Copyright
by
Greg Andrew McIvor
2012

The Dissertation Committee for Greg Andrew McIvor
certifies that this is the approved version of the following dissertation:

Computing Binary Black Hole Merger Waveforms Using *OpenGR*

Committee:

Richard Matzner, Supervisor

Karl Gebhardt

Pawan Kumar

Michael Marder

Philip Morrison

William Press

Computing Binary Black Hole Merger Waveforms Using *OpenGR*

by

Greg Andrew McIvor, B.S.

DISSERTATION
Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
May 2012

Acknowledgments

openGR has been the work of a team of researchers whose efforts I would like to acknowledge. Matt Anderson started the project about a decade ago, and was followed by Paul Walter, Jon Allen, and Andrea Nerozzi. I would particularly like to thank Paul Walter, as his support and insight have has been invaluable during this work. I also want to acknowledge my advisor, Richard Matzner, who has overseen not only my work but the work of those before me and has provided guidance and direction throughout the entire process.

openGR is based on many external libraries, including *SAMRAI*, *PVODE*, and *KINSol*, all developed at the Lawrence Livermore National Laboratory (LLNL). Visualization is done using VisIt, also developed at LLNL. *openGR* would not exist as it does today without the efforts of many people at LLNL who have invested their time developing those tools.

All of the simulations presented in this work were performed on the Ranger supercomputer at the Texas Advanced Computing Center (TACC, `http://www.tacc.utexas.edu`) at the University of Texas at Austin using Teragrid account TGPHY090030. I want to acknowledge the staff of the TACC who maintain all of the high performance computing and visualization resources including Ranger and Spur.

I would like to thank my parents for their support throughout my entire

graduate school career. They have encouraged and supported me throughout this long process.

Finally, I want to thank my girlfriend, Courtney, who selflessly served as the editor of this dissertation. I know she has been anticipating the completion of this work almost as much as I have. Although this has been an incredibly long and difficult process, with her motivation and support I was finally able to complete this work so we can both move on to the next stage of our life together.

Computing Binary Black Hole Merger Waveforms Using *OpenGR*

Publication No. ____________________

Greg Andrew McIvor, Ph.D.
The University of Texas at Austin, 2012

Supervisor: Richard Matzner

One of the most important predictions of General Relativity, Einstein's theory of gravity, is the existence of gravitational radiation. The strongest source of such radiation is expected to come from the merging of black holes. Upgrades to large ground based interferometric detectors (LIGO, VIRGO, GEO 600) have increased their sensitivity to the point that the first direct observation of a gravitational wave is expected to occur within the next few years. The chance of detection is greatly improved by the use of simulated waveforms which can be used as templates for signal processing. Recent advances in numerical relativity have allowed for long stable evolution of black hole mergers and the generation of expected waveforms.

openGR is a modular, open framework black hole evolution code developed at The University of Texas at Austin Center for Relativity. Based on the

BSSN (strongly hyperbolic) formulation of Einstein's equations and the moving puncture method, we are able to model the evolution of a binary black hole system through the merger and extract the gravitational radiation produced. Although we are generally interested in binary interactions, *openGR* is capable of handling any number of black holes. This work serves as an overview of the capabilities of *openGR* and a demonstration of the physics it can be used to explore.

Table of Contents

Acknowledgments **iv**

Abstract **vi**

List of Tables **xi**

List of Figures **xii**

Chapter 1. Introduction **1**

1.1 Conventions 4

Chapter 2. 3+1 Numerical Relativity **5**

2.1 General Relativity 5

2.2 ADM Formalism 6

2.2.1 Evolution Equations 10

2.2.2 Contraint Equations 10

2.2.3 The Conformal Transverse-Traceless Decomposition . . 11

Chapter 3. Puncture Method **13**

3.1 Initial Data 14

3.2 BSSN Formulation 15

3.3 Gauge Conditions 17

Chapter 4. Gravitational Wave Extraction **19**

4.1 Tetrad Formalism 20

4.2 The Newman Penrose Formalism 22

4.2.1 Null Tetrads 22

4.2.2 Weyl Tensor and Weyl Scalars 24

4.3 Tetrad Transformations 25

4.3.1 Type I Rotations 25
4.3.2 Type II Rotations 26
4.3.3 Type III Rotations 27
4.4 Null Tetrads and Null Frames 28
4.5 Curvature Invariants 28
4.6 Principal Null Directions 29
4.7 Petrov Classification 32
4.7.1 Petrov Type I 33
4.7.2 Petrov Type D 33
4.8 Quasi-Kinnersley Frame 34
4.9 Finding the Quasi-Kinnersley Frame 37
4.10 Ψ_4 in Terms of Scalar Quantities 39
4.11 Decomposition of Ψ_4 by Spherical Harmonics 40
4.12 Energy, Momentum, and Angular Momentum 41

Chapter 5. Gravitational Wave Detection **44**
5.1 LIGO 44
5.2 Matched Filtering 47
5.3 NINJA 49

Chapter 6. Computational Framework **52**
6.1 The *SAMRAI* Library 52
6.2 *PVODE* 54
6.3 *KINSol* 58
6.4 Restriction and Prolongation 60
6.5 Outer Boundary Conditions 62
6.5.1 Sommerfield Boundary Conditions 62
6.5.2 Robin Boundary Conditions 62
6.6 Adaptive Mesh Refinement 63

Chapter 7. Scaling, Performance, and Convergence 65
7.1 Scaling . . . 65
7.1.1 Strong Scaling . . . 67
7.1.2 Single Puncture Unigrid Strong Scaling . . . 68
7.1.3 Single Puncture FMR Strong Scaling . . . 69
7.1.4 Weak Scaling . . . 70
7.1.5 Single Puncture Unigrid Weak Scaling . . . 71
7.1.6 Single Puncture FMR Weak Scaling . . . 72
7.2 Memory Use . . . 74
7.2.1 Ghostzones . . . 76
7.3 Performance . . . 78
7.4 Convergence . . . 80

Chapter 8. Results 83
8.1 QC0 . . . 83
8.2 Equal Mass Head-On Collision from Rest . . . 91
8.3 QC0 with Spin . . . 93
8.4 Head-On Collision with Spin . . . 96
8.5 Scattering Interaction . . . 98

Chapter 9. Conclusion 103

Vita 112

List of Tables

4.1 Table of Petrov Classifications. The Petrov type is determined by the number of coinciding principal null directions. 32

7.1 Configurations used for scaling runs. 66
7.2 Summary of strong scaling results for a unigrid single puncture. 68
7.3 Summary of strong scaling results for single puncture with nine levels of FMR. 70
7.4 Summary of weak scaling results for a unigrid single puncture. 72
7.5 Summary of weak scaling results for single puncture with nine levels of FMR. 74
7.6 Details of successful and unsuccessful runs for single puncture unigrid Jobs C and D. 75
7.7 Details of successful and unsuccessful runs for single puncture FMR Jobs C and D. 76
7.8 Details of the ghostzones used in Job D for a single puncture in a unigrid domain. 77
7.9 Rate of evolution of optimal configurations for Jobs A, B, and C. 79
7.10 Configurations used for convergence tests. In all cases the hole is located at the origin and has mass M. 80

8.1 Summary of parameters used for QC0 simulation. 84
8.2 Summary of parameters used for Equal Mass Head-On Collision from Rest. 91
8.3 Summary of parameters used for QC0 simulation with spin added. 93
8.4 Summary of parameters used for an eccentric merger and a scattering interaction. All parameters are identical except for the initial momentum in the x direction. 98

List of Figures

2.1 The definition of the lapse α and shift β^i shown in terms of their relation to two adjacent hypersurfaces. [9] 8

5.1 Schematic view of a LIGO detector [29]. 45
5.2 LIGO strain sensitivity for science runs S1 - S5 [30]. 47

6.1 Example of a grid with mesh refinement. 53
6.2 Zoomed in view of the grid in the region of the holes. 64

7.1 Strong scaling for a single puncture in a unigrid domain. . . . 69
7.2 Strong scaling for a single puncture with nine levels of FMR. . 70
7.3 Weak scaling for a single puncture in a unigrid domain. 72
7.4 Weak scaling for a single puncture with nine levels of FMR. . 73
7.5 Results of a convergence test at time $t = 30M$. 82

8.1 Full simulation of QC0. Snapshots of the lapse α. 85
8.2 Trajectory of the holes in the $x - y$ plane for QC0. 86
8.3 Full simulation of QC0. Snapshots of the real part of $r\Psi_4$. . . 87
8.4 The real part of Ψ_4 $(l = 2, m = 2)$ scaled by the extraction radius shown for various extraction radii. 88
8.5 Comparison of coarse and fine resolution for QC0 wave extraction. 90
8.6 The product $rRe[\Psi_4](2,2)$ shown for two holes undergoing infall from rest. 92
8.7 Trajectory of the holes in the $x - y$ plane for QC0 with spins $a = \pm 0.6$. 94
8.8 The product $rRe[\Psi_4(2,2)]$ extracted at $r = 40M$ shown for QC0 and QC0 with spin $a = \pm 0.6$. 95
8.9 Comparison of the x position vs. time for two holes with and without spin starting from rest at $x = \pm 1.5M$. 96
8.10 The real part of $r\Psi_4$ shown for a head-on collision from rest with and without spin. 97

8.11 Trajectories of the holes in the $x - y$ plane for an eccentric merger and a scattering interaction. 99

8.12 The real part of $r\Psi_4$ shown for an eccentric merger and a scattering interaction. 101

Chapter 1

Introduction

One of the most important predictions of general relativity, Einstein's theory of gravity, is the existence of gravitational radiation. In the linearized theory of general relativity, this radiation exists as a traveling wave solution to the Einstein field equations and is represented as a small perturbation $h_{\mu\nu}$ to a flat background metric $\eta_{\mu\nu}$. These perturbations are generated by a changing quadrupole moment I:

$$h_{\mu\nu} = \frac{2G}{c^4 r}\ddot{I}_{\mu\nu}. \tag{1.1}$$

In the case of two compact objects in circular orbit with orbital frequency f_{orb}, the gravitational waves produced have a strain h of

$$h \approx \frac{4\pi^2 G M R^2 f_{orb}^2}{c^4 r} \tag{1.2}$$

where M is the reduced mass of the system, R is the separation of the two objects, and r is the distance from the source to the detector. Since black holes have very small radii compared to other objects of the same mass, they are able to orbit much closer together than other objects, resulting in very large orbital frequencies (f_{orb}). As such, binary black hole mergers are expected to be the strongest sources of gravitational radiation. In other words, black hole mergers produce gravitational waves with a comparatively large strain h relative to waves produced by other objects.

Upgrades to large ground based interferometric detectors (LIGO [1], VIRGO [2], GEO 600 [3]) have increased their sensitivity to the point that the first direct observation of a gravitational wave is expected to occur within the next few years. To aid in signal processing, the signals from these detectors are analyzed using the process of matched filtering. Matched filtering uses simulated waveforms as templates, which greatly enhances the signal to noise ratio and thus improves the chance of detection. Use of these simulated waveforms also allow for improved parameter estimation such as mass ratio and spin. These waveforms are generated using numerical relativity simulations.

Significant advances in numerical relativity came in 2005 when Frans Pretorius was the first to successfully simulate a full orbit of two black holes and extract the gravitational radiation produced using generalized harmonic coordinates [4]. Shortly thereafter, two groups (University of Texas at Brownsville and NASA Goddard Space Flight Center) both had similar success using an approach based on the moving puncture method and the BSSN formulation of general relativity [5, 6]. Since then, several other groups have developed their own black hole simulation codes, including the University of Texas at Austin's development of *openGR*.

openGR is a framework used for large numerical black hole simulations. It has been developed at The University of Texas at Austin Center for Relativity over the past decade, beginning with Matt Anderson's work [7]. Based on the BSSN (strongly hyperbolic) formulation of Einstein's equations and the moving puncture method, *openGR* can be used to model the evolution of a binary black hole system through the merger and extract the gravitational radiation produced. One of the main differences between *openGR* and other numerical relativity codes is that *openGR* is based on the *SAMRAI* (Structure Adaptive Mesh Refinement Application Infrastructure) library [8] while most other codes are based on *Cactus Code*. The finite-differencing scheme used in *openGR* is accurate to fourth order (second order at the boundaries). Although the work discussed in this paper focuses on binary black hole mergers and the resulting gravitational radiation, *openGR* is capable of evolving any general relativistic spacetime. As the name suggests, *openGR* is an open source code available for download at `http://wwwrel.ph.utexas.edu/openGR`.

openGR has finally been developed to a point that all of its capabilities are functional and it can now be used for scientific research. This paper begins with an overview of the theoretical background upon which *openGR* is based, starting with Einstein's general theory of relativity (Chapter 2). Chapter 3 discusses the puncture method, which *openGR* uses to stably evolve a black hole. Chapter 4 explores the wave extraction and demonstrates how information about gravitational radiation is extracted from the spacetime. Gravitational wave detection is discussed in Chapter 5, including the role numerical relativity plays in detection efforts. Chapter 6 focuses on numerical methods and details the computational framework upon which *openGR* is built. The discussion of scaling in Chapter 7 provides an overview of how the code performs as simulations grow larger. Memory use, performance, and convergence are also detailed in this chapter. Finally, Chapter 8 presents a series of results that demonstrate *openGR*'s capabilities and provide examples of the types of problems it can be used to study.

1.1 Conventions

In this paper the following conventions are used (unless otherwise specified):

- Metric signature: $(-,+,+,+)$
- Greek indices $(\mu, \nu, \dots)$ span $0, 1, 2, 3$.
- Latin indices $(i, j, k, \dots)$ span $1, 2, 3$.
- Geometric units: $G = c = 1$
- Distances and times are given in terms of the ADM mass M:
  - Distance: $d = M \Rightarrow d = \frac{GM}{c^2}$
  - Time: $t = M \Rightarrow t = \frac{GM}{c^3}$
- Covariant derivative operator: ∇
- Vectors indicated with an arrow: $\vec{v}$
- Determinant of the 3-metric g_{ij}: g
- Determinant of the 4-metric $g_{\mu\nu}$: $\mathbf{g}$

Chapter 2

3+1 Numerical Relativity

2.1 General Relativity

The fundamental equations in general relativity are the the Einstein field equations, given by

$$G_{\mu\nu} = 8\pi T_{\mu\nu}. \tag{2.1}$$

$G_{\mu\nu}$ is the Einstein tensor and $T_{\mu\nu}$ is the stress-energy Tensor. The Einstein tensor can be expanded as

$$G_{\mu\nu} \equiv R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R. \tag{2.2}$$

The Ricci curvature tensor $R_{\mu\nu}$ is a contraction of the Riemann tensor $R^{\alpha}_{\beta\gamma\delta}$ on the first and third indices:

$$R_{\mu\nu} \equiv R^{\alpha}_{\mu\alpha\nu} = \Gamma^{\alpha}_{\mu\nu,\alpha} - \Gamma^{\alpha}_{\mu\alpha,\nu} + \Gamma^{\alpha}_{\beta\alpha}\Gamma^{\beta}_{\mu\nu} - \Gamma^{\alpha}_{\beta\nu}\Gamma^{\beta}_{\mu\alpha} \tag{2.3}$$

where

$$\Gamma^{\alpha}_{\beta\gamma} = \frac{1}{2} g^{\alpha\delta}(g_{\beta\delta,\gamma} + g_{\gamma\delta,\beta} - g_{\beta\gamma,\delta}) \tag{2.4}$$

is the Christoffel symbol of the second kind.

It can be shown that a certain combination of covariant derivatives of the Riemann tensor vanish:

$$R^{\alpha}_{\beta\mu\nu;\lambda} + R^{\alpha}_{\beta\lambda\mu;\nu} + R^{\alpha}_{\beta\nu\lambda;\mu} = 0. \tag{2.5}$$

These are known as the Bianchi identities. One important consequence comes from contracting them twice, which reveals the fact that the Einstein tensor has vanishing divergence

$$G^{\mu}_{\nu;\mu} = 0 \tag{2.6}$$

If we consider the vacuum case, $T_{\mu\nu} = 0$ and the Einstein equations (Eq. (2.1)) become

$$G_{\mu\nu} = 0. \tag{2.7}$$

The field equations can be decomposed into four constraint equations

$$G_{0\mu} = 0 \tag{2.8}$$

and six evolution equations

$$G_{ij} = 0. \tag{2.9}$$

It is worth noting that the constraint equations remain satisfied under the action of the evolution equations.

In numerical relativity we want to look at the dynamics of the gravitational field as it changes in time. This can be achieved by decomposing the field equations into spatial and temporal parts, typically referred to as a 3+1 split. The ADM formalism provides a method by which to achieve this split.

2.2 ADM Formalism

The Einstein field equations discussed in the previous section are presented in a covariant form, which makes no distinction between space and time. While this is important from a theoretical perspective, it does not allow us to investigate the evolution of the gravitational field in time. We want to formulate this evolution as an initial value or Cauchy problem. That is, given initial and boundary conditions we can then use the fundamental equations to evolve the system forward in time. To do this we need to re-formulate Einstein's equations with a clear separation between time and space. This is known as the 3+1 formalism of general relativity.

Consider a spacetime with metric $g_{\alpha\beta}$ that is globally hyperbolic, meaning it has a Cauchy surface. As such, it can be completely foliated into three dimensional hypersurfaces in such a way that each of these hypersurfaces is spacelike. Consider two adjacent hypersufaces Σ_t and Σ_{t+dt}. The geometry of the spacetime between these hypersurfaces can be determined using three pieces:

1. The 3-metric g_{ij} that measures proper distance within the hypersurface:

$$dl^2 = g_{ij} dx^i dx^j. \tag{2.10}$$

2. The lapse α that relates the proper time $d\tau$ between the hypersufaces as measured by observers moving in a direction normal to the hypersurfaces (Eulerian observers) to the coordinate time dt:

$$d\tau = \alpha dt. \tag{2.11}$$

3. The shift vector β^i that gives the relative velocity between the Eulerian

observers and the lines of constant spatial coordinates:

$$x^i_{t+dt} = x^i_t - \beta^i dt. \tag{2.12}$$

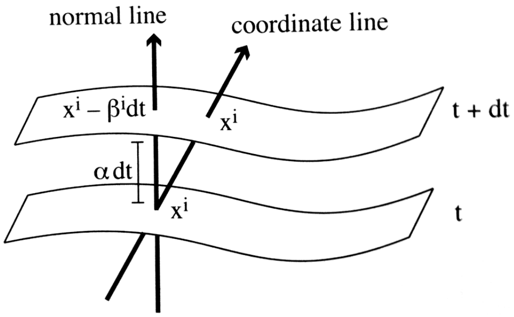


Figure 2.1: The definition of the lapse α and shift β^i shown in terms of their relation to two adjacent hypersurfaces. [9]

The lapse α and the shift β^i are gauge functions, meaning they are freely specifiable and depend on the choice of coordinates chosen. The metric in terms of these quantities is:

$$ds^2 = (-\alpha^2 + \beta_i\beta^i)dt^2 + 2\beta_i dt dx^i + g_{ij}dx^i dx^j. \tag{2.13}$$

This gives:

$$g_{\mu\nu} = \begin{pmatrix} -\alpha^2 + \beta_k\beta^k & \beta_i \\ \beta_j & g_{ij} \end{pmatrix}, \tag{2.14}$$

$$g^{\mu\nu} = \frac{1}{\alpha^2}\begin{pmatrix} -1 & \beta^i \\ \beta^j & g^{ij} - \beta^i\beta_j \end{pmatrix} \tag{2.15}$$

The unit normal vector to the spacial hypersufaces, n^μ, is given by

$$n^\mu = (\frac{1}{\alpha}, \frac{-\beta^i}{\alpha}), \tag{2.16}$$

$$n_\mu = (\alpha, 0). \tag{2.17}$$

By definition, the unit normal vector corresponds to the 4-velocity of the Eulerian observers. This can be expressed in terms of the lapse α and coordinate time t as

$$n^\mu = -\alpha \nabla^\mu t \tag{2.18}$$

where the minus sign ensures that $\vec{n}$ points forward in time.

From the definiton of the shift (Eq. (2.12)) we find that

$$\beta^i = -\alpha(\vec{n} \cdot \nabla x^i) \tag{2.19}$$

As defined, the β^i are scalars and can be used to define a 4-vector $\vec{\beta}$ whose componets are $(0, \beta^i)$. This 4-vector is orthogonal to $\vec{n}$ and we can then use the vectors $\vec{n}$ and $\vec{\beta}$ to construct a time vector $\vec{t}$ defined as

$$t = \alpha n^\mu + \beta^\mu. \tag{2.20}$$

When considering the curvature of the hypersurfaces created by the foliation of spacetime, we need to make a distinction between the intrisic curvature of the hyperfurfaces, which comes from their internal geometry, and the extrinsic curvature which is associated with the way in which these surfaces are immersed in a four-dimensional spacetime. The intrinsic curvature can be calculated using the three-dimensional Riemann tensor defined in terms of the 3-metric g_{ij}. The extrinsic curvature gives a measure of the change in the unit normal vector n^μ as it is parallel-transported from one point on the hypersurface to another. This is a much more useful quantity for our purposes. It is given by

$$K_{\mu\nu} - (\nabla_\mu n_\nu + n_\mu n^\alpha \nabla_\alpha n_\nu) \tag{2.21}$$

It can be seen that $K_{\mu\nu}$ is a purely spatial tensor, i.e. $n^\mu K_{\mu\nu} - n^\nu K_{\mu\nu} = 0$. This means that $K^{00} = K^{0i} = 0$, so we will generally deal with K_{ij}.

