January 2022

An Introduction To Numerical Relativity And Simulations Of Binary Neutron Stars

Hayden Dale Drown

An Introduction to Numerical Relativity and Simulations of Binary Neutron Stars

By

Hayden Dale Drown

Bachelor of Science, University of North Dakota, 2020

A Thesis

Submitted to the Graduate Faculty

of the

University of North Dakota

for the degree of

Master of Science

Grand Forks, North Dakota

August

2022

This thesis, submitted by Hayden Drown in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

Wayne Barkhouse, Committee Chair   Date

Yen Lee Loh, Committee Member   Date

Tim Young, Committee, Member   Date

This thesis is being submitted by the appointed advisory committee as having met all of the requirements of the School of Graduate Studies at the University of North Dakota and is hereby approved.

Chris Nelson, Dean of Graduate Studies   Date

PERMISSIONS

Title	AN INTRODUCTION TO NUMERICAL RELATIVTY AND SIMULATIONS OF BINARY NEUTRON STARS
Department	Physics and Astrophysics
Degree	Master of Science

In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, is his absence, by the Chairperson of the department or the dean of the School of Graduate Studies. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my thesis.

Hayden Drown

Date

Table of Contents

Table of Figures

List of Tables

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to the members of my advisory Committee for their guidance and support during my time in the master’s program at the University of North Dakota.

To my parents and girlfriend.

Without their support this wouldn’t have been possible.

Abstract

The theory of general relativity is currently the best description of gravity. However, the equations in general relativity are highly nonlinear and only the simplest of cases can hope to be solved analytically. As a result, the field of numerical relativity was created to solve some of these issues and to model more complicated dynamical situations. This thesis sets out to give the reader a basic understanding of general relativity, numerical relativity, as well as an understanding of some of the programs that are used in numerical relativity research such as Lorene and the Einstein Toolkit and concludes with a brief set of simulations of binary neutron stars with various masses.

1. Introduction

1.1. Motivation

In 1915, Albert Einstein formulated a theory of gravity that was consistent with his special theory of relativity (Einstein 1918). This theory is currently our best understanding of gravity. To better comprehend gravity in a strong field environment, the field of numerical relativity was created (Hahn & Lindquist 1964). Numerical relativity deals with the evolution of gravitational systems and the distortions of spacetime under conditions that cannot be solved analytically. An example of such a system is the motion of massive, compact binaries, orbiting each other at extreme speeds. These bodies should emit gravitational waves as their orbits decay and objects eventually merge (Einstein 1918).

This decay was first indirectly observed in 1974 with the observation of orbits of binary neutron stars decaying and the system losing energy through these gravitational waves (Taylor & Weisberg 1982). This led to the Nobel Prize in physics in 1993 being awarded to Russel Hulse and Joseph Taylor Jr. for the first indirect evidence for gravitational waves. The field then made another step forward in 2015 with the first direct detection of gravitational waves from the inspiral and merging of two compact objects made by the Laser Interferometer Gravitational-Wave Observatory (LIGO) (Abbott et al. 2016). In the case of this first detection, the compact objects orbiting each other and eventually merging were black holes of masses $36^{+5}_{-4}\ M_{\odot}$ and $29^{+4}_{-4}\ M_{\odot}$, although in principle any massive object experiencing acceleration should create gravitational waves, it is these extremely massive, high speed objects who's gravitational waves can be observed at the moment (Schutz 1999). This observation has ushered in a new age of gravitational wave astronomy, a field that is extremely promising due to the characteristics of gravitational waves, namely that they interact with matter so weakly that there is little obstruction in the signal (Schutz 1999). To fully utilize the detection of gravitational waves from astronomical sources and participate in the new age of gravitational wave astronomy, a more complete understanding of Einstein’s theory of general relativity in extreme environments is needed to be obtained. This includes the development of tools using numerical approximations and simulations to better match observations with known physics.

It is the goal of this thesis to present an introduction to general relativity, as well as give the reader a basic understanding of numerical relativity and its applications to simulations that describe the inspiral and merger of binary neutron star systems. The main method for creating these simulations is the Einstein Toolkit, which is based on the ADM and BSSN formulation of numerical relativity (Löffler et al. 2012), all of which will be discussed in more detail later. The Einstein Toolkit suite of computational tools (Löffler et al. 2012) are used to perform high-level numerical relativity and relativistic astrophysical simulations on a modest computational machine. An important output product from these numerical simulations is the waveform of the gravitational waves that are emitted during the coalescence phase of the merger of compact objects. Waveforms generated using different masses of binary objects can be compared to LIGO detections to better understand the progenitors of the gravitational waves sources that are detected.

This thesis will delve into four cases of binary neutron star systems with various mass values, as well as different parameter files to better understand the importance of the usage of different thorns within the Einstein Toolkit to create meaningful simulations with realistic data. These test cases include; 1) two neutron stars each having the maximum observed mass value for a neutron star of 2.08 solar masses (Mazzali et al. 2007), 2) the minimum mass value of 1.4 solar masses, which is the Chandrasekar limit (Fonseca et al. 2021), 3) an intermediate value between the Chandrasekar value and the maximum observed mass, which is taken to be 1.74 solar masses, and 4) a case where one neutron star has the minimum value of 1.4 solar masses and the companion object has a mass of 1.74 solar masses. Although these cases are limited in their scope, they do provide a proving ground in which simulations can be tested and the input of relevant physics explored.

1.2. Convention

The sign convention that will be used in this thesis will be the $(-,+,+,+)$ sign convention. Some consider this sign convention to be more cumbersome to use in some applications. However, we feel that it makes the most physical sense and will be the convention adopted throughout this thesis. This thesis will also make liberal use of the Einstein summation convention to denote addition over repeated indices, e.g.

$$a^i b_i = \sum_{i=1}^{3} a^i b_i = a^1 b_1 + a^2 b_2 + a^3 b_3. \tag{1.2.1}$$

Additionally, the use of Latin indices will be used to denote values that may range over the spatial components, e.g.

$$a^i b_i = a^1 b_1 + a^2 b_2 + a^3 b_3. \tag{1.2.2}$$

The use of Greek indices will be used to indicate values that may range over the temporal and spatial components, e.g.

$$a^{\mu}b_{\mu} = a^0 b_0 + a^1 b_1 + a^2 b_2 + a^3 b_3. \tag{1.2.3}$$

At times it will also be useful to refer to a tensor as a whole and not by each component. In these cases, I will use the notation || ||, to refer to the whole tensor and not an individual component, e.g.

$$\overline{\overline{T}} = \left\| T_{ij} \right\|. \tag{1.2.4}$$

A convenient shorthand that will be used throughout this thesis will be the use of the symbol , used in the indices that will be used to denote the partial derivative of the object with respect to the thing that follows in the indices, e.g.

$$\frac{\partial S}{\partial x^i} = \partial_i = S_{,i}. \tag{1.2.7}$$

Finally, this thesis will use the convention of $G = c = M_{\odot} = 1$, meaning that any number given will be expressed in solar masses unless otherwise stated.

1.3. Primer on General Relativity

Before describing the research presented in this thesis, an introduction to general relativity is provided. The following derivation follows closely to that found in Misner et al. (2017).

1.3.1. The Metric

To have an understanding of general relativity, one needs at least a basic understanding of the metric tensor. The metric tensor is the object that general relativity is constructed from, and it fully describes the way that coordinate systems are defined. In order to understand the metric, one first needs a coordinate system to work with. The coordinate system will be made up of the coordinate basis $\hat{b}_i$, which is defined by the relationship

$$\hat{b}_i = \frac{\partial \vec{r}}{\partial x^i}, \tag{1.3.1.1}$$

where $\vec{r}$ is an abstract notion of a position vector and is not a vector that depends on the actual coordinates. Also, ∂x^i is an infinitesimal displacement in a direction used to define the basis. A familiar example of a vector depending on position is a vector in polar coordinates

$$\vec{v}(r,\theta) = v^r(r,\theta)\hat{r}(r,\theta) + v^\theta(r,\theta)\hat{\theta}(r,\theta). \tag{1.3.1.2}$$

This explicitly shows that the basis vectors $\hat{r}(r,\theta)$ and $\hat{\theta}(r,\theta)$, depend on the coordinates used. With the basis now defined, it is possible to define the metric tensor. The metric tensor is defined by the relationship

$$g_{ij} = \hat{b}_i \cdot \hat{b}_j. \tag{1.3.1.3}$$

From this definition of the metric, it should be symmetric, thus

$$g_{ij} = g_{ji}. \tag{1.3.1.4}$$

In general relativity the metric is an important object that describes how spacetime is deformed, and it is the natural generalization of the Minkowski metric that is used in special relativity (Einstein 1905). In general relativity it looks like the metric should be made up of 16 independent components. However, equation (1.3.1.4) brings the total number of independent components down to 10.

Another important property of the metric is the definition of the inner product. Consider the generic vectors $\vec{u}$ and $\vec{v}$ defined by

$$\vec{u} = u^i \hat{b}_i, \qquad (1.3.1.5a)$$

$$\vec{v} = v^j \hat{b}_j. \qquad (1.3.1.5b)$$

Taking the inner product of these vectors gives

$$\vec{u} \cdot \vec{v} = u^i \hat{b}_i \cdot v^j \hat{b}_j,$$

$$\vec{u} \cdot \vec{v} = g_{ij} u^i v^j. \qquad (1.3.1.6)$$

From this it is a natural extension to see that the line element is also based off the metric since the line element is defined by

$$ds^2 = d\vec{r} \cdot d\vec{r}. \qquad (1.3.1.7)$$

This generalizes to

$$ds^2 = dx^i \hat{b}_i \cdot dx^j \hat{b}_j = g_{ij} dx^i dx^j. \qquad (1.3.1.8)$$

With this definition of the metric tensor, it is helpful to have a version of the metric that it contravariant in addition to the covariant version of the metric. The key feature of the contravariant version of the metric is that it should be the matrix inverse of the covariant version of the metric, i.e.

$$\|g^{ij}\| \equiv \|g_{ij}\|^{-1}. \qquad (1.3.1.9)$$

This leads to the important relationship that

$$g^{ij} g_{jk} = \delta^i_k, \qquad (1.3.1.10)$$

with δ_k^i being the standard Kronecker delta. The last property of the metric is that it should raise or lower indices of a tensor. This can be seen by recalling the definition of the general vector used above, $\vec{v}$. To get the v_i component of the vector, take the inner product of the vector with the basis vector in that direction i.e.

$$v_i = \vec{v} \cdot \hat{b}_i. \tag{1.3.1.10}$$

Expanding this expression gives

$$\begin{aligned} v_i &= v^j \hat{b}_j \cdot \hat{b}_i, \\ v_i &= g_{ij} v^j. \end{aligned} \tag{1.3.1.11}$$

The contravariant version of the tensor is then able to raise the indices of a tensor from being a covariant component to being a contravariant component

$$v^i = g^{ij} v_j. \tag{1.3.1.12}$$

With a basic understanding of the metric, it is now possible to journey farther into the depths of general relativity with the covariant derivative.

1.3.2. Covariant Derivative

To begin, it is helpful to define a derivative that will work in all reference frames because, as seen in Einstein’s special theory of relativity (Einstein 1905), no one observer has a preferred reference frame over any other. This derivative, defined below, is known as the covariant derivative. For this task it is necessary to use the generic vector

$$\vec{v} = v^i \hat{b}_i. \tag{1.3.2.1}$$

Differentiating equation (1.3.2.1) gives

$$d\vec{v} = \hat{b}_i dv^i + v^i d\hat{b}_i, \tag{1.3.2.2}$$

since this differentiation follows the Leibniz rule, where dv^i is defined to be

$$dv^i = \frac{\partial v^i}{\partial x^j} dx^j, \tag{1.3.2.3}$$

and $d\hat{b}_i$ is given by

$$d\hat{b}_i = \Gamma_{ij}^k dx^j \hat{b}_k, \tag{1.3.2.4}$$

with Γ_{ik}^jbeing known as the connection coefficient, the formula of which will be determined later. It should be noted at this point that the connection coefficient is symmetric with respect to the two covariant indices. This can be seen by writing equation (1.3.2.4) in the form

$$\begin{gathered} \frac{\partial \hat{b}_i}{\partial x^j} = \Gamma_{ij}^k \hat{b}_k, \\ \frac{\partial}{\partial x^j} \frac{\partial \vec{r}}{\partial x^i} = \Gamma_{ij}^k \hat{b}_k. \end{gathered} \tag{1.3.2.5}$$

The left-hand side is normal partial differentiation, which follows Clairaut's theorem, showing that the connection coefficient is symmetric in the two covariant indices. This makes the covariant derivative

$$d\vec{v} = \frac{\partial v^i}{\partial x^j} dx^j \hat{b}_i + v^i \Gamma_{ij}^k dx^j \hat{b}_k. \tag{1.3.2.6}$$

This is the basis in which the covariant derivative is defined. Rearranging and changing the indices, the covariant derivative will be defined as

$$\nabla_j \vec{v} = \frac{d\vec{v}}{dx^j} = \left(\frac{\partial v^i}{\partial x^j} + \Gamma_{ij}^k v^i\right)\hat{b}_k. \qquad (1.3.2.7)$$

With the definition of the covariant derivative firmly established, it is important to know the properties of this derivative and how to use it properly.

The first important thing to note about the covariant derivative is that it does not satisfy Clairaut's theorem. To show this, first consider the covariant derivative taken with respect to x^j. This gives

$$\nabla_j \vec{v} = \left(\frac{\partial v^i}{\partial x^j} + \Gamma_{ij}^k v^i\right)\hat{b}_k. \qquad (1.3.2.8)$$

Taking the covariant derivative of this result with respect to x^m gives

$$\nabla_{mj} \vec{v} = \nabla_m \left(\frac{\partial v^i}{\partial x^j} + \Gamma_{ij}^k v^i\right)\hat{b}_k,$$

$$\nabla_{mj} \vec{v} = \left(\frac{\partial^2 v^i}{\partial x^m \partial x^j} + \frac{\partial v^i}{\partial x^m}\Gamma_{ij}^k + \Gamma_{im}^k \frac{\partial v^i}{\partial x^j} + v^i \frac{\partial}{\partial x^m}\Gamma_{ij}^k + v^i \Gamma_{im}^n \Gamma_{nj}^k\right)\hat{b}_k. \qquad (1.3.2.9)$$

Repeating the same process except interchanging the order of the differentiation between x^m and x^j gives

$$\nabla_{jm} \vec{v} = \left(v^i_{,jm} + v^i_{,j}\Gamma_{im}^k + \Gamma_{ij}^k v^i_{,m} + v^i \Gamma_{im,j}^k + v^i \Gamma_{ij}^n \Gamma_{nm}^k\right)\hat{b}_k. \qquad (1.3.2.10)$$

Finding the difference between these expressions gives

$$\nabla_{mj} \vec{v} - \nabla_{jm} \vec{v} = \left(\Gamma_{ij,m}^k - \Gamma_{im,j}^k + \Gamma_{im}^n \Gamma_{nj}^k - \Gamma_{ij}^n \Gamma_{nm}^k\right) v^i \hat{b}_k. \qquad (1.3.2.11)$$

The first terms cancel because they only involve partial derivatives, which interchange via Clairaut's theorem. Finally, the term in parentheses can be defined as

$$\Gamma^k_{ij,m} - \Gamma^k_{im,j} + \Gamma^n_{im}\Gamma^k_{nj} - \Gamma^n_{ij}\Gamma^k_{nm} = R^k_{imj}, \tag{1.3.2.12}$$

which is the Riemann Tensor. The Riemann Tensor and some of its properties will be discussed in further detail later.

The next question is how does the covariant derivative act on tensor of rank two or higher? First, consider the case of a rank two doubly contravariant tensor. By definition, a tensor of this type must transform in the same way as the product of two contravariant vectors so that the tensor can be substituted as the product of two contravariant vectors. The covariant derivative will be

$$\nabla_i T^{jk} = \nabla_i\left(u^j v^k\right). \tag{1.3.2.13}$$

However, since this is a derivative, the Leibnitz rule still applies so the derivative becomes

$$\nabla_i\left(u^j v^k\right) = u^j_{,i} v^k + u^j v^k_{,i} + \Gamma^j_{il} u^l v^k + \Gamma^k_{il} u^j v^l. \tag{1.3.2.14}$$

Combining the first two terms as a single partial derivative and replacing the vectors with the original tensor gives

$$\nabla_i\left(T^{jk}\right) = T^{jk}_{,i} + \Gamma^j_{il} T^{lk} + \Gamma^k_{il} T^{jl}. \tag{1.3.2.15}$$

This demonstrates that the way to take the contravariant derivative of a rank two doubly contravariant vector is to have two connection coefficients, one for each of the indices. This can be generalized to higher rank contravariant vectors, with one connection coefficient in the covariant derivative for each of the indices.

The final property of the covariant derivative that will be important is how to take the covariant derivative of a covariant tensor. To understand this, consider the scalar, which will be made up of the tensors, $T = u^j v_j$. The covariant derivative of this scalar is

$$\nabla_i T = \nabla_i\left(u^j v_j\right) = \left(\nabla_i u^j\right) v_j + u^j\left(\nabla_i v_j\right). \tag{1.3.2.16}$$

The covariant derivative of a scalar is the partial derivative of the scalar since the scalar does not depend on the coordinate basis. So, equation (1.3.2.16) can also be written as

$$\nabla_i T = \partial_i\left(u^j v_j\right) = \left(\partial_i u^j\right) v_j + u^j\left(\partial_i v_j\right). \tag{1.3.2.17}$$

Equating these equations and taking the covariant derivative of u^j gives

$$\left(\partial_i u^j\right) v_j + u^j\left(\partial_i v_j\right) = \left(u^j_{,i} + \Gamma^j_{il} u^l\right) v_j + u^j\left(\nabla_i v_j\right). \tag{1.3.2.18}$$

Which can be rewritten as

$$\left(\nabla_i v_j\right) u^j = \left(\partial_i v_j - \Gamma^l_{ji} v_l\right) u^j. \tag{1.3.2.19}$$

Meaning that the way the covariant derivative acts on a covariant tensor is given by

$$\nabla_i v_j = \partial_i v_j - \Gamma^l_{ij} v_l. \tag{1.3.2.20}$$

Which is the same way that the covariant acts on a contravariant tensor except that the connection coefficient has a negative sign instead of a positive sign. This can be generalized and combined with the result of equation (1.3.2.16) to find the covariant derivative of any tensor. This generalization is given by

$$\nabla_i T^{jk\dots}_{mn\dots} = T^{jk\dots}_{mn\dots,i} + \Gamma^j_{il} T^{lk\dots}_{mn\dots} + \Gamma^k_{il} T^{jl\dots}_{mn\dots} + \cdots - \Gamma^l_{im} T^{jk\dots}_{ln\dots} - \Gamma^l_{in} T^{jk\dots}_{ml\dots} - \cdots. \tag{1.3.2.21}$$

The rule is to take the covariant derivative of a tensor is to first take the partial derivative of the tensor with respect to x^i, then each index gets a connection coefficient with the contravariant indices getting a positive connection coefficient and the covariant indices getting a negative connection coefficient.

1.3.3. Christoffel Symbols

Now that the definition and some of the properties of the covariant derivative have been firmly established, it is time to determine how to find the connection coefficient which is also called the Christoffel symbol. To begin with recall the definition of the line element which is

$$ds^2 = d\vec{r} \cdot d\vec{r}. \tag{1.3.3.1}$$

Expressing this in an arbitrary basis gives the equation

$$ds^2 = dx^i \hat{b}_i \cdot dx^j \hat{b}_j = g_{ij} dx^i dx^j. \tag{1.3.3.2}$$

From this expression it is obvious that the metric can be defined by the dot product of the coordinate basis

$$\hat{b}_i \cdot \hat{b}_j = g_{ij}. \tag{1.3.3.3}$$

Taking the partial derivative of this definition of the metric in the direction of x^i gives

$$\partial_i g_{jk} = \partial_i \big(\hat{b}_j \cdot \hat{b}_k\big),$$

$$\partial_i g_{jk} = \Gamma^l_{ij} g_{lk} + \Gamma^l_{ik} g_{jl}.$$

Interchanging indices give the three relationships

$$\partial_i g_{jk} = \Gamma^l_{ij} g_{lk} + \Gamma^l_{ik} g_{jl},$$

$$\partial_j g_{ki} = \Gamma^l_{jk} g_{li} + \Gamma^l_{ji} g_{kl},$$

$$\partial_k g_{ij} = \Gamma^l_{jk} g_{li} + \Gamma^l_{ki} g_{jl}. \tag{1.3.3.4}$$

Adding the second expression to the first and subtracting the third gives

$$\partial_i g_{jk} + \partial_j g_{ki} - \partial_k g_{ij} = \Gamma^l_{ij} g_{lk} + \Gamma^l_{ik} g_{jl} + \Gamma^l_{jk} g_{li} + \Gamma^l_{ji} g_{kl} - \Gamma^l_{jk} g_{li} - \Gamma^l_{ki} g_{jl},$$

$$\partial_i g_{jk} + \partial_j g_{ki} - \partial_k g_{ij} = 2\Gamma^l_{ij} g_{lk}. \tag{1.3.3.5}$$

Combining these equations together in this way gives an expression for the Christoffel symbol which is

$$\Gamma^l_{ij} = \frac{1}{2} g^{lk}\left(\partial_i g_{jk} + \partial_j g_{ki} - \partial_k g_{ij}\right). \tag{1.3.3.6}$$

Now that it is possible to write the Christoffel symbol in terms of the metric, it is also possible to write the Riemann tensor in terms of the metric as well.

1.3.4. Riemann Tensor, Ricci Tensor, and Ricci Scalar

Now that the connection has been made that the Christoffel symbol depends on the coordinates, it is now possible to think about the Riemann tensor in terms of the metric as well. Recall that the definition of the Riemann tensor as described above is

$$R^i_{jkl} = \Gamma^i_{jl,k} - \Gamma^i_{jk,l} + \Gamma^m_{jk}\Gamma^i_{ml} - \Gamma^n_{jl}\Gamma^i_{nk}. \tag{1.3.4.1}$$

Lowering the upper index makes it easier to see some of the properties that will be used later. The Riemann tensor in this form looks like

$$R_{ijkl} = g_{io}\left(\Gamma^o_{jl,k} - \Gamma^o_{jk,l} + \Gamma^m_{jk}\Gamma^o_{ml} - \Gamma^n_{jl}\Gamma^o_{nk}\right). \tag{1.3.4.2}$$

In this form it is obvious that the Riemann tensor is antisymmetric in the first and second indices as well in the third and fourth indices. The Riemann tensor is also symmetric if the first and second indices are interchanged with the third and fourth. So, one can see that

$$R_{ijkl} = R_{jikl} = R_{ijlk} = R_{klij}. \tag{1.3.4.3}$$

Using these properties, it can be verified that

$$R_{ijkl} + R_{iklj} + R_{iljk} = 0. \tag{1.3.4.4}$$

Taking the covariant derivative of the Riemann tensor gives

$$\nabla_m R_{ijkl} = \nabla_m \left(g_{io}\left(\Gamma^o_{jl,k} - \Gamma^o_{jk,l} + \Gamma^m_{jk}\Gamma^o_{ml} - \Gamma^n_{jl}\Gamma^o_{nk}\right)\right). \tag{1.3.4.5}$$

However, the covariant derivative of the metric tensor is zero (Misner et al. 2017), so this becomes

$$\nabla_m R_{ijkl} = g_{io}\nabla_m\left(\Gamma^o_{jl,k} - \Gamma^o_{jk,l} + \Gamma^m_{jk}\Gamma^o_{ml} - \Gamma^n_{jl}\Gamma^o_{nk}\right). \tag{1.3.4.6}$$

This is simplified in a locally geodesic coordinate system since the Christoffel symbol vanishes, then the covariant derivative can be replaced by a partial derivative. This results in

$$\nabla_m R_{ijkl} = g_{io}\left(\Gamma^o_{jl,km} - \Gamma^o_{jk,lm}\right). \tag{1.3.4.7a}$$

Interchanging indices it is also true that

$$\nabla_k R_{ijlm} = g_{io}\left(\Gamma^o_{jm,lk} - \Gamma^o_{jl,mk}\right), \tag{1.3.4.7b}$$

$$\nabla_l R_{ijmk} = g_{io}\left(\Gamma^o_{jk,ml} - \Gamma^o_{jm,kl}\right). \tag{1.3.4.7c}$$

Combining these three equations gives

$$\nabla_m R_{ijkl} + \nabla_k R_{ijlm} + \nabla_l R_{ijmk} = 0, \tag{1.3.4.8}$$

which is the Bianchi identity.

Another tensor that is important in general relativity is the Ricci tensor. The Ricci tensor is a tensor that is formed by summing over the upper and the middle lower indices to form a second rank tensor

$$R_{ij} = R^k_{ikj} = \Gamma^k_{ij,k} - \Gamma^k_{ik,j} + \Gamma^m_{ik}\Gamma^k_{mj} - \Gamma^n_{ij}\Gamma^k_{nk}. \tag{1.3.4.9}$$

From this form it is also clear that the Ricci tensor must be symmetric in both of its indices as well. The final value that will be necessary before the derivation of the Einstein Field Equations is the Ricci scalar. The Ricci scalar is found by taking the trace of the Ricci tensor. It is given by

$$R = g^{ij}R_{ij} = g^{ij}\left(\Gamma^k_{ij,k} - \Gamma^k_{ik,j} + \Gamma^m_{ik}\Gamma^k_{mj} - \Gamma^n_{ij}\Gamma^k_{nk}\right). \tag{1.3.4.10}$$

With this final equation it is now time to move onto the Einstein Field Equations.

1.3.5. The Einstein Equations

The Einstein Field Equations will be assumed to take the form

$$G_{\mu\nu} = \kappa T_{\mu\nu}, \tag{1.3.5.1}$$

with $T_{\mu\nu}$ being the stress-energy tensor and $G_{\mu\nu}$ is the Einstein tensor. This equation assumes that the stress-energy tensor is the source of the gravitational field, and they are related through a proportionality constant, κ, which will be determined later. To determine how to construct the Einstein tensor, one must start with some leading assumptions:

1. The Einstein tensor must vanish when spacetime is flat, i.e., when the stress-energy tensor is zero.
2. The Einstein tensor must be constructed from the Riemann tensor and the metric tensor and nothing else.
3. The Einstein Tensor is distinguished from other tensors which can be built from the Riemann tensor and metric tensor from the demands that:
   a. The Einstein tensor be linear in Riemann, as befits any natural measure of curvature.
   b. Like the stress-energy tensor, the Einstein tensor should be symmetric and second rank.
   c. The Einstein tensor should naturally have a vanishing divergence

$$\nabla_\mu G_{\mu\nu} \equiv 0. \tag{1.3.5.2}$$

Applying conditions 2, 3a, and 3b, the most general Einstein tensor is

$$G_{\mu\nu} = R_{\mu\nu} + bRg_{\mu\nu} + \Lambda g_{\mu\nu}, \tag{1.3.5.3}$$

where b and Λ are scalar constants, and technically $R_{\mu\nu}$ should have a scalar constant that would get absorbed into κ in equation (1.3.5.1). Considering condition 1, one needs to know how the Riemann tensor behaves when spacetime is taken to be flat. Rewriting the definition of the Riemann tensor gives

$$R^{\alpha}_{\beta\gamma\delta} = \Gamma^{\alpha}_{\beta\delta,\gamma} - \Gamma^{\alpha}_{\beta\gamma,\delta} + \Gamma^{\epsilon}_{\beta\gamma}\Gamma^{\alpha}_{\epsilon\delta} - \Gamma^{\eta}_{\beta\delta}\Gamma^{\alpha}_{\eta\gamma}. \tag{1.3.5.4}$$

To begin, it is useful to look at how the Christoffel symbol behaves in a flat spacetime. The Christoffel symbol is

$$\Gamma^{\alpha}_{\beta\gamma} = \frac{1}{2} g^{\alpha\varepsilon}\left(g_{\gamma\varepsilon,\beta} + g_{\varepsilon\beta,\gamma} - g_{\beta\gamma,\epsilon}\right). \tag{1.3.5.5}$$

The metric in flat spacetime is

$$\|g_{\mu\nu}\| = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1.3.5.6}$$

Since none of these components depend on coordinates, the partial derivative of the metric is zero, meaning the Christoffel symbol is zero in all its entries as well. This also makes the Riemann tensor, Ricci tensor, and Ricci scalar all zero. So, in a flat spacetime the Einstein equation reduces to

$$\Lambda g_{\mu\nu} = 0. \tag{1.3.5.7}$$

For this equation to hold Λ must go to zero. To determine b, recall the Bianchi identity

$$\nabla_m R_{ijkl} + \nabla_k R_{ijlm} + \nabla_l R_{ijmk} = 0. \tag{1.3.5.8}$$

Multiplying through by $g^{nm} g^{ik} g^{jl}$ yields

$$\nabla_m g^{nm} g^{ik} g^{jl} R_{ijkl} + \nabla_k g^{nm} g^{ik} g^{jl} R_{ijlm} + \nabla_l g^{nm} g^{ik} g^{jl} R_{ijmk} = 0. \tag{1.3.5.9}$$

This simplifies to

$$\nabla_m g^{nm} R - \nabla_k R^{kn} - \nabla_l R^{ln} = 0. \tag{1.3.5.10}$$

Renaming indices gives

$$\nabla_m (g^{nm} R - 2R^{mn}) = 0. \tag{1.3.5.11}$$

Or

$$\nabla_m \left(R^{mn} - \frac{1}{2} g^{nm} R \right) = 0. \qquad (1.3.5.12)$$

So, it is clear that b must be $\frac{1}{2}$. So, the Einstein field equations become

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = \kappa T_{\mu\nu}. \qquad (1.3.5.13)$$

To determine the value for κ one needs to compare to the weak field limit of Newtonian gravitation.

