulting sizes and mass of the boson stars would be so small as to be astrophysically inconsequential.

However, many years later, a seminal paper by Colpi and collaborators [105] showed that if the scalar field was endowed with a nonlinear self-interaction potential, then by suitable choice of the coupling constants appearing in the nonlinear terms, boson stars with masses comparable to neutron stars could be achieved using a plausible value for m . Specifically, for the case of a $\lambda|\phi|^4$ potential, and for sufficiently large values of the dimensionless coupling constant ³, $\Lambda = \lambda\hbar c/4\pi Gm^2$, the structure of the static solutions was found to change significantly relative to the stars with $\lambda = 0$ that had originally been studied. For large Λ the stellar density profile exhibits a relatively slow decline out to some coupling-dependent characteristic radius, and this leads to a significant enhancement of the star's mass. In fact, it was demonstrated in [105] that as $\Lambda \rightarrow \infty$ the maximum stellar mass has the *same* scaling dependence on the particle mass, m , as for fermionic stars. Although this result led to revived interest in astrophysical implications of boson stars—including the possibility that they might constitute at least some fraction of the dark matter that is currently unaccounted for—we must emphasize that no fundamental scalar boson has yet been detected, nor is there any credible evidence for the existence of boson stars.

The work by Colpi *et. al.* was followed by a burst of activity in boson star research, with several

³Colpi and collaborators adopt $c = \hbar = 1$.

groups carrying out investigations focused mainly on their stability properties [106, 107, 108, 109, 110, 111]. In 1991, Jetzer published a comprehensive review of the subject [112], and this remains a very useful general reference, especially for newcomers, as does the review by Lee and Pang [113].

What one might identify as the modern era of boson star research began with work by Seidel and Suen [114]. These authors considered the *dynamical* evolution of the full spherically-symmetric Einstein-Klein-Gordon equations, and were thus able to directly address issues such as dynamical stability. Their numerical experiments indicated that the stars, parametrized by the central modulus, ϕ_0 , of the scalar field, belonged to two “branches”: one stable, the other unstable. Further, and in accord with analogous fluid-star calculations, as well as perturbation theory, the change in stability was found to coincide with the boson star that had maximal mass. When perturbed, stars that were on the stable branch were found to oscillate and radiate scalar radiation with a particular frequency, eventually settling down to a somewhat less massive stable star. On the other hand, stars on the unstable branch that were perturbed would either undergo gravitational collapse to a black hole or radiate some scalar field and migrate to a configuration on the stable branch with smaller mass. Later, in collaboration with Balakrishna, Seidel and Suen [115] extended these studies to include excited states⁴ and boson stars with nonlinear self-interactions. The dynamics of spherically symmetric boson stars have also been studied by other authors, including Choptuik and collaborators [116, 117], who have investigated the nature of critical gravitational collapse [118] in this context.

In addition to the spherically symmetric work, the past decade or so has seen the accumulation of a considerable literature on axially symmetric boson stars. Since most of this work is not closely related to ours, we direct the interested reader to Schunck and Mielke [119], which remains the most recent comprehensive review of boson stars.

Finally, there have been a few previous efforts in which the dynamical evolution of boson stars has been studied in three spatial dimensions. This includes work by Balakrishna *et. al.* [7] and Palenzuela *et. al.* which, in both cases, involved the solution of the full Einstein equations.

1.3 Summary of Thesis

The remainder of this thesis is organized as follows:

⁴The boson stars discussed thus far, and which constitute the focus of this thesis, are so-called *ground state* boson star, where the spherically-symmetric scalar field profile is nodeless. The excited states have one or more nodes (zero-crossings).

Chap. 2 discusses the principal formalisms and (continuum) approximations used in this thesis. A quick overview of general relativity is given, primarily to fix notation, and to introduce key geometrical quantities (metric, curvature, etc.). This is followed by a review of the $3 + 1$ decomposition of spacetime, in which the 4-geometry of spacetime is viewed as the time history of the 3-geometry of a spacelike hypersurface. We then proceed to a description of our matter model (a massive complex scalar field) which includes a derivation of the associated equations of motion for the scalar field. This is followed by a section devoted to a general conformal decomposition of the $3 + 1$ equations. The conformally flat approximation is then introduced, and the final form of the equations of motion for our model is derived. Related issues such as the boundary conditions and the ADM mass are also discussed.

Chap. 3 is concerned with the generation of initial data for our model. We first give a description of the construction of spherically symmetric, ground state boson stars. We then detail the procedures we used to interpolate one of two stars onto the three dimensional computational domain, and to provide the stars with initial velocities through the application of approximate Lorentz boosts.

Chap. 4 begins with a review of some basic concepts related to the finite difference approximation (FDA) of partial differential equations (PDEs). This is followed by a description of a simple example that illustrates the specific scheme we used to discretize our hyperbolic PDEs. We then discuss the strategies and techniques we use to establish the validity of our code and to assess the quality of the numerical results. The chapter concludes with a brief overview of the multigrid method in general, as well as a description of the specific multigrid algorithm we implemented to solve our discretized elliptic PDEs.

The majority of the original research contributions in this thesis are contained in Chap. 5. The chapter starts with a discussion of the construction of the numerical code, including a description of the overall algorithmic flow. This is followed by the results from a thorough series of code tests that employed several different types of initial data. The dynamics of head-on collisions of boson stars is then considered. The last part of the chapter discusses results related to orbital dynamics. We perform a modest parameter space survey and identify three distinct endstates for the calculations: 1) long term orbital motion, 2) merger to a conjectured rotating and pulsating boson star, and 3) merger to a conjectured black hole.

App. A documents the use of a publicly-available FORTRAN subroutine that generates initial

data for a spherically-symmetric boson star, and which implements a general polynomial self-interaction term for the scalar field. We feel that this routine should be of significant utility to others who wish to study the dynamics of general relativistic boson stars.

App. B lists all of the finite difference formulae we have used in the thesis and includes a note on the regularization of differential operators in curvilinear coordinate systems prior to finite-differencing.

App. C contains brief documentation of a set of Maple procedures which facilitate the construction and coding of finite-difference discretizations of generic systems of PDEs. Again, some effort has been expended in designing the routine so that it is potentially useful in other contexts, and for other researchers.

Finally, App. D contains a brief review of the relaxation techniques that we have used in this work.

1.4 Conventions, Notation and Units

We adopt the abstract index notation for tensors as defined and discussed in Wald [120]. In particular, letters from the beginning of the Latin alphabet, $\{a, b, c, \dots\}$, denote abstract indices. We then use two sets of indices for tensor *components*: Greek indices $\{\mu, \nu, \dots\}$ range over the spacetime values 0, 1, 2, 3 (where 0 is the time index), while the subset of Latin indices $\{i, j, k, l, m, n\}$ range over the spatial values 1, 2, 3. The Einstein summation convention applies to both of these component index types. We also adopt Wald's sign conventions, so, in particular, the metric signature is $(-, +, +, +)$.

The totally symmetric and totally antisymmetric parts of a tensor of type $(0, 2)$ are defined by

$$T_{(ab)} = \frac{1}{2}(T_{ab} + T_{ba}), \quad (1.1)$$

and

$$T_{[ab]} = \frac{1}{2}(T_{ab} - T_{ba}), \quad (1.2)$$

respectively.

Additionally, we use a common terminology from computational science in which the sets of time-dependent PDEs with dependence on 1, 2 and 3 spatial dimensions (independent variables) are

referred to as 1D, 2D and 3D, respectively. In particular, PDEs describing spherically symmetric systems in which the field variables depend on time and a radial coordinate, r , are referred to as 1D.

A variety of differentiation operators are used below. ∇_a and D_a denote the covariant derivative operators compatible with the spacetime metric, g_{ab} , and induced hypersurface metric, γ_{ab} , respectively. The Lie derivative along a vector field v^a is denoted by \mathcal{L}_v . Ordinary derivatives are represented by different notations according to the context. For example the ordinary derivative of the function $f = f(x, y, z)$ with respect to the coordinate x will be denoted in one of the following ways: $\partial f / \partial x \equiv \partial_x f \equiv f_{,x}$. For time dependent functions, $g = g(t, x, y, z)$, we also sometimes denote temporal differentiation using an overdot, so $\dot{g} \equiv \partial g / \partial t \equiv \partial_t g$. Similarly, for functions of one spatial variable, $g = g(t, r)$ or $g = g(r)$, we sometimes use the prime notation for the first (spatial) derivative: $g' \equiv \partial g / \partial r$ or $g' \equiv dg / dr$. Finally, a variant of geometric units is used throughout this thesis. Specifically, Newton's gravitational constant, G , the speed of light in vacuum, c , and the scalar field mass parameter, m , are all set to unity: $G = c = m = 1$.

Chapter 2

Formalism and Equations of Motion

As described in the introductory chapter, this thesis focuses on the numerical solution of a model that describes the gravitational interaction of boson stars within an approximation to Einstein’s general theory of relativity. The bulk of this chapter is devoted to a discussion of the formalism underlying the model, and the derivation of the set of partial differential equations (PDEs) which govern it. We start with a brief review of the general relativistic field equations in Sec. 2.1. We then proceed to a discussion of the so called 3 + 1 (or ADM) formalism which underlies most of the computational work⁵ in which systems of general relativistic partial differential equations are solved as initial value problems (or initial-boundary values problems). In Sec. 2.3 we discuss the matter content of our model, which is a massive complex scalar field. The equation of motion for the scalar field is derived, as are related quantities, including its stress-energy tensor. Sec. 2.4 then provides a definition of the approximation we make to the Einstein equations, as well as a detailed derivation of the PDEs that, using this approximation, then govern our model. The key element in the approximation is the demand that the 3-metric (which is the fundamental dynamical variable in the 3 + 1 approach to general relativity) be conformal to a *flat* metric. In addition to this requirement, a specific choice of time coordinatization—known as maximal slicing—is made (Sec. 2.4.1). This combination of an assumption of conformal flatness and a choice of maximal slicing leads to what we call the conformally flat approximation (or CFA) of general relativity.

Principally for the sake of completeness, Sec. 2.4.2 contains a review of a specific approach to the conformal decomposition of the 3 + 1 form of the Einstein equations. Once this is in hand, it is a relatively straightforward matter to impose conformal flatness, adopt maximal slicing and then derive the equations of motion for our model (Sec. 2.4.3).

Ultimately this results in a system of nonlinear PDEs of mixed elliptic-hyperbolic type that are written in Cartesian coordinates (Sec. 2.4.4). In Sec. 2.4.5 we discuss three different strategies that we have investigated for the implementation of boundary conditions for this system. This is

⁵Here we note that an alternate formulation of Einstein’s equations, known as the generalized harmonic approach, has also had a major impact on numerical relativity in the past few years.

followed by a definition of the ADM mass and a short description of our use of that quantity as a diagnostic in our computations (Sec. 2.4.6). Sec. 2.5 then concludes the chapter with a recap of the equations of motion, as well as a synopsis of how they were solved numerically. Thus, the reader interested only in the final form of the equations of motion may wish to proceed directly to that section.

2.1 The Field Equations of General Relativity

Space and time in general relativity are modelled as a four dimensional Lorentzian manifold, \mathcal{M} . Each point in the spacetime manifold corresponds to a physical event. The notion of distance between two points or, in the spacetime case, the interval between two events, is encoded in a symmetric, non-degenerate, tensor field of type $(0, 2)$, the *metric* g_{ab} . The metric is thus the fundamental entity used to quantify the geometry of spacetime, and gravitational effects arise due to the fact that this geometry is, in general, curved.

Let $\{x^\mu\}$ be a coordinate system that we assume covers that part of the manifold which is of interest. The vector and dual vector bases associated with this coordinate system are called coordinate bases and are usually written as $\{(\partial/\partial x^\mu)^a\}$ and $\{(dx^\mu)_a\}$, respectively. Therefore the metric components in this coordinate basis can be defined by:

$$g_{ab} \equiv ds^2 = g_{\mu\nu} (dx^\mu)_a (dx^\nu)_b. \quad (2.1)$$

The covariant derivative map, ∇_a , is written in terms of the ordinary (partial) derivative map, ∂_a , via

$$\nabla_a v^b = \partial_a v^b + \Gamma^b_{ac} v^c, \quad (2.2)$$

where Γ^b_{ac} is the *Christoffel symbol* and v^b is an arbitrary vector. Throughout this thesis we use a covariant derivative that is compatible with the metric, i.e. that satisfies $\nabla_a g_{bc} = 0$. The Christoffel symbol can then be calculated in terms of the ordinary derivative operator:

$$\Gamma^c_{ab} = \frac{1}{2} g^{cd} (\partial_a g_{bd} + \partial_b g_{ad} - \partial_d g_{ab}). \quad (2.3)$$

The intrinsic notion of curvature of the spacetime can be made mathematically precise by

considering the parallel transport of vectors along curves in the spacetime.⁶ For example, a spacetime is curved in a region if an arbitrary vector that is parallel transported along a closed path that bounds that region experiences an overall rotation. Another consequence of curvature is that the result of parallel transporting a vector is, in general, path dependent. In fact, the failure of successive applications of the covariant derivative to commute captures the path-dependence of parallel transport (along with the notion of curvature) and is quantified by the so called intrinsic curvature tensor, the *Riemann curvature tensor*, defined by:

$$2\nabla_{[a}\nabla_{b]}w_c = R_{abc}{}^d w_d. \quad (2.4)$$

Contractions of the Riemann tensor give rise to the *Ricci tensor*:

$$R_{ac} = R_{adc}{}^d, \quad (2.5)$$

and to the *Ricci scalar* or *scalar curvature*:

$$R = g^{ab}R_{ab}. \quad (2.6)$$

A key feature of the Riemann curvature tensor is that it satisfies the contracted *Bianchi identity* which can be expressed as the fact that the *Einstein tensor*, defined in terms of the Ricci tensor and scalar curvature as

$$G_{ab} = R_{ab} - \frac{1}{2}g_{ab}R, \quad (2.7)$$

has vanishing divergence:

$$\nabla^a G_{ab} = 0. \quad (2.8)$$

Given a metric, g_{ab} , the Riemann curvature tensor can be written in terms of the Christoffel symbols through the following tensorial equation:

$$R_{abc}{}^d = -2(\partial_{[a}\Gamma^d{}_{b]c} - \Gamma^e{}_{c[a}\Gamma^d{}_{b]e}). \quad (2.9)$$

Once a coordinate basis is chosen, the Riemann tensor components can be computed from the metric components and its various derivatives in that basis. In particular, since from (2.3) the

⁶See Chap. 3 of [120] for a complete and detailed discussion of this material.

Christoffel symbol components involve first derivatives of the metric components, the components of the curvature tensor generally involve second derivatives of the metric.

The curvature of spacetime arises in response to the distribution of matter-energy in the spacetime, which includes contributions from the gravitational field itself, as well as those from any matter fields present in the spacetime. For any given matter field, the coupling to gravity is described by the stress-energy-momentum tensor (stress-energy for short): T_{ab} . In many cases, including that considered in this thesis, the equation of motion for the matter field can be derived from the vanishing of the divergence of the stress tensor,

$$\nabla^a T_{ab} = 0. \quad (2.10)$$

The *Einstein field equations* expresses the intimate relationship between the physical phenomena taking place in the spacetime and the curvature of the spacetime geometry itself. In their most compact and elegant form the field equations are:

$$G_{ab} \equiv R_{ab} - \frac{1}{2}g_{ab}R = \kappa T_{ab}, \quad (2.11)$$

where the constant κ depends on the system of units, but is also chosen so that in the weak-field limit, the Newtonian description of gravity is recovered. In the geometric units used in this thesis we have $\kappa = 8\pi$.

Once a coordinate basis is chosen, the Einstein equations can be cast as a system of 10 nonlinear second order partial differential equations for the metric components, $g_{\mu\nu}$, in the chosen coordinate system. Since the metric signature is Lorentzian, i.e. $(-, +, +, +)$, this system of partial differential equations has a hyperbolic or wave-like character.

Although there have been many “exact” (closed form) solutions of the Einstein equations discovered over the years, very few of these are considered to be of significant physical interest, and even fewer are thought to be relevant to astrophysics. Typically, exact solutions are obtained by demanding symmetries: the spherically symmetric Schwarzschild solution for a single black hole, as well as the Kerr solution, which is axially symmetric and generalizes Schwarzschild to the case of rotating black holes, are among the most important examples. However, for situations in which 1) the gravitational field is strong and highly dynamical, 2) there are no symmetries, and 3) one does not have substantial *a priori* information about the expected nature of the solution, numeri-

cal analysis provides the only currently viable general approach for solving the Einstein equations. This, of course, is precisely the situation that confronts us in our current study of interacting boson stars.

We end this brief overview of general relativity, with the observation that the complex, tensorial nature of the Einstein field equations has fostered the development of *many* different approaches to their expression and analysis. We will refer to any such approach as a *formalism* for general relativity, and in this thesis will focus exclusively on formalisms that are appropriate for solving the field equations as an initial-value problem. That is, we are interested in approaches where the state of the gravitational field—defined in terms of some set of dynamical variables for the field—can be prescribed at some initial time. The initial data will then be evolved to the future (or past) using partial differential equations of motion for the dynamical variables. A given formalism must thus include prescriptions for 1) the decomposition of spacetime into space and time, 2) the choice of dynamical and auxiliary variables, and 3) the choice of time and space coordinates. Crucially, it must also provide a specific form of the equations of motion that will be suitable for use in numerical computations. As mentioned previously, the $3 + 1$ (or ADM) formalism has formed the basis for most work to date in numerical relativity. It is also the fundamental approach that we adopt here, and is the topic of the next section.⁷

⁷Here we should emphasize that developments over the past decade or so—both theoretical and computational—have made it clear that the traditional form of the $3 + 1$ equations is *not* generally suitable for fully 3D numerical calculations. This is due to the fact that this form of the field equations is not strongly hyperbolic, and thus does not lead, in general cases, to a well posed initial value problem. However, this fact is not relevant to our work since as a result of the approximation we adopt, the quantities describing the gravitational field are all governed by *elliptic* rather than hyperbolic PDEs.

2.2 The 3 + 1 (ADM) Formalism

As we have just indicated, in the 3 + 1 (or ADM) formalism, spacetime is decomposed into space and time.⁸ Fundamental to this approach is the choice of a timelike unit vector field, t^a , in the spacetime and a foliation of spacelike hypersurfaces, Σ_t , parametrized by a time function. The timelike vector field is chosen such that its integral curves represent the time coordinate (or time function), t , throughout the spacetime, i.e. such that $t^a \nabla_a t = 1$. In essence the vector field and the time function are chosen to enable the definition of the notion of dynamical evolution for quantities defined on the hypersurfaces. In order to make the last statement precise, it is necessary to decompose spacetime vector fields (as well as tensor fields of higher rank) into (vector) pieces that are either defined 1) exclusively on any given hypersurface, or 2) in a direction normal to the hypersurface. For example, for the vector field t^a itself we write

$$t^a = \alpha n^a + \beta^a, \quad (2.12)$$

where n^a is the future-directed, timelike unit vector field normal to Σ_t (thus satisfying $g_{ab} n^a n^b = -1$ with our convention for the metric signature), α is the function that gives the component of t^a in the normal direction, and β^a is a vector field that resides on the hypersurface (thus satisfying $g_{ab} n^a \beta^b = 0$). In 3 + 1 parlance, α and β^a are known as the *lapse function* and *shift vector*, respectively.

From $g_{ab} n^a n^b = -1$ and $g_{ab} n^a \beta^b = 0$ it follows from (2.12) that

$$\alpha = -g_{ab} t^a n^b. \quad (2.13)$$

Introducing the projection tensor, γ^a_b , defined by

$$\gamma^a_b = \delta^a_b + n^a n_b, \quad (2.14)$$

and also sometimes denoted \perp^a_b , it follows from (2.12) and (2.14) that the shift vector is given by

$$\beta^a = \gamma^a_b t^b. \quad (2.15)$$

⁸The development in this section closely parallels that of York in [121].

We also note that in order for $t^a \nabla_a t = 1$ to be satisfied, (2.13) implies that we must have

$$n_a \equiv g_{ab} n^b = -\alpha \nabla_a t. \quad (2.16)$$

The decomposition of vectors into “temporal” and “spatial” parts can be readily generalized to tensors of arbitrary rank. Following York [121], for any vector field, W^a , we define

$$W^{\hat{n}} = -W^a n_a, \quad (2.17)$$

and, in general, any upstairs \hat{n} index denotes that the original tensor index has been contracted with $-n_a$. On the other hand, for a dual vector field, W_a , York defines

$$W_{\hat{n}} = +W_a n^a, \quad (2.18)$$

and then any downstairs \hat{n} index denotes contraction with $+n^a$.

Of special interest are tensors which have been completely projected onto the hypersurface. For a general tensor, $T^{ab\dots}_{cd\dots}$ we write

$$\perp T^{ab\dots}_{cd\dots} \equiv \gamma^a_e \gamma^b_f \dots \gamma^g_c \gamma^h_d \dots T^{ef\dots}_{gh\dots} \quad (2.19)$$

and $\perp T^{ab\dots}_{cd\dots}$ is called a *spatial tensor* since

$$n_a \perp T^{ab\dots}_{cd\dots} = n_b \perp T^{ab\dots}_{cd\dots} \dots = n^c \perp T^{ab\dots}_{cd\dots} = n^d \perp \perp T^{ab\dots}_{cd\dots} \dots = 0. \quad (2.20)$$

One particularly important spatial tensor is the induced three dimensional metric, γ_{ab} , of the hypersurface

$$\gamma_{ab} = g_{ab} + n_a n_b, \quad (2.21)$$

which, the reader should note, is also given by lowering the upstairs index of the projection tensor, $\perp^a_b \equiv \gamma^a_b$.

In order to describe parallel transport of spatial tensors and curvature within the hypersurface, a covariant derivative operator must be defined. A natural choice is to project the four dimensional covariant derivative onto the spacelike hypersurface, leading to the definition of a three dimensional

covariant derivative operator, D_a :

$$D_a \equiv \perp \nabla_a. \quad (2.22)$$

The Riemann curvature tensor on the hypersurface is defined analogously to its four dimensional counterpart:

$$2D_{[a}D_{b]}w_c = \mathcal{R}_{abc}{}^d w_d, \quad (2.23)$$

while the Ricci tensor and Ricci scalar are obtained by the usual contractions of the Riemann curvature tensor:

$$\mathcal{R}_{ab} = \mathcal{R}_{acb}{}^c \quad \text{and} \quad \mathcal{R} = \mathcal{R}^a{}_a. \quad (2.24)$$

As mentioned previously, the Riemann curvature tensor describes the curvature *intrinsic* to a manifold. In the current case, which involves the embedding of three-dimensional hypersurfaces in a four-dimensional spacetime, there is a second type of curvature, known as the *extrinsic* curvature, that quantifies the embedding. Since the orientation of the hypersurface within the spacetime is related to the unit normal vector, n^a , the covariant derivative of n^a thus characterizes nearby changes in the orientation. The extrinsic curvature tensor, K_{ab} , can therefore be defined as the projection of the covariant derivative of the dual vector field associated to the normal vector field:

$$K_{ab} = -\perp \nabla_a n_b = -\frac{1}{2} \perp \mathcal{L}_n g_{ab} = -\frac{1}{2} \mathcal{L}_n \gamma_{ab}. \quad (2.25)$$

Since both sides of the Einstein equations must be decomposed in the 3+1 approach, we must also consider various projections of the stress-energy tensor along the normal n^a and onto the hypersurface, Σ_t . First note that the stress-energy tensor T_{ab} is a type (0,2) symmetric tensor. A generic tensor of this type can be decomposed in the following way in the 3+1 formalism:

$$T_{ab} = \perp T_{ab} - 2n_{(a} \perp T_{b)\hat{n}} + n_a n_b T_{\hat{n}\hat{n}}. \quad (2.26)$$

We rewrite the above as

$$T_{ab} = S_{ab} - 2J_{(a} n_{b)} + \rho n_a n_b, \quad (2.27)$$

where the quantities, ρ , J_a and S_{ab} are defined by

$$\rho \equiv T_{\hat{n}\hat{n}} = T_{ab}n^an^b = T^{ab}n_an_b, \quad (2.28)$$

$$J_a \equiv \perp T_{a\hat{n}} = \perp (T_{ab}n^b), \quad (2.29)$$

$$J^a \equiv \perp T^{a\hat{n}} = -\perp (T^{ab}n_b), \quad (2.30)$$

$$S_{ab} \equiv \perp T_{ab}, \quad (2.31)$$

Physically, ρ is interpreted as the local energy density, J^a as the momentum density, and S_{ab} as the spatial stress tensor, all measured by observers moving orthogonally to the slices.

The several possible combinations of projections of Einstein equations along the “temporal” and “spatial” directions give rise to the equations of motion in the 3 + 1 form. Projecting both indices with n^a we find

$$\mathcal{R} + K^2 - K_{ab}K^{ab} = 16\pi\rho, \quad (2.32)$$

where $K \equiv K^a{}_a$ is the trace of the extrinsic curvature. Eq. (2.32) is also known as *Hamiltonian constraint*. On the other hand, if only one index is contracted along n^a , while the other is projected onto the hypersurface, we derive a three-vector equation known as the *momentum constraint*:

$$D_bK^{ab} - D^aK = 8\pi J^a. \quad (2.33)$$

We note that care must be exercised in using the covariant form of this equation, since due to the relative sign in the definitions of J_a and J^a in equations (2.29) and (2.30) (see definitions (2.17) and (2.18)), we have

$$\perp G_{a\hat{n}} = -D^bK_{ab} + D_aK = 8\pi J_a, \quad (2.34)$$

$$\perp G^{a\hat{n}} = D_bK^{ab} - D^aK = 8\pi J^a. \quad (2.35)$$

Key features of the constraint equations are the presence of *only* spatial tensors and the absence of explicit time derivative of these tensors. They must be satisfied by $\{\gamma_{ab}, K_{ab}\}$ on all slices, including the initial slice.

The 3+1 equations that *do* involve time derivatives of the spatial tensors $\{\gamma_{ab}, K_{ab}\}$, are thus

called evolution equations. For the spatial metric, an evolution equation follows from the definition of the extrinsic curvature (2.25):

$$\mathcal{L}_t \gamma_{ab} = -2\alpha K_{ab} + \mathcal{L}_\beta \gamma_{ab}. \quad (2.36)$$

where \mathcal{L}_t is the Lie derivative with respect to the vector field t^a . We note here that the Lie derivative of a general tensor, $T^{a_1 \dots a_k}_{b_1 \dots b_l}$, with respect to a vector field, v^a , is defined by:⁹

$$\begin{aligned} \mathcal{L}_v T^{a_1 \dots a_k}_{b_1 \dots b_l} &= v^c D_c T^{a_1 \dots a_k}_{b_1 \dots b_l} - \sum_{i=1}^k T^{a_1 \dots c \dots a_k}_{b_1 \dots b_l} D_c v^{a_i} \\ &\quad + \sum_{j=1}^l T^{a_1 \dots a_k}_{b_1 \dots c \dots b_l} D_{b_j} v^c. \end{aligned} \quad (2.37)$$

The evolution equation for the extrinsic curvature can be derived by considering the projection of both indices of Einstein equations, which involves computation of $\perp R_{a\hat{n}a\hat{n}}$. After some manipulation, we find

$$\mathcal{L}_t K^a_b = \mathcal{L}_\beta K^a_b - D^a D_b \alpha + \alpha \left[\mathcal{R}^a_b + K K^a_b + 8\pi \left(\frac{1}{2} \gamma^a_b (S - \rho) - S^a_b \right) \right]. \quad (2.38)$$

All of the definitions and decompositions discussed thus far are independent of any choice of coordinate system. Operationally however, we must introduce coordinates, $x^\mu \equiv (t, x^i)$, in order to cast (2.32), (2.33), (2.36) and (2.38) as a system of partial differential equations that can then be solved—using a numerical approach in general—for given initial data and boundary conditions (we remind the reader that Greek indices such as μ range over the spacetime values, 0,1,2,3, while Latin indices such as i are restricted to spatial values, 1,2 and 3). We thus adopt such a coordinate system, with the time coordinate, t , being identified with the time function. In terms of tensor components taken with respect to the coordinate basis, the spacetime displacement can be written as

$$\begin{aligned} ds^2 &= g_{\mu\nu} dx^\mu dx^\nu \\ &= -\alpha^2 dt^2 + \gamma_{ij} (dx^i + \beta^i dt) (dx^j + \beta^j dt). \end{aligned} \quad (2.39)$$

⁹We should emphasize that this expression is valid for *any* derivative operator D_a . The reader should refer to App. C of Wald [120] for a comprehensive discussion.

We emphasize that the lapse function, α , the three shift vector components, β^i , and the six components of the symmetric 3-metric, γ_{ij} , that appear in the above expression are all functions of the coordinates $x^\mu \equiv (t, x^i)$. We also note that the lapse function can be interpreted as the ratio of proper time to coordinate time for an observer travelling normally to the hypersurface, while the shift vector encodes the translation of spatial coordinates from one slice to the other, again relative to propagation in the normal direction. In addition, component indices of spatial tensors are lowered and raised with the 3-metric γ_{ij} and its inverse γ^{ij} , respectively, where γ^{ij} is defined by $\gamma^{ik}\gamma_{kj} = \delta^i_j$.

As discussed above, the decomposition of the stress-energy tensor gives rise to a variety of energy-momentum quantities defined by equations (2.28-2.31). Using the relations $n^\mu = (1/\alpha; -\beta^i/\alpha)$ and $n_\mu = (-\alpha; 0)$, these quantities become

$$\rho = \frac{T_{00}}{\alpha^2} - 2\frac{\beta^i T_{0i}}{\alpha^2} + \frac{\beta^i \beta^j T_{ij}}{\alpha^2} = \alpha^2 T^{00}, \quad (2.40)$$

$$J_i = \frac{T_{i0}}{\alpha} - \frac{T_{ij}\beta^j}{\alpha}, \quad (2.41)$$

$$J^i = \alpha(T^{i0} + T^{00}\beta^i), \quad (2.42)$$

$$S_{ij} = T_{ij}. \quad (2.43)$$

The component form of the evolution equations can then be written as

$$\partial_t \gamma_{ij} = -2\alpha \gamma_{ik} K^k_j + 2D_{(i}\beta_{j)}, \quad (2.44)$$

and

$$\begin{aligned} \partial_t K^i_j &= \beta^k \partial_k K^i_j - \partial_k \beta^i K^k_j + \partial_j \beta^k K^i_k - D^i D_j \alpha \\ &+ \alpha \left[\mathcal{R}^i_j + K K^i_j + 8\pi \left(\frac{1}{2} \delta^i_j (S - \rho) - S^i_j \right) \right]. \end{aligned} \quad (2.45)$$

We will use these versions of the evolution equations for the components of the 3-metric and extrinsic curvature when discussing spherically symmetric spacetimes in Ch. 3. However, for the purpose of performing a conformal decomposition of the 3+1 evolution equations—which is done later in the current chapter—we will start from slightly different forms which may be easily derived from previous formulae using straightforward tensor calculus. Specifically, for the 3-metric components

we can also write

$$\mathcal{L}_{(t-\beta)}\gamma_{ij} = -2\alpha K_{ij} \quad (2.46)$$

which follows immediately from (2.36) as well as

$$\mathcal{L}_{(t-\beta)}\gamma^{ij} = 2\alpha K^{ij}. \quad (2.47)$$

This last result is easily established from (2.46) and the fact that

$$\begin{aligned} 0 &= \mathcal{L}_{(t-\beta)}(\delta^i_k) = \mathcal{L}_{(t-\beta)}(\gamma^{ij}\gamma_{jk}) = \gamma_{jk}\mathcal{L}_{(t-\beta)}\gamma^{ij} + \gamma^{ij}\mathcal{L}_{(t-\beta)}\gamma_{jk} \\ &\implies \gamma_{jk}\mathcal{L}_{(t-\beta)}\gamma^{ij} = -\gamma^{ij}\mathcal{L}_{(t-\beta)}\gamma_{jk}. \end{aligned}$$

For the extrinsic curvature components, K_{ij} , we have

$$\mathcal{L}_{(t-\beta)}K_{ij} = -D_i D_j \alpha + \alpha \left[\mathcal{R}_{ij} + K K_{ij} - 2K_{ik}K^k_j + 8\pi \left(\frac{1}{2}\gamma_{ij}(S - \rho) - S_{ij} \right) \right]. \quad (2.48)$$

Additionally, the constraint equations in component form are:

$$\mathcal{R} + K^2 - K_{ij}K^{ij} = 16\pi\rho, \quad (2.49)$$

$$D_j K^{ij} - D^i K = 8\pi J^i. \quad (2.50)$$

Even within a specific formulation of the Einstein equations, such as the 3+1 approach described above, the coordinate invariance of general relativity ensures that there are generally many distinct possibilities to solve the specific set of PDEs that results once the coordinate system has been fully fixed (full specification of the lapse and shift). Here we are referring to the fact that we have more equations (4 second-order “elliptic” constraints + 12 first-order-in-time evolution for a total of 16 equations) than fundamental dynamical unknowns (6 g_{ij} + 6 K_{ij} = 12 unknowns). The interested reader is referred to the classic paper by Piran [122] in which nomenclature, such as free evolution, constrained evolution and partially-constrained evolution is defined and discussed. Here, the key thing to note is that the approximation (CFA) that is adopted in this thesis has the advantage of providing a single, well defined set of 5 elliptic PDEs for 5 well defined functions that completely fix the spacetime geometry. In this sense, and in an abuse of Piran’s original

classification, we implement a fully constrained evolution for the geometrical field and, further, in contrast to the full general-relativistic situation there are *no* purely gravitational degrees of freedom. That means that in the model considered here, as is the case for any model that adopts the CFA with maximal slicing condition, all dynamics is linked to the dynamics of the matter. This has a host of ramifications, physically, mathematically and computationally, but particularly given the efforts that have been expended on taming instabilities in free evolution approaches for the full Einstein equations, is one of the most attractive features of Isenberg's proposal.

2.3 The Complex Scalar Field

The matter model adopted in this thesis is a complex *Klein-Gordon field*, which satisfies a Klein-Gordon equation as discussed in detail below. This field represents a simple type of matter that when coupled to Einstein gravity, or in the context of the approximation adopted in this thesis, admits star-like solutions. Studies focusing on such solutions—known as boson stars—using a variety of techniques including numerical analysis, have a rich history and we refer the reader to the paper by Schunck and Mielke, [119] (and references therein), for an excellent and thorough review of the subject as of about five years ago.

General relativists have studied Klein-Gordon fields for many purposes over the years. As either 1) a classical field or 2) a quantum-theory of spin 0 particles, scalar fields have been widely exploited for exploratory theoretical studies. A key point is that the simplicity of scalar matter (in terms, e.g. of physical interpretation as well as complexity of the equations of motion), often allows one to investigate and understand basic theoretical issues in Einstein gravity relatively free of the complications a more realistic matter model could bring in. This is a chief motivation for the use of a scalar field in the current work.

In the discussion below, we will refer to the system of a single Klein-Gordon field minimally coupled¹⁰ to Einstein gravity as the Einstein-Klein-Gordon (EKG) system. We note that we adopt a complex scalar field, rather than a real one, since it has been long known that there are no regular, static solutions (i.e. star-like solutions) for a real scalar field in general relativity. Interestingly, for us this turns out to be something of a technical point, since for a real field coupled to Einstein gravity there are quasi-static solutions known as “oscillons” which have decay times that can be *much* longer than the intrinsic dynamical time [123]. Thus in principle one could use a real scalar field to study some of the effects we wish to examine in this thesis and follow-up work. However, for a variety of reasons, not least including the ease with which one can generate star-like solutions, we prefer to work with the complex field.

Additionally, the complex field must interact in a non-trivial potential, which we define to include a mass term. The possibilities for potential choice are endless, and have formed the basis for much previous work. Again, we choose the simplest approach and, at least initially, adopt only a mass term. The boson stars modelled by scalar fields with this self-interaction potential are also known in the literature as *mini-boson stars*.

¹⁰The notion of minimal coupling is defined in the next section.

Since mathematically we are ultimately interested in solving an approximate EKG system as an initial value problem, we note in passing that a scalar field is known to admit a well-posed initial value formulation in the following sense [120]:

- For an initial data in a spacelike Cauchy surface Σ in a globally hyperbolic spacetime ¹¹ (\mathcal{M}, g_{ab}) , there is an open neighbourhood O of Σ such that the Klein-Gordon equation has a solution in O and (O, g_{ab}) is globally hyperbolic.
- The solution in O is unique and propagates causally.
- The solution depends continuously on the initial data.

In addition to possessing star-like solutions, a key advantage of scalar matter relative to the more-astrophysically relevant perfect fluid case, is that the solutions do not tend to develop shocks or rarefaction regions. Rather, as is expected from the structure of the equations, and has been born out by many [114, 123, 124, 115, 116, 125, 126, 127, 128, 7, 8, 9, 129, 117, 10, 130] previous numerical studies, solutions tend to remain as smooth as the initial data, except at actual physical singularities (produced, for example, by gravitational collapse).

2.3.1 Einstein-Klein-Gordon System

One route to study matter models in general relativity is to postulate equations of motion for the matter, derive a suitable stress-energy tensor T_{ab} compatible with those equations and then use the Einstein equations to relate the matter distribution to the spacetime curvature through this stress-energy tensor. However, and as already mentioned, in many cases local conservation of the stress-tensor, implies the matter equation of motion. In such cases it is essentially sufficient to postulate the stress-energy tensor T_{ab} for the matter model in order to study the coupled system of matter distribution and spacetime geometry.

Additionally, and for a variety of reasons, it is often useful to adopt a Lagrangian (or variational) approach to Einstein equations and we will do so here. Here, a basic observation is that the vacuum Einstein equations can be obtained from the functional derivative of the so called Hilbert action functional:

$$S_G[g^{ab}] = \int_{\mathcal{M}} \mathcal{L}_G = \int_{\mathcal{M}} \sqrt{-g} R, \quad (2.51)$$

¹¹When the domain of dependence of a Cauchy surface is the whole spacetime (region of interest) then this spacetime (region) is said to be globally hyperbolic

where $\mathcal{L}_G = \sqrt{-g}R$ is the Einstein Lagrangian density and R the Ricci scalar. It is a standard exercise to show that the functional derivative of the action with respect to the inverse metric g^{ab} is

$$\frac{\delta S_G}{\delta g^{ab}} = \sqrt{-g}G_{ab}, \quad (2.52)$$

which then clearly yields the vacuum Einstein equations $G_{ab} = 0$ when the field configuration satisfies the action extremization condition:

$$\frac{\delta S_G}{\delta g^{ab}} = 0. \quad (2.53)$$

In order to obtain a coupled matter-gravity system, one then simply adds to the matter Lagrangian density to the Hilbert term (this is the so-called minimal coupling prescription). We thus have

$$\mathcal{L} = \mathcal{L}_G + \alpha_M \mathcal{L}_M, \quad (2.54)$$

where α_M is a coupling constant that can typically be rescaled through a redefinition of the matter fields. In the case of the Einstein-Klein-Gordon system one conventional choice that we adopt here is $\alpha_{KG} = 16\pi$.

The stress-energy tensor can now be calculated as the variation of the matter action with respect to the inverse metric field g^{ab} . Specifically, one has

$$T_{ab} = -\frac{\alpha_M}{8\pi} \frac{1}{\sqrt{-g}} \frac{\delta S_M}{\delta g^{ab}}, \quad (2.55)$$

where S_M is the action functional for the matter field M (understood here as a generic collection of matter fields and their higher order covariant derivatives):

$$S_M[g^{ab}, M] = \int_{\mathcal{M}} \mathcal{L}_M. \quad (2.56)$$

Finally, variations of the action S_M with respect to the matter fields themselves generate the equations of motion for the matter.

For the reasons discussed above, we now restrict attention to matter consisting of a single

complex scalar field, Φ . We write the field as

$$\Phi = \phi_1 + i\phi_2 = \phi_0 \exp(i\theta), \quad (2.57)$$

where ϕ_1 , ϕ_2 , ϕ_0 and θ are real-valued functions of the spacetime coordinates x^μ . The Lagrangian density associated with this field is

$$\mathcal{L}_\Phi = -\frac{1}{2}\sqrt{-g} (g^{ab}\nabla_a\Phi\nabla_b\Phi^* + U(|\Phi|^2)), \quad (2.58)$$

where $U(|\Phi|^2)$ is the scalar field self-interaction potential. As also discussed above, we will eventually specialize to the case where U contains only a mass term

$$U(|\Phi|^2) = m^2\Phi\Phi^* = m^2\phi_0^2 = m^2(\phi_1^2 + \phi_2^2), \quad (2.59)$$

but for the time being we will continue the discussion in terms of general potentials.

We now rewrite the Lagrangian (2.58) in terms of the real-valued quantities defined in (2.57):

$$\mathcal{L}_\Phi = -\frac{1}{2}\sqrt{-g} (g^{ab}\nabla_a\phi_1\nabla_b\phi_1 + g^{ab}\nabla_a\phi_2\nabla_b\phi_2 + U(\phi_0^2)). \quad (2.60)$$

Klein-Gordon equations for each real valued component ($\phi_A \in \{\phi_1, \phi_2\}$) can then be obtained by the usual variational procedure, yielding

$$\square\phi_A - \frac{dU(\phi_0^2)}{d\phi_0^2}\phi_A = 0 \quad \text{or} \quad g^{ab}\nabla_a\nabla_b\phi_A - \frac{dU(\phi_0^2)}{d\phi_0^2}\phi_A = 0, \quad A = 1, 2, \quad (2.61)$$

where $\square \equiv g^{ab}\nabla_a\nabla_b$ is the general relativistic D'Alembertian operator.

Once a coordinate system is chosen, each of the above scalar Klein-Gordon equations is a second-order-in-time PDE. In keeping with the 3+1 spirit, it is often conventional to recast these equations in first-order-in-time form, and we do so here. One specific way of doing this is to pass to the Hamiltonian description of the system in the standard fashion. Namely, we consider the Lagrangian as a function of the field and its spatial and time derivatives; we define a conjugate momentum associated with the field; we write down the Hamiltonian functional from the Lagrangian by performing a Legendre transformation for the conjugate momentum; and we then evaluate the Hamilton evolution equations from the Lagrangian. Full details of this procedure can be found in

standard texts such as Wald [120], and here we simply summarize the results for the scalar field.

Since the scalar-field Lagrangian (2.58) does not contain time derivatives higher than first order, the conjugate momentum associated with each component of the scalar field can be defined as:

$$\Pi_A \equiv \frac{\delta(\sqrt{-g}L_{\phi_A})}{\delta\dot{\phi}_A}, \quad (2.62)$$

or, more explicitly

$$\Pi_A = \frac{\sqrt{-g}}{\alpha^2} \left[\dot{\phi}_A - \beta^i \partial_i \phi_A \right], \quad (2.63)$$

where the overdot denotes differentiation with respect to the time coordinate.

The dynamical equations of motion (2.61) can be rewritten in terms of these conjugate fields, leading to four first-order-in-time partial differential equations for the two conjugate pairs of variables $\{\phi_A, \Pi_A\}$ (where $A = 1, 2$):

$$\partial_t \phi_A = \frac{\alpha^2}{\sqrt{-g}} \Pi_A + \beta^i \partial_i \phi_A, \quad (2.64)$$

$$\partial_t \Pi_A = \partial_i (\beta^i \Pi_A) + \partial_i (\sqrt{-g} \gamma^{ij} \partial_j \phi_A) - \sqrt{-g} \frac{dU(\phi_0^2)}{d\phi_0^2} \phi_A. \quad (2.65)$$

These last equations can be further manipulated using the following relationship between the determinants of the spacetime and spatial metrics: ¹²

$$\sqrt{-g} \equiv \alpha \sqrt{\gamma}, \quad (2.66)$$

yielding

$$\partial_t \phi_A = \frac{\alpha}{\sqrt{\gamma}} \Pi_A + \beta^i \partial_i \phi_A, \quad (2.67)$$

$$\partial_t \Pi_A = \partial_i (\beta^i \Pi_A) + \partial_i (\alpha \sqrt{\gamma} \gamma^{ij} \partial_j \phi_A) - \alpha \sqrt{\gamma} \frac{dU(\phi_0^2)}{d\phi_0^2} \phi_A. \quad (2.68)$$

Having obtained equations of motion for the scalar field, we now consider computation of the stress-energy tensor and the 3+1 quantities derived from it. Using the variational prescription

¹²This relation is derived from the definition of inverse metric: $g^{00} = -\alpha^{-2} = (-1)^{0+0} \det(\gamma_{ij}) / \det(g_{\mu\nu}) = \gamma/g$.

sketched above we find

$$T_{ab} = \frac{1}{2} [\nabla_a \Phi \nabla_b \Phi^* + \nabla_b \Phi \nabla_a \Phi^* - g_{ab} (g^{cd} \nabla_c \Phi \nabla_d \Phi^* + U(\Phi \Phi^*))], \quad (2.69)$$

which, adopting a coordinate basis, and working with the real-valued field components becomes

$$T_{\alpha\beta} = \sum_{A=1}^2 \frac{1}{2} [\partial_\alpha \phi_A \partial_\beta \phi_A + \partial_\beta \phi_A \partial_\alpha \phi_A - g_{\alpha\beta} g^{\mu\nu} \partial_\mu \phi_A \partial_\nu \phi_A] - \frac{1}{2} g_{\alpha\beta} U(\phi_0^2). \quad (2.70)$$

From the above, and using (2.40)-(2.43), we compute the 3+1 stress-energy quantities and find:

$$\rho = \frac{1}{2} \sum_{A=1}^2 \left[\frac{\Pi_A^2}{\gamma} + \gamma^{ij} \partial_i \phi_A \partial_j \phi_A \right] + \frac{1}{2} U(\phi_0^2), \quad (2.71)$$

$$J_i = \sum_{A=1}^2 \left[\frac{\Pi_A}{\sqrt{\gamma}} \partial_i \phi_A \right], \quad (2.72)$$

$$J^i = \sum_{A=1}^2 \left[-\frac{\Pi_A}{\sqrt{\gamma}} \gamma^{ij} \partial_j \phi_A \right], \quad (2.73)$$

$$S_{ij} = \frac{1}{2} \sum_{A=1}^2 \left\{ 2\partial_i \phi_A \partial_j \phi_A + \gamma_{ij} \left[\frac{\Pi_A^2}{\gamma} - \gamma^{pq} \partial_p \phi_A \partial_q \phi_A \right] \right\} - \frac{1}{2} \gamma_{ij} U(\phi_0^2). \quad (2.74)$$

Additionally, we need to compute the trace of the spatial stress tensor, $S \equiv S^i_i$ as well as the combination $\rho + S$. These are given by

$$S^i_i = \frac{1}{2} \sum_{A=1}^2 \left[3 \frac{\Pi_A^2}{\gamma} - \gamma^{ij} \partial_i \phi_A \partial_j \phi_A \right] - \frac{3}{2} U(\phi_0^2), \quad (2.75)$$

$$\rho + S = \sum_{A=1}^2 \left[2 \frac{\Pi_A^2}{\gamma} \right] - U(\phi_0^2). \quad (2.76)$$

2.3.2 Noether Charge

The invariance of the Klein-Gordon Lagrangian density, Eq. (2.58), under a global $U(1)$ symmetry transformation $\Phi \rightarrow \Phi e^{i\epsilon}$ gives rise to a conserved current density according to Noether's theorem. Roughly, this result can be obtained as follows:

First, consider the Klein-Gordon action as a functional of the inverse metric, the scalar field and its first covariant derivative, instead of the inverse metric and the scalar field alone, as in the

last subsection:

$$S_{KG}[g^{ab}, \Phi, \nabla_a \Phi] = \int_{\mathcal{M}} \mathcal{L}_{KG}(g^{ab}, \Phi, \nabla_a \Phi). \quad (2.77)$$

Also note that for a scalar field we have $\nabla_a \Phi = \partial_a \Phi$. Defining the variation of a functional or function with respect to a parameter ϵ as:

$$\delta S[\Phi_\epsilon] \equiv \left. \frac{dS}{d\epsilon} \right|_{\epsilon=0} \quad \text{and} \quad \delta \Phi \equiv \left. \frac{d\Phi}{d\epsilon} \right|_{\epsilon=0}, \quad (2.78)$$

the variation of the Klein-Gordon action functional can then be expanded as:

$$\delta S_{KG} = \int_{\mathcal{M}} \frac{\delta S_{KG}}{\delta g^{ab}} \delta g^{ab} + \frac{\delta S_{KG}}{\delta \Phi} \delta \Phi + \frac{\delta S_{KG}}{\delta(\partial_a \Phi)} \delta(\partial_a \Phi). \quad (2.79)$$

Our interest here is in variations that keep the action functional constant; that is, variations such that $\delta S_{KG} = 0$. The inverse metric g^{ab} is invariant under the action of a $U(1)$ transformation: $\delta g^{ab} = \left. \frac{dg^{ab}}{d\epsilon} \right|_{\epsilon=0} = 0$, and the first term in the Eq. (2.79) drops out. Further simplification of Eq. (2.79) results from noting that the variation with respect to the field derivative can be rewritten as:

$$\delta(\partial_a \Phi) = \left. \frac{d(\partial_a \Phi)}{d\epsilon} \right|_{\epsilon=0} = \partial_a \left(\left. \frac{d\Phi}{d\epsilon} \right|_{\epsilon=0} \right) = \partial_a(\delta \Phi), \quad (2.80)$$

since ordinary derivatives commute. Inserting the above relationship in Eq. (2.79), we have after some simple algebraic manipulation:

$$\delta S_{KG} = \int_{\mathcal{M}} \left\{ \left[\frac{\delta S_{KG}}{\delta \Phi} - \partial_a \left(\frac{\delta S_{KG}}{\delta(\partial_a \Phi)} \right) \right] \delta \Phi + \partial_a \left(\frac{\delta S_{KG}}{\delta(\partial_a \Phi)} \delta \Phi \right) \right\}. \quad (2.81)$$

The first term of the equation above is simply the Klein-Gordon equation of motion which vanishes identically. The second term is a total divergence that can be converted to a surface term using Stokes theorem, and which also has to vanish if the action is supposed to be invariant under the field variation $\delta \Phi$. This then implies that the current density, j^a , associated with the $U(1)$ symmetry and defined by

$$j^a \equiv \frac{\delta S_{KG}}{\delta(\partial_a \Phi)} \delta \Phi, \quad (2.82)$$

is conserved

$$\partial_a j^a = 0. \quad (2.83)$$

A conserved (Noether) charge, Q_N , is associated with the “time” component of the current density:

$$Q_N = \int_{\Sigma_t} j^t, \quad (2.84)$$

where Σ_t is a spacelike hypersurface as previously, and a fixed volume element on Σ_t , \mathbf{e} , is understood in the integration.

To compute the explicit form of the Noether current we apply Eq. (2.82) to the Klein-Gordon Lagrangian (2.58), obtaining

$$\delta\mathcal{L}_{KG} \equiv \partial_a \left(\frac{\delta S_{KG}}{\delta(\partial_a \Phi)} \delta\Phi + \frac{\delta S_{KG}}{\delta(\partial_a \Phi^*)} \delta\Phi^* \right) = \partial_a \left(-\frac{1}{2} \sqrt{-g} g^{ab} (\partial_b \Phi^* \delta\Phi + \partial_b \Phi \delta\Phi^*) \right), \quad (2.85)$$

so the current density is

$$j^a = -\frac{1}{2} \sqrt{-g} g^{ab} (\partial_b \Phi^* \delta\Phi + \partial_b \Phi \delta\Phi^*). \quad (2.86)$$

For an infinitesimal $U(1)$ transformation, $\Phi \rightarrow \Phi + i\epsilon\Phi$, we have

$$\delta\Phi = i\epsilon\Phi \quad \text{and} \quad \delta\Phi^* = -i\epsilon\Phi^*, \quad (2.87)$$

and we have

$$j^a = -i\frac{1}{2} \sqrt{-g} g^{ab} (\Phi \partial_b \Phi^* - \Phi^* \partial_b \Phi), \quad (2.88)$$

where the constant ϵ has been factored out. Using the component form of the field, $\Phi = \phi_1 + i\phi_2$, this can also be expressed as:

$$j^a = \sqrt{-g} g^{ab} (\phi_2 \partial_b \phi_1 - \phi_1 \partial_b \phi_2). \quad (2.89)$$

Finally the time component of the current density in a 3+1 coordinate basis assumes the following form:

$$\begin{aligned} j^t &= \sqrt{-g} \left[g^{tt} (\phi_2 \partial_t \phi_1 - \phi_1 \partial_t \phi_2) + g^{ti} (\phi_2 \partial_i \phi_1 - \phi_1 \partial_i \phi_2) \right] \\ &= \sqrt{-g} \left[-\frac{1}{\alpha^2} (\phi_2 \partial_t \phi_1 - \phi_1 \partial_t \phi_2) + \frac{\beta^i}{\alpha^2} (\phi_2 \partial_i \phi_1 - \phi_1 \partial_i \phi_2) \right] \\ &= \phi_1 \Pi_2 - \phi_2 \Pi_1, \end{aligned} \quad (2.90)$$

where Eq. (2.64) was used to simplify the second line of the above, and to express $\partial_t \phi_A$ in terms of their respective conjugate momenta Π_A . Choosing the fixed volume element \mathbf{e} on Σ_t to be the coordinate volume element d^3x , the Noether charge can be written as:

$$Q_N = \int_{\Sigma_t} (\phi_1 \Pi_2 - \phi_2 \Pi_1) d^3x \quad (2.91)$$

and can be expected to be conserved: i.e. to have the same value on each slice Σ_t of the spacetime foliation. This expression is used in Chap. 5 as one diagnostic to ensure that the numerical code used to solve our model system is producing sensible results.

2.4 The Conformally Flat Approximation (CFA)

This section provides a detailed description of the conformally flat approximation (CFA) of general relativity. Our definition of the CFA includes a particular choice of time coordinate, known as maximal slicing, which is briefly discussed in Sec. 2.4.1. We continue in Sec. 2.4.2 with a detailed review of a specific conformal decomposition of the 3 + 1 Einstein equations. Once the conformal Einstein equations have been derived, we introduce the assumption of conformal flatness in 2.4.3 and derive the simplified set of field equations that result. These equations are covariant with respect to a choice of spatial coordinates, and in Sec. 2.4.4 we fix those coordinates to be Cartesian. This then yields the actual PDEs that we solve numerically. Sec. 2.4.5 discusses three different approaches we investigated for imposing boundary conditions on the PDEs. Finally, Sec. 2.4.6 describes the definition and calculation of the ADM mass, which we use as a diagnostic quantity in our computations.

2.4.1 Maximal Slicing

In the initial value, or Cauchy, formulation of Einstein’s equations, the choice of time coordinate is related to the choice of the spacelike hypersurfaces that foliate spacetime since the hypersurfaces are level surfaces of the time coordinate. As briefly discussed in Sec. 2.2, the embedding of these three dimensional hypersurfaces in the four dimensional spacetime is described by the extrinsic curvature K_{ab} . It is therefore natural to choose the time coordinate by imposing a condition on the extrinsic curvature. One particular choice widely used in numerical relativity demands that trace of the extrinsic curvature tensor vanish:

$$K \equiv K^a{}_a = 0 \quad \text{and} \quad \partial_t K = 0. \quad (2.92)$$

This choice is called *maximal slicing* and it is a particularly useful slicing for numerical computation since it tends to avoid spacetime singularities—such as those that arise from gravitational collapse of matter to a black hole—by “freezing” the evolution in regions close to locations where such singularities are developing [131]. Considering a congruence of worldlines for a family of observers travelling normally to the hypersurfaces, the maximal slicing condition implies that the expansion of the congruence vanishes, inhibiting focusing of the worldlines as well as the formation of caustics. Moreover, as the name suggests, and due to the non-Euclidean signature of spacetime,

$K = 0$ slices have *maximal* volume with respect to small, but arbitrary, deformations of the hypersurfaces. The interested reader is directed to reference [132] for a more detailed discussion of this choice of time slicing and its properties.

2.4.2 Conformal Decomposition of the Einstein Equations

Conformal decompositions of the Einstein equations in the context of the 3 + 1 formalism were first introduced by Lichnerowicz in 1944 [33], who proposed a specific conformal decomposition of the Hamiltonian constraint. The goal of the decomposition was to write the constraint as a manifestly elliptic partial differential equation for a specific part of the 3-metric—namely an overall scale, or conformal factor—and then to establish existence and uniqueness of solutions of the PDE. In the 1970’s and early 1980’s, York and his collaborators (most notably Ó Murchadha) [133, 134, 135, 136, 137, 138, 139, 140], made significant and highly influential advances of Lichnerowicz’s work through a general program aimed at understanding which of the basic dynamical variables in the 3+1 approach should be freely specified at the initial time, and which should be fixed by the constraints. An analogy was made to electromagnetism, where the field can be decomposed into longitudinal and transverse parts, with the former representing the gauge degree of freedom, and the latter encoding the dynamical (radiative) content of the theory. Building on previous work by Deser and others [141, 142], York [136] thus considered similar decompositions for the case of the symmetric rank-2 tensor fields that appear in the 3 + 1 approach: namely, the spatial metric and the extrinsic curvature. The main idea was that a covariant¹³ decomposition of these tensors could yield at least a formal (or perhaps approximate) solution to the constraint equations, as well as providing a route to establishing existence and uniqueness of the solutions. In addition, it was hoped that the process would lead to an identification of the “true” dynamical degrees of freedom of the gravitational field within the 3 + 1 framework. Ideally, these degrees of freedom were to be covariant and freely specifiable, and were to encapsulate the radiative content of general relativity. York’s work resulted in a specific conformal decomposition in which the transverse-traceless part of the dynamical variables, $\{\gamma_{ab}, K_{ab}\}$, were to represent the radiative degrees of freedom.

Now, there is a strong argument to be made that the goals of York’s effort were not entirely successful, especially in terms of identifying radiative degrees of freedom on a single slice in the strong-field regime. However there can be no doubt that his program was an absolutely crucial

¹³Here “covariant” means covariant with respect to coordinate transformations in the hypersurfaces, i.e. with respect to spatial diffeomorphisms.

development for numerical relativists since it allowed the constraint equations to be written as a set of coupled, nonlinear elliptic PDEs, for which existence and uniqueness *could* be established. Moreover the equations could be tackled and solved using standard numerical approaches for elliptic systems. Most importantly for this thesis, the conformal transverse-traceless decomposition lies at the heart of the approximation to the Einstein equations that we adopt for our model.

Thus, reemphasizing that the physical interpretation of York's approach is still a matter of debate, we now proceed to work through the details of the decomposition.¹⁴ We start with the general case in which no assumptions are made about the 3-metric, γ_{ab} , and then specialize in Sec. 2.4.3 to the case of a conformally flat γ_{ab} . Our discussion parallels lecture notes on the 3+1 approach due toourgoulhon [144] and, as mentioned previously, we include the development here largely for the sake of completeness.

The first step towards the conformal decomposition of the Einstein equations in 3+1 form is an investigation of how each of the fields that appear in the equations changes under a conformal transformation. Specifically, we must consider the action of conformal scalings on the components of 1) the 3-metric, γ_{ij} , 2) the connection, C^i_{jk} , associated with the spatial covariant derivative D_i , 3) the 3-Ricci tensor, \mathcal{R}_{ij} , and, finally, 4) the extrinsic curvature K_{ij} . Each of these is discussed in turn below, after which results are assembled and used in the decomposition of the field equations themselves.

For the case of the induced spatial metric, γ_{ij} , conformal decomposition means that we introduce a base metric, $\tilde{\gamma}_{ij}$, and a strictly positive function, $\psi \equiv \psi(x^\mu)$, known as the *conformal factor*, and then write

$$\gamma_{ij} \equiv \psi^4 \tilde{\gamma}_{ij}. \quad (2.93)$$

We note that $\tilde{\gamma}_{ij}$ is frequently known as the *conformal metric* and emphasize that it has no direct physical interpretation. Defining the inverse of the conformal metric, $\tilde{\gamma}^{jk}$, in the usual way, so that $\tilde{\gamma}_{ij} \tilde{\gamma}^{jk} = \delta^k_i$, and noting that $\gamma_{ij} \gamma^{jk} = \delta^k_i$ we have

$$\gamma^{ij} = \psi^{-4} \tilde{\gamma}^{ij}. \quad (2.94)$$

Proceeding to the connection, we first recall that any covariant derivative, D_i , can be defined

¹⁴The reader who is interested in more details concerning transverse-traceless decompositions and their relationship to gravitational radiation in general is directed to the review paper by Thorne [143].

in terms of the ordinary derivative operator, ∂_i via

$$D_i v^j = \partial_i v^j + \Gamma^j_{ik} v^k, \quad (2.95)$$

where v^j are the components of a spatial vector and Γ^j_{ik} are the Christoffel symbols associated with D_i . If the covariant derivative is chosen to be compatible with the spatial metric, as will be done here, then we have $D_i \gamma_{jk} = 0$ and the Christoffel symbols can be calculated using the usual formula:

$$\Gamma^j_{ik} = \frac{1}{2} \gamma^{jl} (\partial_i \gamma_{kl} + \partial_k \gamma_{il} - \partial_l \gamma_{ik}). \quad (2.96)$$

Similarly, choosing a covariant derivative \tilde{D}_i compatible with the conformal metric, so that $\tilde{D}_i \tilde{\gamma}_{jk} = 0$, we have:

$$\tilde{\Gamma}^j_{ik} = \frac{1}{2} \tilde{\gamma}^{jl} (\partial_i \tilde{\gamma}_{kl} + \partial_k \tilde{\gamma}_{il} - \partial_l \tilde{\gamma}_{ik}). \quad (2.97)$$

The two covariant derivatives D_i and \tilde{D}_i can be related through the connection tensor, C^j_{ik} :

$$D_i v^j = \tilde{D}_i v^j + C^j_{ik} v^k, \quad (2.98)$$

where C^j_{ik} is calculated in terms of conformal covariant derivatives as follows:

$$C^j_{ik} = \frac{1}{2} \gamma^{jl} \left(\tilde{D}_i \gamma_{kl} + \tilde{D}_k \gamma_{il} - \tilde{D}_l \gamma_{ik} \right). \quad (2.99)$$

This last expression can be rewritten in terms of the conformal metric using equations (2.93) and (2.94) along with the fact that $\tilde{D}_i \tilde{\gamma}_{jk} = 0$. After some manipulation, we find:

$$C^j_{ik} = 2\tilde{\gamma}^{jl} \left[\tilde{D}_i (\ln \psi) \tilde{\gamma}_{kl} + \tilde{D}_k (\ln \psi) \tilde{\gamma}_{il} - \tilde{D}_l (\ln \psi) \tilde{\gamma}_{ik} \right]. \quad (2.100)$$

The contraction of the above equation on its first and second indices, or, due to the symmetry $C^j_{ik} = C^j_{ki}$, on its first and third indices, can be used to provide a useful expression for the divergence of a vector in terms of a conformal divergence of an appropriate conformal scaling of the vector:

$$D_i v^i = \tilde{D}_i v^i + C^i_{ik} v^k = \tilde{D}_i v^i + 6\tilde{D}_i (\ln \psi) v^i = \frac{1}{\psi^6} \tilde{D}_i (\psi^6 v^i). \quad (2.101)$$

We next reexpress the spatial Ricci tensor, \mathcal{R}_{ik} , and Ricci scalar, \mathcal{R} , in terms of their conformal

counterparts, $\tilde{\mathcal{R}}_{ik}$, and $\tilde{\mathcal{R}}$, as well as additional terms involving conformal derivatives of ψ . To that end we start from the definition of the spatial Riemann tensor and then rewrite it in terms of the connection tensor $C^j{}_{ik}$. From the definition (2.23) we have

$$\mathcal{R}_{ijk}{}^l w_l = 2D_{[i}D_{j]}w_k. \quad (2.102)$$

In addition, the covariant derivative of a tensor of type (p, q) is given by

$$D_k T^{i_1 \dots i_p}{}_{j_1 \dots j_q} = \tilde{D}_k T^{i_1 \dots i_p}{}_{j_1 \dots j_q} + \sum_{l=1}^p C^{i_l}{}_{ks} T^{i_1 \dots s \dots i_p}{}_{j_1 \dots j_q} - \sum_{l=1}^q C^s{}_{k j_l} T^{i_1 \dots i_p}{}_{j_1 \dots s \dots j_q}. \quad (2.103)$$

Using (2.103) in (2.102), and performing some algebraic simplifications we find:

$$\mathcal{R}_{ijk}{}^l = \tilde{\mathcal{R}}_{ijk}{}^l - 2\tilde{D}_{[i}C^l{}_{j]k} + 2C^m{}_{k[i}C^l{}_{j]m}, \quad (2.104)$$

The Ricci tensor is now given by contracting this last equation on its second and fourth indices, yielding

$$\mathcal{R}_{ik} \equiv \mathcal{R}_{ilk}{}^l = \tilde{\mathcal{R}}_{ik} - 2\tilde{D}_{[i}C^l{}_{l]k} + 2C^m{}_{k[i}C^l{}_{l]m}. \quad (2.105)$$

Using (2.100) in (2.105), we have the desired relationship between the physical and conformal spatial Ricci tensors.

$$\begin{aligned} \mathcal{R}_{ik} &= \tilde{\mathcal{R}}_{ik} - 2\tilde{D}_i\tilde{D}_k(\ln \psi) - 2\tilde{\gamma}_{ik}\tilde{\gamma}^{lm}\tilde{D}_l\tilde{D}_m(\ln \psi) \\ &+ 4\tilde{D}_i(\ln \psi)\tilde{D}_k(\ln \psi) - 4\tilde{\gamma}_{ik}\tilde{\gamma}^{lm}\tilde{D}_l(\ln \psi)\tilde{D}_m(\ln \psi). \end{aligned} \quad (2.106)$$

Taking the trace of this equation then provides an expression relating the physical and conformal Ricci scalars:

$$\begin{aligned} \mathcal{R} \equiv \gamma^{ik}\mathcal{R}_{ik} &= \psi^{-4} \left[\tilde{\mathcal{R}} - 8\tilde{\gamma}^{ik}\tilde{D}_i\tilde{D}_k(\ln \psi) - 8\tilde{\gamma}^{ik}\tilde{D}_i(\ln \psi)\tilde{D}_k(\ln \psi) \right] \\ &= \psi^{-4}\tilde{\mathcal{R}} - 8\psi^{-5}\tilde{\gamma}^{ik}\tilde{D}_i\tilde{D}_k\psi, \end{aligned} \quad (2.107)$$

where $\tilde{\mathcal{R}} \equiv \tilde{\gamma}^{ik}\tilde{\mathcal{R}}_{ik}$. One sees immediately from this last equation one of the crucial results of Lichnerowicz's approach: namely that the conformal transformation "pulls out" from the extremely complicated expression for \mathcal{R} (when written out in full for the case of a generic 3-metric) a term

that is proportional to the covariant Laplacian of the conformal factor, $\tilde{\gamma}^{ik}\tilde{D}_i\tilde{D}_k\psi$

Before moving on to a conformal treatment of the Einstein equations themselves, we need to perform a conformal decomposition of the extrinsic curvature tensor, K_{ij} . In anticipation of our use of the decomposition in conjunction with maximal slicing, $K \equiv K^i_i = 0$, it is convenient to first write the tensor as a sum of traceless and traceful pieces:

$$K_{ij} = A_{ij} + \frac{1}{3}K\gamma_{ij}, \quad (2.108)$$

$$K^{ij} = A^{ij} + \frac{1}{3}K\gamma^{ij}, \quad (2.109)$$

where A_{ij} and A^{ij} are, by definition, the traceless parts of K_{ij} and K^{ij} , respectively, so that $\gamma^{ij}A_{ij} = \gamma_{ij}A^{ij} = 0$. In the following it will eventually be A^{ij} — i.e. the traceless part of K^{ij} — that we will conformally transform according to

$$A^{ij} \equiv \psi^s \tilde{A}^{ij}. \quad (2.110)$$

The non-zero conformal power, s , appearing in this definition can be chosen so that convenient mathematical relationships result. In this regard it turns out to be natural to start from the evolution equations (2.46) for the spatial metric components. We recall that these are given by

$$\mathcal{L}_m \gamma_{ij} \equiv \mathcal{L}_{(t-\beta)} \gamma_{ij} = -2\alpha K_{ij}, \quad (2.111)$$

where $m^a = t^a - \beta^a = \alpha n^a$. Using (2.108) and (2.93) in the above, we find

$$\mathcal{L}_m \tilde{\gamma}_{ij} = -2\alpha\psi^{-4}A_{ij} - \frac{2}{3}\alpha K \tilde{\gamma}_{ij} - 4\tilde{\gamma}_{ij}\mathcal{L}_m(\ln\psi). \quad (2.112)$$

Note that this equation involves time derivatives (through the Lie derivatives along m^a) for both the conformal metric and the conformal factor itself, and in this sense is a “coupled” evolution equation for $\tilde{\gamma}_{ij}$ and ψ . Since we wish to treat these quantities as dynamically independent, we must perform some manipulation to get decoupled evolution equations. We start by taking the trace of (2.112) by contracting both sides with $\tilde{\gamma}^{ij}$, and making use of $\tilde{\gamma}^{ij}A_{ij} = \psi^{-4}\gamma^{ij}A_{ij} = 0$:

$$\tilde{\gamma}^{ij}\mathcal{L}_m\tilde{\gamma}_{ij} = -2\alpha K - 12\mathcal{L}_m(\ln\psi). \quad (2.113)$$

We now use the well known formula

$$\delta(\ln \det M) \equiv \text{tr} (M^{-1} \cdot \delta M) \quad (2.114)$$

where M is an invertible matrix, δ represents an arbitrary derivative operator, tr denotes the trace operation and \cdot is matrix multiplication, to rewrite the left hand side of (2.113) as

$$\tilde{\gamma}^{ij} \mathcal{L}_m \tilde{\gamma}_{ij} = \mathcal{L}_m (\ln \det \tilde{\gamma}_{ij}) = \mathcal{L}_m (\ln \tilde{\gamma}) , \quad (2.115)$$

where $\tilde{\gamma} \equiv \det(\tilde{\gamma}_{ij})$. Now, in the approach to the conformal decomposition of the Einstein equations that we are following, a key demand is that the determinant of the conformal spatial metric be Lie dragged from hypersurface to hypersurface along the vector field t^a :

$$\mathcal{L}_t \tilde{\gamma} = 0 . \quad (2.116)$$

Using (2.116) in (2.115) we have

$$\tilde{\gamma}^{ij} \mathcal{L}_m \tilde{\gamma}_{ij} = -\mathcal{L}_\beta (\ln \tilde{\gamma}) . \quad (2.117)$$

Moreover, again using (2.114) we can write

$$-\mathcal{L}_\beta (\ln \tilde{\gamma}) = -\tilde{\gamma}^{ij} \mathcal{L}_\beta \tilde{\gamma}_{ij} , \quad (2.118)$$

so that (2.117) becomes

$$\tilde{\gamma}^{ij} \mathcal{L}_m \tilde{\gamma}_{ij} = -\tilde{\gamma}^{ij} \mathcal{L}_\beta \tilde{\gamma}_{ij} = -2\tilde{D}_i \beta^i , \quad (2.119)$$

where we have used the definition (2.37) in the last equality above. We can then use this last result to eliminate the term $\tilde{\gamma}^{ij} \mathcal{L}_m \tilde{\gamma}_{ij}$ that appears in Eq. (2.113). Performing the substitution and some trivial manipulation and rearrangement, we have (recalling that $m^a = t^a - \beta^a$)

$$\mathcal{L}_m \ln \psi = (\partial_t - \mathcal{L}_\beta) \ln \psi = \frac{1}{6} \left(\tilde{D}_i \beta^i - \alpha K \right) . \quad (2.120)$$

This then provides the desired (decoupled) evolution equation for the conformal factor, ψ .

Additionally, (2.120) can now be used in (2.112) to yield a set of evolution equations for the

conformal metric components:

$$\begin{aligned}
(\partial_t - \mathcal{L}_\beta) \tilde{\gamma}_{ij} &= -2\alpha\psi^{-4} A_{ij} - \frac{2}{3}\alpha K \tilde{\gamma}_{ij} - 4\tilde{\gamma}_{ij} \mathcal{L}_m(\ln \psi) \\
&= -2\alpha\psi^{-4} A_{ij} - \frac{2}{3}\alpha K \tilde{\gamma}_{ij} - \frac{2}{3}\tilde{\gamma}_{ij} \left(\tilde{D}_k \beta^k - \alpha K \right) \\
&= -2\alpha \tilde{A}_{ij} - \frac{2}{3}\tilde{\gamma}_{ij} \tilde{D}_k \beta^k,
\end{aligned} \tag{2.121}$$

where we have (suggestively) defined $\tilde{A}_{ij} \equiv \psi^{-4} A_{ij}$. Note that \tilde{A}_{ij} is traceless

$$\tilde{\gamma}^{ij} \tilde{A}_{ij} = \psi^4 \gamma^{ij} \psi^{-4} A_{ij} = 0, \tag{2.122}$$

and that the contravariant components, \tilde{A}^{ij} , are given by

$$\tilde{A}^{ij} \equiv \tilde{\gamma}^{ik} \tilde{\gamma}^{jl} \tilde{A}_{kl} A^{ij} = \psi^{-4} A^{ij}, \tag{2.123}$$

Thus, the conformal decomposition of the evolution equation for the spatial metric suggests that we choose $s = -4$ for the conformal exponent in

$$A^{ij} \equiv \psi^s \tilde{A}^{ij}. \tag{2.124}$$

It is worth emphasizing, however, that the choice $s = -4$ is not unique in terms of leading to simplifications (or “naturalness”) in the conformal equations. Another common scaling that was first adopted by Lichnerowicz [33] is $s = -10$. This choice also arises naturally from a decomposition of one of the Einstein equations, but this time when the decomposition is applied to the momentum constraint equations. Since we do not use this specific decomposition in the current work, the reader who is interested in further details concerning it is directed toourgoulhon’s lectures [144].

We now turn to the task of deriving conformal evolution equations for \tilde{A}_{ij} and K . This will be done in two stages. We will first use the 3 + 1 evolution equations for K_{ij} , as given by (2.48), to derive evolution equations for the trace-free components, A_{ij} , and the trace, K . These equations will be expressed in terms of physical (i.e. non conformally scaled) quantities, but will then be used in the second stage to derive update equations for \tilde{A}_{ij} and K that are written in terms of the conformally scaled variables.