2.2.1 Evolution Equations

As noted above, Einstein's field equations in vacuum (Eq. (2.7)) lead to a set of six evolution equations (Eq. (2.9)). These equations are second order in g_{ij}. We can now write these evolution equations as 12 first order equations in terms of g_{ij} and K_{ij}:

$$\begin{aligned} \partial_t g_{ij} &= -2\alpha K_{ij} + \mathcal{L}_\beta g_{ij} && (2.22) \\ \partial_t K_{ij} &= -\nabla_i \nabla_j \alpha + \alpha(R_{ij} - 2K_{ik} K K^k_j + K K_{ij}) + \mathcal{L}_\beta K_{ij} && (2.23) \end{aligned}$$

where

$$\begin{aligned} \mathcal{L}_\beta g_{ij} &= \beta^k g_{ij,k} + g_{kj} \beta^k_{,i} + g_{ik} \beta^k_{,j} && (2.24) \\ \mathcal{L}_\beta K_{ij} &= \beta^k K_{ij,k} + K_{kj} \beta^k_{,i} + K_{ik} \beta^k_{,j}. && (2.25) \end{aligned}$$

2.2.2 Contraint Equations

In addition to the evolution equations noted in the previous section, the vacuum field equations (Eq. (2.7)) lead to a set of four constraint equations (Eq. (2.8)). These can be divided into the Hamiltonian constraint ($G_{00} = 0$)

and three momentum constrains ($G_{0i} = 0$). Like the evolution equations, in the ADM formalism the constraint equations can now be written in terms of the 3-metric g_{ij} and the extrinsic curvature K_{ij}:

$$\text{Hamiltonian constraints:} \quad {}^{(3)}R + K^2 - K_{ij}K^{ij} = 0 \tag{2.26}$$

$$\text{Momentum constraints:} \quad \nabla_j K^{ij} - \nabla_i K = 0. \tag{2.27}$$

2.2.3 The Conformal Transverse-Traceless Decomposition

In order to more easily solve the constraint equations, we need to get them into a form where they can be solved using numerical methods. To do this we follow the approach known as the conformal transverse-traceless decomposition developed by York and Piran [10]. This consists of converting the constraint equations into elliptic equations for four potentials, ϕ and w^i. Since these potentials will be solved for iteratively, an initial guess is needed for the metric g_{ij} and extrinsic curvature K_{ij}. The potentials appearing in the elliptic equations of the conformal transverse-traceless decomposition relate trial fields, indicated by an overhead tilde, to the solved fields. The trace of the extrinsic curvature, $K = K^i{}_i$, is left unchanged by this method (i.e. $K = \tilde{K}$). The four potentials modify only the trace free parts of the extrinsic curvature:

$$A_{ij} = K_{ij} - \frac{1}{3} g_{ij} K. \tag{2.28}$$

The relations between the four potentials, trial fields, and solved fields are:

$$
\begin{aligned}
g_{ij} &= \phi^4 \tilde{g}_{ij} && (2.29)\\
A^{ij} &= \phi^{-10}(\tilde{A}^{ij} + (\widetilde{lw})^{ij}) && (2.30)
\end{aligned}
$$

where

$$(\widetilde{lw})^{ij} \equiv \tilde{\nabla}^i w^j + \tilde{\nabla}^j w^i - \frac{2}{3}\tilde{g}^{ij}\tilde{\nabla}_k w^k. \qquad (2.31)$$

The conformal factor ϕ and the vector potential w^i are solved for using the elliptic equations:

$$
\begin{aligned}
\tilde{\nabla}^2\phi &= \frac{1}{8}\left[\tilde{R}\phi + \frac{2}{3}\tilde{K}^2\phi^5 - \phi^{-7}\left(\tilde{A}^{ij} + (\widetilde{lw}^{ij}\right)\left(\tilde{A}_{ij} + (\widetilde{lw})_{ij}\right)\right] && (2.32)\\
\tilde{\nabla}_j(\widetilde{lw})^{ij} &= \frac{2}{3}\tilde{g}^{ij}\phi^6\tilde{\nabla}_j\tilde{K} - \tilde{\nabla}_j\tilde{A}^i j. && (2.33)
\end{aligned}
$$

Chapter 3

Puncture Method

One of the difficulties in simulating black holes numerically is dealing with the singularity. One way this challenge can be handled is by essentially removing the region around the holes from the computational domain, a technique known as excision. The idea is that as long as the excised region lies within the apparent horizon of the black hole then no information should be able to propagate outward from within the black hole. Due to numerical error, this is not entirely true in practice. Additionally, *openGR* is not stable using excision and ADM evolution, though excision has been shown to successfully simulate black hole mergers using other formulations of General Relativity [11].

In recent years another approach to handling black hole singularities, known as the puncture method, has been developed by Campanelli [5]. This approach has proven to be very successful for stably evolving binary black hole mergers[6]. We follow the method outlined by Brügman, et al. [12].

3.1 Initial Data

For our initial geometry used to implement the puncture method, we adopt Brill-Lindquist wormhole topology[13] with $N+1$ asymptotically flat ends and thus N "throats". We can use this geometry to model N black hole initial data. The spatial metric g_{ij} is related to the conformal metric $\tilde{g}_{ij}$ via an initial conformal factor ψ_0:

$$g_{ij} = \psi^4 \tilde{g}_{ij}. \tag{3.1}$$

Prior to solving the initial constraint equations, the conformal background metric is chosen to be flat ($\tilde{g}_{ij} = \delta_{ij}$).

The conformal extrinsic curvature is split into trace and trace-free parts:

$$K_{ij} = \psi^{-2} A_{ij} + \frac{1}{3} g_{ij} K \tag{3.2}$$

where K is the trace of the extrinsic curvature ($K = g^{ij} K_{ij}$) and A_{ij} is trace-free. We also impose maximal slicing, $K = 0$, which decouples the Hamiltonian and momentum constraints. Linear and angular momentum are inserted via the momentum constraints, which can be written as

$$\partial_j A_{ij} = 0 \tag{3.3}$$

and allow for Bowen-York solutions[14] for any number of black holes. These solutions are of the form

$$A_{ij} = \frac{3}{2r^2}[P_i n_j + P_j n_i - (g_{ij} - n_i n_j) P_a n^a] + \frac{3}{r^3}(\epsilon_i^{ab} S_a n_b n_j + \epsilon_j^{ab} S_a n_b n_i) \tag{3.4}$$

where n_i is radial normal vector, ϵ_{ijk} is the three dimensional Levi-Civita tensor and P_i and S_i are the linear and angular momentum of the black hole, respectively. For the case of multiple black holes, the extrinsic curvature is simply given by the sum of the contributions from each hole.

The Hamiltonian constraint is an elliptic equation in terms of the conformal factor, which has a solution of the form[15]

$$\psi = u + \sum_{i=1}^{N} \frac{M_i}{2|r - r_i|}. \tag{3.5}$$

This solution is just a superposition of the solutions for the single black hole case plus a correction term u, which we must solve via

$$\nabla^2 u + \frac{1}{8}\psi^5 K_{ij} K^{ij} = 0. \tag{3.6}$$

It has been shown that the solution to Eq. (3.6) exists and is unique[15]. Additionally, since we have removed the singularities in Eq. (3.6), the solution is regular at the black holes, making it solvable numerically.

3.2 BSSN Formulation

In the puncture method, the initial data is evolved using the strongly hyperbolic BSSN formulation [16, 17]. The standard BSSN variables are ϕ, $\tilde{g_{ij}}$, $\tilde{A}_{ij}$, K, and $\tilde{\Gamma}^i$, and are related to those in the conformal transverse-traceless decomposition by:

$$\phi = \ln \psi_0 \tag{3.7}$$

$$\tilde{A}_{ij} = \psi^{-6} A_{ij} \tag{3.8}$$

$$\tilde{\Gamma}^i = -\partial_j \tilde{g}^{ij} \tag{3.9}$$

where $\tilde{g}_{ij}$ and K are unchanged.

The BSSN variables are evolved using the following evolution equations:

$$\partial_0 \phi = -\frac{1}{6}\alpha K \tag{3.10}$$

$$\partial_0 \tilde{g}_{ij} = -2\alpha \tilde{A}_{ij} \tag{3.11}$$

$$\partial_0 \tilde{A}_{ij} = e^{-r\phi}[-\nabla_i \nabla_j \alpha + \alpha R_{ij}]^{TF} + \alpha(K\tilde{A}_{ij} - 2\tilde{A}_{ik}\tilde{A}^k_{,j}) \tag{3.12}$$

$$\partial_0 K = -\nabla^i \nabla_i \alpha + \alpha(\tilde{A}_{ij}\tilde{A}^{ij} + \frac{1}{3}K^2) \tag{3.13}$$

$$\begin{aligned}\partial_t \tilde{\Gamma}^i = {} & \tilde{g}^{ij}\partial_j \partial_k \beta^i + \frac{1}{3}\tilde{g}^{ij}\partial_j\partial_k\beta^k + \beta^j\partial_j\tilde{\Gamma}^i - \tilde{\Gamma}^i\partial_j\beta^i + \frac{2}{3}\tilde{\Gamma}^i\partial_j\beta^j \\ & -2\tilde{A}^{ij}\partial_j\alpha + 2\alpha(\tilde{\Gamma}^i_{jk}\tilde{A}^{jk} + 6\tilde{A}^{ij}\partial_j\phi - \frac{2}{3}\tilde{g}^{ij}\partial_j K).\end{aligned} \tag{3.14}$$

where $\partial_0 = \partial_t - \mathcal{L}_\beta$, ∇_i is the covariant derivative with respect to the physical metric g_{ij}, and "TF" indicates the trace-free part of the expression with respect to the physical metric ($X^{TF}_{ij} = X_{ij} - \frac{1}{3}g_{ij}X^k{}_k$). The lie derivatives of the non-tensoral quantities are[5]:

$$\mathcal{L}_\beta \phi = \beta^k \partial_k \phi + \frac{1}{6}\partial_k \beta^k \tag{3.15}$$

$$\mathcal{L}_\beta \tilde{g}_{ij} = \tilde{g}_{ij}\partial_k \tilde{g}_{ij} + \tilde{g}_{ik}\partial_j\beta^k + \tilde{g}_{jk}\partial_i\beta^k - \frac{2}{3}\partial_k\beta^k \tag{3.16}$$

$$\mathcal{L}_\beta \tilde{A}_{ij} = \tilde{A}_{ij}\partial_k\tilde{A}_{ij}\tilde{A}_{ik}\partial_j\beta^k + \tilde{A}_{jk}\partial_i\beta^k - \frac{2}{3}\tilde{A}_{ij}\partial_k\beta^k \tag{3.17}$$

The Ricci tensor R_{ij} is:

$$
\begin{aligned}
R_{ij} &= \tilde{R}_{ij} + R^{\phi}_{ij} && (3.18)\\
\tilde{R}_{ij} &= -\frac{1}{2}\tilde{g}^{lm}\partial_l\partial_m\tilde{g}_{ij} + \tilde{g}_{k(i}\partial_{j)}\tilde{\Gamma}^k + \tilde{\Gamma}^k\tilde{\Gamma}_{(ij)k} \\
&\quad + \tilde{g}^{lm}(2\tilde{\Gamma}^k_{l(i}\tilde{\Gamma}_{j)jm} + \tilde{\Gamma}^k_{im}\tilde{\Gamma}_{klj}) && (3.19)\\
R^{\phi}_{ij} &= -2\tilde{\nabla}_i\tilde{\nabla}_j\phi - 2\tilde{g}_{ij}\tilde{\nabla}^k\nabla_k\phi + 4\tilde{\nabla}_i\phi\nabla_j\phi - 4\tilde{g}_{ij}\tilde{\nabla}^k\phi\tilde{\nabla}_k\phi && (3.20)
\end{aligned}
$$

where $\tilde{\nabla}$ is the covariant derivative with respect to the conformal metric $\tilde{g_{ij}}$.

It has been shown [9] that when evolving Eqs. (3.10 - 3.14) greater stability is achieved by enforcing the algebraic constraints $det(g) = 1$ and $Tr(A_{ij}) = 0$. We do this after every *PVODE* iteration. Additionally, whenever $\tilde{\Gamma}^i$ appears undifferentiated we substitute $\tilde{\Gamma}^i = -\partial_j\tilde{g}^{ij}$. If $\tilde{\Gamma}^i$ is differentiated we leave it unchanged.

3.3 Gauge Conditions

To allow the punctures to move (the so-called moving-puncture method), we use the covariant form of "1+log" slicing [18] for the lapse:

$$(\partial_t - \beta^i\partial_i)\alpha = -2\alpha K. \tag{3.21}$$

For the shift, we use a modified $\tilde{\Gamma}$-driver condition[6]:

$$
\begin{aligned}
(\partial_t - \beta^i\partial_i)\beta^i &= \frac{3}{4}B^i && (3.22)\\
(\partial_t - \beta^i\partial_i)B^i &= (\partial_t - \beta^i\partial_i)\tilde{\Gamma}^i - \eta B^i. && (3.23)
\end{aligned}
$$

In our case, $\eta = 1$, but it can also be greater than 1. It has been shown [19] that the BSSN equations with the above gauge choices are strongly hyperbolic and yield a well-posed initial value problem.

Chapter 4

Gravitational Wave Extraction

The presence of gravitational radiation associated with strong and changing gravitational fields is one of the most significant predictions of general relativity. Upgrades to existing interferometric detectors should allow direct observation of a gravitational wave in the very near future. As such, simulation of gravitational radiation is a particularly important aspect of numerical relativity because the numerical predictions of gravitational wave signals can be used as templates that can greatly improve the chances of a detection. This chapter serves as an introduction to the method by which we extract gravitational radiation from a simulation.

There are two main approaches to gravitational wave extraction. Originally, extraction was accomplished using various perturbation methods (Regge-Wheeler [20], Zerilli [21], Teukolsky [22]) which depend on knowledge of the background metric. But for simulations involving strong gravitational fields the background metric is generally not known. For example, when modeling merging black holes we do not know the final mass or spin of the merged hole a priori, so we need to use a method that does not depend on the background metric to extract the gravitational waves. In this chapter will present an approach that permits gravitational wave extraction without a priori knowledge of the background metric. We start with the Newman-Penrose formalism [23] which introduces five complex scalars, known as Weyl scalars, that contain all the necessary information about the curvature of spacetime. When these scalars are computed for a particular tetrad class, the Quasi-Kinnersley frame, they take on physical meaning. One scalar in particular, Ψ_4, contains the information about the gravitational radiation. We will then compute Ψ_4 in terms of scalar quantities, making it coordinate independent. Ψ_4 can then be decomposed into various modes and used to calculate the total energy, momentum, and angular momentum being carried away by the gravitational radiation.

Currently, the Weyl scalars are computed within *openGR* and are output on spheres of specified radii (Secs. 4.1 - 4.9). However, the spherical harmonic decomposition of Ψ_4 (Sec. 4.11) is performed using post-processing scripts courtesy of Uli Sperhake. The calculation of energy, momentum, and angular momentum (Sec. 4.12) is also performed using this set of scripts. These calculations will be incorporated into *openGR* in the future.

4.1 Tetrad Formalism

In developing the Newman-Penrose formalism, we want to work with a basis that is coordinate-independent. Consider a set of four linearly independent basis vectors $\vec{e}_{(a)}$ at every point in space. The Latin index in parentheses identifies the vector, and for this chapter we will change the usual convention and allow Latin characters inside parentheses to be $(0, 1, 2, 3)$. These vectors satisfy

$$\vec{e}_{(a)} \cdot \vec{e}_{(b)} = e_{(a)\mu} e^{\mu}_{(b)} = \eta_{(a)(b)}, \tag{4.1}$$

where $\eta_{(a)(b)}$ is a constant matrix independent of the position in spacetime. In this case the set of vectors $\{\vec{e}_{(a)}\}$ is known as a tetrad. Additionally, $\eta_{(a)(b)}$ are just the components of the metric tensor in the tetrad basis. In the case that the tetrad is orthonormal, $\eta_{(a)(b)}$ is just the Minkowski metric tensor.

Since the vectors $\{\vec{e}_{(a)}\}$ are linearly independent, the matrix $\eta_{(a)(b)}$ can be inverted, with inverse $\eta^{(a)(b)}$ such that

$$\eta^{(a)(c)} \eta_{(c)(b)} = \delta^{(a)}_{(b)}. \tag{4.2}$$

$\eta_{(a)(b)}$ is also used to raise and lower tetrad indices, e.g.:

$$\vec{e}_{(a)} = \eta^{(a)(b)} \vec{e}_{(a)}. \tag{4.3}$$

From this we find

$$\vec{e}^{(a)} \cdot \vec{e}_{(b)} = \delta^{(a)}_{(b)}. \tag{4.4}$$

When we express the tetrad vector $\vec{e}^{(a)}$ in terms of the coordinate basis $\{\vec{e}_\mu\}$ we can see

$$e^{(a)}{}_{\mu} e_{(a)}{}^{\mu} = \delta^{\mu}_{\mu}. \tag{4.5}$$

Eqs. (4.4) and (4.5) can be written together as:

$$e_{(a)}{}^{\mu} e^{(b)}{}_{\mu} = \delta^{(b)}_{(a)}, \tag{4.6}$$

$$e_{(a)}{}^{\mu} e^{(a)}{}_{\nu} = \delta^{\mu}_{\nu}. \tag{4.7}$$

From this, we determine that we can recover metric components in the coordinate frame in terms of the tetrad vector using:

$$g_{\mu\nu} = e_{(a)\mu} e^{(a)}{}_{\nu} = \eta^{(a)(b)} e_{(a)\mu} e_{(b)\nu}. \tag{4.8}$$

As an example, let us consider the Schwarzschild metric:

$$ds^2 = -\left(1 - \frac{2M}{r}\right) dt^2 + \left(1 - \frac{2M}{r}\right)^{-1} dr^2 + r^2 \left(d\theta^2 + sin^2\theta d\phi^2\right). \tag{4.9}$$

A straightforward choice for the tetrad in this case is:

$$e^{\mu}_{(0)} = \left(1 - \frac{2M}{r}\right)^{\frac{1}{2}} (dt)^{\mu}, \tag{4.10}$$

$$e^{\mu}_{(1)} = \left(1 - \frac{2M}{r}\right)^{-\frac{1}{2}} (dr)^{\mu}, \tag{4.11}$$

$$e^{\mu}_{(2)} = r\,(d\theta)^{\mu}, \tag{4.12}$$

$$e^{\mu}_{(3)} = r\,sin\theta\,(d\phi)^{\mu}. \tag{4.13}$$

4.2 The Newman Penrose Formalism

4.2.1 Null Tetrads

The Newman Penrose formalism consists of constructing a tetrad of null vectors. We start with an orthonormal tetrad $\{\vec{e}_{(a)}\}$ so that $\eta_{(a)(b)}$ corresponds to the Minkowski metric tensor. We can then use Eq. (4.8) to write the spacetime metric as

$$g_{\mu\nu} = -e_{(0)\mu}e_{(0)\nu} + e_{(1)\mu}e_{(1)\nu} + e_{(2)\mu}e_{(2)\nu} + e_{(3)\mu}e_{(3)\nu}. \tag{4.14}$$

We typically chose $e^{\mu}_{(0)}$ as the unit normal vector to the spacial hypersurfaces, $e^{\mu}_{(1)}$ as the radial unit vector in spherical coordinates, and $(e^{\mu}_{(2)}, e^{\mu}_{(3)})$ as unit vectors in the angular directions, which are found by performing a Gram-Schmidt orthogonalization procedure. We can see these choices reflected in Eqs. (4.10 - 4.13).

Once we have an orthonormal basis we now construct our null tetrad $\{\vec{e}_{(a)}\} = \{l, k, m, \bar{m}\}$. The vectors l and k are real, and m and $\bar{m}$ are complex, with the bar denoting the complex conjugate. The vectors that make up the tetrad are:

$$\begin{aligned} l^\mu &= \frac{1}{\sqrt{2}}\left(e^\mu_{(0)} + e^\mu_{(1)}\right), && (4.15)\\ k^\mu &= \frac{1}{\sqrt{2}}\left(e^\mu_{(0)} - e^\mu_{(1)}\right), && (4.16)\\ m^\mu &= \frac{1}{\sqrt{2}}\left(e^\mu_{(2)} + ie^\mu_{(3)}\right), && (4.17)\\ \bar{m}^\mu &= \frac{1}{\sqrt{2}}\left(e^\mu_{(2)} - ie^\mu_{(3)}\right). && (4.18) \end{aligned}$$

These four vectors have the orthogonality properties

$$l_\mu l^\mu = k_\mu k^\mu = m_\mu m^\mu = \bar{m}_\mu \bar{m}^\mu = 0, \tag{4.19}$$

$$l_\mu m^\mu = l_\mu \bar{m}^\mu = k_\mu m^\mu = k_\mu \bar{m}^\mu = 0, \tag{4.20}$$

$$l_\mu k^\mu = -m_\mu \bar{m}^\mu = -1. \tag{4.21}$$

Using this null tetrad of vectors, $\eta_{(a)(b)}$ becomes

$$\eta_{(a)(b)} = \eta^{(a)(b)} = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{4.22}$$

and the spacetime metric $g_{\mu\nu}$ is

$$g_{\mu\nu} = -l_\mu k_\nu - k_\nu l_\nu + m_\mu \bar{m}_\nu + \bar{m}_\mu m_\nu. \tag{4.23}$$

4.2.2 Weyl Tensor and Weyl Scalars

Now that we have introduced a tetrad of null vectors, we can calculate scalar quantities which contain all of the curvature information we are interested in. We start by writing the Riemann tensor in the tetrad formalism:

$$\begin{aligned} R_{(a)(b)(c)(d)} &= \frac{1}{2}(\eta_{(a)(c)}R_{(b)(d)} - \eta_{(b)(c)}R_{(a)(d)} - \eta_{(a)(d)}R_{(b)(c)} + \eta_{(b)(d)}R_{(a)(c)}) \\ &- \frac{1}{6}(\eta_{(a)(c)}\eta_{(b)(d)} - \eta_{(a)(d)}\eta_{(b)(c)})R + C_{(a)(b)(c)(d)}, \end{aligned} \tag{4.24}$$

where $R_{(a)(b)}$ is the Ricci tensor, R is the Ricci scalar, and $C_{(a)(b)(c)(d)}$ is the Weyl tensor. We will consider the vacuum case in which both the Ricci tensor and the Ricci scalar vanish, leaving the Weyl tensor equal to the Riemann tensor. Given the symmetries of the Riemann tensor (and hence the Weyl tensor), the Weyl tensor has 10 independent components and contains all of the curvature information about the vacuum spacetime. These 10 independent components can be written in terms of 5 complex scalars, known as the Weyl scalars. They expressed as follows:

$$\Psi_0 \equiv C_{(1)(3)(1)(3)} = C_{\alpha\beta\mu\nu} l^\alpha m^\beta l^\mu m^\nu, \tag{4.25}$$

$$\Psi_1 \equiv C_{(1)(2)(1)(3)} = C_{\alpha\beta\mu\nu} l^\alpha k^\beta l^\mu m^\nu, \tag{4.26}$$

$$\begin{aligned} \Psi_2 &\equiv \frac{1}{2}(C_{(1)(2)(1)(2)} + C_{(1)(2)(3)(4)}) \\ &= \frac{1}{2}C_{\alpha\beta\mu\nu}(l^\alpha k^\beta l^\mu k^\nu + l^\alpha k^\beta m^\mu \bar{m}^\nu), \end{aligned} \tag{4.27}$$

$$\Psi_3 \equiv C_{(1)(2)(4)(2)} = C_{\alpha\beta\mu\nu} l^\alpha k^\beta \bar{m}^\mu k^\nu, \tag{4.28}$$

$$\Psi_4 \equiv C_{(2)(4)(2)(4)} = C_{\alpha\beta\mu\nu} k^\alpha \bar{m}^\beta k^\mu \bar{m}^\nu. \tag{4.29}$$

Like the Riemann tensor, these scalars contain all of the curvature information about the spacetime. Additionally, since they are scalars they do not depend on the choice of coordinates. They do, however, depend on the particular tetrad of null vectors that is chosen, as will be shown in the following sections.

4.3 Tetrad Transformations

The tetrad (l^μ,k^μ,m^μ,$\bar{m}^\mu$) depends on the choice of the orthonormal tetrad $\{\vec{e}_{(a)}\}$. We can apply a combination of a spatial rotation and a Lorentz boost in a given direction which will change the tetrad while keeping it orthonormal. Thus, there are six degrees of freedom corresponding to possible transformations that will not change the formalism previously discussed. These transformations are typically separated into three types.