In the case of weak gravitation general relativity needs to reduce to the Newtonian limit. The following is an informal derivation of the constant κ, following Pe'er (2020). The Newtonian potential for gravitation is

$$\nabla^2 \Phi = 4\pi\rho. \qquad (1.3.5.14)$$

In the Newtonian limit the time-time component of the metric is given by

$$g_{00} = -(1 + 2\Phi), \qquad (1.3.5.15)$$

and the density is equal to the time-time component of the stress-energy tensor, $T_{00} = \rho$. Additionally, in the weak field approximation the metric is assumed to be

$$g_{00} = \gamma_{00} + h_{00} = -1 + h_{00}. \qquad (1.3.5.16)$$

Using this and equation (1.3.5.15), equation (1.3.5.14) becomes

$$\nabla^2 h_{00} = -8\pi\rho. \qquad (1.3.5.17)$$

Another valid way of writing the Einstein field equations is in the form

$$R_{\mu\nu} = \kappa \left(T_{\mu\nu} - \frac{1}{2} g_{\mu\nu} T \right). \tag{1.3.5.18}$$

In the weak-field limit it will be assumed that the solution is time independent, and all particles will be slow moving. Additionally, the T_{00} component will be much larger than any other component so it shall be the only component considered. In the weak-field limit spacetime is approximately flat so the covariant and contravariant versions of the metric will be assumed to be

$$\begin{aligned} g_{00} &= \gamma_{00} + h_{00}, \\ g^{00} &= \gamma^{00} - h^{00}, \end{aligned} \tag{1.3.5.19}$$

with γ_{00} being the time-time component of the Minkowski metric and h_{00} being the time-time component of a small perturbation. The trace of the energy-momentum tensor, ignoring small terms, is approximately

$$T = g^{00} T_{00} = -T_{00}. \tag{1.3.5.20}$$

The next piece to consider is the time-time component of the Ricci tensor. Recall that the Ricci tensor is the trace of the Riemann tensor, so it can be written as

$$R_{00} = R^{\mu}_{0\mu0} = \partial_\mu \Gamma^{\mu}_{00} - \partial_0 \Gamma^{\mu}_{\mu0} + \Gamma^{\mu}_{\mu\nu} \Gamma^{\nu}_{00} - \Gamma^{\mu}_{0\nu} \Gamma^{\nu}_{\mu0}. \tag{1.3.5.21}$$

However, since the second term contains a time derivative, it is zero for static fields. Additionally, since the metric is flat with a small perturbation, the Christoffel symbol itself is small and the third and fourth terms are of the form Γ^2 and they are also approximately zero. So, the time-time component of the Ricci tensor is approximately

$$R_{00} = R^{\mu}_{0\mu0} = \partial_\mu \Gamma^{\mu}_{00},$$

$$= \partial_\mu \left(\frac{1}{2} g^{\mu\nu} (\partial_0 g_{\nu 0} + \partial_0 g_{0\nu} - \partial_\nu g_{00}) \right),$$

$$= -\frac{1}{2} \gamma^{\mu\nu} \partial_\mu \partial_\nu h_{00},$$

$$= -\frac{1}{2} \nabla^2 h_{00}. \qquad (1.3.5.22)$$

Combining this with equation (1.3.5.18) gives

$$-\frac{1}{2} \nabla^2 h_{00} = \kappa \left(T_{00} - \frac{1}{2} g_{00} T \right),$$

$$\nabla^2 h_{00} = -\kappa T_{00}. \qquad (1.3.5.23)$$

Comparing this with equation (1.3.5.17), it can be seen that $\kappa = 8\pi$. So, the Einstein field equations become

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = 8\pi T_{\mu\nu}. \qquad (1.3.5.24)$$

Reinserting G and c, the Einstein field equations become

$$R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R = \frac{8\pi G}{c^4} T_{\mu\nu}. \qquad (1.3.5.24)$$

With this form of the Einstein field equations, it is possible to begin to look for solutions.

1.3.6. TOV Equations

The TOV equations are a solution to the Einstein field equations that describe a spherically symmetric, isotropic body. One such object is a neutron star in which gravity is the force binding the object together and neutron degeneracy is the force that is keeping the object from collapsing further. The derivation of the TOV equations will follow along with the original papers (Oppenheimer & Volkoff 1939; Tolman 1939). To begin with, consider the line element of a spherically symmetric object which is given by

$$ds^2 = -e^{\nu}dt^2 + e^{\lambda}dr^2 + r^2d\theta^2 + r^2sin^2\theta d\phi^2. \tag{1.3.6.1}$$

Here

$$\lambda = \lambda(r), \qquad \nu = \nu(r).$$

If there is no transverse stresses and no mass motion, then the stress-energy tensor is given by

$$T_{11} = T_{22} = T_{33} = P, \qquad T_{11} = -\rho,$$

with P and ρ the pressure and the energy density measured in proper coordinates. Considering these equations, the Einstein field equations reduce to

$$8\pi\rho = e^{-\lambda}\left(\frac{\lambda'}{r} - \frac{1}{r^2}\right) + \frac{1}{r^2}, \tag{1.3.6.2}$$

$$8\pi P = e^{-\lambda}\left(\frac{\nu'}{r} + \frac{1}{r^2}\right) - \frac{1}{r^2}, \tag{1.3.6.3}$$

$$\frac{dP}{dr} = -\frac{(P+\rho)}{2}\nu'. \tag{1.3.6.4}$$

The primes above denote derivatives with respect to r. The last equation follows from the continuity equation and demanding that the pressure and density be static and isotropic. These equations along with the equation of state (EOS) will determine the equilibrium of the star. At the boundary of the star, $r = r_b$, there is no matter so $P = 0$. As well as for $r < r_b$, the solution is dependent on the EOS to connect the pressure with the density, and it must be the case that $P > 0$.

In empty space surrounding the body, the metric should reduce to Schwarzschild's exterior solution, which gives

$$e^{-\lambda(r)} = 1 + \frac{A}{r}, \tag{1.3.6.5}$$

$$e^{\nu(r)} = B\left(1 + \frac{A}{r}\right). \tag{1.3.6.6}$$

The constants A and B can be found using the weak-field approximation far from the object and are found to be $A = -2m$, $B = 1$, m being the mass of the object that a distant observer would measure. Placing these substitutions into equation (1.3.6.3) gives

$$v' = \frac{1}{r}\left(1 - \frac{2m}{r}\right)^{-1}\left(\frac{2m}{r} + 8\pi r^2 P\right). \tag{1.3.6.7}$$

This combined with equation (1.3.6.4) gives

$$\frac{dP}{dr} = -\frac{1}{r^2}(\rho + P)(m + 4\pi r^3 P)\left(1 - \frac{2m}{r}\right)^{-1}, \tag{1.3.6.8}$$

which is the TOV equation. This result along with the EOS, which will be described later, is all that is needed to fully describe the matter distribution around a spherically symmetric body.

1.3.7. Gravitational Waves

With a basic understanding of Einstein's general theory of relativity, one can begin with creating a description of waves propagating though spacetime. These waves are the gravitational waves that are seen to propagate outward in numerical simulations, as well as the waves detected by LIGO (Abbott et al. 2016). Because the sources of these gravitational waves are far away, these waves will be approximated as plane waves. The derivation of these plane waves will follow along with Eddington (1922).

To begin this derivation, we start with the metric

$$g_{\mu\nu} = \gamma_{\mu\nu} + h_{\mu\nu}. \tag{1.3.7.1}$$

As defined before, $\gamma_{\mu\nu}$ is the Minkowski flat spacetime metric and $h_{\mu\nu}$ is the small perturbation representing gravitational waves. Consider plane waves proceeding with velocity V in the negative x_1-direction, so that $h_{\mu\nu}$ are periodic functions of the argument $(x_1 + Vx_0)$. Denoting differentiating with respect to this argument by an apostrophe, and differentiating equation (1.3.7.1) twice, gives

$$\frac{\partial^2 g_{\mu\nu}}{\partial {x^1}^2} = h_{\mu\nu}'', \tag{1.3.7.2a}$$

$$\frac{\partial^2 g_{\mu\nu}}{\partial x^1 \partial x^0} = V h_{\mu\nu}'', \tag{1.3.7.2b}$$

$$\frac{\partial^2 g_{\mu\nu}}{\partial {x^0}^2} = V^2 h_{\mu\nu}'', \tag{1.3.7.2c}$$

with all other second derivatives being zero. Recalling the definition of the Riemann tensor in terms of the metric

$$R_{\mu\rho\nu\sigma} = \frac{1}{2}\left(\frac{\partial^2 g_{\mu\nu}}{\partial x^\sigma \partial x^\rho} + \frac{\partial^2 g_{\sigma\rho}}{\partial x^\mu \partial x^\nu} - \frac{\partial^2 g_{\mu\sigma}}{\partial x^\nu \partial x^\rho} - \frac{\partial^2 g_{\nu\rho}}{\partial x^\mu \partial x^\sigma}\right).$$

Substituting equations (1.3.7.2a-c), gives

$$R_{0101} = \frac{1}{2} V^2 h_{11}'' - V\, h_{01}'' + \frac{1}{2} h_{00}'',$$

$$R_{0102} = \frac{1}{2} V^2 h_{21}'' - \frac{1}{2} V\, h_{02}'',$$

$$R_{0103} = \frac{1}{2} V^2 h_{31}'' - \frac{1}{2} V\, h_{03}'',$$

$$R_{0112} = \frac{1}{2} V\, h_{21}'' - \frac{1}{2} h_{02}'',$$

$$R_{0113} = \frac{1}{2} V\, h_{31}'' - \frac{1}{2} h_{03}'',$$

$$R_{0202} = \frac{1}{2} V^2 h_{22}'' \tag{1.3.7.3}$$

$$R_{0123} = 0 \qquad R_{0203} = \frac{1}{2} V^2 h_{32}''$$

$$R_{0213} = \frac{1}{2} V\, h_{32}''$$

$$R_{0212} = \frac{1}{2} V\, h_{22}'' \qquad R_{0223} = 0$$

$$R_{0312} = \frac{1}{2} V\, h_{23}''$$

$$R_{0303} = \frac{1}{2} V^2 h_{33}'' \qquad R_{0313} = \frac{1}{2} V\, h_{33}''$$

$$R_{1212} = \frac{1}{2} h_{22}''$$

$$R_{0323} = 0 \qquad R_{1213} = \frac{1}{2} h_{32}''$$

$$R_{1223} = 0 \qquad R_{1313} = \frac{1}{2} h_{33}'' \qquad R_{1323} = 0$$

$$R_{2323} = 0.$$

To a first order approximation for small quantities, the Einstein tensor is given by

$$G_{\mu\nu} = g^{\sigma\rho} R_{\mu\nu\sigma\rho}. \tag{1.3.7.4}$$

Taking this and the antisymmetric properties of the tensor give

$$G_{00} = -R_{1010} - R_{2020} + R_{3030} = 0,$$

$$G_{11} = -R_{1212} - R_{1313} + R_{1010} = 0,$$

$$G_{22} = -R_{1212} - R_{2323} + R_{2020} = 0,$$

$$G_{33} = -R_{1313} - R_{2323} + R_{3030} = 0,$$

$$G_{10} = R_{1220} + R_{1330} = 0,$$

$$G_{12} = -R_{1323} + R_{1020} = 0,$$

$$G_{13} = R_{1223} + R_{1030} = 0,$$

$$G_{20} = -R_{1214} + R_{2334} = 0,$$

$$G_{23} = -R_{1213} + R_{2030} = 0,$$

$$G_{30} = -R_{1210} - R_{2320} = 0. \qquad (1.3.7.5)$$

Substituting in equation (1.3.7.3), gives

$$-(V^2 h''_{11} - 2V h''_{10} + h''_{00}) - V^2(h''_{22} + h''_{33}) = 0,$$

$$-(h''_{22} + h''_{33}) + V^2 h''_{11} - 2V h''_{10} + h''_{00} = 0,$$

$$-h''_{22} + V^2 h''_{22} = 0,$$

$$-h''_{33} + V^2 h''_{33} = 0,$$

$$-V(h''_{22} + h''_{33}) = 0,$$

$$V^2 h''_{12} - V h''_{20} = 0,$$

$$V^2 h''_{13} - V h''_{30} = 0,$$

$$-h''_{20} + V h''_{12} = 0,$$

$$-h''_{23} + V^2 h''_{23} = 0,$$

$$-h''_{30} + V h''_{13} = 0. \qquad (1.3.5.6)$$

Integrating these equations, which in this case, since $h_{\mu\nu}$ are periodic functions, is akin to removing the apostrophes, gives the following conditions

$$h_{22} + h_{33} = 0,$$

$$(1 - V^2) h_{22} = 0,$$

$$(1 - V^2) h_{33} = 0,$$

$$\begin{aligned}(1-V^2)h_{23} &= 0,\\ h_{20} &= Vh_{12},\\ h_{30} &= Vh_{13},\\ h_{00}-2Vh_{10}+V^2h_{11} &= 0.\end{aligned} \tag{1.3.5.7}$$

These can be separated into different types of waves

Transverse-transverse (TT)	h_{22}, h_{33}, h_{23}
Longitudinal-transverse (LT)	$h_{12}, h_{13}, h_{20}, h_{30}$
Longitudinal-longitudinal (LL)	h_{00}, h_{10}, h_{11}.

For a transverse-transverse wave, h_{22}, h_{33}, and h_{23} cannot vanish so it must be the case that

$$1-V^2=0. \tag{1.3.5.8}$$

Which dictates that TT waves must propagate with a velocity of 1, i.e., the speed of light.

For LL and LT waves h_{22}, h_{33}, and h_{23} must be zero so there is no equation to determine V independent of the coefficients of the disturbance. Additionally, taking the conditions of equation (1.3.5.7) into account, the Riemann tensor only contains components depending on h_{22}, h_{33}, and h_{23}. So, for LL and LT waves, the Riemann tensor vanishes so the spacetime is flat and waves seem to completely vanish. This shows that the only type of waves that should propagate through spacetime are transverse-transverse waves, with the wave travelling in the x-direction and oscillations in the y and z-directions of the same amplitude that are 180° out of phase with each other. These are the same type of waves that have been detected by LIGO (Abbott et al. 2016), as well as the type of waves detected by the thorn `WeylScal4` (Zilhão & Löffler 2013).

1.4. Numerical Relativity

In most cases it is not possible to solve the Einstein field equations analytically. The cases where an analytical solution exists, it is in simple cases like the Schwarzschild solution which describes an eternal, static, rotation-free black hole (Schwarzschild 1916). To see solutions that could physically exist, it is usually necessary to solve the Einstein equations numerically and to step that solution in time to approximate how that system will evolve; this section deals with that topic through the Arnowitt, Deser, Misner (ADM) and the Baumgarte, Shapiro, Shibata, Nakamura (BSSN) formalisms of numerical relativity.

1.4.1. ADM Formulation

To have a numerical formalism of general relativity, first it is necessary to be able to split the four dimensions of spacetime into a 3+1 formalism with three spatial dimensions and one temporal dimension. The slices of space at a given time will be denoted by Σ_t, which are three-dimensional slices of space threaded together through a dimension of time. It is important to have a way to determine which direction the flow of time is facing, i.e., which direction is normal to the three dimensions of space, to have a way to flow from one slice Σ_t to a later slice Σ_{t+dt}. For that, define the 1-form

$$\Omega_\mu = \nabla_\mu t. \tag{1.4.1.1}$$

This is a vector that is normal to the three spatial dimensions since it contains a derivative of time. To normalize this vector, it will be multiplied by the normalization factor α to get the normal unit vector

$$n_\mu = \alpha \Omega_\mu, \tag{1.4.1.2}$$

where α can be shown to be related to the metric via

$$\alpha^2 = \frac{1}{g^{00}}, \tag{1.4.1.3}$$

meaning that α describes how time evolves for each point of the three-dimensional slice. This normalization constant, $\alpha,$ is known as the lapse function and it is a vital component in evolving a system forward in time. Using this normal vector, it is possible to introduce the time projection operator

$$N_{\nu}^{\mu} = n^{\mu} n_{\nu}. \tag{1.4.1.4}$$

In addition to the time projection operator, a space projection operator can be formed from this normal vector, which takes the form

$$P_{\nu}^{\mu} = \delta_{\nu}^{\mu} + n^{\mu} n_{\nu}. \tag{1.4.1.4}$$

With these projection operators, another valuable tool in the 3+1 formalism of numerical relativity is what is known as the shift vector $\vec{\beta}$. Using the lapse function and the shift vector, the time vector can be written as

$$t^{\mu} = \alpha n^{\mu} + \beta^{\mu}. \tag{1.4.1.5}$$

With this definition of a time vector, α describes the temporal distance between the two slices and $\vec{\beta}$ describes how the spatial coordinates vary from slice to slice.

With the lapse function, shift vector, and projection operators as described, it is possible to look at two different curvatures, the intrinsic and extrinsic curvature. The intrinsic curvature is the space-space components of the metric tensor, and they are related through

$$\gamma_{\mu\nu} = P_\nu^\delta P_\mu^\varepsilon g_{\delta\varepsilon} = g_{\mu\nu} + n_\mu n_\nu. \tag{1.4.1.6}$$

The extrinsic curvature is related to the normal vector and is given by

$$K_{\mu\nu} = -P_\mu^\alpha \nabla_\alpha n_\nu = -\nabla_\mu n_\nu - n_\mu n^\alpha \nabla_\alpha n_\nu. \tag{1.4.1.7}$$

Taking the Lie derivative of the spatial metric along the normal vector gives

$$\begin{aligned}
\mathcal{L}_{\vec{n}}\gamma_{\mu\nu} &= n^\sigma \nabla_\sigma \gamma_{\mu\nu} + \gamma_{\sigma\nu}\nabla_\mu n^\sigma + \gamma_{\mu\sigma}\nabla_\nu n^\sigma,\\
&= n^\sigma \nabla_\sigma \left(n_\mu n_\nu\right) + \gamma_{\sigma\nu}\nabla_\mu n^\sigma + \gamma_{\mu\sigma}\nabla_\nu n^\sigma,\\
&= n^\sigma n_\nu \nabla_\sigma n_\mu + n^\sigma n_\mu \nabla_\sigma n_\nu + \gamma_{\sigma\nu}\nabla_\mu n^\sigma + \gamma_{\mu\sigma}\nabla_\nu n^\sigma,
\end{aligned}$$

$$\mathcal{L}_{\vec{n}}\gamma_{\mu\nu} = -2K_{\mu\nu}. \tag{1.4.1.8}$$

To get this in a form that is more useful, it is important to create a basis, $e_{(i)}^j$ with $i = 1,2,3$ in the spatial slices. The basis is defined such that

$$\Omega_j e_{(i)}^j = 0. \tag{1.4.1.9}$$

Inserting equation (1.4.1.2) gives

$$\Omega_\mu e_{(i)}^\mu = -\frac{1}{\alpha} n_\mu e_{(i)}^\mu \Rightarrow n_i = 0. \tag{1.4.1.10}$$

The time basis was defined above in equation (1.4.1.5), which gives

$$t^\mu \Omega_\mu = \alpha n^\mu \Omega_\mu + \beta^\mu \Omega_\mu = 1 \Rightarrow \|t^\mu\| = (1,0,0,0). \tag{1.4.1.11}$$

This gives

$$\mathcal{L}_t = \partial_t. \tag{1.4.1.12}$$

With this new information, equation (1.4.1.5) can now be written as

$$n^{\mu} = \frac{1}{\alpha}(t^{\mu} - \beta^{\mu}). \tag{1.4.1.13}$$

Now the metric can be written as

$$\|g_{\mu\nu}\| = \begin{pmatrix} -\alpha + \beta_l \beta^l & \beta_i \\ \beta_j & \gamma_{ij} \end{pmatrix}. \tag{1.4.1.14}$$

Taking into consideration equation (1.4.1.13), the Lie derivative of the spatial metric becomes

$$\mathcal{L}_{\vec{n}} \gamma_{ij} = \frac{1}{\alpha}\left(\mathcal{L}_t - \mathcal{L}_{\vec{\beta}}\right)\gamma_{ij}, \tag{1.4.1.15}$$

and the Lie derivative of the spatial metric with respect to the shift vector is

$$\mathcal{L}_{\vec{\beta}} \gamma_{ij} = \beta^k \partial_k \gamma_{ij} + \gamma_{kj} \partial_i \beta^k + \gamma_{ik} \partial_j \beta^k. \tag{1.4.1.16}$$

This can be generalized by replacing the partial derivatives with covariant derivatives with respect to the spatial metric, which will be denoted by D_i. Then the Lie derivative of the spatial metric becomes

$$\begin{aligned} \mathcal{L}_{\vec{\beta}} \gamma_{ij} &= \beta^k D_k \gamma_{ij} + \gamma_{kj} D_i \beta^k + \gamma_{ik} D_j \beta^k, \\ &= D_i \gamma_{kj} \beta^k + D_j \gamma_{ik} \beta^k, \\ &= D_i \beta_j + D_j \beta_i. \end{aligned} \tag{1.4.1.17}$$

Combining this with equations (1.4.1.15) and (1.4.1.8), gives

$$K_{ij} = \frac{1}{2\alpha}\left(-\partial_t \gamma_{ij} + D_i \beta_j + D_j \beta_i\right). \tag{1.4.1.18}$$

Taking only the spatial components of this gives an equation for how the spatial metric will evolve in time

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} + D_i\beta_j + D_j\beta_i. \tag{1.4.1.19}$$

To get the other evolution equations, one would start applying the projection operators on the Einstein equation. To begin, consider the three unique projections of the Riemann tensor

$$P_\alpha^\sigma P_\beta^\tau P_\mu^\gamma P_\nu^{\delta\,(4)} R_{\sigma\tau\gamma\delta},$$

$$n^\delta P_\alpha^\sigma P_\beta^\tau P_\mu^{\gamma\,(4)} R_{\sigma\tau\gamma\delta},$$

$$n^\tau n^\delta P_\alpha^\sigma P_\beta^{\gamma\,(4)} R_{\sigma\tau\gamma\delta}.$$

Here ${}^{(4)}R_{\sigma\tau\gamma\delta}$ is used to denote the four-dimensional Riemann tensor versus the three-dimensional Riemann tensor that will be used in the 3+1 equations. These three equations simplify to

$$P_\alpha^\sigma P_\beta^\tau P_\mu^\gamma P_\nu^{\delta\,(4)} R_{\sigma\tau\gamma\delta} = {}^{(3)}R_{\sigma\tau\gamma\delta} + K_{\alpha\mu}K_{\beta\nu} + K_{\alpha\nu}K_{\beta\mu}, \tag{1.4.1.20}$$

$$n^\delta P_\alpha^\sigma P_\beta^\tau P_\mu^{\gamma\,(4)} R_{\sigma\tau\gamma\delta} = D_\beta K_{\alpha\mu} - D_\alpha K_{\beta\mu}, \tag{1.4.1.21}$$

$$n^\tau n^\delta P_\alpha^\sigma P_\beta^{\gamma\,(4)} R_{\sigma\tau\gamma\delta} = \mathcal{L}_{\vec{n}} K_{\alpha\beta} + \frac{1}{\alpha} D_\alpha D_\beta \alpha + K^{\sigma\beta} K_{\alpha\sigma}. \tag{1.4.1.22}$$

Multiplying the first equation twice by the spatial metric gives

$$\begin{aligned}\gamma^{\alpha\mu}\gamma^{\beta\nu} P_\alpha^\sigma P_\beta^\tau P_\mu^\gamma P_\nu^{\delta\,(4)} R_{\sigma\tau\gamma\delta} &= \gamma^{\sigma\gamma}\gamma^{\tau\delta\,(4)} R_{\sigma\tau\gamma\delta}, \\ &= {}^{(3)}R + K^2 - K_{\mu\nu}K^{\mu\nu}. \end{aligned} \tag{1.4.1.23}$$

The first equation's right-hand side can also be written as

$$\begin{aligned}\gamma^{\sigma\gamma}\gamma^{\tau\delta\,(4)} R_{\sigma\tau\gamma\delta} &= (g^{\sigma\gamma} + n^\sigma n^\gamma)\big(g^{\tau\delta} + n^\tau n^\delta\big)^{(4)} R_{\sigma\tau\gamma\delta}, \\ &= 2n^\mu n^{\nu\,(4)} R_{\mu\nu} + {}^{(4)}R. \end{aligned} \tag{1.4.1.24}$$

These expressions will be used later. Similarly, the stress-energy tensor can be projected into

$$n^{\mu}n^{\nu}T_{\mu\nu} = \rho, \tag{1.4.1.25}$$

$$P^{i}_{\mu}n_{\nu}T^{\mu\nu} = j^{i}, \tag{1.4.1.26}$$

$$P^{i}_{\mu}P^{j}_{\nu}T^{\mu\nu} = S^{ij}. \tag{1.4.1.27}$$

Multiplying the Einstein tensor by $2n^{\mu}n^{\nu}$ gives

$$\begin{aligned} 2n^{\mu}n^{\nu}G_{\mu\nu} &= 2n^{\mu}n^{\nu}\left({}^{(4)}R_{\mu\nu} - \frac{1}{2}{}^{(4)}Rg_{\mu\nu}\right), \\ &= 2n^{\mu}n^{\nu\,(4)}R_{\mu\nu} + {}^{(4)}R. \end{aligned} \tag{1.4.1.28}$$

Combining this with equation (1.4.1.23) gives

$$2n^{\mu}n^{\nu}G_{\mu\nu} = {}^{(3)}R + K^{2} - K_{\mu\nu}K^{\mu\nu}. \tag{1.4.1.29}$$

Recall the Einstein field equations are $G_{\mu\nu} = 8\pi T_{\mu\nu}$, and with the time-time projection of the stress-energy tensor, equation (1.4.1.29) becomes

$${}^{(3)}R + K^{2} - K_{\mu\nu}K^{\mu\nu} = 16\pi\rho. \tag{1.4.1.30}$$

This is not a time evolution equation but is a conservation equation that is known as the Hamiltonian Constraint.

Repeating the same procedure with equations (1.4.1.21) and (1.4.1.22) gives another constraint equation, the momentum constraint, as well as another evolution equation

$$D_{i}K - D_{j}K^{j}_{i} = 8\pi j_{i}, \tag{1.4.1.31}$$

$$\partial_t K_{ij} = -D_i D_j \alpha + \alpha\left({}^{(3)}R_{ij} - 2K_{ik}K_j^k + KK_{ij}\right) - 8\pi\alpha\left(S_{ij} - \frac{1}{2}(S-\rho)\right)$$

$$+\beta^k D_k K_{ij} + K_{ik} D_i \beta^k + K_{kj} D_j \beta^k. \qquad (1.4.1.32)$$

This is the last of the evolution equations. To recap, the evolution equations are

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

$$\partial_t K_{ij} = -D_i D_j \alpha + \alpha\left({}^{(3)}R_{ij} - 2K_{ik}K_j^k + KK_{ij}\right) - 8\pi\alpha\left(S_{ij} - \frac{1}{2}(S-\rho)\right) + \beta^k D_k K_{ij}$$

$$+ K_{ik} D_i \beta^k + K_{kj} D_j \beta^k.$$

Along with the Hamiltonian constraint and the momentum constraint

$${}^{(3)}R + K^2 - K_{ij}K^{ij} = 16\pi\rho,$$

$$8\pi j_i = D_i K - D_j K_i^j,$$

these make up the ADM formulation of numerical relativity. However, these are unstable and as such do not make a particularly good method for creating numerical relativistic simulations. A better formulation to create simulations is the BSSN formulation discussed in the next section.