We thus start from (2.48), which we recall reads

$$\mathcal{L}_m K_{ij} = -D_i D_j \alpha + \alpha \left[\mathcal{R}_{ij} + K K_{ij} - 2K_{ik} K^{kj} + 8\pi \left(\frac{1}{2} \gamma_{ij} (S - \rho) - S_{ij} \right) \right]. \quad (2.125)$$

Now from definition (2.108) of the decomposition of K_{ij} , we have $K_{ij} = A_{ij} + \frac{1}{3} K \gamma_{ij}$, so

$$\mathcal{L}_m K_{ij} = \mathcal{L}_m A_{ij} + \frac{1}{3} \gamma_{ij} \mathcal{L}_m K + \frac{1}{3} K \mathcal{L}_m \gamma_{ij} = \mathcal{L}_m A_{ij} + \frac{1}{3} \gamma_{ij} \mathcal{L}_m K - \frac{2}{3} \alpha K K_{ij}, \quad (2.126)$$

where we have used (2.111) in the last step. Furthermore, we have

$$\mathcal{L}_m K = \mathcal{L}_m (\gamma^{ij} K_{ij}) = \gamma^{ij} \mathcal{L}_m K_{ij} + K_{ij} \mathcal{L}_m \gamma^{ij} = \gamma^{ij} \mathcal{L}_m K_{ij} + 2\alpha K_{ij} K^{ij}, \quad (2.127)$$

where we have used (2.47)

$$\mathcal{L}_m \gamma^{ij} \equiv \mathcal{L}_{(t-\beta)} \gamma^{ij} = 2\alpha K^{kj}. \quad (2.128)$$

In addition, taking the trace of (2.125) we find

$$\gamma^{ij} \mathcal{L}_m K_{ij} = -\gamma^{ij} D_i D_j \alpha + \alpha [\mathcal{R} + K^2 - 2K_{ij} K^{ij} + 4\pi (S - 3\rho)]. \quad (2.129)$$

Substitution of this last result in (2.127) gives

$$\mathcal{L}_m K = -\gamma^{ij} D_i D_j \alpha + \alpha [\mathcal{R} + K^2 + 4\pi (S - 3\rho)]. \quad (2.130)$$

Now the 3-Ricci scalar \mathcal{R} is an extremely complicated function of γ_{ij} and its first and second derivatives. It is thus computationally advantageous to use the Hamiltonian constraint (2.49) in the form

$$\mathcal{R} = -K^2 + K_{ij} K^{ij} + 16\pi\rho \quad (2.131)$$

to eliminate \mathcal{R} from (2.130). This yields the evolution equation for K in its final form:

$$\mathcal{L}_m K \equiv (\partial_t - \mathcal{L}_\beta) K = -\gamma^{ij} D_i D_j \alpha + \alpha [K_{ij} K^{ij} + 4\pi (S + \rho)]. \quad (2.132)$$

We now continue by deriving the evolution equation for the the trace free components, A_{ij} , of

the physical extrinsic curvature tensor. We first solve (2.126) for $\mathcal{L}_m A_{ij}$:

$$\mathcal{L}_m A_{ij} = \mathcal{L}_m K_{ij} - \frac{1}{3}\gamma_{ij}\mathcal{L}_m K + \frac{2}{3}\alpha K K_{ij}. \quad (2.133)$$

Using (2.125) and (2.132) to replace the terms $\mathcal{L}_m K_{ij}$ and $\mathcal{L}_m K$, respectively, we find

$$\begin{aligned} \mathcal{L}_m A_{ij} = -D_i D_j \alpha &+ \alpha \left[\mathcal{R}_{ij} + \frac{5}{3} K K_{ij} - 2 K_{ik} K^k{}_j - 8\pi \left(S_{ij} - \frac{1}{3} \gamma_{ij} S \right) \right] \\ &+ \frac{1}{3} \gamma_{ij} \left[\gamma^{lk} D_l D_k \alpha - \alpha (\mathcal{R} + K^2) \right]. \end{aligned} \quad (2.134)$$

Using the decompositions

$$K_{ij} = A_{ij} + \frac{1}{3} K \gamma_{ij}, \quad (2.135)$$

$$K^i{}_j = A^i{}_j + \frac{1}{3} K \delta^i{}_j, \quad (2.136)$$

in the above, we find, after some manipulation

$$\begin{aligned} \mathcal{L}_m A_{ij} = -D_i D_j \alpha &+ \alpha \left[\mathcal{R}_{ij} + \frac{1}{3} K A_{ij} - 2 A_{ik} A^k{}_j - 8\pi \left(S_{ij} - \frac{1}{3} \gamma_{ij} S \right) \right] \\ &+ \frac{1}{3} \gamma_{ij} \left(\gamma^{lk} D_l D_k \alpha - \alpha \mathcal{R} \right). \end{aligned} \quad (2.137)$$

This is the final form of the evolution equation for A_{ij} , which together with the evolution equation (2.132) provides an equivalent system to the original 3 + 1 equation (2.125) for K_{ij} .

We now proceed to the second stage of the current calculation which involves reexpressing (2.137) and (2.132) in terms of the conformally rescaled variables. We begin by considering the left hand side of (2.137), which can be rewritten as

$$\mathcal{L}_m A_{ij} = \mathcal{L}_m \left(\psi^4 \tilde{A}_{ij} \right) = \psi^4 \mathcal{L}_m \tilde{A}_{ij} + 4\psi^3 \mathcal{L}_m \psi \tilde{A}_{ij}. \quad (2.138)$$

Using (2.120) to eliminate the $\mathcal{L}_m \psi$ term in this last expression we have

$$\mathcal{L}_m A_{ij} = \psi^4 \left[\mathcal{L}_m \tilde{A}_{ij} + \frac{2}{3} \tilde{A}_{ij} \left(\tilde{D}_l \beta^l - \alpha K \right) \right]. \quad (2.139)$$

Replacing the left hand side of (2.137) with the right hand side of the above equation, and solving

for $\mathcal{L}_m \tilde{A}_{ij}$, we find

$$\begin{aligned} \mathcal{L}_m \tilde{A}_{ij} &= \psi^{-4} \left\{ -D_i D_j \alpha + \alpha \left[\mathcal{R}_{ij} + \frac{1}{3} K A_{ij} - 2A_{ik} A_j^k - 8\pi \left(S_{ij} - \frac{1}{3} \gamma_{ij} S \right) \right] \right. \\ &\quad \left. + \frac{1}{3} \gamma_{ij} (\gamma^{lk} D_l D_k \alpha - \alpha \mathcal{R}) \right\} - \frac{2}{3} \tilde{A}_{ij} (\tilde{D}_l \beta^l - \alpha K). \end{aligned} \quad (2.140)$$

Proceeding, we must also rewrite the terms involving the physical covariant derivatives of the lapse function, α , using the derivative operator, \tilde{D}_i , which is compatible with the conformal metric $\tilde{\gamma}_{ij}$. Specifically, we have

$$\begin{aligned} D_i D_j \alpha &= \tilde{D}_i (D_j \alpha) - C_{ij}^k (D_k \alpha) \\ &= \tilde{D}_i \tilde{D}_j \alpha - C_{ij}^k \tilde{D}_k \alpha \\ &= \tilde{D}_i \tilde{D}_j \alpha - 2\tilde{D}_k \alpha \left[\tilde{D}_i (\ln \psi) \delta_j^k + \tilde{D}_j (\ln \psi) \delta_i^k - \tilde{\gamma}_{ij} \tilde{\gamma}^{kl} \tilde{D}_l (\ln \psi) \right] \\ &= \tilde{D}_i \tilde{D}_j \alpha - 2\tilde{D}_i (\ln \psi) \tilde{D}_j \alpha - 2\tilde{D}_j (\ln \psi) \tilde{D}_i \alpha + 2\tilde{\gamma}_{ij} \tilde{\gamma}^{kl} \tilde{D}_l (\ln \psi) \tilde{D}_k \alpha, \end{aligned} \quad (2.141)$$

where the definitions (2.103) and (2.100) were used in the first and third lines, respectively. We also need the trace of this equation, which is given by

$$\gamma^{ij} D_i D_j \alpha = \psi^{-4} \tilde{\gamma}^{ij} D_i D_j \alpha = \psi^{-4} \left[\tilde{\gamma}^{ij} \tilde{D}_i \tilde{D}_j \alpha + 2\tilde{\gamma}^{ij} \tilde{D}_i (\ln \psi) \tilde{D}_j \alpha \right]. \quad (2.142)$$

We have now expressed all of the terms that appear in the evolution equation (2.140) for \tilde{A}_{ij} in terms of conformally scaled quantities. Assembling results, we find after a considerable amount of algebraic simplification that

$$\begin{aligned} (\partial_t - \mathcal{L}_\beta) \tilde{A}_{ij} &= \psi^{-4} \left\{ -\tilde{D}_i \tilde{D}_j \alpha + 4\tilde{D}_i (\ln \psi) \tilde{D}_j \alpha + \frac{1}{3} \tilde{\gamma}_{ij} \tilde{\gamma}^{kl} (\tilde{D}_k \tilde{D}_l \alpha - 4\tilde{D}_k (\ln \psi) \tilde{D}_l \alpha) \right. \\ &\quad + \alpha \left[\tilde{\mathcal{R}}_{ij} - \frac{1}{3} \tilde{\gamma}_{ij} \tilde{\mathcal{R}} - 2\tilde{D}_i \tilde{D}_j (\ln \psi) + 4\tilde{D}_i (\ln \psi) \tilde{D}_j (\ln \psi) \right. \\ &\quad \left. \left. + \frac{2}{3} \tilde{\gamma}_{ij} \tilde{\gamma}^{kl} (\tilde{D}_k \tilde{D}_l (\ln \psi) - 2\tilde{D}_k (\ln \psi) \tilde{D}_l (\ln \psi)) \right] \right\} \\ &\quad + \alpha \left[K \tilde{A}_{ij} - 2\tilde{\gamma}^{kl} \tilde{A}_{ik} \tilde{A}_{lj} - 8\pi \left(\psi^{-4} S_{ij} - \frac{1}{3} \tilde{\gamma}_{ij} S \right) \right] - \frac{2}{3} \tilde{A}_{ij} \tilde{D}_l \beta^l. \end{aligned} \quad (2.143)$$

Here we have used equations (2.106), (2.107), (2.141) and (2.142) as well as $A_{ij} = \psi^4 \tilde{A}_{ij}$.

We can now also write down the evolution equation for K in terms of the conformally scaled variables. To do so we first note that from the decompositions (2.108) and (2.109), and using the fact that A_{ij} is traceless, we have

$$K_{ij}K^{ij} = \left(A_{ij} + \frac{1}{3}K\gamma_{ij} \right) \left(A^{ij} + \frac{1}{3}K\gamma^{ij} \right) = \tilde{A}_{ij}\tilde{A}^{ij} + \frac{1}{3}K^2, \quad (2.144)$$

Using this last result as well as (2.142) in (2.132), we find

$$(\partial_t - \mathcal{L}_\beta)K = -\psi^{-4}\tilde{\gamma}^{kl} \left[\tilde{D}_k\tilde{D}_l\alpha + 2\tilde{D}_k(\ln\psi)\tilde{D}_l\alpha \right] + \alpha \left[\tilde{A}_{kl}\tilde{A}^{kl} + \frac{1}{3}K^2 + 4\pi(S + \rho) \right]. \quad (2.145)$$

Eqs. (2.143) and (2.145) are our desired evolution equations for the extrinsic curvature quantities within the conformal framework.

Having dealt with the evolution equations for the spatial metric and extrinsic curvature, we are left with the conformal treatment of the constraint equations.

First, using (2.107) and (2.144) in the Hamiltonian constraint (2.49), we readily find

$$\tilde{\gamma}^{kl}\tilde{D}_k\tilde{D}_l\psi - \frac{1}{8}\psi\tilde{\mathcal{R}} + \left(\frac{1}{8}\tilde{A}_{kl}\tilde{A}^{kl} - \frac{1}{12}K^2 + 2\pi\rho \right) \psi^5 = 0. \quad (2.146)$$

Second, in order to rewrite the momentum constraint, we first calculate

$$\begin{aligned} D_j K^{ij} &= D_j A^{ij} + \frac{1}{3}\gamma^{ij}D_j K \\ &= \tilde{D}_j A^{ij} + C^i_{jk}A^{jk} + C^j_{jk}A^{ik} + \frac{1}{3}\psi^{-4}\tilde{\gamma}^{ij}\tilde{D}_j K \\ &= \tilde{D}_j A^{ij} + 4\tilde{D}_j(\ln\psi)A^{ij} + 6\tilde{D}_j(\ln\psi)A^{ij} + \frac{1}{3}\psi^{-4}\tilde{D}^i K \\ &= \tilde{D}_j A^{ij} + \psi^{-10}\tilde{D}_j(\psi^{10}A^{ij}) + \frac{1}{3}\psi^{-4}\tilde{D}^i K. \end{aligned} \quad (2.147)$$

In the above derivation we have used the following: (2.108) in writing down the first line; (2.103), as well as the fact that the actions of D_i and \tilde{D}_i on scalar functions such as K are identical, in going from the first to the second line; expression (2.100) for the connection, and the fact that A_{ij} is traceless in going from the second to the third line. Using (2.147) in the momentum constraint (2.50) then yields

$$\tilde{D}_j A^{ij} + \psi^{-10}\tilde{D}_j(\psi^{10}A^{ij}) - \frac{2}{3}\psi^{-4}\tilde{D}^i K = 8\pi J^i, \quad (2.148)$$

and using $A^{ij} = \psi^{-4} \tilde{A}^{ij}$ we have

$$\tilde{D}_j \tilde{A}^{ij} + 6\tilde{D}_j (\ln \psi) \tilde{A}^{ij} - \frac{2}{3} \tilde{D}^i K = 8\pi\psi^4 J^i. \quad (2.149)$$

We have now completed the task of conformally decomposing the Einstein field equations within the 3 + 1 formalism, and it is thus appropriate to collect the results.

We have the following set of evolution equations for the conformal factor, ψ , the conformal metric γ_{ij} , the trace of the extrinsic curvature, K , and the conformally scaled, trace-free part of the extrinsic curvature, \tilde{A}_{ij} :

$$(\partial_t - \mathcal{L}_\beta) \psi = \frac{1}{6} \psi \left(\tilde{D}_i \beta^i - \alpha K \right), \quad (2.150)$$

$$(\partial_t - \mathcal{L}_\beta) \tilde{\gamma}_{ij} = -2\alpha \tilde{A}_{ij} - \frac{2}{3} \tilde{\gamma}_{ij} \tilde{D}_k \beta^k, \quad (2.151)$$

$$(\partial_t - \mathcal{L}_\beta) K = -\psi^{-4} \tilde{\gamma}^{kl} \left[\tilde{D}_k \tilde{D}_l \alpha + 2\tilde{D}_k (\ln \psi) \tilde{D}_l \alpha \right] + \alpha \left[\tilde{A}_{kl} \tilde{A}^{kl} + \frac{1}{3} K^2 + 4\pi (S + \rho) \right], \quad (2.152)$$

$$\begin{aligned} (\partial_t - \mathcal{L}_\beta) \tilde{A}_{ij} &= \psi^{-4} \left\{ -\tilde{D}_i \tilde{D}_j \alpha + 4\tilde{D}_{(i} (\ln \psi) \tilde{D}_{j)} \alpha + \frac{1}{3} \tilde{\gamma}_{ij} \tilde{\gamma}^{kl} \left(\tilde{D}_k \tilde{D}_l \alpha - 4\tilde{D}_k (\ln \psi) \tilde{D}_l \alpha \right) \right. \\ &+ \alpha \left[\tilde{\mathcal{R}}_{ij} - \frac{1}{3} \tilde{\gamma}_{ij} \tilde{\mathcal{R}} - 2\tilde{D}_i \tilde{D}_j (\ln \psi) + 4\tilde{D}_i (\ln \psi) \tilde{D}_j (\ln \psi) \right. \\ &+ \left. \left. \frac{2}{3} \tilde{\gamma}_{ij} \tilde{\gamma}^{kl} \left(\tilde{D}_k \tilde{D}_l (\ln \psi) - 2\tilde{D}_k (\ln \psi) \tilde{D}_l (\ln \psi) \right) \right] \right\} \\ &+ \alpha \left[K \tilde{A}_{ij} - 2\tilde{\gamma}^{kl} \tilde{A}_{ik} \tilde{A}_{lj} - 8\pi \left(\psi^{-4} S_{ij} - \frac{1}{3} \tilde{\gamma}_{ij} S \right) \right] - \frac{2}{3} \tilde{A}_{ij} \tilde{D}_l \beta^l. \end{aligned} \quad (2.153)$$

We also have the Hamiltonian and momentum constraints:

$$\tilde{\gamma}^{kl} \tilde{D}_k \tilde{D}_l \psi - \frac{1}{8} \psi \tilde{\mathcal{R}} + \left(\frac{1}{8} \tilde{A}_{kl} \tilde{A}^{kl} - \frac{1}{12} K^2 + 2\pi \rho \right) \psi^5 = 0, \quad (2.154)$$

$$\tilde{D}_j \tilde{A}^{ij} + 6\tilde{D}_j (\ln \psi) \tilde{A}^{ij} - \frac{2}{3} \tilde{D}^i K = 8\pi\psi^4 J^i. \quad (2.155)$$

The above system of 3 + 1 conformal Einstein equations is to be solved for the conformal unknowns ψ , $\tilde{\gamma}_{ij}$, K and \tilde{A}_{ij} . Once the conformal variables have been computed, the physical quantities—namely the spatial metric γ_{ij} and the extrinsic curvature K_{ij} —can be recovered

from (2.93), (2.108) and $A_{ij} = \psi^4 \tilde{A}_{ij}$:

$$\gamma_{ij} = \psi^4 \tilde{\gamma}_{ij}, \quad (2.156)$$

$$K_{ij} = \psi^4 \left(\tilde{A}_{ij} + \frac{1}{3} K \tilde{\gamma}_{ij} \right). \quad (2.157)$$

2.4.3 The CFA Equations

We are now in a position to derive the equations that define the approximation of general relativity that we adopt in this thesis. As discussed previously (Sec. 2.4.2), a key element of the approximation is the demand that the 3-metric, γ_{ij} , be conformally flat, so that $\tilde{\gamma}_{ij}$ is the constant, flat 3-metric, which we denote $\hat{\gamma}_{ij}$. This motivates the nomenclature ‘‘Conformally Flat Approximation’’ (CFA) that we have adopted. However, we emphasize that we combine the demand of conformal flatness with the particular time coordinatization given by the maximal slicing condition, which requires that $K = 0$ on each hypersurface. Together, conformal flatness and maximal slicing result in a significant simplification of the system of conformal Einstein equations that was displayed at the end of the previous section.

First, denoting by \hat{D}_i the covariant derivative compatible with the flat 3-metric $\hat{\gamma}_{ij}$, and using $K = 0$, the evolution equation (2.150) for the conformal factor ψ reduces to

$$(\partial_t - \mathcal{L}_\beta) \psi = \frac{1}{6} \psi \hat{D}_i \beta^i. \quad (2.158)$$

Next, the evolution equation (2.151) for $\hat{\gamma}_{ij}$, becomes

$$(\partial_t - \mathcal{L}_\beta) \hat{\gamma}_{ij} = -2\alpha \tilde{A}_{ij} - \frac{2}{3} \hat{\gamma}_{ij} \hat{D}_k \beta^k. \quad (2.159)$$

Continuing, since K must identically vanish on all slices, we have $(\partial_t - \mathcal{L}_\beta)K = 0$. Thus equation (2.152) becomes

$$0 = -\psi^{-4} \hat{\gamma}^{kl} \left[\hat{D}_k \hat{D}_l \alpha + 2 \hat{D}_k (\ln \psi) \hat{D}_l \alpha \right] + \alpha \left[\tilde{A}_{kl} \tilde{A}^{kl} + 4\pi (S + \rho) \right], \quad (2.160)$$

which is now interpreted as an elliptic equation that the lapse function, α , must satisfy on each slice in order that the hypersurfaces remain maximal.

Proceeding to equation (2.153), we note that since the conformal metric is flat, the associated

Ricci tensor, $\tilde{\mathcal{R}}_{ij} = \hat{\mathcal{R}}_{ij}$ and Ricci scalar, $\tilde{\mathcal{R}} = \hat{\mathcal{R}}$, both vanish. Using this fact, along with $K = 0$, the evolution equation for \tilde{A}_{ij} becomes

$$\begin{aligned}
(\partial_t - \mathcal{L}_\beta) \tilde{A}_{ij} &= \psi^{-4} \left\{ -\hat{D}_i \hat{D}_j \alpha + 4\hat{D}_{(i} (\ln \psi) \hat{D}_{j)} \alpha + \frac{1}{3} \hat{\gamma}_{ij} \hat{\gamma}^{kl} \left(\hat{D}_k \hat{D}_l \alpha - 4\hat{D}_k (\ln \psi) \hat{D}_l \alpha \right) \right. \\
&+ \alpha \left[-2\hat{D}_i \hat{D}_j (\ln \psi) + 4\hat{D}_i (\ln \psi) \hat{D}_j (\ln \psi) \right. \\
&+ \left. \left. \frac{2}{3} \hat{\gamma}_{ij} \hat{\gamma}^{kl} \left(\hat{D}_k \hat{D}_l (\ln \psi) - 2\hat{D}_k (\ln \psi) \hat{D}_l (\ln \psi) \right) \right] \right\} \\
&+ \alpha \left[-2\hat{\gamma}^{kl} \tilde{A}_{ik} \tilde{A}_{lj} - 8\pi \left(\psi^{-4} S_{ij} - \frac{1}{3} \hat{\gamma}_{ij} S \right) \right] - \frac{2}{3} \tilde{A}_{ij} \hat{D}_l \beta^l. \tag{2.161}
\end{aligned}$$

Turning to the Hamiltonian constraint (2.154), and again using $\tilde{\mathcal{R}} = 0$ and $K = 0$, we find

$$\hat{\gamma}^{kl} \hat{D}_k \hat{D}_l \psi + \left(\frac{1}{8} \tilde{A}_{kl} \tilde{A}^{kl} + 2\pi\rho \right) \psi^5 = 0, \tag{2.162}$$

Finally the momentum constraint (2.155) becomes

$$\hat{D}_j \tilde{A}^{ij} + 6\hat{D}_j (\ln \psi) \tilde{A}^{ij} = 8\pi\psi^4 J^i, \tag{2.163}$$

where once again $K = 0$ has been used.

Eq. (2.159) merits further attention. Since the conformal 3-metric is constant from slice to slice we have $\partial_t \hat{\gamma}_{ij} = 0$. Additionally, Eq. (2.37) tells us that we can write

$$\mathcal{L}_\beta \hat{\gamma}_{ij} = \hat{D}_k \hat{\gamma}_{ij} + \hat{\gamma}_{kj} \hat{D}_i \beta^k + \hat{\gamma}_{ik} \hat{D}_j \beta^k = \hat{\gamma}_{kj} \hat{D}_i \beta^k + \hat{\gamma}_{ik} \hat{D}_j \beta^k, \tag{2.164}$$

where we have used $\hat{D}_k \hat{\gamma}_{ij} = 0$ in the last step. Using this last equation along with $\partial_t \hat{\gamma}_{ij} = 0$ in (2.159) we find

$$\tilde{A}_{ij} = \frac{1}{2\alpha} \left(\hat{\gamma}_{kj} \hat{D}_i \beta^k + \hat{\gamma}_{ik} \hat{D}_j \beta^k - \frac{2}{3} \hat{\gamma}_{ij} \hat{D}_k \beta^k \right). \tag{2.165}$$

We can also raise both indices of (2.165)—using the inverse flat metric, $\hat{\gamma}^{ij}$ —to get

$$\tilde{A}^{ij} = \frac{1}{2\alpha} \left(\hat{\gamma}^{ik} \hat{D}_k \beta^j + \hat{\gamma}^{jk} \hat{D}_k \beta^i - \frac{2}{3} \hat{\gamma}^{ij} \hat{D}_k \beta^k \right). \tag{2.166}$$

We thus observe that having made the requirement of conformal flatness, we can view the components of \tilde{A}^{ij} as being *derived* from the shift vector components, β^k . Now our general inventory of

geometric variables within the context of the conformal 3 + 1 approach included the 4 kinematical quantities α and β^k , and the 12 dynamical fields ψ , $\tilde{\gamma}_{ij}$, K and \tilde{A}_{ij} . We now see that of these, once conformal flatness has been imposed and we have chosen $K = 0$, only α , β^k and ψ remain as independent variables. We have already noted that (2.160) provides an elliptic equation that fixes α on each slice, and we will display the final form of that equation momentarily. We now proceed to show that the constraints provide similar equations for ψ and β^k .

To do so, we first consider the divergence of \tilde{A}^{ij} . Using (2.166) we calculate

$$\begin{aligned}\hat{D}_j \tilde{A}^{ij} &= \frac{1}{2\alpha} \hat{D}_j \left(\hat{\gamma}^{ik} \hat{D}_k \beta^j + \hat{\gamma}^{jk} \hat{D}_k \beta^i - \frac{2}{3} \hat{\gamma}^{ij} \hat{D}_k \beta^k \right) - \frac{1}{\alpha} \hat{D}_j \alpha \tilde{A}^{ij} \\ &= \frac{1}{2\alpha} \left(\hat{\gamma}^{jk} \hat{D}_k \hat{D}_j \beta^i + \frac{1}{3} \hat{\gamma}^{ik} \hat{D}_k \hat{D}_j \beta^j - 2 \tilde{A}^{ij} \hat{D}_j \alpha \right).\end{aligned}\quad (2.167)$$

Using (2.167) and (2.166) in the momentum constraint (2.163) and rearranging, we find

$$\begin{aligned}\hat{\gamma}^{jk} \hat{D}_k \hat{D}_j \beta^i &= -\frac{1}{3} \hat{\gamma}^{ik} \hat{D}_k \hat{D}_j \beta^j + 2 \tilde{A}^{ij} \left[\hat{D}_j \alpha - 6\alpha \hat{D}_j (\ln \psi) \right] + 16\pi\alpha\psi^4 J^i \\ &= -\frac{1}{3} \hat{\gamma}^{ik} \hat{D}_k \hat{D}_j \beta^j + \left[\hat{\gamma}^{ik} \hat{D}_k \beta^j + \hat{\gamma}^{jk} \hat{D}_k \beta^i - \frac{2}{3} \hat{\gamma}^{ij} \hat{D}_k \beta^k \right] \hat{D}_j \left[\ln \left(\frac{\alpha}{\psi^6} \right) \right] \\ &+ 16\pi\alpha\psi^4 J^i,\end{aligned}\quad (2.168)$$

which are 3 elliptic equations for the 3 shift vector components, β^k . Next, we use (2.165) and (2.166) to rewrite the product $\tilde{A}_{ij} \tilde{A}^{ij}$ in terms of covariant derivatives of the shift vector:

$$\tilde{A}_{ij} \tilde{A}^{ij} = \frac{1}{2\alpha} \left(\hat{\gamma}_{kj} \hat{\gamma}^{il} \hat{D}_i \beta^k \hat{D}_l \beta^j + \hat{D}_k \beta^i \hat{D}_i \beta^k - \frac{2}{3} \hat{D}_i \beta^i \hat{D}_j \beta^j \right).$$

Substituting the right hand side of this equation into the Hamiltonian constraint (2.162) and rearranging, we have

$$\hat{\gamma}^{kl} \hat{D}_k \hat{D}_l \psi = - \left[\frac{1}{16\alpha} \left(\hat{\gamma}_{kj} \hat{\gamma}^{il} \hat{D}_i \beta^k \hat{D}_l \beta^j + \hat{D}_k \beta^i \hat{D}_i \beta^k - \frac{2}{3} \hat{D}_i \beta^i \hat{D}_j \beta^j \right) - 2\pi\rho \right] \psi^5. \quad (2.169)$$

which is an elliptic equation for the conformal factor, ψ .

Finally, (2.160), which we recall arises from the demand that $K \equiv 0$ can be written as

$$\begin{aligned}
\hat{\gamma}^{kl} \hat{D}_k \hat{D}_l \alpha &= -2\hat{\gamma}^{kl} \hat{D}_k (\ln \psi) \hat{D}_l \alpha + \psi^4 \alpha \left[\tilde{A}_{kl} \tilde{A}^{kl} + 4\pi (S + \rho) \right] \\
&= -2\hat{\gamma}^{kl} \hat{D}_k (\ln \psi) \hat{D}_l \alpha + \frac{1}{2} \psi^4 \left[\hat{\gamma}_{kj} \hat{\gamma}^{il} \hat{D}_i \beta^k \hat{D}_l \beta^j + \hat{D}_k \beta^i \hat{D}_i \beta^k - \frac{2}{3} \hat{D}_i \beta^i \hat{D}_j \beta^j \right] \\
&+ 4\pi \psi^4 \alpha (S + \rho) , \tag{2.170}
\end{aligned}$$

and is, as stated previously, an elliptic equation for the lapse function, α .

We emphasize that a sequence of solutions of equations (2.168)–(2.170)—computed on each hypersurface of the spacetime—completely determines the 4-dimensional metric, $g_{\mu\nu}$. Specifically, we have

$$\begin{aligned}
ds^2 &= g_{\mu\nu} dx^\mu dx^\nu \\
&= -\alpha^2 dt^2 + \gamma_{ij} (dx^i + \beta^i dt) (dx^j + \beta^j dt) \\
&= -\alpha^2 dt^2 + \psi^4 \hat{\gamma}_{ij} (dx^i + \beta^i dt) (dx^j + \beta^j dt) . \tag{2.171}
\end{aligned}$$

Table 2.1 summarizes the steps taken in this chapter to produce the CFA equations of motion. It includes enumerations of the full Einstein-Klein-Gordon equations within both the standard and conformal 3 + 1 approaches, as well as the subset of the conformal equations that were used to derive the CFA system.

In concluding this section, we note that the system (2.168)–(2.170) is still written in a generally 3-covariant form—that is, the equations are valid for any set of spatial coordinates that we might choose to adopt in conjunction with a flat 3-metric. In order to recast the system as a specific set of PDEs that can be solved numerically, we must first fix the spatial coordinates, and this is the topic of the next section.

EKG eqns. in 3 + 1 form.	Conformally decomposed EKG eqns.	CFA equations of motion.
$\mathcal{R} + K^2 - K_{ij}K^{ij} = 16\pi\rho$ (2.49) $D_j K^{ij} - D^i K = 8\pi J^i$ (2.50)	$\tilde{\gamma}^{kl}\tilde{D}_k\tilde{D}_l\psi - \frac{1}{8}\psi\tilde{\mathcal{R}} + \left(\frac{1}{8}\tilde{A}_{kl}\tilde{A}^{kl} - \frac{1}{12}K^2\right)\psi^5 = -2\pi\rho\psi^5$ (2.154) $\tilde{D}_j\tilde{A}^{ij} + 6\tilde{D}_j(\ln\psi)\tilde{A}^{ij} - \frac{2}{3}\tilde{D}^i K = 8\pi\psi^4 J^i$ (2.155)	$\hat{\gamma}^{kl}\hat{D}_k\hat{D}_l\psi = -\left(\frac{1}{8}\tilde{A}_{kl}\tilde{A}^{kl} + 2\pi\rho\right)\psi^5$ (2.162) $\hat{\gamma}^{jk}\hat{D}_k\hat{D}_j\beta^i = -\frac{1}{3}\hat{\gamma}^{ik}\hat{D}_k\hat{D}_j\beta^j + 2\tilde{A}^{ij}\left[\hat{D}_j\alpha - 6\alpha\hat{D}_j(\ln\psi)\right] + 16\pi\alpha\psi^4 J^i$ (2.168)
$\mathcal{L}_{(t-\beta)}\gamma_{ij} = -2\alpha\gamma_{ik}K^k{}_j$ (2.44) $\mathcal{L}_{(t-\beta)}K^i{}_j = -D^i D_j\alpha + \alpha\left[\mathcal{R}^i{}_j + KK^i{}_j + 8\pi\left(\frac{1}{2}\delta^i{}_j(S-\rho) - S^i{}_j\right)\right]$ (2.45)	$(\partial_t - \mathcal{L}_\beta)\psi = \frac{1}{6}\psi\left(\tilde{D}_i\beta^i - \alpha K\right)$ (2.150) $(\partial_t - \mathcal{L}_\beta)\tilde{\gamma}_{ij} = -2\alpha\tilde{A}_{ij} - \frac{2}{3}\tilde{\gamma}_{ij}\tilde{D}_k\beta^k$ (2.151) $(\partial_t - \mathcal{L}_\beta)K = -\psi^{-4}\tilde{\gamma}^{kl}\left[\tilde{D}_k\tilde{D}_l\alpha + 2\tilde{D}_k(\ln\psi)\tilde{D}_l\alpha\right] + \alpha\left[\tilde{A}_{kl}\tilde{A}^{kl} + \frac{1}{3}K^2 + 4\pi(S+\rho)\right]$ (2.152) $(\partial_t - \mathcal{L}_\beta)\tilde{A}_{ij} = \psi^{-4}\left\{-\tilde{D}_i\tilde{D}_j\alpha + 4\tilde{D}_{(i}(\ln\psi)\tilde{D}_{j)}\alpha + \frac{1}{3}\tilde{\gamma}_{ij}\tilde{\gamma}^{kl}\left(\tilde{D}_k\tilde{D}_l\alpha - 4\tilde{D}_k(\ln\psi)\tilde{D}_l\alpha\right) + \alpha\left[\tilde{\mathcal{R}}_{ij} - \frac{1}{3}\tilde{\gamma}_{ij}\tilde{\mathcal{R}} - 2\tilde{D}_i\tilde{D}_j(\ln\psi) + 4\tilde{D}_i(\ln\psi)\tilde{D}_j(\ln\psi) + \frac{2}{3}\tilde{\gamma}_{ij}\tilde{\gamma}^{kl}\left(\tilde{D}_k\tilde{D}_l(\ln\psi) - 2\tilde{D}_k(\ln\psi)\tilde{D}_l(\ln\psi)\right)\right]\right\} + \alpha\left[K\tilde{A}_{ij} - 2\tilde{\gamma}^{kl}\tilde{A}_{ik}\tilde{A}_{lj} - 8\pi\left(\psi^{-4}S_{ij} - \frac{1}{3}\tilde{\gamma}_{ij}S\right) - \frac{2}{3}\tilde{A}_{ij}\tilde{D}_l\beta^l\right]$ (2.153)	$\mathcal{L}_{(t-\beta)}\psi = \frac{1}{6}\psi\hat{D}_i\beta^i$ (2.158) $\tilde{A}_{ij} = \frac{1}{2\alpha}\left(\hat{\gamma}_{kj}\hat{D}_i\beta^k + \hat{\gamma}_{ik}\hat{D}_j\beta^k - \frac{2}{3}\hat{\gamma}_{ij}\hat{D}_k\beta^k\right)$ (2.165) $\hat{\gamma}^{kl}\hat{D}_k\hat{D}_l\alpha = -2\hat{\gamma}^{kl}\hat{D}_k(\ln\psi)\hat{D}_l\alpha + \psi^4\alpha\left[\tilde{A}_{kl}\tilde{A}^{kl} + 4\pi(S+\rho)\right]$ (2.170) $\mathcal{L}_{(t-\beta)}\tilde{A}_{ij} = \psi^{-4}\left\{-\hat{D}_i\hat{D}_j\alpha + 4\hat{D}_{(i}(\ln\psi)\hat{D}_{j)}\alpha + \frac{1}{3}\hat{\gamma}_{ij}\hat{\gamma}^{kl}\left(\hat{D}_k\hat{D}_l\alpha - 4\hat{D}_k(\ln\psi)\hat{D}_l\alpha\right) + \alpha\left[-2\hat{D}_i\hat{D}_j(\ln\psi) + 4\hat{D}_i(\ln\psi)\hat{D}_j(\ln\psi) + \frac{2}{3}\hat{\gamma}_{ij}\hat{\gamma}^{kl}\left(\hat{D}_k\hat{D}_l(\ln\psi) - 2\hat{D}_k(\ln\psi)\hat{D}_l(\ln\psi)\right)\right]\right\} + \alpha\left[-2\hat{\gamma}^{kl}\tilde{A}_{ik}\tilde{A}_{lj} - 8\pi\left(\psi^{-4}S_{ij} - \frac{1}{3}\hat{\gamma}_{ij}S\right) - \frac{2}{3}\tilde{A}_{ij}\hat{D}_l\beta^l\right]$ (2.161)
$\partial_t\phi_A = \frac{\alpha}{\sqrt{\gamma}}\Pi_A + \beta^i\partial_i\phi_A$ (2.67) $\partial_t\Pi_A = \partial_i(\beta^i\Pi_A) + \partial_i(\alpha\sqrt{\gamma}\gamma^{ij}\partial_j\phi_A) - \alpha\sqrt{\gamma}\frac{dU(\phi_0^2)}{d\phi_0^2}\phi_A$ (2.68)	$\partial_t\phi_A = \frac{\alpha}{\psi^6\sqrt{\tilde{\gamma}}}\Pi_A + \beta^i\partial_i\phi_A$ $\partial_t\Pi_A = \partial_i(\beta^i\Pi_A) + \partial_i(\alpha\psi^2\sqrt{\tilde{\gamma}}\tilde{\gamma}^{ij}\partial_j\phi_A) - \alpha\psi^6\sqrt{\tilde{\gamma}}\frac{dU(\phi_0^2)}{d\phi_0^2}\phi_A$	$\partial_t\phi_A = \frac{\alpha}{\psi^6\sqrt{\hat{\gamma}}}\Pi_A + \beta^i\partial_i\phi_A$ $\partial_t\Pi_A = \partial_i(\beta^i\Pi_A) + \partial_i(\alpha\psi^2\sqrt{\hat{\gamma}}\hat{\gamma}^{ij}\partial_j\phi_A) - \alpha\psi^6\sqrt{\hat{\gamma}}\frac{dU(\phi_0^2)}{d\phi_0^2}\phi_A$

Table 2.1: Equations of motion. This table summarizes the steps taken in the derivation of the CFA equations of motion. The fully general relativistic 3 + 1 Einstein-Klein-Gordon (EKG) equations, and their conformal decompositions, are given in the first two columns. The first row lists the constraint equations, while the evolution equations for the geometric and matter variables are listed in the second and third rows, respectively. The maximal slicing condition and assumption of conformal flatness yield the CFA equations; these are given in the third column. Note that within the CFA the components of the traceless conformal extrinsic curvature components, \tilde{A}_{ij} , are *defined* in terms of the lapse function and shift vector, and that the evolution equations for the conformal factor, ψ , and the \tilde{A}_{ij} , shown in red, are not used.

2.4.4 Specialization to Cartesian Coordinates

The choice of spatial coordinates is frequently a crucial issue in numerical relativity calculations. In cases where the spacetime is constrained to have an exact symmetry, or where the physical scenario has an approximate symmetry, coordinates adapted to the symmetry are often used. Adopting such coordinates—which will generally be curvilinear—can be advantageous not only for the significant simplification in the equations of motion that may result, but also for providing a “natural” representation of the solution, which in turn can minimize computational cost. For example, consider the problem of simulating a single star in 3 spatial dimensions plus time, but where the star remains approximately spherically symmetric. Then, irrespective of the method we choose to discretize the continuum equations (i.e. finite difference, finite element, spectral etc.), we can expect that for specified accuracy it will be cheaper to compute a solution using spherical polar coordinates than Cartesian coordinates.

However, especially when dealing with tensor equations, the use of curvilinear coordinates for numerical calculations can also be problematic. Such coordinate systems generically have coordinate singularities, such as $r = 0$ and the “ z -axis” in the spherical polar case, where special attention and care is needed when designing and implementing the numerical approach to ensure that the computed solution remains smooth. In any case, the ultimate goal of the work described here is to simulate boson star binaries, where the expected solutions will not have any specific simple symmetry that would motivate us to adopt some special curvilinear coordinate system. We instead use a Cartesian coordinate system, (x, y, z) , which has the advantages of 1) covering the spatial hypersurfaces in a smooth fashion (i.e. without any coordinate singularities or other pathologies) provided that there are no physical singularities on the slices, and 2) inducing particularly simple forms for the differential operators appearing in the governing PDEs.

In addition to having Cartesian “topology”, and in contrast to some previous related work (most notably that of Wilson, Mathews, Marronetti [16, 17, 18, 19, 20, 21, 23, 25, 26]) we require that our coordinate system be “asymptotically inertial”, i.e. we demand that at large distances from the matter sources the metric components approach those of flat spacetime in an inertial frame, $g_{\mu\nu} \rightarrow \eta_{\mu\nu} = \text{diag}(-1, 1, 1, 1)$. In the work of Wilson and collaborators the coordinate system was typically chosen to be in corotation with the binary system being studied. Finally, we emphasize that the topology of the hypersurfaces is taken to be \mathbb{R}^3 , so that the slices are infinite in extent in all three directions. Naturally, this has significant implications for the numerical treatment of

boundary conditions in our model as will be discussed in following sections.

Thus, from this point on, the set of field variables defining our model system, namely α , ψ , β^i , $i = 1, 2, 3$, and ϕ_A , Π_A , $A = 1, 2$, are all to be understood to be functions of (t, x, y, z) .

With the choice of Cartesian spatial coordinates, the conformally flat 3-dimensional line element is simply

$${}^{(3)}ds^2 = \psi^4(dx^2 + dy^2 + dz^2). \quad (2.172)$$

We can now display the equations of motion for our model in essentially the form used for our numerical computations. We start with the evolution equations for the complex scalar field, (2.67) and (2.68), which become

$$\partial_t \phi_A = \frac{\alpha}{\psi^6} \Pi_A + \beta^i \partial_i \phi_A, \quad (2.173)$$

$$\begin{aligned} \partial_t \Pi_A &= \partial_x (\beta^x \Pi_A + \alpha \psi^2 \partial_x \phi_A) + \partial_y (\beta^y \Pi_A + \alpha \psi^2 \partial_y \phi_A) \\ &+ \partial_z (\beta^z \Pi_A + \alpha \psi^2 \partial_z \phi_A) - \alpha \psi^6 \frac{dU(\phi_0^2)}{d\phi_0^2} \phi_A. \end{aligned} \quad (2.174)$$

Continuing, the energy-momentum quantities defined by (2.71)-(2.76) are

$$\rho = \frac{1}{2} \sum_{A=1}^2 \left[\frac{\Pi_A^2}{\psi^{12}} + \frac{1}{\psi^4} [(\partial_x \phi_A)^2 + (\partial_y \phi_A)^2 + (\partial_z \phi_A)^2] \right] + \frac{1}{2} U(\phi_0^2), \quad (2.175)$$

$$J_i = \sum_{A=1}^2 \left[\frac{\Pi_A}{\psi^6} \partial_i \phi_A \right], \quad (2.176)$$

$$J^i = \sum_{A=1}^2 \left[-\frac{\Pi_A}{\psi^{10}} \partial_i \phi_A \right], \quad (2.177)$$

$$\begin{aligned} S_{ij} &= \frac{1}{2} \sum_{A=1}^2 \left\{ 2\partial_i \phi_A \partial_j \phi_A + \psi^4 \delta_{ij} \left[\frac{\Pi_A^2}{\psi^{12}} \right. \right. \\ &\quad \left. \left. - \frac{1}{\psi^4} [(\partial_x \phi_A)^2 + (\partial_y \phi_A)^2 + (\partial_z \phi_A)^2] \right] \right\} - \frac{1}{2} \psi^4 \delta_{ij} U(\phi_0^2), \end{aligned} \quad (2.178)$$

$$S^i = \frac{1}{2} \sum_{A=1}^2 \left[3 \frac{\Pi_A^2}{\psi^{12}} - \frac{1}{\psi^4} [(\partial_x \phi_A)^2 + (\partial_y \phi_A)^2 + (\partial_z \phi_A)^2] \right] - \frac{3}{2} U(\phi_0^2), \quad (2.179)$$

$$\rho + S = \sum_{A=1}^2 \left[2 \frac{\Pi_A^2}{\psi^{12}} \right] - U(\phi_0^2). \quad (2.180)$$

Next, we have the equations that constrain the geometric variables. The maximal slicing con-

dition (2.170) for the lapse function becomes

$$\frac{\partial^2 \alpha}{\partial x^2} + \frac{\partial^2 \alpha}{\partial y^2} + \frac{\partial^2 \alpha}{\partial z^2} = -\frac{2}{\psi} \left[\frac{\partial \psi}{\partial x} \frac{\partial \alpha}{\partial x} + \frac{\partial \psi}{\partial y} \frac{\partial \alpha}{\partial y} + \frac{\partial \psi}{\partial z} \frac{\partial \alpha}{\partial z} \right] + \alpha \psi^4 \left(\tilde{A}_{ij} \tilde{A}^{ij} + 4\pi(\rho + S) \right), \quad (2.181)$$

while the Hamiltonian constraint (2.169) for ψ is

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \frac{\partial^2 \psi}{\partial z^2} = -\frac{\psi^5}{8} \left(\tilde{A}_{ij} \tilde{A}^{ij} + 16\pi\rho \right). \quad (2.182)$$

For both (2.181) and (2.182) the term $\tilde{A}_{ij} \tilde{A}^{ij}$ is given by

$$\begin{aligned} \tilde{A}_{ij} \tilde{A}^{ij} = & \frac{1}{2\alpha^2} \left[\left(\frac{\partial \beta^x}{\partial x} \right)^2 + \left(\frac{\partial \beta^x}{\partial y} \right)^2 + \left(\frac{\partial \beta^x}{\partial z} \right)^2 + \left(\frac{\partial \beta^y}{\partial x} \right)^2 + \left(\frac{\partial \beta^y}{\partial y} \right)^2 + \left(\frac{\partial \beta^y}{\partial z} \right)^2 \right. \\ & + \left(\frac{\partial \beta^z}{\partial x} \right)^2 + \left(\frac{\partial \beta^z}{\partial y} \right)^2 + \left(\frac{\partial \beta^z}{\partial z} \right)^2 + \left(\frac{\partial \beta^x}{\partial x} \frac{\partial \beta^x}{\partial x} + \frac{\partial \beta^y}{\partial x} \frac{\partial \beta^x}{\partial y} + \frac{\partial \beta^z}{\partial x} \frac{\partial \beta^x}{\partial z} \right) \\ & + \left(\frac{\partial \beta^x}{\partial y} \frac{\partial \beta^y}{\partial x} + \frac{\partial \beta^y}{\partial y} \frac{\partial \beta^y}{\partial y} + \frac{\partial \beta^z}{\partial y} \frac{\partial \beta^y}{\partial z} \right) + \left(\frac{\partial \beta^x}{\partial z} \frac{\partial \beta^z}{\partial x} + \frac{\partial \beta^y}{\partial z} \frac{\partial \beta^z}{\partial y} + \frac{\partial \beta^z}{\partial z} \frac{\partial \beta^z}{\partial z} \right) \\ & \left. - \frac{2}{3} \left(\frac{\partial \beta^x}{\partial x} + \frac{\partial \beta^y}{\partial y} + \frac{\partial \beta^z}{\partial z} \right)^2 \right]. \end{aligned}$$

Finally, from the momentum constraints (2.168), we have the 3 equations that fix the components of the shift vector:

$$\begin{aligned} \frac{\partial^2 \beta^x}{\partial x^2} + \frac{\partial^2 \beta^x}{\partial y^2} + \frac{\partial^2 \beta^x}{\partial z^2} = & -\frac{1}{3} \frac{\partial}{\partial x} \left(\frac{\partial \beta^x}{\partial x} + \frac{\partial \beta^y}{\partial y} + \frac{\partial \beta^z}{\partial z} \right) + \alpha \psi^4 16\pi J^x \\ & - \frac{\partial}{\partial x} \left[\ln \left(\frac{\psi^6}{\alpha} \right) \right] \left[\frac{4}{3} \frac{\partial \beta^x}{\partial x} - \frac{2}{3} \left(\frac{\partial \beta^y}{\partial y} + \frac{\partial \beta^z}{\partial z} \right) \right] \\ & - \frac{\partial}{\partial y} \left[\ln \left(\frac{\psi^6}{\alpha} \right) \right] \left[\frac{\partial \beta^x}{\partial y} + \frac{\partial \beta^y}{\partial x} \right] - \frac{\partial}{\partial z} \left[\ln \left(\frac{\psi^6}{\alpha} \right) \right] \left[\frac{\partial \beta^x}{\partial z} + \frac{\partial \beta^z}{\partial x} \right], \end{aligned} \quad (2.183)$$

$$\begin{aligned} \frac{\partial^2 \beta^y}{\partial x^2} + \frac{\partial^2 \beta^y}{\partial y^2} + \frac{\partial^2 \beta^y}{\partial z^2} = & -\frac{1}{3} \frac{\partial}{\partial y} \left(\frac{\partial \beta^x}{\partial x} + \frac{\partial \beta^y}{\partial y} + \frac{\partial \beta^z}{\partial z} \right) + \alpha \psi^4 16\pi J^y \\ & - \frac{\partial}{\partial y} \left[\ln \left(\frac{\psi^6}{\alpha} \right) \right] \left[\frac{4}{3} \frac{\partial \beta^y}{\partial y} - \frac{2}{3} \left(\frac{\partial \beta^x}{\partial x} + \frac{\partial \beta^z}{\partial z} \right) \right] \\ & - \frac{\partial}{\partial x} \left[\ln \left(\frac{\psi^6}{\alpha} \right) \right] \left[\frac{\partial \beta^x}{\partial y} + \frac{\partial \beta^y}{\partial x} \right] - \frac{\partial}{\partial z} \left[\ln \left(\frac{\psi^6}{\alpha} \right) \right] \left[\frac{\partial \beta^y}{\partial z} + \frac{\partial \beta^z}{\partial y} \right], \end{aligned} \quad (2.184)$$

$$\begin{aligned}
\frac{\partial^2 \beta^z}{\partial x^2} + \frac{\partial^2 \beta^z}{\partial y^2} + \frac{\partial^2 \beta^z}{\partial z^2} &= -\frac{1}{3} \frac{\partial}{\partial z} \left(\frac{\partial \beta^x}{\partial x} + \frac{\partial \beta^y}{\partial y} + \frac{\partial \beta^z}{\partial z} \right) + \alpha \psi^4 16\pi J^z \\
&\quad - \frac{\partial}{\partial z} \left[\ln \left(\frac{\psi^6}{\alpha} \right) \right] \left[\frac{4}{3} \frac{\partial \beta^z}{\partial z} - \frac{2}{3} \left(\frac{\partial \beta^x}{\partial x} + \frac{\partial \beta^y}{\partial y} \right) \right] \\
&\quad - \frac{\partial}{\partial y} \left[\ln \left(\frac{\psi^6}{\alpha} \right) \right] \left[\frac{\partial \beta^z}{\partial y} + \frac{\partial \beta^y}{\partial z} \right] - \frac{\partial}{\partial x} \left[\ln \left(\frac{\psi^6}{\alpha} \right) \right] \left[\frac{\partial \beta^x}{\partial z} + \frac{\partial \beta^z}{\partial x} \right].
\end{aligned} \tag{2.185}$$

The 4 hyperbolic scalar field evolution equations, (2.173) and (2.174), along with the 5 elliptic equations, (2.181)-(2.185), constitute the basic set of equations for our model. This set of PDEs must of course be supplemented by boundary and initial conditions in order to complete the mathematical prescription of the model, and these will be discussed in the sections that follow.

Before moving on to that discussion though, it is worth mentioning that the derivation of equations of motion such as the above set is a non-trivial and error prone process. It is therefore very useful to use symbolic manipulation software to check calculations, and we have done so. Specifically, after having been derived by hand, the equations presented in this thesis were checked using Maple [145], including a tensor manipulation package due to Choptuik [146].

2.4.5 Boundary Conditions

In this section we discuss issues related to the boundary conditions that are to be applied in conjunction with the equations of motion summarized above. Mathematically, our model is to be solved as a Cauchy problem where the $t = \text{const.}$ surfaces extend to spatial infinity.¹⁵ We will restrict attention to cases where the matter source (the complex scalar field) has compact support on the initial time-slice: the hyperbolicity of the scalar wave equation then guarantees that the matter can never reach spatial infinity, which, of course, is appropriate from a physical point of view. This restriction, combined with the demands that 1) the spacetimes we construct be asymptotically flat, 2) that our (x, y, z) coordinate system be “asymptotically inertial”, and 3) that coordinate

¹⁵We note that our use of the term “boundary conditions” here is thus a slight abuse of nomenclature in the sense that the $t = \text{const}$ surfaces are edgeless, and thus have no boundaries.

time is identical to proper time at infinity, provides the following set of boundary conditions:

$$\lim_{r \rightarrow \infty} \phi_A(t, x, y, z) = 0, \quad (2.186)$$

$$\lim_{r \rightarrow \infty} \Pi_A(t, x, y, z) = 0, \quad (2.187)$$

$$\lim_{r \rightarrow \infty} \psi(t, x, y, z) = 1, \quad (2.188)$$

$$\lim_{r \rightarrow \infty} \alpha(t, x, y, z) = 1, \quad (2.189)$$

$$\lim_{r \rightarrow \infty} \beta^k(t, x, y, z) = 0. \quad (2.190)$$

Here and below r is defined by $r \equiv \sqrt{x^2 + y^2 + z^2}$.

The computational problems that arise from the fact that our boundary conditions are naturally specified at infinity are familiar ones, not only in numerical relativity, but in many other areas of numerical analysis and computational science that involve the solution of hyperbolic PDEs on unbounded domains. A key, if rather obvious observation, is that while the spatial domain is infinite, any specific numerical computation based on a discretization technique (such as finite differencing, as used in this thesis) must be restricted to a finite number of discrete unknowns. Given this, there are essentially two basic strategies for dealing with the infinite solution domain. The first involves artificially introducing boundaries at $x = x_{\min}$, $x = x_{\max}$, etc. and then imposing approximate boundary conditions on the solution unknowns. The second involves “compactification” of the infinite domain, using a coordinate transformation which maps infinity to a finite value of the transformed coordinate. The exact boundary conditions can then be applied on the boundaries of the compactified domain. In the current work, we have experimented with two variations of the first approach, and one of the second. We proceed to discuss each of the approaches that we studied, and in the the original order that they were investigated. We highlight specific challenges that we encountered, and indicate possible future directions for improvement. We also note that parts of the discussion rely on computational concepts and techniques that are discussed in detail in Chap. 4. The reader may thus wish to postpone a detailed study of this part of the thesis until that chapter has been perused.

Spatial Compactification

Spatial compactification was the first strategy we considered for a computational treatment of the conditions (2.186)–(2.190). As already mentioned, the basic idea in this case is quite simple: a smooth coordinate transformation is applied to map each infinite coordinate range to a finite interval, which, without loss of generality we can take to be $[-1, 1]$. Since elements of the Jacobian matrix of the coordinate transformation will appear in the PDEs when written in the new coordinates, it is reasonable to require that the transformation be given in some simple closed form. With this in mind we chose the hyperbolic tangent function to define compactified coordinates (ξ, η, ζ) by

$$\xi(x) = \tanh x, \quad \eta(y) = \tanh y, \quad \zeta(z) = \tanh z. \quad (2.191)$$

Clearly, this transformation maps $-\infty < x, y, z < \infty$ to $-1 \leq \xi, \eta, \zeta \leq 1$, as desired, and the exact (Dirichlet) boundary conditions can then be set at $\xi, \eta, \zeta = \pm 1$.

Transforming the PDEs that govern our model requires nothing more than the chain rule for differentiation. For our specific choice of compactification we have

$$\frac{\partial}{\partial x} = \frac{d\xi}{dx} \frac{d}{d\xi} = (1 - \tanh^2 x) \frac{d}{d\xi} = (1 - \xi^2) \frac{d}{d\xi}, \quad (2.192)$$

and

$$\frac{d^2}{dx^2} = (1 - \xi^2) \frac{d}{d\xi} \left[(1 - \xi^2) \frac{d}{d\xi} \right] = (1 - \xi^2)^2 \frac{d^2}{d\xi^2} - 2\xi(1 - \xi^2) \frac{d}{d\xi}, \quad (2.193)$$

as well as analogous formulae for the y and z derivatives. Using these formulae, we derive the transformed equations of motion simply by rewriting all of the derivatives appearing in (2.173)–(2.174) and (2.181)–(2.185), in terms of derivatives taken with respect to the compactified variables.

We note at this point that spatial compactification of the sort we have just described generically leads to well-known computational problems for hyperbolic systems. In particular, for the case of finite differencing, and assuming that the discretization is uniform in the compactified coordinates (i.e. so that the spacing between grid points in each of the coordinate directions is constant), the physical separation between grid points becomes increasingly large as the boundaries of the compactified domain are approached. This means that waves propagating outwards invariably become very poorly resolved. In turn, this can lead to various difficulties including numerical instabilities and spurious reflections of waves back into the interior of the domain.

Nonetheless, spatial compactification of this type *has* proven successful in some previous time dependent calculations in numerical relativity. This includes a study of the Gregory-Laflamme instability of black strings [147], as well as Pretorius' ground-breaking work on binary black hole inspiral and merger [43, 148]. In both of these instances, computational difficulties of the sort mentioned above were kept under control through the addition of numerical dissipation.

Given the previous discussion, as well as the fact that compactification *is* routinely used with great success in the numerical analysis of elliptic PDEs, it is somewhat ironic that it was problems associated with the solution of the elliptic equations of our model that ultimately forced us to abandon compactification. Noting that the specific difficulties we encountered are well documented in the numerical analysis literature, we nonetheless feel it important to discuss them here in some detail. We are especially motivated by the fact that there appear to be very few instances where elliptic systems have been solved in numerical relativity using compactification in conjunction with finite difference techniques.

In order to understand the nature and source of the problem we ran into, it suffices to consider a model elliptic PDE, namely the Poisson equation:

$$\nabla^2 u(x, y, z) = \rho(x, y, z) \quad \iff \quad \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = \rho. \quad (2.194)$$

Here u is the unknown function while ρ is a specified source function, which we will require to have compact support. In analogy with the elliptic equations of our model system we will consider the solution of (2.194) on \mathbb{R}^3 and require that

$$\lim_{r \rightarrow \infty} u(x, y, z) = 0, \quad (2.195)$$

where, again, $r \equiv \sqrt{x^2 + y^2 + z^2}$. We now transform (2.194) to the compactified coordinate system defined by (2.191) using (2.193) and the corresponding formulae for the second y and z derivatives.

We get

$$(1 - \xi^2)^2 \frac{\partial^2 \bar{u}}{\partial \xi^2} + (1 - \eta^2)^2 \frac{\partial^2 \bar{u}}{\partial \eta^2} + (1 - \zeta^2)^2 \frac{\partial^2 \bar{u}}{\partial \zeta^2} - 2\xi(1 - \xi^2) \frac{\partial \bar{u}}{\partial \xi} - 2\eta(1 - \eta^2) \frac{\partial \bar{u}}{\partial \eta} - 2\zeta(1 - \zeta^2) \frac{\partial \bar{u}}{\partial \zeta} = \rho(\xi, \eta, \zeta), \quad (2.196)$$

where $\bar{u} = \bar{u}(\xi, \eta, \zeta) = u(x, y, z)$. The boundary conditions (2.195) now become

$$\bar{u}(1, \eta, \zeta) = \bar{u}(-1, \eta, \zeta) = \bar{u}(\xi, 1, \zeta) = \bar{u}(\xi, -1, \zeta) = \bar{u}(\xi, \eta, 1) = \bar{u}(\xi, \eta, -1) = 0 \quad (2.197)$$

that is, they are simply Dirichlet conditions imposed on the boundaries of the domain $-1 \leq \xi, \eta, \zeta \leq 1$, precisely as we have for the elliptic equations that govern the metric variables in our model when expressed in the (ξ, η, ζ) system. Now, clearly, the transformation to compactified coordinates increases the algebraic complexity of any elliptic PDE to which it is applied, and introduces additional terms involving first derivatives of the unknown. However, dealing with these complications within the context of a finite difference approach is not difficult in principle, and, at least naively, it seems reasonable to expect that the extra computational cost involved will be more than offset by the increase in accuracy and improved convergence properties that the use of exact Dirichlet conditions will provide.

Unfortunately, these expectations are *not* met when one factors in the combination of 1) the specific technique that we have adopted to solve the discretized elliptic equations, and 2) the need to eventually have a code that runs in parallel on multi-processor architectures. We can summarize the situation as follows. As described in detail in Chap. 4, we have chosen the multigrid method to solve our finite-differenced elliptic PDEs due to its efficiency: in particular, it is the only available technique that can solve discretized forms of general nonlinear elliptic systems using $O(N)$ calculations, where N is the total number of discrete unknowns. However, as we will now discuss, the use of compactified coordinates induces a major stumbling block to the parallelization of the multigrid algorithm.

Consider the following schematic form for a Poisson-like equation, where the coordinate system (x, y, z) may be some general curvilinear system such as (ξ, η, ζ)

$$A(x, y, z) \frac{\partial^2 u}{\partial x^2} + B(x, y, z) \frac{\partial^2 u}{\partial y^2} + C(x, y, z) \frac{\partial^2 u}{\partial z^2} = \rho(x, y, z). \quad (2.198)$$

From the coefficient functions $A(x, y, z)$, $B(x, y, z)$ and $C(x, y, z)$ we can construct the ratios $|A/B|$, $|A/C|$ and $|B/C|$. If any of these ratios exhibit large variations in magnitude on the solution domain, then the PDE is said to be *anisotropic*. It is clear, then, that the compactified Poisson equation (2.196), is highly anisotropic in this sense, since each of the ratios $|A/B|$, $|A/C|$ and $|B/C|$ ranges from 0 to ∞ for $-1 \leq \xi, \eta, \zeta \leq 1$

Now—and again as discussed in more detail in Chap. 4—to construct an efficient multigrid algorithm, we must have a way to efficiently *smooth* the errors in the discrete solution independently of the size of the mesh spacing that is used in any given calculation. For equations that are *not* anisotropic, simple relaxation methods such as point-wise Newton-Gauss-Seidel (see App. D) tend to be very effective smoothers. Importantly, point-wise techniques are readily parallelized because the operation of updating any given unknown during a relaxation sweep requires only “local information”: that is the update operation only involves values of other unknowns which are directly coupled through the finite difference equations themselves. For the type of finite differencing used in this thesis (see App. B) this typically amounts to nearest-neighbours in each of the coordinate directions.

For the case of anisotropic elliptic operators, however, point-wise relaxation will generically fail to be an effective smoother [35]. For example, consider the situation where the ratios $|A/B|$ and $|A/C|$ for the schematic equation (2.198) satisfy $|A/B| \approx |A/C| \gg 1$. Then the equation is anisotropic and is said to be *strongly coupled* in the x direction. As shown in [35], point-wise relaxation tends to smooth only along directions of strong coupling, so in this instance, the point-wise approach will not provide effective smoothing along the y and z directions. Now, there are well known strategies for recovering a good smoother when facing anisotropy. Chief among these is the use of *line* relaxation, which for the current example would involve the simultaneous update of all of the unknowns in the x direction for each discrete combination of (y, z) . Unfortunately, line relaxation destroys the locality of the smoothing process, and thus inhibits parallelization. This is the main reason that we decided not to implement compactification in the version of our code that was used to generate the results described in this thesis.

Before concluding this section, however, we point out that there is another approach to deal with anisotropy in multigrid that seems very promising for our application. This is the technique of semi-coarsening [149, 150], whereby the coarsening operation inherent to all multigrid algorithms is first performed only along directions of strong coupling. The coarsening process tends to weaken the degree of coupling in those directions, so that point-relaxation can again be used as a smoother. For compactifying transformations of the form (2.191) the effective strong-coupling directions will be location dependent, so that we would need to implement an algorithm that cycles through the three coordinates ξ , η and ζ , semi-coarsening along each direction in turn. However, all of the operations involved in this method, including the semi-coarsening itself would remain local, and

thus the method would be amenable to parallelization.

Asymptotic and Sommerfeld Boundary Conditions

The next strategy we considered for treating the boundary conditions (2.186)–(2.190) is the one that is probably the most widely used in current 3D numerical relativity codes. Here, the underlying idea is to work in (non-compactified) (x, y, z) coordinates and replace the spatially infinite domain with a finite region defined by

$$x_{\min} \leq x \leq x_{\max}, \quad y_{\min} \leq y \leq y_{\max}, \quad z_{\min} \leq z \leq z_{\max}, \quad (2.199)$$

where x_{\min} , x_{\max} , y_{\min} , y_{\max} , z_{\min} , z_{\max} , become adjustable parameters of the computation. We then use the known and/or expected behaviour of the solution unknowns as $r \rightarrow \infty$ [120]¹⁶ to derive approximate conditions which can be imposed on the boundaries of the domain.

For example, considering the conformal factor, ψ , asymptotic flatness of the spacetime requires that

$$\lim_{r \rightarrow \infty} \psi = 1 + \frac{k}{r} + O(r^{-2}), \quad (2.200)$$

where k is a constant. Using a trick that is well known to numerical relativists [151], we can convert (2.200) to a boundary condition of mixed type as follows. We first differentiate (2.200) with respect to r to get

$$\lim_{r \rightarrow \infty} \frac{\partial \psi}{\partial r} = -\frac{k}{r^2} + O(r^{-3}). \quad (2.201)$$

This implies that

$$k = \lim_{r \rightarrow \infty} \left[-r^2 \frac{\partial \psi}{\partial r} \right] + O(r^{-1}). \quad (2.202)$$

Using this result to eliminate k in (2.200) we have

$$\lim_{r \rightarrow \infty} \psi = 1 - r \frac{\partial \psi}{\partial r} + O(r^{-2}), \quad (2.203)$$

or

$$\lim_{r \rightarrow \infty} \left[\frac{\partial \psi}{\partial r} + \frac{\psi - 1}{r} \right] = O(r^{-3}). \quad (2.204)$$

Neglecting the $O(r^{-3})$ term in the above, the condition that we impose on ψ on the boundaries of

¹⁶Again recall that $r \equiv \sqrt{x^2 + y^2 + z^2}$.

the finite domain is then

$$\frac{\partial\psi}{\partial r} + \frac{\psi - 1}{r} = 0. \quad (2.205)$$

In terms of Cartesian coordinates, (x, y, z) , we have

$$\frac{\partial\psi}{\partial r} = \frac{\partial\psi}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial\psi}{\partial y} \frac{\partial y}{\partial r} + \frac{\partial\psi}{\partial z} \frac{\partial z}{\partial r} = \frac{x}{r} \frac{\partial\psi}{\partial x} + \frac{y}{r} \frac{\partial\psi}{\partial y} + \frac{z}{r} \frac{\partial\psi}{\partial z},$$

where the relations $\partial x/\partial r = x/r$ etc. follow from the standard transformation from rectangular to spherical polar coordinates. Using the above result in (2.205) we have ¹⁷

$$x \frac{\partial\psi}{\partial x} + y \frac{\partial\psi}{\partial y} + z \frac{\partial\psi}{\partial z} + \psi - 1 = 0. \quad (2.206)$$

Similar boundary conditions of mixed type can be derived for the lapse function, α , and the shift vector components, β^k , from the asymptotic behaviours:

$$\lim_{r \rightarrow \infty} \alpha = 1 + O(r^{-1}), \quad (2.207)$$

$$\lim_{r \rightarrow \infty} \beta^k = O(r^{-1}). \quad (2.208)$$

For the case of the scalar field variables, ϕ_A and Π_A , we also have fall-off conditions, but additionally must take into account the radiative nature of the fields. The situation is further complicated by the fact that since the field is massive, it has a non-trivial dispersion relationship. The simplest—and fairly crude—approach is to treat the scalar field as if it were massless. Letting $v(t, r)$ denote any of the ϕ_A, Π_A in the asymptotic region $r \rightarrow \infty$, we then expect

$$\lim_{r \rightarrow \infty} v(t, r) = \frac{h(t - r)}{r} \quad (2.209)$$

where h is a function that describes a purely *outgoing* disturbance propagating at the speed of light (i.e. with speed $c = 1$ in our units). If we impose (2.209) as a boundary condition, then we are requiring that there be no *incoming* radiation (from infinity) at any time during the evolution: such a demand is often called a *Sommerfeld* boundary condition.

The Sommerfeld condition in the form (2.209) cannot be implemented directly in a numerical

¹⁷We note that a mixed boundary condition such as (2.206), which involves the value of the function *and* its normal derivative, is sometimes called a *Robin* condition.

calculation since h is not a known function. However, observing that (2.209) implies

$$\lim_{r \rightarrow \infty} (rv) = h(t - r), \quad (2.210)$$

we immediately have

$$\lim_{r \rightarrow \infty} \left[\frac{\partial}{\partial t} (rv) + \frac{\partial}{\partial r} (rv) \right] = 0, \quad (2.211)$$

or

$$\lim_{r \rightarrow \infty} \left[r \frac{\partial v}{\partial t} + r \frac{\partial v}{\partial r} + v \right] = 0. \quad (2.212)$$

In Cartesian coordinates, and dropping the $\lim_{r \rightarrow \infty}$ this becomes

$$\sqrt{x^2 + y^2 + z^2} \frac{\partial v}{\partial t} + x \frac{\partial v}{\partial x} + y \frac{\partial v}{\partial y} + z \frac{\partial v}{\partial z} = 0. \quad (2.213)$$

The remainder of this section concerns technical details and issues related to our numerical implementation of boundary conditions exemplified by (2.206) for the metric variables, and (2.213) for the scalar field quantities. As such, we again note that the reader may first wish to consult Chap. (4) as well as App. (B) for background information on the specific finite differencing techniques that we used to discretize our model.

The key piece of (2.213) on which we need to focus is

$$x \frac{\partial v}{\partial x} + y \frac{\partial v}{\partial y} + z \frac{\partial v}{\partial z} \quad (2.214)$$

Observe that an expression of identical form appears in the boundary condition (2.206) for the conformal factor, ψ , and also appears in the corresponding equations for α and β^k that we have not displayed here. Denoting the coordinates of the finite difference grid by

$$(x_i, y_j, z_k), \quad i = 1 \dots n_x, \quad j = 1 \dots n_y, \quad k = 1 \dots n_z \quad (2.215)$$

(see Sec. 4.1), we consider the boundary $x = x_{\min}$ which is $x = x_1$ on the grid. Noting that our grid uses the same (constant) mesh spacing, h , in each of the coordinate directions, our finite difference

version of (2.214) is

$$\begin{aligned}
x_{3/2} \frac{v_{2,j,k} - v_{1,j,k}}{h} &+ \frac{1}{2} y_j \left(\frac{v_{1,j+1,k} - v_{1,j-1,k}}{2h} + \frac{v_{2,j+1,k} - v_{2,j-1,k}}{2h} \right) \\
&+ \frac{1}{2} z_k \left(\frac{v_{1,j,k+1} - v_{1,j,k-1}}{2h} + \frac{v_{2,j,k+1} - v_{2,j,k-1}}{2h} \right) \\
j = 2 \dots n_y - 1, \quad k = 2 \dots n_z - 1 & \tag{2.216}
\end{aligned}$$

Note that this formula is centred at the “fictitious” grid point $(x_{3/2}, y_j, z_k) \equiv ((x_1 + x_2)/2, y_j, z_k)$, and that due to this centring the approximation of the x derivative is $O(h^2)$ while only involving grid function values at x_1 and x_2 . Similar formulae are readily derived for the other boundaries $x = x_{\max}$, $y = y_{\min}$, $y = y_{\max}$, $z = z_{\min}$ and $z = z_{\max}$. Additionally, we stress that expressions such as (2.216) cannot be used along the edges or at the corners of the grid. There we use modified versions of (2.216) that maintain the basic strategy of centring the formula at a fictitious grid point (defined as the centre of 4 points along the edges, and 8 points at the corners). We note that although we expect that this differencing technique should work adequately for the scalar field variables (maintaining stable evolution in particular) the current version of our code has some convergence problems when stringent tolerances are set for the overall time-step iteration (see Sec. 5.1). We believe that this is due to one or more code bugs, but have not invested much time on the matter given the issue with the elliptic equations that we now discuss.

Indeed, use of the same differencing scheme to discretize the boundary conditions—such as (2.206)—imposed on the elliptic PDEs led to severe numerical difficulties that we have yet to resolve. Once more the issue is related to our use of multigrid to solve the elliptic equations. Even though our basic relaxation method (used in multigrid as a smoother, as discussed briefly in the previous section) converged to a smooth solution when applied to the full set of difference equations for the elliptic unknowns, our multigrid algorithm would *not* converge. We strongly suspect that the multigrid failure can be traced to our use of the particular differencing strategy for the boundary conditions that we have outlined above. Heuristically, the use of centring at the fictitious points means that the boundary conditions are actually being imposed in the interior of the domain, rather than on the boundaries *per se*. During the execution of a multigrid cycle, as coarser and coarser grids come into play, the locations where the boundary conditions are set thus penetrate further and further into the solution domain, and this seems to destroy the convergence of the method.

One possible remedy for this problem would be to augment the grid with so-called ghost points,

and then to impose discretized versions of the interior PDEs as well as the boundary conditions at the boundary points. Typically, when this approach is adopted one then eliminates the ghost values that are referenced by the interior equations using the discrete boundary conditions. However, although this strategy works well in 1D and 2D, it is not so straightforward to implement in 3D, especially when the PDEs involve mixed derivatives as is the case for our system. Additional complications arise due to the need to maintain smoothness near the boundaries during the multigrid process, as well as in formulating appropriate transfers within the grid hierarchy that are compatible with the boundary conditions.

In brief, although we devoted considerable effort to the task, we were not able to construct a convergent multigrid solver for our elliptic equations in the case that discrete versions of asymptotic boundary conditions such as (2.206) were used. However, we remain hopeful that a resolution to this problem *does* exist, and plan to continue to work towards it.

Dirichlet Boundary Conditions

Our third approach to the numerical treatment of the boundary conditions for our model, and the one that was incorporated in the code used to generate all of the results presented in Chap. 5, is extremely simple. We simply set the conditions (2.186)–(2.190) on the boundaries of the finite domain defined in the previous section:

$$x_{\min} \leq x \leq x_{\max}, \quad y_{\min} \leq y \leq y_{\max}, \quad z_{\min} \leq z \leq z_{\max}, \quad (2.217)$$

Specifically, we have

$$\begin{aligned} U(t, x_{\min}, y, z) = U(t, x_{\max}, y, z) &= 1 \\ U(t, x, y_{\min}, z) = U(t, x, y_{\max}, z) &= 1 \\ U(t, x, y, z_{\min}) = U(t, x, y, z_{\max}) &= 1 \end{aligned} \quad (2.218)$$

where U represents α or ψ , and

$$\begin{aligned} V(t, x_{\min}, y, z) &= V(t, x_{\max}, y, z) &= 0 \\ V(t, x, y_{\min}, z) &= V(t, x, y_{\max}, z) &= 0 \\ V(t, x, y, z_{\min}) &= V(t, x, y, z_{\max}) &= 0 \end{aligned} \tag{2.219}$$

where V is any of the β^k , ϕ_A or Π_A . With this choice, we were able to construct a convergent multigrid solver for the elliptic equations, and, as described in Chap. 5 our numerical evolutions of the coupled elliptic-hyperbolic system of PDEs governing our model were stable and convergent.

Although this choice certainly constitutes quite a crude approximation, especially for the domain sizes that we have used in our calculations to date, we note that we ultimately intend to incorporate adaptive mesh refinement techniques [152] into our code. Particularly for scenarios involving the interaction of two boson stars, we anticipate that fine grids will be needed only in the central region where the stars interact. The hope is then that a “telescoping” sequence of ever coarser grids will allow us to increase the physical size of the numerical solution domain to the point where the use of Dirichlet conditions contributes a relatively minor amount to the overall error in the solutions. Again, however, this is a matter for future investigation.

2.4.6 ADM/York Mass

In the Hamiltonian formulation of general relativity, the total spacetime energy—also called the ADM mass—is associated with the numerical value of the Hamiltonian, which at any time, t , is to be computed on a surface, ∂V , at spatial infinity. The surface encloses the entire volume V of the spacelike hypersurface Σ_t . For an asymptotically flat spacetime, the total energy on Σ_t can be expressed as a surface integral involving derivatives of the components of the 3-metric at spatial infinity. In terms of the 3 + 1 variables and Cartesian coordinates, this integral is

$$M_{\text{ADM}} \equiv E \equiv \frac{H_\infty}{16\pi} = \frac{1}{16\pi} \lim_{r \rightarrow \infty} \oint_{\partial V} \left(\frac{\partial \gamma_{ij}}{\partial x^i} - \frac{\partial \gamma_{ii}}{\partial x^j} \right) N^j dS. \tag{2.220}$$

Here $r = \sqrt{x^2 + y^2 + z^2}$, H_∞ is the numerical value of the Hamiltonian at spatial infinity, N^j is a outwards-directed unit vector normal to the surface ∂V , and dS is the area element on the surface. That is, $dS = \sqrt{q} d^2y$, where q is the determinant of the induced metric on ∂V and d^2y

are differentials associated with the coordinates on the surface. For example, if ∂V is a spherical surface at i^0 , then adopting the usual spherical polar coordinates, (r, θ, ϕ) the measure is simply $dS = \sqrt{q} d^2y = r^2 \sin \theta d\theta d\phi$ and the unit normal, N^i , has components $(N^r, N^\theta, N^\phi) = (1, 0, 0)$.

