

ARISTOTLE UNIVERSITY OF THESSALONIKI
SCHOOL OF PHYSICS

DIPLOMA THESIS
MSc. IN COMPUTATIONAL PHYSICS

*Dynamics of Relativistic Stars with Tabulated
EOS in the Einstein Toolkit*

Author:
THEODOROS SOULTANIS

Advisor:
PROF. NIKOLAOS STERGIOULAS



September, 2018

Contents

1	Introduction	9
2	Preliminary concepts	11
2.1	The perfect fluid approximation	11
2.2	Thermodynamics	12
2.3	Barotropic flow	13
2.4	Adiabatic index	14
2.5	Rotating stars in general relativity	15
3	3+1 split of spacetime	17
3.1	The ADM formalism	17
3.1.1	Foliations of spacetime	17
3.1.2	Intrinsic curvature	20
3.1.3	ADM equations	21
3.2	The BSSNOK formalism	23
3.3	Fixing the gauges	24
3.3.1	Geodesic slicing	24
3.3.2	Maximal slicing	25
3.4	1+log slicing	25
3.5	Hyperbolic Γ -driver	26
4	Gravitational Waves	27
4.1	Linearized gravity	27
4.1.1	Vacuum solutions	28
4.1.2	Slow-motion sources	29
4.2	Newman-Penrose Scalars	32
4.2.1	Decomposition of Ψ_4	34
4.3	Associated physical quantities	35
5	The Code	37
5.1	McLachlan	37
5.2	GRHydro	39
5.2.1	Valencia formulation	40
5.2.2	Reconstruction	42
5.2.3	Reimann Solvers	43
5.2.4	Conservative to Primitive variable transformation	44
5.2.5	Artificial atmosphere	48
5.3	EOS_Omni	49
5.4	Hydro_RNSID	51
5.5	BNS Initial Models	52

6 Results	57
6.1 Oscillation modes	57
6.1.1 Piecewise polytropic EOS	57
6.1.2 Tabulated EOS	64
6.2 Binary neutron star mergers	72
6.3 Simulations for GW170817	74
7 Source code	83
8 Appendices	85
8.1 Reproducibility - HPC infrastructure	85
8.2 Lie derivatives	85
Bibliography	87

Abstract

Acknowledgements

Conventions

Chapter 1

Introduction

Chapter 2

Preliminary concepts

2.1 The perfect fluid approximation

In the study of neutron stars, matter is usually approximated by a *perfect fluid*. That is a model which describes a large assembly of particles, where its macroscopic mass distribution can be reasonably derived from a continuous energy density. The main assumption is that the collisions between the microscopic particles are frequent enough so the mean free path is short compared with the scale on which the density changes. Thus, thermodynamic equilibriums is enforced by the collisions. One assigns a mean velocity field u^μ and a mean stress-energy tensor $T^{\mu\nu}$ in fixed boxes, fluid elements, that are small compared to the macroscopic length scale but large compared to the mean free path. Moreover, on scales which are large compared to the size of the fluid elements, the 4-velocity and thermodynamic quantities have to be accurately described by continuous fields. As a result, an observer moving with the average velocity u^μ equal to that of the fluid will see collisions randomly distribute the nearby particle velocities, leading to an apparent locally isotropic particle distribution.

Let us assume a comoving observer who sees an isotropic particle distribution, with a 4-velocity u^μ , from which we can define the projection operator orthogonal to u^μ as

$$q^{\mu\nu} = g^{\mu\nu} + u^\mu u^\nu. \quad (2.1)$$

Next, we define the *stress energy tensor* (which is synonymous with energy momentum tensor well known from general relativity) $T^{\mu\nu}$ as

$$T^{\mu\nu} = \varepsilon u^\mu u^\nu + P q^{\mu\nu}, \quad (2.2)$$

where ε and P are the *total energy density* (or simply, the energy density) and the *pressure* as measured by the comoving observer with 4-velocity u^μ . To obtain (2.2) one takes advantage of the local isotropy and the fact the specific components of $T^{\mu\nu}$ must be invariant under rotations fixed to u^μ . It follows that the only non-zero parts of $T^{\mu\nu}$, being rotational scalars, are

$$\varepsilon = T^{\mu\nu} u_\mu u_\nu, \quad P = \frac{1}{3} q_{\mu\nu} T^{\mu\nu}. \quad (2.3)$$

Defining and orthonormal frame $\{\mathbf{e}_{\hat{0}}, \mathbf{e}_{\hat{1}}, \mathbf{e}_{\hat{2}}, \mathbf{e}_{\hat{3}}\}$, where $\mathbf{e}_{\hat{0}} = u^\mu$, it can be proven that the stress energy tensor (with respect to this frame) $T^{\hat{\mu}\hat{\nu}}$ reduces to the diagonal form

$$T^{\hat{\mu}\hat{\nu}} = \begin{pmatrix} \varepsilon & & & \\ & P & & \\ & & P & \\ & & & P \end{pmatrix}. \quad (2.4)$$

2.2 Thermodynamics

Next, we denote by n_B the baryon number density, from which we define the *rest mass density* ρ (baryon mass density) as

$$\rho \equiv m_B n, \quad (2.5)$$

where m_B is the mass per baryon. The latter definition originates from the fact that calculations involving microphysics are often carried out using the number density n_B instead of the rest mass density ρ . The properties of compact objects usually rely on numerous parameters, such as fluid and magnetic stresses, entropy gradients, composition, heat flow, and neutrino emission¹. A simple, yet common in the literature, case is the one of a perfect fluid in equilibrium composition, where the energy ε and pressure P depend on two parameters that can be taken to be the rest mass density ρ and *specific entropy* (entropy per unit rest mass) s as

$$\varepsilon = \varepsilon(\rho, s), \quad P = P(\rho, s). \quad (2.6)$$

Since we employ fluids to describe matter, the application of thermodynamics in this study is natural. The first law of thermodynamics expressed in terms of ρ and s is written as

$$d\varepsilon = \rho T ds + h d\rho, \quad (2.7)$$

where T stands for the *temperature* and h is the *specific enthalpy* (enthalpy per unit rest mass) defined as

$$h \equiv \frac{\varepsilon + P}{\rho}. \quad (2.8)$$

To obtain (2.7), one starts from the first thermodynamic law expressed in terms of extensive quantities, namely, energy E , entropy S , volume V , particle number N_i of species i , of a fluid element measured by the comoving observer, as

$$dE = T dS - P dV + \sum_i \mu_i dN_i, \quad (2.9)$$

where μ_i is the *chemical potential* of species i , defined with respect to the comoving observer. This form is rather general, and since we restrict our study in baryons we are allowed to replace the last term in (2.9) with μdN , denoting μ the baryon chemical potential and N the number of baryons. We define the rest mass $M_0 = m_B N$, from which we introduce the relations between ε , s , ρ as

$$\varepsilon = \frac{E}{V}, \quad s = \frac{S}{M_0}, \quad \rho = \frac{M_0}{V}. \quad (2.10)$$

Since in our study the number of baryons is conserved (at least in principle, omitting inconsistencies generated by numerical errors.), using $dM_0 = 0$ one replaces $\frac{dV}{V}$ with $-\frac{d\rho}{\rho}$. Thus, (2.7) is immediately obtained.

It is common to study the properties of the fluid through the *internal specific energy* (internal energy per unit rest mass) which is defined as

$$\varepsilon = \rho(1 + \epsilon). \quad (2.11)$$

¹Neutrino emission is rather crucial in deciphering the mechanism which drives core-collapse supernova explosion. It is considered a highly difficult task to tackle computationally and substantial effort is currently made worldwide. A relevant review article can be found in [?].

In terms of specific internal energy ϵ the specific enthalpy is written as

$$h = 1 + \epsilon + \frac{P}{\rho}. \quad (2.12)$$

One recovers the *Newtonian expression for the specific enthalpy* as

$$h_{\text{Newtonian}} = h - 1 = \epsilon + \frac{P}{\rho}. \quad (2.13)$$

2.3 Barotropic flow

In a *barotropic flow*, one deploys an one-parameter EOS for the description of the compact object. This decision is reasonably justified, at least for newborn neutron stars, due to that fact that within a short time after formation, the neutron star is cooled to $10^{10} K \simeq 1 MeV$ through neutrino emission. This energy scale is much smaller than the corresponding Fermi energy of the interior, in which the baryon density is greater than nuclear density (0.16 fm^{-3}) implying a Fermi energy greater than $E_F(0.16 \text{ fm}^{-3}) \simeq 60 \text{ MeV}$. Under these considerations a neutron star is cold, and, since nuclear reaction times are shorter than the cooling time, one can use a zero-temperature EOS to describe matter. In this context, it follows that

$$\epsilon = \epsilon(\rho), \quad P = P(\rho). \quad (2.14)$$

or equivalently,

$$P = P(\epsilon). \quad (2.15)$$

Using (2.7) and the definition (2.8), one can differentiate the latter as

$$\begin{aligned} dh &= d\left(\frac{\epsilon + P}{\rho}\right) = \frac{1}{\rho}d\epsilon + \frac{1}{\rho}dP - \frac{h}{\rho}d\rho \\ &= Tds + \frac{h}{\rho}d\rho + \frac{1}{\rho}dP - \frac{h}{\rho}d\rho \\ &= Tds + \frac{dP}{\rho} \\ \Rightarrow d\ln h &= \frac{T}{h}ds + \frac{dP}{\epsilon + P}, \end{aligned} \quad (2.16)$$

Since in a stationary, one-component perfect fluid, a one-parameter EOS, such as (2.15) is valid when the specific entropy s is constant throughout the star, implying $\nabla s = 0$, the barotropic one-component perfect fluid is also *homentropic*. Thus, due to $ds = 0$, (2.16) reduces to

$$d\ln h = \frac{dp}{\epsilon + P}, \quad (2.17)$$

which can be integrated to obtain the commonly called *log-enthalpy*

$$H(P) = d\ln h = \int_0^P \frac{dp'}{\epsilon + P'}. \quad (2.18)$$

with $\epsilon/\rho = 1$ at $\rho = 0$, since the gas is not relativistic at low densities. It should be noted that in our simulations, as it is shown in Ch. ??, we employ EOS which depend on ρ and ϵ , which we refer as *general EOS*, following the notation of the Einstein Toolkit. In the presence of shocks,

the local energy density rises ε to values higher than the ones predicted by the barotropic EOS $\varepsilon(\rho)$. These effects are the so called *thermal effects* and can be accounted for either by an EOS dependent on temperature T or specific internal energy ϵ . In general, these types of EOS depend on the electron fraction Y_e (fractional number density of electrons, properly defined at Sec. ??) too, and are represented by three-dimension tables of the form

$$Q = Q(\rho, Y_e, T) \quad \text{or} \quad Q = Q(\rho, Y_e, \epsilon), \quad (2.19)$$

where Q is the given thermodynamic quantity, such as the pressure p . A database with three-dimensional EOS can be found at [?]. Even though we do not consider these types of EOS in our study, it is surely one of our future goals.

2.4 Adiabatic index

As we mentioned in Sec. ??, under specific circumstances we are allowed to ignore the entropy gradients and assume a uniform specific entropy, leading to $ds = 0$. Thus one can identify the increase in pressure and density toward the star's core as an adiabatic one, ignoring the slow change in composition. From (2.7) follows that

$$d\varepsilon = \frac{\varepsilon + p}{p} d\rho, \quad (2.20)$$

which expressed in terms of the differential dp reduces to

$$\frac{\rho dp}{p d\rho} = \frac{\varepsilon + p}{p} \frac{dp}{d\varepsilon}. \quad (2.21)$$

The above expression serves as the two, equivalent definitions of the *adiabatic index* Γ_1 , representing the fractional change in pressure per fractional change in comoving volume at constant entropy. The more general expression, which assumes constant composition is

$$\Gamma_1 \equiv \frac{\partial \log p(\rho, s, Y_1, \dots, Y_N)}{\partial \log p} = \frac{\varepsilon + p}{p} \frac{\partial p(\varepsilon, s, Y_1, \dots, Y_N)}{\partial \log p}. \quad (2.22)$$

Here, Y_k stands for the fractional number density of the k^{th} -species of the constituent particles. It is defined as $Y_k = n_k/n_b$, where n_k is the number density of the k^{th} -species and n_b the total number density of baryon (for electrons, $Y_e = n_e/n_b$). In case of an ideal degenerate Fermi gas, for non-relativistic and ultra-relativistic matter, Γ_1 has constant values $5/3 \simeq 1.667$ and $4/3 \simeq 1.334$ respectively. The former case, corresponds to a pressure of the form $p \sim n^{5/3}$, whereas the latter case leads to a degeneracy pressure of order $p \sim n^4/3$. For this reason, models usually consider a constant effective adiabatic index, chosen to match an average stellar compressibility, even though the neutron star matter is an ideal Fermi gas only in the outer crust. These types of one-parameter EOS are called *polytropic*, and given by

$$p(rho) = K\rho^\Gamma, \quad (2.23)$$

where K and Γ are constants named *polytropic constant* and *polytropic exponent* respectively. Whilst this type of EOS is not realistic, its useful for testing codes. A more realistic description for the neutron star matter can be achieved by parametrizing EOS of cold matter, usually given in tabulated form since they are the result of elaborate calculations, with piecewise polytropic EOS.

2.5 Rotating stars in general relativity

In general relativity the geometry of a rotating star in equilibrium is described by a stationary, axis-symmetric spacetime, which is represented by two Killing vectors t^α and ϕ^α . In a *stationary spacetime* the metric is invariant under time translations generated by an asymptotically timelike symmetry vector t^α , while in an *axis-symmetric spacetime* the metric is invariant under rotations generated by the spacelike vector ϕ^α , which is attributed to circular orbits, an exception being the axis of symmetry, where $\phi^\alpha = 0$. The associated metric is subject to the following assumptions

- The spacetime is asymptotically flat.
- The time and axial symmetries of spacetime are represented in a mathematical point of view by a timelike symmetry Killing vector t^α and a spacelike rotational symmetry Killing vector ϕ^α .
- Commutation between the Killing vectors $[t^\alpha, \phi^\alpha] = 0$. and the existence of an isometry which reserves simultaneously the direction of t^α and ϕ^α ,

$$t^\alpha \rightarrow -t^\alpha, \quad \phi^\alpha \rightarrow -\phi^\alpha \quad (2.24)$$

Thus, an axis-symmetric rotating star in equilibrium is described by the following stationary metric $g_{\alpha\beta}$ which satisfies the aforementioned assumptions:

$$ds^2 = -e^{2\nu}dt^2 + e^{2\psi}(d\phi - \omega dt)^2 + e^{2\mu}(dr^2 + r^2d\theta^2), \quad (2.25)$$

where ν , ψ , ω , and μ are the four metric functions which are coordinate r , θ , dependent. As for the exterior vacuum, the number of the metric functions can be further reduced to three. It is rather convenient to write e^ψ in the form

$$e^\psi = r \sin \theta B e^{-\nu}, \quad (2.26)$$

B being a function of r and θ .

To simplify the problem and minimize the required algebra, a coordinate system has to be carefully chosen. A fine choice is the one where $x^0 = t$, and $x^3 = \phi$ such that $t^\alpha = \delta_0^\alpha$ and $\phi^\alpha = \delta_3^\alpha$ are vector fields. If, additionally the source satisfies the circularity condition, namely absence of meridional convective currents [?], the 2-surfaces orthogonal to t^α and ϕ^α , being surfaces of constant t and ϕ , can be described by x^1 , x^2 . It is common to choose x^1 , x^2 as quasi-isotropic coordinates, where $g_{r\theta} = 0$, $g_{\theta\theta} = r^2 g_{rr}$ (in spherical polar coordinates).

A star in the above consideration is attributed to a circular velocity field u^α expressed in terms of the two Killing vectors t^α and ϕ^α as

$$u^\alpha = u^t(t^\alpha + \Omega\phi^\alpha), \quad (2.27)$$

where from the normalization $u^\alpha u_\alpha = 1$ follows that

$$u^t = [-g_{\alpha\beta}(t^\alpha + \Omega\phi^\alpha)(t^\beta + \Omega\phi^\beta)]^{-1/2}. \quad (2.28)$$

This is the t component of u^α , $u^t = u^\alpha \nabla_\alpha t$, with respect to the previously chosen coordinates t and ϕ . In this context

$$\Omega \equiv \frac{u^\phi}{u^t} = \frac{d\phi}{dt}, \quad (2.29)$$

stands for the angular velocity of the fluid as seen by an observer at rest at infinity (or else, an asymptotic observer with 4-velocity along the asymptotically timelike killing vector t^α). A star is considered to be *uniformly rotating*, as seen by an observer at infinity, when Ω is constant.

Rotating relativistic stars in contrast to their non-rotating counterpart manifest two main effects, the first one being their shape which is flattened by centrifugal forces (an effect which appears at second order in rotation rate), while the second one is the dragging of the local inertial frames by the rotation of the source of gravitational field. Whilst the former effect also appears in the Newtonian approximation, the latter is purely relativistic. Particles dropped from infinity with zero angular momentum acquire a non-zero angular velocity in the direction of the star's rotation, and since inertial observers are free falling observers, the above naming is justified.

Let us now consider the conserved quantities which arise from the rotational symmetry, and therefore from the Killing vector ϕ^α . It can be proven that the quantity defined as the angular momentum $L = mu_\alpha\phi^\alpha$ of a free particle is conserved by ϕ^α . Thus, a particle with $L = 0$ satisfies $u_\alpha\phi^\alpha = u_\phi = 0$, since $\phi^\alpha = \delta_3^\alpha$. For our choice of metric (??), and coordinates,

$$\begin{aligned} u_\phi &= g_{\phi\phi}u^\phi + g_{\phi t}u^t \\ &= e^{2\psi}(u^\phi - \omega u^t), \end{aligned} \quad (2.30)$$

from which follows that

$$e^{2\psi}(u^\phi - \omega u^t) = 0, \quad (2.31)$$

implying the angular momentum velocity $d\phi/dt$ measured by an observer at infinity is given as

$$\frac{u^\phi}{u^t} = \omega. \quad (2.32)$$

Therefore, it follows that an initially (at infinity) radially infalling particle (with zero angular momentum) will acquire velocity as measured by an observer at infinity

$$\Omega = \omega, \quad (2.33)$$

while it preserves its zero angular momentum throughout its trajectory. Let us now define the local zero angular-momentum observers (ZAMOs) or *Eulerian* observers, whose worldlines are normal to $t = \text{const.}$ hypersurfaces. It follows from (2.33), that the metric function ω is the angular velocity the local Eulerian observer with respect to an observer at infinity. In the next chapter where we introduce the $3+1$ decomposition of spacetime we will return to the concept of the Eulerian observers, whereas for now, let us examine some of their properties.

Substitution of the metric components derived by the line element (??) to (2.28) gives

$$u^t = \frac{e^{-\nu}}{\sqrt{1 - (\Omega - \omega)^2 e^{2(\psi - \nu)}}}, \quad (2.34)$$

which for Eulerian observers reduces to $u^t = e^{-\nu}$, leading to the following form of circular velocity (2.27)

$$u_{\text{Eulerian}}^\alpha = e^{-\nu}(t^\alpha + \omega\phi^\alpha). \quad (2.35)$$

In this context, the metric function can be interpreted through the factor $e^{-\nu}$, formally called as the *time dilation factor*, which relates the proper time of the local Eulerian observer to coordinate time t (proper time at infinity). This can be seen from

$$u_{\text{Eulerian}}^t = e^{-\nu} \underbrace{(t^0 + \omega\phi^3)}_{1} = e^{-\nu} \Rightarrow dt = e^{-\nu}d\tau. \quad (2.36)$$

Eulerian observers are highly useful and are frequently used to describe the properties of the fluids. As we show in ??, the quantities related to the fluids are measured by Eulerian observers.

Chapter 3

3+1 split of spacetime

The Einstein equations describe the $4D$ geometry, and in their fully covariant way, treat time and space in equal footing. In this form, the Einstein equations cannot describe the dynamical evolution of the gravitational field parametrized in terms of the "time". There are situations, such as the numerical simulations of astrophysical systems like those we examine in this work, where we know the initial state of the given system and we are interested in the future evolution of the system. This type of problem is also referred in literature as a *Cauchy initial value problem (CIVP)*. To tackle this problem, we concentrate on the *3+1 formalism* formulated by Arnowitt, Deser, and Misner [?].

In this chapter, we first present the basic concepts of the ADM formalism, and the corresponding Einstein equations, namely the constraints and evolution equations. However, the ADM system, being only weakly hyperbolic, is not fit for numerical simulations. Thus, in the following section, we introduce a modified version of the Einstein equations, an evolution system which is strongly hyperbolic, the formalism developed by Baubgarte, Shapiro, Shibata, Nakamura, Oohara and Kojima (BSSNOK)[?]. In this work, we focus on the BSSNOK formalism since it is already implemented and tested in the Einstein Toolkit. Note that this section is heavily based on [?], hence, the reader is highly encouraged to refer to the original publication for more information.

3.1 The ADM formalism

3.1.1 Foliations of spacetime

For the purposes of the $3 + 1$ decomposition, let us assume a spacetime manifold \mathcal{M} , and the associated metric $g_{\mu\nu}$, abbreviated as $(\mathcal{M}, g_{\mu\nu})$. To cast the Einstein equations in the $3 + 1$ form we assume that the spacetime $(\mathcal{M}, g_{\mu\nu})$ is foliated into a family of non-intersecting, spacelike, three-dimensional hypersurfaces Σ_t parametrized by the function t (see Fig. 3.1 for an illustration). This parametrization holds since the hypersurfaces (timeslices) are, locally, level surfaces of a given scalar function t which we identify with the 'universal' time.

From t one obtains the 1-form $\nabla_\alpha t$, and therefore we define a timelike, unit, normal to the spatial hypersurfaces, vector as

$$n^\mu = -\alpha g^{\mu\nu} \nabla_\nu t, \quad (3.1)$$

where α is the so-called *lapse-function* which is a positive definite. For n^μ to be timelike and unit, it follows that

$$\alpha^{-2} = -g^{\mu\nu} \nabla_\mu t \nabla_\nu t. \quad (3.2)$$

As a result, it is obvious that n^μ is timelike and normalized as

$$n^\mu n_\mu = (-\alpha g^{\mu\kappa} \nabla_\kappa t) (-\alpha g_{\mu\lambda} \nabla^\lambda t) = \alpha^2 \delta_\lambda^\kappa \nabla_\kappa t \nabla^\lambda t = \alpha^2 \nabla_\kappa t \nabla^\kappa t = \alpha^2 g^{\mu\nu} \nabla_\mu t \nabla_\nu t = -1 \quad (3.3)$$

where we used the metric's properties to raise and lower indices in addition to relabeling summation indices. Due to this normalization, n^μ can be thought as the 4-velocity of *normal* or *Eulerian observer* whose worldline is always normal to the spatial slices Σ_t (moves along surfaces with $t = \text{const.}$). Note that the negative sign in (3.1) ensures that n^μ points to the direction of increasing t .

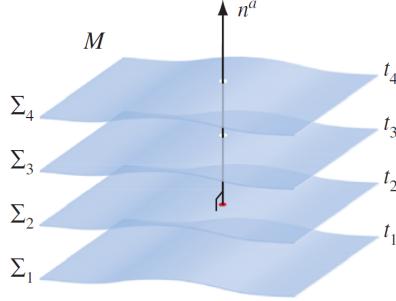


Figure 3.1: A depiction of the foliation of the spacetime \mathcal{M} in a family of non-intersecting the hypersurfaces as level-surfaces of the universal time t . The Eulerian observer n^α is orthogonal to these $t = \text{const.}$ spatial hypersurfaces. The figure is taken from [?].

From the normal vector, we proceed on constructing the induced by $g_{\mu\nu}$ spatial metric, also known as *intrinsic metric*, $\gamma_{\mu\nu}$ on the 3-dimensional hypersurfaces Σ_t , as

$$\gamma_{\mu\nu} = g_{\mu\nu} + n_\mu n_\nu, \quad (3.4)$$

One can see that $\gamma_{\mu\nu}$ is purely spatial (lives entirely in Σ_t without any component along n^μ) by contracting it with the normal n^μ ,

$$n^\mu \gamma_{\mu\nu} = n^\mu g_{\mu\nu} + \underbrace{n^\mu n_\mu}_{-1} n_\nu = 0. \quad (3.5)$$

Thus, $\gamma_{\mu\nu}$ is considered a tensor which projects out all the geometric objects lying along n^μ . The inverse spatial metric is trivially obtained by raising the indices of $\gamma_{\mu\nu}$ using $g^{\mu\nu}$,

$$\gamma^{\mu\nu} = g^{\mu\nu} + n^\mu n^\nu. \quad (3.6)$$

It follows that any 4-dimensional tensor can be decomposed into its purely spatial part, residing in Σ_t , and the timelike part, normal to the spatial surface. The former projection is carried out by contraction with the projection operator

$$\gamma^\mu_\nu = g^\mu_\nu + n^\mu n_\nu = \delta^\mu_\nu + n^\mu n_\nu, \quad (3.7)$$

whereas the latter projection comes from contraction with

$$\mathcal{N}^\mu_\nu = -n^\mu n_\nu, \quad (3.8)$$

where one can easily verify that $\mathcal{N}^\mu_\nu n^\nu = n^\mu$, which is expected since n^μ is purely timelike. In this context, we can define a 4-dimensional covariant derivative that maps spatial tensors into spatial tensors. To accomplish this, we project all indices of a 4-dimensional covariant derivative into the hypersurface Σ_t . It is uniquely defined if we require it to be compatible

with the intrinsic metric $\gamma_{\mu\nu}$. For a given scalar function f , and a rank $(1, 1)$ tensor T^μ_ν , reads respectively as

$$D_\mu f = \gamma^\nu_\mu \nabla_\nu f, \quad D_\mu T^\nu_\rho = \gamma^\sigma_\mu \gamma^\nu_\gamma \gamma^\delta_\rho \nabla_\sigma T^\gamma_\delta. \quad (3.9)$$

The compatibility of the 3-dimensional covariant derivative to the 3-metric follows from $D_\mu \gamma_{\nu\rho} = 0$. Furthermore, we are allowed to express the 3-dimensional covariant derivative in terms of the connection coefficients, which in a coordinate basis, are given by

$$\Gamma^\mu_{\nu\rho} = \frac{1}{2} \gamma^{\mu\sigma} (\partial_\rho \gamma_{\sigma\nu} + \partial_\nu \gamma_{\sigma\rho} - \partial_\sigma \gamma_{\nu\rho}). \quad (3.10)$$

Since the 4-dimensional Riemann tensor, for any vector v^μ , satisfies the relation

$$2\nabla_{[\mu} \nabla_{\nu]} v_\rho = v^\sigma ({}^4R^\sigma_{\rho\mu\nu}), \quad (3.11)$$

we can define the 3-dimensional analogue, namely, the 3-dimensional Riemann tensor associated with the intrinsic metric $\gamma_{\mu\nu}$, requiring that

$$2D_{[\mu} D_{\nu]} w_\rho = w_\sigma R^\sigma_{\rho\mu\nu}, \quad R^\sigma_{\rho\nu\mu} n_\sigma = 0, \quad (3.12)$$

for any spatial vector w_μ . The components of the Riemann tensor, in temrs of a coordinate system, read as

$$R^\mu_{\nu\rho\sigma} = \partial_\rho \Gamma^\mu_{\nu\sigma} - \partial_\sigma \Gamma^\mu_{\nu\rho} + \Gamma^\gamma_{\nu\sigma} \Gamma^\mu_{\gamma\rho} - \Gamma^\gamma_{\nu\rho} \Gamma^\mu_{\gamma\sigma}. \quad (3.13)$$

To obtain the 3-dimensional Ricci tensor, one contracts the Riemann tensor as $R_{\mu\nu} = R^\rho_{\mu\rho\nu}$, while the corresponding Ricci scalar is given as $R = R^\mu_\mu$.

Next, we introduce spatial coordinates on each hypersurface Σ_t such as $(x^i) = (x^1, x^2, x^3)$. Our aim is a coordinate system which varies smoothly between successive hypersurfaces. We can construct a time vector field t^μ (not be confused with the corresponding timelike Killing vector introduced in Ch. [?]) which is tangent to the lines of constant spatial coordinates, also known as *time lines*, as

$$t^\mu = \alpha n^\mu + \beta^\mu, \quad (3.14)$$

where $\beta^\mu = (0, \beta^i)$ is the spatial *shift vector* (Fig. 3.2 depicts these vectors). The time vector t^μ is mathematically preferable (since $t^\mu \nabla_\mu t = 1$), and the corresponding Lie derivative along its direction expresses the natural time derivative of our system. The time vector t^α connects points, having the same spatial coordinates, between successive hypersurfaces. It follows that the shift vector β^μ measures the shift of the spatial coordinates within a slice with respect to the normal (Eulerian) observer, or else

In addition to the Eulerian observers, whose trajectories move along $t = const.$ surfaces, the observers who follow the congruence of t^μ are called coordinate observers, while in the presence of matter, the latter moves independently of the coordinates with a 4-velocity u^μ .

We choose a coordinate basis, where the three spatial vectors $(e_{(1)}^\mu, e_{(2)}^\mu, e_{(3)}^\mu)$ reside entirely in the spatial hypersurface, while the forth basis vector is along the direction of time vector as $e_{(0)}^\mu = t^\mu$. In this basis, the normal vector takes the form

$$n_\mu = (-\alpha, 0, 0, 0), \quad n^\mu = \frac{1}{\alpha} (1, \beta^i), \quad (3.15)$$

and since $n^0 = 1/\alpha$ in (3.15), we obtain the relation

$$\frac{dt}{d\tau} = \frac{1}{\alpha} \Rightarrow d\tau = \alpha dt. \quad (3.16)$$

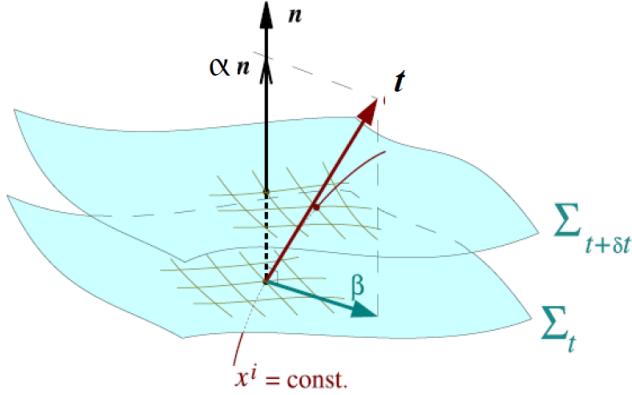


Figure 3.2: Coordinates (x^i) defined on hypersurfaces Σ_t , where lines of $x^i = \text{const.}$ cut through the foliation (Σ_t) defining in this way, the tangent vector t^μ and spatial shift vector β^μ . The figure is taken from [?].