1.4.2. BSSN Formulation

The BSSN formulism of numerical relativity is based on the ADM formalism. The necessity of this formalism stems from the fact that the evolution equations as well as the constraint equations both contain second derivatives of the metric. To begin to derive the equations of the BSSN formalism it is useful to rewrite the extrinsic curvature as a sum of its trace and its traceless parts

$$K_{ij} = A_{ij} + \frac{1}{3}\gamma_{ij}K. \tag{1.4.2.1}$$

Taking the determinants of equations (1.4.1.19) and (1.4.1.32) gives

$$\partial_t \gamma^{\frac{1}{2}} = -\alpha K + D_i\beta^i, \tag{1.4.2.2}$$

$$\partial_t = -D^2\alpha + \alpha\left(K^{ij}K_{ij} + 4\pi(\rho + S)\right) + \beta^i D_i K. \tag{1.4.2.3}$$

The spatial metric is rewritten as

$$\bar{\gamma}_{ij} = e^{-4\Phi}\gamma_{ij}, \tag{1.4.2.4}$$

where Φ is the conformal factor and $\bar{\gamma}_{ij}$ is the conformal metric, with the requirement that $\bar{\gamma} = 1$. The same can be done with the trace-free part of the extrinsic curvature

$$\bar{A}_{ij} = e^{-4\Phi}A_{ij}. \tag{1.4.2.5}$$

Another object that will be useful later is the connection coefficient in terms of the conformal transformation. Substituting equation (1.4.2.4) into the definition of the connection coefficient equation (1.3.3.6), shows that, in three dimensions, the connection coefficient must transform according to

$$\Gamma^i_{jk} = \bar{\Gamma}^i_{jk} + 2\left(\delta^i_j\bar{D}_k\Phi + \delta^i_k\bar{D}_j\Phi - \bar{\gamma}_{jk}\bar{\gamma}^{il}\bar{D}_l\Phi\right), \tag{1.4.2.6}$$

with $\bar{D}$ being the three-dimensional conformal covariant derivative. This can be inserted into the definition of the Ricci tensor, equation (1.3.4.9), to give

$$R_{ij} = \bar{R}_{ij} - 2\left(\bar{D}_i\bar{D}_j\Phi + \bar{\gamma}_{ij}\bar{\gamma}^{lm}\bar{D}_l\bar{D}_m\Phi\right) + 4\left((\bar{D}_i\Phi)(\bar{D}_j\Phi) - \bar{\gamma}_{ij}\bar{\gamma}^{lm}(\bar{D}_l\Phi)(\bar{D}_m\Phi)\right). \tag{1.4.2.7}$$

Taking the trace of equations (1.4.2.2) and (1.4.2.3) gives the expressions

$$\partial_t \Phi = -\frac{1}{6}\alpha K + \frac{1}{6}\beta^i \partial_i \Phi + \frac{1}{6}\partial_i \beta^i, \tag{1.4.2.8}$$

$$\partial_t K = -\gamma^{ij} D_i D_j \alpha + \alpha\left(\bar{A}_{ij}\bar{A}^{ij} + \frac{1}{3}K^2\right) + 4\pi\alpha(\rho + S) + \beta^i \partial_i K. \tag{1.4.2.9}$$

Subtracting these equations from the evolution equations (1.4.1.19) and (1.4.1.32) leaves the trace-free part of the evolution equations for $\bar{\gamma}_{ij}$ and $\bar{A}_{ij}$

$$\partial_t \bar{\gamma}_{ij} = -2\alpha\bar{A}_{ij} + \beta^k \partial_k \bar{\gamma}_{ij} + \bar{\gamma}_{ik}\partial_j \beta^k + \bar{\gamma}_{kj}\partial_i \beta^k - \frac{2}{3}\bar{\gamma}_{ij}\partial_k \beta^k, \tag{1.4.2.10}$$

$$\partial_t \bar{A}_{ij} = e^{-4\Phi}\left(-\left(D_i D_j \alpha\right)^{TF} + \alpha\left(R_{ij}^{TF} - 8\pi S_{ij}^{TF}\right)\right) + \alpha\left(K\bar{A}_{ij} - 2\bar{A}_{il}\bar{A}_j^l\right)$$

$$+\beta^k \partial_k \bar{A}_{ij} + \bar{A}_{ik}\partial_j \beta^k + \bar{A}_{kj}\partial_i \beta^k - \frac{2}{3}\bar{A}_{ij}\partial_k \beta^k, \tag{1.4.2.11}$$

where the superscript TF indicates the trace-free part of the tensor, e.g., $R_{ij}^{TF} = R_{ij} - \frac{1}{3}\gamma_{ij}R$. Additionally, a new variable can be defined, which is the conformal connection coefficient

$$\bar{\Gamma}^i \equiv \bar{\gamma}^{jk}\bar{\Gamma}_{jk}^i = -\partial_j \bar{\gamma}^{ij}, \tag{1.4.2.12}$$

where the last part of the equality holds in Cartesian coordinates when $\bar{\gamma} = 1$. With this definition of the conformal connection coefficient, the conformal Ricci tensor can be written as

$$\bar{R}_{ij} = -\frac{1}{2}\bar{\gamma}^{lm}\partial_m \partial_l \bar{\gamma}_{ij} + \bar{\gamma}_{ki}\partial_j \bar{\Gamma}^k + \bar{\gamma}_{kj}\partial_i \bar{\Gamma}^k + \bar{\Gamma}^k \bar{\Gamma}_{ijk} + \bar{\Gamma}^k \bar{\Gamma}_{jik}$$

$$+\bar{\gamma}^{lm}\left(2\bar{\Gamma}_{li}^k \bar{\Gamma}_{jkm} + 2\bar{\Gamma}_{lj}^k \bar{\Gamma}_{ikm} + \bar{\Gamma}_{im}^k \bar{\Gamma}_{klj}\right). \tag{1.4.2.13}$$

Taking the time derivative of equation (1.4.2.12) and using equation (1.4.2.10), one arrives at

$$\partial_t \bar{\Gamma}^i = -\partial_j \left(2\alpha \bar{A}^{ij} - 2\bar{\gamma}^{mj} \partial_m \beta^i - 2\bar{\gamma}^{mi} \partial_m \beta^j + \frac{2}{3} \bar{\gamma}^{ij} \partial_l \beta^l + \beta^l \partial_l \bar{\gamma}^{ij} \right). \qquad (1.4.2.14)$$

Taking this along with equation (1.4.1.31) gives

$$\partial_t \bar{\Gamma}^i = -2\bar{A}^{ij} \partial_j \alpha + 2\alpha \left(\bar{\Gamma}^i_{jk} \bar{A}^{kj} - \frac{2}{3} \bar{\gamma}^{ij} \partial_j K - 8\pi \bar{\gamma}^{ij} S_j + 6\bar{A}^{ij} \Phi \right) + \beta^j \partial_j \bar{\Gamma}^i - \bar{\Gamma}^i \partial_j \beta^j$$
$$+ \frac{2}{3} \bar{\Gamma}^i \partial_j \beta^j + \frac{1}{3} \bar{\gamma}^{li} \partial_l \partial_j \beta^j + \bar{\gamma}^{lj} \partial_j \partial_l \beta^i. \qquad (1.4.2.15)$$

With this, the evolution equations for the BSSN formulation are

$$\partial_t \Phi = -\frac{1}{6} \alpha K + \frac{1}{6} \beta^i \partial_i \Phi + \frac{1}{6} \partial_i \beta^i,$$

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

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

$$\partial_t \bar{A}_{ij} = e^{-4\Phi} \left(-\left(D_i D_j \alpha \right)^{TF} + \alpha \left(R_{ij}^{TF} - 8\pi S_{ij}^{TF} \right) \right) + \alpha \left(K \bar{A}_{ij} - 2\bar{A}_{il} \bar{A}^l_j \right) + \beta^k \partial_k \bar{A}_{ij}$$
$$+ \bar{A}_{ik} \partial_j \beta^k + \bar{A}_{kj} \partial_i \beta^k - \frac{2}{3} \bar{A}_{ij} \partial_k \beta^k,$$

$$\partial_t \bar{\Gamma}^i = -2\bar{A}^{ij} \partial_j \alpha + 2\alpha \left(\bar{\Gamma}^i_{jk} \bar{A}^{kj} - \frac{2}{3} \bar{\gamma}^{ij} \partial_j K - 8\pi \bar{\gamma}^{ij} S_j + 6\bar{A}^{ij} \Phi \right) + \beta^j \partial_j \bar{\Gamma}^i - \bar{\Gamma}^i \partial_j \beta^j$$
$$+ \frac{2}{3} \bar{\Gamma}^i \partial_j \beta^j + \frac{1}{3} \bar{\gamma}^{li} \partial_l \partial_j \beta^j + \bar{\gamma}^{lj} \partial_j \partial_l \beta^i.$$

These equations are much more stable than the regular ADM equations and these form the basis of most numerical simulations of general relativity.

1.5. Neutron Stars

Neutron stars are remnants of stars that have exhausted their fuel supply and have reached the end of their lives. Neutron stars are among the densest objects in the universe with mass on the order of 1 $M_{\odot}$ and a size on the order of 10 km (Lattimer & Prakash 2004). These objects are formed when a star of moderate size stops fusing material and gravitation is left unmatched and the object collapses down until the degenerate pressure of neutrons is able to halt the collapse. These objects are massive enough and compact enough that when they are orbiting in tight orbits around each other, they should create gravitational waves that are able to be detected. Gravitational waves from BNS were confirmed to exist with the detection of GW170817A (Pian 2021). To model the gravitational waves from an event like this, a relativistic simulation must be used, but before that can be done it is necessary to review some properties of neutron stars.

1.5.1. Equations of State

Along with the TOV equations from section 1.3.6, the EOS is all that is needed to describe the equilibrium of matter inside a neutron star (Oppenheimer & Volkoff 1939). For an EOS to be useful, the number of parameters in the equation needs to be smaller than the number of properties that are related to the EOS, but large enough to provide an accurate approximation to the body as a whole. The best type of EOS that fits these criteria is a piecewise polytrope. Since the temperature of a neutron star is below the Fermi temperature of neutrons, the star can be treated as a cold body (Read et al. 2009a). The EOS will take the form

$$P(\rho) = K\rho^{\Gamma}, \tag{1.5.1.1}$$

with P being the pressure, K being a constant of proportionality, ρ being the density, and Γ being the adiabatic index. For this thesis, the overall EOS will be a piecewise polytrope made up of seven pieces, four for the crust of the neutron star, and three for the core (Read et al. 2009a). An example of values used in the EOS is given in section 2.1 in the included par_eos1.d file. Additionally, a graph of the piecewise polytropic EOS can be seen below in figure 1.

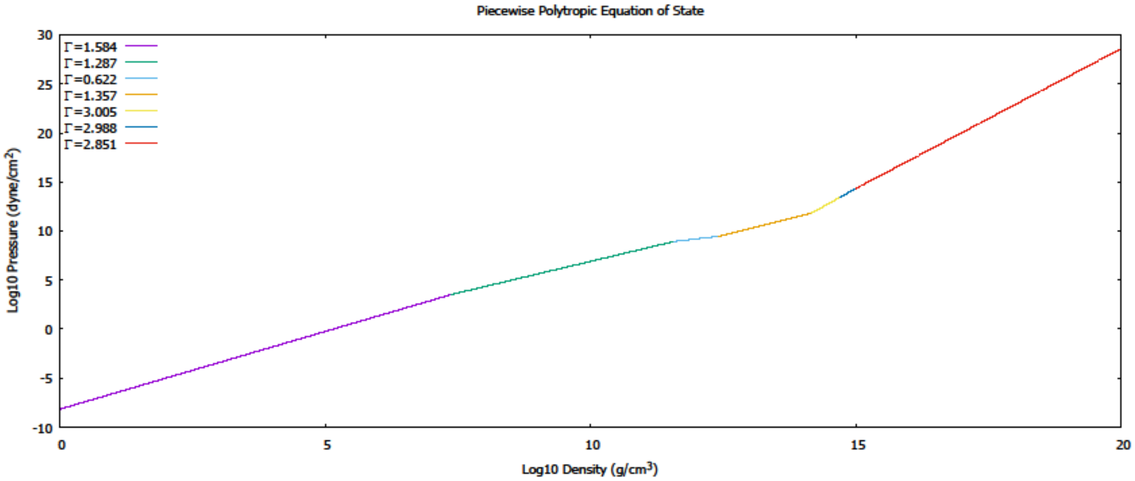


Figure 1: Piecewise polytropic equation of state with seven pieces used in the cases studied.

1.5.2. Bounds of Mass

As stated above, the mass of neutron stars is on the order of 1 $M_\odot$ with the lower bound on the mass being the Chandrasekhar limit of 1.40 $M_\odot$ (Mazzali et al. 2007) and the highest observed mass being 2.08 $M_\odot$ (Fonseca et al. 2021). These are the lower and upper bounds of the neutron stars used in this thesis, as well as an additional mass of 1.74 $M_\odot$ used as an intermediate mass.

1.5.3. Binary Neutron Stars

Neutron stars have two primary methods to form binaries like those studied for this thesis. The first is explored in Renzo et al. (2019) and consists of binary neutron stars (BNS) that formed from massive stars that have exhausted their fuel and have left behind the neutron star remnants that remained locked in a binary throughout this process. High mass stars are likely to be found in binary systems (Sana et al. 2012), so it would be natural to assume that a large fraction of these systems evolve into BNS.

Another possibility is explored in Ye et al. (2019), which looked at the formation of neutron stars in globular clusters where one star is ejected during the first supernova blast and a binary pair is formed later via tidal capture. This process of tidal capture of neutron stars that formed in globular clusters is explored in Lee et al. (2010). Although the process by which BNS form is not completely understood, for the purpose of this thesis it is more important to understand how they evolve and eventually merge.

1.6. Lorene

The Lorene software package, which stands for Langage Object pour la RElativité NumériquE, is a set of C++ classes that are used to solve various problems in numerical relativity. This program is used to create the initial conditions of BNS that is then inserted into the Einstein Toolkit for further evolution. Lorene is a system that provides tools to solve partial differential equations (PDEs) though multi-domain spectral methods (Grandclément & Novak 2009). To understand how Lorene can solve PDEs via spectral methods, it is first important to go over spectral methods.

1.6.1. Spectral Methods

To get a basic understanding of spectral methods, consider a function f that lies within the domain $\mathbb{D}$. The most straight forward approximation of the function's derivative is through finite-difference methods, where first a grid is defined with

$$\{x_i\}_{i=0,1,\dots,N} \subset \mathbb{D}, \tag{1.6.1.1}$$

that is composed of $N + 1$ points in an interval and the function f is defined by its $N + 1$ values on the grid points

$$\{f_i = f(x_i)\}_{i=0,1,\dots,N}. \tag{1.6.1.2}$$

With this way of defining the function f, the derivative of the function is approximated by

$$f_i' = f'(x_i) \simeq \frac{f_{i+1} - f_i}{x_{i+1} - x_i}. \tag{1.6.1.3}$$

This is the method of finite-differences. An alternative way of approximating a solution to a PDE is through spectral methods. In this method the function f is not represented by its values on a finite number of grid points, but instead it is defined by using coefficients $\{c_i\}_{i=0,1,\dots,N}$ in a basis of known functions $\{\Phi_i\}_{i=0,1,\dots,N}$. The function can be approximated by

$$f(x) \simeq \sum_{i=0}^{N} c_i \Phi_i(x). \tag{1.6.1.4}$$

For spectral methods, the trial functions are globally smooth over the entire domain $\mathbb{D}$. Different choices of test functions can be used within spectral methods with the most common being truncated Fourier series, spherical harmonics, or orthogonal families of polynomials.

1.6.2. Compact Binaries

For the way that Lorene handles compact binaries, in this case BNS, a few approximations need to be made. First, systems containing compact binaries are known to emit gravitational waves, so no closed orbits can exist, and objects will follow spiral-like trajectories. However, in the regime where objects are still relatively far apart, the real trajectory can be approximated with a series of closed orbits. To avoid any diverging quantities, an additional demand must be put into place – the spatial metric must be conformally flat. However, even in the case of a single rotating black hole this condition is not true, but comparisons with post-Newtonian results or non-conformally-flat results show this approximation is relatively good (Grandclément & Novak 2009). However, the conformally flat assumption discards all reference to gravitational waves, so this assumption cannot be used during evolution, only during the setup of initial conditions.

With these conditions in place Einstein's equations reduce to a set of five elliptic equations for the lapse, the conformal factor, and the shift vector, with the motion of the fluid being described by another elliptic function for the potential of the flow. These combined with an EOS gives the evolutionary sequence.

In the case of a system with two neutron stars, two sets of domains are used, each centered on one of the stars. Each set consists of sphere-like domains that extend from the surface of the star to infinity, and the functions are expanded in terms of spherical harmonics with respect to the angles (θ, ϕ) and Chebyshev polynomials with respect to the radial coordinates. For a more in-depth description of the iterative process to generate data used in Lorene, refer to Gourgoulhon et al. (2001).

Additionally, in the case of a compact binary it is useful to split some of the quantities that are of interest to create two local versions that are centered on each body and can then be recombined to create a global quantity (Grandclément et al. 2002; Löffler et al. 2012). Some of these quantities that will be useful at a later point are

$$\alpha = 1 + \alpha_{(1)} + \alpha_{(2)}, \tag{1.6.2.1}$$

$$\phi = 1 + \phi_{(1)} + \phi_{(2)}, \tag{1.6.2.2}$$

$$\beta^i = \beta^i_{(1)} + \beta^i_{(2)}, \tag{1.6.2.3}$$

$$K^{ij} = K^{ij}_{(1)} + K^{ij}_{(2)}, \tag{1.6.2.4}$$

with the subscript (1) denoting the quantity around the first star and (2) denoting the quantity around the second star.

1.7. Einstein Toolkit

The Einstein Toolkit (Löffler et al. 2012) is a community driven program used to perform relativistic simulations that are too complex for an analytical solution to exist. Simulations that are ran within the Einstein Toolkit are based on the ADM and BSSN formulations of numerical relativity (Löffler et al. 2012).

1.7.1. Cactus

The Einstein Toolkit is built upon the Cactus framework (Löffler et al. 2012), which means that it is built to be completely modular, giving the user the ability to customize how much of the program is used in each simulation. The Cactus framework consists of two main parts, the ‘flesh’ which provides the infrastructure to build simulations out of independently developed modules and to facilitate communication between these modules. In this framework the modules are referred to as ‘thorns’ and make up the bulk of any simulation.

1.7.2. Thorns

In addition to the ‘flesh’ of the Cactus framework, built on top of it are a variety of thorns (Allen et al.). There are many thorns that the Einstein Toolkit relies upon and many more that add a variety of physics to the numerical system being simulated. This section focuses on a few of the thorns that are important for the binary neutron star simulations that were explored for this thesis.

1.7.2.1. ADMBase

The thorn `ADMBase` provides the basic ADM variables that are used in the 3+1 formalism that the Einstein Toolkit is familiar with. These basic variables consist of:

Quantity	Symbol	Variable Name
3-metric tensor	g_{ij}	gxx, gxy, gxz, gyy, gyz, gzz
Extrinsic curvature tensor	K_{ij}	kxx, kxy, kxz, kyy, kyz, kzz
Lapse function	α	alp
Shift vector	β^i	betax, betay, betaz

Table 1: ADMBase defined variables

This thorn provides the core infrastructure for thorns implementing general relativity on a 3D grid in the 3+1 formalism. These variables are used to communicate between thorns providing the initial data, evolution methods, and analysis routines for the 3+1 formalism. Additionally, the variables can be used as a mechanism to interact with alternative formalisms, as long as the routines can be written to transform the alternative variables into these 3+1 variables (Goodale).

1.7.2.2. ADMCoupling

The thorn `ADMCoupling` allows for seamless coupling of evolution and analysis thorns to any thorns which contribute matter terms to the stress-energy tensor, $T_{\mu\nu}$. By making a spacetime thorn compatible with this thorn, it can know about the variables in all matter thorns that are also compatible with `ADMCoupling`. This avoids explicit dependencies between the spacetime and matter evolution thorns (Hawke & Rideout).

1.7.2.3. AHFinderDirect

The thorn `AHFinderDirec`t locates apparent horizons in a numerically computed slice using a direct method. This direct method uses the fact that in terms of the usual 3+1 variables, an apparent horizon satisfies the equation

$$\theta \equiv \mathrm{D}_i n^i + K_{ij} n^i n^j - K = 0, \qquad (1.7.2.3.1)$$

where n^i is the outward-pointing unit normal to the apparent horizon, and D_i is the covariant derivative associated with the 3-metric γ_{ij} in the slice. As such, this thorn requires the usual Cactus 3-metric, γ_{ij}, and the extrinsic curvature tensor, K_{ij}. However, there may be several such surfaces, some nested inside others. In that case the apparent horizon is an outermost marginally trapped surface (Thornburg).

1.7.2.4. Carpet

`Carpet` is a fixed mesh refinement (FMR) driver for Cactus. FMR, also known as box-in-box, is a way to increase the local resolution in unigrid applications, while retaining the unigrid character of an application. Several grids of varying sizes are overlaid upon each other, where the coarsest grid has the largest extent. The way this works is by creating a coarse grid that encloses the entire domain with successively finer grids which overlay the coarse grid at the locations where a higher resolution is needed. The coarser grids then provide the boundary conditions to the finer grids.

This allows the application to benefit from the higher resolution of the smaller grids where more interesting physics is taking place, while keeping the outer boundary far out at the same time. The biggest advantage of FMR is that it needs far fewer resources than globally increasing the resolution. To increase the resolution in a unigrid application by a factor of two requires a factor of eight more storage in three dimensions, so with this mesh refinement system there is the possibility of finer resolution without the full cost of globally increasing the resolution (Schnetter).

1.7.2.5. EOS_Omni

The `EOS_Omni` thorn provides a unified EOS and implements multiple analytic EOSs. The EOSs that are used in this thorn are the polytropic EOS, the gamma-law EOS, and a hybrid EOS consisting of a n-piece piecewise-polytrope with a thermal gamma-law component. Additionally, the `EOS_Omni` assumes nuclear statical equilibrium with rest-mass density ρ, a specific internal energy ϵ, and electron fraction Y_e being the independent variables (Ott & Schnetter 2013).

1.7.2.6. GRHydro

The `GRHydro` thorn is a general-relativistic three-dimensional hydrodynamics code. This thorn uses the hydro variables defined in `HydroBase` and provides its own "conserved" hydro variables and methods to solve them. However, it does not provide any information about initial a data or the equations of state, so other thorns are needed.

The equations of general relativistic hydrodynamics can be written in the flux conservative form

$$\partial_t q + \partial_i f^i(q) = s(q), \tag{1.7.2.6.1}$$

where q is a set of conserved variables, $f^i(q)$ are the fluxes, and $s(q)$ is the source term. The eight conserved variables are labeled as: D, the generalized particle number density; S_i, the generalized momenta in each direction; τ, an internal energy term; and $\mathcal{B}^k$, the densitized magnetic field. These conserved variables are composed from a set of primitive variables: ρ, the rest-mass density; P, the pressure; v^i, the fluid 3-velocities; ϵ, the specific internal energy; B^k, the magnetic field in the lab frame; and W, the Lorentz factor. The conserved variables are related to the primitive variables through

$$D = \sqrt{\gamma} W \rho, \tag{1.7.2.6.2}$$

$$S_i = \sqrt{\gamma}\left(\rho h * W^2 v_j - \alpha b^0 b_j\right), \tag{1.7.2.6.3}$$

$$\tau = \sqrt{\gamma}\left(\rho h * W^2 - P * -(\alpha b^0)\right) - D, \tag{1.7.2.6.4}$$

$$\mathcal{B}^k = \sqrt{\gamma} B^k, \tag{1.7.2.6.5}$$

where γ is the determinant of the spatial 3-metric γ_{ij}, $h * \equiv 1 + \epsilon + \frac{(P+b^2)}{\rho}$, $P * \equiv P + \frac{b^2}{2}$, and b^μ is the magnetic field in the fluid's rest frame and is related to B^k by

$$b^0 = \frac{WB^k v_k}{\alpha}, \tag{1.7.2.6.6}$$

$$b^i = \frac{B^i}{W} + w(B^k v_k)\left(v^i - \frac{\beta^i}{\alpha}\right), \tag{1.7.2.6.7}$$

$$b^2 = \frac{B^i B_i}{W^2} + \left(B^i v_i\right)^2. \tag{1.7.2.6.8}$$

However, only five of the primitive fluid variables are independent. This is because the Lorentz factor is defined in terms of the velocities and the metric as $W = \left(1 - \gamma_{ij} v^i v^j\right)^{-1/2}$, and the pressure, rest-mass density, and specific internal energy are related through the EOS.

The fluxes are usually defined in terms of both the conserved variables and the primitive variables:

$$F^i(U) = \begin{cases} D\left(\alpha v^i - \beta^i\right) \\ S_j\left(\alpha v^i - \beta^i\right) + P\delta^i_j \\ \tau\left(\alpha v^i - \beta^i\right) + Pv^i \\ \mathcal{B}^k\left(\alpha v^i - \beta^i\right) - \mathcal{B}^i(\alpha v^k - \beta^k) \end{cases}. \tag{1.7.2.6.9}$$

The source terms are

$$s(U) = \begin{cases} 0 \\ T^{\mu\nu}\left(\partial_\mu g_{\nu j} + \Gamma^\delta_{\mu\nu} g_{\delta j}\right) \\ \alpha\left(T^{\mu 0}\partial_\mu \ln\alpha - T^{\mu\nu}\Gamma^0_{\nu\mu}\right) \\ 0 \end{cases}, \tag{1.7.2.6.10}$$

and the stress-energy tensor is given by

$$T^{\mu\nu} = (\rho + \rho\epsilon + P + b^2)u^\mu u^\nu + \left(P + \frac{b^2}{2}\right)g^{\mu\nu} - b^\mu b^\nu. \tag{1.7.2.6.11}$$

With these equations in place, the Einstein Toolkit is able to handle these continuity equations and able to make sure that the system will remain physical for the duration of the simulation (Baiotti et al. 2007; Mösta et al. 2014).

1.7.2.7. Hydro_Analysis

The `Hydro_Analysis` thorn is responsible for providing the basic hydro analysis routines and quantities that are used during the simulation (Löffler 2022).

1.7.2.8. HydroBase

The `HydroBase` thorn extends the Cactus framework to include an interface for magnetohydrodynamics to work within. This thorn's main function is to store the primitive variables, common among hydrodynamic simulations, commonly needed parameters and schedule groups for the main functions of a hydrodynamics code. The prime variables defined in this thorn are:

Quantity	Symbol	Variable Name	Definition
Rest mass density	ρ	rho	
Pressure	P	press	
Specific internal energy	ϵ	eps	
Contravariant fluid three velocity	v^i	vel[3]	$v^i = \frac{u^i}{\alpha u^0} + \frac{\beta^i}{\alpha}$
Electron fraction	Y_e	Y_e	
Temperature	T	temperature	

Specific Entropy per particle	s	entropy	
Contravariant magnetic field vector	B^i	Bvec[3]	$B^i = \frac{1}{\sqrt{4\pi}} n_\nu F^{*\nu i}$

Table 2: HydroBase defined variables

Where the dual to the Faraday tensor is, $F^{*\mu\nu} = \frac{1}{2}\varepsilon^{\mu\nu\alpha\beta}F_{\alpha\beta}$. These variables are then used in any other thorn that is needed to call them (Bode & Löffler 2010).

1.7.2.9. Meudon_Bin_NS

The thorn `Meudon_Bin_NS` is responsible for reading in the initial conditions from Lorene. The initial conditions represent solutions to the equations (Löffler et al. 2012)

$$\bar{\mathrm{D}}^2\nu_{(m)} = 4\pi\phi^4\left(E_{(m)} + S_{(m)}\right) + \phi^4 K_{ij}K^{ij}_{(m)} - \bar{\mathrm{D}}_i\nu_{(m)}\bar{\mathrm{D}}^i\beta, \tag{1.7.2.9.1}$$

$$\bar{\mathrm{D}}^2\beta_{(m)} = 4\pi\phi^4 S_{(m)} + \frac{3}{4}\phi^4 K_{ij}K^{ij}_{(m)} - \frac{1}{2}\left(\bar{\mathrm{D}}_i\nu_{(m)}\bar{\mathrm{D}}^i\nu + \bar{\mathrm{D}}_i\beta_{(m)}\bar{\mathrm{D}}^i\beta\right), \tag{1.7.2.9.2}$$

$$\bar{\mathrm{D}}^2\beta^i_{(m)} + \frac{1}{3}\bar{\mathrm{D}}^i\bar{\mathrm{D}}_j\beta^j_{(m)} = -16\pi\alpha\phi^4\left(E_{(m)} + P_{(m)}\right)v^i_{(m)} + 2\alpha\phi^4 K^{ij}_{(m)}\bar{\mathrm{D}}_j(3\beta - 4\nu), \tag{1.7.2.9.3}$$

with $\bar{\mathrm{D}}^2 = \bar{\mathrm{D}}^i\bar{\mathrm{D}}_i$ being the Laplacian with respect to the conformal metric, $v^i_{(m)}$ the special components of the 4-velocity of star m, and ν and β being defined by

$$\nu \equiv \log\alpha, \qquad \beta \equiv \ln\alpha\phi^2.$$

Equations (1.7.2.9.1) and (1.7.2.9.2) are from the trace of the spatial part of the Einstein equations combined with the Hamiltonian constraint, and equation (1.7.2.9.3) is from the trace of the Einstein equations with the momentum constraint (Gourgoulhon et al. 2001).

