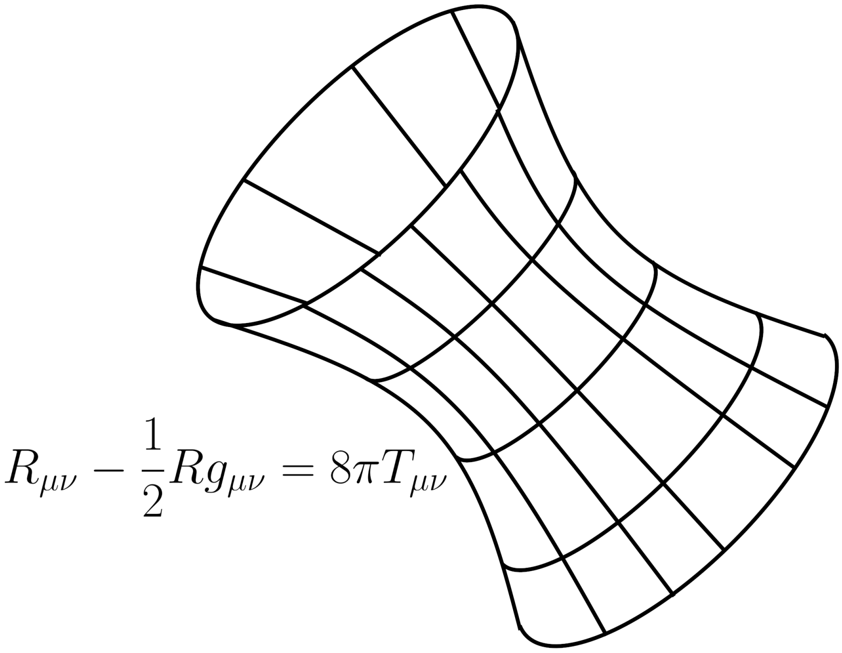
UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE



$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = 8\pi T_{\mu\nu}$$

Master thesis

Numerical Relativity in Axisymmetry

Jorge Expósito Patiño

Supervised by Thomas Baumgarte, Vitor Cardoso and Troels Haugbølle

Submitted on 22-05-2023

Contents

Abstract

Numerical relativity, the branch of general relativity that uses simulations to solve the Einstein equations, has proven an invaluable tool in studying gravity, from gravitational wave astronomy to foundational questions such as cosmic censorship and space-time stability studies. Most current numerical relativity codes use Cartesian or spherical coordinates, which are not ideally suited to many systems that present structure around an axis, examples of which are black hole head-on collisions and critical collapse around an axis. In this thesis, I present the new code I have created to do numerical relativity simulations with a free choice of spatial coordinates, and present results and simulations in both cylindrical and spherical coordinates.

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisors, Thomas Baumgarte, Vitor Cardoso and Troels Haugbølle, for their supervision and guidance during my thesis. I would like to thank Thomas Baumgarte specially for meeting with me every week across the ocean to guide me with his immense knowledge and experience. I also want to acknowledge Vitor Cardoso and Troels Haugbølle for their support and expertise, which greatly contributed to the refinement of my research. It has been a pleasure to work with them, and I am grateful for their time, mentorship, and expertise which have guided me through this thesis and will carry on throughout my future research.

Chapter 1

Introduction: Numerical Relativity

Numerical relativity, the branch of general relativity that uses simulations to solve the Einstein equations, has proven an invaluable tool in studying gravity. Its role, first in the discovery of gravitational waves [1], and nowadays in the crucial field of gravitational wave astronomy [2, 3] cannot be overstated, as it produces the results to which compare observations in order to understand the sources of the gravitational radiation. But its importance doesn't end there, as it is also a main pillar on the study of foundational issues in gravity, some examples being cosmic censorship [4, 5], non-linear stability of space-times [6, 7], the Hoop conjecture [8, 9, 10] and even cosmological evolution [11, 12, 13]. The reason why numerical relativity has such prominence in such a vast array of topics is because, when we are in the strong field regime and non-linearities are important, it is the only tool able to give us results to the precision needed, provided the scales in the problem are amenable [14, 15, 16, 17].

My objective in this thesis is to create a new numerical relativity code, that I will focus on systems with axisymmetry. There exists many numerical relativity codes already [18, 19, 20], so it is important to address the question of why a new code would be worth the effort. The answer is twofold.

First, there is the pedagogical side to the endeavor. Creating a new code is quite a learning experience, forcing the author to engage with all the intricacies of a numerical relativity simulation and understand the problems in depth. Moreover, in this thesis I intend to present a guide of some of the steps necessary, in order to direct a reader interested in developing a new code, or understanding how codes are developed.

Secondly, and perhaps more relevant for my future research, I believe the currently existing codes are not ideal for axisymmetric systems. Those systems are abundant in numerical relativity, both motivated from an astrophysical viewpoint and from a purely relativistic perspective, examples of which could be black hole head on collision or critical collapse around an axis. The system of coordinates that is better suited to describe such systems is cylindrical coordinates, but most existing codes are written either in Cartesian or spherical polar coordinates and those systems require much higher precisions in order to capture axis structure than cylindrical symmetry. The converse is also true: if I want to study a system with spherical or planar structure in axisymmetry, these other systems will be better suited. That is why I believe there is a need to create a code that works with any coordinate system of our choice, although this is not a new idea (see [20]) either.

Importantly, the step of moving to a different system of coordinates is not the last word in Numerical relativity codes either, but that supports my reasoning: in order to keep expanding the systems that we might be able to study, we have to keep the code-making culture alive.

Many of the numerical difficulties in simulating a hyperbolic system in curvilinear coordinates can be studied by looking at the simplest such system: the scalar wave equation. That will motivate the structure of my thesis, that starts with an in-depth treatment of the scalar wave equation in chapter 2, highlighting how many of its numerical characteristics and challenges will translate into numerical relativity, and how to simulate the system correctly. From that, I will move on to the full numerical relativity formulation that I use in chapter 3, including a basic discussion of the system and which extra problems arise with it. In chapter 4 I will present the results of several simulations, that test the correctness of the code, and finally I will end up with some comments about open questions and future research.

Chapter 2

The scalar wave equation

2.1 Analytical properties of the solution

In this section:
Singularities in the equation— Compact solutions — Superposition of ingoing and outgoing waves — Boundary conditions — Boundary error — Analytical solution

This section is a bit of groundwork for the next sections, with two main purposes. The first is to get some theoretical understanding of the solutions of the scalar wave equation: how the solution will look like, which terms can give us complications at the computational level and what are natural boundary conditions. The second is to come up with a specific analytical solution that later we can test our code against to see if it reproduces it successfully.

The equation we want to solve is the scalar wave equation, given as follows:

$$\frac{1}{c^2}\frac{\partial^2 \phi}{\partial t^2} = \nabla^2 \phi. \tag{2.1}$$

This is a good proxy for Einstein's equations for a couple of reasons. First of all, as we will see later, they can both be written in the same form:

$$\frac{\partial u}{\partial t} = \mathcal{L}(u), \tag{2.2}$$

where $\mathcal{L}$ is a differential operator. Both our formulation of Einstein's equations and the scalar wave equation will have this form with different operators, so the code and understanding of one will translate to the other. Second (although not independent) both systems admit wave-like solutions, and this will be useful in particular when getting and understanding of the necessary boundary conditions.

In particular in our code, this equation will be solved in cylindrical coordinates (ρ, φ, z) and axisymmetry $(\partial_\varphi = 0)$, in which the equation looks:

$$\frac{\partial^2 \phi}{\partial \rho^2} + \frac{1}{\rho}\frac{\partial \phi}{\partial \rho} + \frac{\partial^2 \phi}{\partial z^2} - \frac{1}{c^2}\frac{\partial^2 \phi}{\partial t^2} = 0. \tag{2.3}$$

The first thing we notice is that there is a singular term at the axis (the second term in the equation, that scales like $1/\rho$). Numerically this is going to be one of our sources of complication, since although infinities will cancel analytically, they need not do it in our simulation. More about this in the next section.

Solutions to (2.3) by separation of variables look like plane waves that decrease in amplitude far from the axis (see appendix A). Their amplitude far from the axis is given by:

$$|\phi_k|^2 = \frac{A_k}{k_\rho \rho}. \tag{2.4}$$

As we can see, these waves don't decrease in amplitude in the z direction, so solutions have infinite extent. If we think about systems that we might want to study that generate waves, we have the sources localized somewhere in space (an oscillating dipole, a black hole merger...) and the rest is empty space. Therefore, we will not get waves of infinite extent, and these solutions are not well adapted to study our problem analytically. It is important to note, however, that in a simulation the fields are not represented in terms of a functional basis, so the fact that the basis of solutions in cylindrical coordinates doesn't have the behavior we want will not matter in a simulation, except for the choice of initial data.

In order to get waves of finite extent, let's look at the equation in spherical coordinates (r, θ, φ). In that system, it is written like follows:

$$\frac{1}{c^2}\frac{\partial^2 \phi}{\partial t^2} = \frac{1}{r}\frac{\partial^2}{\partial r^2}(r\phi) + \frac{1}{r^2}\frac{1}{\sin^2\theta}\frac{\partial}{\partial \theta}\left(\sin\theta \frac{\partial \phi}{\partial \theta}\right). \tag{2.5}$$

We can do separation of variables, but don't separate totally, in order to keep the information about the time dependence without having to do a superposition of Fourier modes. The decomposition I will consider is the following:

$$\phi_{lm} = \psi_l(r,t) Y_{lm}(\theta, \varphi). \tag{2.6}$$

The angular part is such that I can do separation of variables, but if I went through with all the calculations, I would find that these are the well-known spherical harmonics. Therefore, I can use its defining property in the equation below, and I get:

$$\frac{1}{c^2}\frac{\partial^2}{\partial t^2}(r\psi) - \frac{\partial^2}{\partial r^2}(r\psi) = -\frac{l(l+1)}{r^2}r\psi. \tag{2.7}$$

Let's look at the simplest case, with $l = 0$. This case is spherically symmetric. Then the solution is really easy to find, as a superposition of ingoing and outgoing waves:

$$\psi = \frac{f(r-ct)+g(r+ct)}{r}. \tag{2.8}$$

This solution is important, because, as we can see in equation (2.7) when $r \to \infty$, the term in the right-hand side approaches zero and all modes look like this very far away. This, therefore, can serve as our boundary condition. If we were to be thinking in the cylindrical solutions basis, what that would mean, is that we need a superposition of solutions such that very far away the result doesn't go to infinity, but decay.

Therefore, a reasonable boundary condition for our solutions is to enforce that they look like (2.7) very far away. But if we think about this in terms of characteristics [21], the incoming wave is "bringing information" from very far into our domain. We might say that nothing happens outside our domain, and therefore we don't want any information coming in. That leads us to impose the following boundary condition:

$$\phi \to \frac{f(r-t)}{r} \text{ when } r \to \infty. \tag{2.9}$$

This seems to indicate that to indicate that, referring to causality, there should only be outgoing waves in all our domain. But that is not true for a physical system that is not localized at a point, and it is easy to see by the fact that in that case we would have an irregularity at the origin. A general solution has an ingoing component, and suppressing it will lead to an error that we can study. Let's take the following solution:

$$\phi = \frac{f(r-t)+g(r+t)}{r}. \tag{2.10}$$

If we force it to be totally outgoing at the boundary, we are adding a negative ingoing wave at the boundary

$$\phi_{num} = \phi - \frac{g(r+t)}{r}. \tag{2.11}$$

If g is really small at the boundary anyway, the error will not be big. But it will propagate inwards and increase with r so it has the potential to be a problem. It is important to keep this in mind in simulations.

Now that we have understood the properties of the solutions to the equation, let's look at a particular solution that we can use to test our simulation with. The simplest realistic solution of the ones I have looked at corresponds to the $l = 0$ (spherically symmetric) mode. As we saw earlier, the solution for that case is a superposition of incoming and outgoing waves. In order to be a valid solution in our simulation, it has to be regular at the origin. Therefore the numerator has to vanish at the origin for all times

$$f(-ct) + g(ct) = 0 \Rightarrow g(x) = -f(-x) \quad \forall x. \tag{2.12}$$

With that constraint, we can write our general wave that is regular at the origin as

$$\phi = \frac{f(r-t) - f(-r-t)}{r}. \tag{2.13}$$

Note how we need an ingoing wave in order to have regularity. In order to determine it from initial data, we need both the values of the function and its derivative at $t = 0$. In our case, let's look at the specific case of $\partial_t\phi|_{t=0} = 0$. Let's define the following new variables for simplicity:

$$\sigma u = r - t; \quad \sigma v = r + t. \tag{2.14}$$

For the incoming and outgoing waves that means

$$\begin{aligned} \left.\frac{\partial \phi}{\partial t}\right|_{t=0} = 0 \Rightarrow \left[-f'(\sigma u) + f'(-\sigma v)\right]\big|_{t=0} = -f'(x) + f'(-x) = 0 \Rightarrow \\ \Rightarrow f(x) + f(-x) = C = 2f(0), \end{aligned} \tag{2.15}$$

where x stands for any argument to the function, that has a single variable. The general solution of the type above is

$$\phi(r,t) = \frac{f(\sigma u) + f(\sigma v) - 2f(0)}{r}. \tag{2.16}$$

Now let's impose the initial conditions on the values of the function:

$$\phi(r,0) = h(r) = \frac{f(r) + f(r) - 2f(0)}{r} \Rightarrow 2f(x) - 2f(0) = xh(x)$$

$$\Rightarrow \phi(r,t) = \frac{1}{2}\frac{uh(\sigma u) + vh(\sigma v)}{(r/\sigma)}. \tag{2.17}$$

The above expression is a solution for any choice of initial conditions. In particular, for the tests afterwards I will use a Gaussian for initial data

$$h(r) = e^{-r^2} = e^{-\rho^2 - z^2}. \tag{2.18}$$

2.2 Implementation of the equation

In this section:
General hyperbolic form — evolution algorithm MOL-RK4 — computational grid — back-trace of characteristics — boundary conditions at the origin — Taylor theorem — convergence and error — stability analysis and CFL condition

The scalar wave equation is second order, but if we introduce a new variable we can reduce it to a first order system:

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= \psi \\ \frac{\partial \psi}{\partial t} &= c^2 \nabla^2 \phi. \end{aligned} \tag{2.19}$$

This is of the general form

$$\frac{\partial u}{\partial t} = \mathcal{L}(u), \tag{2.20}$$

with $\mathcal{L}$ an operator that might include spatial derivatives but not ones with respect to time. The equations for numerical relativity are also of this form, and that is the reason why a careful study of the scalar wave is so useful in order to understand and develop the methods for the full non-linear system. A system of this type can be simulated in many different ways, but we will choose a Method Of Lines (MOL) with a Runge-Kutta of 4th order (RK4) to do the time evolution [22]. This method consists on discretizing space first, taken the vector function u to a vector of discrete values $\hat{u}$, which the right-hand side will mix, but there are no longer space derivatives. Then we have an ODE

$$\frac{d\hat{u}}{dt} = L(\hat{u}), \tag{2.21}$$

and as such, we can use any time evolution method that we would use for an ODE. In my case I will choose fourth order Runge-Kutta, in which the system is evolved as follows:

$$\begin{aligned} k_1 &= L(\hat{u}^n) \\ k_2 &= L(\hat{u}^n + k_1 \Delta t/2) \\ k_3 &= L(\hat{u}^n + k_2 \Delta t/2) \\ k_4 &= L(\hat{u}^n + k_3 \Delta t/2) \\ \hat{u}^{n+1} &= \hat{u}^n + \frac{\Delta t}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right). \end{aligned} \tag{2.22}$$

From now on I will make no distinction between the discretized vector of values $\hat{u}$ and the true solution u, and the same for the right-hand side operator, as it will be clear which one is being discussed from the context.

Regarding the space discretization, I will sample the functions on a 2D grid (since we have axisymmetry we only have 2 spatial dimensions) and I will take fourth order centered derivatives. The fourth order partial derivatives have the following formulas, for the first derivative

$$\frac{\partial \phi}{\partial \rho} = \frac{\phi_{i-2,j} - 8\phi_{i-1,j} + 8\phi_{i+1,j} - \phi_{i+2,j}}{12\Delta\rho} + \mathcal{O}(\Delta\rho^4), \tag{2.23}$$

and for the second

$$\frac{\partial^2 \phi}{\partial z^2} = \frac{-\phi_{i,j-2} + 16\phi_{i,j-1} - 30\phi_{i,j} + 16\phi_{i,j+1} - \phi_{i,j+2}}{12\Delta z^2} + \mathcal{O}(\Delta z^4). \tag{2.24}$$

We can notice that each of them need two neighboring points in order to be calculated, therefore I will create my grid with two ghost nodes on each side. An important thing to consider when creating the grid are the coordinate singularities (like $1/\rho$). If we have any grid point with $\rho = 0$, in our floating points we are going to get `inf`, and even though we might divide it by another `inf` later, in such a way that analytically it should be well-behaved, with floating points we have lost all information, and the result will be `Nan` (see [23]). Of course, this is not the behavior we want. The simplest way to solve this problem is to not have any points at $\rho = 0$, by using a cell-centered grid. That way points in ρ will be indexed as

$$\rho_i = (i + 1/2)\Delta\rho. \tag{2.25}$$

And the minimum value for $\rho = \Delta\rho/2$. This will not give any floating point errors in division until $\Delta\rho \sim \epsilon \sim 10^{-16}$, but of course in that regime floating point errors will already start dominating in all our domain, so we will never go to such a high precision. In this way, part of the problem of regularity is solved, but as we will see, there is more to this.

Now that we have studied the requirements of our grid we know that it has to be a 2D cell centered grid with two grid nodes on each side. We can find a plot of the grid in Fig. 2.1.

We need the ghost nodes in order to take derivatives close to the boundary, but which values should we have there? That is going to be imposed by boundary conditions. In our case the boundaries are of two distinct types: the one with $\rho = 0$ is the axis of symmetry, therefore to the left of it our domain is supposed to continue. The rest of the boundaries are simply surfaces that we choose arbitrarily in space to delimit our computational domain, since we can't cover all of space with our method. Therefore the boundary conditions there will be approximate.

For the first type, the boundary condition is that whatever is on the other side of the axis is the same as what is on this side, by symmetry. Mathematically, this means that the derivative at the axis is zero

$$\partial_n \phi|_{\Gamma_{int}} = 0. \tag{2.26}$$

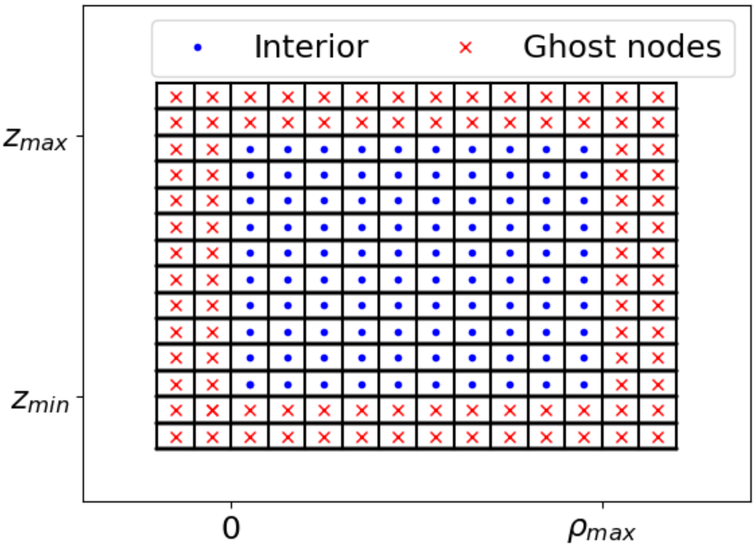


Figure 2.1: Layout of our computational grid.

