Puncture evolutions within the harmonic framework

Diplomarbeit

eingereicht von

Philipp Mösta

Universität Kassel

und

Max-Planck-Institut für Gravitationsphysik
Albert-Einstein-Institut
Potsdam / Golm

25. März 2008

Ich erkläre, dass ich die Arbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder Sinne nach anderen Werken entnommen sind, durch Angabe der Quellen als Entlehnung kenntlich gemacht worden sind.

Kassel, den 25. März 2008 _____

Philipp Mösta

Contents

1 Introduction **1**
1.1 Acknowledgements . . . 3

2 The Einstein equations in Numerical Relativity **5**
2.1 Foliation of spacetime . . . 5
2.2 The ADM equations . . . 8
2.3 Generalized Harmonic equations . . . 12
2.4 The 'Densitized Harmonic' evolution system . . . 12
2.4.1 Constraint Adjustment and Damping . . . 14
2.4.2 Gauge Conditions . . . 14
2.4.3 Blended subluminal-superluminal evolution . . . 15
2.5 The 'ADM Harmonic' evolution system . . . 17
2.5.1 Gauge source functions . . . 23
2.5.2 Constraint adjustment . . . 25

3 Numerical Implementation **27**
3.1 Finite differencing scheme and operators . . . 27
3.2 Conformal decomposition of the metric for finite differencing . . . 28
3.3 Numerical Dissipation . . . 28
3.4 Boundary conditions . . . 29
3.5 Moving Excision . . . 30
3.6 Convergence tests . . . 32

4 Evolutions of single black holes **37**
4.1 Stability tests and different constraint damping parameters . . . 37

5 Evolution of binary black holes **45**
5.1 Head-on black hole collision: 'ADM Harmonic' vs. 'Densitized Harmonic' . . . 45

6 Concluding remarks **55**
6.1 Summary and general remarks . . . 55
6.2 Future outlook . . . 55

7 Deutsche Zusammenfassung (German summary) **57**

8 Appendix A **59**
8.1 The Method of Lines . . . 59
8.2 Convergence . . . 60
8.3 Well-posedness . . . 61

8.4 Overview of stability analysis 62

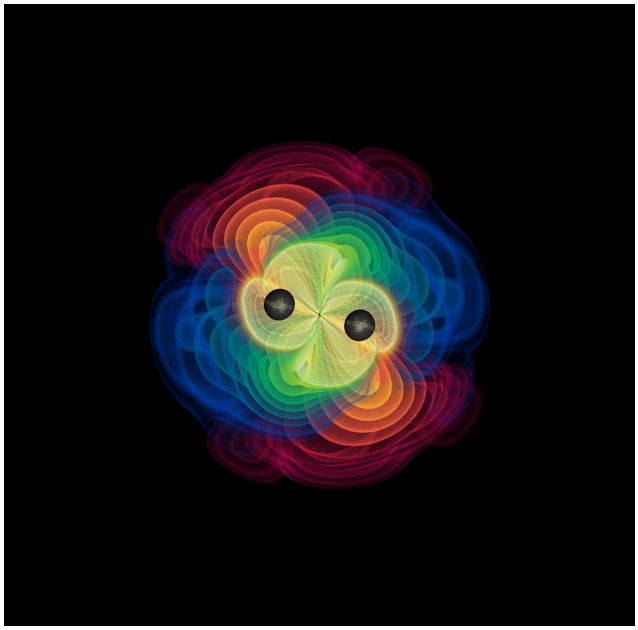
9 Appendix B **67**
9.1 The *Cactus* computational toolkit 67
9.1.1 An example of *Cactus* scheduling 67

1 Introduction

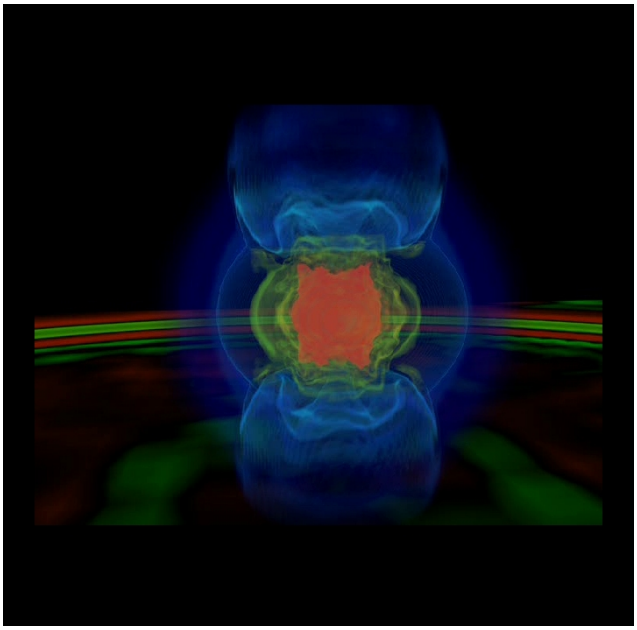
We are at the brink of opening a new window to the universe. With gravitational wave detectors like GEO600, VIRGO, and LIGO already in operation, one expects to observe gravitational wave signals from far away extreme events like stellar core collapses (supernovae) and binary black hole or neutron star mergers. By that one hopes to gain very exciting new insights into the role of gravitation as one of nature's fundamental interactions and the evolution of our universe in general. Due to the very weak nature of gravitational wave signals it is a very delicate and complicated effort to detect waveforms. Therefore it is one of numerical relativity's (NR) primary goals to simulate the events one hopes to observe and investigate the resulting waveforms.

But simulating those events by numerically solving Einstein's equations turned out to be much more complicated than initially expected. Trying to evolve binary black hole mergers introduces delicate numerical difficulties and has thus become one major challenge in numerical relativity. Einsteins equations, a set of coupled non-linear partial differential equations (PDEs), can be formulated in equivalent but different forms. In the harmonic formulation gauge source functions are chosen in a way that the equations for the metric components form a set of wave equations. These equations then have known stability properties deduced from mathematical and numerical analysis. The BSSN system on the contrary is based on a 3+1 formalism which splits Einsteins equations into space- and timelike parts and handles them separately. They are a modification of the ADM equations, which have proven to be numerically stable and well behaved. Another crucial problem is how to handle singularities arising in the evolved quantities on the finite valued numerical grid one uses to simulate the domain. Different approaches have been developed in that attempt. The black hole excision technique tries to excise the singularities from the grid, by usually excluding entire domains that are causally disconnected from spacetime. This is accomplished by tracking the apparent horizon of a black hole and excising everything inside it. However excision techniques lead to complicated grid setups and furthermore introduce the need of imposing excision boundary conditions. This leads to difficult to solve stability problems for the numerical evolution scheme. Another concept of dealing with singularities on the grid is the 'puncture approach'. One here uses different initial data which is constructed by absorbing the singular terms in the black hole interior into an analytic expression. These initial data sets are then evolved without excising regions of the grid at all. While BSSN systems can evolve puncture initial data without major problems and benefit valuably from not having to excise regions of the grid, so far nobody has achieved a working harmonic evolution system that can handle puncture intital data without using excision techniques.

In this thesis I present a 3+1 splitted harmonic evolution system which I implemented in order to investigate its ability to handle puncture initial data. It is based on the ADM variables lapse function α, shift vector β , 3-metric g_{ab} and the extrinsic curvature k_{ab} and uses 3+1 splitted harmonic gauge source functions. The thesis is composed in two major parts. First I concentrate on Einstein's Equations in numerical relativity. I briefly describe the derivation of



(a) Snapshot of two inspiraling black holes. The colors indicate the strength of the complex Weyl-component Ψ_4 which describes the gravitational radiation at future null infinity scri



(b) Snapshot of a simulated supernova explosion. The colors indicate the fluid density. On a plane below, the complex Weyl-component Ψ_4 is drawn.

Figure 1.1: Visualizations of simulation data.

the ADM equations in order to motivate the 3+1 formalism and describe two different harmonic evolution systems. I compare a densitized system implemented by Béla Szilágyi against the ADM based 3+1 splitted evolution system first described by Helmut Friedrich, which the work presented here is based on. In the following section I describe the basic numerical techniques, finite differencing stencils, dissipation operators and a decomposition of the metric components I use in the code.
The second part then mainly focusses on the results obtained by running this code with different black hole spacetime configurations using excision techniques. In the first section I concentrate on results for single black holes. I first show the code's stability for evolving single non spinning black holes on long time scales and investigate the influence of different constraint damping parameters on the evolution. Finally I present convergence tests for the ADM variables. In the second section I focus on binary black holes. I present data and visualizations describing a head-on collision of two non spinning black holes. I describe the merger of the apparent horizons, extract the gravitational wave signal and investigate the time evolution of the harmonic constraints. Furthermore I compare results for this same binary black-hole head-on-collision with the 'densitized' harmonic formulations described earlier.

1.1 Acknowledgements

I would like to thank a few people who helped and supported me during the process of developing and writing this thesis. First of all, I would like to thank Béla Szilágyi, Denis Pollney and Luciano Rezzolla for giving me the opportunity to work on this nice project. I would in particular like to thank Béla for introducing me into the basics of numerical relativity, assisting me with helpful insights and hints for working on the harmonic evolution equations and answering innumerable questions regarding the development of this code. Special thanks also go to Tilman Vogel, who helped and assisted me in deriving the evolution equations, and answered a number of further questions I had when I started working on this project. Furthermore I thank Marcus Ansorg, Sascha Husa, Anil Zenginoglu, Jennifer Seiler, Nils Dorband, Christian Reisswig, Carsten Schneemann, Christian D. Ott, Luca Baiotti, Bruno Giacomazzo, Thorsten Kellermann and Erik Schnetter for fruitful discussions and further insights into mathematical, physical or technical issues. I would like to thank Thomas Radke for his support and help with the *Cactus* computational toolkit and further computer-related questions. Additionally I would like to say special thanks to my supervisors Luciano Rezzolla and Burkhard Fricke. I would like to thank my parents and my family for supporting me during my studies and especially during the time of this thesis. Finally I would like to thank a few very special friends for their support during this time.

Thank you all.

Philipp Mösta

2 The Einstein equations in Numerical Relativity

In this section I will introduce the basic approaches used to evolve Einstein's equations. First, I will give a brief overview of 3+1 spacetime decomposition and the ADM equations commonly used in Numerical Relativity which closely follows [1]. After that I will discuss further a special class of evolution systems, namely the Generalized Harmonic Evolution systems. I will first describe the Densitized Harmonic formulation and then present the Friedrich-Rendall evolution equations on which the 'ADM Harmonic' Code is based. Furthermore I will point out the gauge choices and constraint adjustments I use for the Friedrich-Rendall system.

2.1 Foliation of spacetime

Einstein's Equations unify space and time into the four-dimensional framework of spacetime. But for the majority of astrophysically interesting problems like binary black-hole mergers one cannot find analytical solutions to these equations due to the complexity involved. In order to still tackle these questions one splits Einstein's equations into space- and timelike parts and by that recasts the equations into a "3+1" formulation. One can then evolve initial data given on an initial spatial slice along a timelike vector field. This splitting of Einstein equations is called the Arnowitt-Deser-Misner (ADM) decomposition. A four dimensional space-time $(M, g_{\mu\nu})$ is foliated into three-dimensional non-intersecting spacelike hypersurfaces Σ of a scalar function t which can later be identified with the time coordinate. The notation used here is the following: Greek indices range over 0...3 whereas spatial components 1...3 are labeled with Latin indices.

The 1-form

$$\Omega = dt \tag{2.1.1}$$

is closed ($d\Omega = 0$) and has the norm

$$|\Omega|^2 = g^{\mu\nu}\nabla_\mu t \nabla_\nu t =: -\alpha^2, \tag{2.1.2}$$

here α is the **lapse function** (strictly positive). This makes sure that the hypersurfaces Σ are spacelike. Now the timelike unit normal vector n^μ is defined as

$$n^\mu \equiv -\alpha g^{\mu\nu}\Omega_\nu = -\alpha g^{\mu\nu}\nabla_\nu t. \tag{2.1.3}$$

n^μ points in the direction of increasing t and by the four-dimensional metric $g_{\mu\nu}$ a spatial metric

$$h_{\mu\nu} \equiv g_{\mu\nu} + n_\mu n_\nu \tag{2.1.4}$$

is induced on the hypersurfaces Σ. By that any four dimensional tensor can be decomposed into spatial parts, which live on the hypersurfaces Σ, and timlike parts, which are normal to Σ.

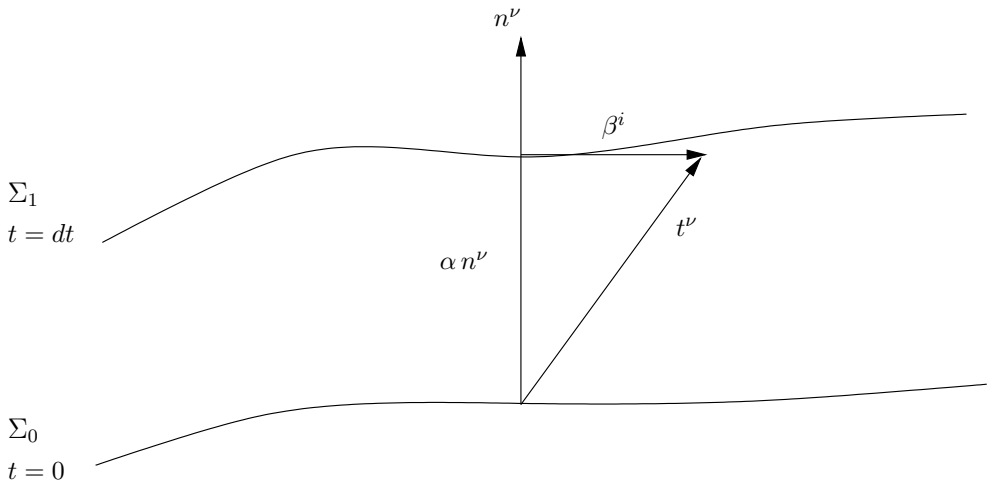


Figure 2.1: The foliation of spacetime into spacelike hypersurfaces in the 3+1 decomposition.

Using the contraction operator

$$h^\mu{}_\nu = g^{\mu\sigma} h_{\sigma\nu} = g^\mu{}_\nu = \delta^\mu{}_\nu + n^\mu n_\nu \tag{2.1.5}$$

the spatial part can be found and the timelike part is obtained by contracting with

$$N^\mu{}_\nu = -n^\mu n_\nu. \tag{2.1.6}$$

The next step is to define the three-dimensional covariant derivative operator acting on a spatial tensor, by projecting all indices of a four-dimensional covariant derivative into Σ:

$$D_\sigma T^\mu{}_\nu \equiv h_\sigma{}^\delta h_\epsilon{}^\mu h_\nu{}^\rho \nabla_\delta T^\epsilon{}_\rho. \tag{2.1.7}$$

It can be easily shown that this definition of the derivative is compatible with the spatial metric,

$$D_\sigma h_{\mu\nu} = 0. \tag{2.1.8}$$

This three-dimensional covariant derivative can also be expressed in terms of the three-dimensional connection coefficients of the spatial metric, given by the following expression (with respect to a coordinate basis):

$$\Gamma^\sigma{}_{\mu\nu} = \frac{1}{2} h^{\sigma\delta} \left(h_{\delta\mu,\nu} + h_{\delta\nu,\mu} - h_{\mu\nu,\delta} \right) \tag{2.1.9}$$

Now the three-dimensional Riemann tensor associated with h_{ij} is defined by requiring that

$$2 D_\mu D_\nu \omega_\sigma = R^\delta{}_{\sigma\nu\mu} \omega_\delta, \qquad R^\delta{}_{\sigma\nu\mu} n_\delta = 0 \tag{2.1.10}$$

for every spatial ω_δ. In respect to a specific coordinate basis, the Riemann tensor can also be obtained from

$$R_{\mu\nu\sigma}{}^\delta = \Gamma^\delta{}_{\mu\sigma,\nu} - \Gamma^\delta{}_{\sigma\nu,\mu} + \Gamma^\epsilon{}_{\mu\sigma} \Gamma^\delta{}_{\epsilon\nu} - \Gamma^\epsilon{}_{\nu\sigma} \Gamma^\delta{}_{\epsilon\mu}. \tag{2.1.11}$$

As in the four-dimensional case contracting the three-dimensional Riemann tensor gives the three-dimensional Ricci-tensor

$$R_{\mu\nu} = R^\sigma{}_{\mu\sigma\nu} \tag{2.1.12}$$

and one more contraction yields the three-dimensional Ricci scalar

$$R = R^{\mu}{}_{\mu}. \tag{2.1.13}$$

The four-dimensional Riemann tensor $R^{\delta}{}_{\mu\nu\sigma}$ contains information about all the derivatives of the four-dimensional metric. But $R^{\delta}{}_{\mu\nu\sigma}$ in contrast is a purely spatial object since it is obtained from spatial derivatives of the metric only. From that it is clear that $R^{\delta}{}_{\mu\nu\sigma}$ can only describe the curvature intrinsic to the slice Σ. In order to correctly depict all the information included in $R^{\delta}{}_{\mu\nu\sigma}$ we need to introduce another tensor called **extrinsic curvature**. This tensor then describes how the slice Σ is embedded in the spacetime M. The extrinsic curvature can be defined as

$$K_{\mu\nu} \equiv -h_{\mu}{}^{\sigma} h_{\nu}{}^{\rho} \nabla_{(\sigma} n_{\rho)}. \tag{2.1.14}$$

$K_{\mu\nu}$ is spatial and symmetric by construction and can alternatively be expressed in terms of "acceleration" of normal observers $a_{\mu} = n^{\nu}\nabla_{\nu} n^{\mu}$,

$$K_{\mu\nu} = -\nabla_{\mu} n_{\nu} - n_{\mu} n_{\nu}. \tag{2.1.15}$$

Using $a^{\mu} n_{\mu} = 0$ we find the trace of the extrinsic curvature to be

$$K \equiv g^{\mu\nu} K_{\mu\nu} = -\nabla^{\mu} n_{\mu} \tag{2.1.16}$$

Additionally the extrinsic curvature can be expressed in terms of the Lie derivative[1] of the spatial metric along the normal vector n^{ν}

$$K_{\mu\nu} = -\frac{1}{2} \mathcal{L}_n h_{\mu\nu} \tag{2.1.18}$$

Equation (2.1.18) illustrates more clearly the meaning associated with the extrinsic curvature. It is an expression for the change of the spatial metric along the normal vector field, which we can later identify as the flow of time. So basically $K_{\mu\nu}$ is the time derivative of the spatial metric $g_{\mu\nu}$. Taking its symmetry properties into account $R^{\delta}{}_{\mu\nu\sigma}$ can be projected in three different ways. These projections then lead to the three equations the ADM-derivation is based on. Projecting all four indices into Σ leads (after some algebraic manipulations) to **Gauss's equation**,

$$R_{\mu\nu\sigma\rho} + K_{\mu\sigma} K_{\nu\rho} - K_{\mu\rho} K_{\nu\sigma} = \gamma^{\epsilon}{}_{\mu} \gamma^{\kappa}{}_{\nu} \gamma^{\xi}{}_{\sigma} \gamma^{\chi}{}_{\rho} R_{\epsilon\kappa\xi\chi}. \tag{2.1.19}$$

Contracting three indices spatially and one with n^{μ} yields the **Codazzi equation**

$$D_{\mu} K_{\nu\sigma} - D_{\nu} K_{\mu\sigma} = \gamma^{\epsilon}{}_{\nu} \gamma^{\kappa}{}_{\mu} \gamma^{\xi}{}_{\sigma} n^{\chi} R_{\epsilon\kappa\xi\chi}. \tag{2.1.20}$$

[1] The Lie derivative is naturally defined by the manifold structure and differentiates a given tensor field along a given vector field. It can be defined as follows:

Definition 2.1.1. *The **Lie derivative** of a tensor field* $\mathbf{T}$ *with respect to* $\mathbf{X}$ *is [?], [?]*

$$\mathcal{L}_{\mathbf{X}} \mathbf{T}|_p = \lim_{t \to 0} \frac{1}{t} \left(\mathbf{T}|_p - (\phi_t)_* \mathbf{T}|_p \right) \tag{2.1.17}$$

where $\phi_t : M \to M$ *is a local one-parameter group of diffeomorphisms (also called a 'flow') mapping each tensor field* $\mathbf{T}$ *at point* p *of the manifold* M *into* $(\phi_t)_* \mathbf{T}|_{\phi_t(p)}$*. A flow is induced by the integral curve of* $\mathbf{X}$ *at* p *and satisfies* $\phi_0(p) = x_0^{\mu}$ *with* x_0^{μ} *being the coordinate representation of the integral curve* $x(t = 0)$ *of* $\mathbf{X}$ *through* p*.*

And finally two spatial contractions and two contractions with n^μ lead to **Ricci's equation**

$$\mathcal{L}_n K_{\mu\nu} = \gamma^\epsilon{}_\nu \gamma^\xi{}_\mu n^\chi n^\kappa R_{\epsilon\kappa\xi\chi} - \frac{1}{\alpha} D_\mu D_\nu \alpha - K^\sigma{}_\nu K_{\mu\sigma}, \tag{2.1.21}$$

where the derivative of the lapse function entered through the identity

$$a_\mu = D_\mu \ln\alpha. \tag{2.1.22}$$

Another useful relation that is later on used is

$$D_\mu V^\nu = \gamma_\mu{}^\rho \nabla_\rho V^\nu + K_{\mu\nu} V^\rho n^\nu, \tag{2.1.23}$$

which is true for every spatial vector V^μ.

2.2 The ADM equations

In this section we will concentrate on the 3+1 decomposition of Einstein's equations using the identities obtained in the previous section. In general relativity the geometry of the four-dimensional spacetime is governed dynamically by Einstein's equations

$$G_{\mu\nu} \equiv R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} = 8\pi T_{\mu\nu}, \tag{2.2.1}$$

where $G_{\mu\nu}$ is the Einstein tensor and $T_{\mu\nu}$ the stress-energy tensor. To obtain the ADM equations we will take projections of (2.2.1) into Σ and n^μ and use the Gauss, Codazzi and Ricci equations to eliminate the four-dimensional Ricci tensor $R_{\mu\nu}$. The ADM equations then will relate three-dimensional curvature quantities to projections of the stress-energy tensor. We will obtain two sets of equations: one determining the properties of $g_{\mu\nu}$ and $K_{\mu\nu}$ on the slices Σ itself and one set governing the dynamics of these quantities. To obtain the **Hamiltonian constraint** we contract Gauss's equation (2.1.19) twice with the spatial metric and insert (2.1.4) into it

$$2 n^\mu n^\nu \, G_{\mu\nu} = R + K^2 - K_{\mu\nu} K^{\mu\nu}. \tag{2.2.2}$$

By now defining the total energy density as measured by a normal observer n^μ as

$$\rho := n^\mu n^\nu T_{\mu\nu} \tag{2.2.3}$$

and, using Einstein's equations (2.2.1), we finally get

$$R + K^2 - K_{\mu\nu} K^{\mu\nu} = 16\pi\rho. \tag{2.2.4}$$

When contracting the Codazzi equation in a similar way once we get the **Momentum constraint**

$$D_\nu K^\nu{}_\mu - D_\mu K = 8\pi j_\mu, \tag{2.2.5}$$

where

$$j_\mu = -\gamma^\nu{}_\mu n^\rho T_{\nu\rho} \tag{2.2.6}$$

is the momentum density as measured by a normal observer n^μ. These constraint equations involve spatial quantities and their spatial derivatives only and so determine $h_{\mu\nu}$ and $K_{\mu\nu}$ on

each and every timeslice Σ. In fact they formulate the necessary integrability conditions for the embedding of the spatial slices $(\Sigma, h_{\mu\nu}, K_{\mu\nu})$ in the spacetime $(M, g_{\mu\nu})$ itself. In order to obtain evolution equations describing how initial data $(h_{\mu\nu}, K_{\mu\nu})$ evolves from one time slice to another, one combines (2.1.19) with Ricci's equation (2.1.21). But first we have to note that the Lie derivative along n^μ is not a natural time derivative since by evaluating the scalar product

$$n^\mu \Omega_\mu = -\alpha h^{\mu\nu} \nabla_\nu t \nabla_\mu t = -\alpha^{-1} \tag{2.2.7}$$

we find that n^μ is not dual to Ω. But by introducing the vector

$$t^\mu = \alpha n^\mu + \beta^\mu \tag{2.2.8}$$

we find that t^μ actually is dual to Ω for any spatial **shift vector** β^μ, since reevaluating the scalar product yields

$$t^\mu \Omega_\mu = t^\mu \nabla_\mu t = -1. \tag{2.2.9}$$

This implies that the Lie derivative along t^μ actually is a natural time derivative because integral curves $x(t)$ of t^μ are identical to $x(t) \equiv t$ and hence directly parametized by t. Furthermore the scalar product being independent of α assures all vectors t^μ originating on one time slice Σ_1 ending up on the same time slice Σ_2. Additionally the Lie derivative of any spatial tensor along t^μ remains spatial. Now the lapse α and the shift β^μ together determine the time evolution of the coordinates from one time slice Σ to the next one. The lapse function prescribes the proper time elapsed between time slices along the unit normal vector n^μ, while the shift vector on the other hand determines by how much the spatial coordinates are shifted in respect to the unit normal vector (Fig. 2.1). Now rewriting (2.1.14) using this new vector t^μ we obtain

$$\mathcal{L}_t h_{\mu\nu} = -2\alpha K_{\mu\nu} + \mathcal{L}_\beta h_{\mu\nu}, \tag{2.2.10}$$

the evolution equation for the spatial metric $h_{\mu\nu}$. To find the evolution equation for the extrinsic curvature $K_{\mu\nu}$ we combine Ricci's equation (2.1.21) together with Einstein's equation (2.2.1)

$$\begin{aligned} \mathcal{L}_t K_{\mu\nu} &= -D_\mu D_\nu \alpha + \alpha(R_{\mu\nu} - 2K_{\mu\rho}K^\rho{}_\nu + KK_{\mu\nu}) \\ &\quad -\alpha 8\pi \left(S_{\mu\nu} - \frac{1}{2} h_{\mu\nu}(S - \rho) \right) + \mathcal{L}_\beta K_{\mu\nu}, \end{aligned} \tag{2.2.11}$$

where

$$S_{\mu\nu} = h_{\mu\rho} h_{\nu\sigma} T^{\rho\sigma} \tag{2.2.12}$$

is the spatial projection of the stress-energy tensor with trace

$$S = h^{\mu\nu} S_{\mu\nu}. \tag{2.2.13}$$

It is now time to introduce a suitable set of coordinates that reflects the 3+1 splitted spacetime. We choose a basis of spatial vectors e_i with $i = 1, 2, 3$ that span each slice Σ so that $\Omega_\mu (e_i)^\mu = 0$. Additionally we introduce a timelike basis vector $(e_0)^\mu = t^\mu$ such that $t^\mu = (1, 0, 0, 0)$. From that it follows immediately that the Lie derivative along t^μ reduces to a partial derivative with respect to the time t. The covariant components of the normal vector n_μ satisfy

$$n_i = n_\mu (e_i)^\mu = \alpha \Omega_\mu (e_i)^\mu = 0. \tag{2.2.14}$$

So zeroth components of contravariant tensors vanish, since contractions of the normal vector with any spatial tensor give zero. It follows for the components of the shift vector

$$\beta^\mu = (0,\, \beta^i). \tag{2.2.15}$$

Using this we obtain the contravariant components of the normal vector as

$$n^\mu = \frac{1}{\alpha}(1, -\beta^i), \tag{2.2.16}$$

and since $n^\mu n_\mu = -1$ the covariant components are

$$n_\mu = (-\alpha, 0, 0, 0). \tag{2.2.17}$$

To obtain a coordinate expression for $h^{\mu\nu}$ we use $h^{ij} = g^{ij}$ which follows directly from the definition (2.1.4) and $h^{\mu 0} = 0$ which is a consequence of (2.2.14)

$$g^{\mu\nu} = \begin{pmatrix} -\alpha^{-2} & \alpha^{-2}\beta^i \\ \alpha^{-2}\beta^j & \gamma^{ij} - \alpha^{-2}\beta^i\beta^j \end{pmatrix}. \tag{2.2.18}$$