4.3.1 Type I Rotations

Type I rotations leave the vector $\vec{l}$ unchanged:

$$l^\mu \to l^\mu, \tag{4.30a}$$

$$m^\mu \to m^\mu + a l^\mu, \tag{4.30b}$$

$$\bar{m}^\mu \to \bar{m}^\mu + \bar{a} l^\mu, \tag{4.30c}$$

$$k^\mu \to k^\mu + \bar{a} m^\mu + a \bar{m}^\mu + a\bar{a} l^\mu. \tag{4.30d}$$

where a is a complex parameter and $\bar{a}$ is its complex conjugate. The Weyl scalars transform as follows:

$$\Psi_0^I \rightarrow \Psi_0, \tag{4.31a}$$

$$\Psi_1^I \rightarrow \Psi_1 + \bar{a}\Psi_0, \tag{4.31b}$$

$$\Psi_2^I \rightarrow \Psi_2 + 2\bar{a}\Psi_1 + \bar{a}^2\Psi_0, \tag{4.31c}$$

$$\Psi_3^I \rightarrow \Psi_3 + 3\bar{a}\Psi_2 + 3\bar{a}\Psi_1 + \bar{a}^3\Psi_0, \tag{4.31d}$$

$$\Psi_4^I \rightarrow \Psi_4 + 4\bar{a}\Psi_3 + 6\bar{a}^2\Psi_2 + 4\bar{a}^3\Psi_1 + \bar{a}^4\Psi_0. \tag{4.31e}$$

4.3.2 Type II Rotations

A type II rotation leaves the vector $\vec{k}$ unchanged:

$$l^\mu \rightarrow l^\mu + \bar{b}m^\mu + b\bar{m}^\mu + b\bar{b}l^\mu, \tag{4.32a}$$

$$m^\mu \rightarrow m^\mu + bk^\mu, \tag{4.32b}$$

$$\bar{m}^\mu \rightarrow \bar{m}^\mu + \bar{b}k^\mu, \tag{4.32c}$$

$$k^\mu \rightarrow k^\mu, \tag{4.32d}$$

where b is a complex parameter and $\bar{b}$ is its complex conjugate. The Weyl scalars transform under a type II rotation as follows:

$$\Psi_0^{II} \rightarrow \Psi_0 + 4b\Psi_1 + 6b^2\Psi_2 + 4b^3\Psi_3 + b^4\Psi_4, \tag{4.33a}$$

$$\Psi_1^{II} \rightarrow \Psi_1 + 3b\Psi_2 + 3b^2\Psi_3 + b^3\Psi_4, \tag{4.33b}$$

$$\Psi_2^{II} \rightarrow \Psi_2 + 2b\Psi_3 + b^2\Psi_4, \tag{4.33c}$$

$$\Psi_3^{II} \rightarrow \Psi_3 + b\Psi_4, \tag{4.33d}$$

$$\Psi_4^{II} \rightarrow \Psi_4. \tag{4.33e}$$

4.3.3 Type III Rotations

A type III rotation scales the vectors $\vec{l}$ and $\vec{k}$ while leaving their direction unchanged, as well as leaving the product $l_\mu k^\mu$ unchanged:

$$l^\mu \quad \rightarrow \quad \lambda^{-1} l^\mu, \tag{4.34}$$

$$m^\mu \quad \rightarrow \quad e^{i\theta} m^\mu, \tag{4.35}$$

$$\bar{m}^\mu \quad \rightarrow \quad e^{-i\theta} \bar{m}^\mu, \tag{4.36}$$

$$k^\mu \quad \rightarrow \quad \lambda k^\mu, \tag{4.37}$$

where λ and θ are two real parameters. These two parameters, together with the complex parameters a and b, make up the six degrees of freedom. A type III rotation can be interpreted as a Lorentz boost in the $(\vec{l},\vec{k})$ plane plus a rotation in the $(\vec{m},\vec{\bar{m}})$ plane. As such, type III rotations are commonly referred to as spin-boost transformations. Under a spin-boost transformation the Weyl scalars transform as follows:

$$\Psi_0^{III} \quad \rightarrow \quad \lambda^{-2} e^{2i\theta} \Psi_0, \tag{4.38a}$$

$$\Psi_1^{III} \quad \rightarrow \quad \lambda^{-1} e^{i\theta} \Psi_1, \tag{4.38b}$$

$$\Psi_2^{III} \quad \rightarrow \quad \Psi_2, \tag{4.38c}$$

$$\Psi_3^{III} \quad \rightarrow \quad \lambda e^{-i\theta} \Psi_3, \tag{4.38d}$$

$$\Psi_4^{III} \quad \rightarrow \quad \lambda^2 e^{-2i\theta} \Psi_4. \tag{4.38e}$$

4.4 Null Tetrads and Null Frames

It is useful to use terminology that makes a clear distinction between null frames and null tetrads. We define the two as follows:

- A *null tetrad* is a specific set of two real null vectors $\vec{l}$ and $\vec{k}$ and two complex conjugate null vectors $\vec{m}$ and $\vec{\bar{m}}$.
- A *null frame* is a class of null tetrads connected by a spin-boost (type III) transformation.

4.5 Curvature Invariants

In general, there are 14 independent invariant quantities which can be constructed from the Riemann curvature tensor in four dimensions. In vacuum, only four of these invariants are non-zero. We can express these four invariants as two complex scalars I and J. These scalars are independent of both the coordinates and the choice of tetrad. The scalar curvature invariants I and J can be written in terms of the Weyl tensor as:

$$I = \frac{1}{16}\left(C_{\mu\nu}{}^{\rho\sigma}C_{\rho\sigma}{}^{\mu\nu} - i\,C_{\mu\nu}{}^{\rho\sigma}\;{}^{\sim}C_{\rho\sigma}{}^{\mu\nu}\right) \tag{4.39}$$

and

$$J = \frac{1}{96}(C_{\mu\nu}{}^{\rho\sigma}C_{\rho\sigma}{}^{\alpha\beta}C_{\alpha\beta}{}^{\mu\nu} - C_{\mu\nu}{}^{\rho\sigma}C_{\rho\sigma}{}^{\alpha\beta}\;{}^{\sim}C_{\alpha\beta}{}^{\mu\nu}), \tag{4.40}$$

where ${}^{\sim}C_{\mu\nu}{}^{\rho\sigma} \equiv \frac{1}{2}{}^{4}\varepsilon_{\mu\nu}{}^{\delta\lambda}C_{\delta\lambda\sigma\rho}$ is the Hodge dual of the Weyl tensor and ${}^{4}\varepsilon_{\mu\nu}{}^{\delta\lambda}$ is the four dimensional Levi-Civita symbol. We can also express I and J in terms of the Weyl scalars as follows:

$$I = 3\Psi_2^2 - 4\Psi_1\Psi_3 + \Psi_0\Psi_4 \tag{4.41}$$

$$J = det\begin{vmatrix} \Psi_0 & \Psi_1 & \Psi_2 \\ \Psi_1 & \Psi_2 & \Psi_3 \\ \Psi_2 & \Psi_3 & \Psi_4 \end{vmatrix}. \tag{4.42}$$

4.6 Principal Null Directions

The path of light emanating from a spherical source and propagating through curved space will cause the image of the spherical source to be distorted as perceived by a distant observer. The amount of this distortion can be quantized by

$$D = \frac{1}{2}C(r,\theta,\phi)r^2 \tag{4.43}$$

where r is the distance to the source and $C(r,\theta,\phi)$ is the projection of the Weyl scalar on the tangent plane of the celestial sphere. There are four distortion-free directions (i.e. $D = 0$) known as principal null directions [24]. Penrose showed that these four directions can be found by setting $\Psi_0 = 0$ after a type II rotation (Eq. 4.33a), i.e.:

$$\Psi_0^{II} = \Psi_0 + 4b\Psi_1 + 6b^2\Psi_2 + 4b^3\Psi_3 + b^4\Psi_4 = 0. \tag{4.44}$$

To solve Eq. (4.44) we will follow the method outlined by Gunnarsen [25]. We begin by introducing a new variable

$$y = \Psi_4 b + \Psi_3. \tag{4.45}$$

Substituting this variable into Eq. (4.44) gives

$$y^4 + 6Hy^2 + 4Gy + K = 0, \tag{4.46}$$

where the coefficients are combinations of the Weyl scalars given by

$$K \equiv \Psi_4^2 I - 3H^2 \tag{4.47}$$

$$H \equiv \Psi_4\Psi_2 - \Psi_3^2 \tag{4.48}$$

$$G \equiv \Psi_4^2\Psi_1 - 3\Psi_4\Psi_3\Psi_2 + 2\Psi_3^2, \tag{4.49}$$

and we use the definitions of the curvature invariants I and J given in Eqs. (4.41) and (4.42). Solutions of Eq. (4.46) can be expressed in terms of solutions to the following equation:

$$\lambda^3 - I\lambda + 2J = 0. \tag{4.50}$$

This equation has the following solutions:

$$\lambda_1 = -\left(P + \frac{I}{3P}\right), \tag{4.51a}$$

$$\lambda_2 = -\left(e^{\frac{2\pi i}{3}}P + e^{\frac{4\pi i}{3}}\frac{I}{3P}\right), \tag{4.51b}$$

$$\lambda_3 = -\left(e^{\frac{4\pi i}{3}}P + e^{\frac{2\pi i}{3}}\frac{I}{3P}\right), \tag{4.51c}$$

where

$$P = \left[J + \sqrt{J^2 - (I/3)^3}\right]^{\frac{1}{3}}. \tag{4.52}$$

From Eqs. (4.51) we determine three new complex variables α,β,γ using the following:

$$\begin{aligned} \alpha^2 &= 2\Psi_4\lambda_1 - 4H && (4.53a) \\ \beta^2 &= 2\Psi_4\lambda_2 - 4H && (4.53b) \\ \gamma^2 &= \alpha^2 + \beta^2 + 4H && (4.53c) \\ \alpha\beta\gamma &= 4G. && (4.53d) \end{aligned}$$

The values of α,β,and γ are determined up to a sign by Eqs. (4.53a - 4.53c) and the sign is determined by Eq. (4.53d). Eqs. (4.51) can be written in terms of these new variables as:

$$\begin{aligned} \lambda_1 &= \frac{\alpha^2 + 4\,H}{2\,\Psi_4}, && (4.54a) \\ \lambda_2 &= \frac{\beta^2 + 4\,H}{2\,\Psi_4}, && (4.54b) \\ \lambda_3 &= \frac{\gamma^2 + 4\,H}{2\,\Psi_4}. && (4.54c) \end{aligned}$$

We can now write the solutions of Eq. (4.46) in terms of α, β, and γ:

$$\begin{aligned} y_1 &= \frac{1}{2}\Big(\alpha + \beta + \gamma\Big), && (4.55a) \\ y_2 &= \frac{1}{2}\Big(\alpha - \beta - \gamma\Big), && (4.55b) \\ y_3 &= \frac{1}{2}\Big(-\alpha + \beta - \gamma\Big), && (4.55c) \\ y_4 &= \frac{1}{2}\Big(-\alpha - \beta + \gamma\Big). && (4.55d) \end{aligned}$$

Setting Eqs. (4.54) equal to Eqs. (4.51) yields the final expressions for α, β, and γ. Once we have those expressions we can now write down the final solution of Eq. (4.44), giving us the principal null directions:

$$b_1 = -\frac{\Psi_3 + \frac{1}{2}\Big(\alpha + \beta + \gamma\Big)}{\Psi_4}, \tag{4.56a}$$

$$b_2 = -\frac{\Psi_3 + \frac{1}{2}\Big(\alpha - \beta - \gamma\Big)}{\Psi_4}, \tag{4.56b}$$

$$b_3 = -\frac{\Psi_3 + \frac{1}{2}\Big(-\alpha + \beta - \gamma\Big)}{\Psi_4}, \tag{4.56c}$$

$$b_4 = -\frac{\Psi_3 + \frac{1}{2}\Big(-\alpha - \beta + \gamma\Big)}{\Psi_4}. \tag{4.56d}$$

4.7 Petrov Classification

We found the principal null directions by solving Eq. (4.44), i.e. finding its roots. These roots are listed in Eqs. (4.56). Spacetimes can be separated into different types, known as Petrov Types, based on the number of distinct roots (and hence principal null directions) in the spacetime. Table 4.1 summarizes the different Petrov classifications. We are particularly interested in Petrov type I and type D spacetimes, so we will highlight those types.

Petrov Type	
I	Four distinct principal null directions
II	Two coinciding principal null directions
D	Two pairs of coinciding principal null directions
III	Three coinciding principal null directions
N	All four principal null directions coincide

Table 4.1: Table of Petrov Classifications. The Petrov type is determined by the number of coinciding principal null directions.

4.7.1 Petrov Type I

In a Petrov type I spacetime there are four distinct principal null directions. By performing a class II transformation (Eqs. 4.33) followed by a class I transformation (Eqs. 4.31), it is possible to set two of the Weyl scalars to zero. In practice will set $\Psi_1 = \Psi_3 = 0$ in order to find the transverse frames, which we will discuss in the following sections.

4.7.2 Petrov Type D

A Petrov type D spacetime has two pairs of coinciding principal null directions. These spacetimes have the relation $27J^2 = I^3$, which is the condition that distinguishes special Petrov types (II, D, III or N) from the general Petrov type I. We can perform a transformation such that the null vector $\vec{k}$ points along one of the principal null directions. This sets $\Psi_0 = \Psi_1 = 0$. Pointing the null vector $\vec{l}$ along the other pair of repeated principal null directions sets $\Psi_3 = \Psi_4 = 0$, leaving only Ψ_2 nonzero. It can be shown that Ψ_2 contains all the information about the background. As an example, we note that both Schwarzschild and Kerr spacetimes are of Petrov type D. For a Schwarzschild spacetime we find

$$\Psi_2{}^{(Schwarzschild)} = -\frac{M}{r^3}. \tag{4.57}$$

For a Kerr spacetime we find

$$\Psi_2{}^{(Kerr)} = -\frac{M}{(r+ia)^3}, \tag{4.58}$$

where a is the Kerr spin parameter. A Petrov type D spacetime is also known as a Kinnersley frame.

4.8 Quasi-Kinnersley Frame

Now that we have calculated the Weyl scalars (Sec. 4.2), we need to chose an appropriate tetrad such that the scalars contain the expected information. In general the information about the spacetime is divided among the Weyl scalars. However, for tetrads that are infinitesimally close to type D the scalars Ψ_0 and Ψ_4 are invariant under both tetrad and gauge transformations and as such have specific physical meaning: Ψ_4 is the outward traveling gravitational radiation and Ψ_0 is the inward traveling radiation. However, we do not have complete knowledge of the spacetime a priori, as it will change with time. We will follow the method outlined by Nerozzi [26] in which we will chose the correct tetrad frame assuming that the spacetime is type D. We start with three definitions:

1. A transverse frame is a frame in which $\Psi_1 = \Psi_3 = 0$. For spacetimes of type I, Ψ_1 and Ψ_3 are gauge choices which can be set to zero via tetrad transformations.

2. A Kinnersley frame for a type D spacetime provides that the two real null tetrad vectors ($\vec{l}$ and $\vec{k}$) coincide with the two repeated principal null directions of the Weyl tensor.

3. A quasi-Kinnersley frame for a type I spacetime converges to a Kinners-

ley frame when $S = \frac{27J^2}{I^3} \to 1$.The quasi-Kinnersley frame is a transverse frame with the added condition that as the spacetime converges to a Kinnersley frame ($S \to 1$), the Weyl scalars Ψ_0 and Ψ_4 both tend towards zero. We introduce a radiation scalar $\xi = \Psi_0\Psi_4$ with the condition that $\xi \to 0$ as $S \to 1$.

To find the quasi-Kinnersley frame in a type I spacetime we start by performing a type I rotation followed by a type II rotation:

$$\begin{aligned}
\Psi_0 &\to \Psi_0^I = \Psi_0, \\
\Psi_1 &\to \Psi_1^I = \Psi_1 + \bar{a}\Psi_0, \\
\Psi_2 &\to \Psi_2^I = \Psi_2 + 2\bar{a}_1^{\Psi} + \bar{a}^2\Psi_0, \\
\Psi_3 &\to \Psi_3^I = \Psi_3 + 3\bar{a}\Psi_2 + 3\bar{a}^2\Psi_1 + \bar{a}^3\Psi_0, \\
\Psi_4 &\to \Psi_4^I = \Psi_4 + 4\bar{a}\Psi_3 + 6\bar{a}^2\Psi_2 + 4\bar{a}^3\Psi_1 + \bar{a}^4\Psi_0, \\
\Psi_0^I &\to \Psi_0^{II} = \Psi_0^I + 4b\Psi_1^I + 6b^2\Psi_2^I + 4b^3\Psi_3^I + b^4\Psi_4^I, \qquad (4.59) \\
\Psi_1^I &\to \Psi_1^{II} = \Psi_1^I + 3b\Psi_2^I + 3b^2\Psi_3^I + b^3\Psi_4^I, \\
\Psi_2^I &\to \Psi_2^{II} = \Psi_2^I + 2b\Psi_3^I + b^2\Psi_4^I, \\
\Psi_3^I &\to \Psi_3^{II} = \Psi_3^I + b\Psi_4^I, \\
\Psi_4^I &\to \Psi_4^{II} = \Psi_4^I.
\end{aligned}$$

Given the definition above, we set $\Psi_1^{II} = \Psi_3^{II} = 0$ and plug this into the above rotation equations. We are then left to solve the following

$$\begin{gathered}\Psi_3 + 3\bar{a}\Psi_2 + 3\bar{a}^2\Psi_1 + \bar{a}^2\Psi_0 + \\ b\left(\Psi_4 + 4\bar{a}\Psi_3 + 6\bar{a}^2\Psi_2 + \bar{a}^3\Psi_1 + \bar{a}^4\Psi_0\right) = 0,\end{gathered} \tag{4.60a}$$

$$\begin{gathered}\Psi_1 + \bar{a}\Psi_0 + 3b\left(\Psi_2 + \bar{a}\Psi_1 + \bar{a}^2\Psi_0\right) + \\ 3b^2\left(\Psi_3 + 3\bar{a}\Psi_2 + 3\bar{a}^2\Psi_1 + \bar{a}^3\Psi_0\right) + \\ b^3\left(\Psi_4 + 4\bar{a}\Psi_3 + 6\bar{a}^2\Psi_2 + 4\bar{a}^3\Psi_1 + \bar{a}^4\Psi_0\right) = 0.\end{gathered} \tag{4.60b}$$

For a Petrov type I spacetime b has the following form:

$$b = -\frac{\Psi_3 + 3\bar{a}\Psi_2 + 3\bar{a}^2\Psi_1 + \bar{a}^2\Psi_0}{\Psi_4 + 4\bar{a}\Psi_3 + 6\bar{a}^2\Psi_2 + \bar{a}^3\Psi_1 + \bar{a}^4\Psi_0}. \tag{4.61}$$

Substituting this into Eq. (4.60b) we obtain a sixth order polynomial for $\bar{a}$:

$$\mathcal{P}_6\bar{a}^6 + \mathcal{P}_5\bar{a}^5 + \mathcal{P}_4\bar{a}^4 + \mathcal{P}_3\bar{a}^3 + \mathcal{P}_2\bar{a}^2 + \mathcal{P}_1\bar{a}^1 + \mathcal{P}_0 = 0, \tag{4.62}$$

whose coefficients $\mathcal{P}_n$ are given by:

$$\mathcal{P}_6 = -\Psi_3\Psi_0^2 - 2\Psi_1^3 + 3\Psi_2\Psi_1\Psi_0, \tag{4.63}$$

$$\mathcal{P}_5 = -2\Psi_3\Psi_1\Psi_0 - \Psi_0^2\Psi_4 + 9\Psi_2^2\Psi_0 - 6\Psi_2\Psi_1^2, \tag{4.64}$$

$$\mathcal{P}_4 = -5\Psi_1\Psi_4\Psi_0 - 10\Psi_3\Psi_1^2 + 15\Psi_3\Psi_2\Psi_0, \tag{4.65}$$

$$\mathcal{P}_3 = -10\Psi_4\Psi_1^2 + 10\Psi_3^2\Psi_0, \tag{4.66}$$

$$\mathcal{P}_2 = 5\Psi_3\Psi_0\Psi_4 + 10\Psi_1\Psi_3^2 - 15\Psi_1\Psi_2\Psi_4, \tag{4.67}$$

$$\mathcal{P}_1 = 2\Psi_3\Psi_1\Psi_4 + \Psi_4^2\Psi_0 - 9\Psi_2^2\Psi_4 + 6\Psi_2\Psi_3^2, \tag{4.68}$$

$$\mathcal{P}_0 = \Psi_1\Psi_4^2 + 2\Psi_3^3 - 3\Psi_2\Psi_3\Psi_4. \tag{4.69}$$

Although Eq.(4.62) is sixth order, there are only three transverse frames due to the the degeneracy of $\vec{l}$ and $\vec{k}$. An exchange of $\vec{l}$ and $\vec{k}$ produces an exchange in the Weyl scalars given by

$$\begin{aligned}\Psi_0 &\leftrightarrow \bar{\Psi}_4,\\ \Psi_2 &\leftrightarrow \Psi_2,\\ \Psi_4 &\leftrightarrow \bar{\Psi}_0.\end{aligned} \tag{4.70}$$

If we also exchange $\vec{m}$ and $\vec{\bar{m}}$ accordingly, then the complex conjugation is unnecessary. In this case we have simply switched which scalars contain the inward and outward radiation. Thus we see that there are three transverse frames for a Petrov type I spacetime, and we will find that one of them is the quasi-Kinnersley frame. We also note that once we find the parameter a from the solution of Eq. (4.62), $\bar{a}$, b can be calculated. We will focus our attention on solving for a, which is done in the following section.

4.9 Finding the Quasi-Kinnersley Frame

In terms of the parameters α, β, and γ, the solutions for the three transverse frames are given by

$$\bar{a}^{I}_{\pm} = \frac{1}{2\,\alpha}\left[\beta\,\gamma\,\sqrt{(\alpha^2-\beta^2)\,(\alpha^2-\gamma^2)}\right], \tag{4.71a}$$

$$\bar{a}^{II}_{\pm} = \frac{1}{2\,\beta}\left[\alpha\,\gamma\,\sqrt{(\beta^2-\alpha^2)\,(\beta^2-\gamma^2)}\right], \tag{4.71b}$$

$$\bar{a}^{III}_{\pm} = \frac{1}{2\,\gamma}\left[\alpha\,\beta\,\sqrt{(\gamma^2-\alpha^2)\,(\gamma^2-\beta^2)}\right]. \tag{4.71c}$$

The ± is due to the degeneracy in $\vec{l}$ and $\vec{k}$. It is necessary to determine which of these frames corresponds to the quasi-Kinnersley frame. To do so, we need to satisfy the criteria $S \rightarrow 1$ as $\xi \rightarrow 0$. It can be seen that $S \rightarrow 1 \implies P \rightarrow$

${}^{3}\!\sqrt{J} \implies {}^{3}\!\sqrt{J} \to \sqrt{I}$. Using Eq. (4.51), we find λ behaves in the following way:

$$\begin{aligned} \lambda_1 &\to -2\sqrt{\frac{\mathrm{I}}{3}}, && (4.72a)\\ \lambda_2 &\to \sqrt{\frac{\mathrm{I}}{3}}, && (4.72b)\\ \lambda_3 &\to \sqrt{\frac{\mathrm{I}}{3}}. && (4.72c) \end{aligned}$$

Ψ_2 in the three frames then has the value:

$$\begin{aligned} \Psi_2^{I} &= \frac{\lambda_1}{2}, && (4.73a)\\ \Psi_2^{II} &= \frac{\lambda_2}{2}, && (4.73b)\\ \Psi_2^{III} &= \frac{\lambda_3}{2}. && (4.73c) \end{aligned}$$

The radiation scalar ξ then becomes:

$$\begin{aligned} \xi^{\,I} &= \frac{(\lambda_2-\lambda_3)^2}{4}, && (4.74a)\\ \xi^{\,II} &= \frac{(\lambda_1-\lambda_3)^2}{4}, && (4.74b)\\ \xi^{\,III} &= \frac{(\lambda_1-\lambda_2)^2}{4}. && (4.74c) \end{aligned}$$

Eqs. (4.72) show that the equations for ξ in the case $S \to 1$ are

$$\begin{aligned} \xi^{\,I} &\to 0, && (4.75a)\\ \xi^{\,II} &\to \frac{3\,\mathrm{I}}{4}, && (4.75b)\\ \xi^{\,III} &\to \frac{3\,\mathrm{I}}{4}. && (4.75c) \end{aligned}$$

Thus, we determine that the transverse frame I is the quasi-Kinnersley frame.