In situations such as ours, where the outer boundaries of the computational domain are *not* in the vicinity of spatial infinity, it is problematic to use (2.220) to compute a good estimate for the ADM mass. Fortunately there is an alternate expression for M_{ADM} due to Ó Murchadha and York [153] that provides the basis for a more accurate calculation of the mass in our simulations. Ó Murchadha and York investigated the relationship between the ADM masses of two spatial metrics that were conformally related, i.e. for $\gamma_{ij} = \psi^4 \tilde{\gamma}_{ij}$. Starting from (2.220), they showed that the difference in energy associated with the two metrics is given by the following volume integral over the spacelike hypersurface:

$$16\pi (M_{\text{ADM}} - \tilde{M}_{\text{ADM}}) = -8 \int_V \sqrt{\tilde{\gamma}} (\tilde{\gamma}^{ij} \partial_i \partial_j \psi) d^3x. \quad (2.221)$$

This expression takes a particularly convenient form when the conformal metric is flat, since in that case the conformal ADM mass, \tilde{M}_{ADM} , vanishes and we have ¹⁸

$$16\pi M_{\text{ADM}} = -8 \int_V \sqrt{\tilde{\gamma}} (\tilde{\gamma}^{ij} \partial_i \partial_j \psi) d^3x. \quad (2.222)$$

Written in Cartesian coordinates this becomes

$$M_{\text{ADM}} = -\frac{1}{2\pi} \int_V (\psi_{,xx} + \psi_{,yy} + \psi_{,zz}) d^3x. \quad (2.223)$$

In the simulations of our model problem we evaluate the above volume integral numerically, and use the value of M_{ADM} computed in this way as a diagnostic quantity. In this regard we mention that the question of whether M_{ADM} is exactly conserved in our model remains, to our knowledge, an open one. It is thus noteworthy that our simulations to date suggest that it *is* a conserved quantity. However in the absence of a theorem to this effect, we emphasize that our use of the conservation of M_{ADM} during an evolution as a measure of the accuracy of the calculation should be viewed with caution.

¹⁸Ó Murchadha and York thus noted that this result provided an alternate definition of M_{ADM} that was valid for for any asymptotically conformally flat hypersurface.

2.5 Overview of the Equations of Motion

In previous sections we have discussed the derivation of the PDEs that govern our model in considerable detail. Here we provide a recap of that development and redisplay the equations in a schematic form that emphasizes the principal parts of the various differential operators involved. In addition, we provide a brief commentary concerning some of the properties of the PDEs as well as few details about how they were solved numerically. We also note that we use a standard tensor calculus notation for partial differentiation below: namely that $f_{,x} = \partial f / \partial x$, $f_{,xx} = \partial^2 f / \partial x^2$ etc.

In summary, our equations of motion constitute a mixed elliptic-hyperbolic system of 9 nonlinear PDEs. There are 5 quasi-linear, elliptic equations for the geometric variables, which include the lapse function, α , the conformal factor, ψ , and the three components of the shift vector, β^i . In addition, there are 4 quasi-linear, hyperbolic equations which govern the components of the complex scalar field, ϕ_1 and ϕ_2 , and their conjugate momenta, Π_1 and Π_2 .

The elliptic equations were derived as follows. The maximal slicing condition for α resulted from taking the trace of the evolution equation for the extrinsic curvature K^i_j , and demanding that the right hand side vanish. Following some manipulation (including the use of the Hamiltonian constraint to eliminate the 3-Ricci scalar) this led to equation (2.181) which can be written schematically as:

$$\alpha_{,xx} + \alpha_{,yy} + \alpha_{,zz} = N_\alpha (\alpha, \alpha_{,j}, \psi, \psi_{,j}, \beta^i_{,j}, \phi_A, \Pi_A). \quad (2.224)$$

Here N_α is a function which is nonlinear in many of its arguments, including the first-order derivatives of α , ψ and β^k .

The elliptic equation (2.182) that governs the conformal factor, ψ , came from the Hamiltonian constraint and has the form:

$$\psi_{,xx} + \psi_{,yy} + \psi_{,zz} = N_\psi (\alpha, \psi, \beta^i_{,j}, \phi_A, \phi_{A,j}, \Pi_A), \quad (2.225)$$

where N_ψ is again nonlinear, but in this case, the key derivative nonlinearities involve only the shift vector components.

Additionally, the momentum constraints were used to derive (2.183)-(2.185), which are elliptic

equations for the components of the shift vector $\beta^k = (\beta^x, \beta^y, \beta^z)$. These take the form

$$\frac{4}{3}\beta^x_{,xx} + \beta^x_{,yy} + \beta^x_{,zz} + \frac{1}{3}(\beta^y_{,yx} + \beta^z_{,zx}) = N_{\beta^x}(\alpha, \alpha_{,j}, \psi, \psi_{,j}, \beta^x_{,j}, \beta^y_{,y}, \beta^y_{,x}, \beta^z_{,z}, \beta^z_{,x}, \phi_{A,x}, \Pi_A), \quad (2.226)$$

$$\beta^y_{,xx} + \frac{4}{3}\beta^y_{,yy} + \beta^y_{,zz} + \frac{1}{3}(\beta^x_{,xy} + \beta^z_{,zy}) = N_{\beta^y}(\alpha, \alpha_{,j}, \psi, \psi_{,j}, \beta^y_{,j}, \beta^z_{,z}, \beta^z_{,y}, \beta^x_{,x}, \beta^x_{,y}, \phi_{A,y}, \Pi_A), \quad (2.227)$$

$$\beta^z_{,xx} + \beta^z_{,yy} + \frac{4}{3}\beta^z_{,zz} + \frac{1}{3}(\beta^x_{,xz} + \beta^y_{,yz}) = N_{\beta^z}(\alpha, \alpha_{,j}, \psi, \psi_{,j}, \beta^z_{,j}, \beta^x_{,x}, \beta^x_{,z}, \beta^y_{,y}, \beta^y_{,z}, \phi_{A,z}, \Pi_A). \quad (2.228)$$

Here, the right-hand-side functions N_{β^x} , N_{β^y} and N_{β^z} , although generally nonlinear, are linear in the first order derivatives of the β^k .

These five elliptic PDEs have to be supplemented by boundary conditions. As discussed in Sec. 2.4.5, for all of the results reported in this thesis, we used Dirichlet boundary conditions as follows

$$\alpha|_B = 1, \quad \psi|_B = 1, \quad \beta^x|_B = 0, \quad \beta^y|_B = 0, \quad \beta^z|_B = 0. \quad (2.229)$$

where the notation $f|_B$ means “ f evaluated on the boundary of the (finite) solution domain”.

Numerically, and as discussed in more detail in Chap. 4 and App. B, the elliptic equations were discretized using second order finite difference methods, on a uniform mesh (constant spacing h in all three of the coordinate directions). The resulting discrete equations were then solved using a multigrid algorithm which is described in Sec. 4.3.

The hyperbolic part of our model consists of the four first-order-in-time PDEs (2.173)–(2.174), all of which originate from the general relativistic Klein-Gordon equation for a complex scalar field, $\phi = \phi_1 + i\phi_2$, where ϕ_1 and ϕ_2 are real-valued. These equations are of the form

$$\dot{\phi}_A = N_{\phi_A}(\alpha, \psi, \beta^i, \phi_{A,j}, \Pi_A) \quad (2.230)$$

and

$$\dot{\Pi}_A = N_{\Pi_A}(\alpha, \alpha_{,j}, \psi, \psi_{,j}, \beta^i, \beta^i_{,i}, \phi_A, \phi_{A,j}, \phi_{A,jj}, \Pi_A, \Pi_{A,j}), \quad (2.231)$$

for $A = 1, 2$, and where Π_A are the momenta conjugate to ϕ_A . In this case we observe that the source functions N_{ϕ_A} and N_{Π_A} are *linear* in the scalar field quantities themselves.¹⁹ As with the

¹⁹This assumes that the scalar field potential has only a mass term, which is the case for the calculations reported here.

elliptic equations, we impose Dirichlet boundary conditions on the hyperbolic variables:

$$(\phi_A)|_B = 0 \quad \text{and} \quad (\Pi_A)|_B = 0. \quad (2.232)$$

The hyperbolic equations were also discretized via second order finite difference techniques, and on the same grid used for the elliptic PDEs. More specifically, we used an implicit Crank-Nicholson scheme (Sec. 4.1.3 for details), and the resulting algebraic equations were solved with a point-wise Newton-Gauss-Seidel iteration (App. D for details). Again, Chap. 4 and App. B provide many more details concerning all of the major numerical techniques and algorithms we used in the current work.

The reader will note that we have not yet discussed the issue of initial data for our model which, of course, is an absolutely crucial part of the complete specification of any initial-boundary value problem. Modulo potential difficulties due to the nonlinearity of the equations, we can, in principle, specify *arbitrary* values for the scalar fields, ϕ_A , and the conjugate momenta, Π_A , as long as those values are sufficiently differentiable,²⁰ and compatible with the boundary conditions. The elliptic equations can then be solved to determine the initial values for the metric functions. The future (or past) development of the initial data will then be given by the solution of the coupled elliptic-hyperbolic system, subject to the boundary conditions.

However, our principal aim is to study the interaction of boson stars, which, we recall, are localized, gravitationally bound configurations of the scalar field that, in isolation, produce time-independent gravitational fields. The process of determining initial data for even a single such star is a non-trivial matter: generating initial states for binaries adds a few more complications. We thus devote the entire next chapter to the issue of computing this specific class of initial data.

²⁰Mathematically, it would suffice to have initial data that is twice differentiable, but especially given that we are using centred finite difference techniques, it will be more convenient to require the data to be smooth—i.e. infinitely differentiable.

Chapter 3

Initial Data

In this chapter we describe the computation of initial configurations for our model that contain boson stars. Although we are most interested in the case of initial data describing boson star binaries, the problem of determining data for a single star is interesting in its own right, and, of course, forms the basis for the task of constructing an initial state for two stars. To generate data for one star, we adopt a particular ansatz, detailed in Sec. 3.4, in which both the spacetime and the scalar field, ϕ , are spherically symmetric, and where due to an assumed harmonic time-dependence for ϕ , the spacetime is time independent.

Now, it transpires that for spherically symmetric spacetimes, the requirement of conformal 3-flatness does *not* imply that we are considering an approximation of the full Einstein equations, as it does in the general 3D case. Rather, as we will discuss in Sec. 3.2, in spherical symmetry conformal flatness amounts to a specific choice of radial coordinate, r , that we can always make, and which we can further combine with maximal slicing to completely fix the spherical coordinate system. For this reason the first five sections below treat the full Einstein equations coupled to a complex scalar field (with some general self-interaction potential), and we thus begin in Sec. 3.1 with a review of the $3+1$ equations for spherically symmetric spacetimes. Additionally, since it is easiest to determine the boson star solutions using a so-called areal radial coordinate, R , and then transform the solutions to the coordinate r in which the 3-metric is conformally flat, we discuss the relevant Einstein equations in both coordinate systems in Secs. 3.3 and 3.2. Sec. 3.4 then defines the ansatz used to generate a single boson star. This reduces the field equations to a system of ordinary differential equations ODEs for the scalar field and metric variables, which is further an eigenvalue problem. We briefly discuss how the system is solved, and then in Sec. 3.5 comment on some of the properties of the solutions.

The last two sections of the chapter then return focus to the 3D case, and the use of the results from the spherically symmetric general relativistic calculations to provide initial conditions for our model. Sec. 3.6 describes the straightforward process that we use to set up initial data in

Cartesian coordinates for one or two stars, and Sec. 3.7 details the further transformations of the field quantities that are used to give stars non-vanishing velocities at the initial time.

We emphasize that apart from the material in Secs. 3.6 and 3.7, none of what is discussed here is original to this work. Indeed, general relativistic spherically symmetric boson stars have been studied by a large number of authors, so there is a considerable literature on the subject (see [119] for a review). In particular, much of the presentation in Secs. 3.1–3.5 follows Chap. 4 of Lai’s PhD thesis [128] (which the interested reader can consult for additional details) as well as unpublished lecture notes due to Choptuik [154].

3.1 Spherically Symmetric Spacetime

In general relativity, spherical symmetry can be precisely defined in terms of the symmetry group, $SO(3)$, whose group orbits are 2-spheres, and which, physically, is associated with rotations in three dimensional space [120]. In particular, spherical symmetry dictates that the spacetime metric, g_{ab} , be invariant under the action of $SO(3)$. Choosing a set of coordinates (t, r, θ, ϕ) which is adapted to the symmetry, the metric induced on any 2 sphere (i.e. any collection of events defined by $t = \text{const.}$, $r = \text{const.}$), is some multiple of the metric on the unit 2-sphere, which itself is given by

$$d\Omega^2 = d\theta^2 + \sin^2\theta d\phi^2. \quad (3.1)$$

It is straightforward to argue that a completely general form for the line element on a $t = \text{const}$ hypersurface in spherical symmetry is then

$${}^{(3)}ds^2 = a^2(t, r) dr^2 + r^2 b^2(t, r) d\Omega^2, \quad (3.2)$$

where we stress that the functions a and b depend only on t and r . Similarly, it can be readily established that the most general line element for spherically spacetime can be written in the 3+1 form

$$ds^2 = (-\alpha(t, r)^2 + a^2\beta(t, r)^2) dt^2 + 2a^2\beta dt dr + a^2 dr^2 + r^2 b^2 d\Omega^2. \quad (3.3)$$

Here, $\alpha(t, r)$ is the lapse function as usual, and $\beta(t, r)$ is the single non-vanishing component of the shift vector $\beta^k = (\beta^r, \beta^\theta, \beta^\phi) \equiv (\beta, 0, 0)$.

Additionally, it is not hard to prove from (2.44) that the extrinsic curvature corresponding to

the spatial metric, γ_{ij} , has only two independent components in spherical symmetry, K^r_r and K^θ_θ :

$$K^i_j = \text{diag}(K^r_r(t, r), K^\theta_\theta(t, r), K^\theta_\theta(t, r)). \quad (3.4)$$

Before writing down the 3 + 1 Einstein equations for spherically symmetric spacetimes, it is convenient to introduce auxiliary fields, $\Phi(t, r)$ and $\Pi(t, r)$, which are defined in terms of the complex scalar field, ϕ as follows:

$$\Phi(t, r) \equiv \phi'(t, r) \equiv \partial_r \phi(t, r) \quad \text{and} \quad \Pi(t, r) \equiv \frac{a}{\alpha}(\dot{\phi} - \beta\phi'). \quad (3.5)$$

(Here and throughout this chapter we use the overdot and prime notations for partial differentiation with respect to the time and radial coordinates respectively.)²¹ In terms of these auxiliary variables, and using (2.40)-(2.43), the non-vanishing components of the stress-energy tensor are:

$$\begin{aligned} \rho &= \frac{|\Phi|^2 + |\Pi|^2}{2a^2} + \frac{U(|\phi|^2)}{2}, & j_r &= -\frac{\Pi^*\Phi + \Pi\Phi^*}{2a} = a^2 j^r, \\ S^r_r &= \rho - U(|\phi|^2), & S^\theta_\theta &= \frac{|\Pi|^2 - |\Phi|^2}{2a^2} - \frac{U(|\phi|^2)}{2}, \\ S^\phi_\phi &= S^\theta_\theta, & S &= \frac{3|\Pi|^2 - |\Phi|^2}{2a^2} - \frac{3}{2}U(|\phi|^2). \end{aligned}$$

where $U(|\phi|^2)$ is the scalar field's interaction potential. The spherically symmetric Klein Gordon equation gives us the following first-order-in-time evolution equations for the scalar field variables, ϕ , Φ and Π :

$$\dot{\phi} = \frac{\alpha}{a}\Pi + \beta\Phi, \quad (3.6)$$

$$\dot{\Phi} = \left(\frac{\alpha}{a}\Pi + \beta\Phi\right)', \quad (3.7)$$

$$\dot{\Pi} = \frac{1}{(rb)^2} \left[(rb)^2 \left(\beta\Pi + \frac{\alpha}{a}\Phi \right) \right]' + 2 \left[\alpha K^\theta_\theta - \beta \frac{(rb)'}{rb} \right] \Pi - \alpha a \frac{dU(|\phi|^2)}{d|\phi|^2} \phi. \quad (3.8)$$

²¹As a technical note, we point out that Π as defined in (3.5) is *not* the conjugate momentum of the field ϕ ; but that $r^2 b^2 \sin(\theta)\Pi$ is. The definition (3.5) is motivated by the form of the dynamical equations for Φ and Π that result (vis. (3.7)-(3.8)).

Continuing, from (2.44) we have evolution equations for the metric functions a and b :

$$\dot{a} = -\alpha a K^r_r + (a\beta)', \quad (3.9)$$

$$\dot{b} = -\alpha b K^\theta_\theta + \frac{\beta}{r}(rb)', \quad (3.10)$$

and from (2.45), evolution equations for the extrinsic curvature components, K^r_r and K^θ_θ :

$$\dot{K}^r_r = \beta K^r_r{}' - \frac{1}{a} \left(\frac{\alpha'}{a} \right)' + \alpha \left\{ -\frac{2}{arb} \left[\frac{(rb)'}{a} \right]' + K K^r_r - 4\pi \left[\frac{2|\Phi|^2}{a^2} + U(|\phi|^2) \right] \right\}, \quad (3.11)$$

$$\dot{K}^\theta_\theta = \beta K^\theta_\theta{}' + \frac{\alpha}{(rb)^2} - \frac{1}{a(rb)^2} \left[\frac{\alpha rb}{a} (rb)' \right]' + \alpha K K^\theta_\theta - 4\pi \alpha U(|\phi|^2). \quad (3.12)$$

Turning to the constraint equations, the Hamiltonian constraint (2.49) reduces to

$$\mathcal{R} + 4K^r_r K^\theta_\theta + 2K^{\theta\theta}{}^2 = 16\pi\rho, \quad (3.13)$$

which, using the explicit form of the 3-Ricci scalar, \mathcal{R} , becomes

$$-\frac{2}{arb} \left\{ \left[\frac{(rb)'}{a} \right]' + \frac{1}{rb} \left[\left(\frac{rb}{a} (rb)' \right)' - a \right] \right\} + 4K^r_r K^\theta_\theta + 2K^{\theta\theta}{}^2 = 8\pi \left[\frac{|\Phi|^2 + |\Pi|^2}{a^2} + U(|\phi|^2) \right]. \quad (3.14)$$

Finally the momentum constraint (2.50) is

$$K^{\theta\theta}{}' + \frac{(rb)'}{rb} (K^\theta_\theta - K^r_r) = 2\pi \frac{\Pi^* \Phi + \Pi \Phi^*}{a}. \quad (3.15)$$

We have now displayed a complete set of equations for the Einstein-Klein-Gordon system in spherical symmetry, which is valid for any system of coordinates (t, r, θ, ϕ) that are adapted to the symmetry. In the next two sections we specialize these equations to the cases of two specific coordinate systems that bear on our current work.

3.2 Maximal-Isotropic Coordinates

As we discussed in Sec. 2.4 the choice of maximal slicing fixes the time coordinate by demanding that the trace of the extrinsic curvature, $K \equiv K^i_i$, vanish at all times. To implement this condition in spherical symmetry then, we require $K(0, r) = 0$ and $\dot{K}(t, r) = 0$ for all t and r . We note that since

we have $K^i_j = (K^r_r, K^\theta_\theta, K^\theta_\theta)$, and thus $K = K^r_r + 2K^\theta_\theta$, the choice $K = 0$ allows us to eliminate one of K^r_r or K^θ_θ from the overall set of equations. In particular, we will take

$$K^\theta_\theta = -\frac{1}{2}K^r_r, \quad (3.16)$$

in the following. The specification of the coordinate system is completed by fixing the radial coordinate. We do this by requiring that the 3-metric be conformally flat, so that again introducing the conformal factor, $\psi(t, r)$, as in Sec. 2.4.2, we have

$${}^{(3)}ds^2 = \psi^4 (dr^2 + r^2 d\Omega^2). \quad (3.17)$$

We reemphasize the point made earlier in this chapter that in spherical symmetry conformal flatness amounts to a coordinate choice: we are always free to write the 3-metric in the form (3.17), and it does *not* imply that we are approximating Einstein's equations in some way, as it does when applied to the generic 3D case.

In terms of the general form (3.2) for the spherically symmetric line element, (3.17) implies

$$a = b \equiv \psi^2(t, r). \quad (3.18)$$

Traditionally, the radial coordinate implied by (3.17), or equivalently (3.18), has been termed *isotropic*, and, although we will also use that terminology, the reader should keep in mind that “isotropic” is synonymous with “conformally flat” in this context.

Operationally, to implement the isotropic condition (3.18), we require that it holds on the initial hypersurface, so that $a(0, r) = b(0, r)$, and that $\dot{a}(t, r) = \dot{b}(t, r)$ for all t and r . Equating the right hand sides of (3.9) and (3.10) and using $a(t, r) = b(t, r)$, we easily derive the following ODE for the shift vector component, β :

$$r \left(\frac{\beta}{r} \right)' = \alpha(K^r_r - K^\theta_\theta), \quad (3.19)$$

or, using (3.16),

$$r \left(\frac{\beta}{r} \right)' = \frac{3}{2}\alpha K^r_r. \quad (3.20)$$

Returning to the maximal slicing condition, we have $K = 0$ and $\dot{K} = 0$. We thus take the trace

of (2.45) and equate the right hand side of the resulting equation to 0. We then use the Hamiltonian constraint to eliminate the 3-Ricci scalar, \mathcal{R} , appearing in the equation in favour of terms involving the extrinsic curvature and stress-energy components, and perform additional simplifications using $K = 0$ and $\dot{K}^\theta_\theta = -\dot{K}^r_r/2$. After some manipulation we find the following equation for the lapse:

$$\alpha'' + \frac{a}{(rb)^2} \left[\frac{(rb)^2}{a} \right]' \alpha' + \left[4\pi a^2 U(|\phi|^2) - 8\pi |\Pi|^2 - \frac{3}{2} a^2 K^r_r{}^2 \right] \alpha = 0,$$

which using $a = b = \psi^2$, becomes

$$\alpha'' + \frac{2}{r\psi^2} \frac{\partial}{\partial(r^2)} (r^2\psi^2) \alpha' + \left[4\pi\psi^4 U(|\phi|^2) - 8\pi |\Pi|^2 - \frac{3}{2} \psi^4 K^r_r{}^2 \right] \alpha = 0. \quad (3.21)$$

Here we note that the operator $\partial/\partial r^2$ takes a derivative with respect to r^2 . Its use here is motivated by the need to maintain regularity in numerical solutions of (3.21), and is an example of a general technique first introduced to the numerical relativity community by Evans [155]. The basic observation is that as $r \rightarrow 0$, we have $\psi(t, r) \rightarrow \psi_0(t) + r^2\psi_2(r) + O(r^4)$, and thus $r^2\psi(t, r) \rightarrow r^2\psi_0(t) + O(r^4)$. Since the leading order term goes like r^2 , finite difference operators based on $\partial/\partial(r^2)$ rather than $\partial/\partial r$ tend to produce numerical solutions that are smoother near $r = 0$. This issue is discussed in more detail in App. B, Sec. B.2.

Eqs. (3.21) and (3.20) determine the kinematical geometrical variables, α and β , respectively. We now need equations for the dynamical geometrical variables, namely the conformal factor, ψ , and the extrinsic curvature component, K^r_r . Here, due to the overdetermined nature of the Einstein equations, which, we recall, results from the coordinate invariance of the theory, we have two choices for each function. We can either use evolution equations derived from (3.9)–(3.11), or we can use the constraint equations (3.14) and (3.15). Here we choose the latter option, which produces what is known as a *fully constrained* scheme [122]. Specifically the Hamiltonian constraint (3.14) becomes an ODE for ψ :

$$\frac{3}{\psi^5} \frac{d}{dr^3} \left(r^2 \frac{d\psi}{dr} \right) + \frac{3}{16} K^r_r{}^2 = -\pi \left(\frac{|\Phi|^2 + |\Pi|^2}{\psi^4} + U(|\phi|^2) \right), \quad (3.22)$$

while the momentum constraint (3.15) provides an ODE for K^r_r :

$$K^r_r{}' + 3 \frac{(r\psi^2)'}{r\psi^2} K^r_r = -\frac{4\pi}{\psi^2} (\Pi^* \Phi + \Pi \Phi^*). \quad (3.23)$$

We note that our ability to completely determine the geometric variables without the explicit use of any evolution equations (and, in particular, without using any equations of the form $\dot{K}^i_j = \dots$) is a reflection of the fact that, in spherical symmetry, the general relativistic gravitational field has no independent dynamics: genuine time dependence of the metric ²² must result from time dependence of a matter field that is coupled to gravity.

Finally, the specialization of (3.6)–(3.8) to maximal-isotropic coordinates yields the following set of evolution equations for the scalar field variables:

$$\dot{\phi} = \frac{\alpha}{\psi^2}\Pi + \beta\Phi, \quad (3.24)$$

$$\dot{\Phi} = \left(\frac{\alpha}{\psi^2}\Pi + \beta\Phi \right)', \quad (3.25)$$

$$\dot{\Pi} = \frac{3}{\psi^4} \frac{d}{dr^3} \left[r^2 \psi^4 \left(\beta\Pi + \frac{\alpha}{\psi^2}\Phi \right) \right] - 2 \left[\frac{1}{2}\alpha K^r_r + \beta \frac{(r\psi^2)'}{r\psi^2} \right] \Pi - \alpha\psi^2 \frac{dU(|\phi|^2)}{d|\phi|^2} \phi. \quad (3.26)$$

The set of equations (3.20)–(3.26) fully determine the spherically symmetric Einstein-Klein-Gordon system in maximal-isotropic coordinates. Note, however, that these equations must be supplemented with initial values and boundary conditions in order to generate a unique solution. However, we have not used this system to generate any of the results discussed below, and have included the derivation of the equations of motion primarily for the sake of completeness. We will thus conclude our discussion of this particular coordinate choice at this point and refer the interested reader to App. B of Lai’s thesis [128] for details of a numerical implementation of (3.20)–(3.26). When we return to the maximal-isotropic coordinate system in Sec. 3.4 it will be in the context of the initial data problem.

3.3 Polar-Areal Coordinates

The second coordinate system for spherically symmetric spacetimes that we consider also uses a condition on the extrinsic curvature to fix the time coordinate. In this case we demand that

$$K \equiv K^i_i = K^R_R, \quad (3.27)$$

²²As opposed to apparent time dependence arising from a choice of coordinates, which can in principle be identified by finding a solution of the Killing equations $\mathcal{L}_\xi g_{ab} = 0$, where ξ is the sought-for timelike Killing vector (see Sec. 3.4).

where we note that we use T and R in this section to denote the time and radial coordinates, respectively, with $\dot{u} \equiv \partial u / \partial T$ and $u' \equiv \partial u / \partial R$. Eq. (3.27) is known as the polar slicing condition [156], and, as for the choice of maximal slicing, has the immediate consequence of reducing the number of independent components of the extrinsic curvature. From the definition of K and (3.27) we have

$$K \equiv K^R_R + 2K^\theta_\theta = K^R_R, \quad (3.28)$$

which implies

$$K^\theta_\theta = 0. \quad (3.29)$$

This last result is then used to derive an equation that must be satisfied by the lapse function, $\alpha(T, R)$, at any time T . Specifically, we require that the initial data satisfy $K^\theta_\theta(0, R) = 0$, and then impose $\dot{K}^\theta_\theta(T, R) = 0$ for all T and R :

$$K^\theta_\theta(T, R) = \dot{K}^\theta_\theta(T, R) = 0. \quad (3.30)$$

As the name ‘‘polar-areal’’ suggests, the spatial coordinate is fixed by demanding that R directly measure the proper surface area of any $T = \text{const.}$, $R = \text{const.}$ 2-sphere. Now, in terms of our general form (3.2) for the 3-metric, this area, $A(T, R)$, is given by

$$A(T, R) = 4\pi R^2 b(T, R)^2, \quad (3.31)$$

so if R is to be areal, we must have $b(T, R) \equiv 1$. Thus, the choice of radial coordinate eliminates another of the geometric dynamical variables from the system. Moreover, the general evolution equation (3.10) reads

$$\dot{b} = -\alpha b K^\theta_\theta + \frac{\beta}{R} (Rb)' \quad (3.32)$$

in the current case. Paralleling the implementation of the other coordinate conditions discussed thus far, to enforce $b(T, R) \equiv 1$ we require $b(0, R) = 1$ at $T = 0$, and $\dot{b}(T, R) = 0$ for all T and R . Using these relations, as well as $K^\theta_\theta(T, R) = 0$ in (3.32), we find that the shift vector component, β , must identically vanish. Thus the spacetime line-element in polar-areal coordinates is simply

$$ds^2 = -\alpha^2(T, R)dT^2 + a^2(T, R)dR^2 + R^2 d\Omega^2, \quad (3.33)$$

which can be viewed as a natural generalization of the familiar Schwarzschild line element

$$ds^2 = - \left(1 - \frac{2M}{R}\right) dT^2 + \left(1 - \frac{2M}{R}\right)^{-1} dR^2 + R^2 d\Omega^2, \quad (3.34)$$

to the case of time dependent spherical geometries. We further note that the elimination of *three* of the geometric quantities—namely b , K^θ_θ and β —using *two* coordinate conditions is a special feature of the polar-areal system.

Not surprisingly, given the simple (and diagonal) form of the line element (3.33), the resulting 3 + 1 Einstein equations also assume a very simple form. Once more, since only two dynamical geometrical variables remain—the metric function, a , and the extrinsic curvature component, K^R_R —we can use the constraint equations in lieu of evolution equations to determine them.

Using $b = 1$ and $K^\theta_\theta = 0$ in the general form of the Hamiltonian constraint (3.14), we find that a must satisfy the ODE

$$a' = \frac{1}{2} \left\{ \frac{a}{R} (1 - a^2) + 4\pi R a [|\Phi|^2 + |\Pi|^2 + a^2 U(|\phi|^2)] \right\}, \quad (3.35)$$

while the momentum constraint (3.15) provides an *algebraic* equation for K^R_R :

$$K^R_R = -\frac{2\pi R}{a} (\Pi^* \Phi + \Pi \Phi^*). \quad (3.36)$$

As discussed above, the polar slicing condition requires that $K^\theta_\theta(T, R) = 0$ and $\dot{K}^\theta_\theta(T, R) = 0$. Setting the right hand side of (3.12) to 0, and using $K = 0$, $\beta = 0$ and $b = 1$, we derive the following ODE for the lapse:

$$\alpha' = \frac{\alpha}{2} \left\{ \frac{a^2 - 1}{R} + 4\pi R [|\Phi|^2 + |\Pi|^2 - a^2 U(|\phi|^2)] \right\}. \quad (3.37)$$

Finally, for the scalar field variables, using $b = 1$, $K^\theta_\theta = 0$ and $\beta = 0$ in (3.6)–(3.8) we find:

$$\dot{\phi} = \frac{\alpha}{a} \Pi, \quad (3.38)$$

$$\dot{\Phi} = \left(\frac{\alpha}{a} \Pi \right)', \quad (3.39)$$

$$\dot{\Pi} = 3 \frac{\partial}{\partial(R^3)} \left[R^2 \frac{\alpha}{a} \Phi \right] - \alpha a \frac{dU(|\phi|^2)}{d|\phi|^2} \phi. \quad (3.40)$$

The set (3.35)–(3.40) constitutes a sufficient set of equations for the spherically symmetric Einstein-Klein-Gordon system in polar-areal coordinates (again, modulo initial values and boundary conditions). In the next section we use a further simplified subset of these equations to compute solutions representing single boson stars.

3.4 Constructing Boson Stars: The Static Ansatz

A spherically symmetric, localized, time independent configuration of matter captures the simplest notion of a star. Provided that the matter is regular everywhere, we should expect such a configuration to produce a gravitational field that is also spherically symmetric, time independent and globally regular. Moreover, if we are to be able to study the time evolution of these objects to any significant degree, then they should also be dynamically stable. For the case of a complex field it is not possible to construct such states on the basis of time independence of ϕ itself. Indeed, Friedberg, Lee and Pang [107] demonstrated that in order for a boson star to be in a (minimal energy) ground state—a necessary condition for stability— ϕ must have harmonic time dependence. Thus, we adopt the following ansatz for the complex scalar field

$$\phi(t, r) = \phi_0(r) e^{-i\omega t}, \quad (3.41)$$

where ω is assumed to be a non-negative real constant, and where, for the time being, (t, r) are a general set of coordinates for spherically symmetric, time dependent spacetimes as discussed in Sec. 3.1. As can be quickly verified by examination of the form of (2.69), all components of the stress-tensor, $T_{\mu\nu}$, become time independent under this assumption. Consequently, the spacetime (i.e. the gravitational field) that is produced *can* also be expected to be time independent.

Now, we note that the most generic sense of “time independence” in general relativity means that the spacetime has a timelike Killing vector field, in which case the spacetime is said to be *stationary*. Additionally, there is a more restrictive definition which requires that we be able to foliate the spacetime with hypersurfaces, Σ_t , such that the Killing vector is everywhere orthogonal to these slices. In this case the spacetime is said to be *static* and it is straightforward to show that in coordinates (t, x^i) adapted to the timelike symmetry, the metric components $g_{\mu\nu}$ must be invariant under the “time reflection” symmetry, $t \rightarrow -t$ [120]. An example of a spacetime which is stationary but not static is provided by an axially symmetric star composed of self-gravitating fluid

that is in rotation about some symmetry axis, and which thus has some net angular momentum. Although one *can* construct boson star models that have angular momentum, we will not consider them here, so the demand that spacetime be static is the appropriate choice. We observe that, assuming that t is adapted to the timelike symmetry—which we will hereafter require—(3.41) is compatible with this demand. We will therefore refer to it as the “static ansatz”.

We now consider what the ansatz implies for the spherically symmetric 3+1 equations displayed in the two previous sections. First, the static requirement immediately implies that all of the metric components, $g_{\mu\nu}$, must be time independent functions, and that in terms of the general 3+1 form of the line element (3.3), the shift vector component β , must vanish:

$$\beta(t, r) = 0. \tag{3.42}$$

We observe that this last condition is always satisfied in polar-areal coordinates, but that it needs to be imposed explicitly when working in maximal-isotropic coordinates.

Since the specific metric functions $a(t, r)$ and $b(t, r)$ must have vanishing time derivatives, and we also have $\beta = 0$, Eqs. (3.9) and (3.10) then imply that the extrinsic curvature components also vanish:

$$K^r_r = K^\theta_\theta = 0. \tag{3.43}$$

As a consequence, for the case of a static solution, polar slicing is also maximal, and vice versa, so that to perform a coordinate transformation of a static spacetime from polar-areal coordinates to the maximal-areal system, we only have to transform the radial coordinate.

Considering the scalar field variables, we note that (3.41) results in the following expressions for the time and radial derivatives of ϕ :

$$\dot{\phi}(t, r) = -i\omega\phi_0(r)e^{-i\omega t}, \tag{3.44}$$

$$\phi'(t, r) = \phi'_0(r)e^{-i\omega t}. \tag{3.45}$$

From these relations we then have

$$\Pi(t, r) = -i\omega \frac{a}{\alpha} \phi_0(r) e^{-i\omega t} \equiv \Pi_0(r) e^{-i\omega t}, \quad (3.46)$$

$$\Phi(t, r) = \phi'_0(r) e^{-i\omega t} \equiv \Phi_0(r) e^{-i\omega t}. \quad (3.47)$$

We also have

$$\Pi^* \Phi + \Pi \Phi^* = 0, \quad (3.48)$$

which means that the momentum constraint (3.15) is satisfied identically.

We now adopt polar-areal coordinates (t, R) (where we use t rather than T in view of the above observation that polar and maximal slicings are identical for static spacetimes), and use the above results in the set of equations for the spherically symmetric EKG system that was given in the previous section. After some manipulation, the Hamiltonian constraint (3.35), the slicing condition (3.37), and the evolution equation for the scalar field (3.40), become the following system of coupled ODEs:

$$a'(R) = \frac{1}{2} \left\{ \frac{a}{R} (1 - a^2) + 4\pi R a \left[\Phi_0^2 + \omega^2 \frac{a^2}{\alpha^2} \phi_0^2 + a^2 U(\phi_0^2) \right] \right\}, \quad (3.49)$$

$$\alpha'(R) = \frac{\alpha}{2} \left\{ \frac{a^2 - 1}{R} + 4\pi R \left[\Phi_0^2 + \omega^2 \frac{a^2}{\alpha^2} \phi_0^2 - a^2 U(\phi_0^2) \right] \right\}, \quad (3.50)$$

$$\phi'_0(R) = \Phi_0, \quad (3.51)$$

$$\Phi'_0(R) = - (1 + a^2 - 4\pi R^2 a^2 U(\phi_0^2)) \frac{\Phi_0}{R} + a^2 \left[\frac{dU(\phi_0^2)}{d\phi_0^2} - \frac{\omega^2}{\alpha^2} \right] \phi_0. \quad (3.52)$$

Note that a prime now denotes *ordinary* differentiation and that the equation $\phi'_0 = \Phi_0$ follows from the *definition* of Φ_0 in (3.47). We will subsequently refer to this set of equations as the polar-areal ODE system.

We observe that this system contains terms such as $a(1 - a^2)/R$ and Φ_0/R which, naively at least, appear to be singular at the origin, $R = 0$. In order to ensure regularity at the origin, certain conditions must be imposed on $a(R)$, $\alpha(R)$ and $\phi_0(R)$. A thorough mathematical treatment of this subject defines a function (or, more generally, a component of a tensor) to be regular at the origin if, using Cartesian coordinates, (x, y, z) , it has a convergent Taylor series in a neighbourhood of $(x, y, z) = (0, 0, 0)$; i.e. at $R = 0$ [156]. For the case at hand (see [156, 157, 158] for additional

details) this requirement turns out to imply the following limiting forms for $a(R)$, $\alpha(R)$ and $\phi_0(R)$:

$$\lim_{R \rightarrow 0} a(R) = a_0 + a_2 R^2 + O(R^4) = 1 + a_2 R^2 + O(R^4), \quad (3.53)$$

$$\lim_{R \rightarrow 0} \alpha(R) = \alpha_0 + \alpha_2 R^2 + O(R^4), \quad (3.54)$$

$$\lim_{R \rightarrow 0} \phi_0(R) = \phi_{00} + \phi_{02} R^2 + O(R^4). \quad (3.55)$$

Here, a_0 , a_2 , α_0 , α_2 , ϕ_{00} and ϕ_{02} are constants, and the fact that $a_0 = 1$ follows from the demand that spacetime be locally flat at $R = 0$. From (3.55) we also have

$$\lim_{R \rightarrow 0} \Phi_0(R) = 2\phi_{02}R + O(R^3). \quad (3.56)$$

Using (3.53)–(3.56) in (3.49)–(3.52), it is straightforward to show that the system of ODEs is regular at $R = 0$.

Since (3.49)–(3.52) is a system of 4 ODEs, it would seem that we need 4 boundary conditions to generate a unique solution. However, there is a twist here which arises from the fact that the ODE system is actually to be treated as an eigenvalue problem, where ω is the eigenvalue. Specifically, for any choice of interaction potential $U(|\phi|^2)$, the solutions of (3.49)–(3.52) form a one-parameter family, where the central modulus of the scalar field, $\phi_0(0)$, is a convenient choice for the family parameter. Since we want the solutions of the ODE system to represent stars, and thus to describe (essentially) localized distributions of matter, we must have

$$\lim_{R \rightarrow \infty} \phi_0(R) = 0. \quad (3.57)$$

For any given choice of $\phi_0(0)$, and in conjunction with the other boundary conditions that we will enumerate below, solutions satisfying (3.57) will only exist for a discrete set of values of $\omega \equiv \omega(\phi_0(0))$ and, further, we will only be interested in the smallest such ω , which will correspond to the lowest energy boson star for a given $\phi_0(0)$. In particular the function $\phi_0(R)$ for that choice of ω will have no zero-crossings (i.e. it will be “nodeless”): higher energy eigenstates will have 1, 2, . . . zero-crossings but, again, will be of no concern to us here.

Bearing this in mind, and using (3.53)–(3.56), the following constitute a sufficient set of bound-

ary conditions for the polar-areal ODE system:

$$a(0) = 1, \quad (3.58)$$

$$\alpha(0) = \alpha_0, \quad (3.59)$$

$$\phi_0(0) = \phi_{00}, \quad (3.60)$$

$$\Phi_0(0) = 0. \quad (3.61)$$

where ϕ_{00} is freely specifiable, as is, at least momentarily, α_0 , but where it is to be understood that $\omega = \omega(\phi_0)$ must be determined so that the asymptotic condition (3.57) is also satisfied.

We now consider the boundary condition (3.59) more closely. We note that the eigenvalue ω enters in the system (3.49)–(3.52) strictly in the combination ω^2/α^2 . Combined with the fact that, apart from that term, the slicing equation (3.50) is linear and homogeneous in α , this means that if we generate a solution of (3.49)–(3.52) subject to the boundary conditions (3.58)–(3.60), then we can take

$$\omega \rightarrow k\omega, \quad (3.62)$$

$$\alpha(R) \rightarrow k\alpha(R), \quad (3.63)$$

where k is an arbitrary positive constant, and still have a solution. Moreover, since the profiles $a(R)$, $\phi_0(R)$ and $\Phi_0(R)$ will be unchanged under the rescaling (3.62)–(3.63), any solutions obtained via such transformations will correspond to the same boson star. This reflects that fact that when we adopt a time-slicing condition such as polar slicing or maximal slicing, we can always freely rescale $\alpha(t, r)$ via $\alpha(t, r) \rightarrow f(t)\alpha(t, r)$ where $f(t)$ is some arbitrary function of the time coordinate, and still have a solution of Einstein's equations. The physical interpretation of this rescaling is that we have complete freedom as to how we are to assign, or reassign, specific labels to the family of hypersurfaces, Σ_t , that foliate the spacetime.

In practice it is a common and sensible choice to perform this labelling so that coordinate time and proper time coincide for observers located at $R = \infty$, and who are at rest in the slices. This means that we want

$$\lim_{R \rightarrow \infty} \alpha(R) = 1. \quad (3.64)$$

As was the case in Sec. 2.4.5 we are now confronted with the issue of dealing with a boundary condition naturally expressed at infinity, which is problematic if our numerical computations can only extend to finite values of R . Fortunately, in this case there is a simple resolution of this issue which is based on the uniqueness of the solution of Einstein's equations in spherical symmetry when no matter is present. This is known as Birkhoff's theorem, and the interested reader can consult Hawking and Ellis [159], for example, for a proof. Here we assert that the solutions of (3.49)–(3.52)—i.e. the boson star solutions—are characterized by scalar field profiles, $\phi_0(R)$, which fall off exponentially after some characteristic radius. Thus, to well within the numerical accuracy to which we work in this thesis, the spacetime can be considered to be vacuum for $R > R_{\max}$, where R_{\max} is the limit of integration of the system (3.49)–(3.52).

Therefore, in terms of the static form of the 3 + 1 element we adopt for polar-areal coordinates:

$$ds^2 = -\alpha(R)^2 dt^2 + a(R)^2 dR^2 + R^2 d\Omega^2, \quad (3.65)$$

Birkhoff's theorem tells us that for $R > R_{\max}$ we must be able to identify (3.65) with the familiar Schwarzschild form:

$$ds^2 = -\left(1 - \frac{2M}{R}\right) dt^2 + \left(1 - \frac{2M}{R}\right)^{-1} dR^2 + R^2 d\Omega^2, \quad (3.66)$$

where the constant M is the total mass of the spacetime (i.e. $M = M_{\text{ADM}}$). Note that since

$$\lim_{R \rightarrow \infty} \left[1 - \frac{2M}{R}\right] = 1, \quad (3.67)$$

the constant time surfaces defined by (3.66) *do* satisfy the condition that proper and coordinate time coincide at infinity. Comparing (3.65) and (3.66) we thus have the following limits for the metric functions $\alpha(R)$ and $a(R)$:

$$\lim_{R \rightarrow \infty} \alpha^2(R) = \left(1 - \frac{2M}{R}\right), \quad (3.68)$$

$$\lim_{R \rightarrow \infty} a^2(R) = \left(1 - \frac{2M}{R}\right)^{-1}, \quad (3.69)$$

which implies that we should choose the rescaling defined by (3.62) and (3.63) so that

$$\alpha(R_{\max}) = \frac{1}{a(R_{\max})}. \quad (3.70)$$

The correspondence $a(R) \rightarrow (1 - 2M/R)^{-1}$, suggested by (3.69), also motivates the definition of a useful diagnostic function, the so-called *mass aspect function*:

$$M(R) \equiv \frac{R}{2} \left(1 - \frac{1}{a(R)^2} \right). \quad (3.71)$$

It can be shown that $M(R)$ defined in this way *does* measure the gravitating mass contained within an $R = \text{const.}$ sphere, so for $R \rightarrow \infty$ it must limit to the total ADM mass:

$$\lim_{R \rightarrow \infty} M(R) = M \equiv M_{\text{ADM}}. \quad (3.72)$$

To summarize, to determine a specific boson star solution we perform the following steps:

1. We choose a particular value, $\phi_0(0)$, for the central modulus of the scalar field. This constitutes the specification of the boundary condition (3.60).
2. Using this choice and the additional boundary conditions

$$a(0) = 1, \quad (3.73)$$

$$\alpha(0) = 1, \quad (3.74)$$

$$\Phi_0(0) = 0, \quad (3.75)$$

we solve the polar-areaal ODE system (3.49)–(3.52) on the interval $0 \leq R \leq R_{\max}$ by determining the eigenvalue $\omega = \omega(\phi_0(0))$ such that $\phi(R_{\max}) \rightarrow 0$. Since $\alpha(R)$ and ω will be rescaled, we choose $\alpha(0) = 1$ arbitrarily and for convenience.

3. Once the solution has been determined, we rescale $\alpha(R)$ and ω using

$$\omega \rightarrow k\omega, \quad (3.76)$$

$$\alpha(R) \rightarrow k\alpha(R), \quad (3.77)$$

where k is chosen so that (3.70) is satisfied.

The only remaining technical issue to be discussed is how we determine the eigenvalue ω in step 2 of above procedure. We do this with a straightforward shooting method [160]. This means that for any specified value of $\phi_0(0)$, we must first determine an initial bracket $[\omega_-, \omega_+]$ satisfying $\omega_- < \omega < \omega_+$. Solutions computed using ω_- and ω_+ will display distinct behaviours as $R \rightarrow R_{\max}$. Specifically, we find integration with $\omega = \omega_-$ results in $\phi_0(R) \rightarrow \infty$ as $R \rightarrow R_{\max}$, whereas for $\omega = \omega_+$ the integration leads to $\phi_0(R) \rightarrow -\infty$. Once the initial bracket has been found, we use a bisection method [160] to compute increasingly accurate estimates of ω , where after each step in the bisection, the appropriate end point of the interval, $[\omega_-, \omega_+]$, is replaced with the current estimate $(\omega_- + \omega_+)/2$.

The solution method that we have just described was coded in a FORTRAN subroutine called `bsidpa` that takes care of the integration of the polar areal ODE system, the shooting process *per se* and the rescaling of the lapse function. The routine is documented in App. A and we note that we have made the code available to others who might find use for it.

Once we have computed a boson star solution in polar-areal coordinates, we can transform the solution to maximal-isotropic coordinates, which are compatible with the coordinates we use in our 3D evolution code. As mentioned above, for static solutions the time coordinates in the two systems are identical, so the transformation only involves the radial coordinates. As shown in full detail in App. D of Lai's PhD thesis [128], this transformation can be made by solving the ODE:

$$\frac{dr}{dR} = a \frac{r}{R}, \quad (3.78)$$

subject to the boundary condition:

$$r|_{R=R_{\max}} = \left[\left(\frac{1 + \sqrt{a}}{2} \right)^2 \frac{R}{a} \right]_{R=R_{\max}}. \quad (3.79)$$

Here r and R are the isotropic and areal radial coordinates, respectively, and $a \equiv a(R)$ is the metric function that appears in the static, areal form of the 3-dimensional spherically symmetric line element

$${}^{(3)}ds^2 = a^2 dR^2 + R^2 d\Omega^2. \quad (3.80)$$

Once $R(r)$ has been determined, the conformal factor $\psi(r)$ is easily computed from

$$\psi(r) = \sqrt{\frac{R}{r(R)}}, \quad (3.81)$$

with

$$\psi(0) = \sqrt{\left(\left.\frac{dr}{dR}\right|_{R=0}\right)^{-1}}, \quad (3.82)$$

as follows from an application of l'Hopital's rule.

3.5 Boson Stars: Some Basic Properties

Fig. 3.1 shows four separate solutions of the polar-areal ODE system, computed for central scalar field values $\phi_0(0) = 0.01, 0.03, 0.05$ and 0.07 , with a potential $U(|\phi|^2)$ having only a mass term

$$U(|\phi|^2) = m^2\phi^2. \quad (3.83)$$

Here m is the mass parameter for the field, which, by our choice of units, satisfies $m = 1$. Each plotted solution represents a distinct boson star.

We note that as $\phi_0(0)$ increases, the stars become increasingly compact—meaning that the star's size (defined, for example, as the radius, R , at which $M(R)$ given by (3.71) is 99% of M_{ADM})—*decreases* as $\phi_0(0)$ increases. We also observe that the central values of the lapse, $\alpha(R)$, decrease as ϕ_0 increases, while the $R = 0$ values of the conformal factor, $\psi(R)$, increase. All of these trends are indicative of the fact that the gravitational self-interaction of the scalar field strengthens as ϕ_0 assumes larger values. Further, although it is not immediately apparent from the plots, we assert that all of these solutions satisfy the appropriate asymptotic boundary conditions, namely that $\lim_{R \rightarrow \infty} \psi(R) = \lim_{R \rightarrow \infty} \alpha(R) = 1$. The figure also shows graphs of the mass aspect function, $M(R)$, for the stars, from which one can see a convergence of $M(R) \rightarrow M_{\text{ADM}}$, with the convergence being more rapid for the stars defined by larger $\phi_0(0)$. Finally we note that all four of these stars are dynamically stable against radial perturbations, although, as we will discuss shortly, this is *not* the case for all solutions in the one-parameter family.

Fig. 3.2 displays plots of the ADM mass, M_{ADM} , as a function of the central scalar field value $\phi_0(0)$ (left panel), as well as a function of two estimates of the stellar radius, R_{99} and

R_{95} (right panel). R_{99} and R_{95} are defined to be the radii for which $M(R) = 0.99 M_{\text{ADM}}$ and $M(R) = 0.95 M_{\text{ADM}}$, respectively: both of these definitions are commonly used in the boson star literature. We note that $M(\phi_0(0))$ has an absolute maximum at $\phi_0(0) = \phi_0^* \approx 0.08$ where it attains a value $M_{\text{max}} = M(\phi_0^*) \approx 0.633$ (again, recall that we work in units in which $c = G = m = 1$). In both plots the solid triangles label stars computed with $\phi_0(0) = 0.01, 0.02, 0.03, 0.04, 0.05, 0.06$ and 0.07 (left to right along the $M(\phi_0(0))$ curve, and right to left along the plots of $M_{\text{ADM}}(R_{99})$ and $M_{\text{ADM}}(R_{95})$).

It is possible to show from perturbation analyses, as well as from evolution of the full equations of motion [110, 116, 161, 128] that stars satisfying $\phi_0(0) < \phi_0^*$ are dynamically stable against radial perturbations, while those defined by $\phi_0(0) \geq \phi_0^*$ are dynamically unstable. The sequences of configurations defined by $\phi_0(0) < \phi_0^*$ and $\phi_0(0) \geq \phi_0^*$ are thus known as the *stable* and *unstable* branches, respectively. Moreover, with reference to the plot of $M(\phi_0(0))$ shown in the left panel of Fig. 3.2, it can be shown that every extremum in the plot corresponds to an additional perturbative mode becoming unstable. We emphasize that all of the results concerning boson star evolutions that are reported in this thesis (Chap. 5) used stars from the stable branch.

The existence of a maximum mass for our one-parameter family of boson stars, and the fact that there is a change of stability at $\phi_0(0) = \phi_0^*$, is completely analogous to the Chandrasekhar limits for spherically symmetric white dwarfs and neutron stars [162]. In both of those instances, above some mass limit there is no stable static configuration and the star is prone to gravitational collapse (to a neutron star in the case of a white dwarf, and, it is widely believed, to a black hole in the case of a neutron star). We note, however, that for boson stars there is no degeneracy pressure, as there is for the fermionic white dwarfs and neutron stars. In the bosonic case, the effective pressure support that counteracts gravity can be viewed heuristically as coming from the uncertainty principle if the system is studied semiclassically, or from the dispersive nature of the Klein-Gordon wave equation when the system is studied classically, as it is in our current work.

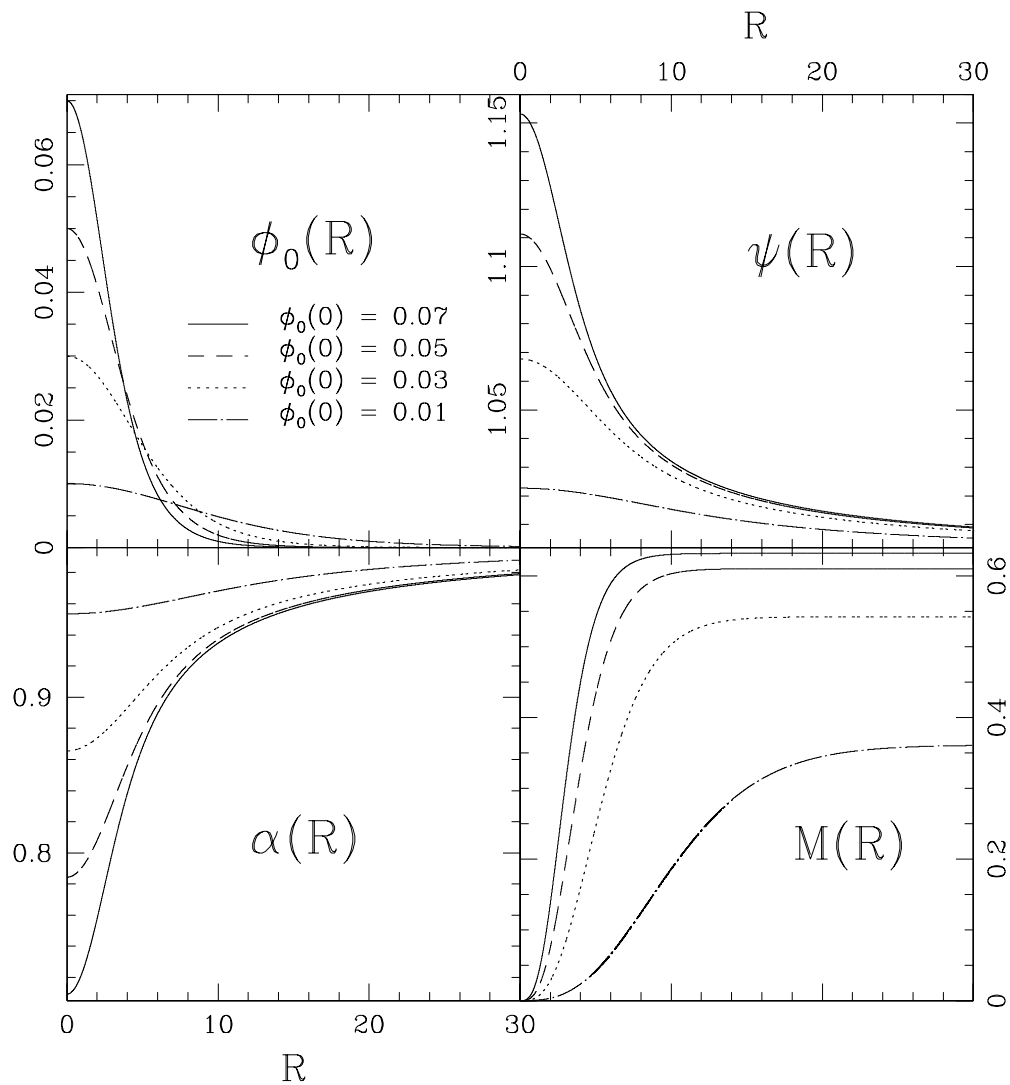


Figure 3.1: Typical Boson Star Solutions. These plots show the values of $\phi_0(R)$, $\psi(R)$, $\alpha(R)$ and $M(R)$ as a function of the areal coordinate, R , for boson stars defined by $\phi_0(0) = 0.01, 0.03, 0.05$ and 0.07 . As discussed in more detail in the text, all of these configurations are dynamically stable against radial perturbations. Note that as $\phi_0(0)$ increases the stars become smaller (more compact).

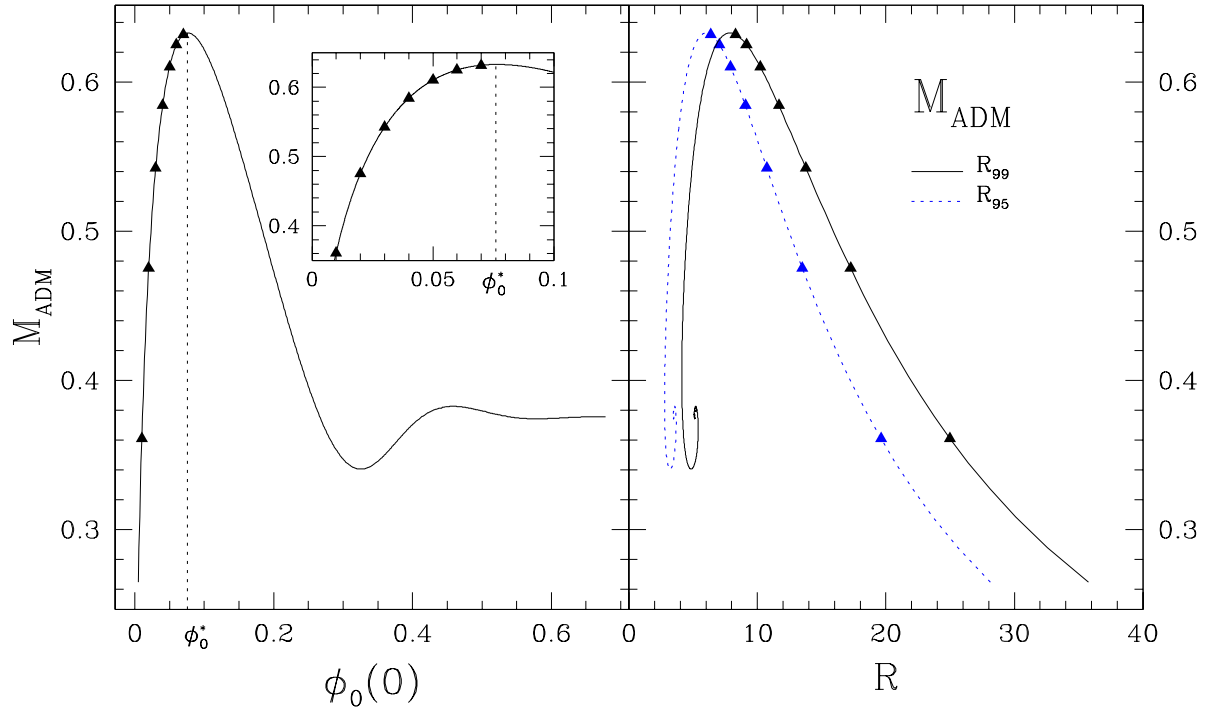


Figure 3.2: This figure shows the ADM mass, M_{ADM} , as a function of the central scalar field value, $\phi_0(0)$ (left panel), as well as a function of two estimates, R_{99} , and R_{95} , of the stellar radius (right panel). R_{99} and R_{95} are defined to be the radii for which $M(R) = 0.99 M_{\text{ADM}}$ and $M(R) = 0.95 M_{\text{ADM}}$, respectively. The $M_{\text{ADM}}(\phi_0(0))$ curve has an absolute maximum at $\phi_0(0) = \phi_0^* \approx 0.08$ with $M_{\text{max}} = M(\phi_0^*) \approx 0.633$. As discussed in the text, the value ϕ_0^* (shown as a dashed line in the plot) signals a change in dynamical stability of the solutions: for $\phi_0(0) < \phi_0^*$ the stars are stable to radial perturbations, while those with $\phi_0(0) \geq \phi_0^*$ are unstable. The solid triangles label specific models computed with $\phi_0(0) = 0.01, 0.02, 0.03, 0.04, 0.05, 0.06$, and 0.07 —these are all on the stable branch. The inset in the left plot highlights the form of $M_{\text{ADM}}(\phi_0(0))$ for $0.01 \leq \phi_0(0) \leq 0.10$. Note that for the curves in the right plot, the triangles corresponding to $\phi_0(0) = 0.01, 0.02, \dots, 0.07$ appear right-to-left along the plots—i.e. as noted in the caption of the previous figure, as well as in the text, the stellar radius *decreases* with increasing $\phi_0(0)$.

3.6 Boson Stars: Representation in Cartesian Coordinates

In Sec. 3.4 we summarized the procedure we use to compute boson star data in areal coordinates, as well as the subsequent transformation of the data to isotropic coordinates. As we noted there, the coordinate transformation is needed to provide initial conditions in a form that is compatible with our 3D code, which assumes conformal flatness of the 3-metric. In addition, since the code *is* 3D, and based in Cartesian coordinates, (x, y, z) , we must also perform some interpolation of the field values.

Let us first consider the case where we wish to evolve a single star with some given central field value, $\phi_0(0)$. We start by solving the static polar-areal ODE system, which yields the functions $\alpha(R_J)$, $a(R_J)$ and $\phi_0(R_J)$, as well as the eigenvalue for the solution, $\omega = \omega(\phi_0(0))$. Here R_J , $J = 1, 2, \dots, N_J$ are the N_J discrete values of the areal radius at which we choose to compute (store) the numerical solution of the ODEs. We then determine the coordinate transformation to isotropic coordinates, $r_J = r(R_J)$, by solving (3.78) with the boundary conditions (3.79). Once this is done, values for the conformal factor, $\psi(r_J)$, are determined from (3.81) and 3.82). Next, since the time coordinates are identical in the polar-areal and maximal-isotropic systems, the lapse function transforms as a scalar, and we have $\alpha(r_J) = \alpha(r(R_J))$. Finally, the matter field, ϕ_0 , *is* a scalar, so we also have $\phi_0(r_J) = \phi_0(r(R_J))$.

Having computed $\psi(r)$, $a(r)$ and $\alpha(r)$, it is a simple matter to interpolate these functions onto the discrete domain which our 3D code uses to solve the PDEs derived in Chap. 2. Without going into the full details (these are given in the next chapter), the finite difference grid points have coordinates (x_i, y_j, z_k) where $i = 1, \dots, n_x$, $j = 1, \dots, n_y$, and $k = 1, \dots, n_k$, so that the grid has dimensions $n_x \times n_y \times n_z$. We choose a point having coordinates (x_0, y_0, z_0) that lies within the computational domain and at which we will “centre” the star. Then for all i, j and k we define

$$r_{ijk} = \sqrt{(x_i - x_0)^2 + (y_j - y_0)^2 + (z_k - z_0)^2} \quad (3.84)$$

and then set

$$\alpha(x_i, y_j, z_k) = \alpha(r_{ijk}), \quad (3.85)$$

$$\psi(x_i, y_j, z_k) = \psi(r_{ijk}), \quad (3.86)$$

$$\phi_0(x_i, y_j, z_k) = \phi_0(r_{ijk}). \quad (3.87)$$

Here, the values $\alpha(r_{ijk})$, $\psi(r_{ijk})$ and $\phi_0(r_{ijk})$ are computed using quadratic Lagrange polynomial interpolation [160] in the values $\alpha(r_J)$, $\psi(r_J)$ and $\phi_0(r_J)$, respectively.

Next, using the values $\phi_0(x_i, y_j, z_k)$, and the scalar ansatz (3.41) evaluated at $t = 0$, we initialize the real and imaginary components of the scalar field by

$$\phi_1(x_i, y_j, z_k) = \phi_0(x_i, y_j, z_k), \quad (3.88)$$

$$\phi_2(x_i, y_j, z_k) = 0. \quad (3.89)$$

In addition, the field conjugate momenta can be computed from (2.173), again, by using the static ansatz (3.41) for the complex scalar field, and the fact that the shift vector components vanish in the static case. In terms of values defined above, and calculated using interpolation of the ODE solution, we have

$$\Pi_1(x_i, y_j, z_k) = \omega \frac{\psi(x_i, y_j, z_k)^6}{\alpha(x_i, y_j, z_k)} \phi_0(x_i, y_j, z_k) \sin(\omega t), \quad (3.90)$$

$$\Pi_2(x_i, y_j, z_k) = -\omega \frac{\psi(x_i, y_j, z_k)^6}{\alpha(x_i, y_j, z_k)} \phi_0(x_i, y_j, z_k) \cos(\omega t). \quad (3.91)$$

Since $t = 0$, the above expressions further reduce to:

$$\Pi_1(x_i, y_j, z_k) = 0, \quad (3.92)$$

$$\Pi_2(x_i, y_j, z_k) = -\omega \frac{\psi(x_i, y_j, z_k)^6}{\alpha(x_i, y_j, z_k)} \phi_0(x_i, y_j, z_k). \quad (3.93)$$

For the case of a binary system, the above process is performed for both stars, and we then

superimpose the solutions as follows:

$$\alpha(x_i, y_j, z_k) = \alpha^{(1)}(x_i, y_j, z_k) + \alpha^{(2)}(x_i, y_j, z_k) - 1, \quad (3.94)$$

$$\psi(x_i, y, z_k) = \psi^{(1)}(x_i, y_j, z_k) + \psi^{(2)}(x_i, y_j, z_k) - 1, \quad (3.95)$$

$$\phi_1(x_i, y_j, z_k) = \phi_1^{(1)}(x_i, y_j, z_k) + \phi_1^{(2)}(x_i, y_j, z_k), \quad (3.96)$$

$$\phi_2(x_i, y_j, z_k) = \phi_2^{(1)}(x_i, y_j, z_k) + \phi_2^{(2)}(x_i, y_j, z_k), \quad (3.97)$$

$$\Pi_1(x_i, y_j, z_k) = \Pi_1^{(1)}(x_i, y_j, z_k) + \Pi_1^{(2)}(x_i, y_j, z_k), \quad (3.98)$$

$$\Pi_2(x_i, y_j, z_k) = \Pi_2^{(1)}(x_i, y_j, z_k) + \Pi_2^{(2)}(x_i, y_j, z_k). \quad (3.99)$$

Here the superscripts (1) and (2) refer to the interpolated solutions for the first and second star, respectively. In practice, when we set up data for a binary we try to ensure that the two stars do not overlap significantly. In this case, we can expect that superposition of the individual lapse and conformal functions, as defined above, provides values of $\alpha(x_i, y_j, z_k)$ and $\psi(x_i, y, z_k)$ which satisfy the slicing and Hamiltonian constraint equations to some degree of precision. However, apart from considerations of the extent that the initial setup *does* represent two distinct and separated stars, the issues of whether or not there is substantial overlap of $\phi_0^{(1)}$ and $\phi_0^{(2)}$, and whether or not superposition of the metric functions approximately holds, are not very important. In particular, the scalar field variables are, in principle, freely specifiable, so an overlapping configuration provides initial data that is no less mathematically valid than one in which there is negligible overlap. Furthermore, once we have fixed the scalar field values at $t = 0$, we *always* solve the elliptic equations PDEs for the metric functions to determine their initial values. Ultimately then, the values obtained using the superposition formulae (3.94) and (3.95) are only used to provide initial estimates for the iterative multigrid method that we use for the elliptic system (see Chap. 4).

3.7 Boson Stars: Applying Approximate Lorentz Boosts

The procedure described in the previous section allows us to initialize our 3D dynamical code with data representing one or two stars that are initially at rest in the (x, y, z) coordinate system. In order to simulate scenarios such as a binary system in which the stars are in mutual orbit about one another, or, in general, where there is to be any motion of one or both stars at $t = 0$, the scalar

field initial conditions must be adjusted to provide the stars with initial velocities.

In this section we describe how this is accomplished through the use of Lorentz transformations, applied to both the scalar field and metric variables. We emphasize at this point that the algorithm that we describe below does *not*, in general, provide initial conditions (even for a single star) that correspond to a pure boost. This is a result of an incompatibility between our straightforward approach to applying the Lorentz transformation and the requirement that our spatial metric be conformally flat. However, at least at this juncture in our research, we do not consider this to be a major shortcoming. What is most important is that we be able to produce initial conditions such that the stars propagate through the solution domain, retaining their overall structure, at least approximately, as they move. As will be seen in Chap. 5, our current method certainly succeeds in this respect.

In the calculations that we report in this thesis (again, in Chap. 5), we have restricted attention to configurations in which one or two boson stars are boosted in the x direction. We therefore adopt an abbreviated notation which suppresses the functional dependence of the unknowns on y and z . Thus, for example, $\phi(t, x)$ is shorthand for $\phi(t, x, y, z)$. We let (t, x) be the coordinates in the “lab” frame (i.e. the coordinate system in which we ultimately solve the PDEs), and denote the coordinates of the rest frame of any boosted star by (t', x') .

We now focus attention on the case of a single boson star. Then if the star is boosted with speed $v > 0$ in the positive x direction, and keeping in mind that we have adopted units in which the speed of light, c , is unity, the Lorentz transformation is

$$x = \gamma(x' + vt'), \quad (3.100)$$

$$t = \gamma(t' + vx'), \quad (3.101)$$

where γ is the usual Lorentz factor defined by $\gamma \equiv (1 - v^2)^{-1/2}$. The inverse Lorentz transformation is:

$$x' = \gamma(x - vt), \quad (3.102)$$

$$t' = \gamma(t - vx). \quad (3.103)$$

Now, from the static ansatz (3.41) we have the following equations for the real and imaginary components of the scalar field, ϕ_1 and ϕ_2 , respectively:

$$\phi'_1(t', x') = \phi_0(x') \cos(\omega t'), \quad (3.104)$$

$$\phi'_2(t', x') = -\phi_0(x') \sin(\omega t'). \quad (3.105)$$

The time derivatives of these quantities, which we denote by $\bar{\Pi}'_1(t', x')$ and $\bar{\Pi}'_2(t', x')$, respectively, are then given by

$$\bar{\Pi}'_1(t', x') \equiv \frac{\partial \phi'_1(t', x')}{\partial t'} = -\omega \phi_0(x') \sin(\omega t'), \quad (3.106)$$

$$\bar{\Pi}'_2(t', x') \equiv \frac{\partial \phi'_2(t', x')}{\partial t'} = -\omega \phi_0(x') \cos(\omega t'). \quad (3.107)$$

As a scalar field is invariant under any coordinate transformation, the values of ϕ_1 and ϕ_2 are determined by

$$\phi_1(t, x) \equiv \phi'_1(t', x') = \phi_0(\gamma(x - vt)) \cos[\omega\gamma(t - vx)], \quad (3.108)$$

$$\phi_2(t, x) \equiv \phi'_2(t', x') = -\phi_0(\gamma(x - vt)) \sin[\omega\gamma(t - vx)], \quad (3.109)$$

where we have used the inverse Lorentz transformations given by (3.102) and (3.103). At the initial time, $t = 0$, we thus have

$$\phi_1(0, x) = \phi_0(\gamma x) \cos(\omega\gamma vx), \quad (3.110)$$

$$\phi_2(0, x) = \phi_0(\gamma x) \sin(\omega\gamma vx). \quad (3.111)$$

We remind the reader that we are suppressing the y and z dependence of the unknowns, so that $\phi_0(\gamma x) \equiv \phi_0(\gamma x, y, z)$. This value can be computed from the ODE solution, $\phi_0(r(R_J))$, via interpolation to $r_{\gamma x} \equiv \sqrt{\gamma^2(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}$, where (x_0, y_0, z_0) are the lab frame coordinates of the point at which the star is centred.

