Università degli Studi di Parma

Tesi di Laurea Magistrale in Fisica

Numerical Simulations of the Merger of Neutron Stars

Michele Pasquali

relatore
Prof. Roberto De Pietri

A.A. 2015/2016

Contents

Introduction

The recent detection [1] of gravitational waves (GW) from a binary black hole merger by Advanced Ligo [2], has opened a new window for investigation of astrophysical compact object. The new generation GW detectors Advanced LIGO and Advanced Virgo [3] are also expected to reveal gravitational signal from Binary Neutron Stars (BNS) coalescence and merger, once their sensitivity at higher frequencies will increase. At the current sensitivity, the rate of BNS signal is predicted to be in the interval (0.2-200) per year [4], making them the next target for GW detection.
Numerical relativity is the main instrument to study the dynamics of the merger of compact objects and this has led to the development of community driven public software like the Einstein Toolkit [5–7] and the LORENE [8, 9] library that allow to simulate such systems and in particular BNS coalescence and merger [10]. Simulations of BNS mergers, where numerical relativity is almost the only available tool, allow to study the possible gravitational waveforms that will be discovered by the present detector. For this reason, there is a considerable effort (using either public or private codes) in analyzing the effect of the EOS used to describe the matter on the gravitational-wave signal and incorporating in the codes more microphysical elements, such as finite temperature nuclear EOSs with neutrino emission [11–14] and magnetic fields [14–19], to accurately simulate the physiscs of the post-merger remnant.
In this work, the attention is focused on the merger process that results in a quite prompt collapse to Black Hole after the merger. We considered three equal mass models of total gravitational mass $1.5M_{\odot}$, $1.55M_{\odot}$ and $1.6M_{\odot}$, respectively and described by a simple polytropic equation of state $P = K\rho^{\Gamma}$ with $\Gamma = 3.00$ and an additional thermal component given by $\Gamma_{th} = 1.8$. These simulations have been done using only publicly available open source software: the Einstein Toolkit code, deployed for the dynamical evolution of the system, and the LORENE code, which provides routines for the generation of initial models based on spectral methods. We used the BSSN-NOK scheme supplemented by a fifth-order WENO reconstruction method. The

study was focused on the properties of the accretion disk that surround the final black hole, its rotational profile as well as its mass.
In detail, in the first Chapter of this thesis we describe the basics of general relativity and introduce the theoretical tools needed. In Chapter 2 we make a brief description of the *3+1 formalism* that gives a more intuitive representation for General Relativity, explaining the theoretical tools the forms Numerical Relativity. Chapter 3 is devoted to the codes we used to run our simulations, the structure of the Einstein Toolkit, used for the dynamical evolution, and LORENE, for the generation of the initial models. The results of the simulations, including the analysis of the GW spectrum obtained, are shown in Chapter 4.

Conventions used

The signature of the spacetime metric we use is $[-1, +1, +1, +1]$.
Greek indices, such as α, β, μ, ν, refer to four-dimensional spacetime and go from 0 to 3, while latin indices go from 1 to 3. use the summation convention, in which identical upper and lower indices are implicitly summed over all their possible values.

1. Introduction to General Relativity

The basic framework of the theory of general relativity arises from considering that we cannot in principle construct inertial observers in the sense of special relativity and measure the gravitational force. Therefore, the hypothesis: *The spacetime metric is not flat, as was assumed in special relativity. The world lines of freely falling bodies in a gravitational field are simply the geodesics of the curved spacetime metric.* As a result we have no meaningful way of describing gravity as a force field; rather, we are forced to view gravity as an aspect of spacetime structure.

1.1 The Einstein Equations

The laws of physics in general relativity are governed by two basic principles:

1. the principle of general covariance which states that the metric $g_{\mu\nu}$ and quantities derivable from it are the only spacetime quantities that can appear in the equations of physics;

2. the requirement that equations must reduce to equations satisfied in special relativity in the case where $g_{\mu\nu}$ is flat.

Therefore, we would like to find an equation which supersedes the Poisson equation for the Newtonian potential:

$$\nabla^2\Phi = 4\pi G\rho \,, \tag{1.1}$$

where $\nabla^2 = \delta^{ij}\partial_i\partial_j$ is the Laplacian in space, Φ the Newotonian potential[1] and ρ the mass density. On the left-hand side we have a second-order differential operator acting on the gravitational potential, and on the right-hand side a measure of the mass distribution. A relativistic generalization should

[1] one can find that the form of the Newtonian potential is $\Phi = -\frac{GM}{r}$ and it's a solution of (1.1)

take the form of an equation between tensors, which is the stress-energy tensor $T_{\mu\nu}$ for mass density. The gravitational potential should get replaced by the metric tensor. For the metric in the Newtonian limit and $T_{00} = \rho$, we see that (1.1) looks like

$$\nabla^2 h_{00} = -8\pi G T_{00} \ , \tag{1.2}$$

but it is still not completely tensorial since the left-hand side does not generalize to a tensor. The substitution of ∇^2 with the d'Alambertian $\Box = \nabla^\mu \nabla_\mu$ on the metric $g_{\mu\nu}$ would lead to zero and it is not a good choice. The Riemann tensor $R^\rho{}_{\sigma\mu\nu}$, which is defined as

$$R^\rho{}_{\sigma\mu\nu} = \partial_\mu \Gamma^\rho_{\nu\sigma} - \partial_\nu \Gamma^\rho_{\mu\sigma} + \Gamma^\rho_{\mu\lambda}\Gamma^\lambda_{\nu\sigma} - \Gamma^\rho_{\nu\lambda}\Gamma^\lambda_{\mu\sigma} \ , \tag{1.3}$$

with $\Gamma^\rho_{\mu\nu}$ as the Christoffel connection defined as

$$\Gamma^\rho_{\mu\nu} = \frac{1}{2} g^{\sigma\rho} \left(\partial_\mu g_{\nu\rho} + \partial_\nu g_{\rho\mu} - \partial_\rho g_{\mu\nu} \right) \ , \tag{1.4}$$

is constructed from second derivatives of the metric and is not zero. It does not have the right number of indices but we can contract it to form the Ricci tensor $R_\mu\nu$

$$R_{\mu\nu} = R^\lambda{}_{\mu\lambda\nu} \ . \tag{1.5}$$

Therefore the gravitational field equations may be written as

$$R_{\mu\nu} = \kappa T_{\mu\nu} \ , \tag{1.6}$$

for some constant κ. This equation leads to a problem with energy conservation

$$\nabla^\mu T_{\mu\nu} = 0 \ , \tag{1.7}$$

which would imply

$$\nabla^\mu R_{\mu\nu} = 0 \ . \tag{1.8}$$

This is not true in an arbitrary geometry, but from the Bianchi identity

$$\nabla^\mu R_{\mu\nu} = \frac{1}{2} \nabla_\nu R \ . \tag{1.9}$$

Eq. (1.6) implies that $R = \kappa g^{\mu\nu} T_{\mu\nu} = \kappa T$, so

$$\nabla_\mu T = 0 \ . \tag{1.10}$$

However, this is highly implausible since $T = 0$ in vacuum and $T > 0$ in matter.

For this reason we define a symmetric tensor, the *Einstein tensor*, constructed from the Ricci tensor, which is automatically conserved:

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} \ . \tag{1.11}$$

$G_{\mu\nu}$ always obeys $\nabla^\mu G_{\mu\nu} = 0$ so (1.6) can be thought as

$$G_{\mu\nu} = \kappa T_{\mu\nu} \ . \tag{1.12}$$

Contracting both sides we obtain $R = \kappa T$ which allows to rewrite (1.12) as

$$R_{\mu\nu} = \kappa \left(T_{\mu\nu} - \frac{1}{2} T g_{\mu\nu} \right) \ . \tag{1.13}$$

It can be shown that (1.13) reduces to (1.2) with $\kappa = 8\pi G$ in the weak-field, time-independent, slowly-moving-particles limit.

Fixing $\kappa = 8\pi G$ as in the previous case, now we have the *Einstein's equations* for general relativity:

$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} = 8\pi G T_{\mu\nu} \ . \tag{1.14}$$

Einstein's equations may be thought of as second-order differential equations for the metric tensor field $g_{\mu\nu}$. There are ten independnt equations, but the Bianchi identity $\nabla^\mu G_{\mu\nu} = 0$ represents four constraints on $R_{\mu\nu}$, so there are only *six* truly independent equations in (1.14). The equations are also non-linear, so that two known solutions cannot be superposed to find a third.

1.2 Initial data problem in GR

Einstein's equations $G_{\mu\nu} = 8\pi G T_{\mu\nu}$ are of course covariant; they don't single out a preferred notion of "time" through which a state can evolve. We can pick a spacelike hypersurface (or "slice" as we see in Chapter 2) Σ, specify initial data on that hypersurface, and see if we can evolve uniquely from it to a hypersurface in the future.
Since the metric is the fundamental variable, our first guess is that we should consider the values $g_{\mu\nu}|_\Sigma$ of the metric on our hypersurface to be the "coordinates" and the time derivatives $\partial_t g_{\mu\nu}|_\Sigma$ (with respect to some specified time coordinate) to be the "momenta", which together specify the state. Considering the Bianchi identity

$$\nabla_\mu G^{\mu\nu} = 0 \ , \tag{1.15}$$

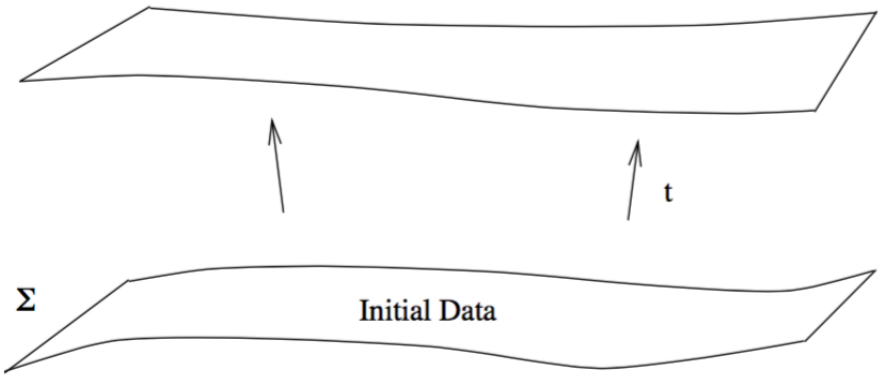

Figure 1.1: A scheme of the initial data problem in General Relativity.

we can rewrite the Einstein's equations as

$$\partial_0 G^{0\nu} = -\partial_i G^{i\nu} - \Gamma^{\mu}_{\mu\lambda} G^{\lambda\nu} - \Gamma^{\nu}_{\mu\lambda} G^{\mu\lambda} \, . \tag{1.16}$$

Looking at the the right-hand side of this equation, we see that there are no third-order time derivatives; therefore there cannot be any on the left-hand side. Although $G^{\mu\nu}$ as a whole involves second-order time derivatives of the metric, the specific components $G^{0\nu}$ do not. So

$$G^{0\nu} = 8\pi G T^{0\nu} \tag{1.17}$$

cannot be used to evolve initial data $(g_{\mu\nu}, \partial_t g_{\mu\nu})_\Sigma$. They serve as constraints on this initial data, while the remaining equations

$$G^{ij} = 8\pi G T^{ij} \tag{1.18}$$

are the dynamical evolution equations for the metric. These are only six equations for the ten unknown functions $g_{\mu\nu}(x^\sigma)$, so the solution will inevitably involve a fourfold ambiguity.
One way to cope with this problem is to simply "choose a gauge". We can do a the same thing we do in electromagnetism also in general relativity, by fixing our coordinate system. A popular choice is the *harmonic gauge* (also known as Lorentz gauge) in which

$$\Box x^\mu = 0 \, . \tag{1.19}$$

Here $\Box = \nabla^\mu \nabla_\mu$ is the covariant D'Alambertian. This condition is therefore simply

$$\begin{aligned} 0 &= \Box x^\mu \\ &= g^{\rho\sigma} \partial_\rho \partial_\sigma x^\mu - g^{\rho\sigma} \Gamma^{\lambda}_{\rho\sigma} \partial_\lambda x^\mu \\ &= -g^{\rho\sigma} \Gamma^{\lambda}_{\rho\sigma} \, . \end{aligned} \tag{1.20}$$

In flat space, Cartesian coordinates (in which $\Gamma^{\lambda}_{\rho\sigma} = 0$) are harmonic coordinates. It can be shown that this choice yields

$$\frac{\partial^2}{\partial t^2}\left(\sqrt{-g}g^{0\nu}\right) = -\frac{\partial}{\partial x^i}\left[\frac{\partial}{\partial t}(\sqrt{-g}g^{i\nu})\right] \ . \tag{1.21}$$

This is a second-order differential equation for the previously unconstrained metric components $g^{0\nu}$, in terms of the given initial data. Note that we still have some freedom remaining; our gauge condition (1.19) restricts how the coordinates stretch from our initial hypersurface Σ throughout spacetime, but we can still choose coordinates x^i on Σ however we like. That corresponds to the fact that making a coordinate transformation does not violate the harmonic gauge condition.
We therefore have a well-defined initial value problem for general relativity; a state is specified by the spacelike components of the metric and their time derivatives on a spacelike hypersurface Σ. For a more detailed description of this topic, see Chapter 2.

1.3 Gravitational radiation

We consider a situation where the field is weak, it can vary with time and there are no restrictions on the motion of test particles. The weakness of the gravitational field is expressed decomposing the metric into the flat Minkowski metric plus a small perturbation,

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \ , \qquad |h_{\mu\nu}| \ll 1 \ . \tag{1.22}$$

$\eta_{\mu\nu}$ takes its canonical form $\eta_{\mu\nu} = \mathrm{diag}(-1, +1, +1, +1)$. The assumption that $h_{\mu\nu}$ is small allows us to ignore anything that is higher than first order in this quantity, so we obtain

$$g^{\mu\nu} = \eta^{\mu\nu} - h^{\mu\nu} \ , \tag{1.23}$$

where $h^{\mu\nu} = \eta^{\mu\rho}\eta^{\nu\sigma}h_{\rho\sigma}$. We can raise and lower indices using $\eta^{\mu\nu}$ and $\eta_{\mu\nu}$ since the corrections would be of a higher order in the perturbation. In fact, we can think of the linearized version of general relativity as describing a theory of a symmetric tensor field $h_{\mu\nu}$ propagating on a flat background spacetime.
We want to find the equation of motion obeyed by the perturbations $h_{\mu\nu}$, which come by examining Einstein's equations to first order. So we rewrite (1.4) in terms of the perturbation

$$\Gamma^{\rho}_{\mu\nu} = \frac{1}{2}g^{\rho\lambda}\left(\partial_\mu g_{\nu\lambda} + \partial_\nu g_{\lambda\mu} - \partial_\lambda g_{\mu\nu}\right) \ . \tag{1.24}$$

Since the connection coefficients are first order quantities, the only contribution to the Rie- mann tensor will come from the derivatives of Γ, not Γ^2. Therefore, we obtain

$$\begin{aligned} R_{\mu\nu\rho\sigma} &= \eta_{\mu\lambda}\partial_\rho\Gamma^\lambda_{\nu\sigma} - \eta - \mu\lambda\partial_\sigma\Gamma^\lambda_{\nu\rho} \\ &= \frac{1}{2}\left(\partial_\rho\partial_\nu h_{\mu\sigma} + \partial_\sigma\partial_\mu h_{\nu\rho} - \partial_\sigma\partial_\nu h_{\mu\rho} - \partial_\rho\partial_\mu h_{\nu\sigma}\right) \ . \end{aligned} \tag{1.25}$$

Contracting over μ and ρ we obtain the Ricci tensor as

$$R_{\mu\nu} = \frac{1}{2}\left(\partial_\sigma\partial_\nu h^\sigma{}_\mu + \partial_\sigma\partial_\mu h^\sigma{}_\nu - \partial_\mu\partial_\nu h - \Box h_{\mu\nu}\right) \ , \tag{1.26}$$

where $h = \eta^{\mu\nu}h_{\mu\nu} = h^\mu{}_\mu$ is the trace of the perturbation. The Ricci tensor is manifestally symmetric in μ and ν. Contracting again to obtain the Ricci scalar yields

$$R = \partial_\mu\partial_\nu h^{\mu\nu} - \Box h \ . \tag{1.27}$$

Using (1.26) and (1.27), the Einstein tensor:

$$G_{\mu\nu} = \frac{1}{2}\left(\partial_\sigma\partial_\nu h^\sigma{}_\mu + \partial_\sigma\partial_\mu h^\sigma{}_\nu - \partial_\mu\partial_\nu h - \Box h_{\mu\nu} - \eta_{\mu\nu}\partial_\mu\partial_\nu h^{\mu\nu} + \eta_{\mu\nu}\Box h\right) \ . \tag{1.28}$$

The linearized field equation is $G_{\mu\nu} = 8\pi G T_{\mu\nu}$, where $G_{\mu\nu}$ is given by (1.28) and $T_{\mu\nu}$ is the stress-energy tensor calculated to zeroth order in $h_{\mu\nu}$. The conservation law to lowest order is simply $\partial_\mu T^{\mu\nu} = 0$.

The linearized field equations have an issue of gauge invariance that arises because the demand that $g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}$ does not completely specify the coordinate system on spacetime; there may be other coordinate systems in which the metric can still be written as the Minkowski metric plus a small perturbation, but the perturbation will be different. Thus, this decomposition is not unique. This problem may be avoided and it can be shown that the linearized Riemann tensor is left unchanged, similarly to gauge invariance of electromagnetism.

When faced with a system that is invariant under some kind of gauge transformations, we may think to fix it. We recall the harmonic gauge $\Box x^\mu = 0$, which is equivalent to

$$g^{\mu\nu} = \Gamma^\rho_{\mu\nu} = 0 \ . \tag{1.29}$$

In the weak filed limit this becomes

$$\frac{1}{2}\eta^{\mu\nu}\eta^{\lambda\sigma}\left(\partial_\mu h_{\nu\lambda} + \partial_\nu h_{\lambda\mu} - \partial_\lambda h_{\mu\nu}\right) = 0 \ , \tag{1.30}$$

or

$$\partial_\mu h^\mu{}_\lambda - \frac{1}{2}\partial_\lambda h = 0 \ . \tag{1.31}$$

This condition is also known as Lorentz gauge. As before, we still have some gauge freedom remaining, since we can change our coordinates by (infinitesimal) harmonic functions. Therefore, the linearized Einstein equations $G_{\mu\nu} = 8\pi G T_{\mu\nu}$ simplify to

$$\Box h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}\Box h = -16\pi G T_{\mu\nu} \ , \tag{1.32}$$

while the vacuum equations $R_{\mu\nu} = 0$ take the form

$$\Box h_{\mu\nu} = 0 \ , \tag{1.33}$$

which is simply the relativistic *wave equation*. (1.31) and (1.33) determine the evolution of a perturbation in the gravitational field in vacuum in the harmonic gauge.
It's convenient to work in the *trace-reversed* description so we define

$$\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h \ . \tag{1.34}$$

The name "trace-reversed" means that $\bar{h}^{\mu}{}_{\mu} = -h^{\mu}{}_{\mu}$. In terms of $\bar{h}_{\mu\nu}$ the harmonic gauge condition becomes

$$\partial_{\mu}\bar{h}^{\mu}{}_{\lambda} = 0 \ . \tag{1.35}$$

Therefore, the field equations are

$$\Box\bar{h}_{\mu\nu} = -16\pi G T_{\mu\nu} \ , \tag{1.36}$$

from which it follows that the vacuum equations are

$$\Box\bar{h}_{\mu\nu} = 0 \ . \tag{1.37}$$

From (1.36) we can derive the weak-field metric for a stationary spherical source such as a planet or star in the Newtonian limit. We recall that h_{00} obeys the Poisson equation, implying that

$$h_{00} = -2\Phi \ , \tag{1.38}$$

where Φ is the Newtonian potential, $\Phi = -GM/r$. Now we assume that the energy-momentum tensor of our source is dominated by its rest energy density $\rho = T_{00}$. The other components of $T_{\mu\nu}$ will be smaller than T_{00} and from (**??**) the same must hold for $\bar{h}_{\mu\nu}$. If $\bar{h}_{00}$ is much larger than $\bar{h}_{ij}$, we have

$$h = -\bar{h} = \eta^{\mu\nu}\bar{h}_{\mu\nu} = \bar{h}_{00} \ , \tag{1.39}$$

and then from (1.34) we obtain

$$\bar{h}_{00} = 2h_{00} = -4\Phi \ . \tag{1.40}$$

The other components of $\bar{h}_{\mu\nu}$ are negligible so

$$h_{i0} = \bar{h}_{i0} - \frac{1}{2}\eta_{i0}\bar{h} = 0 \ , \tag{1.41}$$

and

$$h_{ij} = \bar{h}_{ij} - \frac{1}{2}\eta_{ij}\bar{h} = -2\Phi\delta_{ij} \ . \tag{1.42}$$