4.10 Ψ_4 in Terms of Scalar Quantities

This section describes the treatment for computing Ψ_4 in *openGR*, following the procedure outlined by Nerozzi [27]. We can construct the electric and magnetic parts of the Weyl tensor as follows:

$$
\begin{aligned}
E_{\alpha\gamma} &= -C_{\alpha\beta\gamma\delta} e^{\mu}_{(0)} e^{\nu}_{(0)}, && (4.76)\\
B_{\alpha\gamma} &= -\frac{1}{2} {}^{4}\varepsilon^{\mu\nu}_{\alpha\beta} C_{\alpha\beta\mu\nu} e^{\mu}_{(0)} e^{\nu}_{(0)}, && (4.77)
\end{aligned}
$$

where we recall $e_{(0)^{\mu}}$ is the unit normal vector on the spatial hypersurface first used in Eq. (4.14) and ${}^{4}\varepsilon^{\mu\nu}_{\alpha\beta}$ is the four dimensional Levi-Civita tensor. The Weyl scalars can now be written in terms of the electric and magnetic parts of the Weyl tensor as follows:

$$
\begin{aligned}
\Psi_0 &= -(E_{\beta\gamma} - iB_{\beta\gamma}) m^{\beta} m^{\gamma}, && (4.78a)\\
\Psi_1 &= -(E_{\beta\gamma} - iB_{\beta\gamma}) m^{\beta} e^{\gamma}_{(1)}, && (4.78b)\\
\Psi_2 &= -(E_{\beta\gamma} - iB_{\beta\gamma}) e^{\beta}_{(1)} e^{\gamma}_{(1)}, && (4.78c)\\
\Psi_3 &= -(E_{\beta\gamma} - iB_{\beta\gamma}) \bar{m}^{\beta} e^{\gamma}_{(1)}, && (4.78d)\\
\Psi_4 &= -(E_{\beta\gamma} - iB_{\beta\gamma}) \bar{m}^{\beta} \bar{m}^{\gamma}. && (4.78e)
\end{aligned}
$$

After finding a frame in which $\Psi_1 = \Psi_3 = 0$, Eq. (4.78e) can be written as follows

$$
(\Psi_4)_{TF} = \frac{E_y - E_x}{2} + i\frac{B_x - B_y}{2}, \tag{4.79}
$$

where "TF" denotes the transverse frame. After calculating the eigenvalues of the electric and magnetic parts of the Weyl tensor, the Weyl scalar Ψ_4 can be expressed in terms of a magnitude and phase given by

$$\mathrm{Re}[(\Psi_4)_{TF}] = -\sqrt{3}|\mathcal{E}|\sin\left(\Theta_{\mathcal{E}} + \frac{2k\pi}{3}\right), \tag{4.80}$$

$$\mathrm{Im}[(\Psi_4)_{TF}] = \sqrt{3}|\mathcal{B}|\sin\left(\Theta_{\mathcal{B}} + \frac{2k\pi}{3}\right), \tag{4.81}$$

In general k can assume values $\{-1, 0, 1\}$, but it is set to 0 in *openGR*. The magnitudes $\mathcal{E}$ and $\mathcal{B}$ are given by

$$|\mathcal{E}| = \sqrt{\frac{E_{\alpha\beta}E^{\alpha\beta}}{6}}, \tag{4.82a}$$

$$|\mathcal{B}| = \sqrt{\frac{B_{\alpha\beta}B^{\alpha\beta}}{6}}, \tag{4.82b}$$

and the phases $\Theta_{\mathcal{E}}$ and $\Theta_{\mathcal{B}}$ are

$$\Theta_{\mathcal{E}} = \frac{1}{3}\arccos\left[\sqrt{6}\left(\frac{E_{\alpha\beta}E^{\beta}{}_{\gamma}E^{\gamma\alpha}}{[E_{\alpha\beta}]^{\frac{3}{2}}}\right)\right], \tag{4.83a}$$

$$\Theta_{\mathcal{B}} = \frac{1}{3}\arccos\left[\sqrt{6}\left(\frac{B_{\alpha\beta}B^{\beta}{}_{\gamma}B^{\gamma\alpha}}{[B_{\alpha\beta}]^{\frac{3}{2}}}\right)\right]. \tag{4.83b}$$

Notice that all of the E and B terms in Eqs. (4.82) and (4.83) have summations that leave only scalar quantities in these expressions. We use these terms to express the Weyl scalar Ψ_4 in terms of scalar quantities.

4.11 Decomposition of Ψ_4 by Spherical Harmonics

In this chapter the decomposition Ψ_4 by spherical harmonics is discussed, in accordance with the discussion by Walter [28]. Since the gravitational field has spin weights ±2, we will project Ψ_4 on the spherical harmonics of the same spin-weights. More specifically, we will focus on the spherical harmonic with spin-weight $s = -2$. By doing so we are able to calculate the contributions of the individual l, m modes. The scalar product of Ψ_4 and Y^{-2}_{lm} gives

$$A_{lm} = \langle Y^{-2}_{lm}, \Psi_4 \rangle = \int_0^{2\pi} \int_0^{\pi} \Psi_4 \bar{Y}^{-2}_{lm} \sin\theta d\theta d\phi. \tag{4.84}$$

where Y^s_{lm} are the spin-weighted spherical harmonics given by

$$Y^s_{lm}(\theta,\phi) = (-1)^s \sqrt{\frac{2l+1}{4\pi}} d^l_{m(-s)}(\theta) e^{im\phi}. \tag{4.85}$$

Above, $d^l_{ms}(\theta)$ are the Wigner d-functions, which are defined as follows:

$$\begin{aligned} d^l_{ms}(\theta) \quad = \quad & \sum_{t=C_1}^{C_2} \frac{(-1)^t[(l+m)!(l-m)!(l+s)!(l-s)!]^{\frac{1}{2}}}{(l+m-t)!(l-s-t)!t!(t+s-m)!} \\ & \left(\frac{\cos\theta}{2}\right)^{2l+m-s-2t} \left(\frac{\sin\theta}{2}\right)^{2t+s-m}, \end{aligned} \tag{4.86}$$

where $C_1 = \max(0, m-s)$ and $C_2 = \min(l+m, l-s)$.

4.12 Energy, Momentum, and Angular Momentum

The gravitational radiation emitted by a system carries away energy and momentum. This is the mechanism by which orbiting black holes undergo inspiral and, ultimately, merger. This section follows the discussion by Walter [28] and presents the calculation of energy and momentum carried by gravitational radiation. The rate at which energy and momentum are radiated can be written in terms of the Weyl scalar Ψ_4 as follows:

$$\frac{dE}{dt} = \lim_{r\to\infty}\left[\frac{r^2}{16\pi}\int_\Omega \left|\int_{-\infty}^{t}\Psi_4 dt\right|^2 d\Omega\right], \tag{4.87}$$

$$\frac{dP_i}{dt} = -\lim_{r\to\infty}\left[\frac{r^2}{16\pi}\int_\Omega l_i \left|\int_{-\infty}^{t}\Psi_4 dt\right|^2 d\Omega\right], \tag{4.88}$$

$$\frac{dJ_z}{dt} = -\lim_{r\to\infty}\left\{\frac{r^2}{16\pi}\mathrm{Re}\left[\int_\Omega \left(\partial_\phi \int_{-\infty}^{t}\Psi_4 dt\right)\left(\int_{-\infty}^{t}\int_{-\infty}^{t'}\bar{\Psi}_4 dt dt'\right) d\Omega\right]\right\}, \tag{4.89}$$

where $l_i = (-\sin\theta\cos\phi, -\sin\theta\sin\phi, -\cos\phi)$. By expanding Ψ_4 in terms of the l and m modes, the expression for rate of energy radiated is simplified. The energy radiated in the individual modes is found to be:

$$\frac{dE}{dt} = \lim_{r\to\infty}\left[\frac{r^2}{16\pi}\left|\int_{-\infty}^{t}\sum_{l,m} A_{l,m} dt\right|^2\right]. \tag{4.90}$$

Typically, most of the energy is radiated in the $l = 2, m = \pm 2$ modes.

The total amount of energy in a spacetime is given by the ADM Energy, also known as the ADM Mass M_{ADM} [12]. The energy in the spacetime is computed within a sphere S_r of radius r as

$$E(r) = \frac{1}{16\pi}\int_{S_r}\sqrt{g}g^{ij}g^{kl}(g_{ik,j} - g_{ij,k})dS_l. \tag{4.91}$$

To account for the fact that the boundaries of the simulations do not extend out to spatial infinity, the ADM Energy is found by taking the limit of $E(r)$as r goes to infinity, i.e.:

$$M_{ADM} = \lim_{r\to\infty} E(r).$$

Similarly, the linear momentum P_i and angular momentum J_i are also calculated within the sphere of radius r as:

$$\begin{aligned} P_i(r) &= \frac{1}{8\pi}\int_{S_r}\sqrt{g}(K^j{}_i-\delta^j{}_iK)dS_j, && (4.92)\\ J_i(r) &= \frac{1}{8\pi}\varepsilon_{ij}{}^k\int_{S_r}\sqrt{g}x^j(K^m{}_k-\delta^m{}_kK)dS_m. && (4.93)\end{aligned}$$

where the three dimensional Levi-Civita tensor ε^{ijk} is a contraction of the four dimensional Levi-Civita tensor ${}^4\varepsilon^{\alpha\beta\gamma\delta}$:

$$\begin{aligned} \varepsilon^{ijk} &\equiv {}^4\varepsilon^{\alpha\beta\gamma\delta}\,\hat{n}_\delta && (4.94)\\ &= \hat{x}^i\cdot\left(\hat{x}^j\times\hat{x}^k\right)=\sqrt{g}\;[1\,2\,3]^{ijk}, && (4.95)\end{aligned}$$

and $[123]^{ijk}$ is equal to:

$$[123]^{ijk}=\begin{cases} +1 & \text{for}\,(i,j,k)\in\{(1,\,2,\,3)\,,\,(3,\,1,\,2)\,,\,(2,\,3,\,1)\}\,,\\ -1 & \text{for}\,(i,j,k)\in\{(1,\,3,\,2)\,,\,(2,\,1,\,3)\,,\,(3,\,2,\,1)\}\,,\\ 0 & \text{for any other combination.}\end{cases} \tag{4.96}$$

The total linear and angular momentum is then found by taking the limit of $P_i(r)$ and $J_i(r)$, respectively, as $r\to\infty$:

$$\begin{aligned} P_i &= \lim_{r\to\infty}P_i(r),\\ J_i &= \lim_{r\to\infty}J_i(r).\end{aligned}$$

Chapter 5

Gravitational Wave Detection

Although there is strong indirect evidence that gravitational waves exist, a wave has never been detected. Direct detection would unequivocally prove the existence of gravitational waves and would validate Einstein's theory of general relativity. Direct detection is expected to occur within the next few years and will most likely be achieved by large ground based interferometric detectors such as LIGO [1].

5.1 LIGO

The Laser Interferometer Gravitational Wave Observatory (LIGO) is a US-based collection of three physical gravitational wave detectors in two locations: Hanford, Washington and Livingston, Louisiana. Conceptually, each detector is a large Michelson interferometer comprised of two large vacuum tubes at right angles to each other with mirrored test masses suspended at the ends. In reality, each detector is a modified Michelson interferometer with additional test masses used to create Fabry-Perot cavities and introduce power-recycling, both of which increase the sensitivity. However, for simplicity sake, only a basic Michelson interferometer will be described. In a Michelson interferometer, a laser beam is split at the intersection of the two arms, sent down the arms, reflected off the mirrors at the ends, and returned back to the intersection where the beam is re-combined. The arm lengths can be precisely arranged such that when the beams from the two arms re-combine they perfectly interfere. If the length of one of the arms changes relative to the other arm, the beams will no longer be perfectly out of phase when they re-combine and therefore will not perfectly interfere. Without perfect destructive interference the excess light can be detected with a photodetector. A gravitational wave passing through the detector would have the effect of changing the arm lengths, resulting in a signal which would be detected by the photodetector. A schematic illustration of a LIGO interferometer is shown in Fig. 5.1.

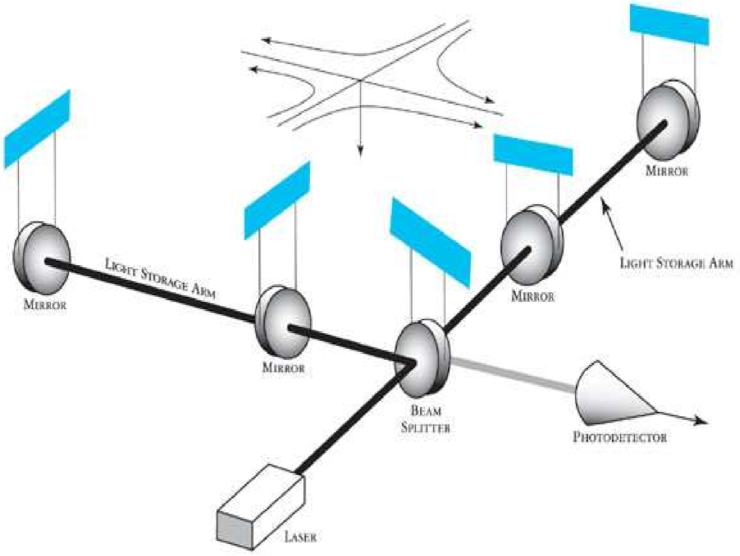


Figure 5.1: Schematic view of a LIGO detector [29].

Using this configuration, the fractional change in arm length, or strain, can be measured with extremely high accuracy. Figure 5.2 shows the strain sensitivity achieved by LIGO for each of the first five science runs. In the fifth science run (S5), completed in 2007, LIGO achieved its design sensitivity with a peak of $3 \times 10^{-23}/\sqrt{\text{Hz}}$ in the $100\text{Hz} - 200\text{Hz}$ frequency range.

Using Eq. (1.2) we determine that this peak sensitivity coincides with the strain that is expected to be produced by two $1 - 2$ solar mass black holes or neutron stars undergoing inspiral and merger at a distance of about 15MPC from Earth. From the number of such objects that are believed to exist within the volume of space out to 15MPC, LIGO expected a detection rate of up to 1/yr. As such, the fact that LIGO has not detected anything yet is not surprising. To have a good chance at detecting a gravitational wave, LIGO's sensitivity has to be improved, which is the goal of Advanced LIGO.

Advanced LIGO [31] is the name given to the next generation of gravitational wave detectors at LIGO facilities. Upgrades to the existing LIGO detectors are currently underway and are expected to be completed by 2014. Once completed, the initial sensitivity of LIGO will be improved by about a factor of 10, increasing the volume of observable space by a factor of 1,000. Consequently, the expected rate of detection will increase substantially. Over the course of a year, Advanced LIGO can expect to see up to 40 neutron star inspiral events, 30 black hole binary mergers, and 10 mixed black hole-neutron star inspirals. Given these numbers, hopes are very high that a direct detection will be made once the upgraded detectors come on-line. Even with the

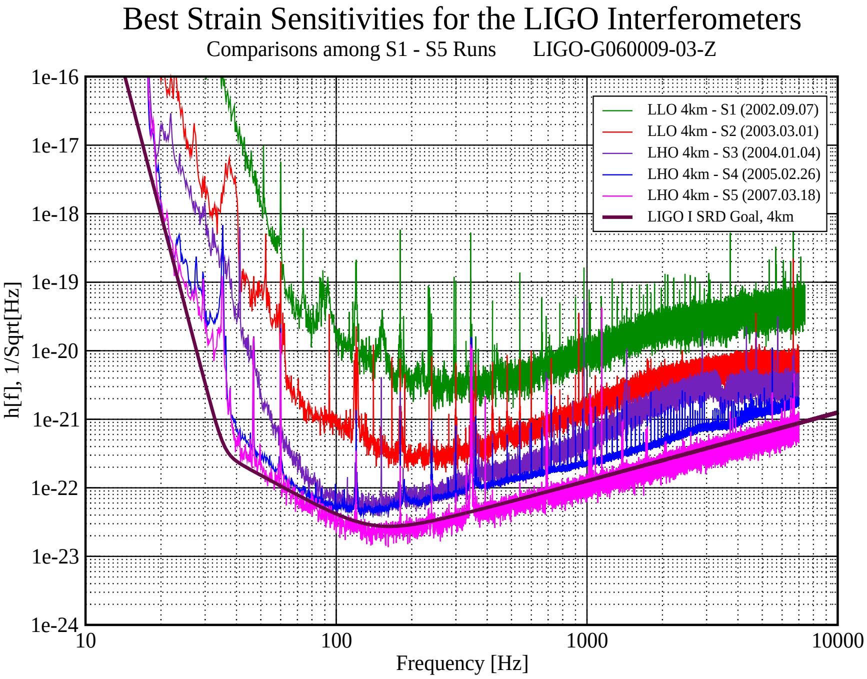


Figure 5.2: LIGO strain sensitivity for science runs S1 - S5 [30].

increased sensitivity, though, the chance of detection is greater still with the use of data analysis techniques such as matched filtering.

5.2 Matched Filtering

The chances of finding a signal in noisy data is greatly increased if we know what signal to expect. Matched filtering is the process of searching for a signal in noisy data by passing the data through a filter $K(t)$, which uses the expected signal as a template. For a detailed treatment of this process, see Fazi [32]. The output Z of the filter $K(t)$ gives a measure of the likelihood that a given signal is present in the data. It is given by

$$Z(t) \equiv \int_{-\infty}^{\infty} K(t)s(t)dt. \tag{5.1}$$

We want to select the filter $K(t)$ such that Z will be large if a signal is present and small if it is not. It can be shown that for signals of known form in colored Gaussian noise this is accomplished by using a filter of the following form:

$$\tilde{K}(f) = C\frac{\tilde{h}(f)}{S_n(f)}, \tag{5.2}$$

where $\tilde{h}(f)$ is the Fourier transform of the signal and $S_n(f)$ is the power spectrum.

Starting from the initial equation for the filter output, Eq. (5.1), and allowing the signal to occur at an unknown time t, the filter output is then a function of t and can be expressed as a convolution as follows:

$$Z(t) = C\int_{-\infty}^{\infty} K(t-t')s(t')dt', \tag{5.3}$$

where C is a constant determined in the derivation of the filter. Using the convolution theorem this can be expressed as

$$Z(t) = C\int_{-\infty}^{\infty} \tilde{s}(f)\tilde{K}^*(f)e^{2\pi ift}df, \tag{5.4}$$

where * denotes a complex conjugate. By substituting the expression for the optimal filter, Eq. (5.2), we get

$$Z(t) = C\int_{-\infty}^{\infty} \frac{\tilde{s}(f)\tilde{h}^*(f)e^{2\pi ift}}{S_n(|f|)}df. \tag{5.5}$$

A threshold for a signal could be put directly on Z, but instead Z is first normalized by the variance of the filter, σ^2, given by:

$$\sigma^2 = 2\int_{-\infty}^{\infty} \frac{\tilde{h}(f)\tilde{h}^*(f)}{S_n(|f|)}df, \tag{5.6}$$

where the constant C above is set to 2. The signal to noise ratio (SNR) of the normalized output of the optimal filter is defined by the quantity $\rho(t)$, given as follows:

$$\rho(t) = \frac{|Z(t)|}{\sigma}. \tag{5.7}$$

A threshold value, ρ_0, is then chosen which determines whether or not a signal is present, i.e.:

$$\rho \geq \rho_0 \rightarrow \text{signal is present}$$

$$\rho < \rho_0 \rightarrow \text{signal is not present}$$

With this procedure there is the possibility of false alarm or false dismissal, i.e., when $\rho \geq \rho_0$ and no signal is present, or when $\rho < \rho_0$ but a signal is actually present, respectively. Thus, the threshold ρ_0 must be chosen carefully. To help determine this threshold and to test the entire data analysis pipeline, simulated waveforms are injected into real data to see if they can be detected. This process is discussed in the following section.

5.3 NINJA

The Numerical INJection Analysis (NINJA) project [33] is a collaboration between the numerical relativity and gravitational wave data analysis communities. The purpose of NINJA is to test the existing gravitational wave data analysis pipeline using simulated numerically generated waveforms inserted into real detector data. This provides a test of whether a signal will be detected with the current data analysis procedure and using the existing gravitational waveform template bank. These tests can also be used to help determine the threshold SNR that constitutes a detection, as described in Section 5.2.

The parameter space needed to cover every possible binary black hole interaction is huge and completely infeasible to simulate directly. Varying parameters such as spin, mass ratio, and momentum all produce different interactions and, hence, different gravitational waveforms. Since numerical relativity simulations are very computationally expensive, this parameter space cannot be investigated simply by brute force. Instead, detailed analysis is done to determine if a signal will be detected using the current template bank or if new templates are needed. For example, when looking at the effect of spin on the template bank, Frei [34] found that a surprisingly high number of signals with spin were detected using the template bank that did not include spin. However, the parameters of the holes (spin, mass ratio, etc.) were recovered much more accurately using templates that did include spin. Using this type of analysis, the effectiveness of the existing template bank can be evaluated and new templates can be generated if necessary.

If it is determined that a new template is needed, new templates can be created using interpolation. In most cases gravitational waveforms vary in a regular manner in response to changing a parameter. This allows the use of interpolation between existing waveforms without loss of accuracy, avoiding the need to run lengthy full numerical relativity simulations. If a full simulation is necessary, a request can be made to the numerical relativity community to produce the waveforms needed.

Chapter 6

Computational Framework

There are three main elements of the computational framework upon which *openGR* is built: the *SAMRAI* library[8] and the two packages *PVODE* and *KINSol*, which are distributed as part of the SUNDIALS suite. *SAMRAI* provides a framework for adaptive mesh refinement while *PVODE* and *KINSol* are used to solve the hyperbolic (time evolution) and elliptic (constraints) equations, respectively. *SAMRAI* comes with some C++ wrappers for the otherwise independent PVODE and KINSol packages so that they are fully embedded in the *SAMRAI* framework. *openGR* is an additional piece of C++ code that provides the physics associated with the problem we want to solve. Given the modular design of *openGR* one can solve differing sets of evolution or constraint equations simply by setting the number of variables to solve and providing the right-hand-sides of the evolution and constraint equations.

6.1 The *SAMRAI* Library

The *SAMRAI* C++ library provides an adaptive mesh refinement (AMR) framework that can be used for numerical relativity simulations. The computational grid is implemented as a collection of structured grid components. The adaptive mesh refinement structure is a hierarchy of levels with different resolutions nested within the *SAMRAI* framework (Fig. 6.1). Each level is divided in a series of rectangular patches that are assigned to the different processors used in the numerical simulation.

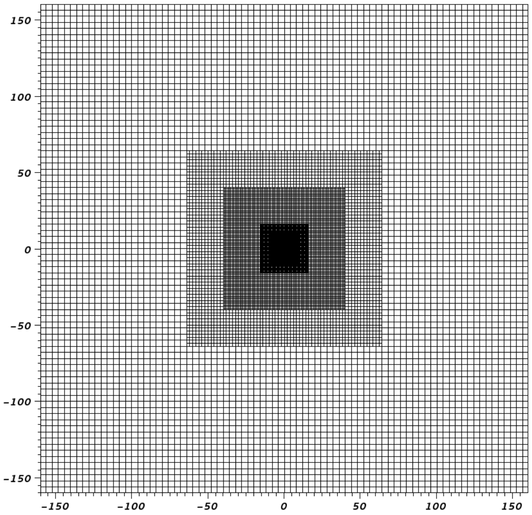


Figure 6.1: Example of a grid with mesh refinement. Finer level grids are nested within coarser levels.

The intrinsic C++ design of the *SAMRAI* library provides significant flexibility by allowing the user to treat some features as a "black box" or to make modifications as necessary. All the fundamental features of *SAMRAI* are defined as object oriented classes whose key functions are declared as virtual and can be inherited and overwritten by the user.