And for our grid nodes, it is direct to see that the ghost nodes should have the same values as the corresponding point on the domain

$$\phi_{-1,j} = \phi_{0,j}; \quad \phi_{-2,j} = \phi_{1,j}. \tag{2.27}$$

Regarding the outer boundaries, we will impose the outgoing wave boundary condition discussed in the previous section. The way to implement it in code will be the following. If the function has this form

$$\phi(r,t) = \frac{f(r-t)}{r}, \tag{2.28}$$

then in the next time step it will be

$$\phi(r,t+\Delta t) = \frac{f(r-t-\Delta t)}{r} = \frac{r-\Delta t}{r}\frac{f(r-\Delta t-t)}{r-\Delta t} = \frac{r-\Delta t}{r}\phi(r-\Delta t,t). \tag{2.29}$$

The interpretation is immediate: if we want to know what value the wave will have at a point later, we just back-trace the characteristic (where the wave is coming from) and get the value at the corresponding point, reduced by the correct geometric factor. This idea is plotted in Fig. 2.2. Another point is worth noting, that the reference value along the characteristic will almost surely not be a grid point. Therefore, we will have to do bi-linear interpolation with the four neighboring points, which will increase the error of the boundaries even more. But we know that the error created in the

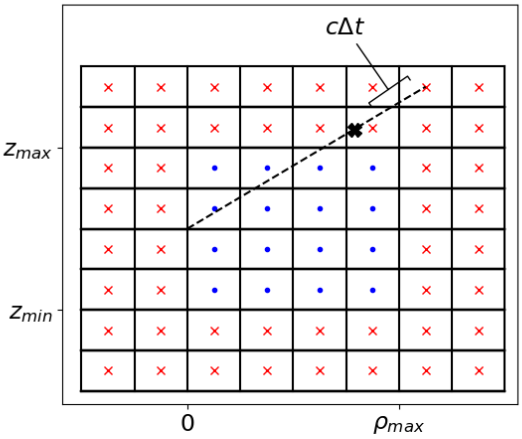


Figure 2.2: The values of the ghost nodes are calculated by tracing the characteristics back to the origin in time and interpolating the value of the function at the previous point, in order to impose outgoing wave boundary conditions.

boundaries satisfies the wave equation in the domain, therefore it will propagate inwards, and we can trust the results in the parts of the domain that have not been in causal contact[1] with the boundary yet.

Now that the scheme is designed and presented, the question is, how big is the error on our simulations? Should we expect the numerical solution to approach the analytical solution? How will the error behave with increasing resolution? The first thing to address is that our discretization is based on the fact that we can approximate derivatives by finite differences, and that is itself based on Taylor's theorem [24]. Therefore, if the conditions for this theorem are not fulfilled, our formulas will not work, and the error will in general be bigger than we expect. This might seem an overly rigorous tangent, but it does have a very important consequence for our simulations.

Taylor's theorem says that for a function n times differentiable at a point c, the function around that point can be expressed as

$$f(x) = \sum_{k=0}^{n-1} \frac{f^{(k)}(c)}{k!}(x-c)^k + \frac{f^{(n)}(x_1)}{n!}(x-c)^n \text{ with } x_1 \text{between } x \text{ and } c. \tag{2.30}$$

Therefore, in order to get an approximation of order n around a point, the function has to be n times differentiable. In terms of our finite differences therefore, we need the functions to be C^5 for the first derivative, and C^6 for

---

[1] Meaning that a wave from the boundary has had time to reach them

the second derivatives[2].

The specific orders will be important later, but for now it is enough to know that we cannot take derivatives across points in which the fields are irregular. This discards the completely outgoing functions as solutions to evolve, for example, since they are not regular in the origin, and therefore we would have unknown error 2 grid points from the origin[3]. We will find a case like this when dealing with black holes. This is also very important, as we will see, because in a curvilinear coordinate basis, some tensor components are not regular, and we cannot differentiate them computationally (even if analytically there is no problem) so we will have to find an alternative procedure.

Assuming that all of our functions will be at least C^6, Taylor's theorem tells us exactly how our error will scale with resolution. Therefore a good test of the implementation is to check that for an arbitrary function, the error in its derivatives scales as expected. Tests like this will be done for all differential methods in the code, and some be presented in the next section. Let's study also how the error looks at a later time step, in which the function to differentiate also has some error from the previous steps. At every step, the error in the calculation of the right-hand side operator is fourth order. The error of the RK4 method is also fourth order, so the solution that we have in each time step is already fourth order. Therefore the total error on the calculation is

$$L(\hat{u}) = \mathcal{L}(\hat{u}) + \alpha N^{-4} = \mathcal{L}(u + \beta N^{-4}) + \alpha N^{-2} = \mathcal{L}(u) + \left[\mathcal{L}(\beta) + \alpha\right] N^{-4}. \tag{2.31}$$

Where α and β are arbitrary functions. The overall error is fourth order, but notice how this relies on the operator being linear. I believe that if $\mathcal{L}(u, \partial_i u)$ is a sufficiently differentiable function of its arguments, meaning it can be expressed as a Taylor series in them, then the leading order of the error in $L(\hat{u})$ will still be the same as the linear case. This is because, if we have non-linear terms of the form u^2 or $u\partial_i u$, the error will be like

$$(u + \beta N^{-4})(u + \beta N^{-4}) = u^2 + 2u\beta N^{-4} + \beta^2 N^{-8}. \tag{2.32}$$

And the leading term is still fourth order. Therefore we can test the evolution by calculating the total error (with the L2 norm for example) at each time step and seeing if it scales like we expect it to. This is the other main test that I will perform.

---

[2] Note that in those formulas we divide by Δx and Δx^2 respectively. Therefore, although the error is fourth order, it comes from the fifth and sixth order remainders.

[3] Since the fourth order finite difference includes points that are two grid spacings away, the approximation at one point is only ensured to converge if the function is sufficiently differentiable in the region between those grid points.

Finally, we would like to know if our scheme is stable. Meaning, will the error increase too fast and completely cover the solution during evolution? This is in principle hard to quantify, and the CFL[4] condition presents a good rule of thumb. If we want to be a bit more precise, we can perform a Von Neumann stability analysis, that consists of studying how the evolution affects the individual modes that make up a solution. This is of course only valid for linear equations, since for non-linear ones, we can't find a basis of the space of solutions since it is not a linear space at all. If we find that some modes are increased without bound, then we know our method will be unstable, but such an analysis gives a necessary but not sufficient condition for stability. In any way, it is useful to get some understanding of the system, and giving us some conditions for our discretization. In the appendix B we can find the stability analysis for our method, and we get the following condition:

$$\frac{c\Delta t}{\Delta_2} \lesssim 1.2; \quad \Delta_2 = \left(\frac{1}{\Delta z^2} + \frac{1}{\Delta \rho^2}\right)^{-1/2}. \tag{2.33}$$

This is a type of CFL condition. The size of the physical domain of dependence $c\Delta t$ has to be smaller than the physical domain of dependence (related to the grid size Δ_2) in order to not lose information in each time step. It is also common to quantify the grid size with the following quantities

$$\Delta_1 = \left(\frac{1}{\Delta z} + \frac{1}{\Delta \rho}\right)^{-1}; \quad \Delta_{min} = \min(\Delta\rho, \Delta z). \tag{2.34}$$

These other measures have the property that

$$\Delta_{min} > \Delta_2 > \Delta_1; \quad \forall \Delta\rho \neq 0, \Delta z \neq 0. \tag{2.35}$$

But also more importantly, we can find a factor such that

$$\Delta_{min}/\alpha < \Delta_2; \quad \Delta_2/\alpha < \Delta_1; \quad \text{with } \alpha > \sqrt{2}. \tag{2.36}$$

That means that if we write the stability condition as

$$\frac{\Delta t}{\Delta} = C. \tag{2.37}$$

with C the Courant factor, the condition can be expressed with any of the measures as long as we choose a small enough Courant factor. In my simulation I will use Δ_1 the most restrictive of the three measures mentioned, and $C = 0.4$, well inside the stability region, that for Δ_1 is $C \lesssim 1.7$.

Now that we have all the computational elements and expectations for the error, let's test the implementation.

---

[4] Courant–Friedrichs–Lewy

2.3 Numerical tests

In this section:
Static tests — snapshots — convergence plot — boundary error

I will start discussing the static tests. These are tests in which we don't use the evolution scheme, but rather we focus only in parts of the right-hand side operator being correctly calculated. The finite difference approximations have an associated order of convergence, meaning the order of the leading order correction in the grid spacing. For my case, this convergence will be fourth order, as presented in formulas (2.23) and (2.24). The idea behind the static tests is the following: I can take a specific function (only restriction is it has to be C^6), compute its derivative in code and compare with the analytic results. I repeat this process for many different grid spacings, and I can quantify how the error varies with resolution. If the method is not correctly implemented, the convergence will be different from theoretically expected, therefore it is a good test for the implementation. The way that I quantify the error is by using the L2 norm of the difference with the analytic solution

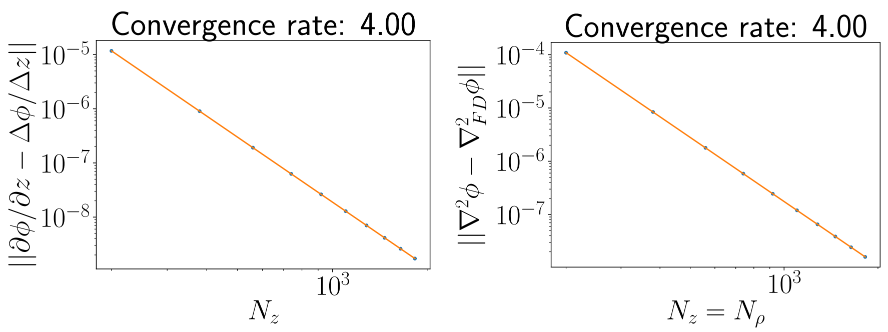
$$(\text{error}) = \sqrt{\sum_{i,j} \left(\hat{\phi}_{ij} - \phi_{ij}\right)^2}. \tag{2.38}$$

This is a common choice, but any norm will be good for studying convergence. Let's look at the results of some specific tests. In Fig. 2.3 we find two tests of this type. We present the results as a linear fit in a log-log plot, such that the expected power law will show as a straight line with the convergence factor as its slope. In general, if the resolution is not high enough, we don't know the specific shape of the error, and we could find something different than the line. But if we are in the regime in which the leading error term is much higher than the others, then it will look like

$$(\text{error}) = \alpha N^{-4}. \tag{2.39}$$

Where N is the number of grid points in the relevant direction, or in both, depending on the method to be tested.

Now for the tests and results of the simulation. In Fig. 2.6 we can see some snapshots of the evolution of the scalar wave. The first test is already passed: we don't see any instabilities of the domain. Once the wave reaches the boundary, we see that we get reflection, which is not what is to be expected from a wave in vacuum. This is a result of the boundary conditions being only approximate, and will be good when they are far away. In this case they are close to study the whole behavior. In order to be able to trust our implementation, we need to quantify the error.



(a) Convergence tests for the first derivative in z.

(b) Convergence test for the Laplacian.

Figure 2.3: Convergence tests for derivatives of a Gaussian function. Notice that both $\Delta\rho$ and Δz are decreased for the Laplacian, since the error scales with both of them.

In order to quantify the error, I will compare with the analytical solution found in section 2.1. The instantaneous error (defined as in formula (2.38) for each time step) will be plotted against time. The simulation will be repeated for several resolutions, and all of those functions will be added to the plot. The result is Fig. 2.4.

In that figure we can see that we increase the resolution, the error looks the same, just smaller. As discussed in the previous section, if our method is well implemented, we expect to it to scale as N^{-4}. Therefore, to test for this we re-scale all the errors times N^4, and if they coincide we have correct convergence. In Fig. 2.5 we can see that plot, and it presents correct fourth order convergence.

This is before the wave reaches the boundary. Once that happens, boundary error becomes important, as we can see in the snapshots of the evolution. Our numerical solution will stop showing correct convergence, since our model (with outgoing waves at the boundary) and our analytical solution don't agree anymore. If the analytical solution is ϕ, and the correct solution to our model (with reflections) is $\tilde{\phi}$, the error that we compute does not approach zero exactly when $\Delta\tilde{\phi} = \phi - \tilde{\phi}$ starts being significant. The error will instead approach $\left\|\Delta\tilde{\phi}\right\|$, and that we can see in Fig. 2.7. This modelling error doesn't scale with resolution, but rather with the distance to the boundaries, since our boundary conditions become more and more precise the farther away they are.

In this chapter we have introduced most methods and ideas that will be necessary in order to do a numerical relativity simulation, but in a simpler setting. The complication in doing the full numerical relativity evolution

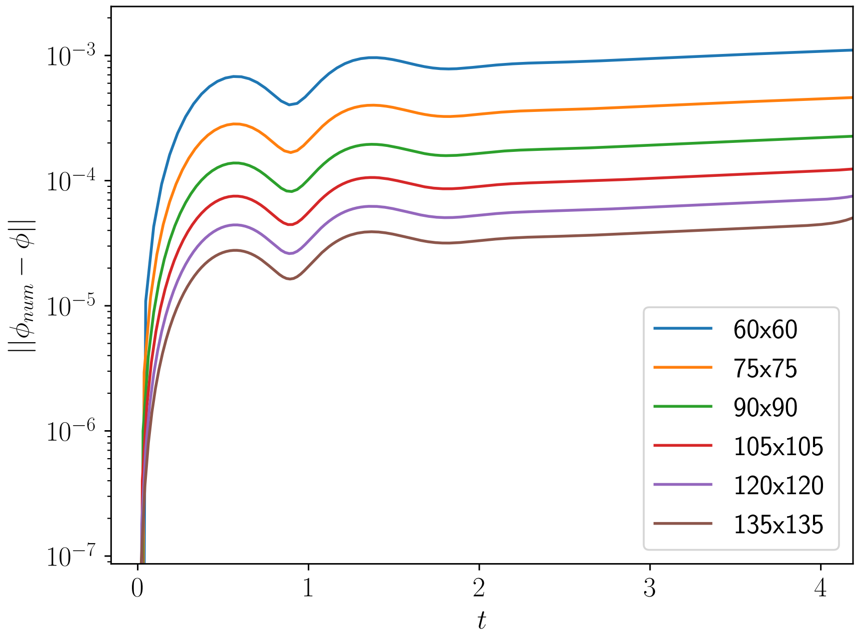
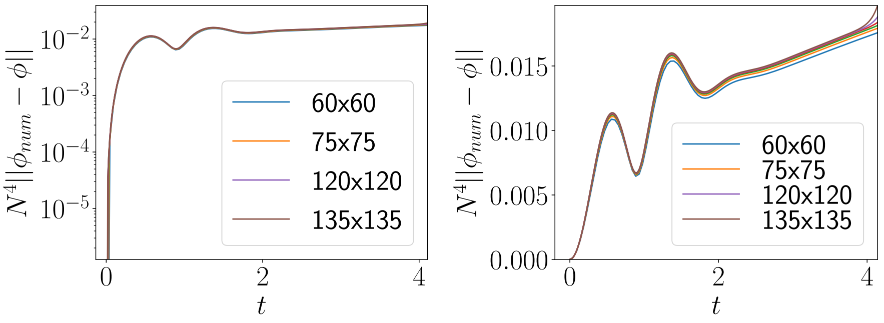


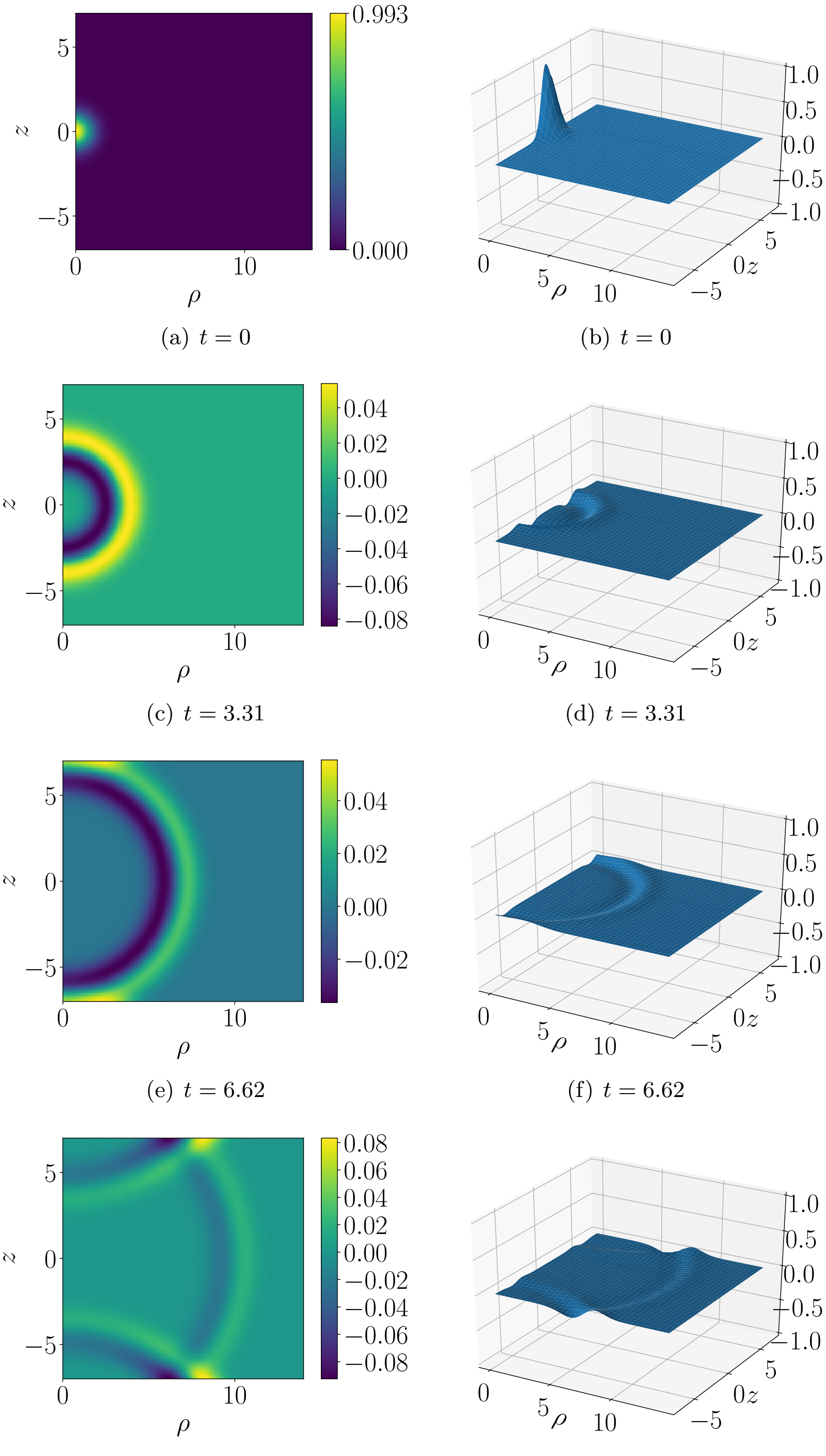
Figure 2.4: L2 norm of the error in the scalar wave simultion as a function of time, for different resolutions.