So the metric for a star or planet in the weak-field limit is

$$ds^2 = -(1+2\Phi)dt^2 + (1-2\Phi)(dx^2+dy^2+dz^2) \ . \tag{1.43}$$

An application of the weak-field limit is the gravitational radiation. So from (1.37), we recognize that a particularly useful set of solutions to this wave equation are the plane waves, given by

$$\bar{h}_{\mu\nu} = C_{\mu\nu}e^{ik_\sigma x^\sigma} \ , \tag{1.44}$$

where $C_{\mu\nu}$ is a constant, symmetric tensor, and k^σ is a constant vector known as the *wave vector*. To check that it is a soluton we can plug it in (1.37). Since for not all of the components of $h_{\mu\nu}$ will be zero everywhere, we must have

$$k_\sigma k^\sigma = 0 \ . \tag{1.45}$$

The plane wave is therefore a solution to the linearized equations if the wavevector is null; this is loosely translated into the statement that *gravitational waves propagate at the speed of light*. The timelike component of the wave vector is often referred to as the *frequency* of the wave, so $k^\sigma = (\omega, k^1, k^2, k^3)$. Then the condition for k^σ to be null becomes

$$\omega^2 = \delta_{ij}k^i k^j \tag{1.46}$$

We now have to eliminate those free parameters which are the result of coordinate freedom and gauge freedom. Imposing the harmonic gauge condition we find that

$$k_\mu C^{\mu\nu} = 0 \ . \tag{1.47}$$

We say that the wave vector is orthogonal to $C^{\mu\nu}$. These four equations reduce the number of independent components of $C_{\mu\nu}$ from ten to six.
Coordinate freedom implies that for any coordinate transformation of the form

$$x^\mu \to x^\mu + \zeta^\mu \tag{1.48}$$

the harmonic condition is satisfied as long as

$$\Box\zeta^{\mu} = 0 \, . \tag{1.49}$$

Therefore, (1.49) is a wave equation, so we choose the solution

$$\zeta_{\mu} = B_{\mu} e^{ik_{\sigma}x^{\sigma}} \ , \tag{1.50}$$

where k_{σ} is the wave vector for the gravitational wave and B_{μ} are constant coefficients.
This remaining freedom allows us to convert from whatever coefficients $C_{\mu\nu}^{(old)}$ that characterize our gravitational wave to a new set $C_{\mu\nu}^{(new)}$, such that

$$C_{\mu}^{(new)\mu} = 0 \tag{1.51}$$

and

$$C_{0\nu}^{(new)} = 0 \, . \tag{1.52}$$

Under (1.48), the resulting change in our metric perturbation can be written

$$h_{\mu\nu}^{(new)} = h_{\mu\nu}^{(old)} - \partial_{\mu}\zeta_{\nu} - \partial_{\nu}\zeta_{\mu} \ , \tag{1.53}$$

which implies

$$\begin{aligned} \bar{h}_{\mu\nu}^{(new)} =& h_{\mu\nu}^{(new)} - \frac{1}{2}\eta_{\mu\nu}h^{(new)} \\ =& h_{\mu\nu}^{(old)} - \partial_{\mu}\zeta_{\nu} - \partial_{\nu}\zeta_{\mu} - \frac{1}{2}\eta_{\mu\nu}(h^{(old)} - 2\partial_{\lambda}\zeta^{\lambda}) \\ =& \bar{h}_{\mu\nu}^{(old)} - \partial_{\mu}\zeta_{\nu} - \partial_{\nu}\zeta_{\mu} + \eta_{\mu\nu}\partial_{\lambda}\zeta^{\lambda} \, . \end{aligned} \tag{1.54}$$

Using the solutions of the wave equations for ζ_{μ} and $\bar{h}_{\mu\nu}$ we obtain

$$C_{\mu\nu}^{(new)} = C_{\mu\nu}^{(old)} - ik_{\mu}B_{\nu} - ik_{\nu}B_{\mu} + i\eta_{\mu\nu}k_{\lambda}B^{\lambda} \ , \tag{1.55}$$

imposing the conditions on $C_{\mu}^{(new)\mu}$ and $C_{0\nu}^{(new)}$ we find that

$$B_0 = -\frac{i}{2k_0}\left(C_{00}^{(old)} + \frac{1}{2}C_{\mu}^{(old)\mu}\right) \tag{1.56}$$

and

$$B_j = \frac{i}{2(k_0)^2}\left[-2k_0C_{0j}^{(old)} + k_j\left(C_{00}^{(old)} + \frac{1}{2}C_{\mu}^{(old)\mu}\right)\right] \, . \tag{1.57}$$

We've used up all of our possible freedom, so these two numbers represent the physical information characterizing our plane wave in this gauge. This

can be seen more explicitly by choosing our spatial coordinates such that the wave is travelling in the x^3 direction, that is

$$k^\mu = (\omega, 0, 0, k^3) = (\omega, 0, 0, \omega) \tag{1.58}$$

and the only nonzero components of $C_{\mu\nu}$ are

$$C_{\mu\nu} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & C_{11} & C_{12} & 0 \\ 0 & C_{12} & -C_{11} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} . \tag{1.59}$$

In using up all of our gauge freedom, we have gone to a subgauge of the harmonic gauge known as the *transverse traceless gauge* or *radiation gauge*. The name comes from the fact that the metric perturbation is traceless and perpendicular to the wave vector. In this gauge we have

$$\bar{h}^{TT}_{\mu\nu} = h^{TT}_{\mu\nu} \qquad \text{(transverse traceless gauge)} . \tag{1.60}$$

Therefore, we can drop the bars over $h_{\mu\nu}$ as long as we are in this gauge. One nice feature of the transverse traceless gauge is that if you are given the components of a plane wave in some arbitrary gauge, you can easily convert them into the transverse traceless components.

To understand the physical effects due to gravitational waves, it is useful to consider the motion of test particles in the presence of a wave. To obtain a coordinate-independent measure of the wave's effects, we consider the relative motion of nearby particles, as described by the geodesic deviation equation. If we consider some nearby particles with four-velocities described by a single vector field $U^\mu(x)$ and a separation vector S^μ, we have

$$\frac{D^2}{d\tau^2} S^\mu = R^\mu{}_{\nu\rho\sigma} U^\nu U^\rho S^\sigma . \tag{1.61}$$

We would like to compute the left-hand side to first order in $h_{\mu\nu}$. If we take our test particles to be moving slowly then we can express the four-velocity as a unit vector in the time direction plus corrections of order $h_{\mu\nu}$ and higher; so we write $U^\nu = (1, 0, 0, 0)$ since the Riemann tensor is already first order. Therefore we compute $R_{\mu 00\sigma}$

$$R_{\mu 00\sigma} = \frac{1}{2} \left(\partial_0 \partial_0 h_{\mu\sigma} + \partial_\sigma \partial_\mu h_{00} - \partial_\sigma \partial_0 h_{\mu 0} - \partial_\mu \partial_0 h_{\sigma 0} \right) . \tag{1.62}$$

But $h_{\mu 0} = 0$, so

$$R_{\mu 00\sigma} = \frac{1}{2} \partial_0 \partial_0 h_{\mu\sigma} . \tag{1.63}$$

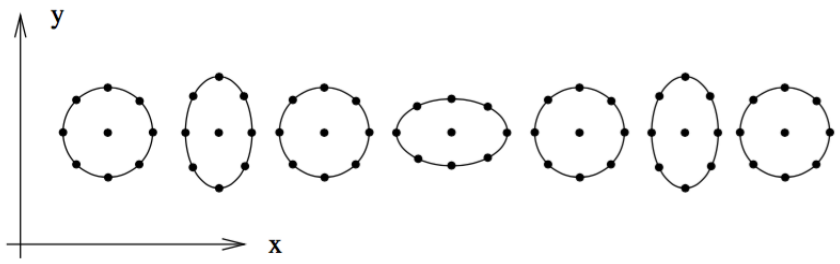

Figure 1.2: This is the effect of the C_+ polarization.

At lowest order $\tau = x^0 = t$ for our slowly-moving particles, so the geodesci deviation equation becomes

$$\frac{\partial^2}{\partial t^2} S^\mu = \frac{1}{2} S^\sigma \frac{\partial^2}{\partial t^2} h^\mu{}_\sigma \ . \tag{1.64}$$

For our wave travelling in the x^3 direction, this implies that only S^1 and S^2 will be affected. Our wave is characterized by the two numbers, which we rename $C_+ = C_{11}$ and $C_\times = C_{12}$. Then, if $C_\times = 0$, we have

$$\begin{aligned} \frac{\partial^2}{\partial t^2} S^1 &= \frac{1}{2} S^1 \frac{\partial^2}{\partial t^2} (C_+ e^{ik_\sigma x^\sigma}) \\ \frac{\partial^2}{\partial t^2} S^2 &= -\frac{1}{2} S^2 \frac{\partial^2}{\partial t^2} (C_+ e^{ik_\sigma x^\sigma}) \ . \end{aligned} \tag{1.65}$$

These yield

$$S^1 = \left(1 + \frac{1}{2} C_+ e^{ik_\sigma x^\sigma}\right) S^1(0) \ , \tag{1.66}$$

and the same for S^2 with a minus sign before the C_+ term (see Figure 1.2). On the other hand, if $C_+ = 0$, we would have

$$\begin{aligned} S^1 &= S^1(0) + \frac{1}{2} C_\times e^{ik_\sigma x^\sigma} S^2(0) \\ S^2 &= S^2(0) + \frac{1}{2} C_\times e^{ik_\sigma x^\sigma} S^1(0) \ . \end{aligned} \tag{1.67}$$

In this case the circle of particles would bounce back and forth in the shape of a "$\times$" (see Figure 1.3).
We could consider right- and left-handed circularly polarized modes by defining

$$\begin{aligned} C_R &= \frac{1}{\sqrt{2}} (C_+ + i C_\times) \ , \\ C_L &= \frac{1}{\sqrt{2}} (C_+ - i C_\times) \ . \end{aligned} \tag{1.68}$$

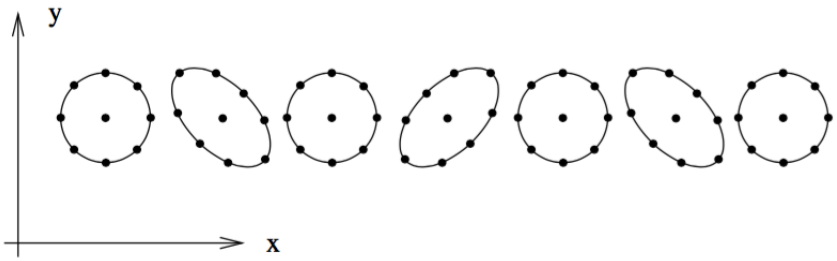

Figure 1.3: This is the effect of the $C_\times$ polarization.

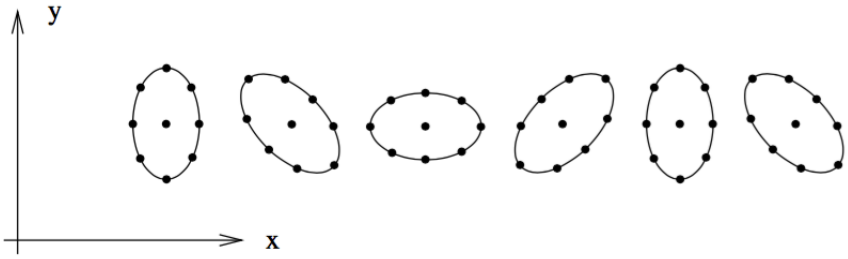

Figure 1.4: This is the effect for circular polarized modes.

With plane-wave solutions to the linearized vacuum equations in our possession, it remains to discuss the generation of gravitational radiation by sources. For this purpose it is necessary to consider the equations coupled to matter,

$$\Box \bar{h}_{\mu\nu} = -16\pi G T_{\mu\nu} \ . \tag{1.69}$$

The solution to such an equation can be obtained using a Green's function, so the general solution can be written

$$\bar{h}_{\mu\nu}(x^\sigma) = -16\pi G \int G(x^\sigma - y^\sigma) T_{\mu\nu}(y^\sigma) d^4y \ . \tag{1.70}$$

The solutions can be thought of as either "retarded" or "advanced" depending on whether they represent waves travelling forward or backward in time. For that reason, we consider the "retarded" solution

$$G(x^\sigma - y^\sigma) = -\frac{1}{4\pi|\mathbf{x}-\mathbf{y}|}\delta\left[|\mathbf{x}-\mathbf{y}| - (x^0 - y^0)\right]\theta(x^0 - y^0) \ . \tag{1.71}$$

Now plugging (1.71) into (1.70), we find

$$\bar{h}_{\mu\nu}(t, \mathbf{x}) = 4G \int \frac{1}{|\mathbf{x}-\mathbf{y}|} T_{\mu\nu}(t - |\mathbf{x}-\mathbf{y}|, \mathbf{y}) d^3y \ , \tag{1.72}$$

where $t = x^0$. The term "retarded time" is used to refer to

$$t_r = t - |\mathbf{x} - \mathbf{y}| \ . \tag{1.73}$$

The disturbance in the gravitational field at $(t, \mathbf{x})$ is a sum of the influences from the energy and momentum sources at the point $(t_r, \mathbf{x} - \mathbf{y})$ on the past light cone.

Let us take this general solution and consider the case where the gravitational radiation is emitted by an isolated source, fairly far away, comprised of nonrelativistic matter. We take the Fourier transform of the metric perturbation

$$\begin{aligned} \tilde{\bar{h}}_{\mu\nu}(\omega, \mathbf{x}) &= \frac{1}{\sqrt{2\pi}} \int dt e^{i\omega t} \bar{h}_{\mu\nu}(t, \mathbf{x}) \\ &= 4G \int d^3y e^{i\omega|\mathbf{x}-\mathbf{y}|} \frac{\tilde{T}_{\mu\nu}(\omega, \mathbf{y})}{|\mathbf{x} - \mathbf{y}|} \ . \end{aligned} \tag{1.74}$$

We now make the approximations that our source is isolated, far away, and slowly moving. This means that we can consider the source to be centered at a (spatial) distance R , with the different parts of the source at distances $R+\delta R$ such that $\delta R \ll R$. Since it is slowly moving: $\delta R \ll \omega^{-1}$ and the term $e^{i\omega t}/|\mathbf{x} - \mathbf{y}|$ can be replaced by $e^{i\omega R}/R$ and the integral in (1.74) becomes

$$\tilde{\bar{h}}_{\mu\nu}(\omega, \mathbf{x}) = 4G \frac{e^{i\omega R}}{R} \int d^3y \tilde{T}_{\mu\nu}(\omega, \mathbf{y}) \ . \tag{1.75}$$

The harmonic gauge condition in Fourier space implies

$$\tilde{\bar{h}}^{0\nu} = \frac{i}{\omega} \partial_i \tilde{\bar{h}}^{i\nu} \ . \tag{1.76}$$

By integrating by parts the $\tilde{T}_{\mu\nu}$ part of the integral and using the gauge conditions we obtain

$$\int d^3y \tilde{T}^{ij}(\omega, \mathbf{y}) = -\frac{\omega^2}{2} \int y^i y^j \tilde{T}^{00} d^3y \ . \tag{1.77}$$

It is conventional to define the *quadrupole moment tensor* of the energy density of the source

$$q_i j(t) = 3 \int y^i y^j T^{00}(t, \mathbf{y}) d^3y \ , \tag{1.78}$$

as a constant tensor on each surface of constant time. In terms of the Fourier transform of the quadrupole moment, our solution takes the compact form

$$\tilde{\bar{h}}_{ij}(\omega, \mathbf{x}) = -\frac{2G\omega^2}{3}\frac{e^{i\omega R}}{R}\tilde{q}_{ij}(\omega) \ , \tag{1.79}$$

or, transforming back to t,

$$\bar{h}_{ij}(t, \mathbf{x}) = \frac{2G}{3R}\frac{d^2 q_{ij}}{dt^2}(t_r) \ , \tag{1.80}$$

where $t_r = t - R$.
The gravitational wave produced by an isolated nonrelativistic object is therefore proportional to the second derivative of the quadrupole moment of the energy density at the point where the past light cone of the observer intersects the source.

1.4 Spherical symmetry and Schwarzschild solution

With the possible exception of Minkowski space, by far the most important such solution is that discovered by Schwarzschild, which describes spherically symmetric vacuum spacetimes. "Spherically symmetric" means "having the same symmetries as a sphere".
The simplest example is flat three-dimensional Euclidean space. If we pick an origin, then $\mathbf{R}^3$ is clearly spherically symmetric with respect to rotations around this origin. Under such rotations (i.e., under the flow of the Killing vector fields) points move into each other, but each points stays on a S^2 at a fixed distance from the origin. We can also have spherical symmetry without an "origin" to rotate things around. An example is provided by a "wormhole", with topology $\mathbf{R} \times S^2$ (see Figure 1.5. In this case the entire manifold can be foliated by two-spheres. In this case, we typically choose coordinates (θ, ϕ) for our submanifold, so the metric takes the form

$$d\Omega^2 = d\theta^2 + \sin^2\theta d\phi^2 \ . \tag{1.81}$$

So the metric on a four-dimensional spherically simmetric spacteime can be put in form

$$ds^2 = g_{aa}(a,b)da^2 + g_{ab}(a,b)(dadb + dbda) + g_{bb}(a,b)db^2 + r^2(a,b)d\Omega^2 \ . \tag{1.82}$$

Here $r(a,b)$ is an undetermined function. We can change coordinates from (a,b) to (a,R) by inverting $r(a,b)$. Therefore, the metric becomes

$$ds^2 = g_{aa}(a,r)da^2 + g_{ar}(a,r)(dadr + drda) + g_{rr}dr^2 + r^2 d\Omega^2 \ . \tag{1.83}$$

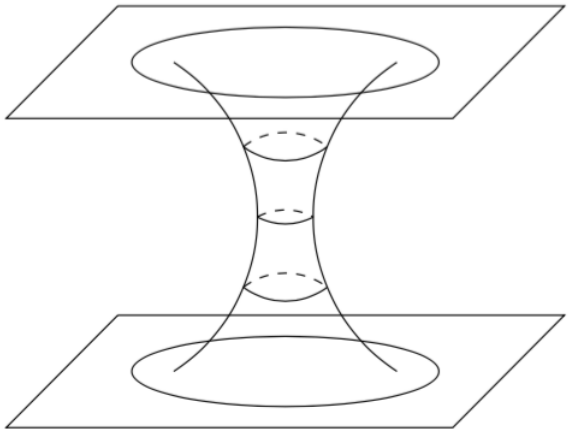

Figure 1.5: An example of "wormhole".

Now, we have to find a function $t(a, r)$ such that in the (t, r) coordinate system, there are no cross terms $dtdr + drdt$ in the metric. This change can be done by taking

$$dt = \frac{\partial t}{\partial a}da + \frac{\partial t}{\partial r}dr \ , \tag{1.84}$$

so that

$$dt^2 = \left(\frac{\partial t}{\partial a}\right)^2 da^2 \left(\frac{\partial t}{\partial a}\right)\left(\frac{\partial t}{\partial r}\right)(dadr + drda) + \left(\frac{\partial t}{\partial r}\right)^2 \ . \tag{1.85}$$

We would like to replace the first three terms in (1.83) by $mdt^2 + ndr^2$ for some function m and n. The metric takes the form

$$ds^2 = m(t, r)dt^2 + n(t, r)dr^2 + r^2 d\Omega^2 \ . \tag{1.86}$$

We know that the spacetime under consideration is Lorentzian, so either m or n have to be negative. We choose m to be negative, so

$$ds^2 = -e^{2\alpha(t,r)}dt^2 + e^{2\beta(t,r)}dr^2 + r^2 d\Omega^2 \ . \tag{1.87}$$

Now we need to solve Einstein's equations, which allow us to determine explicitly the functions $\alpha(t, r)$ and $\beta(t, r)$. The explicit calculation of the Christoffel symbols, the Riemann tensor and its contraction can be found in [20]. So we can rewrite the metric as

$$ds^2 = -e^{2\alpha(r)}dt^2 + e^{\beta(r)}dr^2 + r^2 d\Omega^2 \ . \tag{1.88}$$

All of the metric components are independent of the coordinate t. This proves that *any spherically symmetric vacuum metric possesses a timelike Killing vector.*
This property is so interesting that it gets its own name: a metric which possesses a timelike Killing vector is called *stationary.* There is also a more restrictive property: a metric is called *static* if it possesses a timelike Killing vector which is orthogonal to a family of hypersurfaces. The metric (1.88) is not only stationary, but also static.
Now, since both R_{00} and R_{11} vanish, as it is discussed in [20], we can see that $\alpha = -\beta + \text{const}$. Rescaling our coordinates, we have

$$\alpha = -\beta \ . \tag{1.89}$$

Considering that even $R_{22} = 0$, we find

$$e^{2\alpha}(2r\partial_1\alpha + 1) = 1 \tag{1.90}$$

that is completely equivalent to

$$\partial_1(re^{2\alpha}) = 1 \ . \tag{1.91}$$

If we solve this, we obtain

$$e^{2\alpha} = 1 + \frac{\mu}{r} \ , \tag{1.92}$$

where μ is some undetermined constant. Therefore, the metric becomes

$$ds^2 = -\left(1 + \frac{\mu}{r}\right) dt^2 + \left(1 + \frac{\mu}{r}\right)^{-1} dr^2 + r^2 d\Omega^2 \ . \tag{1.93}$$

We now have no freedom left except for the single constant μ. The only thing left to do is to interpret μ in terms of some physical parameter. The most important use of a spherically symmetric vacuum solution is to represent the spacetime outside a star or planet. In that case we would expect to recover the weak field limit as $r \to \text{inf}$. In this limit

$$\begin{aligned} g_{00}(r \to \text{inf}) &= -\left(1 + \frac{\mu}{r}\right) \ , \\ g_{rr}(r \to \text{inf}) &= -\left(1 - \frac{\mu}{r}\right) \ . \end{aligned} \tag{1.94}$$

On the other hand, the weak field limit implies

$$\begin{aligned} g_{00} &= -(1 + 2\Phi) \ , \\ g_{rr} &= (1 + 2\Phi) \ , \end{aligned} \tag{1.95}$$

with the potential $\Phi = -GM/r$. Therefore the metrics do agree in this limit if $\mu = -2GM/r$.
As a result we obtain the *Schwarzschild metric*

$$ds^2 = -\left(1 - \frac{2GM}{r}\right) dt^2 + \left(1 - \frac{2GM}{r}\right)^{-1} dr^2 + r^2 d\Omega^2 \ . \tag{1.96}$$

This is true for any spherically symmetric vacuum solution to Einstein's equations. Note that as $M \to 0$ we recover Minkowski space, which is to be expected. Note also that the metric becomes progressively Minkowskian as we go to $r \to \inf$, which is known as *asymptotic flatness*. The fact that the Schwarzschild metric is not just a good solution, but is the unique spherically symmetric vacuum solution, is known as *Birkhoff's theorem*.
From (1.96), the metric coefficients become infinite at $r = 0$ and $r = 2GM$, which is the sign of something wrong in our choice. As in Section 7 of [20], it's shown that $r = 0$ is actually a singularity since

$$R^{\mu\nu\rho\sigma} R_{\mu\nu\rho\sigma} = 12G^2M^2/r^6 \ , \tag{1.97}$$

but the behavior at $r = 2GM$ need to be explained. Proceeding like in Section 7 of [20], we can see that the surface $r = 2GM$ is locally perfectly regular, but globally functions as a point of no return. For that reason, $r = 2GM$ is known as the *event horizon*: no event at $r \leq 2GM$ can influence any other event at $r > 2GM$. Notice also that since nothing can escape the event horizon, it is impossible for an external observer to "see inside", thus the name *black hole*.