Therefore, the lapse function measures the amount of proper time elapsed between neighbouring time slices along the Eulerian observer. At this point it useful to stress that the lapse function α and the shift vector β^i (since it is purely spatial we replaced the index with the one indicating spatial vectors) are four freely chosen gauge functions. The lapse function reflects the freedom in opting for the specific sequence of hypersurfaces, forwarding them by different amounts of proper time at different spatial points on a slice. The shift vector represents the freedom to relabel the spatial coordinates on each slice.

Furthermore, it follows that the 3-metric, using (3.4), is the spatial part of the 4-metric, as

$$\gamma_{ij} = g_{ij}, \quad (3.17)$$

and since the 0th components of any spatial contravariant tensor vanish, it follows that $\gamma^{\mu 0} = 0$. Thus, given the functions $\{\alpha, \beta^i, \gamma_{ij}\}$, the 4-metric can be reconstructed as

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij} (dx^i + \beta^i dt) (dx^j + \beta^j dt), \quad (3.18)$$

and its inverse

$$g^{\mu\nu} = \begin{pmatrix} -\alpha^{-2} & a^{-2}\beta^i \\ a^{-2}\beta^j & \gamma^{ij} - \alpha^{-2}\beta^i\beta^j \end{pmatrix}. \quad (3.19)$$

Lastly, since $\gamma_{ik}\gamma^{kj} = \delta^i_j$, the intrinsic metric can be used to raise and lower indices of purely spatial tensors, such as $\beta_i = \gamma_{ij}\beta^j$.

- γ_{0i} is not zero, thus I should not use the term "purely" spatial.

3.1.2 Extrinsic curvature

In the previous subsection, we introduced the associated 3-dimensional Riemann tensor, the Ricci tensor and scalar, as projections of their 4-dimensional counterparts on the spatial hypersurfaces. We did it in an attempt to express the Einstein equations of the 4-dimensional spacetime in terms 3-dimensional objects, however, these projections cannot possibly encapsulate all the information stored in the 4-dimensional objects ($R^\mu_{\nu\rho\sigma}$ being a purely spatial quantity, while ${}^{(4)}R^\mu_{\nu\rho\sigma}$ resides in spacetime). Whilst the 3-dimensional Riemann tensor stores information of the curvature intrinsic to the hypersurface Σ_t , it is inadequate to describe the shape of the hypersurface in the spacetime \mathcal{M} which is embedded in. This job is carried out

by the *extrinsic curvature* $K_{\mu\nu}$, which can be obtained through the projection of the gradients of the normal vector n^μ upon the spatial hypersurfaces Σ_t , as

$$K_{\mu\nu} = -\gamma_\mu^\sigma \gamma_\nu^\rho \nabla_{(\sigma} n_{\rho)}.$$
 (3.20)

This definition represents the negative expansion (see Fig .3.3), and as one easily sees, $K_{\mu\nu}$ is symmetrical with only spatial components. Basically, this quantity calculates the change in the pointing direction between normal vectors defined at nearby points across the hypersurface (these vectors are unit, hence, only differ upon their orientation). In other words, it measures the rate of deformation of the hypersurface.

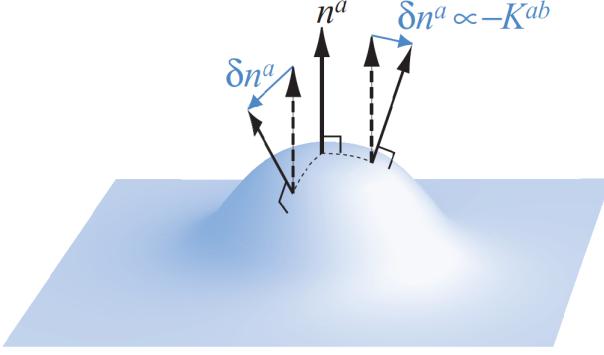


Figure 3.3: The figure illustrates the extrinsic curvature of a hypersurface in an enveloping spacetime. It measures the difference between normal vectors n^μ upon neighbouring points. The figure is taken from [?].

The trace of $K_{\mu\nu}$ is given by

$$K = g^{\mu\nu} K_{\mu\nu} = \gamma^{ij} K_{ij},$$
 (3.21)

where Latin indices were used since K_{ij} is purely spatial (whilst the contravariant metric γ^{ij} has only spatial components, the covariant metric γ_{ij} has not since $\gamma_{0i} = g_{0i} + n_0 n_i = \beta_i$). Alternatively, one can use the identity $n^\sigma \nabla_\mu n_\sigma = 0$, in conjunction with the definition of the Lie derivative, to manipulate (3.20) and ultimately obtain an expression of the extrinsic curvature $K_{\mu\nu}$ in terms of the Lie derivative \mathcal{L}_n along the normal vector n^μ of the 3-metric, as

$$K_{\mu\nu} = -\frac{1}{2} \mathcal{L}_n \gamma_{\mu\nu}.$$
 (3.22)

Thus, due to the normal vector n^μ , being timelike, (3.22) provides another interpretation for $K_{\mu\nu}$ as a generalization of the "time derivative" of the spatial metric γ_μ . However, since the coordinates change between neighbouring hypersurfaces, it suggests that the Lie derivative along the time vector t^μ , gives the natural time derivative $\mathcal{L}_t = \mathcal{L}_{\alpha n + \beta}$ for the evolution of the system.

3.1.3 ADM equations

The ADM equations are equivalent to the original Einstein equations, expressed in terms of the 3+1 quantities: lapse function α , spatial shift vector β^i , intrinsic 3-metric γ_{ij} , extrinsic curvature K_{ij} . The former two express our freedom in opting for a specific foliation (which is not unique), while the latter two describe the properties of spacetime. In order to obtain the ADM equations, one employs the *Gauss*, *Codazzi*, and *Ricci equations*, which arise from

the decomposition of the 4-dimensional Riemann tensor into spatial and normal pieces. Using those, from the 4-dimensional Einstein equations,

$${}^{(4)}G_{\mu\nu} = {}^{(4)}R_{\mu\nu} - \frac{1}{2} {}^{(4)}R g_{\mu\nu} = 8\pi T_{\mu\nu}, \quad (3.23)$$

through projections into the spatial hypersurface Σ_t , and normal direction n^μ , the constraint equations read as

$$\underbrace{R + K^2 - K_{ij}K^{ij}}_{\text{Hamiltonian constraint}} = 16\pi\rho, \quad \underbrace{D_j K_i^j - D_i K}_{\text{Momentum constraint}} = 8\pi S_i, \quad (3.24)$$

where ρ is the total energy density (we use this notation only in this chapter, in order to be compatible with the original publication, while in the later chapters we adopt the Einstein Toolkit's notation where ρ stands for the rest-mass density) as seen by the normal observer n^μ ,

$$\rho \equiv n_\mu n_\nu T^{\mu\nu}, \quad (3.25)$$

and the momentum density, projected into the spatial slice, S_μ as measured by the normal observer n^μ , given by

$$S_\mu = -\gamma^\nu_{\mu\nu} n^\rho T_{\nu\rho}. \quad (3.26)$$

Note that in (3.24) the summations are carried through spatial indices, since all the information of any spatial tensor is stored in their spatial components.

The constraint equations express the fact that the spatial metric γ_{ij} and the extrinsic curvature K_{ij} cannot be arbitrary, so the spatial hypersurfaces fit into the spacetime \mathcal{M} . In other words, the constraint equations need to be satisfied in each spatial slice, and fortunately they are, at least in the continuous case. In simulations, the numerical error due to the discretization, leads to them being approximately true. To assess codes, a common method is to monitor the constraint equations. Moreover, one can see that the constraint equations (3.24) do not rely on the lapse function and shift vector, and this is expected since these quantities determine how the coordinates differ between successive spatial slices, while the constraint equations state the conditions to be satisfied within the slices.

The evolution equations for γ_{ij} and K_{ij} can be obtained via the natural time derivative, namely, the Lie derivative \mathcal{L}_t along the direction of the time vector t^μ , the Ricci equation, and our choice of coordinate basis. The extrinsic curvature evolves as

$$\begin{aligned} \partial_t K_{ij} &= -D_i D_j \alpha + \alpha (R_{ij} - 2K_{ik}K_j^k + K K_{ij}) - 8\pi\alpha \left[S_{ij} - \frac{1}{2}\gamma_{ij}(S - \rho) \right] \\ &\quad + \beta^k D_k K_{ij} + K_{ik} D_j \beta^k + K_{kj} D_i \beta_i. \end{aligned} \quad (3.27)$$

where we introduced the *spatial stress-energy tensor* $S_{\mu\nu} = \gamma^\rho_\mu \gamma^\sigma_\nu T_{\rho\sigma}$ and its trace $S = S^\mu_\mu$. Lastly, the evolution of the spatial metric is given by

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i \beta_j + D_j \beta_i. \quad (3.28)$$

It is evident that the system of equations (3.24), (3.27), (3.28), do not provide any equations for the evolutions of the lapse function and shift vector. This is expected since, as we mentioned, these quantities are freely chosen gauge functions, and their evolutions are determined through gauge conditions which we impose. Different gauge conditions will, without a doubt, lead to different functions for γ_{ij} and K_{ij} during the evolution. Note that even though we do obtain analytical expressions for α and β^i , by comparison of (3.18) and the given spacetime metric $g_{\mu\nu}$, at least for the initial conditions, the evolutions of these functions is dictated by the corresponding gauge conditions.

3.2 The BSSNOK formalism

Even though the ADM formalism is rather elegant, the ADM equations do not behave well in numerical calculations. Their discretized version is only weakly-hyperbolic and inapplicable to numerical simulations due to severe instabilities which arise. The reason is the mixed second derivatives of the spatial metric hidden in the Ricci tensor, which render the system unable to be hyperbolic (the system cannot be expressed as wave equations for the components of the spatial metric). In an attempt to make the system hyperbolic, the BSSNOK formalism discards the mixed derivative terms in the Ricci tensor by the introduction of new auxiliary variables. The BSSNOK system is strongly hyperbolic (see []), and undeniably superior to the ADM system.

To obtain the new system, the spatial metric γ_{ij} becomes subject to *conformal transformations* (conformal quantities are denoted with tildes) with a corresponding *conformal factor* $\psi = e^\phi$. Introducing the *conformal metric* $\tilde{\gamma}_{ij}$, the following relation holds,

$$\tilde{\gamma}_{ij} = e^{-4\phi} \gamma_{ij}. \quad (3.29)$$

We also require equality between the conformal metric's determinant $\tilde{\gamma} = \det \tilde{\gamma}_{ij}$ and the flat metric's $\eta = \det \eta_{ij} = 1$ in the given coordinate system (Cartesian for our case), leading to $\phi = \ln \gamma / 12$. Next, it is imperative to split the trace from the extrinsic curvature, and subsequently conformally rescale the traceless part \tilde{A}_{ij} , using the same conformal factor as in the previous case,

$$\tilde{A}_{ij} = e^{-4\phi} \underbrace{\left(K_{ij} - \frac{1}{3} \gamma_{ij} K \right)}_{\text{traceless part}}. \quad (3.30)$$

(3.29) implies that

$$\begin{aligned} e^{4\phi} \tilde{\gamma}_{ij} &= \gamma_{ij} \Rightarrow \tilde{\gamma}_{ij} e^{4\phi} \gamma^{jk} = \gamma_{ij} \gamma^{jk} \\ &\Rightarrow \tilde{\gamma}_{ij} \tilde{\gamma}^{ik} = \delta_i^k, \end{aligned} \quad (3.31)$$

where $\tilde{\gamma}^{ij} = e^{4\phi} \gamma^{ij}$, in addition to $\tilde{\gamma} = 1$, indices of conformal quantities such as \tilde{A}_{ij} can be raised and lowered with $\tilde{\gamma}_{ij}$, hence, $\tilde{A}^{ij} = e^{4\phi} A^{ij}$. Thus, we obtain the evolution equations for K and ϕ through the traces of (3.28), (3.27) respectively, as

$$\partial_t K = -D^2 \alpha + \alpha \left(\tilde{A}_{ij} \tilde{A}^{ij} + \frac{1}{3} K^2 \right) + 4\pi \alpha (\rho + S) + \beta^i \partial_i K, \quad (3.32)$$

and

$$\partial_t \phi = \frac{1}{6} \alpha K + \beta^i \partial_i \phi + \frac{1}{6} \partial_i \beta^i. \quad (3.33)$$

We used the Hamiltonian constraint (3.24) to get rid of the Ricci scalar which arose from the contraction of the Ricci tensor, and the relation $\partial_t \ln \gamma^{1/2} = \alpha K + D_i \beta^i$ to deal with the determinant γ . As a result, subtraction of (3.32), (3.33) from the evolution equations (3.27), (3.28), leads to the traceless evolution equations for $\tilde{\gamma}_{ij}$ and \tilde{A}_{ij} as

$$\partial_t \tilde{\gamma}_{ij} = -2\alpha \tilde{A}_{ij} + \beta^k \partial_k \tilde{\gamma}_{ij} + \tilde{\gamma}_{ik} \partial_i \beta^k - \frac{2}{3} \tilde{\gamma}_{ij} \partial_k \beta^k, \quad (3.34)$$

and

$$\begin{aligned} \partial_t \tilde{A}_{ij} &= e^{-4\phi} \left[-(D_i D_j \alpha)^{TF} + \alpha (R_{ij}^{TF} - 8\pi S_{ij}^{TF}) \right] + \alpha \left(K \tilde{A}_{ij} - 2\tilde{A}_{ij} - 2\tilde{A}_{il} A^l_j \right) \\ &+ \beta^k \partial_k \tilde{A}_{ij} + \tilde{A}_{ik} \partial_j \beta^k + \tilde{A}_{kj} \partial_i \beta^k - \frac{2}{3} \tilde{A}_{ij} \partial_k \beta^k. \end{aligned} \quad (3.35)$$

The superscript TF in (3.35) represent the traceless part of a given tensor, such as $R_{ij}^{TF} = R_{ij} - R/3$. Next, we proceed on defining the so called "conformal connection functions", which prove to be crucial for eliminating the mixed term derivatives in the Ricci tensor, as

$$\tilde{\Gamma}^i \equiv \tilde{\gamma}^{jk}\tilde{\Gamma}_{jk} = -\partial_j\tilde{\gamma}^{ij}. \quad (3.36)$$

The quantities $\tilde{\Gamma}^i_{jk}$ are the analogue of (3.10), defined with respect to the conformal metric $\tilde{\gamma}_{ij}$ (the last equality follows from $\tilde{\gamma} = 1$). Therefore, the Ricci tensor takes the following form

$$\tilde{R}_{ij} = \frac{1}{2}\tilde{\gamma}^{lm}\partial_m\partial_l\tilde{\gamma}_{ij} + \tilde{\gamma}_{(ki}\partial_{j)}\tilde{\Gamma}^k + \tilde{\Gamma}^k\tilde{\Gamma}_{(ij)k} + \tilde{\gamma}^{lm}\left(2\tilde{\Gamma}^k_{l(i}\tilde{\Gamma}_{j)km} + \tilde{\Gamma}^k_{im}\tilde{\Gamma}_{klj}\right), \quad (3.37)$$

where $\tilde{R}_{ij} = R_{ij} - R_{ij}^\phi$, R_{ij}^ϕ being solely reliant on the conformal factor. In this form, the Laplace operator $\tilde{\gamma}^{lm}\partial_m\partial_l$ acting on $\tilde{\gamma}_{ij}$ is the only one which involves second derivatives, while all the others are absorbed in first derivatives of $\tilde{\Gamma}^i$. In this context, $\tilde{\Gamma}^i$ is considered an independent variable which means that its corresponding evolution equation has to be found. Taking the time derivative of (3.36), and subsequently interchanging the with the space derivatives we obtain

$$\partial_t\tilde{\Gamma}^i = \partial_j\left(2\alpha\tilde{A}^{ij} - 2\tilde{\gamma}^{m(j}\partial_m\beta^{i)} + \frac{2}{3}\tilde{\gamma}^{ij}\partial_l\beta^l + \partial^l\beta_l\tilde{\gamma}^{ij}\right). \quad (3.38)$$

Taking advantage of the momentum constraint (3.24) to eliminate the divergence of the extrinsic curvature, it reduces to

$$\begin{aligned} \partial_t\tilde{\Gamma}^i &= -2\tilde{A}^{ij}\partial_j\alpha + 2\alpha\left(\tilde{\Gamma}^i_{jk}\tilde{A}^{kj} - \frac{2}{3}\tilde{\gamma}^{ij}\partial_jK - 8\pi\tilde{\gamma}^{ij}S_j + 6\tilde{A}^{ij}\partial_j\phi\right) \\ &+ \beta^j\partial_j\tilde{\Gamma}^i - \tilde{\Gamma}^j\partial_j\beta^i + \frac{2}{3}\tilde{\Gamma}^i\partial_j\beta^j + \frac{1}{3}\tilde{\gamma}^{li}\partial_l\partial_j\beta^j + \tilde{\gamma}^{lj}\partial_j\partial_l\beta^i. \end{aligned} \quad (3.39)$$

Thus, the newly formed system of the evolution equations consists (3.32), (3.33), (3.34), (3.35), and (3.39). Furthermore, (3.36) serves as an additional constraint equations, since $\tilde{\Gamma}^i$ is treated as an independent variable.

3.3 Fixing the gauges

To close the system of the evolution equations, one has to choose proper gauge conditions in order to fix the gauge functions, the lapse function α and the shift vector β^i . There are several options available in the literature, each of them having advantages and disadvantages respectively. Nevertheless, there are certain attributes the gauge conditions should manifest:
a) the underlying symmetries of the system under consideration should be expressed (exactly or approximately) in the gauge condition, so that they are apparent through the evolution
b) in the presence of singularities in the spacetime, the gauges should hinder the slices from hitting them
c) the gauge conditions should be easy to implement and mathematically well-behaved (hyperbolic equations are preferable in numerical simulations)
d) the gauge conditions should be independent on the coordinate system.

3.3.1 Geodesic slicing

The simplest choice of gauge conditions corresponds to unit lapse function, and vanishing shift vector:

$$\alpha = 1, \quad \beta^i = 0. \quad (3.40)$$

Under this consideration, the proper time equals the coordinate time and the coordinates have zero velocity (with respect to the normal observer). Coordinate observers coincide with Eulerian observers since $t^\mu = \alpha n^\mu + \beta^\mu = n^\mu$. It can be proven that in this context, the Eulerian observers are free-falling, and therefore their worldlines are geodesics.

As a result, in geodesic slicing coordinate singularities tend to form rather quickly throughout the evolution. This is due to the fact that in the presence of gravitational sources the geodesics tend to focus, and therefore the coordinate observers approach each other, leading to the formation of a coordinate singularity.

3.3.2 Maximal slicing

Another common choice in numerical relativity is the so-called maximal slicing, where the hypersurfaces are chosen so that the trace of the extrinsic curvature vanishes for the initial data and the subsequent evolution. It is expressed via

$$K = 0, \quad \partial_t K = 0. \quad (3.41)$$

Maximal slicing, in contrast to the geodesic slicing, prevents the Eulerian observers from approaching each other. The hypersurfaces preserve their volume along the congruence of n^μ , and normal observers move as irrotational, incompressible fluid elements. The latter property prevents Eulerian observers from focusing, and as a result maximal slicing avoids coordinate singularities. Inserting the conditions (3.41) to (3.27), and taking the trace of the extrinsic curvature, we obtain the evolution equation for the lapse functions as

$$D^i D_i \alpha = \alpha [K^{ij} K_{ij} + 4\pi (\rho + S)]. \quad (3.42)$$

However, the latter is an elliptic equation for α and therefore is computationally expensive. Since it has to be solved numerically at iteration in order to remain in the maximal slicing gauge, it is not the appropriate choice for us.

3.4 1+log slicing

An alternative family of gauge conditions is derived from the harmonic gauge of spacetime coordinates. Let us first consider the 4-dimensional connection coefficients

$${}^{(4)}\Gamma^\mu = g^{\sigma\rho} {}^{(4)}\Gamma^\mu_{\sigma\rho}. \quad (3.43)$$

Setting these quantities equal to predetermined gauge source functions \mathcal{H}^μ , imposes gauge conditions. The simplest case of vanishing gauge source functions, ${}^{(4)}\Gamma^\mu = 0$, results to the coordinates being harmonic functions satisfying

$$\nabla^\nu \nabla_\nu x^\mu = 0. \quad (3.44)$$

A common alternative in numerical relativity is the *harmonic slicing*, where only the time component ${}^{(4)}\Gamma^0$ is set to zero. This choice in conjunction to the assumption of zero shift vector, results to

$$\partial_t \alpha = -\alpha^2 K. \quad (3.45)$$

A generalization of the harmonic slicing condition introduced by [?] suggests that the lapse function and shift vector can be dynamically found from simple evolution equations, of the form

$$\partial_t \alpha = -\alpha^2 f(\alpha) K, \quad (3.46)$$

where $f(\alpha)$ is an arbitrary positive function. For $f = 1$, (3.46) reduces to the harmonic slicing, while for $f = 0$ and initial $\alpha = 1$ reduces to the geodesic sling. Maximal slicing is obtained by assuming $f \rightarrow \infty$. For the choice of $f = 2/\alpha$, (3.46) provides an analytic expression for α through integration (assuming an integration constant equal to unity) as

$$\alpha = 1 + \ln \gamma. \quad (3.47)$$

The latter slicing is the so-called "1+log" condition is very robust in practice. Not only is it an algebraic slicing condition which is easy to implement and fast to solve, but also has a stronger singularity avoidance than the harmonic slicing. It is considered to "mimic" the maximal slicing, since f increases as the lapse function α decreases, hence, it behaves like the condition of the maximal slicing ($f \rightarrow \infty$). A non-zero shift vector, in addition to $f = 2/\alpha$, results to the generalized version of (3.46) which includes an advective shift term as

$$(\partial_t - \beta^j \partial_j) \alpha = -2\alpha K. \quad (3.48)$$

3.5 Hyperbolic Γ -driver

Nevertheless, maximal slicing gauge conditions can cause stretching of the grid near singularities, resulting to the black hole horizon to grow in coordinate space, and subsequently the crash of the simulation. A proper choice of shift vector, which pushes the coordinates outwards, can counter this phenomenon. From the definitions of the conformal connection functions (3.39), an elliptic condition for the shift vector can be obtained by setting $\partial_t \tilde{\Gamma}^i = 0$. This gauge-condition is called Γ -freezing and is closely connected to the minimal distortion shift condition. Since hyperbolic equations are preferable, one can reformulate the corresponding evolution equation for the shift vector to a hyperbolic Γ -driver equation as

$$\partial_t \beta^i - \beta^j \partial_j \beta^i = \frac{3}{4} \alpha B^i, \quad (3.49)$$

$$\partial_t B^i - \beta^j \partial_j B^i = \partial_t \tilde{\Gamma}^i - \beta^j \partial_j \tilde{\Gamma}^i - \eta B^i, \quad (3.50)$$

where the factor $3/4$ is chosen rather arbitrarily but is good for numerical results, η stands for the damping parameter which is of the order of $1/2M$ with M measuring the total mass of spacetime. The hyperbolic Γ -driver (3.50), also referred as "shifting shift", manifests the property of advection of unstable 'gauge' modes off the computational grid. An alternative version of (3.50), called "non-shifting shift" replaces the advective derivatives $\partial_t - \beta^j \partial_j$ with ∂_t . In our code we employ the advective "1+log" slicing condition (3.48), in combination with the "shifting shift" hyperbolic Γ -driver condition (3.50).

Chapter 4

Gravitational Waves

4.1 Linearized gravity

Einstein's equations, in the majority of cases, cannot be solved analytically, and for this reason we deploy computers to derive numerical solutions. Nevertheless, analytical solutions can be obtained by taking advantage of the symmetries of the physical system in examination. Moreover, it is often interesting to examine the behaviour of known solutions under the effects of small perturbations. Thus, gravitational waves can be viewed as small perturbations over a fixed background spacetime. In case of weak gravitational fields, one expects it to resemble the Minkowski flat spacetime, therefore, perturbations of the following form

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}, \quad |h_{\mu\nu}| \ll 1, \quad (4.1)$$

where $\eta_{\mu\nu}$ is the flat Minkowski metric, and $h_{\mu\nu}$ stands for the small perturbations, are referred as *linearized theory of gravity*. This is justified, since $|h_{\mu\nu}| \ll 1$, terms of higher orders of $h_{\mu\nu}$ (and its derivatives) are neglected. By the assumptions of small perturbations, the inverse of the physical metric $g_{\mu\nu}$ is given by

$$g^{\mu\nu} = \eta^{\mu\nu} - h^{\mu\nu}, \quad (4.2)$$

where we used $\eta_{\mu\nu}$ to raise and lower $h_{\mu\nu}$, and discarded higher order products of $h_{\mu\nu}$. Subsequently, one can calculate the Cristoffel symbols of the perturbed metric

$$\begin{aligned} \Gamma^\mu_{\nu\lambda} &= \frac{1}{2} g^{\mu\rho} (\partial_\lambda g_{\rho\nu} + \partial_\nu g_{\lambda\rho} - \partial_\rho g_{\nu\lambda}) \\ &= \frac{1}{2} \eta^{\mu\rho} (\partial_\lambda h_{\rho\nu} + \partial_\nu h_{\lambda\rho} - \partial_\rho h_{\nu\lambda}) \\ &= \frac{1}{2} (\partial_\lambda h^\mu_\nu + \partial_\nu h^\mu_\lambda - \partial^\mu h_{\nu\lambda}), \end{aligned} \quad (4.3)$$

and therefore, the Riemann tensor takes the following form

$$\begin{aligned} R^\mu_{\nu\lambda\rho} &= \partial_\lambda \Gamma^\mu_{\nu\rho} - \partial_\rho \Gamma^\mu_{\nu\lambda} \\ &= \frac{1}{2} (\partial_\lambda \partial_\nu h^\mu_\rho + \partial_\rho \partial^\mu h_{\nu\lambda} - \partial_\lambda \partial^\mu h_{\nu\rho} - \partial_\rho \partial_\nu h^\mu_\lambda). \end{aligned} \quad (4.4)$$

As a result, one can derive the Ricci tensor through a contraction over μ and λ of (4.4) as

$$R_{\mu\nu} = R^\lambda_{\mu\lambda\nu} = \frac{1}{2} (\partial_\lambda \partial_\nu h^\lambda_\mu + \partial^\lambda \partial_\mu h_{\nu\lambda} - \square h_{\mu\nu} - \partial_\mu \partial_\nu h), \quad (4.5)$$

where the trace of the perturbation metric is defined as $h = \eta^{\mu\nu}h_{\mu\nu} = h^\mu_\mu$, and $\square = \partial_\lambda^\lambda = \nabla^2 - \partial_t^2$ stands for the wave operator. The Ricci scalar, the curvature scalar, arises from the contraction of (4.5) as

$$R = R_\mu^\mu = (\partial_\lambda\partial^\mu h^\lambda_\mu - \square h). \quad (4.6)$$

Finally, having all the required components, we are able to derive the expression the Einstein tensor

$$\begin{aligned} G_{\mu\nu} &= R_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}R \\ &= \frac{1}{2}(\partial_\lambda\partial_\nu h^\lambda_\mu + \partial^\lambda\partial_\mu h_{\nu\lambda} - \square h_{\mu\nu} \\ &\quad - \partial_\mu\partial_\nu h - \eta_{\mu\nu}\partial_\lambda\partial^\rho h^\lambda_\rho + \eta_{\mu\nu}\square h). \end{aligned} \quad (4.7)$$

Thus, the field equations in the linearized system hold as $G_{\mu\nu} = 8\pi T_{\mu\nu}$, where the stress-energy tensor $T_{\mu\nu}$ is computed up to the first order in $h_{\mu\nu}$, and the conservation law is simply $\partial_\mu T^{\mu\nu} = 0$. We can further simplify (4.7), by the defining the so-called *trace-reversed* perturbation metric as

$$\bar{h}_{\mu\nu} = h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h, \quad \bar{h}_\mu^\mu = -h, \quad (4.8)$$

and subsequently substituting it to (4.7), which results to

$$G_{\mu\nu} = \frac{1}{2}(\partial_\lambda\partial_\nu\bar{h}^\lambda_\mu + \partial^\lambda\partial_\mu\bar{h}_{\nu\lambda} - \square\bar{h}_{\mu\nu} - \eta_{\mu\nu}\partial_\lambda\partial^\rho\bar{h}^\lambda_\rho). \quad (4.9)$$

However, in the linearized equations, the perturbations $h_{\mu\nu}$ are not uniquely determined since changes on the coordinate system may result to different perturbations $h'_{\mu\nu}$. This phenomenon is commonly referred as *gauge invariance* under gauge transformations. To depict this, we assume an infinitesimal coordinate transformation

$$x'^\mu = x^\mu + \xi^\mu, \quad (4.10)$$

for which, it can be proven that, up to first order in ξ^α , the metric perturbation varies as

$$h'_{\mu\nu} = h_{\mu\nu} - 2\nabla_{(\mu}\xi_{\nu)}, \quad \bar{h}'_{\mu\nu} = \bar{h}_{\mu\nu} - 2\nabla_{(\mu}\xi_{\nu)} + \eta_{\mu\nu}\nabla_\alpha\xi^\alpha. \quad (4.11)$$

As a result, we are free to choose any of the above coordinate transformations, and the underlying physics remain the same. To fix the gauge invariance, we impose the so-called *Lorentz gauge-condition* defined as

$$\partial^\mu\bar{h}_{\mu\nu} = 0, \quad (4.12)$$

which reduces the expression of the Einstein tensor (4.9) to the rather elegant form of

$$G_{\mu\nu} = -\frac{1}{2}\square\bar{h}_{\mu\nu}. \quad (4.13)$$

4.1.1 Vacuum solutions

In an attempt to get intuitive understanding of the gravitational radiation, we first examine the problem in absence of matter, namely, the vacuum solutions where the stress energy tensor $T_{\mu\nu}$ vanishes. Thus, (4.13) reduces to

$$\square\bar{h}_{\mu\nu} = 0. \quad (4.14)$$

Since (4.14) is an ordinary wave equation, its most general solution is a superposition of plane waves such as

$$\bar{h}_{\mu\nu}(\mathbf{x}, t) = \mathcal{R} \int d^3k A_{\mu\nu}(\mathbf{k}) e^{i(\mathbf{k}\cdot\mathbf{x}-\omega t)}, \quad (4.15)$$

where we defined the wave vector $k^\mu = (\omega, \mathbf{k})$, for which $\omega = |\mathbf{k}|$, and $A_{\mu\nu}(\mathbf{k})$ is constant symmetric tensor. These solutions are interpreted as gravitational waves propagating with the speed of light. From the Lorentz condition, the following constraint arises,

$$k^\mu A_{\mu\nu} = 0, \quad (4.16)$$

which reduces the number of independent variables from ten to six. Lastly, for asymptotically flat spacetimes we are allowed to additionally confine the gauges so that time perturbations vanish, as

$$h_{tt} = h_{ti} = 0. \quad (4.17)$$

and the perturbations to be traceless

$$h = h_i^i = 0. \quad (4.18)$$

Thus the Lorentz conditions reads as

$$\partial_i h_{ij} = 0, \quad (4.19)$$

which leads to transverse metric perturbations. Thus, the gauge is now completely fixed, while this gauge is usually referred to literature as *transverse-traceless (TT) gauge*. The perturbation metric $h_{\mu\nu}^{TT}$, in TT gauge, contains solely the physical information of the system, while it is free of any gauge choices.