A specific example demonstrating *SAMRAI*'s flexibility is the refinement operation between different levels to fill the ghost zones of a finer level using interpolation from the coarser one. *SAMRAI* comes with a simple linear interpolation algorithm as the default. In *openGR*, we need to utilize higher order interpolation since the finite differences are calculated at fourth order in space. The flexibility of *SAMRAI* allows the user to inherit the class `RefineOperator` and overwrite the function refine, introducing higher order stencils for interpolation.

Just like the `RefineOperator` class, all the key ingredients of the AMR framework can be inherited and modified. For example, *SAMRAI* and *openGR* currently use uniform load balancing to share the workload among processors, but there are plans to modify this feature to introduce a more adapted and optimized load balancing scheme that should improve the code efficiency. *SAMRAI* allows such modifications to be made in a straightforward way, making the code improvements easy to design and apply. Moreover, the main *SAMRAI* library only needs to be compiled once, even if we plan to modify some key features of the basic library. Using this C++ capability of inheriting an existing class to modify the functions we want, the new functions are present in our code and re-compiling the basic library is not necessary.

6.2 *PVODE*

Einstein's equations generate a set of hyperbolic evolution equations. To solve these evolution equations we use a mixed Adams-Bashforth Adams-Moulton method implemented by *PVODE*. *PVODE* is a general purpose solver for systems of ordinary differential equations (ODEs). It allows the use of a mixed Adams-Moulton Adams-Bashforth method. This is often referred to as a “predictor-corrector”: an Adams-Bashforth method is used to calculate the solution y at timestep n (predictor), then an Adams-Moulton method is used to improve the value (corrector). *PVODE* is used to solve the initial value problem

$$\dot{y} = f(t, y), \qquad y(t_0) = y_0. \tag{6.1}$$

The Adams-Bashforth Adams-Moulton predictor-corrector method is developed by integrating Eq. (6.1) on both sides and replacing the integral with a quadrature formula:

$$y(t) - y(t_0) = \int_{t_0}^{t} f(\tau, y(\tau))\,\mathrm{d}\tau, \tag{6.2}$$

$$y(t) \approx y(t_i) + \sum_{j=0}^{k} A_j f(t_j, y(t_j)), \tag{6.3}$$

where $t_i \leq t0 < t1 < \cdots < t_k \leq t$, and A_j are the appropriate quadrature coefficients.

In general the initial value problem of Eq. (6.1) can be solved using a linear multistep method of the form [35]

$$\sum_{i=0}^{K_1} \alpha_{n,i} y_{n-i} + \Delta t \sum_{i=0}^{K_2} \beta_{n,i} \dot{y}_{n-1} = 0, \tag{6.4}$$

where $\Delta t = t_n - t_{n-1}$ is the timestep and y_n is the computed approximation of $y(t_n)$, the exact solution of Eq. (6.1) at t_n. The coefficients $\alpha_{n,i}$ and $\beta_{n,i}$ are uniquely determined by a particular integration formula, the history of the size of the timestep, and the normalization $\alpha_{n,0} = -1$. Setting $K_1 = 1$ and

$K_2 = q - 1$ yields the general Adams-Moulton Formula. The variable q is the order of integration.

The Adams-Bashforth Adams-Moulton predictor-corrector method is a combination of the Adams-Bashforth method and the Adams-Moulton method. The Adams-Bashforth method is an implicit method obtained by setting $\beta_q = 0$ in the Adams-Moulton formula. It can be written in the form

$$y_n = y_{n-1} + \Delta t \sum_{i=0}^{q-1} \beta_i f(t_{n-1}, y_{n-i}). \tag{6.5}$$

By fixing q the order of the method is determined and the coefficients β_i can be determined using the Lagrange formula for polynomial interpolation:

$$\beta_{q-i-1} = \frac{(-1)^i}{i!(q-i-1)!} \int_0^1 \prod_{i=0}^{q-1} (u+i)\mathrm{d}u, \tag{6.6}$$

where $i = 0 \ldots q-1$. For example, a fourth order Adams-Bashforth method equation has coefficients $\beta_i = [-\frac{3}{3}, \frac{37}{24}, -\frac{59}{24}, \frac{55}{24}]$. The procedure for calculating coefficients for the Adams-Moulton method is similar and also uses the Lagrange formula for polynomial interpolation.

Setting $\beta_q = 0$ to obtain the Adams-Bashforth method implies that this method is implicit, while the Adams-Moulton method is explicit. This also implies that a q-step Adams-Moulton method is of order $q + 1$ while a q-step Adams-Bashforth scheme is of order q.

The Adams-Moulton method is generally more accurate than Adams-Bashforth method due to the fact that it is implicit. However, it also proves to be more computationally expensive since solving an implicit problem requires more advanced numerical techniques. Since the Adams-Bashforth part of the predictor-corrector calculates the solution at timestep n, the Adams-Moulton part is no longer implicit and can be solved using functional iteration until a specified convergence tolerance is met.

PVODE enables the use of the Adams methods up to twelfth order in time. As noted above, a q order Adams-Bashforth method requires q timesteps, but when the simulation starts only one timestep is available. This problem is solved by varying the order of the method and the size of the timesteps. Initially the simulation uses first order Adams-Bashforth and very small timesteps. The order and the size of the timestep are slowly increased until the desired integration order is reached.

Compared to the standard Runge-Kutta scheme used for time integration in most other numerical relativity codes, the Adams-Moulton Adams-Bashforth method used by *openGR* proves to be more accurate and stable due to the implicit nature of the Adams-Moulton method. However, this implicit nature also makes it more computationally demanding in both memory allocation and efficiency. The need to keep information of twelve previous timesteps significantly increases the memory storage of the program. Moreover, we are unable to use AMR technologies like Berger Oliger because of the variability of the timestep. All of the AMR levels must evolve using the same timestep, which results in a less efficient code. We made this tradeoff of efficiency in exchange for more stability and accuracy.

6.3 *KINSol*

The ADM decomposition of Einstein's equations introduces a set of elliptic equations, the constraint equations, defined on each spatial hypersurface. These constraint equations must be solved in order to obtain well defined initial data that are solutions of Einstein's equations. Moreover, these constraint equations can be used to perform constrained evolution, which is the practice of periodically correcting the evolved variables. The constraint equations are a set of non-linear algebraic equations which can be solved using Newton's method. The problem can be stated as follows: given n unknowns with n real valued functions,

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} \tag{6.7}$$

where $x = (x_1, x_2, \ldots, x_n)$, a vector $s = (s_1, s_2, \ldots, s_n)$ is sought such that

$$\mathbf{F}(\mathbf{s}) = \mathbf{0}. \tag{6.8}$$

An inexact Newton method is applied to find the solution of Eq. (6.8), using the following iteration scheme:

1. Set x_0 as initial guess
2. For $n = 0, 1, 2 \ldots$ until convergence do:
    (a) Solve $\mathbf{J}(\mathbf{x_n})\delta\mathbf{n} = -\mathbf{F}(\mathbf{x_n})$
    (b) Set $\mathbf{x_{n+1}} = \mathbf{x_n} + \delta\mathbf{n}$

(c) Test for convergence

where $\mathbf{J}(\mathbf{x_n})$ is the system Jacobian. A Krylov method is used to solve the equation in step 2, requiring only the matrix vector product $J(u)v$, which is approximated by

$$\mathbf{J}(\mathbf{x_n})\mathbf{v_n} \approx \frac{\mathbf{F}(\mathbf{x_n} + \sigma\mathbf{v_n}) - \mathbf{F}(\mathbf{x_n})}{\sigma}, \tag{6.9}$$

where $\mathbf{x_n}$ is the current approximation and σ is a scalar chosen to minimize the numerical error.

The particular elliptic solver used by KINSol is based on the Krylov procedure for systems of non-linear equations. The method starts with an initial guess x_0 for the system, and in the following k iterations produces an approximate solution x_k from a Krylov space generated by a vector b,

$$\kappa(A, b) = \text{span}\{b, Ab, \ldots, A^{k-1}b\} \tag{6.10}$$

KINSol uses the Generalized Miniminum RESidual method (GMRES) to obtain the solution for a linear system of equations

$$Ax = b, \tag{6.11}$$

(which, in our case, is represented by the equation $\mathbf{J}(\mathbf{x_n})\delta\mathbf{n} = -\mathbf{F}(\mathbf{x_n})$).

After an initial solution x_0 is chosen, the residual $r_0 = b - Ax_0$ is computed. The Krylov space is then generated using the Arnoldi method, which basically consists of a Gram-Schmidt orthonormalization of the basis vectors.

The first step in generating the Krylov space is to compute a normalized version of the residual r_0 given by $v1 = r_0/\|r_0\|$. The iteration will then proceed through the following steps:

- Compute the coefficients $h_{i,j} = (Av_j v_i)$
- Add one orthonormal vector $v_{j+1} = Av_j \sum_{i=1}^{j} h_{i,j} v_i$ to the Krylov series,
- Form an approximate solution $x_k = x_{k-1} + V_k y_k$ and then repeat iteration until convergence is met.

The procedure implemented in the KINSol package describes what is done in a single refinement level. To adapt and optimize the procedure for a multi-level simulation, we adopt the following strategy:

1. Solve the elliptic system of equations on the coarsest level using Robin boundary conditions until tolerance is reached.
2. A prolongation operator (see the next section for the description of restriction and prolongation) is used to interpolate the solution on the next finer level in the grid, including the ghost zones.
3. Solve the finer level and go back to step 2 until the finest level is solved.

6.4 Restriction and Prolongation

The presence of a multi-resolution grid introduces the issue of communicating information between different levels of resolution. To accomplish this we need to two operators: (1) a restriction operator I_h^{2h} that copies data from a finer to a coarser level and (2) a prolongation operator I_{2h}^h that performs the opposite procedure. In particular, we need restriction at the end of each *PVODE* iteration to copy data from the more accurate finer levels to the coarser levels. Since we are using node-centered data the fine data lies directly on top of the coarse data so the fine data can be directly copied to the coarse level and no interpolation is necessary. When solving the initial data we start by solving the coarsest level and then prolongation and interpolation are used to fill in the domain of the next finer level. During time evolution, prolongation is only necessary to fill the ghost zones of each level (except the coarsest one).

SAMRAI handles the geometry of the system once the position of each level is defined within the hierarchy of the grid. The prolongation and restriction performed by *SAMRAI* is performed using only a first order interpolation operator. However, this operator is defined in the context of a C++ object (`RefineAlgorithm`) that the user can inherit and modify. This is exactly what we do in *openGR* by using the two classes `RefineAlgorithm` and `CoarsenAlgorithm`, in which we inherit the basic *SAMRAI* C++ class and define alternative operators that perform interpolation up to 6th order. The order of the interpolation is specified in the parameter file.

6.5 Outer Boundary Conditions

Since we are modeling an extended physical space with a limited computational domain, we need to enforce boundary conditions at the outer boundary. We use different boundary conditions during the time evolution and the constraint solver. These boundary conditions are applied to the coarsest level in the AMR grid, as that is the only level that extends all the way to the outer boundary.

6.5.1 Sommerfeld Boundary Conditions

For the time evolution we implement Sommerfeld radiative boundary conditions. These allow outward propagating waves to leave the domain, avoiding back-reflection which would introduce errors into the spacetime. Each variable u is evolved using the equation

$$\frac{\partial u}{\partial t} = -\frac{u - u_0}{r} + n^i \partial_i u, \tag{6.12}$$

where n^i is the normal to the sphere $r = const$ given by $n^i = -[\frac{x}{r}, \frac{y}{r}, \frac{z}{r}]$ and u_0 is the value of the field in the flat limit. We assume $u_0 = 0$ for all the fields except g_{xx}, g_{yy} and g_{zz}.

6.5.2 Robin Boundary Conditions

For the constraint solver we instead use Robin boundary conditions. These assume that the variables fall off as some power of r such that

$$\frac{\partial}{\partial r}[r^n (u - u_0)] = 0, \tag{6.13}$$

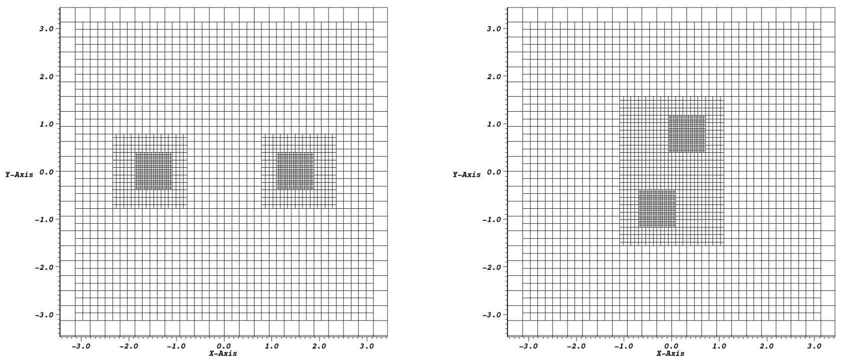
where n is the power of the radial fall-off of a given variable. This leads to the following conditions:

$$u_{i,j,k} = 1 + (u_{i-1,j,k} - 1)\frac{r_{i-1,j,k}}{r_{i,j,k}}. \tag{6.14}$$

6.6 Adaptive Mesh Refinement

openGR is a fully AMR-enabled code that allows all the refinement levels to be moved during the simulation. The user can decide which refinement levels are fixed and which are free to move. In general, different refinement levels can be very nonuniform and change over time. *SAMRAI* allows for a very general definition of where the refinement should take place by means of the function `ApplyGradientDetector`. However, it is too computationally expensive to dynamically determine where refinement should occur using gradients or other such measures. Instead we employ a straightforward "moving boxes" method in which the uniform nested boxes grid as shown in Fig. (6.1) is maintained over much of the domain and only the finest few levels are free to move to track the positions of the holes (Fig. (6.2)). The shape of these adaptive refinement levels remains constant unless they move close enough together to merge into a single box (Fig. (6.2(b))). The boxes track the positions of the holes by integrating in time the shift vector β^i, since in the puncture approach the velocity of the holes satisfies the following equation

$$\frac{\partial x^i}{\partial t} = -\beta^i(x^i). \tag{6.15}$$



(a) (b)

Figure 6.2: Zoomed in view of the grid in the region of the holes. The finest two levels track the positions of the holes. In Fig. 6.2(b) the coarser of the two levels moved close enough together to be combined into one box by *SAMRAI*.

Eq. (6.15) is then integrated using a second order Runge-Kutta time integrator. Once the position of the two black holes is known at a time step, *SAMRAI* re-arranges the grid according to this position, and makes use of the prolongation and restriction operator to update the solution on the new grid structure.

Chapter 7

Scaling, Performance, and Convergence

This chapter will investigate matters of performance and accuracy. We want to understand how well *openGR* performs as we change the number of points in a grid and the number of processors used to carry out a simulation. The following discussion looks at two different types of scaling (strong and weak) to get a sense of how *openGR* scales as well as how the network overhead increases with larger jobs. Memory use is also outlined as it is typically the limiting factor on large simulations. Actual performance is also briefly discussed to provide a sense of how long simulations take to run. Finally, the accuracy of the code is demonstrated by running convergence tests.

7.1 Scaling

Scaling describes how much faster the code runs as the number of processors is increased. We can relate the time it takes to run a simulation (wall-clock time) to the number of processors by

$$t = an^{b}, \tag{7.1}$$

where t is the wall-clock time, n is the number of processors, and a and b are parameters. Taking the log of both sides yields

$$\log t = \log a + b \log n. \tag{7.2}$$

Here we have a clear linear relationship on a log-log plot, where $\log a$ is the intercept and b is the slope. Comparing two simulations using Eq. (7.1) we find:

$$\frac{t_2}{t_1} = \left(\frac{n_2}{n_1}\right)^b. \tag{7.3}$$

To measure the scaling properties of *openGR* we set up a series of jobs detailed in Table 7.1. Since a majority of a simulation is spent doing the evolution, we will only consider the time spent in *PVODE* (used for the evolution in *openGR*). All scaling runs were performed on TACC supercomputer Ranger.

Job	A	B	C	D
Points / Level	40^3	80^3	160^3	320^3
Refinement Levels	9	9	9	9
Domain	$\pm 100M$	$\pm 200M$	$\pm 400M$	$\pm 800M$
Coarse Resolution	$5M$	$5M$	$5M$	$5M$
Fine Resolution	$\frac{5M}{256}$	$\frac{5M}{256}$	$\frac{5M}{256}$	$\frac{5M}{256}$

Table 7.1: Configurations used for scaling runs. In all cases the hole is located at the origin and has mass M. The domain is increased proportionally to the number of points per level to keep the spatial resolution consistent for each job.

As will be seen below, we only considered the case of a single puncture. Having a second puncture only changes the finest two levels of the grid, and looking at the results obtained by Walter [28] we see that no significant difference is found in the scaling results for two punctures as compared to a single puncture. Although actual simulations use AMR, for the purposes of scaling we will also only consider fixed mesh refinement (FMR). The AMR routine is a trivial calculation and introduces very little extra computational overhead. In addition, for time considerations we will not evolve the system long enough for the hole to move an appreciable distance so we can restrict our configurations to FMR without any significant loss of validity.

7.1.1 Strong Scaling

Strong scaling is a measure of the speed-up time achieved by running the same job on a larger number of processors. Ideally, if we double the number of processors used, the wall-clock time should be cut in half. From Eq. (7.3) we can see that this would correspond to a slope of -1 on a log-log plot, giving $t \propto \frac{1}{n}$. In practice, this does not actually happen because of the overhead required for interprocessor communication. The number of points that each individual processor works on is padded by a few points in each direction, known as the ghostzones. As the number of processors is increased, the portion of the job that each processor performs is decreased, but the communication overhead per processor stays the same. Eventually, the size of the computational domain that a processor is handling approaches the size of the ghostzone cells and more time is spent performing communication overhead than is spent actually doing calculations. At this point there is no further performance gained by increasing the number of processors. This would correspond to a slope of 0 on a log-log plot. If you then continue to increase the number of processors the time spent communicating between processors will become greater than the time spent doing calculations and the performance is worse than it was was with fewer processors. This corresponds to a positive slop on a log-log plot.

7.1.2 Single Puncture Unigrid Strong Scaling

To perform a unigrid job the same configurations listed in Table 7.1 are used but with only a single level, i.e. Job A consists of a single grid containing 40^3 points, etc. Comparing these unigrid results to those obtained with FMR allow us to understand how much performance is being affected by having multiple refinement levels. Strong scaling results for jobs run on a unigrid domain are shown in Fig. 7.1. All jobs were run out to a time of $5M$. A summary of results is shown in Table 7.2.

We can see that for all unigrid jobs the slope is relatively close to the average of -0.54, however they are also rather far from the ideal slope of -1. We conclude that for a unigrid domain the scaling is very consistent, although less than ideal.

Strong Scaling: Single Puncture Unigrid				
Job	A	B	C	D
Processor Range	16 - 128	16 - 128	16 - 512	128-512
Slope	-0.48	-0.50	-0.58	-0.61
Average Slope	-0.54			

Table 7.2: Summary of strong scaling results for a unigrid single puncture.

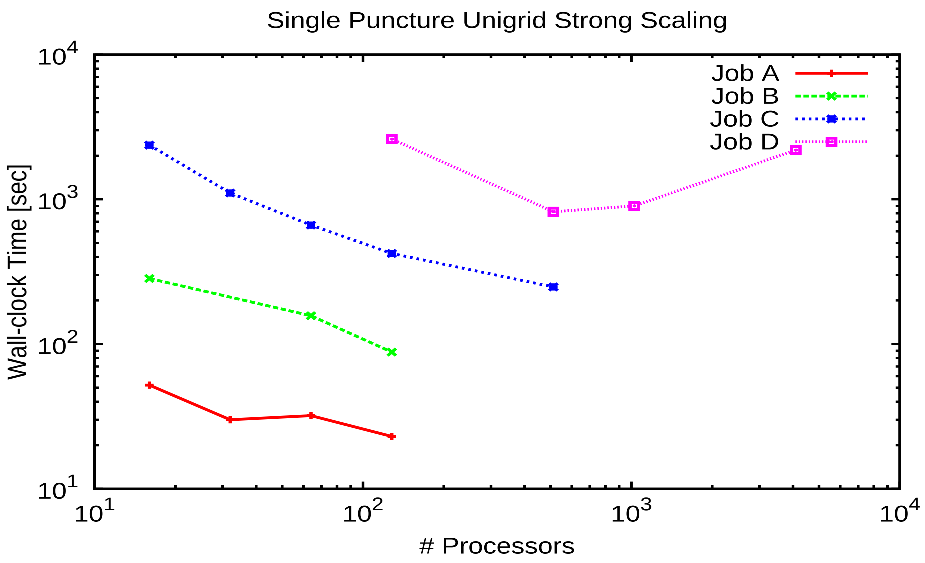


Figure 7.1: Strong scaling for a single puncture in a unigrid domain. All jobs were run out to a time of $5M$.

7.1.3 Single Puncture FMR Strong Scaling

Scaling results from full FMR simulation more closely resemble a typical configuration used in production runs. Strong scaling results for a single puncture in a nine level FMR grid are shown in Fig. 7.2

For a single puncture with nine levels of FMR we find varying scaling performance depending on the job being run. Curve A has a slope of -0.34 over a processor range of 16-64, while curve D has a slope of -0.83 over a processor range of 1024 - 4096. The average of the slope of all four curves is -0.59. We conclude that for nine levels of FMR, *openGR* scales better as the number of points is increased. Having a large number of points ensures that

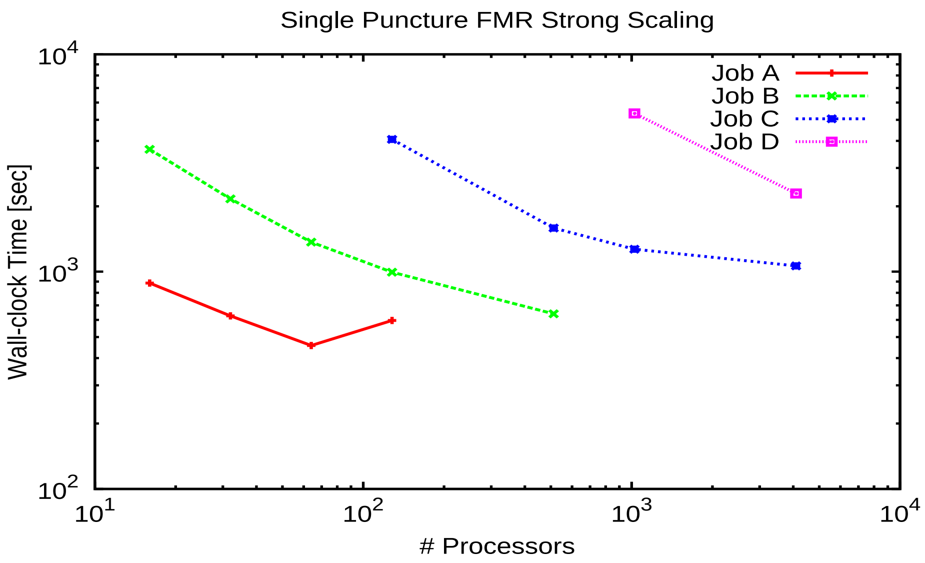


Figure 7.2: Strong scaling for a single puncture with nine levels of FMR.

the time spent doing calculations is significantly longer than the time spent on communication between processors.

Strong Scaling: Single Puncture FMR				
Job	A	B	C	D
Processor Range Slope	16 - 64 -0.34	16 - 512 -0.54	128 - 1024 -0.64	1024-4096 -0.83
Average Slope	-0.59			

Table 7.3: Summary of strong scaling results for single puncture with nine levels of FMR.