2. 3+1 Formalism

The *3+1 formalism* [21] is an approach to general relativity and to Einstein equations that relies on the slicing of the four-dimensional spacetime by three-dimensional *hypersurfaces*. These have to be spacelike, so that the metric induced on them by the spacetime metric [signature(−,+,+,+)] is *Riemannian* [signature(+,+,+)]. From the mathematical point of view, this procedure allows to formulate the problem of resolution of Einstein equations as a *Cauchy problem* with constraints.

2.1 Geometry of foliation

We assume that the spacetime $(\mathcal{M}, g)$ can be foliated into a family of non-intersecting, spacelike, three-dimensional hypersurfaces Σ. At least locally, these timeslices Σ form level surfaces of a scalar function$t$, which we will identify with the coordinate time. The 1-form $\Omega = \mathrm{d}t$ has the norm

$$|\Omega|^2 = g^{\mu\nu}\nabla_\mu t \nabla_\nu t \equiv -\alpha^{-2} \,, \tag{2.1}$$

where the *lapse function* α is strictly positive. This implies that the surfaces Σ are spacelike. We now define the unit normal vector n^μ as

$$n^\mu \equiv -\alpha g^{\mu\nu}\Omega_\nu \equiv -\alpha g^{\mu\nu}\nabla_\nu t \,. \tag{2.2}$$

The negative sign has been chosen so that n^μ points into the direction of increasing t. The four-dimensional spacetime metric $g_{\mu\nu}$ now induces the *spatial metric*

$$\gamma_{\mu\nu} \equiv g_{\mu\nu} + n_\mu n_\nu \tag{2.3}$$

on the hypersurface Σ.
Any four-dimensional tensor can now be decomposed into spatial parts, which live in the hypersurface Σ, and timelike parts, which are normal to Σ and hence aligned with n^μ. The spatial part can be found by contracting with the projection operator

$$\gamma^\mu{}_\nu = g^{\mu\rho}\gamma_{\rho\nu} = g^\mu{}_\nu + n^\mu n_\nu = \delta^\mu{}_\nu + n^\mu n_\nu \,, \tag{2.4}$$

and the timelike part by contracting with

$$N^{\mu}{}_{\nu} = -n^{\mu}n_{\nu} \ . \tag{2.5}$$

We now define the three-dimensional covariant derivative of a spatial tensor by projecting all indices of a four-dimensional covariant derivative into Σ

$$D_{\mu}T^{\nu}{}_{\rho} \equiv \gamma_{\mu}{}^{\sigma}\gamma_{\gamma}{}^{\nu}\gamma_{\rho}{}^{\delta}\nabla_{\sigma}T^{\gamma}{}_{\delta} \ . \tag{2.6}$$

It is easy to show that this derivative is compatible with the spatial metric, $D_{\mu}\gamma_{\nu\rho} = 0$ as it is supposed to be. The three-dimensional covariant derivative can be expressed in terms of three-dimensional connection coefficients, which in a coordinate basis, are given by

$$\Gamma^{\mu}_{\nu\rho} = \frac{1}{2}\gamma^{\mu\sigma}\left(\gamma_{\sigma\nu,\rho} + \gamma_{\sigma\rho,\nu} - \gamma_{\nu\rho,\sigma}\right) \ . \tag{2.7}$$

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

$$2D_{[\mu}D_{\nu]}w_{\rho} = R^{\sigma}{}_{\rho\nu\mu}w_{\sigma}, \qquad R^{\sigma}{}_{\rho\nu\mu}n_{\sigma} = 0 \tag{2.8}$$

for every spatial w_{σ}. In terms of spatial components, Riemann can be computed from

$$R^{\mu}{}_{\nu\rho\sigma} = \Gamma^{\mu}_{\nu\sigma,\rho} - \Gamma^{\mu}_{\nu\rho,\sigma} + \Gamma^{\gamma}_{\nu\sigma}\Gamma^{\mu}_{\gamma\rho} - \Gamma^{\gamma}_{\nu\rho}\Gamma^{\mu}_{\gamma\sigma} \ . \tag{2.9}$$

Contracting the Riemann tensor yields the three-dimensional Ricci tensor $R_{\mu\nu} = R^{\rho}{}_{\mu\nu\rho}$ and the three-dimensional scalar curvature $R = R^{\mu}{}_{\mu}$.
It is intuitive that casting Einstein's equations into a $3+1$ form will necessitate expressing the four-dimensional Riemann tensor ${}^{(4)}R^{\sigma}{}_{\mu\nu\rho}$ in terms of the three-dimensional $R^{\sigma}{}_{\mu\nu\rho}$. It is also clear, however, that the latter cannot contain all the relevant information. This missing information is expressed by a tensor called *extrinsic curvature* which describes how the slice Σ is embedded in the spacetime $\mathcal{M}$. The extrinsic curvature can be defined as

$$K_{\mu\nu} \equiv \gamma_{\mu}{}^{\rho}\gamma_{\nu}{}^{\sigma}\nabla_{(\rho}n_{\sigma)} \ . \tag{2.10}$$

$K_{\mu\nu}$ is spatial and symmetric by construction. Alternatively, the extrinsic curvature can be found in terms of the "acceleration" of normal observers $a_{\mu} = n^{\nu}\nabla_{\nu}n_{\mu}$ as $K_{\mu\nu} = -\nabla_{\mu}n_{\nu} - n_{\mu}n_{\nu}$. Since $a^{\mu}n_{\mu} = 0$ the trace of the extrinsic curvature is

$$K \equiv g^{\mu\nu}K_{\mu\nu} = -\nabla^{\mu}n_{\mu} \ . \tag{2.11}$$

Another equivalent expression is in terms of the Lie derivative (see Appendix A for details) of the spatial metric along the normal vector n^{μ}

$$K_{\mu\nu} = \frac{1}{2}\mathcal{L}_{n}\gamma_{\mu\nu} \ . \tag{2.12}$$

Eq. (2.12) clearly illustrates the interpretation of $K_{\mu\nu}$ as a time-derivative of the spatial metric.
Given its symmetry properties ${}^{(4)}R^{\mu}{}_{\nu\rho\sigma}$ can be projected in three different ways. Projecting all four indices into Σ yields to *Gauss' equation*,

$$R_{\mu\nu\rho\sigma} + K_{\mu\rho}K_{\nu\sigma} - K_{\mu\sigma}K_{\nu\rho} = \gamma^{\alpha}{}_{\mu}\gamma^{\beta}{}_{\nu}\gamma^{\gamma}{}_{\rho}\gamma^{\delta}{}_{\sigma}\ {}^{(4)}R_{\alpha\beta\gamma\delta}\ . \tag{2.13}$$

Three spatial projections and a contraction with n^{μ} yields the *Codazzi equation*

$$D_{\mu}K_{\nu\rho} - D_{\nu}K_{\mu\rho} = \gamma^{\gamma}{}_{\nu}\gamma^{\alpha}{}_{\mu}\gamma^{\beta}{}_{\rho}n^{\delta}\ {}^{(4)}R_{\gamma\alpha\beta\delta} \tag{2.14}$$

Finally, two spatial projections and two contractions with n^{μ} yield *Ricci's equation*

$$\mathcal{L}_n K_{\mu\nu} = n^{\sigma\rho}\gamma^{\beta}{}_{\mu}\gamma^{\gamma}{}_{\nu}\ {}^{(4)}R_{\gamma\sigma\beta\rho} - \frac{1}{\alpha}D_{\mu}D_{\nu}\alpha - K^{\rho}{}_{\nu}K_{\mu\rho}\ , \tag{2.15}$$

where the derivative of the lapse enter through the identity $a_{\mu} = D_{\mu}\ln\alpha$.

2.2 The ADM equations

In Section 2.1 wesimplyrecordedgeometricalidentitiesrelatingthegeometryofthethree-dimensional hypersurfaces Σ to the geometry of the embedding four-dimensional spacetime $\mathcal{M}$. As we have seen in Chapter 1, the geometry of the latter is governed dynamically by Einstein's equations

$${}^{(4)}G_{\mu\nu} \equiv\ {}^{(4)}R_{\mu\nu} - \frac{1}{2}\ {}^{(4)}R g_{\mu\nu} = 8\pi T_{\mu\nu} \tag{2.16}$$

where ${}^{(4)}G_{\mu\nu}$ is the Einstein tensor and $T_{\mu\nu}$ the stress-energy tensor. We take the projections of Einstein's equations into Σ and n^{μ} and use Eq. (2.13), (2.14) and (2.15) to eliminate the four-dimensional Ricci tensor. The result will be the ADM (Arnowitt-Deser-Misner) equations, which relate three-dimensional curvature quantities to projections of the stress-energy tensor. One relation that is very useful in these derivations is

$$D_{\mu}V^{\nu} = \gamma_{\mu}{}^{\rho}\nabla_{\rho}V^{\nu} + K_{\mu\rho}V^{\rho}n^{\nu}\ , \tag{2.17}$$

which holds for any spatial vector V^{μ}.
Contracting Eq. (2.13) twice with the spatial metric and inserting (2.3) yields

$$2n^{\mu}n^{\nu}\ {}^{(4)}G_{\mu\nu} = R + K^2 - K_{\mu\nu}K^{\mu\nu}\ . \tag{2.18}$$

We now de ne the total energy density as measured by a normal observer n^μ as

$$\rho \equiv n^\mu n^\nu T_{\mu\nu} \tag{2.19}$$

and using (2.16) we find the *Hamiltonian constraint*

$$R + K^2 - K_{\mu\nu}K^{\mu\nu} = 16\pi\rho \; . \tag{2.20}$$

Similarly, contracting Eq. (2.14) once we find the *momentum constraint*

$$D_\nu K^\mu{}_\nu - D_\mu K = 8\pi j_\mu \; , \tag{2.21}$$

where

$$j_\mu \equiv -\gamma^\nu{}_\mu n^\rho T_{\nu\rho} \tag{2.22}$$

is the momentum density (mass current) as measured by a normal observer n^μ. The Hamiltonian and momentum constraints are called constraint equations because they only involve spatial quantities andtheirspatialderivatives.They therefore have to hold on each individual spatial slice.
Evolution equations that describe how $\gamma_{\mu\nu}$ and $K_{\mu\nu}$ evolve in time, from one spatial slice to the next, can be found from Eq. (2.12) and the Ricci equation (2.15). However, the Lie derivative along n^μ is not a natural time derivative orthogonal to the spatial slices, since n^μ is not dual to the surface 1-form Ω, i.e. their dot product is

$$n^\mu \Omega_\mu = -\alpha g^{\mu\nu}\nabla_\mu t \nabla_\nu t = \alpha^{-1} \; . \tag{2.23}$$

instead the vector

$$t^\mu = \alpha n^\mu + \beta^\mu \tag{2.24}$$

is dual to Ω for any spatial *shift vector* β^μ. The lapse α and the shift β^μ together determine how the coordinates evolve from one slice to the next. The lapse determines how much proper time elapses between timeslices along the normal vector n^μ while the shift determines by how much spatial coordinates are shifted with respect to the normal vector.
Note that

$$t^\mu \Omega_\mu = t^\mu \nabla_\mu t = 1 \tag{2.25}$$

implies that the Lie derivative along t^μ is a natural time derivative and the integral curves of t^μ are naturally parametrized by t. Therefore, all vectors t^μ originating on one spatial slice Σ_1 will end on the same spatial slice Σ_2 (unlike n^μ, which generally would end on different slices).
Rewriting Eq. (2.12) in terms of t^μ yields the evolution equation for the spatial metric

$$\mathcal{L}_t \gamma_{\mu\nu} = -2\alpha K_{\mu\nu} + \mathcal{L}_\beta \gamma_{\mu\nu} \; . \tag{2.26}$$

Combining Eq. (2.16) with (2.15) we find the evolution equation for the extrinsic curvature

$$\begin{aligned}\mathcal{L}_t K_{\mu\nu} = &- D_\mu D_\nu \alpha + \alpha \left(R_{\mu\nu} - 2K_{\mu\rho}K^\rho{}_\nu + K K_{\mu\nu} \right) \\ &- \alpha 8\pi \left(S_{\mu\nu} - \frac{1}{2}\gamma_{\mu\nu}(S-\rho) \right) + \mathcal{L}_\beta K_{\mu\nu}\end{aligned} \tag{2.27}$$

where $S_{\mu\nu} \equiv \gamma_{\mu\rho}\gamma_{\nu\sigma}T^{\rho\sigma}$ and S is the trace $S \equiv \gamma^{\mu\nu}S_{\mu\nu}$.
While the two constraint equations (2.20) and (2.21) contrain $\gamma_{\mu\nu}$ and $K_{\mu\nu}$ on every spatial slice Σ, the evolution equation (2.26) and (2.27) describe how quantities evolve from one slice to the next. It can be deomstrated that this structure is very similar to that of Maxwell's equations.
So far, we have made no assumptions about the choice of coordinates, and have expressed all quantities in a coordinate-independent way. So we introduce a basis of spatial vectors e_i that span each slice Σ, so that $\Omega_\mu(e_i)^\mu = 0$. It can be shown that this condition is preserved if the spatial vectors are Lie dragged along t^μ. As the fourth basis vector we choose $(e_0)^\mu = t^\mu$, or, in components, $t^\mu = (1,0,0,0)$. Therefore, the Lie derivative along t^μ reduces to a partial derivative with respect to t. Since $n_i = n_\mu(e_i)^\mu = \alpha\Omega_\mu(e_i)^\mu = 0$, the covariant, spatial components of the normal vector vanish. Since contractions of any spatial tensor with the normal vector are zero, this implies that the zeroth components of contravariant spatial tensors have to vanish. For the shift vector, $\beta^\mu n_\mu = \beta^0 n_0 = 0$ so we can write

$$\beta^\mu = (0, \beta^i) \ . \tag{2.28}$$

From (2.24)

$$n^\mu = \frac{1}{\alpha}(1, -\beta^i) \tag{2.29}$$

and, since $n^\mu n_\mu = -1$,

$$n_\mu = (-\alpha, 0, 0, 0) \ . \tag{2.30}$$

From the defnition of the spatial metric we find $\gamma_{ij} = g_{ij}$. The inverse metric can therefore be expressed as

$$g^{\mu\nu} = \begin{pmatrix} -\alpha^{-2} & \alpha^{-2}\beta^i \\ \alpha^{-2}\beta^j & \gamma^{ij} - \alpha^{-2}\beta^i\beta^j \end{pmatrix} \tag{2.31}$$

It is possible to show that γ_{ij} and γ^{ij} are inverses, so that $\gamma^{ik}\gamma_{kj} = \delta^i{}_j$ and they can be used to raise and lower indices of spatial tensors. Inverting (2.31) we find the components of the four-dimensional metric

$$g_{\mu\nu} = \begin{pmatrix} -\alpha^2 + \beta_k\beta^k & \beta_i \\ \beta_j & \gamma_{ij} \end{pmatrix} \tag{2.32}$$

and the equivalent line element

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt) \; . \tag{2.33}$$

We can therefore restrict the constraint and evolution equations to spatial components and restrict all contractions to spatial components. The connection coefficients reduce to

$$\Gamma^i_{jk} = \frac{1}{2}\gamma^{il}\left(\gamma_{lj,k} + \gamma_{il,k} - \gamma_{jk,l}\right) \; , \tag{2.34}$$

and expressing the Ricci tensor in terms of second derivatices of the metric yields

$$R_{ij} = \frac{1}{2}\gamma^{kl}\left(\gamma_{kj,il} + \gamma_{il,kj} - \gamma_{kl,ij} - \gamma_{ij,kl}\right) + \gamma^{kl}\left(\Gamma^m_{il}\Gamma_{mkj} - \Gamma^m_{ij}\Gamma_{mkl}\right) \; . \tag{2.35}$$

With these simplifications, the Hamiltonian constraint (2.20) now becomes

$$R + K^2 - K_{ij}K^{ij} = 16\pi\rho \; , \tag{2.36}$$

the momentum constraint (2.21)

$$D_j K^j_i - D_i K = 8\pi j_i \; , \tag{2.37}$$

the evolution equation for the metric (2.26)

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i\beta_j + D_j\beta_i \; , \tag{2.38}$$

and the evolution equation for the extrinsic curvature (2.27)

$$\begin{aligned}\partial_t K_{ij} = &- D_i D_j \alpha + \alpha(R_{ij} - 2K_{ik}K^k{}_j + KK_{ij}) - \alpha 8\pi\left(S_{ij} - \frac{1}{2}\gamma_{ij}(S-\rho)\right) \\ &+ \beta^k D_k K_{ij} + K_{ik}D_j\beta^k + K_{kj}D_i\beta^k \; .\end{aligned} \tag{2.39}$$

The shift terms in the last two equations arise from the Lie derivatives $\mathcal{L}_\beta\gamma_{ij}$ and $\mathcal{L}_\beta K_{ij}$. Eq. (2.36), (2.37), (2.38), (2.39) are commonly referred to as the *ADM equations*. These equations are not yet in a form that is generally well suited for numerical implementation.

Note that the ADM equations only determine the spatial metric γ_{ij} and the extrinsic curvature K_{ij} but not the lapse α or the shift β^i. The latter determine how the coordinates evolve from one timeslice to the next and re ect the coordinate freedom of general relativity.

For later purposes it is useful to take the traces of the evolution equation (2.38) and (2.39). Since $\partial_t \ln \gamma = \gamma^{ij}\partial_t \gamma_{ij}$, we find

$$\partial_t \ln \gamma^{1/2} = -\alpha K + D_i \beta^i \ , \tag{2.40}$$

for the determinant of the metric γ, and combining (2.36) with the trace of (2.39)

$$\partial_t K = -D^2\alpha + \alpha(K_{ij}K^{ij} + 4\pi(\rho + S)) + \beta^i D_i K \ , \tag{2.41}$$

where $D^2 \equiv \gamma^{ij} D_i D_j$ is the covariant Laplace operator associated with γ_{ij}.