As an example, one should examine the simple plane wave solution which propagates along the z -axis,

$$h_{ij}^{TT} = A_{ij}(\mathbf{k}) e^{+i(\mathbf{k}\cdot\mathbf{x}-\omega t)}, \quad (4.20)$$

where the wave vector is given as

$$k^\mu = (\omega, 0, 0, k^3) = (\omega, 0, 0, \omega). \quad (4.21)$$

It follows that $h_{ij}^{TT} = h_{ij}^{TT}(t - z)$, and as a result, the Lorentz condition (4.19), reduces to $\partial_z h_{zj}^{TT} = 0$, which using (4.20) implies that $h_{zj}^{TT} = \text{const} = 0$. Thus, there are only two independent components, hence, only two polarization states, namely,

$$h_+ = h_{xx}^{TT} = -h_{yy}^{TT}, \quad h_\times = h_{xy}^{TT} = -h_{yx}^{TT}. \quad (4.22)$$

h_+ and h_\times are the only two true radiative degrees of freedom, which are shown in Fig. 4.1, while linear polarizations are obtained when h_+ and h_\times are in phase, and circular or elliptic polarizations can be obtained when h_+ and h_\times are in phase shifted by $\pi/2$.

4.1.2 Slow-motion sources

Gravitational waves are generated in the presence of matter sources, which in the linearized gravity are represented by the stress-energy tensor $T_{\mu\nu}$, as the right hand side terms in

$$\square \bar{h}_{\mu\nu} = -16\pi T_{\mu\nu}, \quad (4.23)$$

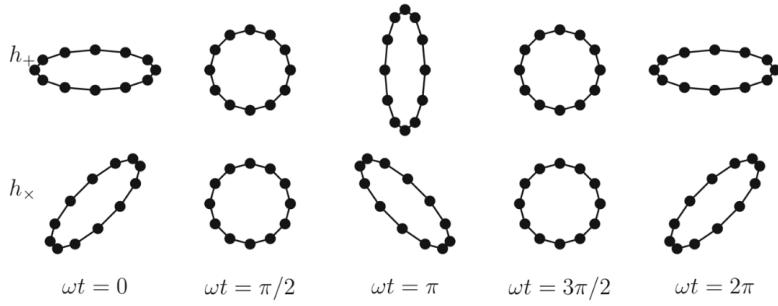


Figure 4.1: To be added.

which is derived from (4.13) by simply replacing the Einstein tensor using $G_{\mu\nu} = 8\pi T_{\mu\nu}$. The above equation (4.23) can be solved through Green's functions. Let us assume a wave equation under the influence of sources of the form

$$\square f(t, \mathbf{x}) = s(t, \mathbf{x}), \quad (4.24)$$

where in our case, $f(t, \mathbf{x})$ stands for the radiative field, which is reliant on time t and position \mathbf{x} , and source function is given by $s(t, \mathbf{x})$. The Green's function $G(t, \mathbf{x}; t', \mathbf{x}')$ is defined as the generated field from a delta function source, and gives the value of the field point (t, \mathbf{x}) per unit source at the source point (t', \mathbf{x}') as

$$\square G(t, \mathbf{x}; t', \mathbf{x}') = \delta(t - t')\delta(\mathbf{x} - \mathbf{x}'). \quad (4.25)$$

Thus the field generated by the actual source is given by the integration of Green's function against $s(t, \mathbf{x})$ as

$$f(t, \mathbf{x}) = \int dt' d^3x' G(t, \mathbf{x}; t', \mathbf{x}') s(t', \mathbf{x}'). \quad (4.26)$$

Fortunately, the literature provides several Green's functions for a variety of well known differential operators, and amongst them, the Green's function associated with the wave operator \square is given by

$$G(t, \mathbf{x}; t', \mathbf{x}') = -\frac{\delta(t' - [t - |\mathbf{x} - \mathbf{x}'|])}{4\pi|\mathbf{x} - \mathbf{x}'|}, \quad (4.27)$$

where $t - |\mathbf{x} - \mathbf{x}'|$ is the *retarded time*, even though its physical interpretation is given at Sec. ??, it is basically a quantity which accounts for the time required for the information from an event at position \mathbf{x} to reach one at \mathbf{x}' . Returning to our problem, since each component of the perturbation metric $\bar{h}_{\mu\nu}$ satisfies (4.23), Green's function can be separately applied to every single one of them, which leads to

$$\bar{h}_{\mu\nu} = 4 \int d^3x' \frac{T_{\mu\nu}(t - |\mathbf{x} - \mathbf{x}'|, \mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|}. \quad (4.28)$$

As we stated for the case of vacuum solutions, at least in the *TT*-gauge, the radiative degrees of freedom are embedded in the spatial part of the perturbation metric, thus, we have to translate (4.28) to the one compatible with the *TT*-gauge. Let us consider the spatial part

$$\bar{h}_{ij} = 4 \int d^3x' \frac{T_{ij}(t - |\mathbf{x} - \mathbf{x}'|, \mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|}, \quad (4.29)$$

which can be further simplified by making the following assumptions. Since we focus our interest in the radiation field emitted far from the source, which allows the factor $|\mathbf{x} - \mathbf{x}'|$ in

the denominator to be replaced by $r = |\mathbf{x}|$. Assuming L is the size of the source, and $r \gg L$, the fractional errors scale as $\frac{L}{r}$, and therefore can be neglected. This replacement is also valid for the argument of T_{ij} as long as the timescale τ over which the system changes, and the size of the source L are connected with the relation $\frac{L}{\tau} \ll 1$ (in $c = 1$ units), which stands for the velocity of the internal motions of the source, hence

$$T_{ij}(t - |\mathbf{x} - \mathbf{x}'|, \mathbf{x}') \simeq T_{ij}(t - r, \mathbf{x}'). \quad (4.30)$$

As a result, the metric perturbations takes the simplified form

$$\bar{h}_{ij}(t, \mathbf{x}) = \frac{4}{r} \int d^3x' T^{ij}(t - r, \mathbf{x}'), \quad (4.31)$$

which is considered the leading term in multipolar decomposition of the radiation field. Note that due to the metric being the flat Minkowski we are allowed to change between lower and upper spatial indices without altering generality. As previously mentioned, the conservation of the stress energy tensor, in the Minkowski flat spacetime, reads as $\partial_\mu T^{\mu\nu}$, which can be decomposed as

$$\partial_t T^{tt} = -\partial_i T^{it}, \quad (4.32)$$

from which, upon differentiating with respect to time, follows that

$$\partial_t^2 T^{tt} = \partial_i \partial_j T^{ij}. \quad (4.33)$$

Subsequently, we multiply each side with $x^i x^j$, and after some manipulation using integration by parts we obtain

$$\partial_t^2 (T^{tt} x^i x^j) = \partial_\kappa \partial_\lambda (T^{\kappa\lambda} x^i x^j) - 2\partial_\kappa (T^{i\kappa} x^j + T^{\kappa j} x^i) + 2T^{ij}. \quad (4.34)$$

Solving (4.34) for T^{ij} , and substituting to (4.31), while making use of the Gauss theorem to recast volume integrals to surface ones which vanish, the boundaries being outside of the sources, one obtains

$$\bar{h}_{ij} = \frac{2}{r} \frac{\partial^2}{\partial t^2} \int d^3x' T^{tt} x'^i x'^j = \frac{2}{r} \frac{\partial^2}{\partial t^2} \int d^3x' \rho x'^i x'^j. \quad (4.35)$$

We define the second moment I_{ij} of the mass distribution as

$$I_{ij}(t) = \int d^3x' \rho(t, \mathbf{x}') x'^i x'^j. \quad (4.36)$$

Combining the aforementioned results to

$$\bar{h}_{ij} = \frac{2}{r} \ddot{I}_{ij}(t - r). \quad (4.37)$$

Finally, in order to obtain the respective perturbation metric in the TT -gauge, we firstly have to subtract the trace of (4.37) as

$$\mathcal{I}_{ij} = I_{ij} - \frac{1}{3} \delta_{ij} I, \quad I = I_{ii}. \quad (4.38)$$

Then, we have to project out the non- TT pieces of (4.37), and since we are working with leading order $1/r$, this procedure reduces to projecting h_{ij} perpendicularly to the direction to $n^i = x^i/r$. This job is carried out by the projection operator

$$P_{ij} = \delta_{ij} - n_i n_j, \quad (4.39)$$

which eliminates components parallel to n^i , letting only the transverse components to survive. Therefore, the metric perturbation takes the following form

$$\bar{h}_{ij}^T = \bar{h}_{\kappa\lambda} P_{i\kappa} P_{j\lambda}, \quad (4.40)$$

and upon subtracting the trace of (4.40), we obtain the perturbation metric in TT -gauge as

$$h_{ij}^{TT} = \bar{h}_{\kappa\lambda} P_{i\kappa} P_{j\lambda} - \frac{1}{2} P_{ij} P_{\kappa\lambda} \bar{h}_{\kappa\lambda}. \quad (4.41)$$

Thus, we derive the quadrupole formula

$$\begin{aligned} h_{ij}^{TT}(t, \mathbf{x}) &= \frac{2}{r} \ddot{\mathcal{I}}_{\kappa\lambda}(t-r) \left[P_{i\kappa} P_{j\lambda} - \frac{1}{2} P_{ij} P_{\kappa\lambda} \right] \\ &= \frac{2}{r} \ddot{\mathcal{I}}_{ij}(t-r)^{TT}. \end{aligned} \quad (4.42)$$

This result implies that a source to radiates gravitational waves when acceleration (due to the dependence on the second time derivative of second moment of mass distribution) of matter occurs.

4.2 Newman-Penrose Scalars

In the previous section, we examined the limit of weak gravitational fields, where we defined the corresponding outgoing gravitational waves, and their two polarization states. In contrast to the aforementioned, BNS coalescences are violent processes which are considered strong gravitational sources, hence, the non-linear Einstein's equations have to be solved with our only available tool being numerical relativity. Thus, the need to successfully extract the gravitational strain emitted by the simulated system is certain. In this work, we employed one of the most commonly used techniques for the determination of the gravitational radiation signal from the numerical simulations, the Newman-Penrose scalar Ψ_4 .

This method is strongly coupled to the *Weyl tensor*, which arises from the fact that the Ricci tensor does not encompass all the information given by the Riemann tensor after its decomposition. In other words, the Weyl tensor carries the excess of information missing by the Ricci tensor, and is defined as

$$C_{\mu\nu\lambda\rho} \equiv R_{\mu\nu\lambda\rho} - [g_{\mu[\lambda} R_{\rho]\nu} - g_{\nu[\lambda} R_{\rho]\nu}] + \frac{1}{3} g_{\mu[\lambda} g_{\rho]\nu} R. \quad (4.43)$$

Its symmetries are identical to Riemann's tensor, while it is also traceless $C_{\lambda\mu\rho}^\mu = 0$. The Weyl tensor coincides with the Riemann tensor in vacuum, which is expected, and it is invariant under the effect of conformal transformations. The Newman-Penrose (or Weyl) scalars are defined as contractions of the Weyl tensor's components with a suitable orthonormal *null tetrad* l^μ , n^μ (not to be confused with the normal to Σ_t -surface vector n^μ), m^μ , \bar{m}^μ (which is the conjugate of m^μ). These vectors are considered to be null if they satisfy the relation

$$l^\mu l_\mu = n^\mu n_\mu = m^\mu m_\mu = \bar{m}^\mu \bar{m}_\mu = 0, \quad (4.44)$$

while, their are called a null tetrad when the following relations additionally hold as

$$l^\mu n_\mu = -m^\mu \bar{m}_\mu = 1, \quad l^\mu m_\mu = n^\mu \bar{m}_\mu = 0. \quad (4.45)$$

The null tetrad is constructed from the orthonormal basis on the spherical coordinates¹ $(\hat{\mathbf{R}}, \hat{\theta}, \hat{\phi})$, on each hypersurface Σ_t , where the normal vector defines a time-like vector $\hat{\mathbf{t}}$. Note that the

¹To be added. Changed notation for the radius coordinate r to R to distinguish it from the surface area r , while in the previous sections it was not necessary.

spherical coordinates are related to our spatial Cartesian (x , y , z) coordinates with the usual relations, while the radial coordinate R is not necessarily a surface area coordinate r (2-surface $r = t = \text{const.}$ where the area equals to $4\pi r^2$) which points to the direction of the propagation of the gravitational waves. The null tetrad implemented in WeylScalar4 module of the Einstein Toolkit, schematic of which is depicted in Fig. 4.2, is a modified version of [?] and reads as

$$\mathbf{l} = \frac{1}{\sqrt{2}} (\hat{\mathbf{t}} - \hat{\mathbf{R}}), \quad \mathbf{n} = \frac{1}{\sqrt{2}} (\hat{\mathbf{t}} + \hat{\mathbf{R}}), \quad \mathbf{m} = \frac{1}{\sqrt{2}} (\hat{\theta} - i\hat{\phi}). \quad (4.46)$$

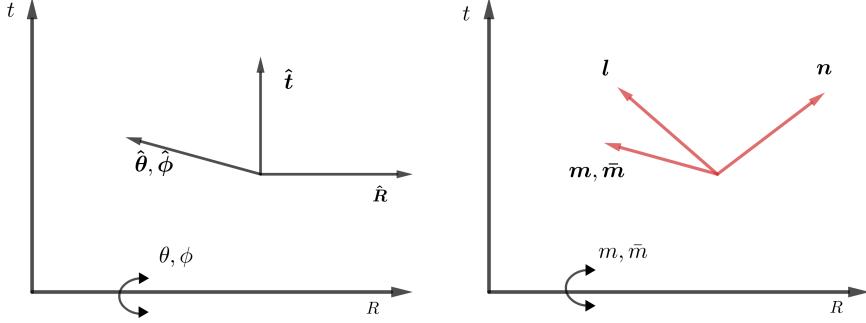


Figure 4.2: A schematic comparison between the orthonormal tetrad (left) and a null tetrad (right), in Minkowski spacetime, in spherical coordinates (t, R, θ, ϕ) .

Thus, the contractions of the Weyl tensor with the elements of the null tetrad (4.46), define the Newman-Penrose scalars, which are tetrad dependent, as

$$\Psi_0 = -C_{\mu\nu\lambda\rho} n^\mu m^\nu n^\lambda m^\rho, \quad (4.47)$$

$$\Psi_1 = -C_{\mu\nu\lambda\rho} n^\mu l^\nu n^\lambda l^\rho, \quad (4.48)$$

$$\Psi_2 = -C_{\mu\nu\lambda\rho} n^\mu m^\nu \bar{m}^\lambda l^\rho, \quad (4.49)$$

$$\Psi_3 = -C_{\mu\nu\lambda\rho} n^\mu l^\nu \bar{m}^\lambda l^\rho, \quad (4.50)$$

$$\Psi_4 = -C_{\mu\nu\lambda\rho} l^\mu \bar{m}^\nu l^\lambda \bar{m}^\rho. \quad (4.51)$$

These quantities carry all the information stored in the Weyl tensor, and for the purposes of this work, the most significant component is Ψ_4 . The latter, in the asymptotic limit, where $r \rightarrow \infty$, can be identified as the outgoing gravitational radiation field. Therefore, far from the isolated source, the gravitational waves and the Ψ_4 are connected to the perturbation metric in the TT gauge with the following relation

$$\Psi_4 = \ddot{h}_+ - i\ddot{h}_\times = \ddot{h}, \quad (4.52)$$

where \ddot{h} stands for complex gravitational wave strain $h = h_+ - ih_\times$. The latter definition, connects the complex strain h associated with the Weyl-scalars with the polarization states h_+ , h_\times , which we introduced in the TT gauge. In the limit of asymptotically flat spacetimes, with appropriate coordinates, the *Peeling theorem* [?] is valid, which states that the Newman-Penrose scalars (4.51) for $r \rightarrow \infty$ fall-off according to $\mathcal{O}(r^{n-5})$, therefore, Ψ_4 falls off as $\mathcal{O}(1/r)$. As a result it common to report results using quantities such as $r\Psi_4$, rh , which facilitates the comparison between waveforms extracted at spheres with different radii. As stated, the Peeling theorem assumes an asymptotic frame, and therefore it is not exactly satisfied at the finite radius where we extract the gravitational signal, which leads to an error in the computed gravitational strains which should be taken into account.

4.2.1 Decomposition of Ψ_4

We follow the widely used method [?] for the extraction of the gravitational signal from the numerical simulations, that is computing the Ψ_4 scalar (4.51) at spherical surfaces of different radii, and subsequently decomposing it to spin-weighted spherical harmonics of spin-weight $s = -2$ as

$$\Psi_4(t, R, \theta, \phi) = \sum_{l=2}^{l_{\max}} \sum_{m=-l}^l \Psi_4^{lm}(t, R) {}_{-2}Y^{lm}(\theta, \phi). \quad (4.53)$$

The above decomposition is carried out by the Multipole module, and the corresponding coefficients $\Psi_4^{lm}(t, R)$ are given by

$$\Psi_4^{lm}(t, R) = \int \Psi_4(t, R, \theta, \phi) {}_{-2}\bar{Y}^{lm}(\theta, \phi) r^2 d\Omega. \quad (4.54)$$

Lastly, it is helpful to define the concept the *retarded time* t_{ret} , which is the time measured by the "detector" spherical surface after the signal reaches the detector. To accomplish this, we approximate the metric, far from the sources, with a Schwarzschild metric, which leads to

$$t_{ret} = t - R^*, \quad R^* = R + 2M_{ADM} \log \left(\frac{R}{2M_{ADM} - 1} \right), \quad (4.55)$$

where M_{ADM} is the ADM mass, and R^* is known as the tortoise coordinate. The next step is to use (4.52) to integrate the Ψ_4^{lm} twice in time to obtain h^{lm} . For this reason, we employ a simple yet accurate method, that is to numerically calculate the double time integral using a rather simple trapezoid rule, and subsequently determine the two unknown integration constants throughout a polynomial fit of the strain. Finally the fitted constants are subtracted from the resulting strain as

$$\bar{h}_{lm}^{(0)}(t, r) = \int_0^t dt' \int_0^{t'} dt'' \Psi_4^{lm}(t'', r), \quad (4.56)$$

$$\bar{h}_{lm}(t, r) = \bar{h}_{lm}^{(0)}(t, r) - Q_1 t - Q_0, \quad (4.57)$$

where the natural interpretation of the fitted parameters is

$$Q_1 = \left. \frac{\partial \bar{h}}{\partial t} \right|_{t=0}, \quad Q_0 = \bar{h}(t=0). \quad (4.58)$$

It should be noted that the integration is carried out from the zero of the ordinary time t , instead of the more natural approach of the retarded time t_{ret} , which according to [?, ?] leads to oscillations in the first few milliseconds of the strain signal. In the above case, we performed a linear fit, where as we mentioned, the constants Q_0 , Q_1 , have a straightforward physical interpretation, however, this method leads to unphysical oscillations and non-linear drifts on the strain signal. This phenomenon is accurately documented by [?] and is attributed to unresolved high-frequency noise which is aliased in the low-frequency signal that does not result in a zero average in the double integration. To tackle this problem we employed a method, introduced in [?], which fits the integrated signal to a second-order polynomial as

$$h_{lm} = h_{lm}^{(0)} - Q_2 t^2 - Q_1 t - Q_0. \quad (4.59)$$

Despite the fact that Q_2 does not have a clear physical interpretation, it eliminates the non-physical effects of the signal strain, at least for the dominant modes ($l = 2$, $m = \pm$), which are

responsible for the majority of the radiated energy and angular momentum, while for the sub-dominant modes more sophisticated methods like the ones presented in [?] should be employed. Possibly, Q_2 can be interpreted as a correction term of finite-radius extraction errors [?].

Having obtained the strain signal h_{lm} , the next step is to compute the physical observable namely, the gravitational wave amplitude *spectral density* defined as

$$|\tilde{h}(f)| = \sqrt{\frac{|\tilde{h}_+|^2 + |\tilde{h}_\times|^2}{2}} \quad (4.60)$$

where \tilde{h}_+ , \tilde{h}_\times are the Fourier transform of h_+ , h_\times respectively. The processes during the evolution of BNS systems, especially in post-merger phase, can be potentially deciphered using the frequency spectrum of the emitted gravitational wave signal, since one can connect its main features with parameters directly linked with the neutron star's EOS.

4.3 Associated physical quantities

In this section we briefly present the formulas we employed to compute the energy, and angular momentum fluxes associated with the gravitation radiation. According to the procedure introduced at (??) the integrated radiated energy flux over the spherical surface of the extraction, associated with a radius R , is given by

$$\frac{dE^{gw}}{dt} = \frac{R^2}{16\pi^2} \int d\Omega |\dot{h}(t, \theta, \phi)|^2, \quad (4.61)$$

where $\dot{h}(t, \theta, \phi)$ stands for the derivative with respect to time of the complex gravitational signal. Since we decomposed Ψ_4 over the basis of spin-weighted spherical harmonics (4.54), we aim to express the above relation (4.61) in terms of the decomposed modes $h_{lm}(t)$. From (4.52), (4.53) and (4.54), follows that

$$\dot{h}(t, \theta, \phi) = \sum_{l=2}^{l_{\max}} \sum_{m=-l}^l \dot{h}_{lm}(t) Y^{lm}(\theta, \phi), \quad \dot{h}_{lm}(t) = \int_0^t dt' \Psi_{lm}(t'). \quad (4.62)$$

In the above we omitted the radius dependence since we examine quantities for specific choices of R , where surface or time dependent integrals are included. To compute the norm squared of the complex gravitational strain, we simply use (??) and orthonormal relations between the spin-weighted spherical harmonics, which leads to the useful relation

$$\int d\Omega |\dot{h}(t, \theta, \phi)|^2 = \sum_{l=2}^{l_{\max}} \sum_{m=-l}^l |\dot{h}_{lm}(t)|^2. \quad (4.63)$$

Thus, (4.61) reduces to

$$\frac{dE^{gw}}{dt} = \sum_{l=2}^{l_{\max}} \sum_{m=-l}^l \frac{dE_{lm}^{gw}}{dt}, \quad (4.64)$$

where

$$\frac{dE_{lm}^{gw}}{dt} = \frac{R^2}{16\pi^2} \sum_{l=2}^{l_{\max}} \sum_{m=-l}^l |\dot{h}_{lm}(t)|^2. \quad (4.65)$$

This result gives us the opportunity to examine the contribution to the total energy flux of each mode separately. Some authors [?] take advantage of (4.64) as tool to determine the merger time of BNS collisions by looking at the corresponding peaks of the dominant $l = 2$, $m = 2$ mode at the curves with respect to time. To be added the part of the angular momentum.

Chapter 5

The Code

In this chapter, we present the core modules (or thorns in the language of the Einstein Toolkit) which we incorporated in our code. Whilst there are modules which form the basis of the Einstein Toolkit code, we restrict ourselves and elaborate particularly in those which are crucial for our work (see [?] for an extensive discussion about the modules of the Einstein Toolkit). We start with the McLachlan module, which implements the BSSNOK formalism. Subsequently, we present the GRHydro module implementing a High Resolution Shock Capturing scheme for the evolution of the hydrodynamical variables. GRHydro provides a variety of alternatives for every of its internal tasks, while we focus on the ones we opted for our work. In addition, in Sec. ??, Sec. ?? we introduce our implementation following the work of [?], [?]. Next, we talk about the EOS_Omni module, which provides a general interface for all the required EOS calls throughout the initial data stage and the subsequent evolution of the system. We present the current implementation focusing on piecewise polytropic EOS, and the modifications we applied in order to include cold, tabulated EOS. In the final two sections, we introduce the Hydro_RNSID module which generates initial configurations of differentially rapidly rotating neutron stars (and non-rotating stars) compatible with the GRHydro module, and the LORENE library which generates binary neutron star (BNS) initial models which can imported to the Einstein Toolkit through the Meudon_NS module. Further information regarding the modules which we omitted in the following description can be found at the complete parameter files at appendix, or at the website our group [?].

5.1 McLachlan

The Einstein Toolkit handles the spacetime evolution through the McLachlan code. It implements the BSSNOK formalism described in Sec. ??, and discretizes the corresponding spacetime variables with an accurate finite differencing scheme up to eighth order (with additional Kreiss-Oliger dissipation terms). The code is auto-generated from the corresponding tensor equations of Mathematica scripts via Kranc [?], and is highly efficient and vectorized. The associated module in the Einstein Toolkit framework is ML_BSSN.

To check the accuracy of the simulations during the evolution of the system, one is able to monitor the violation of the Hamiltonian and Momentum constraint equations (??), through the variables,

$$\tilde{\mathcal{H}} \equiv R + \frac{2}{3}K^2 - \tilde{A}_{ij}\tilde{A}^{ij} = 0, \quad (5.1)$$

$$\tilde{\mathcal{M}} \equiv \tilde{D}_j\tilde{A}^{ij} + 6\tilde{A}^{ij}\partial_j\phi - \frac{2}{3}\tilde{\gamma}^{ij}\partial_jK = 0. \quad (5.2)$$

$$\tilde{\mathcal{L}}^i \equiv \tilde{\Gamma}^i + \partial_j\tilde{\gamma}^{ij} = 0. \quad (5.3)$$

Note that the first two constraint equations are expressed in terms of the conformal quantities according to the BSSNOK formalism (for simplicity we only show the case which corresponds to the absence of matter sources) while the third one stands for the additional constraint (3.36). The ML_BSSN module fixes the gauges, lapse function and shift vector, throughout an interface which allows us to set the parameters associated to the variations of "1+log" slicing. The general evolution of the lapse function is given by

$$\partial_t \alpha = -F\alpha^N K + \text{AlphaDriver}(\alpha - 1) + \text{advectLapse}\beta^i \partial_i \alpha. \quad (5.4)$$

The second term in (5.4) stands for the α -driver for which we have not discussed since we do not apply it in our work. The parameters AlphaDriver, AdvectLapse, N, and F, have to be specified in the parameter file. To obtain the advective "1+log" slicing (3.48) (which we employ in our work) one must set $N = 1$ and $F = 2$, in conjunction to AlphaDriver = 0 and AdvectLapse = 1. Similarly, the hyperbolic Γ -driver is given as

$$\partial_t \beta^i = \text{ShiftGammaCoeff}B^i + \text{AdvectShift}\beta^j \partial_j \beta^i, \quad (5.5)$$

$$\partial_t B^i = \partial_t \tilde{\Gamma}^i - \beta^j \partial_j \tilde{\Gamma}^i - \text{BetaDriver}B^i + \text{AdvectDriver}\beta^j \partial_j B^i. \quad (5.6)$$

Furthermore, it is imperative to impose proper boundary conditions at the external edges of the computational grid. Since our aim is to extract the gravitational wave signal, we must ensure that gravitational waves are reflected to the inner of the grid. This is carried out in the Einstein Toolkit in the NewRad module, which implements a Sommerfeld-type radiative boundary condition under the assumption that near the boundary all fields behave as spherical outgoing waves. Denoting as X any of the tensor components of the evolved variables, the boundary conditions reads as

$$X = X_0 + \frac{u(r - v_0 t)}{r}, \quad (5.7)$$

where X_0 stands for the asymptotic value of the given dynamical variable (typically equals to unity, for the lapse functions and the diagonal components of the metric, and zero for everything else), v_0 is the wave speed (it depends on the given variable), and u is a spherical perturbation (which is true, when the boundary is enough from the source). Since the code employs finite-difference schemes, it is more efficient to use the corresponding differential form of (5.7) in Cartesian coordinates, given by

$$\partial_t X = -\frac{v_0 x^i}{r} \partial_i X - v_0 \frac{X - X_0}{r}. \quad (5.8)$$

Second order central differencing is employed for the calculation of spatial derivatives (one sided finite differencing if it not possible). The parts of the solution which do not behave as a pure outgoing wave, is assumed to decay with a certain power p of the radius. To accomplish this, the time derivative term $\partial_t X$ is replaced with

$$(\partial_t X)^* = (\partial_t X) + \left(\frac{r}{r - n^i \partial_i r} \right)^p n^i \partial_i (\partial_t X), \quad (5.9)$$

where n^i stands for the normal vector of the given boundary surface. In our work, second order decay of $p = 2$ is assumed. To invoke the aforementioned features of ML_BSSN the parameter file section shown in Tab. 5.1 must be added to the corresponding parameter file.

# 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
ML_BSSN::initial_boundary_condition	= "extrapolate-gammas"
ML_BSSN::rhs_boundary_condition	= "NewRad"
ML_BSSN::timelevels	= 3
Boundary::radpower	= 2
ML_BSSN::advectLapse	= 1
ML_BSSN::advectShift	= 1
ML_BSSN::harmonicN	= 1.0 # 1+log
ML_BSSN::harmonicF	= 2.0 # 1+log
ML_BSSN::ShiftGammaCoeff	= 0.75
ML_BSSN::AlphaDriver	= 0.0
ML_BSSN::BetaDriver	= 0.75
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	= 1
ML_BSSN::SpatialBetaDriverRadius	= 50
ML_BSSN::apply_dissipation	= "never"
ML_BSSN::epsDiss	= 0.0

Table 5.1: Parameter file section to invoke the ML_BSSN module.

5.2 GRHydro

The GRHydro module carries out the evolution of the hydrodynamical variables in full 3-dimensional general relativity using a High Resolution Shock Capturing (HRSC) scheme. The original implementation dealt with general relativistic hydrodynamics (GRHD) and was derived from the public version of Whisky code [], while the most recent one [?] implements general relativistic magnetohydrodynamics (GRMHD) using the Valencia formulation []. The evolution of the magnetic fields in full general relativity dynamical spacetimes is accomplished under

the assumption of ideal MHD. In the following sections, we present the basic definitions of the Valencia formulation, in addition to the numerical methods we opted for our code. We present the WENO reconstruction scheme, the HLLE Riemann solver, and three schemes for the conversion of conservative variables to primitive ones.

5.2.1 Valencia formulation

As we previously mentioned, GRHydro evolves the magnetic fields under the assumption of ideal MHD. In the latter, the fluids have zero resistivity and the electric fields vanish in the comoving frame of the fluid (no charge separation), the condition being

$$u_\mu F^{\mu\nu} = 0. \quad (5.10)$$

In the Einstein Toolkit, the magnitude of the relativistic Faraday tensor $F^{\mu\nu}$, and its dual ${}^*F^{\mu\nu} \equiv \frac{1}{2}\epsilon^{\mu\nu\kappa\lambda}F_{\kappa\lambda}$, in addition to the electric and magnetic fields, are rescaled by a factor of $1/\sqrt{4\pi}$. This way, there is no need of including the permittivity and permeability of free space in CGS-Gaussian units.