The time derivatives, $\bar{\Pi}'_1$ and $\bar{\Pi}'_2$, are also scalar fields and thus invariant under coordinate transformations as well. However, in order to write down expressions for them in the lab frame, we need to go through the Lorentz transformation in more detail. We first note that the chain rule can be used to express the time derivatives in the star frame's in terms of derivatives in the lab frame:

$$\bar{\Pi}_1(t, x) \equiv \bar{\Pi}'_1(t', x') = \frac{\partial \phi'_1}{\partial t'}(t', x') = \frac{\partial \phi_1}{\partial t'}(t, x) = \frac{\partial \phi_1}{\partial x}(t, x) \frac{\partial x}{\partial t'} + \frac{\partial \phi_1}{\partial t}(t, x) \frac{\partial t}{\partial t'}, \quad (3.112)$$

$$\bar{\Pi}_2(t, x) \equiv \bar{\Pi}'_2(t', x') = \frac{\partial \phi'_2}{\partial t'}(t', x') = \frac{\partial \phi_2}{\partial t'}(t, x) = \frac{\partial \phi_2}{\partial x}(t, x) \frac{\partial x}{\partial t'} + \frac{\partial \phi_2}{\partial t}(t, x) \frac{\partial t}{\partial t'}. \quad (3.113)$$

The spatial derivatives appearing in the above expressions above can be calculated using equations (3.108) and (3.109):

$$\begin{aligned} \frac{\partial \phi_1}{\partial x}(t, x) &= \frac{\partial \phi_0}{\partial x}(\gamma(x - vt)) \cos[\omega\gamma(t - vx)] \\ &\quad + \omega\gamma v \phi_0(\gamma(x - vt)) \sin[\omega\gamma(t - vx)], \end{aligned} \quad (3.114)$$

$$\begin{aligned} \frac{\partial \phi_2}{\partial x}(t, x) &= -\frac{\partial \phi_0}{\partial x}(\gamma(x - vt)) \sin[\omega\gamma(t - vx)] \\ &\quad + \omega\gamma v \phi_0(\gamma(x - vt)) \cos[\omega\gamma(t - vx)]. \end{aligned} \quad (3.115)$$

Additionally, we have

$$\frac{\partial \phi_0}{\partial t}(\gamma(x - vt)) = \frac{\partial \phi_0}{\partial x'}(x') \frac{\partial x'}{\partial t} = -\gamma v \left. \frac{\partial \phi_0}{\partial x'}(x') \right|_{x'=\gamma(x-vt)}, \quad (3.116)$$

so that

$$\begin{aligned} \frac{\partial \phi_1}{\partial t}(t, x) &= \frac{\partial \phi_0}{\partial t}(\gamma(x - vt)) \cos[\omega\gamma(t - vx)] - \omega\gamma \phi_0(\gamma(x - vt)) \sin[\omega\gamma(t - vx)] \\ &= -\gamma v \left. \frac{\partial \phi_0}{\partial x'}(x') \right|_{x'=\gamma(x-vt)} \cos[\omega\gamma(t - vx)] \\ &\quad - \omega\gamma \phi_0(\gamma(x - vt)) \sin[\omega\gamma(t - vx)], \end{aligned} \quad (3.117)$$

and

$$\begin{aligned}
\frac{\partial \phi_2}{\partial t}(t, x) &= -\frac{\partial \phi_0}{\partial t}(\gamma(x - vt)) \sin[\omega\gamma(t - vx)] - \omega\gamma\phi_0(\gamma(x - vt)) \cos[\omega\gamma(t - vx)] \\
&= \gamma v \frac{\partial \phi_0}{\partial x'}(x') \Big|_{x'=\gamma(x-vt)} \sin[\omega\gamma(t - vx)] \\
&\quad - \omega\gamma\phi_0(\gamma(x - vt)) \cos[\omega\gamma(t - vx)].
\end{aligned} \tag{3.118}$$

Using (3.114), (3.115), (3.117) and (3.118) in (3.112) and (3.113) yields

$$\begin{aligned}
\bar{\Pi}_1(t, x) &= \gamma v \left[\frac{\partial \phi_0}{\partial x}(\gamma(x - vt)) - \gamma \frac{\partial \phi_0}{\partial x'}(x') \right] \cos[\omega\gamma(t - vx)] \\
&\quad - \omega\phi_0(\gamma(x - vt)) \sin[\omega\gamma(t - vx)],
\end{aligned} \tag{3.119}$$

$$\begin{aligned}
\bar{\Pi}_2(t, x) &= \gamma v \left[\gamma \frac{\partial \phi_0}{\partial x'}(x') - \frac{\partial \phi_0}{\partial x}(\gamma(x - vt)) \right] \sin[\omega\gamma(t - vx)] \\
&\quad - \omega\phi_0(\gamma(x - vt)) \cos[\omega\gamma(t - vx)],
\end{aligned} \tag{3.120}$$

where the derivatives with respect to x' are to be evaluated at $x' = \gamma(x - vt)$. At $t = 0$, the solution in the lab frame is then

$$\bar{\Pi}_1(0, x) = \gamma v \left[\frac{\partial \phi_0}{\partial x}(\gamma x) - \gamma \frac{\partial \phi_0}{\partial x'}(x') \Big|_{x'=\gamma x} \right] \cos(\omega\gamma vx) + \omega\phi_0(\gamma x) \sin(\omega\gamma vx), \tag{3.121}$$

$$\bar{\Pi}_2(0, x) = \gamma v \left[\frac{\partial \phi_0}{\partial x}(\gamma x) - \gamma \frac{\partial \phi_0}{\partial x'}(x') \Big|_{x'=\gamma x} \right] \sin(\omega\gamma vx) - \omega\phi_0(\gamma x) \cos(\omega\gamma vx). \tag{3.122}$$

In order to completely determine the initial data—if only to provide good initial estimates for the elliptic solver at $t = 0$ —we also need to consider the effect of the Lorentz boost on the metric components in the lab frame. In the rest frame of the star we have

$$g'_{\mu\nu}(t', x') = \begin{pmatrix} -\alpha'^2(t', x') & 0 & 0 & 0 \\ 0 & \psi'^4(t', x') & 0 & 0 \\ 0 & 0 & \psi'^4(t', x') & 0 \\ 0 & 0 & 0 & \psi'^4(t', x') \end{pmatrix}. \tag{3.123}$$

We now adopt another notation, in which we use a tilde on lab-frame metric components that have had Lorentz transformations applied to them. This distinguishes them from the *actual* lab-frame metric variables that appear in the our CFA model, in view of our above observation that

the simple-minded boost procedure we use is incompatible with conformal flatness. The metric components in the lab frame are then given by

$$\tilde{g}_{\lambda\delta}(t, x) = \frac{\partial x'^{\mu}}{\partial x^{\lambda}} \frac{\partial x'^{\nu}}{\partial x^{\delta}} g'_{\mu\nu}(t', x'), \quad (3.124)$$

where, from the general 3 + 1 form we also have

$$\tilde{g}_{\lambda\delta}(t, x) = \begin{pmatrix} -\tilde{\alpha}^2(t, x) + \tilde{\gamma}^{ij} \tilde{\beta}_i \tilde{\beta}_j & \tilde{\beta}_k(t, x) \\ \tilde{\beta}_l(t, x) & \tilde{\gamma}_{lk}(t, x) \end{pmatrix}. \quad (3.125)$$

The Lorentz boost is given by

$$\Lambda^{\mu}_{\lambda} = \frac{\partial x'^{\mu}}{\partial x^{\lambda}} = \begin{pmatrix} \gamma & -\gamma v & 0 & 0 \\ -\gamma v & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (3.126)$$

Thus, for example, the \tilde{g}_{0x} component of the lab frame metric is given by

$$\tilde{g}_{0x} \equiv \tilde{\beta}_x(t, x) = \Lambda^{\mu}_0 \Lambda^{\nu}_x g'_{\mu\nu} = (\Lambda^0_0 g'_{0\nu} + \Lambda^x_0 g'_{x\nu}) \Lambda^{\nu}_x, \quad (3.127)$$

which yields

$$\tilde{\beta}_x(t, x) = \gamma^2 v [\alpha'^2(t', x') - \psi'^4(t', x')]. \quad (3.128)$$

Similarly, it is possible to show that the spatial metric in the lab frame is related to the metric in the star's rest frame by:

$$\tilde{\gamma}_{ij} = \begin{pmatrix} \gamma^2 [\psi'^4(t', x') - v^2 \alpha'^2(t', x')] & 0 & 0 \\ 0 & \psi'^4(t', x') & 0 \\ 0 & 0 & \psi'^4(t', x') \end{pmatrix}, \quad (3.129)$$

As promised, this lab frame 3-metric is *not* conformally flat. However, as the tilde notation emphasises, it is also *not* the spatial metric with which the 3D code is eventually initialized, and we emphasize that there is no inconsistency being introduced by the current method for determining initial data. In this regard, the metric (3.129) is perhaps best viewed as an intermediary which aids

in the computation of other quantities (such as the scalar field momenta, Π_1 and Π_2) to produce data *approximately* describing a boosted star. A similar comment applies to the other lab frame geometric quantities—namely the lapse, $\tilde{\alpha}$, and the non-vanishing shift component $\tilde{\beta}^x$ —that are defined in this stage of the algorithm. They too should be viewed as “provisional” values that are used primarily to set the scalar field quantities at $t = 0$ so that something close to a boosted boson star results. On the other hand, the true initial values for the geometric quantities are *always* determined—once the scalar field variables are given—by solving the governing set of elliptic PDEs (2.181)–(2.185), in which the scalar field quantities play the role of sources.

Keeping this in mind then, we continue by using (3.129) and (3.128) to find

$$\tilde{\beta}^x(t, x) = \frac{v [\alpha'^2(t', x') - \psi'^4(t', x')]}{[\psi'^4(t', x') - v^2 \alpha'^2(t', x')]} \quad (3.130)$$

However, since the metric components in the star frame are all time independent, the above equation becomes

$$\tilde{\beta}^x(t, x) = \frac{v [\alpha'^2(x') - \psi'^4(x')]}{[\psi'^4(x') - v^2 \alpha'^2(x')]} \quad (3.131)$$

and restricting to the initial time $t = 0$, we have

$$\tilde{\beta}^x(0, x) = \frac{v [\alpha'^2(\gamma x) - \psi'^4(\gamma x)]}{[\psi'^4(\gamma x) - v^2 \alpha'^2(\gamma x)]} \quad (3.132)$$

Using a derivation analogous to that just used to determine $\tilde{\beta}^x(0, x)$, we can show that the initial values of the lapse in the lab frame are given by

$$\tilde{\alpha}(0, x) = \frac{\alpha'(\gamma x) \psi'^2(\gamma x)}{\gamma \sqrt{\psi'^4(\gamma x) - v^2 \alpha'^2(\gamma x)}}, \quad (3.133)$$

while the conformal factor is simply

$$\tilde{\psi}(0, x) = \psi'(\gamma x). \quad (3.134)$$

Finally, using (3.110), (3.111), (3.121), (3.122), (3.132), (3.133) and (3.134), the initial values for the scalar field conjugate momenta are given by

$$\Pi_1(0, x) = \frac{\tilde{\psi}^6(0, x)}{\tilde{\alpha}^2(0, x)} \left[\bar{\Pi}_1(0, x) - \tilde{\beta}^x(0, x) \frac{\partial \phi_1}{\partial x}(0, x) \right], \quad (3.135)$$

$$\Pi_2(0, x) = \frac{\tilde{\psi}^6(0, x)}{\tilde{\alpha}^2(0, x)} \left[\bar{\Pi}_2(0, x) - \tilde{\beta}^x(0, x) \frac{\partial \phi_2}{\partial x}(0, x) \right]. \quad (3.136)$$

To summarize, equations (3.110), (3.111), (3.135) and (3.136) provide initial values for the scalar field components and their conjugate momenta in the lab reference frame. On the other hand, equations (3.132), (3.133) and (3.134) are used to provide initial *estimates* for the values of β^x , α and ψ that are ultimately computed by solving the elliptic equations at $t = 0$. However, they are also used to construct the $t = 0$ values of the scalar field conjugate momenta as given by equations (3.135) and (3.136).

As was the case in the previous section, for configurations involving two stars, the above calculations are performed for each star individually. We then set values for the fields using the following superposition formulae:

$$\phi_1(0, x) = \phi_1^{(1)}(0, x) + \phi_1^{(2)}(0, x), \quad (3.137)$$

$$\phi_2(0, x) = \phi_2^{(1)}(0, x) + \phi_2^{(2)}(0, x), \quad (3.138)$$

$$\Pi_1(0, x) = \Pi_1^{(1)}(0, x) + \Pi_1^{(2)}(0, x), \quad (3.139)$$

$$\Pi_2(0, x) = \Pi_2^{(1)}(0, x) + \Pi_2^{(2)}(0, x), \quad (3.140)$$

$$\tilde{\beta}^x(0, x) = \tilde{\beta}^{x(1)}(0, x) + \tilde{\beta}^{x(2)}(0, x), \quad (3.141)$$

$$\tilde{\alpha}(0, x) = \tilde{\alpha}^{(1)}(0, x) + \tilde{\alpha}^{(2)}(0, x) - 1, \quad (3.142)$$

$$\tilde{\psi}(0, x) = \tilde{\psi}^{(1)}(0, x) + \tilde{\psi}^{(2)}(0, x) - 1. \quad (3.143)$$

Here the superscripts (1) and (2) again refer to the solutions computed for the two individual stars. For the scalar field variables, ϕ_1 , ϕ_2 , Π_1 and Π_2 , (3.137)–(3.140) are used to set the actual initial values. Again, however, (3.141)–(3.143) are used only to provide initial estimates for the multigrid elliptic solver which computes the actual $t = 0$ values of β^x , α and ψ .

We end this section by noting that it is probably possible to refine the algorithm described here—by iterating the steps of computing the conjugate momenta using expressions such as (3.135)

and (3.136), and of solving the elliptic equations for the metric variables—in order to determine initial data that more closely approximates a purely boosted boson star. This however, is another matter that will require further investigation: as stated earlier, for the purposes of the computations described in Chap. 5, the procedure detailed here appears adequate.

Chapter 4

Numerical Techniques

This chapter discusses the key numerical techniques that were used to find approximate solutions of the partial differential equations (PDEs) that comprise our model. Furthermore, the methods that were employed to assess the basic correctness of the numerical implementation and the accuracy of the generated solutions are presented. Although some issues are covered here in more detail than others, for completeness we have tried to describe all of the main numerical approaches and algorithms that factored into the construction of our code. In this regard, however, we note that we have relegated some of the more basic and well-known concepts and techniques to Appendices B and D. Portions of this chapter closely follow the presentation in [163] and the interested reader is referred to those notes (as well as references contained therein) for examples, as well as more detailed discussions of some of the approaches we have used. In addition, for the uninitiated, the venerable text due to Burden and Faires [160] provides an excellent general introduction to the field of numerical analysis.

A crucial step in the construction of numerical solutions of PDEs is the choice of an appropriate discretization approach. The one adopted here is finite difference approximation (FDA), and, fundamentally, involves replacement of the differential operators appearing in the PDEs with suitable finite difference operators. While the differential operators are applied to functions defined on a particular subdomain, Ω , of the continuum—which in our case is $\mathbb{R} \times \mathbb{R}^3$ —in practice, the discrete operators are applied to functions defined on a discrete set of points, Ω^h , often referred to as a grid, or mesh.²³ Sec. 4.1 introduces the main concepts we use in the finite difference discretization of our PDEs. Based on the FDA concepts and assumptions discussed in Sec. 4.1, Sec. 4.2 then continues with a description of the analysis tools that were used to evaluate the fidelity of our numerical solutions, as presented in detail in Chap. 5.

As discussed previously, our model consists of a coupled system of 4 hyperbolic (first order in time) and 5 elliptic (second order in space) PDEs. Since the equations belong to two different

²³We note, however, that it can be useful to view the discrete operators as acting on functions defined on the continuum, particularly when considering error analysis of the type discussed, for example, in 4.2.2.

classes, it is natural to expect that different techniques will be needed to efficiently obtain the full numerical solutions. In brief, we treat the hyperbolic equations using a second order (in the mesh spacing, h) Crank-Nicholson approximation that is solved iteratively, while the elliptic equations, also discretized to $O(h^2)$, are solved using a multigrid technique. Here we note that, as illustrated by the pseudo-code of the overall flow of our code in Fig. 5.1 (Sec. 5.1), at each stage of the Crank-Nicholson iteration that performs the basic time-step-advance of the hyperbolics, we re-solve the elliptics. A review of basic relaxation techniques, including the point-wise Newton-Gauss-Seidel method used in the Crank-Nicholson iteration is given App. D, while the multigrid method is discussed in some detail in Sec. 4.3.

4.1 Discretization of Partial Differential Equations: Finite Difference Approximation

Two fundamental assumptions underlying a successful finite difference approximation (FDA) of a set of PDEs are:

1. Given suitable boundary and/or initial conditions, there is a unique solution of the system of PDEs to be discretized.
2. This unique solution is smooth.

As argued in Chap. 2, any solution of the coupled hyperbolic-elliptic system of equations constituting the initial-boundary value problem to be solved in this thesis is expected to be smooth provided the prescribed initial data is smooth. Also, although no general proof exists, there is no reason *a priori* to expect solutions of our system (again given appropriate initial and boundary conditions) not to be unique. Therefore the choice of finite difference approximation as the discretization approach becomes a natural one for the problem at hand.

The first step in the discretization of PDEs using finite differencing involves the definition of a grid (or mesh) over the solution domain $\Omega \subset \mathbb{R}^n$. (We note that for the time being we will restrict our attention to discretization of the *spatial* domain, so that Ω is a spatial volume: we will return to the issue of temporal discretization below.) Loosely speaking, the grid is defined as a discrete set of points from the continuum domain that satisfy some criteria. These criteria can be as diverse as the physical problems that are governed by systems of PDEs. In general, though, the choice of

grid points is governed by the geometry of the domain, (and hence the geometries of the domain boundaries), as well as the coordinate system in which the PDEs are expressed. For domains with very complex structure, for example, it is common to use so-called unstructured grids that adapt themselves to the geometry. Grids used for simulations of car crashes or airplane aerodynamics are frequently of this kind. Considering another example, a physical problem with approximate spherical symmetry will often be formulated in a coordinate system adapted to the symmetry, and it will therefore be natural to use a curvilinear grid which reflects that symmetry. For the case of the differential equations defining our model, the coordinate system in which the equations are written is the familiar Cartesian (or rectangular) one. Thus it is only logical that the grid adopted in the discretization also be Cartesian. In addition, the simplicity of our solution domain, combined with the fact that the objects that we will be studying will 1) tend to have comparable extents in each of the three coordinate directions, and 2) tend to move through the computational domain, suggests the use of the simplest grid possible: a uniform Cartesian grid. For our 3D spatial domain $\Omega \subset \mathbb{R}^3$ this consists of an ordered set of points $(x_i, y_j, z_k) \equiv \Omega^h \subset \Omega$ such that the separation between adjacent points in any of the coordinate directions is some constant, h , known as the grid (or mesh) spacing, or discretization scale. Specifically we have

$$x_{i+1} - x_i = h, \quad y_{j+1} - y_j = h, \quad z_{k+1} - z_k = h. \quad (4.1)$$

For any FDA the various grid spacings that may appear in the definition of the mesh are the fundamental control parameters for the approximation: in particular one hopes to recover the continuum solution in the limit that all of the grid spacings tend to 0. We emphasize that in this thesis we will exclusively use discretizations (including that of the time variable) that are characterized by the *single* discretization scale, h .

The specific grid points comprising our uniform Cartesian mesh are readily defined from the coordinate ranges of the continuum domain. Thus, for

$$x_{\min} \leq x \leq x_{\max}, \quad (4.2)$$

$$y_{\min} \leq y \leq y_{\max}, \quad (4.3)$$

$$z_{\min} \leq z \leq z_{\max}, \quad (4.4)$$

the grid points are

$$x_i = x_{\min} + (i - 1)h, \quad i = 1 \dots n_x, \quad (4.5)$$

$$y_j = y_{\min} + (j - 1)h, \quad j = 1 \dots n_y, \quad (4.6)$$

$$z_k = z_{\min} + (k - 1)h, \quad k = 1 \dots n_z, \quad (4.7)$$

such that

$$x_1 = x_{\min} \quad \text{and} \quad x_{n_x} = x_{\max}, \quad (4.8)$$

$$y_1 = y_{\min} \quad \text{and} \quad y_{n_y} = y_{\max}, \quad (4.9)$$

$$z_1 = z_{\min} \quad \text{and} \quad z_{n_z} = z_{\max}. \quad (4.10)$$

There are therefore $N_g = n_x \times n_y \times n_z$ points in the mesh, and we further note that there is an implicit assumption that each of the ranges $x_{\max} - x_{\min}$, $y_{\max} - y_{\min}$ and $z_{\max} - z_{\min}$ is evenly divisible by h .

Once the grid Ω^h has been defined, we can introduce grid functions $u^h : \Omega^h \rightarrow \mathbb{R}$ on it. For the 3D Cartesian grid defined above, the grid function u^h is simply equivalent to the N_g values u_{ijk} defined on the grid points (x_i, y_j, z_k) —i.e. u_{ijk} denotes a discrete approximation to the continuum value $u(x_i, y_j, z_k)$. We also observe that the way that we have defined our discrete unknowns is often called a *vertex centred* approach, since the unknowns are defined at the grid points *per se*, instead of, for example, at the centre of a cubic cell composed of 8 neighbouring grid points. Discretizations based on this latter approach, and which *have* frequently been used in numerical relativity—especially in studies involving sets of conservation laws such as hydrodynamics—are known as *cell centred*. Finally, Fig. 4.1 illustrates a sample 3D rectangular grid of the type used throughout this thesis.

As already mentioned in the introductory section of this chapter, the derivative operators appearing in the system of PDEs governing our model are approximated by finite difference operators. It is conventional to characterize any such operator by its *difference stencil* or *difference star*. This involves specification of the set of neighbouring grid points that appear in the definition of the difference operator. For example, if a real function of one variable, $u(x)$, is restricted to its values, u_i , defined on a uniform 1D grid, $\Omega^h = \{x_i \mid x_{i+1} - x_i = h\}$, then the second derivative of u at

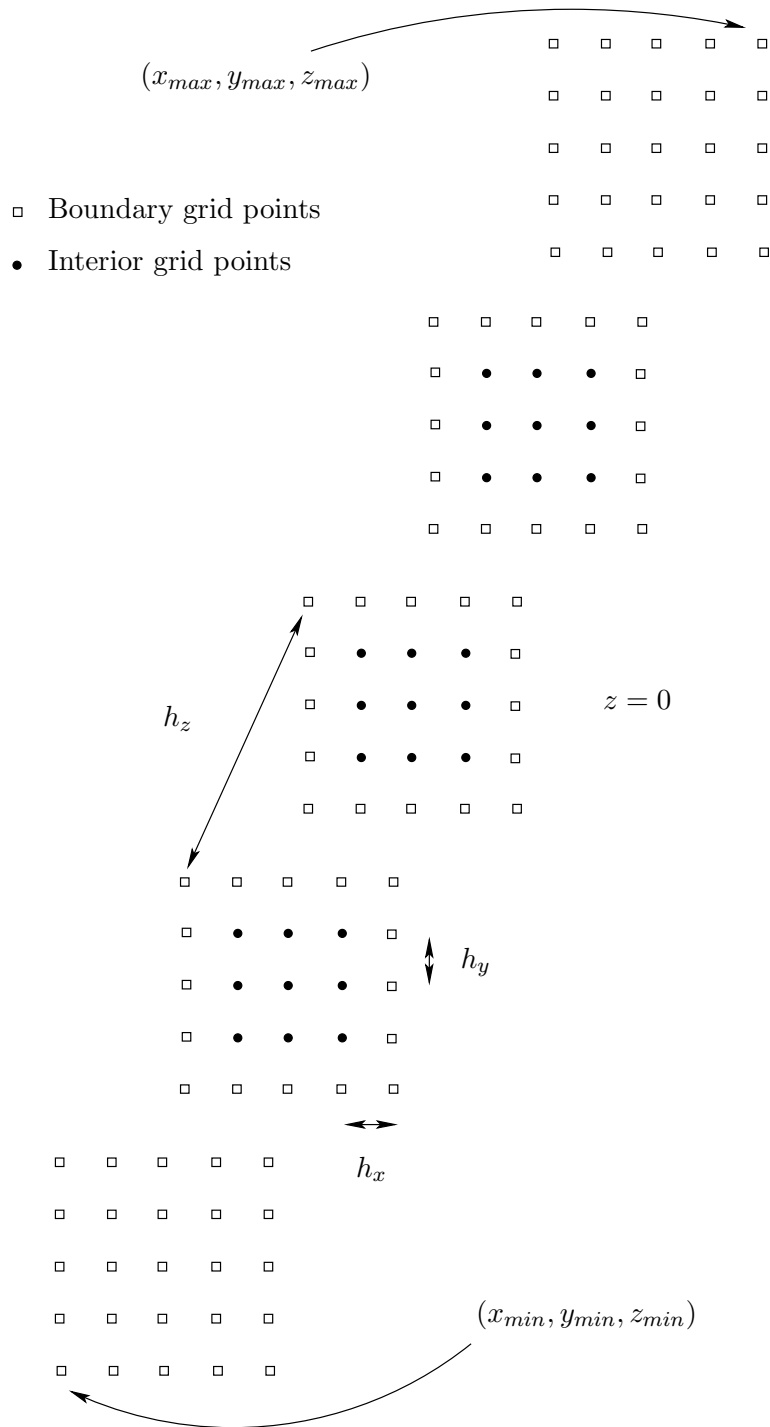


Figure 4.1: Illustration of sample 3D finite difference grid of the type used throughout this thesis. The figure shows a 3D Cartesian grid drawn in perspective. This particular grid is composed of $5 \times 5 \times 5$ grid points covering the domain $x_{min} \leq x \leq x_{max}$, $y_{min} \leq y \leq y_{max}$, $z_{min} \leq z \leq z_{max}$. The solid circles represent interior grid points, while the hollow squares denote boundary grid points. For clarity, the grid spacing in the z direction is drawn out of scale, and we emphasize that throughout this manuscript we take $h_x = h_y = h_z \equiv h$.

$x = x_i$ can be approximated with the following combination of grid function values:

$$\frac{d^2}{dx^2}u(x) \approx \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}. \quad (4.11)$$

The stencil of this finite difference operator (see Fig. 4.2) has an overall multiplicative factor, $1/h^2$, and is weighted by the ordered set of integers $(1, -2, 1)$ corresponding to the coefficients that multiply the grid function values in the numerator of the approximation. In general, the particular grid function values, as well as the corresponding weights, that go into the definition of any finite difference operator depend on both the order of the differential operator being approximated and the level of accuracy (as a function of the mesh spacing) desired in the approximation. App. B tabulates all the FDAs used in the discretization of our model equations. In addition, App. C describes a set of Maple procedures we have written with an aim to automate 1) the derivation of finite difference operators of any order and 2) the subsequent discretization of PDEs, of any *differential* order, using these operators. The appendix includes several examples that illustrate the usage of these procedures.

We now turn our attention to some additional basic concepts which are very useful in the analysis of FDAs.

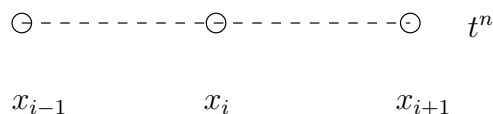


Figure 4.2: Illustration of stencil for a finite difference approximation of the second spatial derivative operator.

4.1.1 Basic Concepts Related to FDAs

Consider a system of partial differential equations described abstractly by

$$Lu - f = 0, \quad (4.12)$$

where, in the most general case, L is a set of m differential operators acting on a vector, u , of m unknown dependent functions: $u = (u_1, \dots, u_m)$. Each of the u_i is a function of the n independent variables (including time, as appropriate); that is, $u_i = u_i(x_1, \dots, x_n)$ for $i = 1 \dots m$. Similarly, f

is a vector of m prescribed functions—often known as *sources*—with $f_i = f_i(x_1, \dots, x_n)$ in general. For the remainder of this section, and strictly for the sake of presentation, we will specialize to the case $m = 1$ (i.e. for a single unknown function), while emphasizing that all of the discussion is equally valid for the general- m case.

We can write any finite difference approximation of (4.12) as

$$L^h u^h - f^h = 0, \quad (4.13)$$

where the operator L^h is an FDA of the differential operator, L , u^h is the discrete solution and f^h are the values of the source function f restricted to the grid points. The superscript h notation emphasizes that h is the fundamental control parameter of the discretization, and we observe that (4.13) will in general constitute a set of algebraic equations—possibly nonlinear—for the discrete unknowns, u^h .

In order to quantify how much the discrete solution, u^h , deviates from the continuum solution, u , for any specific value of h , it is natural to define the *solution error*, e^h , as

$$e^h \equiv u^h - u. \quad (4.14)$$

Clearly, in most cases of interest, the solution error *will* depend on the (finite) value of h . Thus, a key element in the analysis of finite difference approximation schemes is the investigation of the relation between e^h and h as $h \rightarrow 0$. Does the discrete solution approach the continuum one? If so, at what rate? Roughly speaking, numerical analysts judge the quality of the FDA by how fast the error goes to zero as a function of h . This motivates the introduction of the concept of *convergence* of the approximate solution. The approximation is said to converge if and only if

$$\lim_{h \rightarrow 0} u^h = u, \quad (4.15)$$

or, equivalently,

$$\lim_{h \rightarrow 0} e^h = 0. \quad (4.16)$$

The *order* of convergence (not to be confused with the differential order of the system being

approximated) measures the rate at which the error converges to zero. If

$$\lim_{h \rightarrow 0} e^h = O(h^p), \quad (4.17)$$

then the discrete solution is said to converge to the continuum one with order p , or to be p -th order accurate. For example, if $p = 1$, the solution is said to converge linearly (first-order accurate), while if $p = 2$, the solution converges quadratically (second-order accurate).

In principle, the discrete solution, u^h , of the algebraic system of equations (4.13) can be calculated exactly (assuming that a solution exists). In the context of numerical computation, “exact” typically means “as accurate as the particular representation of floating-point numbers being used allows”; this is often referred to as “at the level of machine precision”. Let us momentarily restrict our attention to the case that the system (4.13) is linear in the unknowns u^h . Then, again in principle (and ignoring issues concerning the conditioning of the linear system), we could employ standard numerical linear algebra software that implements some variant of Gaussian elimination to *directly* compute u^h to machine precision. Moreover, for the case that (4.13) is nonlinear in the u^h , we could employ a so-called global Newton iteration: at each stage this would require the solution of a linear system, which again could be achieved via Gaussian elimination. However, especially for the algebraic systems that result from the finite difference discretization of time-independent PDEs with dependence on multiple spatial variables (e.g. 2D and 3D systems), the “exact” calculation of u^h in this manner is often prohibitively expensive computationally. In particular, for such problems, the amount of computational work that must be expended to compute the solution *per unknown* tends to increase as the mesh is refined. The reason for this increase in work per unknown as $h \rightarrow 0$ can be traced to the phenomenon of *fill in* that occurs during the Gaussian elimination procedure, which is typically based on the so called **LU** decomposition, wherein a matrix, \mathbf{A} is factored as $\mathbf{A} = \mathbf{L}\mathbf{U}$, where \mathbf{L} and \mathbf{U} are, respectively, upper- and lower-triangular matrices. Even with special orderings of the unknowns (such as that provided by nested dissection [164]), the matrices that result from finite difference discretization of elliptic operators in 2- and 3-D—though sparse—have bandwidths that increase as $h \rightarrow 0$. This leads to an increasing density of non-zero elements appearing in \mathbf{L} and \mathbf{U} , which in turn results in the increasing cost per unknown as the mesh is refined.

Given this situation, it often turns out to be more computationally efficient to calculate an

approximate solution of the linear system arising from an FDA of an elliptic equation (or a linearization thereof) through an iterative method. In this case, an “exact” solution is only obtained in the limit of an infinite number of iterations (assuming the iteration converges), but in practice the iteration can be terminated when some convergence criterion is achieved. We note that iterative techniques form the basis of the multigrid method that we use to solve the elliptic PDEs in our model, and which we discuss in Sec. 4.3. Additionally, they are also used *directly* to solve our time-implicit (Crank-Nicholson) discretization of the hyperbolic equations, as is described in more detail in Sec. 4.1.3.

As the name suggests, an iterative method involves the computation of a sequence of approximations of the solution unknown, u^h , and we will generically denote any of these approximations as \tilde{u}^h . It should be noted that implicit in the use of an iterative technique is the fact that \tilde{u}^h must be *initialized* in some fashion.

Given any iterate, \tilde{u}^h , an important quantity is the *residual*, r^h , associated with \tilde{u}^h , and defined by

$$r^h \equiv L^h \tilde{u}^h - f^h. \quad (4.18)$$

Thus, the residual quantifies the amount by which \tilde{u}^h fails to satisfy the FDA (4.13), and the solution of (4.13) by an iterative process is then equivalent to driving the residual to 0.

One last quantity that is very useful in the analysis of FDAs is the *truncation error*, defined by

$$\tau^h \equiv L^h u - f^h, \quad (4.19)$$

where we note that u is the *continuum solution of the differential system* (4.12). Note that from Eq. (4.12) we have $f^h = Lu$, where the right hand side of this last expression is understood to be evaluated on the discrete mesh. We can thus rewrite (4.19) as

$$(L^h - L)u = \tau^h, \quad (4.20)$$

and from this form we see that the truncation error directly measures the deviation between the actions of the finite difference and continuum operators on the continuum solution.

A finite difference approximation is said to be *consistent* (with the underlying PDE) if and only

if the truncation error goes to zero as h tends to zero:

$$\lim_{h \rightarrow 0} \tau^h = 0. \quad (4.21)$$

Clearly, consistency is a necessary condition for convergence of the FDA. Finally, the FDA is said to be p -th order accurate if:

$$\lim_{h \rightarrow 0} \tau^h = O(h^p), \quad (4.22)$$

and where p is strictly positive integer.

4.1.2 Richardson Expansions

We note that for any given FDA, L^h , of some differential operator, L , the functional form of the truncation error, τ^h , can be explicitly computed, typically using Taylor series expansion. On the other hand, it is not immediately clear what, if anything, we can say about the functional form of the actual solution error, $e^h = u^h - u$. However, this was, in fact, addressed almost a century ago in a landmark paper by L. F. Richardson [165]. Richardson posited (stated without proof) that provided a finite difference scheme was $O(h^2)$ and centred, so that the truncation error, τ^h , had the form

$$\tau^h = h^2 \tau_2(t, x, y, z) + h^4 \tau_4(t, x, y, z) + h^6 \tau_6(t, x, y, z) + \dots \quad (4.23)$$

then the solution error, e^h , would have a similar expansion

$$e^h = h^2 e_2(t, x, y, z) + h^4 e_4(t, x, y, z) + h^6 e_6(t, x, y, z) + \dots \quad (4.24)$$

Here, a key observation is that the functions e_2, e_4, e_6, \dots appearing in the expansion of the solution error have *no h -dependence*. This seemingly innocuous observation has far-reaching consequences for the analysis of the error in finite difference calculations, and we will refer to an asymptotic ($h \rightarrow 0$) expansion of the form (4.24) as a *Richardson expansion*. Note that one immediate consequence of (4.24)—if it is indeed true for the particular second-order, centred FDA under consideration—is that an $O(h^2)$ truncation error implies an $O(h^2)$ solution error.

In addition, we can postulate the existence of Richardson expansions in schemes where non-centred difference approximations are employed: in this case, the expansion will include terms proportional to h^p , where p can be both even *and* odd. Similarly in situations where the approx-

imation is $O(h^p)$ accurate for $p > 2$ the expansion will begin with a term $h^p e_p(t, x, y, z)$, and will contain terms that have strictly even powers of h , or even and odd powers of the mesh scale, depending on the type of finite differences used.

In all cases, should a Richardson expansion exist, we can argue that *consistency* leads to *convergence*:

$$\lim_{h \rightarrow 0} \tau^h = O(h^p) \quad \Rightarrow \quad \lim_{h \rightarrow 0} e^h = O(h^p). \quad (4.25)$$

In general it is not feasible to *prove* the existence of a Richardson expansion for a given FDA of a set of PDEs: such a proof will be at least as difficult as proving global existence and uniqueness of the PDEs themselves. However, in simple cases where both the PDE and the discretization are amenable to closed form analysis it may be possible to provide a proof—see page 111 of reference [166] for an example. Most importantly for us, in practice one can always establish the existence of the expansion (non-rigourously) by construction, i.e. through examination of the h -dependence of the numerical solutions themselves. In particular, if the discrete solution converges at the expected rate as $h \rightarrow 0$, then one can usually be quite certain that a Richardson expansion exists. We will return to this point in Sec. 4.2.1.

4.1.3 The Crank-Nicholson Discretization Scheme

To conclude this section we consider a specific type of FDA that can be applied to time dependent PDEs, and which was used in the finite-differencing of the hyperbolic equations appearing in our model. We illustrate the technique using a very simple PDE, namely the one dimensional advection equation. However, as discussed at the end of this sub-section, the method can be generalized to virtually any system of PDEs that are first-order in time.

The 1D dimensional advection equation is perhaps the simplest example of a hyperbolic PDE. Posing it as an initial-boundary-value problem on a finite domain we have:

$$\frac{\partial u(t, x)}{\partial t} = \frac{\partial u(t, x)}{\partial x} \quad -x_{\min} < x < x_{\max} \quad , \quad t \geq 0, \quad (4.26)$$

$$u(0, x) = u_0(x), \quad (4.27)$$

$$u(t, x_{\max}) = 0. \quad (4.28)$$

where the last condition can be interpreted as meaning that no disturbances are entering the domain

from the right. Note that there is *no* boundary condition at $x = x_{\min}$, since the only characteristics in the problem are given by $t = -x + \text{const}$, so that at $x = x_{\min}$, the single characteristic field is strictly outgoing. (Since it is largely irrelevant to the current discussion, and to keep the following presentation straightforward, we will not stipulate any details concerning the update of the grid function value at $x = x_{\min}$.)

One rule-of-thumb that is often used when constructing FDAs is to try to keep the stencils as centred as possible (i.e. the geometric structure of the stencil is to be kept symmetric about the grid point at which the approximation is applied). Use of centred schemes typically results in several benefits, including:

1. For given accuracy (e.g. $O(h^2)$), a minimization of the number of unknowns appearing in the stencil.
2. Conversely, for a given number of unknowns in the stencil, a maximization of the accuracy of the scheme (e.g. $O(h^2)$ vs $O(h)$).
3. Symmetry of the matrices, L^h , that operate on the grid function, u^h , and that result from the discretization (or a linearization of the FDA).

Concerning the last point, the properties of symmetric matrices are well known, and can often be used to prove existence and uniqueness of solutions of the discrete equations. In addition, stability theorems that establish convergence of iterative techniques for solving the linear systems resulting from finite-difference approximation are generally easier to establish for the case of symmetric matrices. Here and throughout this thesis, then, we use centred difference approximations (in time as well as in space) whenever possible.

Returning to the advection equation, we introduce a discrete domain (t^n, x_i) as follows:

$$t^n \equiv n\Delta t, \quad n = 0, 1, 2, \dots \quad (4.29)$$

$$x_i \equiv x_{\min} + (i - 1)h, \quad i = 1, 2, \dots, n_x, \quad (4.30)$$

$$u_i^n \equiv u^h(t^n, x_i), \quad (4.31)$$

$$\Delta x \equiv h = \frac{(x_{\max} - x_{\min})}{(n_x - 1)}, \quad (4.32)$$

$$\Delta t = \lambda h, \quad (4.33)$$

where λ is known as the Courant factor. Here, and throughout the thesis, we will assume that for any specific sequence of calculations in which the mesh resolution is changed, λ is held fixed. This means that the overall difference scheme will always be characterized by the *single* mesh spacing, h .

The Crank-Nicholson discretization scheme is a two-level (i.e. involves unknowns at two discrete instants of time, t^n and t^{n+1}), $O(h^2)$ method, which, as shown in Fig. 4.3, is centred in both space and time about the fictitious grid point, $(t^{n+1/2}, x_i)$. One obvious advantage of this method is that it uses only two time levels. Related to this is the fact that among schemes which are $O(h^2)$ accurate in time, the actual magnitudes of the temporal contributions to the truncation and solution errors tend to be minimized. Finally, another appealing feature of Crank-Nicholson discretization is that it tends to minimize the appearance of instabilities that frequently arise in the finite difference approximation of hyperbolic equations. As applied to the advection equation,

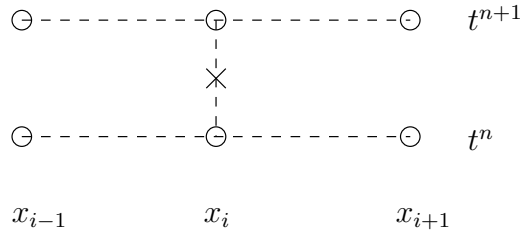


Figure 4.3: Crank-Nicholson Stencil. The fictitious grid point, $(t^{n+1/2}, x_i)$, about which the scheme is centred, is marked with the \times symbol.

and using the standard $O(h^2)$ approximation of the first spatial derivative, the Crank-Nicholson scheme is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \mu_t \left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \right), \quad 2 \leq i \leq n_x - 1. \quad (4.34)$$

Here μ_t is an $O(\Delta t^2)$ ($O(h^2)$) time-averaging operator. Its action on any grid function is defined by

$$\mu_t u_i^n = \frac{1}{2} (u_i^{n+1} + u_i^n). \quad (4.35)$$

Inserting the above definition into (4.34) yields the following explicit expression for the Crank-Nicholson scheme as applied to the advection equation:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{2} \left[\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x} + \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} \right], \quad 2 \leq i \leq n_x - 1. \quad (4.36)$$

These last equations can be rewritten as

$$-\frac{\lambda}{4}u_{i+1}^{n+1} + u_i^{n+1} + \frac{\lambda}{4}u_{i-1}^{n+1} = u_i^n + \frac{\lambda}{4}(u_{i+1}^n - u_{i-1}^n), \quad 2 \leq i \leq n_x - 1. \quad (4.37)$$

The above equations, supplemented with the boundary condition $u_{n_x}^{n+1} = 0$ (that follows from (4.28)), and some auxiliary condition of the form $u_1^{n+1} = U(u_i^{n+1}, i = 2 \dots n_x; u_i^n, i = 1 \dots n_x)$, constitute a linear system for the unknowns $u_i^{n+1}, i = 1, 2, \dots, n_x$ (provided that U is linear in $u_i^{n+1}, i = 2 \dots n_x$). That is, we can write the update in the form

$$\mathbf{A}\mathbf{u}^{n+1} = \mathbf{b}. \quad (4.38)$$

where \mathbf{A} is an $n_x \times n_x$ matrix, and \mathbf{u}^{n+1} and \mathbf{b} are length- n_x column vectors.

We note that \mathbf{A} is *not* diagonal—that is, there is coupling between the individual advanced-time unknowns, u_i^{n+1} , and hence our discretization is an example of what is known as an *implicit* scheme. (In contrast, an *explicit* scheme would be one for which it would be possible to write, for arbitrary n_x , explicit expressions for the u_i^{n+1} .) Now, assuming that the update for u_1^{n+1} is of the form $u_1^{n+1} = U(u_2^{n+1}; u_i^n, i = 1 \dots n_x)$, then \mathbf{A} is actually a tridiagonal matrix. That is, only the elements of its diagonal, as well as the the immediate upper and lower diagonals are non-vanishing. We note in passing that these tridiagonal systems can be solved very efficiently—with $O(n_x)$ operations—using specialized solvers such as those available in the widely used LAPACK linear algebra package [167].

The evolution equations used in this dissertation were cast as a system of first order PDEs in (t, x, y, z) , with each individual PDE assuming the following general form:

$$\frac{\partial u(t, x, y, z)}{\partial t} = F(x, y, z, u, u_x, u_{xx}). \quad (4.39)$$

Each evolution equation was then discretized using an $O(h^2)$ Crank-Nicholson scheme. As explained above in the context of the advection equation, and emphasizing that *all* mesh spacings, $\Delta x, \Delta y, \Delta z$ and Δt are $O(h^2)$, this involves the following approximation for the time derivative:

$$\frac{\partial u(t, x, y, z)}{\partial t} \approx \frac{u_{ijk}^{n+1} - u_{ijk}^n}{\Delta t} = \left. \frac{\partial u(t, x, y, z)}{\partial t} \right|_i^{n+1/2} + O(h^2). \quad (4.40)$$

Again, as was the case for the advection equation, the spatial part of the PDEs are replaced by approximations that average between the advanced and current time levels using the time averaging operator, μ_t . Specifically, for a general right hand side, F , we have

$$F(x, y, z, u, u_x, u_{xx}) \approx \frac{1}{2} \left(\hat{F}_{ijk}^{n+1} + \hat{F}_{ijk}^n \right) = F(x, y, z, u, u_x, u_{xx})|_i^{n+1/2} + O(h^2), \quad (4.41)$$

where \hat{F} denotes an approximation to F in which u_x and u_{xx} have been replaced by the usual $O(h^2)$, centred finite-difference formulae. The result of the Crank-Nicholson discretization, as applied to our system of 4 hyperbolic PDEs, Eqs. 2.173–2.174, is a non-linear system of algebraic equations to be solved for the advanced-time unknowns, u_{ijk}^{n+1} . Here we remind the reader that u represents any of the hyperbolic variables appearing in our model; that is u is any of ϕ_1 , ϕ_2 , Π_1 or Π_2 .

Fortunately there are efficient iterative methods to solve this type of algebraic system of equations. In particular, we chose to apply point-wise iterative Newton-Gauss-Seidel relaxation.²⁴ We also note that the relaxation was applied to each of the 4 hyperbolic unknowns individually (the *decoupled* approach), rather than *collectively*, which would have involved the solution of a 4×4 system at each grid point.

We should also remark that discretizations of hyperbolic systems of equations are notorious for being susceptible to numerical instability [169, 170]. However, the implicit Crank-Nicholson discretization scheme is well known for being an *unconditionally* stable scheme—i.e. stable for arbitrary values of the Courant factor—for large classes of time dependent problems. Although we have no mathematical proof of stability for the scheme applied to our hyperbolic equations, we have not encountered any issues with numerical stability in the calculations described in this thesis.

²⁴The reader can refer to App. D for a brief overview of the method or to Varga [168] for a more comprehensive treatment.

4.2 Convergence Testing of Finite Difference Solutions

This section introduces two techniques that we employ to establish that our finite difference solutions *do* converge to continuum solutions of the PDEs governing our model. The first technique, which is now almost universally employed in numerical relativity work (as well as in other fields), is a basic convergence test that, for fixed initial data, examines the behaviour of the finite difference solution as the discretization scale, h , is varied. In our case, this is done by defining a quantity, which we call the convergence factor, $Q^h(t)$, and which is easily computed from a series of calculations in which successive discretization scales are typically related by a factor of 2. When the empirically measured $Q^h(t)$ behaves in the expected fashion, we establish that the finite difference solution is converging to *some* continuum solution, but there is then still the possibility that it may not actually be a solution of the original PDEs. This could easily happen if the overall finite difference approximation was *not* consistent with the model PDEs, but *was* consistent with some other set of PDEs (an omission or accidental modification of some term in any of the original PDEs would generally lead to precisely such a situation). We thus use a second technique, which is perhaps not so commonly used in our field, which is termed independent residual evaluation, wherein we essentially directly verify that our discrete solutions *do* approach the desired continuum solutions as $h \rightarrow 0$.

4.2.1 Convergence Factor

As suggested above, we perform convergence tests in a very straightforward manner: we fix initial data, and then calculate numerical solutions using at least three different discretization scales (which we will often refer to as *levels* of discretization). Successive levels are characterized by mesh spacings that are in a 2 : 1 ratio. Thus, at a minimum—and again emphasizing that we fix the initial data, as well as all parameters defining the numerical solution, *except* for h —we calculate u^h , u^{2h} and u^{4h} , which are finite difference solutions generated with mesh spacings h , $2h$ and $4h$, respectively. In addition, and again for the sake of simplicity and convenience, we ensure that each grid in the sequence “aligns” with the next coarsest grid. That is, the grid points of the level- h mesh are a subset of those of the level- $2h$ mesh, which in turn are a subset of those of the level- $4h$

mesh. We then define our convergence factor, $Q^h(t)$, as follows

$$Q^h(t) \equiv \frac{\|u^{4h} - u^{2h}\|}{\|u^{2h} - u^h\|}, \quad (4.42)$$

where $\|\cdot\|$ is any appropriate spatial norm, such as l_2 ($\|\cdot\|_2$) or l_∞ ($\|\cdot\|_\infty$),²⁵

$$\|u^h\|_2 = \frac{1}{\sqrt{n}} \left[\sum_{i=1}^n |u_i^h|^2 \right]^{1/2}, \quad (4.43)$$

$$\|u^h\|_\infty = \max_{1 \leq i \leq n} |u_i^h|, \quad (4.44)$$

and where it is to be understood that the subtraction of individual grid function values implicit in the expressions $u^{2h} - u^h$ and $u^{4h} - u^{2h}$ occurs only at the set of grid points common to the two meshes. Now, assuming that

1. our FDA is $O(h^2)$ and completely centred, so that the truncation error, τ^h , contains no terms proportional to h^{p_o} , where p_o is an odd integer,
2. any finite difference solution, u^h has a Richardson expansion of the form (4.24),

then the discrete solutions u^h , u^{2h} and u^{4h} can be expanded as

$$u^h = u + h^2 e_2 + h^4 e_4 + \dots \quad (4.45)$$

$$u^{2h} = u + (2h)^2 e_2 + (2h)^4 e_4 + \dots \quad (4.46)$$

$$u^{4h} = u + (4h)^2 e_2 + (4h)^4 e_4 + \dots \quad (4.47)$$

Substitution of the above expressions into (4.42) yields

$$\begin{aligned} Q^h(t) &= \frac{\|(u + (4h)^2 e_2 + (4h)^4 e_4 + O(h^6)) - (u + (2h)^2 e_2 + (2h)^4 e_4 + O(h^6))\|}{\|(u + (2h)^2 e_2 + (2h)^4 e_4 + O(h^6)) - (u + h^2 e_2 + h^4 e_4 + O(h^6))\|} \quad (4.48) \\ &\approx \frac{\|12h^2 e_2 + 240h^4 e_4\|}{\|3h^2 e_2 + 15h^4 e_4\|}. \end{aligned}$$

²⁵We note that, in contrast to what is often done in numerical analysis, we define the l_2 norm here to include a normalization by an appropriate power of the number, n , of grid points—i.e. by $1/\sqrt{n}$. When dealing with functions defined on different meshes, this is the natural and convenient approach: for example, grid functions defined on meshes with different resolutions will then tend to have approximately the same norm.

Thus, as the mesh spacing tends to 0, we have

$$\lim_{h \rightarrow 0} Q^h(t) = \lim_{h \rightarrow 0} \frac{\|12h^2 e_2 + 240h^4 e_4\|}{\|3h^2 e_2 + 15h^4 e_4\|} = \lim_{h \rightarrow 0} \frac{12h^2 \|e_2\|}{3h^2 \|e_2\|} = 4. \quad (4.49)$$

Note that in deriving (4.49) we have assumed that the terms in the (asymptotic) Richardson expansion are monotonically decreasing. This will usually be the case when the mesh scale, h , is significantly smaller than the typical scale over which u varies. Conversely, if h is of the order of the variation scale of u , one cannot reasonably expect that $Q^h(t)$ will be close to 4.

Finally, as we have already mentioned, one needs a *minimum* of 3 levels of discretization to compute $Q^h(t)$. In practice, it is best to use as many levels as possible in testing for convergence.²⁶ As finer and finer discretization scales are used, we expect $Q^h(t)$ to get closer and closer to the constant 4, and deviations from this anticipated behaviour can be used as a powerful diagnostic for detecting subtle mistakes in the FDA (e.g. where certain terms that are relatively small have only been discretized to $O(h)$ accuracy, or perhaps not even consistently with the PDEs).

4.2.2 Independent Residual Evaluation

As mentioned previously, although a convergence test of the form detailed above can provide strong evidence that a discrete solution is converging to *something* as $h \rightarrow 0$, it does *not* directly establish that the limiting solution satisfies the original set of PDEs. We therefore introduce a second technique, known as independent residual evaluation²⁷, that aims to remedy this shortcoming. Importantly, the method can be used to test the correctness of an arbitrary FDA of an arbitrary set of PDEs, as well as the actual computer code that solves the FDA. We feel that is difficult to overemphasize the value of using this technique for testing the *complete* process of discretizing a set of PDEs, and then solving for the discrete solutions. Particularly for complicated PDEs in multiple dimensions, the FDAs used, and the algorithms needed to solve them can be extremely complex: there are thus many places where mistakes can be made, and it will not always be obvious from the numerical solutions that something has gone awry. Independent residual evaluation has the potential to detect virtually any and all errors that have been made, assuming only the following:

²⁶Available computational resources tend to be the limiting factor in this respect: for 3+1 dimensional computations such as that performed in this thesis, a calculation with mesh size $h/2$ requires about $2^4 = 16$ as much computer time as one with mesh size h .

²⁷This technique was first introduced by Choptuik in his studies of critical phenomena in gravitational collapse [171], where its use was crucial to the validation of new and unexpected results.

1. The PDEs that we start from are correct.
2. The discrete solutions, u^h , that are computed have Richardson expansions.

We also note that one of the greatest advantages of independent residual evaluation is that it does *not* require the existence of any exact or previously-computed solution of the PDEs.

We illustrate the technique by again considering a general set of PDEs written in the form introduced in Sec. 4.1.1:²⁸

$$Lu - f = 0. \quad (4.50)$$

The continuum equation is discretized as

$$L^h u^h - f^h = 0, \quad (4.51)$$

with the presumption being that the discretization is correct. Thus, assuming that L^h is a second order approximation of L , we will have

$$L^h = L + h^2 E_2 + O(h^4), \quad (4.52)$$

where E_2 is a differential operator, whose form can be explicitly computed, and that will be of higher degree than L (e.g. if L involves second derivatives, then E_2 will generally involve fourth derivatives, given that the FDA is $O(h^2)$). We now assume that we have computed numerically an approximate discrete solution, \tilde{u}^h , and have shown that it is converging at second order to *some* continuum function, \bar{u} (e.g. so that $Q^h(t) \approx 4$). We can therefore be confident that \tilde{u}^h has a Richardson expansion of the form

$$\tilde{u}^h = \bar{u} + h^2 \bar{e}_2 + O(h^4). \quad (4.53)$$

Furthermore, by construction (i.e. by virtue of the actual numerical calculation), we assert that the residual, r^h , associated with \tilde{u}^h has a magnitude less than some convergence tolerance, ϵ , where ϵ can generally be much smaller than the typical magnitude of the truncation error, τ^h . That is,

²⁸As in Sec. 4.1.1, and for clarity of exposition, we suppress the (t, x, y, z) dependence of continuum functions (u , f , τ_2 etc.) as well as their discrete counterparts (u^h , f^h etc.). In addition, we will proceed by assuming that L and any finite difference approximations of L , as well as the operators E_2 and \hat{E}_2 appearing in (4.52) and (4.56), respectively, are linear; however, the technique is equally applicable to nonlinear equations.

we have

$$\|L^h \bar{u}^h - f^h\| = \|r^h\| < \epsilon \ll \|\tau_h\|. \quad (4.54)$$

Relative to the discretization (4.51), then, we have established that we have a convergent solution, which limits to \bar{u} . What we have *not* established, however, is whether $\bar{u} \equiv u$. In order to do so, we need to simultaneously establish that L^h is a consistent approximation of L , and that our implementation correctly solves the algebraic equations (4.51).

We thus consider an independent (distinct) discretization of the PDE:

$$\hat{L}^h \hat{u}^h - f^h = 0. \quad (4.55)$$

This new discrete operator \hat{L}^h can be expanded in the same manner as L^h was:

$$\hat{L}^h = L + h^2 \hat{E}_2 + O(h^4) \quad (4.56)$$

where \hat{E}_2 is another higher-order differential operator that will be *not* be the same as the operator E_2 appearing in (4.52). Note that we assume here that \hat{L}^h is also an $O(h^2)$ approximation to L , but this is not essential (e.g. an $O(h)$ approximation could also be adopted). Given this second discretization, the process of independent residual evaluation consists simply of applying the left hand side of (4.55) to our putative numerical solution, \tilde{u}^h . Defining I^h to be the independent residual we have

$$I^h \equiv \hat{L}^h \tilde{u}^h - f^h = \left[L + h^2 \hat{E}_2 + O(h^4) \right] [\bar{u} + h^2 \bar{e}_2 + O(h^4)] - f^h \quad (4.57)$$

$$= L\bar{u} - f^h + h^2 \left(\hat{E}_2 \bar{u} + L\bar{e}_2 \right) + O(h^4) \quad (4.58)$$

$$= L\bar{u} - f^h + O(h^2). \quad (4.59)$$

Here we have again used the (inessential) assumption that L and E_2 , are both linear.

Now, if the computed continuum solution, \bar{u} is *not* a solution of the original PDE we will generically have $\bar{u}(t, x, y, z) = u(t, x, y, z) + e_0(t, x, y, z)$, where $e_0 = O(1)$, and as $h \rightarrow 0$ we will find

$$I^h \equiv \hat{L}^h \tilde{u}^h - f^h = Lu - f^h + Le_0 + O(h^2) = Le_0 + O(h^2). \quad (4.60)$$

Thus, unless $Le_0 \equiv 0$, which is extremely improbable, then as $h \rightarrow 0$, we will find that I^h will converge to some non-zero *function*, given by Le_0 . In other words we will have $I^h = O(1)$. Conversely, if \bar{u} and u are the same function, we will have

$$I^h \equiv \hat{L}^h \tilde{u}^h - f^h = Lu - f^h + O(h^2) = O(h^2), \quad (4.61)$$

with the crucial observation being that $I^h \rightarrow 0$ as $h \rightarrow 0$, i.e. that I^h really is “residual” quantity with respect to both the independent discretization, and the continuum PDE. If we compute I^h in a typical sequence of calculations aimed establishing convergence (i.e. with all problem parameters, save h , fixed), and observe behaviour as given by (4.61) then we have provided very strong evidence that the approximate discrete solution \tilde{u}^h is converging to the true continuum solution u of the PDE.

In order for the technique of independent residual evaluation to be effective, it is vital that the second discretization, \hat{L}^h , is consistent with the original PDE operator L . Otherwise, measurement of an I^h which is $O(1)$ could signal an inconsistency in \hat{L}^h , instead of some problem with the principal discretization, L^h , or in the solution of the algebraic equations that result from that discretization. In this regard, one should note the following. First, there is no need to *solve* the system (4.55) for \hat{u}^h : rather, the independent discretization \hat{L}^h is simply *applied* to the computed solution \tilde{u}^h . Intimately related to this observation is the fact that for the case of time dependent systems, one does not need to worry about the stability of \hat{L}^h . In addition, as already noted, there is no need for \hat{L}^h to be of the same order of accuracy as L^h . For example, if L^h is $O(h^2)$ accurate, but \hat{L}^h is only $O(h)$, then if \tilde{u}^h is converging to the true continuum solution, u , we will find that I^h is $O(h)$ (i.e. still a residual quantity). However, if convergence is to some function $\bar{u} \neq u$, then I^h will still be $O(1)$.

All of this means that one has lot of latitude in how \hat{L}^h is constructed, and, perhaps more importantly, one can easily use symbolic computing software to generate \hat{L} from a high-level description of the PDEs that is relatively easy to check for correctness. This significantly decreases the probability that a I^h that is measured to be $O(1)$ is due to a mistake in the derivation or implementation of \hat{L} .

For example, in the construction of the independent residual operator for our set of equations, we used $O(h)$ forward difference approximations for time derivatives, and $O(h^2)$ forward difference

approximations for the spatial derivatives.²⁹ This is to be contrasted to the $O(h^2)$ approximations used for all derivatives in the construction of the basic difference scheme itself (Crank-Nicholson for the time derivatives, standard $O(h^2)$ centred approximations for the spatial derivatives). In addition, the Maple procedure described in App. C was used to generate the actual code to evaluate I^h . The results from our application of independent residual evaluation to series of calculations for several distinct initial data sets are described in the next chapter, along with the results from measurement of the basic convergence factor $Q^h(t)$ discussed in Sec 4.2.1.

²⁹For grid points close to the outer boundary of the computational domain where the forward difference formulae could not be applied, we used $O(h^2)$ backwards difference expressions.

4.3 Multigrid Techniques

4.3.1 Introduction

As stated in the previous section, iterative techniques such as point-wise Newton-Gauss-Seidel (NGS), provide efficient methods for solving the systems of nonlinear algebraic equations that result from our time-implicit discretization of the hyperbolic PDEs of our model (those that govern the scalar field). Specifically, we can generally expect convergence to the level of intrinsic discretization error using a number of sweeps through the mesh which is independent of the mesh scale, h . Thus, in the absence of any elliptic PDEs, the operation count for solving the hyperbolics would be linear in the number of discrete unknowns.³⁰

However, as is well known, relaxation methods—even when they are accelerated using the successive-overrelaxation (SOR) technique—are *not* very efficient for solving the systems that result from finite-difference discretization of elliptic PDEs. In particular, the amount of computational work *per unknown* needed to solve such systems generally increases as $h \rightarrow 0$.

For example, consider what one might call the Poisson equation in one spatial dimension:

$$u(x)_{xx} = f(x), \quad (4.62)$$

and, neglecting boundary conditions, discretize this equation on a uniform mesh using the usual centred, $O(h^2)$ FDA of u_{xx} :

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} - f_i = 0. \quad (4.63)$$

As described in App. D, assuming that the NGS iteration visits the grid points in so-called lexicographic order (i.e. $i = 2, 3, \dots, n_x - 1$), the i -th component of the *running residual* vector is given by

$$[\mathbf{r}_i^{(k)}]_i = \frac{\tilde{u}_{i-1}^{(k)} - 2\tilde{u}_i^{(k-1)} + \tilde{u}_{i+1}^{(k-1)}}{h^2} - f_i. \quad (4.64)$$

Each relaxation sweep (iteration) is supposed to bring the approximate solution $\tilde{u}_i^{(k)}$ closer to the exact discrete solution u_i . The convergence of this process can also be viewed in terms of driving the running residuals to 0. In fact, in the analysis of the convergence of relaxation methods [168], the effect of the iteration on the running residual vector is often described in terms of the action

³⁰Indeed, as noted by Teukolsky [172], for many purely hyperbolic systems of PDEs, one can expect the so-called “iterative Crank-Nicholson” method to converge to the level of the solution error in precisely 2 iterations. However, in our case, due to the additional coupling with the elliptic equations, this observation does not apply.

of the *residual amplification matrix*, \mathbf{A} . Assuming a linear set of discrete equations³¹ this matrix maps the residual vector at iteration k , $\mathbf{r}^{(k)}$, to the corresponding vector at iteration $k + 1$, $\mathbf{r}^{(k+1)}$:

$$\mathbf{r}^{(k+1)} = \mathbf{A}\mathbf{r}^{(k)}. \quad (4.65)$$

Clearly, \mathbf{A} must be a contraction map in order for the iteration to converge, and, ideally, the spectral radius, ρ , of the amplification matrix would always be bounded away from unity—i.e. $\rho(\mathbf{A}) < 1$ —to *ensure* rapid convergence. Unfortunately, for the case of NGS applied to the simple, but representative, set of equations defined by (4.62), this is not the case. Indeed, it can be shown that the spectral radius of the amplification matrix, \mathbf{A}_{NGS} satisfies

$$\lim_{h \rightarrow 0} \rho(\mathbf{A}_{\text{NGS}}) = 1 - O(h^2). \quad (4.66)$$

Operationally, this means that one must perform $O(h^{-2}) = O(n_x^2)$ relaxation sweeps in order to achieve convergence, so that, for 1D problems, the overall operation count is $O(n_x^3)$, rather than the optimal $O(n_x)$. Furthermore, the type of behaviour given by (4.66) is seen when NGS is applied to virtually any finite differenced elliptic system, in any number of spatial dimensions. Although the use of overrelaxation (SOR) can significantly improve convergence, one still generally has

$$\lim_{h \rightarrow 0} \rho(\mathbf{A}_{\text{SOR}}) = 1 - O(h), \quad (4.67)$$

and the work per unknown needed to achieve convergence still increases as the mesh is refined.³² In short, especially for cases such as ours, where an elliptic system must be solved at each discrete time step, relaxation techniques do not provide a viable route for solving discretized elliptic equations.

On the other hand—and again using the simple toy problem (4.62) for illustrative purposes—by expanding the residual vector in terms of the eigenvectors of the amplification matrix, it is possible to show that the asymptotic convergence rate (4.66) is dominated by the eigenvectors with wavelengths that are long compared to the mesh scale, h . That is, each relaxation sweep damps long-wavelength components of the running residual by factors given by the associated eigenvalues,

³¹Again, the analysis can be extended to the nonlinear case, when a linearization method such as Newton iteration is used.

³²Furthermore, one can only attain (4.67) when the choice of relaxation parameter is precisely optimal, and determining an optimal value is non-trivial in practice, particularly for coupled nonlinear elliptic systems such as ours.

σ_i^{long} , where $\sigma_i^{\text{long}} = 1 - O(h^2)$. Conversely, for short-wavelength modes—which we will define as modes having wavelengths, λ_i , in the range $2h \leq \lambda_i \leq 4h$ —one finds that the corresponding eigenvalues are bounded away from unity, and, importantly, *are independent of h* . In other words, as the amplification matrix is repeatedly applied to the residual vector, the short wavelength (high frequency) components of the residual are rapidly annihilated, while the long wavelength (low frequency) components are very slowly damped. In short, relaxation methods such as Gauss-Seidel tend to be very good *smoothers* when applied to finite difference discretizations of elliptic PDEs, and that this smoothing property applies both to the (running) residual vector, $\mathbf{r}^{(k)}$ and the deviation between the approximate solution after the k -th relaxation sweep, $\tilde{\mathbf{u}}^k$, and the exact solution, $\tilde{\mathbf{u}}^h$ of the discrete equations.

Fortunately there is an extremely efficient methodology for solving finite difference approximations of elliptic equations. This is the *multigrid* technique [35, 149], which was largely developed by A. Brandt in the late 1970’s and early 1980’s. In terms of computational complexity, the most striking feature of multigrid is that, in many cases, it is able to provide solutions of the discrete equations with $O(N)$ computational work and $O(N)$ storage, where N is the total number of discrete unknowns.³³ Importantly, this computationally-optimal performance can be achieved for quite general nonlinear systems of elliptic PDEs, such as the one encountered in our model. It should be noted that from the fact that multigrid can provide solutions with $O(N)$ work one can immediately deduce that its convergence rate in such instances must be h -independent.

The classic multigrid method (i.e. the approach due to Brandt) builds on two fundamental observations. The first of these we have just made: relaxation methods, such as (Newton)-Gauss-Seidel, can be used to efficiently *smooth* both the residuals and solution errors on any particular grid. In multigrid then, one uses relaxation not to *solve* the discrete set of equations, but only to smooth the residuals and solution errors.³⁴

The second observation is that once the discrete elliptic problem has been smoothed, it can be well represented on an coarser grid, having, for example, a discretization scale $2h$. Especially for multidimensional equations, the work need to solve the coarse-grid problem will be a fraction of that

³³The multigrid method is itself iterative, so by solving the discrete equations we again mean “to the level of the intrinsic solution error, $u^h - u$ ”, where u is the continuum solution and u^h is the exact solution of the discrete system.

³⁴Again, when we speak of “solution errors” in the context of an iterative method such as multigrid, we generally mean the errors in the approximate solution, \tilde{u}^h , at any stage of the iteration, relative to the exact solution, u^h , of the difference equations. However, assuming that the true solution error, $u^h - u$ is smooth, smoothing $\tilde{u}^h - u^h$ is clearly equivalent to smoothing $\tilde{u}^h - u$.

required for the fine-grid solution. Moreover, we can then apply these ideas recursively: relaxation on the $2h$ grid quickly annihilates modes in the residuals and errors that are high-frequency (short-wavelength) with respect to $2h$, and then we can pose a version of the problem on an even coarser grid having mesh spacing, $4h$, and so on. Eventually, this smoothing and coarsening process on grids with ever increasing mesh spacings leads to a grid with so few points that *solution* of the discrete equations using relaxation (or even a direct technique) requires a negligible amount of computational work. The process of working on coarser and coarser grids is then reversed: smooth corrections to grid functions on finer grids are interpolated from the coarse grids. This process of interpolation introduces new high frequency error components in the fine-grid functions, but these can be effectively damped with a few more relaxation sweeps. Once the algorithm returns to the original, finest mesh, the relaxation sweeps performed after the interpolation of the correction from the second-finest mesh typically results in a solution in which the error has been reduced by a substantial amount.

In the next subsection we will discuss these basic ideas in somewhat more detail for the case of a particular multigrid method, known as the Full Approximation Storage (FAS) scheme, which is suitable for the treatment of nonlinear problems.

4.3.2 The FAS Algorithm

Again adopting a notation in which the dependence of functions on the relevant set of independent variables (e.g. (x, y, z)) is implicit, we write a general nonlinear system of elliptic PDEs as

$$N[u] = f. \tag{4.68}$$

Here N denotes the set of nonlinear differential operators acting on the solution vector, u , while f is a vector of source functions which is independent of u or any of its spatial derivatives. We then consider some finite difference approximation of (4.68) which, as usual, is characterized by a *single* discretization scale h . This yields a nonlinear set of algebraic equations that we write as

$$N^h[u^h] = f^h. \tag{4.69}$$

The solution of (4.69) is expected to be computed through some iterative process. Each step of the iteration defines the residual vector, r^h , corresponding to the current approximation, \tilde{u}^h , of the

discrete solution, u^h :

$$r^h \equiv N^h[\tilde{u}^h] - f^h. \quad (4.70)$$

The goal of the iterative process is to drive the residual to zero or, equivalently, to drive the solution error, v^h , to zero, where v^h is defined by

$$u^h = \tilde{u}^h + v^h. \quad (4.71)$$

As mentioned in the last section, relaxation methods such as Newton-Gauss-Seidel tend to be excellent smoothers. The FAS multigrid method takes advantage of this fact to (implicitly) smooth the correction v^h on each member of a *hierarchy* of grids. Each grid in the hierarchy is labelled by a parameter l (the *level* parameter), where $l = 1$ is the coarsest level and $l = l_{\max}$ is the finest. From considerations of both total computational cost, as well as ease of implementation, the grids in the hierarchy usually satisfy $h_{l+1} = h_l/2$, and we have adopted this choice in our current work. Starting on the finest mesh, i.e. on level l_{\max} , we perform a few relaxation sweeps using a method such as NGS.³⁵ Provided that these sweeps *do* smooth both r^h and v^h on the fine grid, we can then proceed to define a coarse grid problem. This is done by setting up an appropriate equation for the correction (solution error), v^h , on the coarse grid.

In order to define this equation, first note that if N^h were *linear*, we could apply it to both sides of (4.71) to get

$$N^h[u^h] = N^h[\tilde{u}^h + v^h] = N^h[\tilde{u}^h] + N^h[v^h] = r^h + f^h + N^h[v^h]. \quad (4.72)$$

Using (4.69) this could be further simplified to

$$N^h[v^h] = -r^h. \quad (4.73)$$

This last equation actually forms the basis of a multigrid algorithm known as the *linear correction scheme* (LCS). However, since N^h is a nonlinear operator, we cannot proceed along this route.

Nonetheless, Eq. (4.73) *does* suggest a similar treatment for the nonlinear case. Subtracting the

³⁵We note that, as discussed in Sec. 2.4.5, *point-wise* Newton-Gauss-Seidel is not always an effective smoother for elliptic discretizations, but it is for our system, provided that we work in the standard (i.e. non-compactified) Cartesian coordinates.

definition of the residual, Eq. (4.70), from the basic difference equation (4.69) yields

$$N^h[u^h] - N^h[\tilde{u}^h] = N^h[\tilde{u}^h + v^h] - N^h[\tilde{u}^h] = -r^h. \quad (4.74)$$

Thus, rather than having an explicit equation for the correction v^h , (4.74) is to be viewed as an equation for the “full approximation”, $u^h = \tilde{u}^h + v^h$, and hence the name “Full Approximation Scheme”.

From the smoothing assumption, all terms appearing in (4.74) are smooth on the scale, h , of the finer grid. One can then sensibly pose a coarse grid form of (4.74) as follows:

$$N^{2h}[u^{2h}] - N^{2h}[I_h^{2h}\tilde{u}^h] = -I_h^{2h}r^h, \quad (4.75)$$

Here, u^{2h} is the unknown which is to be computed on the coarse grid (note that it does *not*, in general, satisfy $N^{2h}u^{2h} = f^{2h}$), and I_h^{2h} —known as a *restriction* operator—transfers a fine grid function to the coarse grid. Once u^{2h} has been determined by solving (4.75), the fine grid unknown, u^h , is updated using

$$\tilde{u}^h = \tilde{u}^h + I_{2h}^h(u^{2h} - I_h^{2h}\tilde{u}^h). \quad (4.76)$$

Here, I_{2h}^h —known as a *prolongation* operator transfers a coarse grid function to the fine grid. In practice, I_{2h}^h generally performs polynomial interpolation of an order that can depend on: the differential order of the elliptic system, the order of accuracy of the FDA, and the specific smoother being used. As Brandt [35] emphasizes, Eq. (4.76) is to be preferred over the more obvious

$$\tilde{u}^h = I_{2h}^h u^{2h} \quad (4.77)$$

since the former retains (useful) high frequency information already computed in u^h , whereas the latter does not.

Eqs. (4.75) and (4.76) constitute the core of the FAS algorithm. Our quick derivation of the FAS scheme also corresponds to a specific point of view, wherein the multigrid method is seen as a solver that uses a hierarchy of coarser grids to accelerate the convergence of error components which have long wavelengths on the fine grid.

Again as stressed by Brandt [35], there is a useful “dual” interpretation of the FAS algorithm in which the fine grids are used to provide correction terms to coarse grid systems, essentially allowing

unknowns on coarse grids to be determined to the same accuracy—relative to the continuum solution—as the unknowns on the finest grid. It is instructive to quickly work through the alternate derivation of (4.75) from this vantage point, and to do so we must introduce the concept of *relative truncation error*.

Recall from Sec.4.1.1, Eq. (4.19), that the truncation error, τ^h , of a finite difference scheme is defined in terms of the action of the discrete operator on the continuum solution:

$$\tau^h \equiv N^h[u] - f^h. \quad (4.78)$$

For the purposes of the current development, it is convenient to use the FDA (4.69) in the form $f^h = N^h[u^h]$ to rewrite the above equation as

$$\tau^h = N^h[u] - N^h[u^h]. \quad (4.79)$$

Now, if it was possible to know the exact value of the truncation error in advance, then by adding it to the right hand side of (4.69) we would have an equation

$$N^h[u^h] = f^h + \tau^h, \quad (4.80)$$

whose solution, u^h , would be identical to the restriction of the continuum solution, u , of the PDE (4.68) to the mesh points. Unfortunately *a priori* knowledge of τ^h is, of course, equivalent to *a priori* knowledge of u , so at first glance, this observation does not seem very useful.

However, suppose that it is possible to compute some approximation, $\tilde{\tau}^h$ of τ^h . Then, by adding $\tilde{\tau}^h$ to the right hand side of Eq. (4.69), we would get an equation

$$N^h[u_\star^h] = f^h + \tilde{\tau}^h, \quad (4.81)$$

for a new grid function, u_\star^h , which should be more accurate than the solution u^h of (4.69). That is, we should find $\|u_\star^h - u\| < \|u^h - u\|$, where $\|\cdot\|$ denotes some norm.

In analogy to the definition of the truncation error, τ^h , we now define the relative truncation error, τ_h^{2h} , which involves quantities defined on two adjacent levels within the multigrid hierarchy,

having discretization scales h and $2h$ respectively:

$$\tau_h^{2h} \equiv N^{2h}[I_h^{2h}u^h] - I_h^{2h}(N^h[u^h]). \quad (4.82)$$

Again, I_h^{2h} is a fine-to-coarse transfer operator (restriction operator). Once more using Eq. (4.69) in the form $f^h = N^h[u^h]$, and assuming that $I_h^{2h}f^h = f^{2h}$, this definition can be rewritten as

$$N^{2h}[u^{2h}] = f^{2h} + \tau_h^{2h}. \quad (4.83)$$

Thus, completely parallelling our previous interpretation of τ^h , the relative truncation error, τ_h^{2h} , can be viewed as the correction that must be added to the source term of the coarse grid difference equations in order that the coarse grid solution actually coincide with the (restricted) fine grid solution. Once more, we are unable to compute τ_h^{2h} precisely unless we have the exact solutions, u^h and u^{2h} , of the two discrete systems in hand. However, during the multigrid solution process, we can certainly calculate an approximation, $\tilde{\tau}_h^{2h}$, of τ_h^{2h} using the current estimate, \tilde{u}^h , of the fine grid unknown:

$$\tilde{\tau}_h^{2h} \equiv N^{2h}[I_h^{2h}\tilde{u}^h] - I_h^{2h}(N^h[\tilde{u}^h]). \quad (4.84)$$

Then, replacing τ_h^{2h} with $\tilde{\tau}_h^{2h}$ in (4.83), we have derived the “dual” equation for the coarse grid unknown, u^{2h} (i.e. the full approximation):

$$N^{2h}[u^{2h}] = f^{2h} + \tilde{\tau}_h^{2h}. \quad (4.85)$$

Now from (4.75), and again assuming that $I_h^{2h}f^h = f^{2h}$ and that I_h^{2h} is linear (which in practice it invariably is), we have

$$N^{2h}[u^{2h}] = N^{2h}[I_h^{2h}\tilde{u}^h] - I_h^{2h}r^h \quad (4.86)$$

$$= N^{2h}[I_h^{2h}\tilde{u}^h] - I_h^{2h}(N^h\tilde{u}^h - f^h) \quad (4.87)$$

$$= f^{2h} + \tilde{\tau}_h^{2h} \quad (4.88)$$

which is precisely Eq. (4.85). Thus the two viewpoints lead to the same coarse grid correction equations, and therefore are indeed equivalent.

As summarized in Sec. 4.3.1, the FAS coarse grid correction equations are applied recursively, with relaxation sweeps on level l being followed by the initiation of a coarse grid correction on level $l - 1$. When the coarsest grid ($l = 1$) is reached, u^{h_1} can generally be computed very inexpensively using relaxation or perhaps some other method (such as a direct solution of the difference equations, using a global Newton iteration). Once u^{h_1} has been calculated, another sequence—consisting of fine grid function updates (using prolongation) followed by additional smoothing sweeps—is performed for $l = 2, \dots, l_{\max}$. This entire process of “working down” the hierarchy from the finest to coarsest grid, then “back up” to the finest grid, is known as a V -cycle, due to the V -shape that results from a standard pictorial representation of the algorithm in which the vertical direction encodes the discretization level (with h increasing downwards), while horizontal displacement represents successive stages of the procedure. A pseudo-code version of the basic FAS algorithm described above, and which was used in this thesis, is shown in Fig. 4.4.

To conclude this chapter we reiterate that the elliptic PDEs appearing in our model, Eqs. 2.173–2.185 were discretized using a centred $O(h^2)$ FDA (see App. B for the specific difference operators used). In addition, we used *red-black* ordering for the NGS relaxation sweeps, *full-weighted* restriction for I_h^{2h} and *trilinear* interpolation for I_{2h}^h . The interested reader is directed to [35, 149] for further discussion of these technicalities. Finally, although we *did* experiment with collective relaxation, due to an original concern that the mixed second derivatives in some of the equations might be problematic, we ultimately found that the decoupled approach produced faster running code.

```

subroutine MG_update()
  cycle = 1
  do while (residual > tolerance)
    call vcycle(lmax,cycle,p,q)
    cycle = cycle + 1
  end do
end subroutine

subroutine vcycle(lmax,cycle,p,q)
  do l = lmax to 1 ( cycle from fine to coarse levels )
    if cycle = 1 or l not equal to lmax then
      repeat p times:
        perform a NGS relaxation sweep on level(h[l]):
          u[l] = relax_rb_NGS(u[l],f[l],h[l])

        restrict grid functions to level(h[l-1]):
          u[l-1] = I[l-1][l] u[l]

        compute the relative truncation error of the solution
        on level(h[l-1]):
          tau[l-1][l] = N[l-1] I[l-1][l] u[l] - I[l-1][l] N[l] u[l]

        compute the new RHS vector for level(h[l-1]), by adding the
        restricted RHS to the truncation error:
          f[l-1] = tau[l-1][l] + I[l-1][l] f[l]
      end if
    end do

    solve the system of FDAs on level(h[l]) exactly:
    do while (residual > coarsest grid tolerance)
      u[l] = relax_rb_NGS(u[l],f[l],h[l])
    end do

    do l = 2 to lmax ( cycle from coarse to fine levels )
      compute the CGC from level(h[l-1]) to level(h[l])
      and apply the CGC to unknown variables at level(h[l]):
      u[l] = u[l] + I[l][l-1] ( u[l-1] - I[l-1][l] u[l] )

      repeat q times:
        perform a relaxation sweep on level(h[l]):
          u[l] = relax_rb_NGS(u[l],f[l],h[l])
      end do
    end do
  end subroutine

```

Figure 4.4: A pseudo-code representation of the FAS V-cycle multigrid algorithm used in this thesis.