1.7.2.10. NaNChecker

The `NaNChecker` thorn is used to analyze Cactus grid variables of real or complex data type for Not-a-Number (NaN) and infinite values. If this thorn finds a NaN, the actions it can perform are to display a warning about where and how many NaN’s were found, to gracefully terminate the simulation, or to immediately terminate Cactus (Radke).

1.7.2.11. NSTracker

The thorn `NSTracker` is responsible for tracking the neutron stars during the simulation and shifting the grid so that the finest resolution of the grid will move along with the neutron stars to maximize the resolution where the material from the binary is located (Löffler et al. 2012).

1.7.2.12. TmunuBase

The thorn `TmunuBase` provides core infrastructure for thorns implementing some kind of energy or matter in general relativity. This thorn provides the basic variables in the stress-energy tensor, $T_{\mu\nu}$, to be communicated between thorns contributing to the stress-energy content of the spacetime as well as thorns needed to evaluate the stress-energy tensor as the spacetime evolves. The variables defined in the 3+1 formulism are:

Quantity	Symbol	Variable Name

“scalar” time-time component	T_{00}	eTtt
“vector” time-space components	T_{0i}	eTtx, eTty, eTtz
“tensor” space-space components	T_{ij}	eTxx, eTxy, eTxz, eTyy, eTyz, eTzz

Table 3: TmunuBase defined variables

These components have a prefix e to avoid naming conflicts with existing variables. These variables are then able to be used in any thorn that calls them during the simulation (Schnetter 2007).

1.7.2.13. VolumeIntergral_GRMHD

The thorn `VolumeIntergal_GRMHD` allows for integration of spacetime quantities. To do this the authors of the thorn start by defining the “densitised density” which is given by

$$\rho_* \equiv \alpha\sqrt{\gamma}\rho_b u^0 = W\rho_b\sqrt{\gamma}, \qquad (1.7.2.17.1)$$

where α is the lapse function, γ is the determinant of the physical spatial metric γ_{ij}, ρ_b is the baryonic density, u^μ is the fluid four-velocity, and $W \equiv \alpha u^0$ is the Lorentz factor. The rest mass integral is then given by

$$M_0 = \int dV\rho_* = \int dV\left(W\rho_b\sqrt{\gamma}\right). \qquad (1.7.2.17.2)$$

Similar integrals are defined in similar ways (Etieene & Werneck 2021).

1.7.2.14. WeylScal4

The thorn `WeylScal4` uses the methods described in Newman & Renrose (1962) to calculate the portion of the Weyl tensor that is associated with gravitational waves and outputs that into a plottable ASCII file (Löffler et al. 2012).

2. Process

2.1. Lorene

For Lorene to find a binary neutron star configuration, a few things must first be specified. These being (Gourgoulhon et al. 2001):

1. The EOS of each neutron star.
2. The rotation state, either being rigidly rotating or irrotational flow.
3. The distance between the stellar cores.
4. The central enthalpy of each star.

Once these parameters have been set, Lorene is able to create the initial data to be used in the Einstein Toolkit. An example of the EOS used in Lorene follows.

```
110    Type of the EOS (cf. documentation of
Eos::eos_from_file)

Star 1  EOS

7                   number of polytropes

1.58425             array of adiabatic index

1.28733

0.62223

1.35692

3.005

2.988

2.851

6.8011e-09          kappa value for the lowest density
                    region i.e., the crust of the
                    neutron star

3.53623             log10 pressure along first
                    boundary, in natural units
```

```
7.3875              array of the exponent of fiducial
                    densities logRho
11.5779
12.4196
14.165
14.7
15
0.                      array of percentage
0.
0.
0.
0.
0.
```

The number of polytropes used within the piecewise function is specified along with the adiabatic index for each piece, the constant of proportionality at the crust of the neutron star, the pressure at the boundary between piece one and two, and an array of fiducial densities for each piece. Initial data in this form is needed for each of the neutron stars.

In each case the neutron star was said to be irrotational, this assumption is made because although neutron stars rotate rapidly, they can be assumed to not rotate compared to the time span of the simulation. For each case the distance was taken to be 40 kilometers, because this was assumed to be far enough to give a good understanding of the system but close enough to conform to the computational limitations presented. The central enthalpy, calculated using Lorene to find a desired baryonic mass, is given in Table 4.

Baryonic Mass ($M_{\odot}$)	Central Enthalpy (Log base 10)

1.40	0.1246
1.74	0.1894
2.08	0.3081

Table 4: Central Enthalpy Log10 based on baryonic mass

2.2. Parameter File

To get a clear understanding of the Einstein Toolkit and how it takes the input from Lorene and the parameter file, and creates an output, it will be useful to take the example of the parameter file for case 1, which consists of two 2.08 solar mass BNS, and go through the parameter file. The complete parameter file is included in Appendix B.

2.2.1. Cactus Parameters

The first section of the parameter file are the Cactus parameters. These contain basic information about the simulation.

```
#------------------------------------------------------
# Cactus parameters:
#------------------------------------------------------
Cactus::cctk_run_title     = "May23-
MagneticFieldVolumeCase1"
Cactus::cctk_full_warnings = "yes"
Cactus::highlight_warning_messages = "no"

Cactus::terminate       = "time"
Cactus::cctk_final_time = 2000.0
```

This section tells Einstein Toolkit what the name of the run is, in this case it is called “May23-MagneticFieldVolumeCase1”, as well as telling the Einstein Toolkit how long the simulation should run, in this case it was told to run for 2000 hours. Although it did not run for that whole time, a time larger than the expected run time was selected, and the simulation was stopped manually.

2.2.2. Active Thorns

Next, the Einstein Toolkit needs to know which thorns are available at its disposal. That is taken care of in the Active Thorns section:

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


ActiveThorns = "Boundary CartGrid3D CoordBase Fortran
InitBase IOUtil LocalReduce SymBase Time"

ActiveThorns = "AEILocalInterp"

ActiveThorns = "MoL Slab SpaceMask SphericalSurface"

ActiveThorns = "Carpet CarpetInterp CarpetInterp2
CarpetIOASCII CarpetIOHDF5 CarpetIOScalar CarpetLib
CarpetIOBasic CarpetReduce CarpetRegrid2 CarpetSlab
CarpetTracker CarpetMask LoopControl"

ActiveThorns = "Formaline"

ActiveThorns = "NaNChecker TerminationTrigger
TimerReport"

ActiveThorns = "ADMbase ADMcoupling ADMmacros
CoordGauge StaticConformal"

ActiveThorns = "RotatingSymmetry180
ReflectionSymmetry"

ActiveThorns = "Constants TmunuBase HydroBase"

ActiveThorns = "QuasiLocalMeasures"

ActiveThorns = "EOS_Omni"

ActiveThorns = "GRHydro"

ActiveThorns = "SummationByParts"
ActiveThorns = "GenericFD NewRad"
ActiveThorns = "ML_BSSN ML_BSSN_Helper
ML_ADMConstraints"
ActiveThorns = "Hydro_Analysis NSTracker"
ActiveThorns = "Dissipation"
ActiveThorns = "SystemStatistics SystemTopology"
ActiveThorns = "VolumeIntegrals_GRMHD"
# Wave extraction (Psi4)
ActiveThorns = "WeylScal4 Multipole"
ActiveThorns = "AHFinderDirect"
```

This section is responsible for telling the Einstein Toolkit which thorns should be used by writing all the active thorns in a list to let the Einstein Toolkit know what thorns are available. Since some of the more important thorns have been covered above, thorns that need more detail are covered further below.

2.2.3. Diagnostic Parameters

The next section of the parameter file is the Diagnostic Parameters. This section deals with simulation output during run-time such as logging by the timer indicating how long the simulation has been running and communicating with thorns to output commands that are currently executing.

```
#-----------------------------------------------------
# Diagnostic parameters:
#-----------------------------------------------------

Carpet::output_timers_every = 0
Carpet::storage_verbose   = "no"
```

```
Carpet::verbose           = "no"
Carpet::veryverbose       = "no"
Carpet::grid_structure_filename   = "carpet-grid-
structure"
Carpet::grid_coordinates_filename = "carpet-grid-
coordinates"


CarpetLib::output_bboxes  = "no"


CarpetMask::verbose    = "no"
CarpetReduce::verbose  = "no"
CarpetRegrid2::verbose = "no"
CarpetRegrid2::veryverbose    = "no"
CarpetTracker::verbose = "no"



TimerReport::out_every    = 4096
TimerReport::out_filename = "TimerReport"
TimerReport::output_all_timers          = "yes"
TimerReport::output_all_timers_together = "yes"
TimerReport::output_all_timers_readable = "yes"
TimerReport::n_top_timers               = 40



QuasiLocalMeasures::verbose   = "no"
SphericalSurface::verbose   = "no"
```

As seen above, many of the quantities are set to ‘verbose= “no”’ which tells the Einstein Toolkit not to print the command that it is executing.

2.2.4. Utility Parameters

The utility parameter is set up to allow for NaNChecker to check different variables for imaginary or infinite values and stopping the simulation if any are found.

```
#-----------------------------------------------------
# Utility parameters:
#-----------------------------------------------------

NaNChecker::check_every    =  128 # twice for
every_coarse
NaNChecker::check_vars = "
            ADMBase::curv
            ADMBase::metric
            ADMBase::lapse
            ADMBase::shift
            HydroBase::rho
            HydroBase::eps
            HydroBase::press
            HydroBase::vel
"
NaNChecker::action_if_found   =  "terminate"
```

As can be seen in the parameter file above, NaNChecker will check variables defined in the ADMBase and the HydroBase thorns every 128 iterations for imaginary or infinite values. The setting is to terminate instead of ‘just warn’ or ‘abort’.

2.2.5. Run Parameters

The next section of the parameter file is dedicated to the run parameters which are made up of the grid, the model, the EOS, as well as the numerics and evolution sections.

2.2.5.1. Grid

In the grid section, the size and refinement of the grid is specified along with symmetries that the system may use to simplify the calculations, as well as the regrid size that is used around the neutron stars.

```
#------
# Grid:
#------

MoL::ODE_Method             = "rk4"
MoL::MoL_Intermediate_Steps = 4
MoL::MoL_Num_Scratch_Levels = 1
# use dt = 0.4 dx (works for core collapse)
Time::dtfac = 0.35

CoordBase::domainsize = "minmax"
CoordBase::xmin =    0.00
CoordBase::ymin = -960.00
CoordBase::zmin =    0.00
CoordBase::xmax = +960.00
CoordBase::ymax = +960.00
CoordBase::zmax = +960.00
CoordBase::dx   =  10.00
CoordBase::dy   =  10.00
CoordBase::dz   =  10.00
```

```
CoordBase::boundary_size_x_lower     = 3
CoordBase::boundary_size_y_lower     = 3
CoordBase::boundary_size_z_lower     = 3
CoordBase::boundary_size_x_upper     = 3
CoordBase::boundary_size_y_upper     = 3
CoordBase::boundary_size_z_upper     = 3

CoordBase::boundary_shiftout_x_lower = 1
CoordBase::boundary_shiftout_y_lower = 0
CoordBase::boundary_shiftout_z_lower = 1

reflectionsymmetry::avoid_origin_x       = no
reflectionsymmetry::avoid_origin_y       = no
reflectionsymmetry::avoid_origin_z       = no
reflectionsymmetry::reflection_x         = no
reflectionsymmetry::reflection_y         = no
reflectionsymmetry::reflection_z         = yes

CartGrid3D::type = "coordbase"
Carpet::domain_from_coordbase = "yes"

Driver::ghost_size                     = 3

# General Carpet parameters:
Carpet::enable_all_storage       = "no"
Carpet::use_buffer_zones         = "yes"
Carpet::schedule_barriers        = "no"

Carpet::poison_new_timelevels    = "yes"
```

```
Carpet::check_for_poison         = "no"

Carpet::init_3_timelevels        = "no"
Carpet::init_fill_timelevels     = "yes"

CarpetLib::poison_new_memory         = "yes"
CarpetLib::poison_value              = 114
CarpetLib::check_bboxes              = "no"
CarpetLib::interleave_communications = "yes"
CarpetLib::combine_sends             = "yes"

CarpetInterp::tree_search = "yes"
CarpetInterp::check_tree_search = "no"

CarpetRegrid2::freeze_unaligned_levels = "yes"
CarpetRegrid2::freeze_unaligned_parent_levels = "yes"
CarpetRegrid2::ensure_proper_nesting   = "yes"
CarpetRegrid2::snap_to_coarse          = "yes"
CarpetRegrid2::symmetry_rotating180    = "yes"

# System specific Carpet parameters:
Carpet::max_refinement_levels    = 8
Carpet::prolongation_order_space = 5
Carpet::prolongation_order_time  = 2

Carpet::refinement_centering     = "vertex"

CarpetRegrid2::regrid_every = 64 # as often as
required
CarpetRegrid2::num_centres  = 3
```

```
CarpetRegrid2::min_distance = 0

CarpetRegrid2::num_levels_1 = 7
CarpetRegrid2::position_x_1 = 0
CarpetRegrid2::radius_1[1]  = 960
CarpetRegrid2::radius_1[2]  = 228
CarpetRegrid2::radius_1[3]  = 114
CarpetRegrid2::radius_1[4]  = 66
CarpetRegrid2::radius_x_1[5]  = 35
carpetregrid2::radius_y_1[5]  = 35
carpetregrid2::radius_z_1[5]  = 24
CarpetRegrid2::radius_1[6]  = 13
CarpetRegrid2::radius_1[7]  = 6.5

CarpetRegrid2::num_levels_2 = 7
CarpetRegrid2::position_x_2 = -15
CarpetRegrid2::radius_2[1]  = 320
CarpetRegrid2::radius_2[2]  = 164
CarpetRegrid2::radius_2[3]  = 96
CarpetRegrid2::radius_2[4]  = 48
CarpetRegrid2::radius_2[5]  = 18
CarpetRegrid2::radius_2[6]  = 11
CarpetRegrid2::radius_2[7]  = 5.5

CarpetRegrid2::num_levels_3 = 7
CarpetRegrid2::radius_3[1]  = 320
CarpetRegrid2::radius_3[2]  = 164
CarpetRegrid2::radius_3[3]  = 96
CarpetRegrid2::radius_3[4]  = 48
CarpetRegrid2::radius_3[5]  = 18
```

```
CarpetRegrid2::radius_3[6]  = 11
CarpetRegrid2::radius_3[7]  =  5.5

carpetmask::excluded_surface[0]          = 0
carpetmask::excluded_surface[1]          = 1
carpetmask::excluded_surface_factor[0]   = 1
carpetmask::excluded_surface_factor[1]   = 1

CarpetTracker::surface_name[0] = "Righthand NS"
CarpetTracker::surface_name[1] = "Lefthand NS"
```

As seen above in the parameter file, the size of the grid used in the simulations is 960 by 960 in the x and z directions and from −960 to 960 in the y-direction. The first level of refinement is 10 in each direction and progressively gets finer closer to the neutron star. Also specified in this section is the symmetries that are used like reflection symmetry about the x and y-axes.

2.2.5.2. Model

This section of the parameter file is dedicated to the loading of the initial conditions from Lorene using the thorn `Meudon_Bin_NS`.

```
#------
# MODEL:
#------

ActiveThorns = "Meudon_Bin_NS"
HydroBase::initial_hydro         = "Meudon_Bin_NS"
ADMBase::initial_data            = "Meudon_Bin_NS"
```

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

Meudon_Bin_NS::filename ="case208208.d"
```

As can be seen above, the initial data is taken from the `Meudon_Bin_NS` thorn, which gets the data from `"case208208.d"`, which is the initial data that is generated by Loren for case1.

2.2.5.3. Equation of State

In this section of the parameter file, the EOS is specified again to model how the material evolves in each star throughout the run-time of the simulation.

```
#------------------EOS-----------------------------

EOS_Omni::poly_K = 123.613314525753

EOS_Omni::hybrid_gamma_th        = 1.8 #gamma thermal
from Hotokezaka et al. 2013 (arxiv.org/abs/1307.5888)

EOS_Omni::n_pieces               = 7 #3 for the core +
4 for the crust (Read et al 2009)

##k0=6.8011e-09 in cgs units and Kcu =
k0_cgs*(cu_to_cgs**(gamma-1)) where
cu_to_cgs=6.1762691458861658e+17 converts from CU to
g/cm^3

EOS_Omni::hybrid_k0              = 168.58190246577206
```

```
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_rho[0]         = 3.95160737e-11
EOS_Omni::hybrid_rho[1]         = 6.12595478e-07
EOS_Omni::hybrid_rho[2]         = 4.25474745e-06
EOS_Omni::hybrid_rho[3]         = 2.36741168e-04
EOS_Omni::hybrid_rho[4]         = 8.11472463e-04
EOS_Omni::hybrid_rho[5]         = 1.61910043e-03
```

The model used in Lorene to generate the initial state consists of seven parts to the piecewise function with three for the core and four for the crust of the neutron star (Read et al. 2009b), with the same values for the constants as used in the polytropic equations.

2.2.5.4. Numerics

This section of the parameter file is dedicated to the thorns TmunuBase and SphericalSurface. Both thorns define variables that other thorns are then allowed to use.

```
#----------
# Numerics:
#----------
```

```
InitBase::initial_data_setup_method =
"init_some_levels"


TmunuBase::stress_energy_storage = "yes"

TmunuBase::stress_energy_at_RHS  = "yes"

TmunuBase::timelevels            =  1

TmunuBase::prolongation_type     = "none"

TmunuBase::support_old_CalcTmunu_mechanism = "no"


HydroBase::timelevels            = 3


SpaceMask::use_mask      = "yes"


SphericalSurface::nsurfaces = 5

SphericalSurface::maxntheta = 39

SphericalSurface::maxnphi = 76


SphericalSurface::ntheta      [0] = 39

SphericalSurface::nphi        [0] = 76

SphericalSurface::nghoststheta[0] = 2

SphericalSurface::nghostsphi  [0] = 2

SphericalSurface::name        [0] = "Righthand NS"


SphericalSurface::ntheta      [1] = 39

SphericalSurface::nphi        [1] = 76

SphericalSurface::nghoststheta[1] = 2

SphericalSurface::nghostsphi  [1] = 2

SphericalSurface::name        [1] = "Lefthand NS"
```

```
SphericalSurface::ntheta      [3] = 39
SphericalSurface::nphi        [3] = 76
SphericalSurface::nghoststheta[3] = 2
SphericalSurface::nghostsphi  [3] = 2
SphericalSurface::set_spherical[3] = yes
SphericalSurface::radius      [3] = 100
SphericalSurface::name        [3] = "waveextract
surface at 100"


SphericalSurface::ntheta      [4] = 39
SphericalSurface::nphi        [4] = 76
SphericalSurface::nghoststheta[4] = 2
SphericalSurface::nghostsphi  [4] = 2
SphericalSurface::set_spherical[4] = yes
SphericalSurface::radius      [4] = 250
SphericalSurface::name        [4] = "waveextract
surface at 250"
```

2.2.5.5. Evolution

The evolution section of the parameter file is dedicated to the evolution of the system. It specifies the process in which the system is evolved during each time step. The defaults of this section are based on Colella & Sekora (2008) and Mccorquodale & Colella (2011).

```
#-----------
# Evolution:
#-----------


# test
```

```
HydroBase::initial_Bvec = "zero"
Hydrobase::Bvec_evolution_method   = "GRHydro"
GRHydro::transport_constraints = yes

HydroBase::evolution_method      = "GRHydro"

ADMMacros::spatial_order = 4
GRHydro::sources_spatial_order = 4

GRHydro::riemann_solver            = "HLLE"
GRHydro::recon_method              = "ppm"
GRHydro::GRHydro_stencil            = 3
GRHydro::bound                     = "flat"
GRHydro::rho_abs_min               = 1.e-11
GRHydro::GRHydro_atmo_tolerance    = 0.01

GRHydro::c2p_reset_pressure        = "yes"

GRHydro::GRHydro_eos_type           = "General"
GRHydro::GRHydro_eos_table          = "Ideal_Fluid"

GRHydro::GRHydro_MaxNumSandRVars = 0

GRHydro::use_enhanced_ppm            = "yes"
GRHydro::sync_conserved_only     = "yes"
GRHydro::reconstruct_Wv          = "yes"
GRHydro::c2p_resort_to_bisection = "yes"
GRHydro::use_cxx_code            = "yes"
```

```
# 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::timelevels                     = 3

ML_BSSN::initial_boundary_condition  = "extrapolate-
gammas"
ML_BSSN::rhs_boundary_condition       = "NewRad"
Boundary::radpower                      = 2

ML_BSSN::harmonicN           = 1       # 1+log
ML_BSSN::harmonicF           = 2.0     # 1+log
ML_BSSN::ShiftGammaCoeff     = 0.75
ML_BSSN::AlphaDriver         = 0.0
ML_BSSN::BetaDriver          = 1.0
ML_BSSN::advectLapse         = 1.0
ML_BSSN::advectShift         = 1.0

ML_BSSN::MinimumLapse = 1.0e-8
ML_BSSN::ML_log_confac_bound = "none"
ML_BSSN::ML_metric_bound     = "none"
ML_BSSN::ML_Gamma_bound      = "none"
ML_BSSN::ML_trace_curv_bound = "none"
ML_BSSN::ML_curv_bound       = "none"
```

```
ML_BSSN::ML_lapse_bound      = "none"
ML_BSSN::ML_dtlapse_bound    = "none"
ML_BSSN::ML_shift_bound      = "none"
ML_BSSN::ML_dtshift_bound    = "none"

ML_BSSN::UseSpatialBetaDriver = 1
ML_BSSN::SpatialBetaDriverRadius = 50

#ML_BSSN::apply_dissipation   = "never"

ML_BSSN::epsDiss              =0.0

Dissipation::epsdis = 0.1
Dissipation::order = 5
Dissipation::vars                       = "
        ML_BSSN::ML_log_confac
        ML_BSSN::ML_metric
        ML_BSSN::ML_trace_curv
        ML_BSSN::ML_curv
        ML_BSSN::ML_Gamma
        ML_BSSN::ML_lapse
        ML_BSSN::ML_shift
        ML_BSSN::ML_dtlapse
        ML_BSSN::ML_dtshift
"
```

2.2.6. Output

This section of the parameter file is designated to create the output for the thorns that have a value that the user would like saved. These outputs usually take the form of a HDF5, or the ASCII file formats.

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

IOBasic::outInfo_every = 1
IOBasic::outInfo_reductions = "maximum"
IOBasic::outInfo_vars  = "
 Carpet::physical_time_per_hour
 HydroBase::rho
 ML_ADMConstraints::ML_Ham
 SystemStatistics::maxrss_mb
 GRHydro::dens{reductions = 'sum maximum'}
 HydroBase::w_lorentz
"

IOScalar::outScalar_every      = 256 # every_coarse
IOScalar::all_reductions_in_one_file = "yes"
IOScalar::one_file_per_group   = "yes"
IOScalar::outScalar_reductions = "minimum maximum
average norm1 norm2"
IOScalar::outScalar_vars       = "
 ADMBase::lapse
 ADMBase::shift
 ADMBase::metric
 ADMBase::curv
 HydroBase::rho
 HydroBase::vel
 HydroBase::w_lorentz
 GRHydro::dens{reductions = 'minimum maximum average
norm1 norm2 sum'}
 SystemStatistics::process_memory_mb
 SphericalSurface::sf_radius
 ML_ADMConstraints::ML_Ham
"

IOASCII::one_file_per_group     = "yes"
IOASCII::compact_format  = "yes"

IOASCII::out0D_every     = 256 # every_coarse
IOASCII::out0D_vars      = "
 Carpet::timing
 QuasiLocalMeasures::qlm_scalars
 SphericalSurface::sf_active
 SphericalSurface::sf_valid
 SphericalSurface::sf_info
 SphericalSurface::sf_radius
 SphericalSurface::sf_origin
 SphericalSurface::sf_coordinate_descriptors
 Hydro_Analysis::Hydro_Analysis_rho_max_loc

Hydro_Analysis::Hydro_Analysis_rho_max_origin_distance
"

#Set these IOASCII options for initial data only:
IOASCII::out1D_every     = 0
IOASCII::out1D_d         = "no"
```

```
IOASCII::out1D_vars      = "
 HydroBase::rho
 HydroBase::vel
 ADMBase::lapse
 ADMBase::shift
 ADMBase::metric
 ADMBase::curv
 ML_ADMConstraints::ML_Ham
"


CarpetIOHDF5::one_file_per_group            = "no"
# this is required by multipatch
CarpetIOHDF5::open_one_input_file_at_a_time  = "yes"
CarpetIOHDF5::out2D_every                   = 1536
# 6*every coarse
CarpetIOHDF5::out2D_xy                      = "yes"
CarpetIOHDF5::out2D_xz                      = "no"
CarpetIOHDF5::out2D_yz                      = "no"
CarpetIOHDF5::out2D_xyplane_z               = 0.0
CarpetIOHDF5::out2D_vars      = "
  CarpetReduce::weight
  Grid::coordinates
  HydroBase::rho
  HydroBase::vel
  HydroBase::entropy
  HydroBase::press
  HydroBase::eps
  ADMBase::lapse
  ADMBase::shift
  ADMBase::metric
  ML_ADMConstraints::ML_Ham
 "

IOHDF5::out3D_every = 8192 # = 32*every_coarse
IOHDF5::out3D_vars  = "
 CarpetReduce::weight
 HydroBase::rho
 HydroBase::vel
 HydroBase::eps
 ADMBase::lapse
 ADMBase::shift
 ML_ADMConstraints::ML_Ham
 grid::coordinates
"
```

2.2.7. Analysis

The analysis section of the parameter file is dedicated to specifying details for the thorns: `Hydro_Analysis`, `NSTracker`, and `QuasiLocalMeasures`.

```
#------------------------------------------------------
# Analysis:
#------------------------------------------------------
Hydro_Analysis::Hydro_Analysis_comp_rho_max = "true"
Hydro_Analysis::Hydro_Analysis_rho_max_loc_only_positive_x = "true"
Hydro_Analysis::Hydro_Analysis_comp_rho_max_origin_distance = "yes"
Hydro_Analysis::Hydro_Analysis_average_multiple_maxima_locations = "yes"
Hydro_Analysis::Hydro_Analysis_interpolator_name = "Lagrange polynomial interpolation (tensor product)"
```

```
NSTracker::NSTracker_SF_Name          = "Righthand NS"
NSTracker::NSTracker_SF_Name_Opposite = "Lefthand NS"
NSTracker::NSTracker_max_distance = 10
NSTracker::NSTracker_verbose = "yes"
NSTracker::NSTracker_tracked_location =
"Hydro_Analysis::Hydro_Analysis_rho_max_loc"


QuasiLocalMeasures::num_surfaces   = 2
QuasiLocalMeasures::spatial_order  = 4
QuasiLocalMeasures::interpolator = "Lagrange
polynomial interpolation"
QuasiLocalMeasures::interpolator_options = "order=4"
QuasiLocalMeasures::surface_name[0] = "waveextract
surface at 100"
QuasiLocalMeasures::surface_name[1] = "waveextract
surface at 250"
```

2.2.8. Wave Extraction

This section of the parameter file is designated for calculating the fourth Weyl scalar which is associated with gravitational waves at various distances for various multipole modes ranging from $l = 0$ to $l = 6$.