The corresponding line element is

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt). \tag{2.2.19}$$

Since $h^{ik}h_{kj} = \delta^i{}_j$ is an isomorphism between the covariant and contravariant spatial tensor bundle it serves as an index raising and lowering operator for spatial components. Now all the information of a spatial tensor is encoded in its spatial components and we can therefore express the constraint and evolution equations in spatial components only. Furthermore, since in our coordinate system all zeroth components of contravariant, spatial tensors are zero, we can restrict all contractions to spatial components. This yields for the connection coefficients

$$\Gamma^i_{jk} = \frac{1}{2}h^{il}\left(h_{lj,k} + h_{lk,j} - h_{jk,l}\right), \tag{2.2.20}$$

and consequently the Ricci tensor is expressed

$$\begin{aligned} R_{ij} &= \frac{1}{2}h^{kl}\left(h_{kj,il} + h_{il,kj} - h_{kl,ij} - h_{ij,kl}\right) \\ &\quad + h^{kl}\left(\Gamma^m_{il}\Gamma_{mkj} - \Gamma^m_{ij}\Gamma_{mkl}\right). \end{aligned} \tag{2.2.21}$$

Finally we can now re-express the **ADM equations** in their simplified form: First the constraint equations read

$$R + K^2 - K_{ij}K^{ij} = 16\pi\rho \qquad \text{Hamiltonian constraint} \tag{2.2.22}$$

$$D_j K^j{}_i - D_i K = 8\pi j_i \qquad \text{momentum constraint.} \tag{2.2.23}$$

And second the evolution equations for h_{ij} and K_{ij} become

$$\partial_t h_{ij} = -2\alpha K_{ij} + D_i\beta_j + D_j\beta_i, \tag{2.2.24}$$

$$\begin{aligned}\partial_t K_{ij} &= -D_i D_j \alpha + \alpha(R_{ij} - 2K_{ik}K^k{}_j + KK_{ij}) \\ &\quad -\alpha 8\pi(S_{ij} - \frac{1}{2}h_{ij}(S-\rho)) \\ &\quad +\beta^k D_k K_{ij} + K_{ik}D_j\beta^k + K_{kj}D_i\beta^k. \end{aligned} \tag{2.2.25}$$

In these equations lapse α and shift β^i are gauges. However lapse and shift determine how the coordinates evolve from one slice Σ to another and special care must be taken to choose a gauge that allows well-behaved and stable simulations in numerical relativity. For example we can choose lapse profiles which effectly prevent spatial slices from touching singularities present in black-hole spacetimes. In general it is crucial to successful simulations to choose appropriate gauges for the problem. However the standard ADM formulation is not very well suited for numerical relativity since it is only weakly hyperbolic and therefore not a well-posed problem. Strong hyperbolicity is a desired feature, since existing mathematical theorems guarantee local existence and uniqueness of bounded solutions that depend continuously on initial data (well-posedness). In addition to that, stability theorems for discretized problems could be applied, assuring stable evolution schemes. Therefore I will in the following section discuss in detail another class of evolution systems, the Generalized Harmonic evolution systems.

2.3 Generalized Harmonic equations

Harmonic and generalized harmonic coordinates have from the very beginning played a key role in understanding general relativity. Einstein himself already used harmonic coordinates in analyzing different candidates for 'his' theory of gravitation. Later on deDonder [22] then used them to analyze the structure of general relativity and further extensive development by Fock [23]and Choquet-Bruhat [24] gave the first well-posed version of the Cauchy problem for the Einstein equations. The idea of generalizing harmonic coordinates by using arbitrary coordinate functions, was first introduced by Friedrich[33]. Recently generalized harmonic coordinates were used in the first successful numerical simulations of the final inspiral and merger phase of binary black holes by Pretorius[30]. Generalized harmonic spacetime coordinates x^μ satisfy the curved space wave equation

$$\Gamma^\mu = \Box x^\mu = \nabla_\nu \nabla^\nu x^\mu = F^\nu(x^\mu), \tag{2.3.1}$$

where $\Gamma^\mu = g^{\sigma\nu}\Gamma_{\mu\nu\sigma}$ is the trace of the Christoffel symbol related to the metric $g_{\mu\nu}$

$$\Gamma_{\nu\sigma}{}^\mu = \frac{1}{2}(\partial_\nu g_{\mu\sigma} + \partial_\sigma g_{\mu\nu} - \partial_\mu g_{\nu\sigma}). \tag{2.3.2}$$

The Ricci curvature tensor can in any coordinate system be written as

$$R_{\mu\nu} = -\frac{1}{2}g^{\sigma\rho}\partial_\sigma\partial_\rho g_{\mu\nu} + \nabla_{(\mu}\Gamma_{\nu)} + g^{\sigma\rho}g^{\delta\epsilon}(\partial_\delta g_{\sigma\mu}\partial_\epsilon g_{\rho\nu} - \Gamma_{\mu\sigma\delta}\Gamma_{\nu\rho\epsilon}). \tag{2.3.3}$$

In vacuum now Einsteins equations $R_{\mu\nu} = 0$ can be written in harmonic coordinates as

$$g^{\sigma\rho}\partial_\sigma\partial_\rho g_{\mu\nu} = 2g^{\sigma\rho}g^{\delta\epsilon}(\partial_\delta g_{\sigma\mu}\partial_\epsilon g_{\rho\nu} - \Gamma_{\mu\sigma\delta}\Gamma_{\nu\rho\epsilon}). \tag{2.3.4}$$

So choosing harmonic coordinates or generalized harmonic coordinates we can cast the principal part of the Einstein equations into a second-order hyperbolic form or a first-order symmetric hyperbolic form. In the following sections I will in detail describe two evolution systems formulated in generalized harmonic coordinates, the 'Densitized Harmonic' [38] and the 'Friedrich-Rendall' evolution system [34].

2.4 The 'Densitized Harmonic' evolution system

This section closely follows [38] and describes the 'Densitized Harmonic' evolution system. As noted in the previous section in a generalized harmonic gauge [33], the coordinates $x^\mu = (t, x^i) = (t, x, y, z,)$ satisfy

$$-\delta_a\delta^a x^\mu = \Gamma^\mu = F^\mu \ , \tag{2.4.1}$$

where

$$\Gamma^\mu := g^{\rho\sigma}\Gamma^\mu_{\rho\sigma} = -\frac{1}{\sqrt{-g}}\partial_\nu\tilde{g}^{\mu\nu} \ , \tag{2.4.2}$$

with *gauge source functions* $F^\mu(x^\rho, g^{\rho\sigma})$ (which may depend on the spacetime coordinates and the metric) and with the Densitized 4-metric

$$\tilde{g}^{\mu\nu} := \sqrt{-g}g^{\mu\nu} \tag{2.4.3}$$

playing the role of the basic evolution variable. In this harmonic formulation, the constraints reduce to the gauge condition

$$C^\mu := \Gamma^\mu - F^\mu = 0 \ , \tag{2.4.4}$$

and the evolution system is based upon the reduced Einstein tensor

$$E^{\mu\nu} := G^{\mu\nu} - \nabla^{(\mu}\Gamma^{\nu)} + \frac{1}{2}g^{\mu\nu}\nabla_\alpha\Gamma^\alpha \ . \tag{2.4.5}$$

Here Γ^ν is treated formally as a vector in constructing the "covariant" derivative $\nabla^\mu\Gamma^\nu$. When the constraints (2.4.4) are satisfied, this gives rise to a hyperbolic evolution system

$$E^{\mu\nu} = -\nabla^{(\mu}F^{\nu)} + \frac{1}{2}g^{\mu\nu}\nabla_\rho F^\rho. \tag{2.4.6}$$

Provided the gauge source functions do not depend upon derivatives of the metric, they do not enter the principle part of the system and do not affect its well-posedness or numerical stability. The evolution system (2.4.6) takes the specific form of quasi-linear wave equations

$$\begin{aligned}\partial_\rho\left(g^{\rho\sigma}\partial_\sigma\tilde{g}^{\mu\nu}\right) - 2\sqrt{-g}g^{\rho\sigma}g^{\tau\lambda}\Gamma^\mu_{\rho\tau}\Gamma^\nu_{\sigma\lambda} - \sqrt{-g}(\partial_\rho g^{\rho\sigma})(\partial_\sigma g^{\mu\nu}) + \tfrac{g^{\rho\sigma}}{\sqrt{-g}}(\partial_\rho g^{\mu\nu})(\partial_\sigma g)\\ +\tfrac{1}{2}g^{\mu\nu}\left(\tfrac{g^{\rho\sigma}}{2g\sqrt{-g}}(\partial_\rho g)(\partial_\sigma g) + \sqrt{-g}\Gamma^\tau_{\rho\sigma}\partial_\tau g^{\rho\sigma} + \tfrac{1}{\sqrt{-g}}(\partial_\sigma g)\partial_\rho g^{\rho\sigma}\right)\\ +2\sqrt{-g}\delta^{(\mu}F^{\nu)} - \sqrt{-g}g^{\mu\nu}\delta_\rho F^\rho + \sqrt{-g}A^{\mu\nu} = 0 \ ,\end{aligned} \tag{2.4.7}$$

where we have included the possibility of a constraint-adjustment term

$$A^{\mu\nu} := C^\rho A^{\mu\nu}_\rho(x^\rho, g_{\rho\sigma}, \partial_\tau g_{\rho\sigma}) \ , \tag{2.4.8}$$

i.e., a term which vanishes when the constraints are satisfied and which does not affect the principle part of the evolution system. In this system the constraints (2.4.4) are not explicitly enforced during the evolution. Instead, we invoke the Bianchi identities which imply wave equations of the homogeneous form

$$g^{\rho\sigma}\partial_\rho\partial_\sigma C^\mu + L^{\mu\rho}_\sigma\partial_\rho C^\sigma + M^\mu_\sigma C^\sigma = 0 \ , \tag{2.4.9}$$

where the matrices $\boldsymbol{L}$ and $\boldsymbol{M}$ are functions of the metric and its first and second derivatives.

Given constraint-preserving initial and boundary conditions, the uniqueness of the solutions to (2.4.9) guarantees that the harmonic constraints be conserved during the evolution. On the other hand, constraint-preserving initial data also requires that the initial Cauchy data satisfy the standard Hamiltonian and momentum constraints. Also, since the harmonic constraints imply evolution equations for the lapse and shift, the only remaining free initial data in addition to the usual Cauchy data (the 3-metric and extrinsic curvature of the Cauchy hypersurface) are the initial choices of lapse and shift and of the gauge source functions. Note that an initial choice of the gauge source functions is effectively equivalent to a choice in the initial evolution of the lapse and shift.

2.4.1 Constraint Adjustment and Damping

The constraint-adjustments used in this evolution system are those investigated by Babiuc[29] and have the general form

$$A^{\mu\nu} := -\frac{a_1}{\sqrt{-g}} C^\rho \partial_\rho \tilde{g}^{\mu\nu} + \frac{a_2 C^\rho \delta_\rho t}{\varepsilon + e_{\sigma\tau} C^\sigma C^\tau} C^\mu C^\nu - \frac{a_3}{\sqrt{-g^{tt}}} C^{(\mu} \delta^{\nu)} t \ , \tag{2.4.10}$$

where the $a_i > 0$ are adjustable parameters, $e_{\sigma\tau}$ is the natural metric of signature $(++++)$ associated with the Cauchy slicing and ε is a small positive number chosen to ensure numerical regularity. The effects of these adjustments in suppressing long wavelength instabilities in standardized tests for periodic boundary conditions have been discussed by Babiuc[29].
In particular, the first and second terms in the adjustments (2.4.10) have been shown to be effective in suppressing constraint-violating nonlinear instabilities in shifted gauge-wave tests. The third term in (2.4.10), on the other hand, was first considered in [35] and leads to constraint damping in the linear regime. Although it has been used effectively by Pretorius [21, 30] in black-hole simulations, it was not effective in the nonlinear regime of the shifted gauge-wave test [29].
We also note that the work reported in [29] has shown that adjustments which scale quadratically with C^μ (or with higher powers) take effect too late to counter the growth of a constraint-violating instability; and this has led to the specific form for the denominator of the second term in (2.4.10).

2.4.2 Gauge Conditions

As noted earlier, the gauge source functions F^μ may be chosen to be arbitrary functions of the spacetime coordinates and metric. They can be viewed as differential gauge conditions on the densitized metric. This serves two important purposes. Firstly, it allows for convergence tests based upon a known spacetime, whose analytic metric $g^{\mu\nu}_{(0)}(x^\rho)$ is specified in a non-harmonic gauge, by choosing

$$F^\mu = -\frac{1}{\sqrt{-g_{(0)}}} \, \partial_\nu \tilde{g}^{\mu\nu}_{(0)} \ . \tag{2.4.11}$$

Using these analytic gauge source functions, in combination with initial and boundary data consistent with the analytic solution, gives rise to numerically evolved spacetimes that are identical to the analytic solution up to discretization error. This is how the convergence tests reported here have been carried out for the Schwarzschild spacetime expressed in (non-harmonic) Kerr-Schild coordinates. Secondly, and most importantly, the gauge source functions can be used to avoid gauge pathologies.

A major restriction for the form of the gauge source functions is that they cannot depend on the derivatives of the metric. In particular, they cannot depend on the location or shape of the MOTS and this is a problem when moving black holes are present, and where it is important for the coordinates to be able to "respond" to the black hole motion. In our simulation of binary black holes, we have used the gauge source function

$$F^\mu = \frac{\omega}{\sqrt{-g}} \, (\tilde{g}^{t\mu} - \eta^{t\mu}) \ , \tag{2.4.12}$$

where $\eta^{\mu\nu}$ is the Minkowski metric and where $\omega = \omega(x^i)$ is a smooth, spherically symmetric, time-independent weighting function with $\omega = 1$ over most of the computational domain, but with $\omega = 0$ in some neighborhood of the outer boundary. When spatial derivatives are neglected and $\omega = 1$, the resulting gauge condition takes the simpler form

$$\partial_t \tilde{g}^{t\mu} = -(\tilde{g}^{t\mu} - \eta^{t\nu}) \,, \tag{2.4.13}$$

showing that it forces the densitized lapse and shift to relax to their Minkowski values.
In our first attempts at binary black hole simulations, we have found that this choice of gauge source function keeps the lapse and shift under reasonable control. Similar choices of gauge source functions have been used with success in other binary black hole simulations [21, 30].

2.4.3 Blended subluminal-superluminal evolution

The evolution system (2.4.7) consists of coupled quasi-linear wave equations whose numerical stability is determined by the principle part. By the principle of frozen coefficients [45], the stability analysis can be reduced to a consideration of the wave equation with shift. Although finite difference approximations to the wave equation is a well studied problem, the complications introduced by a non-zero shift are peculiar to the black hole excision problem. This was first recognized in [46], where it was suggested that the superluminal shift introduced by tracking the excision boundary could be treated by implicit methods.

Subsequent studies established the stability of explicit finite-difference algorithms, with second order accuracy, for the case of superluminal evolution. This was first achieved for the 1D wave equation with shift

$$g^{tt}\partial_t^2\Phi + 2\partial_t\partial_x g^{xt}\Phi + g^{xx}\partial_x^2\Phi = 0, \tag{2.4.14}$$

in work by Calabrese [36] and Szilágyi [37]. The standard choice of energy for this system,

$$E^{(t)} = \frac{1}{2}\int dx \left[(-g^{tt}(\partial_t\Phi)^2 + g^{xx}(\partial_x\Phi)^2\right] , \tag{2.4.15}$$

gives rise to a norm when the evolution direction ∂_t is timelike. In that subluminal case, summation by parts (SBP) can be used to establish stability of the semi-discrete approximation

$$g^{tt}\partial_t^2\Phi + 2g^{xt}D_0\partial_t\Phi + g^{xx}D_+D_-\Phi = 0, \tag{2.4.16}$$

where D_+, D_- and D_0 are, respectively, the standard forward, backward and centered finite difference approximations for ∂_x. This ensures that the numerical error is controlled by an estimate for the semi-discrete version of the energy norm $E^{(t)}$. For most 'Method of Lines' time integrators, e.g. Runge-Kutta, this estimate extends to the fully discretized system. The algorithm (2.4.16) has been extended to the 3D subluminal case to give a stable SBP boundary treatment [27].

However, the algorithm (2.4.16) is unstable (and cannot be stabilized by Kreiss-Oliger type dissipation) when the evolution is superluminal, *i.e.*, when the shift is large enough so that ∂_t is spacelike and

$$g^{xx} = h^{xx} + \frac{(g^{xt})^2}{g^{tt}} < 0, \tag{2.4.17}$$

where $h^{xx} > 0$ is the inverse to the spatial metric of the $t =$ const Cauchy hypersurfaces. In that case, when the energy $E^{(t)}$ is no longer a norm, stability can be based upon the positive energy associated with the timelike normal n^μ to the Cauchy hypersurfaces,

$$E^{(n)} = \frac{1}{2}\int dx \left[-\frac{1}{g^{tt}}(g^{tt}\partial_t\Phi + g^{tx}\partial_x\Phi)^2 + h^{xx}\partial_x^2\Phi\right]. \tag{2.4.18}$$

As discussed in the 1D case [36, 37], the discretization

$$g^{tt}(\partial_t + \frac{g^{xt}}{g^{tt}}D_0)^2\Phi + h^{xx}D_+D_-\Phi = 0. \tag{2.4.19}$$

yields a stable second-order accurate superluminal algorithm. Stable superluminal evolution algorithms for the 3D case have been given by Motamed [40], where the global stability of a model black hole excision problem is treated.

Although a stable boundary treatment for the superluminal algorithm (2.4.19) has been proposed [39], its extended stencil (due to the D_0^2 operator) makes this complicated and an SBP boundary version has not yet been formulated. For this reason we use the 3D version of the subluminal algorithm (2.4.16) in the neighborhood of the outer boundary and blend it to the superluminal algorithm (2.4.19) by introducing the vector

$$\hat{n}^\mu = (g^{tt}, wg^{it}) \tag{2.4.20}$$

and the evolution variable

$$\hat{Q} = \hat{n}^\mu\partial_\mu\Phi, \tag{2.4.21}$$

where $w(x^i)$ is a spherically symmetric smooth blending function, with $w = 0$ near the outer boundary and $w = 1$ (so that $\hat{n}^\mu = n^\mu$) in the interior. It suffices to discuss the frozen coefficient case in which the 1D wave equation (2.4.14) gives rise to the evolution system for $\hat{Q}$ and Φ,

$$\begin{aligned} g^{tt}\partial_t\hat{Q} &= -(2g^{xt} - \hat{n}^x)\partial_x(Q - \hat{n}^x\partial_x\Phi) - g^{tt}g^{xx}\partial_x^2\Phi \\ g^{tt}\partial_t\Phi &= \hat{Q} - \hat{n}^x\partial_x\Phi. \end{aligned} \tag{2.4.22}$$

Note that introduction of the auxiliary variable $\hat{Q}$, which reduces the system to first-order in time, introduces no associated constraints.

For a second-order accurate approximation, we discretize (2.4.22) according to

$$\begin{aligned} g^{tt}\partial_t\hat{Q} = -(2g^{xt} - \hat{n}^x)D_0Q + (2g^{xt} - \hat{n}^x)\hat{n}^xD_+D_-\Phi - g^{tt}g^{xx}D_+D_-\Phi \\ g^{tt}\partial_t\Phi = \hat{Q} - \hat{n}^xD_0\Phi. \end{aligned} \tag{2.4.23}$$

In the neighborhood of the outer boundary, this reduces to the subluminal algorithm (2.4.16) and in the interior where $w = 1$ it reduces to the superluminal algorithm (2.4.19). The harmonic code uses a fourth-order accurate version of (2.4.23) in the interior region. An alternative scheme for switching between stable subluminal and superluminal algorithms across the "artificial horizon" where $\det(g^{ij}) = 0$ is given in [40].

2.5 The 'ADM Harmonic' evolution system

Defining χ by

$$\chi_{\mu\nu} = \frac{1}{2}\mathcal{L}_n h_{\mu\nu}. \tag{2.5.1}$$

we can write the constraint equations on a spacelike hypersurface (2.2.4) and (2.2.5) in a slightly different form

$$0 = Z^H = R - \chi_{ab}\chi^{ab} + (\chi_a{}^a)^2 - 2\lambda - 2\kappa\rho \tag{2.5.2}$$

and

$$0 = Z^M_b = D_a\chi_b{}^a - D_b\chi_a{}^a + \kappa j_b, \tag{2.5.3}$$

with $\rho = n^\mu n^\nu T_{\mu\nu}$ and $j_\mu = -n^\mu h^\rho{}_\nu T_{\mu\rho}$.

Now the evolution equations written in terms of χ_{ab} and not substituting the Lie derivatives $\mathcal{L}_\beta h_{ab}$ and $\mathcal{L}_\beta \chi_{ab}$ yield

$$\partial_t h_{ab} = 2\alpha\chi_{ab} + \mathcal{L}_\beta h_{ab} \tag{2.5.4}$$

$$\begin{aligned}\partial_t\chi_{ab} \;=\; & -\alpha\left(R_{ab} + \chi_c{}^c\chi_{ab} - 2\chi_{ac}\chi_b{}^c\right) \\ & + D_aD_b\alpha + \mathcal{L}_\beta\chi_{ab} \\ & + \alpha\left(\lambda h_{ab} + \kappa\left(T_{ab} - \frac{1}{2}T h_{ab}\right)\right).\end{aligned} \tag{2.5.5}$$

Solving (2.5.4) for χ_{ab} and inserting it into (2.5.5) leads to a second order system for h_{ab} of the form

$$\begin{aligned}\tfrac{1}{\alpha^2}\quad & \left[\partial_t^2 - \beta^c\partial_c\partial_t - \beta^c\partial_t\partial_c + \beta^c\beta^d\partial_c\partial_d\right]h_{ab} \\ & - h^{cd}\left(\partial_c\partial_d h_{ab} + \partial_a\partial_b h_{cd} - \partial_a\partial_c h_{bd} - \partial_b\partial_c h_{ad}\right) \\ & - \frac{2}{\alpha}\left[\partial_a\partial_b\alpha - \left(h_{c(a}\partial_{b)}\partial_t\beta^c - \beta^d h_{c(a}\partial_{b)}\partial_d\beta^c\right)\right] \\ & = \text{terms of lower order in } h_{ab}, \alpha, \beta^c.\end{aligned} \tag{2.5.6}$$

While the induced constraint equations on a given hypersurface is unique we have a huge freedom to modify the evolution equations. By using the constraints and choosing specific lapse and shift functions we can try to bring the principal part of (2.5.6) into a suitable form. Assuming S to be some spacelike hypersurface and x^a coordinates on some open subset U of S one can always find coordinates $x^{\nu'}$ on some neighborhood of U in M with $x^{0'} = 0$, $x^{a'} = x^a$ such that the Christoffel coefficients of g in these coordinates satisfy

$$\Gamma^{\mu'}(x^{\nu'}) = F^{\mu'}(x^{\nu'}), \tag{2.5.7}$$

with $\Gamma^{\mu'} = g^{\lambda'\rho'}\Gamma^{\mu'}{}_{\lambda'\rho'}$ and $F^{\mu'} = F^{\mu'}(x^{\nu'})$ being four smooth real functions defined in $\mathbf{R}^4$. We can write the four-dimensional Ricci tensor in the form

$$\begin{aligned}R_{\mu\nu} \;=\; & -\frac{1}{2}g^{\sigma\rho}g_{\mu\nu,\sigma\rho} + \nabla_{(\mu}\Gamma_{\nu)} \\ & + \Gamma^\eta{}_{\lambda\mu}g_{\eta\delta}g^{\lambda\rho}\Gamma^\delta{}_{\rho\nu} + 2\Gamma^\lambda{}_{\delta\eta}g^{\delta\rho}g_{\lambda(\mu}\Gamma^\eta{}_{\nu)\rho},\end{aligned} \tag{2.5.8}$$

where

$$\Gamma_\nu = g_{\mu\nu}\Gamma^\mu \qquad and \qquad \nabla_\mu \Gamma_\nu = \partial_\mu \Gamma_\nu - \Gamma^\lambda{}_{\mu\nu}\Gamma_\lambda.$$

If we now consider the Γ_ν as given functions, the Einstein equations form a set of wave equations for lapse, shift and the spatial metric. Writing the four dimensional Christoffel symbols in terms of the 3+1 splitted spacetime quantities we get

$$\partial_t \alpha - \beta^c \partial_c \alpha = \alpha^2 \left(\chi + n_\nu \Gamma^\nu\right), \tag{2.5.9}$$

$$\partial_t \beta^a - \beta^c \partial_c \beta^a = \alpha^2 \left(\gamma^a - D^a \log\alpha - h^a{}_\nu \Gamma^\nu\right). \tag{2.5.10}$$

From these equations we see that the gauge source functions Γ^ν may fix the evolution of lapse and shift. We now want obtain a set of wave equations for lapse α , shift β^c and 3-metric h_{ab}, which will then act as the basic equations of the evolution system: First we use (2.5.4) to get

$$D^2 h_{ab} = 2\alpha D\chi_{ab} + 2\chi_{ab}\alpha^2(\chi + n_\nu\Gamma^\nu) + D\left[h_{c(a}\partial_{b)}\beta^c\right], \tag{2.5.11}$$

with $D \equiv (\partial_t - \beta^c\partial_c)$.