To include magnetic fields into the hydrodynamics equation, one has to add the corresponding electromagnetic contribution to the energy-momentum tensor. Combining the hydrodynamic and electromagnetic contributions, one obtains the energy-momentum tensor for ideal GRMHD as

$$T^{\mu\nu} = (\rho h + b^2) u^\mu u^\nu + \left(P + \frac{b^2}{2} \right) g^{\mu\nu} - b^\mu b^\nu, \quad (5.11)$$

where ρ , ϵ , P , u^μ , and $h = 1 + \epsilon + P/\rho$ stand for the fluid rest mass density, specific internal energy, pressure (all three are measured in the comoving with the fluid frame), 4-velocity of the fluid, specific enthalpy. Moreover, b^μ is the magnetic 4-vector in the comoving with fluid frame defined as

$$b^\mu = u_\nu {}^*F^{\mu\nu}, \quad (5.12)$$

from which one has $b^2 = b^\mu b_\mu$. Commonly in literature, the magnetic pressure is defined as $P_m = b^2/2$, which results to the definition of the magnetically modified pressure and enthalpy (which we do not adopt in this work) as

$$P^* = P + P_m = P + \frac{1}{2}b^2, \quad h^* = 1 + \epsilon + \frac{P_m}{\rho} = 1 + \epsilon + \frac{2P + b^2}{2\rho}. \quad (5.13)$$

Next, we define the magnetic field measured by the Eulerian observer as

$$B^\mu = n_\nu {}^*F^{\mu\nu}, \quad (5.14)$$

from which one can easily see, using $n_\mu = (-\alpha, 0, 0, 0)$ and the antisymmetric properties of the Faraday tensor, that the spatial components of its projection to spatial hypersurface ($\gamma_\nu^\mu n_\rho {}^*F^{\nu\rho}$) reduce to

$$B^i = n_\mu {}^*F^{i\mu} = -\alpha {}^*F^{i0}. \quad (5.15)$$

GRHydro evolves the ideal GRMHD equations which are obtained from the local conservation laws of mass and energy momentum, and the Maxwell's equations as

$$\nabla_\mu(\rho u^\mu) = 0 \quad (5.16)$$

$$\nabla_\mu T^{\mu\nu} = 0 \quad (5.17)$$

$$\nabla_\nu {}^*F^{\mu\nu} = 0 \quad (5.18)$$

The GRHydro scheme is written in a first order hyperbolic flux-conservative evolution system with respect to the conserved variables D , \mathcal{S}^i , τ , and \mathcal{B}^i defined in terms of the primitive variables ρ , ϵ , v^i , and B^i as

$$D = \sqrt{\gamma} \rho W, \quad (5.19)$$

$$\mathcal{S}_j = \sqrt{\gamma} (-n_\mu T^\mu_i) = \sqrt{\gamma} \left((\rho h + b^2) W^2 v_j - \alpha b^0 b_j \right), \quad (5.20)$$

$$\tau = \sqrt{\gamma} (n_\mu n_\nu T^{\mu\nu}) - D = \left((\rho h + b^2) W^2 - \left(P + \frac{b^2}{2} \right) - (\alpha b^0)^2 \right) - D, \quad (5.21)$$

$$\mathcal{B}^k = \sqrt{\gamma} B^k, \quad (5.22)$$

where following the same notation with Ch. ??, γ stands for the $\det \gamma_{ij}$, v^i is the 3-velocity of the fluid as measured by the Eulerian observer, defined as

$$v^i = \frac{\gamma^i_\mu u^\mu}{-n_\mu u^\mu} = \frac{u^i}{W} + \frac{\beta^i}{\alpha}, \quad (5.23)$$

where

$$W = -n_\mu u^\mu = \alpha u^0 = \frac{1}{\sqrt{1 - v^2}}, \quad (5.24)$$

corresponds to the relative Lorentz factor between u^μ and n^μ , with $v^2 = \gamma_{ij} v^i v^j$. We remind the reader that the vectors which reside on the spatial hypersurfaces namely, v^i , B^i , \mathcal{S}^i , and β^i , raise and lower their indices through the spatial metric γ_{ij} . The corresponding evolution system for the conserved variables, according to (5.18) (only the spatial components of the third equation considered) is given by

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}^i}{\partial x^i} = \mathbf{S}, \quad (5.25)$$

where,

$$\mathbf{U} = \begin{bmatrix} D \\ \mathcal{S}_j \\ \tau \\ \mathcal{B}^k \end{bmatrix}, \quad \mathbf{F}^i = \alpha \times \begin{bmatrix} D \tilde{v}^i \\ \mathcal{S}_j \tilde{v}^i + \sqrt{\gamma} \left(P + \frac{b^2}{2} \right) \delta^i + j - b_j \mathcal{B}^i / W \\ \tau \tilde{v}^i + \sqrt{\gamma} \left(P + \frac{b^2}{2} \right) v^i - \alpha b^0 \mathcal{B}^i / W \\ \mathcal{B}^k \tilde{v}^i - \mathcal{B}^i \tilde{v}^k \end{bmatrix}, \quad (5.26)$$

$$\mathbf{S} = \alpha \sqrt{\gamma} \times \begin{bmatrix} 0 \\ T^{\mu\nu} \left(\partial_\mu g_{\nu j} - {}^{(4)}\Gamma_{\mu\nu}^\lambda g_{\lambda j} \right) \\ \alpha \left(T^{\mu 0} \partial_\mu \ln \alpha - T^{\mu\nu} {}^{(4)}\Gamma_{\mu\nu}^0 \right) \\ \mathbf{0} \end{bmatrix}. \quad (5.27)$$

In the above, $\tilde{v}^i = v^i - \beta^i / \alpha$. Lastly, let us show a few useful relations (especially for the conversion of the conservative to primitive variables) between the magnetic fields measured by the comoving observers (b^μ) and Eulerian observers (B^i):

$$b^0 = \frac{WB^k v_k}{\alpha}, \quad (5.28)$$

$$b^i = \frac{B^i}{W} + WB^k v_k \left(v^i - \frac{\beta^i}{\alpha} \right), \quad (5.29)$$

$$b^2 = \frac{B^i B_i}{W^2} + (B^k v_k)^2. \quad (5.30)$$

5.2.2 Reconstruction

The GRHydro module provides a variety of methods for the reconstruction of the matter sources, the fluid and magnetic field variables, to cell interfaces. These are: TVD (total variation diminishing) [], PPM (piecewise parabolic method) [], ENO (essentially non-oscillatory) [], WENO (5th order weighted ENO) [], and MP5 (5th order monotonically preserving) [].

Amongst the aforementioned, for the analysis presented in this work, we employed WENO, since the merger time of BNS at lower resolution simulations appears to be less than in the cases where the alternative methods (PPM, MP2) [] were chosen. Thus, it minimizes the required computational resources, and facilitates test BNS simulations.

The idea of WENO reconstruction originates from the original ENO approach. While the latter, suffers from several drawbacks such as the order of accuracy not being maximal given the available stencil points, and the unnecessary slow performance due to the large number of the required if statements, the former attempts to overcome the above. WENO aims to combine all of the available ENO reconstruction stencils through the assignment of weights to each stencil. In the case where the full set of stencils is employed, the weights are non-zero, and maximal accuracy for a given number of points is obtained. The weights are determined according to the local smoothness of the flow resulting to fifth-order accuracy when the flow is smooth, while for less smooth flow, some weights are suppressed and the lower order interpolation is dominant. The WENO implementation of **GRHydro**, as introduced in [], is the following.

We introduce three interpolation polynomials (*Lagrange interpolation*) which approximate $U_{i+1/2}^L$ from cell-averages of a given quantity U_i :

$$U_{i+1/2}^{L,1} = +\frac{3}{8}U_{i-2} - \frac{5}{4}U_{i-1} + \frac{15}{8}U_i, \quad (5.31)$$

$$U_{i+1/2}^{L,2} = -\frac{1}{8}U_{i-1} + \frac{3}{4}U_i - \frac{3}{8}U_{i+1}, \quad (5.32)$$

$$U_{i+1/2}^{L,3} = +\frac{3}{8}U_i + \frac{3}{4}U_{i+1} - \frac{1}{8}U_{i+2} \quad (5.33)$$

Every single one of the above polynomials stands for a third-order accurate approximation of U at the cell-interface, where a convex linear combination of the three interpolation polynomials can potentially lead to a higher-order accurate interpolation polynomial. More specifically, introducing

$$U_{i+1/2}^L = w^1 U_{i+1/2}^{L,1} + w^2 U_{i+1/2}^{L,2} + w^3 U_{i+1/2}^{L,3} \quad (5.34)$$

with w^i being the weights which satisfy $\sum_i w^i = 1$, one can possibly obtain a fifth-order interpolation polynomial which spans all five stencil points $\{U_{i-2}, \dots, U_{i+2}\}$. The associated weights w^i are obtained from the so-called smoothness indicators β^i , where in the original WENO algorithm [], are given by

$$\beta^1 = \frac{1}{3} (4U_{i-2}^2 - 194U_{i-2}U_{i-1} + 25U_{i-1}^2 + 11U_{i-2}U_{i-1} - 31U_{i-1}U_i + 10U_i^2), \quad (5.35)$$

$$\beta^2 = \frac{1}{3} (4U_{i-2}^2 - 134U_{i-1}U_i + 13U_i^2 + 5U_{i-1}U_{i+1} - 13U_iU_{i+1} + 4U_{i+1}^2), \quad (5.36)$$

$$\beta^3 = \frac{1}{3} (10U_i^2 - 31U_iU_{i+1} + 25U_{i+1}^2 + 11U_iU_{i+2} - 19U_{i+1}U_{i+2} + 4U_{i+2}^2). \quad (5.37)$$

For this particular section β^i refers to the smoothness indicators, and not to the previously defined shift vector. The corresponding weights are given by

$$w^i = \frac{\bar{w}^i}{\bar{w}^1 + \bar{w}^2 + \bar{w}^3}, \quad \bar{w}^i = \frac{\gamma^i}{(\epsilon + \beta^i)^2} \quad (5.38)$$

where $\gamma^i = \{1/16, 5/8, 5/16\}$, and ϵ is a small constant in order to avoid a potential division by zero. It should be noted that the choice of ϵ is scale dependent, hence, choosing a fixed number is not appropriate for cases with variation in scales. Employing the improvement introduced in [], the problem is overcome by modifying the smoothness indicators (5.38). In the most recent implementation of **GRHydro** the *modified* smoothness indicators are obtained by

$$\bar{\beta}^i = \beta^i + \epsilon (|U^2| + 1), \quad |U^2| = U_{i-2}^2 + U_{i-1}^2 + U_i^2 + U_{i+1}^2 + U_{i+2}^2 \quad (5.39)$$

where in **GRHydro**, the default value of $\epsilon = 10^{-26}$ is chosen. Due to $|U^2|$, the modified smoothness indicators are reliant on the scale of the reconstructed field. The derivation of $U_{i+1/2}^R$ follows the same recipe, except that one has to replace the values of U_{i-2}, \dots, U_{i+2} with U_{i+2}, \dots, U_{i-2} .

Alternatively, one has the option of using the so-called *WENO Z* [] reconstruction method, which we also employed in our study of single stars. This method introduces new smoothness indicators of higher order than the original counterpart.

5.2.3 Riemann Solvers

Consequently, as the Riemann problem is constructed, its solution can be obtained through the approximate solver Harten-Lax-van Leer-Einfeldt (HLLE) [], []. In addition to the latter, **GRHydro** implements more accurate solvers, namely Roe and Marquina, however, for the purpose of this work, we employed HLLE, since it is computationally less expensive than the others. More specifically, HLLE follows a two-wave approximation to update the terms across the discontinuity at the cell interface, whereas the other solvers, Roe and Marquina, require the knowledge of the eigenvalues which characterize the linearising hydrodynamic evolution scheme, which is extremely resource intensive and does not provide a decisive advantage in accuracy. In the HLLE solver, ξ_- and ξ_+ stand for the most negative and most positive wave speed eigenvalues present on either side of the interface, and the solution state vector \mathbf{U} is assumed to take the following form:

$$\mathbf{U} = \begin{cases} \mathbf{U}^L & \text{if } 0 < \xi_-, \\ \mathbf{U}_* & \text{if } \xi_- < 0 < \xi_+, \\ \mathbf{U}^R & \text{if } 0 > \xi_+, \end{cases} \quad (5.40)$$

where the state \mathbf{U}_* is equal to

$$\mathbf{U}_* = \frac{\xi_+ \mathbf{U}^R - \xi_- \mathbf{U}^L - \mathbf{F}(\mathbf{U}^R) + \mathbf{F}(\mathbf{U}^L)}{\xi_+ - \xi_-}. \quad (5.41)$$

The numerical flux along the interface is given by

$$\mathbf{F}(\mathbf{U}) = \frac{\hat{\xi}_+ \mathbf{F}(\mathbf{U}^L) - \hat{\xi}_- \mathbf{F}(\mathbf{U}^R) + \hat{\xi}_+ \hat{\xi}_- (\mathbf{U}^R - \mathbf{U}^L)}{\hat{\xi}_+ \hat{\xi}_-} \quad (5.42)$$

where we impose that

$$\hat{\xi}_- = \min(0, \xi_-), \quad (5.43)$$

$$\hat{\xi}_+ = \max(0, \xi_+). \quad (5.44)$$

These flux terms (5.42) are employed for the evolution of the hydrodynamic quantities. As previously stated, HLLE solver is an approximate one, hence, the computed wave speeds overstate the true wave speeds by no more than a factor of $\sqrt{2}$ [], only for certain magnetic fields and

fluid velocity configurations. To obtain the wavespeeds, the full MHD dispersion relations is replaced by the approximate quadratic form

$$\omega_d^2 = k_d^2 \left[v_A^2 + c_s^2 \left(1 - \frac{v_A^2}{c^2} \right) \right]. \quad (5.45)$$

In the above, the wave-vector is defined as $k_\mu \equiv (-\omega, k_i)$, c_s is the fluid velocity, and finally, the Alfvén velocity,

$$v_A \equiv \sqrt{\frac{b^2}{\rho h + b^2}} = \sqrt{\frac{b^2}{\rho h^*}}. \quad (5.46)$$

Moreover, we define the projected wave vector as

$$K_\mu \equiv (\delta_\mu^\nu + u_\mu u^\nu) k_\nu, \quad (5.47)$$

and the dispersion relation between frequency and (squared) wave number, given by

$$\omega_d = k_\mu u^\mu = -\omega u^0 + k_i u^i \quad (5.48)$$

$$k_d^2 = K_\mu K^\mu = \omega_d^2 + g^{\nu\sigma} k_\nu k_\sigma \quad (5.49)$$

The substitution of the above leads to the followinf quadratic

$$\xi^2 [W^2(V^2 - 1) - V^2] - 2\xi [\alpha W^2 \tilde{v}^i (V^2 - 1) + V^2 \beta^i] + \quad (5.50)$$

$$[(\alpha W \tilde{v}^i)^2 (V^2 - 1) + V^2 (\alpha \gamma^{ii} - \beta^i \beta^i)] = 0 \quad (5.51)$$

where $V^2 = v_A^2 + c_s^2(1 - v_A^2)$, ξ is the resulting wavespeed. There is no implicit summation over the repeated indices, and as a result, there are different wavespeeds at different directions.

5.2.4 Conservative to Primitive variable transformation

At this point, it is important to clarify the connection between different definitions used in literature which can invoke misunderstandings (at least based on personal experience). The momentum density measured by the normal observer is given by the quantity $-n_\nu T^\nu_\mu$, from which we defined its associated projection to the hypersurface (3.26) as

$$S_\mu = -\gamma^\nu_\mu n_\rho T^\rho_\nu = \gamma^\nu_\mu \alpha T^0_\nu, \quad (5.52)$$

where in the last step, we further simplified the expression using $n_\mu = (-\alpha, 0, 0, 0)$. Thus, we obtain the spatial components S_i as

$$\begin{aligned} S_i &= \gamma^0_i \alpha T^0_\nu \\ &= (\delta^0_i + n^0 n_i) \alpha T^0_\nu \\ &= \alpha T^0_i, \end{aligned} \quad (5.53)$$

from which, using the definition of the stress energy tensor (??), we immediately see that $S_i = \hat{S}_i$, since

$$S_i = \alpha T^0_i = (\rho h + b^2) W^2 v_i - \alpha b^0 b_i = \hat{S}_i. \quad (5.54)$$

we remind that γ_{ij} raises and lowers indices of spatial tensors, hence, $S^i = \gamma^{ij} S_j$.

In this work, we concentrate on the cases where magnetic fields are absent, however, not only do we employ pure GRHD treatments but we mostly focus on GRMHD formulations on

the limit of $B^i = 0$. This way, as a next step, we will be able to investigate phenomena where magnetic fields are not negligible.

In GRHD, the procedure of the recovery of primitive from the conservative variables is rather straightforward. One has to invert (??), (??), with all of the magnetic field quantities set to zero. Currently, **GRHydro** implements a 1D Newton-Raphson scheme [], which iterates by estimating the fluid pressure, and therefore determining the density ρ , internal specific energy ϵ and using those, in addition to the evolved conserved variables, to calculate the new value for the pressure. The latter becomes the seed for the next iteration, and the procedure is repeated until the specified tolerance is achieved.

For the purposes of this work, we present the general scheme, that is when a general equation of state $P(\rho, \epsilon)$ is used, and omit the simplified cases of the known analytical equations of state (polytrope, ideal gas). Firstly, we define the so-called (known) undensitized conservative variables as follows

$$\hat{D} \equiv \frac{D}{\sqrt{\gamma}} = \rho W, \quad \hat{S}^i \equiv \frac{S^i}{\sqrt{\gamma}} = \rho h W^2 v^i, \quad (5.55)$$

$$\hat{\tau} \equiv \frac{\tau}{\sqrt{\gamma}} = \rho h W^2 - P - \hat{D}. \quad (5.56)$$

Secondly we utilize the auxiliary quantities given by

$$Q \equiv \rho h W^2 = \hat{\tau} + \hat{D} + P, \quad (5.57)$$

$$\hat{S}^2 = \gamma_{ij} \hat{S}^i \hat{S}^j = (\rho h W)^2 (W^2 - 1). \quad (5.58)$$

Note, that Q is P dependent, and \hat{S}^2 is known. Subsequently, we find that

$$\sqrt{Q^2 - \hat{S}^2} = \rho h W, \quad (5.59)$$

from which we can update values of ρ , ϵ as

$$\rho = \frac{\hat{D} \sqrt{Q^2 - \hat{S}^2}}{Q}, \quad W = \frac{Q}{\sqrt{Q^2 - \hat{S}^2}}, \quad \epsilon = \frac{\sqrt{Q^2 - \hat{S}^2} - PW - \hat{D}}{\hat{D}}. \quad (5.60)$$

Thus, given the new updated values of ρ , ϵ , one can compute the residual between the pressure (initial guess or the one at the current iteration step) and $P(\rho, \epsilon)$, and compute the next estimate using the Newton Raphson method as long the derivatives $\frac{dP}{d\rho}$, $\frac{dP}{d\epsilon}$ are known. In other words, the goal is to find the solution which satisfies the above equations in addition to:

$$P - P(\rho, \epsilon) = 0, \quad (5.61)$$

hence, due to the nature of Einstein Toolkit, one should compute the following derivative,

$$\frac{dP(\rho, \epsilon)}{dP} = \frac{\partial P(\rho, \epsilon)}{\partial \rho} \frac{\partial \rho}{\partial P} + \frac{\partial P(\rho, \epsilon)}{\partial \epsilon} \frac{\partial \epsilon}{\partial P} \quad (5.62)$$

where

$$\frac{\partial \rho}{\partial P} = \frac{\hat{D} \hat{S}^2}{\sqrt{Q^2 - \hat{S}^2 Q^2}}, \quad \frac{\partial \epsilon}{\partial P} = \frac{P \hat{S}^2}{\rho (Q^2 - \hat{S}^2) Q}. \quad (5.63)$$

In the case of MHD, which are mostly interested in, the recovery of the primitive variables is rather complicated. The inversion of the additional equation (??) is trivial, since $B^i = \mathcal{B}^i / \sqrt{\gamma}$,

however the primary difficulty lies in b^μ , which cannot be trivially determined, hence due the absence of analogue to GRHD expressions one cannot calculate the density one the pressure is specified. In a similar fashion to GRHD, we define the undensitized conservative variables which satisfy the GRMHD equations

$$\hat{D} \equiv \rho W, \quad (5.64)$$

$$\hat{S}^i \equiv (\rho h + b^2)W^2 v^i - \alpha b^0 b^i, \quad (5.65)$$

$$\hat{\tau} \equiv (\rho h + b^2)W^2 - \left(P + \frac{b^2}{2} \right) - (\alpha b^0)^2 - \hat{D}. \quad (5.66)$$

The currently implemented recovery scheme in **GRHydro** is a multi-dimensional (3-dimensional for a general equation of state) Newton-Raphson solver, which is mostly based on the methods presented in [], and more specifically, on $2D$, $1D_w$ solvers. In the following we present the method¹ responsible for the recovery of primitive variables for a general equation of state.

Firstly, we must define a few auxiliary quantities to facilitate the numerical calculations. Note that we use a notation compatible with the Einstein Toolkit in contrast to the one adopted by the original publication [?]. However, the corresponding implemented routine employs the latter notation, hence, the curious reader is advised to refer to Tab. ?? for the changes between the two notations.

First, we recall the definition of the momentum density, projected in the hypersurface Σ_t , measured by the normal observer (??). Therefore, the expression

$$\mathcal{S}_\mu = -n_\nu T_\mu^\nu = \alpha T_\mu^0, \quad \text{where } \mathcal{S}_i = \hat{S}_i, \quad (5.67)$$

stands for the 4-dimensional momentum density (we again changed the notation so it compatible with the publication in examination). The normal projection $\tilde{\mathcal{S}}$ is given by

$$\tilde{\mathcal{S}} = (\delta_\mu^\nu + n_\mu n^\nu) \mathcal{S}_\nu, \quad \text{where } \tilde{\mathcal{S}}_i = \mathcal{S}_i, \quad (5.68)$$

since $n_\mu = (-\alpha, 0, 0, 0)$. In addition, we make use of an auxiliary variable Q which is defined as

$$Q \equiv \rho h W^2. \quad (5.69)$$

Thus, for the general equation of state, a three-dimensional Newton - Raphson scheme solves the following three equations for the unknown variables Q , v^2 , u , where the latter one is the so-called *internal energy density* defined as $u = \rho \epsilon$.

$$\tilde{\mathcal{S}}^2 = \hat{S}_i \hat{S}^i = v^2 (B^2 + Q)^2 - (\mathcal{S} \cdot \mathbf{B})^2 \frac{B^2 + 2Q}{Q^2} \quad (5.70)$$

$$\mathcal{S} \cdot n = -(\hat{\tau} + \hat{D}) = \frac{B^2}{2} (1 + v^2) + \frac{(\mathcal{S} \cdot \mathbf{B})^2}{2} Q^{-2} - v^2 Q - \hat{D} \sqrt{1 - v^2} - u, \quad (5.71)$$

$$P(\rho, u) = (1 - v^2) Q - \hat{D} \sqrt{1 - v^2} - u \quad (5.72)$$

Note that all dot products are four-dimensional when four-vectors are involved and three-dimensional when the time components vanish e.g. $\mathcal{S} \cdot \mathbf{B} = \mathcal{S}_\mu B^\mu = \mathcal{S}_i B^i$, since $B^0 = 0$. However, since a few steps have been omitted, additional explanation is required. In the above equations we utilized the pressure expression

$$P = (1 - v^2) Q - \hat{D} \sqrt{1 - v^2} - u \quad (5.73)$$

¹The implemented routine can be found in **GRHydro** module under the "src" directory, with function name "GRHydro_Con2PrimM_pt2".

so the second of (5.72) is only Q , v^2 dependent. Lastly, the third equation of (5.72) is the equivalent of (5.61) for MHD. To summarize, in terms of the Newton Raphson scheme, we initially compute $\rho = \hat{D}/W = \hat{D}\sqrt{1 - v^2}$, and then solve (5.72) for Q^2 , v^2 , u .

Within the Newton Raphson scheme², one must calculate the partial derivatives of the pressure with respect to the variables v^2 , u , as

$$\frac{\partial P}{\partial(v^2)} = \left(\frac{\partial P}{\partial\rho} \right)_u \frac{d\rho}{(v^2)} = \left[\left(\frac{\partial P}{\partial\rho} \right)_\epsilon - \frac{\epsilon}{\rho} \left(\frac{\partial P}{\partial\epsilon} \right)_\rho \right] \frac{d\rho}{d(v^2)}, \quad (5.74)$$

$$\frac{\partial P}{\partial u} = \frac{1}{\rho} \left(\frac{\partial P}{\partial\epsilon} \right)_\rho. \quad (5.75)$$

The reason for choosing these partial derivatives, including the chain rule, is twofold. On one hand, the pressure depends on ρ , ϵ , and since the density is only reliant on \hat{D} and v^2 , only the partial derivatives $\partial P/\partial(v^2)$, $\partial P/\partial u$ are required. On the other hand, the current version of the Einstein Toolkit's EOS interface computes partial derivatives against ρ , ϵ rather than ρ , u .

In this work, we attempt to contribute to the Einstein Toolkit, by implementing an alternative recovery scheme for GRMHD, presented by [?], and [?], where a comparison of several schemes including this one can be found at [?]. This method significantly reduces the dimensionality of the recovery problem to one-dimension (1D), and can potentially be highly useful when the pressure depends only on ρ and ϵ , which is the case for the Einstein Toolkit's EOS interface. This scheme utilizes Brent's method to minimize

$$f(x) = x - \hat{h}\hat{W} = x - \left(1 + \hat{\epsilon} + \frac{P(\hat{\rho}, \hat{\epsilon})}{\hat{\rho}} \right) \hat{W}, \quad (5.76)$$

in the unknown $x = hW$. The quantities with a hat (only in this scheme, not to be confused with undesitized conserved variables) are computed at every iteration step from x and the conserved variables as shown in the following.

As presented in [?], [?], we rescale the conserved quantities in order to get order-unity quantities, namely

$$\begin{aligned} q &\equiv \frac{\hat{\tau}}{\hat{D}}, & r &\equiv \frac{\hat{\mathcal{S}}^2}{\hat{D}^2}, \\ s &\equiv \frac{B^2}{\hat{D}}, & t &\equiv \frac{B_i \hat{\mathcal{S}}^i}{\hat{D}^{3/2}}. \end{aligned} \quad (5.77)$$

According to the authors, the solution x in (5.76) is bracketed by

$$1 + q - s < x < 2 + 2q - s, \quad (5.78)$$

hence, employing Brent's method is a natural choice for finding the root. Thus, the recovery scheme can be summarized in the following steps.

- From (5.67):

$$\mathcal{S}_i = (\rho h W^2 + B^2) v_i - (B_j v^j) B_i, \quad \text{using } B^i \mathcal{S}_i = \rho h W^2 B^i v_i, \quad (5.79)$$

one can write

$$\hat{W}^{-2} = 1 - \frac{x^2 r + (2x + s)t^2}{x^2(x + s)^2}. \quad (5.80)$$

²For the minimization of three-dimensional system of equations $f_i(Q, v^2, u) = 0$, with $i = 1, 2, 3$, the Jacobian is required.

- Subsequently, one computes $\rho = \hat{D}/\hat{W}$, while $\hat{\epsilon}$ can be obtained from $\epsilon = h - 1 - P/\hat{\rho}$, which after some algebra takes the following form

$$\hat{\epsilon} = -1 + \frac{x}{\hat{W}}(1 - \hat{W}^2) + \hat{W} \left[1 + q - s + \frac{1}{2} \left(\frac{t^2}{x^2} + \frac{s}{\hat{W}^2} \right) \right]. \quad (5.81)$$

- In our case the pressure $P(\hat{\rho}, \hat{\epsilon})$ can be trivially calculated from the EOS, while for temperature-dependent EOS, one must invert $\hat{\epsilon}$ to find the corresponding temperature, and therefore the pressure.
- Once Brent's method solves (5.76), x serves as the input for the next cycle. Note that for the first initial guess, data from the previous time-step can be used.

This procedure is repeated until convergence to the specified tolerance is achieved.

Finally, once ρ and ϵ are obtained, the velocity components v^i can be computed by setting $Q = x\rho W$ and employing

$$v^i = \frac{\mathcal{S}^i}{Q + B^2} + \frac{(B^j \mathcal{S}_j) B^i}{Q(Q + B^2)}. \quad (5.82)$$

Fixing Inequalities

One of the major obstacles one faces replacing analytical EOS with tabulated EOS is the recovery of primitive variables to the conservative variables during the evolution of the hydrodynamics variables. Not only is it computationally expensive, since a system of non-linear equations has to be solved numerically for each fluid element, but also is one of the most common reasons for a code to crash. Due to numerical errors, the conservative variables obtain values which lie outside of the physically accepted region resulting the recovery to fail, which is highly problematic. To counter this problem we slightly modify the conservative variables, in order to yield solutions in the physical range, following the prescription of [?]. In the limit of vanishing magnetic fields (which is the one we use in our simulations) the inversion failure can be partially avoided, since we use tabulated EOS while the authors employ other types of EOS, by enforcing the constraints

$$\hat{S}^2 \equiv \gamma^{ij} \hat{S}_i \hat{S}_j \leq \hat{\tau} (\hat{\tau} + 2\hat{D}), \quad \hat{\tau} \geq 0. \quad (5.83)$$

When the first inequality is violated, we rescale \hat{S}_i so that its new magnitude is $\hat{S}_i^2 = \hat{\tau} (\hat{\tau} + 2\hat{D})$ by replacing

$$\hat{S}_i \rightarrow \hat{S}_i \sqrt{\frac{\hat{\tau} (\hat{\tau} + 2\hat{D})}{\hat{S}^2}}. \quad (5.84)$$

When the latter condition is not met, we simply set $\hat{\tau}$ to a significantly low value. For our simulations of single stars, this simple fixing of the inequalities [?] avoided failures of the recovery scheme when tabulated EOS were employed.

5.2.5 Artificial atmosphere

In our simulations, matter is located on a rather small portion of the computational grid, whereas the rest can be approximated by vacuum. However, in vacuum limit the continuity equations describing the fluid break, and therefore and recovery of the primitive variables fails. This problem is tackled by the inclusion of an artificial, low and constant density atmosphere which surrounds the compact objects. Nevertheless, it is considered one the drawbacks of the HRSC schemes. In this work, the atmosphere is set to 10^{-11} , 10^{-12} in CU units.