(a) Log scale

(b) Linear scale

Figure 2.5: Error in the scalar wave simulation multiplied by the resolution factor N to the fourth power. Coincidence of lines means correct convergence. Each line corresponds to the resolution marked in the legend. The maximum grid spacing is $\Delta\rho = \Delta z \simeq 0.47$, and the minimum is approximately 0.049.



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

(c) $t = 3.31$ (d) $t = 3.31$

(e) $t = 6.62$ (f) $t = 6.62$

(g) $t = 9.94$ (h) $t = 9.94$

Figure 2.6: Here we can see some snapshots of the evolution of the initial data discussed in section 2.1

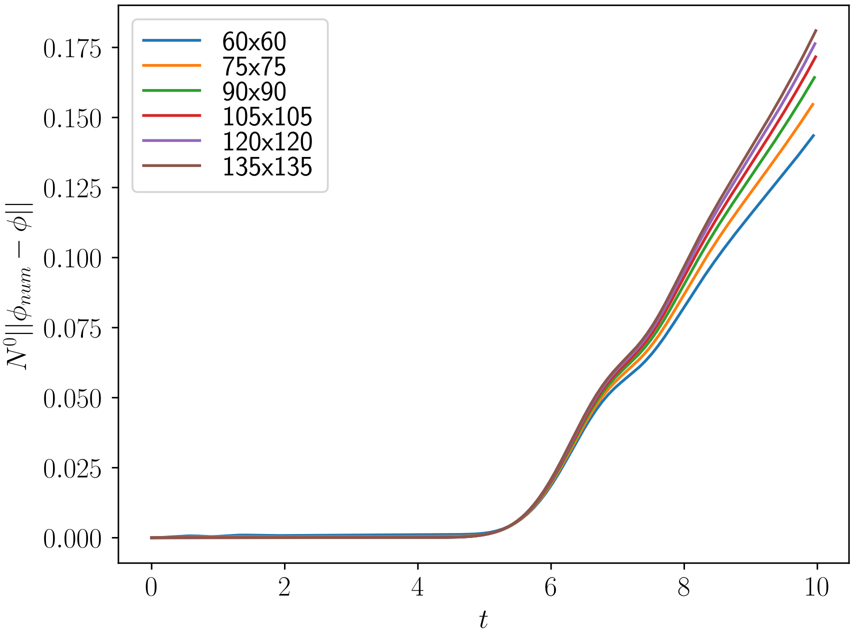


Figure 2.7: The error doesn't converge to zero anymore once the boundary effects become important around $t = 6$, but it converges to the norm of the reflected wave instead. Each line corresponds to the resolution in the legend.

might seem just a practical one, since the equations are going to be much more complex, and that is certainly a big part of it. Implementing the full equations, with all the tensorial terms, is certainly not an easy task. We will find, however, that the complications are not only practical, but we will have to put even more focus on regularity, and also on choosing a correct slicing. These matters will be discussed in the next chapter.

Chapter 3

Numerical Relativity

3.1 BSSN equations

In this section:
General BSSN explanation — definition of the variables — hidden terms
— constraint equations

The problem of solving Einstein's equations numerically is deeply complex, one of the main reasons being coordinate invariance, that mixes gauge and physical degrees of freedom in the solutions. As written by Einstein, they are not specially well suited to be solved as an initial value problem, so I will work with an alternative equivalent formalism, that separates the system of equations in evolution equations for some new variables, and constraints. These are the BSSN equations [25, 26, 27]. An in-depth discussion of the form of this equation is outside the scope of this text, but can be found in [28] and [29] for the specific form used here. I will limit myself to offer the most basic intuition of the terms and make the necessary definitions.

There are many re-formulations of Einstein's equations, but most of them are not stable. The BSSN (by Thomas W. Baumgarte, Stuart L. Shapiro, Masaru Shibata and Takashi Nakamura) formulation has proven empirically to be stable, and that is one of the reasons why nowadays it is mostly the standard in numerical relativity simulations, and that is why it is the formulation I use. It is based on a 3+1 decomposition of space-time, in which it is sliced into spatial hyper-surfaces that cover our space-time parametrized by a time coordinate. This makes it more directly applicable as an initial value problem.

The terms in the equations can loosely be separated into five groups, that

I will mark with different colors in order to make the meaning stand out in the equations

1. Metric terms: they tell us how the geometry of space looks at a single moment in time. They are $(\bar{\gamma}_{ij}, \phi, \bar{\Lambda}^i)$, called the *conformal metric*, the *conformal factor* and the *connection vector*. The first two reproduce the metric on every slice by a conformal transformation
$$\gamma_{ij} = e^{4\phi}\bar{\gamma}_{ij}, \tag{3.1}$$
and hence the name. The connection vector is analytically redundant, since we are going to have a constraint on its value, but one of the main ideas of BSSN is promoting it to an independent variable to evolve. We will see what is the constraint later.

2. Second fundamental form terms: they tell us how the geometry looks when considering the embedding of space into space-time. They carry the degrees of freedom of the conjugate momenta of the metric components, and tell us how those evolve. They are $(K, \bar{A}_{ij})$, which allow us to reconstruct the second fundamental form or extrinsic curvature on each slice
$$K_{ij} = e^{4\phi}\bar{A}_{ij} + \frac{1}{3}\gamma_{ij}K. \tag{3.2}$$
We see that $\bar{A}_{ij}$ corresponds to the trace-free part of the extrinsic curvature, conformally transformed, and K is the trace. I will refer to these terms as the *conformal extrinsic curvature* (sometimes conformal trace-less extrinsic curvature) and the *trace of the extrinsic curvature*.

3. Gauge terms: These terms specify the *slicing*, how the space-time is separated into individual spaces. They are (α, β^i) the *lapse* and the *shift*. Roughly, the lapse tells us how fast time moves at a certain point, and the shift how the spatial coordinates are shifted from one slice to the next (see [28] for a more detailed explanation). If we know these terms and the metric in one slice, we can reproduce the metric in the space-time as follows:
$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij}\left(dx^i + \beta^i dt\right)\left(dx^j + \beta^j dt\right). \tag{3.3}$$
The shift and lapse are freely specifiable, as they represent the freedom in choosing spatial slices of our space-time. In practice, the best way to specify them is through evolution equations, of them the following choice being standard
$$\partial_t\alpha = -2\alpha K; \quad \partial_t\beta^i = B^i; \quad \partial_t B^i = \frac{3}{4}\partial_t\bar{\Lambda}^i. \tag{3.4}$$

This is referred to as the *1+log* slicing with *Gamma-driver* condition for the lapse and shift, respectively. The new variable that appears, B^i, is just used for reducing the order of the time derivative. This is not the only option, but it is important to keep in mind when looking at the equations, that we need another two equations for the lapse and shift that we are free to choose.

4. Matter terms: They represent the matter and energy present at each moment in space. They are (ρ, S, S^i, S_{ij}). These are contractions of the energy momentum tensor that represent the momentum and energy that a normal observer to the spatial slices would measure. The simulations I present in this thesis are in vacuum, so I will not go in more detail about these terms.

5. Background terms: In order to have correct tensorial behavior for all our terms, a reference metric is introduced. These terms do not evolve, so they are not part of our variables, but they will appear in the equations. In particular, we are going to choose the reference metric to be the flat metric, and background terms will be marked with a hat. We have the flat metric $\hat{\gamma}_{ij}$, the connection that is compatible with it and torsion-free $\hat{\Gamma}^k_{ij}$, and the covariant derivative that corresponds to that connection $\hat{\mathcal{D}}_i$.

Due to the introduction of the background terms, we basically have two geometries in our slices: both the one inherited from the space-time curvature, that we conformally re-scale and denote with a bar, and the flat geometry, denoted with a hat. I will refer to the covariant derivatives compatible with each as *conformal derivatives* $\bar{D}_i$ and *flat derivatives* $\hat{\mathcal{D}}_i$.

Now that a basic introduction to all the terms has been presented, these are the BSSN equations, that follow from Einstein equations:

$$\partial_\perp \bar{\gamma}_{ij} = -\frac{2}{3}\bar{\gamma}_{ij}\bar{D}_k\beta^k - 2\alpha\bar{A}_{ij} \tag{3.5a}$$

$$\begin{aligned}\partial_\perp \bar{A}_{ij} =& e^{-4\phi}\left[-2\alpha\bar{D}_i\bar{D}_j\phi + 4\alpha\bar{D}_i\phi\bar{D}_j\phi + 4\bar{D}_{(i}\alpha\bar{D}_{j)}\phi - \bar{D}_i\bar{D}_j\alpha + \right.\\ &\left.\alpha\left(\bar{R}_{ij} - 8\pi S_{ij}\right)\right]^{\mathrm{TF}} + \alpha\left(K\bar{A}_{ij} - 2\bar{A}_{ik}\bar{A}^k_j\right) - \frac{2}{3}\bar{A}_{ij}\bar{D}_k\beta^k\end{aligned} \tag{3.5b}$$

$$\partial_\perp \phi = -\frac{1}{6}\alpha K + \frac{1}{6}\bar{D}_k\beta^k \tag{3.5c}$$

$$\begin{aligned}\partial_\perp K =& \frac{\alpha}{3}K^2 + \alpha\bar{A}_{ij}\bar{A}^{ij} - e^{-4\phi}\left(\bar{D}^2\alpha + 2\bar{D}^i\alpha\bar{D}_i\phi\right) + \\ &4\pi\alpha(\rho + S)\end{aligned} \tag{3.5d}$$

$$\begin{aligned}\partial_\perp \bar{\Lambda}^i =& \bar{\gamma}^{jk}\hat{\mathcal{D}}_j\hat{\mathcal{D}}_k\beta^i + \frac{2}{3}\Delta\Gamma^i\bar{D}_j\beta^j + \frac{1}{3}\bar{D}^i\bar{D}_j\beta^j \\ &- 2\bar{A}^{jk}\left(\delta^i_j\partial_k\alpha - 6\alpha\delta^i_j\partial_k\phi - \alpha\Delta\Gamma^i_{jk}\right) - \frac{4}{3}\alpha\bar{\gamma}^{ij}\partial_j K - 16\pi\alpha\bar{\gamma}^{ij}S_j.\end{aligned} \tag{3.5e}$$

The right-hand side looks complicated, but there are still many terms that conceal further computational complexity. These terms are:

1. Covariant derivatives. These change when the geometry of the space changes, so at every moment in time they are not the same. The flat derivatives don't change during evolution, so the conformal derivatives can be expressed as
$$\bar{D}_i V_j = \partial_i V_j - \bar{\Gamma}^k_{ij} V_K = \partial_i V_j - \hat{\Gamma}^k_{ij} V_K - \Delta\Gamma^k_{ij} V_K = \hat{\mathcal{D}}_i V_j - \Delta\Gamma^k_{ij} V_K. \tag{3.6}$$
Equivalent equations can be found for second derivatives. This still requires us to calculate the connection functions at every time step, which is our next hidden term.

2. The difference of the Christoffel symbols $\Delta\Gamma^k_{ij} = \bar{\Gamma}^k_{ij} - \hat{\Gamma}^k_{ij}$ or *connection functions*, appear in several places in the equations and have to be calculated for taking conformal derivatives, as we just have seen. We can do the calculation in Cartesian coordinates, in which the flat Christoffel symbols vanish
$$\Delta\Gamma^k_{ij} = \frac{1}{2}\bar{\gamma}^{kl}\left(\partial_i \bar{\gamma}_{lj} + \partial_j \bar{\gamma}_{il} - \partial_l \bar{\gamma}_{ij}\right) = \frac{1}{2}\bar{\gamma}^{kl}\left(\hat{\mathcal{D}}_i \bar{\gamma}_{lj} + \hat{\mathcal{D}}_j \bar{\gamma}_{il} - \hat{\mathcal{D}}_l \bar{\gamma}_{ij}\right). \tag{3.7}$$
The resulting formula is coordinate independent, by virtue of partial and flat derivatives coinciding in Cartesian coordinates, and therefore will be valid in general.

3. The conformal Ricci tensor $\bar{R}_{ij}$. A formula for the tensor can be found in [29], as
$$\begin{aligned}\bar{R}_{ij} = -\frac{1}{2}\bar{\gamma}^{kl}\hat{\mathcal{D}}_k\hat{\mathcal{D}}_l\bar{\gamma}_{ij} + \bar{\gamma}_{k(i}\hat{\mathcal{D}}_{j)}\bar{\Lambda}^k + \Delta\Gamma^k\Delta\Gamma_{(ij)k} + \\ +\bar{\gamma}^{kl}\left(2\Delta\Gamma^m_{k(i}\Delta\Gamma_{j)ml} + \Delta\Gamma^m_{ik}\Delta\Gamma_{mjl}\right).\end{aligned} \tag{3.8}$$
As we can see, this term is not trivial, and will pose one of the biggest implementation challenges for a single term.

4. The Lie derivatives. In the equations those are hidden, but we have one for every variable hidden in the time derivative
$$\partial_\perp = \partial_t - \mathcal{L}_\beta. \tag{3.9}$$

5. Raised indices and contractions. Although they make index notation a lot cleaner, it is useful when we need to perform computations to choose a version of the tensor (with all indices down for example) and add a term of the conformal metric (in this case it is the one used to

lower and raise indices) to represent the other versions. Also, we have contractions in the equations like the following:

$$\Delta\Gamma^i = \bar{\gamma}^{jk}\Delta\Gamma^i_{jk}. \tag{3.10}$$

Of course, that is only part of the system. As discussed earlier, we need gauge conditions too, and if there was matter in our space-time we would have to evolve the matter terms too, through hydrodynamics equations. But the other part that we are missing are the constraints, which in our system are the following:

$$\begin{aligned}
\mathcal{M}^i &= e^{-4\phi}\left(6\bar{A}^{ij}\partial_j\phi + \bar{D}_j\bar{A}^{ij} - \frac{2}{3}\bar{\gamma}^{ij}\bar{D}_jK\right) = 0,\\
\mathcal{H} &= \frac{2}{3}K^2 - \bar{A}_{ij}\bar{A}^{ij} + e^{-4\phi}\left(\bar{R} - 8\bar{D}^i\phi\bar{D}_i\phi - 8\bar{D}^2\phi\right) = 0,\\
\mathcal{C}^i &= \bar{\Lambda}^i - \Delta\Gamma^i = 0.
\end{aligned} \tag{3.11}$$

3.2 Regularity problems

In this section:
Curvilinear irregularity and orthonormal frame formalism — parity — moving puncture

One of the main complications in using curvilinear coordinates for our simulations is irregularity problems in special points, like the axis in cylindrical coordinates or the origin in spherical coordinates. This problem is twofold: first, in the differential operators there appear irregular terms, as in the Laplacian, that in cylindrical coordinates and axisymmetry is

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial\rho^2} + \frac{1}{\rho}\frac{\partial\phi}{\partial\rho} + \frac{\partial^2\phi}{\partial z^2} - \frac{1}{c^2}\frac{\partial^2\phi}{\partial t^2}. \tag{3.12}$$

As it was discussed in section 2.2, terms like $1/\rho$ will not be problematic as long as there are no grid points exactly in the irregular places (the axis in this case). The problem we will face here is related to the other type of error I introduced in section 2.2, differentiating irregular fields. These come about because of using a coordinate basis in curvilinear coordinates, as will be discussed in detail below. Afterwards, I will discuss the conditions on the components of a tensor in curvilinear coordinates in order to represent a smooth field, and finally I will present the methods that exist in order to simulate space-times in which the fields are truly irregular, as in a black hole.

Irregularity in a curvilinear basis

To illustrate the problem, let's consider a field in Cartesian coordinates

$$V = V^x \partial_x + V^y \partial_y + V^z \partial_z. \tag{3.13}$$

In Cartesian coordinates, covariant derivatives are simply partial derivatives. Therefore, the regularity condition that we want for our simulation, that covariant derivatives of a field are regular, reduces to the condition that the components are smooth. Later we will see some examples of how that is not true for other coordinate systems. Let's assume that our field is smooth, which implies that all the Cartesian components are smooth, and change to cylindrical coordinates for example. The same vector field will then be expressed as

$$V = \left(\cos\varphi V^x + \sin\varphi V^y\right)\partial_\rho + V^z \partial_z + \rho^{-1}\left(-\sin\varphi V^x + \cos\varphi V^y\right)\partial_\varphi. \tag{3.14}$$

The φ component is no longer regular. A way to understand the problem is to look at the flat metric in cylindrical coordinates

$$\hat{\gamma}_{ij} \doteq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \rho^2 \end{bmatrix}. \tag{3.15}$$

With this metric, the basis vector in the φ direction has zero norm at the axis

$$||\partial_\varphi||_{\hat{\gamma}} = \rho. \tag{3.16}$$

The basis vector shrinks, and the corresponding component has to grow to compensate for that, resulting in an irregular component. We can solve this problem by translating our components into an orthonormal frame, in which all our basis vectors are orthonormal with respect to the flat metric. This will re-scale the components to get rid of the irregularities, and the components will then be differentiable through finite differences.

This method could bring the question, where does the flat metric come from, why use it instead of any other metric? After all, this problem is not geometric, since even if we were on a manifold without metric it will appear. The answer is that, any metric will pick up the problematic factors when doing a change of coordinates

$$\begin{aligned}
\gamma_{\rho\rho} &= \gamma_{xx}\cos^2\phi + \gamma_{yy}\sin^2\phi + (\gamma_{xy} + \gamma_{yx})\sin\phi\cos\phi \\
\gamma_{\rho z} &= \gamma_{xz}\cos\phi + \gamma_{yz}\sin\phi \\
\gamma_{\rho\phi} &= \rho\left[(\gamma_{yy} - \gamma_{xx})\sin\phi\cos\phi + \gamma_{xy}\cos^2\phi - \gamma_{yx}\sin^2\phi\right] \\
\gamma_{z\phi} &= \rho(\gamma_{zy}\cos\phi - \gamma_{zx}\sin\phi) \\
\gamma_{\phi\phi} &= \rho^2\left[\gamma_{xx}\sin^2\phi - (\gamma_{xy} + \gamma_{yx})\sin\phi\cos\phi + \gamma_{yy}\cos^2\phi\right].
\end{aligned} \tag{3.17}$$

And any choice of (smooth) metric with respect to which the basis is orthonormal will get rid of the irregular terms. Using the flat metric is just the simplest choice. It is worth stressing that this is a geometric solution to a non-geometric problem.