7.1.4 Weak Scaling

Weak scaling provides a measure of how much performance is affected by interprocessor communication. This is accomplished by increasing both the overall size of a job and the number of processors such that the number of points per processor is unchanged. The domain is also extended so that the spatial resolution of each job is identical. For example, Job B has double the number of points in each dimension as Job A, and thus has 8× as many total points. If Job B were run on 8× as many processors as Job A, the time it takes to run both of these Jobs should be equal, corresponding to a slope of 0 on a log-log plot. Of course in practice this does not happen and this is directly attributable to the overhead required for the communication between processors. Looking at the deviation from ideal allows us to put limits on the size of job that can be run.

7.1.5 Single Puncture Unigrid Weak Scaling

We again start by looking at the unigrid case. Weak scaling results for a single puncture run on a uniform grid are shown in Fig. 7.3. The label of each line denotes the job and number of processors of the first data point on that line. Each subsequent point on the line corresponds to a job with 8× as many points per level and 8× the number of processors. For example, the first point on $A16$ is job A run on 16 processors. The second data point is Job B run on 128 processors, followed by Job C run on 1024 processors.

Similar to the strong scaling results for the unigrid domain, the weak scaling results are very consistent through a processor range of 16 - 2048 processors. The average slope of all the curves is 0.26. This is less than ideal scaling (which would be a slope of 0), but is reasonable.

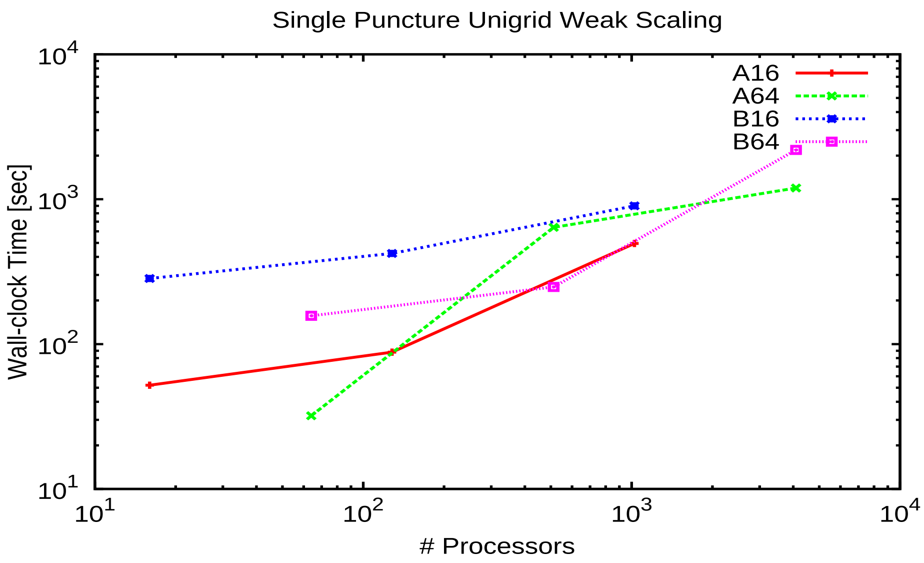


Figure 7.3: Weak scaling for a single puncture in a unigrid domain. The label of each line denotes the job run and the number of processors of the leftmost data point on the line. Each subsequent data point on a line represents an increase in the number of points per level by a factor of 8 and an 8× increase in the number of processors. For example, the first point on line *A*16 is Job A run on 16 processors. The second point is Job B run on 128 processors, followed by Job C run on 1024 processors.

Weak Scaling: Single Puncture Unigrid				
Job	A16	A64	B16	B64
Processor Range	16 - 128	512 - 2048	16 - 1024	64-512
Slope	0.25	0.30	0.28	0.22
Average Slope	0.26			

Table 7.4: Summary of weak scaling results for a unigrid single puncture.

7.1.6 Single Puncture FMR Weak Scaling

As noted before, the full FMR configuration is a more realistic representation of a typical configuration used for full-scale runs. Fig. 7.4 shows weak scaling results for a single puncture with nine levels of FMR.

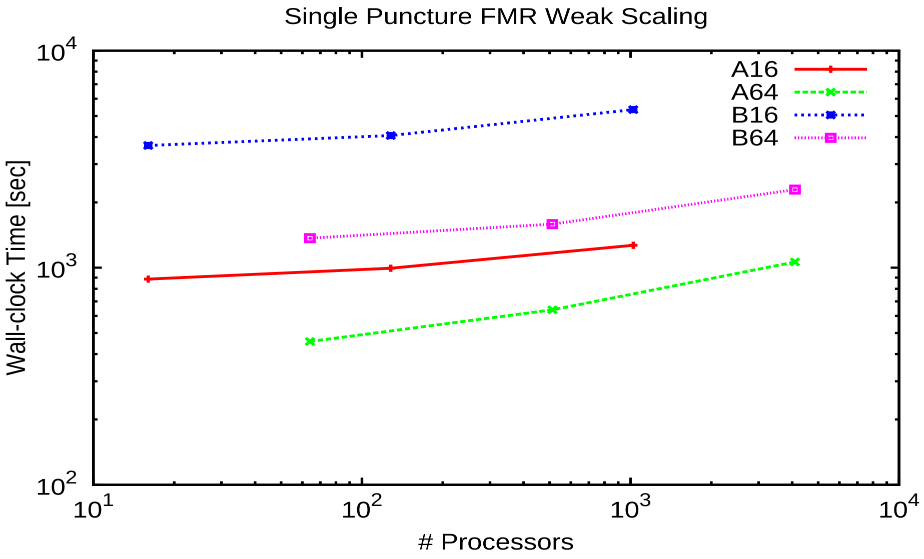


Figure 7.4: Weak scaling for a single puncture with nine levels of FMR. The label of each line denotes the job run and the number of processors of the leftmost data point on the line. Each subsequent data point on a line represents an increase in the number of points per level by a factor of 8 and an 8× increase in the number of processors. For example, the first point on line $A16$ is Job A run on 16 processors. The second point is Job B run on 128 processors, followed by Job C run on 1024 processors.

Weak scaling for a single puncture with nine levels of FMR is very good over a large processor range, from 16 to 4096 processors. The average slope of the four curves over this range is 0.13. Again this is larger than the ideal of 0, but this shows that communication between processors does not hurt performance very much.

Given the results shown in Sections 7.1.2 - 7.1.6 we conclude that in-

Weak Scaling: Single Puncture FMR				
Job	A16	A64	B16	B64
Processor Range	16 - 1024	64 - 4096	16 - 1024	64 - 4096
Slope	0.09	0.20	0.09	0.12
Average Slope	0.13			

Table 7.5: Summary of weak scaling results for single puncture with nine levels of FMR.

terprocessor communication is not a significant bottleneck and that *openGR* scales well provided an appropriate number of processors is chosen for a particular grid layout.

7.2 Memory Use

Memory is the primary bottleneck when running large jobs on *openGR*, even when using large supercomputers. All results obtained in this work were run on the Ranger supercomputer at the Texas Advanced Computing Center (TACC), which has 2GB of memory per processing core. The limits on job size are demonstrated by revisiting the scaling runs presented in section 7.1.

When running on multiple processors, *SAMRAI* divides the grid into patches to distribute the computation among processors. Currently *openGR* uses a uniform load balancing strategy in which *SAMRAI* divides each level into a number of patches equal to the number of processors being used. If possible, *SAMRAI* creates patches that are all the same size on any given level so that every processor has the same number of points to handle. This is not always possible, and when it is infeasible *SAMRAI* divides the grid up as evenly as possible. Each patch is also padded with "ghostzones," which are the points used for interprocessor communication. These ghostzones will not be taken into account when computing the limits on a job size but will be discussed in section 7.2.1 below.

When trying to determine job size limits, we look at runs that approach the memory capacity. Table 7.6 details several runs for Job C and Job D for the Unigrid domain outlined in Table 7.1. Job D was unable to run on 64 processors, with 80^3 points allocated per processor (excluding ghostzones). However, Job D was able to run on 128 processors, with 63.5^3 points allocated per processor. Job C, with the same number of points per processor, successfully ran on 16 processors. Note that the number of points on a processor is obviously always a whole number, but the decimal number comes from the fact that for *SAMRAI* to achieve optimal load balancing the patches will not all be a perfect cube and so the number of points on each processor is not exactly identical. From the numbers depicted in Table 7.6 we conclude that jobs can be run with a maximum number of points per processor in the range of $65^3 - 75^3$.

Single Puncture Unigrid				
Job	# Points	# Processors	$\frac{\#\text{Points}}{\text{Processor}}$	Result
C	4,096,000	16	63.5^3	Successfully Ran
D	32,768,000	64	80^3	Failed
D	32,768,000	128	63.5^3	Successfully Ran

Table 7.6: Details of successful and unsuccessful runs for single puncture unigrid Jobs C and D.

A similar analysis to that performed above was done for single puncture FMR Job C and Job D. Table 7.7 details these runs. Job C with 9 levels of FMR was unable to run on 64 processors. Job D run on 512 processors also failed. Both of these configurations would require 83.2^3 points per processor, which was already shown to be too large in the unigrid case. Job C was able to run on 128 processors, as was Job D on 1024 processors. These configurations both had 66^3 points allocated per processor. These numbers are consistent with those obtained in the unigrid case, and we can now conclude that the largest job we can run is somewhere in the range of about $70^3 - 75^3$ points per processor.

Single Puncture FMR				
Job	# Points	# Processors	$\frac{\#\text{Points}}{\text{Processor}}$	Result
C	36,864,000	64	83.2^3	Failed
C	36,864,000	128	66^3	Successfully Ran
D	294,912,000	512	83.2^3	Failed
D	294,912,000	1024	66^3	Successfully Ran

Table 7.7: Details of successful and unsuccessful runs for single puncture FMR Jobs C and D.

7.2.1 Ghostzones

Ghostzones are the regions around each patch that are used for communication between processors. Each patch is padded on all sides by a slab which is 4 points thick. For example, a cubic patch of 40^3 points has ghostzones with $40^2 \times 4$ points on all six sides. Table 7.8 details the number of ghostzone points used in Job D with a single puncture in a unigrid domain.

Ghostzones for Single Puncture Unigrid Job D			
# Processors	$\frac{\#\text{Points}}{\text{Processor}}$	$\frac{\#\text{Ghostzone Points}}{\text{Patch}}$	$\frac{\#\text{Ghostzone Points}}{\#\text{Compute Points}}$
128	63.5^3	45.9^3	0.38
512	40^3	33.7^3	0.60
1024	31.7^3	28.8^3	0.75
4096	20^3	21.3^3	1.21

Table 7.8: Details of the ghostzones used in Job D for a single puncture in a unigrid domain. Although the ghostzone points are not arranged in a cube, they are expressed as a cube to get a sense of the size relative to the number of compute points. Note: Only the jobs run on 512 and 4096 processors have a layout in which all patches are perfect cubes.

Table 7.8 demonstrates that when Job D is run on 128 processors on a unigrid domain, every patch has just under 40% as many ghostzone points as the number of points used for computation. Obviously this is a larger percentage than we want, but we will have to settle for this until *openGR* is run on a computer with more memory than Ranger. As the number of processors is increased this ratio gets larger, and for 4096 processors there are actually more ghostzone points than there are compute points. As the ratio of ghostzone points to compute points gets larger, the scaling performance clearly gets worse, as can be seen in Fig. 7.1. It is clear that any performance gained by increasing the number of processors is lost once the number of ghostzone points approaches the number of compute points, which is seen as a leveling out or upward turn of the curves in Figs. 7.1 and 7.2.

Obviously we would like to minimize the ratio of ghostzone points to compute points. This is achieved by making the size of a patch (or number of points per processor) as large as possible. Unfortunately, due to the memory limitations outlined in section 7.2, Job D run on 128 processors is almost as large of a patch size as can be used (65 points/processor), which means the ratio of ghostzone points to compute points will not be much lower than 0.38.

Although the current state regarding memory use is less than ideal, there is a potential solution that will enable the use of more memory per processor. Ranger is divided up into nodes that have 16 processors and 32GB of memory (hence, 2GB of memory per processor). We can get more memory per processor by using less than 16 processors per node. For example, if we use only 8 processors per node, those 8 processors have access to all 32GB of memory on the node, so now each processor has 4GB of memory available to it. This allows the creation of patches with more points, thus reducing the ratio of ghostzone points to compute points. We are currently testing the performance improvement by using this type of configuration.

7.3 Performance

To get a sense of how long computations actually take we refer to Fig. 7.2. Job B is a typical configuration which would be used for long production runs. Job A is somewhat coarse, which allows for faster evolution, but is also less accurate. For high accuracy runs a configuration such as Job C is most appropriate. Job D is generally higher resolution than we would use with any regularity, unless we specifically wanted to look at certain features with extremely high accuracy. To look at performance we will consider only the optimal configurations of Jobs A, B, and C, i.e. we will use the points at which there is no significant improvement in performance by increasing the number of processors. Since each of these runs was evolved to a time of $t = 0.5M$ we can calculate the rate of evolution in M/hr. A summary of evolution rates is shown in Table 7.9.

Job	A	B	C
# Processors	64	128	512
Evolution Rate [M/hr]	3.9	1.8	1.1
Time required for 200M evolution [days]	2.1	4.6	7.6

Table 7.9: Rate of evolution of optimal configurations for Jobs A, B, and C.

Note that as mentioned above, the rates listed are not necessarily the fastest that can be achieved, but increasing speed would come at the expense of a significant increase in CPU time. For instance, running job B on 512 processors instead of 128 gains about a 50% improvement in performance with a 400% increase in computational expense.

Since we need to run a simulation long enough to allow for the gravitational radiation to propagate out to the wave zone ($r = 60M - 100M$), we typically need to evolve the system to a time of about $200M$. For Job B, this would take about $4\frac{1}{2}$ days. Job A would take a little over 2 days to run out to $t = 200M$, while Job C would take over a week. In light of this, we strive to improve performance in any way possible and we will continue to do so into the future.

7.4 Convergence

In addition to benchmarking the performance of *openGR*, we also need to ascertain its accuracy. To do this we perform a straightforward convergence test. This consists of running the same initial configuration on three different resolution grids, which we refer to as coarse, medium, and fine. We then compare the values of a particular variable along a straight line in each of the three simulations. Table 7.10 details the configurations for a single puncture FMR convergence test.

	Coarse	Medium	Fine
Points / Level	40^3	60^3	80^3
Refinement Levels	8	8	8
Domain	$\pm 100M$	$\pm 100M$	$\pm 100M$
Outer Boundary Resoltion	$5M$	$\frac{10M}{3}$	$\frac{5M}{2}$
Puncture Resolution	$\frac{5M}{128}$	$\frac{5M}{192}$	$\frac{5M}{256}$

Table 7.10: Configurations used for convergence tests. In all cases the hole is located at the origin and has mass M.

To compare values from different resolutions, we need to appropriately scale them to ensure consistency. We compute a scaling ratio R as follows:

$$R = \frac{(\frac{P_m}{P_f})^d \left[(\frac{P_c}{P_m})^d - 1\right]}{(\frac{P_m}{P_f})^d - 1}, \tag{7.4}$$

where P is the number of points along the line in one dimension, the subscript of P denotes the coarse, medium, and fine configurations, and d is the order, which in this case is 4. Notice that when the ratio $\frac{P_c}{P_m}$ is equal to the ratio $\frac{P_m}{P_f}$ (for example, using grids which have 40^3, 80^3, and 160^3 points per level, respectively), Eq. 7.4 reduces to

$$R = \left(\frac{P_m}{P_f}\right)^d. \tag{7.5}$$

We then look at a particular value for the coarse and fine configurations and compare against the medium configuration. For example, looking at the conformal factor ϕ we would calculate:

$$\Delta_{cm} = |(\phi_m - \phi_c)|/R \tag{7.6a}$$

$$\Delta_{fm} = |(\phi_f - \phi_m)|. \tag{7.6b}$$

Figure 7.5 shows the results of a convergence test performed at $t = 30M$. For perfect convergence the two lines would exactly overlap. We see very good convergence demonstrated out to $x = 60M$, particularly close to the hole (as $x \to 0$). We see less overlap further from the hole, but this is to be expected since the resolution is coarser due to the FMR grid structure.

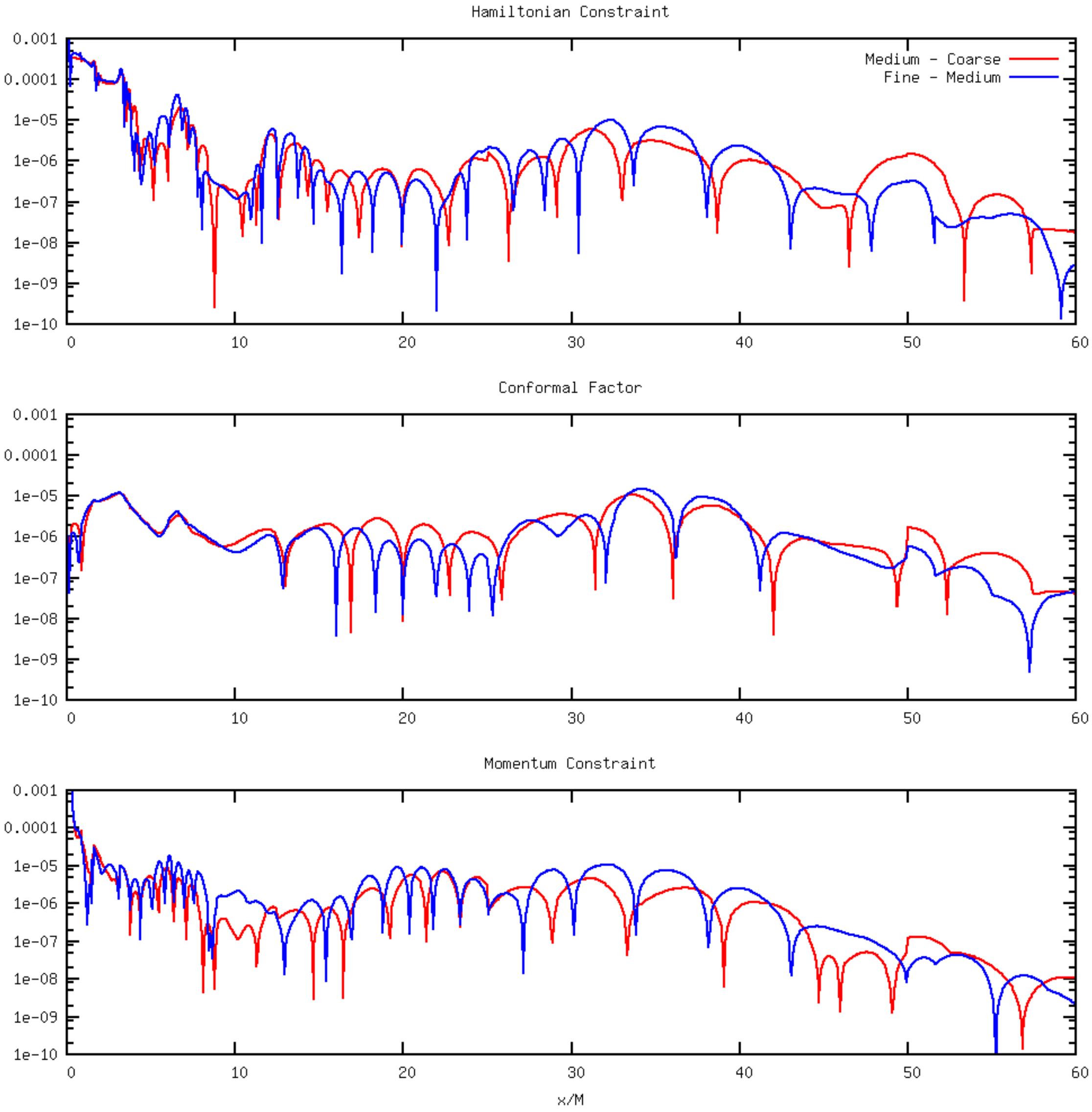


Figure 7.5: Results of a convergence test at time $t = 30M$. Plotted above are the Hamiltonian constraint, the conformal factor, and the x momentum constraint.

Chapter 8

Results

This chapter presents results from various simulations that demonstrate the capabilities of *openGR*. Specifically, it discusses a quasi-circular orbit (QC0), which undergoes inspiral, merger, and ringdown, and a straight infall from rest resulting in a head-on collision of the two holes. This chapter also shows how the dynamics of these cases are affected when spin is added to the holes. Finally, A comparison between an eccentric merger and a scattering interaction is shown. In all cases the gravitational radiation generated by the interaction is calculated.

8.1 QC0

Modeling quasai-circular orbits has become a standard test for numerical relativity because such models produce very regular waveforms while exhibiting all three stages of a binary black hole interaction: inspiral, merger, and ringdown. We use the parameters from the QC sequence given by Baker [36], which were adapted from Cook [37]. These parameters are summarized in Table 8.1.

Snapshots of the lapse α are shown in Fig. 8.1 at various stages during

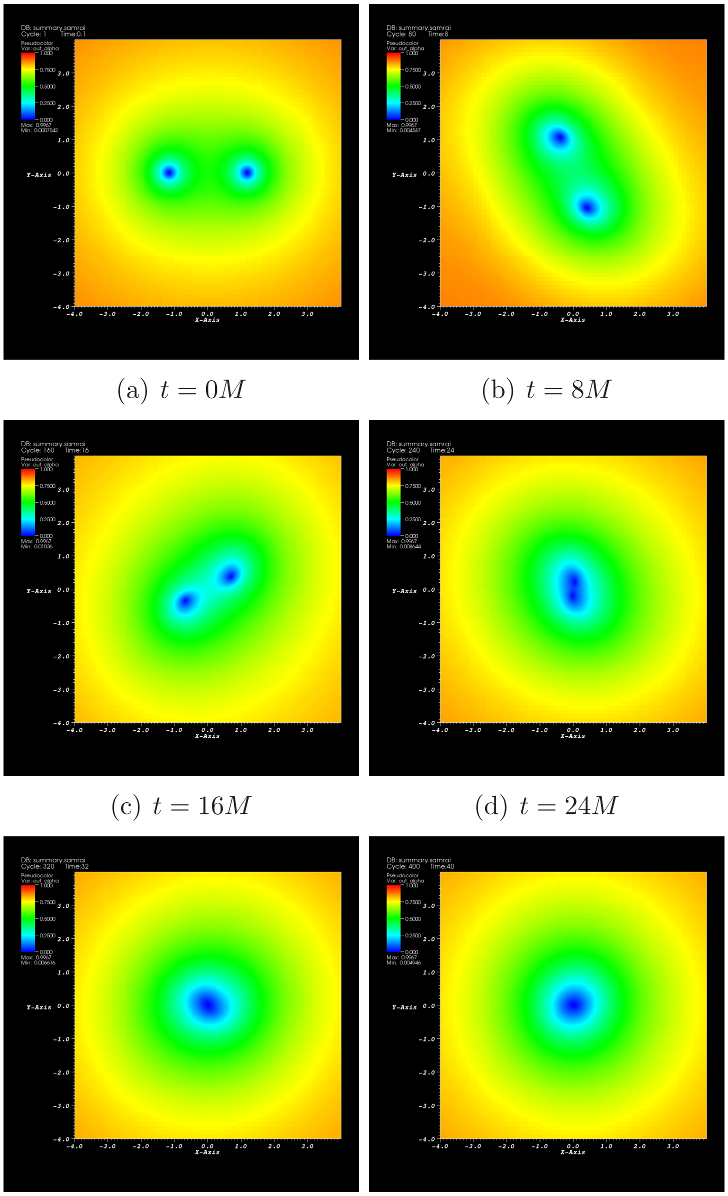
QC0	
Initial Positions	$(\pm 1.169M, 0, 0)$
Initial Momenta	$(\pm 0.333M, 0, 0)$
Initial Spins	$(0, 0, 0)$
Masses	$0.453M$
Physical Domain	$\pm 160M$
# Refinement levels	9
# Moving levels	2
Coarse Resolution	$4M$
Fine Resolution	$\frac{M}{64}$

Table 8.1: Summary of parameters used for QC0 simulation.

the evolution. The three stages of black hole interactions can be seen: inspiral, merger and ringdown. Inspiral (Figs 8.1(a) - 8.1(c)) is when the physical separation of the holes decreases as they orbit one another due to energy and momentum being radiated away. Once the separation of the holes decreases enough to the point that the holes share a common horizon, they are in the merger phase (Fig. 8.1(d)). After the merger a single distorted black hole remains. This black hole then begins to ringdown, during which time it continues to radiate until all the asymmetries are removed (Figs. 8.1(e) - 8.1(f)). After the merger is complete a single symmetric black hole remains.