2.3 BSSNOK Formulation

The ADM equations can be shown to be weakly hyperbolic and hence the problem is "ill-posed". For that reason, we need to make the system hyperbolic and we introduce new evolution variables to obtain a set of strongly hyperbolic equations. Starting from the *conformal transverse-traceless decomposition* with $\gamma_{ij} = \psi^4 \tilde{\gamma}_{ij}$ and writing the conformal factor ψ as $\psi = e^{\phi}$ so that

$$\tilde{\gamma}_{ij} = e^{-4\phi}\gamma_{ij} \ , \tag{2.42}$$

and by choosing it such that the determinant of the conformally related metric $\tilde{\gamma}_{ij}$ is unity

$$\phi = \frac{1}{12}\ln \gamma \ . \tag{2.43}$$

Then we need to re-scale the traceless part of A_{ij} (the conformal extrinsic curvature)

$$\tilde{A}_{ij} = e^{-4\phi} A_{ij} \tag{2.44}$$

Evolution equations for ϕ and K can now be found from Eq. (2.40), yielding

$$\partial_t \phi = -\frac{1}{6}\alpha K + \beta^i \partial_i \phi + \frac{1}{6}\partial_i \beta^i \tag{2.45}$$

and (2.41)

$$\partial_t K = -\gamma^{ij} D_j D_i \alpha + \alpha\left(\tilde{A}_{ij}\tilde{A}^{ij} + \frac{1}{3}K^2\right) + 4\pi\alpha(\rho + S) + \beta^i \partial_i K \ . \tag{2.46}$$

Subtracting these from the evolution equations (2.38) and (2.39) yields the traceless evolution equations for $\tilde{\gamma}_{ij}$

$$\partial_t \tilde{\gamma}_{ij} = -2\alpha\tilde{A}_{ij} + \beta^k \partial_k \tilde{\gamma}_{ij} + \tilde{\gamma}_{ik}\partial_j \beta^k + \tilde{\gamma}_{kj}\partial_i \beta^k - \frac{2}{3}\tilde{\gamma}_{ij}\partial_k \beta^k \ , \tag{2.47}$$

and $\tilde{A}_{ij}$

$$\begin{aligned}\partial_t \tilde{A}_{ij} =& e^{-4\phi}\left(-(D_i D_j \alpha)^{TF} + \alpha(R_{ij}^{TF} - 8\pi S_{ij}^{TF})\right) + \alpha(K\tilde{A}_{ij} - 2\tilde{A}_{il}\tilde{A}^l{}_j) \\ &+ \beta^k \partial_k \tilde{A}_{ij} + \tilde{A}_{ik}\partial_j \beta^k + \tilde{A}_{kj}\partial_i \beta^k - \frac{2}{3}\tilde{A}_{ij}\partial_k \beta^k \ .\end{aligned} \tag{2.48}$$

The superscript *TF* denotes the trace-free part of a tensor. The shift terms arise from the Lie derivatives of the respective variable.
We can now split the Ricci tensor into

$$R_{ij} = \tilde{R}_{ij} R^{\phi}_{ij} \ . \tag{2.49}$$

The conformally related Ricci tensor $\tilde{R}_{ij}$ could be computed by inserting $\tilde{\gamma}_{ij}$ into (2.35). We introduce a new variable $\tilde{\Gamma}$ (called *conformal connection*) to eliminate the mixed derivatives that arise in $\tilde{R}_{ij}$

$$\tilde{\Gamma}^i \equiv \tilde{\gamma}^{jk}\tilde{\Gamma}^i{}_{jk} = -\tilde{\gamma}^{ij}_{,j} \ , \tag{2.50}$$

where the $\tilde{\Gamma}^i$ are the connection coefficients associated with $\tilde{\gamma}_{ij}$, and where the last equality holds because $\tilde{\gamma} = 1$. In terms of these, the Ricci tensor can be written

$$\tilde{R}_{ij} = \frac{1}{2}\tilde{\gamma}^{lm}\tilde{\gamma}_{ij,lm} + \tilde{\gamma}_{k(i}\partial_{j)}\tilde{\Gamma}^k + \tilde{\Gamma}^k\tilde{\Gamma}_{(ij)k} + \tilde{\gamma}^{lm}\left(2\tilde{\Gamma}^k{}_{l(i}\tilde{\Gamma}_{j)km} + \tilde{\Gamma}^k{}_{im}\tilde{\Gamma}_{klj}\right) \ . \tag{2.51}$$

We now promote the $\tilde{\Gamma}^i$ to independent functions, and hence need to derive their evolution equation. So we find

$$\partial_t \tilde{\Gamma}^i = -\partial_j\left(2\alpha\tilde{A}^{ij} - 2\tilde{\gamma}^{m(j}\beta^{i)}{}_{,m} + \frac{2}{3}\tilde{\gamma}^{ij}\beta^l{}_{,l} + \beta^l\tilde{\gamma}^{ij}{}_{,l}\right) \ . \tag{2.52}$$

The divergence of the extrinsic curvature can now be eliminated with the help of the momentum constraint, which yields the evolution equation

$$\begin{aligned}\partial_t \tilde{\Gamma}^i =& -2\tilde{A}^{ij}\partial_j \alpha + 2\alpha\left(\tilde{\Gamma}^i{}_{jk}\tilde{A}^{kj} - \frac{2}{3}\tilde{\gamma}^{ij}\partial_j K - 8\pi\tilde{\gamma}^{ij}S_j + 6\tilde{A}^{ij}\partial_j\phi\right) \\ &+ \beta^j\partial_j\tilde{\Gamma}^i - \tilde{\Gamma}^j\partial_j\beta^i + \frac{2}{3}\tilde{\Gamma}^i\partial_j\beta^j + \frac{1}{3}\tilde{\gamma}^{li}\beta^j{}_{,jl} + \tilde{\gamma}^{li}\beta^i{}_{,lj} \ .\end{aligned} \tag{2.53}$$

So equations (2.45), (2.46), (2.47), (2.48) and (2.53) form a new system of evolution equations, the BSSNOK scheme. Eq. (2.50) serves as a new constraint (together with Hamiltonian and momentum constraints) since $\tilde{\Gamma}^i$ are evolved as independent functions.

The BSSNOK evolution scheme is widely used and it has been demonstrated to lead to stable and accurate evolution of vacuum (binary black holes) and non-vacuum (neutron stars) spacetimes. For that reason, we used the BSSNOK scheme for the evolution of the spacetime metric in our simulations, as we set in the paremeter files we describe in Chapter 4. Here we report a small part of these parameter files to show our choice.

```
# MacLachlan evolution parameters

ADMBase::metric_type                    = physical
ADMBase::evolution_method               = ML_BSSN
ADMBase::lapse_evolution_method         = ML_BSSN
ADMBase::shift_evolution_method         = ML_BSSN
ADMBase::dtlapse_evolution_method       = ML_BSSN
ADMBase::dtshift_evolution_method       = ML_BSSN
```

2.4 Gauge Conditions

The lapse α and β^i are not solutions of the Einstein equations but represent the *gauge freedom*. This allows us to choose the coordinate system that better fits to our problem to reduce the computational cost of our simulations. In this section, we describe the possible choices for α and β^i and we refer to the first one as *slicing condition* and the second one as *spatial gauge condition*.

2.4.1 Geodesic slicing

The simplest choice of foliation one might think about is the *geodesic slicing*, for it corresponds to a unit lapse:

$$\alpha = 1. \tag{2.54}$$

This choice implies that the acceleration of the Eulerian observers is $\mathbf{a} = 0$ so the wordlines of the Eulerian observer are geodesics, hence the name *geodesic slicing*. Moreover the choice implies that the proper time along these wordlines coincides with the coordinate time t. The spacetime metric tensor with this choice of coordinates (called *Gaussian normal coordinates*) takes a simple form:

$$g_{\mu\nu}dx^\mu dx^\nu = -dt^2 + \gamma_{ij}dx^i dx^j. \tag{2.55}$$

However, this is not the best choice we can make since timelike geodesics without vorticity, such as the wordlines of the Eulerian observers, have the tendency to focus and eventually cross. In general, this makes also impossible to get a Gaussian normal coordinate system that covers all $\mathcal{M}$.

2.4.2 Maximal Slicing

Alternatively, the foliation $(\Sigma_t)_{t\in\mathbb{R}}$ may be chosen so that

$$K = 0 \tag{2.56}$$

i.e. the hypersurfaces Σ_t have a vanishing mean curvature. This is called the *maximal slicing* condition since it leads to hypersurfaces of maximal volume. An interesting property of maximal slicing is the *singularity avoidance*. This is related to the fact that the set of Eulerian observers of a maximal foliation define an incompressible flow. If we compare to the Eulerian observers of geodesic slicings, who have the tnedency to squeeze, we may say maximal slicing Eulerian observers do not converge. However, this is just a good mathematical idea since combining the maximal slicing condition with the evolution equation for K yields the following elliptic equation for the lapse function:

$$D^i D_i \alpha = \alpha[K_{ij}K^{ij} + 4\pi(\rho + S)]. \tag{2.57}$$

Elliptic equations are expensive to solve at each time with numerical calculation so this makes this would not be a good choice for us.

2.4.3 1+log Slicing

Another important category of time slicing is deduced from the harmonic gauge for the spacetime coordinates (x^α):

$$\Box_{\mathbf{g}} x^\alpha = 0 \tag{2.58}$$

where $\Box_{\mathbf{g}} := \nabla_\mu \nabla^\mu$ is the d'Alambertian associated with metric $\mathbf{g}$ and each coordinate x^α is considered as a scalar field on $\mathcal{M}$. The *harmonic slicing* is defined by requiring that the harmonic condition holds for the $x^0 = t$ coordinate, but not necessarily for the other coordinates, leaving the freedom to choose any coordinate x^i in each hypersurface Σ_t:

$$\Box_{\mathbf{g}} t = 0. \tag{2.59}$$

Therefore, this choice yields an evolution equation for the lapse function:

$$(\partial_t - \mathcal{L}_\beta)\,\alpha = -K\alpha^2. \tag{2.60}$$

So the harmonic slicing has some singularity avoidance feature, but weaker than that of maximal slicing.
Bona, Massò, Seidel and Stela (1995) [22] have generalized the harmonic slicing condition requiring that lapse and shift dynamically satisfy simple evolution equations. The generalized form is:

$$(\partial_t - \mathcal{L}_\beta)\,\alpha = -K\alpha^2 f(\alpha) \tag{2.61}$$

where $f(\alpha)$ is an arbitrary function. We can see that the $f = 0$ choice corresponds to the geodesic slicing, $f = 1$ to the harmonic one, $f \to \inf$ to the maximal slicing while choosing $f(\alpha) = 2/\alpha$ leads to

$$(\partial_t - \mathcal{L}_\beta)\,\alpha = -2K\alpha. \tag{2.62}$$

So, using Eq. (2.47) for $-K\alpha$ we obtain

$$(\partial_t - \mathcal{L}_\beta)\,\alpha = \partial_t(\ln\gamma) - 2D_i\beta^i. \tag{2.63}$$

If we use normal coordinates, $\beta^i = 0$ and Eq. (2.63) reduces to

$$\partial_t\alpha = \partial_t(\ln\gamma), \tag{2.64}$$

which has a solution

$$\alpha = 1 + \ln\gamma. \tag{2.65}$$

For this reason, a foliation whose lapse obeys to Eq. (2.62) is called *1+log slicing*[1].
The 1+log slicing is widely used in vacuum and non-vacuum spacetimes thanks to its strong singularity avoidance properties (it has been found to "mimic" maximal slicing), although in a slightly modified form of Eq. (2.62):

$$(\partial_t - \mathcal{L}_\beta)\,\alpha = -2\alpha(K - K_0) \tag{2.66}$$

where K_0 is the value of K at $t = 0$.

2.4.4 Normal coordinates

A similar choice to the geodesic slicing as seen in section 2.4.1, would be setting the shift vector to zero:

$$\beta = 0. \tag{2.67}$$

For this choice, $x^i = \text{const}$ are normal to Σ_t, hence the name *normal coordinates*. An advantage of normal coordinates is to be as regular as foliation without introducing some pathology. On the other hand, they may lead to large values of the metric coefficients γ_{ij}, especially if rotation is present.

[1] note that even if $\beta^i \neq 0$ we still define the 1+log slicing by condition (2.62), although Eq. (2.65) does no longer hold.

2.4.5 Minimal distortion and "Gamma freezing"

The idea of this choice of coordinates is to minimize the variation of $\tilde{\gamma}$ during the evolution. So, we define

$$\dot{\tilde{\gamma}}_{ij} = \partial_t \tilde{\gamma}_{ij} \tag{2.68}$$

that is what we are trying to minimize. There is not a unique way to proceed so we decided to use the *distortion tensor* $\mathbf{Q}$, defined as:

$$Q_{ij} = \partial_t \gamma_{ij} - \frac{1}{3}\gamma^{kl}\partial_t \gamma_{kl}\gamma_{ij}. \tag{2.69}$$

Now we can rewrite the distortion in terms of the conformal metric $\tilde{\gamma}$ and the conformal factor ϕ:

$$Q_{ij} = e^{4\phi}\dot{\tilde{\gamma}}_{ij}. \tag{2.70}$$

At this point, working on the decomposition of Q into its longitudinal and traceless transverse part, we obtain the *minimal distortion* condition:

$$D^j Q_{ij} = 0. \tag{2.71}$$

Using Eq. (2.47) we may write Eq. (2.69) as

$$Q_{ij} = -2\alpha K_{ij} - \mathcal{L}_\beta \gamma_{ij} + \frac{1}{3}\left(-2\alpha K + 2D_k\beta^k\right)\gamma_{ij} \tag{2.72}$$

and thanks to the momentum constraint as in Eq. (2.21) we obtain

$$D_j D^j \beta^i + \frac{1}{3}D^i D_j \beta^j + R^i{}_j \beta^j = 16\pi\alpha p^i + \frac{4}{3}\alpha D^i K + 2A^{ij}D_j\alpha \tag{2.73}$$

which is the elliptic equation on the shift vector that one has to solve in order to enforce the minimal distortion. We may rewrite it in terms of $\tilde{\gamma}$ using Eq. (2.70) but we do not show the details of the calculation. As we mentioned before, elliptic equations are too expensive to solve.
A very similar choice is the *Gamma freezing* prescription for the evolution of spatial coordinates. Briefly, the Gamma freezing conditions is equivalent to

$$\partial_t \tilde{\Gamma}^i = 0 \tag{2.74}$$

hence the name "Gamma freezing": for such a choice, $\tilde{\Gamma}^i$ does not evolve. This condition also leads to an elliptic equation for the shift vector.

2.4.6 Gamma drivers

As seen above, the minimal distortion and Gamma freezing conditions yield to elliptic equations for the shift vector so Alcubierre and Brügmann in [23] proposed to turn it into a parabolic equation by considering

$$\partial_t \beta^i = k \partial_t \tilde{\Gamma}^i \tag{2.75}$$

where k is a positive function. This is also called *parabolic Gamma driver* and leads to a parabolic equation for the shift vector.
In 2003 Alcubierre, Brügmann, Diener, Koppitz, Pollney, Seidel and Takahashi [24] introduced an alternative with the *hyperbolic Gamma driver* prescription:

$$\partial_t^2 \beta^i = k \partial_t \tilde{\Gamma}^i - (\eta - \partial_t k)\, \partial_t \beta^i \tag{2.76}$$

where k and η are two positive functions. It has been found in [24] that η is a dissipation term. This condition leads to a hyperbolic equation for the shift vector which makes it a good choice for numerical calculation.

3. The Code

3.1 The Einstein Toolkit

The Einstein Toolkit is a community-driven, freely accessible computational infrastructure intended for use in numerical relativity, relativistic astrophysics and other applications. It's based on a set of core modules that provide basic functionality to create, deploy and manage numerical simulations. It is made of four parts:

1. the `Cactus` framework, on which the majority of the code is based, provides the underlying infrastructure to build complex simulation codes out of independently developed modules;

2. the adaptive mesh refinement driver, `Carpet`, that is build on top of `Cactus` and provides problem independent adaptive mesh refinement support for simulations that eed to resolve physics on lenght scales differing by many order of magnintude;

3. `Kranc`, which generates code in a computer language from a high-level description in *Mathematica*;

4. the *Simulation Factory*, which provides a uniform, high-level interface to run and manage the simulations.

3.2 Cactus Framework and Basic Modules

The `Cactus` computational toolkit consists in general modules which provide parallel drivers, coordinates, boundary conditions, interpolators, reduction operators and efficient I/O in different data formats. The structure is completely modular, with only a small core (the "flesh") which provides the interfaces between modules both at compile- and run-time. The `Cactus` modules (called "thorns") compiled into an eecutable can remain dormant at run-time if there's no need to activate them.

3.2.1 ADMBase

As we described in Chapter 2, 4-dimensional spacetime can be foliated into sequences of spacelike 3-dimensional hypersurfaces (slices) connected by timelike normal vectors. This module provides the codes to solve Einstein's general relativity equation in the 3+1 ADM formulation, as we discussed in Section 2.2.
According to this, the spacetime metric is assumed to take the form of equation (**??**). The remaining dynamical component of the spacetime is contained in the definition of the extrinsic curvature K_{ij}, which is defined in terms of the time derivative of the metric after incorporating a Lie derivative with respect to the shift vector:

$$K_{ij} = -\frac{1}{2\alpha}\left(\partial_t - \mathcal{L}_\beta\right)\gamma_{ij} \tag{3.1}$$

By default, `ADMBase` initializes the 3-metric and extrinsic curvature to Minkowski (i.e., $\gamma_{ij} = \delta_{ij}$ and $K_{ij} = 0$) the shift to zero and the lapse to unity. Otherwise, the type of the initial data chosen for the simulation may be set manually in `ADMBase::initial_data` for 3-metric and extrinsic curvature, `ADMBase::initial_lapse`, `ADMBase::initial_shift`. This thorn also allows to read these parameters from an external source, as described below for our parameter files.

```
ADMBase::initial_data            = "Meudon_Bin_NS"
ADMBase::initial_lapse           = "Meudon_Bin_NS"
ADMBase::initial_shift           = "zero"
ADMBase::initial_dtlapse         = "zero"
ADMBase::initial_dtshift         = "zero"
```

`ADMBase` may be used to hold the variables static during the dynamical evolution, so we set

```
ADMBase::metric_type                    = physical
ADMBase::evolution_method               = ML_BSSN
ADMBase::lapse_evolution_method         = ML_BSSN
ADMBase::shift_evolution_method         = ML_BSSN
ADMBase::dtlapse_evolution_method       = ML_BSSN
ADMBase::dtshift_evolution_method       = ML_BSSN
```

The variables defined in `ADMBase` typically are not used for the actual evolution of the curvature so this enables modules which perform analysis on the

spacetime variables to use these variables without any dependency on any of the existing curvature evolution methods.

3.2.2 HydroBase

Similar to `ADMBase` we described in section 3.2.1, the module `HydroBase` defines a common basis for interactions between modules of a given evolution problem, in this case relativistic hydrodynamics. `HydroBase`'s main function is to store variables which are common to most if not all hydrodynamics codes solving the Euler equations, the so-called primitive variables.
`HydroBase` uses a set of conventions known as the Valencia formulation [25, 26], defining the following primitive variables (see [27] for details):

- **`rho`**: rest mass density ρ;
- **`press`**: pressure P;
- **`eps`**: internal energy density ϵ;
- **`vel[3]`**: contravariant fluid three-velocity v^i defined as

$$v^i = \frac{u^i}{\alpha u^0} + \frac{\beta^i}{\alpha}; \tag{3.2}$$

- **`Y_e`**: electron fraction Y_e;
- **`temperature`**: temperature T;
- **`entropy`**: specific entropy per particle s;
- **`Bvec[3]`**: contravariant magnetic field vector defined as

$$B^i = \frac{1}{\sqrt{4\pi}} n_\nu F^{*\nu i} \tag{3.3}$$

  in terms of the dual $F^{*\mu\nu} = \frac{1}{2}\varepsilon^{\mu\nu\alpha\beta}F_{\alpha\beta}$ to the Faraday tensor and the unit normal of the foliation of spacetime $n^\mu \equiv \alpha^{-1}\left[1, -\beta^i\right]^T$.