We write the three-dimensional Ricci tensor in terms of the spatial metric and its connection coefficients

$$R_{ab} = -\frac{1}{2}h^{cd}h_{ab,cd} + D_{(a}\gamma_{b)} + \gamma^d{}_{ca}h_{fd}h^{ce}\gamma^f{}_{eb} + 2\gamma^d{}_{ce}h^{cf}h_{d(a}\gamma^e{}_{b)f}, \tag{2.5.12}$$

where

$$\gamma_a = h_{ab}h^{cd}\gamma^b{}_{cd} = h^{cd}(h_{ac,d} - \frac{1}{2}h_{cd,a}). \tag{2.5.13}$$

Using this and (2.5.5) we get

$$\begin{aligned}
\tfrac{1}{2\alpha^2}D^2 h_{ab} \quad & -\alpha\big[\chi_{ab}(n_\nu\Gamma^\nu) - \gamma_{(ab)c}\gamma^c - \chi_a\chi_b + \chi^c\gamma_{(ab)c} - D_{(a}h_{b)\nu}\Gamma^\nu + h^c{}_\nu\Gamma^\nu\gamma_{(ab)c} \\
& +2\chi_{(a}\gamma_{b)} - 2\chi_{(a}h_{b)\nu}\Gamma^\nu + \frac{1}{2}h^{cd}h_{ab,cd} + \gamma^d{}_{ca}h_{fd}h^{ce}\gamma^f{}_{eb} + 2\gamma^d{}_{ce}h^{cf}h_{d(a}\gamma^e{}_{b)f} + 2\chi_{ac}\chi_b{}^c\big] \\
& -\frac{1}{\alpha}\left[2h_{d(a}\beta^c{}_{,b)}\beta^d{}_{,c} + h^{cd}\beta_{a,c}\beta^b{}_{,d}\right] - 4\chi_{c(a}\beta^c{}_{,b)} - \alpha(\lambda h_{ab} + \kappa(T_{ab} - \frac{1}{2}Th_{ab})) \\
& = 0,
\end{aligned} \tag{2.5.14}$$

where $\chi_a \equiv \frac{1}{\alpha}D_a\alpha$.

With

$$-g^{\mu\nu}h_{ab,\mu\nu} = \frac{1}{\alpha^2}D^2 h_{ab} + (\gamma^c - \chi^c - h^c{}_\nu\Gamma^\nu)h_{ab,c} - h^{cd}h_{ab,cd}, \tag{2.5.15}$$

(2.5.9), and again (2.5.4) we finally get

$$\begin{aligned}
-g^{\mu\nu}\partial_\mu\partial_\nu h_{ab} = & \frac{2}{\alpha^2}\chi_{ab}\left(\partial_t\alpha - \mathcal{L}_\beta\alpha\right) + \frac{2}{\alpha^2}D_a\alpha D_b\alpha \\
& - 2D_{(a}\left[h_{b)c}h^c_\nu\Gamma^\nu\right] + \frac{4}{\alpha^3}D_{(a}\alpha h_{b)c}\left(\partial_t\beta^c - \beta^d\partial_d\beta^c\right) \\
& + \frac{4}{\alpha^2}\left(\partial_{(b}\beta^c\right)\partial_t h_{a)c} + \frac{2}{\alpha^2}h_{d(a}\left(\partial_{b)}\beta^c\right)\left(\partial_c\beta^d\right) \\
& - \frac{2}{\alpha^2}\left(\partial_{(b}\beta^c\right)\mathcal{L}_\beta h_{a)c} - \frac{2}{\alpha^2}\left(\partial_{(b}\beta^c\right)\beta^d\left(\partial_d h_{a)c}\right) \\
& + 4\chi_{ac}\chi^c_b - 2\chi_{ab}\chi - 2\gamma^d_{ca}h_{fd}g^{ce}\gamma^f_{eb} - 4\gamma^d_{ce}h^{cf}h_{d(a}\gamma^e_{b)f} \\
& + 2\lambda h_{ab} + 2\kappa\left(T_{ab} - \frac{1}{2}Th_{ab}\right) \\
& =: S_{ab},
\end{aligned} \tag{2.5.16}$$

which is a wave equation for the spatial metric h_{ab}.

Using (2.5.13), (2.5.4) and (2.5.3) we get

$$\begin{aligned}
(\partial_t - \beta^c\partial_c)^2\gamma^a \; & = -\beta^a{}_{,d} + 2\chi^{da}D_d\alpha + h^{de}\beta^a{}_{,de} + \alpha D^a\chi \\
& \quad + 2\alpha\kappa j^a - \chi D^a\alpha - 2\alpha\chi^{ed}\gamma^a{}_{de}.
\end{aligned} \tag{2.5.17}$$

Together with (2.5.9), (2.5.10) and again (2.5.4) this yields the wave equation for the shift vector

$$\begin{aligned}
\frac{1}{\alpha^2}\left(\partial_t - \beta^c\partial_c\right)^2\beta^a - h^{de}\partial_d\partial_e\beta^a = & \; 2\alpha\kappa j^a + 4\left(\chi^{ad} - \chi h^{ad}\right)D_d\alpha \\
& - 2\alpha\left(\chi^{ed} - \chi h^{ed}\right)\gamma^a_{de} - \left(\partial_d\beta^a\right)\gamma^d \\
& + \left(\partial_d\beta^a\right)D^d\log\alpha \\
& + 2\alpha n_\nu\Gamma^\nu\left[\gamma^a - D^a\log\alpha - h^a_\nu\Gamma^\nu\right] \\
& - 2\alpha\chi h^a_\nu\Gamma^\nu - D^a\left(\alpha n_\nu\Gamma^\nu\right) \\
& - \left(\partial_t - \beta^c\partial_c\right)\left(h^a_\nu\Gamma^\nu\right) \\
& =: S^a - h^{de}\partial_d\partial_e\beta^a.
\end{aligned} \tag{2.5.18}$$

Combining (2.5.1) and (2.5.10) yields

$$\begin{aligned}
\tfrac{1}{\alpha^2}(\partial_t - \beta^c\partial_c)^2\alpha \; & = (\partial_t - \beta^c\partial_c)\chi + 2\alpha\chi^2 + 4\alpha\chi n_\nu\Gamma^\nu \\
& \quad + 2\alpha(n_\nu\Gamma^\nu)^2 + \alpha(\partial_t - \beta^c\partial_c)(n_\nu\Gamma^\nu).
\end{aligned} \tag{2.5.19}$$

Now taking the trace of (2.5.5) and using (2.5.4) together with (2.5.3) we furthermore get

$$(\partial_t - \beta^c\partial_c)\chi = D_aD^a\alpha - \alpha\left(\chi_{ab}\chi^{ab} - \lambda + \kappa\frac{1}{2}(\rho + h^{ab}T_{ab})\right), \tag{2.5.20}$$

which when plugged into (2.5.19) leads to a wave equation for the lapse

$$\begin{aligned}\frac{1}{\alpha^2}\left(\partial_t-\beta^c\partial_c\right)^2\alpha-D_aD^a\alpha=-\alpha\Big[&\chi_{ab}\chi^{ab}-\lambda+\frac{\kappa}{2}\left(2\rho+T\right)\\&-4\chi n_\nu\Gamma^\nu-2\left(n_\nu\Gamma^\nu\right)^2-2\chi^2\Big]+\\&\left(\partial_t-\beta^c\partial_c\right)\left(n_\nu\Gamma^\nu\right)\\&=:S-D_aD^a\alpha\end{aligned}\tag{2.5.21}$$

Now equations (2.5.16),(2.5.18) and (2.5.21) form a set of wave equations for the metric components, the shift vector and the lapse function. The following evolution system is based these equations.

In order to bring our evolution system in a form of first-order in time and second-order in space we introduce additional auxiliary variables. These quantities obey an extra set of evolution equations besides the ones already in place for lapse α, shift β^i and spatial metric h_{ab}. In the following section we use a modified definition of the unit normal vector to the time slices $\vec{n}$, where we include a weighting factor $(0<w\leq 1)$.

$$n^\nu=\frac{1}{\alpha}\left(\delta^\nu_0-w\beta^\nu\right),\tag{2.5.22}$$

to ensure that n^μ remains timelike everywhere. The weight w is a function $w=w(x^i)$ $(\partial_t w=0$ but $\partial_i w\neq 0)$ that vanishes near the outer boundary and is unity in the interior of the domain.

The auxiliary variables are defined using Lie-derivatives along the normal vector n^μ

$$A:=\mathcal{L}_n\alpha=n^\mu\partial_\mu\alpha=\frac{1}{\alpha}\partial_t\alpha-\frac{w}{\alpha}\beta^i\partial_i\alpha\tag{2.5.23}$$

$$\begin{aligned}A^i:=\mathcal{L}_n\beta^i&=n^\mu\partial_\mu\beta^i-\beta^h\partial_h n^i\\&=\frac{1}{\alpha}\partial_t\beta^i-\frac{w}{\alpha^2}\beta^h\partial_h\alpha\beta^i+\frac{1}{\alpha}\beta^h\partial_h w\beta^i\end{aligned}\tag{2.5.24}$$

$$\begin{aligned}A_{ij}:=\frac{1}{2}\mathcal{L}_n g_{ij}&=\frac{1}{2}[n^\mu\partial_\mu g_{ij}+g_{i\mu}\partial_j n^\mu+g_{\mu j}\partial_i n^\mu]\\&=\frac{1}{2}\Big[\frac{1}{\alpha}\partial_t g_{ij}-\frac{w}{\alpha}\beta^h\partial_h g_{ij}\\&-\frac{1}{\alpha^2}\left(g_{ix}\beta^x+g_{iy}\beta^y+g_{iz}\beta^z\right)\partial_j\alpha\\&+g_{ih}\left(\frac{w}{\alpha^2}\partial_j\alpha\beta^h-\frac{1}{\alpha}\partial_j w\beta^h-\frac{w}{\alpha}\partial_j\beta^h\right)\\&-\frac{1}{\alpha^2}\left(g_{xj}\beta^x+g_{yj}\beta^y+g_{zj}\beta^z\right)\partial_i\alpha\\&+g_{hj}\left(\frac{w}{\alpha^2}\partial_i\alpha\beta^h-\frac{1}{\alpha}\partial_i w\beta^h-\frac{w}{\alpha}\partial_i\beta^h\right)\Big].\end{aligned}\tag{2.5.25}$$

Now the evolution equations of the ADM variables are expressed in terms of the definition of the corresponding auxiliary variables A, A^i and A_{ij}.

$$\partial_t \alpha = \alpha A + w\beta^i \partial_i \alpha \tag{2.5.26}$$

$$\partial_t \beta^i = \alpha A^i + \frac{w}{\alpha}\beta^h \partial_h \alpha \beta^i - \beta^h \partial_h w \beta^i \tag{2.5.27}$$

$$\begin{aligned}
\partial_t g_{ij} &= 2\alpha A_{ij} - \alpha[n^h \partial_h g_{ij} + g_{i\mu}\partial_j n^\mu + g_{\mu j}\partial_i n^\mu] \\
&= 2\alpha A_{ij} + w\beta^h \partial_h g_{ij} \\
&+ \frac{1}{\alpha}\left(g_{ix}\beta^x + g_{iy}\beta^y + g_{iz}\beta^z\right)\partial_j \alpha \\
&- g_{ih}\left(\frac{w}{\alpha}\partial_j \alpha \beta^h - \partial_j w \beta^h - w\partial_j \beta^h\right) \\
&+ \frac{1}{\alpha}\left(g_{xj}\beta^x + g_{yj}\beta^y + g_{zj}\beta^z\right)\partial_i \alpha \\
&- g_{hj}\left(\frac{w}{\alpha}\partial_i \alpha \beta^h - \partial_i w \beta^h - w\partial_i \beta^h\right).
\end{aligned} \tag{2.5.28}$$

To obtain an equation governing the time evolution of A we first expand the box operator in (2.5.21)

$$\begin{aligned}
\frac{1}{\alpha^2}\left(\partial_t - \beta^c \partial_c\right)^2 \alpha = \frac{1}{\alpha^2}\Big[&\partial_t \partial_t - \left(\partial_t \beta^c\right)\partial_c - \beta^c \partial_t \partial_c \\
&- \beta^c \partial_c \partial_t + \beta^c \left(\partial_c \beta^d\right)\partial_d + \beta^c \beta^d \partial_c \partial_d\Big]\alpha,
\end{aligned} \tag{2.5.29}$$

which yields an expression for $\partial_t \partial_t \alpha$ in terms of its source terms

$$\begin{aligned}
\partial_t \partial_t \alpha = \Big[&\left(\partial_t \beta^c\right)\partial_c + \beta^c \partial_t \partial_c \\
&+ \beta^c \partial_c \partial_t - \beta^c \left(\partial_c \beta^d\right)\partial_d - \beta^c \beta^d \partial_c \partial_d\Big]\alpha + \alpha^2 D_a D^a \alpha + \alpha^2 S + B_{tt}.
\end{aligned} \tag{2.5.30}$$

Applying the operator ∂_t to (2.5.23) we get

$$\begin{aligned}
\partial_t A &= \partial_t n^\mu \partial_\mu \alpha + n^\mu \partial_t \partial_\mu \alpha \\
&= -\frac{1}{\alpha^2}\partial_t \alpha \partial_t \alpha + \frac{1}{\alpha}\partial_t \partial_t \alpha \\
&+ \frac{w}{\alpha^2}\partial_t \alpha \beta^i \partial_i \alpha - \frac{w}{\alpha}\partial_t \beta^i \partial_i \alpha - \frac{w}{\alpha}\beta^i \partial_t \partial_i \alpha,
\end{aligned} \tag{2.5.31}$$

using (2.5.30) to substitute for the second time derivative of the lapse. We see that thus is the explicit evolution equation for A.

Expanding the box operator in (2.5.22) we get:

$$\begin{aligned}
\frac{1}{\alpha^2}\left(\partial_t - \beta^c \partial_c\right)^2 \beta^a = \frac{1}{\alpha^2}\Big[&\partial_t \partial_t - \left(\partial_t \beta^c\right)\partial_c - \beta^c \partial_t \partial_c \\
&- \beta^c \partial_c \partial_t + \beta^c \left(\partial_c \beta^d\right)\partial_d + \beta^c \beta^d \partial_c \partial_d\Big]\beta^a.
\end{aligned} \tag{2.5.32}$$

This then yields the expression

$$\begin{aligned}\partial_t\partial_t\beta^a = &\Big[\left(\partial_t\beta^c\right)\partial_c + \beta^c\partial_t\partial_c \\ &+ \beta^c\partial_c\partial_t - \beta^c\left(\partial_c\beta^d\right)\partial_d - \beta^c\beta^d\partial_c\partial_d\Big]\beta^a \\ &+ \alpha^2 h^{de}\partial_d\partial_e\beta^a + \alpha^2 S^a + B_{ta},\end{aligned} \tag{2.5.33}$$

which when plugged into $\partial_t A^i$ results in the evolution equation for A^i

$$\begin{aligned}\partial_t A^i = &-\frac{1}{\alpha^2}\partial_t\alpha\partial_t\beta^i + \frac{1}{\alpha}\partial_t\partial_t\beta^i + \frac{2}{\alpha^3}\left(\partial_t\alpha\right)w\beta^h\left(\partial_h\alpha\right)\beta^i \\ &- \frac{w}{\alpha^2}\left(\partial_t\beta^h\right)\left(\partial_h\alpha\right)\beta^i - \frac{w}{\alpha^2}\beta^h\left(\partial_t\partial_h\alpha\right)\beta^i \\ &- \frac{w}{\alpha^2}\beta^h\left(\partial_h\alpha\right)\partial_t\beta^i - \frac{1}{\alpha^2}\left(\partial_t\alpha\right)\beta^h\left(\partial_h w\right)\beta^i \\ &+ \frac{1}{\alpha}\left(\partial_t\beta^h\right)\left(\partial_h w\right)\beta^i + \frac{1}{\alpha}\beta^h\left(\partial_h w\right)\left(\partial_t\beta^i\right).\end{aligned} \tag{2.5.34}$$

Expanding the left hand side of (2.5.23) we get

$$-g^{\mu\nu}\partial_\mu\partial_\nu h_{ab} = \frac{1}{\alpha^2}\Big[\partial_t\partial_t - 2\beta^i\partial_i\partial_t + \beta^i\beta^j\partial_i\partial_j\Big]h_{ab} - h^{ij}\partial_i\partial_j h_{ab}, \tag{2.5.35}$$

which we use to substitute for $\partial_t\partial_t h_{ab}$

$$\partial_t\partial_t h_{ab} = \Big[2\beta^i\partial_i\partial_t - \beta^i\beta^j\partial_i\partial_j\Big]h_{ab} + \alpha^2 h^{ij}\partial_i\partial_j h_{ab} + \alpha^2 S_{ab} + B_{ab} \tag{2.5.36}$$

in $\partial_t A_{ij}$

$$\begin{aligned}
\partial_t A_{ij} = \frac{1}{2}\Big[& \left(\partial_t n^\mu\right)\partial_\mu h_{ij} + n^\mu \partial_t \partial_\mu h_{ij} + \left(\partial_t h_{i\mu}\right)\partial_j n^\mu \\
& + h_{i\mu}\partial_t\partial_j n^\mu + \left(\partial_t h_{\mu j}\right)\partial_i n^\mu + h_{\mu j}\partial_t\partial_i n^\mu \Big] \\
= \frac{1}{2}\Big[& -\frac{1}{\alpha^2}\left(\partial_t\alpha\right)\partial_t h_{ij} + \frac{1}{\alpha}\partial_t\partial_t h_{ij} \\
& + \frac{w}{\alpha^2}\left(\partial_t\alpha\right)\beta^h\partial_h h_{ij} - \frac{w}{\alpha}\left(\partial_t\beta^h\right)\partial_h h_{ij} \\
& - \frac{w}{\alpha}\beta^h\partial_t\partial_h h_{ij} + \frac{2}{\alpha^3}\left(\partial_t\alpha\right)h_{it}\partial_j\alpha \\
& - \frac{1}{\alpha^2}\left(\partial_t h_{it}\right)\partial_j\alpha - \frac{1}{\alpha^2}h_{it}\partial_t\partial_j\alpha \\
& + 2\Big[\partial_t h_{(ih}\left(\frac{w}{\alpha^2}\partial_{j)}\alpha\beta^h - \frac{1}{\alpha}\partial_{j)}w\beta^h - \frac{w}{\alpha}\partial_{j)}\beta^h\right) \\
& + h_{(ih}\Big(-\frac{2w}{\alpha^3}\left(\partial_t\alpha\right)\partial_{j)}\alpha\beta^h + \frac{w}{\alpha^2}\left(\partial_t\partial_{j)}\alpha\right)\beta^h \\
& + \frac{w}{\alpha^2}\left(\partial_{j)}\alpha\right)\partial_t\beta^h + \frac{1}{\alpha^2}\left(\partial_t\alpha\right)\left(\partial_{j)}w\right)\beta^h \\
& - \frac{1}{\alpha}\left(\partial_{j)}w\right)\partial_t\beta^h + \frac{w}{\alpha^2}\partial_{j)}\beta^h - \frac{w}{\alpha}\partial_t\partial_{j)}\beta^h\Big)\Big]\Big]
\end{aligned} \tag{2.5.37}$$

to obtain the evolution equation for A_{ij}. The extra source terms $B_{\mu\nu}$ are constraint adjustment terms, which weuse to control the propagation of constraint violating modes. We will in detail discuss the constraint adjustment technique in section 2.5.2.

2.5.1 Gauge source functions

As we have seen from (2.5.9) and (2.5.10) we can try to influence the time evolution of lapse and shift by specifying proper gauge source functions. To better match the nature of the evolution system which is in 3+1 splitted form already, we also use 3+1 splitted gauge source functions

$$\left(n_\nu\Gamma^\nu\right)\left(x^\nu\right) = F(x^\nu) \tag{2.5.38}$$

$$\left(h^a_\nu\Gamma^\nu\right)\left(x^\nu\right) = F^a(x^\nu), \tag{2.5.39}$$

whose expression in terms of lapse α and shift β^i are obtained from (2.5.9) and (2.5.10)

$$\left(n_\nu\Gamma^\nu\right) = \frac{1}{\alpha^2}\left(\partial_t\alpha - \beta^a\partial_a\alpha\right) - \chi \tag{2.5.40}$$

$$\left(h^a_\nu\Gamma^\nu\right) = -\frac{1}{\alpha^2}\left(\partial_t\beta^a - \beta^b\partial_b\beta^a\right) - \gamma^a - D^a\log\alpha. \tag{2.5.41}$$

In order to control the evolution of lapse and shift, we introduce evolution equations for the gauge source functions. Their evolution then is mostly governed by our choice of source terms

$$\frac{1}{\alpha^2}\left(\partial_t - \beta^c\partial_c\right)^2\left(n_\nu\Gamma^\nu\right) - D_aD^a\left(n_\nu\Gamma^\nu\right) = S\_nF \tag{2.5.42}$$

$$\frac{1}{\alpha^2}\left(\partial_t - \beta^c\partial_c\right)^2\left(h^a_\nu\Gamma^\nu\right) - h^{de}\partial_d\partial_e\left(h^a_\nu\Gamma^\nu\right) = S\_hF^a. \tag{2.5.43}$$

As for the evolution system for the ADM-variables we again introduce auxiliary variables to cast the evolution system into a coupled first order form:

$$(A\_nF) := \mathcal{L}_n n_\nu\Gamma^\nu = n^\mu\partial_\mu n_\nu\Gamma^\nu = \frac{1}{\alpha}\partial_t n_\nu\Gamma^\nu - \frac{w}{\alpha}\beta^i\partial_i n_\nu\Gamma^\nu \tag{2.5.44}$$

$$\begin{aligned}(A\_hF)^i &:= \mathcal{L}_n h^i_\nu\Gamma^\nu = n^\mu\partial_\mu h^i_\nu\Gamma^\nu - h^h_\nu\Gamma^\nu\partial_h n^i \\ &= \frac{1}{\alpha}\partial_t\left(h^i_\nu\Gamma^\nu\right) - \frac{w}{\alpha}\beta^h\partial_h h^i_\nu\Gamma^\nu - h^h_\nu\Gamma^\nu\partial_h n^i.\end{aligned} \tag{2.5.45}$$

First the evolution equations for the gauge source functions themselves

$$\partial_t\left(n_\nu\Gamma^\nu\right) = \alpha(A\_nF) + w\beta^i\partial_i\left(n_\nu\Gamma^\nu\right) \tag{2.5.46}$$

and

$$\begin{aligned}\partial_t\left(h^i_\nu\Gamma^\nu\right) = \alpha(A\_hF)^i + \frac{w}{\alpha}\beta^h\partial_h\alpha\left(h^i_\nu\Gamma^\nu\right) - \\ \beta^h\partial_h w\left(h^i_\nu\Gamma^\nu\right).\end{aligned} \tag{2.5.47}$$

Now we obtain the equations controlling the time evolution of the auxilary variables by using their definitions (2.5.44) and (2.5.45)

$$\begin{aligned}\partial_t(A\_nF) &= \partial_t n^\mu\partial_\mu\left(n_\nu\Gamma^\nu\right) + n^\mu\partial_t\partial_\mu\left(n_\nu\Gamma^\nu\right) \\ &= -\frac{1}{\alpha^2}\partial_t\alpha\partial_t\left(n_\nu\Gamma^\nu\right) + \frac{1}{\alpha}\partial_t\partial_t\left(n_\nu\Gamma^\nu\right) \\ &+ \frac{w}{\alpha^2}\partial_t\alpha\beta^i\partial_i\left(n_\nu\Gamma^\nu\right) - \frac{w}{\alpha}\partial_t\beta^i\partial_i\left(n_\nu\Gamma^\nu\right) - \frac{w}{\alpha}\beta^i\partial_t\partial_i\left(n_\nu\Gamma^\nu\right)\end{aligned} \tag{2.5.48}$$

with

$$\begin{aligned}\partial_t\partial_t\left(n_\nu\Gamma^\nu\right) = \Big[&\left(\partial_t\beta^c\right)\partial_c + \beta^c\partial_t\partial_c \\ &+ \beta^c\partial_c\partial_t - \beta^c\left(\partial_c\beta^d\right)\partial_d - \beta^c\beta^d\partial_c\partial_d\Big]\left(n_\nu\Gamma^\nu\right) \\ &+ \alpha^2 D_a D^a\left(n_\nu\Gamma^\nu\right) + \alpha^2 S\_nF.\end{aligned} \tag{2.5.49}$$

For $(A\_hF)^i$ we find

$$\begin{aligned}\partial_t(A\_hF)^i = -\frac{1}{\alpha^2}\partial_t\alpha\partial_t\left(h^i_\nu\Gamma^\nu\right) + \frac{1}{\alpha}\partial_t\partial_t\left(h^i_\nu\Gamma^\nu\right) + \\ \frac{w}{\alpha^2}\left(\partial_t\alpha\right)\beta^h\partial_h\left(h^i_\nu\Gamma^\nu\right) - \frac{w}{\alpha}\left(\partial_t\beta^h\right)\partial_h\left(h^i_\nu\Gamma^\nu\right) - \\ \frac{w}{\alpha}\beta^h\partial_h\partial_t\left(h^i_\nu\Gamma^\nu\right) - \partial_t\left(h^h_\nu\Gamma^\nu\right)\partial_h n^i - \left(h^h_\nu\Gamma^\nu\right)\partial_h\partial_t n^i,\end{aligned} \tag{2.5.50}$$

with

$$\begin{aligned}\partial_t\partial_t\left(h^i_\nu\Gamma^\nu\right) = &\Big[\left(\partial_t\beta^c\right)\partial_c + \beta^c\partial_t\partial_c \\ &+ \beta^c\partial_c\partial_t - \beta^c\left(\partial_c\beta^d\right)\partial_d - \beta^c\beta^d\partial_c\partial_d\Big]\left(h^i_\nu\Gamma^\nu\right) \\ &+ \alpha^2 h^{de}\partial_d\partial_e\left(h^i_\nu\Gamma^\nu\right) + \alpha^2 S\_hF^i.\end{aligned} \tag{2.5.51}$$

Although having this evolution system for the gauge source functions already in place to allow for future development the results presented in this thesis were all obtained using gauge source functions of a pure harmonic form

$$\left(n_\nu\Gamma^\nu\right)\left(x^\nu\right) = F(x^\nu) = 0 \tag{2.5.52}$$

$$\left(h^a_\nu\Gamma^\nu\right)\left(x^\nu\right) = F^a(x^\nu) = 0. \tag{2.5.53}$$

2.5.2 Constraint adjustment

We include the constraint adjustment terms $B_{\mu\nu}$ in the evolution system. These are terms that vanish when the constraints are satisfied and which do not effect the principal part of the evolution system, but successfully damp exponential growth in the constraints which might occur during evolution. The constraint adjustment terms we actually use were orignally proposed by Babiuc, Szilágyi and Winicour for standardized tests for periodic boundary conditions. We translated these adjustments to match our evolution system. The adjustment terms are of the following form

$$A^{\mu\nu} = -\frac{a_1}{\sqrt{-g}}\mathcal{C}^\alpha\partial_\alpha\tilde{g}^{\mu\nu} \tag{2.5.54}$$

$$+\frac{a_2\mathcal{C}^\alpha\nabla_\alpha t}{\varepsilon + e_{\rho\sigma}\mathcal{C}^\rho\mathcal{C}^\sigma}\mathcal{C}^\mu\mathcal{C}^\nu \tag{2.5.55}$$

$$-\frac{a_3}{\sqrt{-g^{tt}}}\mathcal{C}^{(\mu}\nabla^{\nu)}t, \tag{2.5.56}$$

where ε is a small positive number and $a_i > 0$ are positive adjustable parameters.

$$e_{\rho\sigma} = g_{\rho\sigma} - \frac{2}{g^{tt}}(\nabla_\rho t)\nabla_\sigma t \tag{2.5.57}$$

is the natural metric of signature $(+++{+})$ associated with the Cauchy slicing. The adjustments (2.5.54) and (2.5.55) were effective in suppressing nonlinear instabilities in the shifted gauge wave tests. The adjustment (2.5.56) leads to constraint damping in the linear regime and has been used effectively by Pretorius in black hole simulations but was not effective in the nonlinear shifted gauge wave test.