Chapter 5

Code Validation and Results

This chapter presents some results from the numerical solution of the system of PDEs derived in Chap. 2 and summarized in Sec. 2.5. The results discussed here are restricted to the case of boson stars having a potential that only includes a mass term, so that $U(|\phi|^2) = m^2|\phi|^2$.³⁶ As discussed in Chap. 3, the mass parameter m can be chosen arbitrarily as part of our overall specification of a system of units, and we have performed all of our calculations with $m = 1$.

Sec. 5.1 provides some details concerning our numerical code as well as the nature of the various initial data configurations that we have considered. This is followed by three sections devoted to validation and error analysis of the code. Specifically, we report the results of convergence tests and independent residual evaluation, as defined and discussed in Ch. 4, for representative calculations involving:

1. A generic initial configuration of the scalar field (Sec. 5.2).
2. Single boson stars at rest in the computational domain (Sec. 5.3).
3. A single boson star moving through the computational domain (Sec. 5.4).

The overall success of these convergence tests—most notably the convergence of the independent residuals—is considered the strongest evidence we have that our code *does* correctly compute solutions of our model that converge to the continuum limit as the basic mesh scale, h , approaches 0. In addition, we provide evidence that our numerical results conserve ADM mass and Noether charge to the expected order of accuracy.

The final two sections of the chapter focus on the dynamics of two boson stars, and the calculations discussed therein constitute the major new computational results of this thesis. Sec. 5.5 is concerned with the simulation of a head-on collision between two boson stars, in which we observe “solitonic” behaviour that has previously been seen in other studies of self-gravitating scalar field

³⁶As mentioned in Chap. 2, these configurations are sometimes referred to as *mini boson stars*, since for any plausible particle mass, the total gravitating mass of the star is very small. Consequently, such objects are thought to be highly unlikely to be of any astrophysical significance, especially as dark matter candidates.

configurations [173, 174, 128, 130]. Sec. 5.6 then considers the interesting case of the simulation of boson star binaries for three different choices of initial data parameters, each of which produces a distinct end state:

1. Long-lived orbital motion.
2. Merger of the stars that results in a conjectured rotating and pulsating boson star.
3. Merger of the stars that leads to the conjectured formation of a black hole.

Finally, Sec. 5.7 summarizes our main results and outlines some possible directions for improvements and additional developments of this project.

5.1 Summary of the Numerical Code

The set of PDEs governing our model—as derived and discussed in Ch. 2—was discretized using the finite difference techniques described in Ch. 4. Specifically, an $O(h^2)$ Crank-Nicholson scheme was applied to:

1. The 2 first-order-in-time equations that govern the time evolution of the real and imaginary components of the complex scalar field, $\phi_1(t, x, y, z)$ and $\phi_2(t, x, y, z)$, respectively.
2. The 2 first-order-in-time equations for the corresponding conjugate momenta, $\Pi_1(t, x, y, z)$ and $\Pi_2(t, x, y, z)$.

Additionally, $O(h^2)$ centred FDAs were used to discretize the following elliptic PDEs appearing in the model:

1. The slicing condition for the lapse function, $\alpha(t, x, y, z)$.
2. The Hamiltonian condition for the conformal factor, $\psi(t, x, y, z)$.
3. The momentum constraints for the three components of the shift vector, $\beta^x(t, x, y, z)$, $\beta^y(t, x, y, z)$ and $\beta^z(t, x, y, z)$.

We reemphasize that in generating the results described below, Dirichlet boundary conditions were imposed on all variables (see Sec. 2.4.5).

The numerical code that was written to solve the finite difference equations (FDEs) originating from the discretization sketched above consisted of separate components for the hyperbolic and

elliptic variables. For the hyperbolic unknowns, the Crank-Nicholson FDEs were generated and solved using RNPL (Rapid Numerical Prototyping Language) [175, 176]. RNPL takes a high-level specification of difference equations written in a natural operator form, and produces routines that employ point-wise Newton-Gauss-Seidel iteration to compute advanced-time unknowns. As discussed in Chap. 4 this iteration typically converges rapidly for time-implicit discretizations of wave equations, and our experience was consistent with that observation. For the case of the FDEs governing the discrete elliptic unknowns we wrote FORTRAN 77 routines that implemented an FAS multigrid algorithm, as also described in the previous chapter.

A pseudo-code description of the overall code flow is given in Fig. 5.1. Execution of the program begins with a call to the initial data solver denoted `IVP_solver` in the figure.³⁷ In turn, this solver makes use of a set of routines that generate solutions representing the static spherically symmetric boson stars that were discussed in detail in Chap. 3. The interested reader is referred to App. A for documentation of the highest level routine, `bsidpa`, that can be used to determine spherically symmetric boson star profiles for arbitrary polynomial self-interaction potentials.

The solver for a single boson star returns a set of spherically symmetric functions, including $\alpha(R)$, $a(R)$, $\phi(R)$ and $\Phi(R)$, where R is areal radius. These functions³⁸ must then be transformed to the isotropic radial coordinate, r , and interpolated to the Cartesian computational domain as described in Secs. 3.4 and 3.6, respectively. In addition, if the star is to be boosted at the initial time, the transformations detailed in Sec. 3.7 are applied.

In instances where the evolution is to describe the dynamics of a boson star binary, the above process is carried out for each star, and the resulting functions from the computations for each individual star are added to produce the data for the binary. Although not necessary mathematically—since the initial values of the scalar field variables are essentially unconstrained, other than requirements of smoothness and sufficiently rapid decay as $r \rightarrow \infty$ —when we set up binaries in this manner we try to ensure that the stars are sufficiently well separated that the initial configuration really *does* describe two isolated objects.

Once the scalar field variables have been fixed at $t = 0$, the multigrid solver is invoked to determine the initial values of the elliptic variables, $\alpha(0, x, y, z)$, $\psi(0, x, y, z)$ and $\beta^x(0, x, y, z)$. Here we note that $\beta^x(0, x, y, z)$ is only non-zero when a boost has been applied (either to a single

³⁷We note that our code can handle essentially arbitrary initial configurations for the scalar field. Here we focus discussion on the case of most interest, in which the initial data represents one or more boson stars.

³⁸Refer to Sec. 3.7 for details concerning the computation of the conjugate momenta, Π_1 and Π_2 , from the spherically symmetric solution for a single boson star.

star, or to both stars in the binary case). Furthermore, since we only apply boosts in the x direction we always have $\beta^y(0, x, y, z) = \beta^z(0, x, y, z) = 0$. In performing the multigrid solution we start from initial estimates given by the values of the geometric variables computed from the boson star solver—and post-processed in the manner described in Sec. 3.7.

Once the initial data has been determined, the code enters the main time-stepping loop. After initialization of the advanced values $\mathbf{f}(\mathbf{t}+\mathbf{dt})$ and $\mathbf{g}(\mathbf{t}+\mathbf{dt})$ to those from the previous step $\mathbf{f}(\mathbf{t})$ and $\mathbf{g}(\mathbf{t})$, each time step proceeds by a sub-iteration in which:

1. The advanced values, $\mathbf{g}(\mathbf{t}+\mathbf{dt})$, of the geometry variables are updated using a single multigrid FAS V -cycle with the scalar field values $\mathbf{f}(\mathbf{t}+\mathbf{dt})$ acting as sources.
2. The advanced values, $\mathbf{f}(\mathbf{t}+\mathbf{dt})$, of the scalar field variables are updated using a single point-wise Newton-Gauss-Seidel relaxation sweep, with the values $\mathbf{g}(\mathbf{t})$, $\mathbf{g}(\mathbf{t}+\mathbf{dt})$ and $\mathbf{f}(\mathbf{t})$ acting as sources.
3. The residuals of the finite difference equations for the advanced values of the scalar field are computed.

This sub-iteration continues until the l_2 norm of the scalar field residuals is below some specified tolerance, which was set to 10^{-7} for the calculations described below. Once the sub-iteration has converged, the advanced ($\mathbf{t}+\mathbf{dt}$) values are relabelled as current values (pseudo-routine `swap_levels`). Invocation of the `analysis` routine then effects calculation of quantities such as the independent residuals, the ADM mass and Noether charges, as well as the periodic output of grid function values to disk. This completes one pass of the main loop: the time-stepping procedure is then repeated until the specified final integration time, t_{\max} , is reached.

In order to facilitate the presentation of the numerical results that follow, we now introduce some convenient nomenclature and notation. Recalling that we use a single, uniform grid (mesh) to perform our computations, the actual number of grid points associated with any of our simulations can be represented by a vector we call the `shape` of the mesh. `shape` is defined by

$$\mathbf{shape} = [N_x, N_y, N_z], \tag{5.1}$$

where N_x , N_y and N_z are the number of grid points in the x , y and z , coordinate directions, respectively. The total number of grid points in the computational domain is thus simply given by

```
f(t) = scalar field variables at time t
g(t) = geometry variables at time t

t = 0
call IVP_solver[f(t),g(t)]

do while (t < tmax)

    call initial_guess[f(t) -> f(t+dt), g(t) -> g(t+dt)]

    do while (residual[f(t+dt)] > tolerance)

        call MG_update[source = f(t+dt); solve for g(t+dt)]

        call CN_update[source = g(t),g(t+dt),f(t); solve for f(t+dt)]

        call evaluate_residual[f(t+dt)]

    end do

    call swap_levels[f(t) <--> f(t+dt); g(t) <--> g(t+dt)]

    t = t + dt

    call analysis[g(t),g(t+dt),f(t),f(t+dt)]

end do
```

Figure 5.1: A pseudo-code representation of the numerical code. See the text for details concerning the overall program flow.

$N_x N_y N_z$. In addition, when performing computations on a series of meshes with grid spacings in which each finer scale is 1/2 that of the previous grid, it is useful to introduce the notion of the discretization *level*. Let $N = \min(N_x, N_y, N_z)$. Then we demand that there be an integer l such that

$$N = 2^l + 1, \quad (5.2)$$

where l is precisely what we call the level of the calculation. In many cases we run our code with $N_x = N_y = N_z$. In instances where this is not true, then tacit in this definition is the assumption that we will compute with some minimum level l_{\min} , with corresponding $N_{\min} = 2^{l_{\min}} + 1$. Further assuming that $N = N_x$, the coarsest grid will then be characterized by

$$\mathbf{shape}_{l_{\min}} = [N_{\min}, N_{y0}, N_{z0}], \quad (5.3)$$

with $N_{y0} \geq N$, $N_{z0} \geq N$. Finer grids will then have

$$\mathbf{shape}_{l_{\min}+i} = [2^i(N-1)+1, 2^i(N_{y0}-1)+1, 2^i(N_{z0}-1)+1]. \quad (5.4)$$

Another definition concerns the coordinates spanned by a particular grid; i.e. the limits of the computational domain. As discussed in 4.1 this domain is given by

$$x_{\min} \leq x \leq x_{\max}, \quad (5.5)$$

$$y_{\min} \leq y \leq y_{\max}, \quad (5.6)$$

$$z_{\min} \leq z \leq z_{\max}. \quad (5.7)$$

We thus introduce the notion of a *bounding box vector*, denoted by \mathbf{bbox} , and defined by

$$\mathbf{bbox} = [x_{\min}, x_{\max}, y_{\min}, y_{\max}, z_{\min}, z_{\max}]. \quad (5.8)$$

We conclude this section with some brief remarks concerning the computing requirements for our calculations, as well as the nature of some of the figures we use to illustrate key results.

First, the simulations presented here were run on single nodes of one of two clusters located at UBC, which are known as `vnfe4` and `vnfe5`. Each node of `vnfe5` has four 2.4 GHz Dual-Core

AMD(R) Opteron(TM) processors with a total of 4 GB of memory. Nodes on `vnfe4` have two 2.4 GHz Intel(R) Xeon(TM) CPUs and a total of 2 GB of memory. The most extensive calculations were run on `vnfe5`, lasted approximately 260 hours, and used about 700 MB of memory. Larger simulations (up to 1 GB), although possible, were deemed impractical in terms of providing results on the timescale of a few days at most.

Second, many of the figures that follow are surface plots of functions, and for these plots the following should be noted:

1. The “grid lines” in each direction—which are included as a visual aid—are generally only a subset of the total that are available: i.e. the number of grid lines does not reflect the true resolution of the computation.
2. Unless otherwise stated, the vertical displacement of the surface corresponds to the value of the function being plotted.

5.2 Generic Initial Data

A straightforward way to test our code for convergence and consistency is to choose suitably generic initial data which can be specified in closed form, and then evaluate the results of its time evolution. The initial configuration for the results discussed in this section was based on the following generalized gaussian profile for one component of the complex scalar field:

$$\phi(x, y, z) = \phi_0 \exp \left[- \left(\frac{x - x_0}{\delta_x} \right)^2 - \left(\frac{y - y_0}{\delta_y} \right)^2 - \left(\frac{z - z_0}{\delta_z} \right)^2 \right]. \quad (5.9)$$

Here, ϕ_0 , x_0 , y_0 , z_0 , δ_x , δ_y and δ_z are adjustable parameters. The freely specifiable variables $\{\phi_1, \phi_2, \Pi_1, \Pi_2\}$ were then initialized as follows:

- $\phi_1(0, x, y, z) = \phi(x, y, z)$, with $(x_0, y_0, z_0) = (4, 3, 2)$, $\phi_0 = 0.05$ and $(\delta_x, \delta_y, \delta_z) = (4.5, 4, 4)$,
- $\Pi_2(0, x, y, z) = -3\phi_1(0, x, y, z)$,
- $\phi_2(0, x, y, z) = \Pi_1(0, x, y, z) = 0$.

This produces a off-origin, slightly non-spherical, gaussian lump of field that exhibits non-trivial dynamics, but whose evolution also mimics, to some extent, that of a distorted stable boson star. We would like to emphasize that the presence of symmetries in the initial data could lead to a failure in detecting certain implementation errors. Thus, in particular, we designed this initial configuration so that it did *not* have any of the reflection symmetries $x \rightarrow -x$, $y \rightarrow -y$, and $z \rightarrow -z$. We should note, however, that the other initial datasets we subsequently describe *do* have at least one of these symmetries, and thus in principle, should lead to time evolutions that also possess the symmetry.³⁹ We are aware that we could have exploited this fact to reduce the cost of many of our computations. Nonetheless, we did not pursue this option since we wanted to keep our code general, and also wanted to avoid the complications of implementing symmetry conditions in the multigrid solver at this stage.

The bounding box for the experiments described here was `bbox` = $[-11, 19, -12, 18, -13, 17]$, and three different grids with resolutions in a 1 : 2 : 4 ratio were used for the convergence tests and independent residual evaluations. The grid with the finest resolution (level 7), had `shape` = $[129, 129, 129]$, while the coarsest grid (level 5) had `shape` = $[33, 33, 33]$. The Courant factor adopted here—and for all simulations discussed in this chapter—was $\lambda = 0.4$.

³⁹This assumes that there are no dynamically unstable modes in the configurations considered that would break the symmetries, and that could be excited by truncation error and/or roundoff error effects.

Fig. 5.2 shows a time-series of $z = 2$ cuts of the scalar field from a short evolution (roughly one dynamical time) of the initial configuration defined above, and where the finest resolution grid was used. Here, and for other plots of this type shown subsequently, evolution proceeds left-to-right and top-to-bottom. The solution oscillates, but is apparently gravitationally bound, as there is no evidence for dispersal of the scalar field. However, we are not so much interested in the evolution of this data for its physical content, as for what it can tell us about the convergence properties of our code.

In that regard, we first note that all grid functions remain quite smooth during the calculations. Second, as can be seen in the left panels of Figs. 5.3, 5.4 and 5.5, the convergence factors $Q^h(t)$ (defined by 4.42) for the dynamic variables provide clear evidence of second-order convergence. That is, $Q^h(t)$ for all of the variables remains close to the value 4 that is expected for our second-order scheme in the limit that the mesh spacings go to 0 (according to 4.49). Third, and perhaps most importantly, the right panels of these figures show that the independent residuals are also converging. Here we note that in order to more easily assess the convergence rates of the independent residuals from the graphs, we have rescaled them so that, should they be converging at the expected order, the plots from computations at different discretization levels will be roughly coincident. Thus, for the geometric variables, where we expect second order convergence, the l_2 -norms of the independent residuals were rescaled in the following way:

$$\|I^l(t)\|_2 \rightarrow 4^{(l-5)}\|I^l(t)\|_2, \quad (5.10)$$

where l refers to the level of discretization: 5, 6 or 7 in this case. The independent residuals for the scalar field quantities were rescaled in similar fashion. However, since we expect first-order convergence in this instance, the scaling factor is modified accordingly:

$$\|I^l(t)\|_2 \rightarrow 2^{(l-5)}\|I^l(t)\|_2. \quad (5.11)$$

We also observe that, unless otherwise specified, all of the subsequent plots of independent residuals in this chapter show rescaled values as well.

Thus, Figs. 5.3 and 5.4 provide good evidence that the rescaled independent residuals for all of the geometric variables are converging as $O(h^2)$, as expected. We also note that the functional form of $I^h(t)$ for the metric variables is roughly resolution-independent, so that we apparently *do*

have

$$\lim_{h \rightarrow 0} I^h = I_2(t) h^2 \quad \text{and not} \quad \lim_{h \rightarrow 0} I^h = I_2(t, h) h^2. \quad (5.12)$$

Similar remarks can be made concerning the independent residuals for the variables ϕ_1 , ϕ_2 , Π_1 , and Π_2 , except that the convergence of the $I^h(t)$ in this case is linear in h . This is to be expected since, as also discussed in Chap. 4, the independent discretization adopted for approximate time derivatives was a first-order forward difference.

Fig. 5.6 shows the ADM mass, M_{ADM} , and the Noether charge, Q_N , as a function of time for levels 5, 6 and 7. As the resolution increases, $M_{\text{ADM}}(t)$ and $Q_N(t)$ tend to constant functions, indicating that the continuum conservation laws are being recovered as the mesh spacing tends to 0.⁴⁰ In order to better examine the rate of “convergence to conservation” of these values, Fig. 5.7 plots *deviations* of the ADM mass $M_{\text{ADM}}(t)$ and the Noether charge $Q_N(t)$ from their initial values, $M_{\text{ADM}}(0)$ and $Q_N(0)$, as a function of time. Additionally, parallelling what we did for the independent residuals, the deviations computed at different levels of discretization have been rescaled assuming an $O(h^2)$ convergence rate:

$$\Delta M_{\text{ADM}}^l(t) = 4^{l-5} (M_{\text{ADM}}^l(t) - M_{\text{ADM}}^l(0)), \quad (5.13)$$

$$\Delta Q_N^l(t) = 4^{l-5} (Q_N^l(t) - Q_N^l(0)), \quad (5.14)$$

for $l = 5, 6$ and 7 . Again, the fact that the plots of the rescaled deviations are nearly coincident provides strong evidence of second order convergence (to conservation, as a function of time) of both Q_N and M_{ADM} .

Before proceeding to describe our next set of numerical experiments, we make a brief digression to emphasize a key assumption that underpins the efficacy of finite differencing. To keep the discussion simple, we consider the case of a scalar function of one independent variable, $u(x)$, and assume that we have discretized the continuum domain with a uniform grid with mesh spacing h . Then when any differential operator, L , acting on u is approximated with a finite difference operator, L^h , the error in the approximation will be of the form

$$L[u(x)] - L^h[u(x)] = h^p E_p[u(x)] + \dots \quad (5.15)$$

⁴⁰We reemphasize, however, that we know of no proof that $M_{\text{ADM}}(t)$ *should* be conserved for other than spherically symmetric cases.

where the positive integer p is the order of the approximation, and E_p is a differential operator of *higher (differential) degree* than L . For example, the usual $O(h^2)$ centred approximation to $du/dx \equiv u'(x)$ can be written as

$$\frac{u_{i+1} - u_{i-1}}{2h} \equiv \frac{u(x+h) - u(x-h)}{2h} = u'(x) + \frac{h^2}{12}u'''(\xi), \quad (5.16)$$

where $\xi \in [x-h, x+h]$ and $h^2/12u'''(\xi)$ is the error in the approximation. Note, however, that this result is valid *only* if $u'''(x)$ exists and is continuous on the $[x-h, x+h]$.

This simple example highlights the importance of *smoothness* of both the continuum *and* discrete solutions in the context of finite differencing. Specifically, for the discrete case, “smooth” naturally means “smooth on the scale of the mesh”, and when grids that have h comparable to the scale of variation of the solution unknowns are used, one cannot expect “good” results from convergence tests, including independent residual evaluation. The reader should keep this point in mind, especially in later sections of this chapter where the coarsest mesh spacings used *are* comparable to the scales on which the solution is changing. Moreover, as we will see in the next section, (discrete) non-smoothness related to issues such as the treatment of boundary conditions can also easily spoil the convergence of our finite difference scheme.

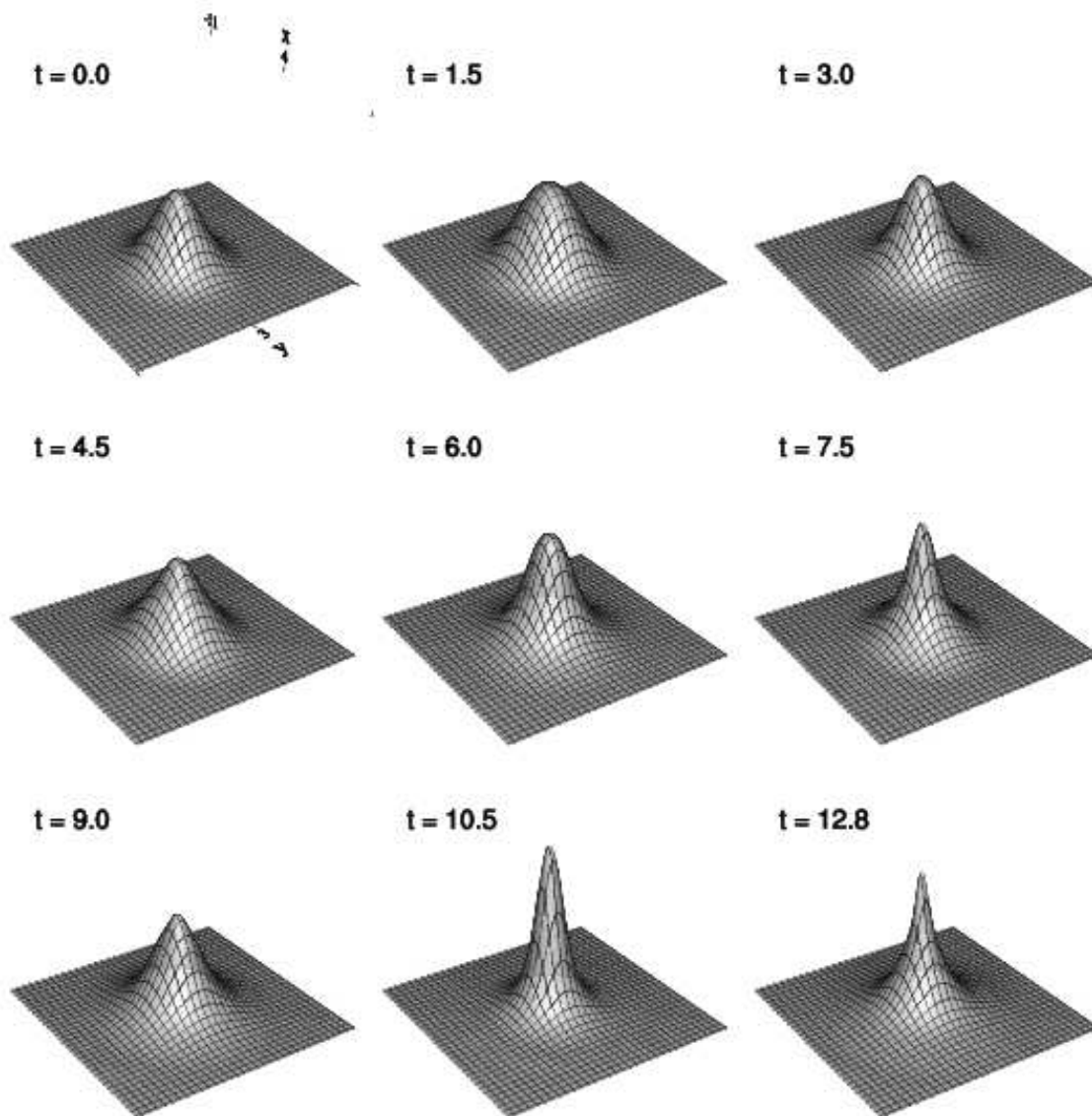


Figure 5.2: Time Evolution of a Generic Gaussian Profile. This figure displays a time-series of $z = 2$ cuts of the scalar field modulus $|\phi(t, x, y, z)|$ from a short evolution (roughly one dynamical time) of the initial configuration defined by Eq. 5.9. Here the value of ϕ_0 appearing in (5.9) is 0.05, the range of the plotted data is $0 \leq |\phi(t, x, y, z)| \leq 0.12$, and the solution was computed using a grid with `shape = [129, 129, 129]` (level 7). Note that the solution oscillates, but is apparently gravitationally bound, as there is no evidence for dispersal of the scalar field

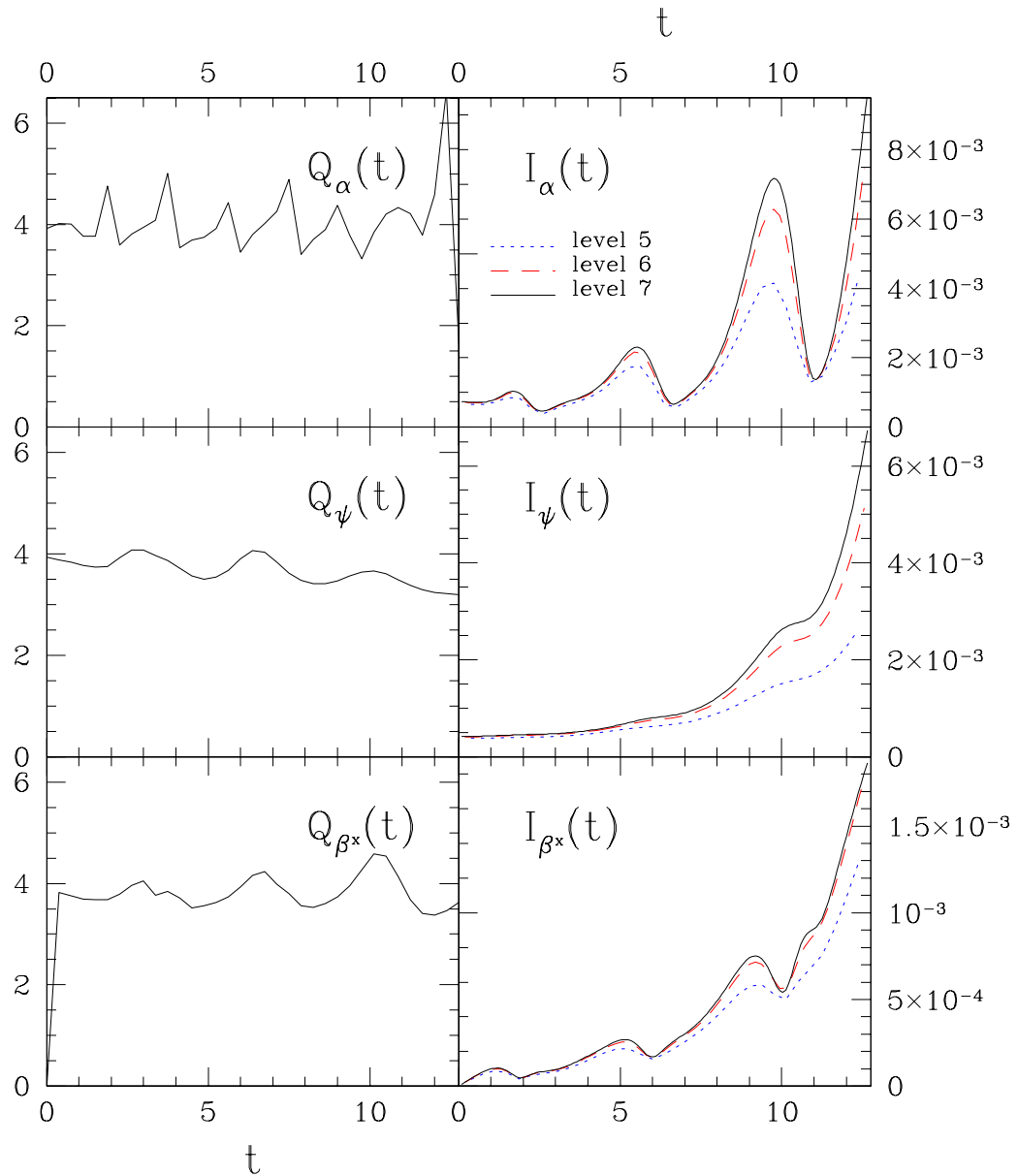


Figure 5.3: Left panels: plots of the convergence factor $Q^h(t)$, as a function of time, t , for the geometric variables α , ψ and β^x , using data from the calculation described in the caption of Fig. 5.2. The convergence factor remains close to the value 4 that is expected for our second-order scheme. Note that $\beta^x(0) = 0$ at all resolutions, and we have thus defined $Q_{\beta^x}^h(0) = 0/0 \equiv 0$. Right panels: l_2 norms of the rescaled independent residuals (see Eq. 5.10), $\|I(t)\|_2$, for the same set of variables. As the level of refinement increases, the plots of the rescaled $I(t)$ become increasingly coincident, providing strong evidence of convergence of our code to the continuum solution.

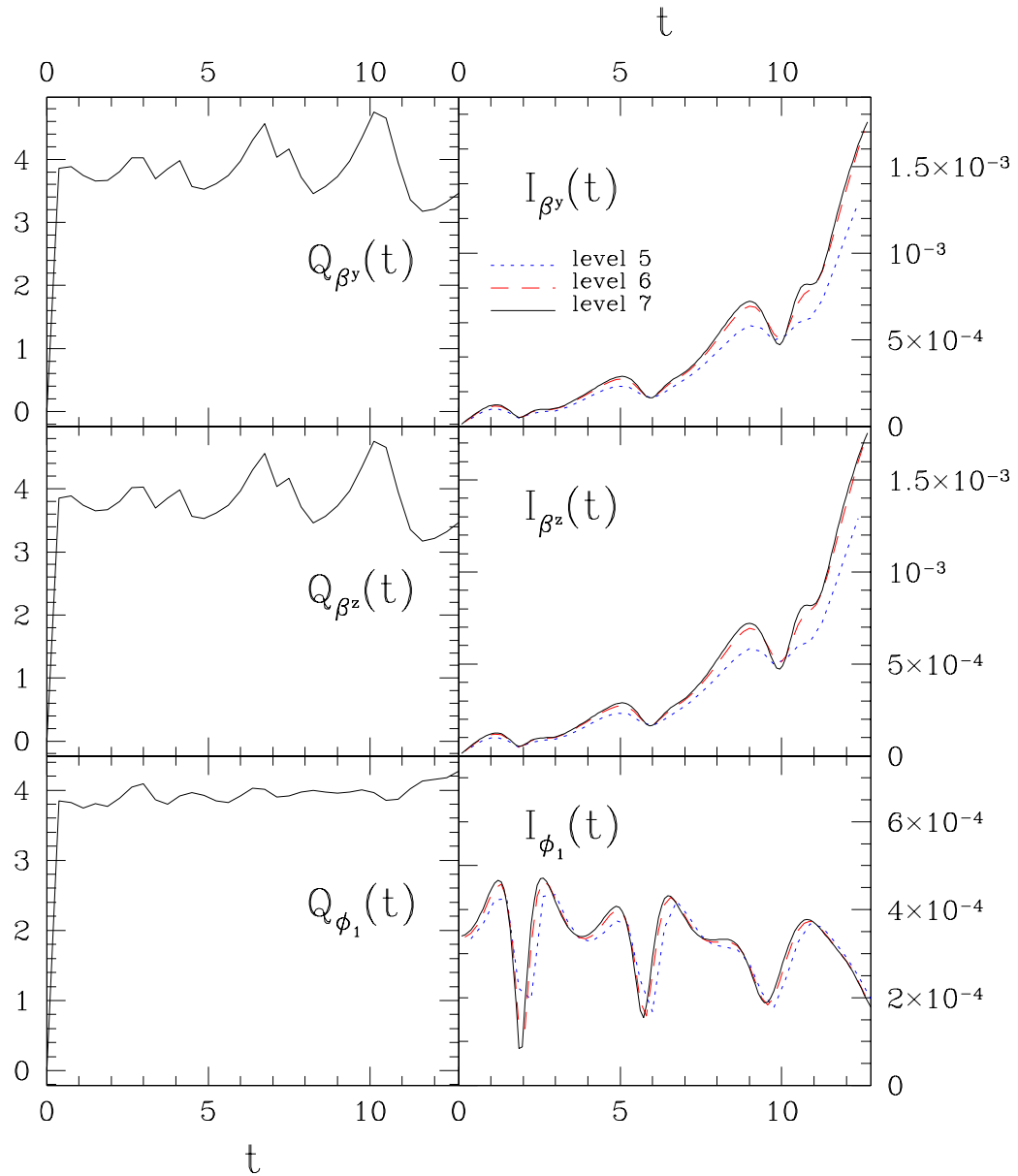


Figure 5.4: Left panels: Plots of the convergence factor $Q^h(t)$ for β^y , β^z and ϕ_1 , using data from the calculation described in the caption of Fig. 5.2. Again, all convergence factors are close to the expected value of 4. Right panels: l_2 norms of scaled independent residuals, $\|I(t)\|_2$, for the same set of variables, and where we have defined $Q_{\beta^y}^h(0) = 0/0 \equiv 0$ and $Q_{\beta^z}^h(0) = 0/0 \equiv 0$. Scaling factors for the two shift vector components and the scalar field are given by (5.10) and (5.11) respectively. We see clear evidence for the convergence of the independent residuals at their expected rates ($O(h^2)$ for β^y and β^z , $O(h)$ for ϕ_1).

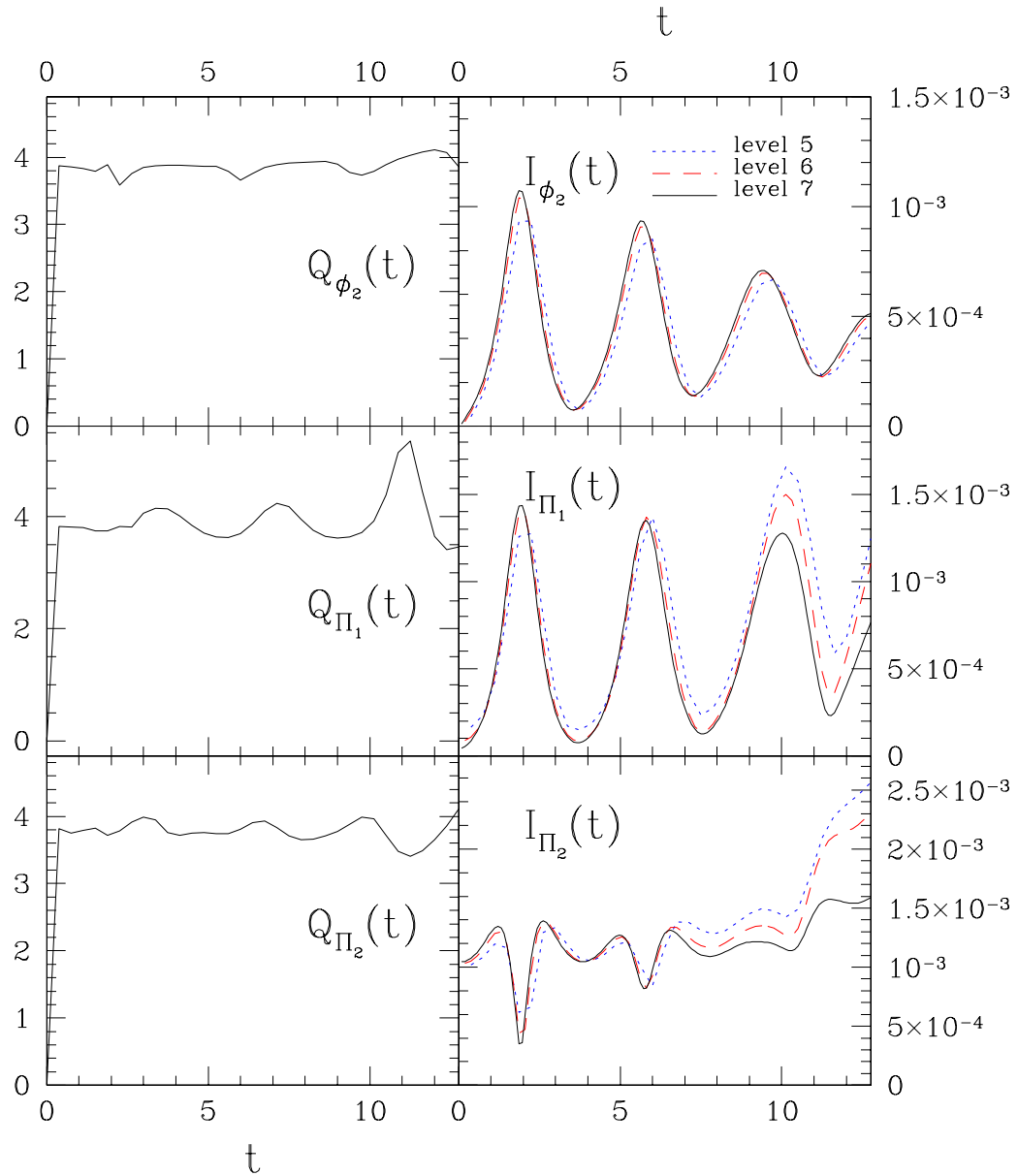


Figure 5.5: Left panels: Plots of the convergence factor $Q^h(t)$ as a function of time for the scalar field variables ϕ_2 , Π_1 and Π_2 , using data from the calculation described in the caption of Fig. 5.2. Right panels: l_2 norms of scaled independent residuals, $\|I(t)\|_2$, for the same set of variables. The residuals were rescaled using (5.11) and clearly are converging as $O(h)$, as expected.

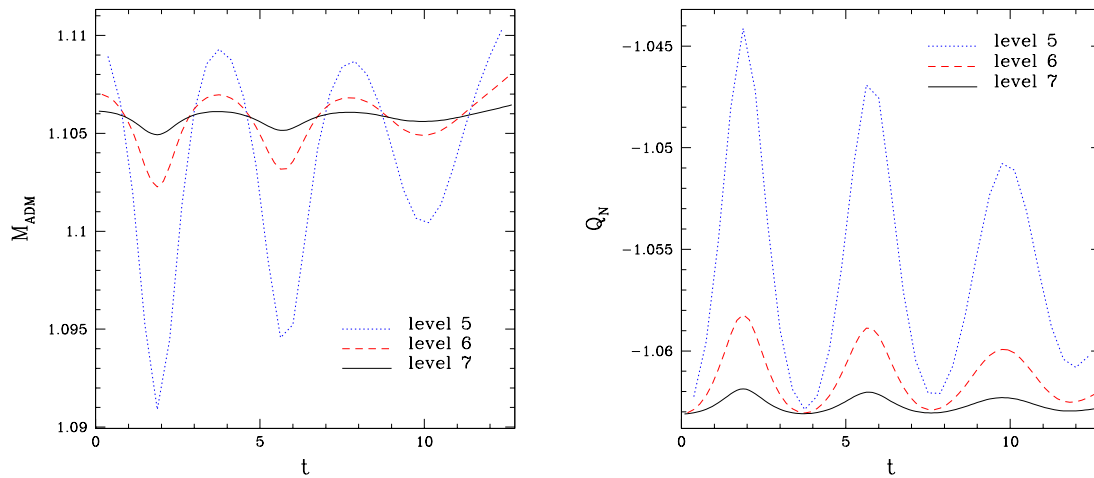


Figure 5.6: ADM mass $M_{\text{ADM}}(t)$ and Noether charge $Q_N(t)$ from the calculation described in the caption of Fig. 5.2. Both plots indicate a trend to conservation of the respective quantity, with a convergence rate of $O(h^2)$, as clearly illustrated in Fig. 5.7 below.

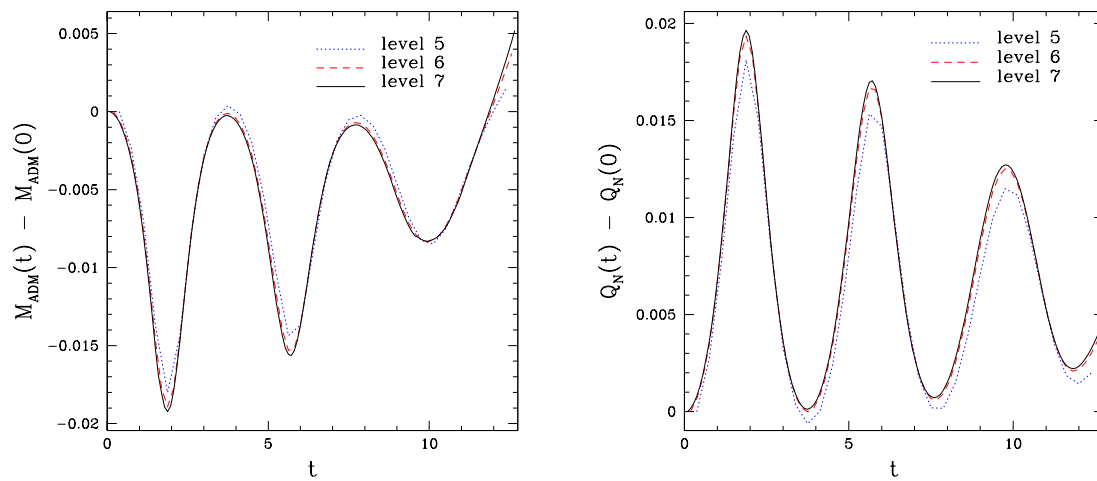


Figure 5.7: Rescaled deviations in ADM mass and Noether charge, as functions of time, t , from the calculation described in the caption of Fig. 5.2. The deviations were rescaled using (5.13) and (5.14), and the near-coincidence of the curves is strong evidence of “convergence to conservation”.

5.3 Static Spherically Symmetric Initial Data - One Star

In the previous section we used a generic type of initial data to establish the correctness and convergence of our numerical code. The results of our convergence tests (including those of the conserved quantities), and independent residual evaluation provide strong evidence that the discrete equations of motion are consistent, have been implemented correctly, and that the numerical solution obtained converges to the continuum solution. In fact, we consider the generic initial data test to be sufficiently comprehensive to establish the overall validity of the code. Nevertheless, in this section and the next we continue with code tests, albeit with slightly different goals.

If there are no problems with the discrete equations of motion, or with convergence of sufficiently smooth solutions, then any deficiencies identified in the numerical results can likely be traced to characteristics of the initial data, or the resolution(s) used in the calculations. With this in mind, there were two key motivations for the tests performed in this section.

First, it is natural to use the spherically-symmetric static configurations that describe single boson stars to simultaneously test 1) the evolution code, and 2) our solutions of the ODEs that determine the static solutions. Recall from Sec. 3.4 that these solutions are computed from the following ansatz for the complex scalar field:

$$\phi(t, R) = \phi_0(R) e^{-i\omega t}, \quad (5.17)$$

where R is the areal radius. It is important to note that even though the stars are characterized by a time *independent* gravitational field, $\phi(t, r)$ is still time *dependent* due to the $e^{-i\omega t}$ factor in the above expression. We should thus expect that the metric variables computed from such initial data should exhibit only approximate time independence, but that the time independence should become more exact as the discretization scale, h , tends to 0.

Second, as also discussed in Chap. 3, the boson stars computed using the above ansatz form a one-parameter family, where $\phi_0(0)$ is a convenient choice for the family parameter. Different members of this family have different ADM masses, as well as different overall strengths of the gravitational field (as quantified, for example, by $\max_R 2m(R)/R$). In this context it seems reasonable to expect that for fixed discretization parameters stronger gravitational fields should lead to larger numerical errors.

We thus consider the numerical evolution of two distinct boson stars using our code. Both

are on the stable branch (see Fig. 3.2) and have central scalar field values $\phi_0 \equiv \phi_0(0) = 0.03$ and $\phi_0 \equiv \phi_0(0) = 0.06$, respectively. Here, and in the following, we adopt a notation such that $S_{\phi_0(0)}$ denotes the boson star having a central field modulus $\phi_0(0)$. With this nomenclature, the stars that we will study are $S_{0.03}$ and $S_{0.06}$. $S_{0.03}$ is less massive, and thus less compact than $S_{0.06}$, as can be inferred from Fig. 3.1. To evolve each of these configurations, we choose a set of grid parameters that is slightly different than that adopted in the previous section: namely `bbox` = $[-15, 15, -15, 15, -15, 15]$, with the coarsest and finest of the three grids used given by `shape` = $[33, 33, 33]$ and `shape` = $[129, 129, 129]$, respectively.

Figs. 5.8 and 5.9 show some snapshots from the time evolution of $S_{0.03}$ and $S_{0.06}$, respectively, as computed on the finest grid. At least to the naked eye these figures seem to indicate that the computed geometries are time independent in both cases. It is thus tempting to conclude that with respect to the first purpose of this particular experiment, the code has passed the test.

However, as Figs. 5.10, 5.11 and 5.12 show, the measured convergence rates for $S_{0.03}$ quantities are significantly poorer than those from the $S_{0.06}$ calculation. This runs counter to the expectation that stronger gravitational fields and more compact scalar field configurations should lead to higher discretization error for a fixed mesh size. As argued above, we are then led to suspect that the convergence problem is related to the nature of the specific data that defines $S_{0.03}$.

The defect in the $S_{0.03}$ computations, relative to the $S_{0.06}$ results, is even more apparent when one examines the rescaled independent residuals. These are shown in Figs. 5.13, 5.14 and 5.15, which plot $\|I(t)\|_2$ for all dynamical variables and at all three resolutions used, and where the rescaling has been done using (5.10) and (5.11) as appropriate. In each panel, the left and right plots correspond to the $S_{0.06}$ and $S_{0.03}$ calculations respectively. Particularly worrisome is the fact that for $S_{0.03}$ the independent residuals for the shift vector components and Π_1 are actually *increasing* as the resolution decreases (this may not be immediately obvious due to the rescaling, but is in fact the case). On the other hand, consistent with the measured values of $Q^h(t)$, all independent residuals for $S_{0.06}$ —with the exception of I_{Π_1} and I_{β^i} —are converging as expected.⁴¹

Finally, although Fig. 5.16 provides some evidence that there is convergence to conservation of mass and Noether charge for both sets of calculations, it is clear that the rate of convergence for mass conservation in the $S_{0.03}$ is indeterminate at best, and that even for the $S_{0.06}$ computations

⁴¹We do not yet understand the difference in the behaviour of I_{Π_1} relative to the independent residuals for the other matter variables, but note that there is also a large “glitch” in Q_{Π_1} at early times (see Fig. 5.12) that is almost certainly related.

it is not definitively second order, as we can see from Fig. 5.17.

We have already mentioned that these results run counter to our intuition that numerical calculations of $S_{0.06}$, with its stronger gravitational fields and steeper scalar field gradients, should be less accurate than those of $S_{0.03}$, for any given resolution. Moreover, since $S_{0.03}$ is less centrally condensed, the observed lack of convergence cannot be due to any inherent non-smoothness of the solution relative to the scales of the grids being used. What has been omitted from the analysis thus far is consideration of the boundary conditions. As discussed in Sec. 2.4.5, Dirichlet boundary conditions were used for all of the calculations reported in this thesis: this means that the outer boundary values for all dynamical variables remain unchanged relative to their initial values as the evolution proceeds. As we argued in that section, Dirichlet conditions can provide a reasonable approximation to the true boundary conditions—which are to be imposed at spatial infinity—so long as the boundaries of the computational domain are sufficiently far removed from the region containing matter. This leads us to suspect that for $S_{0.03}$ the computational domain defined by $\text{bbox} = [-15, 15, -15, 15, -15, 15]$ is simply not large enough.

Indeed, a closer examination of the $t = 3.0$ frame of Fig. 5.8, as shown in Fig. 5.18, reveals a discontinuous wave of scalar matter that originates at the boundaries of the numerical domain and then propagates inwards. The jump in the solution is estimated to be of the order of 1% of the central value, ϕ_0 . Clearly, this relatively small, but spurious, boundary effect has a dramatic impact on the measured convergence of the solution. As emphasized at the end of the previous section, when computing convergence factors and independent residuals, we always assume that the continuum solution is smooth, and that our discrete approximations are similarly smooth (on the scale of the mesh). Here we see vividly what can happen to calculated measures of convergence when this assumption is violated.

The observed propagating discontinuity in the $S_{0.03}$ calculation can be immediately connected to the fact that the star has a relatively long tail which extends beyond $r = 15$. This in turn means that there is significant dynamical behaviour in the scalar field variables near the boundaries of the computational domain—in particular at the next-to extremal grid points. Since Dirichlet conditions are used, the time variation in the function values at these grid points generates the discontinuity which then travels inwards, eventually contaminating the entire solution.

For $S_{0.06}$ this effect is much less pronounced, as can be seen from Fig. 5.19. Thus, although there is without doubt some level of discontinuous behaviour at and near the boundaries in this case,

the resulting impact on the solution is well below the level of truncation error at the resolutions used. In other words, we can consider $S_{0.06}$ as providing another example of an initial dataset that is “sufficiently smooth”, given the computational domain, discretization parameters and boundary conditions.

To conclude this section we return to the issue of the anomalous behaviour of the independent residuals, I_{β^i} , that are associated with the shift vector components (see Figs. 5.13 and 5.14). Here we argue that the interpretation of the measured values of I_{β^i} requires special attention for the case of static initial data.

As discussed in Sec. 3.4, a static spherically symmetric spacetime which has a time coordinate adapted to the time symmetry—as ours is—implies that $\beta(t, x^i) \equiv 0$. Moreover, the finite difference approximations we use for the elliptic equations that govern β^i , in combination with the multigrid algorithm that we use to solve the resulting algebraic equations, guarantees that $[\beta^i]_{ijk}^n \equiv 0$ satisfies the discretized elliptic equations, provided that $[J^i]_{ijk}^n \equiv 0$. Here the $[J^i]_{ijk}^n$ are the grid values (at discrete time t^n) of the components of the 3-momentum, as defined by (2.177). Now, at the initial time $t = t^1 = 0$ we *do* have $[J^i]_{ijk}^1 \equiv 0$, so that $[\beta^i]_{ijk}^1 \equiv 0$ as well. However, the subsequent evolution on the Cartesian grid (x_i, y_j, z_k) generates non-sphericities in the scalar field variables, which leads to non-zero values for $[J^i]_{ijk}^n$ for any $t^n > 0$. This in turn results in $[\beta^i]_{ijk}^n \neq 0$ for any discrete time $t^n \neq 0$. This fact is clearly illustrated in the plots on the left hand side of Fig. 5.20, which show $\|\beta^i(t)\|_2$ from the $S_{0.06}$ computations performed at discretizations levels 5, 6 and 7. At any given resolution there is clearly approximately linear growth in the norms of all of the shift vector components. However, the plots on the right hand side of the figure—which display values of the norms that have been rescaled by factors 4^{l-5} —show that all of the components are converging to 0 as $O(h^2)$. Thus, although we do not yet understand why the independent residuals for the shift components fail to scale as expected for static data, it seems clear that the failure is not related to a convergence problem of the β^i themselves.

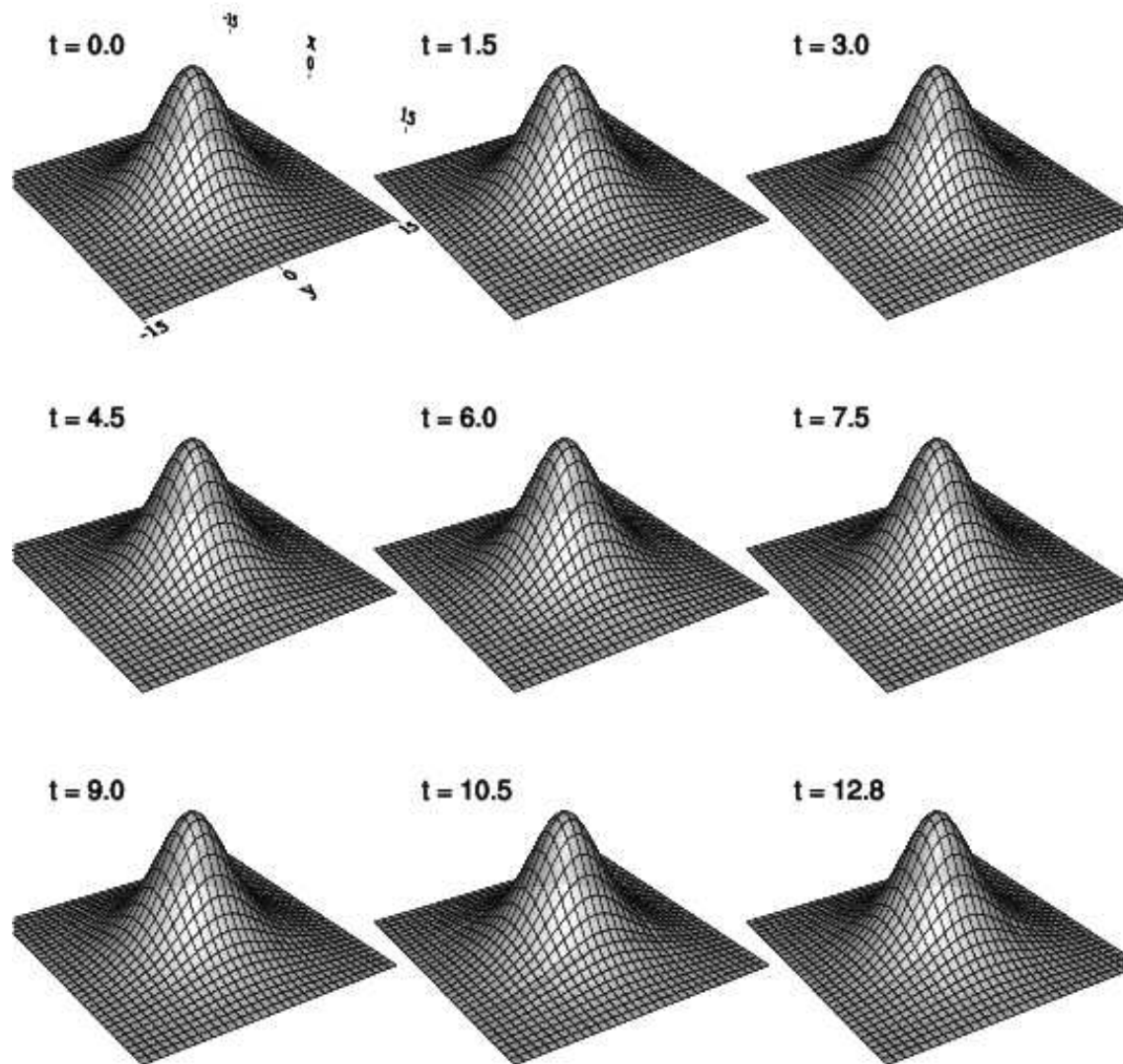


Figure 5.8: Snapshots of the time evolution of $|\phi(t, x, y, 0)|$ for the static boson star, $S_{0.03}$, as computed on the finest grid (`shape = [129, 129, 129]`). At least to the naked eye, the evolution appears time independent.

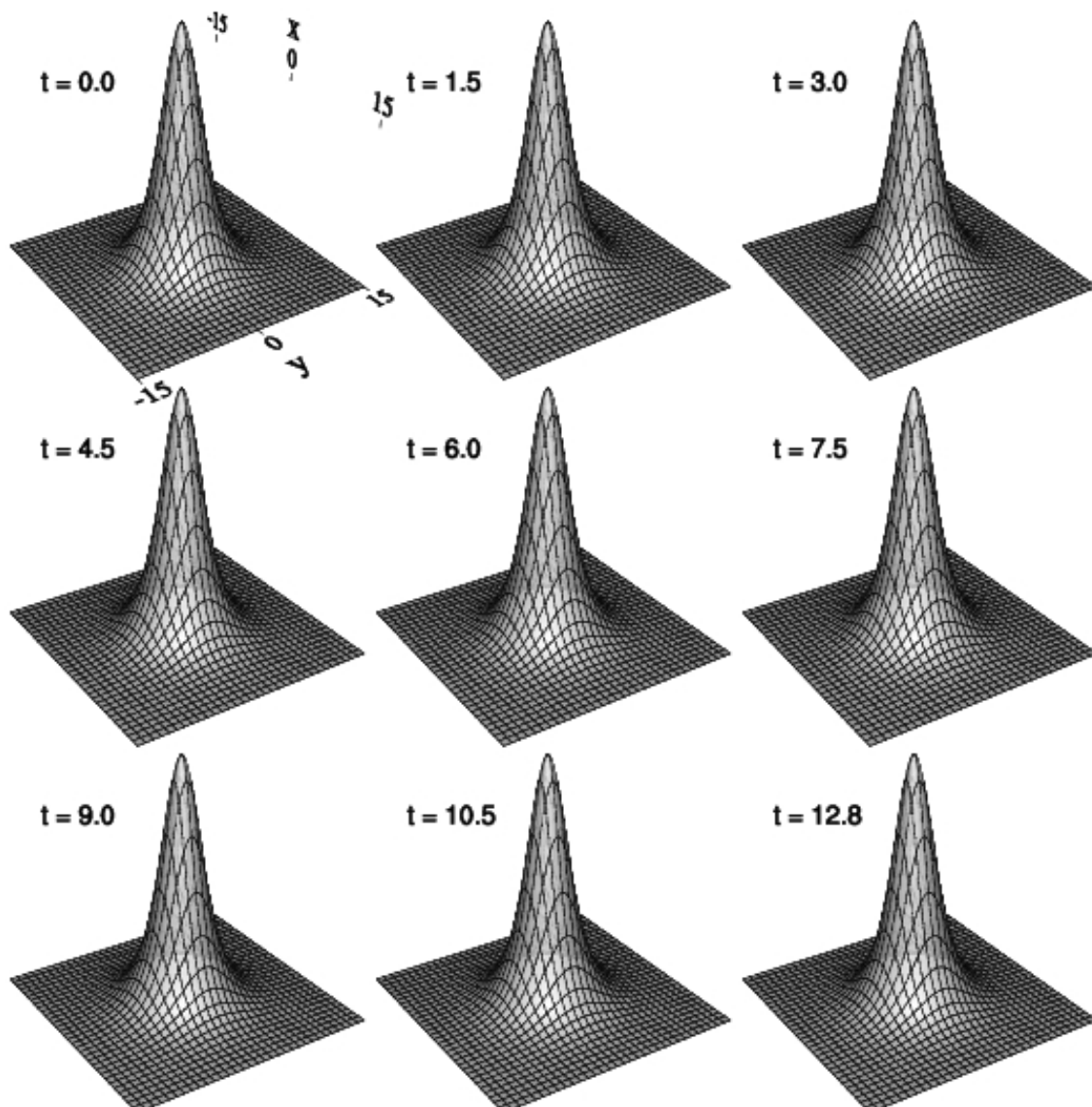


Figure 5.9: Snapshots of the time evolution of $|\phi(t, x, y, 0)|$ for the static boson star, $S_{0.06}$, as computed on the finest grid (`shape = [129, 129, 129]`). As for the data displayed in Fig. 5.8, the evolution appears time independent.

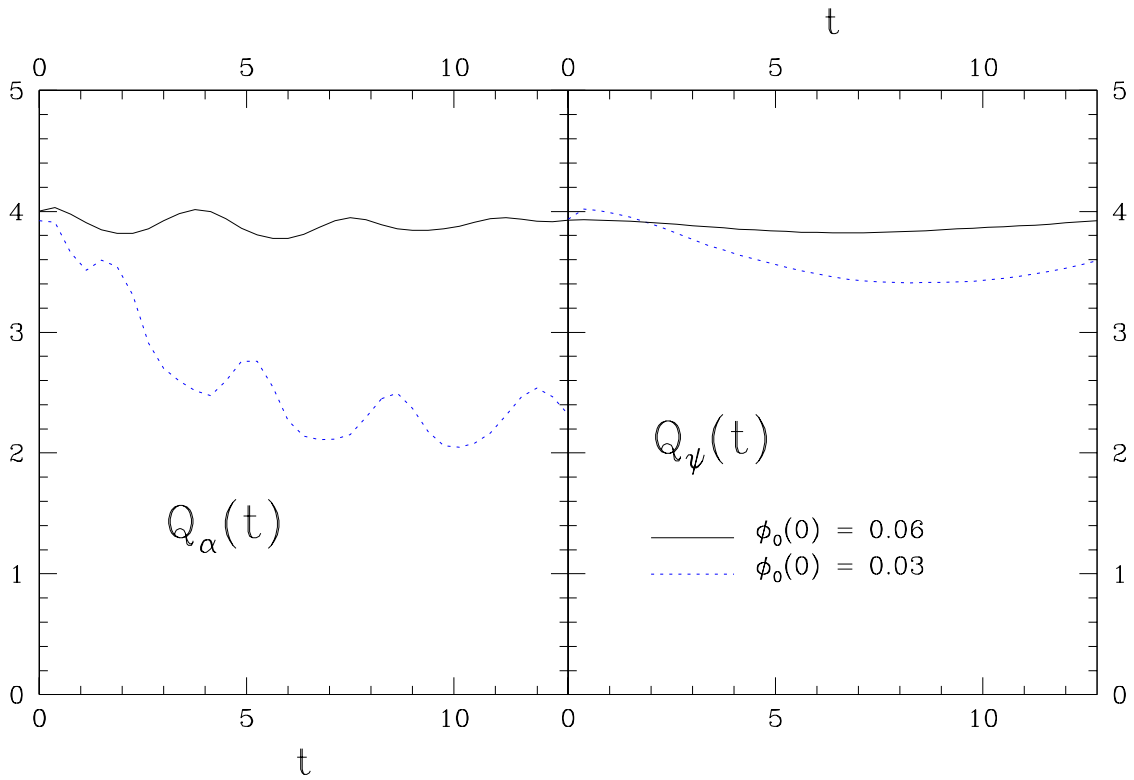


Figure 5.10: Convergence factors $Q^h(t)$ of α and ψ for evolutions of the two static configurations $S_{0.03}$ (solid black curves) and $S_{0.06}$ (blue dashed curves). The superior convergence of the $S_{0.06}$ data at the resolutions used (`shape = [33, 33, 33]`, `[65, 65, 65]` and `[129, 129, 129]`) is apparent.

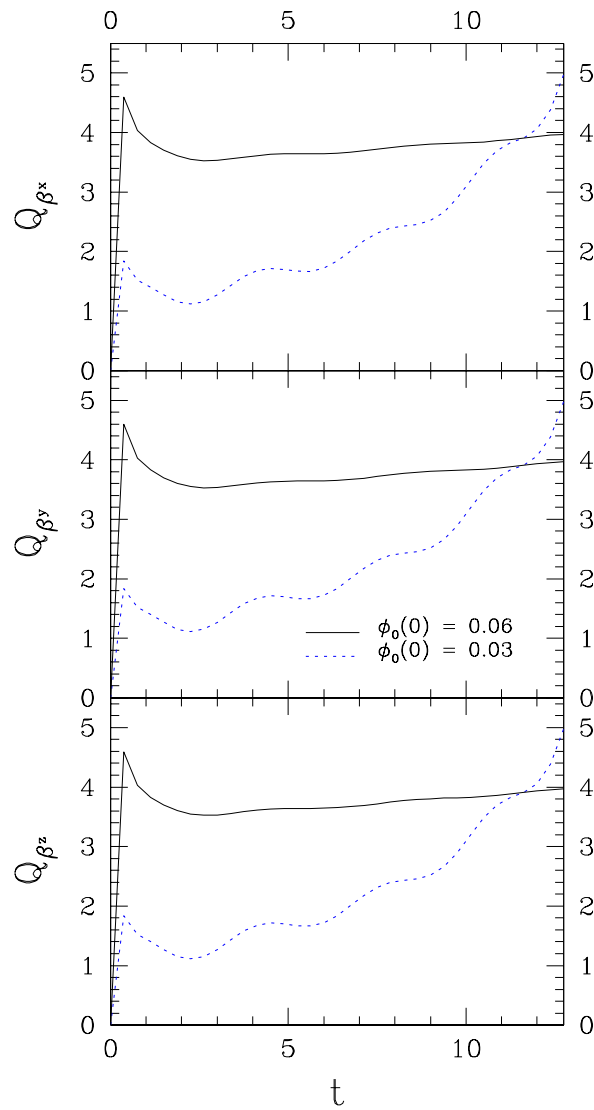


Figure 5.11: Convergence factors $Q^h(t)$ of β^k for evolutions of the two static configurations $S_{0.03}$ (solid black curves) and $S_{0.06}$ (blue dashed curves). As in Fig. 5.10, the superior convergence of the $S_{0.06}$ is clear.

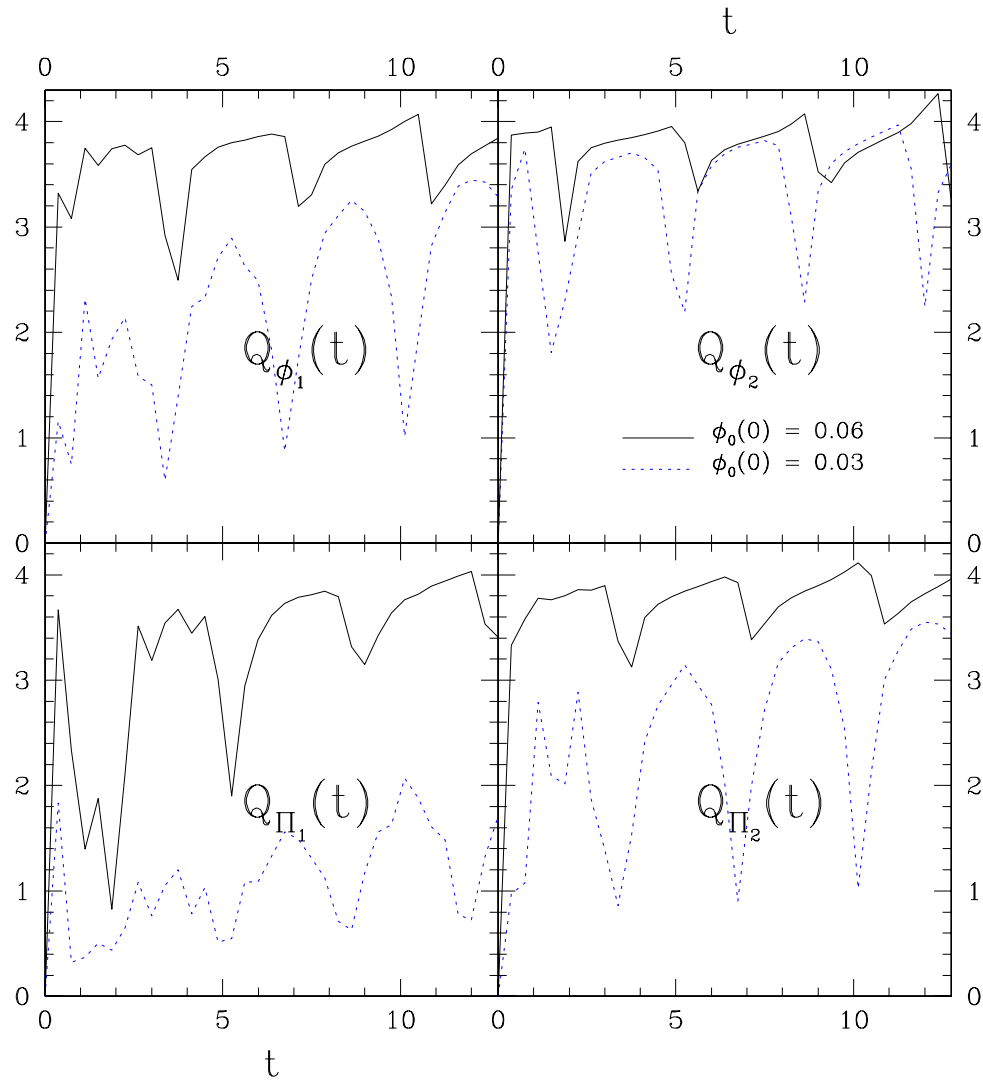


Figure 5.12: Convergence factors $Q^h(t)$ of ϕ_A and Π_A for evolutions of the two static configurations $S_{0.03}$ (solid black curves) and $S_{0.06}$ (blue dashed curves). As in the previous two figures, better convergence is observed in the $S_{0.06}$ data. There is, however, a noticeable “glitch” in the $S_{0.06}$ results for $Q_{\Pi_1}(t)$ in the interval $1 \leq t \leq 3$. Although we do not completely understand this behaviour at the current time, we suspect that spurious reflections from the boundaries may be to blame.

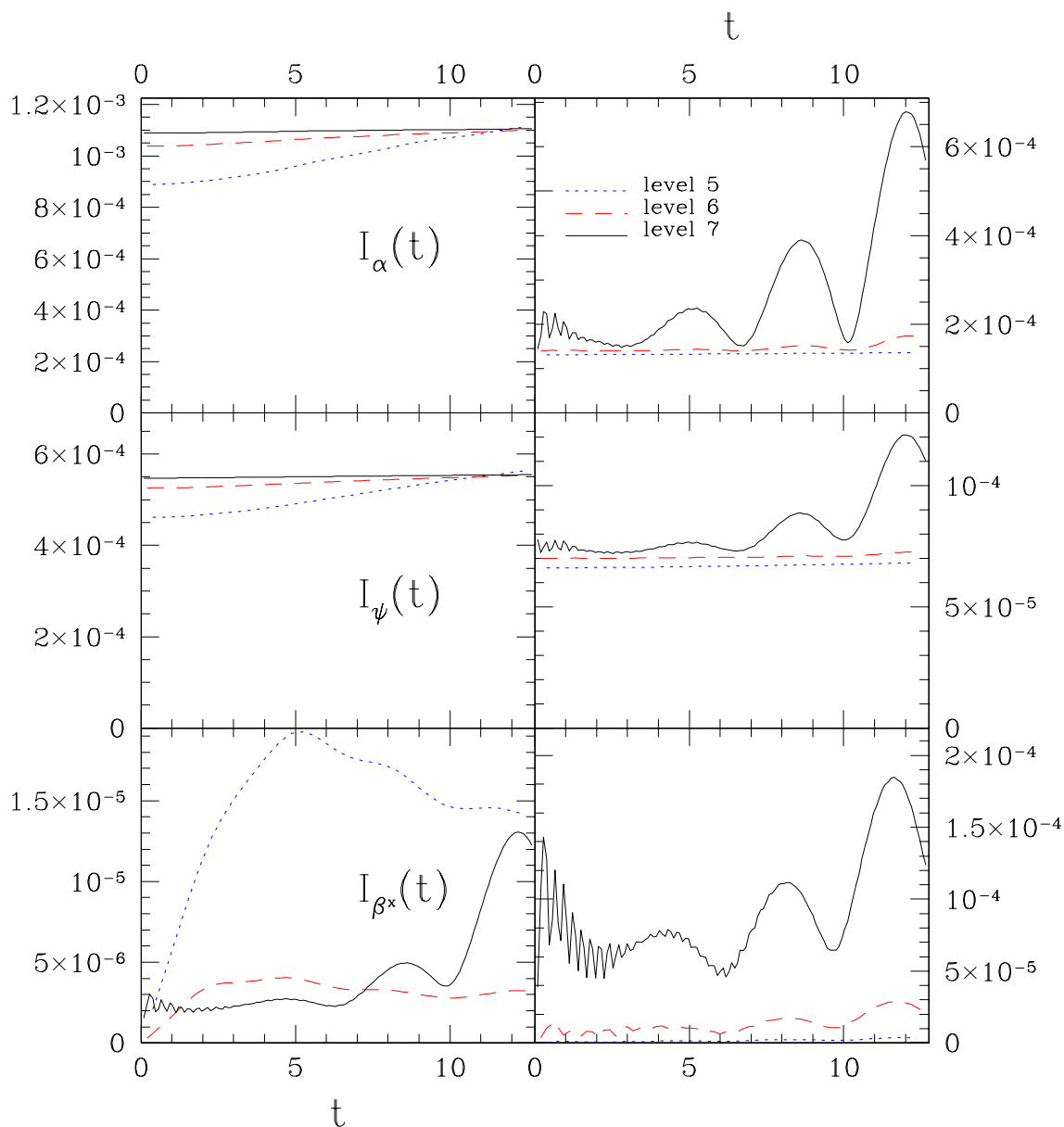


Figure 5.13: Rescaled independent residuals, $\|I(t)\|_2$, for α , ψ and β^x from evolutions of the two static configurations $S_{0.03}$ (left panels) and $S_{0.06}$ (right panels). As previously, the residuals were rescaled using (5.10) so that coincidence of curves computed at different levels of discretization corresponds to convergence of the residuals at the expected $O(h^2)$ rate. Apart from the residuals associated with β^x (as well as β^y and β^z , see Fig. 5.14), there is strong evidence that $I(t)$ is $O(h^2)$ for the geometric quantities. However, as discussed in the text, spurious boundary effects adversely impact the convergence of the independent residuals for the $S_{0.03}$ case.