In our parameter files the `HydoBase` thorn is used to load the initial data parameters from the `Meudon_Bin_NS` module

```
HydroBase::initial_hydro          = "Meudon_Bin_NS"
```

set the time levels for the simulation

```
HydroBase::timelevels             = 3
```

define the evolution methods

```
HydroBase::evolution_method       = "GRHydro"
HydroBase::Bvec_evolution_method = "GRHydro"
```

and the initial value for **Bvec[3]**

```
HydroBase::initial_Bvec        = "
   bin_ns_poloidalmagfield"
```

3.2.3 TmunuBase

The last of the base modules of the Einstein Toolkit we describe is `TmunuBase`, which provides the grid function for the stress-energy tensor $T_{\mu\nu}$ as well as the schedule groups to manage when $T_{\mu\nu}$ is calculated.
Typically, we choose the stress-energy tensor $T^{\mu\nu}$ to be that of an ideal relativistic fluid. We recall the form of $T^{\mu\nu}$ in this form

$$T^{\mu\nu} = \rho h u^{\mu} u^{\nu} - g^{\mu\nu} P \tag{3.4}$$

where ρ is the rest mass density and $h = 1 + \epsilon + P/\rho$ is the relativistic specific enthalpy.
We used this thorn to set the storage instructions for the stress-energy tensor, as we report here

```
TmunuBase::stress_energy_storage = "yes"
TmunuBase::stress_energy_at_RHS  = "yes"
TmunuBase::timelevels            =  1
TmunuBase::prolongation_type     = "none"
TmunuBase::support_old_CalcTmunu_mechanism = "no"
```

3.2.4 McLachlan

The Einstein Toolkit curvature evolution code `McLachlan` is generated via `Kranc` (see [28] and at the beginning of Chapter 3 for details) and implements the Einstein equations in a 3+1 split as a Cauchy boundary

value problem [29]. This module employs the Baumgarte-Shapiro-Shibata-Nakamura (BSSN) conformal-tracefree formulation we described in section 2.3. In `McLachlan`, there are a few adjustment implemented that can be found in the `Kranc` source code which generates the `ML_BSSN` module.
The consraints are the same described in Eq. (2.20), (2.21) and the additional ones for the BSSN formulation

$$\begin{aligned}
\tilde{\mathcal{H}} &\equiv R + \frac{2}{3}K^2 - \tilde{A}_{ij}\tilde{A}^{ij} = 0 \\
\tilde{\mathcal{M}}^i &\equiv \tilde{D}_j\tilde{A}^{ij} + 6\tilde{A}^{ij}\partial_j\phi - \frac{2}{3}\tilde{\gamma}^{ij}\partial_j K = 0 \\
\tilde{\mathcal{G}} &\equiv \tilde{\gamma} - 1 = 0 \\
\tilde{\mathcal{A}} &\equiv \tilde{\gamma}^{ij}\tilde{A}_{ij} = 0 \\
\tilde{\mathcal{L}}^i &\equiv \tilde{\Gamma}^i + \partial_j\tilde{\gamma}^{ij} = 0 \,.
\end{aligned} \tag{3.5}$$

The constraints $\tilde{\mathcal{G}}$ and $\tilde{\mathcal{A}}$ are enforced actively at each time-step, while $\tilde{\mathcal{H}}$, $\tilde{\mathcal{M}}^i$ and $\tilde{\mathcal{L}}^i$ are not. In addition, where derivatives of $\tilde{\Gamma}^i$ are needed, the evolved $\tilde{\Gamma}^i$ is used, otherwise $\tilde{\gamma}^{jk}\tilde{\Gamma}^i{}_{jk}$ are used.
As we described in section 2.4, α and β^i are our gauge variables so we have to choose the gauge conditions. The module `McLachlan` allows to choose the foliation in the *1 + log* family. The gauge condition for the evolution of α is implemented as

$$\partial_t\alpha = -F\alpha^N K + \mathbf{alphaDriver}(\alpha - 1) + \mathbf{advectLapse}\beta^i\partial_i\alpha \tag{3.6}$$

where **alphaDriver**, **alphaDriver**, N and F are manually set in the parameter file. For *1 + log* slicing one must choose $N = 1$ and $F = 2$.
Therefore, the hyperbolic gamma driver condition:

$$\partial_t\beta^i = \mathbf{shiftGammaCoeff}B^i + \mathbf{advectShift}\beta^j\partial_j\beta^i \tag{3.7}$$

$$\partial_t B^i = \partial\tilde{\Gamma}^i - \mathbf{betaDriver}B^i + \mathbf{advectDriver}\beta^j\partial_j B^i \tag{3.8}$$

The gamma driver shift condition is symmetry-seeking, driving the shift β^i to a stat that renders the conformal connection functions $\tilde{\Gamma}^i$ stationary. This stationary state cannot be achieved while the metric is evolving, but in a stationary spacetime evolution of the shift β^i and thus that of the spatial coordinates x^i will be exponentially damped. This is how we set the variables for the `McLachlan` thorn in our simulations:

```
ML_BSSN::timelevels                          = 3
ML_BSSN::my_initial_data                     = "ADMBase"
```

```
ML_BSSN::my_initial_boundary_condition  = "extrapolate
   -gammas"
ML_BSSN::my_boundary_condition          = "none"
ML_BSSN::my_rhs_boundary_condition       = "NewRad"
Boundary::radpower                      = 2

ML_BSSN::harmonicN           = 1
ML_BSSN::harmonicF           = 2.0
ML_BSSN::ShiftGammaCoeff     = 0.75
ML_BSSN::AlphaDriver         = 0.0
ML_BSSN::BetaDriver          = 0.75
ML_BSSN::LapseAdvectionCoeff = 1.0
ML_BSSN::ShiftAdvectionCoeff = 1.0

ML_BSSN::MinimumLapse = 1.0e-8
ML_BSSN::ML_log_confac_bound = "none"
ML_BSSN::ML_metric_bound     = "none"
ML_BSSN::ML_Gamma_bound      = "none"
ML_BSSN::ML_trace_curv_bound = "none"
ML_BSSN::ML_curv_bound       = "none"
ML_BSSN::ML_lapse_bound      = "none"
ML_BSSN::ML_dtlapse_bound    = "none"
ML_BSSN::ML_shift_bound      = "none"
ML_BSSN::ML_dtshift_bound    = "none"

ML_BSSN::UseSpatialBetaDriver = "yes"
ML_BSSN::SpatialBetaDriverRadius = 50

ML_BSSN::apply_dissipation   = "never"
```

We make a short description of how this module manage the boundary conditions, as set in

```
ML_BSSN::my_boundary_condition          = "none"
ML_BSSN::my_rhs_boundary_condition       = "NewRad"
```

The main part of the boundary condition assumes that one has an outgoing radial wave with some speed v_0:

$$X = X_0 + \frac{u(r - v_0 t)}{r} \tag{3.9}$$

where X is any of the tensor components of evolved variables, X_0 the value at infinity, and u a spherically simmetric perturbation. Both X_0 and v_0 depend on the particular variable and have to be specified, implying that

$$\partial_t X = -v^i \partial_i X - v_0 \frac{X - X_0}{r} \tag{3.10}$$

where $v^i = v_0 x^i / r$. The spatial derivatives ∂_i are evaluated using centered finite differencing where possible, and one-sided finite differencing elsewhere. In addition, it is assumed that these parts decay with a certain power p of the radius. This is implemented by considering the radial derivative of the source term above, and extrapolating according to this power-law decay. Given a source term $\partial_t X$, one defines the corrected source term $(\partial_t X)^*$:

$$(\partial_t X)^* = (\partial_t X) + \left(\frac{r}{r - n^i \partial_i r} \right)^p n^i \partial_i (\partial_t X) \tag{3.11}$$

where n^i is the normal vector of the corresponding boundary face. The spatial derivatives ∂_i are evaluated by comparing neighboring grid points. Second-order decay is assumed, hence $p = 2$.
As with the initial conditions above, this boundary condition is evaluated on several layers of grid points, starting from the innermost layer. This is an approximation of the actual decay behavior of the BSSN vector, however it leads to stable evolutions if applied far from the source. This boundary condition are implemented in the `NewRad` component of the Einstein Toolkit.

3.2.5 GRHydro

`GRHydro` is a code derived from the public `Whisky` [30] for relativistic hydrodynamics evolution. It includes a high resolution shock capturing (HRSC) scheme to evolve hydrodynamic quantities, with several different reconstruction methods and Riemann solvers, as we discuss below. `GRHydro` operates in three stages for each timestep:

- The variables are "reconstructed" at cell faces using shock-capturing techniques like total variation diminishing (TVD), piecewise parabolic (PPM) and essentially non-oscillatory (ENO) methods. We discuss only the WENO (weighted essentially non-oscillatory) method which we used in our simulations (see [31] for details on the other methods).

- It solves a Riemann problem at each cell face using one of the implemented solvers: HLLE (Harten-Lax-van Leer-Einfeldt), which we used, Roe and Marquina.

- The conserved variables are advanced and used to recalculate the new values of the primitive variables.

GRHydro employs the *ideal MHD* (magneto-hydrodynamics) approximation – fluids have infinite conductivity and there is no charge separation. Thus, electric fields $E_\nu = u_\mu F^{\mu\nu}$ in the rest frame of the fluid vanishes and ideal MHD corresponds to imposing the following condition:

$$u_\mu F^{\mu\nu} = 0 \tag{3.12}$$

As in [31] we rescale $F^{\mu\nu}$ and its dual ${}^*F^{\mu\nu} \equiv \frac{1}{2}\epsilon^{\mu\nu\kappa\lambda}F_{\kappa\lambda}$ by a factor $1/\sqrt{4\pi}$. The hydrodynamic and electromagnetic contributions to the stress-energy tensor are given, respectively, by

$$T^{\mu\nu}_H = \rho h u^\mu u^\nu + P g^{\mu\nu} = (\rho + \rho\epsilon + P)u^\mu u^\nu + P g^{\mu\nu} \tag{3.13}$$

and

$$T^{\mu\nu}_{EM} = F^{\mu\lambda}F^\nu{}_\lambda - \frac{1}{4}g^{\mu\nu}F^{\lambda\kappa}F_{\lambda\kappa} = b^2 u^\mu u^\nu - b^\mu b^\nu + \frac{b^2}{2}g^{\mu\nu} \tag{3.14}$$

where ρ, ϵ, P, u^μ and h are the same variables as seen in section **??**, while b^μ is the magnetic four-vector (the projected component of the Maxwell tensor parallel to the four-velocity of the fluid):

$$b^\mu = u_\nu \, {}^*F^{\mu\nu}. \tag{3.15}$$

Note that $b^2 = b^\mu b_\nu = 2P_m$ whereP_m is the magnetic pressure. Combining Eq. (3.13) and (3.14), the stress-energy tensor takes the form:

$$\begin{aligned} T^{\mu\nu} &= \left(\rho + \rho\epsilon + P + b^2\right) u^\mu u^\nu + \left(P + \frac{b^2}{2}\right) g^{\mu\nu} - b^\mu b^\nu \\ &\equiv \rho h^* u^\mu u^\nu + P^* g^{\mu\nu} - b^\mu b^\nu \end{aligned} \tag{3.16}$$

where the * quantities are the magnetically modified ones. The spatial magnetic field (living on spacelike three-hypersurfaces) is defined as the Eulerian component of the Maxwell tensor

$$B^i = n_\mu \, {}^*F^{i\mu} = -\alpha \, {}^*F^{i0}, \tag{3.17}$$

where $n_\mu = [-\alpha, 0, 0, 0]$ is the normal vector to the hypersurface as defined in Chapter 2. The equations of ideal GRHMD (general relativistic magnetohydrodynamics) evolved by GRHYDRO are derived from the local conservation laws of mass and energy-momentum

$$\nabla_\mu J^\mu = 0, \qquad \nabla_\mu T^{\mu\nu} = 0, \tag{3.18}$$

where ∇_μ denotes the covariant derivative with respect to the four-metric and $J^\mu ? \rho u^\mu$ is the mass current, and from Maxwell equations,

$$\nabla_\nu \, {}^*F^{\mu\nu}. \tag{3.19}$$

The `GRHydro` scheme is written in a first-order hyperbolic flux-conservative evolution system for the conserved variables D, S^i, τ and $\mathcal{B}^i$ defined in terms of the primitive variables such that

$$\begin{aligned} D &= \sqrt{\gamma}\rho W, \\ S_j &= \sqrt{\gamma}\left(\rho h^* W^2 v_j - \alpha b^0 b_j\right), \\ \tau &= \sqrt{\gamma}\left(\rho h^* W^2 - P^* - (ab^0)^2\right) - D, \\ \mathcal{B}^k &= \sqrt{\gamma} B^k, \end{aligned} \tag{3.20}$$

where γ is the determinant of γ_{ij}. The three-velocity v^i is defined to correspond to the velocity seen by an Eulerian observer at rest in the current spatial three-hypersurface,

$$v^i = \frac{u^i}{W} + \frac{\beta^i}{\alpha} \tag{3.21}$$

and $W \equiv (1 - v^i v_i)^{-1/2}$ is the Lorentz factor.
The evolution system for Eq. (3.18) and (3.19) is

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}^i}{\partial x^i} = \mathbf{S} \tag{3.22}$$

where

$$\begin{aligned} \mathbf{U} &= \left[D, S_j \tau, \mathcal{B}^k\right], \\ \mathbf{F}^i &= \alpha \times \begin{bmatrix} D\tilde{v}^i \\ S_j \tilde{v}^i + \sqrt{\gamma} P^* \delta^i_j - b_j \mathcal{B}^i / W \\ \tau \tilde{v}^i + \sqrt{\gamma} P^* v^i - a b^0 \mathcal{B}^i / W \\ \mathcal{B}^k \tilde{v}^i - \mathcal{B}^i \tilde{v}^k \end{bmatrix}, \\ \mathbf{S} &= \alpha\sqrt{\gamma} \times \begin{bmatrix} 0 \\ T^{\mu\nu}\left(\frac{\partial g_{\nu j}}{\partial x^\mu} - \Gamma^\lambda_{\mu\nu} g_{\lambda j}\right) \\ \alpha\left(T^{\mu 0} \frac{\partial \ln \alpha}{\partial x^\mu} - T^{\mu\nu}\Gamma^0_{\mu\nu}\right) \\ \overrightarrow{0} \end{bmatrix}. \end{aligned} \tag{3.23}$$

Here, $\tilde{v}^i = v^i - \beta^i/\alpha$ and $\Gamma^\lambda_{\mu\nu}$ are the four-Christoffel symbols. In Eq. (3.19), the time component yields the "no monopoles" constraint (the magnetic field is divergence-free):

$$\nabla \cdot B \equiv \frac{1}{\sqrt{\gamma}} \partial_i \left(\sqrt{\gamma} B^i\right) = 0 \tag{3.24}$$

which also implies

$$\partial_i \mathcal{B}^i = 0. \tag{3.25}$$

These conditions are implemented in `GRHydro` as *divergence cleaning method* and *constrained transport* which are discussed in [31].

Numerical methods for GRHydro

Now we describe the the numerical methods implemented in `GRHydro` we used for our simulations to reconstruct fluid and magnetic field variables (WENO), the HLLE Riemann solver and how to transform the primitive variables into the conserved ones.
As we explained at the beginning of section 3.2.5, the first stage of the evolution at each timestep is the "reconstructon" of the primitive variables at the cell faces in order to construct the Riemann problem to solve. The *reconstruction* step interpolates the fluid state from cell averaged values to values at cell interfaces without introducing oscillations at shocks and other discontinuities. We describe the fifth-order *weighted essentially non-oscillatory* (WENO5) method, which we chose for our simulations.
First of all, we define $U^L_{i+1/2}$ to be the value of an element of our conservative variable state vector $\mathbf{U}$ on the left side of the face between $U_i \equiv U(x_i, y, z)$ and $U_{i+1} \equiv U(x_{i+1}, y, z)$ where x_i is the i^{th} point in the x-direction, and U^R_{i+2} the value on the right side of the same face. Then we introduce three interpolation polynomials that approximate U^L_{i+1} from cell-averages of a given quantity U_i:

$$\begin{aligned} U^{L,1}_{i+1/2} &= \frac{3}{8}U_{i-2} - \frac{5}{4}U_{i-1} + \frac{15}{8}U_i \ , \\ U^{L,2}_{i+1/2} &= -\frac{1}{8}U_{i-1} + \frac{3}{4}U_i - \frac{3}{8}u_{i+1} \ , \\ U^{L,3}_{i+1/2} &= \frac{3}{8}U_i + \frac{3}{4}U_{i+1} - \frac{1}{8}U_{i+2} \ . \end{aligned} \tag{3.26}$$

Each of the three polynomials yields a third-order accurate approximation of $\mathbf{U}$ at the cell interface. We introduce

$$U^L_{i+1/2} = w^1 U^{L,1}_{i+1/2} + w^2 U^{L,2}_{i+1/2} + w^3 U^{L,3}_{i+1/2} \ , \tag{3.27}$$

which is a linear combination of the three interpolants, where w^i are the weights and satisfy $\sum_i w^i = 1$. Therefore, is the fifth-order interpolation polynomial that spans alla five stencil points $U_{i-2}, ..., U_{i+2}$. The weights w^i are computed using so-called smoothness indicators β^i, which are given by

(in the original WENO algorithm [32])

$$\begin{aligned}
\beta^1 &= \frac{1}{3}\left(4u_{i-2}^2 - 19u_{i-2}u_{i-1} + 25u_{i-1}^2 + 11u_{i-2}u_i - 31u_{i-1}u_i + 10u_i^2\right) \ , \\
\beta^2 &= \frac{1}{3}\left(4u_{i-1}^2 - 13u_{i-1}u_i + 13u_i^2 + 5u_{i-1}u_{i+1} - 13u_iu_{i+1} + 4u_{i+1}^2\right) \ , \\
\beta^3 &= \frac{1}{3}\left(10u_i^2 - 31u_iu_{i+1} + 25u_{i+1}^2 + 11u_iu_{i+2} - 19u_{i+1}u_{i+2} + 4u_{i+2}^2\right) \ .
\end{aligned} \tag{3.28}$$

Note that in this case β^i refers to the smoothness indicators and not to the shift vector. The weights are obtained using

$$w^i = \frac{\bar{w}^i}{\bar{w}^1 + \bar{w}^2 + \bar{w}^3} \qquad \text{with} \qquad \bar{w}^i = \frac{\gamma^i}{(\epsilon + \beta^i)^2} \tag{3.29}$$

where $\gamma^i = 1/16, 5/8, 5/16$ and ϵ is a small constant to avoid division by zero. The choice of ϵ is scale dependent and cannot be fixed with for cases with a large variation in scales. This can't be improved defining the *modified* smoothness indicators (3.28):

$$\bar{\beta}^i = \beta^i + \epsilon|U^2| + \delta \ , \tag{3.30}$$

where $|U^2|$ is the sum of the U_j^2 in the i-th stencil and δ is the smallest number chosen floating point variable type can hold. In `GRHydro` these parameters are set to be

$$\bar{\beta}^i = \beta^i + \epsilon\left(|U^2| + 1\right) \ , \tag{3.31}$$

with $\epsilon = 10^{-26}$.

Once the Riemann problem is constructed, `GRHydro` proceeds by solving it. One of the methods implemented is the Harten-Lax-van Leer-Einfeldt (HLLE) [33] approximate solver which we chose for our simulations. HLLE uses a two-wave approximation to compute the update terms across the discontinuity at the cell interface with ξ_- and ξ_+ the most negative and most positive wave speed eigenvalues present on either side of the interface. The solution state vector $\mathbf{U}$ is assumed to take the form

$$\mathbf{U} = \begin{cases} \mathbf{U}^L & \text{if } \ 0 < \xi_- \ , \\ \mathbf{U}_* & \text{if } \ \xi_- < 0 < \xi_+ \ , \\ \mathbf{U}^R & \text{if } \ 0 > \xi_+ \ , \end{cases} \tag{3.32}$$

with $\mathbf{U}_*$ defined as

$$\mathbf{U}_* = \frac{\xi_+\mathbf{U}^R - \xi_-\mathbf{U}^L - \mathbf{F}(\mathbf{U}^R) + \mathbf{F}(\mathbf{U}^L)}{\xi_+ - \xi_-}. \tag{3.33}$$

The numerical flux along the interface takes the form

$$\mathbf{F}(\mathbf{U}) = \frac{\hat{\xi}_+\mathbf{F}(\mathbf{U}^L) - \hat{\xi}_-\mathbf{F}(\mathbf{U}^R) + \hat{\xi}_+\hat{\xi}_-(\mathbf{U}^R - \mathbf{U}^L)}{\hat{\xi}_+ - \hat{\xi}_-} \,, \tag{3.34}$$

where $\hat{\xi}_- = \min(0, \xi_-)$ and $\hat{\xi}_+ = \max(0, \xi_+)$. The flux terms in Eq. (3.34) are used to evolve the hydrodynamics quantities. When computing the wavespeeds, we replace the full MHD dispersion relation by the approximate quadratic form

$$\omega_d^2 = k_d^2 \left[v_A^2 + c_s^2 \left(1 - \frac{v_A^2}{c^2} \right) \right] \,, \tag{3.35}$$

where the wavefactor is defined as $k_\mu \equiv (-\omega, k_i)$, the fluid speed c_s, the Alfvén velocity

$$v_A \equiv \sqrt{\frac{b^2}{\rho h + b^2}} = \sqrt{\frac{b^2}{\rho h^*}} \,, \tag{3.36}$$

the projected wave vector

$$K_\mu \equiv (\delta_\mu^\nu + u_\mu u^\nu) k_\nu, \tag{3.37}$$

and the dispersion relation between frequency and (squared) wave number

$$\begin{aligned} \omega_d &= k_\mu u^\mu = -\omega u^0 + k u^i \,, \\ k_d^2 &= K_\mu K^\mu = \omega_d^2 + g^{\nu\sigma} k_\nu k_\sigma \,. \end{aligned} \tag{3.38}$$

The resulting quadratic may be written

$$\begin{aligned} &\xi^2 \left[W^2(V^2 - 1) - V^2 \right] - 2\xi \left[\alpha W^2 \tilde{v}^i (V^2 - 1) + V^2 \beta^i \right] + \\ &+ \left[(\alpha W \tilde{v}^i)^2 (V^2 - 1) + V^2 (\alpha^2 \gamma^{ii} - \beta^i \beta^i) \right] = 0 \end{aligned} \tag{3.39}$$

where $V^2 \equiv v_A^2 + c_s^2(1 - v_A^2)$ and ξ is the resulting wavespeed[1].