To apply these terms to our evolution equations we have to first lower both incides of the adjustment terms using the four-dimensional metric $g_{\mu\nu}$ and subtract the trace $A = g_{\mu\nu}A^{\mu\nu}$ to translate from the densitized formulation to our non-densitized formulation

$$A_{\mu\nu} = g_{\mu\sigma}g_{\nu\rho}A^{\sigma\rho} - \frac{1}{2}g_{\mu\nu}A. \tag{2.5.58}$$

Now we use the following combination of the $A_{\mu\nu}$'s to obtain the $B_{\mu\nu}$'s we actually include as extra source terms in our evolution equations

$$B_{tt} = -\alpha\beta^i\beta^j A_{ij} + 2\alpha(g_{ij}\beta^j\beta^k g^{il} A_i kl + g_{ij}\beta^j g^{ik} A_{tk}) \tag{2.5.59}$$

$$B_{ti} = -2\alpha^2(\beta^j g^{ik} A_{kj} + g^{ij} A_{tj}) \tag{2.5.60}$$

$$B_{ij} = -2\alpha^2 A_{ij}. \tag{2.5.61}$$

3 Numerical Implementation

3.1 Finite differencing scheme and operators

The code solves the finite-difference equations on a Cartesian grid with finest resolution $\Delta x^i = h$, using a cubic outer boundary and with excision boundaries for each black hole. Vertex-centered mesh refinement is applied using the `Carpet` driver [31], within the framework of the `Cactus` computational toolkit [32]. The time evolution is carried out by the method of lines using a third- or fourth-order Runge-Kutta scheme, with a fifth-order spatial prolongation and a second-order time interpolation to provide fine-grid boundary data at mesh-refinement boundaries. While the bulk of the code uses fourth-order accurate centered difference operators to approximate spatial derivatives, in a neighborhood of the outer boundary I approximate the spatial derivatives by diagonal norm SBP difference operators of fourth-order interior accuracy and of second-order accuracy at the boundary. More specifically, on a grid $x_I = x_0 + ih$ with boundary at x_0, these operators, as described by Mattsson and Nordström [43], are

$$(\partial_x f)_{i=1} \to \frac{1}{h}\left(\frac{1}{2}f_{[2]} - \frac{1}{2}f_{[0]}\right) , \tag{3.1.1}$$

$$(\partial_x f)_{i=2} \to \frac{1}{h}\left(-\frac{4}{43}f_{[4]} + \frac{59}{86}f_{[3]} - \frac{59}{86}f_{[1]} + \frac{4}{43}f_{[0]}\right) , \tag{3.1.2}$$

$$(\partial_x f)_{i=3} \to \frac{1}{h}\left(-\frac{4}{49}f_{[5]} + \frac{32}{49}f_{[4]} - \frac{59}{98}f_{[2]} + \frac{3}{98}f_{[0]}\right) , \tag{3.1.3}$$

and

$$(\partial_x^2 f)_{i=1} \to \frac{1}{h^2}\left(f_{[2]} - 2f_{[1]} + f_{[0]}\right) , \tag{3.1.4}$$

$$(\partial_x^2 f)_{i=2} \to \frac{1}{h^2}\left(-\frac{4}{43}f_{[4]} + \frac{59}{43}f_{[3]} - \frac{110}{43}f_{[2]} + \frac{59}{43}f_{[1]} - \frac{4}{43}f_{[0]}\right) , \tag{3.1.5}$$

$$(\partial_x^2 f)_{i=3} \to \frac{1}{h^2}\left(-\frac{4}{49}f_{[5]} + \frac{64}{49}f_{[4]} - \frac{118}{49}f_{[3]} + \frac{59}{49}f_{[2]} - \frac{1}{49}f_{[0]}\right) . \tag{3.1.6}$$

For grid points at the outer boundary, on the other hand, all components of the auxiliary variables A, A^i, A_{ij} are updated using a flat-spacetime, homogeneous Sommerfeld boundary condition. With the outer boundary located in the weak-field regime, a Sommerfeld condition applied to A, A^i, A_{ij} is equivalent to setting the Sommerfeld derivative of α, β^i and g_{ij} to the value of this Sommerfeld derivative at $t = 0$, as determined by the initial data. (By implication, our boundary algorithm is compatible with the initial data.) This boundary condition is effective in maintaining numerical stability. However, it is not constraint-preserving and is a prime target for future code improvement. In the part of the computational domain near the outer boundary where $w = 0$, the evolution algorithm for A, A^i, A_{ij} and α, β^i and g_{ij} reduces

to a fourth-order version of the subluminal evolution algorithm for evolving the wave equation with shift discussed in [40]. This algorithm is known to be unstable in the region where the shift is superluminal (e.g. near the excision boundary). In this superluminal region, I set $w = 1$ so that the definition of the auxiliary variables α, β^i and g_{ij} becomes the Lie derivative of the corresponding evolution variables in the normal direction to the Cauchy hypersurfaces, which stabilizes the algorithm. A more detailed discussion of this subluminal-superluminal blending is discussed in 2.4.

3.2 Conformal decomposition of the metric for finite differencing

In order to avoid NaN values resulting from applying the finite differencing operators to the metric components near singularities I decompose the metric g_{ij} in a conformal factor $\Psi^4 \equiv g^{\frac{1}{3}}$ and a resulting (conformally flat) metric $\tilde{g}_{ij}$

$$\tilde{g}_{ij} = g^{-\frac{1}{3}} g_{ij}, \tag{3.2.1}$$

where $g \equiv det(g_{ij})$ is the determinant of g_{ij}. From that I can obtain the expression for the first order derivative

$$\partial_a g_{ij} = \tilde{g}_{ij} \partial_a g^{\frac{1}{3}} + g^{\frac{1}{3}} \partial_a \tilde{g}_{ij}, \tag{3.2.2}$$

with

$$\partial_a g^{\frac{1}{3}} = g^{\frac{1}{3}} \partial_a \left(\ln g^{\frac{1}{3}} \right). \tag{3.2.3}$$

Here I take the derivative of the conformal factor $g^{\frac{1}{3}}$ with respect to its logarithm to avoid very big numerical values of the derivative operators. Now for derivatives of second order I can obtain the expression

$$\begin{aligned} \partial_a \partial_b g_{ij} = \; & \tilde{g}_{ij} \partial_a \partial_b g^{\frac{1}{3}} + (\partial_b g^{\frac{1}{3}})(\partial_a \tilde{g}_{ij}) \\ & + (\partial_a g^{\frac{1}{3}})(\partial_b \tilde{g}_{ij}) + g^{\frac{1}{3}} \partial_a \partial_b \tilde{g}_{ij}, \end{aligned} \tag{3.2.4}$$

where I again use the derivative of the logarithm to obtain

$$\partial_a \partial_b g^{\frac{1}{3}} = g^{\frac{1}{3}} (\partial_a \ln g^{\frac{1}{3}})(\partial_b \ln g^{\frac{1}{3}}) + g^{\frac{1}{3}} \partial_a \partial_b \ln g^{\frac{1}{3}}. \tag{3.2.5}$$

This technique helps in handling puncture initial data in the areas near the actual 'puncture location'. We tested this method to obtain the derivatives by evolving the same initial data set once with ordinary derivates and once with derivatives obtained by this conformal decomposition of the metric. We see good agreement of the two methods for testbeds with exact solutions and for numerical initial data sets.

3.3 Numerical Dissipation

Another important ingredient of this code is numerical dissipation. We find this essential in keeping the algorithm stable in the neighborhood of the excision domain. In addition, this is

also helpful in killing off high-frequency noise generated at the mesh-refinement boundaries. In the interior of the grid, numerical dissipation is added at $O(h^5)$ in the form

$$\begin{aligned}
\partial_t^2\alpha \to &\quad \partial_t^2\alpha + \frac{1}{64}h^5\sum_i \epsilon_i (D_{+i}D_{-i})^3\partial_t\alpha \\
\partial_t^2\beta^i \to &\quad \partial_t^2\beta^i + \frac{1}{64}h^5\sum_i \epsilon_i (D_{+i}D_{-i})^3\partial_t\beta^i \\
\partial_t^2 g_{ij} \to &\quad \partial_t^2 g_{ij} + \frac{1}{64}h^5\sum_i \epsilon_i (D_{+i}D_{-i})^3\partial_t g_{ij}
\end{aligned} \tag{3.3.1}$$

where $D_{\pm i}$ are the forward and backward difference operators in the x^i direction and ϵ_i is a smooth weighting function. In the neighborhood of a face of the outer boundary with normal in the x-direction, I set $\epsilon_x = 0$ so that the dissipation applies only in the tangential directions. In carrying out convergence tests for a Schwarzschild black hole, I choose $\epsilon_i = 0.2$ outside the apparent horizon (AH) and $\epsilon_i = 2$ inside the AH (except for a transition region). In the two black hole simulations, I choose $\epsilon_i = 1$ outside the AH and $\epsilon_i = 2$ inside the AH.

3.4 Boundary conditions

Our evolution domain has a timelike outer boundary and a smooth, spacelike excision boundary inside each MOTS. The harmonic evolution system, in the second-order form (2.4.7), consists of quasilinear wave equations whose characteristics are identical to the null directions determined by the metric. As a result, all characteristics leave the spacelike excision boundaries and no boundary conditions are necessary (or allowed). At the timelike outer boundary, any dissipative boundary condition for the wave equation with shift leads to a well-posed initial-boundary value problem (IBVP). Such dissipative boundary conditions were first worked out in the one-dimensional (1D) case [36, 37, 39] and general results for the 3D case have been discussed recently in [27, 40]. For a boundary with normal in the $+x$ direction, such dissipative boundary conditions have the form

$$\left[(1-\kappa)\partial_t + \kappa g^{z\rho}\partial_\rho\right]\tilde{g}^{\mu\nu} = q^{\mu\nu}, \qquad 0 \le \kappa \le 1, \tag{3.4.1}$$

for each component $\tilde{g}^{\mu\nu}$, where $q^{\mu\nu}$ is the boundary data. The choice $\kappa = 0$ gives a Dirichlet condition and $\kappa = 1$ gives a Neumann condition. Dirichlet and Neumann conditions are marginally dissipative in the sense that they are purely reflective for modes with $q^{\mu\nu} = 0$. A strictly dissipative Sommerfeld-type condition arises when κ is chosen so that the derivative in the left hand side of (3.4.1) lies in the outgoing null direction.

In order for the IBVP to be constraint preserving, the boundary data $q^{\mu\nu}$ must be assigned to enforce a homogeneous, dissipative boundary condition on the constraints C^μ. Then, with proper initialization, the uniqueness of solutions to eqs. (2.4.9) ensures that the constraints are satisfied throughout the evolution. The first proposal for such constraint-preserving boundary conditions for the harmonic system consisted of a combination of 3 Dirichlet and 7 Neumann conditions on the components of $\tilde{g}^{\mu\nu}$ [26]. However, numerical studies [27] showed that these Dirichlet-Neumann boundary conditions were effective in carrying the signal off the grid but that their marginally dissipative nature reflected the noise and gave poor results in highly nonlinear tests.

The first example of strictly dissipative constraint-preserving boundary conditions which would in principle allow numerical error to leave the grid, was given for a tetrad formulation of the Einstein equations by Friedrich and Nagy [41]. Constraint-preserving boundary conditions of the Sommerfeld type which lead to a well-posed IBVP for the non-linear harmonic formulation have subsequently been formulated [42]. These Sommerfeld-type boundary conditions have been incorporated in a numerical code which gives vastly superior performance in nonlinear test problems than the Dirichlet-Neumann scheme [28]. However, I have not yet implemented these condition here; instead I have used a naive version of Sommerfeld boundary conditions which not only is not constraint-preserving but whose numerical implementation is only second-order accurate. In addition, as explained further in Sect. 3.1, the code uses summation-by-parts (SBP) difference operators which are fourth-order accurate in the interior but only second-order accurate in the vicinity of the outer boundary.

3.5 Moving Excision

The excision algorithm is driven by the apparent horizon finder algorithm [14, 15]. Strictly speaking, this algorithm searches for MOTS, regardless of whether these are apparent horizons or not. We recall that MOTS are defined as smooth, compact 2-dimensional surfaces whose outgoing normal null geodesics have zero expansion. With respect to a $3+1$ foliation, the apparent horizon is defined as the 3-dimensional hypersurface traced out by the outer boundary of the trapped region in each time slice. If sufficiently smooth, the apparent horizon is foliated by MOTS. A smooth spacelike boundary is used to excise a region inside each MOTS, resulting in a jagged boundary in the Cartesian grid. The excision boundary is chosen to be centered inside the MOTS and scaled in coordinate size to be 0.7 the size of the MOTS in evolutions presented. We keep the same interior evolution stencil near the excision boundary by introducing the necessary ghost points. Because the dissipation operator (3.3.1) would require an excessive number of ghost points I replace it with a third-order form $h^3(D_{+i}D_{-i})^2$ near the excision boundary. Values at the required ghost points are supplied by an extrapolation scheme which was proved to be stable for the case of a boundary aligned with the grid [40]. We have generalized this to the case of a generic smooth boundary in a Cartesian grid following the "embedded-boundary" method developed by Kreiss and Petersson [44] for formulating a stable Neumann condition. More specifically, I construct a vector v^i by taking the flat-space displacement from the centroid of the excised region to the current position, and require that

$$\begin{aligned} \textstyle\sum_i (v^i D_{\pm i})^3 \alpha &= 0 \\ \textstyle\sum_i (v^i D_{\pm i})^3 \beta^i &= 0 \\ \textstyle\sum_i (v^i D_{\pm i})^3 g_{ij} &= 0 \end{aligned} \tag{3.5.1}$$

and

$$\begin{aligned} \textstyle\sum_i (v^i D_{\pm i})^3 A &= 0 \\ \textstyle\sum_i (v^i D_{\pm i})^3 A^i &= 0 \\ \textstyle\sum_i (v^i D_{\pm i})^3 A_{ij} &= 0 \end{aligned} \tag{3.5.2}$$

where the one-sided differences $D_{\pm i}$ correspond to the sign of v^i. The extrapolation conditions (3.5.1 and 3.5.2) are applied iteratively at the points near the boundary until the full stencil of

ghost points is updated.

Extra care needs to be paid to the identification of ghost points when the excision domain is moving across the grid. In particular, grid-points that become interior points at $t^N := t_0 + N\Delta t$ but were ghost points at t^{N-1} need to be treated as ghost points, *i.e.*, the excision algorithm must fill them with extrapolated values during the time-integration from t^{N-1} to t^N. However, when evolving from t^N to t^{N+1}, these same grid-points need no longer be treated as ghost points and can then be labeled as evolution points.

3.6 Convergence tests

In this section I present data describing the convergence behavior of the ADM variables for a shifted gauge wave testbed. The wavelength for this test was chosen to be $\lambda = 1.0M$. The outer boundary for this test setup was located at $2M$ and I have run this tests for $1M$ of evolution time. The test was run with periodic boundary conditions and the error was calculated using the known analytic solution. First I show the L2 norm of the error compared to the analytic solution for two different resolutions $h = 0.02M$ and $h = 0.01M$. In addition I present the convergence rates as functions of time t for selected ADM variables as a result of a three-level convergence test with resolutions $h = 0.04M$, $h = 0.02M$ and $h = 0.01M$.

Figures 3.1 - 3.3 show the L2 norm of the errors with respect to the analytic solution for α, β^x and g_{xx} for the two different resolutions $h = 0.04M$ and $h = 0.02M$. I see clean second order convergence for the quantities but for g_{xx} where I see an area of lower convergence rate between $0.5M$ and $0.9M$.

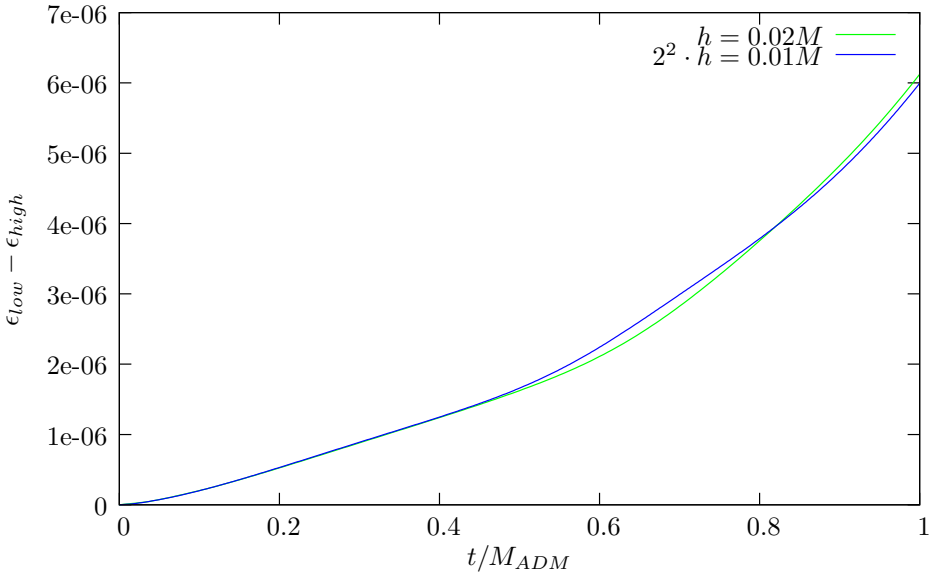


Figure 3.1: Difference between low resolution error ϵ_{low} and high resolution error ϵ_{high} for α.

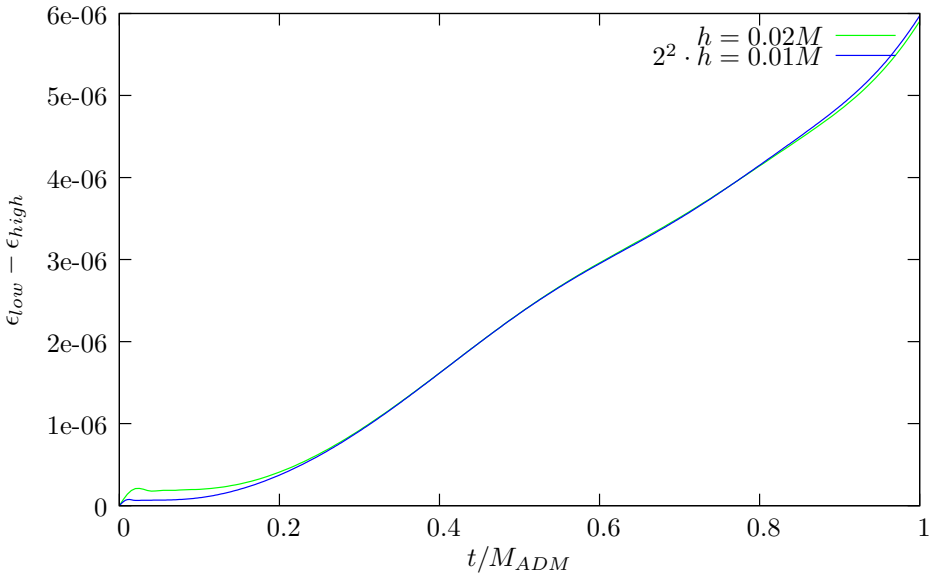


Figure 3.2: Difference between low resolution error ϵ_{low} and high resolution error ϵ_{high} for β^x.

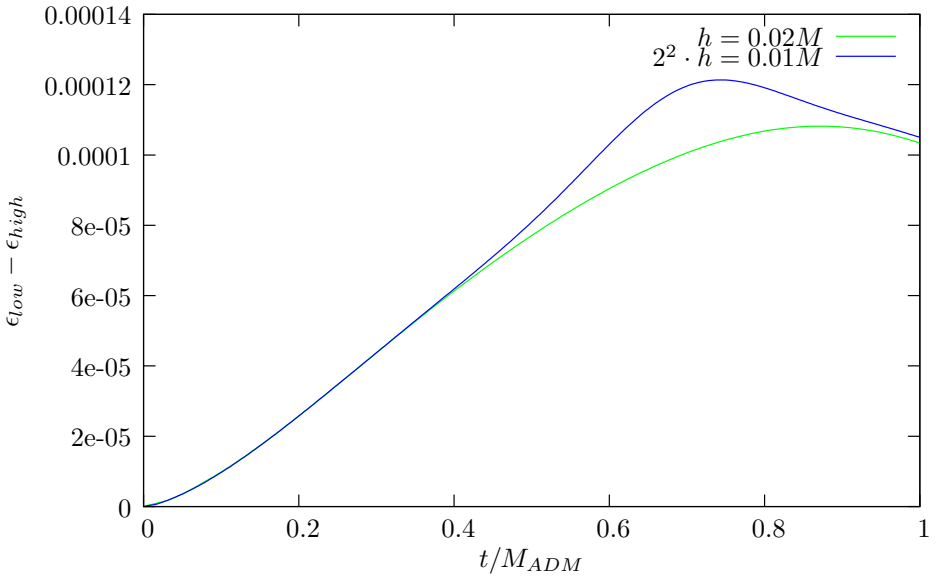


Figure 3.3: Difference between low resolution error ϵ_{low} and high resolution error ϵ_{high} for g_{xx}.

Figures 3.4 - 3.6 now show the convergence rate for α, β^x and g_{xx} at point $(1.08, 1.0, 1.0)$ on the Cartesian grid over evolution time. In the first timesteps I see a rapidly varying convergence rate, which is due to the truncation error still influencing the convergence rate. After a few timesteps I then see a constant convergence factor of 4. This reflects the fourth-order accurate stencils for finite differencing in the interior region of the grid. For late timesteps, the effects of the only second order accurate boundary stencils start to affect the convergence rate also at this point in the central region of the grid setup.

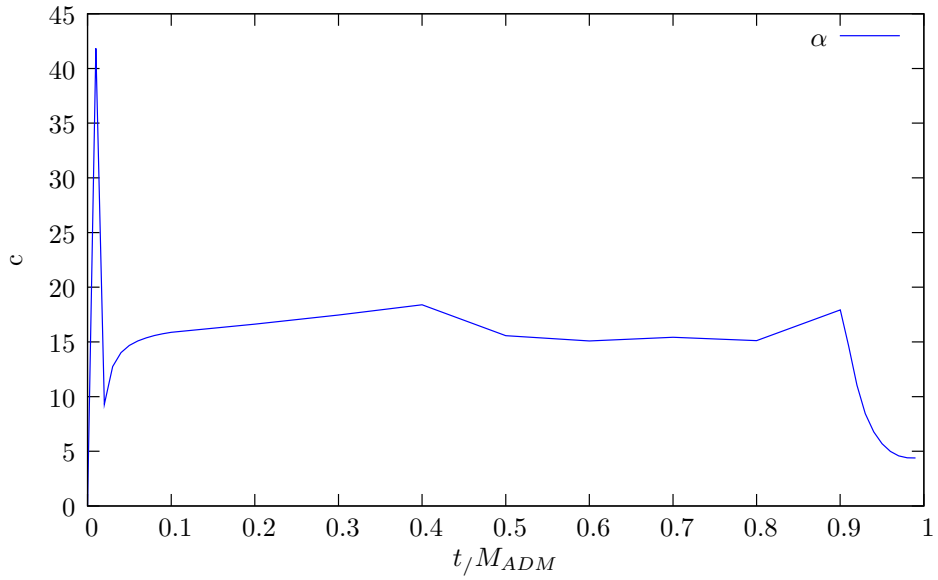


Figure 3.4: Convergence rate c for the lapse function α at $x = 1.08$,$y = 1.0$,$z = 1.0$.

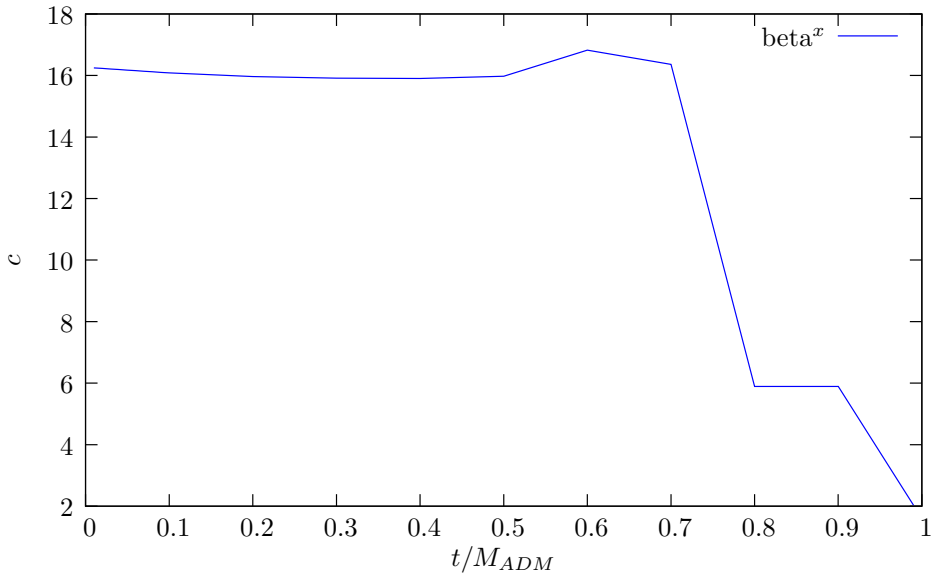


Figure 3.5: Convergence rate c for β^x at $x = 1.08$,$y = 1.0$,$z = 1.0$.