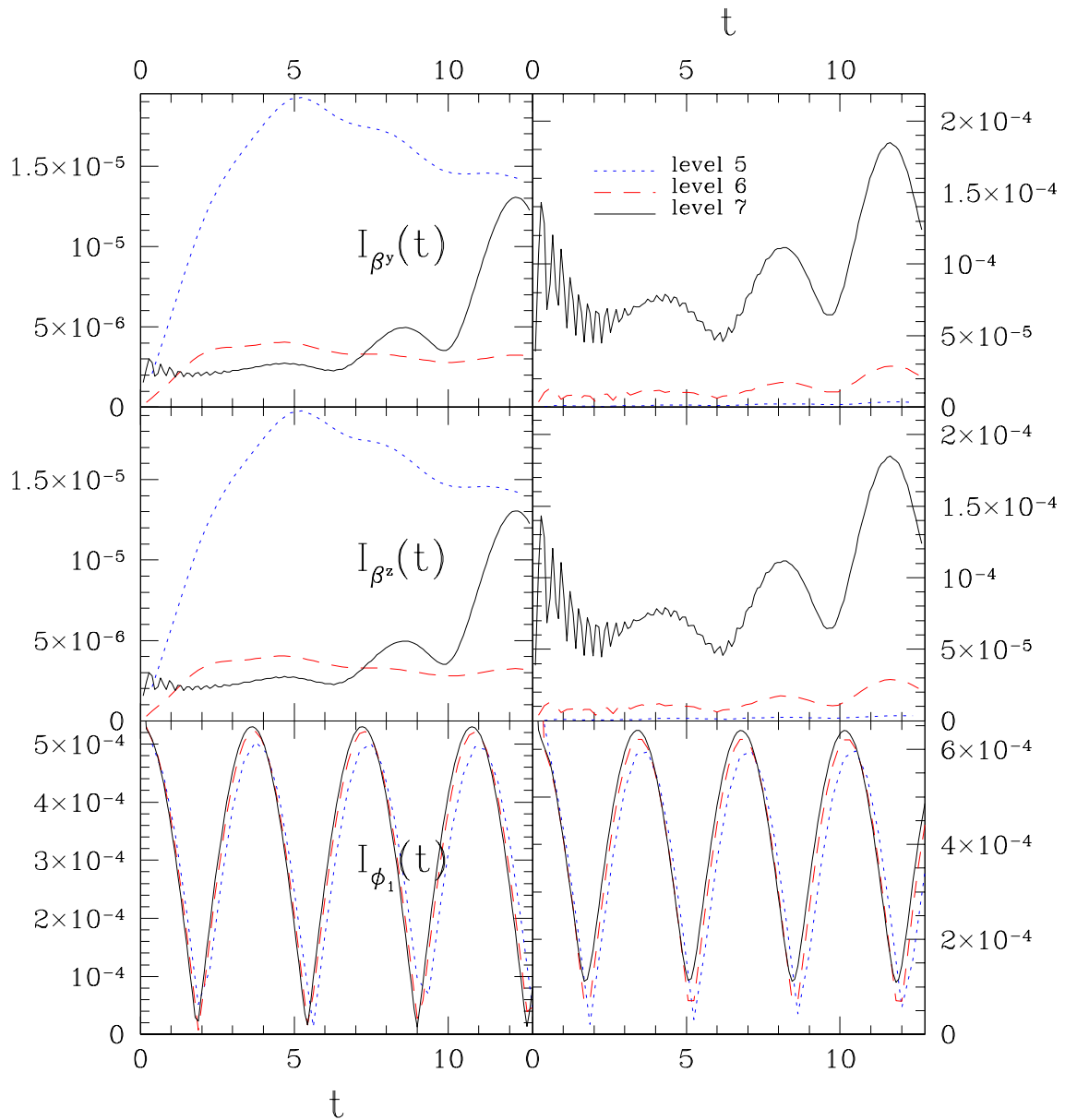


Figure 5.14: Rescaled independent residuals, $\|I(t)\|_2$, for β^y , β^z and ϕ_1 from evolutions of the two static configurations $S_{0.03}$ (left panels) and $S_{0.06}$ (right panels). Residuals were rescaled using (5.10) for β^y and β^z , and (5.11) for ϕ_1 . Although there is clear evidence for the expected $O(h)$ convergence of $I_{\phi_1}(t)$ for both calculations, for the case of the shift vector components (and as in the previous figure), the independent residuals are clearly *not* $O(h^2)$ quantities. However, as discussed in the text, the continuum static solutions have $\beta^i(t, x, y, z) \equiv 0$, and Fig. 5.20 provides clear evidence that for both calculations, all components of the shift vector *do* converge to 0 as $O(h^2)$. Combined with the results from the test that used generic initial data (Sec. 5.2), this suggests that the anomalous behaviour of $I_{\beta^i}(t)$ can be traced to the fact that the β^i vanish in the continuum limit, and, possibly to boundary effects. However, additional investigation is needed to provide a completely satisfactory explanation of the observed results.

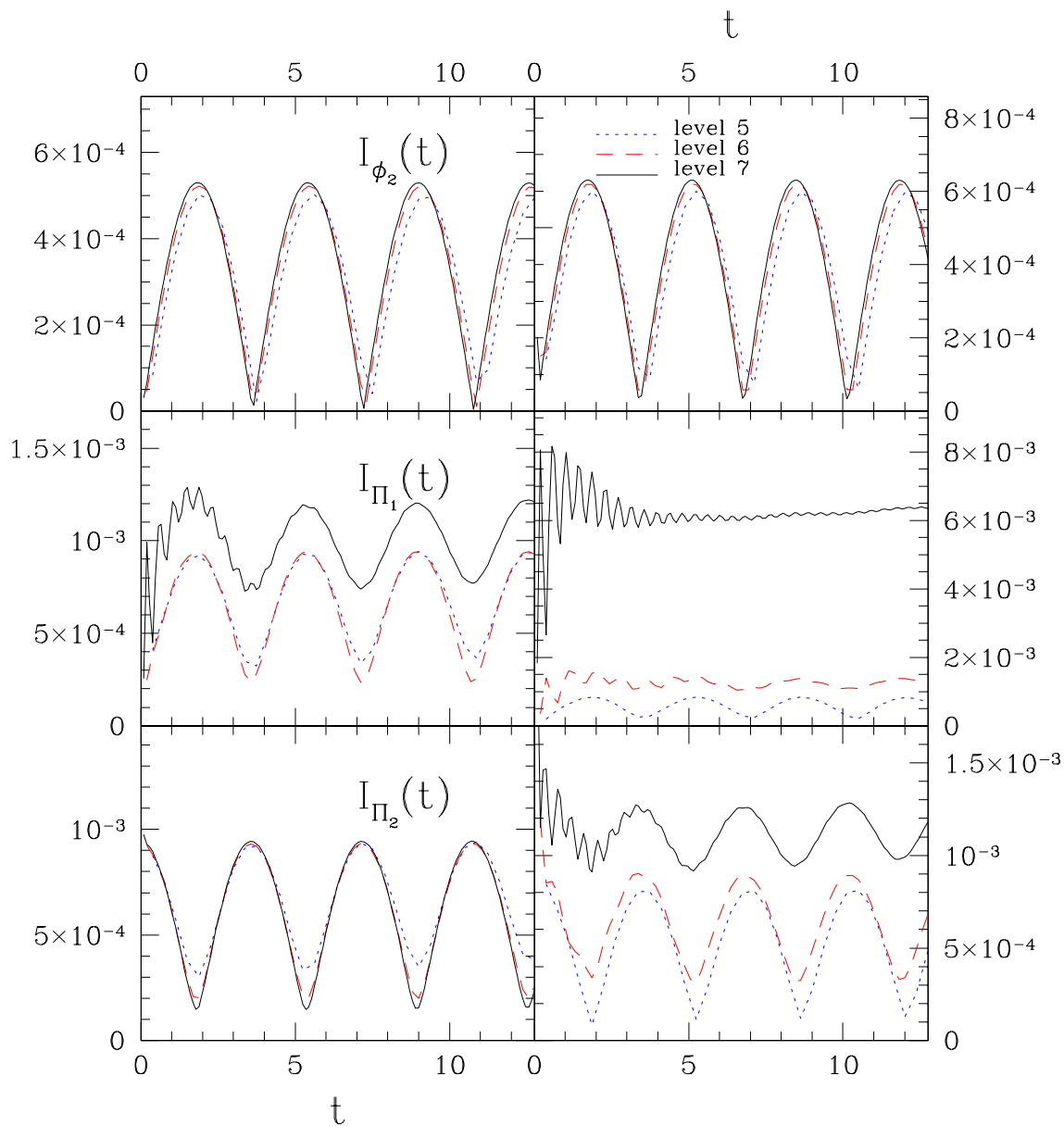


Figure 5.15: Rescaled independent residuals, $\|I(t)\|_2$, for ϕ_2 , Π_1 and Π_2 from evolutions of the two static configurations $S_{0.03}$ (left panels) and $S_{0.06}$ (right panels). Residuals were rescaled using (5.11) for ϕ_1 . Except for $I_{\Pi_1}(t)$, the expected $O(h)$ convergence of the residuals is observed for the $S_{0.06}$ data. As previously noted in the caption of Fig. 5.13, and as discussed in detail in the text, boundary effects produce a deterioration of the convergence of the $S_{0.03}$ residuals. We suspect the same effect is at play in the case of $I_{\Pi_1}(t)$ from the $S_{0.06}$ calculations.

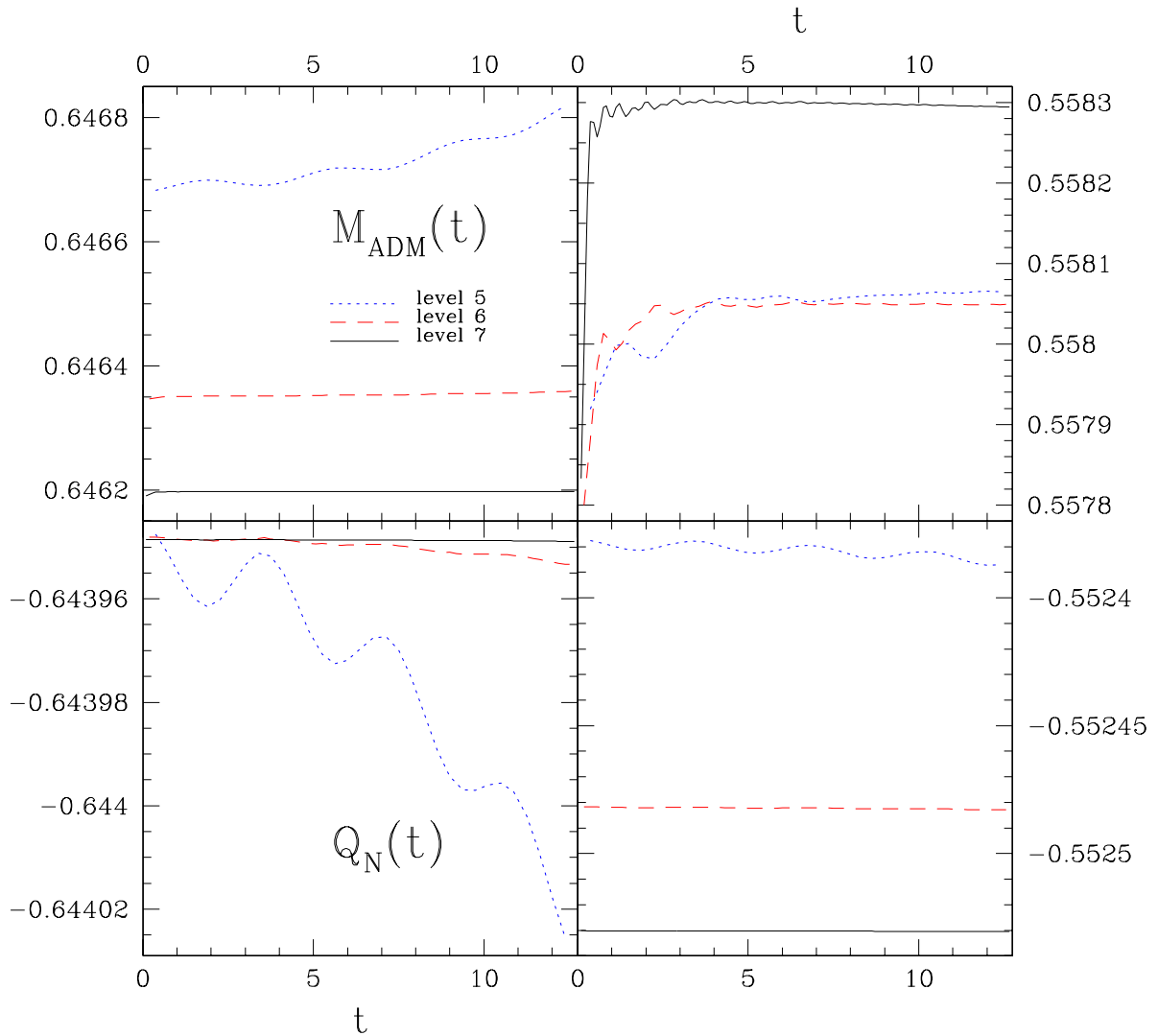


Figure 5.16: Plots of the ADM mass $M_{\text{ADM}}(t)$ and Noether charge $Q_N(t)$ for $S_{0.06}$ (left) and $S_{0.03}$ (right), and for the three resolutions used in the calculations. “Convergence to conservation” is observed in both quantities for $S_{0.06}$, but not for $S_{0.03}$.

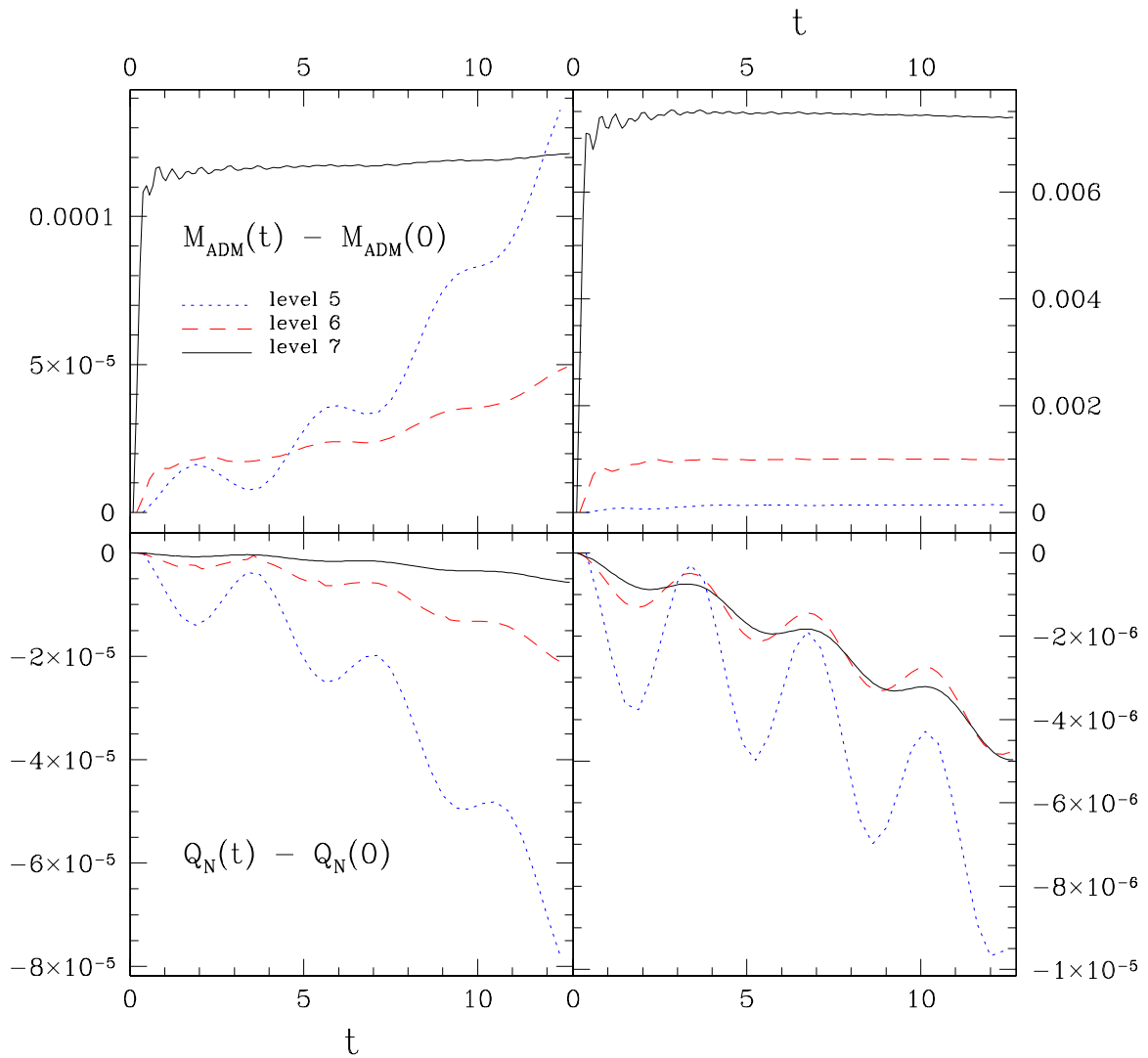


Figure 5.17: Plots of the rescaled deviations of the ADM mass $M_{\text{ADM}}(t)$ and Noether charge $Q_N(t)$ for $S_{0.06}$ (left) and $S_{0.03}$ (right), and for the three resolutions used in the calculations. $O(h^2)$ “convergence to conservation” is evident for the $S_{0.06}$ data, while for $S_{0.03}$, the convergence rate is indeterminate at best.

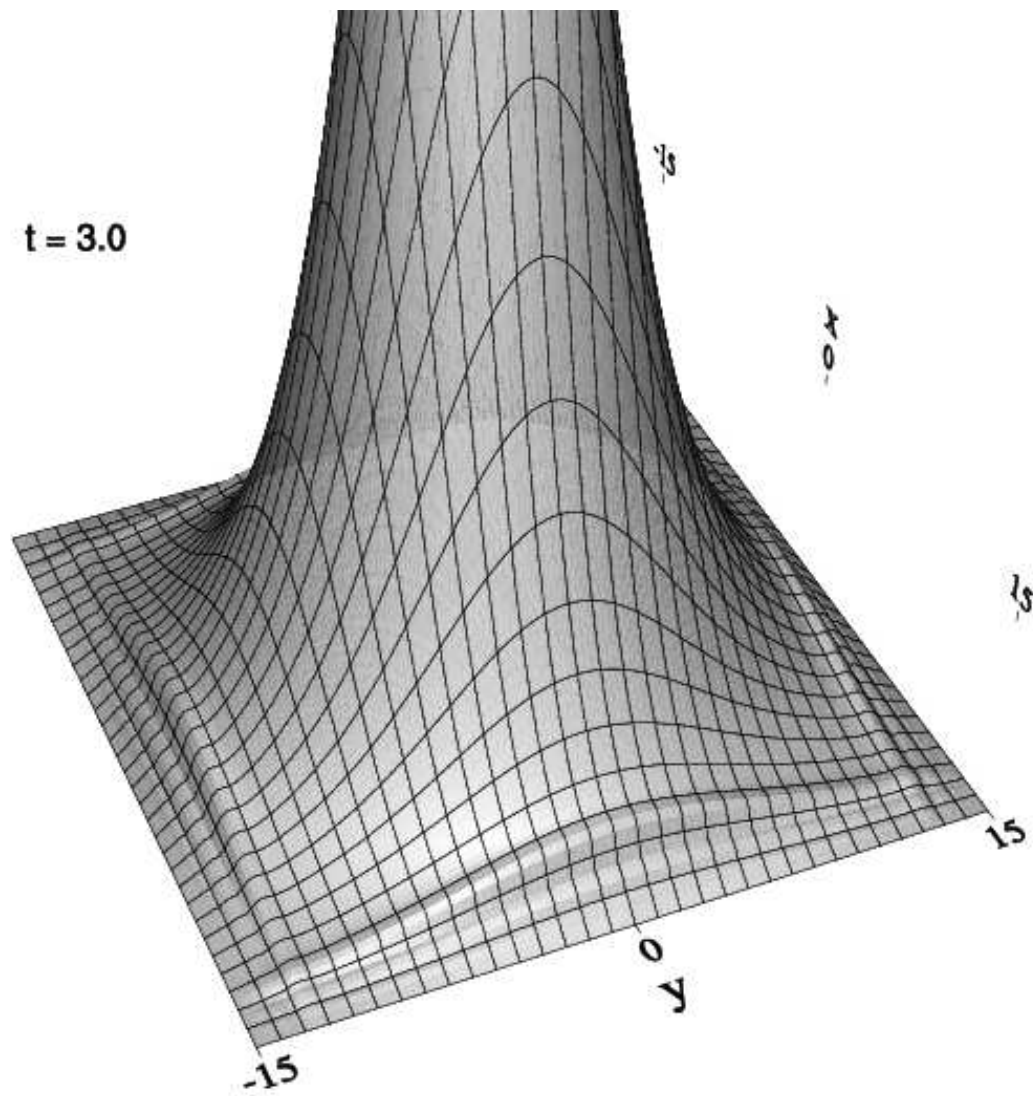


Figure 5.18: Magnified view of $|\phi(3.0, x, y, 0)|$ (i.e. $t = 3.0$) from the $S_{0.03}$ evolution. The jump in the solution is estimated to be of the order of 1% of $\phi_0 \equiv |\phi(0, 0, 0, 0)|$.

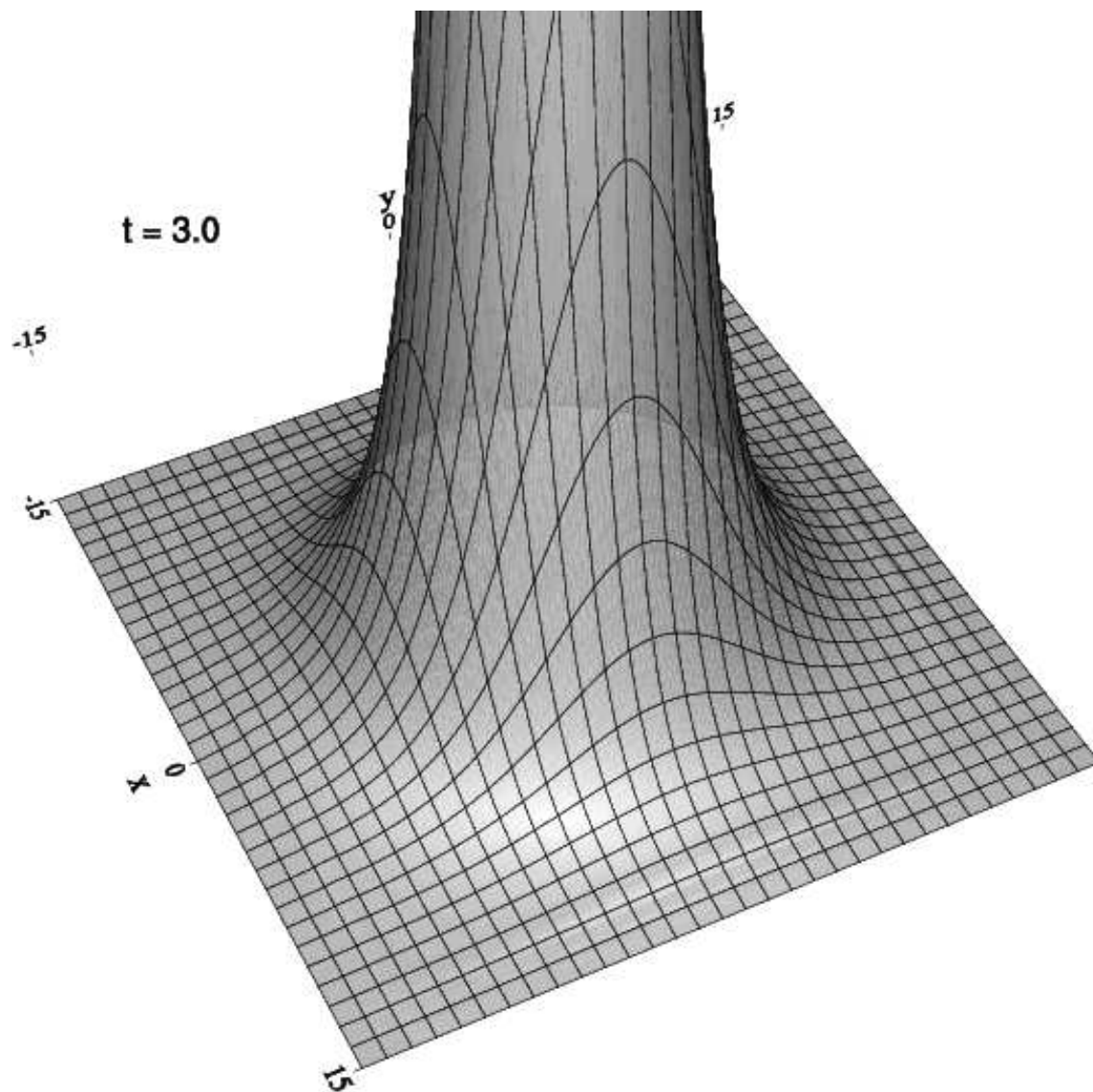


Figure 5.19: Magnified view of $|\phi(3.0, x, y, 0)|$ from the $S_{0.06}$ evolution. Here it virtually impossible to visually detect the discontinuity in the scalar field near the computational boundaries.

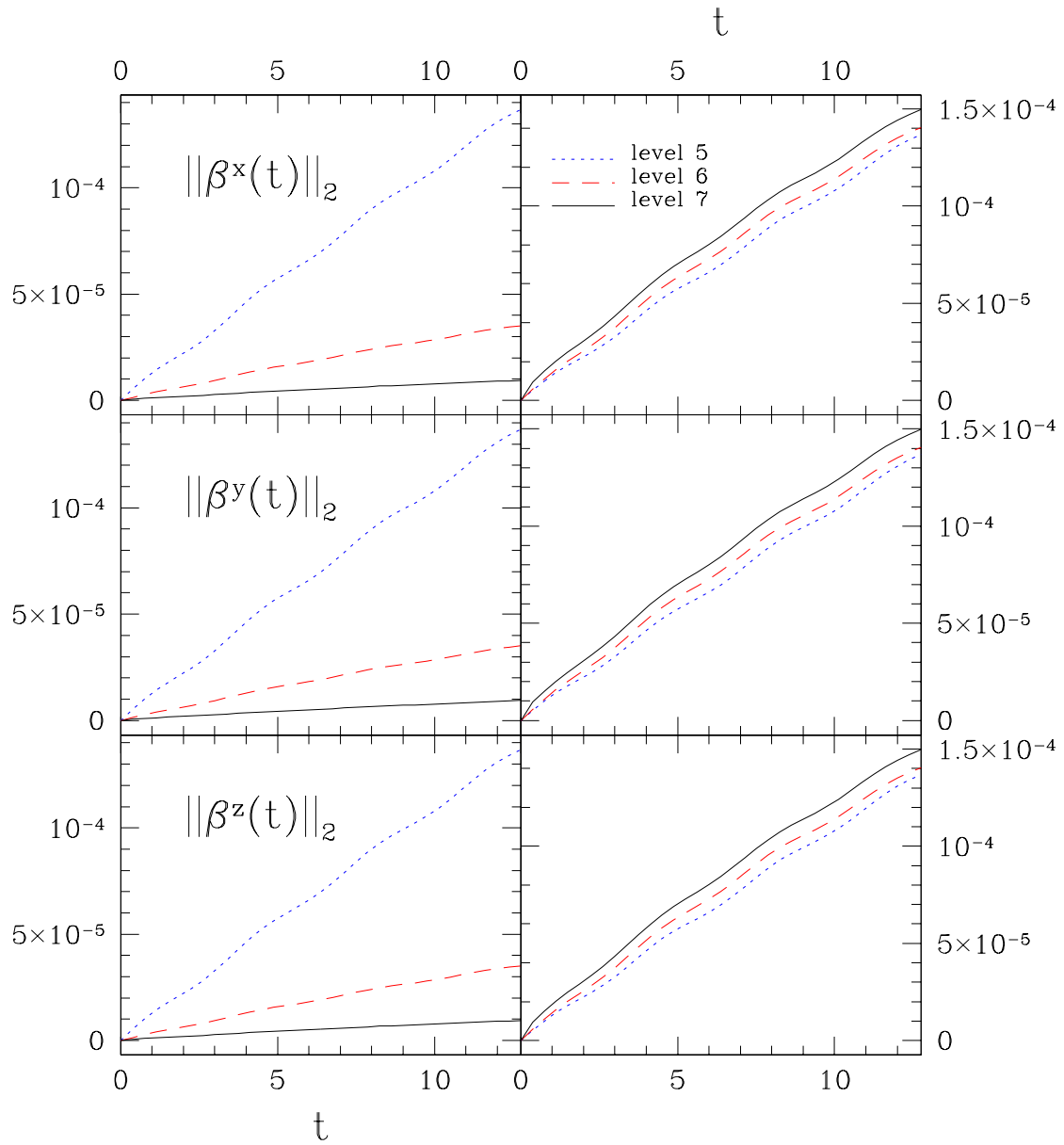


Figure 5.20: Left panels: Spatial l_2 -norms of the shift vector components, $\|\beta^x(t)\|_2$, $\|\beta^y(t)\|_2$ and $\|\beta^z(t)\|_2$ from the $S_{0.06}$ calculation. Right panels: Values of these l_2 norms rescaled using (5.10). These plots provide strong evidence that despite the issues with the independent residuals $I_{\beta^i}(t)$ displayed in previous figures, and discussed in the text, all of the shift vector components are converging to the continuum limit, $\beta^i(t, x, y, z) \equiv 0$, as $O(h^2)$.

5.4 Single Boosted Boson Star

In the previous section we tested our code using spherically-symmetric initial datasets representing single boson stars. Modulo the problems identified when using stars not sufficiently compact with respect to the size of the computational domain, we observed satisfactory convergence of the numerical results to static continuum solutions, which in addition to providing a further test of the PDE code, also established that our procedure for computing the boson star initial data was working. However, there is one significant piece of our overall process for setting up initial data for boson stars that has not yet been tested. To evolve configurations describing stars with non-zero initial velocities—as is done in the next two sections—then once a spherically symmetric solution for a star has been generated and interpolated to the 3D numerical domain (Sec. 3.6), an (approximate) Lorentz boost is applied (Sec. 3.7). This section thus reports the results of the same battery of tests used in the previous section, but now applied to the special case of a boosted spherically symmetric solution.

We perform the tests on the star $S_{0.06}$ and maintain the same grid parameters employed in the previous section: `bbox` = $[-15, 15, -15, 15, -15, 15]$, using three levels of resolution ranging from `shape` = $[33, 33, 33]$ to `shape` = $[129, 129, 129]$. The star is boosted in the x -direction with a boost parameter $v_x = -0.1$. Fig. 5.21 displays some snapshots of $|\phi(t, x, y, 0)|$ from an evolution computed on the finest grid. Careful study of the sequence shows that the star is slowly (relative to the time scale of the sequence) moving across the computational domain from right to left (i.e. in the $-x$ direction). A rough numerical estimate of the rate of change of the star’s coordinate position with respect to coordinate time yields $\Delta x/\Delta t \approx -0.1$, consistent with expectations.

Convergence factors, $Q^h(t)$, and independent residual norms, $\|I(t)\|_2$, are shown in the left and right panels, respectively, of Figs. 5.22, 5.23 and 5.24. Once again, these results provide strong evidence that the numerical solution is converging to a continuum solution of the PDEs governing our model, and at the expected $O(h^2)$ rate. Additionally, Fig. 5.25 and Fig. 5.26 indicate that both the ADM mass and the Noether charge are being conserved as $h \rightarrow 0$, and that the deviations from conservation are also $O(h^2)$.

We hope that the results from the extensive and comprehensive set of tests that we have described thus far in this chapter will have convinced the reader of the following:

- That we have consistently discretized the equations of motion for our model to $O(h^2)$ in both

space and time.

- That our implementation of the resulting finite difference approximation is correct, and produces convergent (and thus, implicitly, stable) results.
- That our procedures for computing and boosting single boson stars are also correct.
- That convergence testing will reveal situations where the numerical solution has developed significant non-smoothness on the scale of the mesh.

We now proceed to a discussion of the key numerical calculations in this thesis: those that involve two boson stars in interaction with one another.

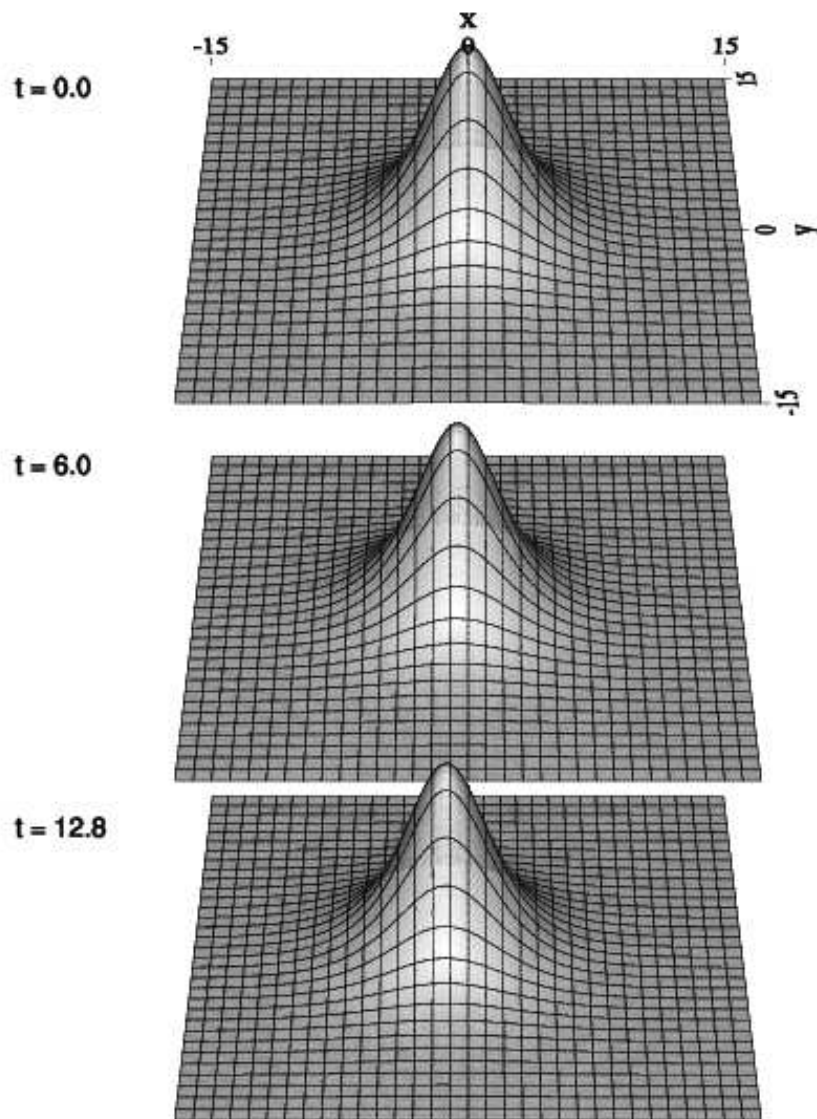


Figure 5.21: Time evolution of a single $S_{0.06}$ star which has been boosted in the x -direction with a velocity parameter, $v_x = -0.1$ (see Sec. 3.7 for a full discussion of the algorithm used to determine the boosted data). Plotted are snapshots of $|\phi(t, x, y, 0)|$ for $t = 0.0, 6.0$ and 12.8 . Careful inspection of these frames reveals that the star is slowly moving in the $-x$ direction. A rough numerical estimate of the rate of change of the star's coordinate position with respect to coordinate time yields $\Delta x / \Delta t \approx -0.1$, consistent with expectations.

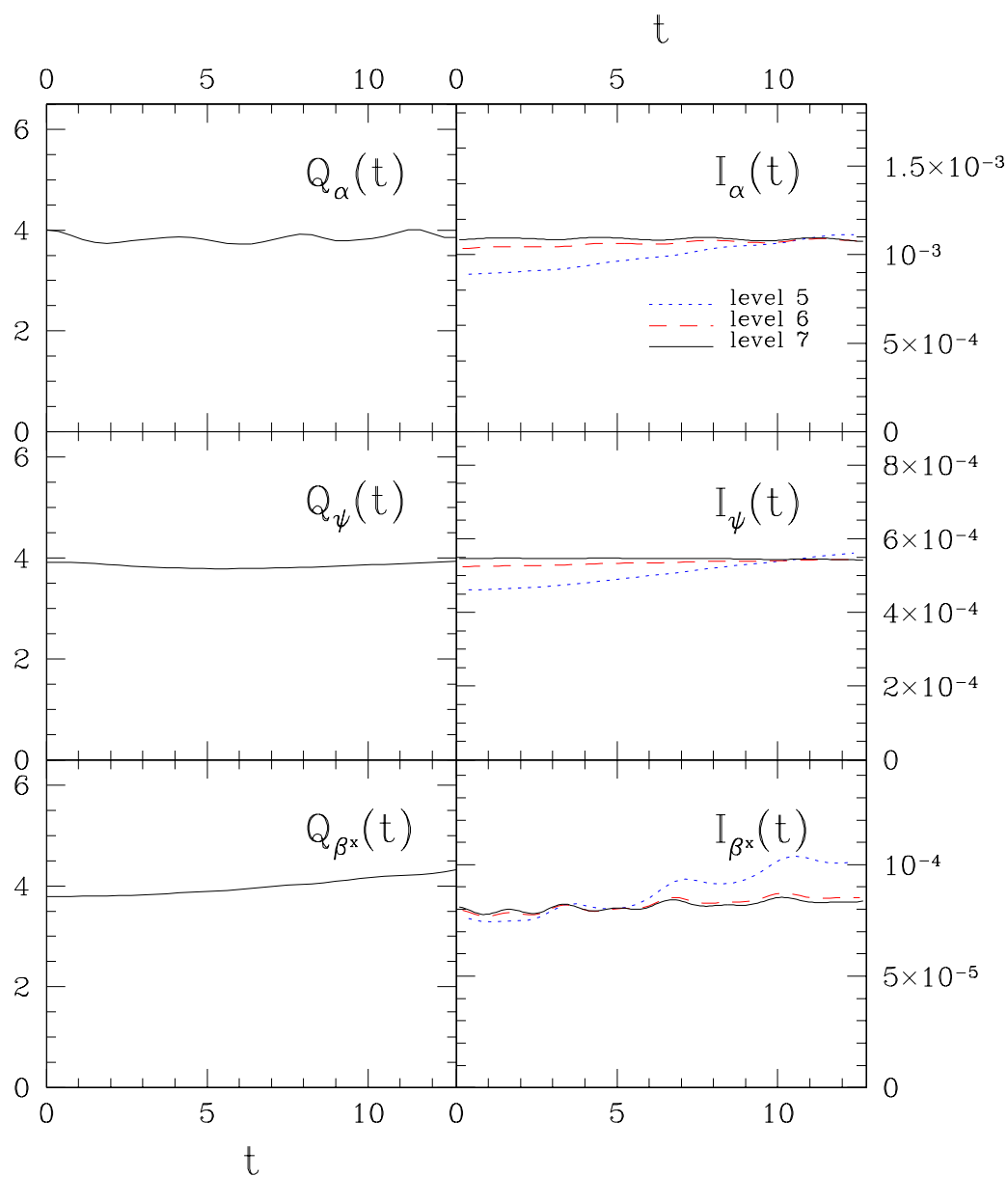


Figure 5.22: $Q^h(t)$ (left) and rescaled $\|I(t)\|_2$ (right) for α , ψ and β^x from the boosted $S_{0.06}$ experiment. Values of the independent residuals norms $\|I(t)\|_2$ have been rescaled using (5.10). These plots provide strong evidence of $O(h^2)$ convergence for all three of the geometric variables.

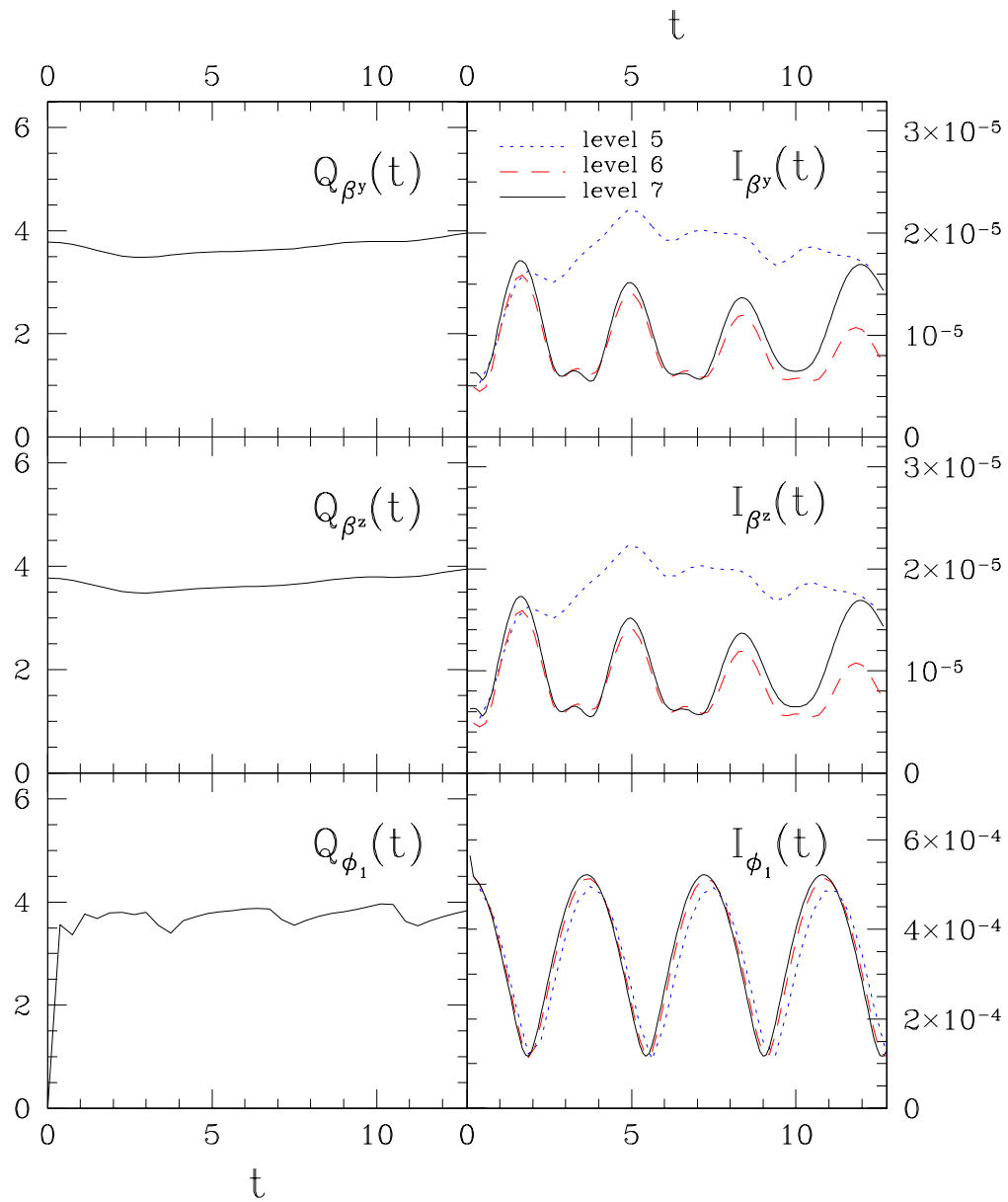


Figure 5.23: $Q^h(t)$ (left) and rescaled $\|I(t)\|_2$ (right) for β^y , β^z and ϕ_1 from the boosted $S_{0.06}$ calculations. Values of the independent residual norms $\|I(t)\|_2$ have been rescaled using (5.10) for $I_{\beta^y}(t)$ and $I_{\beta^z}(t)$ and (5.11) for $I_{\phi_1}(t)$. These plots provide strong evidence of $O(h^2)$ convergence for all of β^y , β^z and ϕ_1 .

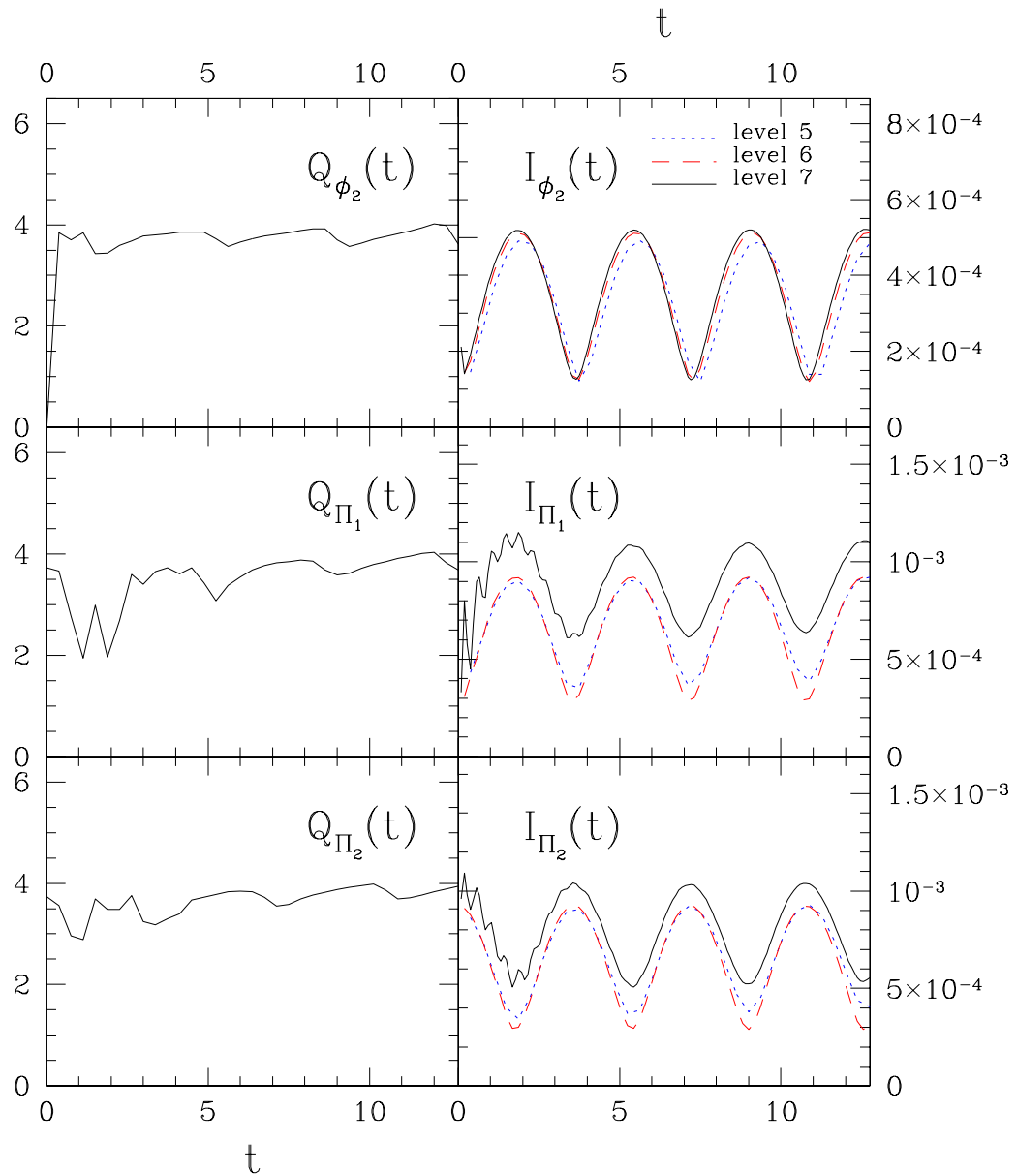


Figure 5.24: $Q^h(t)$ (left), and rescaled $\|I(t)\|_2$ (right) for β^y , β^z and ϕ_1 from the boosted $S_{0.06}$ calculations. Values of the independent residual norms $\|I(t)\|_2$ have been rescaled using (5.11). Here we observe some irregularities in the norms of $I_{\Pi_1}(t)$ and $I_{\Pi_2}(t)$ that we suspect may again be due to our use of Dirichlet conditions on a computational domain whose extent is comparable to the size of the star. Nonetheless, these plots still provide strong evidence for $O(h^2)$ convergence of the scalar field variables.

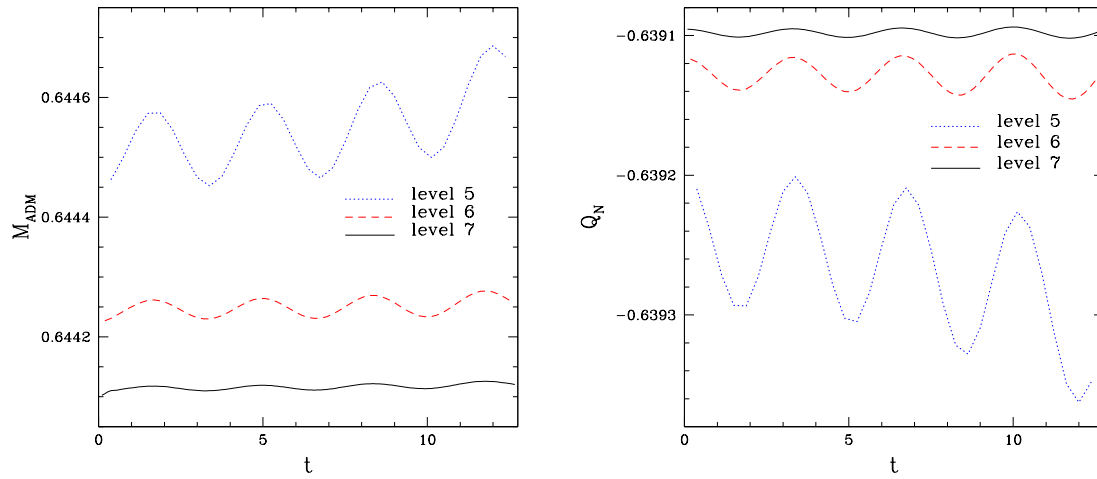


Figure 5.25: ADM mass $M_{\text{ADM}}(t)$ (left) and Noether charge $Q_N(t)$ (right) for the boosted $S_{0.06}$ calculation. These plots suggest that both the ADM mass and Noether charge are conserved as $h \rightarrow 0$.

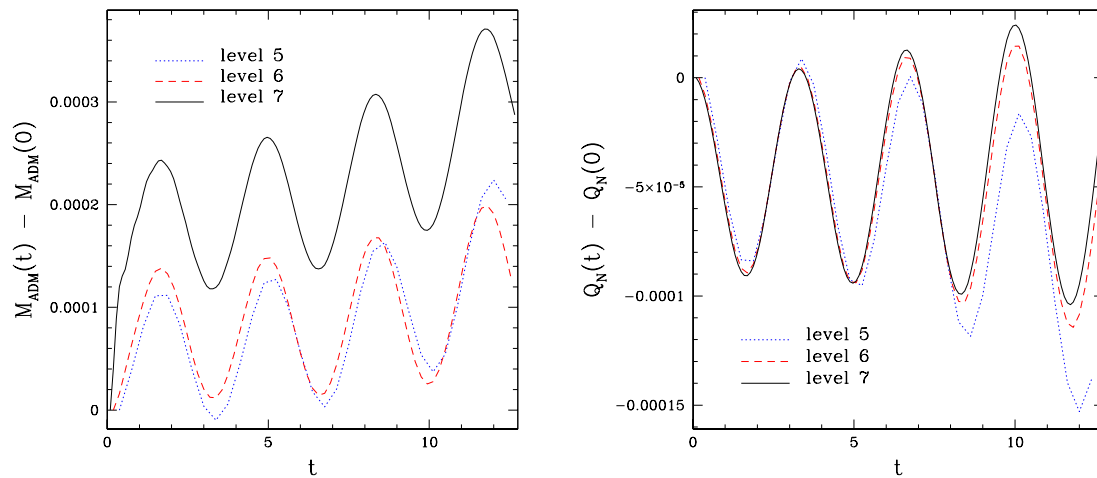


Figure 5.26: Rescaled deviations of ADM mass (left) and Noether charge (right), relative to their initial values, for the boosted $S_{0.06}$ calculation. The deviations were rescaled using (5.13) and (5.14). Although it seems evident that the computed Noether charge is converging to a conserved value as $O(h^2)$, the convergence rate for M_{ADM} cannot be determined with certainty using the available data.

5.5 Head-on Collision of Two Boson Stars

In this section we report the results from an numerical experiment describing the head-on collision of two boson stars which are initially boosted towards one another. We note that head-on collisions of self-gravitating boson stars have been investigated previously by other researchers. The prior studies include work by: Choi [173, 174] who implemented an axisymmetric Newtonian code,⁴² Balakrishna [6], using a 3D fully relativistic code, Lai [128] who used a fully general relativistic, but axisymmetric code, Palenzuela *et al* [9], using a 3D general-relativistic code, and most recently Choptuik and Pretorius [130] who used an axisymmetric general-relativistic code. One particularly interesting effect that Choi first demonstrated for the Newtonian case is so-called “solitonic behaviour”, wherein upon collision the stars interpenetrate and effectively pass through one another, emerging from the interaction relatively unscathed. Subsequently, this behaviour has also been seen in the general relativistic calculations reported in [128] and [130]. Here we show that the solitonic effect is also present within the context of our current model, which in some loose sense is an intermediary between the purely Newtonian and fully general relativistic calculations.

The experiment was prepared as follows. First, we adopted a discrete domain defined by $\text{bbox} = [-50, 50, -25, 25, -25, 25]$. Note that this yields a computational volume with an $x : y : z$ aspect ratio of $2 : 1 : 1$, and that the coordinate extent in the y and z directions is roughly double that used in the calculations reported in previous sections. This last point is important since the star configuration that we used in the collision is relatively extended, so we needed to be careful to ensure that our calculations were not significantly affected by the boundary-discontinuity issue described in Sec. 5.3.

Next we generated initial data for the star $S_{0.02}$ (i.e. a star with central field modulus $\phi_0 = 0.02$) and the star $S_{0.03}$. Both solutions were then interpolated onto the 3D numerical domain with centres at $(25, 0, 0)$ and $(-25, 0, 0)$, respectively. The stars were then given boosts in the x direction having equal magnitudes but opposite senses: $v_x^1 = -0.4$ and $v_x^2 = 0.4$. This of course results in an initial configuration in which the stars are already approaching one another with a significant relative velocity.

The computations performed in this section were made using three different grid resolutions. Let the grids be denoted by G_1 , G_2 , and G_3 , where G_1 is the coarsest and G_3 is the finest mesh,

⁴²This required the solution of the Schrödinger equation for the complex scalar field, and the Poisson equation for the Newtonian gravitational field.

the associated `shape` parameters are given by:

$$G_1 : \quad \text{shape}_1 = [129, 65, 65], \quad (5.18)$$

$$G_2 : \quad \text{shape}_2 = [161, 81, 81], \quad (5.19)$$

$$G_3 : \quad \text{shape}_3 = [193, 97, 97]. \quad (5.20)$$

Fig. 5.27 shows the time evolution of $|\phi(t, x, y, 0)|$, $0 \leq t \leq 125$ as calculated on the grid G_3 , using the initial data described above. Significant direct interaction of the stars begins at $t \approx 40$, and by $t = 55$ it is essentially impossible to identify two distinct objects. The period of strong interaction—during which “interference patterns” are clearly visible—persists until $t \approx 100$. The stars then emerge from the collision with their initial shapes roughly preserved, and continue to propagate until the end of the calculation.

During the period of interaction, the scalar field modulus reaches a maximum value $|\phi(t, x, y, z)| \simeq 0.059$. In addition, the stars come out of the collision with an estimated average coordinate velocity $\Delta x / \Delta t \sim 0.24$. The rescaled l_2 norms of the independent residual, $I_{\phi_1}(t)$, for the scalar field component, ϕ_1 , are plotted in Fig. 5.28. They provide validation of the computation since they are converging at the expected $O(h)$ rate.

Given the previous observations of the solitonic nature of head-on boson stars in fully general relativistic cases, the results of this experiment can be interpreted as providing qualitative evidence that the CFA is capturing at least some of the essential physics described by a solution of the full Einstein-Klein-Gordon equations. However, we must again note that Newtonian calculations also yield the same type of behaviour. Thus, without some sort of direct comparison with fully general relativistic results, our computations should be viewed as providing a relatively weak validation of the suitability of the CFA as a replacement for the Einstein equations in the treatment of head-on collisions of boson stars.

Finally, we note that the initial conditions used in this section have $y \rightarrow -y$ and $z \rightarrow -z$ symmetries. As we have mentioned previously, these were *not* exploited in this (or any other) calculation. However, in this case we *have* estimated how well the reflection symmetry $y \rightarrow -y$ is maintained during the evolution. Results of this test are shown in Fig. 5.29, which suggest that, at any of the resolutions used, the maximum deviation from exact symmetry in a typical dynamical variable— $|\phi|$ in this instance—was about 1 part in 10^6 .

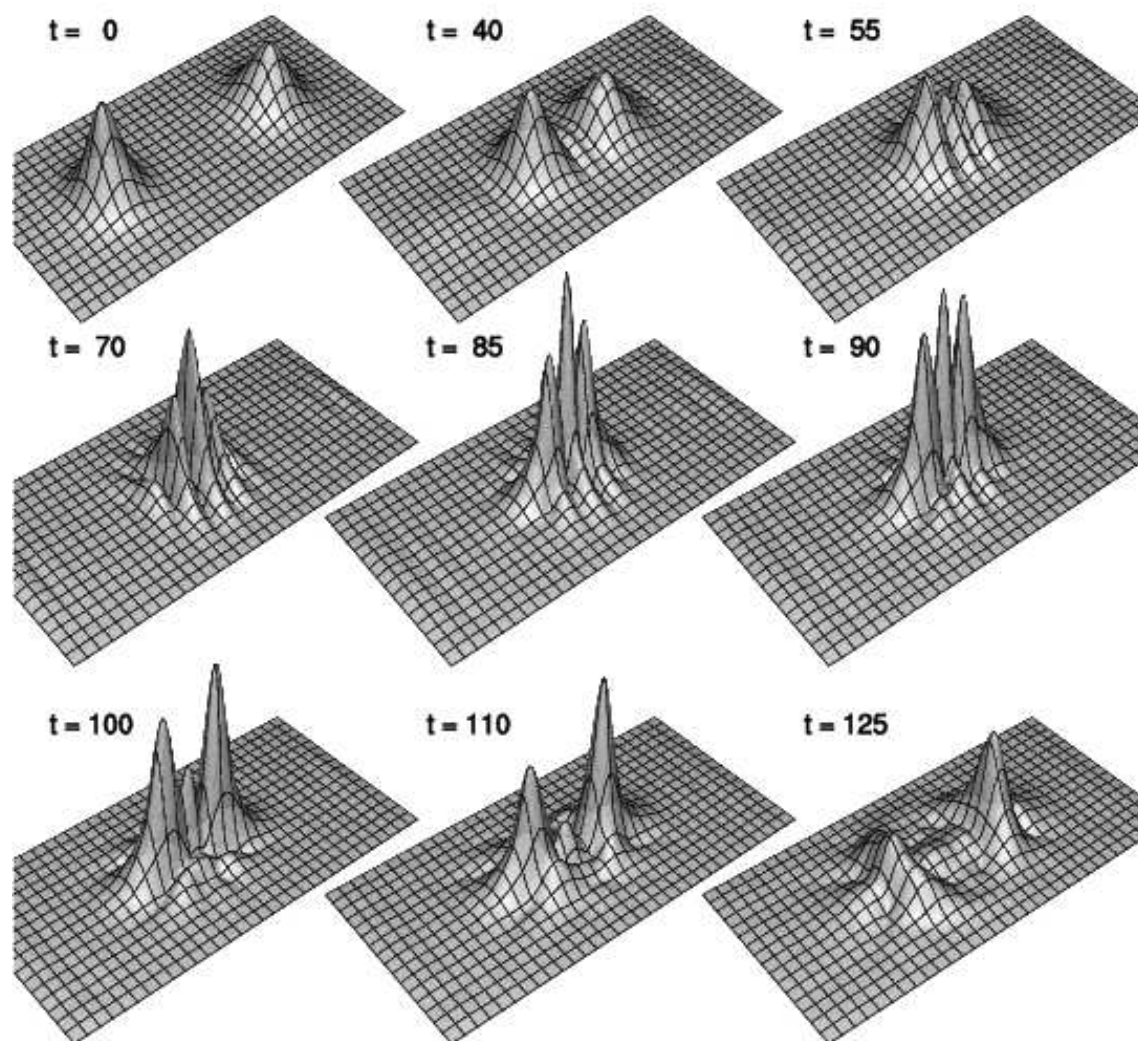


Figure 5.27: Head-on Collision of Boson Stars: Solitonic Dynamics. This figure shows a series of snapshots from the time evolution of $|\phi(t, x, y, 0)|$, for $0 \leq t \leq 125$. The initial configuration describes two different boson stars, $S_{0.02}^1$ and $S_{0.03}^2$, centred at $(25, 0, 0)$ and $(-25, 0, 0)$, respectively, and boosted with velocity parameters $v_x^1 = -0.4$ and $v_x^2 = 0.4$. At time $t \approx 40$, the stars are interacting substantially, and by $t = 55$ it is virtually impossible to identify the individual stars. During this interaction epoch “interference patterns” that are characteristic of boson star collisions of this type are evident. By $t \approx 100$ the period of interaction has essentially concluded and distinct star-like configurations can again be identified. The stars emerge with their shape roughly preserved—thus exhibiting “solitonic” dynamics—and continue to propagate in the directions of their respective original boosts.

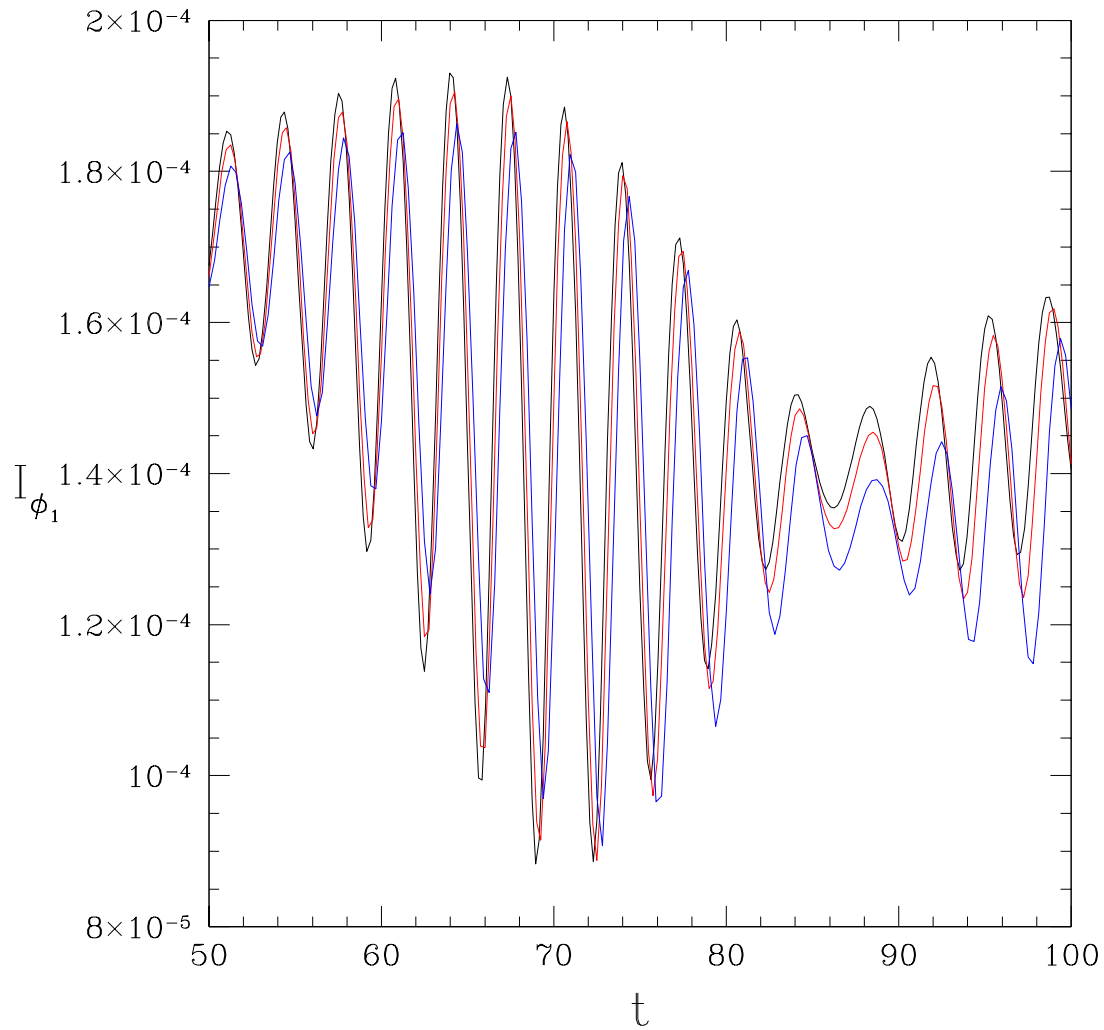


Figure 5.28: This figure shows the rescaled l_2 norms of the independent residuals, $\|I_{\phi_1}(t)\|_2$, from the calculation of the head-on collision of $S_{0.02}^1$ and $S_{0.03}^2$. The plot focuses on the time interval $50 \leq t \leq 100$, when the interaction between the stars is strongest. The blue, red and black curves correspond to runs using grids G_1 , G_2 and G_3 (see text), which had mesh scales h_1 , h_2 and h_3 , respectively. The red and black values have been rescaled by factors of h_1/h_2 and h_1/h_3 respectively, and the near-coincidence of the curves shows that the independent residuals are $O(h)$, as expected.

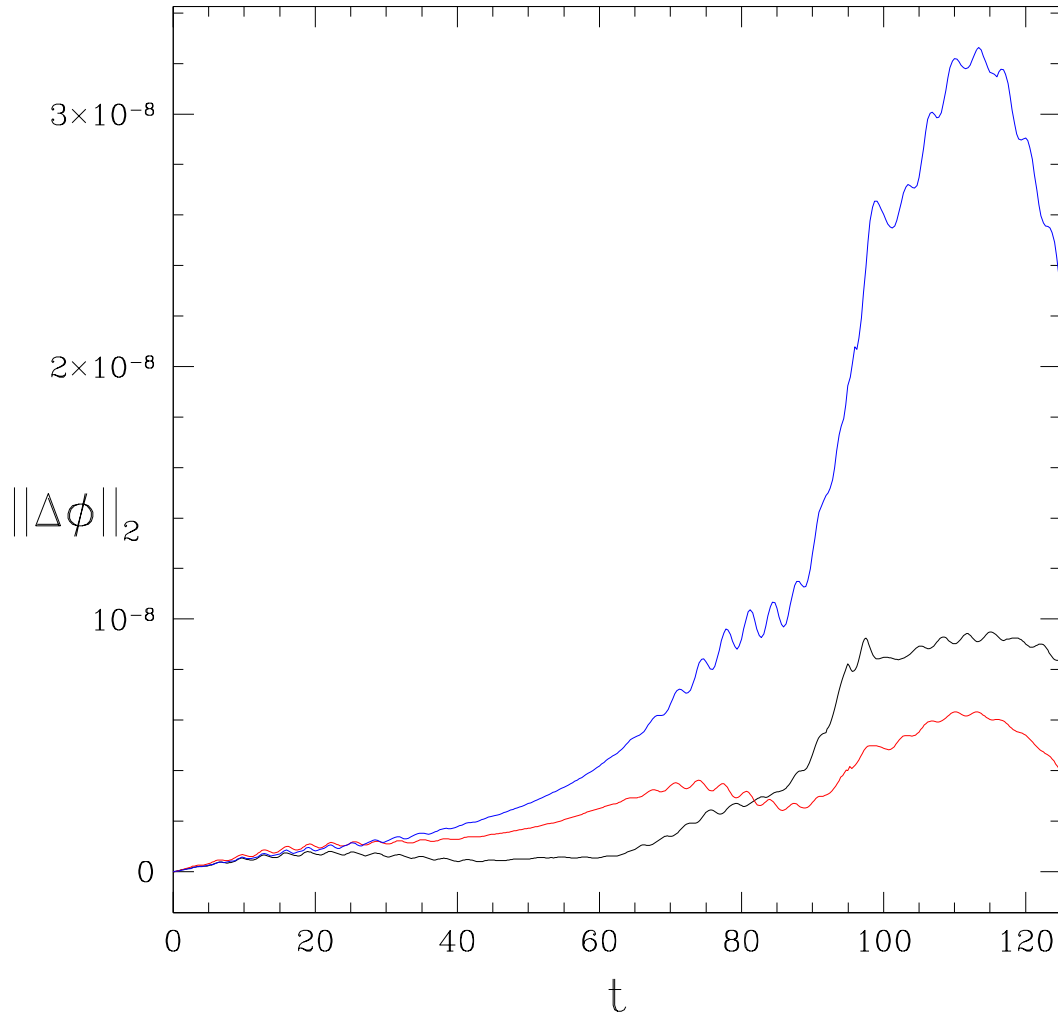


Figure 5.29: Head-on Collision of Boson Stars: $y \rightarrow -y$ Symmetry Assessment. This figure plots a measure of the extent to which our code breaks the $y \rightarrow -y$ symmetry that is expected in the continuum limit for the head-on collision of $S_{0.02}^1$ and $S_{0.03}^2$. Specifically, we define $\Delta\phi(t, x) \equiv |\phi(t, x, 5, 0)| - |\phi(t, x, -5, 0)|$ and then plot $\|\Delta\phi(t, x)\|_2$, where the norm is taken over the remaining spatial coordinate, x . Blue, red and black curves correspond to calculations on G_1 (coarsest resolution), G_2 and G_3 (finest resolution), respectively. The maximum value of $|\phi(t, x, 5, 0)|$ attained during the computation is about 10^{-2} . We thus conclude that our code preserves the $y \rightarrow -y$ symmetry to about a part in 10^6 for this particular set of calculations.

5.6 Orbital Dynamics of Two Boson Stars

In this section we summarize what we consider to be the most interesting and important calculations that we have performed with our code to date. We focus on dynamics that involves two identical boson stars— $S_{0.02}$ —but now set initial data so that, at least initially, some form of orbital motion results. The stars continue to be centred in the $z = 0$ plane, but at $t = 0$ have centres on the y axis which are equidistant from the origin. We then give the stars initial boosts in the x direction which are of equal magnitude, v_x , but of opposite sign. By adjusting the size of v_x , we can produce evolutions that lead to one of three distinct end states, as already summarized in the introductory section of this chapter. Specifically, we have run calculations for $v_x = 0, 0.03, 0.05, 0.07, 0.08, 0.09, 0.1$ and 0.11 . From this sequence of computations, and again noting that we used stars with $\phi_0 = 0.02$, we have identified approximate ranges for v_x associated with the three types of behaviour:

1. Long-lived orbital motion: $0.09 \leq v_x \leq 0.11$.
2. Merger of the stars that results in a conjectured rotating and pulsating boson star: $0.07 \leq v_x < 0.09$.
3. Merger of the stars that leads to the conjectured formation of a black hole: $0 \leq v_x < 0.07$.

Following a description of the computational setup used for our parameter survey (in v_x), we will proceed to discuss representative examples from each class in some detail.

All the calculations documented in this section were performed on a computational domain defined by $\mathbf{bbox} = [-60, 60, -60, 60, -60, 60]$, and all were made using four different grid resolutions. Denoting the grids G_1, G_2, G_3 and G_4 , where G_1 is the coarsest mesh and G_4 is the finest, the corresponding `shape` parameters are given by:

$$G_1 : \quad \mathbf{shape}_1 = [65, 65, 65], \quad (5.21)$$

$$G_2 : \quad \mathbf{shape}_2 = [81, 81, 81], \quad (5.22)$$

$$G_3 : \quad \mathbf{shape}_3 = [97, 97, 97], \quad (5.23)$$

$$G_4 : \quad \mathbf{shape}_4 = [113, 113, 113]. \quad (5.24)$$

As usual, there is a $z \rightarrow -z$ symmetry in the computations, and the plots that are displayed were generated from $z = 0$ cuts of the full 3D solutions. The two identical stars, $S_{0.02}^1$ and $S_{0.02}^2$, are initially centred at $C^1 = (0, 20, 0)$ and $C^2 = (0, -20, 0)$, and given boosts in the x direction with $v_x^1 = -v_x$ and $v_x^2 = v_x$. We emphasize that v_x is being treated as a control parameter for our experiments: in particular, the *only* difference in parameter settings for the three cases described below is the value of v_x that is used.

5.6.1 Case 1: $v_x = 0.09$ —Long Lived Orbital Motion

Results from this calculation are shown in Figs. 5.30–5.35. Fig. 5.30 displays a series of snapshots of $|\phi(t, x, y, 0)|$ for $0 \leq t \leq 4211$. During this period of coordinate time the stars orbit one another about $2\frac{1}{4}$ times. The orbit is slightly eccentric, as can be seen from Fig. 5.31 which shows the orbital trajectory of the centre of each star (defined to be the position of the local maximum of $|\phi(t, x, y, 0)|$). We note, however, that we have not yet made any attempt to estimate the eccentricity. Detailed examination of the trajectories suggests that the orbit is also precessing. Since orbital precession is a purely general relativistic effect (assuming that the stars remain spherical in their respective rest-frames) this qualitative result provides additional encouraging evidence that the CFA is capturing some of the key physical effects that one would see in a solution of the full Einstein equations. It would therefore be very interesting to attempt to quantify the observed rate of precession and compare it to the value obtained (or estimated) from a general relativistic calculation. This, however, is something that remains to be done.

Plots of the l_2 norms of two of the independent residuals— $I_{\phi_1}(t)$ and $I_\alpha(t)$ —from the computations performed on the four different grids are shown in Figs. 5.32–5.35. Here, and for all of the remaining figures in this chapter, the colours green, blue, red and black are used for quantities computed on grids G_1 , G_2 , G_3 and G_4 respectively. Additionally, and in contrast to the corresponding plots in Secs. 5.2–5.4 we have *not* scaled the independent residuals in this section or the next, as the resulting graphs are difficult to interpret (exceptions are Fig. 5.33 and Fig. 5.35). Nonetheless, a detailed quantitative examination of the residuals, such as that provided by Fig. 5.33 and Fig. 5.35, reveals that the independent residuals are converging at the rates expected.

5.6.2 Case 2: $v_x = 0.07$ —Formation of Pulsating & Rotating Boson Star

Results from this calculation are summarized in Figs. 5.36–5.38. Fig. 5.36 shows a series of surface plots of $|\phi(t, x, y, 0)|$ for $0 \leq t \leq 2089$, while Fig. 5.37 shows trajectories of the stars prior to their merger. In this case orbital motion persists for a little less than one half of a full rotation before the stars graze one another. This is followed by a rapid “plunge” phase which leads to the formation of what we conjecture to be essentially a single spinning and pulsing boson star, with a central field modulus $\phi_0 \simeq 0.05$.

As usual, we monitor independent residuals to ensure that our calculations are reliable, and Fig. 5.38 shows the l_2 norm of $I_\alpha(t)$. We first note that the plot shows that I_α is tending to 0 as the mesh spacing decreases. However, it is also clear from the figure that the convergence deteriorates at later times, especially once the stars have merged. Indeed, the apparent divergence of $I_\alpha(t)$ on G_1 (green) and G_2 (blue) for $t \gtrsim 500$ is a clear indication that the numerical solutions computed on those grids are not trustworthy once the collision has occurred. In addition, the merger results in the ejection (radiation) of a significant amount of scalar matter. Some of this ejecta hits the outer computational boundary and, due to our use of Dirichlet boundary conditions, is largely reflected back into the solution domain. These spurious reflections eventually contaminate the entire calculation, so that even though the solution may still exhibit convergence as measured via independent residuals, the physical interpretation at late times is dubious at best.

5.6.3 Case 3: $v_x = 0.05$ —Formation of a Black Hole

Results from our last numerical experiment are shown in Figs. 5.39–5.41: here the initial boost parameter is $v_x = 0.05$. As usual, we start with surface plots of $|\phi(t, x, y, 0)|$, this time on the interval $0 \leq t \leq 411$. Following a very brief period of orbital motion, the two stars quickly plunge towards one another and merge, as can clearly be seen in the trajectory plots shown in Fig. 5.40. In this case we believe that the evolution results in the formation of a single black hole. This hypothesis could best be tested by looking for marginally trapped surfaces [120] in our data—that is, surfaces with S^2 topology for which the divergence of outgoing null geodesics emanating from the surface vanishes. If such a surface was located, then, assuming cosmic censorship [120] holds, we could infer that the data described is a black hole. However, at the current time we have not implemented this approach. Instead, our conjecture is based on the behaviour of the metric functions in the central interaction region. For example, the lapse function attains extremely small

values in that part of the solution domain, effectively freezing the evolution there. This “collapse of the lapse” is a very well known feature of maximal slicing and is usually correlated with the formation of a black hole.

However, we must emphasize that various indicators—such as the plot of the independent residuals for the lapse function shown in Fig. 5.41—strongly suggest that the results for this specific choice of v_x are not trustworthy for times $t \gtrsim 380$, even on the finest grid used. At later times the matter is highly centrally condensed, and, assuming that collapse to a black hole *is* occurring, would become increasingly so as the evolution proceeded. Thus, only a substantial increase in resolution in the interaction region would allow us to provide a definitive answer concerning the end state of members of this class of initial data.

Finally, since code “crashes”—due to floating point overflows, for example—are rather frequent occurrences in numerical relativity calculations, we should point out that our code did *not* “crash” at late times for this configuration (or for any other simulation reported in this thesis, for that matter). Rather, we simply stopped the evolution at a coordinate time when, even on the finest grid, the solution was obviously very poorly resolved.