The last step is converting the conservative variables back to the primitives, which can be accomplished by inverting (3.20) with all the magnetic field quantities set to zero. In `GRHydro` there's implemented a 1D Newton-Raphson scheme (see [34]). The scheme iterates by estimating the fluid pressure, determining the density, internal energy and the conservative quantities given this estimate, and using those in turn to calculate a new value for the pressure and its residual.
While the last equation of (3.20) is immediately invertible, b^μ cannot be

[1] note that the indices are not to be summed over

immediately determined. We consider the values of the undensitised conservative variables,

$$\begin{aligned} \hat{D} &\equiv \frac{D}{\sqrt{\gamma}} = \rho W \ , \qquad\qquad \hat{S}^i \equiv \frac{S^i}{\sqrt{\gamma}} = \rho h^* W^2 v^i - \alpha b^0 b^i \ , \\ \hat{\tau} &\equiv \frac{\tau}{\sqrt{\gamma}} = \rho h^* W^2 - P^* - (\alpha b^0)^2 - \hat{D} \end{aligned} \tag{3.40}$$

the GRHD system (in which $P^* = P$, $h^* = h$ and $b^\mu = 0$) allows us to define

$$Q \equiv \hat{\tau} + \hat{D} + P = \rho h W^2 \ , \tag{3.41}$$

and then determine the density as a function of pressure through the relation

$$\rho = \frac{\hat{D}\sqrt{Q^2 - \gamma_{ij}\hat{S}^i\hat{S}^j}}{Q} \ , \tag{3.42}$$

thus the Lorentz factor and internal energy as well. The most efficient approach for inverting the conservative variable set is often to use multi-dimensional Newton-Raphson solver, with simplifications for barotropic EOS, for whichc the internal energy is assumed to be a function of the density only, eliminating the need to evolve the energy equation.
In `GRHydro` a few other quantities are defined for use in numerical calculations. From the values of the conservative variables and the metric, the momentum density is defined as

$$\mathcal{S}_\mu \equiv -n_\nu T^\nu{}_\mu = \alpha T^0{}_\mu \ , \tag{3.43}$$

whose spatial components are given by the relation $\mathcal{S}_i \equiv \hat{S}_i$ and its normal projection $\tilde{\mathcal{S}}$ given by

$$\tilde{\mathcal{S}}_\mu = (\delta^\nu_\mu + n_\mu n^\nu)\mathcal{S}_\nu \ . \tag{3.44}$$

For cases where the pressure and internal energy are functions of the rest mass density only, the EOS is barotropic (the polytropic $P = K\rho^\Gamma$ EOS we chose is a special case of a barotropic EOS). The inversion uses $\mathcal{S}^2 = \hat{S}_i\hat{S}^i$ only to eliminate the variable v^2 from the scheme by solving

$$v^2(Q) = \frac{Q^2\mathcal{S}^2 + (\mathcal{S}\cdot B)^2(B^2 + 2Q)}{Q^2(B^2+Q)^2} \ . \tag{3.45}$$

For a polytropic EOS, we solve for $\rho(Q)$ first through an independent Newton-Raphson loop over the equation

$$\rho Q = \hat{D}\left(1 + \frac{\Gamma K \rho^{\Gamma-1}}{\Gamma - 1}\right) \ . \tag{3.46}$$

Using (3.45) and $v^2 = \frac{\rho^2}{\hat{D}^2} - 1$ we obtain

$$\begin{aligned} 0 &= Q^2(B^2+Q)^2v^2 - Q^2(B^2+Q)^2v^2 \ , \\ &= Q^2\mathcal{S}^2 + (\mathcal{S}\cdot B)^2(B^2+2Q) - \left(\frac{\rho^2}{\hat{D}^2} - 1\right) Q^2(B^2+Q)^2 \ . \end{aligned} \tag{3.47}$$

When performing the Newton-Raphson step, all quantities in (3.47) are known a priori except the iteration variable Q and $\rho(Q)$, which depends upon it. The loop over Q uses the derivative of

$$\frac{d\rho}{dQ} = \frac{\rho}{\hat{D}^2\gamma K\rho^{\Gamma-2} - Q} \ . \tag{3.48}$$

3.2.6 EOS Omni

The module `EOS_Omni` provides a unified general equation of state (EOS) interface and back-end for simple analytic and complex microphysical EOSs. In our simulations, we used the polytropic EOS

$$P = K\rho^{\Gamma} \tag{3.49}$$

where K is the polytropic constant and Γ (we set it to be $\Gamma = 3$) the adiabatic index, which is appropriate for adiabatic (= isentropic) evolution without shocks. When using the polytropic EOS, one does not need to evolve the total fluid energy equation, since the specific internal energy ϵ is fixed to

$$\epsilon = \frac{K\rho^{\Gamma}}{(\Gamma-1)\rho} \ . \tag{3.50}$$

More precisely, we chose a hybrid EOS which includes a thermal component designed for the application in simple models of stellar collapse. In particular, $P(\rho,\epsilon) = P_{cold}(\rho) + P_{th}(\rho,\epsilon)$ and (3.49) becomes

$$P_{cold} = K\rho^{\Gamma} \ , \tag{3.51}$$

$$\epsilon_{cold} = \epsilon + \frac{K}{\Gamma-1}\rho^{\Gamma-1} \ , \tag{3.52}$$

$$P_{th} = \Gamma_{th}\rho(\epsilon - \epsilon_{cold}) \ , \tag{3.53}$$

where the parameters are chosen to be:

```
EOS_Omni::poly_gamma                    = 3
EOS_Omni::poly_k                        = 80000
EOS_Omni::hybrid_gamma_th               = 1.8
EOS_Omni::n_pieces                      = 1
EOS_Omni::hybrid_gamma[0]               = 3
EOS_Omni::hybrid_k0                     = 80000
```

We followed the discussion in [35] for the choice of $\Gamma_{th} = 1.8$.
This thor also allows to get the settings from other modules and read external EOS tables.

3.2.7 AHFinder

When spacetimes contain a BH, localizing its horizon is necessary for describing time-dependent quasi-local measures such as its mass and spin. `AHFinder` allows to locate the apparent horizons (AHs) defined locally on a hypersurface.
This module makes use of the fact that null geodesics have vanishing expansion on an AH which, in the usual 3+1 quantities, can be written

$$\Theta \equiv \nabla_i n^i + K_{ij} n^i n^j - K = 0 \, . \tag{3.54}$$

The module `AHFinder` provides two algorithms for locating AHs. The minimization algorithm, which finds the local minimum of $\oint_S (\Theta - \Theta_o)^2 d^2S$ corresponding to a surface of constant expansion Θ_o with $\Theta_o = 0$ corresponding to the AH. An alternative algorithm provided by `AHFinder`, the flow algorithm (see [36]), defines a surface as a level set $f(x^i) = r - h(\theta, \phi) = 0$ and introduce an unphysical timelike parameter λ to parametrize the flow of h towards a solution. So (3.54) can be rewritten

$$\partial_\lambda = -\left(\frac{\alpha}{l_{max}(l_{max}+1)} + \beta\right)\left(1 - \frac{\beta}{\alpha}L^2\right)^{-1} \rho\Theta \tag{3.55}$$

where ρ is a strictly positive weight, L^2 is the Laplacian of the 2D metric and α, β and l_{max} are free parameters. Decomposing $h(\theta, \phi)$ onto a basis of spherical harmonics, the coefficients a_{lm} evolve iteratively towards a solution as

$$a_{lm}^{(n+1)} = a_{lm}^{(n)} - \frac{\alpha + \beta l_{max}(l_{max}+1)}{l_{max}(l_{max}+1)(1 + \beta l(l+1)/\alpha)} (\rho\Theta)_{lm}^{(n)} \tag{3.56}$$

The `AHFinderDirect` module, which is the one we use in our simulations, is a faster alternative to `AHFinder`. Its approach is to view (3.54) as an

elliptic PDE for $h(\theta, \phi)$ on S^2 using standard finite differencing methods. So (3.54) becomes

$$\Theta \equiv \Theta\left(h, \partial_u h, \partial_{uv} h; \gamma_{ij}, K_{ij}, \partial_k \gamma_{ij}\right) = 0 \,, \tag{3.57}$$

the expansion Θ is evaluated on a trial surface, then iterated using a Newton-Raphson method to solve $\mathbf{J} \cdot \delta\mathbf{h} = -\Theta$, where $\mathbf{J}$ is the Jacobian matrix.
In our parameter files we set the variable for the `AHFinderDirect` module as

```
ActiveThorns = "AHFinderDirect"
# This is later steered
AHFinderDirect::find_every = 0
AHFinderDirect::run_at_CCTK_POST_RECOVER_VARIABLES =
   no
AHFinderDirect::move_origins            = yes
AHFinderDirect::reshape_while_moving    = yes
AHFinderDirect::predict_origin_movement = yes
AHFinderDirect::geometry_interpolator_name = "Lagrange
    polynomial interpolation"
AHFinderDirect::geometry_interpolator_pars = "order=4"
AHFinderDirect::surface_interpolator_name  = "Lagrange
    polynomial interpolation"
AHFinderDirect::surface_interpolator_pars  = "order=4"
AHFinderDirect::output_h_every = 128
AHFinderDirect::N_horizons = 1
AHFinderDirect::which_surface_to_store_info
   [1] = 0
AHFinderDirect::reset_horizon_after_not_finding
   [1] = no
AHFinderDirect::initial_guess__coord_sphere__radius
   [1] = 1.3528

AHFinderDirect::track_origin_source_x[1] = "
   Hydro_Analysis::Hydro_Analysis_rho_max_loc[0]"
AHFinderDirect::track_origin_source_y[1] = "
   Hydro_Analysis::Hydro_Analysis_rho_max_loc[1]"
AHFinderDirect::track_origin_source_z[1] = "
   Hydro_Analysis::Hydro_Analysis_rho_max_loc[2]"

AHFinderDirect::track_origin_from_grid_scalar[2] = "
   yes"
```

```
AHFinderDirect::track_origin_source_x[2] = "
   Hydro_Analysis::Hydro_Analysis_rho_max_loc[0]"
AHFinderDirect::track_origin_source_y[2] = "
   Hydro_Analysis::Hydro_Analysis_rho_max_loc[1]"
AHFinderDirect::track_origin_source_z[2] = "
   Hydro_Analysis::Hydro_Analysis_rho_max_loc[2]"
```

3.2.8 Gravitational Waves extraction

The Einstein Toolkit includes modules for extracting gravitational waves via either the Moncrief formalism of a perturbation on a Schwarzschild background or the calculation of the Weyl scalar ψ_4. In this section, we briefly how the `WaveExtract` (based on the Regge-Wheeler-Zerilli theory) and `WeylScal4` (based on the Newman-Penrose ψ_4 scalar) modules work. For a more detailed description of the two methods, see [37] for the ψ_4-based method and [38] and [39] for the Regge-Wheeler-Zerilli theory of perturbations of the Schwarzschild spacetime in the gauge-invariant formulation by Moncrief [40].
We describe in Chapter 4 how we used these methods for our simulations.

3.3 LORENE

LORENE is a set of C++ classes to solve various problems arising in numerical relativity, and more generally in computational astrophysics. It provides tools to solve partial differential equations by means of multi-domain spectral methods. In particular, we use the LORENE code for the generation of the initial models we used in our simulations. In this section, we follow the description in [41] to show how the LORENE code works for generating initial models of irrotational binary neutron stars.

A quasiequilibrium binary neutron star configuration is obtained by specifying the equation of state for each star, the rotation state: either rigidly rotating or irrotational flow, the coordinate distance $d := |X_{\langle 2\rangle} - X_{\langle 1\rangle}|$ between the two stellar centers and the central enthalpies $H^c_{\langle 1\rangle}$ and $H^c_{\langle 2\rangle}$ in each star, or equivalently, the central density in each star.
Once we set the above parameters, we start by computing initial conditions for the iterative procedure. These initial conditions are constitued by two numerical solutions for spherically symmetric static isolated neutron stars, of respective enthalpy $H^c_{\langle 1\rangle}$ and $H^c_{\langle 2\rangle}$. $M_{\langle 1\rangle}$ and $M_{\langle 2\rangle}$ being the gravitational

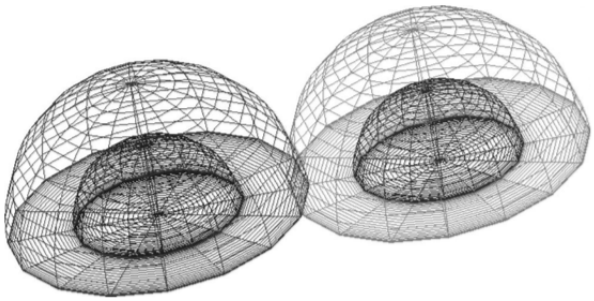

Figure 3.1: Domains used in the numerical computations.

masses of these spherical symmetric models, we set the X coordinates of the two stellar centers according to the Newton-like formulas

$$X_{\langle 1\rangle} = -\frac{M_{\langle 2\rangle}}{M_{\langle 1\rangle} + M_{\langle 2\rangle}} d \quad \text{and} \quad X_{\langle 2\rangle} = \frac{M_{\langle 1\rangle}}{M_{\langle 1\rangle} + M_{\langle 2\rangle}} d \,. \tag{3.58}$$

These coordinates will remain fixed during the iteration. Only the location of the rotation axis X_{rot}, initially set to 0, will change. The location of this origin is a priori arbitrary, only the distance d between the two stellar centers having a physical meaning; the setting (3.58) simply insures that this origin is not too far from the rotation axis.
The angular velocity Ω is initialized according to a formula for second order post-Newtonian spherical stars

$$\begin{aligned} \Omega^2_{ini} = \frac{M_{ini}}{d^3} \Bigg\{ 1 - \frac{M_{ini}}{d} \left[\frac{11}{4} + \frac{2R^2}{d^2}\gamma - \frac{12}{25}\frac{R^4}{d^4}\gamma^2 \right] + \\ + \left(\frac{M_{ini}}{d} \right)^2 \left[\frac{69}{8} + \frac{11}{4}\frac{R^2}{d^2}\gamma + \frac{17}{25}\frac{R^4}{d^4}\gamma^2 \right] \Bigg\} \,, \end{aligned} \tag{3.59}$$

where $M_{ini} := M_{\langle 1\rangle} + M_{\langle 1\rangle}$, R is the coordinate radius of one of the two stars (which is spherical initially) and $\gamma = \gamma_{irrot} := 0$ for irrotational binaries, as we set in our parameter files.
The metric autopotentials are initialized as follows: $\nu_{\langle a\rangle}$ and $\beta_{\langle a\rangle}$ are set to the values of ν and β for the static spherical models. The shift $N^i_{\langle a\rangle}$ is initialized to be

$$N^i_{\langle a\rangle} = \frac{7}{8} W^i_{\langle a\rangle} - \frac{1}{8} \left(\bar{\nabla}^i \chi_{\langle a\rangle} + \bar{\nabla}^i W^j_{\langle a\rangle} x_j \right), \quad a = 1, 2 \tag{3.60}$$

with $W^i_{\langle a\rangle}$ and $\chi_{\langle a\rangle}$ defined as in equations (93) and (94) of [41].

At a given step, we start by determining the value of the orbital angular velocity Ω and the value of the X coordinate of the rotation axis, X_{rot} by

taking the gradient along X of the first integral of motion demanding H to be maximal at the center of each star. This results in the two equations

$$\frac{\partial}{\partial X}\ln\Gamma_0|_{(X_{\langle a\rangle},0,0)} = \frac{\partial}{\partial X}\left(\nu+\ln\Gamma\right)|_{(X_{\langle a\rangle},0,0)} \quad a=1,2 \tag{3.61}$$

where Γ_0 can be expressed in terms of Ω and X_{rot}. It results in a system of two equations for the unknowns Ω and X_{rot} which can be solved by standard methods. The components of the orbiting velocity $\mathbf{U}_0$ can be computed in terms of Ω, X_{rot}, the lapse N and the components of the nonrotating-coordinates shift vector. From this, the Lorentz factor Γ_n and the fluid three-velocity $\mathbf{U}$ are computed.
Then the computational domanins are adapted as follows. The first integral of motion is written, following the splitting

$$H = H^c_{\langle a\rangle} + \nu^c_{\langle a\rangle} + \Phi^c_{\langle a\rangle,ext-\nu_{\langle a\rangle}-\Phi_{\langle a\rangle,ext}} \tag{3.62}$$

where Φ is an external potential defined as

$$\Phi_{\langle a\rangle,ext} := \nu_{\langle b\to a\rangle} - \ln\Gamma_0 + \ln\Gamma \tag{3.63}$$

Then, the auto-potential $\nu_{\langle a\rangle}$ si rescaled by a factor α^2 (as in Eq. (100) of [41]) to make sure that the enthalpy vanishes at the point $\theta_{\langle a\rangle}=\pi/2$, $\phi_{\langle a\rangle}=0$ on the extremal boundary of domain $\mathcal{D}^{\langle a\rangle}_{M_{\langle a\rangle}-1}$.
From this new value of H, the fluid proper baryon density n, proper energy density e and pressure p are computed and then we get the Eulerian energy density E and the trace of the stress-energy tensor S. These last quantities are subsequently used to evaluate the source terms of the elliptic equations for gravitational potentials.
Before the beginning of a new step, some relaxation is performed onto the enthalpy field and the autopotentials, according to

$$Q^J \leftarrow \lambda Q^J + (1-\lambda)Q^{J-1} \tag{3.64}$$

where Q stands for any of the fields H, $\nu_{\langle a\rangle}$, $\beta_{\langle a\rangle}$ and $N^i_{\langle a\rangle}$, J $(J-1)$ labels the current step (previous step), and λ is the relaxation factor.
To control the convergence of the iterative procedure, the relative difference between the enthalpy fields of two successive steps is used

$$\delta H := \frac{\sum_i \left|H^J(x_i) - H^{J-1}(x_i)\right|}{\sum_i |H^{J-1}(x_i)|}\,, \tag{3.65}$$

where the summation is extended to all the collocation points inside the star and J is the step label. Typically $\delta H = 10^{-7}$ is chosen to be the criterion to end the iteration. For very high precision calculations $\delta H = 10^{-12}$ is used instead.

4. Results and Conclusions

In this chapter we report the results of the simulations of the three binary neutron star systems with polytropic equation of state $P = K\rho^{\Gamma}$ and initial baryonic masses of $1.6M_{\odot}$, $1.5M_{\odot}$ and $1.55M_{\odot}$. We choose to set $\Gamma = 3.00$, $K = 80000$ and include an additional term $\Gamma_{th} = 1.8$ following the discussions in the previous chapters and in [10].
In the following table, we reported all the initial values of the three models we studied.

Model	$M_0^{(1)}$ $\left[M_{\odot}\right]$	$M_0^{(2)}$ $\left[M_{\odot}\right]$	$M^{(1)}$ $\left[M_{\odot}\right]$	$M^{(2)}$ $\left[M_{\odot}\right]$	Ω $\left[\frac{krad}{s}\right]$	M_{ADM} $\left[M_{\odot}\right]$	J $\left[\frac{GM_{\odot}^2}{c}\right]$
16vs16	1.60	1.60	1.36	1.36	2.121	2.81	7.635
155vs155	1.55	1.55	1.32	1.32	2.099	2.73	7.280
15vs15	1.50	1.50	1.28	1.28	2.076	2.66	6.928

Table 4.1: Properties of the initial irrotational BNS models simulated. All these models were generated using the public LORENE code [8]. The columns show, in this order, the baryonic masses of the two stars ($M_0^{(1)}$,$M_0^{(2)}$), their gravitational masses ($M^{(1)}$,$M^{(2)}$) at infinite separation, the initial rotational angular velocity (Ω), the total initial ADM mass (M_{ADM}) and the angular momentum (J).

4.1 Gravitational waves signal

The merger of binary neutron stars is associated with the emission of gravitational radiation, as explained in Section 1.3, and a consequent loss of energy and angular momentum associated to such emission. We birefly describe the method we used to extract the GW signal, following [10].
As we explained in Chapter 3, the module `WeylScal4` of the Einstein Toolkit calculates the Newman-Penrose ψ_4 which is decomposed in spin-weighted spherical harmonics of spin-weight $s = -2$ by the module `Multipole`:

$$\psi_4(t, r, \theta, \phi) = \sum_{l=2}^{\infty} \sum_{m=-l}^{l} \psi_4^{lm}(t, r)\, {}_{-2}Y_{lm}(\theta, \phi)\ . \tag{4.1}$$

ψ_4 is linked to the GW strain by

$$\psi_4 = \ddot{h}_+ - \ddot{h}_\times \ . \tag{4.2}$$

We therefore need to integrate each component of ψ_4^{lm} twice in time to get h^{lm}. We performed those integrals using a simple trapezoidal rule. We then performed a fit to a second order polynomial, as in [10], obtaining

$$\tilde{h}_{lm} = \tilde{h}_{lm}^{(0)} - Q_2 t^2 - Q_1 t - Q_0 \ . \tag{4.3}$$

Therefore, we proceeded to compute the h_{22} component of the GW signal, which provides the maximum signal.