Changing the basis everywhere implies changing the components by the change of basis matrix [30]

$$V^A = e_i^A V^i. \tag{3.18}$$

The change of basis matrix to an orthonormal frame is usually referred to in the literature as the *vielbein*, which is calculated from the orthonormality condition

$$e_i^A e_j^B \hat{\gamma}_{ij} = \delta_{AB}. \tag{3.19}$$

Let's look at one example to see how the components are regular in this basis. In cylindrical coordinates the vielbein is

$$e_i^A \doteq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \rho \end{bmatrix}. \tag{3.20}$$

Since it is diagonal, we can identify the directions of the basis vectors one to one with the ones before. The new basis vectors are

$$e_P = \partial_\rho; \quad e_Z = \partial_z; \quad e_\Phi = \rho^{-1}\partial_\varphi. \tag{3.21}$$

And the vector field of our example in that base looks like

$$V = (\cos\varphi V^x + \sin\varphi V^y)\, e_P + V^z e_Z + (-\sin\varphi V^x + \cos\varphi V^y)\, e_\Phi. \tag{3.22}$$

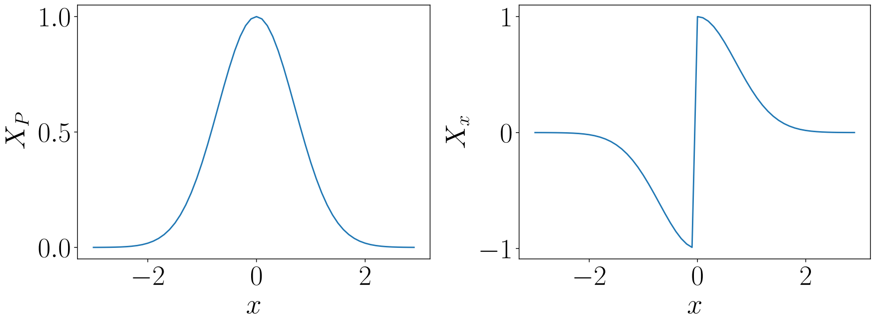
which is regular. This example illustrates how the orthonormal frame formalism works, and it means that all smooth fields have smooth components in any coordinate system. The formulas for all differential terms have to be changed to account for the new basis, and the details can be found in the appendix C.

Parity conditions

Let's assume we have a co-vector field $\boldsymbol{X}$ and in cylindrical components it is given by

$$X_\rho = e^{-\rho^2}; \quad X_z = X_\Phi = 0. \tag{3.23}$$

It looks smooth in cylindrical coordinates, but in Fig. 3.1 we can see that it results in non-continuous Cartesian components, and therefore the first derivative will not be regular. This shows that even when all the components look well-behaved in cylindrical coordinates, they could be representing a



(a) ρ component of a vector field.

(b) The corresponding x component.

Figure 3.1: In (a) the ρ component of a vector field that doesn't go to zero at the origin is presented. In (b) we can see that the the corresponding x component is not regular, signifying that that choice of X_ρ doesn't represent a smooth vector field.

non-smooth field at the origin, and it is important to understand which conditions on the cylindrical components ensure smoothness.

This problem comes about because the basis vectors change sign under a parity inversion

$$\hat{e}_P(\varphi + \pi/2) = \hat{e}_P(\varphi); \quad \hat{e}_\Phi(\varphi + \pi/2) = \hat{e}_\Phi(\varphi). \tag{3.24}$$

Therefore, if we have a continuous vector field at the axis

$$V^P(0, \varphi + \pi/2)\hat{e}(\varphi + \pi/2) = V^P(0, \varphi)\hat{e}(\varphi), \tag{3.25}$$

the component itself has to change sign to account for the change of sign of the basis. This will happen to all tensor fields, if they have a basis term of the form

$$\underbrace{e_\rho \otimes .. \otimes e_\rho}_{n_\rho} \underbrace{e_\Phi \otimes .. \otimes e_\Phi}_{n_\Phi} \underbrace{e_z \otimes .. \otimes e_z}_{n_z}. \tag{3.26}$$

That term has parity $(-1)^{n_\rho + n_\Phi}$, therefore the corresponding term has to have the same parity when $\rho = 0$. In the particular case of axisymmetry, all objects don't depend on φ, and coming back to the example before

$$V^P(0) = -V^P(0). \tag{3.27}$$

All parity -1 objects have to be zero at the axis.

This explains why our example above did not represent a smooth vector field. Incidentally, this also tell us how to implement boundary conditions at the origin, as will be discussed in section 3.3.

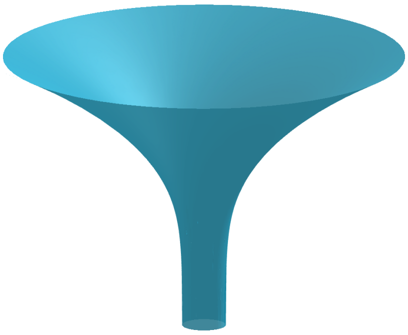


Figure 3.2: Embedding diagram of a trumpet slice of Schwarzschild. An embedding diagram represents the geometry of a two-dimensional surface by embedding it in 3D space [34, 35]

Moving puncture

All the analysis so far was to study when components of a smooth tensor field are smooth and vice versa, in order to be sure we don't take derivatives of non-smooth fields or components. But in numerical relativity, we want to study systems with real irregularities: black holes. If we have such a system, how do we deal with the fact that we are differentiating non-differentiable functions? One of the original ideas, referred to as static puncture nowadays [31, 32], was to treat the offending term analytically in the derivatives, and evolve the rest. This method would be the most direct choice, but it has an important problem: since the irregular term does not evolve, the position and character of the singularity stays fixed. This is a detrimental flaw if we want to evolve moving black holes, for simulations for example of binary mergers. Another idea was needed for such simulations.

That idea is what we now call *moving puncture* [33], for its similarity with the static puncture method and ability to move the singularities throughout the grid. The idea is at the same time straight-forward and subtle. It is straightforward because avoiding regularity concerns, we directly differentiate the irregular terms, without analytic treatment of the derivatives or anything of that sort. It is subtle because in order to have the error be localized in the irregular region, we need to adopt a certain type of slices for our spacetime. These slices are called trumpet slices, because of the shape of their embedding diagrams (Fig. 3.2).

These slices have a couple of important properties. First of all, they don't include the black hole singularity, since they end at a finite areal radius. That is desirable, because that is a very irregular point, that analytically will not affect the rest of our domain, since it is covered by a horizon. The second property is that the boundary around the singularity (which is inside the horizon) is infinitely far away. In the embedding diagram, it is all the way down in the trumpet. That means that although the metric is not regular there, analytically those errors don't propagate outwards.

In simulations, we find that this method works, and although there is the uncontrolled error at the origin, it doesn't contaminate all the domain. Why it works is not completely understood from a computational point of view, but the heuristic picture goes like follows: the grid nodes that are less than one stencil away from the origin have a big error from the irregularity at the origin. That error does propagate outwards computationally, but once we are a couple of grid points away from the singularity, the equations are numerically well described (meaning that the error is convergent to zero). In that area, the extra error will approximately follow the equations, and since signals cannot propagate out of the black hole horizon, that will also be the case approximately in the simulation. That means that the amount of error that gets out of the horizon depends on the accuracy of our method, therefore presenting the same convergence as the rest of the error.

We want our spatial slices to have trumpet geometry, and we can impose that in two ways:

1. We can start with initial data that corresponds to analytical trumpet slices, if those are known. Such initial data for Schwarzschild and Kerr black holes can be found in [36, 37].

2. In a more general case in which trumpet initial data is not available, we can start with another type of slices and use gauge conditions to evolve our slices into the trumpet geometry [33, 38, 39]. The combination of 1+log slicing for the lapse

$$\partial_t \alpha = -2\alpha K, \tag{3.28}$$

   and a Gamma driver condition for the shift

$$\partial_t \beta^i = B^i; \quad \partial_t B^i = \frac{3}{4}\bar{\Lambda}^i, \tag{3.29}$$

   have been found to have the desired property.

To illustrate both methods, I will simulate a Schwarzschild black hole both with trumpet initial data, and with wormhole initial data with the necessary gauge to have it evolve into a trumpet geometry.

3.3 Implementation

In this section:
General implementation strategy —computational objects — boundary conditions — non-fixed constraints

In the previous chapter, we presented the equations, and the main ideas to deal with the irregularity, that is the biggest problem of doing simulations in curvilinear coordinates. Now the general strategy behind the implementation will be discussed.

From section 3.1 we see that implementing the equations, with all those hidden terms, will be highly non-trivial. There are many ways to go about it, for example, one could choose to have a field for every tensor component, and write explicitly in the code the combination of terms that is hidden by the tensor notation. That would be a valid method, but it would run into complications once we wanted to change the spatial coordinates, since the formulas might change. Therefore, I will adopt a different strategy: I will create computational objects that correspond to the mathematical objects as closely as possible, in order to be able to write the equations as similarly as possible to the tensor notation in which we present it. For that I am going to use the following objects:

- GridFunction. It is the most basic object, represents a field in a 2D grid, the same as in the scalar wave case.
- Tensor. Another direct choice. It can be indexed to find its components, which are expressed in the orthonormal frame. We can take partial derivatives of it, but it doesn't have methods for the flat derivatives, since those will depend on our coordinates.
- Metric. Inherits from tensor, but the inverse can be calculated, and it serves to do contractions.
- Coordinates. This object contains all the necessary information about the flat geometry on the grid, and the coordinates we are using for it. It includes the connection and the vielbein to change into the orthonormal frame. It allows us to take flat derivatives of the other objects.
- State. This is the grouping of all different fields into a single object that represents the state at each time and that is evolved. It contains information about the coordinates through a coordinates object, and dedicated functions to calculate and store the hidden terms at each time-step.

It is important to mention that we have to store all tensor components not only because of preserving the possibility of changing coordinates, but also because in axisymmetry, we don't have simplification of tensor components. In spherical symmetry, all vectors have to be radial, for example, so we can only store the radial component of the vectors and write the formulas by hand. Since this doesn't happen in cylindrical coordinates, there is no simplification. This has the effect that the code structure is simple to extend to 3D, if that was my intention in the future.

In terms of how the system is evolved, I will use a MOL-RK4 algorithm to evolve the state object, in the same way that was done for the scalar wave. The constraints will not be imposed, but instead they will be tracked to study error and convergence.

The error will be second order this time. Most partial derivatives are calculated as before, but we have advective terms inside the Lie derivatives

$$\mathcal{L}_\beta T_{ij} = \beta^k \partial_k T_{ij} + ... \tag{3.30}$$

Such terms are calculated through an upwind method for stability reasons. This method has a one-sided stencil, so in order to achieve the same fourth order accuracy, we would need extra ghost nodes. With only two ghost nodes on the boundary, we have to reduce the order of the derivative. According to the discussion of error in section 2.2, even in the non-linear case, the convergence of the error at each time step is given by the part with the slowest convergence, so the total error will be second order.

In terms of outer boundary conditions, the ones in the outer boundary will be the same, with the small difference that the fields will be allowed to oscillate around a non-zero value. The boundary condition this time is

$$\phi(r,t) = \phi_0 + \frac{f(r-t)}{r}. \tag{3.31}$$

Regarding the interior boundary, and as a result of our parity discussion, the fields will be copied into the ghost nodes with a corresponding factor of the parity of that term

$$\phi_{-1,j} = (-1)^{n_\rho + n_\Phi} \phi_{0,j}. \tag{3.32}$$

In the case that we do a simulation in a different system of coordinates, the interior boundary condition will change. For example, in spherical coordinates

$$\phi(-r,\theta) = (-1)^{n_\rho + n_\Phi} \phi(r, \pi - \theta) \Rightarrow \phi_{-1,j} = (-1)^{n_\rho + n_\Phi} \phi_{0,-j}. \tag{3.33}$$

And what also will change is the CFL condition. Since the grid is no longer constituted of equally sized cells, the limiting grid size will be the smallest.

The smallest grid size corresponds to the first cell, that has an area of $\Delta r \times \Delta r \Delta\theta$, therefore, if we use one of the measures discussed before

$$\Delta_{min} = \min(\Delta r, \Delta r \Delta\theta) = \Delta r \Delta\theta. \tag{3.34}$$

The time-step, which is limited by this quantity, will be limited quadratically in resolution, which means that a simulation in these coordinates will be significantly slower.

Chapter 4

Tests and results

In this chapter, I present a discussion of the necessary tested to be confident in the code being correct, and present the results I have obtained by doing three different simulations. The simulations have been chosen to be complementary and give a high degree of confidence on the code. My main tool for testing is comparison with an analytical solution, but I also present a setup without an analytical counterpart to see how we can track the error in that case using the constraints of the system.

4.1 Code testing

The code has many parts that could possibly be wrong individually, so when testing, it is important to keep in mind all the moving pieces and try to isolate them as much as possible with well-chosen tests that only depend on some of them. A schematic view of the parts of my code would be:

1. Object implementation and methods
2. Evolution methods
3. Mathematical operators

The first one is easy to understand and test. My object that represents fields (`GridFunction`), for example, has an operator to sum it with another field, that iterates over the elements and sums them individually. This is the kind of low-level details that it is very important to test for upfront, because at the end there will be a lot such methods. If one of them were to be wrong (for example, if the iteration in the sum is badly implemented and it skips the corners of the domain) it would be really hard to track this error down in a simulation. Therefore, for every object method, I set up several tests with example cases and check that they produce the correct results. Testing tensor methods also makes some simplifications later, since if a formula works for a given tensor, there is no need to test it for the specific components.

The second group encompasses the implementation of the evolution operator itself and the boundary conditions. It is partly tested with the correct scalar wave simulation in chapter 2, but in the full numerical relativity case the boundary conditions will be different, so it needs to be tested again. We gain confidence on them by running simulations that don't crash, and, provided we trust the implementation of the evolution operator on the right-hand side of the evolution equation

$$\partial_t u = \mathcal{L}(u), \tag{4.1}$$

by finding correct convergence to an analytical solution. Among my tests that I will present later, the Teukolsky wave will address mainly this part. It has the advantage that since it is a linear solution, most of the terms in the evolution equations will be zero, so that the correct implementation of those terms and of the evolution operator can be tested separately.

Lastly, the mathematical operators have to be checked. There are a lot of them, including but not limited to:

1. All derivative operators (∂_i ,$\hat{\mathcal{D}}_i$, $\hat{\mathcal{D}}_i\hat{\mathcal{D}}_j$, $\mathcal{L}_\beta$ ...)
2. The calculation of the connection functions $\Delta\Gamma_{ijk}$.
3. The Ricci tensor $\bar{R}_{ij}$.
4. The Hamiltonian and momentum constraints $\mathcal{H}$, $\mathcal{M}^i$.
5. The right-hand side operator in equation (4.1).

The strategy behind most of them is to compare with an analytic solution. They depend on several variables, and for a good test, all the terms should be non-zero. For example, take the Hamiltonian constraint

$$\mathcal{H} = \frac{2}{3}K^2 - \bar{A}_{ij}\bar{A}^{ij} + e^{-4\phi}\left(\bar{R} - 8\bar{D}^i\phi\bar{D}_i\phi - 8\bar{D}^2\phi\right). \tag{4.2}$$

The last term can be separated into the flat derivative and the connection function contribution,

$$\bar{D}^2\phi = \hat{\mathcal{D}}^2\phi - \Delta\Gamma^k\hat{\mathcal{D}}_k\phi. \tag{4.3}$$

In order to test the second term, one needs an analytic solution in which both $\Delta\Gamma^k$ and ϕ are non-zero. This type of considerations will be specially relevant when testing the right-hand side operator, since there are so many terms that it helps to keep in mind which are tested already and which aren't.

I have tested all the mathematical operators. In appendix D an example can be found of how this is done for the Ricci tensor. But the right hand side operator is more complicated: not only finding the analytic solution for a specific state is hard, but it also includes the gauge information. It is possible that a gauge choice has been made that leads to unstable evolution, so that operator needs to be tested in evolution. The simulations presented will work as tests for this operator.

4.2 Teukolsky waves

In this section:
Simulation set-up and analytical comparison — snapshots of the solution — convergence plots

For my first test, I am going to look at a gravitational wave in vacuum. Gravitation waves are solutions to the linearized vacuum Einstein's equations [30]

$$\Box h_{ij} = 0, \tag{4.4}$$

and therefore will only be valid approximations for small amplitudes. The specific waves I am going to look at are Teukolsky waves [40], which are spherical quadrupolar solutions. In an orthonormal frame in spherical coordinates they are given by

$$h_{AB} \doteq \begin{bmatrix} Af_{rr} & Bf_{r\theta} & Bf_{r\phi} \\ Bf_{r\theta} & Cf^{(1)}_{\theta\theta} + Af^{(2)}_{\theta\theta} & (A-2C)f_{\theta\phi} \\ Bf_{r\phi} & (A-2C)f_{\theta\phi} & \left(Cf^{(1)}_{\phi\phi} + Af^{(2)}_{\phi\phi}\right) \end{bmatrix}, \tag{4.5}$$

where the capital constants (A, B, C) are functions of only the radius and time, and the f_{ab} functions are angular, and have five different versions, corresponding to $m = \pm 2, \pm 1, 0$. Of these, only the ones with $m = 0$ are axisymmetric, so those are the ones I will concern myself with here. These are given by

$$\begin{gathered} f_{rr} = 2 - 3\sin^2\theta; \quad f_{r\theta} = -3\sin\theta\cos\theta; \quad f_{r\phi} = 0; \quad f^{1}_{\theta\theta} = 3\sin^2\theta \\ f^{(2)}_{\theta\theta} = -1; \quad f_{\theta\phi} = 0; \quad f^{(1)}_{\phi\phi} = -3\sin^2\theta; \quad f^{(2)}_{\phi\phi} = 3\sin^2\theta - 1. \end{gathered} \tag{4.6}$$

And the radial-time parts are given in terms of a seed function and its derivatives

$$\begin{aligned} A(r,t) &= 3\left[\frac{F^{(2)}}{r^3} + \frac{3F^{(1)}}{r^4} + \frac{3F}{r^5}\right], \\ B(r,t) &= -\left[\frac{F^{(3)}}{r^2} + \frac{3F^{(2)}}{r^3} + \frac{6F^{(1)}}{r^4} + \frac{6F}{r^5}\right], \\ C(r,t) &= \frac{1}{4}\left[\frac{F^{(4)}}{r} + \frac{2F^{(3)}}{r^2} + \frac{9F^{(2)}}{r^3} + \frac{21F^{(1)}}{r^4} + \frac{21F}{r^5}\right]. \end{aligned} \tag{4.7}$$

$F(t,r) = F_1(t-r) + F_2(t+r)$ is the seed function, and its derivatives are defined as

$$F^{(n)} := \left[\frac{d^n F_1(x)}{dx^n}\right]_{x=t-r} + (-1)^n \left[\frac{d^n F_2(x)}{dx^n}\right]_{x=t+r}. \tag{4.8}$$

In my case I will choose the following seed function

$$F(t,r) = \mathcal{A}\left\{(t-r)\exp\left[-\left(\frac{t-r}{\lambda}\right)^2\right] - (t+r)\exp\left[-\left(\frac{t+r}{\lambda}\right)^2\right]\right\}. \tag{4.9}$$

This choice follows the same reasoning as the scalar wave solution discussed in section 2.1, a superposition of ingoing and outgoing waves in such a way that we will have regularity at the origin.