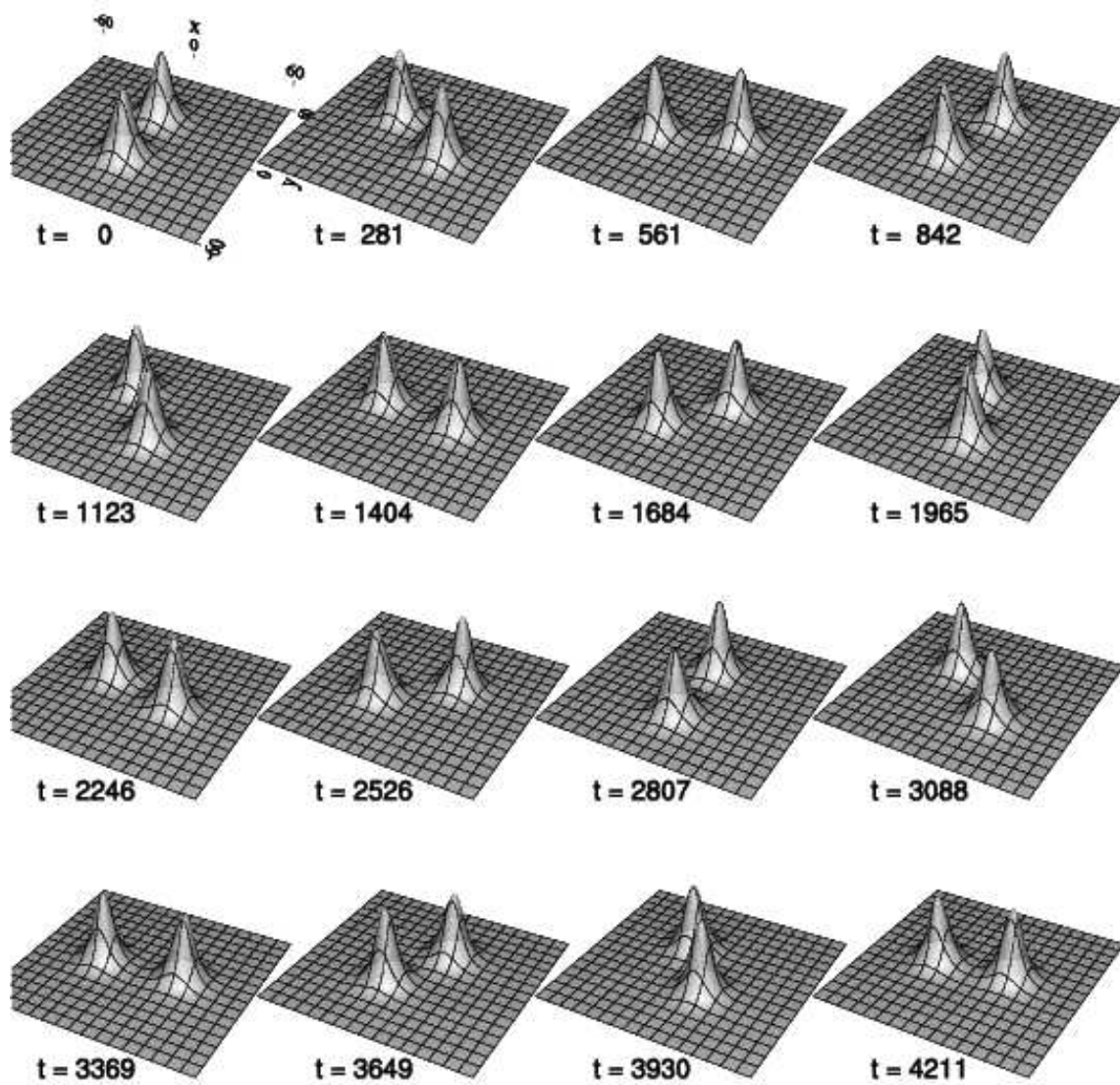


Figure 5.30: This figure shows a series of snapshots of $|\phi(t, x, y, 0)|$ for $0 \leq t \leq 4211$ from a calculation describing long lived orbital motion of two identical boson stars. The two stars, $S_{0.02}^1$ and $S_{0.02}^2$, are initially centred at $C^1 = (0, 20, 0)$ and $C^2 = (0, -20, 0)$ and boosted with parameters $v_x = -v_x^{(1)} = v_x^{(2)} = 0.09$. During this period of time that is displayed, the stars execute about $2\frac{1}{4}$ orbits, and the maximum value of $|\phi(t, x, y, 0)|$ remains approximately constant during the evolution.

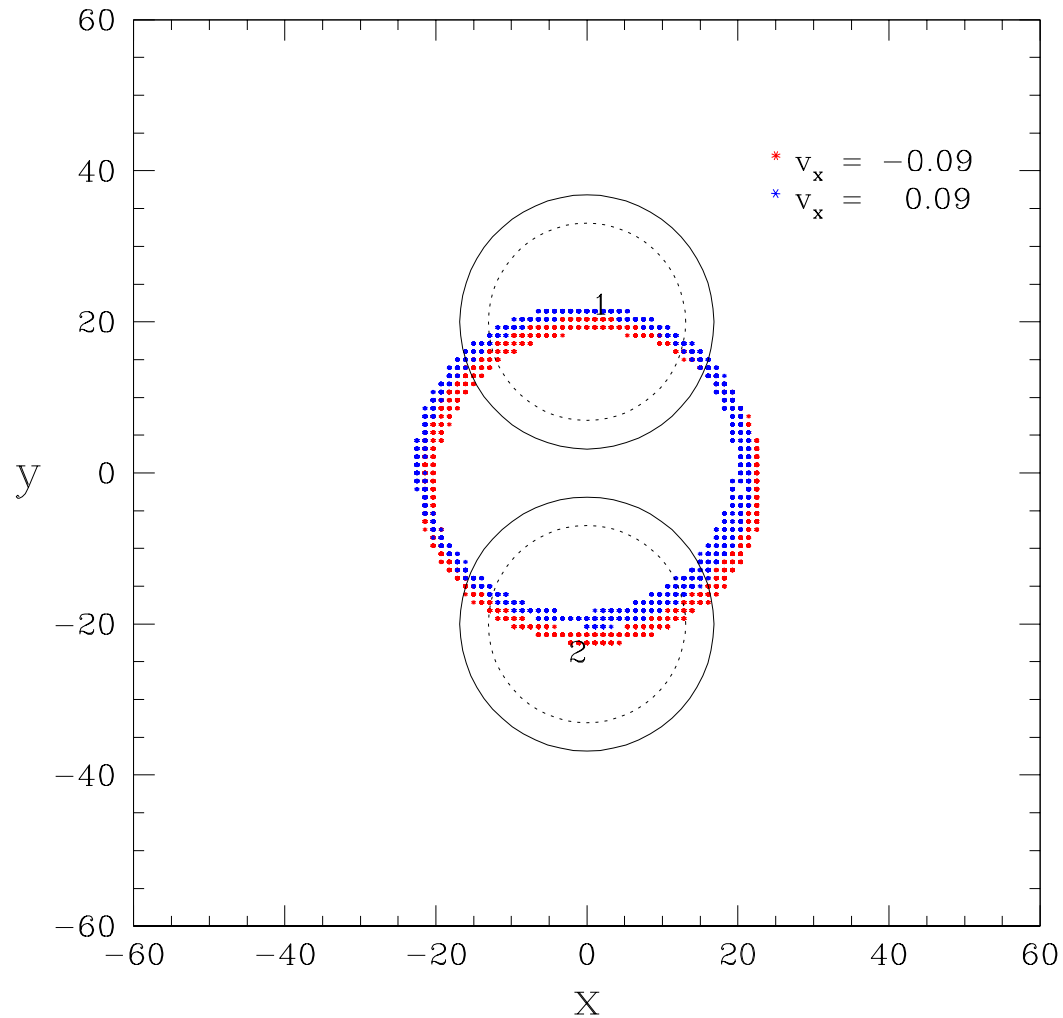


Figure 5.31: Long lived orbital motion: stellar trajectories. This figure shows trajectories of the two stars, $S_{0.02}^1$ and $S_{0.02}^2$, as they orbit one another during the coordinate time interval $0 \leq t \leq 4500$. Each red or blue dot represents the local maximum of $|\phi(t, x, y, 0)|$ at a particular instant of time. The orbit is slightly eccentric and is apparently precessing with time. In this and other figures of this type, in order to provide some sense of the size of the stars we have also plotted circles that approximately delimit the stellar surfaces. Specifically, the solid and dashed lines have radii R_{99} (containing 99% of M_{ADM}) and R_{95} (containing 95% of M_{ADM}), respectively.

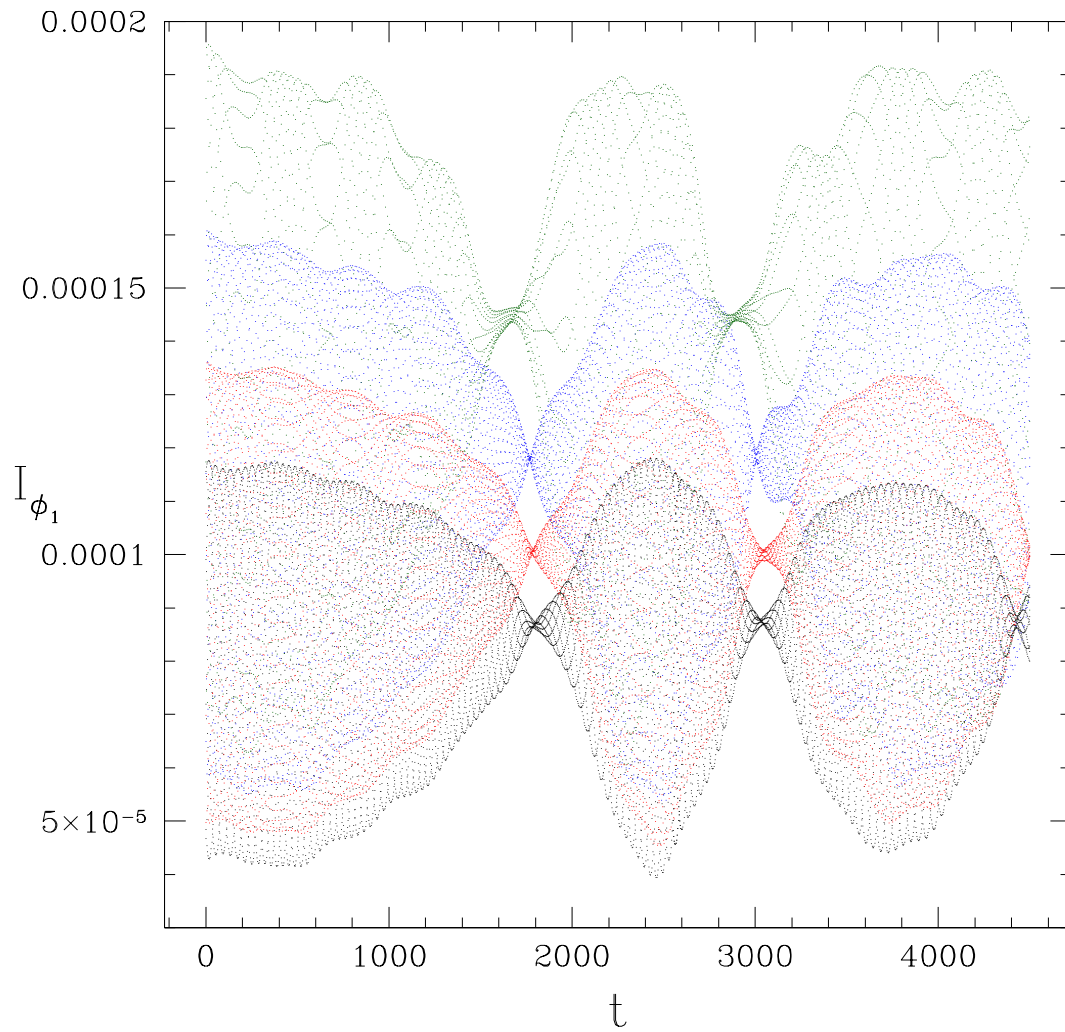


Figure 5.32: Long Lived Orbital Motion: $\|I_{\phi_1}(t)\|_2$. This figure shows the l_2 -norm of the independent residuals $I_{\phi_1}(t)$ for $0 \leq t \leq 4500$. During this interval of coordinate time the stars perform about $2\frac{1}{4}$ orbits. Residuals plotted with green, blue, red and black dots were computed using grids G_1 (coarsest resolution), G_2 , G_3 and G_4 (finest resolution), respectively, with specific grid parameters defined in the text. Note that these residuals have *not* been rescaled. The plots provide strong evidence of convergence of the independent residuals as $h \rightarrow 0$.

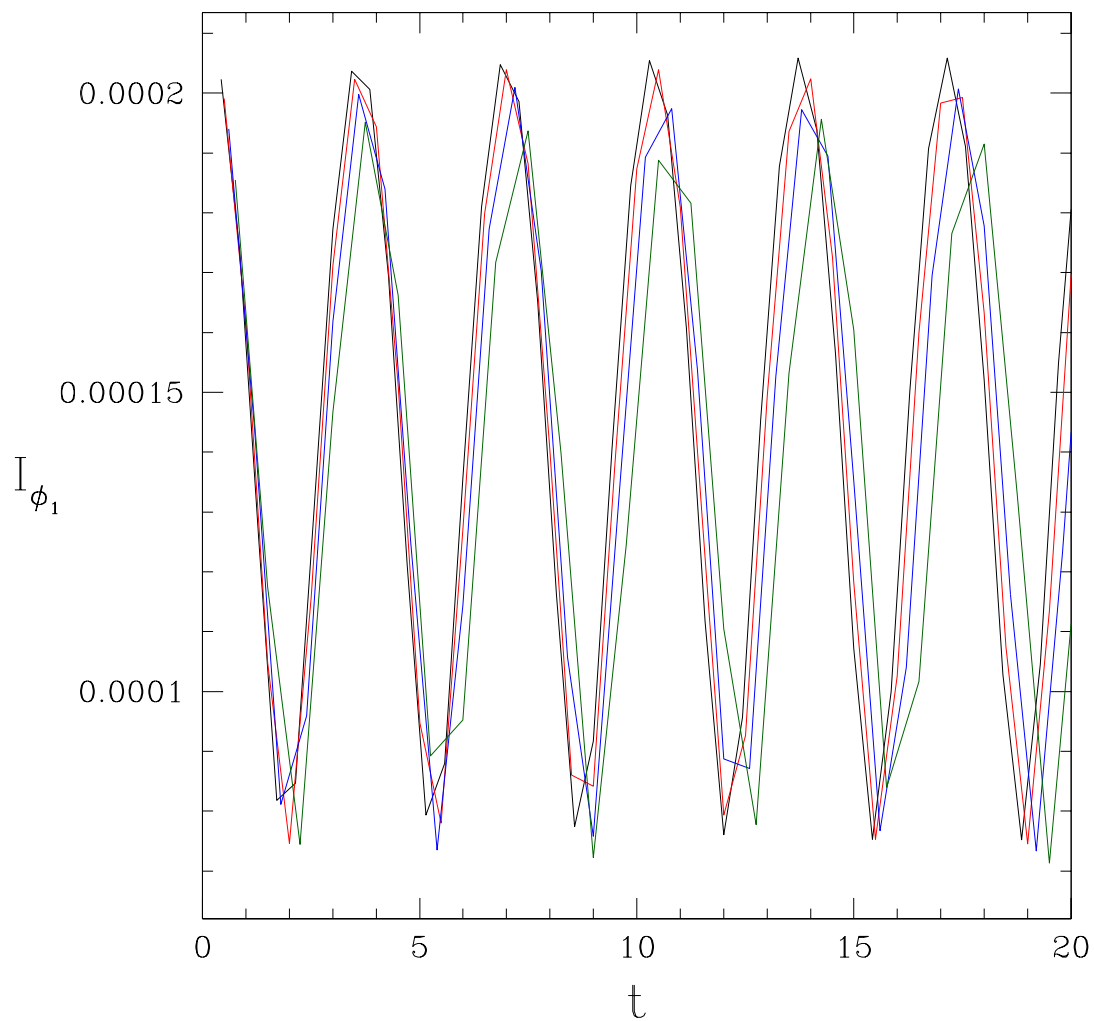


Figure 5.33: Long Lived Orbital Motion: Rescaled $\|I_{\phi_1}(t)\|_2$. This figure shows the same independent residuals displayed in Fig. 5.32, but on the shorter time interval $0 \leq t \leq 20$, and with scaling factors h_1/h_2 , h_1/h_3 , and h_1/h_4 multiplying the values computed on G_2 (blue), G_3 (red) and G_4 (black), respectively. The expected $O(h)$ convergence of the residuals is apparent.

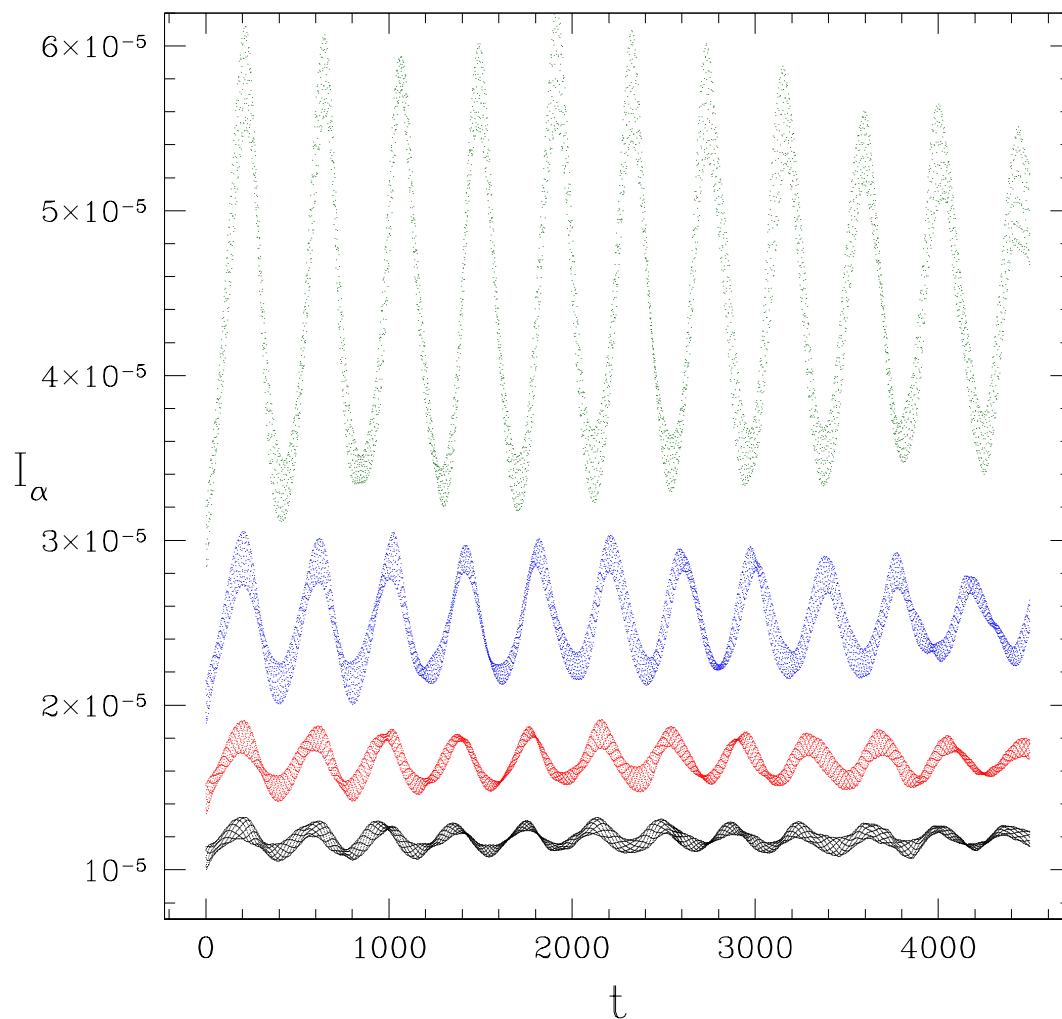


Figure 5.34: Long Lived Orbital Motion: $\|I_\alpha(t)\|_2$. This figure shows the l_2 -norm of the independent residuals $I_\alpha(t)$ for $0 \leq t \leq 4500$, and as computed on grids G_1 (green, coarsest resolution), G_2 (blue), G_3 (red), and G_4 (black, finest resolution). As with Fig. 5.32, these residuals have *not* been rescaled. Once more, the plots strongly suggest convergence of the independent residuals as $h \rightarrow 0$.

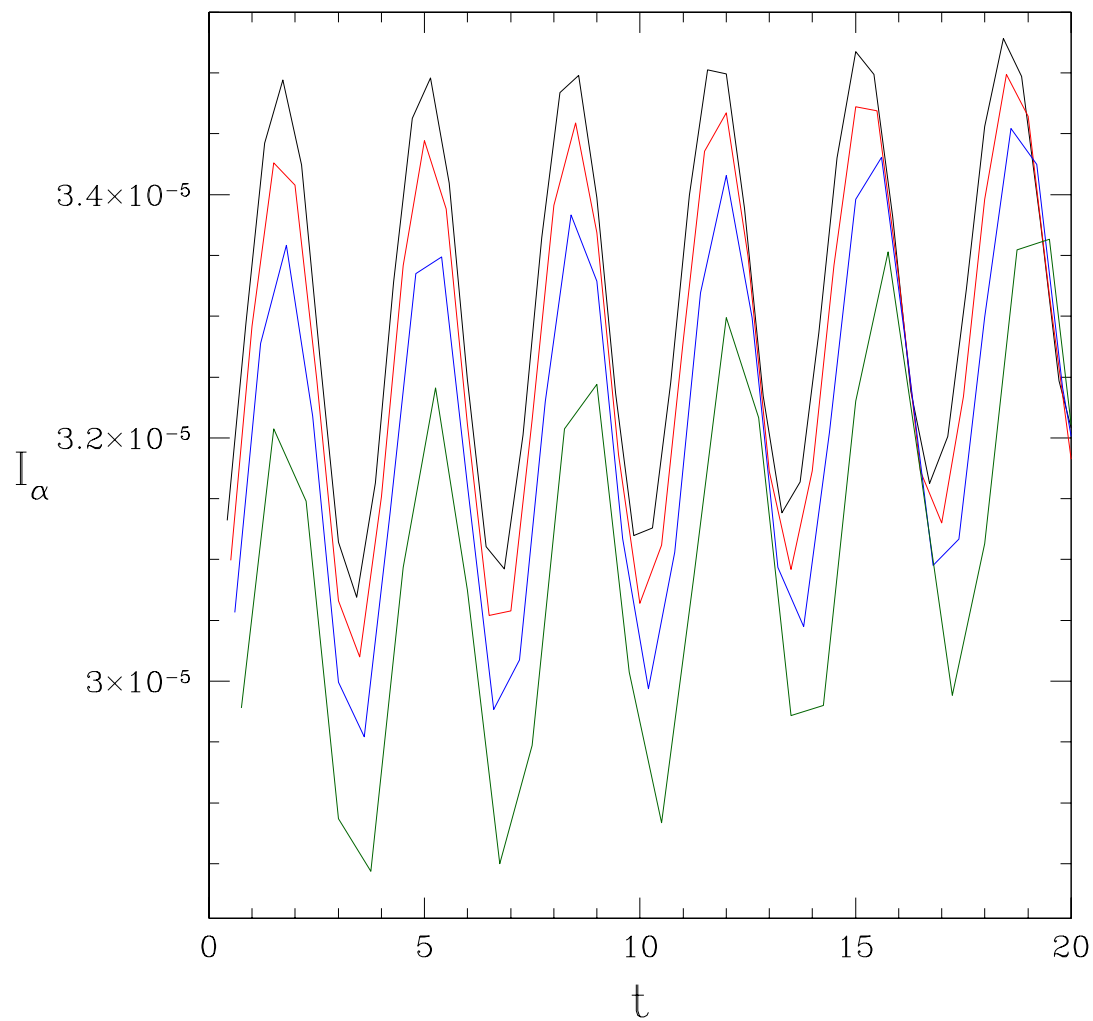


Figure 5.35: Long lived Orbital Motion: Rescaled $\|I_\alpha(t)\|_2$. This figure shows the same independent residuals displayed in Fig. 5.34, but on the shorter time interval $0 \leq t \leq 20$, and with scaling factors $(h_1/h_2)^2$, $(h_1/h_3)^2$, and $(h_1/h_4)^2$ multiplying the values computed on G_2 (blue), G_3 (red) and G_4 (black), respectively. The $O(h^2)$ convergence rate expected for all the residuals associated with the metric variables is apparent.

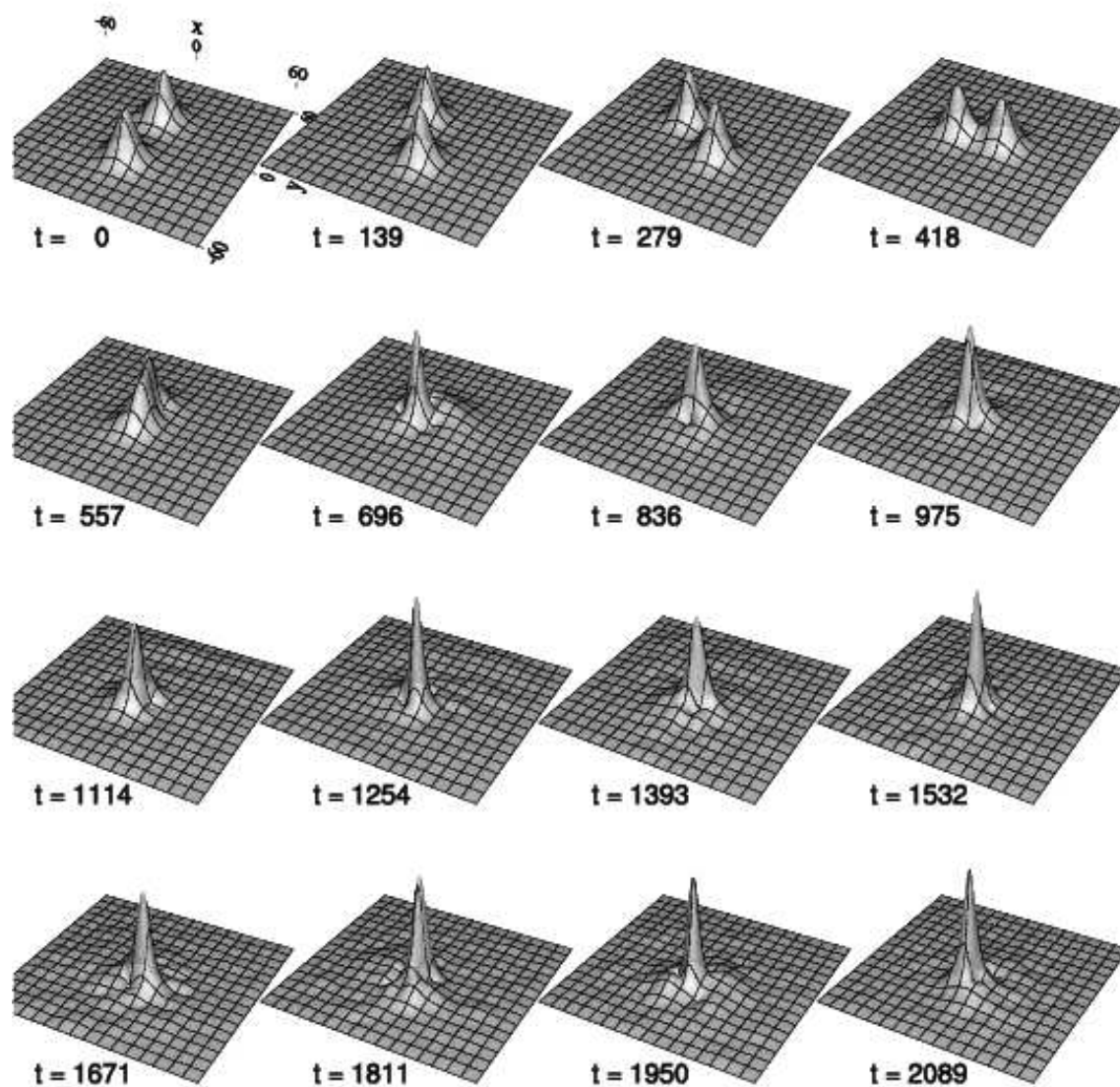


Figure 5.36: This figure displays a series of surface plots of $|\phi(t, x, y, 0)|$ for $0 \leq t \leq 2089$, from a calculation in which two identical stars, $S_{0.02}^1$ and $S_{0.02}^2$, are initially centred at $C^1 = (0, 20, 0)$ and $C^2 = (0, -20, 0)$, and boosted with parameters $v_x = -v_x^{(1)} = v_x^{(2)} = 0.07$. The stars subsequently execute somewhat less than half of an orbit before they graze one another. A quick “plunge” phase follows, leading to the formation of a (conjectured) pulsating and rotating boson star characterized by a central field modulus $\phi_0 \simeq 0.05$.

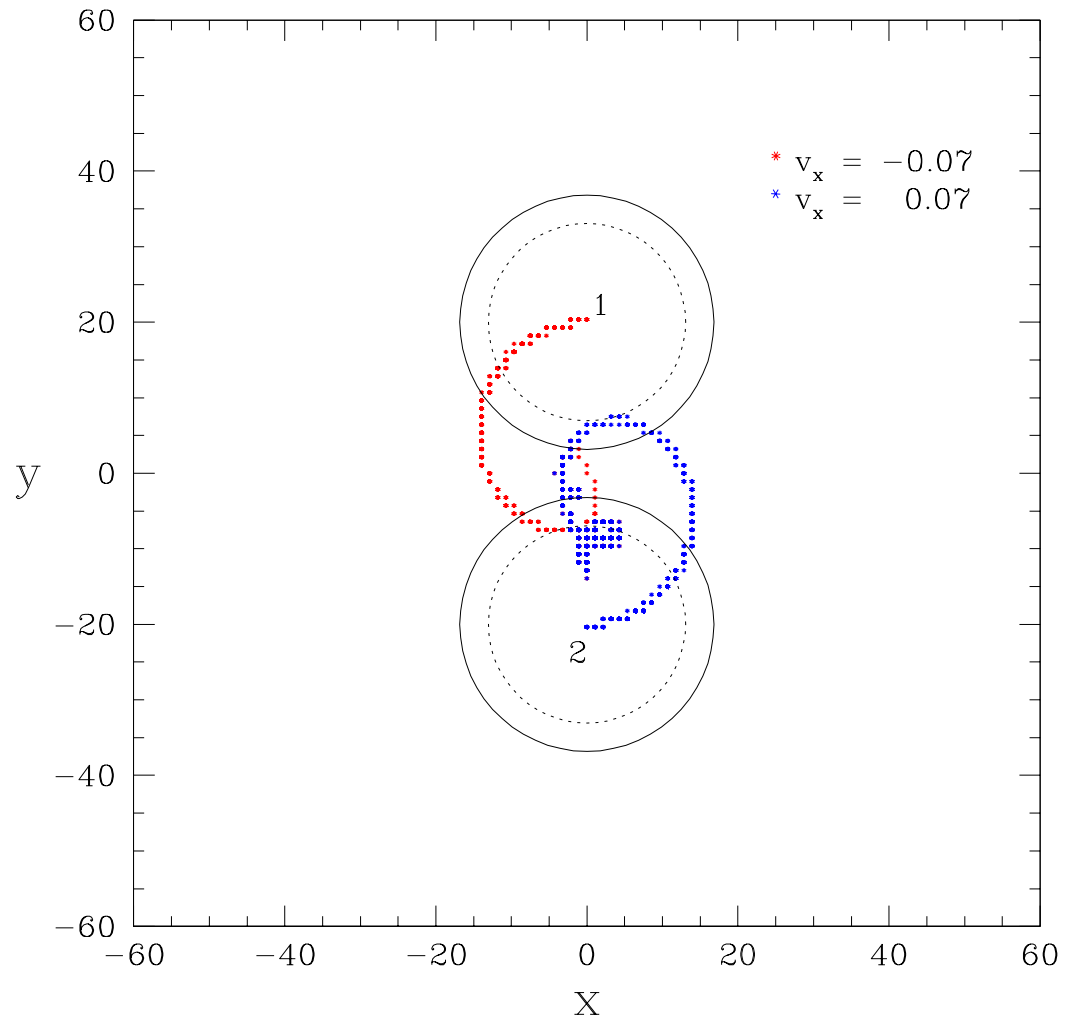


Figure 5.37: Formation of Pulsating and Rotating Boson Star: Trajectories. This figure shows the trajectories of the stars $S_{0.02}^1$ and $S_{0.02}^2$ for the computation described in the caption of Fig. 5.36. Each red or blue dot plots the position a local maximum of $|\phi(t, x, y, 0)|$ at a particular instant of time. Here the tracks clearly reveal a short period of orbital motion which is followed by a rapid “plunge” and merger of the stars.

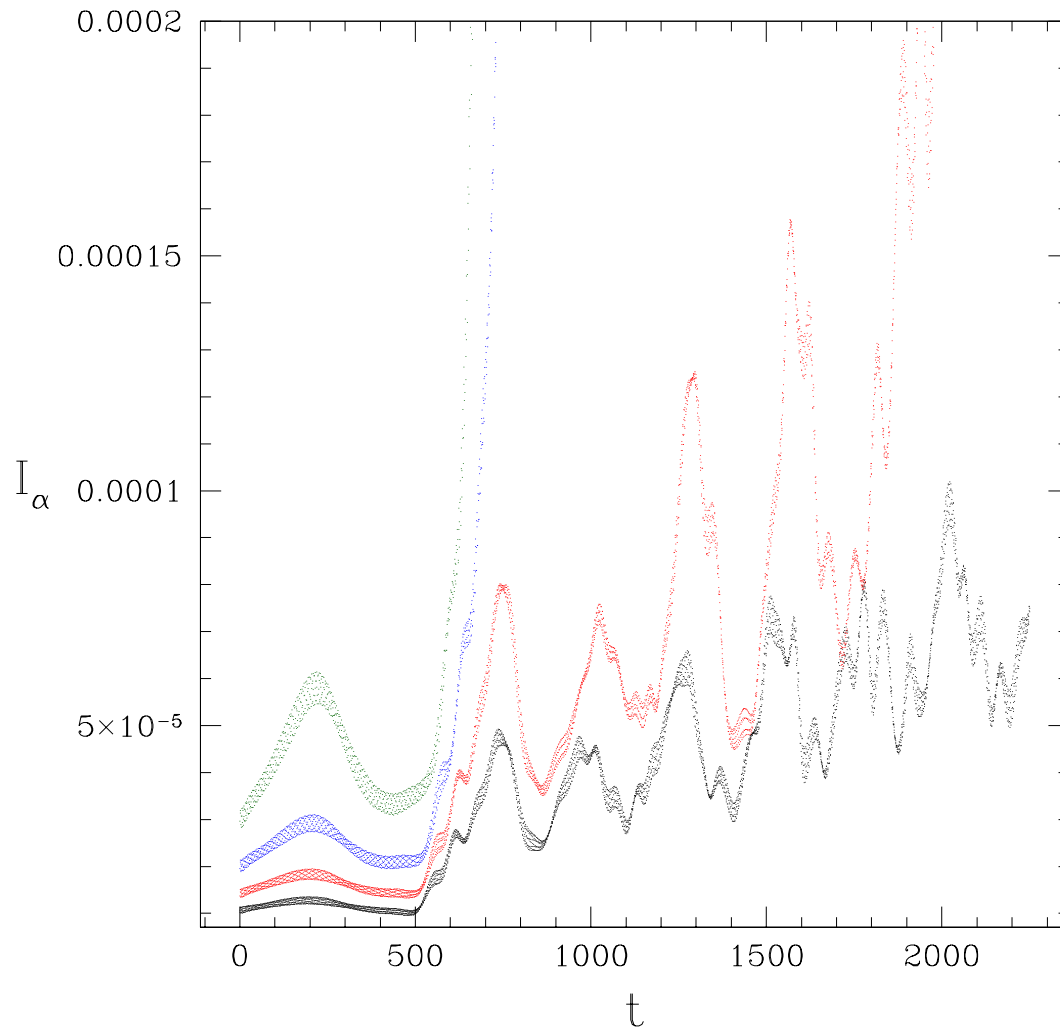


Figure 5.38: Formation of Pulsating and Rotating Boson Star: $\|I_\alpha(t)\|_2$. This figure shows the l_2 -norm of the independent residual $I_\alpha(t)$ for the calculation described in Fig. 5.36. We note that the residuals have *not* been rescaled in this case, but that the values computed on grids G_1 (green, coarsest resolution), G_2 (blue), G_3 (red), and G_4 (black, finest resolution) *do* seem to be converging to 0 as $h \rightarrow 0$. However, it also clear that there is a qualitative change in the nature of the independent residuals computed on G_1 and G_2 once the collision has occurred at $t \sim 500$. This is a good indication that the calculations on those relatively coarse meshes are not reliable for $t \gtrsim 500$.

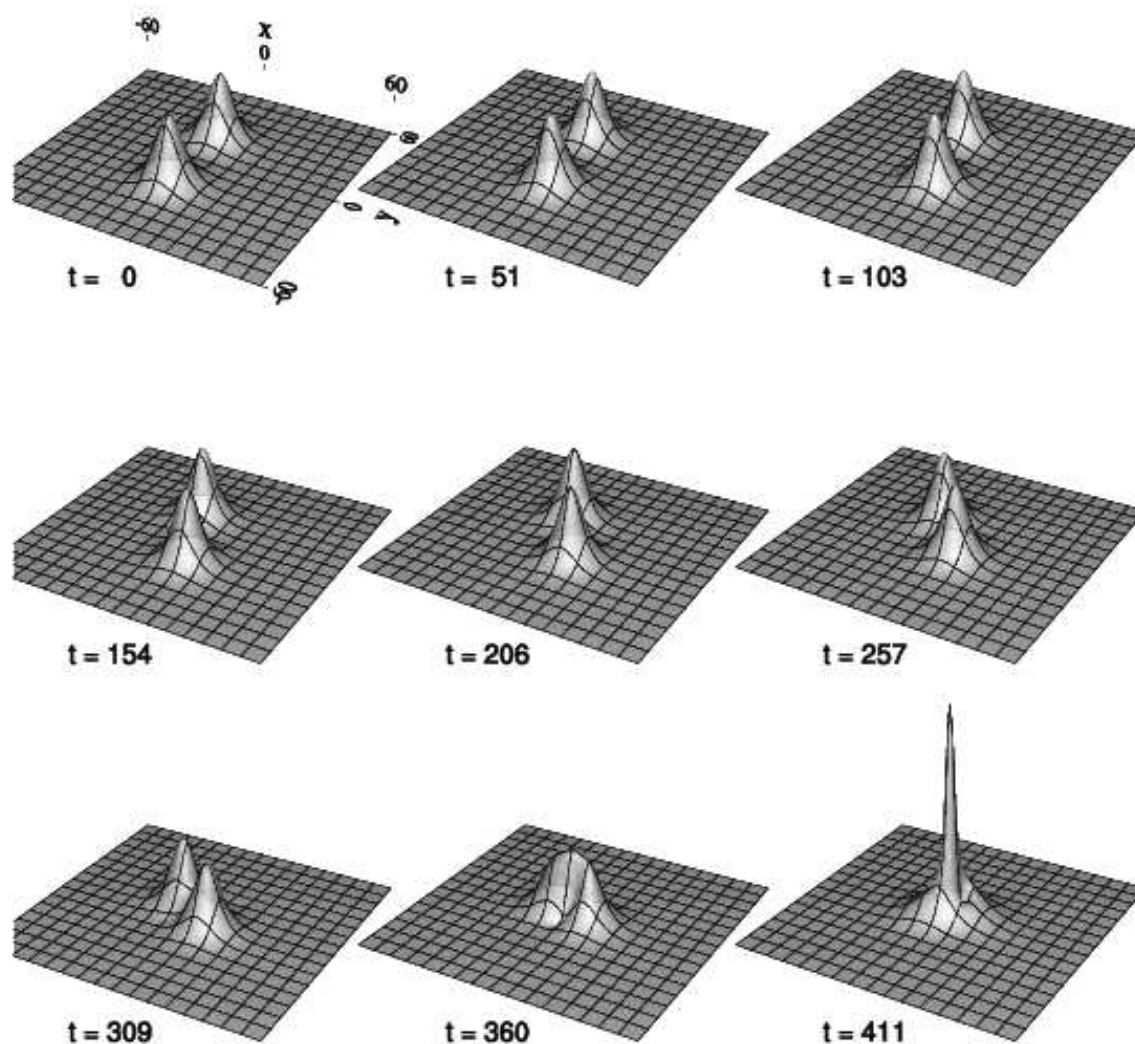


Figure 5.39: This figure displays a series of surface plots of $|\phi(t, x, y, 0)|$ for $0 \leq t \leq 411$, from a calculation in which two identical stars, $S_{0.02}^1$ and $S_{0.02}^2$, are initially centred at $C^1 = (0, 20, 0)$ and $C^2 = (0, -20, 0)$, and boosted with parameters $v_x = -v_x^{(1)} = v_x^{(2)} = 0.05$. As with all previous plots of this type, the specific results visualized were computed on the finest grid, G^4 , which has $\text{shape} = [113, 113, 113]$. In this case, the stars rapidly approach each other, and, after a plunge and merger phase, apparently undergo gravitational collapse to form a single black hole. At $t = 411$ the maximum value of $|\phi(t, x, y, z)|$ (the height of the “spike” in the last frame) is $\simeq 0.06$, and even larger values— $|\phi(t, x, y, z)| \gtrsim 0.14$ —are seen at later times. We note that as can be verified from Fig. 3.2, such large values are associated with boson stars on the *unstable* branch. It thus seems plausible that the evolution will lead to black hole formation, but computations with considerably more resolution in the central, strongly interacting region would be required to test this hypothesis.

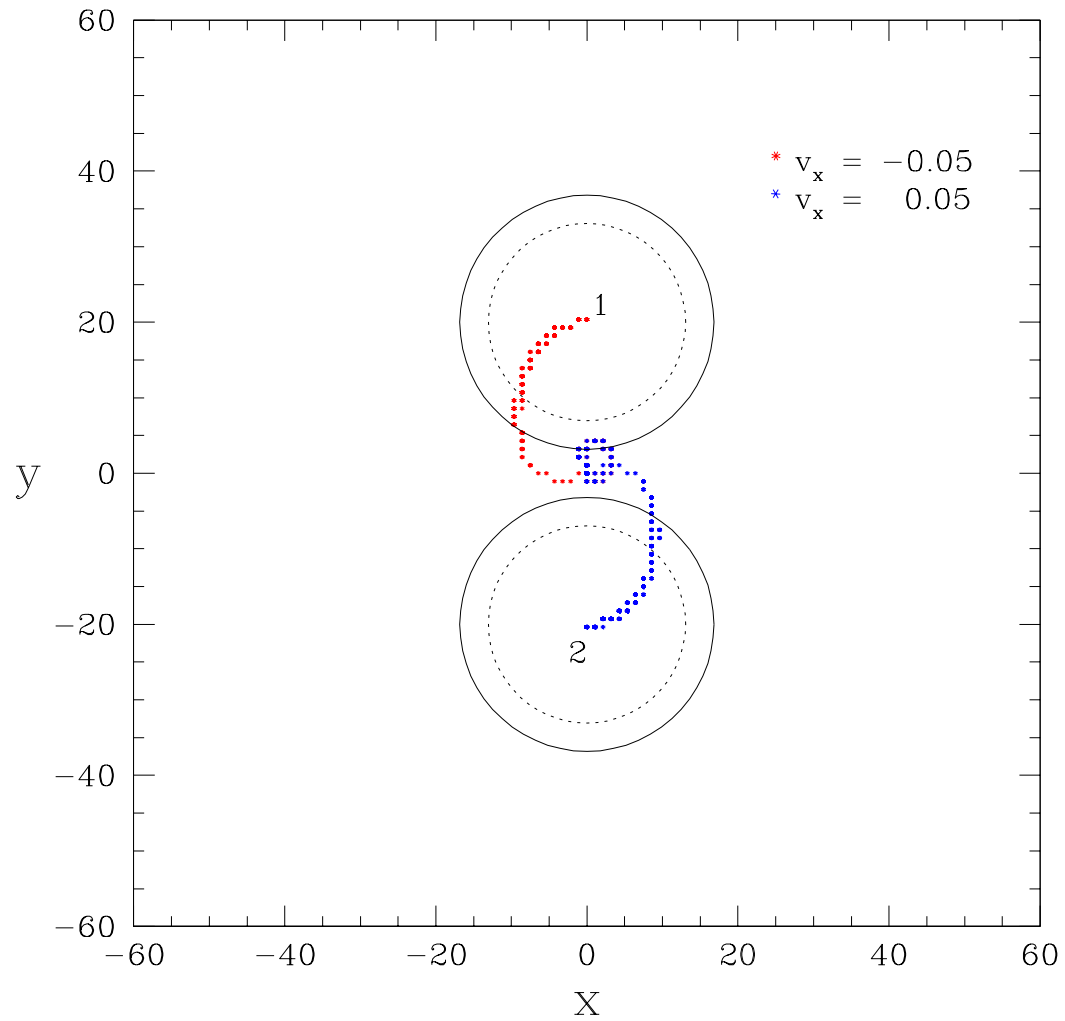


Figure 5.40: Formation of a Black Hole: Stellar Trajectories. This figure shows the trajectories of the stars $S_{0.02}^1$ and $S_{0.02}^2$ from the calculation described in the caption of Fig. 5.39, and during the coordinate time interval $0 \leq t \leq 1500$. Once more, each red and blue dot represents a local maximum of $|\phi(t, x, y, 0)|$ at a particular instant of time. The resulting tracks show a rapid plunge of the stars towards one another, resulting in a merger that we believe ends in black hole formation.

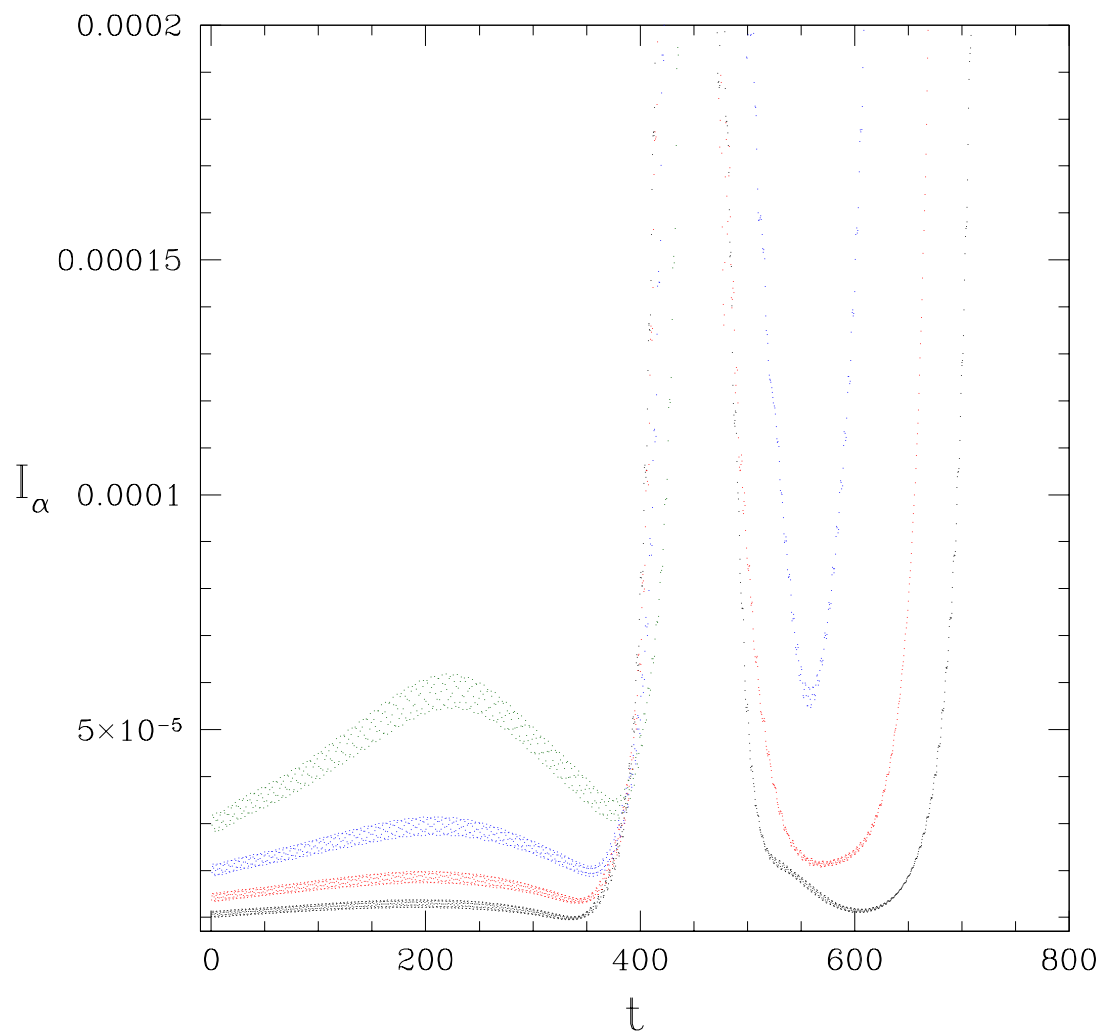


Figure 5.41: Formation of a Black Hole: $\|I_\alpha(t)\|_2$. This figure shows the l_2 -norm of the independent residuals $I_\alpha(t)$ for the calculation described in the caption of Fig. 5.39. The residuals have not been rescaled. At early times, the residual values computed on grids G_1 (green, coarsest resolution), G_2 (blue), G_3 (red), and G_4 (black, finest resolution) once more appear to be converging to 0 as $h \rightarrow 0$. However, for $t \gtrsim 400$, when we suspect that gravitational collapse to a black hole would result if the finite difference resolution was high enough, we see an obvious deterioration in the convergence of the residuals computed on *all* grids. We interpret this as a clear indication that none of the computations are reliable for $t \gtrsim 400$.

5.7 Discussions and Further Developments

It is worth noting that the second and third experiments discussed in the previous section are qualitatively consistent with some of the results reported by Palenzuela *et. al.* in [10]. Those authors have implemented a fully general relativistic 3D code for the Einstein-Klein-Gordon system, and have also performed calculations in which the initial data contains two identical spherically symmetric boson stars that have been given equal but opposite boosts. One of the calculations discussed in [10] is quite similar in setup to our $v_x = 0.07$ calculation described in Sec. 5.6.2, in which the final state appears to be a pulsating and rotating boson star. The specific stars Palenzuela and collaborators used in that case each had an ADM mass, $M_{\text{ADM}} = 0.5$, while the ones used here ($S_{0.02}$) had $M_{\text{ADM}} \simeq 0.475$. They also positioned their stars so that, at least in coordinate space, they were closer to one another at the initial time than ours were. Defining d to be the coordinate separation of the centres of the stars, they took $d = 32$. On the other hand, we chose $d = 40$ so that our stars—which have $R_{99} \sim 17.25$ —were essentially completely separated at $t = 0$. These differences in the initial configurations are relatively minor, so we feel that a general comparison of our results with theirs is not unreasonable. Specifically, they identified the end state of their evolution as a spinning “bar” configuration, accompanied by dispersal of scalar matter, and this at least roughly agrees with the interpretation of our end state as a spinning and pulsing star. In addition, the range of values of the boost parameter, v_x , that led to black hole formation in their calculations, namely $0 \leq v_x < 0.04$, is in rough agreement with our findings, where we conjecture that black hole formation occurs for $0 \leq v_x < 0.07$.

Palenzuela *et. al.* also report a very interesting case in which the stars merge into a transient single lump of scalar field from which two configurations of approximately the same form as the initial stars eventually emerge, and propagate away from one another. We have *not* observed any behaviour of this type in the calculations we have done with our code thus far.

Arguably, the most novel of the results presented here are those that describe long term orbital motion. To our knowledge, the calculations reported in Sec. 5.6 are the first in which more than two orbits of a binary boson star have been simulated. Furthermore, we have noted that there is strong evidence of orbital precession, which indicates that our model has captured some of the key physical effects predicted by general relativity.

We note that one of the original motivations for this work was to see whether the compression effect reported by Wilson and Mathews [16, 17, 18, 19, 20, 21, 23, 25, 26] for the case of binary

neutron star inspiral (and using the conformally flat approximation with maximal slicing) was present in our boson star analog of that setup. In this regard our current results are inconclusive. In one calculation, where the boost parameter was $v_x = 0.08$, there *did* appear to be compression of the stars as they orbited. However, this is another case where significantly higher resolution is needed to provide a clear answer. Moreover, should something like the Wilson-Mathews effect be convincingly seen in our calculations, we would want to vary the scalar field self-interaction potential, $U(|\phi_0|^2)$ to see what impact that had on the results. Here the key point is that the reported compression effect for the case of fluid stars depends on the equation of state that is adopted.

We conclude this chapter with a few comments concerning future plans for improvements and extensions of our code. The most pressing issue is clearly that of getting adequate finite-difference resolution for those instances where the boson stars interact strongly, and especially when black holes appear to be forming. This is simply not possible given that our current code uses a single uniform mesh, and only runs on one processor. We therefore plan to parallelize the code but, perhaps more importantly, to incorporate adaptive mesh refinement (see, for example, the classic paper by Berger and Oliger [152]), so that the increased resolution can be concentrated where it is needed. In fact, using the AMRD/PAMR software infrastructure developed by Pretorius [177, 178], we have already developed a version of the code that should both run in parallel (with good scaling on up to 100's of processors) and provide AMR. However, this implementation is still being debugged and tested.

A second component of our algorithm that obviously needs improvement is the treatment of the boundary conditions. As discussed in Sec. 2.4.5 a (formally) exact implementation of the physical boundary conditions could be achieved through compactification of the spatial domain. Although it is promising, this approach would require the development of a more sophisticated multigrid solver, capable of efficiently solving highly anisotropic equations, and doing so in parallel. This will not be an easy task. Given that we are incorporating AMR into our code—which should allow us to substantially extend the boundaries of the computational domain—we feel that it would be most fruitful at this time to implement the strategy described in Sec. 2.4.5, which uses mixed (Robin) conditions based on known falloff behaviour for the metric variables, and Sommerfeld conditions for the scalar field unknowns.

We end by emphasizing that the experiments described above represent only a start of a thor-

ough investigation of the multi-dimensional parameter space of boson star orbital dynamics. We hope that once we have implemented the improvements just discussed, we will be able to use our code to survey this space at significantly lower computational cost relative to fully general relativistic calculations. Ideally, this survey would identify regions of parameter space that warrant more detailed investigation using codes that solve the full Einstein equations.

Chapter 6

Conclusion

The main new work presented in this thesis involved the implementation of a finite-difference code to solve a nonlinear system of elliptic-hyperbolic partial differential equations in 3 space dimensions plus time. The PDEs describe the evolution of a self-gravitating massive complex scalar field, where the gravitational interaction is given by the conformally flat approximation of the Einstein equations. This code was thoroughly tested to establish overall correctness of the implementation (in particular, consistency with the continuum PDEs), and was shown to generate results that converge to the continuum limit at the expected rate in the mesh spacing. The implementation was then used to study interactions between two boson stars.

In simulations of head-on collisions, we observed the same type of solitonic behaviour that has been seen in previous computations using this type of matter [173, 174, 128, 130]. We also performed a preliminary survey of orbital dynamics of boson star binaries, finding evidence for three distinct end states: a long-lived, precessing elliptic orbit, merger of the two stars into a final, more compact boson star, and merger that we conjecture produces a black hole. These results are consistent with the findings of Palenzuela and collaborators [10], who solved the full Einstein-scalar system. This concordance suggests that the CFA may be sufficiently accurate to capture at least some of the key features of general relativistic dynamics, even in fully 3D scenarios.

The major weakness of our current code is its limited resolution. For many of the runs we have described, the resolution was sufficient for us to be confident that our modeling was capturing the key physics exhibited in the evolutions. However, in other instances, and especially in those cases where we suspect that black holes are forming, the calculations are *not* reliable at late integration times, and only a substantial increase in resolution will fix this flaw. As we have already discussed, we are currently working on a parallel, adaptive version of the code that should provide this fix.

Determining conditions under which the CFA does or does not provide a good approximation of general relativity is another research priority. Ultimately, this pursuit may even be able to shed some light on the elusive issue of identifying the true dynamical (radiative) degrees of freedom of

Einstein's theory.

Bibliography

- [1] Keita Kawabe. Status of LIGO. *J. Phys. Conf. Ser.*, 120:032003, 2008.
- [2] F. Acernese et al. The status of Virgo. *J. Phys. Conf. Ser.*, 110:062025, 2008.
- [3] R. Takahashi et al. Operational status of TAMA300 with the Seismic Attenuation System (SAS). *Class. Quant. Grav.*, 25:114036, 2008.
- [4] Benno Willke. GEO600: Status and plans. *Class. Quant. Grav.*, 24:S389–S397, 2007.
- [5] Curt Cutler and Eanna E. Flanagan. Gravitational waves from merging compact binaries: How accurately can one extract the binary’s parameters from the inspiral wave form? *Phys. Rev.*, D49:2658–2697, 1994, gr-qc/9402014.
- [6] J. Balakrishna. *A Numerical Study of Boson Stars: Einstein Equations with a Matter Source*. PhD thesis, Washington University, (1999), gr-qc/9906110.
- [7] F. Siddhartha Guzman. Evolving spherical boson stars on a 3d cartesian grid. *Phys. Rev.*, D70:044033, 2004, gr-qc/0407054.
- [8] Jayashree Balakrishna, Ruxandra Bondarescu, Gregory Daues, F. Siddhartha Guzman, and Edward Seidel. Evolution of 3d boson stars with waveform extraction. *Class. Quant. Grav.*, 23:2631–2652, 2006, gr-qc/0602078.
- [9] C. Palenzuela, I. Olabarrieta, L. Lehner, and S. Liebling. Head-on collisions of boson stars. *Phys. Rev.*, D75:064005, 2007, gr-qc/0612067.
- [10] C. Palenzuela, L. Lehner, and S. L. Liebling. Orbital Dynamics of Binary Boson Star Systems. *Phys. Rev.*, D77:044036, 2008, 0706.2435.
- [11] Gregory B. Cook, Stuart L. Shapiro, and Saul A. Teukolsky. Testing a simplified version of Einstein’s equations for numerical relativity. *Phys. Rev. D*, 53(10):5533–5540, May 1996.

- [12] Willy Kley and Gerhard Schaefer. Relativistic dust disks and the Wilson-Mathews approach. *Phys. Rev.*, D60:027501, 1999, gr-qc/9812068.
- [13] Harald Dimmelmeier, Jose A. Font, and Ewald Muller. Relativistic simulations of rotational core collapse. I. Methods, initial models, and code tests. *Astron. Astrophys.*, 388:917–935, 2002, astro-ph/0204288.
- [14] Harald Dimmelmeier, Jose A. Font, and Ewald Muller. Gravitational waves from relativistic rotational core collapse. *Astrophys. J.*, 560:L163–L166, 2001, astro-ph/0103088.
- [15] James A. Isenberg. Waveless approximation theories of gravity. *Int. J. Mod. Phys.*, D17:265–273, 2008, gr-qc/0702113.
- [16] J. R. Wilson and G. J. Mathews. Instabilities in close neutron star binaries. *Phys. Rev. Lett.*, 75:4161–4164, 1995.
- [17] J. R. Wilson, G. J. Mathews, and P. Marronetti. Relativistic numerical model for close neutron star binaries. *Phys. Rev.*, D54:1317–1331, 1996, gr-qc/9601017.
- [18] G. J. Mathews and J. R. Wilson. Binary induced neutron-star compression, heating, and collapse. *Astrophys. J.*, 482:929–941, 1997, astro-ph/9701142.
- [19] G. J. Mathews, P. Marronetti, and J. R. Wilson. Relativistic studies of close neutron star binaries. 1997, gr-qc/9710101.
- [20] G. J. Mathews, P. Marronetti, and J. R. Wilson. Relativistic hydrodynamics in close binary systems: Analysis of neutron-star collapse. *Phys. Rev. D*, 58(4):043003, Jul 1998.
- [21] Pedro Marronetti, Grant J. Mathews, and James R. Wilson. Binary neutron stars systems: Irrotational quasi- equilibrium sequences. 1998, gr-qc/9903105.
- [22] P. Marronetti, G. J. Mathews, and J. R. Wilson. Binary neutron-star systems: From the newtonian regime to the last stable orbit. *Phys. Rev. D*, 58(10):107503, Oct 1998.
- [23] P. Marronetti, G. J. Mathews, and J. R. Wilson. Irrotational binary neutron stars in quasiequilibrium. *Phys. Rev. D*, 60(8):087301, Sep 1999.
- [24] Éanna É. Flanagan. Possible explanation for star-crushing effect in binary neutron star simulations. *Phys. Rev. Lett.*, 82(7):1354–1357, Feb 1999.

- [25] G. J. Mathews and J. R. Wilson. Revised relativistic hydrodynamical model for neutron-star binaries. *Phys. Rev.*, D61:127304, 2000, gr-qc/9911047.
- [26] James R. Wilson. Precompression of colliding neutron stars. *Phys. Rev.*, D66:084015, 2002.
- [27] Mark A. Miller. General relativistic decompression of binary neutron stars during inspiral. *Phys. Rev.*, D75:024001, 2007, gr-qc/0510020.
- [28] Marc Favata. Are neutron stars crushed? Gravitomagnetic tidal fields as a mechanism for binary-induced collapse. *Phys. Rev.*, D73:104005, 2006, astro-ph/0510668.
- [29] Masaru Shibata, Koji Uryu, and John L. Friedman. Deriving formulations for numerical computation of binary neutron stars in quasicircular orbits. *Phys. Rev.*, D70:044044, 2004, gr-qc/0407036.
- [30] Gerhard Schaefer and Achamveedu Gopakumar. A minimal no-radiation approximation to Einstein's field equations. *Phys. Rev.*, D69:021501, 2004, gr-qc/0310041.
- [31] Koji Uryu, Francois Limousin, John L. Friedman, Eric Gourgoulhon, and Masaru Shibata. Binary neutron stars in a waveless approximation. *Phys. Rev. Lett.*, 97:171101, 2006, gr-qc/0511136.
- [32] 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).
- [33] A. Lichnerowicz. L'intégration des équations de la gravitation relativiste et le problème des n corps. *J. Math. Pures et Appl.*, **23**:37, (1944).
- [34] A.R. Mitchell and D.F. Griffiths. *The Finite Difference Method in Partial Differential Equations*. John Wiley and Sons, Inc, (1980).
- [35] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comput.*, **31**:333, (1977).
- [36] Frederic A. Rasio and Stuart L. Shapiro. Coalescing binary neutron stars. *Class. Quant. Grav.*, 16:R1–R29, 1999, gr-qc/9902019.
- [37] T.W. Baumgarte and S.L. Shapiro. Numerical relativity and compact binaries. *Phys. Rep.*, **376**:41–131, (2003).

- [38] Frans Pretorius. Binary Black Hole Coalescence. 2007, 0710.1338.
- [39] L. Smarr, A. Cadez, Bryce S. DeWitt, and K. Eppley. Collision of Two Black Holes: Theoretical Framework. *Phys. Rev.*, D14:2443–2452, 1976.
- [40] Peter Anninos, David Hobill, Edward Seidel, Larry Smarr, and Wai-Mo Suen. The Collision of two black holes. *Phys. Rev. Lett.*, 71:2851–2854, 1993, gr-qc/9309016.
- [41] G. B. Cook et al. Boosted three-dimensional black-hole evolutions with singularity excision. *Phys. Rev. Lett.*, 80:2512–2516, 1998, gr-qc/9711078.
- [42] Bernd Bruegmann, Wolfgang Tichy, and Nina Jansen. Numerical simulation of orbiting black holes. *Phys. Rev. Lett.*, 92:211101, 2004, gr-qc/0312112.
- [43] Frans Pretorius. Evolution of Binary Black Hole Spacetimes. *Phys. Rev. Lett.*, 95:121101, 2005, gr-qc/0507014.
- [44] Frans Pretorius. Simulation of binary black hole spacetimes with a harmonic evolution scheme. *Class. Quant. Grav.*, 23:S529–S552, 2006, gr-qc/0602115.
- [45] 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, gr-qc/0511048.
- [46] Manuela Campanelli, C. O. Lousto, and Y. Zlochower. The last orbit of binary black holes. *Phys. Rev.*, D73:061501, 2006, gr-qc/0601091.
- [47] Y. Zlochower, J. G. Baker, Manuela Campanelli, and C. O. Lousto. Accurate black hole evolutions by fourth-order numerical relativity. *Phys. Rev.*, D72:024021, 2005, gr-qc/0505055.
- [48] 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, gr-qc/0511103.
- [49] John G. Baker, Joan Centrella, Dae-Il Choi, Michael Koppitz, and James van Meter. Binary black hole merger dynamics and waveforms. *Phys. Rev.*, D73:104002, 2006, gr-qc/0602026.
- [50] Bela Szilagyi, Denis Pollney, Luciano Rezzolla, Jonathan Thornburg, and Jeffrey Winicour. An explicit harmonic code for black-hole evolution using excision. *Class. Quant. Grav.*, 24:S275–S293, 2007, gr-qc/0612150.

- [51] Peter Diener et al. Accurate evolution of orbiting binary black holes. *Phys. Rev. Lett.*, 96:121101, 2006, gr-qc/0512108.
- [52] Bernd Bruegmann et al. Calibration of Moving Puncture Simulations. *Phys. Rev.*, D77:024027, 2008, gr-qc/0610128.
- [53] John G. Baker, Manuela Campanelli, Frans Pretorius, and Yosef Zlochower. Comparisons of binary black hole merger waveforms. *Class. Quant. Grav.*, 24:S25–S31, 2007, gr-qc/0701016.
- [54] Carlos F. Sopuerta, Nicolas Yunes, and Pablo Laguna. Gravitational recoil velocities from eccentric binary black hole mergers. *Astrophys. J.*, 656:L9–L12, 2007, astro-ph/0611110.
- [55] John G. Baker et al. Getting a kick out of numerical relativity. *Astrophys. J.*, 653:L93–L96, 2006, astro-ph/0603204.
- [56] Jose A. Gonzalez, Ulrich Sperhake, Bernd Bruegmann, Mark Hannam, and Sascha Husa. Total recoil: the maximum kick from nonspinning black-hole binary inspiral. *Phys. Rev. Lett.*, 98:091101, 2007, gr-qc/0610154.
- [57] Frank Herrmann, Ian Hinder, Deirdre Shoemaker, and Pablo Laguna. Unequal mass binary black hole plunges and gravitational recoil. *Class. Quant. Grav.*, 24:S33–S42, 2007.
- [58] Dae-Il Choi et al. Recoiling from a kick in the head-on collision of spinning black holes. *Phys. Rev.*, D76:104026, 2007, gr-qc/0702016.
- [59] Frank Herrmann, Ian Hinder, Deirdre M. Shoemaker, Pablo Laguna, and Richard A. Matzner. Binary Black Holes: Spin Dynamics and Gravitational Recoil. *Phys. Rev.*, D76:084032, 2007, 0706.2541.
- [60] Pedro Marronetti, Wolfgang Tichy, Bernd Bruegmann, Jose Gonzalez, and Ulrich Sperhake. High-spin binary black hole mergers. *Phys. Rev.*, D77:064010, 2008, 0709.2160.
- [61] John G. Baker et al. Modeling kicks from the merger of non-precessing black-hole binaries. *Astrophys. J.*, 668:1140–1144, 2007, astro-ph/0702390.
- [62] J. A. Gonzalez, M. D. Hannam, U. Sperhake, Bernd Bruegmann, and S. Husa. Supermassive kicks for spinning black holes. *Phys. Rev. Lett.*, 98:231101, 2007, gr-qc/0702052.

- [63] Bernd Bruegmann, Jose A. Gonzalez, Mark Hannam, Sascha Husa, and Ulrich Sperhake. Exploring black hole superkicks. *Phys. Rev.*, D77:124047, 2008, 0707.0135.
- [64] Manuela Campanelli, C. O. Lousto, and Y. Zlochower. Gravitational radiation from spinning-black-hole binaries: The orbital hang up. *Phys. Rev.*, D74:041501, 2006, gr-qc/0604012.
- [65] Manuela Campanelli, C. O. Lousto, and Yosef Zlochower. Spin-orbit interactions in black-hole binaries. *Phys. Rev.*, D74:084023, 2006, astro-ph/0608275.
- [66] Manuela Campanelli, Carlos O. Lousto, Yosef Zlochower, Badri Krishnan, and David Merritt. Spin Flips and Precession in Black-Hole-Binary Mergers. *Phys. Rev.*, D75:064030, 2007, gr-qc/0612076.
- [67] Manuela Campanelli, Carlos O. Lousto, Yosef Zlochower, and David Merritt. Maximum gravitational recoil. *Phys. Rev. Lett.*, 98:231102, 2007, gr-qc/0702133.
- [68] Ian Hinder, Birjoo Vaishnav, Frank Herrmann, Deirdre Shoemaker, and Pablo Laguna. Universality and Final Spin in Eccentric Binary Black Hole Inspirals. *Phys. Rev.*, D77:081502, 2008, 0710.5167.
- [69] Ulrich Sperhake et al. Eccentric binary black-hole mergers: The transition from inspiral to plunge in general relativity. *Phys. Rev.*, D78:064069, 2008, 0710.3823.
- [70] Frans Pretorius and Deepak Khurana. Black hole mergers and unstable circular orbits. *Class. Quant. Grav.*, 24:S83–S108, 2007, gr-qc/0702084.
- [71] Matthew C. Washik et al. Binary Black Hole Encounters, Gravitational Bursts and Maximum Final Spin. *Phys. Rev. Lett.*, 101:061102, 2008, 0802.2520.
- [72] John G. Baker et al. Binary black hole late inspiral: Simulations for gravitational wave observations. *Phys. Rev.*, D75:124024, 2007, gr-qc/0612117.
- [73] Alessandra Buonanno, Gregory B. Cook, and Frans Pretorius. Inspiral, merger and ring-down of equal-mass black-hole binaries. *Phys. Rev.*, D75:124018, 2007, gr-qc/0610122.
- [74] Emanuele Berti et al. Inspiral, merger and ringdown of unequal mass black hole binaries: A multipolar analysis. *Phys. Rev.*, D76:064034, 2007, gr-qc/0703053.

- [75] Mark Hannam, Sascha Husa, Ulrich Sperhake, Bernd Bruegmann, and Jose A. Gonzalez. Where post-Newtonian and numerical-relativity waveforms meet. *Phys. Rev.*, D77:044020, 2008, 0706.1305.
- [76] John G. Baker et al. Mergers of non-spinning black-hole binaries: Gravitational radiation characteristics. *Phys. Rev.*, D78:044046, 2008, 0805.1428.
- [77] Mark A. Scheel et al. Solving Einstein's equations with dual coordinate frames. *Phys. Rev.*, D74:104006, 2006, gr-qc/0607056.
- [78] Harald P. Pfeiffer et al. Reducing orbital eccentricity in binary black hole simulations. *Class. Quant. Grav.*, 24:S59–S82, 2007, gr-qc/0702106.
- [79] Michael Boyle et al. High-accuracy comparison of numerical relativity simulations with post-Newtonian expansions. *Phys. Rev.*, D76:124038, 2007, 0710.0158.
- [80] Ulrich Sperhake, Vitor Cardoso, Frans Pretorius, Emanuele Berti, and Jose A. Gonzalez. The high-energy collision of two black holes. *Phys. Rev. Lett.*, 101:161101, 2008, 0806.1738.
- [81] Masaru Shibata, Hirotada Okawa, and Tetsuro Yamamoto. High-velocity collision of two black holes. *Phys. Rev.*, D78:101501, 2008, 0810.4735.
- [82] R. Penrose. Presented at Cambridge University Seminar, Cambridge, England, 1974 (unpublished).
- [83] P. D. D'Eath and P. N. Payne. Gravitational radiation in high speed black hole collisions. 1. Perturbation treatment of the axisymmetric speed of light collision. *Phys. Rev.*, D46:658–674, 1992.
- [84] Masaru Shibata and Koji Uryu. Simulation of merging binary neutron stars in full general relativity: $\Gamma = 2$ case. *Phys. Rev.*, D61:064001, 2000, gr-qc/9911058.
- [85] Masaru Shibata and Koji Uryu. Gravitational waves from the merger of binary neutron stars in a fully general relativistic simulation. *Prog. Theor. Phys.*, 107:265, 2002, gr-qc/0203037.
- [86] Masaru Shibata, Keisuke Taniguchi, and Koji Uryu. Merger of binary neutron stars of unequal mass in full general relativity. *Phys. Rev.*, D68:084020, 2003, gr-qc/0310030.

- [87] Masaru Shibata, Keisuke Taniguchi, and Koji Uryu. Merger of binary neutron stars with realistic equations of state in full general relativity. *Phys. Rev.*, D71:084021, 2005, gr-qc/0503119.
- [88] Masaru Shibata and Keisuke Taniguchi. Merger of binary neutron stars to a black hole: Disk mass, short gamma-ray bursts, and quasinormal mode ringing. *Phys. Rev.*, D73:064027, 2006, astro-ph/0603145.
- [89] Matthew D. Duez, Pedro Marronetti, Stuart L. Shapiro, and Thomas W. Baumgarte. Hydrodynamic simulations in 3+1 general relativity. *Phys. Rev.*, D67:024004, 2003, gr-qc/0209102.
- [90] Mark A. Miller, Philip Gressman, and Wai-Mo Suen. Towards a Realistic Neutron Star Binary Inspiral: Initial Data and Multiple Orbit Evolution in Full General Relativity. *Phys. Rev.*, D69:064026, 2004, gr-qc/0312030.
- [91] Matthew Anderson et al. Simulating binary neutron stars: dynamics and gravitational waves. *Phys. Rev.*, D77:024006, 2008, 0708.2720.
- [92] Luca Baiotti, Bruno Giacomazzo, and Luciano Rezzolla. Accurate evolutions of inspiralling neutron-star binaries: prompt and delayed collapse to black hole. *Phys. Rev.*, D78:084033, 2008, 0804.0594.
- [93] Yuk Tung Liu, Stuart L. Shapiro, Zachariah B. Etienne, and Keisuke Taniguchi. General relativistic simulations of magnetized binary neutron star mergers. *Phys. Rev.*, D78:024012, 2008, 0803.4193.
- [94] Matthew Anderson et al. Magnetized Neutron Star Mergers and Gravitational Wave Signals. *Phys. Rev. Lett.*, 100:191101, 2008, 0801.4387.
- [95] Masaru Shibata and Koji Uryu. Merger of black hole - neutron star binaries: Nonspinning black hole case. *Phys. Rev.*, D74:121503, 2006, gr-qc/0612142.
- [96] Masaru Shibata and Koji Uryu. Merger of black hole-neutron star binaries in full general relativity. *Class. Quant. Grav.*, 24:S125–S138, 2007, astro-ph/0611522.
- [97] Frank Löffler, Luciano Rezzolla, and Marcus Ansorg. Numerical evolutions of a black hole-neutron star system in full general relativity: Head-on collision. *Phys. Rev.*, D74:104018, 2006.