The trajectories of the holes for the entire simulation are shown in Fig. 8.2. The dots plotted along the curve correspond to the positions at $5M$ intervals in time.

We can visualize the gravitational radiation being emitted by looking at snapshots of the real part of $\Psi_4(2, 2)$ (scaled by the extraction radius) at various times during the evolution, shown in Figure 8.3.



(a) $t = 0M$ (b) $t = 8M$

(c) $t = 16M$ (d) $t = 24M$

(e) $t = 32M$ (f) $t = 40M$

Figure 8.1: Full simulation of QC0. The lapse α is shown for different times in the evolution. The three stages of evolution can clearly be seen: inspiral (Figs. 8.1(a) - 8.1(c)), merger (Fig. 8.1(d)), and ringdown (Figs. 8.1(e) - 8.1(f)) Movie at: `http://wwwrel.ph.utexas.edu/Members/gmcivor/openGR/qc0_alpha.mp4`

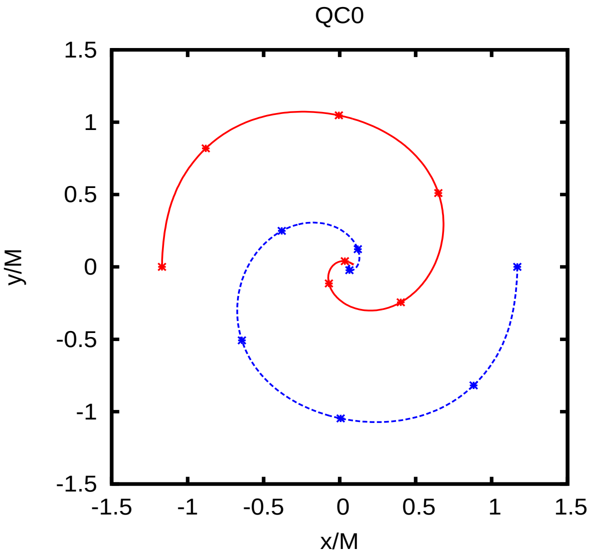
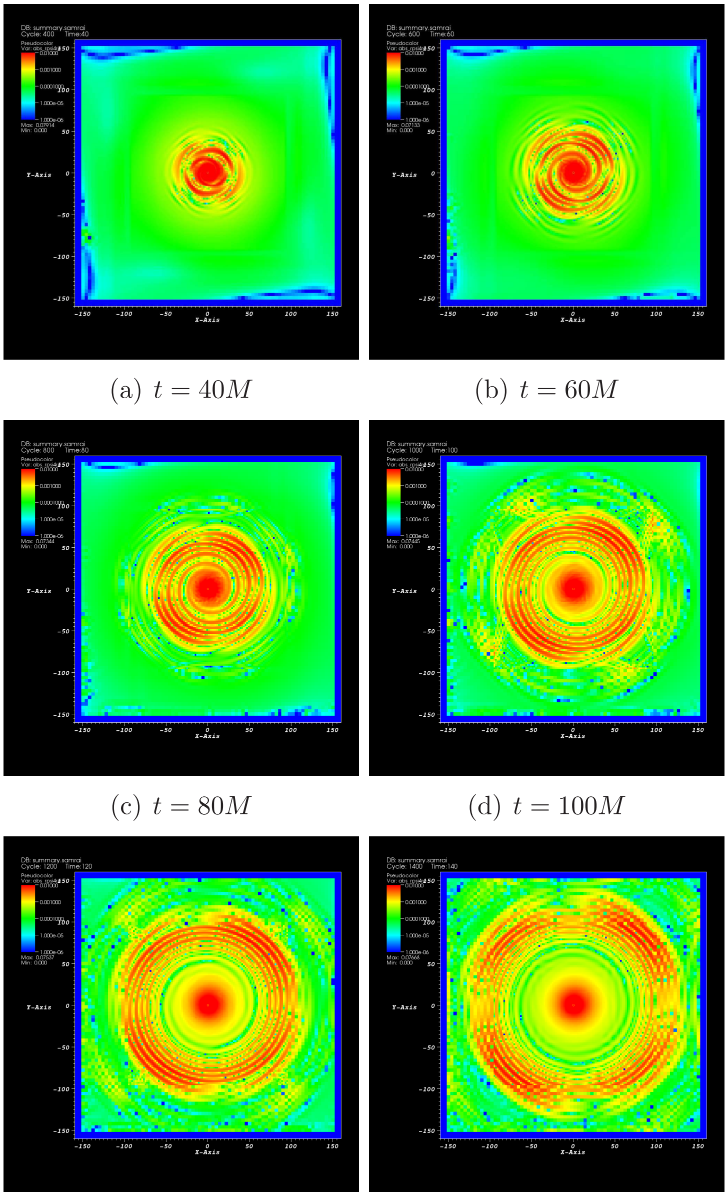


Figure 8.2: Trajectory of the holes in the $x - y$ plane for QC0. The points on the curves represent the positions of the holes at $5M$ intervals in time.

To generate a gravitational waveform, the Weyl scalar Ψ_4 is integrated on a sphere (weighted by spherical harmonics). We want to do this far away from the holes so that the weak field limit applies. That is, the radiation is propagating on a flat background. Fig. 8.4 shows the waveform of the real part of Ψ_4 $(l = 2, m = 2)$ (scaled by the extraction radius) plotted for various radii.

In an effort save time, some simulations were run on a more coarse grid than was used for QC0 (Table 8.1). In order to understand how the resolution affects the accuracy of the waveform, we perform a simulation of QC0 on the coarse grid and compare the results to that obtained using the fine grid. Fig. 8.5 shows $rRe[\Psi_4](2,2)$ for various extraction radii calculated on the coarse and



(a) $t = 40M$ (b) $t = 60M$

(c) $t = 80M$ (d) $t = 100M$

(e) $t = 120M$ (f) $t = 140M$

Figure 8.3: Full simulation of QC0. The product $rRe[\Psi_4](l = 2, m = 2)$ is shown for different times in the evolution. Movies at: http://wwwrel.ph.utexas.edu/Members/gmcivor/openGR/qc0_rpsi4re.mp4 and http://wwwrel.ph.utexas.edu/Members/gmcivor/openGR/qc0_rpsi4re_3d.mp4

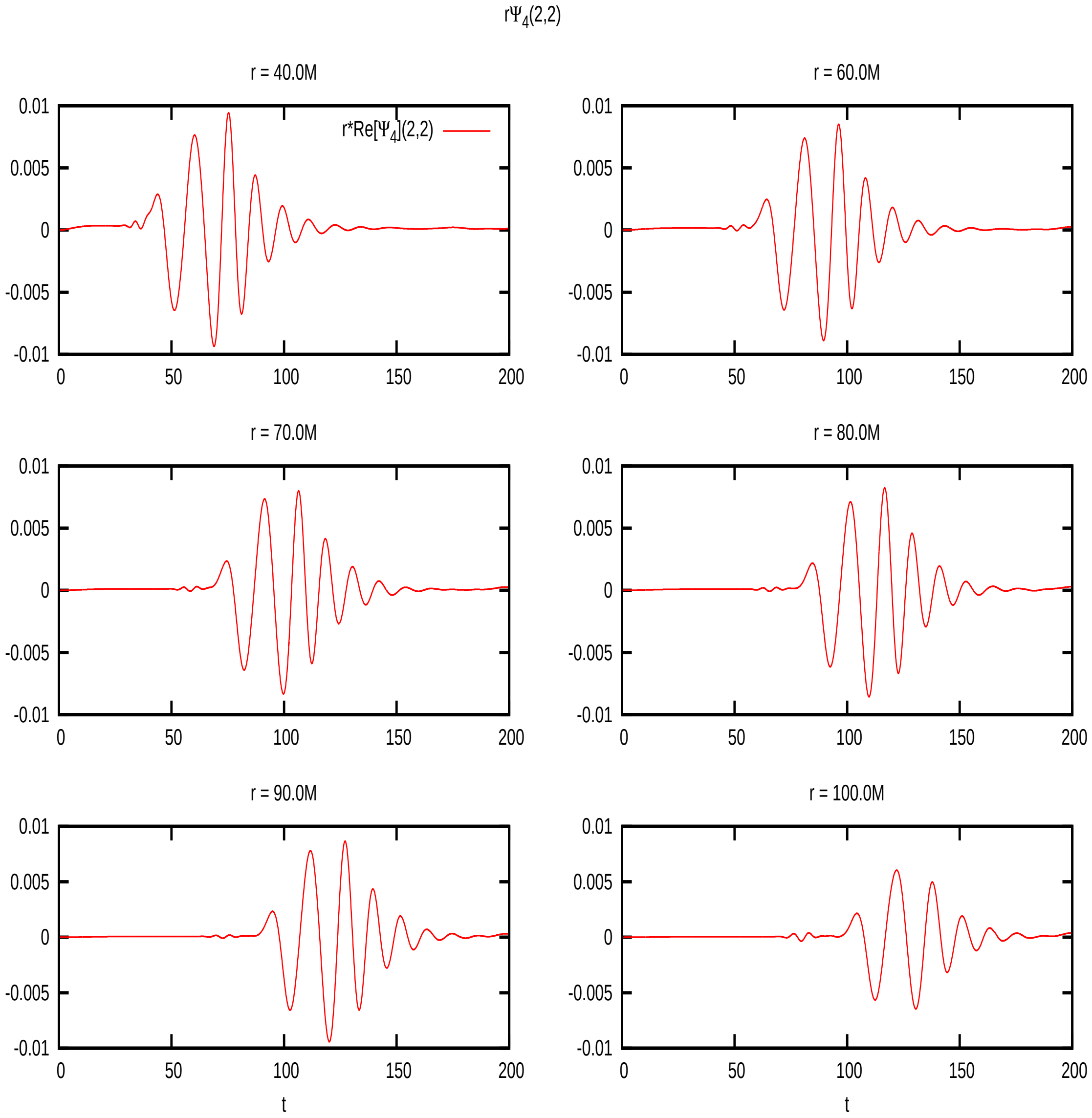


Figure 8.4: The real part of Ψ_4 $(l = 2, m = 2)$ scaled by the extraction radius shown for various extraction radii. The wave is clearly propagating outward and maintaining its shape very well.

fine grids. The two waveforms agree very well for $r = 40M$ due to the fact that the fine and coarse configurations have similar spatial resolution at that radius ($2M$ and $2.5M$, respectively). Beyond that, the coarse grid has significantly lower resolution than the fine grid ($5M$ and $2M$, respectively). This results in an artificial damping of the radiation on the coarse grid. Given the excellent agreement between the two waveforms at $r = 40M$, results obtained using the coarse grid are perfectly valid when extracted at this radius. Outside of this we have to be mindful that error is being introduced due to the coarseness of the grid.

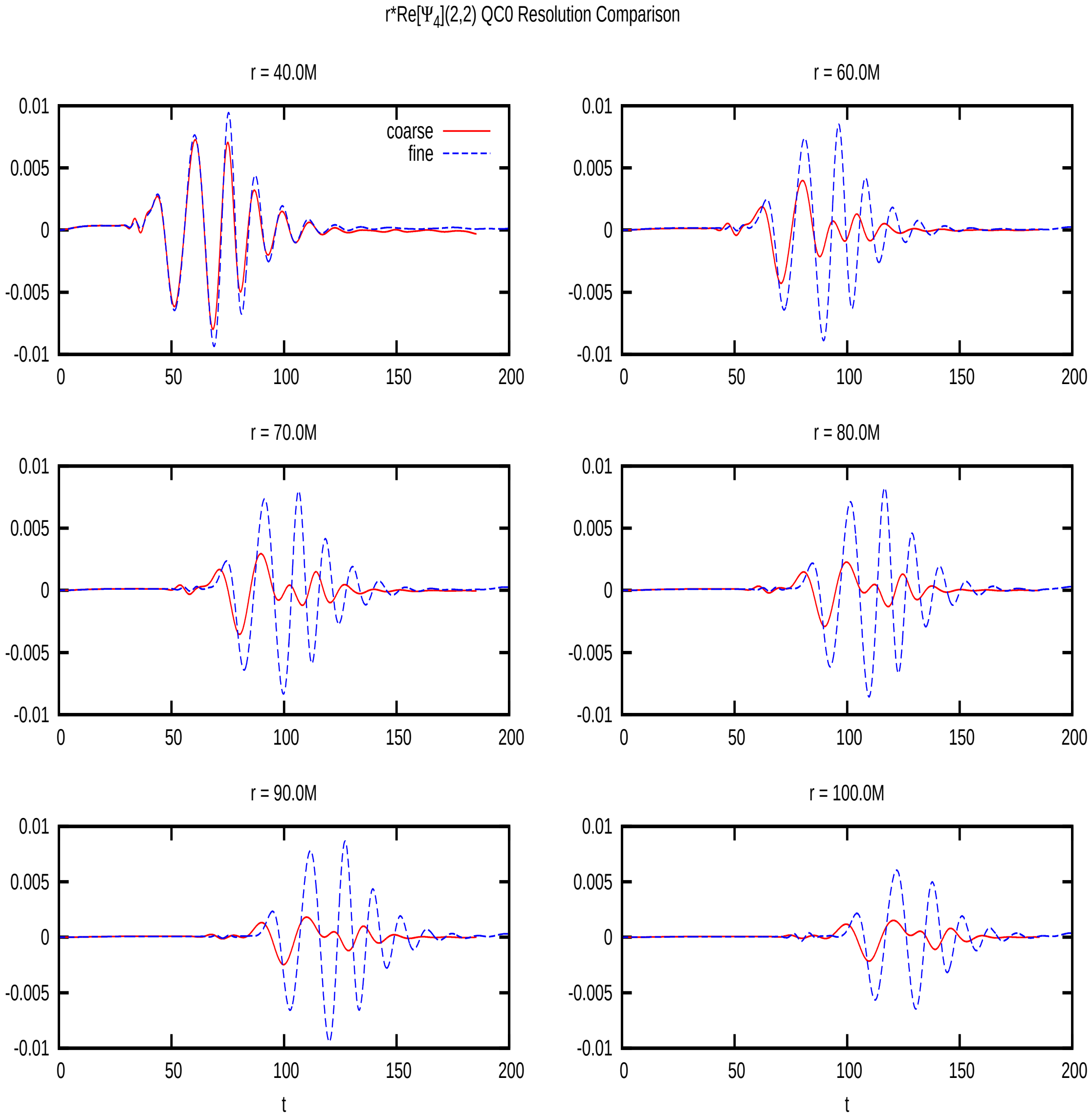


Figure 8.5: Comparison of coarse and fine resolution for QC0 wave extraction. $rRe[\Psi_4](2,2)$ is shown for various extraction radii. The two waveforms agree very well for $r = 40M$ because of similar spatial resolution between the two files at that radius. Further out the coarse grid becomes appreciably more coarse than the fine grid, and a clear difference in waveforms is seen.

8.2 Equal Mass Head-On Collision from Rest

Another standard test case of numerical relativity is the simulation of two equal mass black holes undergoing infall starting from rest. Table 8.2 details the parameters used for the simulation.

Equal Mass Head-On Collision from Rest	
Initial Positions	$(\pm 1.5M, 0, 0)$
Initial Momenta	$(0, 0, 0)$
Initial Spins	$(0, 0, 0)$
Masses	$0.464M$
Physical Domain	$\pm 200M$
# Refinement levels	8
# Moving levels	2
Coarse Resolution	$5M$
Fine Resolution	$\frac{5M}{128} \approx \frac{M}{26}$

Table 8.2: Summary of parameters used for Equal Mass Head-On Collision from Rest.

Fig. 8.6 shows $rRe[\Psi_4](2,2)$ for various radii. As we saw earlier in Fig. 8.5, the shape of the wave is changing at different radii due to error introduced by the coarse grid.

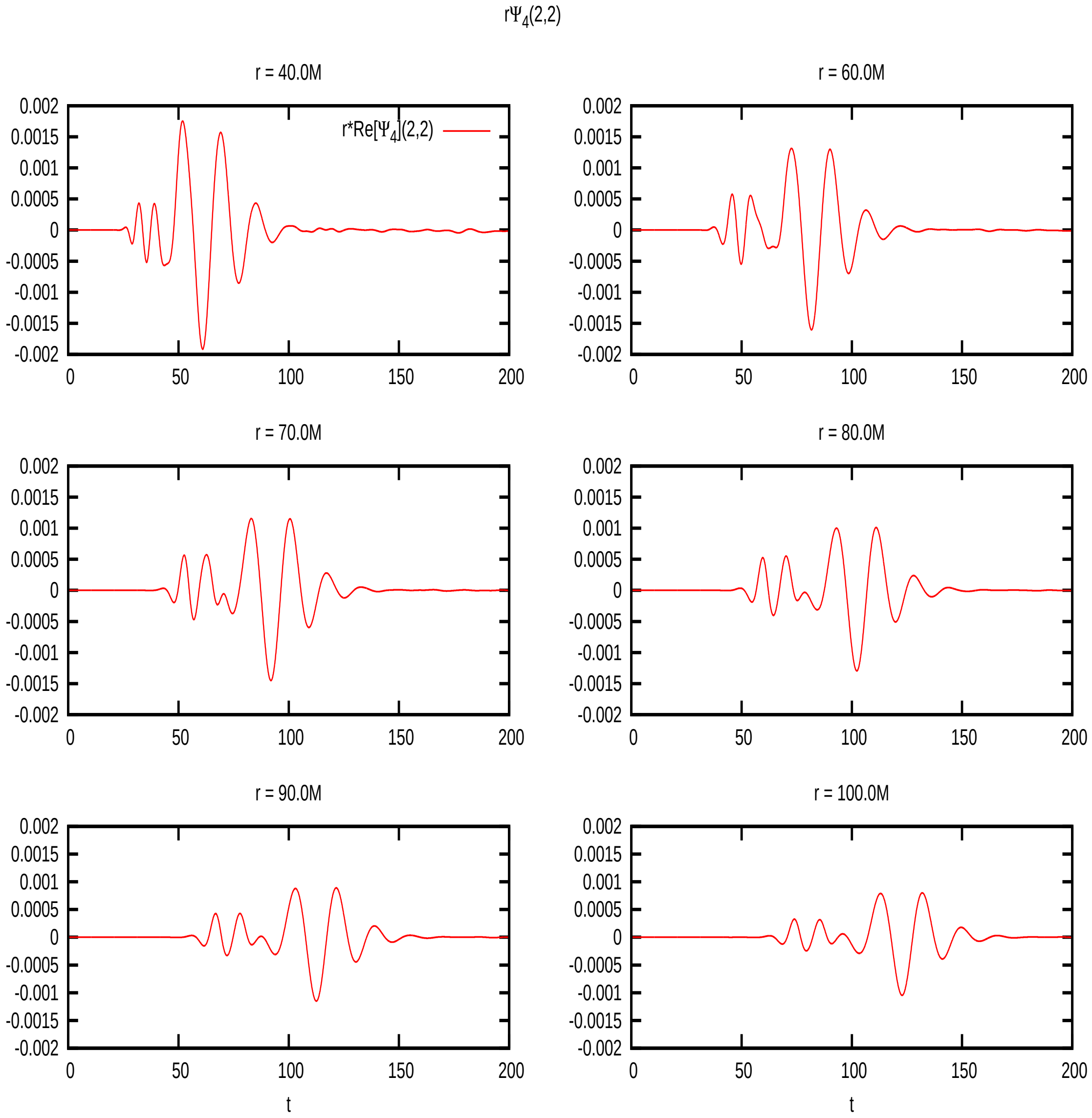


Figure 8.6: The product $rRe[\Psi_4](2,2)$ shown for two holes undergoing infall from rest.

8.3 QC0 with Spin

To model QC0 with spin, we add spin to the QC0 simulation to see how the spin affects the dynamics of the interaction and the radiation produced. Table 8.3 summarizes the parameters used to carry out the simulation of QC0 with spin. The black hole parameters are the same as those used earlier for QC0 (Table 8.1) with the addition of spin, and the grid configuration is that used for the case of straight infall (Table 8.2).

QC0 with Spin	
Initial Positions	$(\pm 1.169M, 0, 0)$
Initial Momenta	$(\pm 0.333M, 0, 0)$
Initial Spins	$(0, 0, \pm 0.6)$
Masses	$0.453M$
Physical Domain	$\pm 200M$
# Refinement levels	8
# Moving levels	2
Coarse Resolution	$5M$
Fine Resolution	$\frac{5M}{128} \approx \frac{M}{26}$

Table 8.3: Summary of parameters used for QC0 simulation with spin added.

Fig. 8.7 shows the trajectories of the holes in the $x - y$ plane for QC0 with spin. As before, the dots plotted along the curve correspond to the positions at $5M$ intervals in time.

Comparing Fig. 8.7 to Fig. 8.2 it is clear that the addition of spin changes the dynamics of the interaction. However, this could be the result of the initial data solver. Since the initial data solver is not fully implemented to utilize the multi-grid setup, we do not get the benefit of the accuracy of the finer levels. As such, errors generated by the initial data on the coarse

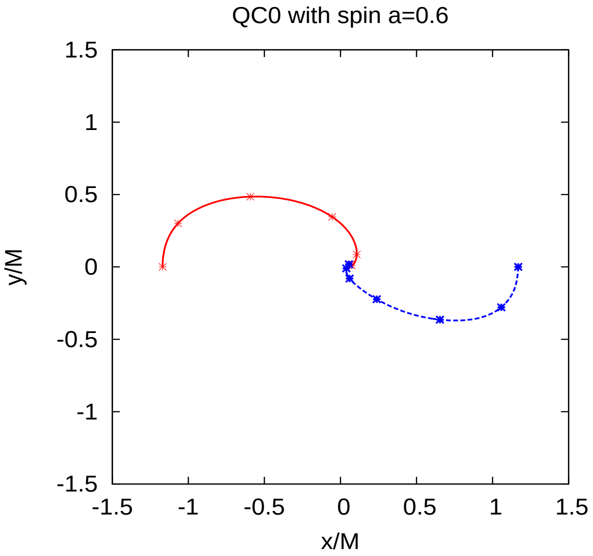


Figure 8.7: Trajectory of the holes in the $x - y$ plane for QC0 with spins $a = \pm 0.6$. The points along the curves represent the positions of the holes at $5M$ intervals in time.

level will be propagated onto the finer levels. This is generally not an issue, but holes with spin are a much more complicated initial configuration than those without spin, particularly when they are close together. Further testing is required to understand the interaction shown in Fig. 8.7.

We also want to see the effect spin has on the radiation produced. Fig. 8.8 shows a comparison of the QC0 waveform extracted at $r = 40M$ with and without spin. Much like the trajectory, we see a significant difference in the appearance of the waveform. Again, this could be due to the errors introduced by the initial data solver.