I will perform a test in both spherical coordinates and cylindrical coordinates, the solution in cylindrical coordinates being

$$\begin{aligned} h_{PP} &= A\left[f^{(2)}_{\theta\theta}\cos^2\theta + f_{rr}\sin^2\theta\right] + Bf_{r\theta}\sin(2\theta) + Cf^{(1)}_{\theta\theta}\cos^2\theta, \\ h_{PZ} &= A\left[f_{rr} - f^{(2)}_{\theta\theta}\right]\cos\theta\sin\theta + Bf_{r\theta}\cos(2\theta) - Cf^{(1)}_{\theta\theta}\cos\theta\sin\theta, \\ h_{P\Phi} &= (A-2C)f_{\theta\phi}\cos\theta + Bf_{r\phi}\sin\theta, \\ h_{ZZ} &= A\left[f_{rr}\cos^2\theta + f^{(2)}_{\theta\theta}\sin^2\theta\right] - 2Bf_{r\theta}\sin\theta\cos\theta + Cf^{(1)}_{\theta\theta}\sin^2\theta, \\ h_{z\Phi} &= -(A-2C)f_{\theta\phi}\sin\theta + Bf_{r\phi}\cos\theta, \\ h_{\Phi\Phi} &= Cf^{(1)}_{\phi\phi} + Af^{(2)}_{\phi\phi}. \end{aligned} \tag{4.10}$$

A schematic view of how these terms are calculated can be found in appendix E.

As mentioned in the previous section, this will be good to test the evolution methods in my code, since it is dynamical, and because it is linear, most terms in the right-hand side operator will not play a role. That is helpful because in the case that the simulation didn't work and all the terms played a role in it, the error would be very hard to track down.

The error will be calculated in the same way as in the scalar wave case: the numerical solution is to be compared with the analytic solution at each time step and the norm of the error is compared for different resolutions. We expect fourth order convergence at this point. The gauge conditions used are zero shift and harmonic slicing

$$\partial_t \alpha = -\alpha^2 K; \quad \beta^i = 0. \tag{4.11}$$

Given all that, we have the necessary ingredients to perform the simulation. We can see snapshots of the evolution of the $h_{PP} = h_{\rho\rho}$ component in Fig. 4.1. These results are from the cylindrical coordinates simulation. As before, simple inspection doesn't ensure the correctness of the results, so I have presented the convergence tests for both cylindrical and spherical coordinates simulations in Fig. 4.2, where we can see that the error is fourth order as expected, since the shift is zero and therefore there are no upwind derivatives. This test gives us good confidence in the evolution algorithm of my code.

4.3 Trumpet initial data

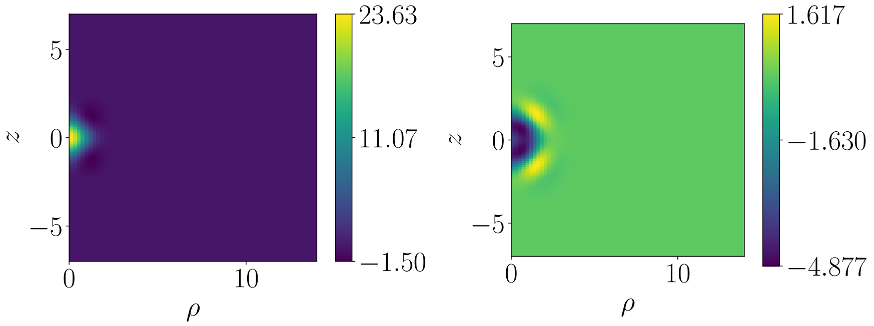
In this section:
Initial data — Comparison of X and ϕ as a variable — Convergence plots

As mentioned in the previous chapter, I am going to simulate the Schwarzschild black hole with two types of initial data: one that already has the trumpet geometry and one that will evolve into it. The trumpet initial data is given in [36] and translated to cylindrical coordinates it is

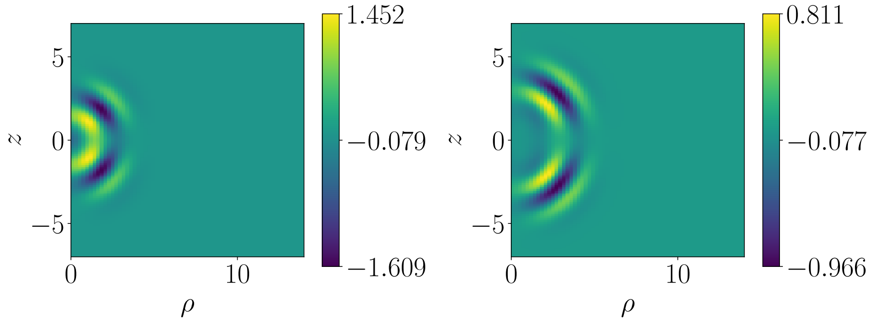
$$\begin{aligned}
&\alpha = \frac{\bar{r}}{\bar{r}+1}; \quad MK = \frac{1}{(\bar{r}+1)^2}; \quad \psi = e^{\phi} = \sqrt{1+\frac{1}{\bar{r}}}; \quad \bar{\gamma}_{ij} = \eta_{ij};\\
&M\bar{A}_{AB} \doteq \frac{2}{3}\frac{1}{(\bar{r}+1)^2}\begin{bmatrix} 1-3\rho^2/r^2 & -3z\rho/r^2 & 0 \\ -3z\rho/r^2 & 1-3z^2/r^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \\
&\beta^A \doteq \frac{1}{(\bar{r}+1)^2}\begin{bmatrix} \bar{\rho} \\ \bar{z} \\ 0 \end{bmatrix}.
\end{aligned} \tag{4.12}$$

Where $\bar{r} = r/M = \sqrt{\rho^2+z^2}/M$ is the radius in units of the black hole mass. This radius is related to the areal radius as

$$R = (\bar{r}+1)M. \tag{4.13}$$

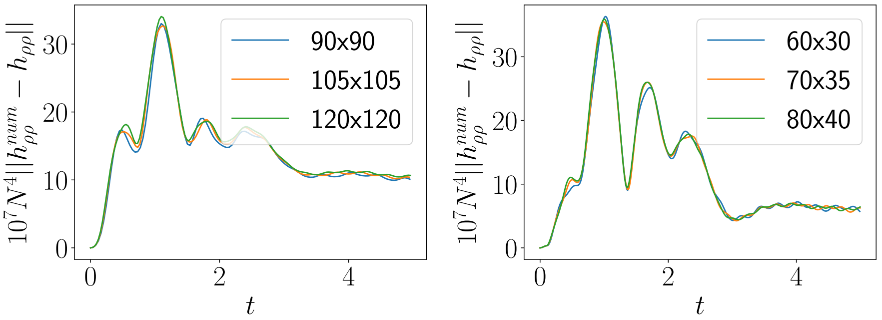


(a) $t = 0$ (b) $t = 1.2$



(c) $t = 2.4$ (d) $t = 3.6$

Figure 4.1: Snapshots of the evolution of the Teukolsky wave. I am plotting $10^7 h_{\rho\rho}$.



(a) Results in cylindrical coordinates. (b) Results in spherical coordinates.

Figure 4.2: The norm of the difference between the computational and analytical solution, scaled with the resolution to fourth order. Coincidence of lines means correct convergence. Each line corresponds to the resolution given in the legend. In (a), the lines range from a grid spacing of 0.15 to 0.12. In (b) those resolutions correspond to $\Delta r = 0.3$, $\Delta\theta \simeq 0.13$ and $\Delta r = 0.19$, $\Delta\theta \simeq 0.079$.

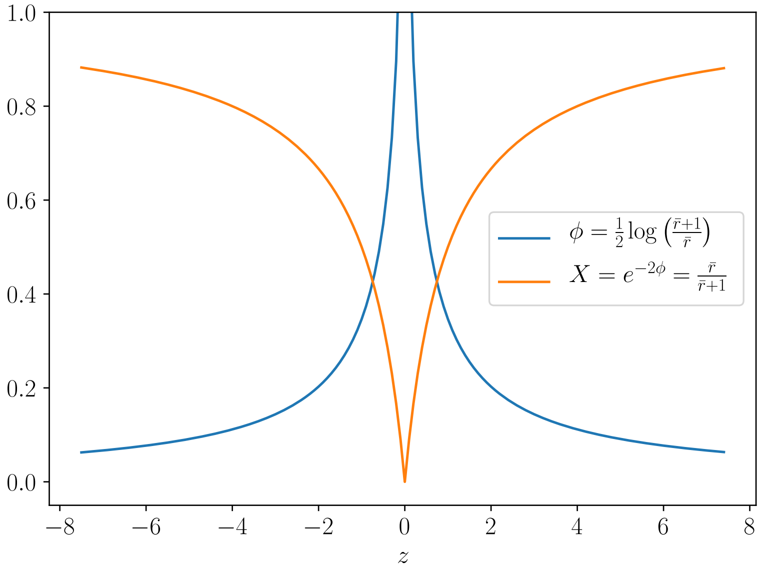


Figure 4.3: Comparison of X and ϕ for the initial data. Both are not differentiable.

Using the variable $X = e^{-2\phi}$ instead of ϕ turns out to give better results numerically [33, 41, 42], as I also found in my code. In this case we have

$$X = \frac{\bar{r}}{\bar{r}+1}. \tag{4.14}$$

We can see a plot of both fields along the $\rho = 0$ line in Fig. 4.3. Both are not differentiable, so a priori the error should be the same

I think the difference comes in the following way. Although we cannot use Taylor's expansion (2.30) in this case, looking at it might give us a hint. The error term depends on the derivative in a point of the stencil domain. Although X is not differentiable at $\bar{r} = 0$, if we look at the derivatives they are all bounded in the region $\mathbb{R} - \{0\}$, while that is not the case for ϕ. Therefore, the error from differentiating ϕ can be much bigger than it is for X. This is, of course, hand-wavy, because we don't have any formula for the error, but I think it gives us a hint of why X could be a better variable numerically, even though it is still not differentiable. Also, the effect of the finite difference is to "smooth" features smaller than the grid spacing. For X, this means that the kink in the function will be lost, but that is not such a big error. For ϕ, the irregularity gets smoothed out, resulting in a formally infinite error.

Another thing to consider is that the solution will only be time-independent if a suitable gauge is chosen. In this case that choice is

$$\partial_\perp \alpha = \alpha(\alpha - 1)K. \tag{4.15}$$

Given that the solution has to be time-independent, the error in our simulation can be studied by looking at the time derivative of the state

$$\partial_t u = \mathcal{L}(u) = 0 \text{ analytically.} \tag{4.16}$$

We can see the results in Figs. 4.4(a) and 4.4(b), both in log and linear scale. There are a couple of things we can see from those plots: first, we have correct second order convergence everywhere except on the area close to the origin (as expected per moving puncture) and the region that is in causal contact with the boundaries. That error is also as expected, and it doesn't improve with resolution since it comes from an approximation that does not depend on resolution, but rather on distance from the origin. It is equivalent to the error studied in section 2.3. On the linear scale plot, we can observe how the boundary that is closer (the left one) produces a bigger error.

The results of the test are therefore satisfactory. Most of the terms are in the evolution equations are tested in this simulation, except the ones that depend on $\Delta\Gamma_{ijk}$ or $\bar{\Lambda}^i$, since those are zero here.

4.4 Wormhole initial data

In this section:
Initial data — Hamiltonian constraint for testing — Gauge choice —
Wrong convergence and possible reasons

Let's turn our attention now to wormhole initial data

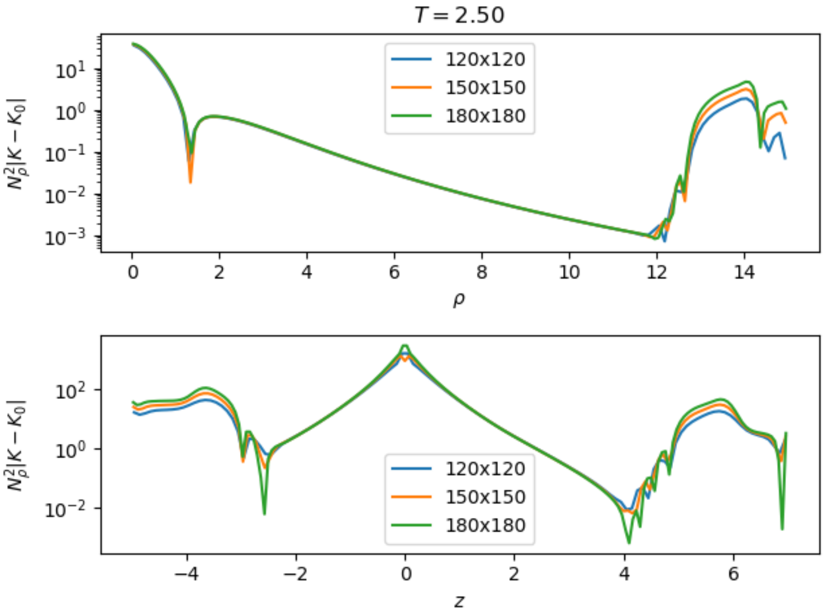
$$\begin{aligned}&\alpha = \left(\frac{\bar{r}}{\bar{r}+1/2}\right)^2; \quad MK = 0; \quad \psi = e^{\phi} = 1 + \frac{1}{2\bar{r}}; \quad \bar{\gamma}_{ij} = \eta_{ij};\\ &M\bar{A}_{AB} = 0; \quad \beta^A = 0.\end{aligned} \tag{4.17}$$

The lapse choice $\alpha = \psi^{-2} = X$ is referred to as a "pre-collapsed" lapse, because it starts being zero at the puncture, instead of evolving towards that. For this initial data to develop into a trumpet geometry we need to use the standard slicing conditions mentioned in section 3.2, that are given by the 1+log lapse condition

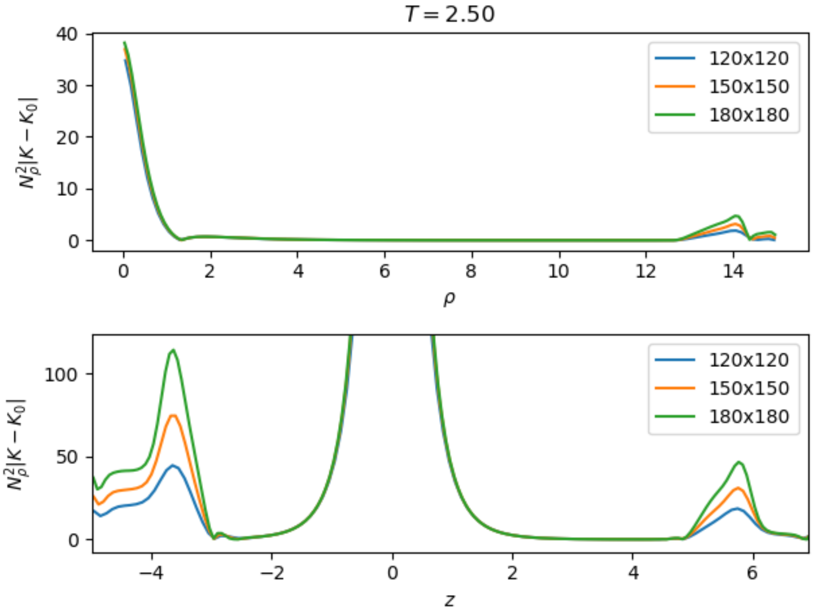
$$\partial_t \alpha = -2\alpha K, \tag{4.18}$$

and a Gamma driver condition for the shift

$$\partial_t \beta^i = B^i; \quad \partial_t B^i = \frac{3}{4}\bar{\Lambda}^i. \tag{4.19}$$

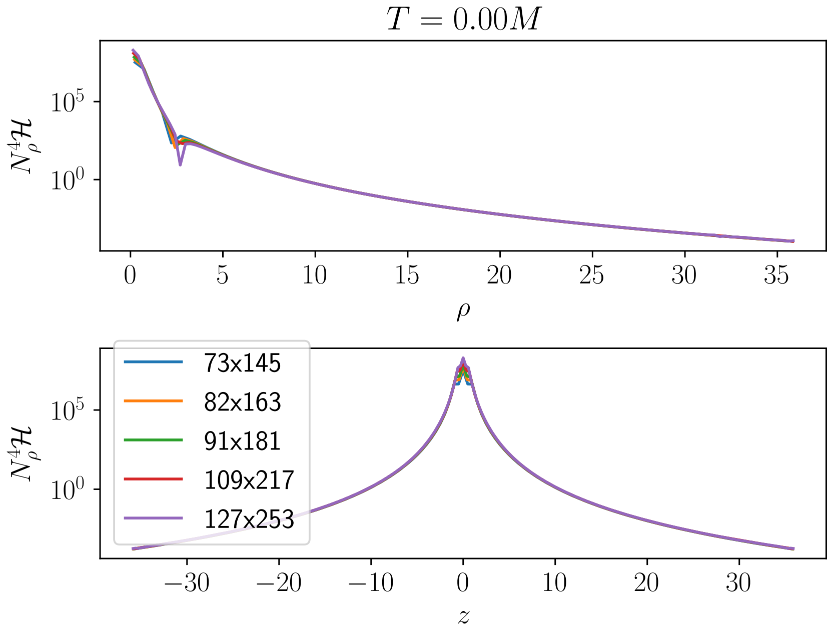


(a) Results in log scale

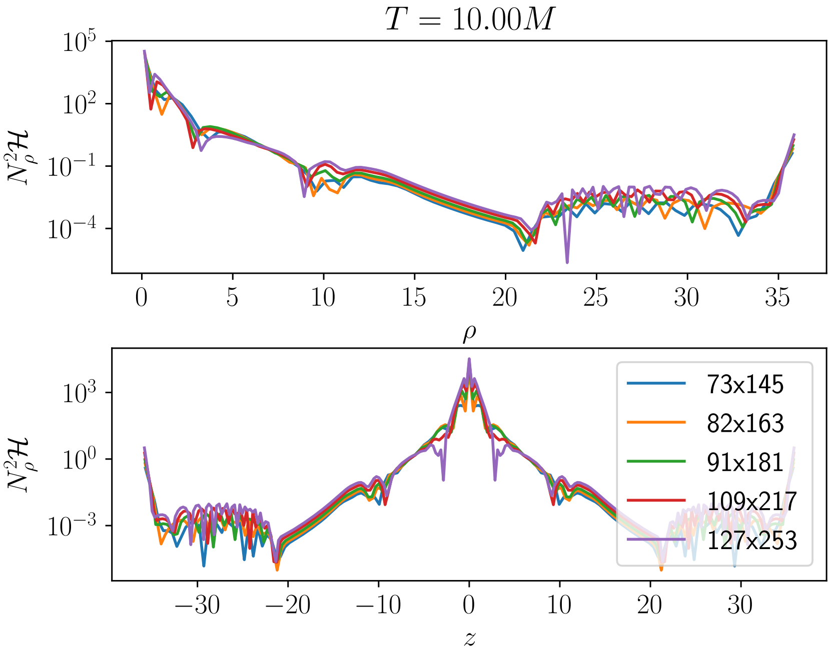


(b) Results in a linear scale

Figure 4.4: Plots of the error in the simulation with trumpet initial data, calculated by comparison with the analytical solution. The difference lines correspond to the different resolutions in the legend. Coincidence of lines means correct convergence. In (a) we can see that the convergence is correct except inside the horizon and near the boundaries, as expected. In (b) we can see how the area that is in causal contact with the boundary has a constant error with resolution, since the boundary conditions are approximate, and the error is bigger the closer the boundaries are.