5.3 EOS_Omni

Equations of state are particularly important for hydrodynamics, and it is imperative to have a general interface which performs all the required EOS related calls for each grid point throughout the calculations. In the Einstein Toolkit, the job is carried out by the EOS_Omni module, which provides a unified EOS interface and implements simple analytic EOS, in addition to complex microphysical EOS.

EOS for the study of neutron stars are usually generated under the assumption of zero temperature,[?]. This assumption is justified for cold and old neutron stars, however in BNS simulations, shocks will drive the temperature up, resulting to a thermal pressure component. Since we are interested in investigating phenomena which take place in BNS mergers, thermal effects cannot be omitted. For this reason, in our study, we employed hybrid EOS, where the description of cold matter is given either by *piecewise polytropes* or *cold tabulated* EOS, while an *ideal gas* EOS component, also known as *thermalgammalaw* EOS, is added to account for the thermal effects. Thus, we restrict ourselves in elaborating on the aforementioned EOS, while for more information regarding EOS_Omni the reader is encouraged to refer at the corresponding documentation page [?].

Piecewise Polytropic EOS

A common tactic in the study of neutron stars' cold matter is to parametrize the EOS as a piecewise polytrope, where each segment is given analytically by a polytropic EOS (5.85). These EOS are extremely beneficial, since they can be used in the construction of realistic models, while simultaneously are computationally less expensive than tabulated EOS, due to the absence of multiple scans over the tables. The pressure of each piece is given by

$$P_{cold}(\rho) = K_i \rho^{\Gamma_i}, \quad \text{for } \rho_{i-1} < \rho < \rho_i, \quad (5.85)$$

K_i and Γ_i being the *polytropic constant* and *adiabatic index* of each segment. Given the initial value of K_0 , the rest $K_{i \neq 0}$ are determined from the continuity conditions of $P_{cold}(\rho)$ as

$$K_{i+1} = K_i \rho_i^{\Gamma_i - \Gamma_{i+1}}. \quad (5.86)$$

In addition, in polytropes ϵ is given analytically as

$$\epsilon_{cold} = \frac{K_i}{\Gamma_i - 1} \rho^{\Gamma_i - 1}, \quad \text{for } \rho_{i-1} < \rho < \rho_i. \quad (5.87)$$

In this work we focus on 7-segment piecewise polytropes following the work of [?, ?], Tab. ?? shows the corresponding parameters of each EOS. The four low-density pieces of each EOS are identical, and come from the fitting of the crust and low density matter modelled with the SLy EOS [?], in contrast to the three high density pieces which vary for each parametrized EOS.

Cold Tabulated EOS

Cold tabulated EOS, behave similarly to the previous case, however, P_{cold} and ϵ_{cold} is now extracted from tables on the fly. To accomplish this task, we modified the current implementation of EOS_Omni. In our version, the user must supply the code with a four-column cold tabulated EOS file, including a preamble from which all parameters are read. The columns, in increasing order, are the following and must be given in *ET* units:1) *internal specific energy* ϵ , 2) *adiabatic index* Γ ³, 3) *speed of sound* of the cold component of the EOS, defined as $c_s = \frac{dP}{d\epsilon(\rho)}$, 4) *pressure*

³The adiabatic index is defined $\Gamma = \frac{\epsilon + P}{P} \frac{dP}{d\epsilon}$, while, in our implementation Γ is not really used, but we kept it since it was originally introduced by the Einstein Toolkit.

$P(\rho)$. The preamble sets the total number of points in EOS, 'Nrho', and the range in baryon rest mass density, namely 'RhoMin', 'RhoMax' in *cgs* units, from which the array of $\log_{10} \rho$ is constructed using log spacing as

$$\log_{10} \rho_i = \log_{10} \rho_{\min} + i \cdot \frac{\log_{10} \rho_{\max} - \log_{10} \rho_{\min}}{\text{Nrho} - 1}. \quad (5.88)$$

The parameter 'GammaTh' stands for Γ_{th} of the thermal gamma law, while the rest of the parameters, 'Kappa' and 'HeatCapacityE', refer to the current implementation of EOS_Omni and are not used throughout our study, nevertheless, are kept intact. An example EOS file is presented at Tab. ??, whereas some of the tables employed in this work are uploaded at the web-page of the *Gravitational Waves Group of A.U.TH.* [1] to facilitate the reproducibility of our results.

In our approach, we perform either Lagrange or Hermite interpolations [?], [?] to the original tables prior to the runs, and linear interpolations, whilst being less accurate are faster than the aforementioned, to obtain the required intermediate values throughout the evolution. Alternatively, one can perform the more accurate interpolation schemes, Lagrange and Hermite, directly to original tables during the evolution, however, Hermite interpolation schemes require the knowledge of partial derivatives of the EOS's variables, which can proven to be rather computationally expensive and error prone.

EoSType = Tabulated
Nrho = 100NYe = 1 NT = 1
RhoMin = 1e-17RhoMax = 0.013
HeatCapacityE = 1
GammaTh = 2
Kappa = 1
RhoSpacing = Log
1.57382773480208e-10 3.77878557909129e+00 5.40358988738925e-05 1.96461236073586e-26
2.08465489315302e-09 2.82041891837468e+00 6.52899872974996e-05 5.45063586370432e-26
⋮
1.70975003817364e+00 2.54362786815193e+00 1.10890637724553e+00 2.41276935680173e-02
2.87419790551732e+00 2.41769308040555e+00 1.12649562394688e+00 5.77582921394838e-02

Table 5.2: Example file for cold tabulated EOS

Thermal Effects

As previously mentioned, in our study the thermal pressure component is approximated by an ideal gas as

$$P_{th}(\rho, \epsilon) = (\Gamma_{th} - 1)\rho \epsilon_{th}, \quad (5.89)$$

where ϵ_{th} is the internal specific energy associated with the thermal effects, and defined as

$$\epsilon_{th} = \epsilon - \epsilon_{cold}. \quad (5.90)$$

Thus, the pressure (general EOS) is given by

$$P(\rho, \epsilon) = P_{cold}(\rho) + P_{th}(\rho, \epsilon). \quad (5.91)$$

In terms of the EOS_Omni module, the former case is referred as 'Hybrid' EOS, while the latter is 'Cold_tabulated' EOS. Tab. 5.3 illustrates the corresponding parameter sections which are required for calling these EOS during the evolution. Note that not only do we have to properly define the EOS_Omni parameters but also the GRHydro ones, namely 'GRHydro_eos_type' and 'GRHydro_eos_table', to be compatible with EOS_Omni.

<pre> EOS_Omni::hybrid_gamma_th = 1.8 EOS_Omni::n_pieces = 7 EOS_Omni::hybrid_gamma[0] = 1.58425 EOS_Omni::hybrid_gamma[1] = 1.28733 EOS_Omni::hybrid_gamma[2] = 0.62223 EOS_Omni::hybrid_gamma[3] = 1.35692 EOS_Omni::hybrid_gamma[4] = 3.005 EOS_Omni::hybrid_gamma[5] = 2.988 EOS_Omni::hybrid_gamma[6] = 2.851 EOS_Omni::hybrid_k0 = 1.685819e2 EOS_Omni::hybrid_rho[0] = 3.951156e-11 EOS_Omni::hybrid_rho[1] = 6.125960e-07 EOS_Omni::hybrid_rho[2] = 4.254672e-06 EOS_Omni::hybrid_rho[3] = 0.0002367449 EOS_Omni::hybrid_rho[4] = 8.114721e-04 EOS_Omni::hybrid_rho[5] = 1.619100e-03 GRHydro::GRHydro_eos_type = "General" GRHydro::GRHydro_eos_table = "cold_tabulated" </pre>	<pre> EOS_Omni::coldeos_read_table = yes EOS_Omni::coldeos_use_thermal_gamma_law = yes EOS_Omni::coldeos_table_name = "path_to_EOS_table" GRHydro::GRHydro_eos_type = "General" GRHydro::GRHydro_eos_table = "cold_tabulated" </pre>
--	--

Table 5.3: Parameter file sections to induce 'Hybrid'(left) or 'Cold_tabulate' (right) EOS. The 'Hybrid' EOS's section refers to the parametrization of SLy introduced in [?].

5.4 Hydro_RNSID

Throughout our simulations of single stars, we generate initial models for either static or differentially rotating relativistic stars using the Hydro_RNSID module. The latter is based on the RNS code, which constructs models for rapidly rotating relativistic compact stars using the method written by KEH [], in addition to the modifications introduced by Cook, Shapiro & Teukolsky (1994) []. It supports polytropic or zero-temperature tabulated EOS, while we concentrate only on the second case. Further information regarding the code, and its publicly available version (apart from the one of ET) [?], can be found at [?], [?], whereas a review article on this subject at [?].

The Hydro_RNSID generates initial data (ID) compatible to Hydro_Base and GRHydro (these modules are the two prerequisites, necessary for its successful activation) by defining the following required quantities. The user must specify the central rest mass density ρ_c (unlike the case of the original version of RNS where the central energy density ε_c/c^2 must be specified), the oblateness of star namely the axes ratio, and one of the two kinds of rotational profiles currently implemented. These are 'uniform' for uniformly rotating stars and 'diff' for differentially rotating stars, where the latter is described by j -law profile

$$\Omega_c - \Omega = \frac{1}{A^2 r_e^2} \left[\frac{(\Omega - \omega)r^2 \sin^2 \theta e^{-2\nu}}{1 - (\Omega - \omega)^2 r^2 \sin^2 \theta e^{-2\nu}} \right], \quad (5.92)$$

where $\Omega = u^\phi/u^t$ is the angular velocity measured by an observer at infinity, Ω_c is its value on the rotation axis, r_e is the equatorial radius, and A is a parameter with units of length and controls the magnitude of the differential rotation. Note that when $A \rightarrow \infty$ (5.92) reduces to uniform rotation.

The format of the zero-temperature tabulated EOS follows the one introduced by the original RNS code, and assumes that all quantities are given in *cgs* units. The first line in the EOS file must contain the number of the tabulated points, while the remaining lines should consist of four columns: energy density ε/c^2 (gr/cm^3 units), pressure P (dyn/cm^2), log-enthalpy H (cm^2/s^2), and baryon number density n_B ($1/cm^3$). The log-enthalpy is defined as

$$H(P) = \ln h = \int_0^P \frac{c^2 dP'}{\varepsilon + P'}. \quad (5.93)$$

As previously mentioned, Hydro_RNSID not only does generate rotating star equilibrium solutions but also static star solutions, by imposing uniform rotation, oblateness equal to one,

and $A = 0$. In Tab. 5.4 we present the parameter file sections for the construction of static and rapidly differentially rotating star solutions for the DD2 [?] EOS, which was extracted from the general 3-dimensional (dependence on ρ, T, Y_e) EOS by imposing beta-equilibrium on zero-temperature 2-dimensional slice. The corresponding ”/par/eos_files” directory contains example EOS files (APR4pp, H4pp, MS1pp, SLypp) compatible with Hydro_RNSID, which can be combined with EOS_Omni ’Hybrid’, since these come from the piecewise polytropic parametrization [?], to properly generate initial models and subsequently evolve them with GRHydro.

Hydro_rnsid::accuracy = 1e-11	Hydro_rnsid::accuracy = 1e-11
Hydro_rnsid::rho_central = 0.00128	Hydro_rnsid::rho_central = 0.0025
Hydro_rnsid::axes_ratio = 1	Hydro_rnsid::axes_ratio = 0.7
Hydro_rnsid::rotation_type = ”uniform”	Hydro_rnsid::rotation_type = ”diff”
Hydro_rnsid::A_diff = 0	Hydro_rnsid::A_diff = 1.0
Hydro_rnsid::model2D_file = ”/path_to_ID/DD2.h5”	Hydro_rnsid::model2D_file = ”/path_to_ID/DD2.h5”
Hydro_rnsid::eos_file = ”/path_to_EOS/DD2”	Hydro_rnsid::eos_file = ”/path_to_EOS/DD2”
Hydro_rnsid::eos_type = ”tab”	Hydro_rnsid::eos_type = ”tab”

Table 5.4: Example parameter file sections for ”Hydro_RNSID”, for the construction of static star models (left) and differentially rotating star models (right) with the tabulated zero-temperature DD2 EOS.

Lastly, Hydro_RNSID allows the user to save, and thus recover the generated initial model from a binary file (h5), where its location is specified in the respective parameter file. For more information on this topic and how to do it, the user should refer to the module’s documentation page, or to the parameter files used for this work which can be found at [?]. Lastly, a highly useful tool for the preparation of the simulations and testing of the initial model’s parameters, is the self-executable version of the Hydro_RNSID, namely RNS, which accepts the same parameters as the module and is capable of creating a binary file of the initial data that can be directly imported into the evolution code. In our repository, we provide a python script which determines the required ρ_c (input of Hydro_RNSID) for the given ε_c/c^2 (input of RNS), to facilitate the comparison between results obtained with RNS, and the Hydro_RNSID.

5.5 BNS Initial Models

For our study of BNS mergers we followed a different approach, where the initial models were externally generated using the open-source library LORENE [?], and subsequently imported to the evolution code using the Meudon_NS module. The reader is strongly advised to refer to the original publication [?] for a thorough and elaborate description of the method. In the following, we briefly present the method’s characteristics, and the instructions for setting up the initial BNS configurations, compatible to the Einstein Toolkit, using tabulated EOS.

LORENE constructs quasi-equilibrium configurations of BNS models on the verge of coalescence (pre-coalescing stage), based on the idea that in the late inspiral phase, prior to orbital instabilities or merging of the two companions, the evolution of BNS can be approximated by a succession of circular orbits. The hydrodynamical flow, or the specific velocity state, is either rigidly rotational or irrotational, while for our study, we opted for the latter state, since it is considered a more realistic depiction of the late evolutionary stages of a BNS system [?]. As for the gravitational field, a conformally flat metric is assumed introduced by [?], in order to reduce the computational cost. Employing a multi-domain spectral method, the system of

partial differential equations is numerically solved, while two spherical coordinate systems are used, with their origins placed at the center of the stars respectively.

To generate the initial BNS models the user must specify several parameters, while the most significant are: the EOS, the rotation state (rigidly rotating or irrotational flow), the coordinate distance d between the stellar centres, the initial central enthalpy of each star $\mathcal{H}_{(1)}^c$ and $\mathcal{H}_{(2)}^c$, the final baryon masses $M_{(1)}$ and $M_{(2)}$ of each star. Once the aforementioned parameters are set, the initial conditions for the iterative procedure are given as two solutions for spherically symmetric, static, and isolated neutron stars of respective central enthalpies $\mathcal{H}_{(1)}^c$ and $\mathcal{H}_{(2)}^c$. Subsequently, the iterative procedure described in [?], proceeds on the construction of the quasi-equilibrium solutions. However, since gravitational waves from BNS systems have already been detected [], simulations with exact values of observed BNS models are required, hence, one should be able to specify the mass (either baryon mass or gravitational mass) configurations of BNS. The current implementation of LORENE modifies the enthalpies at each iteration, so the the solutions converges to the given baryon masses, while for our models, we employ a python script which numerically determines the proper baryon masses (depending on the given coordinate distance) for the given gravitational masses, using a simple Newton Raphson type method. It is worth to mention that LORENE, deploys the fact that the baryon mass is an increasing function of the central enthalpy for stable star solutions, thus, the latter can be adjusted at each iteration by the multiplication of itself with a factor defined as

$$\eta \equiv \left(\frac{2 + \zeta}{2 + 2\zeta} \right)^{1/4}, \quad (5.94)$$

where ζ stands for the relative error between the step's baryon mass $M_B^{(1)J}$ and the given baryon mass $M_B^{(1)}$, defined as

$$\zeta \equiv \frac{M_B^{(1)J}}{M_B^{(1)}} - 1. \quad (5.95)$$

It is apparent that when the iterative procedure converges, the factor η equals to one. One could argue that this method can also be applied to specify the gravitational masses of the two stellar objects, without imposing significant alterations to the current, however, even though we do agree with it, we have not yet tested it.

LORENE supports several types of EOS ⁴, while we focus on EOS tables compatible with the standard LORENE format. The latter, is comprised by a preamble which establishes the total number of lines of the table, followed by four columns, which are, in increasing order, the number of each line i , the baryon density n_b ($1/fm^3$), the energy density ε/c^2 (gr/cm^3), pressure p (dyn/cm^2). It should be noted that the current version (as of today) expects a format similar to the one of Tab. 5.5, meaning that the deletion of first five lines (which for the example are commented) leads to unsuccessful import of the table. In addition, for this particular study, we incorporated part of LORENE2, which is a modified by University of Parma version of LORENE, to our code to ultimately ensure thermodynamic consistency of the EOS. This is accomplished by slightly altering the EOS during the generation of the initial models. To be more precise, instead of computing the specific enthalpy h as

$$h = \frac{\varepsilon + p}{\rho} = \epsilon + 1 + \frac{p}{\rho}, \quad (5.96)$$

we substitute it by the expression derived from the second thermodynamic law (for the isentropic case) given by

$$h = \exp \left[\int_0^p \frac{dp'}{\varepsilon + p'} \right]. \quad (5.97)$$

⁴Refer to "Lorene::Eos Class Reference" [] for an extensive list of the available EOS related routines.

Generated by Theodoros Soulantasis
#
#
#
#
200 # number of lines
#
i $n_b [fm^{-3}]$ $\varepsilon/c^2 [g/cm^3]$ $p [dyn/cm^2]$
#
1 4.69879518072300e-15 7.80000000001800e+00 1.01000000036800e+08
2 5.56008759164000e-15 9.22974540209700e+00 2.34166897028300e+11
⋮
199 1.39578979405400e+00 3.55258020053000e+15 1.78789649724900e+36
200 1.65163903000000e+00 4.65235218049900e+15 2.66320000000000e+36

Table 5.5: Example file for cold tabulated EOS compatible with LORENE's standard format.

This modification is proven to be crucial for the successful converge to a solution for some of the used EOS. LORENE2 can be obtained from Gravitational Physics Group of Parma's University website [?], whereas the above modification is stored at the "eos_tabul.C" file. To specify the type of tabulated EOS, and the directory where it is located, the user must provide two parameter files (one for each stellar object), namely, par_eos1.d and par_eos2.d, as is shown in Tab. 5.6. The next step, is to define the parameters related to the computational grid of each star, such as the total number of domains, the domains inside the star, and the number of points for each domain, as showed in Tab. 5.6.

17 Type of the EOS (cf. documentation of Eos::eos_from_file) 0 Standard Lorene format Tabulated EOS path_to_eos/eos_sly4-pp.d	# Multi-grid parameters ##### 4 nz: total number of domains 1 nzet: number of domains inside the star 21 nt: number of points in theta (the same in each domain) 20 np: number of points in phi (the same in each domain) # Number of points in r and (initial) inner boundary of each domain: 33 0. ← nr & min(r) in domain 0 (nucleus) 33 1. ← nr & min(r) in domain 1 33 1.5 ← nr & min(r) in domain 2 33 2.5 ← nr & min(r) in domain 2
--	--

Table 5.6: Example parameter files for par_eos1.d (par_eos2.d) on the left, and par_grid1.d (par_grid2.d) on the right.

Physical parameters for the binary initial conditions ##### 1 ← 1 for a relativistic computation, 0 for a Newtonian one 100. ← coordinate distance between the two stellar centers [km] 0.20714286 ← initial central enthalpy of star 1 1 ← rotational state of star 1 : 1 = irrotational, 0 = corotating 0.20714286 ← initial central enthalpy of star 2 1 ← rotational state of star 2 : 1 = irrotational, 0 = corotating
--

Table 5.7: Example parameter file for par_init.d.

Lastly, the physical parameters which dictate the properties of the initial stars, the isolated solutions, have to be defined in the par_init.d, while the numerous parameters closely connected with the numerical method are set in parcoal.d, both of which are shown in Tab. 5.7, Tab. 5.8.

The criterion for the evaluation of the iterative procedure's convergence is the relative difference between the enthalpy fields of the two successive steps given by

$$\delta\mathcal{H} \equiv \frac{\sum_i |\mathcal{H}^J(x_i) - \mathcal{H}^{J-1}(x_i)|}{\sum_i \mathcal{H}^{J-1}(x_i)}, \quad (5.98)$$

```
# Parameters for the binary equilibrium computation by coal
#####
ini.d
0.4    fact_separ : Factor by which the initial separation is multiplied
1.4    mbar_voulue[0] : Baryon mass required for star 1 [M_sol]
1.4    mbar_voulue[1] : Baryon mass required for star 2 [M_sol]
#####
300    mermax : Maximum number of steps in the main iteration
0.5    relax : Relaxation factor in the main iteration
1      mermax_eqb : Maximum number of steps in Etoile_bin::equilibrium
1      prompt : 1 if no pause during the computation
0      graph : 1 if graphical outputs during the computation
1.E-6   seuil : Threshold on the enthalpy relative change for ending the computation
2      fmer_stop : Step interval between pauses in the main iteration
1      fmer_save : Step interval between safeguards of the whole configuration
4      mermax_poisson : Maximum number of steps in Map_et::poisson
1.5    relax_poisson : Relaxation factor in Map_et::poisson
4      mermax_potvit : Maximum number of steps in Map_radial::poisson_compact
0.5    relax_potvit : Relaxation factor in Map_radial::poisson_compact
25    mer_masse : Step from which the baryon mass is forced to converge
0.5    aexp_masse : Exponent for the increase factor of the central enthalpy
4      fmer_udp_met : Step interval between metric updates
1      ind_rel_met : 1 if relaxation of the metric, 0 if not
0.75   relax_met : Relaxation factor of the metric (used only if ind_rel_met=1)
0.75   relax_omeg : Relaxation factor on Omega (orbital angular velocity)
0.5    fact_omeg_min : fact_omeg_min * omega = low bound in the omega search
1.5    fact_omeg_max : fact_omeg_max * omega = high bound in the omega search
0.     thres_adapt1 : threshold on dH/dr for the adaptation of the mapping in star 1
0.     thres_adapt2 : threshold on dH/dr for the adaptation of the mapping in star 2
0.6    reduce_shift : factor by which the initial analytical shift is reduced
```

Table 5.8: Example parameter file for parcoal.d.

where the summation is carried out throughout all the points inside the star, and J stands for the iteration step. Its typical value is often $\delta\mathcal{H} = 10^{-6}$, or $\delta\mathcal{H} = 10^{-11}$ for high precision simulations.

- Muedon_NS

Chapter 6

Results

In this chapter, we report the results of this work. First, we present the procedure we followed in order to evaluate the code, that is to identify the radial oscillation-mode fundamental frequencies through simulations of single stars. Next, a low resolution toy model BNS simulation for equal mass companions is presented using the cold, tabulated EOS LS220. Finally, we show the results of our high resolution BNS simulations, ran in ARIS HPC infrastructure, using the piecewise polytropic EOS MPA1, and APR4. We performed four simulations in total where for each EOS two configurations were considered: a) equal mass companions ($q = m_2/m_1 = 1$) b) unequal mass companions ($q = 0.73$). The masses were fixed by imposing a chirp mass $M_{\text{chirp}} = 1.186 M_\odot$ as the one reported by the recent BNS merger observation GW170817 [?]. In all of the simulations (single stars, BNS), the system is evolved in 3-dimensions with the exception of reflection symmetry with respect to the equatorial plane (XY).

6.1 Oscillation modes

There are several publications using the Einstein Toolkit [?, ?, ?, ?], however as far as we know, there has not been a systematic study of the frequency modes associated with radial pulsations of compact objects in the Einstein Toolkit. Based on previous studies of the radial oscillation modes [?, ?, ?], we attempt to expand our understanding and evaluate the code by reproducing results taken from literature. This serves as a test of our implementation, while in the future we will perform more systematic calculations.

In theory, equilibrium solutions of single stars (spherical for non-rotating stars and axis-symmetrical for rotating stars), generated by the initial data modules, are stationary. However, this is not the case for numerical simulations, where truncation errors of the finite-differencing schemes at the surface of the star trigger small amplitude radial oscillations. These pulsations might be amplified in our case where we employ a Cartesian grid and therefore the surface of the given star cannot be an exact sphere. In the following, we investigate these radial pulsations using piecewise polytropic, and cold tabulated, EOS.

6.1.1 Piecewise polytropic EOS

We verified the compatibility between the the newly added (with respect to the moment this work started) Hydro_RNSID module with EOS_Omni, and GRHydro modules by performing a series of 3-dimensional simulations of single non-rotating stars using the 7-segment piecewise polytropic EOS SLy, H4, according to the prescription introduced in [?]. In the latter, numerous EOS have been parametrized as 7-segment piecewise-polytropic EOS, where the low-(rest mass) density section (4-parts) is the same for all the parametrized EOS. The associated parameters

of each segment, namely, the polytropic exponent Γ_i , the rest mass density density ρ_i , and the polytropic constant K_0 ($K_{i \neq 0}$ are obtained automatically via continuity of pressure in the Einstein Toolkit) are displayed in Tab. 6.1, Tab. 6.2. Fig. (6.1) illustrates the pressure to rest mass density relation for SLy and H4 in high density regime.

i	ρ_i [g/cm ³] (CU)	K_i [cgs] (CU)	Γ_i
0	—	$6.801 \cdot 10^{-11}$ (168.5819)	1.584
1	$2.440 \cdot 10^7$ ($3.951 \cdot 10^{-11}$)	—	1.287
2	$3.784 \cdot 10^{11}$ ($6.126 \cdot 10^{-7}$)	—	0.622
3	$2.628 \cdot 10^{12}$ ($4.255 \cdot 10^{-6}$)	—	1.357

Table 6.1: Parameters of the piecewise polytropic EOS for the low rest mass density regime. The quantities are expressed in *cgs* or computational units (CU) denoted by the the parenthesis. $K_{i \neq 0}$ are omitted since they are obtained through the continuity of the pressure.

	ρ_4	ρ_5	ρ_6
SLy	$1.462 \cdot 10^{14}$ ($2.368 \cdot 10^{-4}$)	$5.012 \cdot 10^{14}$ ($8.115 \cdot 10^{-4}$)	$1.000 \cdot 10^{15}$ ($1.619 \cdot 10^{-3}$)
H4	$8.879 \cdot 10^{13}$ ($1.438 \cdot 10^{-4}$)	$5.012 \cdot 10^{14}$ ($8.115 \cdot 10^{-4}$)	$1.000 \cdot 10^{15}$ ($1.619 \cdot 10^{-3}$)
	Γ_4	Γ_5	Γ_6
SLy	3.005	2.988	2.851
H4	2.909	2.246	2.144

Table 6.2: Parameters of the piecewise polytropic EOS for SLy, H4. The units are in the same format as Tab. 6.1.

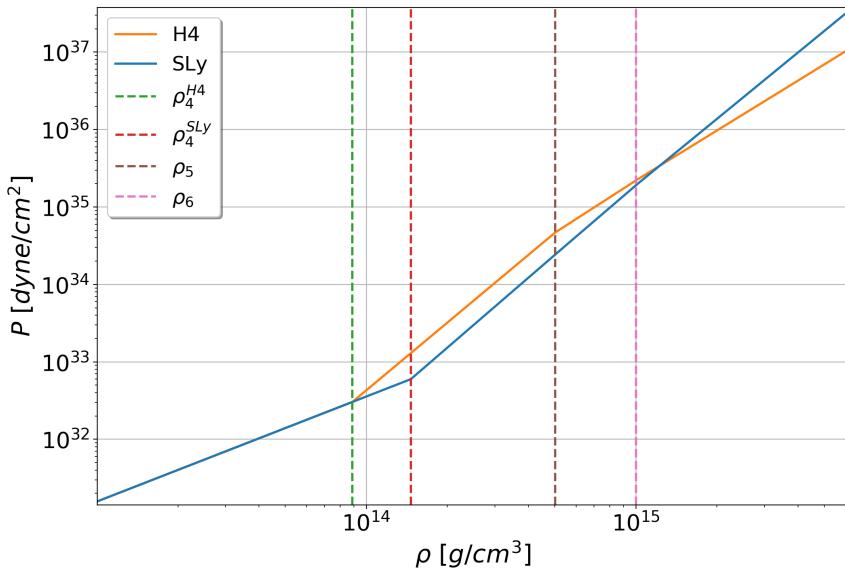


Figure 6.1: Pressure to rest mass density relation for the piecewise polytropic EOS, SLy and H4. The dashed lines denote the densities, ρ_4^{SLy} , ρ_4^{H4} , ρ_5 , ρ_6 , where the adiabatic index Γ_i changes its value.

Furthermore, Fig. 6.2 illustrates the gravitational mass M to circumferential equatorial radius R_e relation for SLy and H4, which predict maximum masses $M_{SLy}^{max} = 2.04 M_\odot$ and $M_{H4}^{max} = 2.06 M_\odot$ respectively. These values are compatible with the lower limit of the allowed maximum mass given by [?], and also, are in agreement with the recently proposed by [?] limits. In the latter publication, the authors propose a method based on GW170817 [], to constraint

the radius of the non-rotating neutron star. More precisely, the radius $R_{1.6}$ of non-rotating neutron star with gravitational mass $M = 1.6 M_\odot$ has to be larger than roughly 10.7 km, while the radius R_{max} of the maximum mass configuration (per EOS) must be above 9.6 km. The aforementioned non-rotating neutron star sequences were computed with the RNS code (original version), where the piecewise polytropic EOS were supplied in tabulated form.

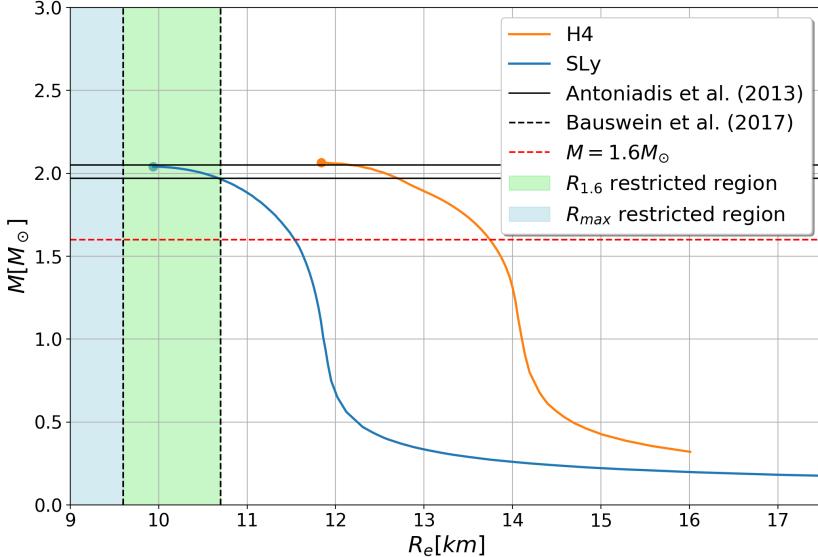


Figure 6.2: Gravitational mass M to circumferential equatorial radius R_e relation for the piecewise polytropic EOS, SLy and H4. Black, dashed horizontal lines represent the lower limits (including uncertainties) of the maximum mass configuration, black vertical lines represent the constraints set on the minimum accepted neutron star radius $R_{1.6}$ and R_{max} . The circular points denote the corresponding maximum mass predicted by SLy and H4.