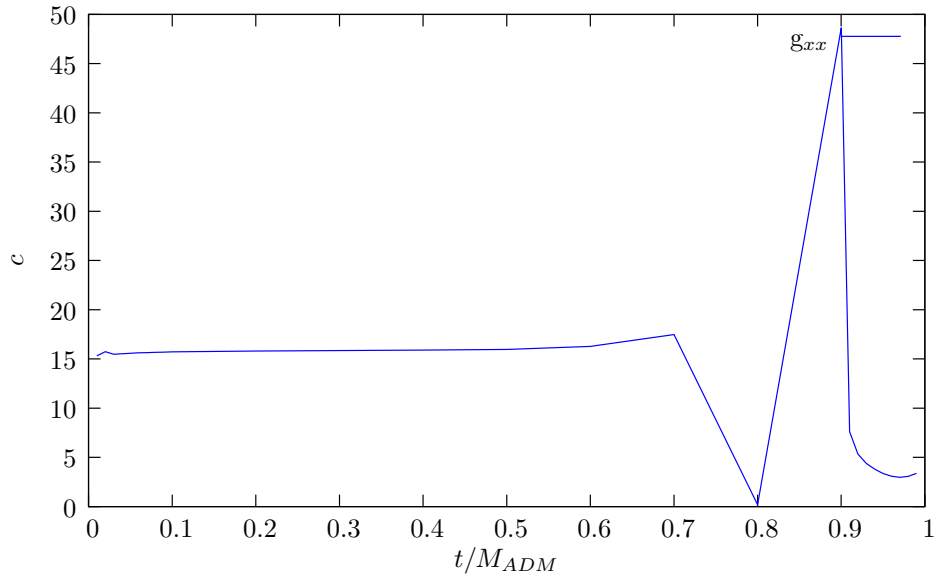


Figure 3.6: Convergence rate c for g_{xx} at $x = 1.08$,$y = 1.0$,$z = 1.0$.

4 Evolutions of single black holes

In this chapter I present results obtained for the 'ADM Harmonic' evolution system when used to evolve single black holes. To test the stability of the evolution equations and the numerical evolution system in general I evolve puncture initial data sets for long time scales (1000 M) and monitor the constraint violations as well as properties of the horizon itself to show the stability, robustness and validity of the results obtained. I present an analysis of different parameters for the constraint adjustment terms and their impact on the constraint violations in long term evolutions. Furthermore I investigate highly spinning black holes (dimensionless spin parameter up to 0.9). First I show the angular momentum conservation for these spinning black holes for the BSSN evolution system of the numerical relativity group of the Albert-Einstein-Institute and later compare this with results obtained by using the 'ADM Harmonic' evolution system. In the last section of this chapter I show the convergence behavior of the 'ADM Harmonic' evolution system for evolving puncture initial data single black holes to validate the results presented.

4.1 Stability tests and different constraint damping parameters

The initial data setup used for the results presented in this section consists of one Brill-Lindquist black-hole located in the origin of the Cartesian grid. The black hole mass is $1.0M$. The outer boundaries for this setup are located at $204.80M$. I use eight levels of mesh refinement with a grid step of $h^{(n)} = 2.56 \cdot 2^{-n}, n = 0, ..., 8$ resulting in a grid-step of $h = 2.56M$ for the coarsest grid and $h = 0.02M$ for the finest one. I adjust the refinement levels in order to keep the in the beginning growing apparent horizon well inside the finest grid. After $10.240M$ the finest refinement level is dropped and the grid structure is kept unchanged. The simulations are all run with reflection symmetries in the $x = 0$,$y = 0$ and $z = 0$ planes using pure harmonic gauge source functions (i.e. $n_\nu \Gamma^\nu = 0 = h^a{}_\nu \Gamma^\nu$).

Figure 4.1 shows the time evolution of the minimum radius of the apparent horizon over the whole evolution time of $1000M$ and in an inset zooms in on the first $50M$ of time evolution time. We see the minimum radius growing from $0.5M$ to nearly $1.1M$ and then after some oscillations settling down to $1M$. In the late stage of the evolution we see a mass loss of approximatly five percent which is probably due to resolution effects. The growth in the minimum radius we see in the beginnig is an effect of our choice of coordinates (i.e. the choice of pure harmonic gauge source functions).

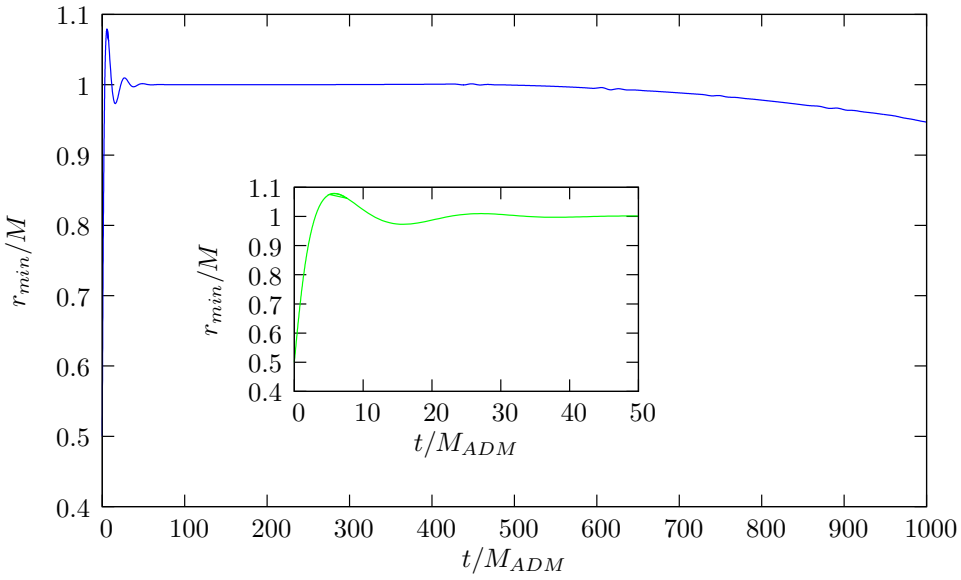
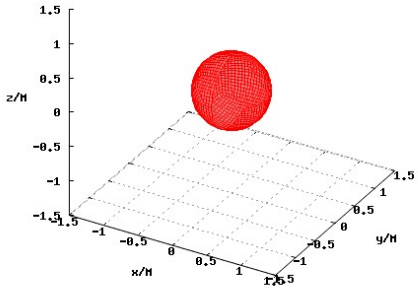
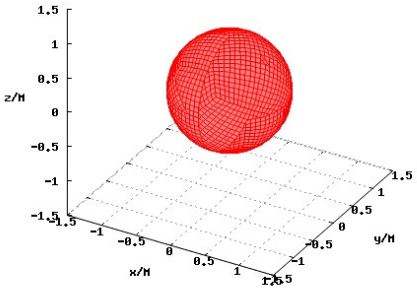


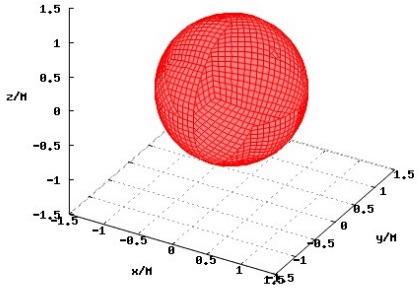
Figure 4.1: Minimum radius of the apparent horizon during the first 50M of evolution time.



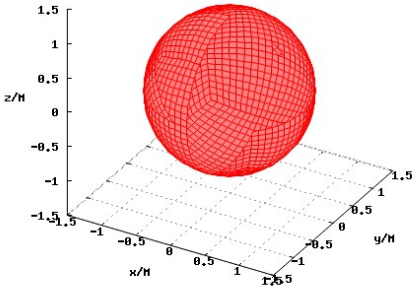
(a) Apparent horizon at t = 0M.



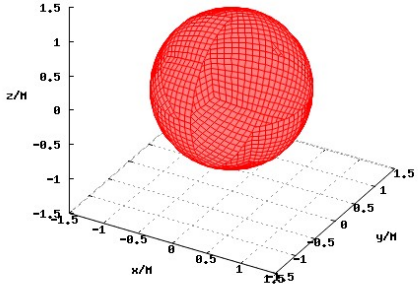
(b) Apparent horizon at t = 1.280M.



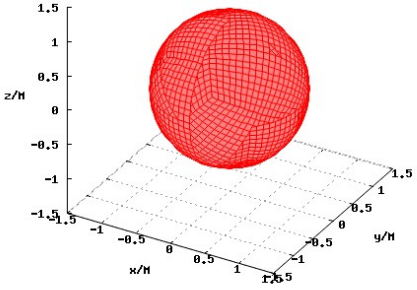
(c) Apparent horizon at t = 2.560M.



(d) Apparent horizon at t = 5.120M



(e) Apparent horizon at t = 10.240M.



(f) Apparent horizon at t = 50.176M

Figure 4.2: Visualizations of single black hole apparent horizon evolution.

Figure 4.2 shows the time evolution of the apparent horizon shape over the first $50M$ of evolution time. We again see the previously mentioned growth in radius due to gauge choice effects reaching its maximum at 5.120M. After that the horizon radius settles down to $1M$ and remains constant over the timescale presented here.

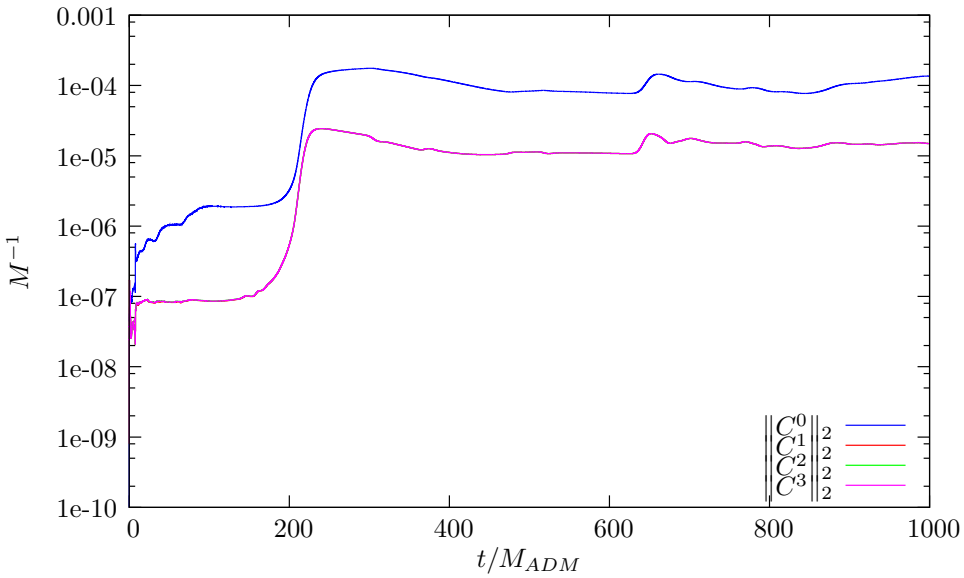


Figure 4.3: L2 norm of the harmonic constraints components

Figure 4.3 shows the L2 norms of the harmonic constraints components respectively for the time evolution of the described initial data set. This simulation was carried out up to 1000 to test the robustness of the implemented evolution system. I see no sign of instability for this simulation. At $t = 200M$ (one crossing time for the grid setup chosen) we see the non-constraint preserving boundary conditions starting to hit the norms. After that we see the constraint violations basically staying in constant until $1000M$. In order to test the effectiveness of the constraint damping terms (described in detail in 2.5.2) I first compare the implemented terms against terms suggested and used by Pretorius (ref paper) and then investigate the influence of different constraint adjustment term coefficients a_2 and a_3 itself.

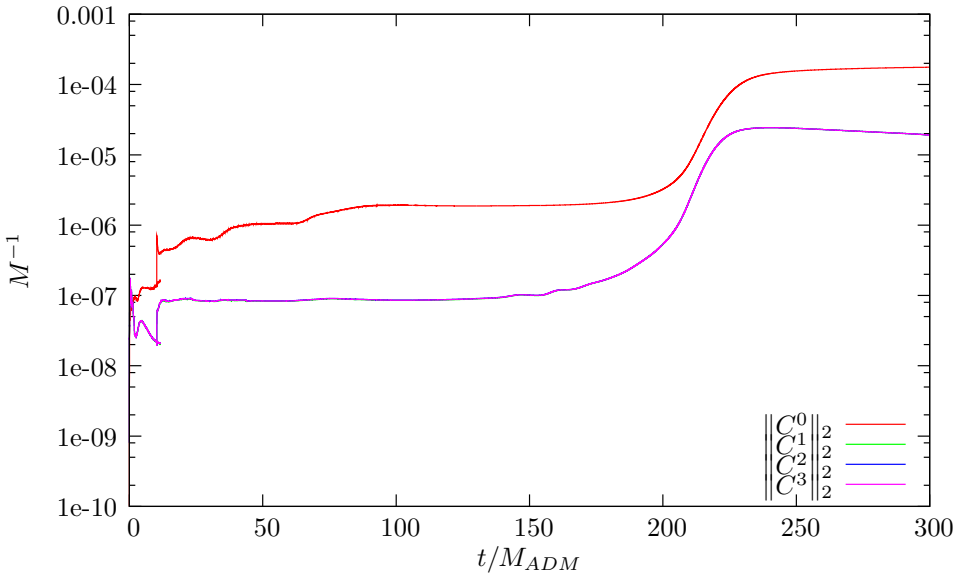


Figure 4.4: L2 norm of the harmonic constraints for AEIHarmonic damping terms.

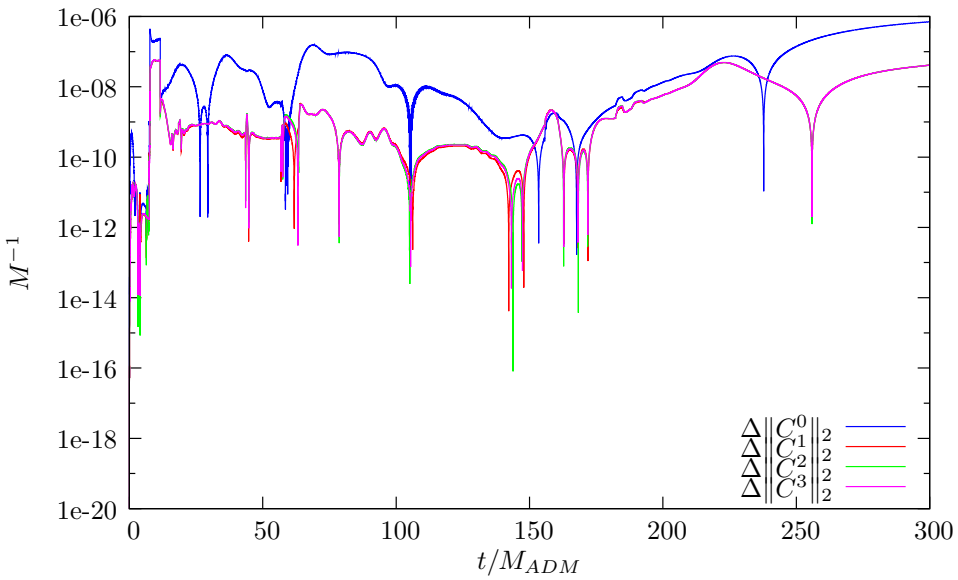


Figure 4.5: Difference in the L2 norms of the harmonic constraints between Pretorius and AEIHarmonic damping terms

Figure 4.4 shows the harmonic constraints for the densitized harmonic damping terms with an initial choice of the parameters of $a_2 = 1.0 = a_3$. Figure 4.5 then shows the difference in the harmonic constraints between the damping terms suggested by Pretorius and the ones implemented in the densitized harmonic evolution system. I find both being effective in damping exponential growing modes in the constraints.

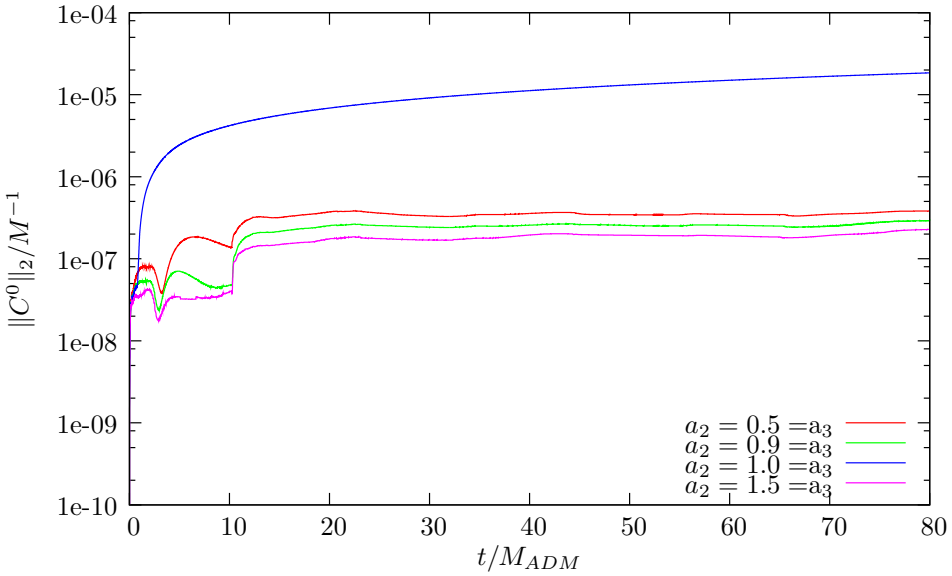


Figure 4.6: L2 norm of the harmonic constraints time component for different constraint damping parameters a_2 and a_3

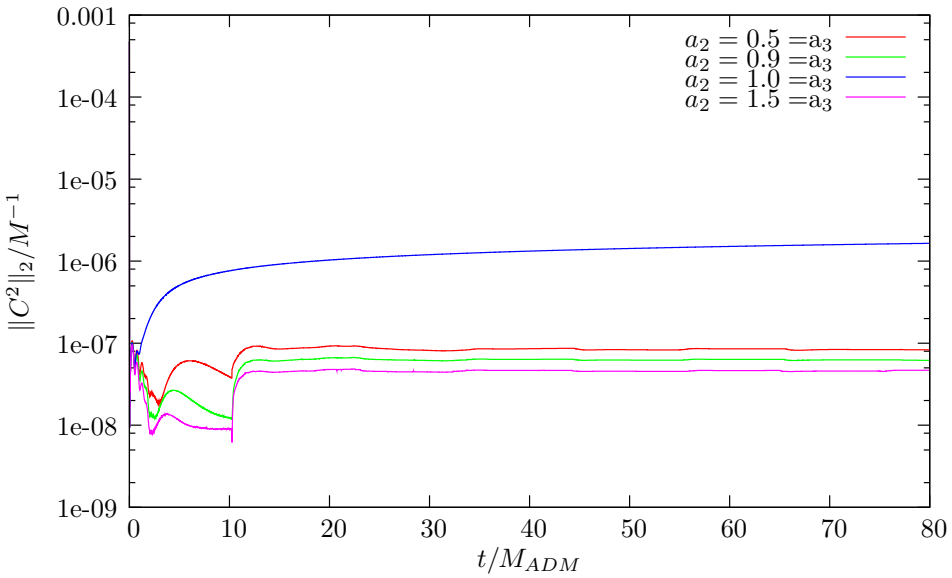


Figure 4.7: L2 norm of the harmonic constraints y component for different constraint damping parameters a_2 and a_3

Figures 4.6 and 4.7 show the impact of different parameter choices for the AEIHarmonic constraint damping terms two and three. In principle all parameter choices that we have tried lead to a stable evolution and a successful damping of the otherwise happening exponential growth in the constraint violations. Bigger parameter values show not surprisingly a more effective damping of the constraint violations. The choice of $a_2 = 1.0 = a_3$ for this parameter study surprisingly shows a different behavior for the constraint violations. I haven't had time to investigate this issue further for this thesis althoug this aspect is very interesting. However still all parameter choices I studied here show the desired damping of otherwise exponentially growing modes. Nearly all simulations presented here were run with the initial choice of $a_2 = a_3 = 1.0$ and $a_1 = a_4 = 0$ for the constraint damping parameters. The choice of $a_1 = a_4 = 0$ was only motivated by time saving reasons since we found the parameter choice of $a_2 = a_3 = 1.0$ already effective in damping the exponential growth in the constraints.

In this chapter I have carried out evolutions of single black holes using the ADM harmonic evolution code and demoinstrated that they are stable and accurate.
I have studied different constraint damping terms and demonstrated that are comparable in damping exponentially growing modes in the constraints.
In addition various values of constraint dampoing parameters haven been tested and I find that a value of $a_2 = 1.5 = a_3$ is optimal to keep the constraint violation at a minimal level.

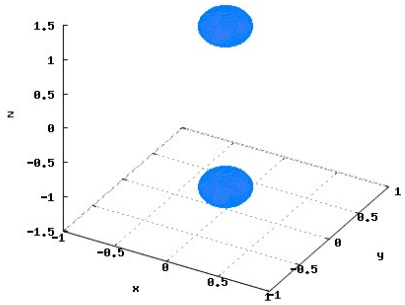
5 Evolution of binary black holes

In this section I present results obtained for a head-on collision of two black holes with the ADM Harmonic evolution code. I investigate the merger of the horizons and the constraint violations. Furthermore I compare the achieved results against the same simulation done with the 'Densitized Harmonic' evolution code.

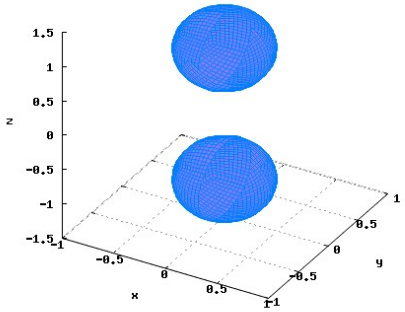
5.1 Head-on black hole collision: 'ADM Harmonic' vs. 'Densitized Harmonic'

The initial data setup used for this test consists of two equal mass non-spinning Brill-Lindquist black holes located at (0,0,1.16) and respectively (0,0,-1.16) on the z-axis of the Cartesian grid. The masses of the black holes are $0.5M$ each. The outer boundaries for this test setup are located at $144M$. I carried out the test using nine levels of mesh refinement with a grid step of $h^{(n)} = 3.2 \cdot 2^{-n}, n = 0, ..., 8$ resulting in a grid-step of $h = 3.2M$ for the coarsest grid and $h = 0.0125M$ for the finest one. The grid setup is chosen such that no movement of the refinement levels is necessary since both black holes are contained within the bounding box of the finest grid. The simulation is run wth refleciton symmetries in the $x = 0$,$y = 0$ and $z = 0$ planes using pure harmonic gauge source functions (i.e. $n_\nu \Gamma^\nu = h^a{}_\nu \Gamma^\nu = 0$ for the 'ADM Harmonic' formulation and $F^\nu = 0$ for the 'Densitized Harmonic' formulation). After the merger of the apparent horizons, when the final horizon has reached an ellipsoidal shape with maximum and minimum radii satisfying $r_{min}/r_{max} > 0.6$, the finest refinement level is dropped. This is reasonable since the coordinate radius of the final apparent horizon is more than two times that of the two original individual horizons. Both simulations show no obvious signs of instability.

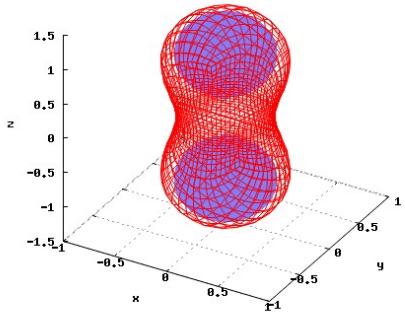
Figures 5.1 and 5.2 present snapshots at different time steps during the evolution of the head-on collision run. We see first the initial configuration with two individual black holes located at $\pm 1.16M$ on the z-axis. The next snapshot shows both individual horizons at $t = 4.080M$. We see the horizons growing in size due to gauge effects (e.g. the coordinates chosen) and getting closer together. After that the common horizon is already found at $t = 4.400M$ but still also the two individual ones are present. After t=$5.380M$ the individuals horizons are lost. We then see the common horizon evolve in shape from a peanut like configuration shortly after merger to its spherical end shape rather quickly.



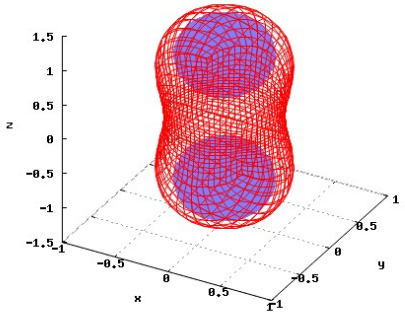
(a) Individual apparent horizons at t = 0M.



(b) Individual apparent horizons at t = 4.080M.

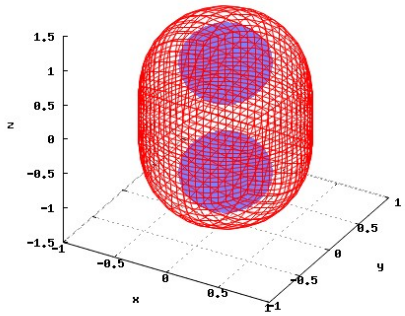


(c) Individual apparent horizons and the common apparent horizon at t = 4.400M.

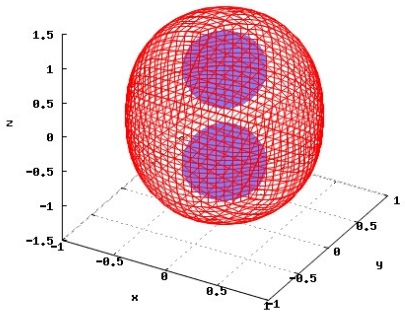


(d) Individual apparent horizons and the common apparent horizon at t = 4.480M

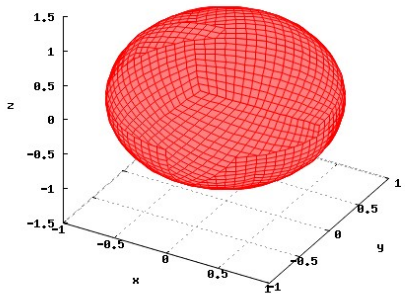
Figure 5.1: Visualization of the horizons for the head-on collision.