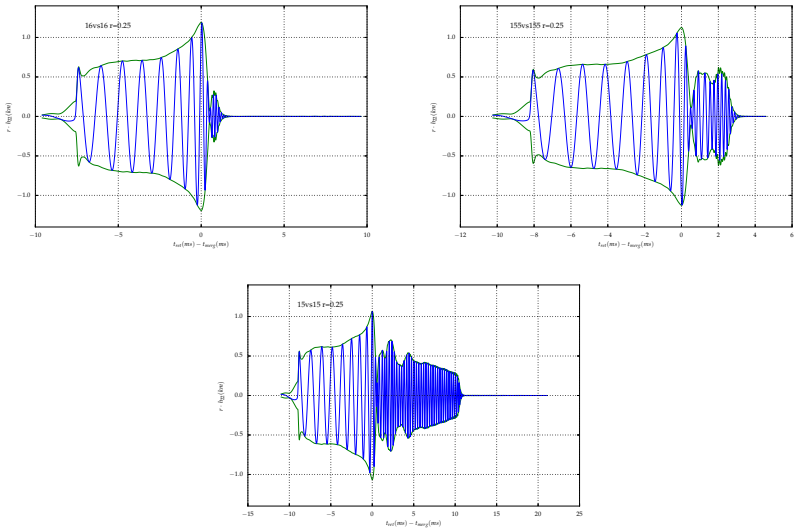

Figure 4.1: Overview of the h_{22} component for the three models at $dx = 0.25$ CU resolution. We show the envelop of the GW amplitude and its real part, multiplied by the distance of the observer to the origin. The observer has been chosen to be at $r = 400$ CU.

From figure 4.1 we can clearly see that the three models are different. The first two panels on the left refers to the model with both stars of gravitational mass $1.6M_\odot$, which presents a prompt collapse to a black hole after the merger. In effect, the gravitational wave signal emission immediately stops after the merger, when the black hole is formed. The panels referring to the $1.55M_\odot$ and $1.5M_\odot$ model respectively, show that the system does not collapse to a black hole after the merger. However, they merge into hyper-massive neutron star which then collapses to a black hole after a few milliseconds. We do not report the results here, but we oserved that lower resolution simulations do not show a collapse to a black hole for the 15vs15 model even 15 ms after the merger.

Model	$t_{merger}^{dx=0.25}$ (ms)	$t_{merger}^{dx=0.375}$ (ms)	$t_{merger}^{dx=0.5}$ (ms)
16vs16	7.48	6.66	5.51
155vs155	8.13	7.15	5.96
15vs15	8.88	7.79	6.38

Table 4.2: Computed merger time at different resolutions for the three models.

As explained in [10], smaller values of t_{merger} at lower resolutions should be expected due to the lower precision of the simulations.

Model	t_{BH} (ms)	$t_{BH} - t_{merger}$ (ms)
16vs16	8.33	0.85
155vs155	10.65	2.51
15vs15	19.32	10.44

Table 4.3: Time to collapse to a black hole. We show here both the time from the beginning of the simulation and the time from the merger to the final collapse. We consider only the simulations at $dx = 0.25$ CU.

In Tab. 4.1, we show that the model with mass $1.6M_{\odot}$ promptly collapses to a black hole, as the difference between the time of the merger and the time of the collapse is 0.85 ms. Both 15vs15 and 155vs155 models show a delayed collapse, with the first one collapsing 10.44 ms after the merger and the second after 2.51 ms.

4.2 Gravitational wave spectrum

As previously noted, the model 16vs16 results in a direct collapse and formation of a BH just after the merger. Models 155vs155 and 15vs15, instead, are characterized by a delayed collapse to a black hole a few milliseconds after the merger, as shown in Tab. 4.1.
The GW spectrum shows the presence of a dominating peak for the 15vs15 and 155vs155 models, while 16vs16 does not. As noted in [10], this peak is well known and corresponds to the frequency f_p (also called f_2 or f_{peak}) of the fundamental quadrupolar $m = 2$ oscillation mode of the bar-deformed massive neutron star formed after the merger.

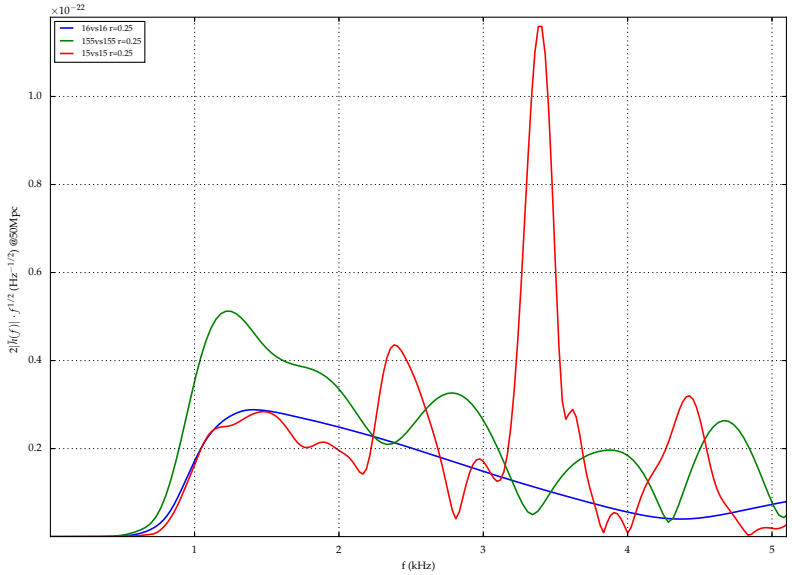

Figure 4.2: This figure shows the power spectral density (Fourier transform) of the effective GW signal $\bar{h}(f)$ in the optimal oriented case for a source at 50 Mpc.

Model	f_0 (kHz)	f_p (kHz)	f_- (kHz)	f_+ (kHz)
16vs16	1.40	–	–	–
155vs155	1.23	3.78	2.78	4.67
15vs15	1.48	3.39	2.38	4.43

Table 4.4: Main peak frequencies and damping times of the postmerger phase of the simulated models at $dx = 0.25$ CU.

Secondary postmerger peaks at frequencies f_- and f_+ are also present and recognizable in the spectrum and could be useful in extracting the neutron stars parameters from future gravitational waves detection. The physical origin of these secondary peaks is still debated (see [42, 43]), but none of the hypothesis have been confirmed yet.

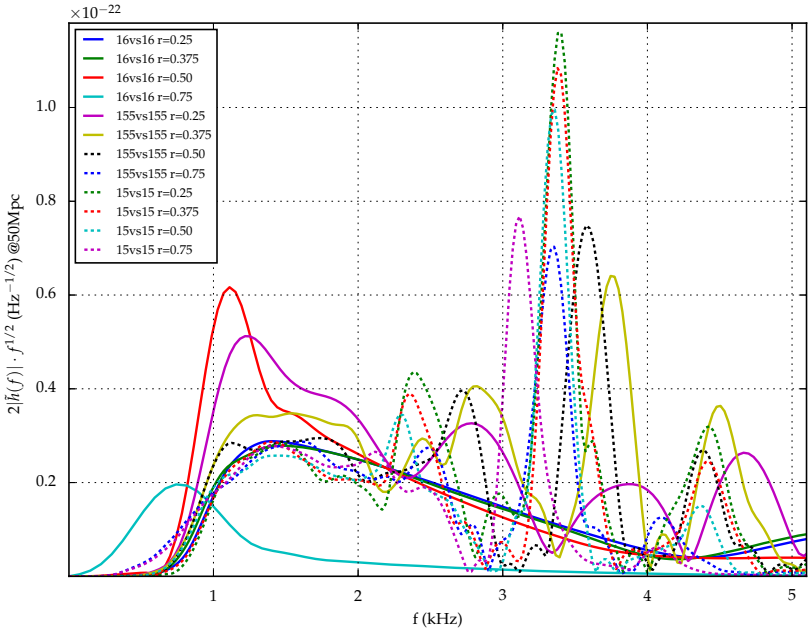

Figure 4.3: Overview of the power spectral density of the effective GW signal of all the models simulated.

In table 4.2, we report the frequencies of all clearly recognizable spectral peaks for the three models we considered. We also report the quasiradial oscillation frequency f_0. The models 15vs15 and 155vs155 show a dominant peak f_p at 3.39 kHz and 3.78 kHz respectively (see Fig. 4.2), while the 16vs16 model only show the peak of the quasiradial mode f_0, due to its prompt collapse to a black hole. The secondary peaks of the 15vs15 and 155vs155 models are close to the central f_p, with a maximum difference of the order of $\simeq 1$ kHz, as it has already been observed in [10]. Also, we observed that the central f_p peak of 155vs155 is smaller than the two secondary peaks. We think that this could be due to the fact that this model collapses to a black hole quickly, as it is shown in section 4.1.
We noted that there is no significant difference among the spectra at different resolution in terms of peaks frequencies, as it is shown in figure 4.3.

4.3 Energy-Angular Momentum budget

To characterize the dynamics of these configurations, we found it useful to analyze two quantities that can be easily extracted from simulation data: the mass and/or energy and the angular momentum.
We analyzed the mass-angular momentum (MJ) budget for the three systems

at $dx = 0.25$, which collapse to a black hole. The quantities we analyzed are the energy (E^{gw}) and angular momentum (J_z^{gw}) carried away by gravitational radiation and the mass (M_{BH}) and angular momentum (J_{BH}) of the black hole. These are computed by the module `QuasiLocalMeasures` of the Einstein Toolkit, which uses the isolated horizon formalism to compute them on the apparent horizon located by `AHFinderDirect` (see section 3.2.7 for details).

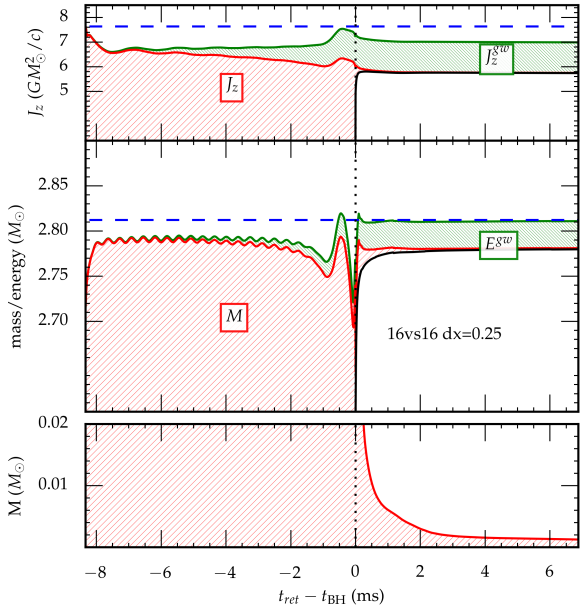

Figure 4.4: Angular momentum and mass budget for the evolution of model 16vs16. The blue, dashed, horizontal line indicates the ADM mass of the initial data, as calculated by Lorene. The red area (hatched from the bottom left to the top right) shows the contribution from matter left on the grid, while the green area, hatched from the top left to the bottom right, shows contributions from emitted GW, plotted in retarded time with regard to the position of the detector, assuming emission from the origin. The bottom-most, black, solid line indicates the angular momentum and mass of the BH.

For the 16vs16 model, which quickly collapses to a black hole, we found a mass of $\approx 2.78\ M_\odot$ and an angular momentum of $\approx 5.74\ GM_\odot^2/c$ while the total gravitational mass still present in the disk is only $0.001\ M_\odot$, 10 ms after the collapse.

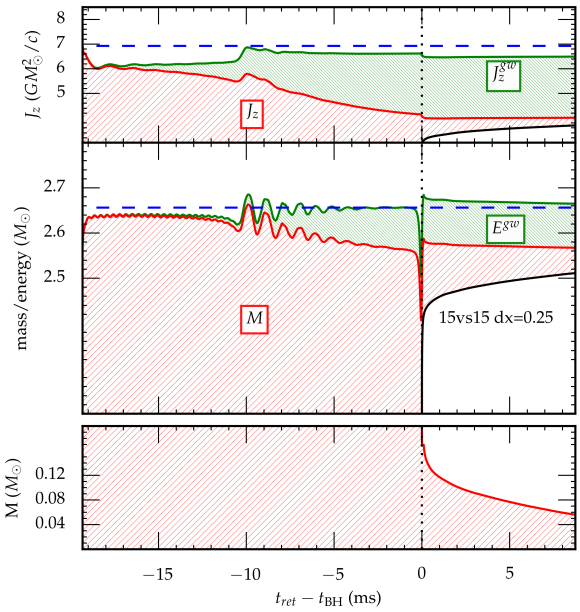

Figure 4.5: Angular momentum and mass budget for the evolution of model 15vs15. The quantities reported are the same as Figure 4.4

For the 15vs15 (see Fig. 4.5) model the mass of the black hole, 13 ms after the collapse, is $\approx 2.52\ M_\odot$ and its angular momentum $\approx 3.77\ GM_\odot^2/c$, while the mass still in the disk is $0.022 M_\odot$.

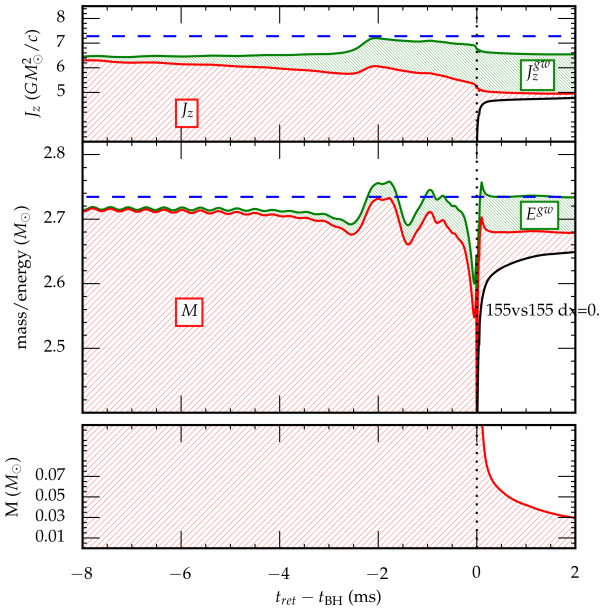

Figure 4.6: Angular momentum and mass budget for the evolution of model 155vs155. The quantities reported are the same as Figure 4.4

Model 155vs155 (see Fig. 4.6) shows a final black hole mass of $\approx 2.66\ M_{\odot}$ and angular momentum $\approx 4.82\ GM_{\odot}^2/c$, 4 ms after the merger, while the mass still in the disk is $0.043 M_{\odot}$. We note that the values of the mass of the remaining matter for 15vs15 and 155vs155 is only 10^{-2} times smaller than the actual black hole mass, so they are relatively high. This could mean that the remaining matter is still forming and driving an accretion disk. However, for 16vs16 the remaining matter mass is 10 smaller than in the other models, which means that the disk is collapsing to the black hole quicker.
We also note that the sum of the "red" and "green" contribution of Fig. 4.4, Fig. 4.5 and Fig. 4.6 are always close to the value of dashed blue line, which means that overall mass and angular momentum are conserved.

Appendices

A Lie derivative

Generically the "derivative" of some vector field V^μ on $\mathcal{M}$ is to be constructed for the variation δV^μ of V^μ between two neighbouring points p and q. Naively, one may write $\delta V^\mu = V^\mu(q) - V^\mu(p)$ but this is an ill-defined operation since the two vector fields belong to different vector spaces. For this reason, it's necessary to proceed in the definition of the derivative of a vector field by introducing another vector field U^μ, leading to the derivative of V^μ along U^μ which is the *Lie derivative*.
Consider a vector field U^μ on $\mathcal{M}$ and let be V^μ another vector field on $\mathcal{M}$. We can use U^μ to transport the vector V^μ from one point p to a neighbouring one q and then define rigorously the variation of V^μ as the difference between the actual value of V^μ at q and the transported value via U^μ. We first define the image $\Phi_\varepsilon(p)$ of the point p by the transport by an infinitesimal "distance" ε along field lines U^μ as $\Phi_\varepsilon(p) = q$, where q is the point close to p such that $\overrightarrow{pq} = \varepsilon U^\mu(p)$. Then if we multiply $V^\mu(p)$ by some infinitesimal parameter λ, it becomes an infinitesimal vector at p. Therefore, there exists a unique point p' close to p such that $\lambda V^\mu(p) = \overrightarrow{pp'}$. We may transport p' to q' along the field lines of U^μ by the same distance ε: $q' = \Phi_\varepsilon(p')$. We define the transport by the distance of the vector $V^\mu(p)$ along the field lines of U^μ according to

$$\Phi_\varepsilon\left(V^\mu(p)\right) := \frac{1}{\lambda}\overrightarrow{qq'} \ . \tag{.4}$$

We may then subtract it from the actual value of the field V^μ at q and define the *Lie derivative* of V^μ along U^μ by

$$\mathcal{L}_u V^\mu := \lim_{\varepsilon\to 0} \frac{1}{\varepsilon}\left[V^\mu(q) - \Phi_\varepsilon\left(V^\mu(p)\right)\right] \ . \tag{.5}$$

If we consider a coordinate system x^α adapted to the field U^μ in the sense that $U^\mu = (e_0)^\mu$ where $(e_0)^\mu$ is the first vector of the natural basis associated with the coordinates x^α, then the Lie derivative is simply given by the partial

derivative of the vector components with respect to x^0, while in an arbitrary coordinte system it is generalized to

$$\mathcal{L}_u V^\alpha = u^\mu \partial_\mu V^\alpha - v^\mu \partial_\mu u^\alpha \ . \tag{.6}$$

B Using LORENE

The procedure explained in needs 7 parameter files named **par_eos1.d**, **par_eos2.d**, **par_grid1.d**, **par_grid2.d**, **par_init.d**, **parcoal.d**. The first two files contain the informations about each neutron star EOS, then two of them set the grid points and size, while **par_init.d** contains the physical values for the initial conditions and **parcoal.d** the parameters for the equilibrium condition.
Once the code is compiled following the instructions in [8], we only need to use the class `bin_ns` to generate our initial models. This class includes the sources of the executables we need, namely `init_bin` and `coal`.

Now, from the terminal we need to go to the directory where we put our parameter files and then run `init_bin`. This executable will generate the `ini.d` file that is necessary to execute `coal`, which generates the initial models as explained in B. The file containing the data for the initial model will be named `resu.d`.

C How to set up the Einstein Toolkit

C.1 Introduction

This guide has the purpose to explain how to run a simulation of a binary neutron star system using the Einstein toolkit [5, 6]. The preliminary steps that include the preparation, download and configuration of the machine are all taken from the guide you can find on the Einstein toolkit wiki [44].

C.2 Prepare your machine

Before starting to talk about the Einstein toolkit and how to download it, your machine should have all the necessary packages in order to successfully complete the instructions you'll find in this guide. You can find the instructions to download and configure these packages and get ready to download

the Einstein toolkit in [44][1].

C.3 Download

First of all, you need to get a script called `GetComponents` to fetch the components of the Einstein toolkit. It uses both `svn` and `git` to download the codes that make up the Einstein toolkit from their repositories so it's the most convenient way method to complete this procedure and then keep it updated.
So, you may download it using

```
curl -kLO https://raw.githubusercontent.com/gridaphobe
   /CRL/ET_2015_11/GetComponents
```

for the last tested version released in November 2015 or

```
curl -kLO https://raw.githubusercontent.com/gridaphobe
   /CRL/master/GetComponents
```

for the development version. Then you have to make it executable:

```
chmod a+x GetComponents
```

`GetComponents` accepts a thorn list as argument. The thorn list contains the list of the codes, including the repository, details and destination folder, that the script `GetComponents` will download. We take a small part of the default thorn list `einsteintoolkit.th` (ET_2015_11) as example

```
# Component list for the Einstein Toolkit <http://
   einsteintoolkit.org/>
# $Revision$
# $Date$
# $HeadURL$

!CRL_VERSION = 1.0

!DEFINE ROOT = Cactus
!DEFINE ARR  = $ROOT/arrangements
!DEFINE COMPONENTLIST_TARGET = $ROOT/thornlists/

!DEFINE ET_RELEASE = ET_2015_11
```

[1]The instructions for the OS X environment with MacPorts also work for OS X 10.11 "El Capitan".

```
# CactusBase thorns
!TARGET   = $ARR
!TYPE     = git
!URL      = https://bitbucket.org/cactuscode/
   cactusbase.git
!REPO_PATH= $2
!REPO_BRANCH = $ET_RELEASE
!CHECKOUT =
CactusBase/Boundary
CactusBase/CartGrid3D
CactusBase/CoordBase
CactusBase/Fortran
CactusBase/InitBase
CactusBase/IOASCII
CactusBase/IOBasic
CactusBase/IOUtil
CactusBase/SymBase
CactusBase/Time
```

To check out the components of the Einstein Toolkit:

```
./GetComponents --parallel https://bitbucket.org/
   einsteintoolkit/manifest/raw/ET_2015_11/
   einsteintoolkit.th #for the 2015_11 version
```

```
./GetComponents --parallel https://bitbucket.org/
   einsteintoolkit/manifest/raw/master/einsteintoolkit
   .th #for the development version
```

The source tree is saved in a folder called `Cactus` for the default versions of the Einstein toolkit or the name defined with `!DEFINE ROOT` for custom thorn lists.