```
######################################################
######################################################
# Wave extraction
######################################################
######################################################


WeylScal4::offset                    = 1e-8
```

```
WeylScal4::fd_order                  = "4th"
WeylScal4::verbose                   = 0

Multipole::nradii = 8
Multipole::out_every = 128
Multipole::radius[0] = 45
Multipole::radius[1] = 70
Multipole::radius[2] = 100
Multipole::radius[3] = 125
Multipole::radius[4] = 150
Multipole::radius[5] = 200
Multipole::radius[6] = 250
Multipole::radius[7] = 300
Multipole::variables = "WeylScal4::Psi4r{sw=-2
cmplx='WeylScal4::Psi4i' name='Psi4'}"
Multipole::l_max = 6
```

2.2.9. Checkpoint/Recovery

The Checkpoint section is responsible for making backups of the simulation that can be restarted if the system experiences some type of failure that cancels the simulation.

```
#-----------------------------------------------------
# Checkpoint/Recovery:
#-----------------------------------------------------
IOHDF5::checkpoint                  = "yes"
IO::checkpoint_dir                  = "../checkpoint"
IO::checkpoint_ID                   = "yes"
IO::checkpoint_every_walltime_hours = 6.0
```

```
CarpetIOHDF5::checkpoint_every_divisor = 55552
IO::checkpoint_keep=2
IO::checkpoint_on_terminate         = "yes"

IO::recover     = "autoprobe"
IO::recover_dir = "../checkpoint"
```

The parameter settings listed above informs the simulation to create a checkpoint every six hours of simulation time and to save the two most-recent checkpoints.

2.2.10. AHFinderDirect

The `AHFinderDirect` section of the parameter file is dedicated to how the thorn `AHFinderDirect` is used during the simulation.

```
#------------------------------------------------------
# AHFinderDirect:
#------------------------------------------------------

AHFinderDirect::find_every = 0

AHFinderDirect::run_at_CCTK_POST_RECOVER_VARIABLES = 
no

AHFinderDirect::move_origins            = yes
AHFinderDirect::reshape_while_moving    = yes
AHFinderDirect::predict_origin_movement = yes

AHFinderDirect::geometry_interpolator_name = "Lagrange 
polynomial interpolation"
AHFinderDirect::geometry_interpolator_pars = "order=4"
```

```
AHFinderDirect::surface_interpolator_name  = "Lagrange
polynomial interpolation"
AHFinderDirect::surface_interpolator_pars  = "order=4"


AHFinderDirect::output_h_every = 128


AHFinderDirect::N_horizons = 1


AHFinderDirect::which_surface_to_store_info
[1] = 0
AHFinderDirect::reset_horizon_after_not_finding
[1] = no
AHFinderDirect::initial_guess__coord_sphere__radius
[1] = 1.3528
```

This section tells the `AHFinderDirect` thorn to look for an apparent horizon at the origin every 128 iterations. This is necessary to allow the simulation to run smoothly once the merger of the BNS has taken place and the resultant object collapses down into a black hole. If this thorn is not implemented properly, the mass will be concentrated in an area that is smaller than the resolution of the simulation and the simulation will fail.

2.2.11. Control

This section of the parameter file is designated for the turning on and off specific thorns at either a set time or iteration of the simulation. For example, as seen below there is a point where the lapse value falls below a specific value, $\alpha < 0.2$. This causes the thorn `AHFinderDirect` to activate and begin searching for an apparent horizon.

```
#-------------------------------------------------------
# Control
```

```
#------------------------------------------------------

ActiveThorns = "Trigger"

Trigger::Trigger_Number = 3

# Check for lapse < 0.5, and if so, set this trigger
to 'once'
Trigger::Trigger_Once           [0] = 0
Trigger::Trigger_Checked_Variable[0] = "ADMBase::alp"
Trigger::Trigger_Reduction      [0] = "minimum"
Trigger::Trigger_Relation       [0] = "<"
Trigger::Trigger_Checked_Value   [0] = 0.5
Trigger::Trigger_Reaction              [0] =
"steerparam"
Trigger::Trigger_Steered_Parameter_Thorn[0] =
"Trigger"
Trigger::Trigger_Steered_Parameter_Name [0] =
"Trigger_Once[0]"
Trigger::Trigger_Steered_Parameter_Value[0] = "1"

# if lapse < 0.2 enable AHFinder
Trigger::Trigger_Once           [1] = 1
Trigger::Trigger_Checked_Variable[1] = "ADMBase::alp"
Trigger::Trigger_Reduction      [1] = "minimum"
Trigger::Trigger_Relation       [1] = "<"
Trigger::Trigger_Checked_Value   [1] = 0.3
Trigger::Trigger_Reaction              [1] =
"steerparam"
Trigger::Trigger_Steered_Parameter_Thorn[1] =
"AHFinderDirect"
```

```
Trigger::Trigger_Steered_Parameter_Name [1] =
"find_every"
Trigger::Trigger_Steered_Parameter_Value[1] = "128"

# if trigger 0 was set 'once' (lapse < 0.5)
# add refinement level
Trigger::Trigger_Once                   [2] = 1
Trigger::Trigger_Checked_Variable       [2] = "param"
Trigger::Trigger_Checked_Parameter_Thorn[2] =
"Trigger"
Trigger::Trigger_Checked_Parameter_Name [2] =
"Trigger_Once[0]"
Trigger::Trigger_Relation               [2] = "=="
Trigger::Trigger_Checked_Value          [2] = 1
Trigger::Trigger_Reduction              [2] = ""
Trigger::Trigger_Reaction               [2] =
"steerscalar"
Trigger::Trigger_Steered_Scalar         [2] =
"CarpetRegrid2::num_levels[0]"
Trigger::Trigger_Steered_Scalar_Value   [2] = "7"
```

2.2.12. VolumeIntegrals_GRMHD

This section of the parameter file is designated for setting up the `VolumeIntegrals_GRMHD` thorn. Recall from above that this thorn is used for the volume integration of specific scalar quantities such as the rest mass and other such scalars.

```
#-------VolumeIntegrals_GRMHD----------------

VolumeIntegrals_GRMHD::NumIntegrals = 6
```

```
VolumeIntegrals_GRMHD::VolIntegral_out_every = 64

VolumeIntegrals_GRMHD::enable_file_output = 1

VolumeIntegrals_GRMHD::outVolIntegral_dir =
"volume_integration"

VolumeIntegrals_GRMHD::verbose = 1


## The AMR centre will only track the first referenced
integration quantities that track said centre.

##   Thus, centeroflapse output will not feed back
into the AMR centre positions.


VolumeIntegrals_GRMHD::Integration_quantity_keyword[1]
= "one"

VolumeIntegrals_GRMHD::Integration_quantity_keyword[2]
= "centerofmass"

VolumeIntegrals_GRMHD::Integration_quantity_keyword[3]
= "one"

VolumeIntegrals_GRMHD::Integration_quantity_keyword[4]
= "centerofmass"

VolumeIntegrals_GRMHD::Integration_quantity_keyword[5]
= "one"

VolumeIntegrals_GRMHD::Integration_quantity_keyword[6]
= "restmass"


#Use output from volume integral to move AMR box
centre 2


VolumeIntegrals_GRMHD::volintegral_sphere__center_x_in
itial         [2] = -15.0

VolumeIntegrals_GRMHD::volintegral_inside_sphere__radi
us            [2] =  10.0

VolumeIntegrals_GRMHD::amr_centre__tracks__volintegral
_inside_sphere[2] =  1

VolumeIntegrals_GRMHD::volintegral_sphere__center_x_in
itial         [3] = -15.0
```

```
VolumeIntegrals_GRMHD::volintegral_inside_sphere__radi
us            [3] =  10.0