First we considered a non-rotating star using SLy as a toy model. Since the initial configurations were generated with Hydro_RNSID, on the limit of uniform rotation with $\Omega = 0$, we test the module's compatibility by examining the star's properties throughout its evolution. Our model has an intermediate gravitational mass, $M = 1.2072 M_\odot$, and circumferential radius $R_e = 11.7983$ km, while the rest of the parameters are shown in Tab. 6.3.

Toy model using SLy					
ρ_c [g/cm ³] (CU)	ε/c^2 [g/cm ³] (CU)	M [M_\odot]	M_0 [M_\odot]	R_e (CU)	r
$7.906 \cdot 10^{14} (0.00128)$	$8.575 \cdot 10^{14} (0.001388)$	1.2072	1.311	11.7983 (7.9903)	1

Table 6.3: Parameters of the toy model, non-rotating star using the piecewise polytropic EOS SLy: rest mass central density ρ_c , central energy density ε_c/c^2 , gravitational mass M , baryonic mass M_0 , circumferential equatorial radius R_e , oblateness r . The units follow the format introduced in the previous figures.

In order to determine the appropriate grid spacing for the subsequent simulations we ran four test simulations from low to rather high resolution. To minimize some of the computational cost, we included refinement levels in our grid, where in each of them the distance between successive points dx (dy, dz) is divided by a factor of 2 (for a given grid with dx and rl refinement levels, the finest region corresponds to $dx^{\text{finest}} = dx/2^{rl}$), with the finest region located inside the star. In our test simulations, the finest grid spacings were $dx = (0.184, 0.368, 0.738, 1.475)$ km, or $dx = (0.125, 0.25, 0.5, 1.0)$ CU, and for simplicity we refer as "high resolution" simulations

when $dx = (0.184, 0.368)$ km and "low resolution" simulations when $dx = (0.738, 1.475)$ km. The refinement levels varied for each case, given the fact that the high resolution simulations and especially the one with $dx = 0.184$ km were computationally, highly demanding. Nevertheless, at minimum, a grid-radius of at least two times the radius of the star was considered. Lastly, the evolution is carried out in 3-dimensions with the exception of reflection symmetry imposed along the equatorial plane.

In Fig. 6.3, we see the time evolution of central rest mass density (normalized to its value at $t = 0$) for the test simulations. Clearly, the central rest mass density is subject to oscillations, where for low resolution simulations their amplitudes are larger than the ones of the high resolution simulations. These oscillations are connected, as we have already mentioned, with the radial oscillation modes of the star and originate from truncation errors. Moreover, one can see that the central rest mass density ρ_c has also an additional drift, which increases as the resolution lessens. This may be caused by the artificial atmosphere (of constant density) which surrounds the star, in conjunction with the fact that in crude grids the finite differencing schemes at surface of the star lead to larger errors. As stated in [?], the artificial atmosphere increases the damping of these pulsations, which can be seen for the low resolution simulation with $dx = 1.475$ km. Thus, it is safe to assume that the one with $dx = 1.475$ km is not reliable for long term simulations, and even though the drift is certainly smaller for the case of $dx = 0.738$ km, it is not negligible especially for long term evolution. The high resolution simulations have certainly small enough oscillation amplitudes, and the drift effect is greatly lessened. However, the highest one with $dx = 0.184$ km, is computational expensive and running long term simulations is rather difficult, therefore, grid spacing of $dx = 0.738$ km is the proper choice for the subsequent single star simulations.

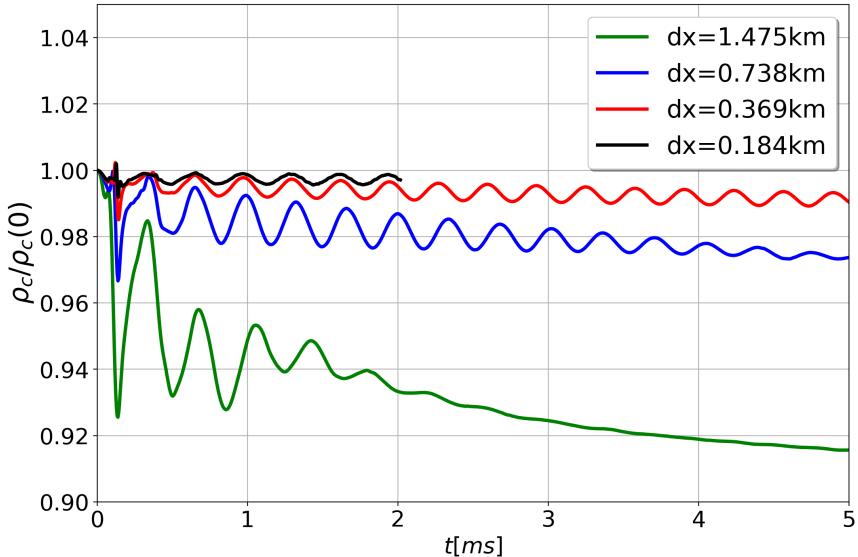


Figure 6.3: Time evolution of the normalized to its initial value central rest mass density ρ_c for the test simulations.

Next, using the SLy model with $dx = 0.369$ km (0.25 CU), we examine the hydrodynamics variable profiles for different time snapshots in order verify their expected behaviour. In Fig. 6.4 we see the pressure, rest mass density, and internal specific energy profiles during the evolution of the toy model, at $t = (0.0, 2.22, 4.80)$ ms. As it is expected, the pressure and rest mass density profiles are rather stable (no substantial amount of matter leaves the surface of the star), while the specific internal energy profiles indicate the slow transfer of energy from the surface of the star to the artificial atmosphere, which is again expected.

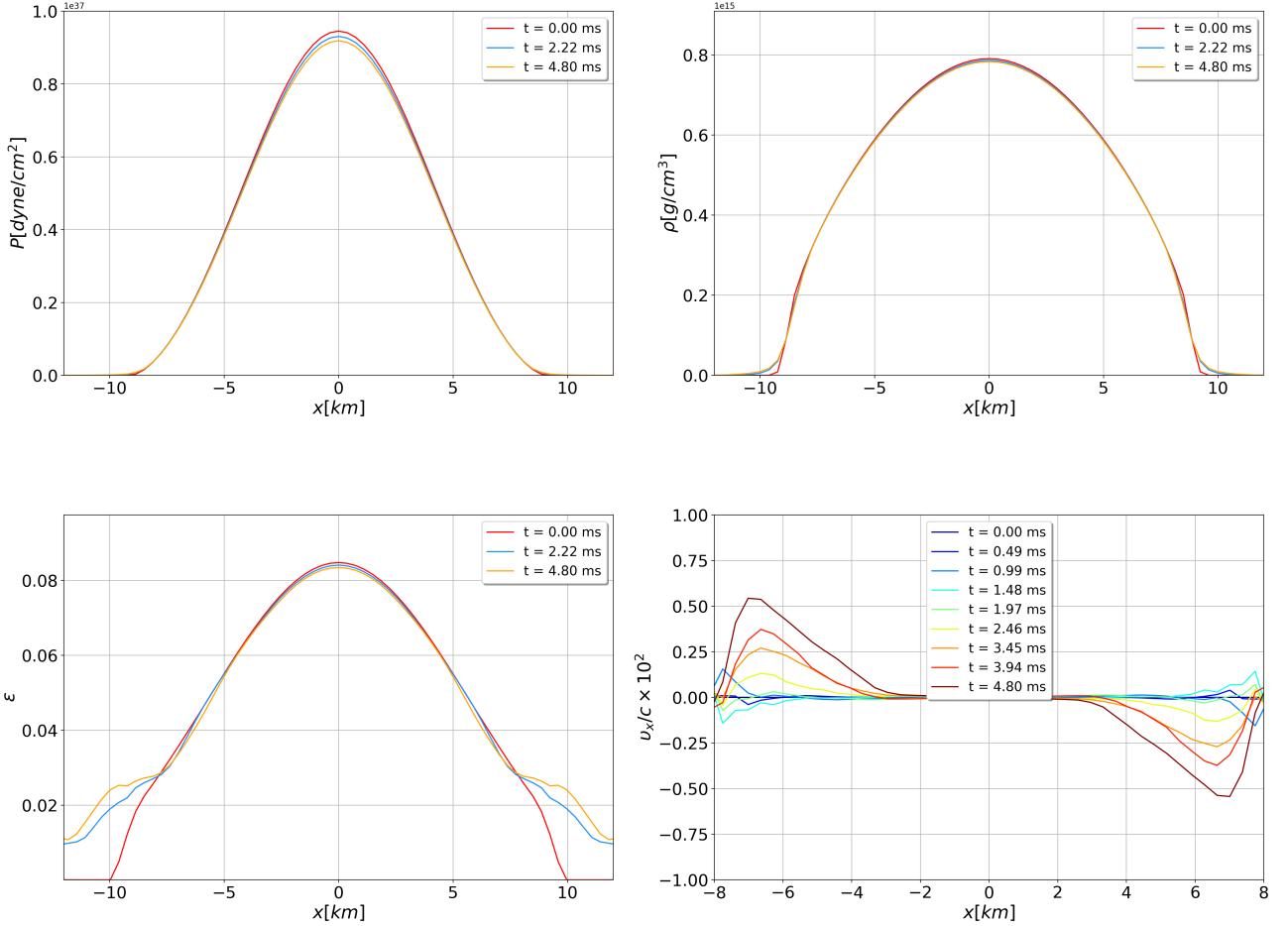


Figure 6.4: (top-left, top-right, bottom-left) Profiles of pressure, rest mass density, and internal specific energy at $t = (0.0, 2.22, 4.80)$ ms, (bottom-right) v_x -component (equivalent to the radial velocity v_r) profile for the toy model at different time-snapshots, for the toy model with $dx = 0.369$ km.

The radial fluid velocity profile, and more specifically in our case, the x -component of the fluid velocity v_x , manifests these pulsations too. As it is expected, in Fig. 6.4 we see that their amplitude is rather small (its maximum is atleast three orders of magnitude less than the speed of light). Moreover, regions with higher densities, closer to the star's core, oscillate with larger amplitudes than the ones near to the surface. Thus, we proceed on computing the associated frequencies of these oscillations.

The prescription for computing the radial oscillation mode associated frequencies through numerical simulations is rather simple. One has to monitor the time evolution of the rest mass density or radial velocity of the fluid (v_i in our case since $\Omega = 0$), and subsequently Fourier transform to obtain the frequency spectrum. As for the rest-mass density, it is more appropriate to use the time-evolution of the central rest mass density, for which the effect of the radial oscillation is maximal. On the contrary, the radial velocity time-evolution at the center should vanish due to the large density values. Thus, an option is to examine the time evolution of a fluid element at distance close to the center, but far from the surface. The ideal combination, should be the distance where the pulsations have large enough amplitudes, but do not exhibit further numerical noise. In Fig. 6.5 we compare the radial velocity time evolution of fluid elements at three different radii, where we see that the velocities are also subject to the previously mentioned drift induced by the artificial atmosphere even for the case of $dx = 0.369$

km, making the distance of $r = 1.85$ km the right choice for the calculation the associated frequencies.

In Fig. 6.6, we present the Fourier transforms of central rest mass density, and radial velocity v_x for our toy model, from which we see their agreement in computing the fundamental frequency with a relative error of 1.17%. In the latter figure, we also used the detrended data to discard the small drift associated with the density curve, which results to a smooth frequency background since the FFT algorithm treats it as a superposition of low frequency modes. Furthermore, it should be noted that since the Fourier transform is discrete, the algorithm being the *Fast-Fourier Transform (FFT)*, the resolution in frequency domain is depends on maximum time value t_{\max} .

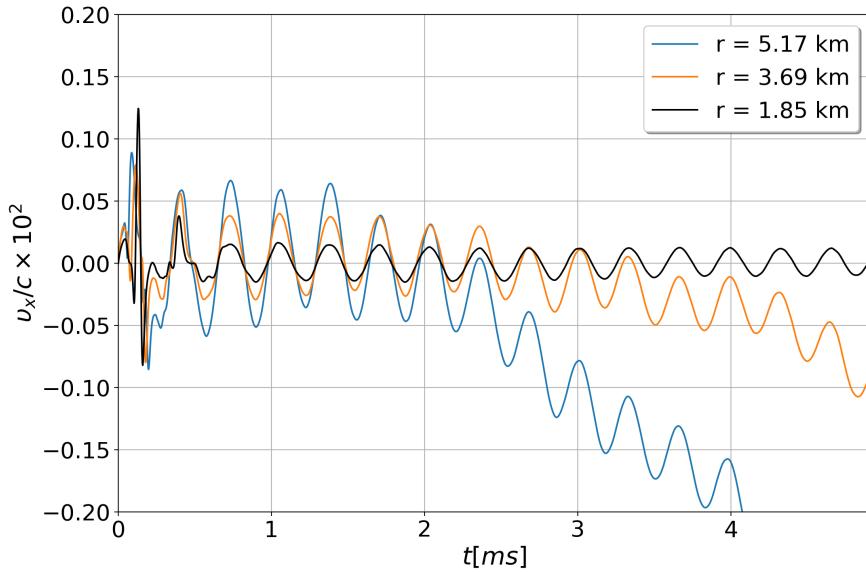


Figure 6.5: Time evolution of radial velocity v_x of fluid elements at three separate distances of $r = (1.85, 3.69, 5.17)$ km

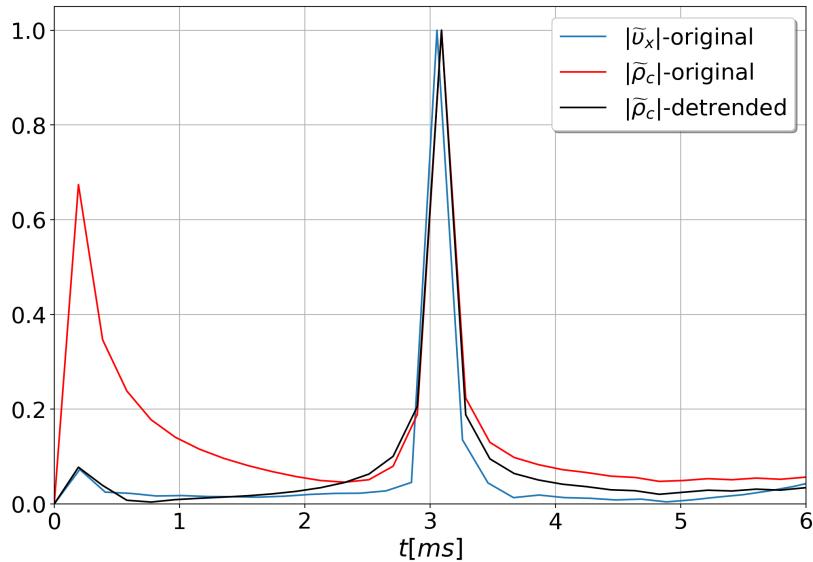


Figure 6.6: Fourier transform of the central rest mass density, the detrended central rest mass density, and radial velocity at $r = 1.85$ km.

As a next step, we considered another model using the piecewise polytropic EOS H4, its parameters are given in Tab. ??, in an attempt to identify the associated frequencies and compare them

with the ones predicted by [?]. In the latter, the authors solve the eigenvalue problem for piecewise polytropic EOS, such as H4. For this simulation, we employed a computational grid with $x_{\max} = 118.126$ km (80 CU) with five refinement levels leading to minimum $dx = 0.369$ km (0.25 CU). We exceeded the run up to $t_{\max} = 15$ ms in order to improve the resolution in frequency domain. As it is shown in Fig. 6.7 the agreement between the fundamental frequency (first peak) obtained from the simulation and the one predicted by perturbation theory, namely F , is undeniable with corresponding relative error $|F^{\text{simulation}} - F^{\text{perturbation}}|/F^{\text{perturbation}} = 0.305\%$. The rest two peaks, H1 and H2, correspond to the overtones (similar to musical instruments), where the respective relative errors are 2.742% and 4.281%. Thus, we can safely conclude that the Einstein Toolkit can be used as an other tool for validation of the results of [?]. At this point, we should mention that the generated initial model comes from Hydro_RNSID, which requires a tabulated EOS, and during the evolution of the system, EOS_Omni uses analytical expressions for the pressure, and internal specific energy. As a result, there are errors which originate from the interpolation scheme of Hydro_RNSID which can slightly alter the initial configuration, and therefore, potentially shift the frequencies of the sub-dominant oscillation modes. Moreover, in this simulation, our implementation of GRHydro was included.

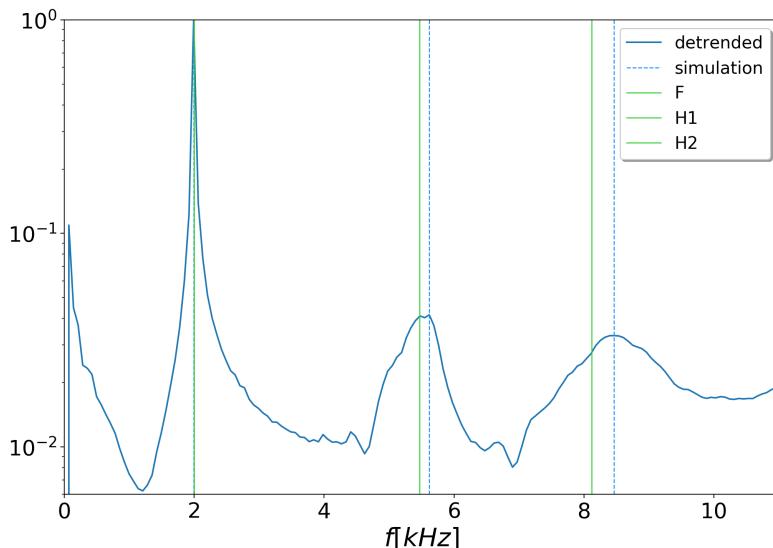


Figure 6.7: FFT on the central rest mass density for the H4 model. Lightblue, dashed, vertical lines represent the frequency peaks computed from the simulation, whereas light green, vertical lines represent the values obtained from perturbation theory.

Comparison model using H4						
ρ_c [g/cm ³] (CU)	ε/c^2 [g/cm ³] (CU)	M [M_\odot]	M_0 [M_\odot]	R_e [km] (CU)	r	
$7.8745 \cdot 10^{14}$ (0.0012749)	$8.9253 \cdot 10^{14}$ (0.001445)	1.7095	1.8970	13.5517 (9.1778)	1	

Table 6.4: Parameters of the non-rotating, comparison model using the piecewise polytropic EOS H4: The format is the same as Tab. 6.3.

Model using H4 taken from literature			
ρ_c [g/cm ³]	M [M_\odot]	M_0 [M_\odot]	R_e [km]
$7.8745 \cdot 10^{14}$	1.7090	1.8964	13.5515

Table 6.5: Parameters of the model taken from literature.

Frequencies extracted from simulation					
F [kHz]	H1 [kHz]	H2 [kHz]	error F [%]	error H1 [%]	error H2 [%]
1.991	5.620	8.465	0.305	2.742	4.281

Table 6.6: Comparison between the mode frequencies extracted from the simulation, and the ones taken from literature.

6.1.2 Tabulated EOS

In this subsection we describe the method we followed in order to properly, to the extent a MSc Thesis project allows us, implement the cold, tabulated EOS in the Einstein Toolkit. Firstly, we had to ensure the initial configurations, using Hydro_RNSID, can be successfully generated for any given (well constructed) tabulated EOS. We encountered cases where even though the RNS code [?] successfully computed the equilibrium models, the RNS executable provided with Hydro_RNSID (located in the "src" directory) in the Einstein Toolkit failed. Furthermore, the presence of the artificial atmosphere, its typical constant value being 10^{-11} (10^{-12} for the BNS simulations), imposed by GRHydro during the evolution of the system, implies that the given table must have values at these low densities, or else the code will surely crush. However, this extreme low density regime is not guaranteed for every EOS in literature. As it turned out, both of the problems can be solved by matching the given EOS table with a low density region. The idea is that the Hydro_RNSID's iterative procedure can lead to densities which exceed the boundaries of the given table and then crush. Similarly, since GRHydro's algorithm for converting the conservative variables to primitive ones is also iterative, it is subject to the same problem.

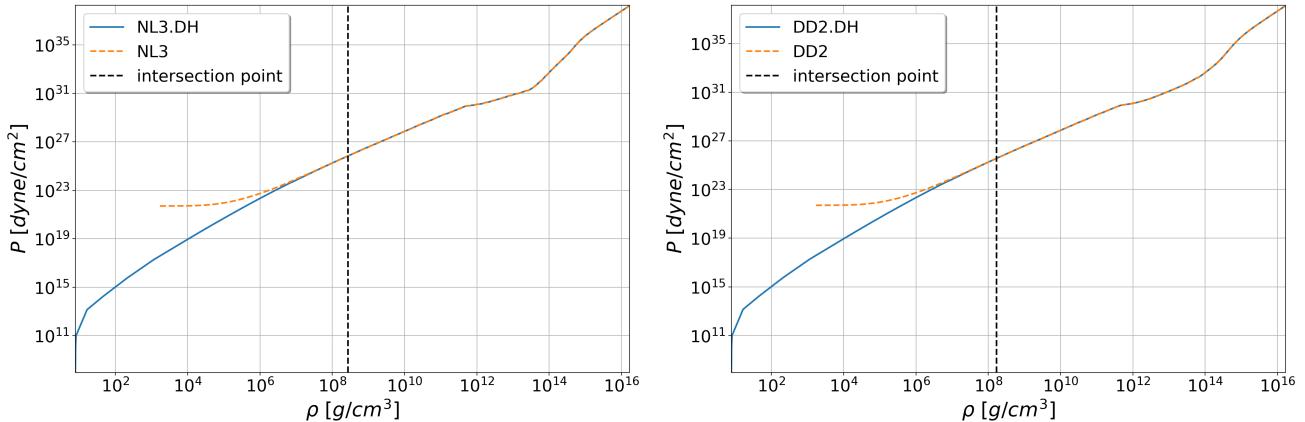


Figure 6.8: Matching of the tabulated EOS NL3, DD2, with the low density EOS DH by their intersection point. Orange lines denote the original tables, blue, dashed lines denote the final tables, and black, dashed vertical lines denote the intersection points.

Fortunately, there are two general accepted low-density EOS candidates for the description of cold matter, one introduced by Douchin and Haensel (DH) [?, ?, ?, ?] and the alternative by Negele and Vautherin [?, ?, ?]. In this work we chose the former one, and then matched it with the EOS tables which lacked the required low density regime. For the matching process, we numerically determined the intersection point on the pressure to rest mass density curves between the given EOS table and the DH EOS using the bisection method, in order to ensure the continuity of the resulted EOS table. We performed the matching for at least four tabulated EOS, namely, DD2, NL3, LS220, LS375, which we are interested in using for future works. The

extended EOS tables are denoted by the suffix ".DH", and their pressure to rest mass density curves are illustrated in Fig. 6.8, Fig. 6.9.

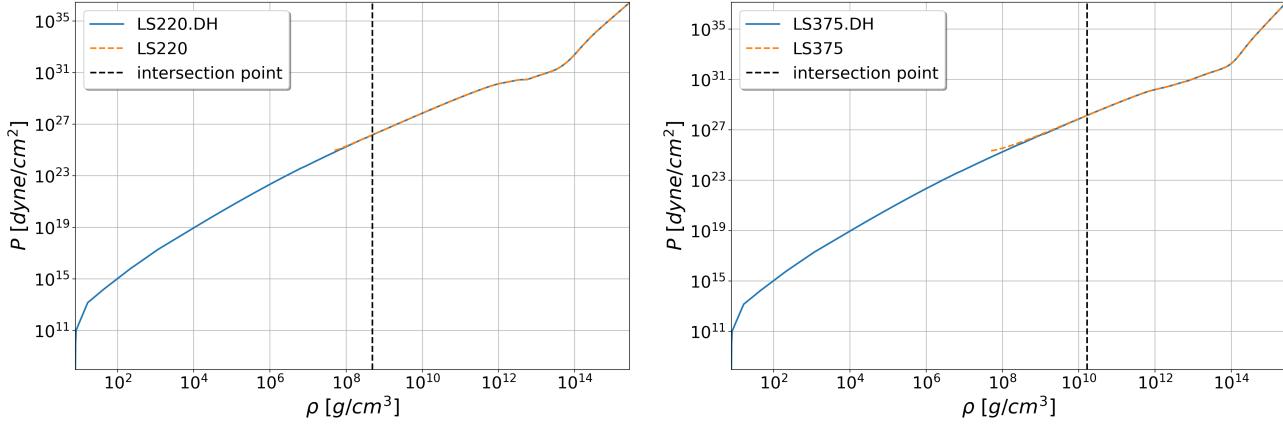


Figure 6.9: Matching of the tabulated EOS LS220, LS375 with the low density EOS DH by their intersection point. The format is the same with Fig. 6.8

It is worth mentioning that the contribution of the low density regime of the EOS table to the bulk properties such as the gravitational mass and circumferential equatorial radius is insignificant [?], and therefore the matching method can be chosen arbitrarily. This is also verified from the gravitational mass to radius relation shown in Fig. 6.10. In the latter, we see that there is a slight difference between the original EOS and the extended EOS tables for the models with low central energy density, hence, low mass models. To construct the sequences, the original RNS code (the same format is followed by Hydro_RNSID) was employed for the extended EOS tables, while the latest version of RNS (not implemented in ET) was used for the original EOS.

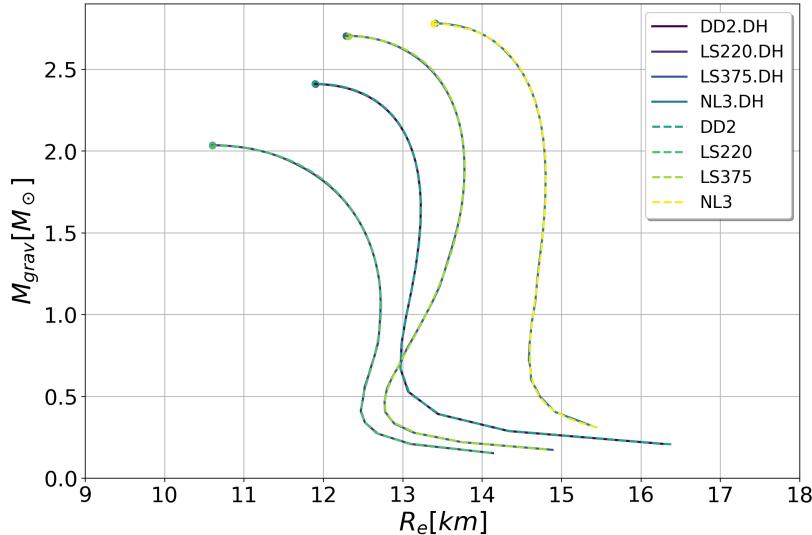


Figure 6.10: Gravitational mass to circumferential radius relation for the original EOS tables (NL3, DD2, LS220, LS375) and the extended EOS tables (NL3.DH, DD2.DH, LS220.DH, LS375.DH).

Subsequently, it is imperative to verify if the extended EOS tables are thermodynamic consistent, meaning the first law of thermodynamics for barotropic flow (2.7) is satisfied to a certain

degree. To accomplish this, we computed the corresponding adiabatic indices Γ_1 for each EOS table using the two separate, but in principle equivalent, definitions (2.22), and then obtained their absolute difference $\Delta\Gamma_1$ as illustrated in Fig. 6.11. This serves as a measure of violation of the first law of thermodynamics, whereas in principle the absolute difference should vanish. Moreover, the absolute difference $\Delta\Gamma_1$ depends on the finite differencing scheme, therefore increasing the number of points in the interpolated tables decreases the corresponding violation.

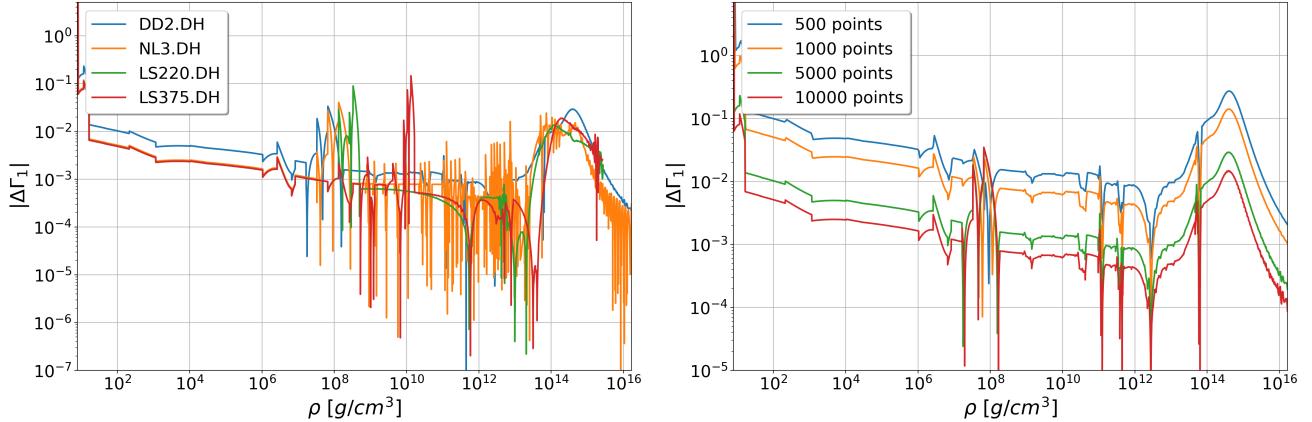


Figure 6.11: (left) Violation of the first law of thermodynamics expressed with the absolute difference of the two definitions of Γ_1 calculated for each value of rest mass density. (right) Demonstration of the decrease in the violation of the first thermodynamic law due to errors originating from the finite differencing scheme.

The aforementioned prescription deals with the proper preparation of EOS tables, which do not cover sufficiently the low-density regime. Nevertheless, there are available EOS tables, an example being the EOS tables which can be downloaded along the original RNS version from [?]. For our first model using a tabulated EOS, we opted for the EOS formulated by Bethe and Johnson 1974 [], abbreviated by eos C. We considered a model with intermediate gravitational mass $M = 1.6204 M_\odot$ and circumferential radius $R_e = 11.4392$ km as shown in Tab. 6.7 in an attempt to reproduce the associated frequencies of radial oscillations of [?]. The thermal effects, which in the case of single stars in equilibrium should vanish, are accounted for by the gamma-law pressure component with $\Gamma_{th} = 2$.

Comparison model using eosC					
ρ_c [g/cm³] (CU)	ε/c^2 [g/cm³] (CU)	M [M_\odot]	M_0 [M_\odot]	R_e [km] (CU)	r
$1.211 \cdot 10^{15} (0.0019604)$	$1.413 \cdot 10^{14} (0.0022876)$	1.6204	1.8182	11.4392 (7.7471)	1

Table 6.7: Parameters of our model using the cold tabulated eos C.