C.4 Configuration

Now you're ready to configure and build the codes on your machine. As explained in [44], each machine has to be treated in slightly different ways, but the general procedure is still the same.

To configurate your machine before building the codes, you have to edit

the `defs.local.ini`[2] file and fill all the required fields. Using the Simulation Factory:

```
cd Cactus
./simfactory/bin/sim setup
```

You'll need to enter your name, e-mail address, the folder where the source tree is and the default folder where all the simulations will be automatically saved. These two last fields already have their default options, so you can just press enter and skip them if you want to keep the default configurations. As explained before, each machine has to be treated differently so you can also add specific configurations to the setup, even if it's still not necessary at this stage.

```
./simfactory/bin/sim setup --optionlist /path/to/<
   optionlist>.cfg --runscript /path/to/<
   runscript_name> --thornlist /path/to/<thornlist>.th
```

C.5 Build

You may build the codes using two different methods: the Simulation Factory and the *classical* one.

Building with the Simulation Factory

Using the Simulation Factory:

```
./simfactory/bin/sim build <config-name> --<option-
   name> /path/to/<option> --thornlist /path/to/<
   thornlist>.th
```

To speed up the building process you can also specify how many parallel threads to open, e.g.

```
./simfactory/bin/sim build <config-name> --mdbkey '
   make -j 4' --<option-name> /path/to/<option> --
   thornlist /path/to/<thornlist>.th
```

That will build all the components included in the thorn list in a subfolder called `configs/<config-name>` that will be used by the executable file named `cactus_<config-name>` and saved in the subfolder `exe`.
In this guide we will not talk about the Simulation Factory anymore but you can check [45].

[2]placed in the `Cactus/simfactory/etc` folder.

Classical method

Alternatively, you can build the codes using the command `make`:

```
make -j 4 <config-name> <OPTIONS>=/path/to/<option-
   name> THORNLIST=/path/to/<thornlist-name>.th
```

The option `-j 4` will open 4 parallel processes and will considerably speed up all the operations.
All the compiled components will be in `configs/<config-name>` and the executable `cactus_<config-name>` in `exe`.

C.6 General procedure to run a simulation

Now you're ready to run a complete simulation using the executable created in the last passage. `exe/cactus_<config-name>` accepts a parameter file as argument.

```
./exe/cactus_<config-name> /path/to/<parameter-name>.
   par
```

The executable will open as many parallel processes possible but this could slow down the simulation so it's better specify how many OpenMP processes you want to open by using `OMP_NUM_THREADS=<number-of-threads>`.

```
OMP_NUM_THREADS=4 ./exe/cactus_<config-name> /path/to
   /<parameter-name>.par
```

In this case the simulation runs on 4 parallel threads.

C.7 Running a simulation for a binary neutron star system

Before running a complete simulation for a binary neutron star system, you need to set up your parameter file accordingly. The `.par` file has a precise structure that we're going to explain below.

Setting the Cactus parameters

First of all, you have to set the `Cactus` parameters for your simulation:

```
#--------------------------------------------
# Cactus parameters:
#--------------------------------------------
```

```
Cactus::cctk_run_title     = ``Meudon BNS''
Cactus::cctk_full_warnings = ``yes''
Cactus::highlight_warning_messages = ``yes''
```

You can also decide to end your simulation after a given time by setting

```
Cactus::terminate       = ``time''
Cactus::cctk_final_time = 10000.0
```

if you want to terminate after 10,000 minutes. Note that the parts parts following # are comments and are ignored by the code while executing.

Activate the thorns

Then you have to activate all the necessary thorns for your simulation. In paritcular, for a binary neutron star system you'll need to activate these thorns:

```
#-------------------------------------------------
# Activate all necessary thorns:
#-------------------------------------------------

ActiveThorns = ``Boundary CartGrid3D CoordBase Fortran
    InitBase IOUtil LocalReduce SymBase Time''
ActiveThorns = ``AEILocalInterp''
ActiveThorns = ``MoL Slab SpaceMask''
ActiveThorns = ``Carpet CarpetInterp CarpetInterp2
   CarpetIOASCII CarpetIOHDF5 CarpetIOScalar CarpetLib
    CarpetIOBasic CarpetReduce CarpetSlab CarpetMask
   LoopControl CarpetEvolutionMask''
ActiveThorns = ``NaNChecker TerminationTrigger
   TimerReport''
ActiveThorns = ``ADMbase ADMcoupling ADMmacros
   CoordGauge StaticConformal''
ActiveThorns = ``ReflectionSymmetry''
ActiveThorns = ``Constants TmunuBase HydroBase ``
ActiveThorns = ``EOS_Omni''
ActiveThorns = ``GRHydro''
ActiveThorns = ``SummationByParts''
ActiveThorns = ``GenericFD NewRad''
ActiveThorns = ``ML_BSSN ML_BSSN_Helper
   ML_ADMConstraints''
ActiveThorns = ``Dissipation''
```

```
ActiveThorns = ''GRHydro_Analysis TempPool''


ActiveThorns = ''SystemStatistics''


##------------------------------------
## Magentic Field initilal data
##------------------------------------
ActiveThorns = ''GRHydro_InitData''

# Wave extraction (Zerilli,Psi4 and QuasiLocalMeasure)
ActiveThorns = ''WaveExtract''
ActiveThorns = ''WeylScal4 Multipole''
ActiveThorns = ''SphericalSurface''
```

You have now to set up the grid and all the preliminar parameters of the configuration. In this guide, we'll skip the details but you may find a good example in [5].

Initial data and equation of state

An important part of the `.par` file is the definition of the model you want to use for your simulation and reading the initial data file. Activating the thorn `Meudon_Bin_NS` you'll be able to run simulations of binary neutron star systems and load initial data files. In this example, we generated the initial data using the public code LORENE [8], as you can see below (see the `.resu_d` file), and a polytropic (`2D_Polytrope`) law for the equation of state.

```
#-----------------------------------------------
# MODEL:
#-----------------------------------------------
ActiveThorns = ''Meudon_Bin_NS''
HydroBase::initial_hydro         = ''Meudon_Bin_NS''
ADMBase::initial_data            = ''Meudon_Bin_NS''
ADMBase::initial_lapse           = ''Meudon_Bin_NS''
ADMBase::initial_shift           = ''zero''
ADMBase::initial_dtlapse         = ''zero''
ADMBase::initial_dtshift         = ''zero''
Meudon_Bin_NS::filename = ''path/to/POLY_G300_14vs14.
   resu_d''
```

```
Meudon_Bin_NS::init_eos_table = ``2D_Polytrope''

#-------------------------------------------------
# M_ADM      2.8
# separation 40km
# K          80000
# Gamma      3.00
#-------------------------------------------------


EOS_Omni::poly_gamma                    = 3
EOS_Omni::poly_k                        = 80000
EOS_Omni::hybrid_gamma_th               = 1.8
EOS_Omni::n_pieces                      = 1
EOS_Omni::hybrid_gamma[0]               = 3
EOS_Omni::hybrid_k0                     = 80000
```

In this case both neutron stars will have a mass equal to 1.4 $M_\odot$, $\Gamma = 3.00$, an additional thermal contribution with $\Gamma_{th} = 1.8$ and their initial distance will be 40 km. You can also set the initial values for the magnetic field using `GRHydro_InitData` but we won't explain it in this guide.

Output files

As explained in section C.6, we won't run the simulation using the Simulation Factory. The output files won't be automatically saved in the `simulation` folder set in the `defs.local.ini` file, your data will be saved in a folder with the same name as the parameter file in your working folder instead. Then you may set the output subfolders for the simulation:

```
#-------------------------------------------------
# Output:
#-------------------------------------------------


##-------------------------------------------------
## Note that all the output form Multipole are
## saved in ``IO::out_dir''
##-------------------------------------------------
```

```
IO::out_dir = ``${parfile}/data/Multipole''

##-----------------------------------------------
##  Files created by
##    - 0D ASCII output created by CarpetIOASCII
##    - ASCII output created by CarpetIOScalar
##  have the seme format
##-----------------------------------------------
IOScalar::outScalar_dir  = ``${parfile}/data/Scalar''
CarpetIOASCII::out0D_dir = ``${parfile}/data/Scalar''

##-----------------------------------------------
##  Directory for HDF5 output
##-----------------------------------------------
IOHDF5::out_dir    = ``${parfile}/data/''
CarpetIOHDF5::out0D_dir = ``${parfile}/data/H5_0d''
CarpetIOHDF5::out2D_dir = ``${parfile}/data/H5_2d''
CarpetIOHDF5::out3D_dir = ``${parfile}/data/H5_3d''
```

If you want to display some of the parameters of your simulation:

```
IOBasic::outInfo_vars  = ``Carpet::
   physical_time_per_hour HydroBase::press{reductions
   ='maximum'} ML_BSSN::ML_Ham{reductions='maximum'}
   ADMBase::alp{reductions='minimum maximum'}''
```

To set what to save in the output folders:

```
IOScalar::outScalar_every      = 128
IOScalar::all_reductions_in_one_file = ``no''
IOScalar::one_file_per_group   = ``yes''
IOScalar::outScalar_reductions = ``minimum maximum
   average norm1 norm2''
IOScalar::outScalar_vars       = ``
 ADMBase::lapse
 ADMBase::shift
 ADMBase::metric
 HydroBase::rho
 HydroBase::vel
 HydroBase::eps
 HydroBase::press
 HydroBase::w_lorentz
 ML_BSSN::ML_Ham
```

```
ML_BSSN::ML_Mom
GRHydro_Analysis::GRHydro_CylDiagnostic{reductions='
   maximum'}
GRHydro_Analysis::GRHydro_Quadrupole{reductions='
   maximum'}
GRHydro_Analysis::GRHydro_DerivativeQuadrupole{
   reductions='maximum'}
  ''
```

This will save the parameters set in `IOScalar::outScalar_vars` into the `<parfile-name>/Scalar` subfolder every 128 iterations, as set in `IOScalar::outScalar_every`. The procedure for the other parameters is not really different so we won't report the details here.
The output files will be put in six different folders:

- `H5_0d` for scalar quantities saved in the `HDF5` format;
- `H5_2d` for 2D projections (`HDF5` format)
- `H5_3d` for the complete 3D output (`HDF5` format);
- `Multipole` for the components of the ψ_4 scalar (data from `WeylScal4`);
- `Scalar` for the quantities in the previous example;
- `WaveExtract` for the GW components in the Regge-Zerilli-Wheeler formalism (data from `WaveExtract`).

Checkpoints

If you have to submit your work to a queue, you can create checkpoint files in order to recover your simulation starting from that point.

```
#--------------------------------------------
# Checkpoint/Recovery:
#--------------------------------------------
IOHDF5::checkpoint                  = ''yes''
IO::checkpoint_dir                  = ${parfile}
IO::checkpoint_dir                  = ''../CHECKPOINTS
   ''
IO::checkpoint_ID                   = ''yes''
IO::checkpoint_every_walltime_hours = 12.0
IO::checkpoint_keep=2
IO::checkpoint_on_terminate         = ''yes''
```

```
IO::recover     = ``autoprobe''
IO::recover_dir = ${parfile}
IO::recover_dir = ``../CHECKPOINTS''
```

In this example, a checkpoint file will be created every 12 hours and saved in the HDF5 format.

Bibliography

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

[2] Aasi J. *et al.* (LIGO Scientific). Advanced ligo. *Classical and Quantum Gravity*, 32(7):074001, 2015.

[3] F Acernese *et al.* (Virgo). Advanced virgo: a second-generation interferometric gravitational wave detector. *Classical and Quantum Gravity*, 32(2):024001, 2015.

[4] J. Aasi, J. Abadie, B. P. Abbott, R. Abbott, T. D. Abbott, M. Abernathy, T. Accadia, F. Acernese, and *et al.* Parameter estimation for compact binary coalescence signals with the first generation gravitational-wave detector network. *Phys. Rev. D*, 88:062001, Sep 2013.

[5] Einstein toolkit: Open software for relativistic astrophysics url: `http://einsteintoolkit.org/`.

[6] Frank Löffler, Joshua Faber, Eloisa Bentivegna, Tanja Bode, Peter Diener, Roland Haas, Ian Hinder, Bruno C Mundim, Christian D Ott, Erik Schnetter, Gabrielle Allen, Manuela Campanelli, and Pablo Laguna. The einstein toolkit: a community computational infrastructure for relativistic astrophysics. *Classical and Quantum Gravity*, 29(11):115001, 2012.

[7] Philipp Mösta, Bruno C Mundim, Joshua A Faber, Roland Haas, Scott C Noble, Tanja Bode, Frank Löffler, Christian D Ott, Christian Reisswig, and Erik Schnetter. Grhydro: a new open-source general-relativistic magnetohydrodynamics code for the einstein toolkit. *Classical and Quantum Gravity*, 31(1):015005, 2014.

[8] Lorene: Langage objet pour la rélativité numérique url `http://www.lorene.obspm.fr/`.

[9] Eric Gourgoulhon, Philippe Grandclément, Keisuke Taniguchi, Jean-Alain Marck, and Silvano Bonazzola. Quasiequilibrium sequences of synchronized and irrotational binary neutron stars in general relativity: Method and tests. *Phys. Rev. D*, 63:064029, Feb 2001.

[10] Roberto De Pietri, Alessandra Feo, Francesco Maione, and Frank Löffler. Modeling equal and unequal mass binary neutron star mergers using public codes. *Phys. Rev. D*, 93:064047, Mar 2016.

[11] Wolfgang Kastaun and Filippo Galeazzi. Properties of hypermassive neutron stars formed in mergers of spinning binaries. *Phys. Rev. D*, 91:064027, Mar 2015.

[12] Francois Foucart, Roland Haas, Matthew D. Duez, Evan O'Connor, Christian D. Ott, Luke Roberts, Lawrence E. Kidder, Jonas Lippuner, Harald P. Pfeiffer, and Mark A. Scheel. Low mass binary neutron star mergers: Gravitational waves and neutrino emission. *Phys. Rev. D*, 93:044019, Feb 2016.

[13] Sebastiano Bernuzzi, David Radice, Christian D. Ott, Luke F. Roberts, Philipp Mösta, and Filippo Galeazzi. How loud are neutron star mergers? *Phys. Rev. D*, 94:024023, Jul 2016.

[14] Carlos Palenzuela, Steven L. Liebling, David Neilsen, Luis Lehner, O. L. Caballero, Evan O'Connor, and Matthew Anderson. Effects of the microphysical equation of state in the mergers of magnetized neutron stars with neutrino cooling. *Phys. Rev. D*, 92:044045, Aug 2015.

[15] Luciano Rezzolla, Bruno Giacomazzo, Luca Baiotti, Jonathan Granot, Chryssa Kouveliotou, and Miguel A. Aloy. The missing link: Merging neutron stars naturally produce jet-like structures and can power short gamma-ray bursts. *The Astrophysical Journal Letters*, 732(1):L6, 2011.

[16] Carlos Palenzuela, Luis Lehner, Marcelo Ponce, Steven L. Liebling, Matthew Anderson, David Neilsen, and Patrick Motl. Electromagnetic and gravitational outputs from binary-neutron-star coalescence. *Phys. Rev. Lett.*, 111:061105, Aug 2013.

[17] Kenta Kiuchi, Pablo Cerdá-Durán, Koutarou Kyutoku, Yuichiro Sekiguchi, and Masaru Shibata. Efficient magnetic-field amplification due to the kelvin-helmholtz instability in binary neutron star mergers. *Phys. Rev. D*, 92:124034, Dec 2015.

[18] Kyriaki Dionysopoulou, Daniela Alic, and Luciano Rezzolla. General-relativistic resistive-magnetohydrodynamic simulations of binary neutron stars. *Phys. Rev. D*, 92:084064, Oct 2015.

[19] Kenta Kiuchi, Yuichiro Sekiguchi, Koutarou Kyutoku, Masaru Shibata, Keisuke Taniguchi, and Tomohide Wada. High resolution magnetohydrodynamic simulation of black hole-neutron star merger: Mass ejection and short gamma ray bursts. *Phys. Rev. D*, 92:064034, Sep 2015.

[20] Sean M. Carroll. Lecture notes on general relativity. 1997.

[21] Thomas W. Baumgarte and Stuart L. Shapiro. Numerical relativity and compact binaries. *Physics Reports*, 376(2):41 – 131, 2003.

[22] Carles Bona, Joan Massó, Edward Seidel, and Joan Stela. New formalism for numerical relativity. *Phys. Rev. Lett.*, 75:600–603, Jul 1995.

[23] Miguel Alcubierre and Bernd Brügmann. Simple excision of a black hole in $3 + 1$ numerical relativity. *Phys. Rev. D*, 63:104006, Apr 2001.

[24] Miguel Alcubierre, Bernd Brügmann, Peter Diener, Michael Koppitz, Denis Pollney, Edward Seidel, and Ryoji Takahashi. Gauge conditions for long-term numerical black hole evolutions without excision. *Phys. Rev. D*, 67:084023, Apr 2003.

[25] José Ma. Martí, José Ma. Ibáñez, and Juan A. Miralles. Numerical relativistic hydrodynamics: Local characteristic approach. *Phys. Rev. D*, 43:3794–3801, Jun 1991.

[26] Francesc Banyuls, José A. Font, José Ma. Ibáñez, José Ma. Martí, and Juan A. Miralles. Numerical $3 + 1$ general relativistic hydrodynamics: A local characteristic approach. *The Astrophysical Journal*, 476(1):221, 1997.

[27] José A. Font. Numerical hydrodynamics and magnetohydrodynamics in general relativity. *Living Reviews in Relativity*, 11(7), 2008.

[28] Sascha Husa, Ian Hinder, and Christiane Lechner. Kranc: a mathematica package to generate numerical codes for tensorial evolution equations. *Computer Physics Communications*, 174(12):983 – 1004, 2006.

[29] Jr J W York. Kinematics and dynamics of general relativity. *Sources of Gravitational Radiation*, pages 83–126, 1979.

[30] Whisky – relativistic hydrodynamics and magnetohydrodynamics url: `http://www.whiskycode.org/`.

[31] Philipp Mösta, Bruno C Mundim, Joshua A Faber, Roland Haas, Scott C Noble, Tanja Bode, Frank Löffler, Christian D Ott, Christian Reisswig, and Erik Schnetter. Grhydro: a new open-source general-relativistic magnetohydrodynamics code for the einstein toolkit. *Classical and Quantum Gravity*, 31(1):015005, 2014.

[32] Chi-Wang Shu. *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, pages 325–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[33] Bernd Einfeldt. On godunov-type methods for gas dynamics. *SIAM Journal on Numerical Analysis*, 25(2):294–318, 1988.

[34] Grhydro documentation url: `http://einsteintoolkit.org/documentation/ThornDoc/EinsteinEvolve/GRHydro/`.

[35] A. Bauswein, H.-T. Janka, and R. Oechslin. Testing approximations of thermal effects in neutron star merger simulations. *Phys. Rev. D*, 82:084043, Oct 2010.

[36] Carsten Gundlach. Pseudospectral apparent horizon finders: An efficient new algorithm. *Phys. Rev. D*, 57:863–875, Jan 1998.

[37] Ezra Newman and Roger Penrose. An approach to gravitational radiation by a method of spin coefficients. *Journal of Mathematical Physics*, 3(3), 1962.

[38] Tullio Regge and John A. Wheeler. Stability of a schwarzschild singularity. *Phys. Rev.*, 108:1063–1069, Nov 1957.

[39] Frank J. Zerilli. Effective potential for even-parity regge-wheeler gravitational perturbation equations. *Phys. Rev. Lett.*, 24:737–738, Mar 1970.

[40] Vincent Moncrief. Gravitational perturbations of spherically symmetric systems. i. the exterior problem. *Annals of Physics*, 88(2):323 – 342, 1974.

[41] Eric Gourgoulhon, Philippe Grandclément, Keisuke Taniguchi, Jean-Alain Marck, and Silvano Bonazzola. Quasiequilibrium sequences of synchronized and irrotational binary neutron stars in general relativity: Method and tests. *Phys. Rev. D*, 63:064029, Feb 2001.

[42] Nikolaos Stergioulas, Andreas Bauswein, Kimon Zagkouris, and Hans-Thomas Janka. Gravitational waves and non-axisymmetric oscillation modes in mergers of compact object binaries. *Monthly Notices of the Royal Astronomical Society*, 418(1):427–436, 2011.

[43] A. Bauswein and N. Stergioulas. Unified picture of the post-merger dynamics and gravitational wave emission in neutron star mergers. *Phys. Rev. D*, 91:124056, Jun 2015.

[44] Einstein toolkit: Simplified tutorial for new users url: `https://docs.einsteintoolkit.org/et-docs/Simplified_Tutorial_for_New_Users`.

[45] Simulation factory: Advanced guide url: `https://docs.einsteintoolkit.org/et-docs/Simulation_Factory_Advanced_Tutorial`.