(a) Violation of the hamiltonian constraint for the initial data.



(b) Violation of the Hamiltonian constraint at $t = 10M$

Figure 4.5: Violation of the Hamiltonian constraint at $t = 18M$ for different resolutions, each line corresponding to the resolution given in the legend. Those resolutions correspond to $\Delta\rho \simeq \Delta z \simeq 0.49$ to $\Delta\rho \simeq \Delta z \simeq 0.33$. In (a) we can see correct fourth order convergence, for initial data, as expected. For (b) we see second order convergence (as expected because of the upwind derivatives) in part of the domain, but not all of it, even excluding the areas that are in causal contact with the boundary. This should be investigated further.

In this case, we use the violation of the Hamiltonian constraint to track the error, since the analytical solution is not known. The results can be found in Fig. 4.5(a) and Fig. 4.5(b). The convergence is plotted both for the initial data, in which the expected convergence is fourth order since $\beta^i = 0$ and therefore there are no advective derivatives, and for a later time. The correct convergence for initial data indicates that it is correctly set up. For a later time, correct convergence is not found in all of the domain, as in the figure the region $10M < r < 20M$ seems to have constant error. This should be investigated further, although the two previous tests give us a strong indication of where the error lies. The only terms that were not tested before are those that depend on $\bar{\Lambda}_i$ and $\Delta\Gamma_{ijk}$, which leaves us with few candidates for the erroneous term. Alternatively, since the grid spacing is rather big in this simulation $0.33 \leq \Delta \leq 0.49$, it is possible that we are still not in the region of convergence, since the leading error term doesn't dominate.

Chapter 5

Conclusion and future work

In this thesis, I have presented my work in creating a new numerical relativity code in axisymmetry, with arbitrary spatial coordinates. This, although not innovative, is important for two reasons. The first being the low-level understanding of the methods of numerical relativity that I have developed through the endeavor, that is going to be very useful in my future research. The second is because I have developed a new tool, with the prospect that it will be specially useful for problems I am interested in, as critical collapse [43, 44]. Among the main takeaways from this work are the requirements and methods needed to do stable simulations when working with irregular terms, how the numerical error behaves in numerical relativity simulations and how to translate the complicated system of equations into a code that is as simple as possible.

But this work is far from done, there are many improvements that would increase the usefulness of the code. In the short term, I would like to fix the faulty wormhole initial data simulation and simulate a black hole head on collision, since that would just involve setting up the analytical initial data. In the medium term, the code would benefit from an implementation of Kreiss-Oliger dissipation [45] and parallelization. This would allow me to run for far longer times, and see more interesting physics. A solver for the initial data (solving the constraint equations at $t = 0$) [46] could be added in this stage, what would open the door for far more complex and interesting simulations. Finally, once all of that was done, my code would be suitable for studying critical collapse for long times, which is one of my interests. At this stage the code could be extended to 3D, since this wouldn't be such a big change implementation-wise, and it would allow me to simulate a vast array of systems, from binary black hole mergers, to non-symmetric collapse etc...

As a conclusion, this works leaves me with my feet firmly planted in the field of numerical relativity, which has proven of invaluable importance to the study of gravity and the universe. Armed with my code and the knowledge obtained from it, I hope to be able to reach interesting and exciting findings and results in the near future.

Appendix A

Analytical solutions of the wave equation in cylindrical coordinates

To find analytical solutions to the equation I will use the method of separation of variables, in which the choice of coordinates is equivalent to choosing a basis for the solution.[1] In this appendix, I present that basis in cylindrical coordinates. Let's remember that the axisymmetry condition implies that we are restricting our study to the subspace of solutions that don't depend on φ, over which derivating with respect to φ is always zero $\partial_\varphi = 0$. The equation in these coordinates is

$$\frac{\partial^2 \phi}{\partial \rho^2} + \frac{1}{\rho}\frac{\partial \phi}{\partial \rho} + \frac{\partial^2 \phi}{\partial z^2} - \frac{1}{c^2}\frac{\partial^2 \phi}{\partial t^2} = 0. \tag{A.1}$$

Let's look for a separation of variables basis of solutions, in which each element will be of the form

$$\phi_\alpha = T(t)R(\rho)Z(z). \tag{A.2}$$

The procedure is straight-forward, and we are left with three terms that only depend on one of the variables, therefore each of them has to be constant individually,

$$\underbrace{\frac{R''}{R} + \frac{1}{\rho}\frac{R'}{R}}_{\lambda} + \underbrace{\frac{Z''}{Z}}_{\mu} - \underbrace{\frac{1}{c^2}\frac{T''}{T}}_{\eta} = 0, \tag{A.3}$$

---

[1] This, of course, is only a product of our analytical method. At the numerical level, the function is not represented in terms of a basis, so the motivation to choose one coordinate system will mainly be symmetry and accuracy related.

with the constraint $\lambda + \mu - \eta = 0$. Without further information, we should include the solutions with both positive and negative constants in our basis. But if we restrict our interest to a particular space of solutions, like L2, it turns out that some solutions are linearly dependent on the others, and we don't need them as part of the basis. Our interest lays in simulations though, so one could ask: how do we restrict the results of our simulation to represent functions in a specific function space? The solution is simpler than it looks: if the initial data is in L2 (what we could call realistic initial data) then the solution will also be.[2] The only terms that we need for our basis are the oscillatory terms, therefore I will make the re-definitions

$$\lambda = -k_\rho^2, \quad \mu = -k_z^2, \quad \eta = -\omega^2/c^2 \Rightarrow \omega^2/c^2 = k_\rho^2 + k_z^2. \tag{A.4}$$

The separated equations for our basis terms are then

$$\begin{cases} \rho^2 R'' + \rho R' + k_\rho^2 \rho^2 R = 0 \\ Z'' + k_z^2 Z = 0 \\ T'' + \omega^2 T = 0 \end{cases}. \tag{A.5}$$

The first equation is the only one that is not trivial. If we change the variable $\rho \to k_\rho \rho$, we can see by inspection that is the zeroth order Bessel function. Therefore, the elements of our basis are the following functions

$$\phi_{\boldsymbol{k}} = \left[A_{k_\rho} H^{(1)}(k_\rho \rho) + B_{k_\rho} H^{(2)}(k_\rho \rho)\right] \left[C_{\boldsymbol{k}} e^{-i(k_z z - \omega t)} + D_{\boldsymbol{k}} e^{-i(k_z z + \omega t)}\right]. \tag{A.6}$$

The Hankel functions of order zero have the following asymptotic form

$$H^{(1)}(k_\rho \rho) \approx \sqrt{\frac{-2i}{\pi}} \frac{e^{ik_\rho \rho}}{\sqrt{k_\rho \rho}}; \quad H^{(2)}(k_\rho \rho) \approx \sqrt{\frac{2i}{\pi}} \frac{e^{-ik_\rho \rho}}{\sqrt{k_\rho \rho}}. \tag{A.7}$$

Far away, the basis of the solution just look like regular waves, but scaled by a geometric factor

$$\phi_{\boldsymbol{k}} \approx \frac{1}{\sqrt{k_\rho \rho}} \left[A_{\boldsymbol{k}} e^{-i(k_z z + k_\rho \rho - \omega t)} + B_{\boldsymbol{k}} e^{-i(k_z z + k_\rho \rho + \omega t)}\right]. \tag{A.8}$$

I think this asymptotic form, although not exact, gives us a lot better intuition of how this waves look like. Let's think about how physical these solutions are. In particular, let's consider we have some system, that has some structure around the axis. Maybe it is a vibrating dipole, or maybe it is a black hole head on collision that is sending gravitational waves very far away. For these systems, the sources are localized in a given area, and if we go far away, the radiation that reaches us is only the one coming from that

---

[2] Keep in mind that this will no longer be the case when we move to the non-linear case. But in that case we will not try to introduce a basis either, as that is a linear space concept.

part of space, we cannot have waves with infinite wave-fronts. But that is exactly how our solutions (A.8) look like. In practice, this means that in order to resolve any "physical" wave in this basis, we will need a big superposition of elements so that they cancel far away. That is what motivates us to think about spherical waves as a basis in the text instead.

Appendix B

Stability analysis of the method

Let's perform a Von Neumann stability analysis of the evolution method. In such an analysis, we take the basis functions (calculated in the previous appendix) and expect the algorithm to act linearly on them. Then we check whether the solutions grow exponentially, in order to find a necessary condition for stability in the case that they don't. In mathematical terms, we have the solution at one time-step, and we get the solution on the next time-step

$$u^{n+1} = \alpha u^n, \tag{B.1}$$

and the necessary condition for stability is $\alpha < 1$. α typically depends on the algorithm parameters $(\Delta t, \Delta x, ...)$ so this is a condition on them. Let's calculate that factor for my algorithm, starting with the finite differences acting on the basis functions, given by

$$\phi = J_0(k_\rho \rho) e^{-ik_z z}. \tag{B.2}$$

Since it is a product of functions of the individual variables, we can look at the action of the finite difference Laplacian on each individually.

$$\nabla^2_{FD}\phi = J_0(k_\rho \rho)\nabla^2_{FD} e^{-ik_z z} + e^{-ik_z z}\nabla^2_{FD} J_0(k_\rho \rho). \tag{B.3}$$

The first term is easy to calculate

$$\begin{aligned}\nabla^2_{FD} e^{-ik_z z} &= e^{-ik_z z}\frac{-(e^{ik_z 2\Delta z} + e^{-ik_z 2\Delta z}) + 16(e^{ik_z \Delta z} + e^{-ik_z \Delta z}) - 30}{12\Delta z^2} = \\ &= -\frac{1}{\Delta z^2}\frac{\cos(2k_z\Delta z) - 16\cos(k\Delta z) + 15}{6} e^{-ik_z z} = -\frac{F_z(k_z\Delta z)}{\Delta z^2} e^{-ik_z z}.\end{aligned} \tag{B.4}$$

For the Bessel function, I didn't calculate the proportionality factor analytically, but looked at the effect of the finite difference operator on the function and tried to find a proportionality constant. This, of course, is a weak point of this analysis, since I can not be sure the effect is exactly linear in that case, although the numerical results give some confidence about it. It seems to be the case that

$$\nabla^2_{FD} J_0(k_\rho \rho) = -\frac{F_\rho(k_\rho \Delta\rho)}{\Delta\rho^2} J_0(k_\rho \rho), \tag{B.5}$$

with F_ρ an oscillating function bounded by $F_\rho(x) \in [0, 5.324..]$. If that is the case, the effect on the basis function of the discretized operator is

$$\nabla^2_{FD}\phi = -\left[\frac{F_z(k_z\Delta z)}{\Delta z^2} + \frac{F_\rho(k_\rho\Delta\rho)}{\Delta\rho^2}\right]\phi. \tag{B.6}$$

Let's look now at the fourth order Runge-Kutta, to see what are the bounds on that factor From the previous discussion, the RHS operator of the scalar wave equation, when discretized, acting on a basis element is

$$L(u) = \left[\begin{array}{c} \psi \\ c^2\nabla^2_{FD}\phi \end{array}\right] = \left[\begin{array}{cc} 0 & 1 \\ -c^2\hat{F}(k_z\Delta z, k_\rho\Delta\rho) & 0 \end{array}\right]\left[\begin{array}{c} \phi \\ \psi \end{array}\right], \tag{B.7}$$

where, for simplicity, I have made the definition:

$$\hat{F} := \left[\frac{F_z(k_z\Delta z)}{\Delta z^2} + \frac{F_\rho(k_\rho\Delta\rho)}{\Delta\rho^2}\right]. \tag{B.8}$$

The algorithm is the following:

$$\begin{aligned} k_1 &= L(\hat{u}^n), \\ k_2 &= L(\hat{u}^n + k_1\Delta t/2), \\ k_3 &= L(\hat{u}^n + k_2\Delta t/2), \\ k_4 &= L(\hat{u}^n + k_3\Delta t), \\ \hat{u}^{n+1} &= \hat{u}^n + \frac{\Delta t}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right). \end{aligned} \tag{B.9}$$

Which in our case reduces to

$$\begin{aligned} k_1 &= M\hat{u}^n, \\ k_2 &= M\hat{u}^n + Mk_1\Delta t/2, \\ k_3 &= M\hat{u}^n + Mk_2\Delta t/2, \\ k_4 &= M\hat{u}^n + Mk_3\Delta t, \\ \hat{u}^{n+1} &= \left(1 + \Delta t M + \frac{1}{2}\Delta t^2 M^2 + \frac{1}{6}\Delta t^3 M^3 + \frac{1}{24}\Delta t^4 M^4\right)\hat{u}^n. \end{aligned} \tag{B.10}$$

Note how the evolution term is a truncation to fourth order of the expansion of the exponential, which would be the exact solution for the problem $\partial_t u =$

Mu. The stability condition can be expressed as the determinant of the evolution matrix being less than one

$$R := 1 + \Delta t M + \frac{1}{2}\Delta t^2 M^2 + \frac{1}{6}\Delta t^3 M^3 + \frac{1}{24}\Delta t^4 M^4; \quad \det(R) \leq 1. \tag{B.11}$$

The determinant of that matrix is

$$\det(R)(\lambda = \hat{F}\Delta t^2) = \left(1 - \frac{\lambda}{2} + \frac{\lambda^2}{4!}\right)^2 + \lambda\left(1 - \frac{\lambda}{3!}\right)^2, \tag{B.12}$$

and the condition of this being less than one implies $\lambda < 8$, therefore

$$c^2 \Delta t^2 \left[\frac{F_z(k_z \Delta z)}{\Delta z^2} + \frac{F_\rho(k_\rho \Delta\rho)}{\Delta\rho^2}\right] \leq \left(\frac{c\Delta t}{\Delta_2}\right)^2 F_M \leq 8. \tag{B.13}$$

Where $F_M = \max(F_z, F_\rho) \approx 5.4$, which results in the stability condition

$$\frac{c\Delta t}{\Delta_2} \lesssim 1.21. \tag{B.14}$$

Appendix C

Changes in the orthonormal frame formalism

When using the orthonormal frame formalism, the derivatives will change, since at each point we have a non-constant matrix acting on the components. The covariant derivatives will have a similar form by introducing the spin connection ω^{B}_{CA} [30]

$$D_C X_A = e^{i}_{C}\partial_i X_A - \omega^{B}_{CA} X_B; \quad \partial_C = e^{i}_{C}\partial_i. \tag{C.1}$$

We will be justified in writing the covariant derivatives this way as long as it will reduce to the usual formula when changing back to the coordinate basis. That will always happen if we define the spin-connection as

$$\omega^{B}_{CA} := e^{B}_{k} e^{j}_{A} e^{i}_{C} \Gamma^{k}_{ij} - e^{k}_{A} e^{i}_{C} \partial_i e^{B}_{k}. \tag{C.2}$$

This formula has a very important property, in that it relates all possible spin connections with their Christoffel symbols. In particular, it works for connections that are not compatible with the metric for which the vielbeins are orthonormal. In our case, that means that we can use this formula for conformal derivatives too. In general, the above formula works for any change of basis, not just to an orthonormal one.

The above definition implies

$$D_C e^{A}_{i} = 0. \tag{C.3}$$

For any possible covariant derivative. That is very important since it will allow us to change basis inside derivatives.[1]

---

[1] It is important to note that this convention is not universal, some people define the

The fact that these formulas work for every connection, independently of whether or not it is related to the metric with respect to which the basis is orthonormal has one important consequence. That is that the difference of spin-connections corresponds to the difference of Christoffel symbols, translated into the orthonormal frame

$$\Delta\omega^{C}_{AB} = \bar{\omega}^{C}_{AB} - \hat{\omega}^{C}_{AB} = e^{i}_{A} e^{j}_{B} e^{C}_{k} \Delta\Gamma^{k}_{ij}. \tag{C.5}$$

So that it will not be a new variable, and the formula used for translating derivatives will be the same in the orthonormal frame. In general, tensorial terms in the equations will have the same expressions, but non-tensorial terms (as the connection) might change.

The Lie derivative is another of the terms that have to be modified. In a coordinate basis it is given by

$$\mathcal{L}_\beta T_{ij} = \beta^k \partial_k T_{ij} + (\partial_i \beta^k) T_{kj} + (\partial_j \beta^k) T_{ik}. \tag{C.6}$$

The individual terms in the formula are not tensorial, and therefore they are subject to change. In fact, since we have partial derivatives, there are going to be some extra terms with derivatives of the vielbein

$$e^{C}_{AB} = e^{C}_{k} e^{i}_{A} \partial_i e^{k}_{B}. \tag{C.7}$$

The full equation in this case is

$$\begin{aligned}(\mathcal{L}_\beta T)_{AB} =& \beta^C \partial_C T_{AB} + \left(\partial_A \beta^C\right) T_{CB} + (\partial_B V^C) T_{AC} + \\ &+ 2 e^{C}_{[AD]} V^D T_{CB} + 2 e^{C}_{[BD]} V^D T_{AC}.\end{aligned} \tag{C.8}$$

Let's now look at a specific example, cylindrical coordinates. The vielbein was given in the text as

$$e^{A}_{i} \doteq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \rho \end{bmatrix}. \tag{C.9}$$

The only non-zero Christoffel symbols are

$$\hat{\Gamma}^{\rho}_{\phi\phi} = -\rho; \quad \hat{\Gamma}^{\phi}_{\rho\phi} = \hat{\Gamma}^{\phi}_{\phi\rho} = \rho^{-1}. \tag{C.10}$$

The only non-zero spin-connection values are, then

$$\hat{\omega}^{\Phi}_{\Phi P} = \rho^{-1}; \quad \hat{\omega}^{P}_{\Phi\Phi} = -\rho^{-1}. \tag{C.11}$$

---

spin connection as

$$D_C e^{A}_{i} = \omega^{A}_{iC}, \tag{C.4}$$

and that will result in a different formula for the covariant derivative. In our case, I think that the convention I used is more useful.

Some derivatives in this system of coordinates are presented.

The first derivative of a first rank tensor is

$$\hat{\mathcal{D}}_A X_B \doteq \begin{bmatrix} \partial_\rho X_\rho & \partial_\rho X_z & \partial_\rho X_\Phi \\ \partial_z X_\rho & \partial_z X_z & \partial_z X_\Phi \\ \rho^{-1}\left(\partial_\phi X_\rho - X_\Phi\right) & \rho^{-1}\partial_\phi X_z & \rho^{-1}\left(\partial_\phi X_\Phi + X_\rho\right) \end{bmatrix}. \tag{C.12}$$

In our case, since we have axisymmetry, all the ϕ derivatives are zero.