Comparing this to waveforms produced by Campanelli [38], it is sur-

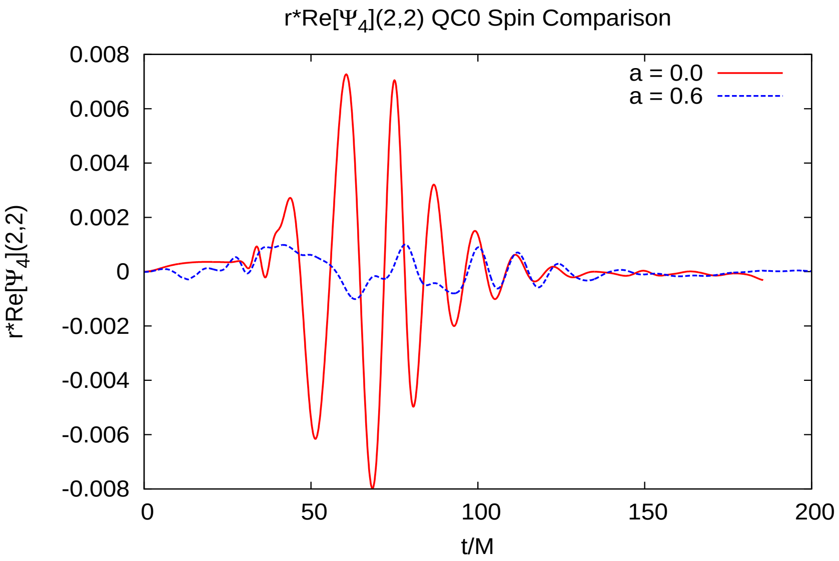


Figure 8.8: The product $rRe[\Psi_4(2,2)]$ extracted at $r = 40M$ shown for QC0 and QC0 with spin $a = \pm 0.6$.

prising to see such a big difference in our waveform with spin as compared to the waveform produced without spin. This indicates that what we are seeing is, in fact, a result produced by the initial data. One thing to note, however, is that the ringdown phase of the interaction starting from about $t = 100M$ is similar with and without spin. This further indicates that no serious issue exists, but rather that the initial data is not very accurate for this configuration. This is not all that surprising given the close proximity of the holes.

New tests are currently underway which set the locations of the holes further apart at the start of the simulation. This should yield better initial data and resolve any discrepancies we are seeing when comparing to other work.

8.4 Head-On Collision with Spin

To make one final comparison we will revisit the equal mass head-on collision from rest, but add spin to both holes. The configuration is exactly the same as shown in Table 8.2 with the addition of the initial spin of $a = 0.8$ in the z direction (i.e. spin $= (0, 0, 0.8)$). Fig. 8.9 shows the x positions of both of the holes as a function of time. The difference in time it takes to merge in the two scenarios can clearly be seen; the spin interaction slows down the merger.

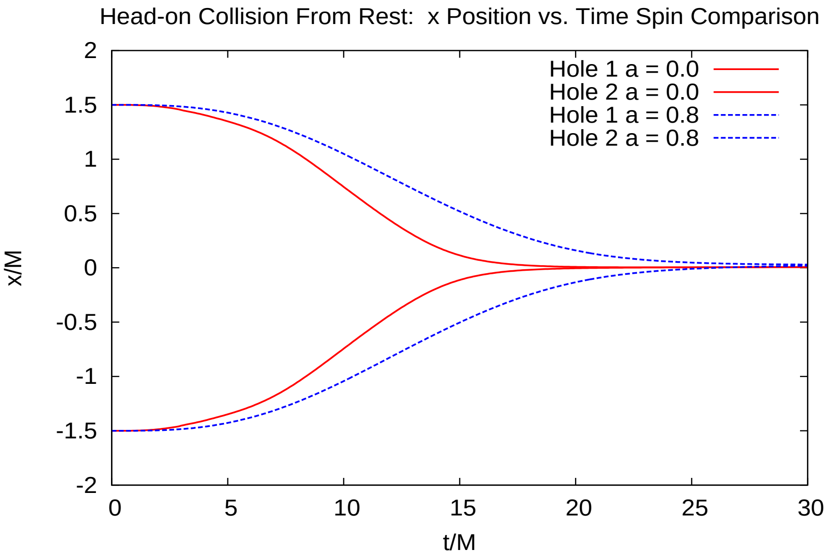


Figure 8.9: Comparison of the x position vs. time for two holes with and without spin starting from rest at $x = \pm 1.5M$. The spin interaction clearly slows down the merger.

Fig. 8.10 shows the waveform extracted at $r = 40M$ for the equal mass head-on collision, comparing the cases with and without spin.

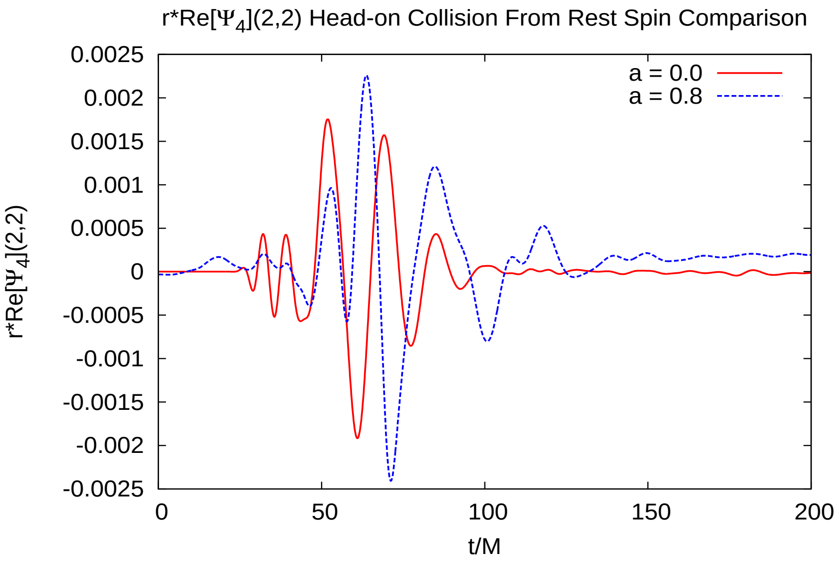


Figure 8.10: The real part of $r\Psi_4$ shown for a head-on collision from rest with and without spin.

We can see that for the case with spin the waveform is slightly stretched out and has a larger amplitude, but the shape is generally the same as in the case without spin. As with the QC0 case with spin, we have to question the accuracy of the initial data. However, in this case (as compared to QC0) the holes are further apart and have no initial velocity, conditions which should lead to better behaved initial data. Given that, it is not surprising that the difference in spinning versus non-spinning waveforms is not as drastic as those differences shown for QC0.

8.5 Scattering Interaction

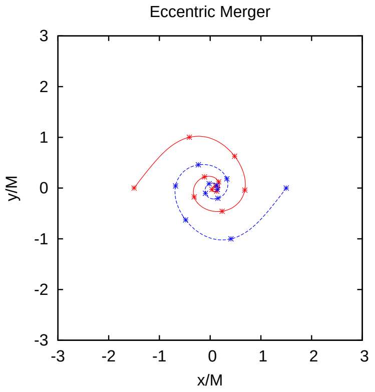
This section moves beyond simple test cases and highlights an example problem that warrants further investigation: scattering interactions. Specifically, scattering interactions provide an example of how varying one parameter (in this case the initial momentum in the x direction) can lead to very different interactions. Table 8.4 summarizes the parameters used in the two initial configurations that demonstrate this principle.

Eccentric Merger and Scattering Interaction		
	Merger	Scatter
Initial Positions	$(\pm 1.5M, 0, 0)$	$(\pm 1.5M, 0, 0)$
Initial Momenta	$(\pm 0.4M, \pm 0.8M, 0)$	$(\pm 0.6M, \pm 0.8M, 0)$
Initial Spins	$(0, 0, 0)$	
Masses	$0.464M$	
Physical Domain	$\pm 200M$	
# Refinement levels	8	
# Moving levels	2	
Coarse Resolution	$5M$	
Fine Resolution	$\frac{5M}{128} \approx \frac{M}{26}$	

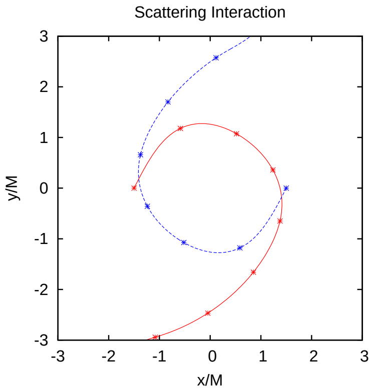
Table 8.4: Summary of parameters used for an eccentric merger and a scattering interaction. All parameters are identical except for the initial momentum in the x direction.

Fig. 8.11 shows the trajectories for the two cases with initial parameters summarized in Table 8.4. Fig. 8.11(a) shows an eccentric orbit that ends with a merger of the two holes. In Fig. 8.11(b), which also starts as an eccentric orbit, it is clear that changing the momentum causes the holes not to merge, but instead to undergo a scattering interaction in which they are left unbound and move away from each other. It should be noted that the initial parameters

(Table 8.4) were chosen somewhat arbitrarily and are not intended to be ideal. Rather, they are a simple demonstration of the very different behavior that can be achieved with a small change in one parameter.



(a) Eccentric merger trajectory in the $x-y$ plane.



(b) Scattering interaction trajectory in the $x-y$ plane.

Figure 8.11: Trajectories of the holes in the $x-y$ plane for an eccentric merger and a scattering interaction. The points on the curves represent the positions of the holes at $5M$ intervals in time. Movies at: `http://wwwrel.ph.utexas.edu/Members/gmcivor/openGR/scatter1_alpha.mp4` and `http://wwwrel.ph.utexas.edu/Members/gmcivor/openGR/scatter2_alpha.mp4`

By precisely tuning parameters such as momentum, very complicated interactions can be achieved in which the holes undergo a series of close orbits as well as large orbits before finally merging or scattering apart. This type of trajectory is known as "zoom-whirl" by Pretorius and Khurana [39] and Healy et al. [40]. Similar behavior was also found by Washik et al. [41], who refer to it as "splash-skip." Using *openGR*, we would like to build on these efforts and perform a parameter investigation in the region of the merger/scattering boundary. By performing a systematic analysis on parameters such as initial momentum and initial separation we can determine the threshold of these parameters that lead to merger. We can also investigate the effect spin will have on these thresholds. Additionally, the gravitational radiation from these interactions will be studied to understand how these threshold parameters are reflected in the waveforms. A simple example how these parameter are reflected in the waveform is shown in Fig. 8.12.

Fig. 8.12 shows the waveform extracted at $r = 40M$ for an eccentric merger and a scattering interaction. The two initial configurations (outlined in Table 8.4) are very similar, with the momentum of one case adjusted slightly to produce a scattering interaction instead of a merger. Accordingly, the waveforms are very similar during the early part of the interaction. As one might expect given the differences in trajectories, the waveforms vary greatly later in the interaction. The very end of the eccentric merger waveform is particularly interesting, as it apparently exhibits some sort of excitation or ringing. The simulation ended at that point, but if run longer we would

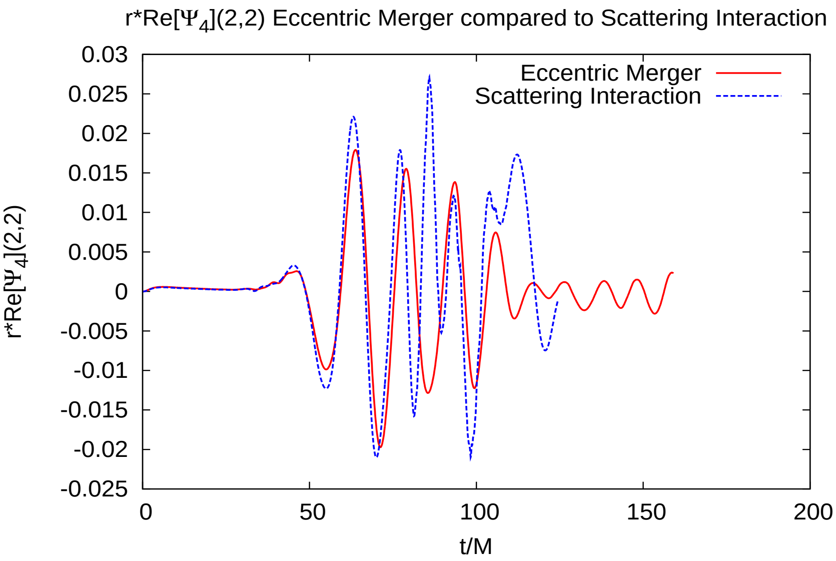


Figure 8.12: The real part of $r\Psi_4$ shown for an eccentric merger and a scattering interaction. The two initial configurations are very similar, with the momentum of one case adjusted slightly to give a scattering interaction instead of a merger. As such, the waveforms are very similar during the early part of the interaction.

expect to see ringdown occur shortly thereafter.

It should be noted that the waveform for the scattering interaction should be considered suspect after about $t = 70M$. Since the holes are moving apart one would not expect to see such variation in the radiation. We believe that these variations are due to the fact that the holes are moving outward, toward the region in which the radiation is being extracted. Additionally, the momentum in this case is very large, so the holes are trailing right behind the waves and moving almost as fast. As such, the gravitational wave signal is very noisy. Future analysis will use more carefully chosen parameters and the radiation will be extracted at a radius further from the origin, which should minimize these noisy effects and yield much cleaner waveforms.

Chapter 9

Conclusion

After the collective efforts of many people over the past decade, *openGR* is now fully functional and ready to be used for full-scale scientific research. After a major overhaul in which the code was re-organized to improve readability as well as eliminate redundant calculations, certain features that used to work were broken, such as the moving boxes. I worked to restore the moving boxes as well as ensure the conversion from coordinate space to grid points was done correctly. Other pieces of *openGR*, such as wave extraction, were partially implemented in the past but never completely operational. After running many test cases I was able to isolate and repair the issue and now the wave extraction is fully implemented and producing the first ever waveforms generated by *openGR*, as shown in Chapter 8.

Part of the development of *openGR* that takes a significant amount of time, but is not at all reflected in the results, is debugging, particularly as it pertains to grid configurations. I spent a considerable amount of time running jobs attempting to find stable grid layouts and better understand how to get *SAMRAI* to operate as optimally as possible. Although this is an ongoing endeavor, we now understand the nuances of this process better than ever before.

One aspect that sets *openGR* apart from other black hole codes is accuracy. *openGR* is fourth-order accurate in both space and time while most other codes are second-order in time. This yields high accuracy and very good convergence. Another way in which *openGR* differs from other numerical relativity codes is the use of the *SAMRAI* library. We have shown that *openGR* provides good scaling to very large number of processors. As simulations get pushed to higher resolution this will prove to be a very useful quality.

This is an exciting time for those involved in numerical relativity and gravitational wave physics. Advances in numerical relativity that have occurred in the past decade now allow for long stable evolution of countless initial configurations of black holes. With upgrades currently underway, Advanced LIGO will improve on the strain sensitivity of initial LIGO by a factor of 10. Advanced LIGO, which has an expected completion of 2014, has the potential to have upwards of 100 gravitational wave detections per year. For the first time, *openGR* is now in a state in which it can be used to make significant contributions to that effort.

I am very confident that *openGR* will join the ranks alongside other established black hole simulation codes and will be widely accepted and used by the numerical relativity community.

Bibliography

[1] A. A. Abramovici, W. Althouse, R. P. Drever, Y. Gursel, S. Kawamura, F. Raab, D. Shoemaker, L. Sievers, R. Spero, K. S. Thorne, R. Vogt, R. Weiss, S. Whitcomb, and M. Zuker. Ligo: The laser interferometer gravitational-wave observatory. *Science*, 256:325–333, 1992.

[2] B. Caron, A. Dominjon, C. Drezen, R. Flaminio, X. Grave, F. Marion, L. Massonnet, C. Mehmel, R. Morand, B. Mours, et al. The virgo interferometer for gravitational wave detection. *Nuclear Physics B-Proceedings Supplements*, 54(3):167–175, 1997.

[3] K. Danzmann. The geo project: a long baseline laser interferometer for the detection of gravitational waves. *Lecture Notes in Physics*, 410:184–209, 1992.

[4] Frans Pretorius. Evolution of binary black hole spacetimes. *Phys. Rev. Lett.*, 95:121101, 2005.

[5] 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.

[6] 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.

[7] Matthew Anderson. *Constrained Evolution in Numerical Relativity*. PhD thesis, University of Texas at Austin, 2004.

[8] Andrew M. Wissink, Richard D. Hornung, Scott R. Kohn, Steve S. Smith, and Noah Elliott. Large scale parallel structured AMR calculations using the SAMRAI framework. *sc*, 00:22, 2001. doi: http://doi.ieeecomputersociety.org/10.1109/SC.2001.10029.

[9] M. Alcubierre. *Introduction to 3+ 1 numerical relativity*, volume 140. Oxford University Press, USA, 2008.

[10] Jr. James W. York and Tsvi Piran. The initial value problem and beyond. In Richard A. Matzner and Lawrence C. Shepley, editors, *Spacetime and Geometry: The Alfred Schild Lectures*, pages 147–176. University of Texas Press, Austin (Texas), 1982. ISBN 0-292-77567-9.

[11] Frans Pretorius. Evolution of binary black hole spacetimes. *Phys. Rev. Lett.*, 95:121101, 2005.

[12] Bernd Bruegmann, Jose A. Gonzalez, Mark Hannam, Sascha Husa, Ulrich Sperhake, and Wolfgang Tichy. Calibration of moving puncture simulations. *Physical Review D*, 77:024027, 2008. URL `doi:10.1103/PhysRevD.77.024027`.

[13] D. Brill and R. Lindquist. Interaction energy in geometrostatics. *Phys. Rev.*, 131(1):471–476, 1963.

[14] Jeffrey M. Bowen and James W. York. Time asymmetric initial data for black holes and black hole collisions. 21:2047–2056, 1980.

[15] S. Brandt and B. Brügmann. A simple construction of initial data for multiple black holes. *Phys. Rev. Lett.*, 78(19):3606–3609, 1997.

[16] M. Shibata and Takashi Nakamura. Evolution of three-dimensional gravitational waves: Harmonic slicing case. *Phys. Rev. D*, 52:5428, 1995.

[17] Thomas W. Baumgarte and Stuart L. Shapiro. On the numerical integration of Einstein's field equations. *Phys. Rev. D*, 59:024007, 1999.

[18] C. Bona, Joan Massó, E. Seidel, and J. Stela. First order hyperbolic formalism for numerical relativity. *Phys. Rev. D*, 56:3405–3415, 1997.

[19] C. Gundlach and J.M. Martin-Garcia. Well-posedness of formulations of the Einstein equations with dynamical lapse and shift conditions. *Physical Review D*, 74(2):024016, 2006.

[20] T. Regge and J. Wheeler. Stability of a Schwarzschild singularity. *Phys. Rev.*, 108(4):1063–1069, 1957.

[21] F.J. Zerilli. Gravitational field of a particle falling in a Schwarzschild geometry analyzed in tensor harmonics. *Phys. Rev. D.*, 2:2141, 1970.

[22] S.A. Teukolsky. Perturbations of a rotating black hole. I. fundamental equations for gravitational, electromagnetic, and neutrino-field perturbations. *Astrophys. J.*, 185:635–647, 1973.

[23] E. Newman and R. Penrose. An approach to gravitational radiation by a method of spin coefficients. *J. Math. Phys.*, 3:566, 1962.

[24] Roger Penrose and W. Rindler. *Spinors and Spacetime*, volume 1, 2. Cambridge University Press, 1984, 1986.

[25] L. Gunnarsen, H. Shinkai, and K. Maeda. A '3+1' method for finding principal null directions. *Class. Quantum Grav.*, 12:133–140, 1995.

[26] Andrea Nerozzi, Christopher Beetle, Marco Bruni, Lior M. Burko, and Denis Pollney. Towards wave extraction in numerical relativity: The quasi-Kinnersley frame. *Phys. Rev.*, D72:024014, 2005.

[27] Andrea Nerozzi. Scalar functions for wave extraction in numerical relativity. *Phys. Rev.*, D75:104002, 2007.

[28] P Walter. *Using openGR for Numerical Relativity*. PhD thesis, The University of Texas at Austin, 2009.

[29] Ligo.gif. URL `http://en.wikipedia.org/wiki/File:Ligo.gif`.

[30] LIGO-G060009-03. Ligo best strain sensitivity. URL `https://dcc.ligo.org/public/0036/G060009/000/G060009-03.pdf`.

[31] G.M. Harry. Advanced ligo: the next generation of gravitational wave detectors. *Classical and Quantum Gravity*, 27:084006, 2010.

[32] Diego Fazi. *Development of a physical-template search for gravitational waves from spinning compact-object binaries with LIGO*. PhD thesis, Universita di Bologna, 2009.

[33] Benjamin Aylott, John G. Baker, William D. Boggs, Michael Boyle, Patrick R. Brady, et al. Testing gravitational-wave searches with numerical relativity waveforms: Results from the first Numerical INJection Analysis (NINJA) project. *Class.Quant.Grav.*, 26:165008, 2009. doi: 10.1088/0264-9381/26/16/165008.

[34] Melissa Frei. *Comparative Efficiency and Parameter Recovery of Spin Aligned Templates for Compact Binary Coalescence Detection*. PhD thesis, University of Texas at Austin, 2011.

[35] G.D. Byrne and A.C. Hindmarsh. Pvode, an ode solver for parallel computers. *International Journal of High Performance Computing Applications*, 13(4):354–365, 1999.

[36] J. Baker, M. Campanelli, C. O. Lousto, and R. Takahashi. Modeling gravitational radiation from coalescing binary black holes. *Phys. Rev. D*, 65:124012, 2002.

[37] Gregory B. Cook. Three-dimensional initial data for the collision of two

black holes II: Quasi-circular orbits for equal-mass black holes. *Phys. Rev. D*, 50(8):5025–5032, 1994.

[38] Manuela Campanelli, C. O. Lousto, and Y. Zlochower. Gravitational radiation from spinning-black-hole binaries: The orbital hang up. 2006.

[39] Frans Pretorius and Deepak Khurana. Black hole mergers and unstable circular orbits. *Classical and Quantum Gravity*, 24(12):S83, 2007. URL `http://stacks.iop.org/0264-9381/24/i=12/a=S07`.

[40] James Healy, Janna Levin, and Deirdre Shoemaker. Zoom-whirl orbits in black hole binaries. *Phys. Rev. Lett.*, 103: 131101, Sep 2009. doi: 10.1103/PhysRevLett.103.131101. URL `http://link.aps.org/doi/10.1103/PhysRevLett.103.131101`.

[41] M.C. Washik, J. Healy, F. Herrmann, I. Hinder, D.M. Shoemaker, P. Laguna, and R.A. Matzner. Binary-black-hole encounters, gravitational bursts, and maximum final spin. *Physical review letters*, 101(6):61102, 2008.

[42] B. Schutz. *A First Course in General Relativity*. Cambridge University Press, 1985.

[43] Charles W. Misner, Kip S. Thorne, and John A. Wheeler. *Gravitation*. W. H. Freeman, San Francisco, 1973.

[44] 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.

[45] A. Bechinger. Gravitational wave extraction in numerical relativity. Master's thesis, University of Texas at Austin, 2005.

Vita

Greg McIvor was born in Philadelphia, PA. He attended high school at Friends' Central School in Wynnewood, PA, where he graduated in 1999. He then attended Drexel University in Philadelphia, PA, graduating summa cum laude in 2004 with a B.S. in Physics. In 2005, after working one year in the aerospace/defense industry in Los Angeles, CA, he entered graduate school in the Department of Physics at the University of Texas at Austin.

Permanent address: 1721 Laguna St. Apt A
Santa Barbara, CA 93101

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.