In contrast to the analytic expressions of the piecewise polytropic EOS, the tabulated EOS is subject to errors originating from the interpolation schemes due to the fact the the total number of points is limited. For our model, we refined the original table, having only 96 points, using a four-point Lagrange interpolation, to a table with total 1000 points (during the evolution, logarithmic linear interpolation is employed). Since our code assumes logarithmic spacing in rest mass density ρ , there are cases where the interpolation is necessary in order to achieve the compatibility. The bulk parameters of the star, in principle, should not depend on the interpolation scheme and this can be seen in Tab. 6.7, Tab. 6.8, where for the same central energy density as in [?], the obtained model agrees to the original one up to one decimal digit

(for gravitational mass). However, the authors state that the interpolation scheme can alter the corresponding mode frequencies, since the profile of speed of sound (and therefore the adiabatic index which enters into the oscillation equations) is highly sensitive to the interpolation scheme.

Model using H4 taken from literature			
ε_c/c^2 [g/cm ³]	M [M_\odot]	R_e [km]	f_0 [kHz]
$1.413 \cdot 10^{15}$	1.619	11.443	2.36

Table 6.8: Parameters of the model using eosC taken from literature [?].

Firstly, we tested the effect of the physical grid's size, namely, the maximum value $x_{\max}(y_{\max}, z_{\max})$, on the corresponding oscillation frequencies. We performed three simulations, using (finest) $dx = 0.369$ km (0.25 CU), for three different physical grid sizes $x_{\max} = (29.53, 118.12, 236.25)$ km, or $x_{\max} = (20, 80, 160)$ CU. We employed two, five, and six refinement levels for each case respectively, with the two inner ones being located at $r = (14.76, 22.15)$ km, or $r = (10, 15)$ CU. Up to roughly 6 ms, we observe only slight differences in the time evolution curves of the central rest mass density as shown in Fig. 6.12, whereas after 6 ms the simulation with the smallest physical size crushed since the star collapsed. Due to the large amplitude of the oscillations, small portions of matter are transferred to the outer artificial atmosphere, which are usually negligible, especially for high resolution simulations. However, in case of small physical grid, the multiple pulsations lead to the excitation of the whole computational grid and at some point matter from the boundary is pushed towards the inner direction, and therefore the star collapses.

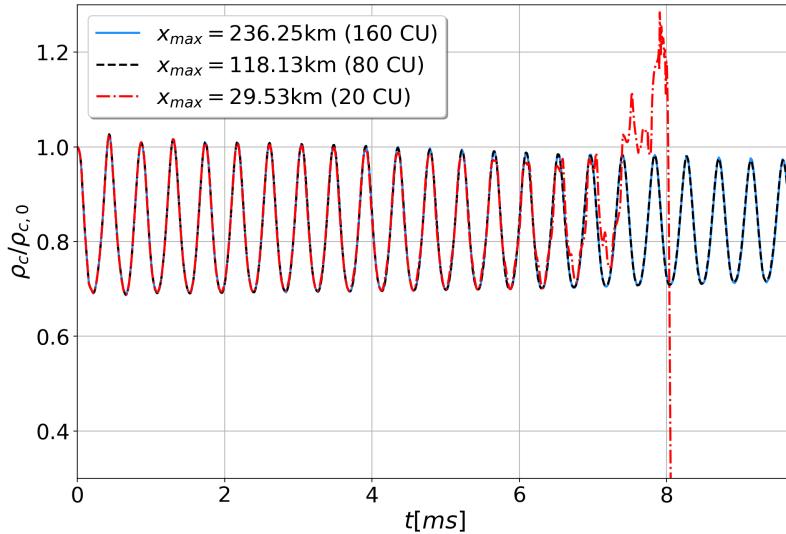


Figure 6.12: Time-evolution of the central rest mass density for three different physical grid sizes.

To compare the associated frequencies on equal footing, we performed the FFT on data up to $t \simeq 8.13$ ms, from which we report the fundamental frequencies in Tab. 6.9. The obtained frequencies $f^{236.25}$, $f^{118.13}$, $f^{29.53}$ agree up to one decimal point, while the relative errors, with respect to $f^{236.25}$, are 1.211% and 1.287% respectively. Furthermore, this agreement also holds for value taken from [?] $f^{Kokkotas} = 2.36$ kHz, with relative errors 2.238%, 1.055%, 0.980%. Thus, the physical size of the system does not alter significantly the mode frequencies, however,

a rather small grid is not appropriate for long term evolution which is required in order to achieve sufficient resolution in frequency domain.

$f^{236.25}$ [kHz]	$f^{118.13}$ [kHz]	$f^{29.53}$ [kHz]	$f^{Kokkotas}$ [kHz]
2.30	2.33	2.33	2.36
relative to $f^{236.25}$ errors %			
-	1.211	1.287	-
relative to $f^{Kokkotas}$ errors %			
2.238	1.055	0.980	-

Table 6.9: Fundamental frequency modes using the tabulated eos C for three different physical sizes.

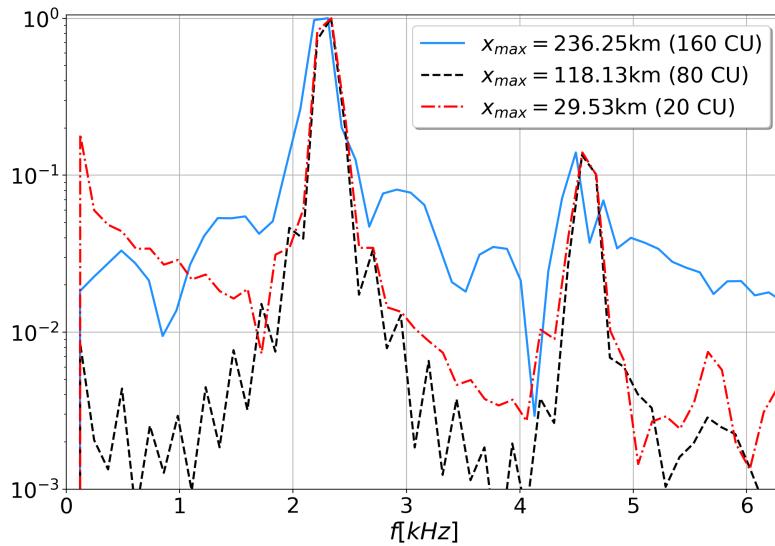


Figure 6.13: FFT for the three simulations with three different physical sizes.

Interestingly, as we see in Fig. 6.13, the simulation with the largest physical domain ($x_{max} = 236.25$ km) displays small bumps in the spectra which represent the superposition of the oscillation frequency modes (since we used a proper window function for the FFT, these bumps cannot be completely unphysical). The latter pulsations could be triggered from the numerical errors associated with outer refinement level where the grid is rather coarse. To support this idea, we further evolved the simulation with $x_{max} = 118.13$ km up to 15.456 ms, and subsequently applied FFT using data with three maximum values $t^{max} = (10.000, 12.31, 15.45)$ ms. As it is demonstrated in Fig. 6.14, we see that as the time variable increases (and therefore the numerical error adds up) the previous mentioned frequency bumps are shown in this simulation too, however, in this case we applied no window function to the original data.

Till now, we have achieved the agreement between our calculations and the ones from literature for the fundamental frequency mode, however we failed to reproduce the secondary frequency modes (we roughly position our second peak at 4.6 kHz, while the authors of [?] predict a second peak at 6.33 kHz). The latter discrepancy can be due to several reasons, the first one being our numerical grid. As we have seen, the effect of the actual physical size of the grid does not influence the corresponding frequencies, at least for short term simulations, however, the way we positioned the refinement levels may have severe implications. For example, we considered a refinement level at $x = 14.765$ km (10 CU), meaning that the resolution at $r \in (R_e, 2R_e)$ varies, and since the radial pulsations originate from the surface of the star, it may play a important

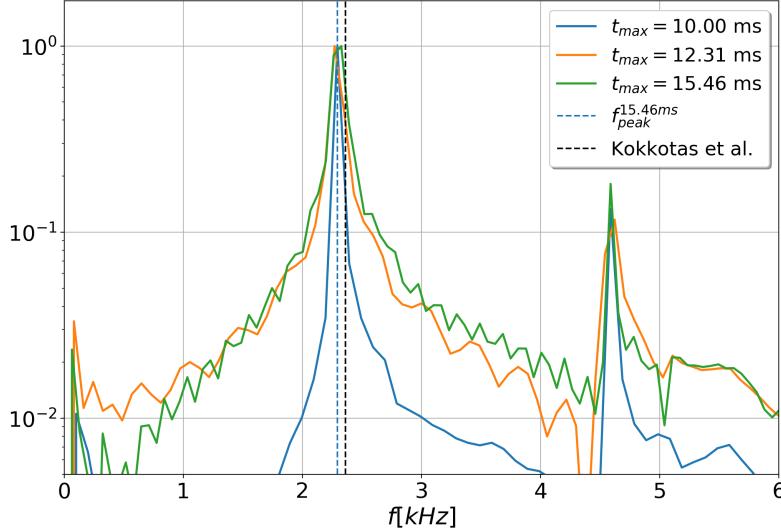


Figure 6.14: FFT for the simulation with $R^{\max} = 118.13$ km, for three different values of t^{\max} .

role. Secondly, in our work we also included the thermal pressure component, with $\Gamma_{th} = 2$ (which may be an overestimation), and therefore by definition we do not have the exact same EOS as the authors of [?].

Moreover, whenever a fluid element obtains lower density than the constant threshold defined by GRHydro, its hydrodynamic variables are set to the atmospheric ones. The atmospheric pressure and internal specific energy are described by a polytropic EOS (this treatment is included in numerous routines, hence, in the future careful modifications have to be applied), and since the oscillations come from the surface of the star substantial changes can occur. In an attempt to fix this inconsistency, we provided the specific internal energy associated with our cold tabulated EOS, and we encountered the following problem. Some authors have considered different values for the baryonic mass (we obtain the rest mass density through multiplication baryionic number density with the bayrioc mass), and thus, the limit ($\rho \rightarrow 0$) $\varepsilon/\rho \rightarrow 1$ is not valid. This leads to negative values of ϵ which can be problematic. We will further investigate this section of the code in future works.

ε_c/c^2 [g/cm ³]	f_0 [kHz] (simulation)	f_0 [kHz] (Kokkotas et al.)	relative error %
$1.995 \cdot 10^{15}$	2.01	1.80	11.85
$1.778 \cdot 10^{15}$	2.13	2.02	5.43
$1.413 \cdot 10^{15}$	2.32	2.36	1.49
$1.122 \cdot 10^{15}$	2.42	2.56	5.48
$1.100 \cdot 10^{15}$	2.41	2.59	7.04

Table 6.10: Comparison of the fundamental frequency oscillation modes

Subsequently, we considered a sequence of models using the tabulated eos C, in order reproduce the fundamental frequencies in a wider range. We employed the aforementioned type of computational grid ($dx = 0.369$ km, and several refinement levels) where we either considered physical size of $x_{\max} = 236.25$ km or $x_{\max} = 118.13$ km. Similarly, we used the refined eos C table with 1000 points, and a thermal component with $\Gamma_{th} = 2$. We constructed five models (four new ones plus the previous one) with the same central energy density ε_c/c^2 as the ones of [?], and evolved them up to roughly 10 ms. From Fig. 6.15, where the time-evolution of the

central rest mass density for each model is shown, we observe that the augmentation in the central energy results to larger amplitudes in the corresponding oscillations.

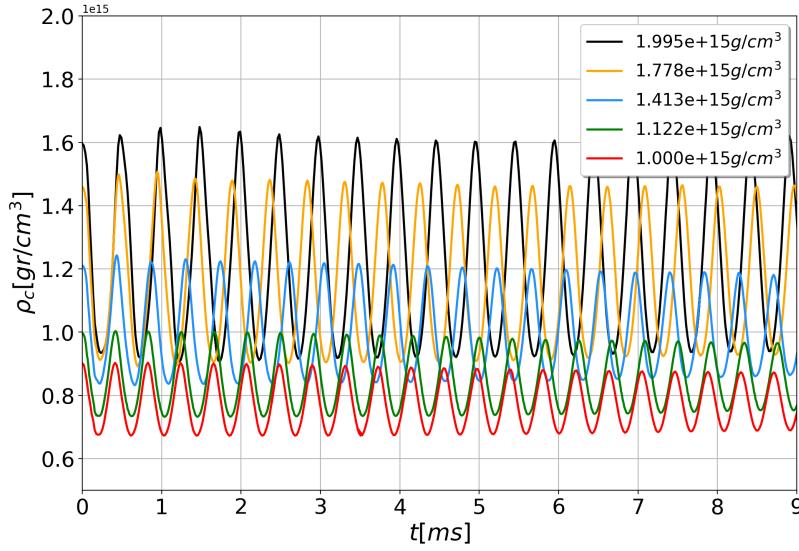


Figure 6.15: Time evolution of the central rest mass density for a sequence of models using eos C

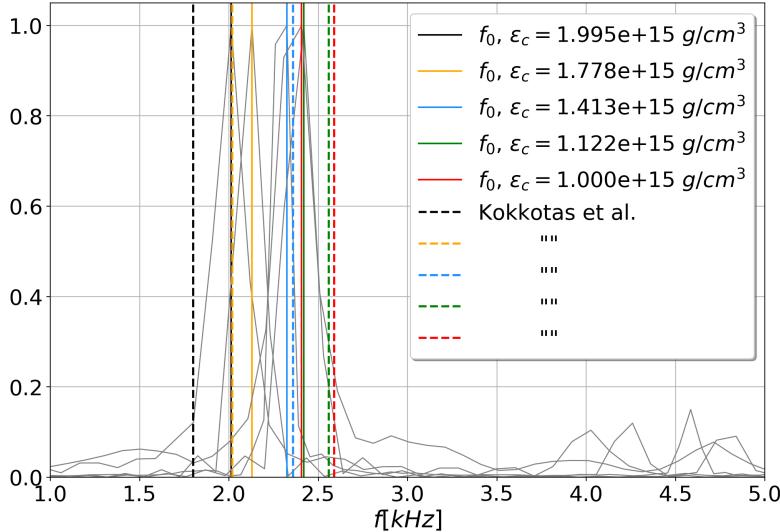


Figure 6.16: FFT for the sequence using eos C. Gray lines represent the FFT functions, vertical lines represent the modes frequencies computed through the simulations, dashed vertical lines represent the theoretical predictions [?].

Unfortunately, we obtained only quantitative agreement for the corresponding fundamental frequencies (see Tab. 6.10, and Fig. 6.16), with the maximum relative error being 11.85%. Nevertheless, as we previously mentioned, several parameters influence the associated frequencies, and for the purposes of this work the quantitative agreement suffices since we test whether the implementation is correct or not.

During the initial part of this work, in several test simulations using the aforementioned tabulated EOSs, with numerous grid configurations, we encountered the failure of the recovery scheme, especially at the region surrounding the star. The latter region is completely artificial

and originates from the fact that low portions of matter are transferred to the constant density atmosphere (this phenomenon is amplified by the radial pulsations of the star). We tackled this failure (at least for single star models) with our implementation described in Ch. ???. It slightly modifies the conservative variables (when it is necessary) to ensure a solution in the physical range, so the code continues. As long this occurs at the artificial region, the physics which we investigate should not be influenced very much. During the evolution, we use both of the recovery schemes (to the limit where the magnetic fields vanish), where our implementation (Brent's method) carries out the recovery when the first method (modified version of Einstein Toolkit's) fails.

Having already tested the implementation for the previously mentioned models, we then attempted to the same at extreme conditions. We considered a rapidly rotating star, generated with the cold, tabulated, and extended DD2.DH EOS. The model's gravitational mass $M = 2.5435 M_{\odot}$ is larger than the maximum mass of the non-rotating star configuration (see Tab. 6.11). Due to the rapid rotation of the star, and the presence of the artificial atmosphere, a low density torus is formed around the star, hence, making it a good candidate for testing the recovery scheme.

Rapidly differentially rotating model using DD2					
ρ_c [g/cm ³] (CU)	ε/c^2 [g/cm ³] (CU)	M [M_{\odot}]	M_0 [M_{\odot}]	R_e [km] (CU)	r
$1.5374 \cdot 10^{15} (0.0024891)$	$2.2146 \cdot 10^{15} (0.0035854)$	2.5435	3.0045	11.8080 (7.9969)	0.75

Table 6.11: Parameters of the rapidly differentially rotating star ($A=1.0$) model using the tabulated DD2.

We performed two simulations, employing a large computational grid of $x_{\max} = 354.38$ km, with grid spacings $dx = (0.538, 1.1074)$ km, or $dx = (0.365, 0.75)$ CU, and evolved them without any problem, up to roughly $t \simeq 16$ ms and $t \simeq 20$ ms respectively. We report the corresponding fundamental (quasi) radial frequency at $f_0 = (1.536, 1.613)$ kHz for $dx = (0.538, 1.1074)$ km respectively. Using these data, we can (roughly) extrapolate to $dx \rightarrow 0$ using a linear fit to finally obtain $f_0^{dx \rightarrow 0} = 1.46$ kHz, as shown in Fig. 6.17.

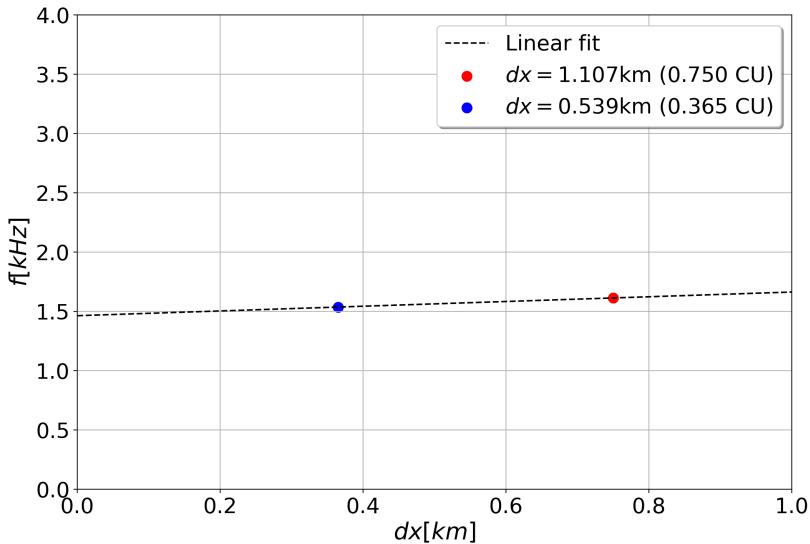


Figure 6.17: Fundamental frequency modes for two different grid spacing, and extrapolation to $dx \rightarrow 0$ using a linear fit.

6.2 Binary neutron star mergers

Since we have tested our implementation for single star models, it naturally follows to question whether or not we are able to simulate binary neutron star coalescences using tabulated EOS. Unfortunately, we cannot yet guarantee that our code works for every (available and well-constructed) tabulated EOS, since we encountered situations where the code crushed (especially for large gravitational mass configurations) during the recovery of the primitive variables from the conservative ones on the inspiral phase. Nevertheless, in the following we present the main features of a toy BNS model using the (extended) LS220 6.8.

We considered a binary system of two equal mass companions ($q = m_2/m_1 = 1$) with gravitational mass $M = 1.2 M_\odot$, and initial distance between the centres of each star $d = 40$ km. The physical grid is rather large, $x_{\max} = 1063.139$ km (720 CU), since as we mentioned in Sec. ?? the gravitational signal is extracted under the assumption of an asymptotic frame. We included six refinement levels, and imposed reflection symmetry at the equatorial plane (XY). Whilst the computational grid being rather coarse, with finest grid spacing $dx = 0.922$ km (0.625 CU), it certainly fulfils its purpose in demonstrating the basic features of the BNS simulation.

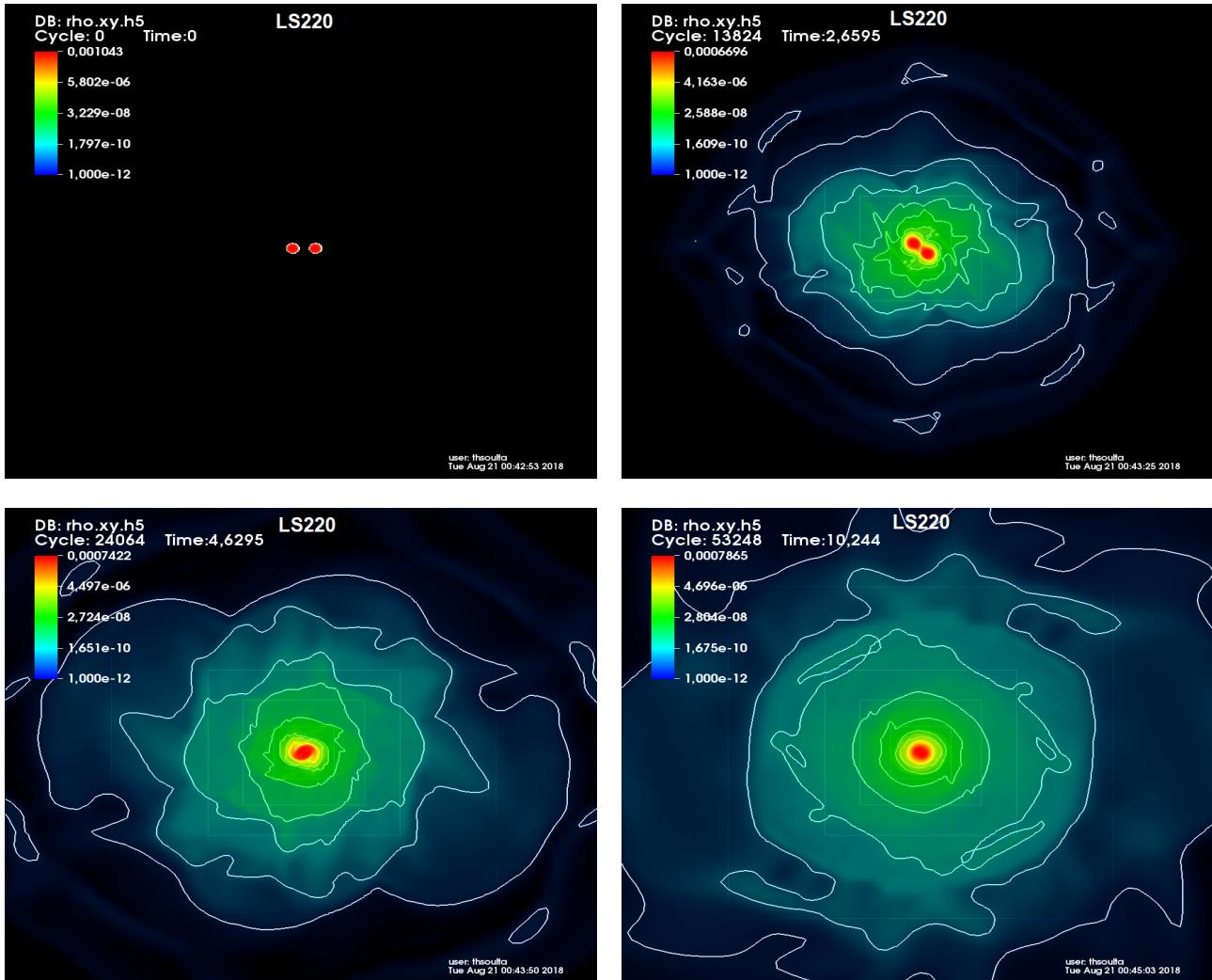


Figure 6.18: Rest mass density (CU) at the XY-plane. The colouring is in logarithmic scale, and the white curves represent contour lines. To facilitate the visualization, only a region at the vicinity of the companions is showed.

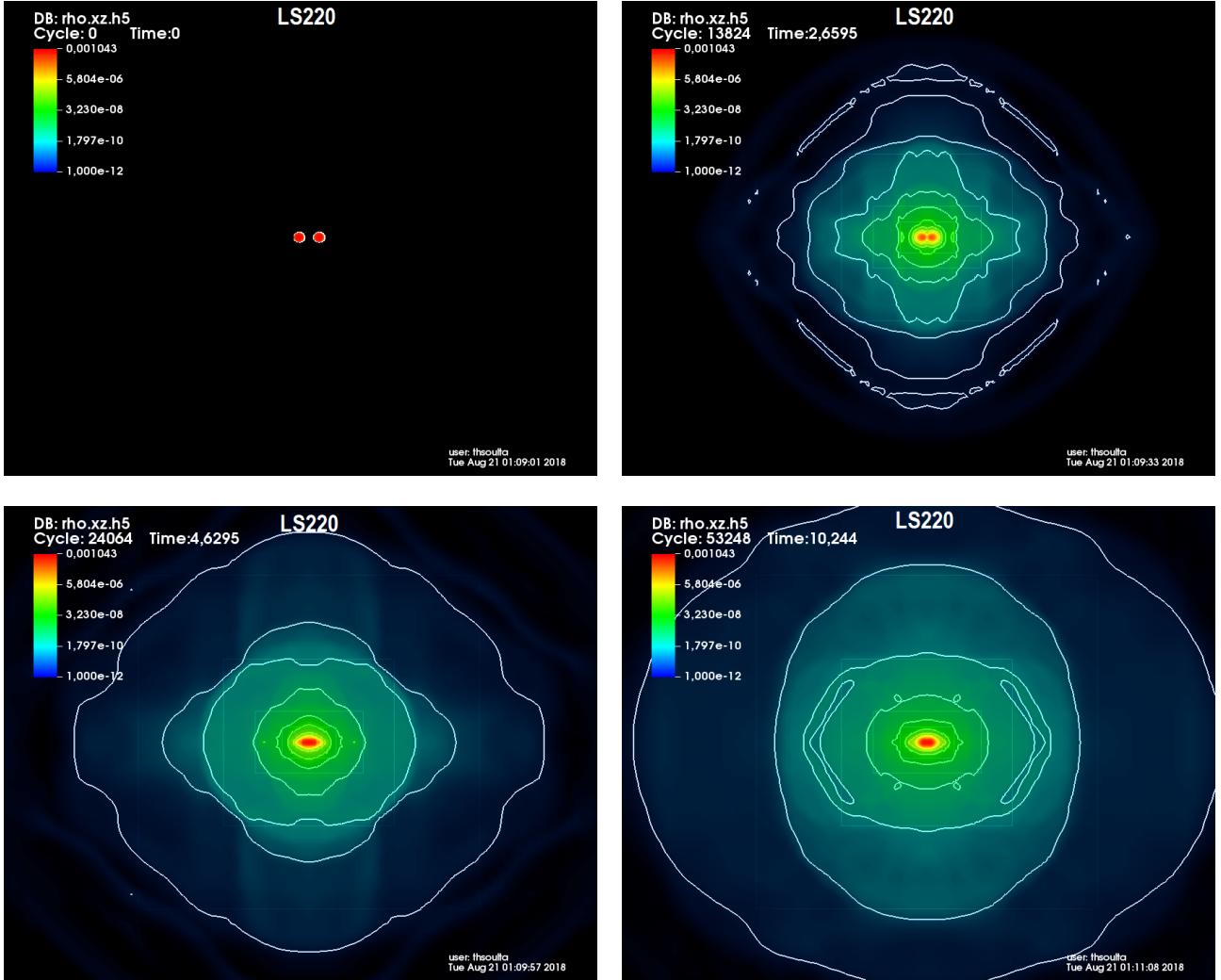


Figure 6.19: Rest mass density (CU) at the XZ-plane. The colouring is in logarithmic scale, and the white curves represent contour lines. The images were produced using reflection symmetry on the original data. To facilitate the visualization, only a region at the vicinity of the companions is showed.

The simulation results to a differentially rotating object (DRO) supported against the gravitational collapse by the centrifugal forces. Fig. 6.18, and Fig. 6.19 illustrate the rest mass density profiles (measured by the comoving with the fluid observer) at XY-plane (equatorial plane), and XZ-plane at different times (initial positions, in spiral, merger, remnant), where the colouring is carried out in logarithmic scale. During the inspiral at roughly $t \simeq 2.65$ ms, even though the merger has not occurred yet we observe the formation of a low density torus surrounding the stars. The latter is again an artefact of the imposed constant density atmosphere, spread throughout the computational grid. It does not influence the dynamical evolution of the system, and it is not to be confused with real low density torus which is later formed around the remnant star at $t \simeq 10.24$ ms. Nevertheless, one should account for the artificial atmosphere's impact in order to study the ejecta of these simulations.

Subsequently, we extract the gravitational strain component h_{22} at radius $R = 1033.607$ km (700 CU) using the prescription described in Sec. ???. In Fig. 6.20 we present the associated polarization states h_+ and h_\times with respect to the retarded time t_{ret} , the corresponding FFT denoted by $|\tilde{h}_+|$ and $|\tilde{h}_\times|$ respectively (computed using the ordinary time t variable), and the spectral density $|\tilde{h}|f^{1/2}$. Its gross behaviour is similar to the typical gravitational strain of a DRO. Moreover, we identify the peak in the spectra at $f_{peak} \simeq 2.13$ kHz as the dominant gravitational wave frequency peak in the postmerger phase, which is thoroughly investigated

in [?]. In the latter, the authors computed the aforementioned frequency peak (for the same mass configuration, and using the complete description of the thermal effects with the full 3-dimensional LS220 EOS table) at $f_{peak}^{literature} = 2.55$ kHz. Given the fact that our simulation's grid is rather coarse, and that we approximate the thermal effects with a gamma-law EOS, the relative error 16% is acceptable.