- [98] Zachariah B. Etienne et al. Fully General Relativistic Simulations of Black Hole- Neutron Star Mergers. *Phys. Rev.*, D77:084002, 2008, 0712.2460.
- [99] Tetsuro Yamamoto, Masaru Shibata, and Keisuke Taniguchi. Simulating coalescing compact binaries by a new code SACRA. 2008, 0806.4007.
- [100] J. A. Wheeler. Geons. *Phys. Rev.*, **97**:511, (1955).
- [101] Edwin A. Power and John A. Wheeler. Thermal Geons. *Rev. Mod. Phys.*, 29:480–495, 1957.
- [102] David J. Kaup. Klein-gordon geon. *Phys. Rev.*, 172(5):1331–1342, Aug 1968.
- [103] R. Ruffini and S. Bonazzola. Systems of self gravitating particles in general relativity and the concept of an equation of state. *Phys. Rev.*, **187**:1767, (1969).
- [104] Subrahmanyan Chandrasekhar. The maximum mass of ideal white dwarfs. *Astrophys. J.*, 74:81–82, 1931.
- [105] M. Colpi, S. L. Shapiro, and I. Wasserman. Boson stars: Gravitational equilibria of self-interacting scalar fields. *Phys. Rev. Lett.*, **57**:2485–2488, (1986).
- [106] T. D. Lee. Soliton stars and the critical masses of black holes. *Phys. Rev.*, **D35**:3637, (1987).
- [107] R. Friedberg, T.D. Lee, and Y. Pang. Mini-soliton stars. *Phys. Rev.*, **D35**:3640–57, (1987).
- [108] Marcelo Gleiser. Stability of Boson Stars. *Phys. Rev.*, D38:2376, 1988.
- [109] R. Ferrell and M. Gleiser. Gravitational atoms: Gravitational radiation from excited boson stars. *Phys. Rev.*, **D40**:2524, (1989).
- [110] M. Gleiser and R. Watkins. Gravitational stability of scalar matter. *Nucl. Phys.*, **B319**:733, (1989).
- [111] T. D. Lee and Y. Pang. Stability of mini-boson stars. *Nucl. Phys.*, **B315**:477, (1989).
- [112] Philippe Jetzer. Boson stars. *Phys. Rept.*, 220:163–227, 1992.
- [113] T. D. Lee and Y. Pang. Nontopological solitons. *Phys. Rept.*, 221:251–350, 1992.
- [114] E. Seidel and W. M. Suen. Dynamical evolution of boson stars: Perturbing the ground state. *Phys. Rev.*, **D42**:384–403, (1990).

- [115] J. Balakrishna and W. M. Suen. Dynamical evolution of boson stars. II. excited states and self-interacting fields. *Phys. Rev.*, **D58**:104004, (1998).
- [116] Scott H. Hawley and Mathew W. Choptuik. Boson stars driven to the brink of black hole formation. *Phys. Rev. D*, 62(10):104024, Oct 2000.
- [117] Chi Wai Lai and Matthew W. Choptuik. Final Fate of Subcritical Evolutions of Boson Stars. 2007, 0709.0324.
- [118] C. Gundlach. Critical phenomena in gravitational collapse. *Phys. Rep.*, **376**:339–405, (2003).
- [119] F. E. Schunck and E. W. Mielke. General relativistic boson stars. *Class. Quantum Grav.*, **20**:R301–R356, (2003).
- [120] R. M. Wald. *General Relativity*. University of Chicago Press, Chicago IL, (1984).
- [121] J. W. York, Jr. Kinematics and dynamics of general relativity. In L. Smarr, editor, *Sources of Gravitational Radiation*. Seattle, Cambridge University Press, (1979).
- [122] Tsvi Piran. Numerical codes for cylindrical general relativistic systems. *Journal of Computational Physics*, **35**:254–283, (1980).
- [123] E. Seidel and W. M. Suen. Oscillating soliton stars. *Phys. Rev. Lett.*, 66:1659–1662, 1991.
- [124] M. W. Choptuik. Universality and scaling in gravitational collapse of a massless scalar field. *Phys. Rev. Lett.*, **70**:9, (1993).
- [125] Scott H. Hawley and Matthew W. Choptuik. Numerical evidence for 'multi-scalar stars'. *Phys. Rev.*, D67:024010, 2003, gr-qc/0208078.
- [126] B. Rousseau. Axisymmetric boson stars in the conformally flat approximation. Master's thesis, The University of British Columbia, (2003).
- [127] M. W. Choptuik, E. W. Hirschmann, S. L. Liebling, and F. Pretorius. Critical collapse of the massless scalar field in axisymmetry. *Phys. Rev.*, **D68**:044007, (2003).
- [128] C. W. Lai. *A Numerical Study of Boson Stars*. PhD thesis, The University of British Columbia, (2004), gr-qc/0410040.

- [129] Ignacio Olabarrieta, Jason F. Ventrella, Matthew W. Choptuik, and William G. Unruh. Critical Behavior in the Gravitational Collapse of a Scalar Field with Angular Momentum in Spherical Symmetry. *Phys. Rev.*, D76:124014, 2007, 0708.0513.
- [130] Matthew W. Choptuik and Frans Pretorius. Ultra Relativistic Particle Collisions. 2009, 0908.1780.
- [131] Frank Estabrook, Hugo Wahlquist, Steven Christensen, Bryce DeWitt, Larry Smarr, and Elaine Tsang. Maximally slicing a black hole. *Phys. Rev. D*, 7(10):2814–2817, May 1973.
- [132] C. W. Misner, K. S. Thorne, and J. A. Wheeler. *Gravitation*. W.H. Freeman and Company, New York, (1973).
- [133] Jr. York, James W. Gravitational degrees of freedom and the initial-value problem. *Phys. Rev. Lett.*, 26:1656–1658, 1971.
- [134] Jr. York, James W. Role of conformal three geometry in the dynamics of gravitation. *Phys. Rev. Lett.*, 28:1082–1085, 1972.
- [135] J. W. York. Mapping onto solutions of the gravitational initial value problem. *J. Math. Phys.*, 13:125–130, 1972.
- [136] Jr. York, James W. Conformally invariant orthogonal decomposition of symmetric tensors on Riemannian manifolds and the initial value problem of general relativity. *J. Math. Phys.*, 14:456–464, 1973.
- [137] Niall O’Murchadha and James W. York. Initial - value problem of general relativity. 1. General formulation and physical interpretation. *Phys. Rev.*, D10:428–436, 1974.
- [138] Niall O’Murchadha and James W. York. Initial-value problem of general relativity. 2. Stability of solutions of the initial-value equations. *Phys. Rev.*, D10:437–446, 1974.
- [139] Jeffrey M. Bowen and James W. York, Jr. Time asymmetric initial data for black holes and black hole collisions. *Phys. Rev.*, D21:2047–2056, 1980.
- [140] J. W. York, Jr. The initial value problem and dynamics. In N. Deruelle and T. Piran, editors, *Gravitational Radiation*. North-Holland, Amsterdam, (1983).

- [141] S. Deser. Covariant decomposition of symmetric tensors and the gravitational cauchy problem. *Annales de l'institut Henri Poincaré (A) Physique théorique*, **7**(2):149–188, (1967).
- [142] M. Berger and D. Ebin. Some decompositions of the space of symmetric tensors on a riemannian manifold. *J. Differential Geometry*, **3**:379–392, (1969).
- [143] Kip S. Thorne. Multipole expansions of gravitational radiation. *Rev. Mod. Phys.*, 52(2):299–339, Apr 1980.
- [144] Ericourgoulhon. 3+1 Formalism and Bases of Numerical Relativity. 2007, gr-qc/0703035.
- [145] <http://www.maplesoft.com/>.
- [146] M. Choptuik. <ftp://laplace.physics.ubc.ca/pub/TensorV6/>.
- [147] Matthew W. Choptuik et al. Towards the final fate of an unstable black string. *Phys. Rev.*, D68:044001, 2003, gr-qc/0304085.
- [148] Frans Pretorius. Numerical Relativity Using a Generalized Harmonic Decomposition. *Class. Quant. Grav.*, 22:425–452, 2005, gr-qc/0407110.
- [149] U. Trottenberg, C. W. Oosterlee, and A. Schueller. *Multigrid*. Academic Press, (2001).
- [150] A. Brandt. Guide to multigrid development. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods: Proceedings of the Conference Held at Köln-Porz, November 23-27, 1981*. Berlin: Springer-Verlag, (1982).
- [151] Jr. J. W. York and T. Piran. The initial value problem and beyond. In R. A. Matzner and L. Shepley, editors, *Spacetime and Geometry / Alfred Schild Lectures*, page 552, (1982).
- [152] Joseph Olinger Marsha J Berger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53:484–512, March 1984.
- [153] Niall O’Murchadha and James W. York. Gravitational energy. *Phys. Rev.*, D10:2345–2357, 1974.
- [154] M. Choptuik. Unpublished Numerical Relativity Lecture Notes (1997).
- [155] Evans C. R. *A method for numerical relativity: Simulation of axisymmetric gravitational collapse and gravitational radiation generation*. PhD thesis, The University of Texas at Austin, (1984).

- [156] Tsvi Piran James M. Bardeen. General relativistic axisymmetric rotating systems: Coordinates and equations. *Physics Reports*, **96**:205–250, (1983).
- [157] M. W. Choptuik. *A Study of Numerical Techniques for Radiative Problems in General Relativity*. PhD thesis, The University of British Columbia, (1986).
- [158] O. Rinne. *Axisymmetric Numerical Relativity*. PhD thesis, Trinity College, Cambridge, (2005), gr-qc/0601064.
- [159] S. W. Hawking and G. F. R. Ellis. *The Large Scale Structure of Space-Time*. Cambridge University Press, Cambridge, (1973).
- [160] R. Burden and J. Faires. *Numerical Analysis*. Thomson Brooks-Cole Publishing, 8th edition, (2005).
- [161] S. H. Hawley. *Scalar Analogues of Compact Astrophysical Systems*. PhD thesis, The University of Texas at Austin, (2000).
- [162] S. L. Shapiro and S. A. Teukolsky. *Black Holes, White Dwarfs, and Neutron Stars*. Wiley, (1983).
- [163] M. W. Choptuik. *Lectures for VII Mexican School on Gravitation and Mathematical Physics, Numerical Analysis for Numerical Relativists*, (2006).
- [164] A. George and J. W. H. Liu. *Computer Solutions of Large Sparse Positive Definite Systems*. Prentice Hall, (1981).
- [165] L. F. Richardson. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A*, **210**:307–357, (1911).
- [166] B. Gustafsson. *High Order Difference Methods for Time Dependent PDEs*. Springer, (2008).
- [167] <http://www.netlib.org/lapack>.
- [168] Richard S. Varga. *Matrix Iterative Analysis, 2nd Ed*. Springer, Berlin, (2000).
- [169] B. Gustafsson, H-O. Kreiss, and J. Olinger. *Time Dependent Problems and Difference Methods*. John Wiley and Sons, Inc, (1995).

- [170] R.D. Richtmyer and K.W. Morton. *Difference Methods and Initial-Value Problems*. John Wiley and Sons, Inc, 2nd edition, (1967).
- [171] M.W. Choptuik. Critical behaviour in scalar field collapse. In D. Hobill, A. Burd, and A. Coley, editors, *Deterministic Chaos in General Relativity*, pages 155–175. Springer, (1994).
- [172] Saul A. Teukolsky. On the Stability of the Iterated Crank-Nicholson Method in Numerical Relativity. *Phys. Rev.*, D61:087501, 2000, gr-qc/9909026.
- [173] D. Choi. *Numerical Studies of Nonlinear Schrödinger and Klein-Gordon Systems: Techniques and Applications*. PhD thesis, The University of Texas at Austin, (1998).
- [174] Dae-II Choi. Collision of gravitationally self-bound Bose-Einstein condensates. *Phys. Rev.*, **A66**:063609, 2002.
- [175] R. L. Marsa. *Radiative Problems in Black-Hole Spacetimes*. PhD thesis, The University of Texas at Austin, (1995).
- [176] M. Choptuik and R. L. Marsa. <http://laplace.physics.ubc.ca/People/matt/Rnpl/index.html>.
- [177] F. Pretorius. http://laplace.physics.ubc.ca/Doc/pamr/AMRD_ref.pdf.
- [178] F. Pretorius. http://laplace.physics.ubc.ca/Doc/pamr/PAMR_ref.pdf.
- [179] B. C. Mundim. See BSIDPA distribution at <ftp://laplace.physics.ubc.ca/pub/>.
- [180] <http://www.netlib.org/odepack>.
- [181] B. C. Mundim. See PDEFDAOFF distribution at <ftp://laplace.physics.ubc.ca/pub/>.

Appendix A

BSIDPA—Boson Star Initial Data Function in Polar Areal Coordinates

`bsidpa` is a freely available⁴³ FORTRAN 77 subroutine that generates numerical solutions describing static, spherically symmetric boson stars as described in Chap. 3 (the reader should refer to that chapter for details concerning notation, definitions of functions used below, etc. which will not be repeated here). The routine name is an acronym for **B**oson **S**tar **I**nitial **D**ata in **P**olar-**A**real coordinates. `bsidpa` has been written so that the user can specify a general polynomial self-interaction potential, $U(\phi_0(R))$, for the scalar field:

$$U(\phi_0) = \sum_{i=1}^{p_n} p_i \phi_0^i, \quad (\text{A.1})$$

where p_n is the degree of the interaction polynomial, and the coefficients p_i , $i = 1, 2, \dots, p_n$ are user-supplied. Note that $p_n = 2$, with $p_1 = 0$ and $p_2 = 1$, is the case considered in this thesis. In other words, the potential only has a mass term, which leads to static configurations that are sometimes called mini boson stars. As detailed in Chap. 3, for any given potential, the static boson stars form a one-parameter family, and it is convenient to use the central value of $\phi_0(r)$ —i.e. $\phi_0(0)$ —as the family parameter. We also recall that the ODE system (3.49)–(3.52) (hereafter often referred to as “the ODEs”), which `bsidpa` solves numerically using the `lsoda` integrator from ODEPACK [180], constitutes an eigenvalue problem, with eigenvalue (eigenfrequency) $\omega = \omega(\phi_0(0))$. Thus, another key input to `bsidpa` is the central field value, $\phi_0(0)$. If the user knows the corresponding eigenfrequency, that can also be supplied. However, if ω is not known for the given value of $\phi_0(0)$, `bsidpa` will attempt to compute it using a bisection algorithm. The bisection is based on the observation that, generically, as $R \rightarrow \infty$ we have $\phi_0(R) \rightarrow \infty$ for $\omega_{\text{hi}} < \omega$, while $\phi_0(R) \rightarrow -\infty$ for $\omega_{\text{lo}} > \omega$ ⁴⁴. In this latter case, the user can either supply an initial bracket

⁴³`bsidpa` and supporting routines are maintained in the UBC numerical relativity group’s `ftp` repository [179].

⁴⁴Observe that in Chap. 3 we used a notation such that $\omega_- \equiv \omega_{\text{hi}}$ and $\omega_+ \equiv \omega_{\text{lo}}$.

$[\omega_{hi}, \omega_{lo}]$ such that $\omega_{hi} \leq \omega \leq \omega_{lo}$, or `bsidpa` can attempt to locate such a bracket automatically.

A typical invocation of `bsidpa` is illustrated by the following fragment of FORTRAN 77 code:

```

...

integer  sht, n, pn, pt, dft, tail_type
real*8   phi0, rmax, lsoda_tol, w_tol, dr, whi, wlo, wshoot, wresca
real*8   a(n), alpha(n), phi(n), pp(n), m(n), zr(n), r(n)
real*8   p(pn)

...

call bsidpa(a,alpha,phi,pp,m,zr,r,p,phi0,rmax,lsoda_tol,w_tol,dr,
&          whi,wlo,wshoot,wresca,sht,n,pn,pt,dft,tail_type)

...

```

The various arguments to the routine can be classified as being either inputs or outputs. Further, some input arguments are required, while others are “optional”, in the sense that a flag can be set so that they are assigned default values.⁴⁵

Input Arguments

The following input arguments must be supplied:

phi0: The central value of the scalar field, $\phi_0(0)$

rmax: The range of the integration: i.e. the ODEs are integrated from $R = 0$ to $R = \mathbf{rmax}$.

pt: The potential type. Currently, the only valid value for **pt** is

- **pt = 1**: Polynomial potential as given by (A.1).

p(pn): real*8 vector of length **pn** defining the coefficients p_i of the polynomial potential.

n: Number of grid points in the interval $0 \leq R \leq \mathbf{rmax}$ at which to compute the solution of the ODEs. Specifically, the solution will be computed at $R_j \equiv (j - 1)\Delta_R, j = 1, 2, \dots, \mathbf{n}$, where $\Delta_R \equiv \mathbf{rmax}/(\mathbf{n} - 1)$.

⁴⁵Note, however, that since FORTRAN 77 has no provision for subprograms with varying-length argument lists, any invocation of `bsidpa` must have exactly 23 arguments

sht: Flag that controls how the eigenfrequency, **wshoot**, is to be determined by **bsidpa**. Valid values are as follows:

- **sht** = 0: User supplies **wshoot**.
- **sht** = 1: Routine uses a shooting technique and bisection to determine **wshoot** to within the user-specified tolerance **w_tol** (see below). Also see documentation of **dft** below which controls whether the routine or user is responsible for determining an initial bracket for **wshoot**.

wshoot: Solution eigenfrequency as defined above. Note: for **sht** = 0 this is an input argument, while for **sht** = 1 it is an output argument.

dft: Flag that controls use of default values for “optional” arguments as follows:

- **dft** = 0: Routine uses default values for **lsoda_tol**, **w_tol**, **dr**, **whi** and **wlo**.
- **dft** = 1: Routine uses default values for **dr**, **whi** and **wlo**; user must supply values for **lsoda_tol** and **w_tol**.
- **dft** = 2: User must supply values for **lsoda_tol**, **w_tol**, **dr**, **whi** and **wlo**.

The “optional” input parameters as well as their default values, are defined as follows:

lsoda_tol: **lsoda** tolerance parameter. **lsoda** will attempt to keep the local error in the solution of the ODEs below this value. Default: **lsoda_tol** = 10^{-10} .

w_tol: Tolerance for computation of the eigenfrequency estimate, **wshoot**. The bisection/shooting algorithm stops when $|\omega_{\text{low}} - \omega_{\text{hi}}| \leq \mathbf{w_tol}$. Default: **w_tol** = 10^{-9}

dr: Solution output interval used by **bsidpa** when the algorithm is determining a bracket for the eigenvalue **wshoot**. It is useful to be able to set **dr** independently of the value Δ_R defined from **rmax** and **n** (see above) since if the estimate **wshoot** is sufficiently poor, **lsoda** may fail on the interval $0 \leq R \leq \Delta_R$. In this case, setting **dr** $\ll \Delta_R$ may aid in the bracket-locating process. Default: **dr** = 10^{-4} .

[**whi**, **wlo**]: Values defining the initial bracket for eigenfrequency **wshoot**, and which should thus satisfy **whi** \leq **wshoot** \leq **wlo**. More specifically, for $\omega = \mathbf{whi}$ and **wlo**, respectively the solution $\phi_0(R)$ should tend to $+\infty$ and $-\infty$, respectively, for large R . *bsidpa* does

have an algorithm encoded to automatically look for an appropriate bracket, but this must be used with caution. Automatic bracketing becomes particularly difficult in cases where eigenfrequencies corresponding to the ground state, and one or more of the excited states, are close to one another. That can happen, for example, when one or more of the potential coefficients, p_i , are large. When performing parameter space surveys in such instances, it is recommended that a continuation method be used, whereby an initial bracket for a solution with $\phi_0(0) + \delta\phi_0(0)$ is given by $[\omega(\phi_0(0)) - \delta\omega, \omega(\phi_0(0)) + \delta\omega]$, and where $\delta\phi_0(0)$ and $\delta\omega$ are adjusted as necessary to ensure proper bracketing. Default: `[whi,wlo] = [1.00,1.01]` (appropriate for a mini boson star with $m = 1$ ($p_1 = 0, p_2 = 1$) and $\phi_0(0) = 0.01$).

`tail_type`: Flag for controlling what closed form expression is used to define ϕ_0 at large values of R , via a fitting procedure to the numerical solution of the ODEs. Currently, only `tail_type = 1` is implemented, which results in a fit to

$$T_1(R) = A \exp(-BR), \quad (\text{A.2})$$

where A and B are coefficients determined from the fit.

Output Arguments

`bsidpa` returns the following output arguments, all of which are vectors of length `n`:

`r(n)`: The areal coordinate, R .

`a(n)`: The computed radial metric function, $a(R)$.

`alpha(n)`: The computed lapse function, $\alpha(R)$, rescaled so that $\lim_{R \rightarrow \infty} \alpha(R) = 1$.

`phi(n)`: The computed scalar field modulus, $\phi_0(R)$.

`pp(n)`: The computed derivative of the scalar field modulus, $\Phi_0(R)$.

`m(n)`: The mass aspect function $m(R) \equiv R(1 - a(R)^{-2})/2$.

`zr(n)`: The “compactness function” $z(R) \equiv 2m(R)/R$. This function provides a measure of how gravitationally compact a given star is.

`wresca`: Rescaled value of ω .

Tab. A.1 lists eigenfrequencies calculated using `bsidpa` for a sequence of values of the family parameter, $\phi_0(0)$, for the case $U(\phi_0) = (\phi_0)^2$. Various properties of these stars are given in Tab. A.2. These include the cutoff radius, R_{cutoff} , the derivative of $m(R)$ at the cutoff, the ADM mass, M_{ADM} , and the two measures of the stellar radius, R_{99} and R_{95} , that were defined in Sec. 3.4. Note that the contribution to the total mass from the tail is negligible in all cases. Tab. A.3 lists additional properties including $\min_R \alpha(R)$, $\max_R a(R)$, $\max_R z(R)$, the radius, $R_{\text{max}z}$, at which $z(R)$ attains its maximum, and the Schwarzschild radius, $R_S \equiv 2M_{\text{ADM}}$, associated with each star. The values of $\max_R z(R)$ are particularly noteworthy, clearly showing that the overall gravitational field of the stars gets stronger as $\phi_0(0)$ increases.

$\phi_0(0)$	wshoot	wresca
0.005	1.01173913657665	0.987921656870952
0.006	1.01414229214191	0.985551416322374
0.007	1.01656427949667	0.983196152260810
0.008	1.01900526873767	0.980855788252950
0.009	1.02146543309092	0.978530270851457
0.01	1.02394494622946	0.976219519074950
0.02	1.04984409445897	0.953910295230624
0.03	1.07788555219769	0.933010708580807
0.04	1.10828046126291	0.913467522154004
0.05	1.14126635619905	0.895235350281201
0.06	1.17711107464507	0.878276194485022
0.07	1.21611731750891	0.862559036385476
0.08	1.25862797105685	0.848059515066958
0.09	1.30503232180141	0.834759601661149

Table A.1: Central scalar field values and eigenfrequencies for mini boson stars (i.e. for boson stars with interaction potential $U(\phi_0) = (\phi_0)^2$). Listed are bare and rescaled values of the eigenfrequency, `wshoot` and `wresca`, respectively, for a sequence of central scalar field values, $\phi_0(0)$. The calculations were performed using `rmax` = 100, `n` = $2^{16} + 1$, `dr` = 10^{-4} , and default values for all other parameters.

$\phi_0(0)$	R_{cutoff}	$dm/dR _{R_{\text{cutoff}}}$	M_{ADM}	R_{99}	R_{95}
0.005	91.12	1.34×10^{-9}	0.2646	35.78	28.15
0.006	81.42	1.13×10^{-9}	0.2878	32.58	25.62
0.007	73.58	1.56×10^{-9}	0.3086	30.09	23.65
0.008	71.63	0.84×10^{-9}	0.3275	28.08	22.06
0.009	65.72	0.92×10^{-9}	0.3449	26.40	20.74
0.01	63.85	1.22×10^{-9}	0.3609	24.99	19.63
0.02	45.49	2.06×10^{-9}	0.4751	17.26	13.51
0.03	36.62	2.62×10^{-9}	0.5424	13.77	10.74
0.04	31.99	3.25×10^{-9}	0.5844	11.67	9.065
0.05	28.39	3.94×10^{-9}	0.6103	10.22	7.909
0.06	25.44	4.83×10^{-9}	0.6251	9.148	7.048
0.07	23.74	5.37×10^{-9}	0.6319	8.311	6.375
0.08	22.16	5.99×10^{-9}	0.6327	7.637	5.830
0.09	20.84	7.21×10^{-9}	0.6290	7.081	5.379

Table A.2: Mini Boson Star Properties: Tail, Mass and Radius. This table lists the location, R_{cutoff} , at which $\phi_0(R)$ was fit to (A.2), $dm/dR|_{R_{\text{cutoff}}}$, the ADM mass of the star, M_{ADM} , and the areal radii containing 99% M_{ADM} and 95% M_{ADM} . The computational parameters used to obtain this data are described in the caption of Tab. A.1. Note that the derivative of the mass aspect function at R_{cutoff} is always extremely small, implying that the contribution to M_{ADM} from the tail is negligible.

$\phi_0(0)$	$\min \alpha(R)$	$\max \psi(R)$	$\max z(R)$	$R_{\max z}$	R_S
0.005	0.9764	1.0111	0.0207	19.03	0.529
0.006	0.9718	1.0134	0.0248	17.32	0.575
0.007	0.9672	1.0157	0.0288	15.98	0.617
0.008	0.9626	1.0180	0.0328	14.90	0.655
0.009	0.9580	1.0203	0.0367	14.00	0.690
0.01	0.9534	1.0226	0.0406	13.24	0.722
0.02	0.9086	1.0454	0.0780	9.05	0.950
0.03	0.8656	1.0678	0.1124	7.14	1.085
0.04	0.8242	1.0898	0.1441	5.98	1.169
0.05	0.7844	1.1114	0.1734	5.17	1.220
0.06	0.7461	1.1325	0.2004	4.56	1.250
0.07	0.7093	1.1532	0.2253	4.08	1.264
0.08	0.6738	1.1735	0.2483	3.69	1.265
0.09	0.6397	1.1933	0.2694	3.36	1.258

Table A.3: Mini Boson Star Properties: Extrema of Metric Components. This table lists the minimum value of the lapse function, $\min \alpha(R)$, the maximum value of the conformal factor, $\max \psi(R)$ and the maximum values of the compactness function $\max z(R)$, as well as the location, $R_{\max z}$, at which $z(R)$ attains its maximum. For comparison purposes, the Schwarzschild radius, $R_S \equiv 2M_{\text{ADM}}$, associated with each star, is also tabulated. The computational parameters used to obtain this data are described in the caption of Tab. A.1.

Appendix B

Finite Difference Approximations

The technique of finite differencing is the most common approach to the discretization of time dependent PDEs that has been used in numerical relativity to date. The specific finite difference approximations (FDAs) that we employ in this thesis are all straightforward, and the novice can consult any basic text on the subject, such as Mitchell and Griffiths [34], for details on how they can be derived using, for example, Taylor series expansion. The main purpose of this appendix is to collect and explicitly display all of the FDAs that are used in the 3D finite difference code described in the body of the thesis. In addition, in Sec. B.2 we briefly describe a technique which is useful for deriving FDAs with good regularity properties when one is working in curvilinear coordinates (spherical polar, cylindrical etc.).

B.1 Finite Difference Operators

We first recall (Sec. 4.1) that we adopt Cartesian (rectangular) coordinates for our computations, and that our spatial solution domain is then defined by

$$x_{\min} \leq x \leq x_{\max}, \tag{B.1}$$

$$y_{\min} \leq y \leq y_{\max}, \tag{B.2}$$

$$z_{\min} \leq z \leq z_{\max}, \tag{B.3}$$

where x_{\min} , x_{\max} , y_{\min} , y_{\max} , z_{\min} and z_{\max} are prescribed values.

We discretize this domain by introducing a finite difference grid (or mesh), in which the same constant mesh spacing, h , is used in each of the three coordinate directions. The discrete spatial

coordinates (x_i, y_j, z_k) that label the grid points are given by

$$x_i = x_{\min} + (i - 1)h, \quad i = 1 \dots n_x, \quad (\text{B.4})$$

$$y_j = y_{\min} + (j - 1)h, \quad j = 1 \dots n_y, \quad (\text{B.5})$$

$$z_k = z_{\min} + (k - 1)h, \quad k = 1 \dots n_z, \quad (\text{B.6})$$

such that

$$x_1 = x_{\min} \quad \text{and} \quad x_{n_x} = x_{\max}, \quad (\text{B.7})$$

$$y_1 = y_{\min} \quad \text{and} \quad y_{n_y} = y_{\max}, \quad (\text{B.8})$$

$$z_1 = z_{\min} \quad \text{and} \quad z_{n_z} = z_{\max}. \quad (\text{B.9})$$

Here there is an implicit assumption that each of the ranges $x_{\max} - x_{\min}$, $y_{\max} - y_{\min}$ and $z_{\max} - z_{\min}$ is evenly divisible by h .

Furthermore, assuming that the calculation is performed on the time interval $0 \leq t \leq t_{\max}$, we similarly discretize the time coordinate by defining values, t^n , given by

$$t^n = (n - 1)\lambda h, \quad t = 1 \dots n_t. \quad (\text{B.10})$$

Here, λ —which is the ratio of the temporal and spatial mesh spacings—is known as the Courant number. It will generally satisfy $\lambda < 1$ from stability considerations, and will be held constant when we perform a sequence of computations in which h is varied (when we are performing a convergence test, for example). Again, there is a tacit assumption that λh evenly divides t_{\max} .

For a generic solution unknown, $u(t, x, y, z)$, which is to be approximated on the grid, we then introduce the standard finite difference notation

$$u_{i,j,k}^n \equiv u(t^n, x_i, y_j, z_k), \quad (\text{B.11})$$

for the so-called grid-function values, $u_{i,j,k}^n$.

Having dispensed with basic definitions, we observe that the finite difference approximations that we have used fall into two sets. Members of the first set, which are listed in Tables B.1–

B.4, were used in the actual discretization of the PDEs governing our model, and are all so-called centred approximations. The second set, enumerated in Tables B.5–B.12, were used to compute independent residuals (see Sec. 4.2.2), and all of these are either forward or backward approximations. In all cases, the approximations are second order in the mesh spacing, h . This includes the FDA (B.15) for the first time derivative, $\partial u/\partial t$, which is used in differencing the evolution equations (2.173) and (2.174) using a Crank-Nicholson scheme, and where the difference equations are centred at the “fictitious” grid point $(t^{n+1/2}, x_i, y_j, z_k) \equiv (t^n + \lambda h/2, x_i, y_j, z_k)$.

$$\frac{\partial u}{\partial x} \rightarrow \frac{u_{i+1,j,k}^n - u_{i-1,j,k}^n}{2h} \quad (\text{B.12})$$

$$\frac{\partial u}{\partial y} \rightarrow \frac{u_{i,j+1,k}^n - u_{i,j-1,k}^n}{2h} \quad (\text{B.13})$$

$$\frac{\partial u}{\partial z} \rightarrow \frac{u_{i,j,k+1}^n - u_{i,j,k-1}^n}{2h} \quad (\text{B.14})$$

$$\frac{\partial u}{\partial t} \rightarrow \frac{u_{i,j,k}^{n+1} - u_{i,j,k}^n}{\lambda h} \quad (\text{B.15})$$

Table B.1: Centred FDAs of First Order Derivatives.

$$\mu_t u \equiv \frac{u_{i,j,k}^{n+1} + u_{i,j,k}^n}{2} \quad (\text{B.16})$$

Table B.2: Time Averaging Operator.

$$\frac{\partial^2 u}{\partial x^2} \rightarrow \frac{u_{i+1,j,k}^n - 2u_{i,j,k}^n + u_{i-1,j,k}^n}{h^2} \quad (\text{B.17})$$

$$\frac{\partial^2 u}{\partial y^2} \rightarrow \frac{u_{i,j+1,k}^n - 2u_{i,j,k}^n + u_{i,j-1,k}^n}{h^2} \quad (\text{B.18})$$

$$\frac{\partial^2 u}{\partial z^2} \rightarrow \frac{u_{i,j,k+1}^n - 2u_{i,j,k}^n + u_{i,j,k-1}^n}{h^2} \quad (\text{B.19})$$

Table B.3: Centred FDAs of Second Order Derivatives.

$$\frac{\partial^2 u}{\partial x \partial y} \rightarrow \frac{u_{i-1,j-1,k}^n - u_{i-1,j+1,k}^n - u_{i+1,j-1,k}^n + u_{i+1,j+1,k}^n}{4h^2} \quad (\text{B.20})$$

$$\frac{\partial^2 u}{\partial y \partial z} \rightarrow \frac{u_{i,j-1,k-1}^n - u_{i,j+1,k-1}^n - u_{i,j-1,k+1}^n + u_{i,j+1,k+1}^n}{4h^2} \quad (\text{B.21})$$

$$\frac{\partial^2 u}{\partial x \partial z} \rightarrow \frac{u_{i-1,j,k-1}^n - u_{i+1,j,k-1}^n - u_{i-1,j,k+1}^n + u_{i+1,j,k+1}^n}{4h^2} \quad (\text{B.22})$$

Table B.4: Centred FDAs of Mixed Second Order Derivatives.

$$\frac{\partial u}{\partial x} \rightarrow \frac{-3u_{i,j,k}^n + 4u_{i+1,j,k}^n - u_{i+2,j,k}^n}{2h} \quad (\text{B.23})$$

$$\frac{\partial u}{\partial y} \rightarrow \frac{-3u_{i,j,k}^n + 4u_{i,j+1,k}^n - u_{i,j+2,k}^n}{2h} \quad (\text{B.24})$$

$$\frac{\partial u}{\partial z} \rightarrow \frac{-3u_{i,j,k}^n + 4u_{i,j,k+1}^n - u_{i,j,k+2}^n}{2h} \quad (\text{B.25})$$

Table B.5: Forward FDAs of first order derivatives.

$$\frac{\partial u}{\partial x} \rightarrow \frac{3u_{i,j,k}^n - 4u_{i-1,j,k}^n + u_{i-2,j,k}^n}{2h} \quad (\text{B.26})$$

$$\frac{\partial u}{\partial y} \rightarrow \frac{3u_{i,j,k}^n - 4u_{i,j-1,k}^n + u_{i,j-2,k}^n}{2h} \quad (\text{B.27})$$

$$\frac{\partial u}{\partial z} \rightarrow \frac{3u_{i,j,k}^n - 4u_{i,j,k-1}^n + u_{i,j,k-2}^n}{2h} \quad (\text{B.28})$$

Table B.6: Backward FDAs of First Order Derivatives.

$$\frac{\partial^2 u}{\partial x^2} \rightarrow \frac{2u_{i,j,k}^n - 5u_{i+1,j,k}^n + 4u_{i+2,j,k}^n - u_{i+3,j,k}^n}{h^2} \quad (\text{B.29})$$

$$\frac{\partial^2 u}{\partial y^2} \rightarrow \frac{2u_{i,j,k}^n - 5u_{i,j+1,k}^n + 4u_{i,j+2,k}^n - u_{i,j+3,k}^n}{h^2} \quad (\text{B.30})$$

$$\frac{\partial^2 u}{\partial z^2} \rightarrow \frac{2u_{i,j,k}^n - 5u_{i,j,k+1}^n + 4u_{i,j,k+2}^n - u_{i,j,k+3}^n}{h^2} \quad (\text{B.31})$$

Table B.7: Forward FDAs of Second Order Derivatives.

$$\frac{\partial^2 u}{\partial x^2} \rightarrow \frac{2u_{i,j,k}^n - 5u_{i-1,j,k}^n + 4u_{i-2,j,k}^n - u_{i-3,j,k}^n}{h^2} \quad (\text{B.32})$$

$$\frac{\partial^2 u}{\partial y^2} \rightarrow \frac{2u_{i,j,k}^n - 5u_{i,j-1,k}^n + 4u_{i,j-2,k}^n - u_{i,j-3,k}^n}{h^2} \quad (\text{B.33})$$

$$\frac{\partial^2 u}{\partial z^2} \rightarrow \frac{2u_{i,j,k}^n - 5u_{i,j,k-1}^n + 4u_{i,j,k-2}^n - u_{i,j,k-3}^n}{h^2} \quad (\text{B.34})$$

Table B.8: Backward FDAs of Second Order Derivatives.

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial y} \rightarrow & \frac{9u_{i,j,k}^n - 12u_{i,j+1,k}^n + 3u_{i,j+2,k}^n - 12u_{i+1,j,k}^n + 16u_{i+1,j+1,k}^n - 4u_{i+1,j+2,k}^n}{4h^2} \\ & + \frac{3u_{i+2,j,k}^n - 4u_{i+2,j+1,k}^n + u_{i+2,j+2,k}^n}{4h^2} \end{aligned} \quad (\text{B.35})$$

$$\begin{aligned} \frac{\partial^2 u}{\partial y \partial z} \rightarrow & \frac{9u_{i,j,k}^n - 12u_{i,j,k+1}^n + 3u_{i,j,k+2}^n - 12u_{i,j+1,k}^n + 16u_{i,j+1,k+1}^n - 4u_{i,j+1,k+2}^n}{4h^2} \\ & + \frac{3u_{i,j+2,k}^n - 4u_{i,j+2,k+1}^n + u_{i,j+2,k+2}^n}{4h^2} \end{aligned} \quad (\text{B.36})$$

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial z} \rightarrow & \frac{9u_{i,j,k}^n - 12u_{i,j,k+1}^n + 3u_{i,j,k+2}^n - 12u_{i+1,j,k}^n + 16u_{i+1,j,k+1}^n - 4u_{i+1,j,k+2}^n}{4h^2} \\ & + \frac{3u_{i+2,j,k}^n - 4u_{i+2,j,k+1}^n + u_{i+2,j,k+2}^n}{4h^2} \end{aligned} \quad (\text{B.37})$$

Table B.9: Forward-Forward FDAs of Mixed Second Order Derivatives.

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial y} \rightarrow & \frac{-3u_{i,j-2,k}^n + 12u_{i,j-1,k}^n - 9u_{i,j,k}^n + 4u_{i+1,j-2,k}^n - 16u_{i+1,j-1,k}^n}{4h^2} \\ & + \frac{12u_{i+1,j,k}^n - u_{i+2,j-2,k}^n + 4u_{i+2,j-1,k}^n - 3u_{i+2,j,k}^n}{4h^2} \end{aligned} \quad (\text{B.38})$$

$$\begin{aligned} \frac{\partial^2 u}{\partial y \partial z} \rightarrow & \frac{-3u_{i,j,k-2}^n + 12u_{i,j,k-1}^n - 9u_{i,j,k}^n + 4u_{i,j+1,k-2}^n - 16u_{i,j+1,k-1}^n}{4h^2} \\ & + \frac{12u_{i,j+1,k}^n - u_{i,j+2,k-2}^n + 4u_{i,j+2,k-1}^n - 3u_{i,j+2,k}^n}{4h^2} \end{aligned} \quad (\text{B.39})$$

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial z} \rightarrow & \frac{-3u_{i,j,k-2}^n + 12u_{i,j,k-1}^n - 9u_{i,j,k}^n + 4u_{i+1,j,k-2}^n - 16u_{i+1,j,k-1}^n}{4h^2} \\ & + \frac{12u_{i+1,j,k}^n - u_{i+2,j,k-2}^n + 4u_{i+2,j,k-1}^n - 3u_{i+2,j,k}^n}{4h^2} \end{aligned} \quad (\text{B.40})$$

Table B.10: Forward-Backward FDAs of Mixed Second Order Derivatives.

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial y} \rightarrow & \frac{-3u_{i-2,j,k}^n + 4u_{i-2,j+1,k}^n - u_{i-2,j+2,k}^n + 12u_{i-1,j,k}^n - 16u_{i-1,j+1,k}^n}{4h^2} \\ & + \frac{4u_{i-1,j+2,k}^n - 9u_{i,j,k}^n + 12u_{i,j+1,k}^n - 3u_{i,j+2,k}^n}{4h^2} \end{aligned} \quad (\text{B.41})$$

$$\begin{aligned} \frac{\partial^2 u}{\partial y \partial z} \rightarrow & \frac{-3u_{i,j-2,k}^n + 4u_{i,j-2,k+1}^n - u_{i,j-2,k+2}^n + 12u_{i,j-1,k}^n - 16u_{i,j-1,k+1}^n}{4h^2} \\ & + \frac{4u_{i,j-1,k+2}^n - 9u_{i,j,k}^n + 12u_{i,j,k+1}^n - 3u_{i,j,k+2}^n}{4h^2} \end{aligned} \quad (\text{B.42})$$

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial z} \rightarrow & \frac{-3u_{i-2,j,k}^n + 4u_{i-2,j,k+1}^n - u_{i-2,j,k+2}^n + 12u_{i-1,j,k}^n - 16u_{i-1,j,k+1}^n}{4h^2} \\ & + \frac{4u_{i-1,j,k+2}^n - 9u_{i,j,k}^n + 12u_{i,j,k+1}^n - 3u_{i,j,k+2}^n}{4h^2} \end{aligned} \quad (\text{B.43})$$

Table B.11: Backward-Forward FDAs of Mixed Second Order Derivatives.

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial y} \rightarrow & \frac{u_{i-2,j-2,k}^n - 4u_{i-2,j-1,k}^n + 3u_{i-2,j,k}^n - 4u_{i-1,j-2,k}^n + 16u_{i-1,j-1,k}^n - 12u_{i-1,j,k}^n}{4h^2} \\ & + \frac{3u_{i,j-2,k}^n - 12u_{i,j-1,k}^n + 9u_{i,j,k}^n}{4h^2} \end{aligned} \quad (\text{B.44})$$

$$\begin{aligned} \frac{\partial^2 u}{\partial y \partial z} \rightarrow & \frac{u_{i,j-2,k-2}^n - 4u_{i,j-2,k-1}^n + 3u_{i,j-2,k}^n - 4u_{i,j-1,k-2}^n + 16u_{i,j-1,k-1}^n - 12u_{i,j-1,k}^n}{4h^2} \\ & + \frac{3u_{i,j,k-2}^n - 12u_{i,j,k-1}^n + 9u_{i,j,k}^n}{4h^2} \end{aligned} \quad (\text{B.45})$$

$$\begin{aligned} \frac{\partial^2 u}{\partial x \partial z} \rightarrow & \frac{u_{i-2,j,k-2}^n - 4u_{i-2,j,k-1}^n + 3u_{i-2,j,k}^n - 4u_{i-1,j,k-2}^n + 16u_{i-1,j,k-1}^n - 12u_{i-1,j,k}^n}{4h^2} \\ & + \frac{3u_{i,j,k-2}^n - 12u_{i,j,k-1}^n + 9u_{i,j,k}^n}{4h^2} \end{aligned} \quad (\text{B.46})$$

Table B.12: Backward-Backward FDAs of Mixed Second Order Derivatives.

B.2 FDAs for Operators of the Form $\partial/\partial(r^p)$

In this section we briefly discuss a technique that is useful for deriving finite difference approximations that yield discrete solutions with good regularity behaviour in the vicinity of coordinate singularities. We note that the method was introduced to the numerical relativity community by Evans in his PhD dissertation [155].

The technique is best illustrated by way of example. We thus consider the radial part of the Laplacian operator in spherical polar coordinates, (r, θ, ϕ) . Assuming our unknown function, u , depends only on r , we then have

$$\nabla^2 u = \nabla^2 u(r) = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial u}{\partial r} \right). \quad (\text{B.47})$$

In order that the solution be regular (smooth) at the origin, $r = 0$, $u(r)$ must satisfy

$$\lim_{r \rightarrow 0} u(r) = u_0 + u_2 r^2 + O(r^4), \quad (\text{B.48})$$

where u_0 and u_2 are constants.

A key observation is that if we insert the above expansion into the right hand side of (B.47) we find

$$\begin{aligned} \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial u}{\partial r} \right) &= \frac{1}{r^2} \frac{\partial}{\partial r} \left[r^2 \frac{\partial}{\partial r} (u_0 + u_2 r^2 + O(r^4)) \right] \\ &= \frac{1}{r^2} \frac{\partial}{\partial r} (2u_2 r^3 + O(r^5)) = 6u_2 + O(r^2). \end{aligned} \quad (\text{B.49})$$

We now discretize the right hand side of (B.47) by first introducing a uniform radial grid, $r_j \equiv (j-1)h$, $j = 1, 2, \dots, n_r$, where h is the mesh spacing, and then writing

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial u}{\partial r} \right) \simeq \frac{1}{r_j^2} \frac{\left[r_{j+\frac{1}{2}}^2 (u_{j+1} - u_j) - r_{j-\frac{1}{2}}^2 (u_j - u_{j-1}) \right]}{(r_{j+1/2} - r_{j-1/2})^2}. \quad (\text{B.50})$$

Here $r_{j+1/2} \equiv r_j + h/2$ and $r_{j-1/2} \equiv r_j - h/2$. The approximation (B.50) is second order and seems natural enough, especially if one requires that the expression involve only the values u_{j-1} , u_j and u_{j+1} . However, we now consider evaluating (B.50) for $u(r)$ given by truncating (B.48) at $O(r^2)$, i.e. for $u(r) \equiv u_0 + u_2 r^2$, where, again, u_0 and u_2 are constants. Then we find after some

elementary algebra that

$$\frac{1}{r_j^2} \frac{\left[r_{j+\frac{1}{2}}^2 (u_{j+1} - u_j) - r_{j-\frac{1}{2}}^2 (u_j - u_{j-1}) \right]}{(r_{j+1/2} - r_{j-1/2})^2} = 6u_2 + \frac{1}{2} u_2 \frac{h^2}{r_j^2}. \quad (\text{B.51})$$

This is *not* the correct leading order behaviour given by (B.49). In particular, at the second grid point, $r_2 = h$, this evaluates to $(6\frac{1}{2})u_2$, instead of $6u_2$, so that irrespective of how small we make h , we will never capture the correct leading order behaviour of the differential operator in the vicinity of $r = 0$. Especially in dynamical evolutions, this can lead a lack of solution smoothness near the origin, and even to numerical instability.

However, with a simple change of variables, and an application of the chain rule, we can rewrite the radial Laplacian operator as follows

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial u}{\partial r} \right) = 3 \frac{\partial}{\partial(r^3)} \left(r^2 \frac{\partial u}{\partial r} \right) \quad (\text{B.52})$$

which can then be discretized as

$$3 \frac{\partial}{\partial(r^3)} \left(r^2 \frac{\partial u}{\partial r} \right) \simeq \frac{3}{\left(r_{j+\frac{1}{2}}^3 - r_{j-\frac{1}{2}}^3 \right)} \left[r_{j+\frac{1}{2}}^2 \frac{(u_{j+1} - u_j)}{(r_j - r_{j-1})} - r_{j-\frac{1}{2}}^2 \frac{(u_j - u_{j-1})}{(r_j - r_{j-1})} \right]. \quad (\text{B.53})$$

When this finite difference approximation is evaluated for $u(r) \equiv u_0 + u_2 r^2$ we find

$$\frac{3}{\left(r_{j+\frac{1}{2}}^3 - r_{j-\frac{1}{2}}^3 \right)} \left[r_{j+\frac{1}{2}}^2 \frac{(u_{j+1} - u_j)}{(r_j - r_{j-1})} - r_{j-\frac{1}{2}}^2 \frac{(u_j - u_{j-1})}{(r_j - r_{j-1})} \right] = 6u_2. \quad (\text{B.54})$$

Thus, the correct leading order behaviour as $r \rightarrow 0$ is mirrored in the discretization through this change-of-variables trick.

The generalization of this strategy is straightforward. When confronted with a differential expression of the general form

$$\frac{\partial f(r)}{\partial r}, \quad (\text{B.55})$$

where $f(r)$ satisfies the regularity condition

$$\lim_{r \rightarrow 0} f(r) = c_p r^p + c_{p+2} r^{p+2} + O(r^{p+4}), \quad (\text{B.56})$$

for some integer $p > 0$, we rewrite (B.55) as

$$p r^{p-1} \frac{\partial f}{\partial(r^p)} \tag{B.57}$$

prior to finite differencing.

Appendix C

PDEFDAOFF: A MAPLE Package for FDAS

This appendix documents the use of the package PDEFDAOFF (an acronym for Partial Differential Equation to Finite Difference Approximation using OFF-centred and centred approximation schemes). The package is a set of Maple⁴⁶ procedures we wrote to assist in the process of discretizing the PDEs governing our model using finite difference methods. The package contains two key Maple procedures, `dfdandoff` and `resndoff` which are discussed with illustrative examples in Secs. C.1 and C.2 below. We note that the examples were extracted directly from the demonstration worksheet that is included in the distribution [181].

C.1 FDAs of n -dimensional Differential Operators

`dfdandoff`⁴⁷ is a Maple procedure that computes a finite difference approximation to a differential operator, as applied to function of an arbitrary number of coordinates (x^1, x^2, \dots, x^n) . Given a user-specified order of accuracy, p , the routine returns a finite difference approximation which is $O(h^p)$ accurate in all coordinate directions. This procedure can calculate both centred and off-centred finite difference formula.

The routine has a header given by

```
dfdandoff := proc(u::function, x::list(name), ud::name, j::list(name),
                 h::list(name), q::list(integer), p::integer,
                 off::list(integer))
```

where the procedure arguments are defined as follows:

1. `u`: A generic Maple function of n variables.
2. `x`: Length- n list of independent variables (coordinates), in alphabetical order.

⁴⁶Maple is a registered trademark of Waterloo Maple Inc.

⁴⁷`dfdandoff` is an acronym for differential finite difference approximation for n -dimensional functions including off-centred schemes.

3. **ud**: Maple name for the finite difference unknown.
4. **j**: Length- n list of names of indices corresponding to each coordinate.
5. **h**: Length- n list of coordinate spacings in each direction.
6. **q**: Length- n list specifying the order of the differential operator with respect to each coordinate direction.
7. **p**: Approximation order for finite difference scheme
8. **off**: Length- n list of offsets (one of $-1, 0$ or 1) which defines the off-centring of the approximation in each coordinate direction.

C.1.1 1-dimensional Examples

We first define a Maple alias for $u(x)$ which suppresses the functional dependence on output, and allows us to omit the explicit dependence on input:

```
> alias(u=u(x));
```

u

Our initial examples deal with first derivatives and illustrate the importance of proper specification of the “off-centring” argument, **off**. Thus, if one naively attempts to find a first order centred scheme for du/dx , by using $[0]$ as the value of **off**, the procedure executes as follows:

```
> dfdandoff(u, [x], ud, [i], [hx], [1], 1, [0]);
Error, (in dfdandoff) invalid input: rhs received 1, which is not valid
for its 1st argument, expr
```

We do not consider this a bug since there is *no* $O(h)$ centred scheme for d/dx . Indeed, the simplest FDAs of d/dx are the first order backward and forward approximations. We can generate the backward formula using **off** := $[-1]$:

```
> dfdandoff(u, [x], ud, [i], [hx], [1], 1, [-1]);
```

$$\frac{\partial}{\partial x} u_i = -\frac{u_{i-1}}{hx} + \frac{u_i}{hx}$$

To verify the correctness of the finite difference operators that `dfdandoff` computes—which means that we ensure that that approximations *are* $O(h^p)$ accurate—we use another Maple procedure, `fdaeval`, due to Choptuik and which is included in `PDEFDAOFF`. `fdaeval` has the header

```
fdaeval := proc(c::array(integer), sigma::algebraic)
```

with arguments defined as follows:

1. `c`: A coefficient array such that `c[j]` is the relative weight of the unknown, u_{i+j} , in the finite difference approximation centred at x_i .
2. `sigma`: An overall scale factor to be applied to the formula.

The input to `fdaeval` thus defines an expression of the form

$$\sigma \sum_{j=j_{\min}}^{j_{\max}} c_j u_{i+j} \equiv \sigma \sum_{j=j_{\min}}^{j_{\max}} c_j u(x_i + jh) \quad (\text{C.1})$$

where h is the (constant) mesh spacing that is hard-coded in the procedure. `fdaeval` then replaces all of the $u_{i+j} \equiv u(x_i + jh)$ with Taylor series expansions (in h) about the value $u_i \equiv u(x_i)$. Provided that (C.1) *does* define a consistent approximation to some differential operator applied at $x = x_i$, `fdaeval` will return that operator (acting on u , which again is hard coded into the routine), as well as the leading order truncation error terms.

Thus, we can check the above invocation of `dfdandoff` as follows:

```
> abwh := array(-1..0, [-1,1]);
```

```
abwh := ARRAY(-1..0, [0 = 1, -1 = -1])
```

```
> fdaeval(abwh, 1/h);
```

$$D(u)(x) - \frac{1}{2}D^{(2)}(u)(x)h + \frac{1}{6}D^{(3)}(u)(x)h^2$$

This shows that, as expected, our backward formula *is* a valid first order finite difference approximation of du/dx .

Continuing, we can generate and check the $O(h^2)$ centred, forward and backward approximations of du/dx as follows:

```
> # Centred approximation
```

```
> dfdandoff(u, [x], ud, [i], [hx], [1], 2, [0]);
```

$$\frac{\partial}{\partial x} u_i = -\frac{1}{2} \frac{u_{i-1}}{hx} + \frac{1}{2} \frac{u_{i+1}}{hx}$$

```
> ach2 := array(-1..1, [-1,0,1]):
```

```
> fdaeval(ach2, 1/(2*h));
```

$$D(u)(x) + \frac{1}{6} D^{(3)}(u)(x) h^2$$

```
> # Backward approximation
```

```
> dfdandoff(u, [x], ud, [i], [hx], [1], 2, [-1]):
```

$$\frac{\partial}{\partial x} u_i = \frac{1}{2} \frac{u_{i-2}}{hx} - 2 \frac{u_{i-1}}{hx} + \frac{3}{2} \frac{u_i}{hx}$$

```
> abwh2 := array(-2..0, [1,-4,3]):
```

```
> fdaeval(abwh2, 1/(2*h));
```

$$D(u)(x) - \frac{1}{3} D^{(3)}(u)(x) h^2 + \frac{1}{4} D^{(4)}(u)(x) h^3$$

```
> # Forward approximation
```

```
> dfdandoff(u, [x], ud, [i], [hx], [1], 2, [1]):
```

$$\frac{\partial}{\partial x} u_i = -\frac{3}{2} \frac{u_i}{hx} + 2 \frac{u_{i+1}}{hx} - \frac{1}{2} \frac{u_{i+2}}{hx}$$

```
> afwh2 := array(0..2, [-3,4,-1]):
```

```
> fdaeval(afwh2, 1/(2*h));
```

$$D(u)(x) - \frac{1}{3} D^{(3)}(u)(x) h^2 - \frac{1}{4} D^{(4)}(u)(x) h^3$$

We conclude this section with examples illustrating the computation of $O(h^4)$ accurate approximations for the second derivative, d^2u/dx^2 .

```
> # Centred approximation
```

```
> dfdandoff(u, [x], ud, [i], [hx], [2], 4, [0]):
```

$$\frac{\partial^2}{\partial x^2} u_i = -\frac{1}{12} \frac{u_{i-2}}{hx^2} + \frac{4}{3} \frac{u_{i-1}}{hx^2} - \frac{5}{2} \frac{u_i}{hx^2} + \frac{4}{3} \frac{u_{i+1}}{hx^2} - \frac{1}{12} \frac{u_{i+2}}{hx^2}$$

```
> a2ch4 := array(-2..2, [-1,16,-30,16,-1]):
```

```
> fdaeval(a2ch4, 1/(12*h**2));
```

$$D^{(2)}(u)(x) - \frac{1}{90} D^{(6)}(u)(x) h^4$$


```

> # Forward approximation
> dfdandoff(u, [x], ud, [i], [hx], [2], 4, [2]);

```

$$\frac{\partial^2}{\partial x^2} u_i = \frac{15}{4} \frac{ud_i}{hx^2} - \frac{77}{6} \frac{ud_{i+1}}{hx^2} + \frac{107}{6} \frac{ud_{i+2}}{hx^2} - 13 \frac{ud_{i+3}}{hx^2} + \frac{61}{12} \frac{ud_{i+4}}{hx^2} - \frac{5}{6} \frac{ud_{i+5}}{hx^2}$$

```

> a2fw2h4 := array(0..5, [45,-154,214,-156,61,-10]);
> fdaeval(a2fw2h4, 1/(12*h**2));

```

$$D^{(2)}(u)(x) - \frac{137}{180} D^{(6)}(u)(x) h^4 - \frac{19}{12} D^{(7)}(u)(x) h^5$$

C.1.2 3-dimensional Example

The following example illustrates the use of `dfdandoff` for functions with dependence on three independent variables.

```

> alias(w=w(x,y,z));

```

$$w$$

```

> # Second order forward approximation of d/dx
> dfdandoff(w, [x,y,z], wd, [i,j,k], [hx,hy,hz], [0,0,1], 2, [0,0,1]);

```

$$\frac{\partial}{\partial z} w_{i,j,k} = -\frac{3}{2} \frac{wd_{i,j,k}}{hz} + 2 \frac{wd_{i,j,k+1}}{hz} - \frac{1}{2} \frac{wd_{i,j,k+2}}{hz}$$

C.2 Residual Evaluators for n -dimensional PDEs

`resndoff`⁴⁸ is a Maple procedure that automatically generates residuals evaluators associated with the finite-difference discretization of a general partial differential equation in n -dimensions. This routine was initially written with elliptic PDEs in mind, and in conjunction with the multigrid method, where the evaluation of residuals plays an important role when a relaxation technique is used as a smoother. However `resndoff` can also be employed in the context of time-dependent PDEs. In particular, it is very useful for generating independent residual evaluators for *all* types of PDEs, including time-dependent ones. As discussed in Chap. 4, independent residual evaluation is a very powerful tool for verifying the implementation of finite difference codes, especially when

⁴⁸`resndoff` is an acronym for **r**esidual evaluator for **n**-dimensional PDEs including **o**ff-centred finite difference schemes.

one can be confident that the independent residual evaluators are themselves error-free. Use of a routine such as `resndoff` helps ensure that this is the case.

`resndoff` has the following header:

```
resndoff := proc(eqn::equation, u::list(function), x::list(name),
                ud::list(name), j::list(name), h::list(name), p::integer,
                off::list(integer))
```

with procedure arguments defined as follows:

1. `eqn`: Partial differential equation for which independent residual is to be computed.
2. `u`: List of Maple names of the functions appearing in the PDE.
3. `x`: List of independent variables (coordinates).
4. `ud`: List of Maple names for the discrete representations of the functions (i.e. the grid function names).
5. `j`: Length- n list of names of indices corresponding to each coordinate.
6. `h`: Length- n list of coordinate spacings in each direction.
7. `p`: Approximation order for finite difference scheme.
8. `off`: Length- n list of offsets which defines the off-centring of the approximation in each coordinate direction.

Its use is illustrated here for the case of a nonlinear Poisson equation in 3 dimensions, where compactified Cartesian coordinates are adopted:

```
> # Define aliases for the functional dependence of u and f.
> alias(u=u(chi,eta,zeta), f=f(chi,eta,zeta));
```

$$u, f$$

```
> # Define the PDE.
> POI3DCP := (1-chi**2)*diff((1-chi**2)*diff(u,chi),chi)+
              (1-eta**2)*diff((1-eta**2)*diff(u,eta),eta)+
              (1-zeta**2)*diff((1-zeta**2)*diff(u,zeta),zeta)+sigma*u**2=f;
```

$$(1 - \chi^2) \left(-2\chi \frac{d}{d\chi} u + (1 - \chi^2) \frac{d^2}{d\chi^2} u \right) + (1 - \eta^2) \left(-2\eta \frac{d}{d\eta} u + (1 - \eta^2) \frac{d^2}{d\eta^2} u \right) \\ + (1 - \zeta^2) \left(-2\zeta \frac{d}{d\zeta} u + (1 - \zeta^2) \frac{d^2}{d\zeta^2} u \right) + \sigma u^2 = f$$

> # Generate the residual

```
> resndoff(POI3DCP, [u,f], [chi,eta,zeta], [ud,fd], [i,j,k], [hx,hy,hz],
2, [0,0,0]);
```

$$(1 - \chi_i^2) \left[-2\chi_i \left(-\frac{1}{2} \frac{ud_{i-1,j,k}}{hx} + \frac{1}{2} \frac{ud_{i+1,j,k}}{hx} \right) + (1 - \chi_i^2) \left(\frac{ud_{i-1,j,k}}{hx^2} - 2 \frac{ud_{i,j,k}}{hx^2} + \frac{ud_{i+1,j,k}}{hx^2} \right) \right] \\ + (1 - \eta_j^2) \left[-2\eta_j \left(-\frac{1}{2} \frac{ud_{i,j-1,k}}{hy} + \frac{1}{2} \frac{ud_{i,j+1,k}}{hy} \right) + (1 - \eta_j^2) \left(\frac{ud_{i,j-1,k}}{hy^2} - 2 \frac{ud_{i,j,k}}{hy^2} + \frac{ud_{i,j+1,k}}{hy^2} \right) \right] \\ + (1 - \zeta_k^2) \left[-2\zeta_k \left(-\frac{1}{2} \frac{ud_{i,j,k-1}}{hz} + \frac{1}{2} \frac{ud_{i,j,k+1}}{hz} \right) + (1 - \zeta_k^2) \left(\frac{ud_{i,j,k-1}}{hz^2} - 2 \frac{ud_{i,j,k}}{hz^2} + \frac{ud_{i,j,k+1}}{hz^2} \right) \right] \\ + \sigma ud_{i,j,k}^2 - fd_{i,j,k} = 0$$

Appendix D

Review of Relaxation Methods

In this appendix we provide a quick review of relaxation techniques as applied to large sparse systems of algebraic equations (both linear and nonlinear). Relaxation methods are used in two distinct capacities in the 3D code that we describe in the main body of the thesis:

1. In the case of the hyperbolic PDEs, point-wise Newton-Gauss-Seidel relaxation is used to *solve* the time-implicit algebraic equations that result from our use of an $O(h^2)$ Crank-Nicholson scheme.
2. For the elliptic PDEs, point-wise Newton-Gauss-Seidel is used as a *smoother* in the context of the multigrid solution of the discrete equations, which again arise from an $O(h^2)$ approximation of the PDEs.

The reader who is interested in full details concerning relaxation methods and related iterative techniques for the solution of large systems of equations should consult the classic reference by Varga [168].

D.1 The Linear Case: Jacobi and Gauss-Seidel Relaxation

We write a general linear system of equations in the form

$$\mathbf{L}\mathbf{u} = \mathbf{f}, \tag{D.1}$$

where \mathbf{L} is a $N \times N$ matrix, and \mathbf{u} and \mathbf{f} are N -component column vectors. In the typical case of interest, \mathbf{u} will enumerate all of the discrete unknowns, $u(t^n, x_i, y_j, z_k)$, associated with the finite difference approximation of some continuum function, $u(t, x, y, z)$, at some specific discrete time, $t = t^n$. The size of the system will then be given by $N = n_x \times n_y \times n_z$, where n_x , n_y and n_z are the number of grid points in the x , y and z directions, respectively. Due to the locality of finite difference operators, the matrix \mathbf{L} will generally be very sparse: that is, unless n_x , n_y

and n_z are very small, most of its entries will be 0, and the fraction of non-zeros will in fact tend to 0 as the mesh spacing h approaches 0. As is well known, even if one adopts a specially crafted ordering of the unknowns (see [164] for example), the solution of (D.1) using standard methods of numerical linear algebra—that is, using some form of Gaussian elimination such as LU decomposition [160]—is extremely inefficient. This is already the case for discretizations of problems in 2 spatial dimensions, and is even more so for 3D calculations. For example, consider the situation where the 3D Poisson equation in (x, y, z) coordinates has been discretized on a uniform mesh with $n = n_x = n_y = n_z$ grid points per edge of the computational domain. Then the size of the system is $N = n^3$. We assume that we have used the standard $O(h^2)$ 7-point approximation for the discrete Laplacian, which couples a particular unknown to its nearest neighbours in each of the three coordinate directions, and that we have numbered the unknowns in so called lexicographic order ($((u_{ijk}, i = 1, \dots, n), j = 1, \dots, n), k = 1, \dots, n)$). Then the bandwidth of \mathbf{L} is $O(n^2) = O(N^{2/3})$, and the cost for solution via Gaussian elimination will be $O(N^{7/3})$ in computational time and up to $O(N^{5/3})$ in memory. In practice this precludes direct solution of (D.1) for any but the smallest 3D problems.

The class of *iterative* techniques known as relaxation methods were developed decades ago largely in response to the need to efficiently solve the large systems of algebraic equations arising from the finite-difference discretization of PDEs, particularly those of elliptic type. Relaxation methods directly exploit the sparsity structure of these systems, and have the further advantage of having extremely economical memory requirements, since the system of equations is not explicitly stored.

We thus start with the general notion of an iterative, or *fixed point* technique for the solution of (D.1). We note that the system can be rewritten as

$$\mathbf{u} = \mathbf{T}\mathbf{u} + \mathbf{c}, \quad (\text{D.2})$$

where \mathbf{T} is the $N \times N$ *update matrix* and \mathbf{c} is a constant vector that depends on the specific iterative method used. The idea is to choose \mathbf{T} and \mathbf{c} such that starting from some initial estimate, $\mathbf{u}^{(0)}$, of \mathbf{u} , the iteration

$$\mathbf{u}^{(k+1)} = \mathbf{T}\mathbf{u}^{(k)} + \mathbf{c}, \quad (\text{D.3})$$

has a fixed point \mathbf{u} which satisfies (D.1). That is, we want

$$\lim_{k \rightarrow \infty} \mathbf{u}^{(k)} = \mathbf{u}. \quad (\text{D.4})$$

Whether or not the iteration (D.3) converges is determined by the eigenvalues of the update matrix \mathbf{T} . More precisely, the necessary and sufficient condition for convergence is that the *spectral radius*, $\rho(\mathbf{T})$, of \mathbf{T} satisfy $\rho(\mathbf{T}) < 1$. We recall that the spectral radius may be defined as

$$\rho(T) = \max_{1 \leq i \leq N} |\lambda_i|, \quad (\text{D.5})$$

where λ_i are the eigenvalues of \mathbf{T} . For strictly diagonally dominant matrices \mathbf{T} , which satisfy

$$|T_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^N |T_{ij}|, \quad (\text{D.6})$$

two iterative schemes become particularly interesting since they are known to converge for any initial estimate $\mathbf{u}^{(0)}$. This observation is pertinent to the case of equations resulting from FDA approximation since such systems often satisfy some form of diagonal dominance, which although frequently weaker than (D.6), is still enough to ensure convergence of the methods we will outline.

Although the formal analysis of relaxation methods (again, see [168]) typically proceeds via a matrix approach which involves a decomposition, or splitting, of \mathbf{L} , for our purposes it is more intuitive to describe the techniques as they are typically *implemented* as computer code. We thus write (D.1) in the form

$$\sum_{j=1}^N L_{ij} u_j = f_i, \quad i = 1, 2, \dots, N, \quad (\text{D.7})$$

which amounts to explicitly writing out the action of each row of \mathbf{L} on \mathbf{u} , and equating it to the corresponding element of \mathbf{f} . We then view any of the N subsystems of equations defined by (D.7) as an equation for the *single* unknown, u_i . Solving for u_i we get

$$u_i = \frac{1}{L_{ii}} \left[- \sum_{\substack{j=1 \\ j \neq i}}^N L_{ij} u_j + f_i \right], \quad (\text{D.8})$$

and the sequence of approximations starting from the initial guess $\mathbf{u}^{(0)}$ can be generated component

by component using

$$u_i^{(k)} = \frac{1}{L_{ii}} \left[- \sum_{\substack{j=1 \\ j \neq i}}^N L_{ij} u_j^{(k-1)} + f_i \right], \quad i = 1, 2, \dots, N. \quad (\text{D.9})$$

The method defined by (D.9) is known as *Jacobi iteration*. We note that it requires that we store values for both the current approximation, $\mathbf{u}^{(k)}$, as well as the previous estimate, $\mathbf{u}^{(k-1)}$. Crucially, however, for systems derived from FDAs there is no need to explicitly store the elements of \mathbf{L} . Rather, the computation of the right hand side of (D.9) simply amounts to the evaluation of the finite difference formula at a given grid point, so that the components L_{ij} are effectively “hard coded” into the computer program. Thus, the memory requirements for the technique—and for all relaxation methods—are the optimal $O(N)$. We also note at this point that the complete process of iterating the solution estimate from $\mathbf{u}^{(k)}$ to $\mathbf{u}^{(k+1)}$ is frequently called a *relaxation sweep*, in appeal to the notion that one “sweeps” through the finite difference grid, updating one discrete unknown at a time.

The other relaxation technique we wish to consider can be viewed as a simple modification of the Jacobi iteration that aims to speed convergence by using the most recently computed components of the solution estimate when they are available. So, assuming that we are updating the unknowns $u_i^{(k)}$ in the order $i = 1, 2, \dots, N$, then for $j < i$ we will have already computed $u_j^{(k)}$, while for $j > i$ we will need to use values $u_j^{(k-1)}$. We thus modify (D.9) as follows:

$$u_i^{(k)} = \frac{1}{L_{ii}} \left[- \sum_{j=1}^{i-1} L_{ij} u_j^{(k)} - \sum_{j=i+1}^N L_{ij} u_j^{(k-1)} + f_i \right], \quad i = 1, 2, \dots, N. \quad (\text{D.10})$$

This defines the *Gauss-Seidel* iteration for the solution of (D.1). In practice, Gauss-Seidel relaxation *does* tend to require fewer sweeps than Jacobi to achieve convergence to a specified tolerance.⁴⁹ Additionally, Gauss-Seidel iteration only needs storage for a single instance of the unknown vector, which is half of that required by the Jacobi method.

We should also emphasize at this juncture that the methods given by (D.9) and (D.10) can be further qualified as defining *point-wise* Jacobi and *point-wise* Gauss-Seidel relaxations, respectively. This distinguishes the techniques from extensions of the methods in which groups of unknowns—

⁴⁹It should be noted, however, that both Gauss-Seidel and Jacobi generally have the same asymptotic convergence rate [168].

$u(x_i, y_j, z_k)$, $i = 1, \dots, N$, for fixed j and k , for example—are updated simultaneously. These generalizations are known as *line* relaxation methods, and although we have mentioned them briefly in Sec. 2.4.5, we did not use them in any of the work described in this thesis.

In preparation for our discussion of nonlinear systems in the next section it is useful to describe the point-wise Gauss-Seidel iteration (D.10) in terms of the *residual vector* associated with the approximate solution. First, given the approximation $\mathbf{u}^{(k)}$ that is defined after the completion of the k -th relaxation sweep, we define the residual vector $\mathbf{r}^{(k)}$ by

$$\mathbf{r}^{(k)} \equiv \mathbf{L}\mathbf{u}^{(k)} - \mathbf{f}, \quad (\text{D.11})$$

so that, assuming that the Gauss-Seidel iteration converges, we have

$$\lim_{k \rightarrow \infty} \mathbf{r}^{(k)} = \mathbf{0}, \quad (\text{D.12})$$

where $\mathbf{0}$ is the N -component 0-vector. We again emphasize that we can only compute \mathbf{r}^k *after* the k -th pass through the solution unknowns has been completed. We therefore further introduce the concept of a *running residual*, generically denoted $\tilde{\mathbf{r}}$, which has components that are continually changing as individual unknowns are updated. We thus define the (running) unknown vector $\tilde{\mathbf{u}}_i^{(k)}$, which at any iteration k , has the following “instantaneous” values when we have just updated unknown $i - 1$ and are about to update unknown i :

$$\tilde{\mathbf{u}}_i^{(k)} = \left[u_1^{(k)}, u_2^{(k)}, \dots, u_{i-1}^{(k)}, u_i^{(k-1)}, \dots, u_N^{(k-1)} \right]^T. \quad (\text{D.13})$$

With this definition, the running residual, $\tilde{\mathbf{r}}_i^{(k)}$, is given by

$$\tilde{\mathbf{r}}_i^{(k)} \equiv \mathbf{L}\tilde{\mathbf{u}}_i^{(k)} - \mathbf{f}, \quad (\text{D.14})$$

where the subscript i reminds us that $\tilde{\mathbf{r}}_i^{(k)}$ has components that are constantly changing as we sweep through the grid, updating unknowns one by one. Using (D.10) and (D.14), we can explicitly write the components, $[\tilde{\mathbf{r}}_i^{(k)}]_m$, of the running residual vector as

$$[\tilde{\mathbf{r}}_i^{(k)}]_m = \sum_{j=1}^{i-1} L_{mj} u_j^{(k)} + \sum_{j=i+1}^N L_{mj} u_j^{(k-1)} + L_{mi} u_i^{(k-1)} - f_m. \quad (\text{D.15})$$

For the particular case of the i -th residual this last expression gives us

$$[\tilde{\mathbf{r}}_i^{(k)}]_i = \sum_{j=1}^{i-1} L_{ij} u_j^{(k)} + \sum_{j=i+1}^N L_{ij} u_j^{(k-1)} + L_{ii} u_i^{(k-1)} - f_i, \quad (\text{D.16})$$

which upon insertion into (D.10) yields a more compact form of the point-wise Gauss-Seidel iteration:

$$u_i^{(k)} = u_i^{(k-1)} - \frac{[\tilde{\mathbf{r}}_i^{(k)}]_i}{L_{ii}}. \quad (\text{D.17})$$

This expression will be generalized to the case of nonlinear relaxation in the next section.

D.2 The Nonlinear Case: Newton-Gauss-Seidel Relaxation

In situations such as ours, where the governing PDEs are nonlinear, finite difference approximation will naturally lead to large sparse, systems of *nonlinear* algebraic equations. Newton's method [160] then provides a route to extend the techniques described above so that they can be used to solve nonlinear systems.

We begin then by reviewing Newton's method for the solution of single nonlinear equation in a single unknown, u . We write the equation in the canonical form

$$f(u) = 0, \quad (\text{D.18})$$

and note that the technique is itself iterative. Thus we must start with some initial estimate, $u^{[0]}$, of the solution, and then generate iterates, $u^{[n]}$, such that, assuming that the method converges (which generally is dependent on the quality of the initial guess), we have $\lim_{n \rightarrow \infty} u^{[n]} = u$. The Newton iteration for (D.18) is then given by

$$u^{[n+1]} = u^{[n]} - \frac{f(u^{[n]})}{f'(u^{[n]})}, \quad (\text{D.19})$$

where $f'(u) \equiv df(u)/du$. Defining the residual, $r^{[n]}$, associated with the iterate, $u^{[n]}$, as

$$r^{[n]} \equiv f(u^{[n]}), \quad (\text{D.20})$$

we can rewrite (D.19) as

$$u^{[n+1]} = u^{[n]} - \frac{r^{[n]}}{f'(u^{[n]})}. \quad (\text{D.21})$$

We now assume that the system of N equations that result from the finite difference discretization of our PDEs has been written in the canonical form

$$F_i[\mathbf{u}] = 0, \quad i = 1, 2, \dots, N, \quad (\text{D.22})$$

where each of the F_i , $i = 1, \dots, N$ is a nonlinear function of the unknowns, $\mathbf{u} = [u_1, u_2, \dots, u_N]^T$ (but only a few of them in general, again due to the locality of finite difference operators).

The application of relaxation to nonlinear equations again involves relaxation sweeps through the unknowns, so focusing on the extension of the Gauss-Seidel method, and in analogy to what we did in the previous section, we define the components, $[\tilde{\mathbf{r}}_i^{(k)}]_m$, of the running residual vector by

$$[\tilde{\mathbf{r}}_i^{(k)}]_m \equiv F_m[\tilde{\mathbf{u}}_i^{(k)}]. \quad (\text{D.23})$$

Here, the running solution vector, $\tilde{\mathbf{u}}_i^{(k)}$, is defined by (D.13) as previously, and the subscript i again emphasizes that the elements of the running vectors are constantly changing as we sweep through the grid, updating each unknown in turn.

With this definition, and using (D.21), the (one-step) point-wise Newton-Gauss-Seidel relaxation method is defined by

$$u_i^{(k)} = u_i^{(k-1)} - \frac{[\tilde{\mathbf{r}}_i^{(k)}]_i}{J_{ii}}, \quad (\text{D.24})$$

where the “diagonal Jacobian element”, J_{ii} is given by

$$J_{ii} = \left. \frac{\partial F_i[\mathbf{u}]}{\partial u_i} \right|_{u_i = u_i^{(k-1)}} \quad (\text{D.25})$$

The additional nomenclature “one step” indicates that we do not use Newton’s method to fully *solve* any of the individual nonlinear equations as we visit the unknowns; instead, we apply only *one* iteration of the Newton technique before moving on to the next unknown. This strategy is largely motivated by the observation that a complete solution of any individual equation will generally represent some amount of wasted computational work, since as soon as one of the neighbouring unknowns is modified by the subsequent relaxation process, the equation will no longer be satisfied.