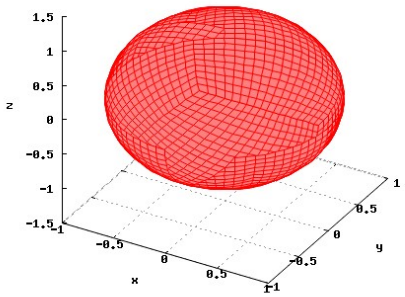
(a) Individual apparent horizons and the common apparent horizon at t = 5.238M



(b) Individual apparent horizons and the common apparent horizon at t = 5.840M



(c) The common apparent horizon at t = 66M



(d) The common apparent horizon at t = 66M

Figure 5.2: Visualizations of head on collision 2.

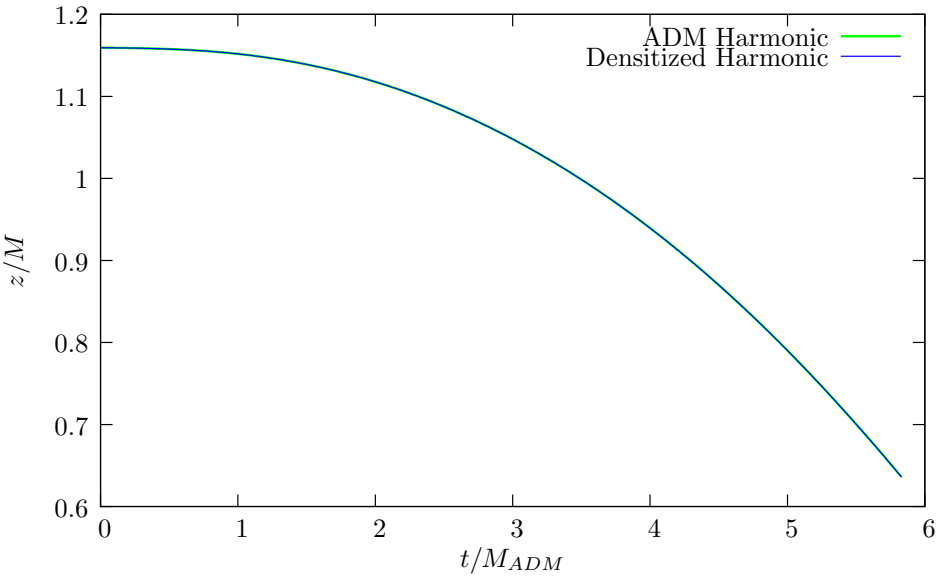


Figure 5.3: z-position of the individual apparent horizons before merger.

Figure 5.3 shows the z-coordinate of the center of the original individual horizons before merger (around $5M$). Since this simulation was run in octant symmetry only the location of the horizon on the positive part of the z-axis is shown here. This describes the behavior of the second horizon in the negative z-octant as well. The green line represents the result obtained using the 'ADM Harmonic' formulation and the blue one the 'Densitized Harmonic' formulation results. One sees the horizon centers initially at $1.16M$ starting to come closer together ever faster with elapsed time until they merge between $5M$ and $6M$. During this time a common apparent horizon has already formed. I see very good agreement between the two formulations for the time evolution of this quantity.

Figure 5.4 presents the mean radius ($r_{mean} = (r_{min} + r_{max})/2$) of the final common apparent horizon formed during merger as a function of time t. The oscillations shortly after merger indicate ellipsoidal distortions of the horizon shape. Directly after merger the common black hole is in an excited state. This is reflected in the shape of the horizon. The horizon then settles down rather quickly (in the first $50M$ after merger) to a spherical shape which indicates that the final black hole is in the end a non-spinning one. The horizon shapes and their time evolution are represented in more detail in figures 5.1 and 5.2. Again I see very good agreement for the results achieved with the two different harmonic formulations.

Figures 5.5 and 5.6 show the behavior of ratio of the circumferences of different planes through the final apparent horizon over time. Both the xz/xy-plane and the yz/xy-plane circumference ratio indicate a measure for the spin of the final apparent horizon. Both figures show the ratio of the circumferences settling down to one after some early oscillations. That indicates the formed final apparent horizon settling down to a non-spinning state after some quasi-normal ringing.

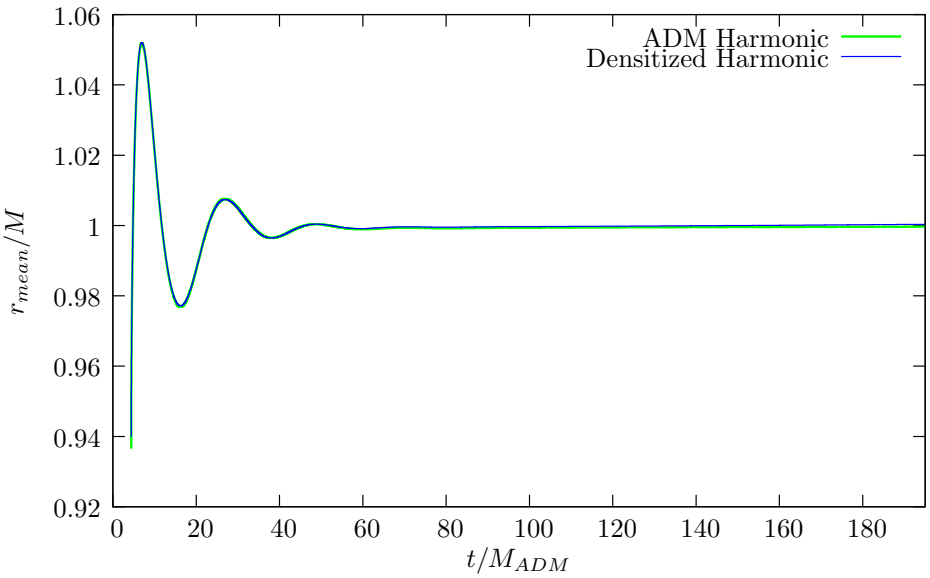


Figure 5.4: Mean radius of the common apparent horizon after merger.

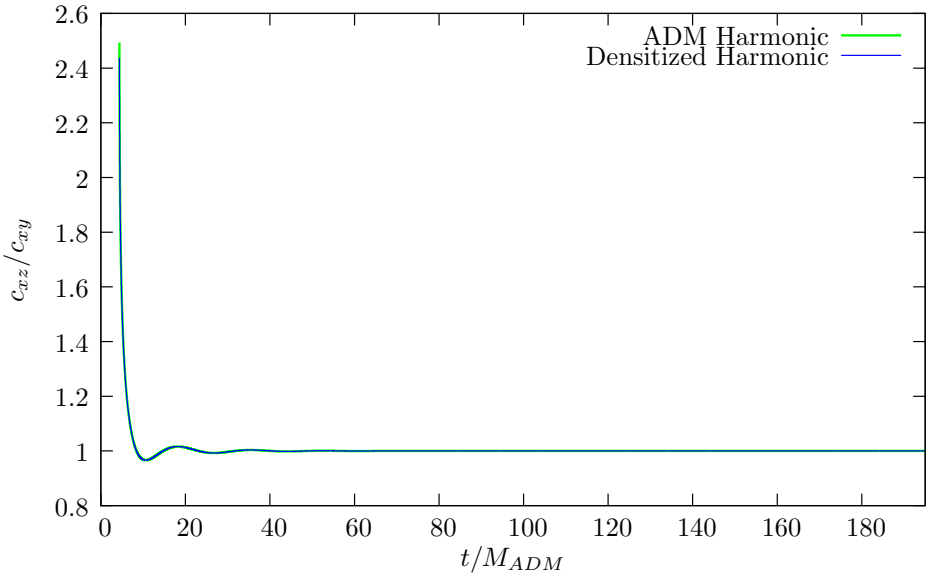


Figure 5.5: ratio of xz/xy-plane circumferences of the common apparent horizon after merger.

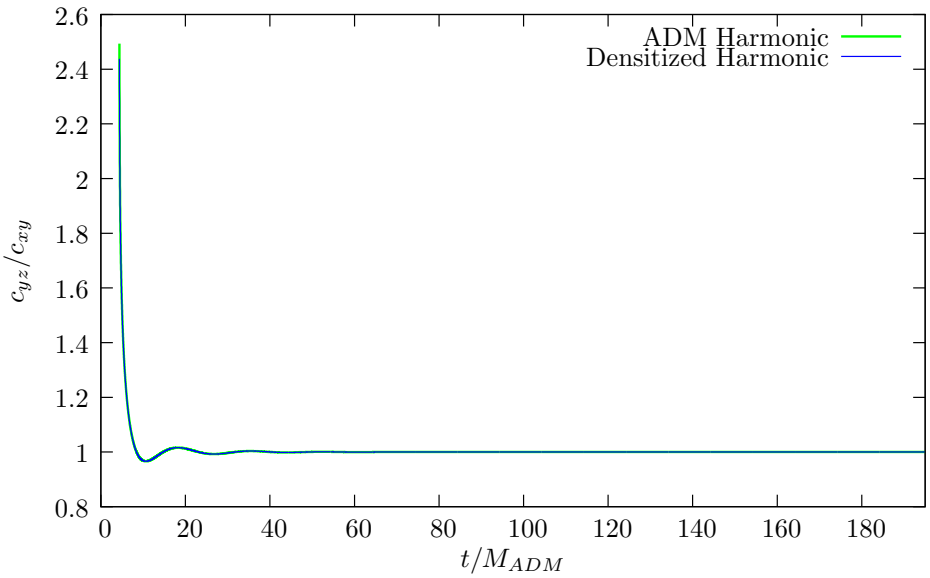
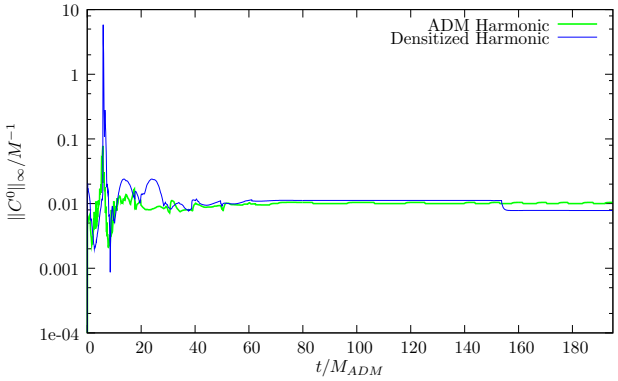
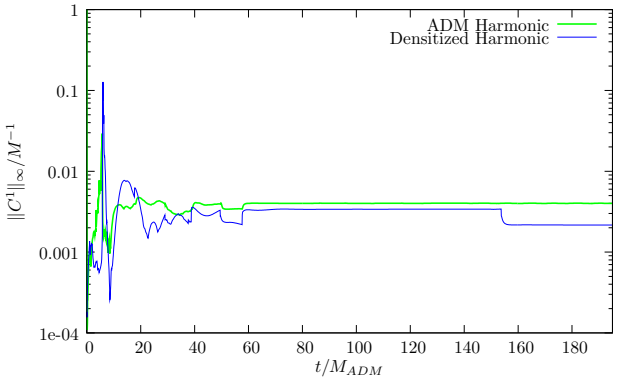


Figure 5.6: ratio of yz/xy-plane circumferences of the common apparent horizon after merger.

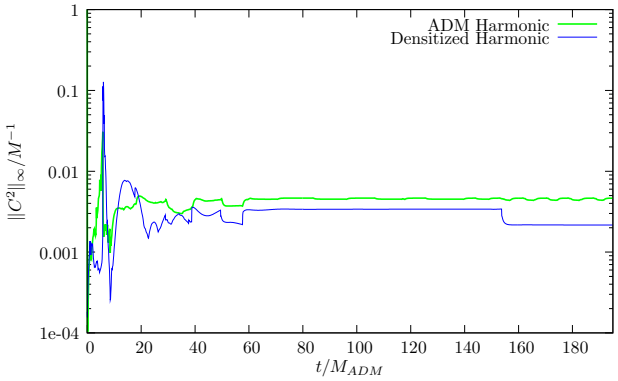
I use the constraint equations 2.5.9 and 2.5.10 posed on the spatial hyper-surfaces to estimate the error of the numerical evolution. This is done by comparing the gauge source functions computed by using the constraint equations 2.5.40 and 2.5.41 against their definition $n_\nu \Gamma^\nu = 0 = h_\nu{}^a \Gamma^\nu$ itself. Then the constraint violations give a measure for the accuracy of the numerical evolution scheme. Figures 5.7 now show the infinity norms of the harmonic constraint violations of both the 'ADM Harmonic' and the 'Densitized Harmonic' evolution system. I see a similar behavior with progressing time evolution. The peak in the constraint violation between $5M$ and $6M$ reflects the merger of the two black holes with strong changes in the metric components and thus also stronger constraint violations during this time. Both runs are not using constraint preserving boundary conditions. The infinity norms of the constraint violations show that the maximum error is not affected by that, which is the expected behavior for this kind of boundary conditions. The infinity norm of the constraint violations is basically staying constant after $50M$. This reflects the final black hole after some quasi normal ringing settling down to its non-spinning end configuration. Figure 5.8 then show the differences in the constraint violations between the 'ADM'- and 'Densitized' Harmonic formulations. I here see again the biggest differences at time of merger.
I see that the constraint violations are of the same order for the two codes, except for a short peak at merger time, where the Densitized Harmonic formulation is two orders of magnitude larger.



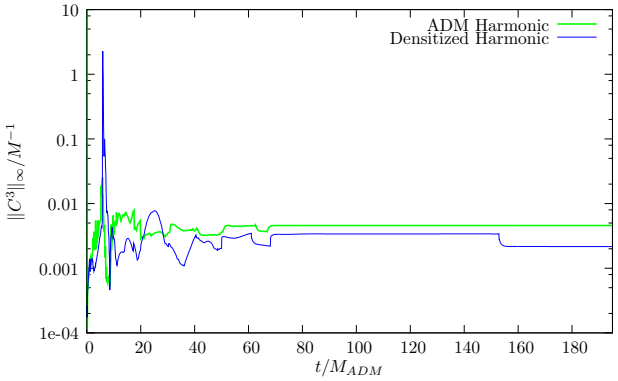
(a) Infinity norm of the constraint violation t component



(b) Infinity norm of the constraint violation x component



(c) Infinity norm of the constraint violation y component



(d) Infinity norm of the constraint violation z component

Figure 5.7: Infinity norms of the harmonic constraints

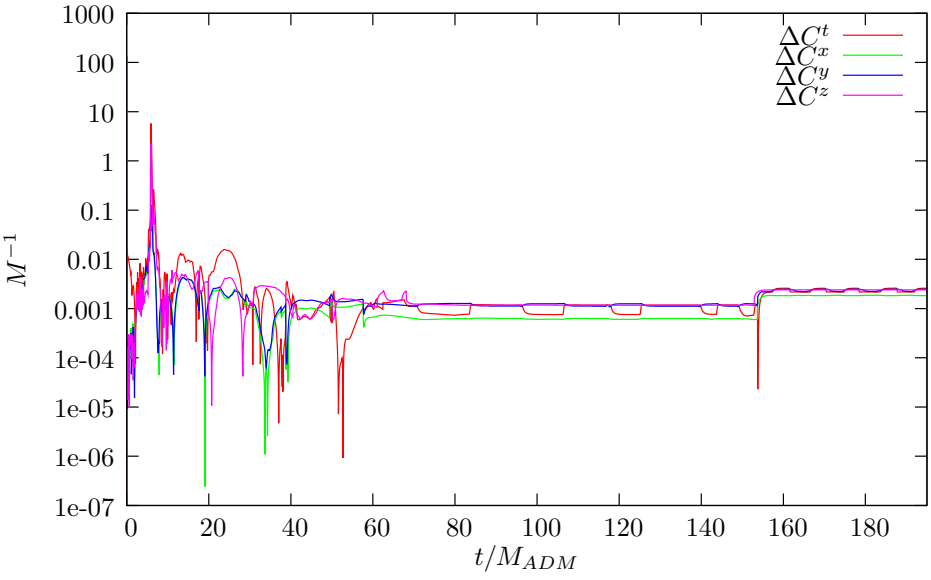


Figure 5.8: Difference between 'ADM' and 'Densitized' Harmonic formulations for the infinity norms of the harmonic constraints.

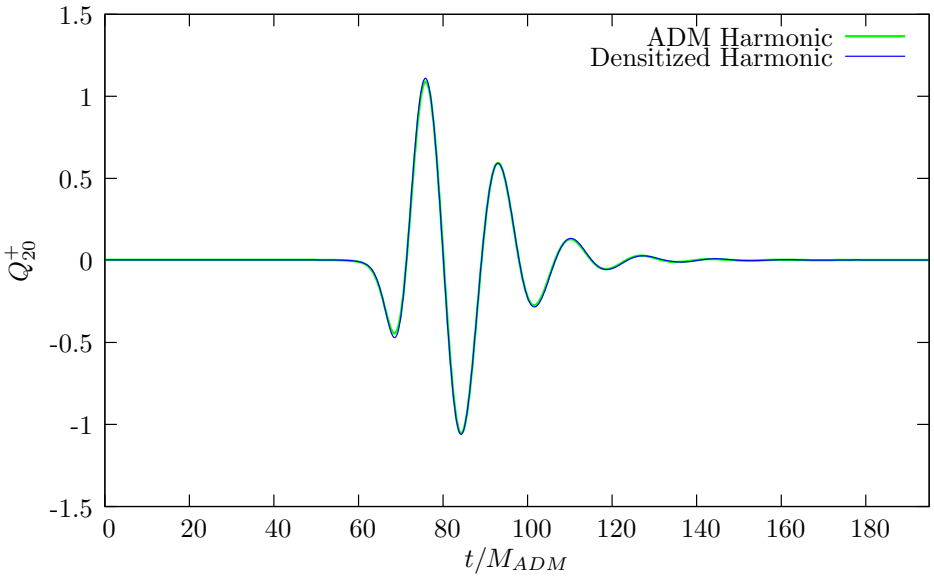


Figure 5.9: Zerilli waveform for the head-on collision extracted at $R = 60M$.

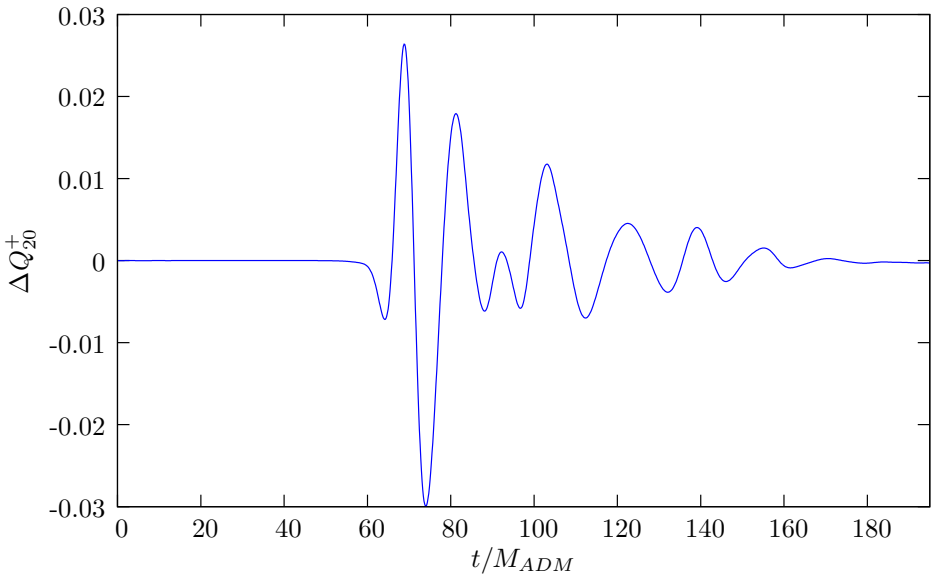


Figure 5.10: Difference in the Zerilli waveform for the head-on collision extracted at $R = 60M$ between the 'ADM' and 'Densitized' Harmonic formulations.

Figure 5.9 shows the $l = 2, \quad m = 2$, even parity multipole of the Zerilli function Q^+_{20} extracted at $R = 60M$. I see a well captured quasi-normal ringing after the merger of the final black hole. Fig. 5.10 shows the difference in the $l = 2, \quad m = 2$, even parity multipole of the Zerilli function Q^+_{20} between the 'ADM'- and 'Densitized' harmonic formulations. I see good agreement between the two different evolution systems with a relative difference of three percent in amplitude.

In this chapter I have carried out evolutions of a binary black hole head-on collision using the 'ADM Harmonic' evolution code and demonstrated that they are stable and accurate.
I have studied horizon porperties and showed that they are comparable to the 'Densitized Harmonic' evolution code. In addition I have compared and showed good agreement for the constraint violations and the gravitaional wave signal extracted between the 'ADM Harmonic' and 'Densitized Harmonic' evolution codes.

6 Concluding remarks

6.1 Summary and general remarks

In this thesis I have presented an harmonic evolution system formulated in the ADM variables. For this I rederived and implemented the evolution equations and discussed the auxiliary variables I introduced to reduce the system to first order in time. In order to test the accuracy of the evolution system I have run convergence tests using a shifted gauge wave testbed for which an analytic solution already has been implemented. I was able to show that the ADM variables converge to the analytical solution in that case. To damp exponential growth in the constraints and allow for stable evolution in the case of black holes present I have implemented, tested and compared different constraint damping terms. I could show that all constraint damping terms I invetigated work effectively for the evolutions presented in this thesis. Furthermore I had to apply dissipation operators to my the right hand sides of the evolution equations to successfully kill high frequency noise arising in the evolution of black hole initial data.
I could show that the evolution system is able to evolve single black holes successfully on long time scales and shows no sign of instability. In addition I have compared results achieved for a hean-on black-hole collision with a densitized harmonic evolution system implemented by Bela Szilagyi and found very good agreement in the black-hole horizon properties, the constraint violations and the extracted gravitational waveform.

6.2 Future outlook

One of the primary motivations to implemented the presented evolution system was to investigate its ability to evolve puncture initial data without using excision techniques. I investigated the effects of different numerical approaches which seemed promising. Although I haven't been able to successfully evolve puncture initial data in general I have found some insights which motivate future development. Additionally the choice of gauge source functions, the implementation of constraint-preserving SBP (summation by parts) boundary conditions leave room for future development and improvement of the code.

7 Deutsche Zusammenfassung (German summary)

Es öffnet sich ein neues Fenster zum Universum. Nachdem Gravitationswellendetektoren wie GEO600 (Deutschland), VIRGO (Italien), und LIGO (USA) in Betrieb gegangen sind, wird es vielleicht schon bald moeglich sein Gravitationswellensignale von extremen kosmischen Ereignissen, wie dem Verschmelzen von schwarzen Löchern und Neutronensternen oder dem Kollabieren von Sternenkernen (Supernovae) aufzunehmen und zu klassifizieren. Dadurch hofft man neue Einsichten in die Rolle der Gravitation als eine der fundamentalen Wechselwirkungen in der Natur und über die Entwicklung unseres Universums ganz allgemein zu erlangen. Da Gravitationswellensignale generell allerdings eine sehr schwache Amplitude haben, ist es ein kompliziertes und schwieriges Vorhaben Gravitationswellenformen überhaupt nachzuweisen. Deswegen ist eines der Hauptziele der numerischen Relativitätstheorie die Ereignisse wie das Verschmelzen von schwarzen Löchern oder Supernovae zu simulieren und die zu erwartenden Wellenformen vorherzusagen.

Allerdings stellte sich heraus, dass die numerische Evolution der Einstein Gleichungen mit wesentlich mehr Schwierigkeiten verbunden ist, als ursprünglich angenommen. Die Modellierung eines Binärsystems bestehend aus zwei verschmelzenden schwarzen Löchern in einer Vakuum-Raumzeit induziert schwerwiegende numerische Probleme und ist deswegen zu einer der Hauptherausforderungen der numerischen Relativitaetstheorie geworden. Die Einstein Gleichungen, ein System gekoppelter, quasi-linearer partieller Differentialgleichungen (PDEs), können in unterschiedlichen Formulierungen ausgedrückt werden. In den sogenannten harmonischen Formulierungen werden durch die Wahl von speziellen Eichfunktionen die Evolutionsgleichungen für die Metrikkomponenten in die Form von Wellengleichungen gebracht. Diese Formulierung der Gleichungen besitzt aus der mathematischen Analyse bekannte und vorteilhafte Stabilitäts- und Wohldefiniertheitseigenschaften. Bis jetzt wurden diese harmonischen Gleichungen durch die direkte Diskretisierung der vierdimensionalen Raumzeitmannigfaltigkeit gelöst und evolviert. Die BSSN (Baumgarte, Shapiro und Shibata, Nakamura) Formulierung hingegen basiert auf dem 3+1 Formalismus, bei dem die Raumzeit in Zeit- und Raumanteile aufgespaltet wird und diese dann separat behandelt werden. Die BSSN Formulierung ist eine Modifikation und Weiterentwicklung der ADM-Gleichungen (Arnowitt, Deser, Misner), die zwar numerisch stabil, aber nicht optimal für die Evolution sind. Ein weiteres Problem ist das Verfahren mit dem man Singularitäten, die in den Evolutionsvariablen auftreten, auf dem endlichen numerischen Gitter, welches benutzt wird um die Simulationsdomäne zu diskretisieren, behandelt. Hier haben sich verschiedene Verfahren entwickelt. In einer Herangehensweise, der Ausschneide-Methode, wird die Singularität aus dem Gitter ausgeschnitten, genauer man schneidet eine von der Raumzeit kausal unabhängige Region aus (z.B. die Region innerhalb eines Ereignishorizonts eines schwarzen Lochs). Dafür muss man die Evolution dieses Horizonts beobachten, um zu jeder Zeit die richtige Region auszuschneiden. Allerdings

führt diese Technik zu komplizierten Gitterkonfigurationen und man muss für den Rand der ausgeschnittenen Region extra Randbedingungen wählen. Dieses kann zu Instabilitäten in der numerischen Evolution führen und verlangsamt diese. Ein anderer Ansatz ist die sogenannte 'Moving-Punctures'-Methode. Hierfür werden Anfangsdaten benutzt, bei denen die singulären Terme innerhalb des Ereignishorizonts in einen analytischen Ausdruck absorbiert werden. Diese Anfangsdaten werden dann ohne Ausschneidetechniken evolviert. Während die BSSN Evolutionssysteme 'Puncture'-Anfangsdaten ohne größere Probleme evolvieren können und stark davon profitieren keine weiteren Ausschneidetechniken anwenden zu müssen, ist es bis jetzt noch nicht gelungen 'Puncture'-Anfangsdaten in einem harmonischen Evolutionssystem zu evolvieren.