The second flat derivative of a scalar

$$\begin{aligned} \hat{\mathcal{D}}_A \hat{\mathcal{D}}_B f = \partial_A \partial_B f - \hat{\omega}^C_{AB} \partial_C f = e^i_A \partial_i \left(e^j_B \partial_j f \right) - \hat{\omega}^C_{AB} e^k_C \partial_k f = \\ = e^i_A e^j_B \partial_i \partial_j f + \left(e^i_A \partial_i e^j_B \right) \partial_j f - \hat{\omega}^C_{AB} e^k_C \partial_k f. \end{aligned} \tag{C.13}$$

In cylindrical coordinates we can make the appropriate substitutions to find

$$\hat{\mathcal{D}}_A \hat{\mathcal{D}}_B f \doteq \begin{bmatrix} \partial^2_\rho f & \partial_\rho \partial_z f & \rho^{-1}\partial_\rho \partial_\phi f - \rho^{-2}\partial_\phi f \\ \partial_z \partial_\rho f & \partial^2_z f & \partial_z \partial_\phi f \\ \rho^{-1}\partial_\phi \partial_\rho f - \rho^{-2}\partial_\phi f & \partial_\phi \partial_z f & \rho^{-2}\partial^2_\phi f + \rho^{-1}\partial_\rho f \end{bmatrix}. \tag{C.14}$$

In particular, we find that if we take the trace with the flat metric δ^{AB}, we get the formula for the Laplacian in section 2.1. This is because the traditional calculus operators are written in orthonormal coordinates.

Let's look at the second derivative of a covector field,

$$\begin{aligned} \hat{\mathcal{D}}_A \hat{\mathcal{D}}_B X_C = \hat{\mathcal{D}}_A \left(\partial_B X_C - \hat{\omega}^D_{BC} X_D \right) = \\ = e^i_A e^j_B \partial_i \partial_j X_C - (\hat{\omega}^D_{BC} e^i_A + \hat{\omega}^D_{AC} e^i_B) \partial_i X_D - \left(\hat{\omega}^D_{AB} e^i_D - e^j_A \partial_j e^i_B \right) \partial_i X_C + \\ + \left(\hat{\omega}^E_{AB} \hat{\omega}^D_{EC} + \hat{\omega}^E_{AC} \hat{\omega}^D_{BE} - \partial_A \hat{\omega}^D_{BC} \right) X_D. \end{aligned} \tag{C.15}$$

The easiest contribution is the derivative of the z component, since many terms will vanish

$$\hat{\mathcal{D}}_A \hat{\mathcal{D}}_B X_Z = e^i_A e^j_B (\partial_i \partial_j - \hat{\Gamma}^k_{ij} \partial_k) X_Z. \tag{C.16}$$

In fact that contribution will be in the three components, so they will all have the same general form as the z component plus some extra factors. The z component is

$$\hat{\mathcal{D}}_A \hat{\mathcal{D}}_B X_z \doteq \begin{bmatrix} \partial^2_\rho X_z & \partial_\rho \partial_z X_z & \rho^{-1}\partial_\rho \partial_\phi X_z - \rho^{-2}\partial_\phi X_Z \\ sym & \partial^2_z X_z & \rho^{-1}\partial_z \partial_\phi X_z \\ sym & sym & \rho^{-2}\partial^2_\phi X_z + \rho^{-1}\partial_\rho X_z \end{bmatrix}. \tag{C.17}$$

The ρ component is

$$\hat{\mathcal{D}}_A\hat{\mathcal{D}}_B X_\rho \doteq \begin{bmatrix} \partial_\rho^2 X_\rho & \partial_\rho\partial_z X_\rho & \rho^{-1}\partial_\rho\partial_\phi X_\rho - \rho^{-2}\partial_\phi X_\rho - \rho^{-1}\partial_\rho X_\Phi + \rho^{-2} X_\Phi \\ sym & \partial_z^2 X_\rho & \rho^{-1}\partial_z\partial_\phi X_\rho - \rho^{-1}\partial_z X_\Phi \\ sym & sym & \rho^{-2}\partial_\phi^2 X_\rho + \rho^{-1}\partial_\rho X_\rho - 2\rho^{-2}\partial_\phi X_\Phi - \rho^{-2} X_\rho \end{bmatrix}. \tag{C.18}$$

And finally the Φ component

$$\hat{\mathcal{D}}_A\hat{\mathcal{D}}_B X_\Phi \doteq \begin{bmatrix} \partial_\rho^2 X_\Phi & \partial_\rho\partial_z X_\Phi & \rho^{-1}\partial_\rho\partial_\phi X_\Phi - \rho^{-2}\partial_\phi X_\Phi + \rho^{-1}\partial_\rho X_\rho - \rho^{-2} X_\rho \\ sym & \partial_z^2 X_\Phi & \rho^{-1}\partial_z\partial_\phi X_\Phi + \rho^{-1}\partial_z X_\rho \\ sym & sym & \rho^{-2}\partial_\phi^2 X_\Phi + \rho^{-1}\partial_\rho X_\Phi + 2\rho^{-2}\partial_\phi X_\rho - \rho^{-2} X_\Phi \end{bmatrix}. \tag{C.19}$$

In axisymmetry

$$\hat{\mathcal{D}}_A\hat{\mathcal{D}}_B X_\rho \doteq \begin{bmatrix} \partial_\rho^2 X_\rho & \partial_\rho\partial_z X_\rho & -\rho^{-1}\partial_\rho X_\Phi + \rho^{-2} X_\Phi \\ sym & \partial_z^2 X_\rho & -\rho^{-1}\partial_z X_\Phi \\ sym & sym & +\rho^{-1}\partial_\rho X_\rho - \rho^{-2} X_\rho \end{bmatrix}, \tag{C.20}$$

$$\hat{\mathcal{D}}_A\hat{\mathcal{D}}_B X_\Phi \doteq \begin{bmatrix} \partial_\rho^2 X_\Phi & \partial_\rho\partial_z X_\Phi & \rho^{-1}\partial_\rho X_\rho - \rho^{-2} X_\rho \\ sym & \partial_z^2 X_\Phi & +\rho^{-1}\partial_z X_\rho \\ sym & sym & \rho^{-1}\partial_\rho X_\Phi - \rho^{-2} X_\Phi \end{bmatrix}. \tag{C.21}$$

In the formulas there are terms of the type

$$\rho^{-1}\partial_\rho X_\rho - \rho^{-2} X_\rho, \tag{C.22}$$

which seem to require the components to be proportional to ρ^2 in order to ensure regularity of the second derivative, but we can see that those terms are really

$$\rho^{-1}\partial_\rho X_\rho - \rho^{-2} X_\rho = \partial_\rho (X_\rho/\rho), \tag{C.23}$$

and from our discussion in section 3.2 we know that X_ρ/ρ has to be smooth if $\boldsymbol{X}$ is a smooth vector field, so those terms are well-behaved.

Another term that looks like it could cause problems is $\rho^{-1}\partial_\rho X_z$, but because of axisymmetry X_z has to be even on ρ, and smooth if it describes a smooth field. If we furthermore assume that it is analytic around the origin, we can write as a Taylor series

$$X_z(\rho) = \sum_{k=0}^{\infty} \alpha_{2k}\rho^{2k} = \sum_{n=0}^{\infty} \tilde{\alpha}_n (\rho^2)^n = \tilde{f}(\rho^2), \tag{C.24}$$

where $\tilde{f}$ is an analytic function (therefore also smooth). Then we can see that term will be regular

$$\rho^{-1}\partial_\rho X_z = 2\tilde{f}'(\rho^2). \tag{C.25}$$

We can see how the formulas for the derivatives will look and how the derivatives are smooth despite the fact that we are in curvilinear coordinates. An important point to mention is that these formulas were calculated in order to do tests of the code, but in the simulation these formulas are not written like this. The derivatives are written in its most general form with terms of the spin-connection, in such a way that it is possible to change coordinates just by changing $\hat{\omega}^{A}_{BC}$

Appendix D

Tests of the Ricci tensor

The Ricci tensor is one of the most complicated single terms, so it is a good example of how terms are tested. In the code, the Ricci tensor is calculated with the following formula

$$\begin{gathered}\bar{R}_{AB} = -\frac{1}{2}\bar{\gamma}^{CD}\hat{\mathcal{D}}_C\hat{\mathcal{D}}_D\bar{\gamma}_{AB} + \hat{\mathcal{D}}_{(A}\bar{\Lambda}_{B)} - \left[\hat{\mathcal{D}}_{(A}\bar{\gamma}_{B)C}\right]\bar{\Lambda}^C + \\ \bar{\gamma}^{CD}\bar{\gamma}^{EF}\left[\Delta\Gamma_{FCD}\Delta\Gamma_{(AB)E} + 2\Delta\Gamma_{FC(A}\Delta\Gamma_{B)ED} + \Delta\Gamma_{FAC}\Delta\Gamma_{EBD}\right].\end{gathered} \tag{D.1}$$

For all the terms to be tested, we just need $\bar{\gamma}_{AB} \neq \hat{\gamma}_{AB}$. Remember that in the code I am using an orthonormal frame, denoted in the components by a capital Latin letter, because of the regularity of the components. I am going to perform three different tests, the first of which using the following metric:

$$\bar{\gamma}_{ij} \doteq \begin{bmatrix} f(\rho) & 0 & 0 \\ 0 & g(z) & 0 \\ 0 & 0 & \rho^2 \end{bmatrix}. \tag{D.2}$$

The corresponding Ricci tensor is

$$R_{AB} \doteq \begin{bmatrix} \frac{f'(\rho)}{2\rho f(\rho)} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{f'(\rho)}{2\rho f(\rho)^2} \end{bmatrix}. \tag{D.3}$$

For a specific example, let's make a choice for the functions informed by parity conditions,

$$f(\rho) = 1 + \rho^2 e^{-\rho^2} \Rightarrow f'(\rho) = 1 + 2\rho(1 - 2\rho^2)e^{-\rho^2}; \quad g(z) = 1 + e^{-z^2}. \tag{D.4}$$

which that choice the Ricci tensor is

$$R_{AB} \doteq \begin{bmatrix} \frac{(1-\rho^2)e^{-\rho^2}}{1+\rho^2 e^{-\rho^2}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{(1-\rho^2)e^{-\rho^2}}{(1+\rho^2 e^{-\rho^2})^2} \end{bmatrix}. \tag{D.5}$$

The connection vector is non-zero, as we want for a good test of all the terms

$$\bar{\Lambda}_P = \frac{\rho(1-\rho^2)}{e^{\rho^2}+\rho^2} + \rho e^{-\rho^2}; \quad \bar{\Lambda}_Z = -\frac{z}{1-e^{z^2}}; \quad \bar{\Lambda}_\Phi = 0. \tag{D.6}$$

Now I do a convergence test. The diagonal tests are expected to converge to their values, for non-diagonal terms we have derivatives of terms that do not depend on the given variable, which result in terms like

$$\partial_z f(\rho) = \frac{f(\rho)-f(\rho)}{\Delta z} = \frac{\varepsilon}{\Delta z} \text{ numerically.} \tag{D.7}$$

Although we have a term that should be zero analytically, in the code there is imprecise cancellation of floating points, that result in a term of the order of machine precision, and we will be able to see this effect in the plots. In our case, we have a second derivative, so the slope of the increase of error with resolution is going to be two. This error is nothing to worry about, it is always present when taking numerical derivatives, but usually it is masked by the truncation error in the Taylor expansion. It is several orders of magnitude smaller than any other error in the problem, so it is negligible. We see the results of the convergence test in Fig. D.1.

Now let's do another test for the Schwarzschild spacetime in isotropic coordinates. In these coordinates we have the metric

$$\gamma_{ij} = \psi^4 \hat{\gamma}_{ij}; \quad \psi = \left(1+\frac{M}{2r}\right). \tag{D.8}$$

And therefore the Ricci tensor in cylindrical coordinates is

$$R_{ij} \doteq \lambda(r) \begin{bmatrix} 1-3\rho^2/r^2 & -3z\rho/r^2 & 0 \\ -3z\rho/r^2 & 1-3z^2/r^2 & 0 \\ 0 & 0 & \rho^2 \end{bmatrix}, \tag{D.9}$$

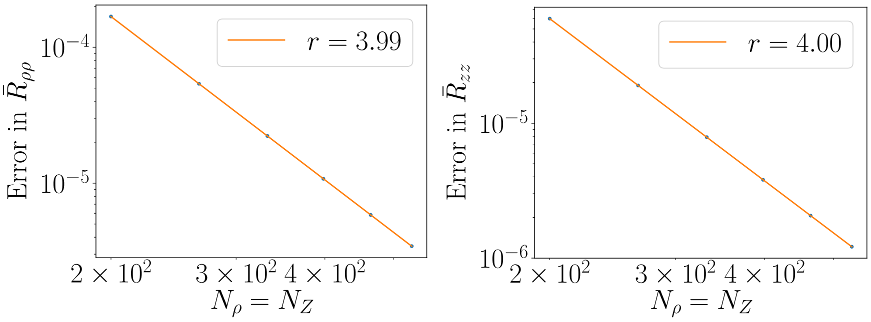
with the common factor

$$\lambda(r) = \frac{4rM}{(2r^2+Mr)^2}. \tag{D.10}$$

When calculated in code, we get the results in Fig. D.2. Note that this metric is irregular at the origin, so one stencil away from it we are going to have uncontrolled error. For a static test, is enough to avoid that area in the calculation of the norm.

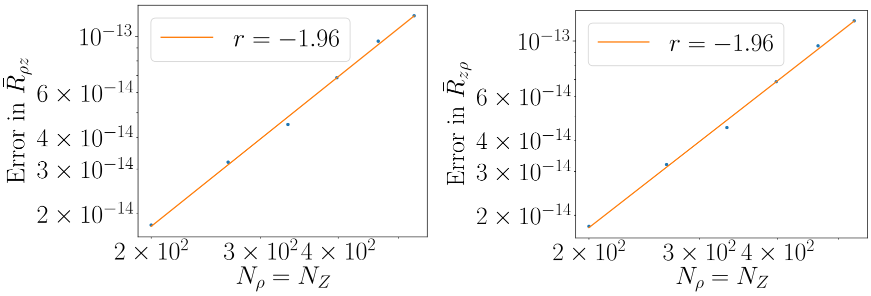
Now I would like to test and use the code also in spherical coordinates. For that I have to supply the new connection vielbein and its derivatives to the coordinates object. The analytic solution for the Ricci tensor is

$$R_{ij} \doteq \lambda(r) diag(-2, r^2, r^2\sin^2\theta) \Rightarrow R_{AB} \doteq \lambda(r) diag(-2,1,1). \tag{D.11}$$



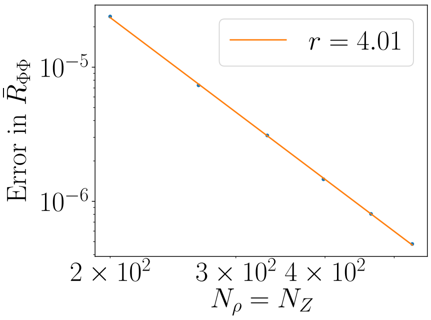
(a) Convergence of R_{PP}.

(b) Convergence of R_{ZZ}.



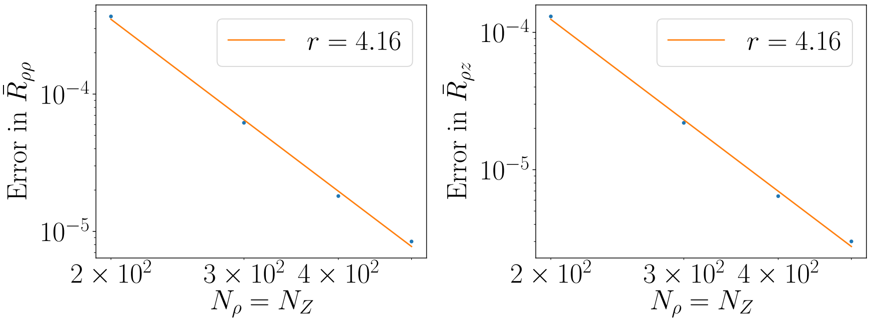
(c) Convergence of R_{PZ}.

(d) Convergence of R_{ZP}.



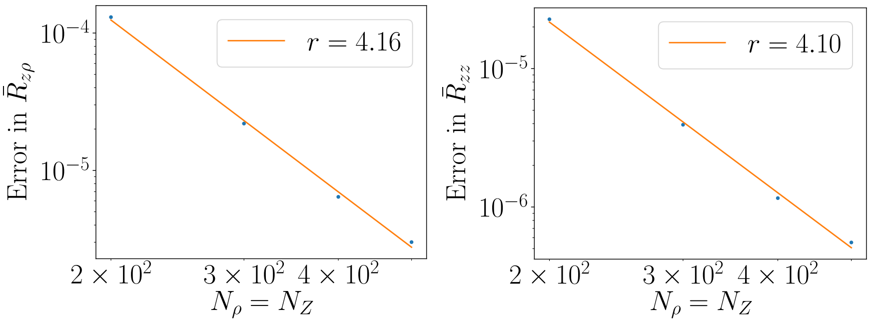
(e) Convergence of $R_{\Phi\Phi}$.

Figure D.1: Convergence of the Ricci tensor for the first test.



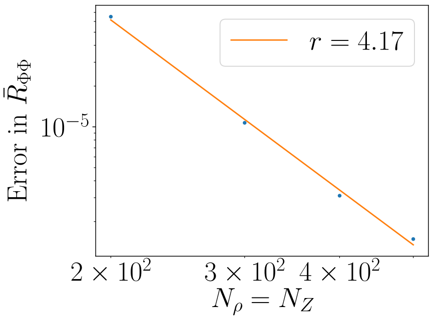
(a) Convergence of R_{PP}.

(b) Convergence of R_{PZ}.



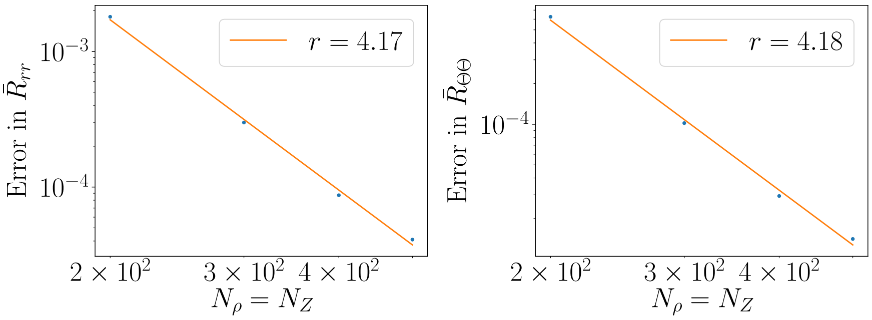
(c) Convergence of R_{ZP}.

(d) Convergence of R_{ZZ}.