#Use output from volume integral to move AMR box
centre 3


VolumeIntegrals_GRMHD::volintegral_sphere__center_x_in
itial         [4] =  15.0

VolumeIntegrals_GRMHD::volintegral_inside_sphere__radi
us            [4] =  10.0

VolumeIntegrals_GRMHD::amr_centre__tracks__volintegral
_inside_sphere[4] =  2

VolumeIntegrals_GRMHD::volintegral_sphere__center_x_in
itial         [5] =  15.0

VolumeIntegrals_GRMHD::volintegral_inside_sphere__radi
us            [5] =  10.0
```

3. Data

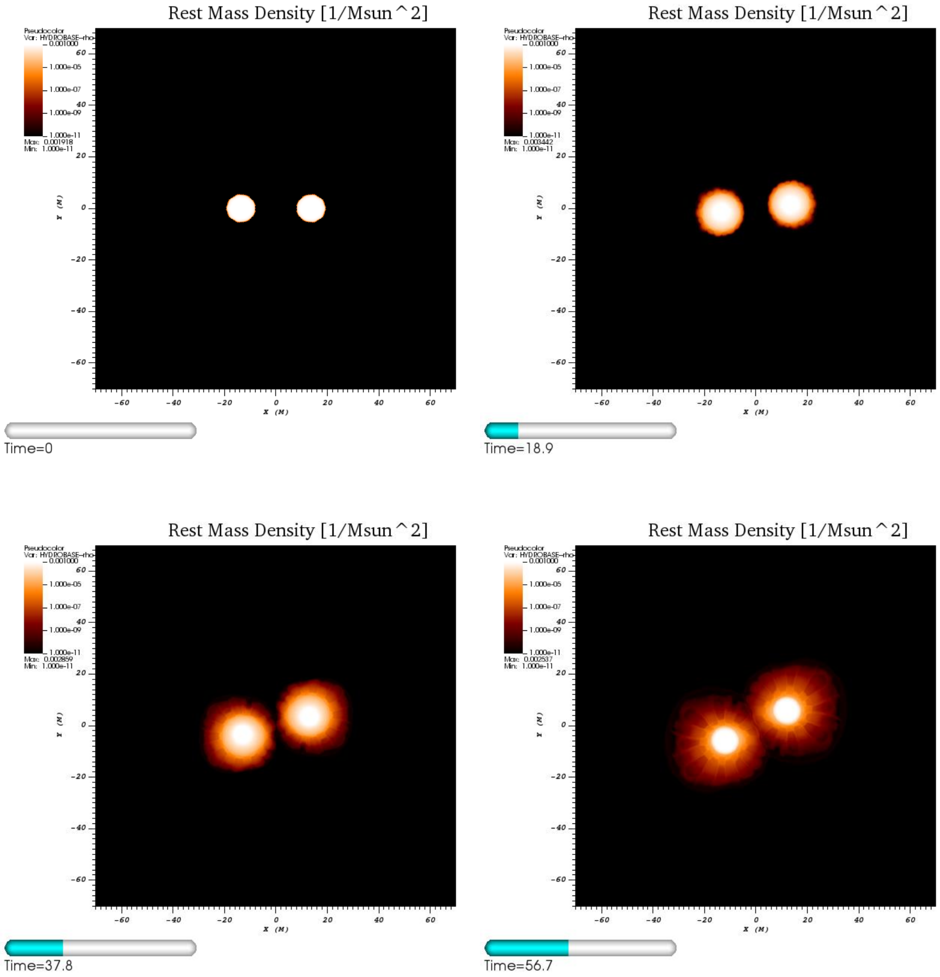
3.1. Case 1: 2.08 and 2.08 $M_{\odot}$

The first case that was simulated was the case of two maximum mass neutron stars that were observed by Mazzali et al. (2007) at 2.08 $M_{\odot}$ each. At the start of the simulation the stars were separated by a distance of 40 km from each other using Lorene to create the initial conditions. Unfortunately, this simulation failed a number of times, all at approximately the same point. This indicates that there is something wrong with either the initial conditions that are generated by Lorene, the way the Einstein Toolkit handles the evolution for a system this massive, or the computational power (*e.g.*, memory) that was dedicated to this task was insufficient to handle a simulation of this size, with the latter being most likely. To fully test the simulation, more computational resources would need to be dedicated to this task.

3.1.1. Matter Distribution

During the evolution of the binary neutron star system, one of the best ways to see how matter is behaving is through watching how the density distribution changes over time. Given below is the density distribution of the evolution of case1 at various points along the simulation, indicated at each step. As can be seen below, the matter distribution starts as two concentrated areas of mass that slightly expand over time and orbit around each other.

Figure 2: Density distribution during case 1 merger

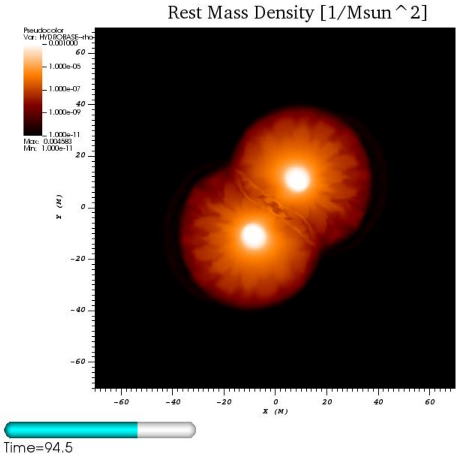


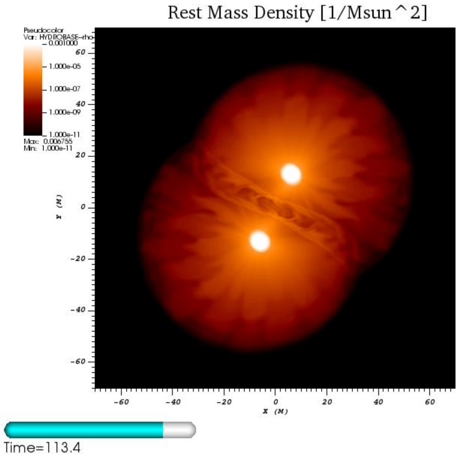
Rest Mass Density [1/Msun^2]

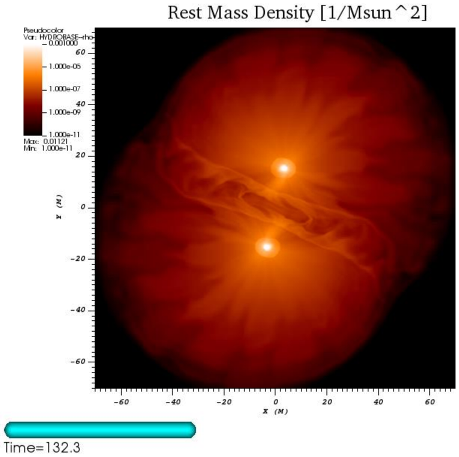
Rest Mass Density [1/Msun^2]

Rest Mass Density [1/Msun^2]

Rest Mass Density [1/Msun^2]





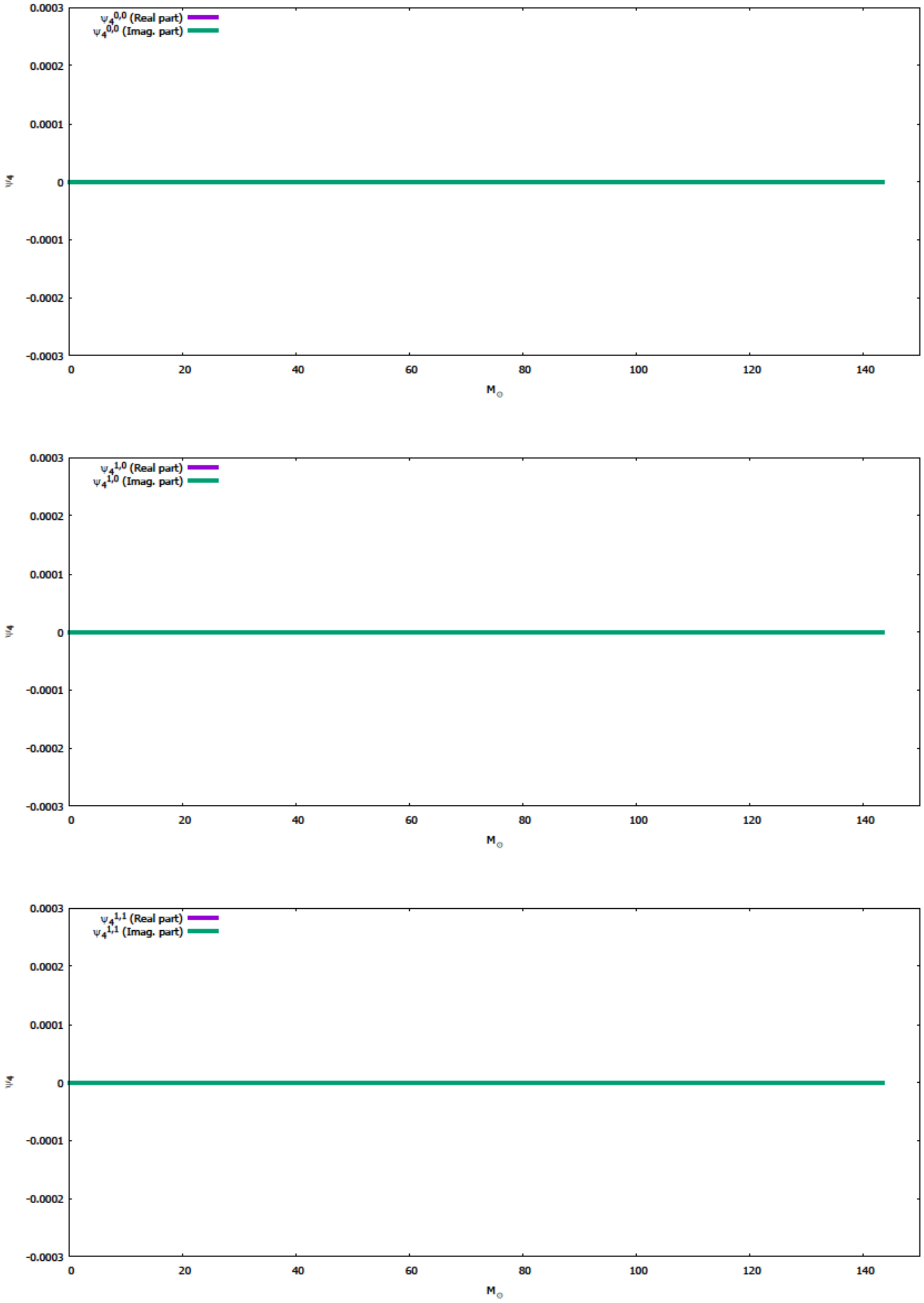


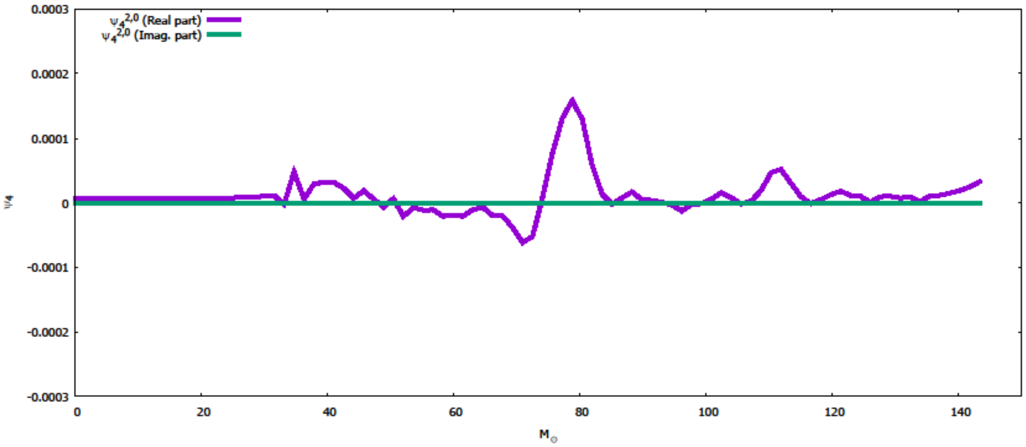
As can be seen above, the bodies orbit around each for approximately one-quarter of an orbit before the simulation fails at producing any further iterations. The time for this simulation is approximately $132.3\ M_{\odot}$ or $6.518 \times 10^{-4}\ s$.

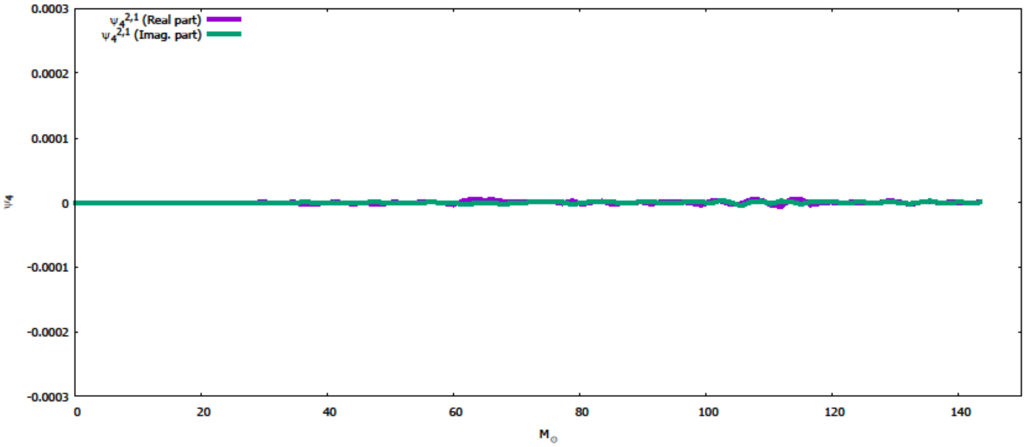
3.1.2. Gravitational Waves

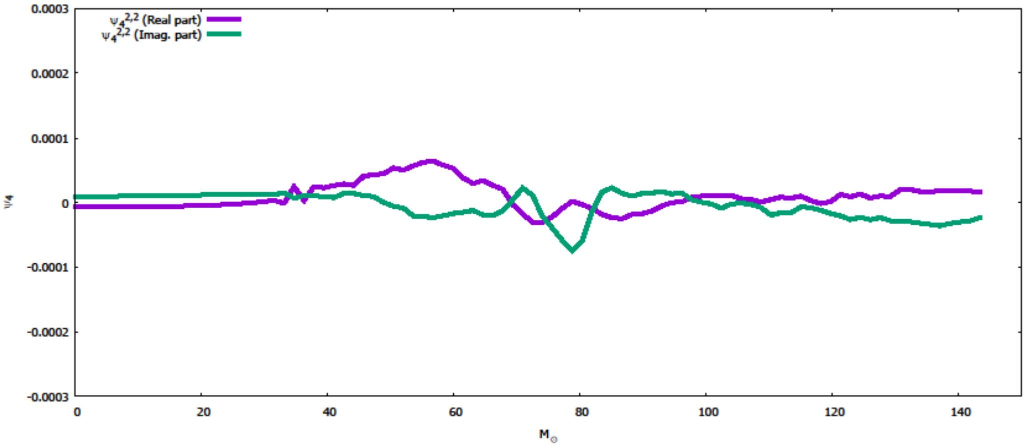
Like described above during the inspiral of the BNS, gravitational waves should be produced and be the driving factor behind the dissipation of energy that eventually causes neutron stars to coalesce. Various modes of gravitational waves follow, each measured at $45\ M_{\odot}$ or approximately 67 km.

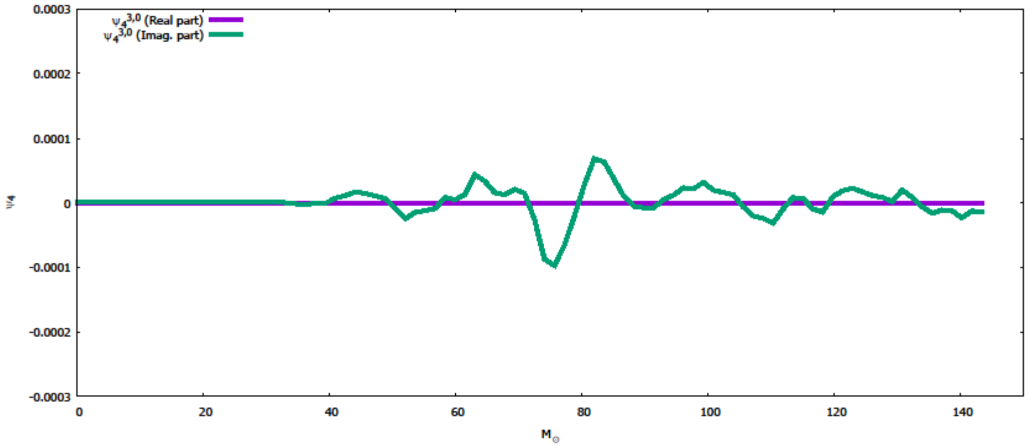
Figure 3: Gravitational wave strength of different rotation modes during case 1 merger

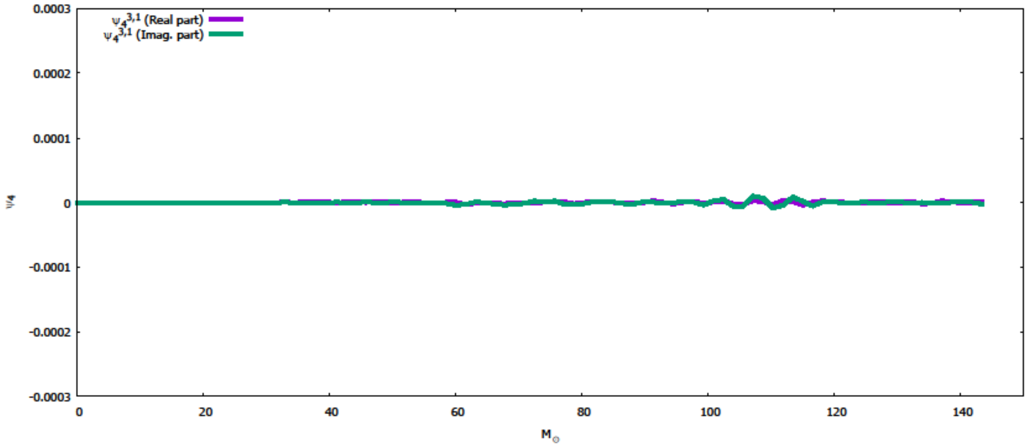


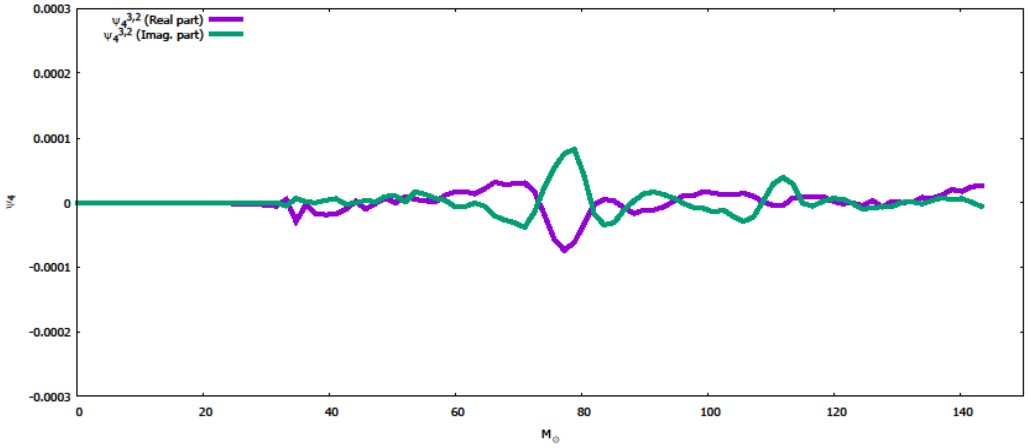

$\psi_4^{2,0}$ (Real part)
$\psi_4^{2,0}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$

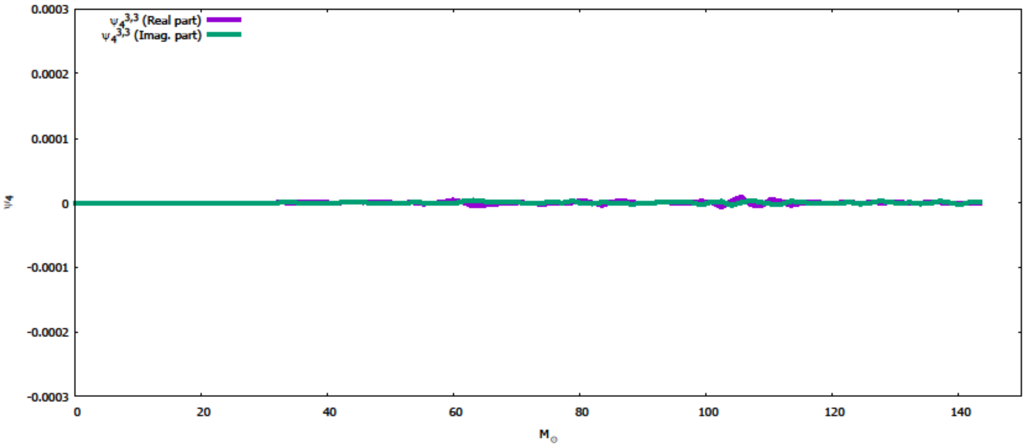

$\psi_4^{2,1}$ (Real part)
$\psi_4^{2,1}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$

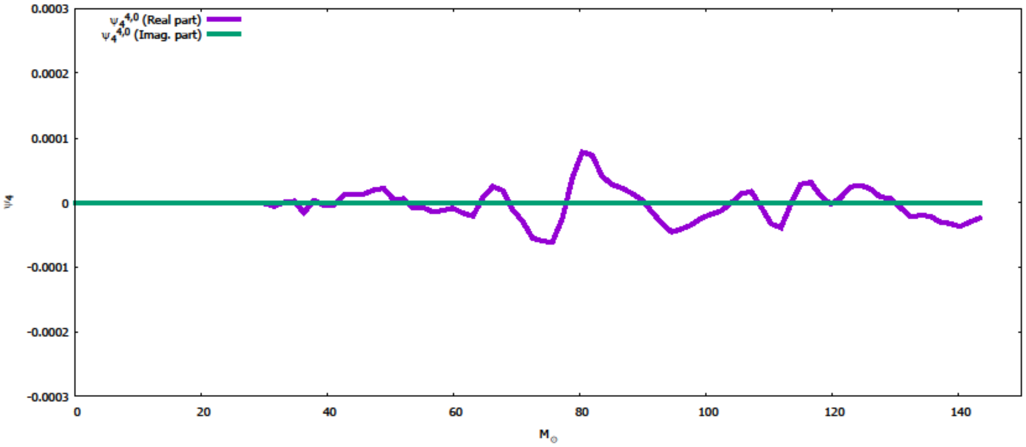

$\psi_4^{2,2}$ (Real part)
$\psi_4^{2,2}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$

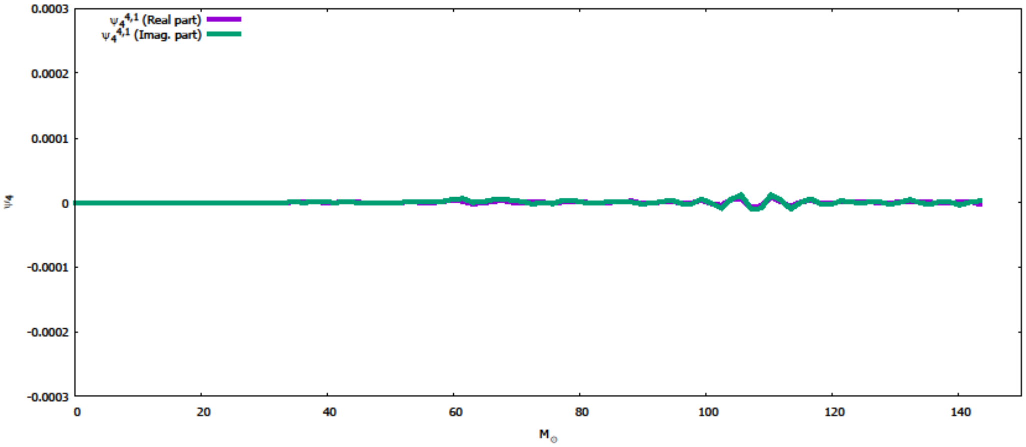

$\psi_4^{3,0}$ (Real part)
$\psi_4^{3,0}$ (Imag. part)
ψ_4
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
0
20
40
60
80
100
120
140
$M_\odot$

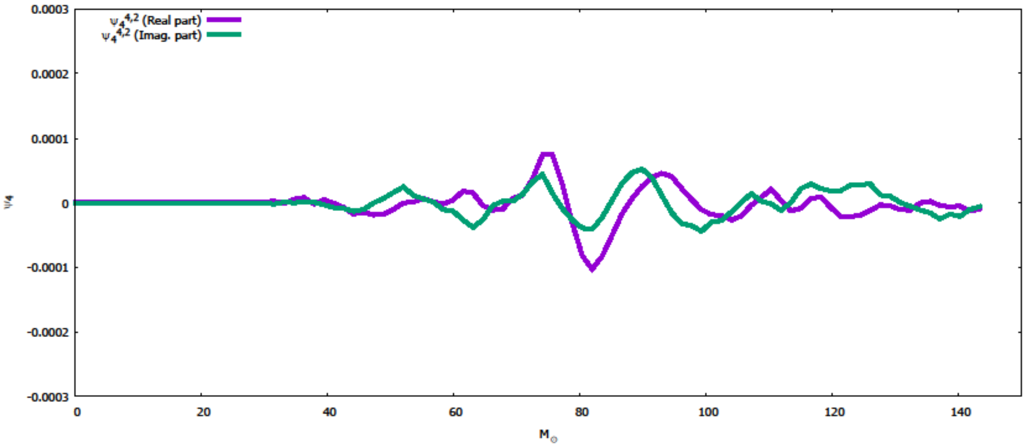

$\psi_4^{3,1}$ (Real part)
$\psi_4^{3,1}$ (Imag. part)
ψ_4
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
0
20
40
60
80
100
120
140
$M_\odot$

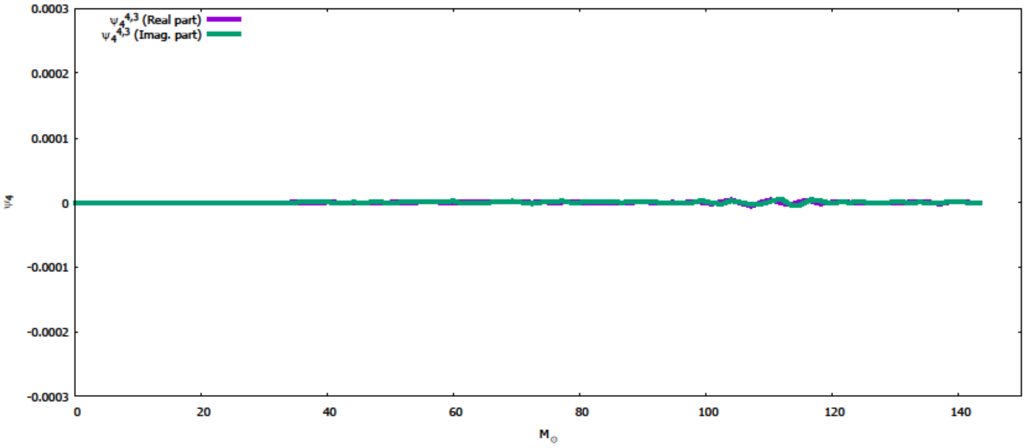

$\psi_4^{3,2}$ (Real part)
$\psi_4^{3,2}$ (Imag. part)
ψ_4
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
0
20
40
60
80
100
120
140
$M_\odot$

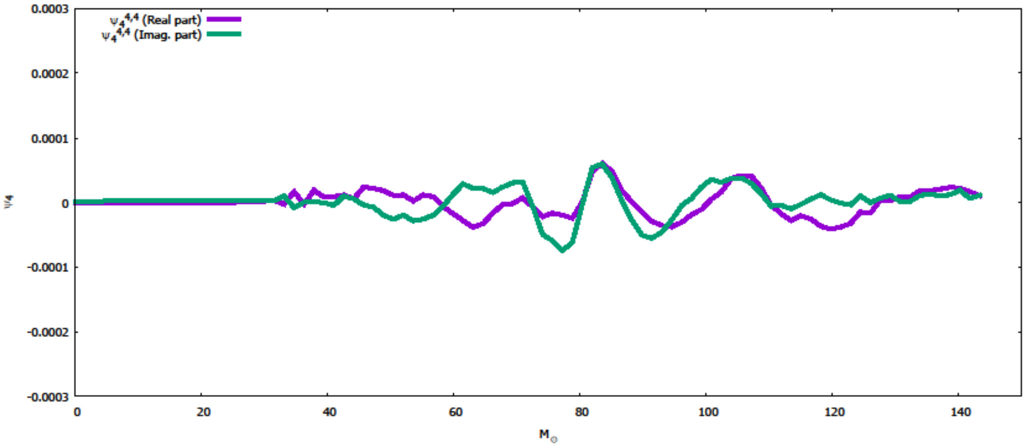

$\psi_4^{3,3}$ (Real part)
$\psi_4^{3,3}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$

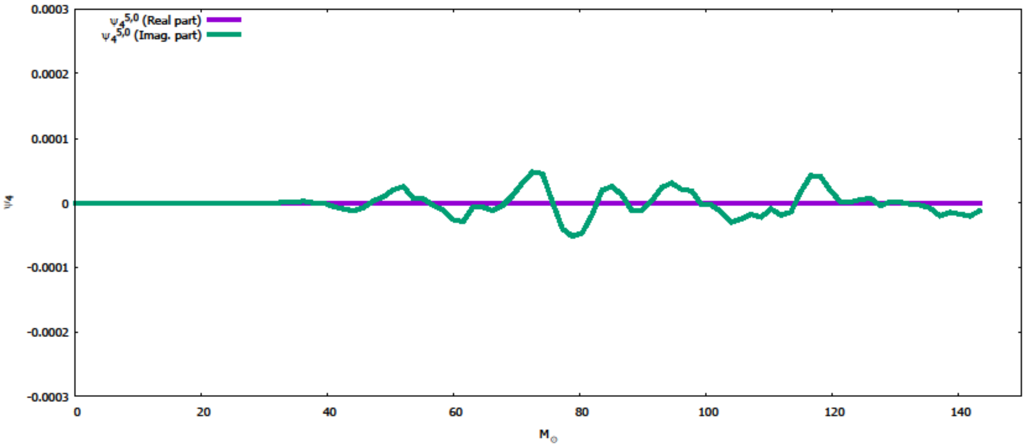

$\psi_4^{4,0}$ (Real part)
$\psi_4^{4,0}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$

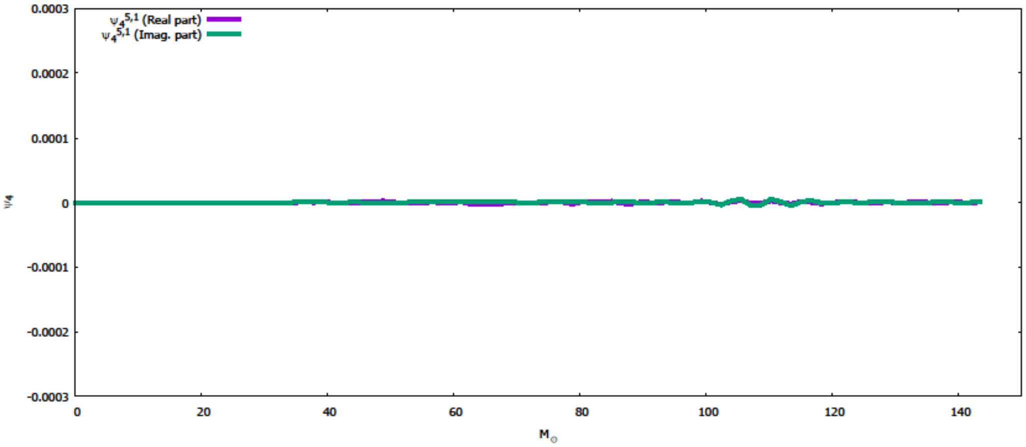

$\psi_4^{4,1}$ (Real part)
$\psi_4^{4,1}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$

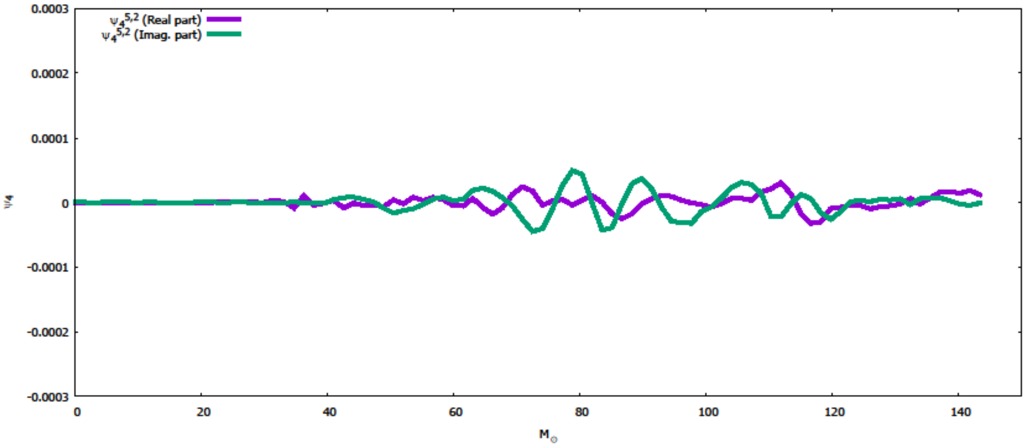

ψ4^{4,2} (Real part)
ψ4^{4,2} (Imag. part)
ψ4
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
0
20
40
60
80