In der vorliegenden Arbeit präsentiere ich ein harmonisches Evolutionssystem, dass im 3+1 Formalismus formuliert ist, und untersuche seine Fähigkeiten Puncture-Anfangsdaten zu evolvieren. Es ist auf den ADM Variablen lapse Funktion α, shift Vektor β, 3-Metrik g_{ab} und der äußeren Krümmung k_{ab} basiert und benutzt Eichquellfunktionen, die ebenfalls im 3+1 Formalismus implementiert sind. Diese Arbeit besteht aus zwei Teilen. Im ersten Teil beschreibe ich die Einstein Gleichungen und ihre Formulierungen für die numerische Relativitätstheorie. Ich beschreibe kurz die ADM-Gleichungen, um den 3+1 Formalismus zu motivieren und konzentriere mich im Folgenden darauf zwei harmonische Evolutionssysteme zu beschreiben. Ich vergleiche ein vier-dimensionales 'Densitized' harmonisches Evolutionssystem, implementiert von Bela Szilagyi, mit dem im 3+1 Formalismus formulierten Evolutionssystem auf dem diese Arbeit basiert, zuerst beschrieben von Helmut Friedrich. In einem weiteren Kapitel beschreibe ich die grundlegenden numerischen Techniken, die finiten Differenzen- und Dissipationsoperatoren sowie eine Zerlegung der Metrikkomponenten für die finiten Differenzenoperatoren, die ich in der Implementierung des Evolutionssystems benutze.
Im zweiten Teil der Arbeit konzentriere ich mich auf die Präsentation der erzielten Ergebnisse. Ich beschreibe Ergebnisse für verschiedene Vakuum-Raumzeit Systeme bestehend aus einzelnen und binären schwarzen Löchern. Im ersten Kapitel konzentriere ich mich auf Ergebnisse, die ich bei der Simulation von einzelnen schwarzen Löchern erzielt habe. Zuerst analysiere ich die Stabilität und Robustheit des implementierten Evolutionssystems für die Langzeitevolution von einzelnen schwarzen Löchern ohne Drehimpuls und untersuche den Einfluss von verschiedenen Parameterm für die 'Constraint-Damping'-Technik. Die 'Constraint-Damping'-Technik ist hierbei essentiell um mit diesem Evolutionssystem Vakuum-Raumzeitkonfigurationen mit schwarzen Löchern hinreichend lange zu simulieren. Weiterhin zeigen die Resultate, dass die 'Constraint-Damping'-Technik in einem Parameterbereich um Größenordnung 1 effektiv für eine große Anzahl an Parameterwerten ist.
Im zweiten Abschnitt dieses Teils präsentiere ich Daten und Visualiserungen für eine Frontalkollision von zwei schwarzen Löchern ohne Drehimpuls. Ich beschreibe die Verschmelzung der Ereignishorizonte, extrahiere das Gravitationswellensignal und untersuche die Zeitentwicklung der harmonischen 'Constraints'. Zur Validierung der erzielten Ergebnisse vergleiche ich diese mit den Ergebnissen für die identische Anfangsdatenkonfiguration, evolviert mit dem 'Densitized' harmonischen Evolutionssystem. Ich sehe hierbei eine sehr gute Übereinstimmung zwischen den beiden unabhängig formulierten und implementierten Evolutionssystemen, was die Richtigkeit der Ergebnisse untermauert.

8 Appendix A

8.1 The Method of Lines

The method of lines (MoL) [2], [3] involves the separate discretization of space and time which enables one to independently analyze the stability properties of the time evolution scheme with the same spatial finite difference operator S. This method therefore allows for different discretization techniques of the time integration and the spatial difference scheme. The time integration is usually done with finite differences, whereas the spatial discretization can be of any kind of finite difference, finite element or even spectral method [4] techniques. It is most common to split the time derivatives in a system of PDEs to first-order, and since MoL is equivalent to transforming the PDEs to ordinary differential equations ODEs, we can apply ODE integrators like the iterative Crank-Nicholson or the Runge-Kutta scheme. To illustrate the idea, we consider the semi-discrete system in which only the time derivative is discretized and the spatial derivatives are left continuous. Let $\mathbf{u}$ be an array of dynamical fields and S a continuous spatial differential operator. The evolution of $\mathbf{u}$ can be written as

$$\partial_t \mathbf{u} = S(\mathbf{u}) \tag{8.1.1}$$

The continuous flow of time is then replaced by a succession of discrete time instants $\{t_n\}$, and we label $\mathbf{u}^n = \mathbf{u}(t_n)$. The Euler step (forward time difference) is a first-order scheme and can be used as a simple approximation. It reads

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t S(t_n, \mathbf{u}^n) \tag{8.1.2}$$

Since the Euler step is an insufficient approximation and is also unconditional unstable [2] in most cases, we need better approximations. A scheme that is sufficient in most cases is the classical fourth-order Runge-Kutta scheme

$$\begin{aligned}
k_1 &= S(t_{n-1}, \mathbf{u}^{n-1}) \\
k_2 &= S\left(t_{n-1} + \frac{\Delta t}{2}, \mathbf{u}^{n-1} + \frac{\Delta t}{2} k_1\right) \\
k_3 &= S\left(t_{n-1} + \frac{\Delta t}{2}, \mathbf{u}^{n-1} + \frac{\Delta t}{2} k_2\right) \\
k_4 &= S(t_{n-1} + \Delta t, \mathbf{u}_{n-1} + \Delta t\, k_3) \\
\mathbf{u}^n &= \mathbf{u}^{n-1} + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) + \mathcal{O}(\Delta t^5).
\end{aligned} \tag{8.1.3}$$

Of course, the time integrator should have at least the same accuracy as the spatial discretization. Otherwise, the overall accuracy would be limited by the time integrator. Another important issue is the fact that not all time integrators provide an (un)conditional stable scheme. By means of Fourier mode analysis [2] of the discretized PDE, one can determine the stability

regions of a given time integrator scheme and decide whether it provides a stable scheme or not. For example, the wave equation fails to work with the Euler step or with a second-order Runge-Kutta scheme [5].

In *Cactus*, MoL is implemented in the thorn `MoL`. One only needs to provide the spatial difference operator, i.e. the RHS of (8.1.1). The ODE time integration is then carried out by `MoL`. Supported integration schemes include iterative Crank-Nicholson (ICN) with N iterations[1] and Runge-Kutta (RK) up to fourth order.

8.2 Convergence

Convergence is a necessary criterion for any numerical approximation to be a reasonable approximation to a continuum solution. In the limit of a vanishing grid size $\Delta x, \Delta t \to 0$, the approximation should result in a continuum solution. Every numerical approximated system should therefore be tested for convergence, since the order of the truncation (or approximation) error is a measure of accuracy of a given scheme [2], [6] and can help to assure a bug-free code. There are two ways of showing convergence. If one knows the analytical solution of a given problem, one can make use of the 2-level convergence check. If the analytical solution is unknown, one can still test for convergence with the 3-level convergence check.
Let us suppose the solution v to a PDE is reasonably smooth. We can then Taylor-expand it around a gridpoint x_i in order to estimate the function in $x_{i-1} = x_i - h$ and $x_{i+1} = x_i + h$. We get

$$v_{i-1} = v_i - v_i' h + \frac{1}{2} v_i'' h^2 - \frac{1}{6} v_i''' h^3 + \frac{1}{24} v_i^{(4)} h^4 + \mathcal{O}(h^5) \tag{8.2.1}$$

$$v_{i+1} = v_i + v_i' h + \frac{1}{2} v_i'' h^2 + \frac{1}{6} v_i''' h^3 + \frac{1}{24} v_i^{(4)} h^4 + \mathcal{O}(h^5). \tag{8.2.2}$$

$$\tag{8.2.3}$$

Inserting these expressions into a difference approximation of a given spatial derivative operator, one can determine its order of accuracy. Let us take, for example, the centered difference approximation to ∂_x applied tp a function v. We have

$$\frac{v_{i+1} - v_{i-1}}{2h} = v_i' + \frac{1}{6} v_i''' h^2 + \mathcal{O}(h^4). \tag{8.2.4}$$

We can see here that the difference approximation is second-order accurate, since terms of higher order are truncated. If higher order errors a negligible, i.e. if we are in the convergent regime, we would expect that by doubling the number of gridpoints, i.e. decreasing the gridsize h by one half, the truncation error should decrease by a factor of 4. By knowing the analytical solution u, we therefore expect [5]

$$\begin{aligned} v_{2h} - u &= C(2h)^2 = 4Ch^2 \\ v_h - u &= Ch^2 \end{aligned}$$

[1] Using a fixed number of iterations misses the idea of iterating until a previously defined accuracy is reached.

where the constant C is independent of the grid resolution h and only depends on u. More generally, we consider higher order schemes by

$$\begin{aligned} v_{2h} - u &= C(2h)^\alpha = 2^\alpha C h^\alpha \\ v_h - u &= C h^\alpha, \end{aligned} \tag{8.2.5}$$

where α is the order of accuracy of the difference approximation. A fourth-order scheme should therefore decrease the error by a factor of 16 when reducing the gridstep h by one half. We call this scaling factor between the error of both resolutions the **convergence rate** Q (in other sections also called C, but here we want to avoid confusion with the constant C).

If the analytical solution is not known, one can still use the 3-level convergence test. It requires to run a given problem at least at 3 different grid resolutions, e.g. h, $2h$ and $4h$. With negligible higher-order errors, we expect [5]

$$\begin{aligned} v_{4h} &= u + C(4h)^\alpha = u + 4^\alpha C h^\alpha \\ v_{2h} &= u + C(2h)^\alpha = u + 2^\alpha C h^\alpha \\ v_h &= u + C h^\alpha. \end{aligned}$$

We get

$$v_{4h} - v_{2h} = 2^\alpha (v_{2h} - v_h). \tag{8.2.6}$$

The convergence rate for a second-order scheme ($\alpha = 2$) should therefore be $Q = 4$ and for a fourth-order scheme ($\alpha = 4$) $Q = 16$. It is possible to consider the convergence rate pointwise, i.e. at each gridpoint of the coarsest grid, or to take the discretized version of the L_2-norm $\|v\|_2 = \sqrt{h \sum_{i=0}^{N} v_i^2}$ with which the convergence rate at any given timestep can be written as [**?**]

$$Q(t) = \frac{\|v_{4h} - v_{2h}\|_2}{\|v_{2h} - v_h\|_2}, \qquad \text{or} \qquad Q(t) = \frac{\|v_{2h} - u\|_2}{\|v_h - u\|_2} \tag{8.2.7}$$

If the finite difference scheme is converging, we should find in the limit of infinite resolution [**?**]

$$\lim_{h \to 0} Q(t) = 2^\alpha. \tag{8.2.8}$$

By using MoL, one can discretize the time integration and the spatial derivatives separately. In order to maintain the accuracy of the spatial difference scheme, it is clear that the accuracy of the time integrator should at least be of the same order.

8.3 Well-posedness

Well-posedness is a reasonable requirement for any given Cauchy initial (boundary) value problem (IBVP). Roughly speaking, it states that a well-posed problem should have a solution, that this solution is unique and that it should depend continuously on the problem's data. For a

given system of PDEs, an IBVP on the domain $x \in [0,1]$ is given by

$$\begin{aligned} \mathbf{u}_t &= P\left(x,t,\frac{\partial}{\partial x},\frac{\partial^2}{\partial x^2},...\right)\mathbf{u}+F, \qquad t \geq t_0, \\ \mathbf{u}(x,t_0) &= \mathbf{f}(x) \\ L_0\left(t,\frac{\partial}{\partial x},\frac{\partial^2}{\partial x^2},...\right)\mathbf{u}(0,t) = g_0, & \qquad L_1\left(t,\frac{\partial}{\partial x},\frac{\partial^2}{\partial x^2},...\right)\mathbf{u}(1,t) = g_1, \end{aligned} \tag{8.3.1}$$

where $\mathbf{u} = (u^1,...,u^m)$ is a set of dynamical fields, $P = \sum_{\nu=0}^{p} A(x_\nu,t)\frac{\partial^\nu}{\partial x^\nu}$ is a differential operator of order p with smooth coefficients A, L_0 and L_1 are differential operators usually of order $r < p-1$ that incorporate the boundary conditions, and F is a forcing function. This results in the following definition for homogeneous boundary conditions $F = g_0 = g_1 = 0$ [2]

Definition 8.3.1. *The problem (8.3.1) with $F = g_0 = g_1 = 0$ is **well-posed** if, for every $\mathbf{f} \in C^\infty$ that vanishes in a neighborhood of $x = 0, 1$, it has a unique solution that satisfies the estimate*

$$\|\mathbf{u}(,t)\| \leq K\mathrm{e}^{\alpha(t-t_0)}\|\mathbf{u}(,t_0)\| \tag{8.3.2}$$

where K and α do not depend on $\mathbf{f}$ and t_0.

For general inhomogeneous boundary data, we have

Definition 8.3.2. *The problem (8.3.1) is **strongly well-posed** if it is well-posed and instead of (8.3.2), the solution satisfies the estimate*

$$\|\mathbf{u}(,t)\|^2 \leq K(t,t_0)\left(\|\mathbf{u}(,t_0)\| + \int_{t_0}^{t}(\|F(,\tau)\|^2 + |g_0(\tau)|^2 + |g_1(\tau)|^2)d\tau\right) \tag{8.3.3}$$

$K(t,t_0)$ is supposed to be a function that is bounded in every finite time interval, i.e. $\infty > \|K(t,t_0)\|_\infty$ and does not depend on the data.

In particular cases, stronger estimates might be obtained such that the functional growth in each step can even be suppressed entirely. It is important to note, that not the PDE itself is well-posed (or ill-posed), but rather the entire problem including initial and boundary data.

8.4 Overview of stability analysis

Analyzing the numerical stability properties of any finite difference scheme is an important issue for numerical approximations to PDEs and is the discrete analogue of well-posedness. Stability analysis helps to determine whether a given difference scheme can principally run for ever or 'blows up' and 'crashes'. 'Blow-ups' usually occur because exponentially growing modes develop which ultimately lead to overflow errors.
A general difference approximation of a linear system of PDEs can be written as [2]

$$\begin{aligned} Q_{-1}\mathbf{u}^{n+1} &= \sum_{\sigma=0}^{q} Q_\sigma \mathbf{u}^{n-\sigma}, \qquad n = q,\, q+1,... \\ \mathbf{u}^\sigma &= \mathbf{f}^{(\sigma)}, \qquad \sigma = 0,1,...,q \end{aligned} \tag{8.4.1}$$

where the Q_σ are difference operators. By assuming that Q_{-1} is uniformly bounded[2] and has a uniformly bounded inverse Q_{-1}^{-1} as grid and time stepsize $h, k \to 0$, we can advance the solution step by step in time. For simplicity, I restrict myself to the case $q = 0$, which means that the discretized time scheme only involves the current and next timelevel. A more general discussion can be found in [2], although the following statements do not lose any validity.
We can rewrite (8.4.1) as

$$\begin{aligned} \mathbf{u}^{n+1} &= Q(t_n)\mathbf{u}^n, \qquad n = 0, 1, ... \\ \mathbf{u}^0 &= \mathbf{f}, \end{aligned} \tag{8.4.2}$$

where $Q(t_n) = Q_{-1}^{-1}Q_0$ and f defines the initial data on the initial timestep. The discrete solution operator can be defined as

$$\mathbf{u}^n = S_h(t_n, t_\nu)\mathbf{u}^\nu. \tag{8.4.3}$$

We then have the following definition:

Definition 8.4.1. *Given the constants α_S, C and K_S, the difference approximation (8.4.2) is called **stable** for $h \le h_0$, if for all h the discrete operator norms satisfy*

$$\|Q_{-1}^{-1}\|_h \le C, \qquad \|S_h(t_n, t_\nu)\|_h \le K_S \mathrm{e}^{\alpha_S(t_n - t_\nu)}. \tag{8.4.4}$$

The stability requirement results in the estimate

$$\|\mathbf{u}^n\|_h \le K(t_n)\|\mathbf{f}\|_h, \qquad K(t_n) = K_S \mathrm{e}^{\alpha_S t_n}, \tag{8.4.5}$$

stating that the growth of the solution at later times is bounded by the initial data times some exponential factor. We allow here for an exponential factor in order to factor in possible exponential solutions. However, in specialized cases where the continuum problem contains no exponentially growing solutions, more restrictive definitions might be used, and the functional growth might even be suppressed entirely.

For practical purposes, a more refined stability definition is useful.

Definition 8.4.2. *Suppose the continuous solution operator $S(t, t_0)$ is bounded by the estimate*

$$\|S(t, t_0)\|_{op} \le K \mathrm{e}^{\alpha(t - t_0)}. \tag{8.4.6}$$

[2]

Theorem 8.4.1. *(5th principle of linear functional analysis: principle of uniformly boundedness) [?]*
Let $(V, \|\cdot\|_V)$, $(W, \|\cdot\|_W)$ be normed K-vectorspaces, $S \subseteq V$ a set of second category in $(V, \tau_{\|\cdot\|_V})$ and $F \subseteq L(V, W)$. Then it is equivalent:

1. $\forall s \in S: \quad \sup\{\|f(s)\|_W \mid f \in F\} < \infty$
   (i.e. F is pointwise bounded in S)
2. $\sup\{\|f\|_{op} \mid f \in F\} < \infty$
   (i.e. F is bounded in $(L(V, W), \|\cdot\|_{op})$)

A difference approximation is then said to be ***strictly stable*** *if in addition to (Definition 8.4.1), we have*

$$\|S_h(t_n, t_0)\|_h \leq K_S \mathrm{e}^{\alpha_S(t_n - t_0)} \tag{8.4.7}$$

where

$$\alpha_S \leq \alpha + \mathcal{O}(k). \tag{8.4.8}$$

This provides for difference operators that are of order k, because it is possible to show [2] that perturbations to a difference operator of order k, e.g. $\mathbf{u}^{n+1} = (Q_0 + k\mathbb{1})\mathbf{u}^n$ automatically satisfy (Definition 8.4.1) if their unperturbed part satisfies (Definition 8.4.1). But this may lead to an unwanted behaviour, since terms of order k can indeed play an important role for stability. The functional growth of (8.4.6) and (8.4.7) can be adapted to the problem, resulting in a more restrictive estimate.
Different analysis techniques exist, where the most popular certainly is the **Fourier mode analysis**. Using the Method of Lines, the stability properties of the PDE can be analyzed in two steps. First, the semi-discretized PDE in space is being transformed to Fourier space, i.e. the spatial difference operators are applied to the Fourier modes of form $\xi \exp(i\omega h x_j)$, where ξ is the amplitude of the Fourier mode, ω is the wave number, h is the grid spacing and j is the index of the grid point. For example, with the discrete Fourier mode $\widehat{u}_j = \xi \exp(i\omega h x_j)$, the centered finite difference operator approximating $\partial/\partial x$ becomes

$$\widehat{D}_0 \widehat{u}_j = \frac{\xi \exp(i\omega h x_{(j+1)}) - \xi \exp(i\omega h x_{(j-1)})}{2h} = \xi \exp(i\omega h x_j) i \frac{\sin(\omega h)}{h}. \tag{8.4.9}$$

Solving then the Fourier-transformed PDE (which in the semi-discretized sense is an ODE)

$$\partial_t \widehat{\mathbf{u}} = \widehat{Q}\widehat{\mathbf{u}} \tag{8.4.10}$$

reveals whether we get solutions of type

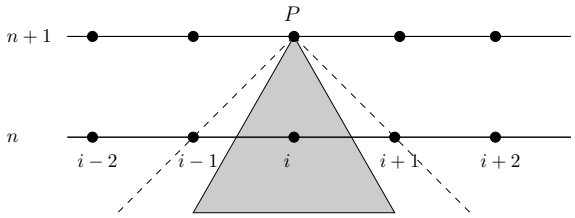
$$\widehat{u} = \mathrm{e}^{\lambda t}, \qquad \mathrm{Re}(\lambda) = 0, \tag{8.4.11}$$

where the λ are the eigenvalues of $\widehat{Q}$.

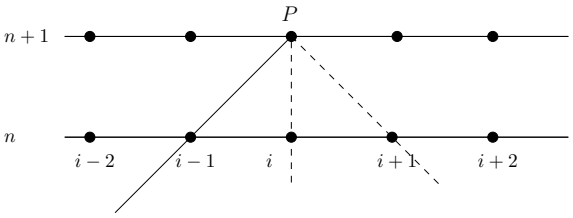
It is a necessary condition for stability that these eigenvalues satisfy

$$|\lambda| \leq \mathrm{e}^{\alpha_S k}. \qquad \text{(the von Neumann condition).} \tag{8.4.12}$$

An ODE method will then yield a stable evolution if the λk lie within the method's region of stability. It is beyond the scope of this section to describe how these regions can be determined, but it is clear for the case (8.4.11) that it should encompass parts of the imaginary axis. In this case, the Euler step or low-order RK schemes fail to give an (un-)conditional stable scheme, since they have stability regions tangent to the imaginary axis at the origin.
Another technique is the **energy method** [2]. It provides another way of proving stability and is simpler to apply than Fourier analysis in cases of PDEs with non-constant coefficients. In this approach, no transformations will be used and the calculations are carried out in physical space. Instead, we construct a suitable norm such that the growth in each step will not exceed



(a) Suppose a problem like the wave equation. In a correct discretization, the theoretical (gray triangle) is enclosed by the numerical domain of dependence (dashed triangle). Here, the difference scheme approximates the value at P with a spatial stencil that spans over the grid points $i-1, i, i+1$.



(b) Incorrect discretization, e.g. for an advection equation. The theoretical domain of dependence (triangle on the left between the straight line and the dashed one in the middle) does not coincide with the numerical one (dashed triangle on the right). A correct discretization would involve $i-1, i$ (upwind scheme) instead of $i, i+1$ (downwind scheme).

Figure 8.1: Two examples of the theoretical domain of dependence and the numerical domain of dependence. It becomes apparent that, since the numerical domain of dependence has to enclose the theoretical domain of dependence and is therefore usually bigger, a-causal information from outside the theoretical domain of dependence can influence the simulation. This effect will grow for very stretched stencils.

$e^{\alpha k}$. With the help of this method, we can construct difference schemes that are strictly stable.

Another necessary condition for numerical stability is the **Courant-Friedrich-Lewy (CFL) condition**, stating that the numerical domain of dependence should always contain the theoretical domain of dependence (fig. 8.1). The theoretical domain of dependence of the value of a solution at one point only depends on the characteristic lines crossing through that point. In other words, the largest characteristic speed v_{max} along every given direction n_i cannot exceed the corresponding numerical speed

$$v_{max} < n_i v^i_{num}, \qquad v^i_{num} = \frac{s\Delta x^i}{\Delta t}, \tag{8.4.13}$$

where s is the stencil width, i.e. the maximum number of points involved in a difference scheme along any direction [3].

9 Appendix B

9.1 The Cactus computational toolkit

Cactus is an open source problem-solving environment [7], [8], [9] that incorporates different programming languages such as *C/C++* and *Fortran*. Its primary application is for large-scale grid-computations, and it is heavily parallelized to allow for data distribution and processing across multiple cluster-nodes of a supercomputer. It is designed to run on different architectures and is being collaboratively developed and maintained between different groups across the world (AEI, LSU and others).

Cactus is made up of a core module (the 'flesh') which connects to application modules ('thorns') that can overtake various tasks like I/O, MoL time integration, mesh-refinement (*Carpet* [10], [11]), multipatch infrastructure (*GZPatchSystem* [12]) or horizon finding [13] - just to enumerate a few. Self-written (physics-)thorns can then interact with other thorns and process or calculate grid-data for a given problem. The flesh itself provides a message passing interface (MPI) [16], [17] parallelization infrastructure with variable types (grid functions/arrays) that are ready to be distributed across multiple processors with local memory[1], i.e. that are not necessarily sharing one common memory block[2]. Furthermore, the 'flesh' handles the scheduling of the various thorns that are supposed to be compiled-in and activated for a specific task. There are a number of so-called schedule bins that are being executed during runtime, e.g. `CCTK_INITIAL` for routines that are supposed to run on the initial step like providing initial data or `CCTK_EVOL` for evolution routines which is being looped through until the final time is reached.

The thorns can be written in any of both, *C/C++* or *Fortran*. It is even possible to have a mixture of *C/C++* or *Fortran* code within one single thorn. A collection of thorns is called an 'arrangement'.

After compilation of *Cactus*, single thorns can be inherited from inside *Cactus* by consigning a parameter (par-)file upon execution. These par-files specify which thorns shall be activated during this run and let the user set up the grid's $\Delta x, \Delta t$, the time integrator if the MoL thorn is being used, start- and end-time and other parameters that are thorn-specific.

9.1.1 An example of Cactus scheduling

To illustrate the idea of *Cactus* scheduling, I have extracted the *Cactus* screen output on startup. Schedule bins are marked in squared brackets. The various routines of activated thorns display a short description of them beginning with the thorn's name. This output has

[1] The AEI's own cluster 'Peyote' [18] is made up of 195 dual-CPU nodes with a total memory of 1132 GB and 9x1.5 terabytes of disk space. It develops a total speed of 706 gigaflops. Further examples of computers with distributed memory include IBM Blue Gene, Cray T3E series or Beowulf-clusters (PCs connected via Ethernet).