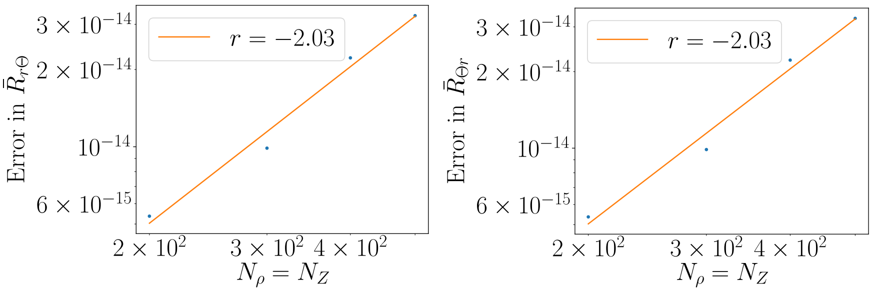
(e) Convergence of $R_{\Phi\Phi}$.

Figure D.2: Convergence of the Ricci tensor for a Schwarzschild spacetime. The region next to the origin is not included in the norm, since the fields are not differentiable there.



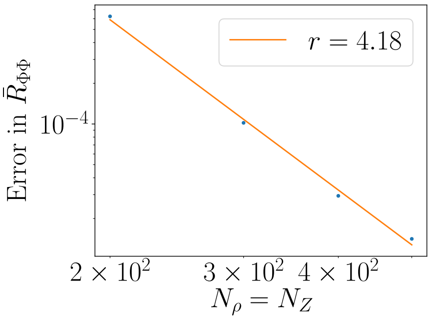
(a) Convergence of R_{rr}.

(b) Convergence of $R_{\Theta\Theta}$.



(c) Convergence of $R_{r\Theta}$.

(d) Convergence of $R_{\Theta r}$.



(e) Convergence of $R_{\Phi\Phi}$.

Figure D.3: Convergence of the Ricci tensor calculated in spherical coordinates.

We can find the results of the convergence test in figure D.3. All of the terms present the expected convergence.

These tests give us high confidence of the Ricci tensor term being correctly implemented, and similar tests are done for all mathematical operators.

Appendix E

Calculation of the Teukolsky waves

In this section:
Linearized equations — tensor harmonics — specific solution

In the literature, I haven't found a source that compiles the ideas of how the formulas for the Teukolsky waves are derived, so in this appendix I will present a schematic view of how that calculation can be done.

Teukolsky waves are a solution to the following equation:

$$\Box h_{ij} = 0. \tag{E.1}$$

This is a form of the linearized Einstein's equations. How these come about is described in any basic text in General Relativity [30]. Basically, this describes how the perturbations from Minkowski space-time will look if the deviation is not really big. But since it is just a linear equation, the solutions are going to be very similar to the scalar wave that I discussed at length in chapter 2, the extra complication coming from the fact that now we are dealing with a tensor quantity.

As for the scalar case, we will look at solutions in spherical coordinates, for the same reasons as before: we want waves that don't have an infinite extent, since those are the ones that model radiation by real localized sources. In General Relativity, the equations and the geometry are two faces of the same coin, so what that means in geometric terms is that the space-time is asymptotically flat. Asymptotic flatness is the condition that when we go really far away our space-time looks like Minkowski space-time, and in particular this means that all perturbations die out with increasing distance, as we want for our waves.

Now, in this case, we cannot look at spherically symmetric solutions because of the well-known result that there are no gravitational waves in spherical symmetry. The solution that we will use instead is a Teukolsky wave [40]. These are quadrupolar solutions, which correspond to the main mode of a radiating system. Here I will present a sketch of how these waves are derived, following the methods of [47], and the expressions presented in the appendix of [48], in order to get a general understanding of how they work, and then present the solutions both in spherical and cylindrical coordinates.

The idea is very similar to the one behind the solutions to the scalar wave, but this time the solutions are separated into tensor spherical harmonics

$$h_{ab}^{(jlm)}(t,r,\theta,\phi) = \varphi_{(l)}(r,t)T_{ab}^{(jlm)}(\theta,\phi). \tag{E.2}$$

The tensor spherical harmonics are defined by the following eigenfunction equations:

$$\begin{aligned} J^2 Y_{ab}^{(jlm)} &= j(j+1)Y_{ab}^{(jlm)}, \\ L^2 Y_{ab}^{(jlm)} &= l(l+1)Y_{ab}^{(jlm)}, \\ J_z Y_{ab}^{(jlm)} &= mY_{ab}^{(jlm)}, \end{aligned} \tag{E.3}$$

where J^2, L^2 and J_z are differential operators, that include spin (matrix) terms. The term L^2,

$$L^2 T_{ab}^{(jlm)} = -\left[\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\sin\theta\frac{\partial}{\partial\theta} + \frac{1}{\sin^2\theta}\frac{\partial^2}{\partial\phi^2}\right]T_{ab}^{(jlm)} = l(l+1)T_{ab}^{(jlm)}, \tag{E.4}$$

is straightforward, as it will help us separate the wave equation in the radial and the angular part.

The radial equation reduces to

$$W_l\varphi_{(l)} = r^{-1}\frac{\partial^2}{\partial r^2}(r\varphi) - l(l+1)r^{-2}\varphi - \partial_t\varphi = 0. \tag{E.5}$$

Defining the following operators

$$D_l^+ := \partial_r - l/r, \quad D_l^- := \partial_r + (l+1)/r, \tag{E.6}$$

we find

$$W_{l+1}\left(D_l^+\varphi_{(l)}\right) = 0, \quad W_{l-1}\left(D_l^-\varphi_{(l)}\right) = 0. \tag{E.7}$$

So we have raising and lowering operators, and therefore we can calculate the solution for $l = 0$, which is easy, and raise it to the order that we want. That way we have

$$\begin{aligned} \varphi_{(0)}(r,t) &= \frac{f(t\pm r)}{r}, \\ \varphi_{(1)}(r,t) &= \frac{f'(t\pm r)}{r} \mp \frac{f(t\pm r)}{r^2}, \\ \varphi_{(2)}(r,t) &= \frac{f''(t\pm r)}{r} \mp \frac{3f'(t\pm r)}{r^2} + \frac{3f(t\pm r)}{r^3}, \end{aligned} \tag{E.8}$$

where the different signs imply an outgoing or ingoing wave, and the general solution is a superposition of both. Of course, we can also solve the equation (E.7) in terms of spherical Bessel functions, but for that we have to Fourier transform and we lose the explicit time dependence.

The Teukolsky wave corresponds to the quadrupole mode ($j = 2$) and thus is a superposition of the modes $L = 0, 1, 2, 3, 4$. Which superposition? Well, we have to take into account two things:

- Gravitational waves have two independent polarizations, so we can divide the solutions into the polarization types.
- The solution has to be in Lorentz gauge, thus its divergence must be zero.

How to separate the polarizations is up to us, and the common choice is to separate the polar modes (that transform under parity as $(-1)^l$) and the axial modes (that pick a sign $(-1)^{l+1}$). In our case, the modes with $l = 0, 2, 4$ are polar, and the modes with $l = 1, 3$ are axial.

Thus, the polar wave will be of the form

$$h_{ab}^{(pol)} = a_0\varphi_{(0)}T_{ab}^{(40m)}(\Omega) + a_2\varphi_{(0)}T_{ab}^{(42m)}(\Omega) + a_4\varphi_{(0)}T_{ab}^{(44m)}(\Omega). \tag{E.9}$$

In order to find the specific constants of the linear combination, we have to impose the Lorentz gauge. In [48] they give the formula for the divergence of each of the solution modes. For our purposes it is enough to know that is of the form

$$\partial_a\left(\varphi_{(l)}T^{ab}_{(jlm)}\right) = \alpha_{(jlm)}\varphi_{(l-1)}T^{ab}_{(jl-1m)} + \beta_{(jlm)}\varphi_{(l+1)}T^{ab}_{(jl+1m)}, \tag{E.10}$$

where α and β are complicated constants with 6-j symbols and so on, but nevertheless constants that depend only on l, j and m. Thus, the condition that the polar wave is divergenceless corresponds to the following equations for the constants:

$$\begin{aligned} a_0\beta_{40m} + a_2\alpha_{42m} &= 0, \\ a_2\beta 42m + a_4\alpha_{44m} &= 0. \end{aligned} \tag{E.11}$$

We have three constants and two conditions because the overall scale corresponds to the polarization. The axial wave will be constructed in exactly the same way.

As a recap, the Teukolsky waves are calculated in the following way:

1. Find the relevant spherical tensor harmonics by solving the eigenvalues equations.

2. Raise the radial $l = 0$ solution to the highest multipole order involved in the desired solution (in the Teukolsky wave, that is $l = 4$).
3. Divide the solution in polar and axial waves.
4. Solve the Lorentz gauge equation to find the specific combination of constants in each polarization.

The detailed calculations are outside the scope of this thesis, but I believe a knowledge of the steps required for the derivation is helpful to understand better the solutions we are dealing with.

In the text the solutions are given, but in order to perform a comparison in the code, we need the specific form of the functions for all times. This can get very complicated soon, as exemplified by the simplest of the three radial functions,

$$\begin{aligned} A(t,r) = \frac{3\mathcal{A}}{\lambda^4 r^5} e^{-\frac{(r+t)^2}{\lambda^2}} \Big\{ e^{\frac{4rt}{\lambda^2}} \left[-4r^2(r-t)^3 + 6\lambda^2 rt(r-t) + 3\lambda^4 t \right] - \\ 4r^2(r+t)^3 - 6\lambda^2 rt(r+t) - 3\lambda^4 t \Big\} . \end{aligned} \tag{E.12}$$

By the time we get to C, each formula is several lines long, therefore I will introduce some new variables and functions in order to simplify the formulas. Let's consider the new variables

$$\lambda u = r - t; \quad \lambda v = r + t; \quad \lambda w = r. \tag{E.13}$$

The last variable is redundant, but it will simplify the formulas. With these variables, the functions for all times are

$$\begin{aligned} A(r,t) &:= \frac{3\mathcal{A}}{w^5} \left[a(u,w) + a(v,w) \right], \\ B(r,t) &:= \frac{2\mathcal{A}}{w^5} \left[b(u,w) + b(v,w) \right], \\ C(r,t) &:= \frac{-\mathcal{A}}{4w^5} \left[c(u,w) + c(v,w) \right], \end{aligned} \tag{E.14}$$

with the auxiliary functions

$$\begin{aligned} a(u,w) &= -e^{-u^2}(4u^3w^2 + 6u^2w - 6uw^2 + 3u - 3w) \\ b(u,w) &= e^{-u^2}(4u^4w^3 + 6u^3w^2 - 12u^2w^3 + 6u^2w - 9uw^2 + 3u + 3w^3 - 3w) \\ c(u,w) &= e^{-u^2}(16u^5w^4 + 16u^4w^3 - 80u^3w^4 + 36u^3w^2 - 48u^2w^3 + \\ &\quad 42u^2w + 60uw^4 - 54uw^2 + 21u + 12w^3 - 21w). \end{aligned} \tag{E.15}$$

This is the way I implemented the analytical solutions in the code, since I found it to be the simplest.

Bibliography

[1] B. P. Abbott et al. Observation of gravitational waves from a binary black hole merger. *Phys. Rev. Lett.*, 116(6):061102, 2016.

[2] Ivan Markin, Anna Neuweiler, Adrian Abac, Swami Vivekanandji Chaurasia, Maximiliano Ujevic, Mattia Bulla, and Tim Dietrich. General-Relativistic Hydrodynamics Simulation of a Neutron Star - Sub-Solar-Mass Black Hole Merger. 4 2023.

[3] B. P. Abbott et al. GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs. *Phys. Rev. X*, 9(3):031040, 2019.

[4] Stuart L. Shapiro and Saul A. Teukolsky. Formation of naked singularities: The violation of cosmic censorship. *Phys. Rev. Lett.*, 66:994–997, Feb 1991.

[5] Luis Lehner and Frans Pretorius. Black strings, low viscosity fluids, and violation of cosmic censorship. *Physical Review Letters*, 105(10), sep 2010.

[6] Roberto Emparan and Robert C Myers. Instability of ultra-spinning black holes. *Journal of High Energy Physics*, 2003(09):025–025, sep 2003.

[7] Piotr Bizoń and Andrzej Rostworowski. Weakly turbulent instability of anti–de sitter spacetime. *Physical Review Letters*, 107(3), jul 2011.

[8] William E. East and Frans Pretorius. Ultrarelativistic black hole formation. *Physical Review Letters*, 110(10), mar 2013.

[9] Matthew W. Choptuik and Frans Pretorius. Ultrarelativistic particle collisions. *Physical Review Letters*, 104(11), mar 2010.

[10] Luciano Rezzolla and Kentaro Takami. Black-hole production from ultrarelativistic collisions. *Classical and Quantum Gravity*, 30(1):012001, dec 2012.

[11] BingKan Xue, David Garfinkle, Frans Pretorius, and Paul J. Steinhardt. Nonperturbative analysis of the evolution of cosmological perturbations through a nonsingular bounce. *Physical Review D*, 88(8), oct 2013.

[12] Xinghai Zhao and Grant J. Mathews. Effects of structure formation on the expansion rate of the universe: An estimate from numerical simulations. *Physical Review D*, 83(2), jan 2011.

[13] David Garfinkle, Anna Ijjas, and Paul J. Steinhardt. Initial conditions problem in cosmological inflation revisited. 4 2023.

[14] Vitor Cardoso, Leonardo Gualtieri, Carlos Herdeiro, and Ulrich Sperhake. Exploring New Physics Frontiers Through Numerical Relativity. *Living Rev. Relativity*, 18:1, 2015.

[15] Richard Brito, Vitor Cardoso, and Paolo Pani. Superradiance: New Frontiers in Black Hole Physics. *Lect. Notes Phys.*, 906:pp.1–237, 2015.

[16] Ulrich Sperhake, Vitor Cardoso, Frans Pretorius, Emanuele Berti, and Jose A. Gonzalez. The High-energy collision of two black holes. *Phys. Rev. Lett.*, 101:161101, 2008.

[17] William E. East. Vortex String Formation in Black Hole Superradiance of a Dark Photon with the Higgs Mechanism. *Phys. Rev. Lett.*, 129(14):141103, 2022.

[18] Frank Löffler et al. The einstein toolkit: a community computational infrastructure for relativistic astrophysics. *Classical and Quantum Gravity*, 29(11):115001, may 2012.

[19] Tomas Andrade et al. GRChombo: An adaptable numerical relativity code for fundamental physics. *Journal of Open Source Software*, 6(68):3703, dec 2021.

[20] Ian Ruchlin, Zachariah B. Etienne, and Thomas W. Baumgarte. SENR/NRPy+: Numerical relativity in singular curvilinear coordinate systems. *Phys. Rev. D*, 97:064036, Mar 2018.

[21] L.C. Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 1998.

[22] Heinz-Otto Kreiss and Godela Scherer. Method of lines for hyperbolic differential equations. *SIAM Journal on Numerical Analysis*, 29:640–646, 1992.

[23] Ieee standard for floating-point arithmetic. *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, page 52, 2019.

[24] T.M. Apostol. *Calculus, Volume 1.* Wiley, 1991.

[25] Thomas W. Baumgarte and Stuart L. Shapiro. Numerical integration of einstein's field equations. *Phys. Rev. D*, 59:024007, Dec 1998.

[26] Masaru Shibata and Takashi Nakamura. Evolution of three-dimensional gravitational waves: Harmonic slicing case. *Phys. Rev. D*, 52:5428–5444, Nov 1995.

[27] J David Brown. BSSN in spherical symmetry. *Classical and Quantum Gravity*, 25(20):205004, sep 2008.

[28] T.W. Baumgarte and S.L. Shapiro. *Numerical Relativity: Solving Einstein's Equations on the Computer*. Cambridge University Press, 2010.

[29] Thomas W. Baumgarte, Pedro J. Montero, Isabel Cordero-Carrión, and Ewald Müller. Numerical relativity in spherical polar coordinates: Evolution calculations with the bssn formulation. *Phys. Rev. D*, 87:044026, Feb 2013.

[30] S.M. Carroll. *Spacetime and Geometry.* Cambridge University Press, 2019.

[31] Larry Smarr, Andrej Čadež, Bryce DeWitt, and Kenneth Eppley. Collision of two black holes: Theoretical framework. *Phys. Rev. D*, 14:2443–2452, Nov 1976.

[32] Susan G Hahn and Richard W Lindquist. The two-body problem in geometrodynamics. *Annals of Physics*, 29(2):304–331, 1964.

[33] M. Campanelli, C. O. Lousto, P. Marronetti, and Y. Zlochower. Accurate evolutions of orbiting black-hole binaries without excision. *Phys. Rev. Lett.*, 96:111101, Mar 2006.

[34] C.W. Misner, K.S. Thorne, J.A. Wheeler, and D.I. Kaiser. *Gravitation.* Princeton University Press, 2017.

[35] Stanislav Hledík, Zdeněk Stuchlík, and Alois Cipko. Visualizing spacetimes via embedding diagrams. In *The Eleventh Marcel Grossmann Meeting.* World Scientific Publishing Company, sep 2008.

[36] Kenneth A. Dennison and Thomas W. Baumgarte. A simple family of analytical trumpet slices of the schwarzschild spacetime. *Classical and Quantum Gravity*, 31, 2014.

[37] Kenneth A. Dennison, Thomas W. Baumgarte, and Pedro J. Montero. Trumpet slices in kerr spacetimes. *Phys. Rev. Lett.*, 113:261101, Dec 2014.

[38] Mark Hannam, Sascha Husa, Niall O Murchadha, Bernd Brügmann, José González, and U. Sperhake. Where do moving punctures go? *Journal of Physics: Conference Series*, 66, 01 2007.

[39] Thomas W. Baumgarte and Stephen G. Naculich. Analytical representation of a black hole puncture solution. *Phys. Rev. D*, 75:067502, Mar 2007.

[40] Saul A. Teukolsky. Linearized quadrupole waves in general relativity and the motion of test particles. *Phys. Rev. D*, 26:745–750, Aug 1982.

[41] M. Campanelli, C. O. Lousto, and Y. Zlochower. Last orbit of binary black holes. *Phys. Rev. D*, 73:061501, Mar 2006.

[42] Miguel Alcubierre and Martha Mendez. Formulations of the 3+1 evolution equations in curvilinear coordinates. *General Relativity and Gravitation - GEN RELATIV GRAVIT*, 43, 10 2010.

[43] Matthew W. Choptuik. Universality and scaling in gravitational collapse of a massless scalar field. *Phys. Rev. Lett.*, 70:9–12, 1993.

[44] Carsten Gundlach. Critical phenomena in gravitational collapse. *Phys. Rept.*, 376:339–405, 2003.

[45] H. Kreiss, J. Oliger, Global Atmospheric Research Programme. Joint Organizing Committee, International Council of Scientific Unions, and World Meteorological Organization. *Methods for the Approximate Solution of Time Dependent Problems*. GARP publications series. International Council of Scientific Unions, World Meteorological Organization, 1973.

[46] Gregory B. Cook. Initial data for numerical relativity. *Living Reviews in Relativity*, 3(1), nov 2000.

[47] Jon Mathews. Gravitational multipole radiation. *Journal of the Society for Industrial and Applied Mathematics*, 10(4):768–780, 1962.

[48] William L. Burke. Gravitational radiation damping of slowly moving systems calculated using matched asymptotic expansions. *Journal of Mathematical Physics*, 12(3):401–418, 1971.