100
120
140
M⊙

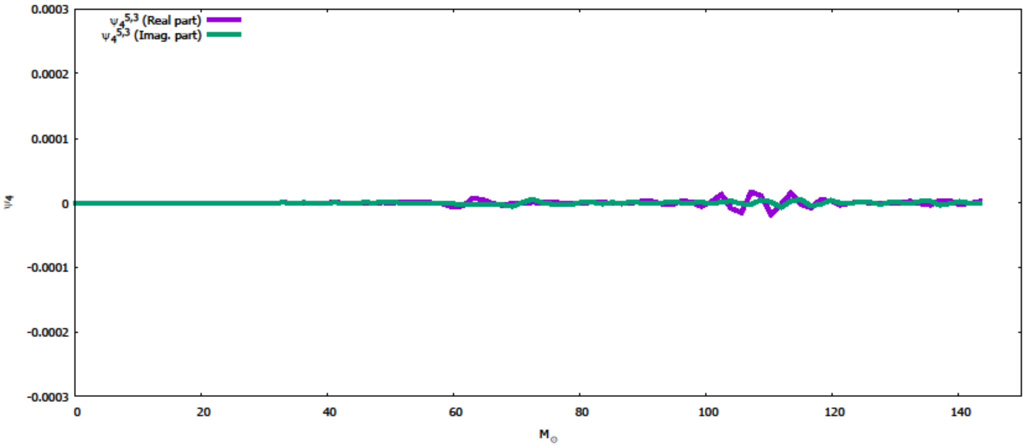

ψ4^{4,3} (Real part)
ψ4^{4,3} (Imag. part)
ψ4
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
0
20
40
60
80
100
120
140
M⊙

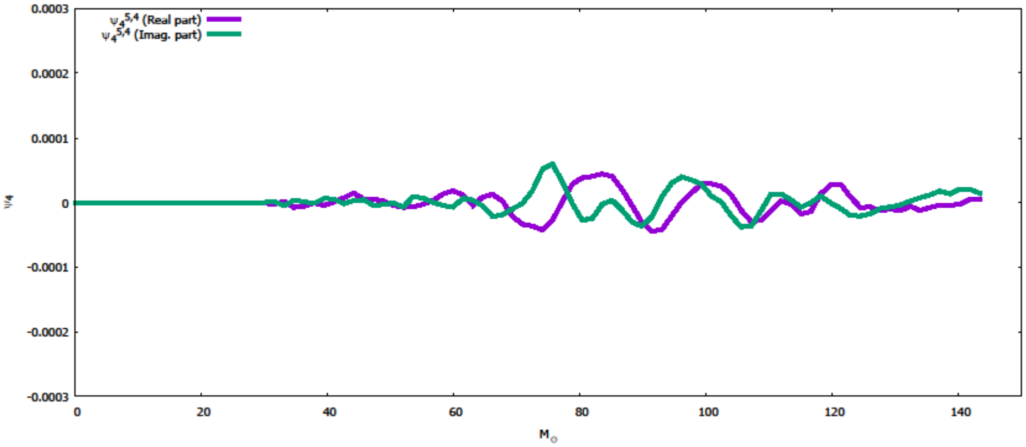

ψ4^{4,4} (Real part)
ψ4^{4,4} (Imag. part)
ψ4
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
0
20
40
60
80
100
120
140
M⊙

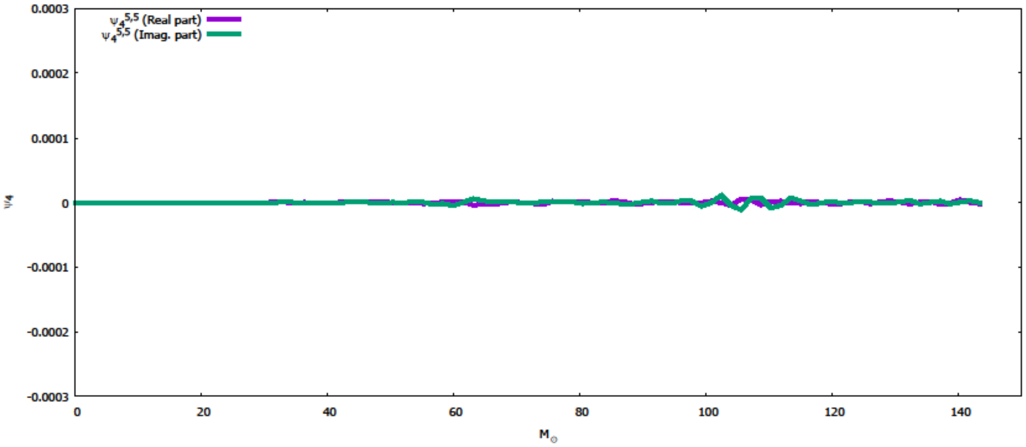

$\psi_4^{5,0}$ (Real part)
$\psi_4^{5,0}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ4
0
20
40
60
80
100
120
140
$M_\odot$

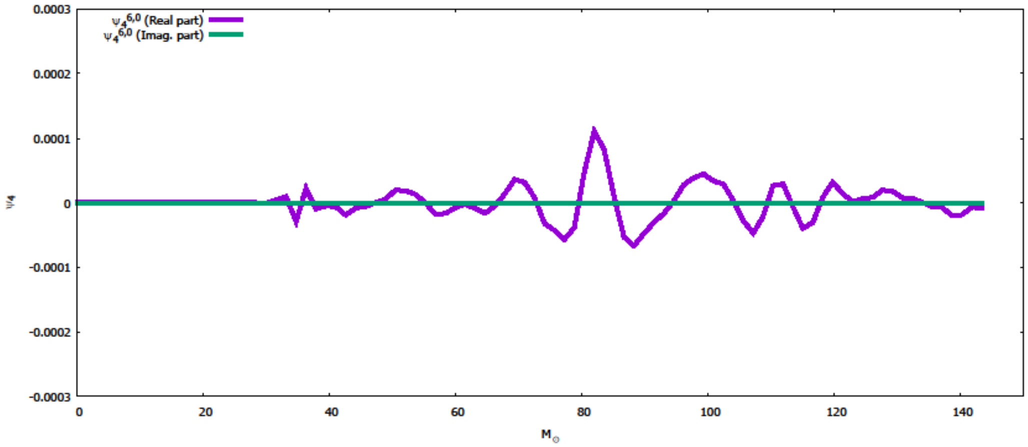

$\psi_4^{5,1}$ (Real part)
$\psi_4^{5,1}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ4
0
20
40
60
80
100
120
140
$M_\odot$

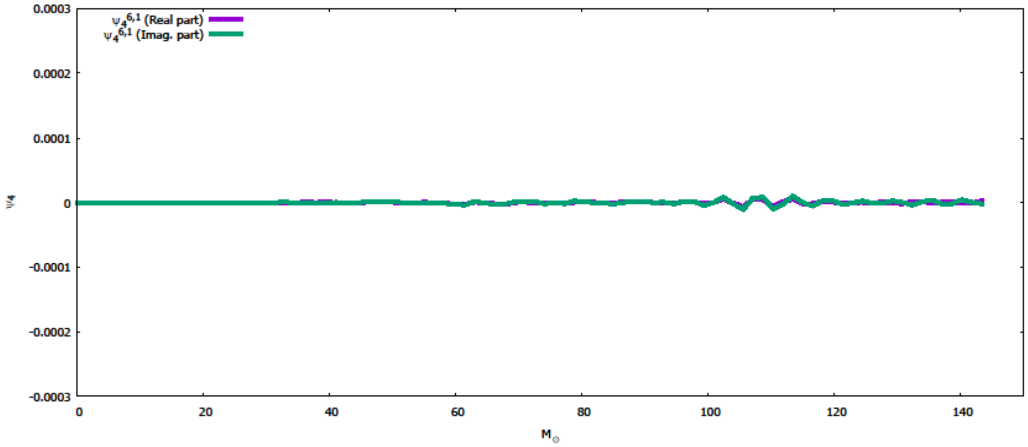

$\psi_4^{5,2}$ (Real part)
$\psi_4^{5,2}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ4
0
20
40
60
80
100
120
140
$M_\odot$

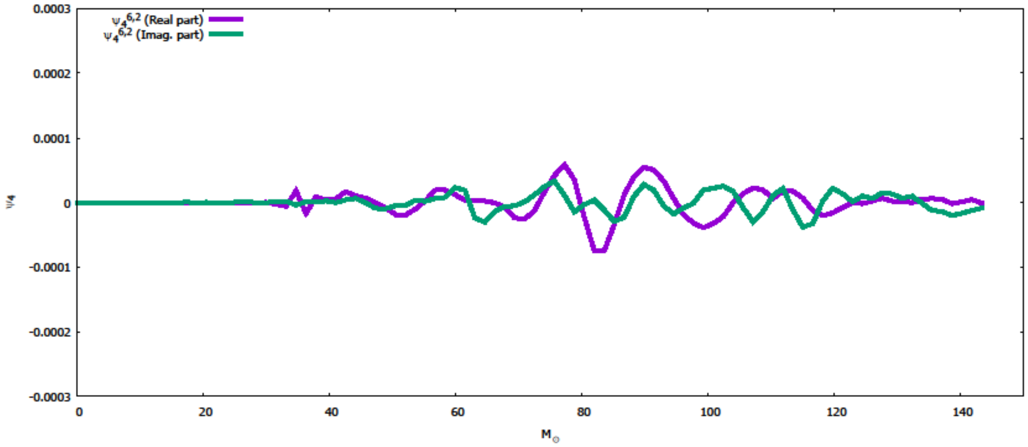

$\psi_4^{5,3}$ (Real part)
$\psi_4^{5,3}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$

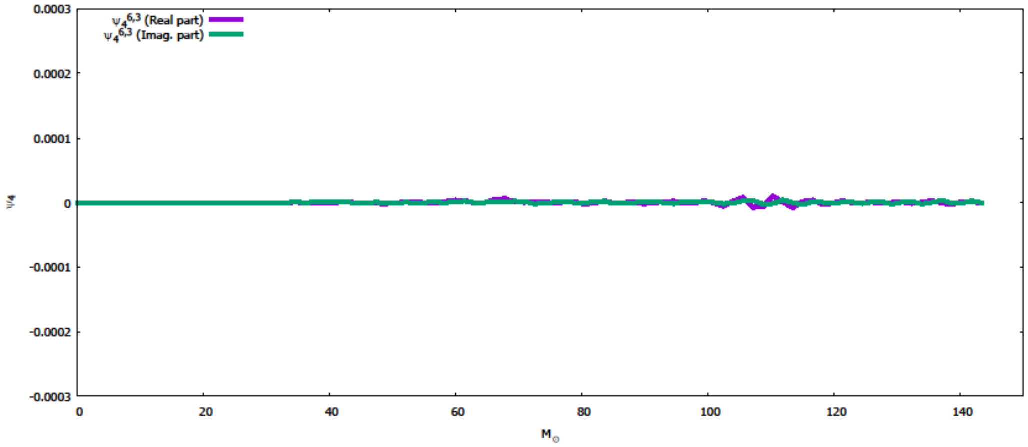

$\psi_4^{5,4}$ (Real part)
$\psi_4^{5,4}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$

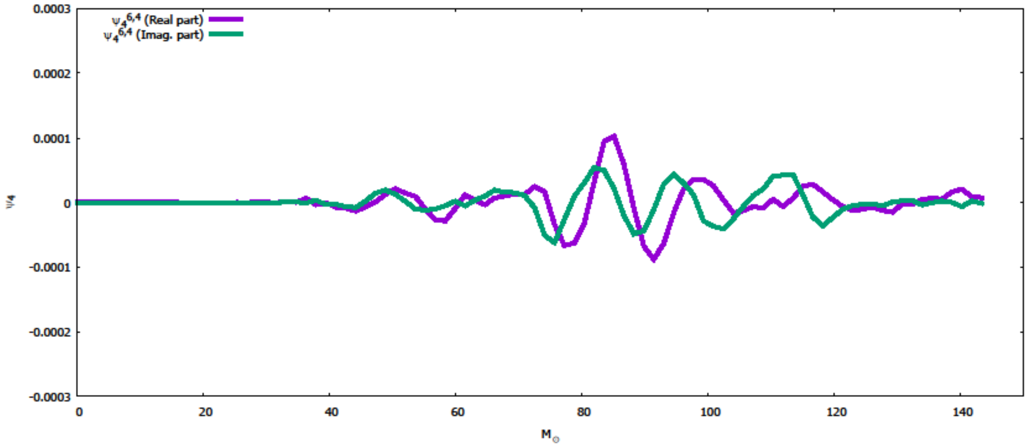

$\psi_4^{5,5}$ (Real part)
$\psi_4^{5,5}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$

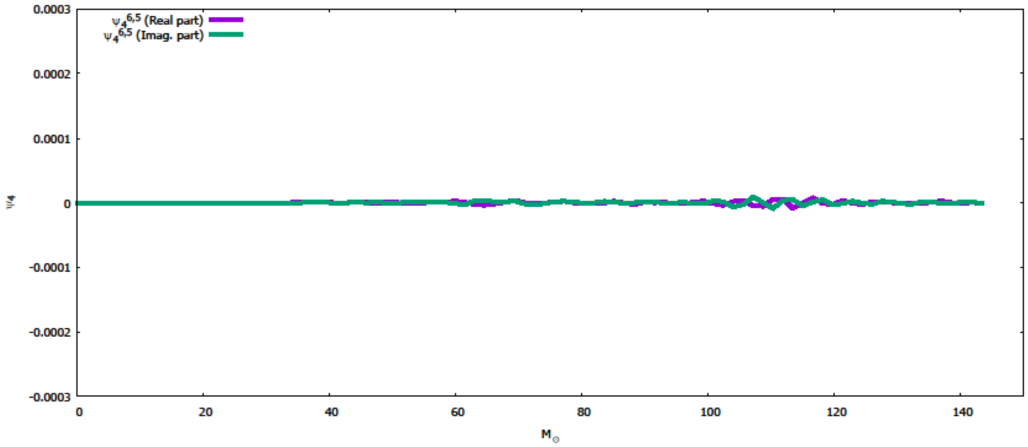

$\psi_4^{6,0}$ (Real part)
$\psi_4^{6,0}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$

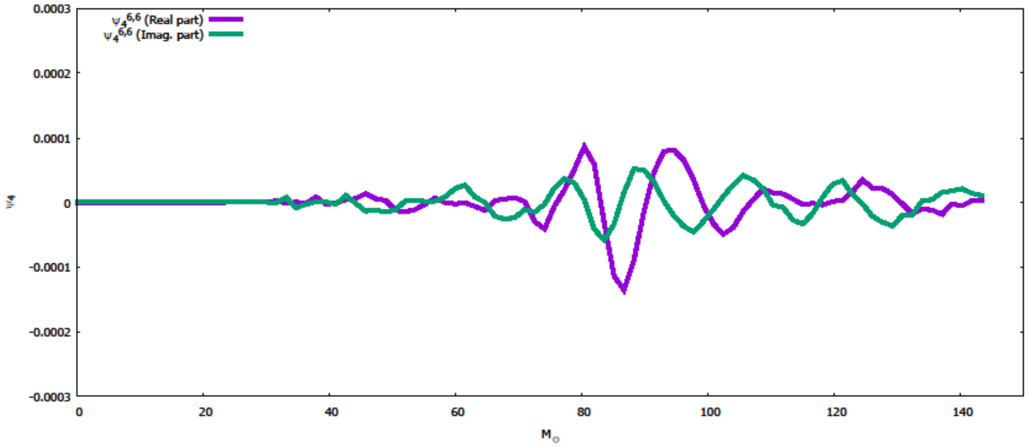

$\psi_4^{6,1}$ (Real part)
$\psi_4^{6,1}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$


$\psi_4^{6,2}$ (Real part)
$\psi_4^{6,2}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$


$\psi_4^{6,3}$ (Real part)
$\psi_4^{6,3}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$


$\psi_4^{6,4}$ (Real part)
$\psi_4^{6,4}$ (Imag. part)
0.0003
0.0002
0.0001
0
-0.0001
-0.0002
-0.0003
ψ_4
0
20
40
60
80
100
120
140
$M_\odot$





As seen above, for the majority of the rotational modes there are no gravitational waves produced. The waves that are being produced begin around 40 $M_\odot$, which is approximately the distance at which the gravitational waves are being measured from. The initial ripples when the simulation starts is presumably from the metric being conformally flat in the Lorene initial data to being allowed to change shape in the Einstein Toolkit.

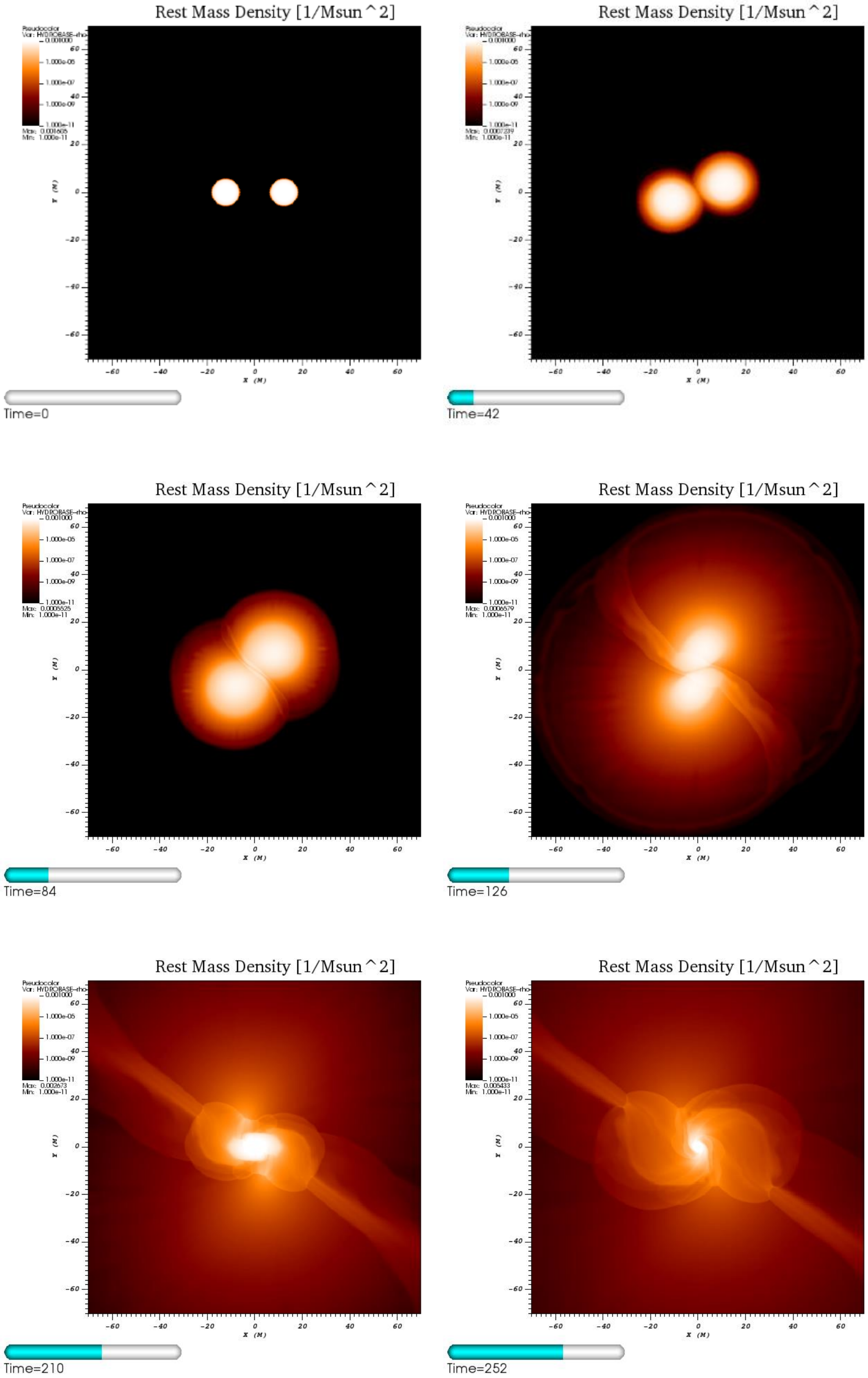
3.2. Case 2: 1.40 and 1.74 $M_\odot$

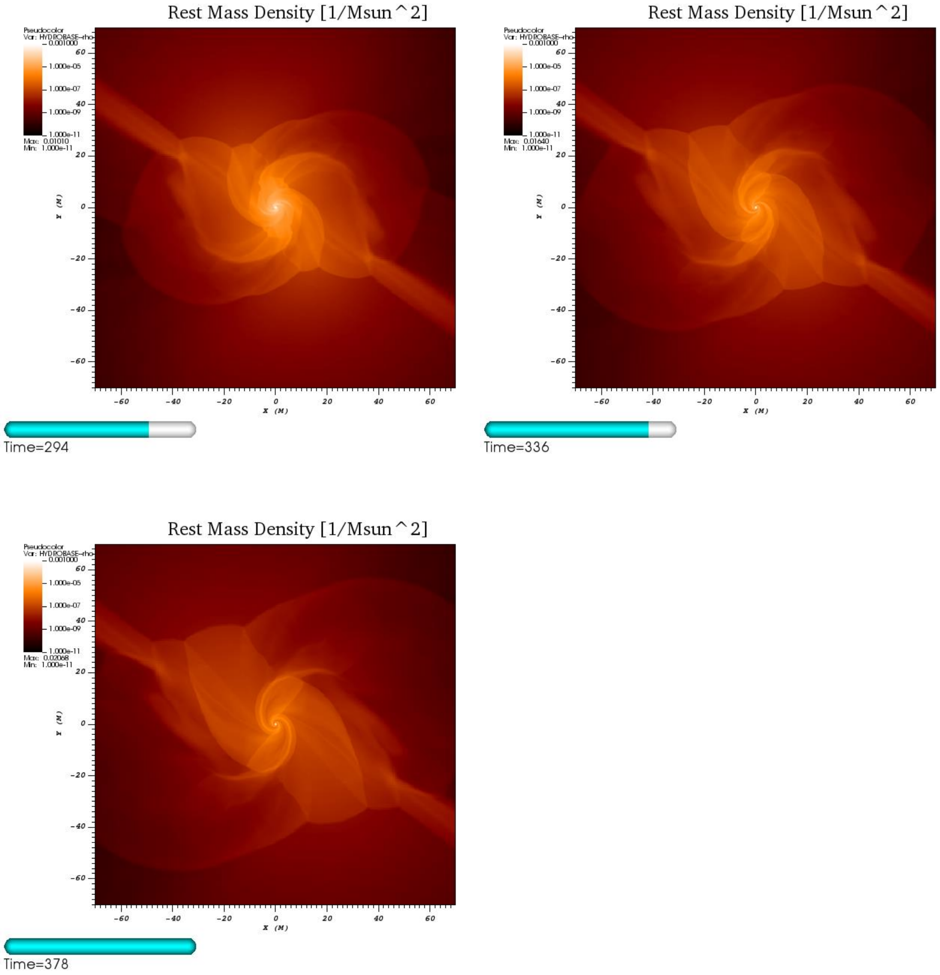
The second case consists of a neutron star of the minimum mass, 1.40 $M_\odot$, and an intermediate mass of 1.74 $M_\odot$ starting at a separation distance of 40 km. This simulation was more successful than case1 being that it resulted in the merger of the neutron stars and the subsequence collapse into a black hole. For this case to be stable, the thorn `AHFinderDirect` needs to be implemented otherwise, the simulation would halt once the distribution of mass was in a region smaller than the finest grid size. Since the `AHFinderDirect` thorn is implemented, it is possible to find the mass of the resultant black hole, which is found to be $2.23\ M_{\odot}$.

3.2.1. Matter Distribution

During the simulations, it was seen that neutron stars start at their intial positions and material expands filling the sourounding area as the stars begin to orbit each other. Below it can be seen that the stars complete approximately half an orbit before they lose enough energy to graviational waves that they begin to merge.

Figure 4: Density distribution during case 2 merger



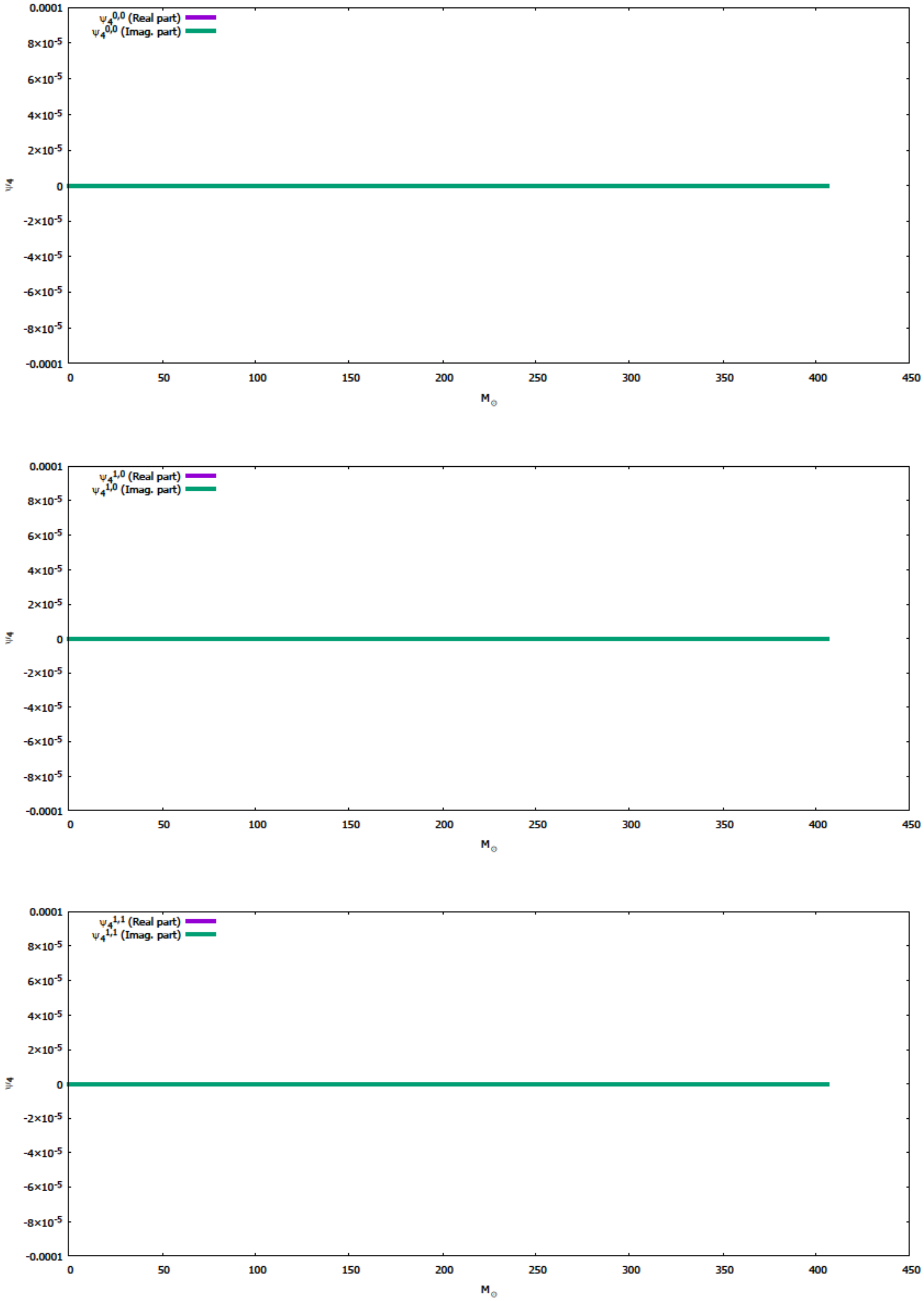


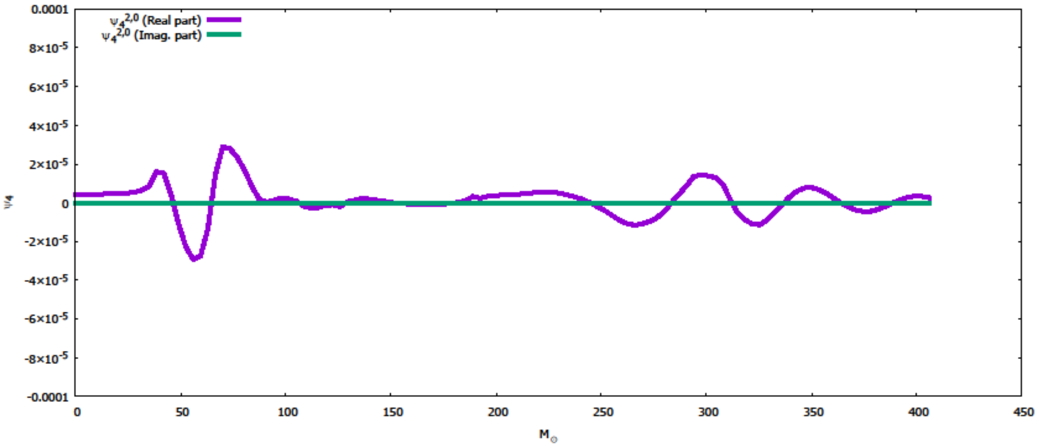
It is at approximately $250\ M_{\odot}$ or $1.2 \times 10^{-3}\ s$ when the neutron stars being to coalesce and collapse into a black hole. The system then evolves further until most of the material is within the event horizon of the resultant black hole but stopped before the black hole can clear anything within its vicinity.

3.2.2. Gravitational Waves

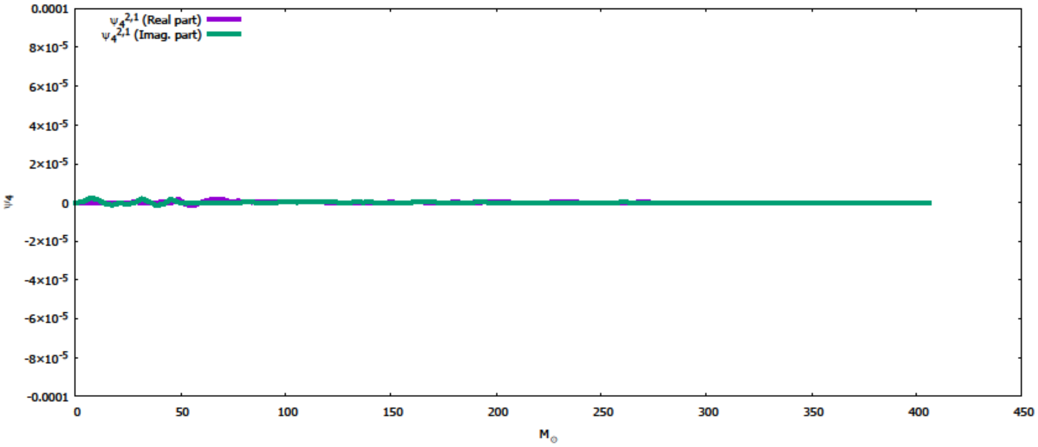
During the inspiral of the BNS, gravitational waves of various modes are produced. What follows is the release of gravitational waves of various modes produced during the inspiral of neutron stars measured at $45\ M_{\odot}$ or approximately 67 km.

Figure 5: Gravitational wave strength of different rotation modes during case 2 merger

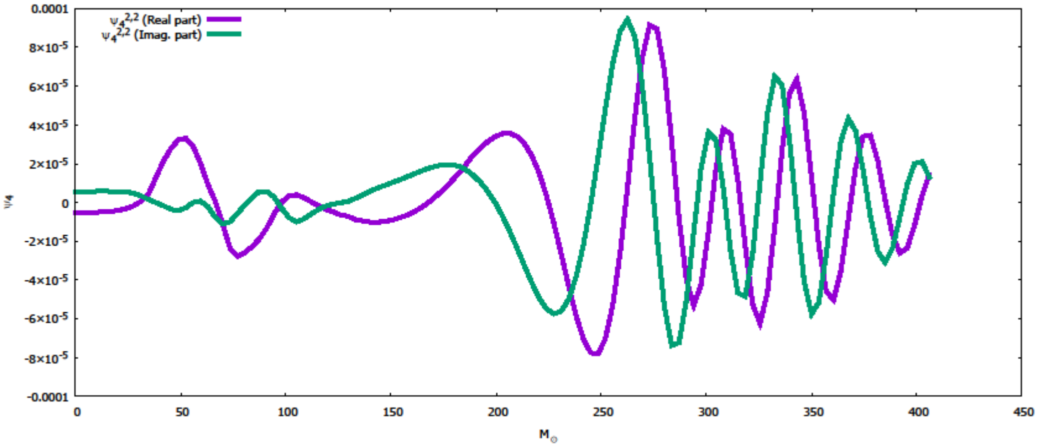




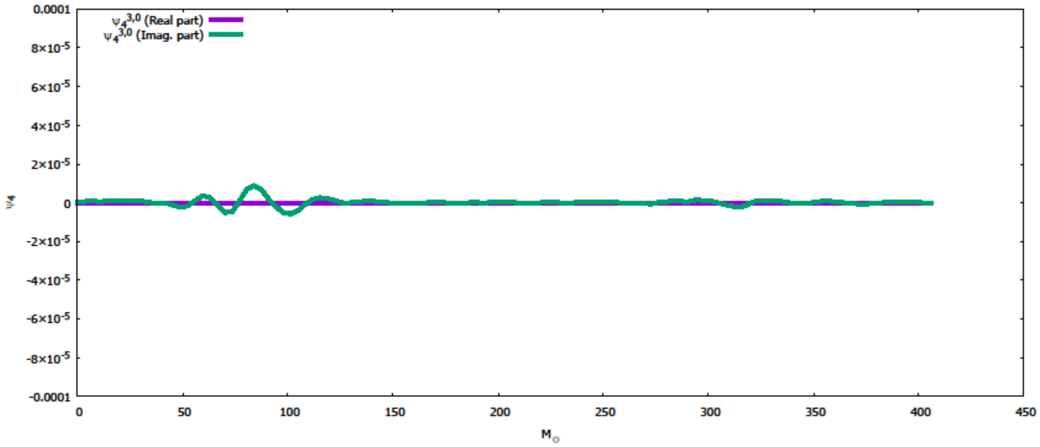
$\psi_4^{2,0}$ (Real part)
$\psi_4^{2,0}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

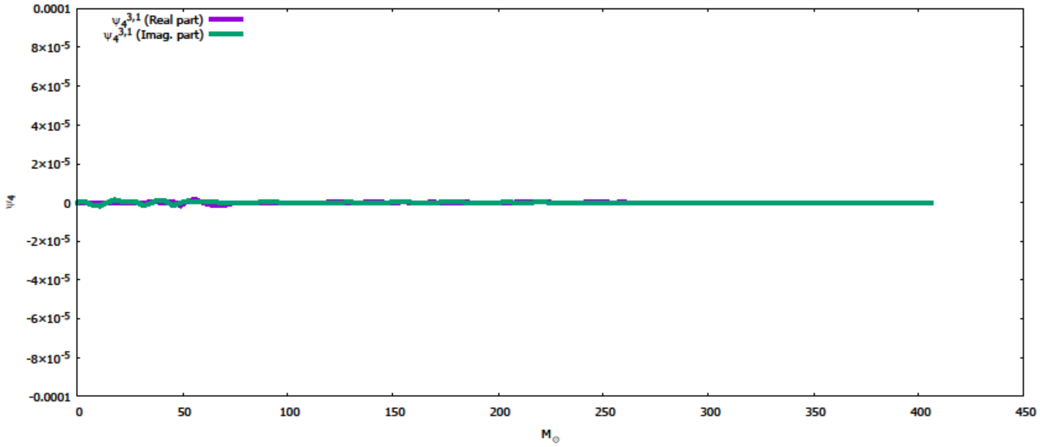


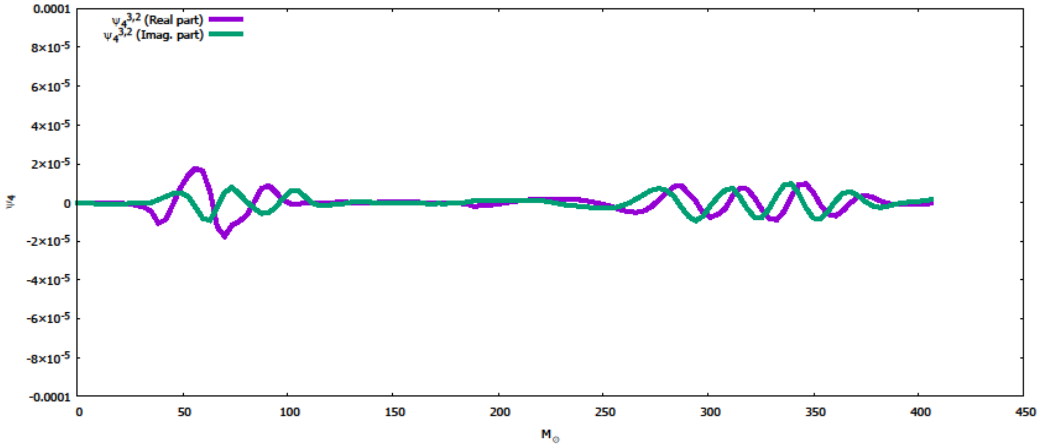
$\psi_4^{2,1}$ (Real part)