[2] Such as Enterprise 10.000 or SUN's Fire 15k

been generated during one of the single black hole runs to test the stability of the evolution system.

```
--------------------------------------------------------------------------------

      10
  1   0101       ************************
  01  1010 10      The Cactus Code V4.0
 1010 1101 011      www.cactuscode.org
  1001 100101    ************************
    00010101
     100011     (c) Copyright The Authors
      0100      GNU Licensed. No Warranty
      0101

--------------------------------------------------------------------------------

Cactus version:    4.0.b16
Compile date:      Jul 31 2007 (18:17:30)
Run date:          Jul 31 2007 (19:13:48+0200)
Run host:          node084.damiana.admin
Working directory: /data34/moesta/pe_d2_a23
Executable:        cactus_hadm-0(mpi:3761@node084)
Parameter file:    /data34/moesta/pe_d2_a23/pe_d2_a23.par
--------------------------------------------------------------------------------

Activating thorn Cactus...Success -> active implementation Cactus
Activation requested for
--->IOUtil<---
Activating thorn IOUtil...Success -> active implementation IO
Activation requested for
--->LocalInterp AEILocalInterp<---
Activating thorn AEILocalInterp...Success -> active implementation AEILocalInterp
Activating thorn LocalInterp...Success -> active implementation LocalInterp
Activation requested for
--->Carpet CarpetLib CarpetInterp CarpetReduce CarpetSlab CarpetIOHDF5<---
Activating thorn Carpet...Success -> active implementation Driver
Activating thorn CarpetInterp...Success -> active implementation interp
Activating thorn CarpetIOHDF5...Success -> active implementation IOHDF5
Activating thorn CarpetLib...Success -> active implementation CarpetLib
Activating thorn CarpetReduce...Success -> active implementation reduce
Activating thorn CarpetSlab...Success -> active implementation Hyperslab
Activation requested for
--->NaNChecker<---
Activating thorn NaNChecker...Success -> active implementation NaNChecker
Activation requested for
--->Boundary CartGrid3D CoordBase SymBase<---
Activating thorn Boundary...Success -> active implementation boundary
Activating thorn CartGrid3D...Success -> active implementation grid
Activating thorn CoordBase...Success -> active implementation CoordBase
Activating thorn SymBase...Success -> active implementation SymBase
Activation requested for
--->CarpetRegrid<---
Activating thorn CarpetRegrid...Success -> active implementation CarpetRegrid
Activation requested for
--->Time<---
Activating thorn Time...Success -> active implementation time
Activation requested for
--->MoL<---
Activating thorn MoL...Success -> active implementation MethodOfLines
Activation requested for
--->CarpetIOBasic<---
Activating thorn CarpetIOBasic...Success -> active implementation IOBasic
Activation requested for
--->CarpetIOScalar<---
Activating thorn CarpetIOScalar...Success -> active implementation IOScalar
Activation requested for
--->CarpetIOASCII<---
Activating thorn CarpetIOASCII...Success -> active implementation IOASCII
```

```
Activation requested for
--->ADMBase SpaceMask StaticConformal IDAnalyticBH<---
Activating thorn ADMBase...Success -> active implementation ADMBase
Activating thorn IDAnalyticBH...Success -> active implementation idanalyticbh
Activating thorn SpaceMask...Success -> active implementation SpaceMask
Activating thorn StaticConformal...Success -> active implementation StaticConformal
Activation requested for
--->SphericalSurface AHFinderDirect<---
Activating thorn AHFinderDirect...Success -> active implementation AHFinderDirect
Activating thorn SphericalSurface...Success -> active implementation SphericalSurface
Activation requested for
--->AHUtil AHTracker<---
Activating thorn AHTracker...Success -> active implementation AHTracker
Activating thorn AHUtil...Success -> active implementation AHUtil
Activation requested for
--->AHUtil HADManalysis HADMevol HADMgauge HADMidata HADMvars HarmonicFD<---
Warning: thorn AHUtil already active
Activating thorn HADManalysis...Success -> active implementation HADManalysis
Activating thorn HADMevol...Success -> active implementation HADMevol
Activating thorn HADMgauge...Success -> active implementation HADMgauge
Activating thorn HADMidata...Success -> active implementation HADMidata
Activating thorn HADMvars...Success -> active implementation HADMvars
Activating thorn HarmonicFD...Success -> active implementation HarmonicFD
Activation requested for
--->AbigelExcision AHUtil Fortran ExcisionMask<---
Warning: thorn AHUtil already active
Activating thorn AbigelExcision...Success -> active implementation AbigelExcisionDist
Activating thorn ExcisionMask...Success -> active implementation ExcisionMask
Activating thorn Fortran...Success -> active implementation Fortran
--------------------------------------------------------------------------------
  if (recover initial data)
    Recover parameters
  endif

  Startup routines
    [CCTK_STARTUP]
    Carpet: Multi-model Startup routine
    Carpet: Startup routine
    AEILocalInterp: register CCTK_InterpLocalUniform() interpolation operators
    IOUtil: Startup routine
    CarpetIOBasic: Startup routine
    CarpetIOHDF5: Startup routine
    CarpetIOScalar: Startup routine
    CarpetInterp: Startup routine
    CarpetReduce: Startup routine
    CartGrid3D: Register GH Extension for GridSymmetry
    CoordBase: Register a GH extension to store the coordinate system handles
    CarpetIOASCII: Startup routine
    LocalInterp: register LocalInterp's interpolation operators
    MoL: Startup banner
    SymBase: Register GH Extension for SymBase

  Startup routines which need an existing GH
    [CCTK_WRAGH]
    ADMBase: Set up GF symmetries
    Boundary: Register boundary conditions that this thorn provides
    AbigelExcision: register symmetry properties of local GFs
    CartGrid3D: Register coordinates for the Cartesian grid
    GROUP SymBase_Wrapper: Wrapper group for SymBase
      GROUP SymmetryRegister: Register your symmetries here
        CartGrid3D: Register symmetry boundaries
      SymBase: Print symmetry boundary face descriptions
    HADMvars: Set symmetries for ADM Harmonic grid functions
    MoL: Set up the MoL bookkeeping index arrays
    MoL: Set the flag so it is ok to register with MoL
    GROUP MoL_Register: The group where physics thorns register variables with MoL
      HADMevol: register our (state vector,RHS) groups with MoL for evolution
    MoL: Report how many of each type of variable there are
   SpaceMask: Set grid symmetries for mask
    SpaceMask: Set grid symmetries for emask (compatibility mode)
```

```
    AHTracker: set CarpetRegrid parameters as requested
  Parameter checking routines
    [CCTK_PARAMCHECK]
    ADMBase: Check consistency of parameters
    AbigelExcision: check parameters
    Boundary: Check dimension of grid variables
    CarpetRegrid: Check Parameters
    CartGrid3D: Check coordinates for CartGrid3D
    HADMevol: check parameters
    IDAnalyticBH: Construct parameters for analytic black hole solutions
    MoL: Basic parameter checking

  Initialisation
    [CCTK_BASEGRID]
    ADMBase: Set the shift_state variable to 0
    CartGrid3D: Set up ranges for spatial 3D Cartesian coordinates
    CartGrid3D: Set up spatial 3D Cartesian coordinates on the GH
    AHTracker: initialize local variables
    AHUtil: reset variables
    GROUP AbigelExcision_InitialGroup: initalize excision algorithm
      AbigelExcision: initalize excision algorithm
      AbigelExcision: set up local grid-layout information
    CarpetIOASCII: Initialisation routine
    CarpetIOBasic: Initialisation routine
    CarpetIOHDF5: Initialisation routine
    CarpetIOScalar: Initialisation routine
    GROUP MaskBase_SetupMask: Set up the weight function
      CarpetReduce: Initialise the weight function
      GROUP SetupMask: Set up the weight function (schedule other routines in here)
        CarpetReduce: Set up the outer boundaries of the weight function
        CarpetReduce: Set up the weight function for the restriction regions
    AHFinderDirect: setup data structures
    AHFinderDirect: save apparent horizon(s) into Cactus variables
    ExcisionMask: initialize mask variables
    HADMvars: weight function for A's
    SpaceMask: Initialise mask to zero
    SpaceMask: Set old style mask to one
    SphericalSurface: Calculate surface coordinate descriptors
    SymBase: Check whether the driver set up the grid consistently
    Time: Initialise Time variables
    Time: Set timestep based on Courant condition (courant_static)
    if (NOT (recover initial data AND recovery_mode is 'strict'))
      [CCTK_INITIAL]
      StaticConformal: Set the conformal_state variable to 0
      GROUP ADMBase_InitialData: Schedule group for calculating ADM initial data
        IDAnalyticBH: Construct initial data for Brill Lindquist black holes
      GROUP ADMBase_InitialGauge: Schedule group for the ADM initial gauge condition
        ADMBase: Set the lapse to 1 at all points
      ExcisionMask: initialize mask variables
      GROUP HADM_group_for_gauge: set gauge functions
        HADMgauge: set gauge functions zero
      HADMidata: Set Aalp, Abetai and Aij
      HADMidata: apply boundary conditions to the HADM initial data
      HADMvars: initialize dissipation weight to its outer value
      MoL: Initialise the step size control
      MoL: Ensure the correct time and timestep are used - initial
      GROUP ADMBase_PostInitial: Schedule group for modifying the ADM initial data, such as e.g. adding n
oise
      [CCTK_POSTINITIAL]
      AHFinderDirect: import the excision mask
      AHFinderDirect: find apparent horizon(s) after this time step
      AHFinderDirect: store apparent horizon(s) into spherical surface(s)
      AHFinderDirect: save apparent horizon(s) into Cactus variables
      AHFinderDirect: set mask(s) based on apparent horizon position(s)
      CarpetIOHDF5: Close all filereader input files
      GROUP HADM_group_for_analysis: Calculating the Constraint Violation
        GROUP HADM_group_for_gauge: set gauge functions
          HADMgauge: set gauge functions zero
        HADManalysis: Calculating the Constraint Violation
        HADManalysis: Calculating the kij from the Aij
```

```
        HADManalysis: apply boundary conditions to analysis quantities
      MoL: A bad routine. Fills all previous timelevels with data copied from the current.
  GROUP MoL_PostStep: Ensure that everything is correct after the initial data have been set up
      GROUP AbigelExcision_ExcisionGroup: distributed excision group
        AbigelExcision: reset own GFn values
        AbigelExcision: set up local grid-layout information
        AbigelExcision: set up mask of points used by excision
        AbigelExcision: fill excised points with dummy values
        AbigelExcision: distributed excision algorithm
      HADMevol: apply boundary conditions to the HADM state vector
      GROUP MoL_OldBdry_Wrap: Wrapper group, do not schedule directly into this group
        MoL: Store and change dt
        GROUP MoL_OldStyleBoundaries: Place old style boundary routines here
        MoL: Reset dt
    [CCTK_POSTSTEP]
    AHFinderDirect: import the excision mask
    AHFinderDirect: find apparent horizon(s) after this time step
    AHFinderDirect: store apparent horizon(s) into spherical surface(s)
    AHFinderDirect: save apparent horizon(s) into Cactus variables
    AHFinderDirect: set mask(s) based on apparent horizon position(s)
    ExcisionMask: check if the mask needs to be initialized
    AbigelExcision: excision on initial slice (active only if cctk_iteration=0)
    GROUP ExcisionMask_SetMask: set horizon mask (post-step)
      GROUP ExcisionMask_OldMask: old mask group
        ExcisionMask: create old copy of emask
      GROUP ExcisionMask_NewMask: new mask group
        ExcisionMask: check if the current algorithm needs to be skipped
        ExcisionMask: reset excision mask state and counter
        ExcisionMask: set up horizon mask for excision
        ExcisionMask: perform various reductions for excision mask
    NaNChecker: Reset the NaNChecker::NaNsFound counter
    NaNChecker: Check for NaNs and count them in NaNChecker::NaNsFound
    NaNChecker: Output NaNChecker::NaNmask and take action according to NaNChecker::action_if_found
    SpaceMask: Ensure that all mask values are legal
    SphericalSurface: Set surface radii
    GROUP SphericalSurface_HasBeenSet: Set the spherical surfaces before this group, and use it afterwa
rds
  endif
  if (recover initial data)
    [CCTK_RECOVER_VARIABLES]
    [CCTK_POST_RECOVER_VARIABLES]
    AHFinderDirect: import horizon data from Cactus variables
    AHFinderDirect: import the excision mask
    AHFinderDirect: find apparent horizon(s) after this time step
    AHFinderDirect: store apparent horizon(s) into spherical surface(s)
    AHFinderDirect: save apparent horizon(s) into Cactus variables
    AHFinderDirect: set mask(s) based on apparent horizon position(s)
    AHTracker: set up some variables after recovery
    ExcisionMask: set up excision mask
    GROUP MaskBase_SetupMask: Set up the weight function
      CarpetReduce: Initialise the weight function
      GROUP SetupMask: Set up the weight function (schedule other routines in here)
        CarpetReduce: Set up the outer boundaries of the weight function
        CarpetReduce: Set up the weight function for the restriction regions
    AbigelExcision: post-recovery excision setup
  endif
  if (checkpoint initial data)
    [CCTK_CPINITIAL]
  endif
  if (analysis)
    [CCTK_ANALYSIS]
    CarpetLib: Print timing statistics if desired
    CarpetLib: Print memory statistics if desired
  endif
  Do periodic output of grid variables

do loop over timesteps
  Rotate timelevels
  iteration = iteration + 1
  t = t+dt
```

```
    [CCTK_PRESTEP]
    HADMvars: weight function for dissipation
    HADMvars: obtain maximum of dissipation weight
    HADMvars: normalize dissipation weight
    [CCTK_EVOL]
    MoL: Initialise the step size control
    while (MoL::MoL_Stepsize_Bad)
 GROUP MoL_Evolution: A single Cactus evolution step using MoL
        GROUP MoL_StartStep: MoL internal setup for the evolution step
          MoL: Set the counter for the ODE method to loop over
          MoL: Ensure the correct time and timestep are used
        GROUP MoL_PreStep: Physics thorns can schedule preloop setup routines in here
        MoL: Ensure the data is in the correct timelevel
        while (MoL::MoL_Intermediate_Step)
          GROUP MoL_Step: The loop over the intermediate steps for the ODE integrator
            MoL: Initialise the RHS functions
            GROUP MoL_CalcRHS: Physics thorns schedule the calculation of the discrete spatial operator i
n here
              GROUP HADM_group_for_gauge: set gauge functions
                HADMgauge: set gauge functions zero
              HADMevol: calculate RHS of the HADM system
            GROUP MoL_PostRHS: Modify RHS functions
              HADMevol: add dissipation to the HADM variables
            GROUP MoL_RHSBoundaries: Any 'final' modifications to the RHS functions (boundaries etc.)
            MoL: Updates calculated with the efficient Runge-Kutta 3 method
            MoL: Alter the counter number
            MoL: If necessary, change the time
            GROUP MoL_PostStep: The group for physics thorns to schedule boundary calls etc.
              GROUP AbigelExcision_ExcisionGroup: distributed excision group
                AbigelExcision: reset own GFn values
                AbigelExcision: set up local grid-layout information
                AbigelExcision: set up mask of points used by excision
                AbigelExcision: fill excised points with dummy values
                AbigelExcision: distributed excision algorithm
              HADMevol: apply boundary conditions to the HADM state vector
              GROUP MoL_OldBdry_Wrap: Wrapper group, do not schedule directly into this group
                MoL: Store and change dt
                GROUP MoL_OldStyleBoundaries: Place old style boundary routines here
                MoL: Reset dt
            MoL: If necessary, change the timestep
        end while
        MoL: Restoring the Save and Restore variables to the original state
        MoL: Control the step size
    end while
    GROUP HADM_group_for_analysis: Calculating the Constraint Violation
      GROUP HADM_group_for_gauge: set gauge functions
        HADMgauge: set gauge functions zero
      HADManalysis: Calculating the Constraint Violation
      HADManalysis: Calculating the kij from the Aij
      HADManalysis: apply boundary conditions to analysis quantities
    [CCTK_POSTSTEP]
    AHFinderDirect: import the excision mask
    AHFinderDirect: find apparent horizon(s) after this time step
    AHFinderDirect: store apparent horizon(s) into spherical surface(s)
    AHFinderDirect: save apparent horizon(s) into Cactus variables
    AHFinderDirect: set mask(s) based on apparent horizon position(s)
    ExcisionMask: check if the mask needs to be initialized
    AbigelExcision: excision on initial slice (active only if cctk_iteration=0)
    GROUP ExcisionMask_SetMask: set horizon mask (post-step)
      GROUP ExcisionMask_OldMask: old mask group
        ExcisionMask: create old copy of emask
      GROUP ExcisionMask_NewMask: new mask group
        ExcisionMask: check if the current algorithm needs to be skipped
        ExcisionMask: reset excision mask state and counter
        ExcisionMask: set up horizon mask for excision
        ExcisionMask: perform various reductions for excision mask
    NaNChecker: Reset the NaNChecker::NaNsFound counter
    NaNChecker: Check for NaNs and count them in NaNChecker::NaNsFound
    NaNChecker: Output NaNChecker::NaNmask and take action according to NaNChecker::action_if_found
    SpaceMask: Ensure that all mask values are legal
```

```
    SphericalSurface: Set surface radii
    GROUP SphericalSurface_HasBeenSet: Set the spherical surfaces before this group, and use it afterward
s
    if (checkpoint)
      [CCTK_CHECKPOINT]
      CarpetIOHDF5: Evolution checkpoint routine
    endif
    if (analysis)
      [CCTK_ANALYSIS]
      CarpetLib: Print timing statistics if desired
      CarpetLib: Print memory statistics if desired
  endif
    Do periodic output of grid variables
  enddo

  Termination routines
    [CCTK_TERMINATE]
    CarpetIOHDF5: Termination checkpoint routine
    MoL: Free the MoL bookkeeping index arrays

  Shutdown routines
    [CCTK_SHUTDOWN]

  Routines run after restricting:
    [CCTK_POSTRESTRICT]
    HADManalysis: apply boundary conditions to analysis quantities
    HADMevol: skip outer boundary algorithm for this scheduling bin
    GROUP MoL_PostStep: Ensure that everything is correct after restriction
      GROUP AbigelExcision_ExcisionGroup: distributed excision group
        AbigelExcision: reset own GFn values
        AbigelExcision: set up local grid-layout information
        AbigelExcision: set up mask of points used by excision
        AbigelExcision: fill excised points with dummy values
        AbigelExcision: distributed excision algorithm
      HADMevol: apply boundary conditions to the HADM state vector
      GROUP MoL_OldBdry_Wrap: Wrapper group, do not schedule directly into this group
        MoL: Store and change dt
        GROUP MoL_OldStyleBoundaries: Place old style boundary routines here
        MoL: Reset dt

  Routines run after changing the grid hierarchy:
    [CCTK_POSTREGRID]
    SpaceMask: Initialise mask to zero
    GROUP AbigelExcision_InitialGroup: re-initialize excision
      AbigelExcision: initalize excision algorithm
      AbigelExcision: set up local grid-layout information
    GROUP MaskBase_SetupMask: Set up the weight function
      CarpetReduce: Initialise the weight function
      GROUP SetupMask: Set up the weight function (schedule other routines in here)
        CarpetReduce: Set up the outer boundaries of the weight function
        CarpetReduce: Set up the weight function for the restriction regions
    CartGrid3D: Set Coordinates after regridding
    ExcisionMask: initialize mask variables
    SpaceMask: Set mask to one
    HADManalysis: apply boundary conditions to analysis quantities
    HADMevol: skip outer boundary algorithm for this scheduling bin
    HADMvars: weight function for A's
    HADMvars: initialize dissipation weight to its outer value
    GROUP MoL_PostStep: Ensure that everything is correct after regridding
      GROUP AbigelExcision_ExcisionGroup: distributed excision group
        AbigelExcision: reset own GFn values
        AbigelExcision: set up local grid-layout information
        AbigelExcision: set up mask of points used by excision
        AbigelExcision: fill excised points with dummy values
        AbigelExcision: distributed excision algorithm
      HADMevol: apply boundary conditions to the HADM state vector
      GROUP MoL_OldBdry_Wrap: Wrapper group, do not schedule directly into this group
        MoL: Store and change dt
        GROUP MoL_OldStyleBoundaries: Place old style boundary routines here
        MoL: Reset dt
```

```
    AHFinderDirect: set mask(s) based on apparent horizon position(s)
    GROUP ExcisionMask_SetMask: set horizon mask (post-regrid)
      GROUP ExcisionMask_NewMask: new mask group
        ExcisionMask: check if the current algorithm needs to be skipped
        ExcisionMask: reset excision mask state and counter
        ExcisionMask: set up horizon mask for excision
        ExcisionMask: perform various reductions for excision mask
      GROUP ExcisionMask_OldMask: old mask group
        ExcisionMask: create old copy of emask
--------------------------------------------------------------------------------
```

Bibliography

[1] Thomas W. Baumgarte and Stuart L. Shapiro. Numerical relativity and compact binaries.

[2] Bertil Gustafsson, Heinz-Otto Kreiss, and Joseph Oliger. *Time Dependent Problems And Difference Methods*. Pure And Applied Mathematics: A Wiley-Interscience Series of Texts, Monographs, and Tracts, 1995.

[3] C. Bona and C. Palenzuela-Luque. *Elements of Numerical Relativity*. Springer, 2005.

[4] John P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, Inc., second (revised) edition, 2001.

[5] P. Diener and F. S. Guzmán. The wave equation in 1+1, 2003.

[6] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Texts in Applied Mathematics 12. Springer, 2002.

[7] Tom Goodale, Gabrielle Allen, Gerd Lanfermann, Joan Massó, Thomas Radke, Edward Seidel, and John Shalf. The cactus framework and toolkit: Design and applications. *Vector and Parallel Processing - VECPAR '2002, 5th International Conference, Springer*, 2003.

[8] Gabrielle Allen, Thomas Dramlitsch, Ian Foster, Nicholas T. Karonis, Matei Ripeanu Edward Seidel, and Brian Toonen. Supporting efficient execution in heterogeneous distributed computing environments with cactus and globus. *Proceedings of Supercomputing*, 2001.

[9] Cactus computational toolkit. www.cactuscode.org.

[10] Erik Schnetter, Scott H. Hawley, and Ian Hawke. Evolutions in 3d numerical relativity using fixed mesh refinement. *Classical and quantum gravity*, 21:1465–1488, 2004.

[11] Carpet mesh-refinement driver. www.carpetcode.org.

[12] Jonathan Thornburg. Black hole excision with multiple grid patches. *gr-qc/0404059 v4*, 2004.

[13] Jonathan Thornburg. A fast apperent horizon finder for three-dimensional cartesian grids in numerical relativity. *Classical and quantum gravity*, 21:743–766, 2004.

[14] Jonathan Thornburg. Finding apparent horizons in numerical relativity. *Phys. Rev. D*, 54(8):4899–4918, October 15 1996.

[15] Jonathan Thornburg. A fast apparent-horizon finder for 3-dimensional Cartesian grids in numerical relativity. *Class. Quantum Grav.*, 21(2):743–766, 21 January 2004.

[16] Message passing interface (mpi) standard. www-unix.mcs.anl.gov/mpi/index.htm.

[17] Griebel, Knapek, Zumbusch, and Caglar. *Numerische Simulation in der Moleküldynamik*. Springer-Verlag, 2004.

[18] AEI supercomputer equipment. http://supercomputers.aei.mpg.de/peyote

[19] AEI supercomputer equipment. http://supercomputers.aei.mpg.de/belladonna

[20] AEI supercomputer equipment. http://supercomputers.aei.mpg.de/damiana

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

[22] T. de Donder. *La Gravifique Einsteinienne*. Gauthiers-Villars, Paris, 1921.

[23] V. Fock. *The Theory of Space, Time, and Gravitation*. Pergamon, New York, 1959.

[24] Y. Choquet-Bruhat and T. Ruggeri. Hyperbolicity of the 3+1 system of Einstein equations. *Comm. Math. Phys*, 89:269–275, 1983.

[25] B. Szilágyi, B. Schmidt, and Jeffrey Winicour. Boundary conditions in linearized harmonic gravity. *Phys. Rev. D*, 65:064015, 2002.

[26] B. Szilágyi and Jeffrey Winicour. Well-posed initial-boundary evolution in general relativity. *Phys. Rev. D*, 68:041501, 2003.

[27] Maria C. Babiuc, Béla Szilágyi, and J.Winicour. Harmonic initial-boundary evolution in general relativity. *Phys. Rev. D*, 73:064017, 2006.

[28] Maria C. Babiuc and J.Winicour. Constraint-preserving sommerfeld conditions for the harmonic einstein equations. gr-qc/0612051, 2006.

[29] Maria C. Babiuc, Béla Szilágyi, and Jeffrey Winicour. Testing numerical relativity with the shifted gauge wave. *Class. Quantum Grav.*, 23:S319–S342, 2006.

[30] Frans Pretorius. Simulation of binary black hole spacetimes with a harmonic evolution scheme. *Class. Quantum Grav.*, 23:S529–S552, 2006.

[31] Erik Schnetter, Scott H. Hawley, and Ian Hawke. Evolutions in 3D numerical relativity using fixed mesh refinement. *Class. Quantum Grav.*, 21(6):1465–1488, 21 March 2004.

[32] Cactus Computational Toolkit. `http://www.cactuscode.org`.

[33] Helmut Friedrich. Hyperbolic reductions for Einstein's equations. *Class. Quantum Grav.*, 13:1451–1469, 1996.

[34] Helmut Friedrich and Alan Rendall. The Cauchy Problem for the Einstein Equations. *Lect.Notes Phys.*, 540:127-224, 2000.

[35] Carsten Gundlach, Jose M. Martin-Garcia, G. Calabrese, and I. Hinder. Constraint damping in the Z4 formulation and harmonic gauge. *Class. Quantum Grav.*, 22:3767–3774, 2005.

[36] Gioel Calabrese. Finite differencing second order systems describing black hole spacetimes. *Phys. Rev. D*, 71:027501, 2005.

[37] B. Szilágyi, H-O. Kreiss, and J. Winicour. Modeling the black hole excision problem. *Phys. Rev. D*, 71:104035, 2005.

[38] B. Szilágyi, Denis Pollney, Luciano Rezzolla, Jonathan Thornburg and Jeffrey Winicour. An explicit harmonic code for black-hole evolution using excision. *Class. Quantum Grav.*, 24:275–293, 2007.

[39] Gioel Calabrese and Carsten Gundlach. Discrete boundary treatment for the shifted wave equation. gr-qc/0509119, 2005.

[40] Mohammad Motamed, M. C. Babiuc, B. Szilágyi, H-O. Kreiss, and J.Winicour. Finite difference schemes for second order systems describing black holes. *Phys. Rev. D*, 73:124008, 2006.

[41] Helmut Friedrich and Gabriel Nagy. The initial boundary value problem for Einstein's vacuum field equations. *Commun. Math. Phys.*, 201:619–655, 1999.

[42] Heinz-Otto Kreiss and Jeffrey Winicour. Problems which are well-posed in a generalized sense with applications to the Einstein equations. *Class. Quantum Grav.*, 23:S405–S420, 2006.

[43] Ken Mattsson and Jan Nordström. Summation by parts operators for finite difference approximations of second derivatives. *Journal of Computational Physics*, 199(2):503–540, 20 September 2004.

[44] Heinz-Otto Kreiss and N. Anders Petersson. A second order accurate embedded boundary method for the wave equation with Dirichlet data. *SIAM J. Sci. Comput.*, page 31, 2004.

[45] Heinz Otto Kreiss and J. Lorenz. *Initial-Boundary Value Problems and the Navier-Stokes Equations*. Academic Press, New York, 1989.

[46] Miguel Alcubierre and Bernard Schutz. Time–symmetric ADI and causal reconnection: Stable numerical techniques for hyperbolic systems on moving grids. *J. Comput. Phys.*, 112:44, 1994.