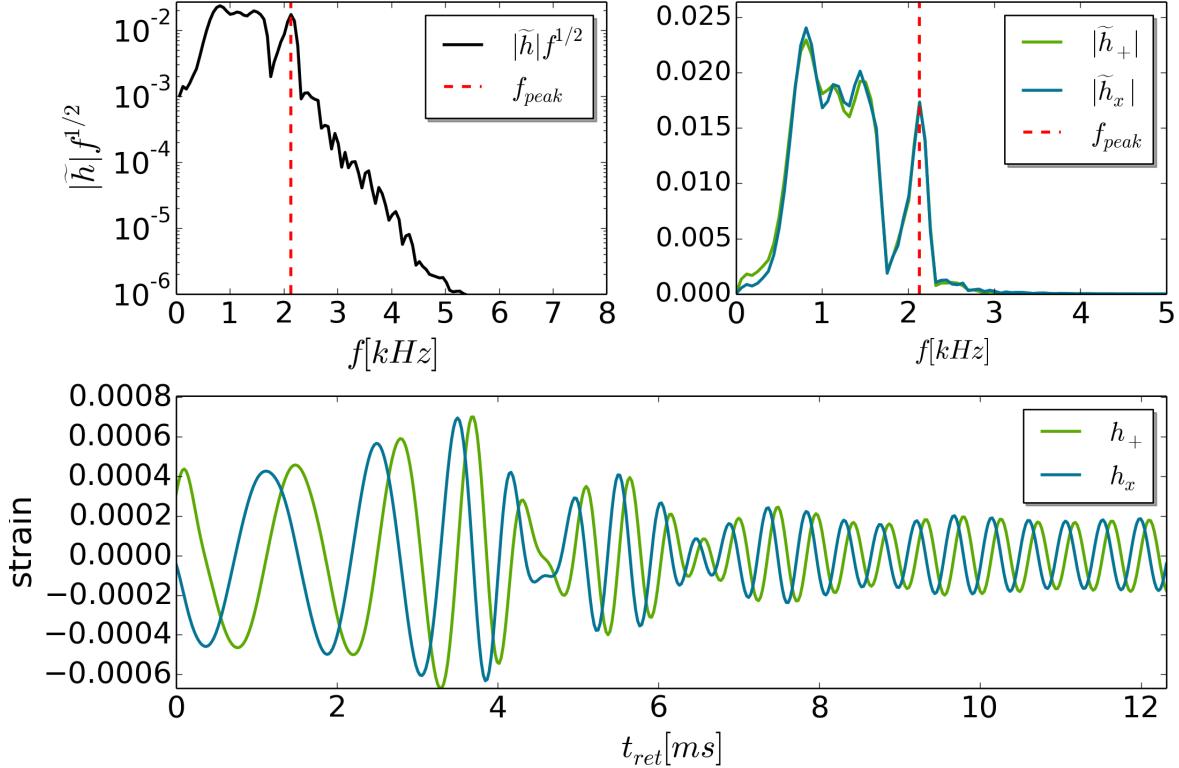


Figure 6.20: Gravitational wave strain extracted at $R = 1033.607$ km (700 CU): (top-left) Spectral density $|\tilde{h}|f^{1/2}$, (top-right) FFT of the polarization states denoted by \tilde{h}_+ and \tilde{h}_x , (bottom) gravitational strain h_+ and h_x with respect to retarded time t_{ret} .

6.3 Simulations for GW170817

In 2017, the LIGO Collaboration announced the detection (GW170817) of the first gravitational wave source involving matter, and more specifically a BNS system. The significance of this detection is undeniable especially for stellar astrophysics and nucleosynthesis, while simultaneously provides the opportunity to infer weakly-constrained properties of neutron stars. The gravitational signal either from finite-size effects during the inspiral phase, or the from the associated oscillation modes of the corresponding postmerger remnant can provide such information.

The BNS mergers can result either in the immediate black hole formation on a dynamical scale (prompt collapse) or the formation of an (temporally) stable differentially rotating neutron star merger remnant (delayed collapse or no collapse). The prompt collapse occurs when the total gravitational mass of the binary system exceed a threshold mass M_{thres} (sensitively depending on the EOS), while the delayed/no collapse happens when the total gravitational mass is lower than M_{thres} . GW170817 is categorized (the evidence suggest so) as a delayed/no collapse scenario, and therefore provides a lower bound of the threshold mass $M_{thres} > M_{tot}^{GW170817} = 2.74_{-0.01}^{+0.04} M_\odot$.

In this section, we present the results of four high-resolution simulations using the seven-segment polytropic EOS MPA1, and APR4, in an attempt to simulate the GW170817 event. For each EOS, we examine two configurations, symmetric BNS system ($q = 1$) and asymmetric BNS ($q = 0.73$), where the mass ratios were chosen in order to be compatible with the limits suggested by Table II of [1]. The final masses of our models were fixed by choosing a chirp mass $M_{chirp} = 1.186$ equal to that of GW170817 (again Table II of [1]). More specifically, one can express the model gravitational masses in terms q and M_{chirp} as

$$m_1 = M_{chirp} q^{-3/5} (q + 1)^{1/5}, \quad (6.1)$$

$$m_2 = M_{chirp} q^{+2/5} (q + 1)^{1/5}. \quad (6.2)$$

Using these relations, we obtain the theoretical mass configurations as $m_1 = m_2 = 1.362 M_\odot$ resulting $M_{tot} = 2.724 M_\odot$, and $m_1 = 1.598 M_\odot$, $m_2 = 1.668 M_\odot$ resulting $M_{tot} = 2.765 M_\odot$ for $q = 1$ and $q = 0.73$ respectively. The parameters of our considered models are given in Tab. 6.12, while the associated parameters (high density region) of MPA1, and APR4, along with the gravitational mass M to circumferential equatorial radius R_e relation, are shown in Fig. 6.13, Fig. 6.21 respectively. One can see that our models's gravitational masses agree with the theoretical values up to two decimal digits, and this is partially due to the fact we used the LORENE library in order to generate the initial data, and since the code converges to specific baryonic masses instead of gravitational masses we applied an iterative Newton-Raphson type scheme to obtain the proper baryonic mass configurations which correspond to the gravitational masses of our choice. Moreover, in order to achieve the ratio $q = 0.73$, we included a patch for LORENE which can be found at [?].

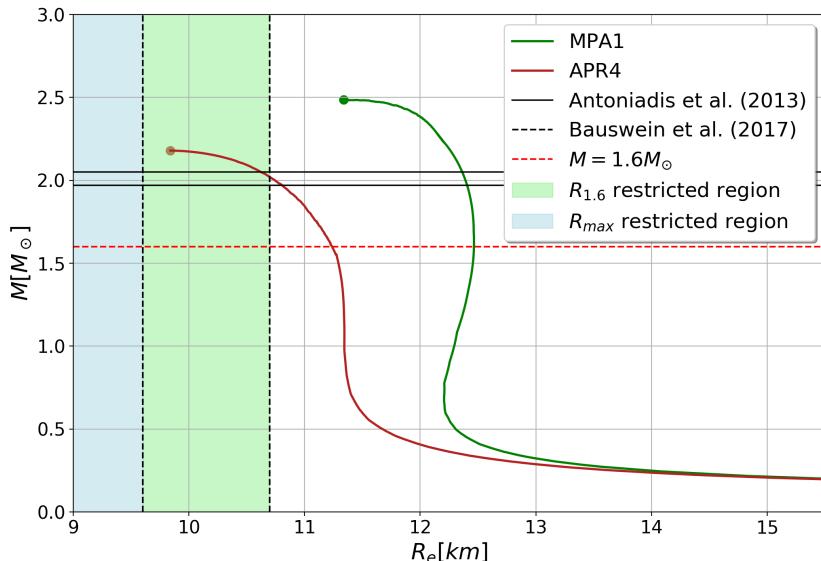


Figure 6.21: Gravitational mass M to circumferential equatorial radius R_e relation for the piecewise polytropic EOS MPA1, APR4. These relations were calculated like those in Fig. 6.2.

	$q = 1$					$q = 0.73$				
	M_1	M_2	M_1^{bar}	M_2^{bar}	d [km]	M_1	M_2	M_1^{bar}	M_2^{bar}	d [km]
MPA1	1.3623	1.3623	1.55548	1.55548	50	1.59853	1.1668	1.878	1.326	40
APR4	1.3623	1.3623	1.59308	1.59308	40	1.59871	1.16754	1.90518	1.34257	40

Table 6.12: Parameters of the BNS models using the polytropic EOS MPA1, APR4. The masses are given in M_\odot , while d denotes the distance between the center of the stars at the initial configuration.

	ρ_4	ρ_5	ρ_6
MPA1	$1.678 \cdot 10^{14}$ ($2.7172 \cdot 10^{-4}$)	$5.012 \cdot 10^{14}$ ($8.115 \cdot 10^{-4}$)	$1.000 \cdot 10^{15}$ ($1.619 \cdot 10^{-3}$)
APR4	$1.512 \cdot 10^{14}$ ($2.448 \cdot 10^{-4}$)	$5.012 \cdot 10^{14}$ ($8.115 \cdot 10^{-4}$)	$1.000 \cdot 10^{15}$ ($1.619 \cdot 10^{-3}$)
	Γ_4	Γ_5	Γ_6
MPA1	3.446	3.572	2.887
APR4	2.83	3.445	3.348

Table 6.13: Parameters of the piecewise polytropic EOS for MPA1, APR4. The units are in the same format as Tab. 6.1.

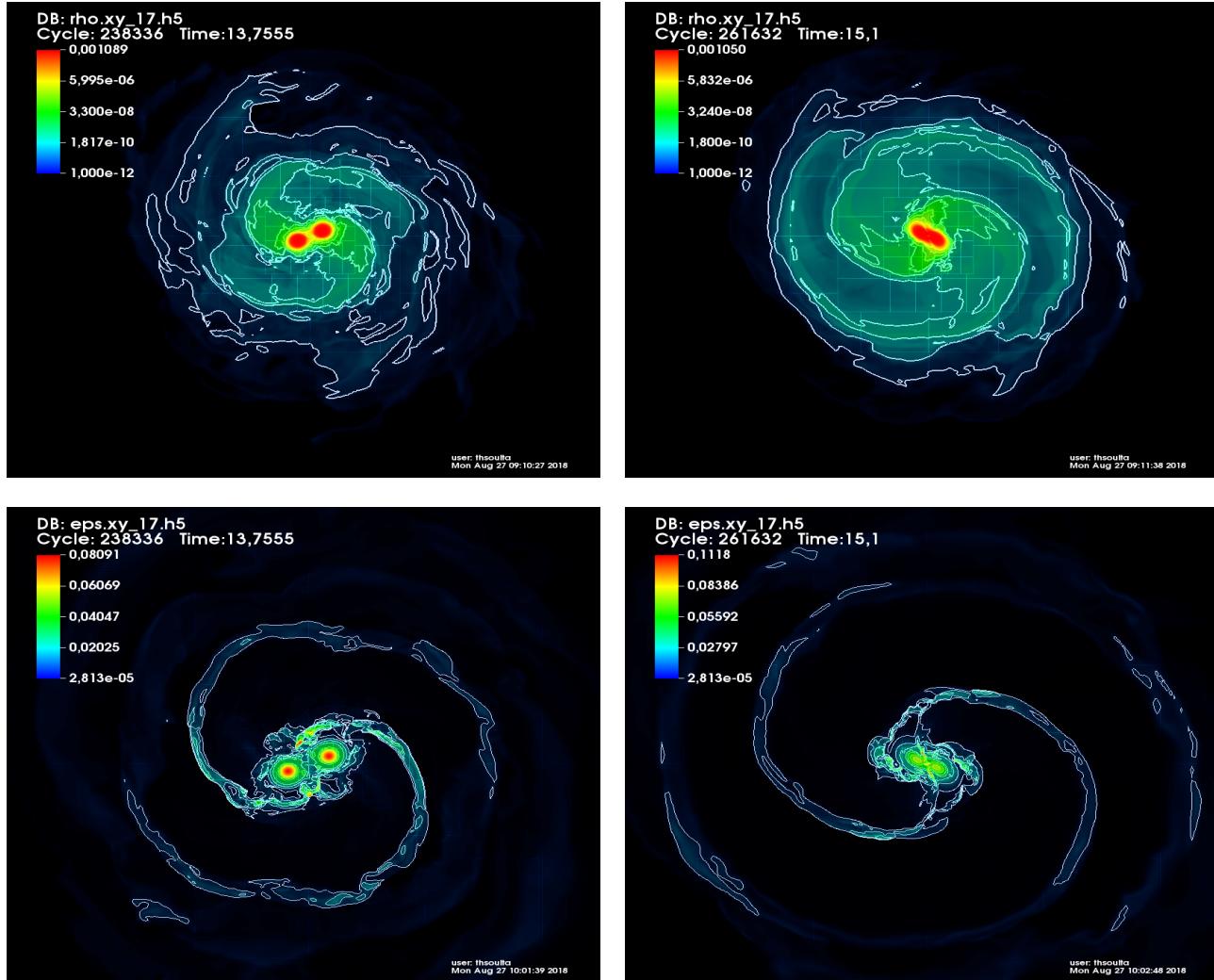


Figure 6.22: Rest mass density (CU) and specific internal energy at the XY-plane for the MPA1, $q = 1$ model for the inspiral and merging phase. The colouring is in logarithmic scale, and the white curves represent contour lines. To facilitate the visualization, only a region at the vicinity of the companions is showed.

As for the numerical setup, we considered a large computational grid with $x_{\max} = 1063.139$ km (720 CU), and six refinement levels corresponding to finest $dx = 0.2768$ km (0.1875 CU) at radii $R = 22.148$ km (15 CU), $R = 11.074$ km (7.5 CU) at x(y) and z direction respectively. In addition, we included one more refinement level, corresponding to a radius $R = 5.537$ km (3.75 CU), which is triggered when the the minimum of the lapse function obtains a value below a specific threshold ($\alpha_{\min} \leq 0.3$). This accounts for the potential formation of a black hole and prevents, to some extend, the code from crashing. As with the previous cases, the simulations are carried out in 3-dimensions and reflection symmetry is applied along the equatorial plane. These configurations (even with reflection symmetry) require substantial computational resources since a typical run corresponds to roughly 200 Gb RAM.

Let us first analyse the results of the MPA1 model with $q = 1$, starting with the rest mass density profile, measured by the comoving observer, at the equatorial plane (XY) as shown in Fig. 6.22, Fig. 6.23. As it was expected, the effect of the artificial atmosphere is certainly decreased in contrast to what we encountered in the previous section (Fig. 6.18). Since we opted for (at least for the model with MPA1, $q = 1$) initial distance of the star centres equal to $d = 50$ km, the inspiral phase is rather long. In Fig. ??, Fig. ?? the specific internal energy profiles are also shown, and it is apparent that energy is emitted as gravitational waves during the different stages of the merger (by comparison to the rest mass density). We continued this run for up to $t \simeq 32.00$ ms, reporting a no collapse scenario (delayed collapse).

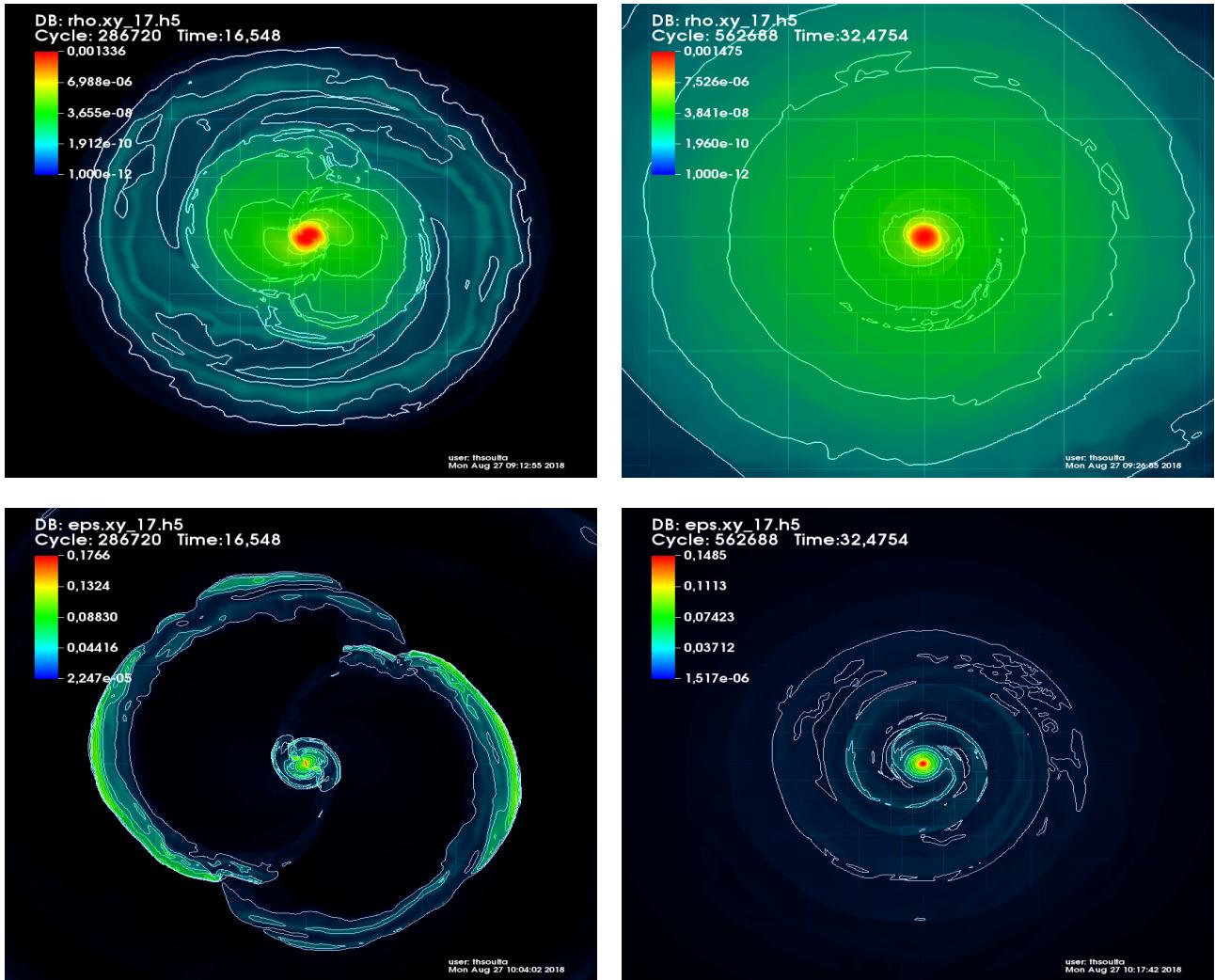


Figure 6.23: As in Fig. 6.22 but for postmerger phase.

Subsequently, we focus our analysis on the dominant h_{22} component of the gravitational signal extracted at radius $R = 1033.607$ km (700 CU) using the prescription described in Sec. ???. Similarly to Fig. 6.20, in Fig. 6.24 we present the two polarization states h_+ , h_x , of the gravitational strain with respect to the retarded time, in addition to the corresponding FFT curves namely, $|\tilde{h}_+|$, $|\tilde{h}_x|$, and $|\tilde{h}|f^{1/2}$. Thus, we identify the dominant gravitational wave frequency $f_{peak} = 2.84$ kHz of the postmerger phase, which agrees with the one reported in [?] (Table II) at $f_{literature} = 2.80$ kHz for a slightly different configuration ($M_1 = M_2 = 1.35 M_\odot$). At this point, it is important to clarify that in Fig. 6.24 we used the complete data of the BNS merger, and the term of "dominant" gravitational wave frequency f_{peak} corresponds to the peak associated with the postmerger data (which we also compute).

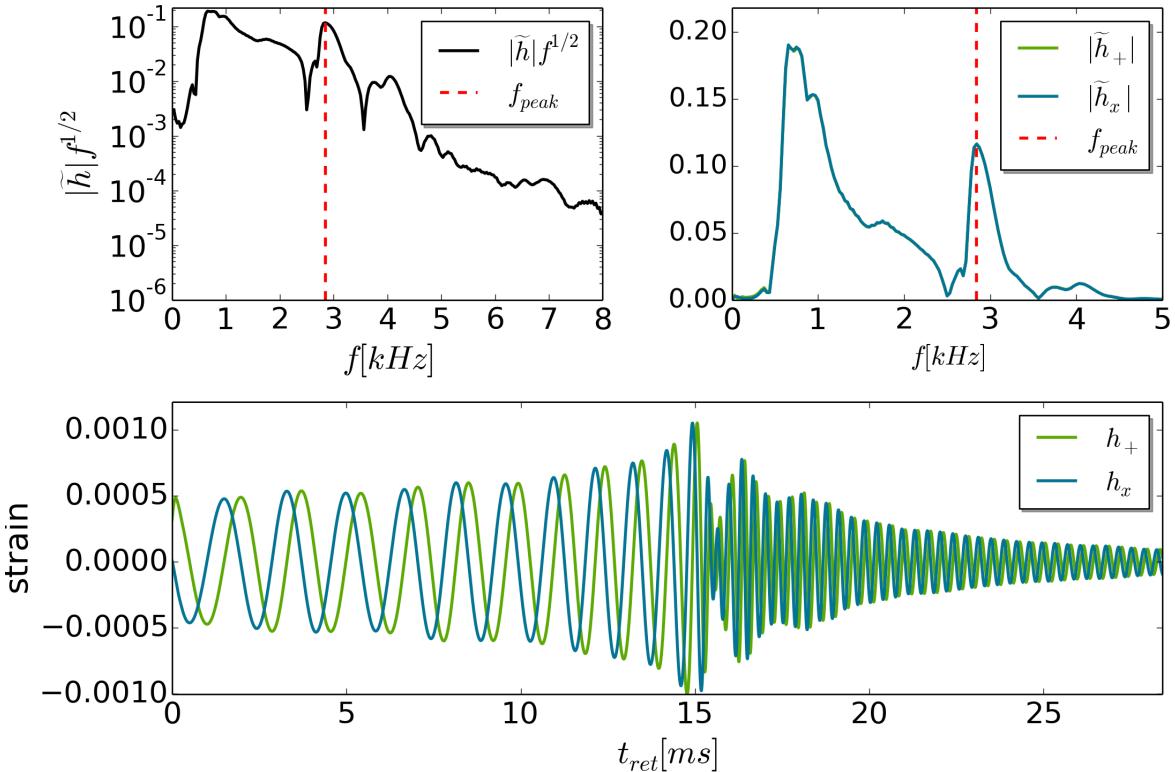


Figure 6.24: Gravitational wave strain extracted at $R = 1033.607$ km (700 CU) for the MPA1, $q = 1$ model: (top-left) Spectral density $|\tilde{h}|f^{1/2}$, (top-right) FFT of the polarization states denoted by \tilde{h}_+ and \tilde{h}_x , (bottom) gravitational strain h_+ and h_x with respect to retarded time t_{ret} .

So far, we purposely omitted the discussion of the window function applied to the gravitational signal for FFT. Windowing our data is significant since it facilitates the identification of the corresponding frequency peaks. In Fig. 6.20, Fig. 6.24, we applied the Hanning window function, and this choice is justified by Fig. (6.26) where it illustrates the comparison of four window functions, namely, Blackman, Bartlett, Hamming, Hanning, to raw (none) data. From these, Bartlett and Hanning seem to be the most appropriate choices, hence we chose the latter one.

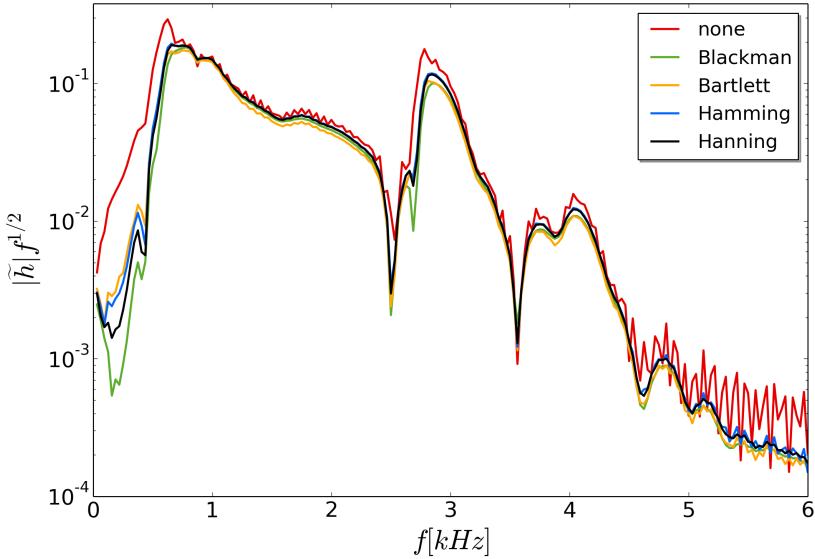


Figure 6.25: Spectral density $|\tilde{h}|f^{1/2}$ calculated using raw data, and four different window functions.

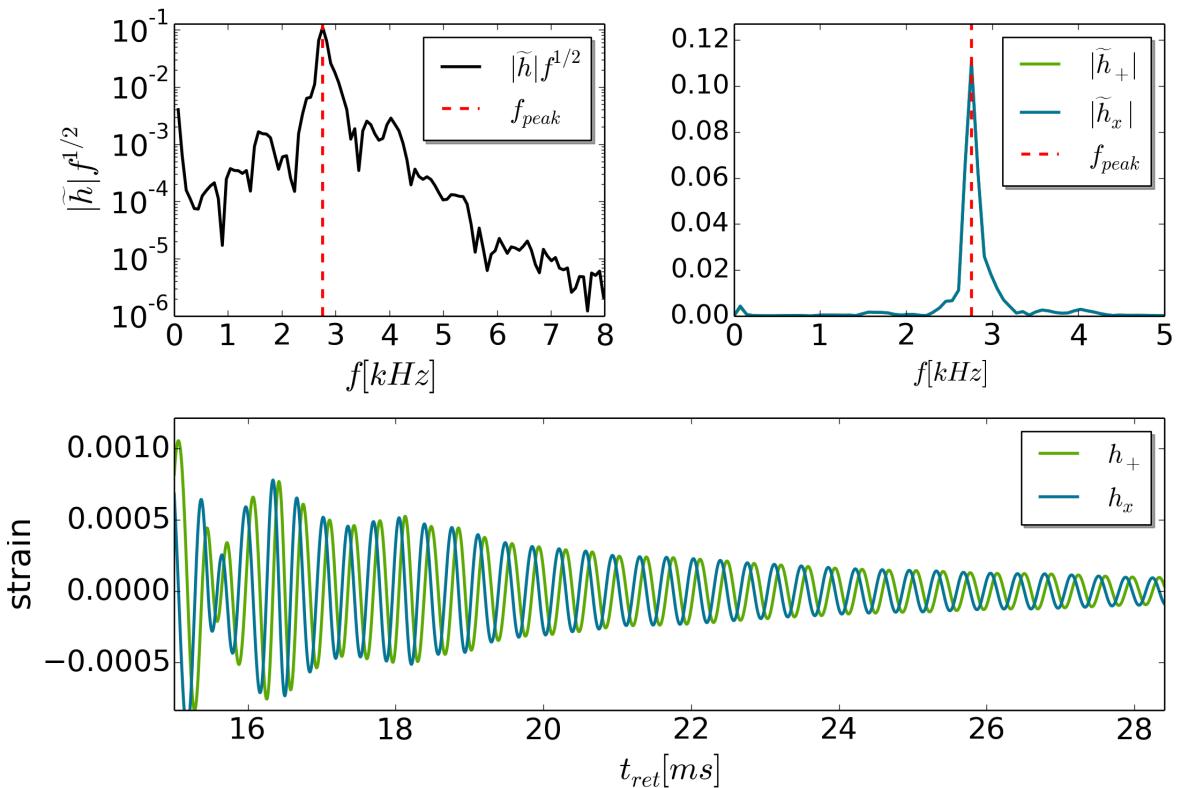


Figure 6.26: Gravitational wave strain extracted at $R = 1033.607$ km (700 CU) for the MPA1, $q = 1$ model, at the postmerger phase: (top-left) Spectral density $|\tilde{h}|f^{1/2}$, (top-right) FFT of the polarization states denoted by \tilde{h}_+ and \tilde{h}_x , (bottom) gravitational strain h_+ and h_x with respect to retarded time t_{ret} .

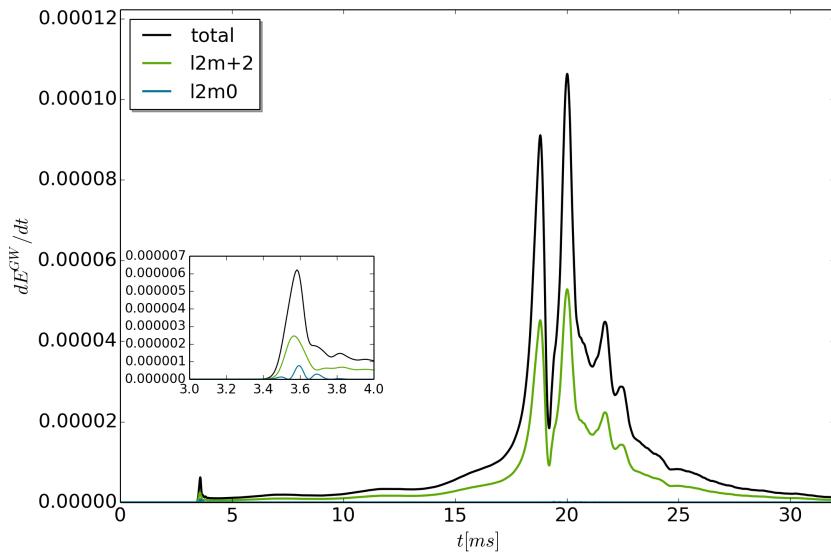


Figure 6.27: Total power of the gravitational wave emission, and the corresponding contributions of its dominant components.

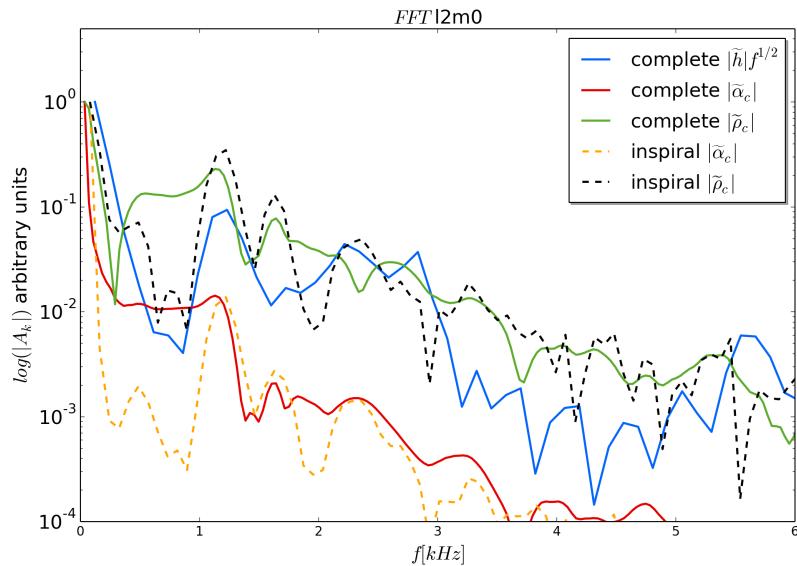


Figure 6.28: FFT spectra for the spectral density, central rest mass density, and minimum lapse function.

Chapter 7

Source code

Chapter 8

Appendices

8.1 Reproducibility - HPC infrastructure

Simulations in 3-dimensions in full general relativity, especially for highly accurate runs (high resolution simulations), require substantial computational resources. For this matter, we would like to thank the Observatory of School of Physics of AUTH, HPC infrastructure of AUTH and the people who helped us to set up the Einstein Toolkit as one of their modules, and finally the National HPC infrastructure ARIS which allowed us to realize the highly accurate simulations. In this section, we provide instructions in order to ensure the reproducibility of our results.

8.2 Lie derivatives

Bibliography

[1] <https://niksterg.github.io/gw-group/>