$\psi_4^{2,1}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

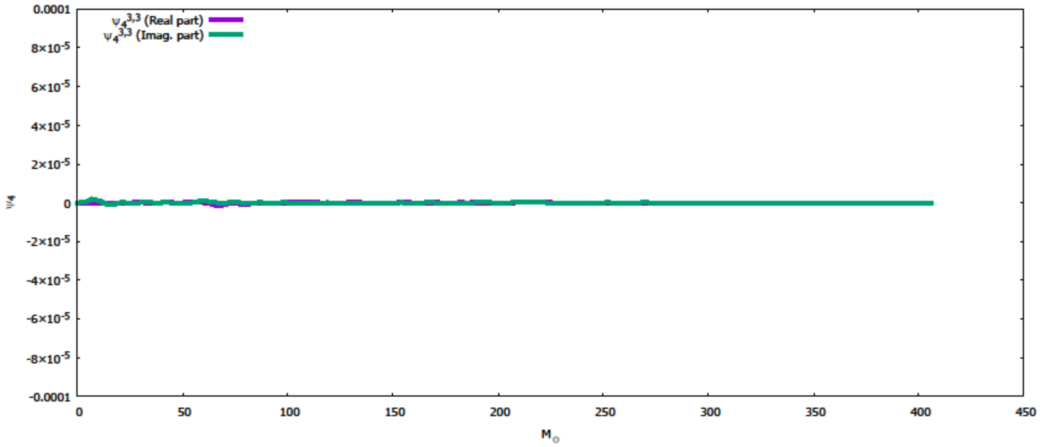


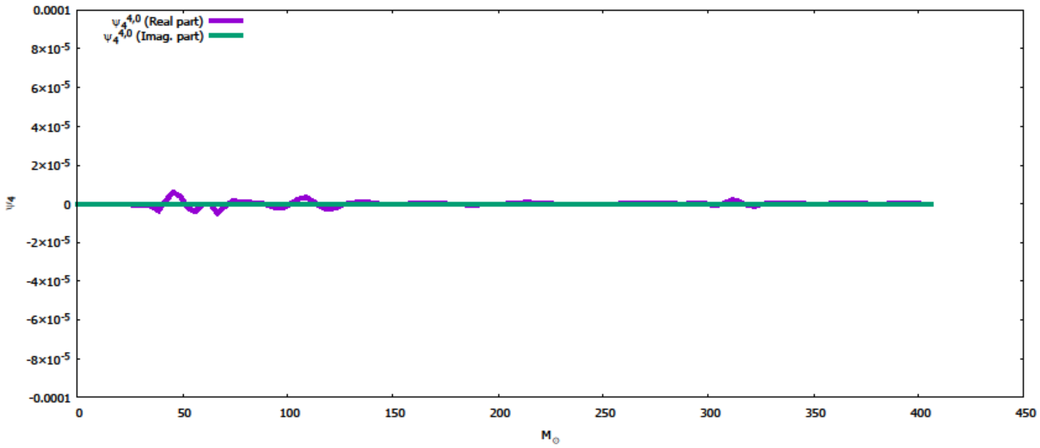
$\psi_4^{2,2}$ (Real part)
$\psi_4^{2,2}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

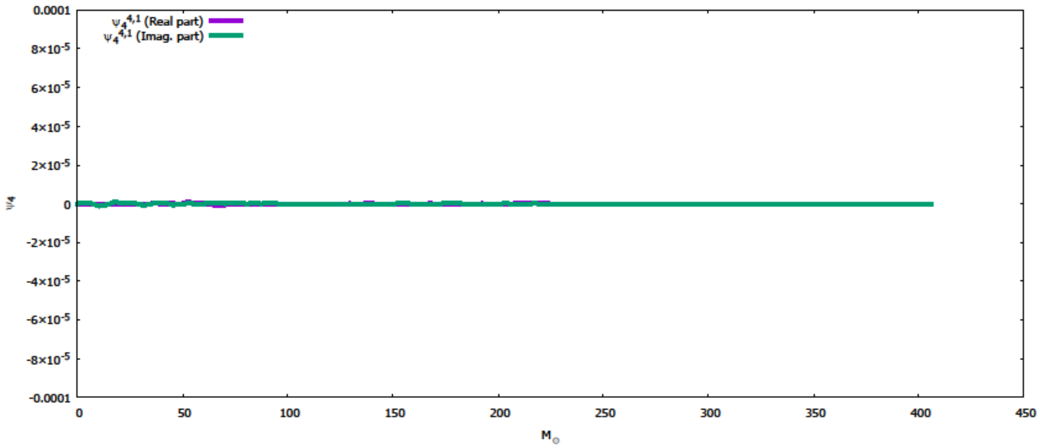

$\psi_4^{3,0}$ (Real part)
$\psi_4^{3,0}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

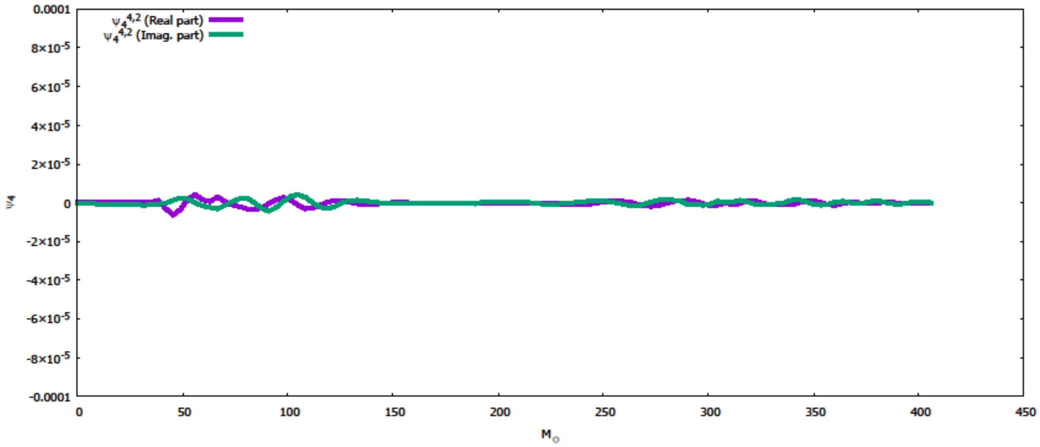

$\psi_4^{3,1}$ (Real part)
$\psi_4^{3,1}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

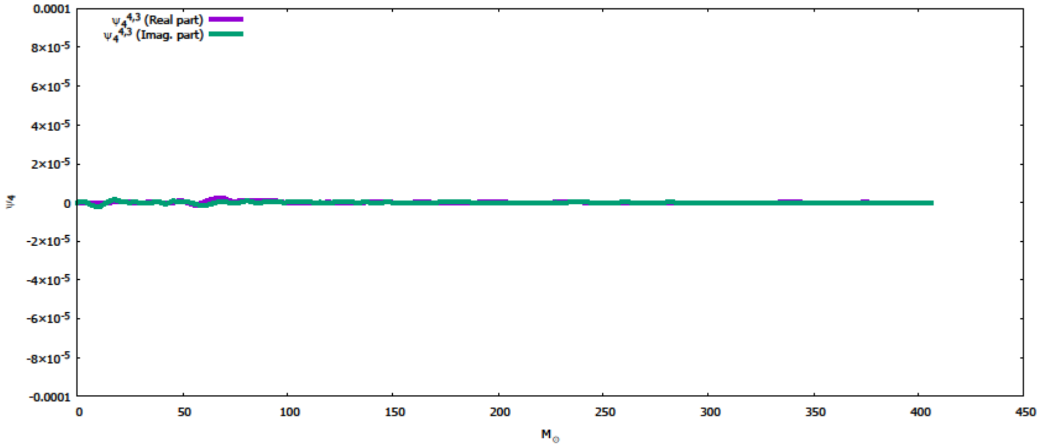

$\psi_4^{3,2}$ (Real part)
$\psi_4^{3,2}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

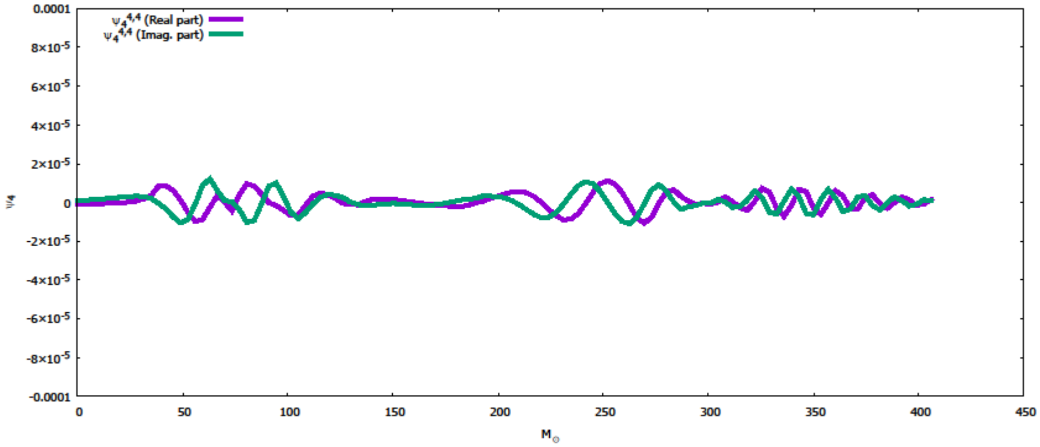

ψ4^3,3 (Real part)
ψ4^3,3 (Imag. part)
ψ4
0.0001
8×10^-5
6×10^-5
4×10^-5
2×10^-5
0
-2×10^-5
-4×10^-5
-6×10^-5
-8×10^-5
-0.0001
0
50
100
150
200
250
300
350
400
450
M⊙

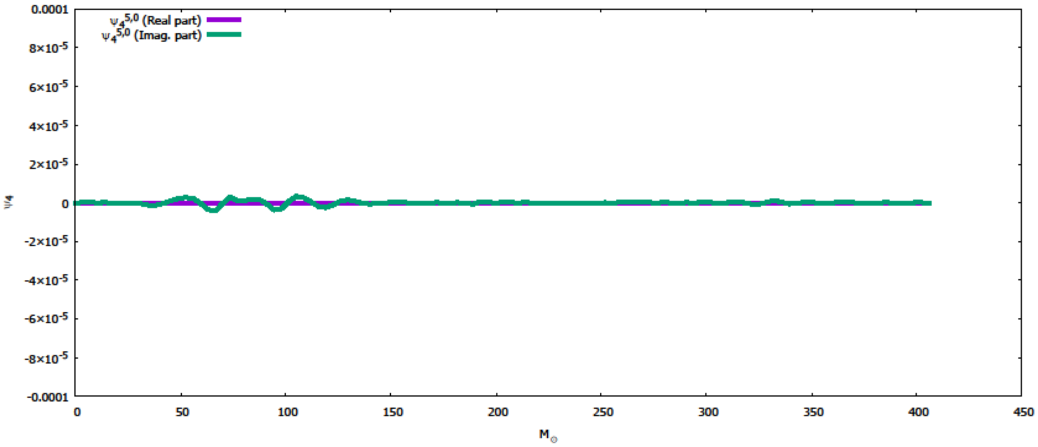

ψ4^4,0 (Real part)
ψ4^4,0 (Imag. part)
ψ4
0.0001
8×10^-5
6×10^-5
4×10^-5
2×10^-5
0
-2×10^-5
-4×10^-5
-6×10^-5
-8×10^-5
-0.0001
0
50
100
150
200
250
300
350
400
450
M⊙

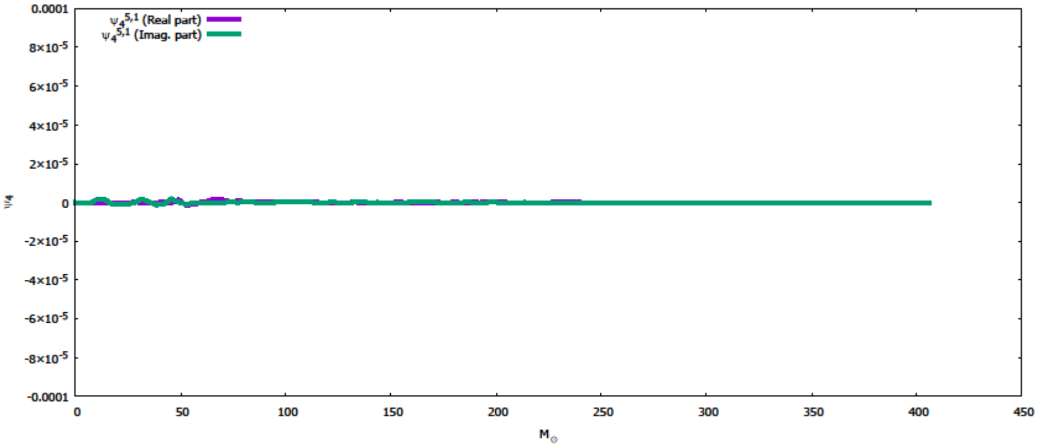

ψ4^4,1 (Real part)
ψ4^4,1 (Imag. part)
ψ4
0.0001
8×10^-5
6×10^-5
4×10^-5
2×10^-5
0
-2×10^-5
-4×10^-5
-6×10^-5
-8×10^-5
-0.0001
0
50
100
150
200
250
300
350
400
450
M⊙

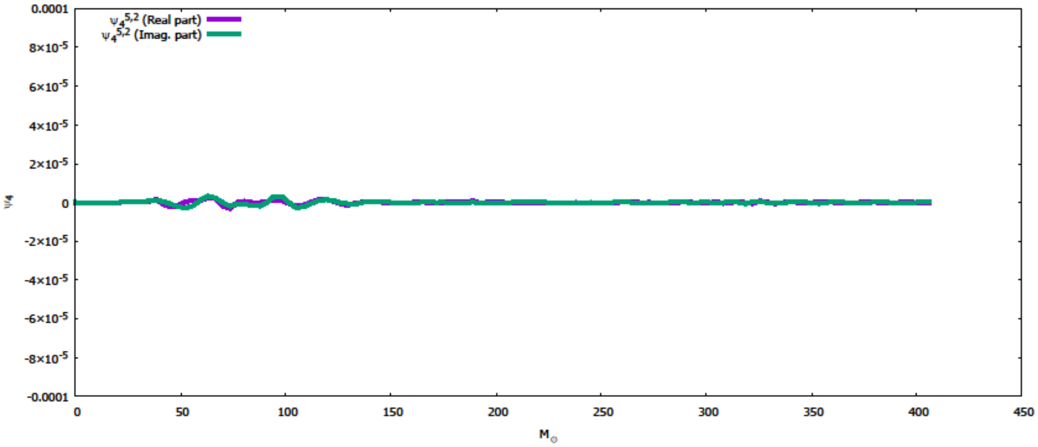

$\psi_4^{4,2}$ (Real part)
$\psi_4^{4,2}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

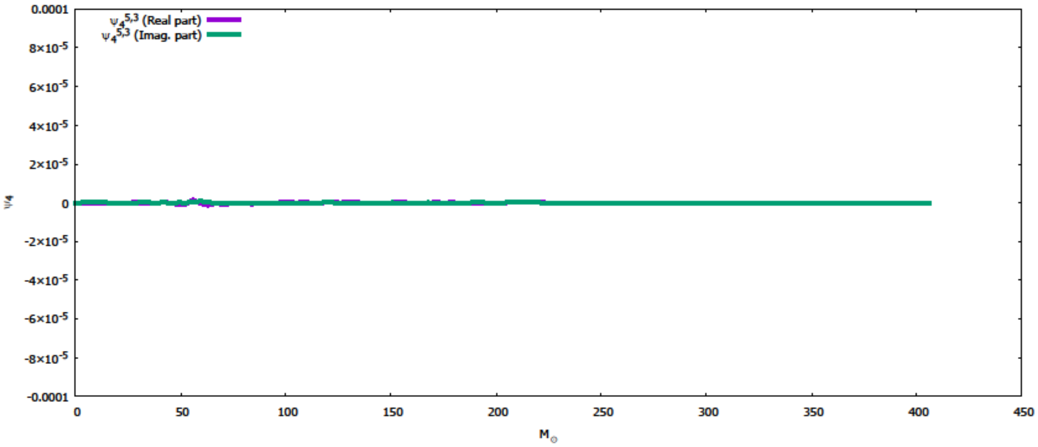

$\psi_4^{4,3}$ (Real part)
$\psi_4^{4,3}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

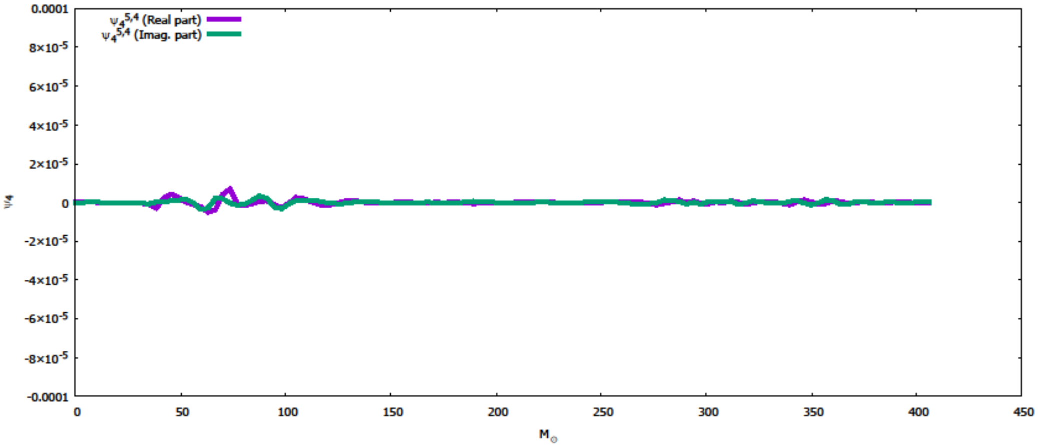

$\psi_4^{4,4}$ (Real part)
$\psi_4^{4,4}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

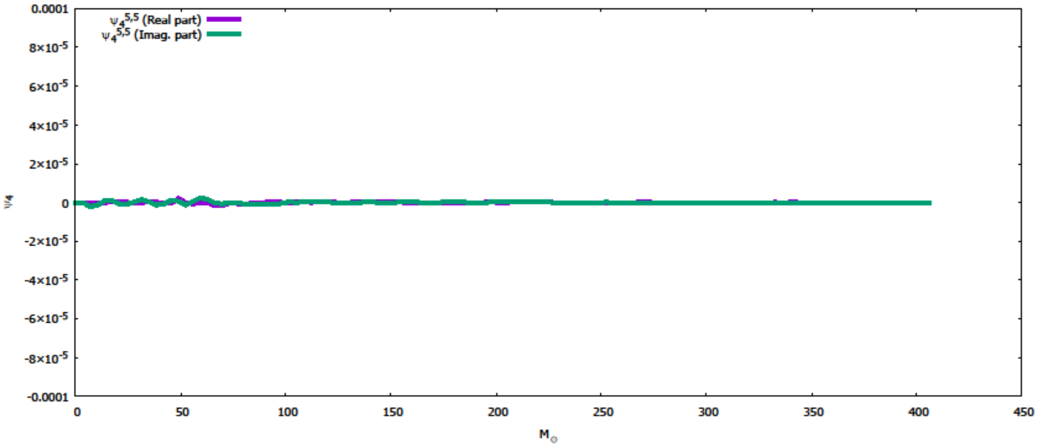

$\psi_4^{5,0}$ (Real part)
$\psi_4^{5,0}$ (Imag. part)
ψ_4
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
0
50
100
150
200
250
300
350
400
450
$M_{\odot}$

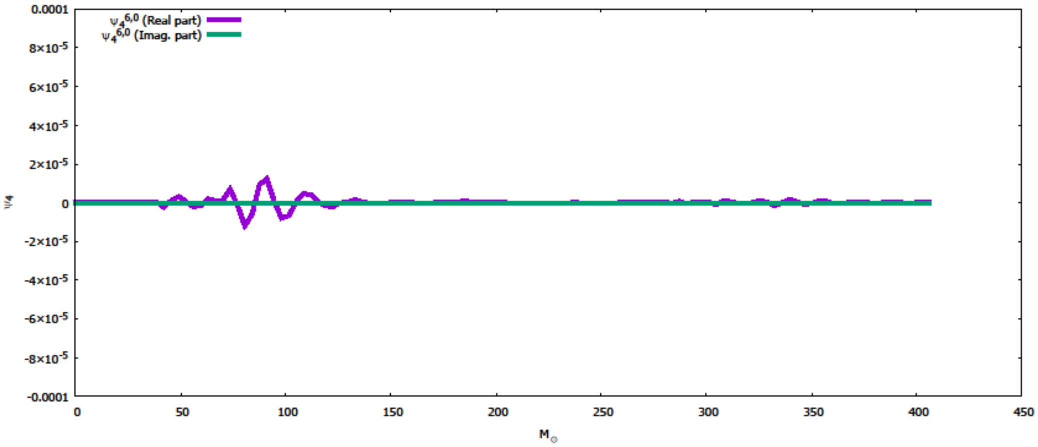

$\psi_4^{5,1}$ (Real part)
$\psi_4^{5,1}$ (Imag. part)
ψ_4
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
0
50
100
150
200
250
300
350
400
450
$M_{\odot}$

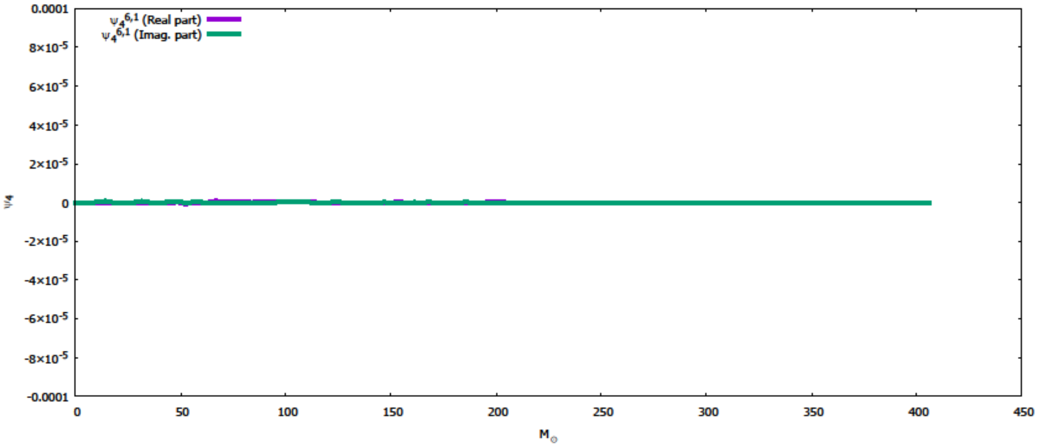

$\psi_4^{5,2}$ (Real part)
$\psi_4^{5,2}$ (Imag. part)
ψ_4
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
0
50
100
150
200
250
300
350
400
450
$M_{\odot}$

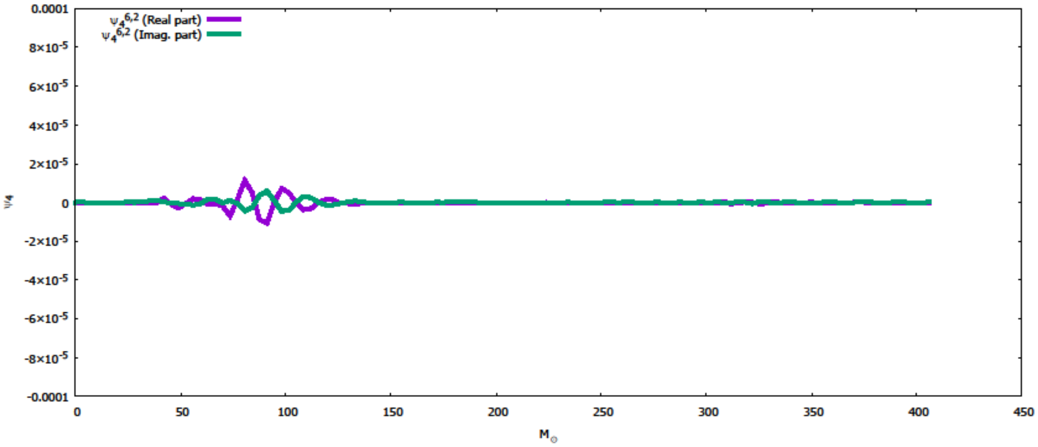

$\psi_4^{5,3}$ (Real part)
$\psi_4^{5,3}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

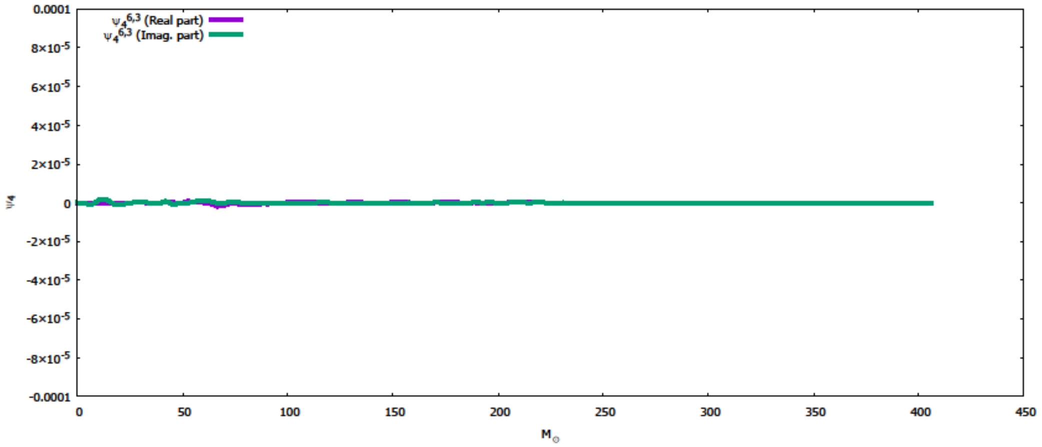

$\psi_4^{5,4}$ (Real part)
$\psi_4^{5,4}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$


$\psi_4^{5,5}$ (Real part)
$\psi_4^{5,5}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$

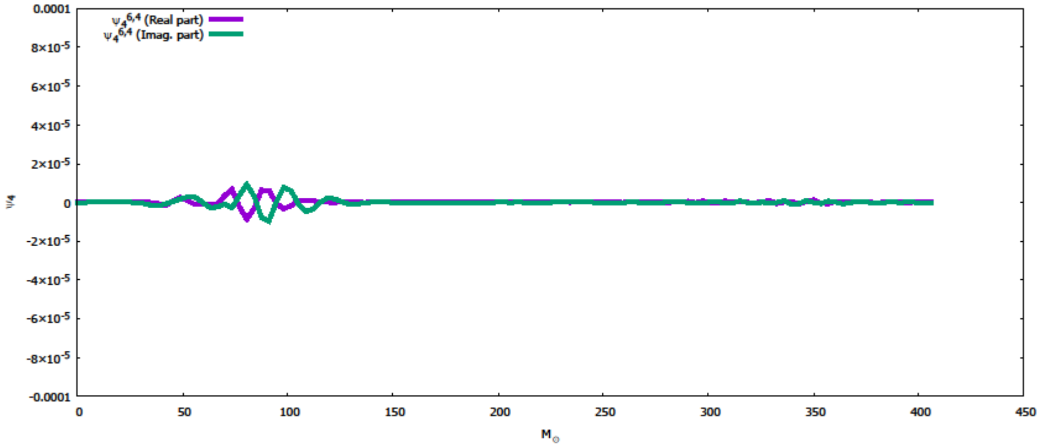

ψ4 6,0 (Real part)
ψ4 6,0 (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ4
0
50
100
150
200
250
300
350
400
450
M⊙


ψ4 6,1 (Real part)
ψ4 6,1 (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ4
0
50
100
150
200
250
300
350
400
450
M⊙

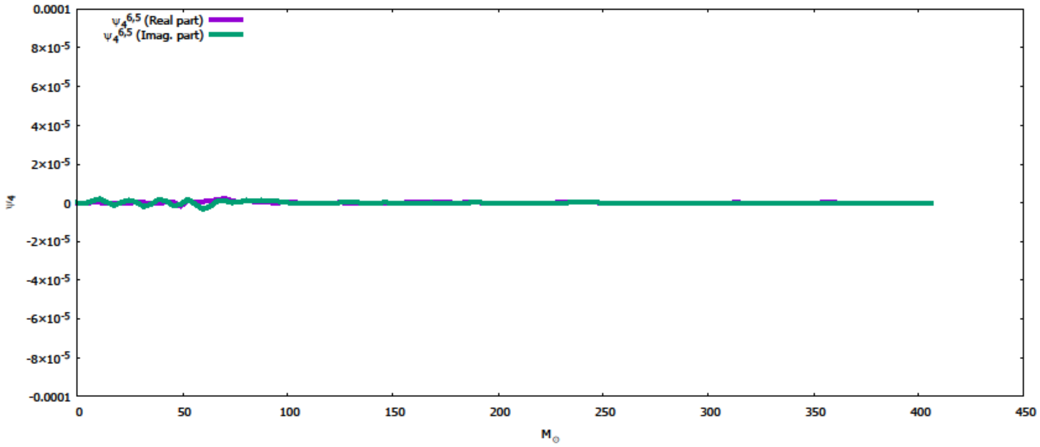

ψ4 6,2 (Real part)
ψ4 6,2 (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ4
0
50
100
150
200
250
300
350
400
450
M⊙

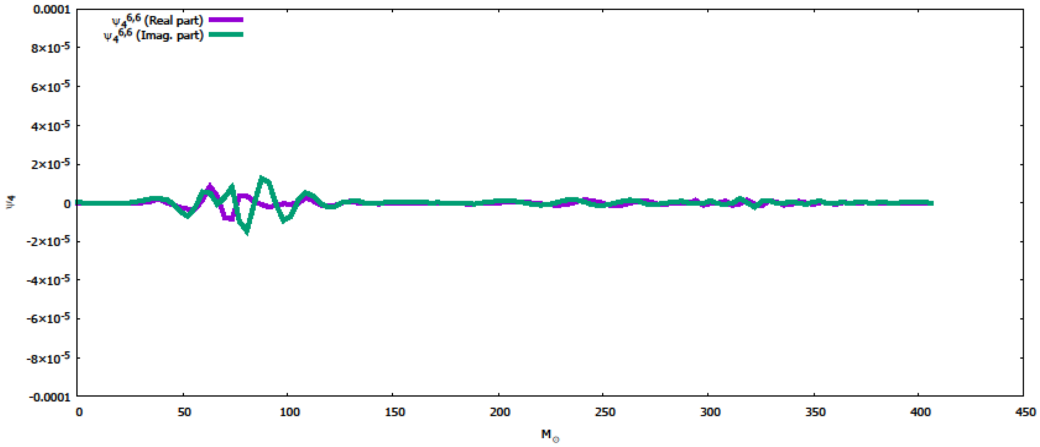


$\psi_4^{6,3}$ (Real part)
$\psi_4^{6,3}$ (Imag. part)
0.0001
8×10⁻⁵
6×10⁻⁵
4×10⁻⁵
2×10⁻⁵
0
-2×10⁻⁵
-4×10⁻⁵
-6×10⁻⁵
-8×10⁻⁵
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$



$\psi_4^{6,4}$ (Real part)
$\psi_4^{6,4}$ (Imag. part)
0.0001
8×10⁻⁵
6×10⁻⁵
4×10⁻⁵
2×10⁻⁵
0
-2×10⁻⁵
-4×10⁻⁵
-6×10⁻⁵
-8×10⁻⁵
-0.0001
ψ_4
0
50
100
150
200
250
300
350
400
450
$M_\odot$





As seen above, most of the modes of gravitational waves are zero with the $\Psi_4^{2,2}$ mode being the most prevalent and usually it is the only mode that is considered. What is seen is that each of the modes is approximately zero until approximately $40\ M_{\odot}$, which is when any gravitational wave that is produced at the start of the simulation would reach the radius where the gravitational waves are being measured. Then many of the modes will experience a perturbation of some kind, presumably from the metric becoming non-conformal. Only a few modes are nonzero once the simulation has started, and the initial wave dissipates away, namely the $\Psi_4^{2,2}$ mode. However, it can also be seen that the simulation was stopped too soon since the gravitational waves never completely return to zero in any of the modes. This is due to the travel time of the wave not being considered and further evolution of the system would need to be done to see the gravitational waves that are produced from the final merger of the system.

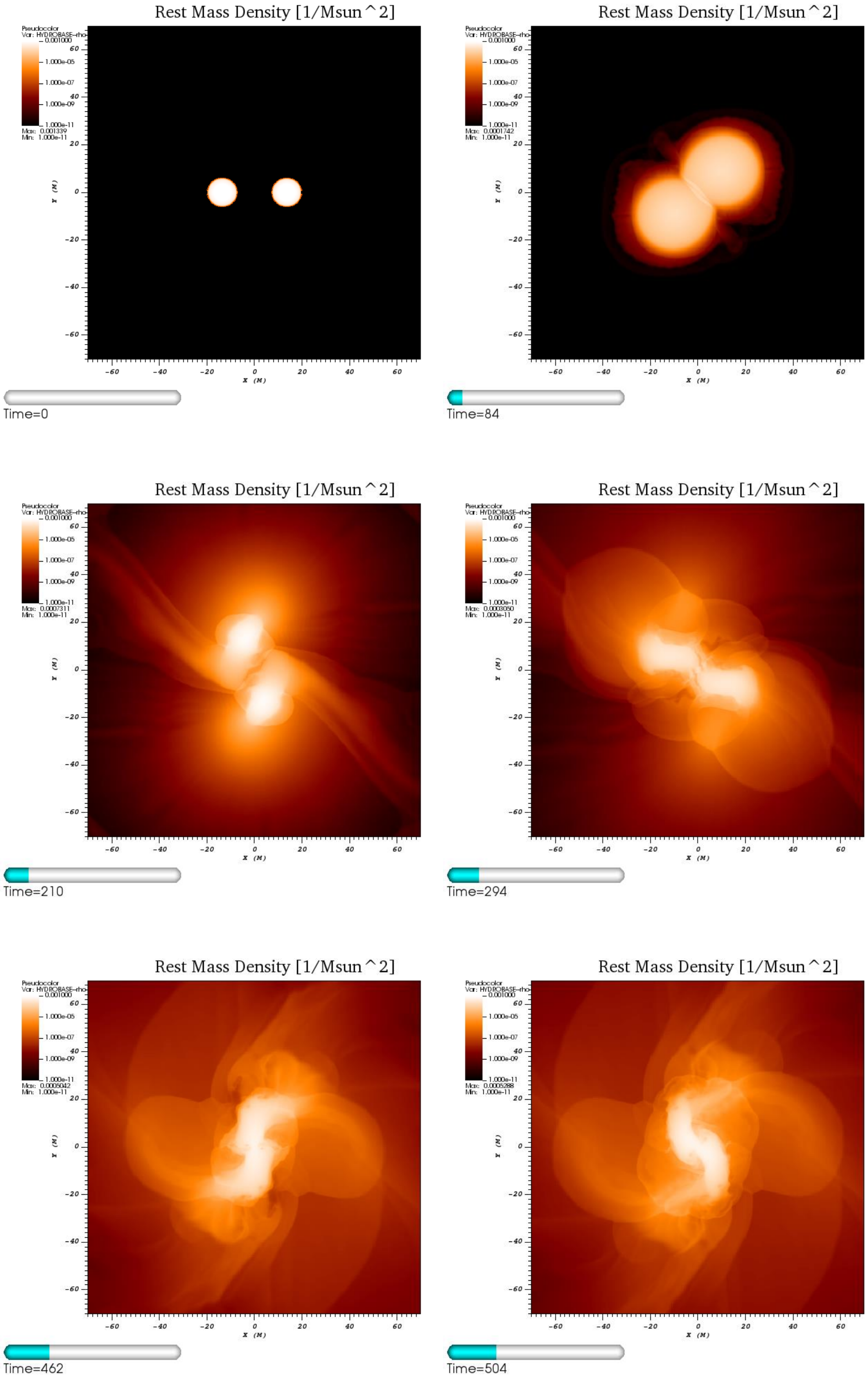
3.3. Case 3: 1.40 and 1.40 $M_{\odot}$

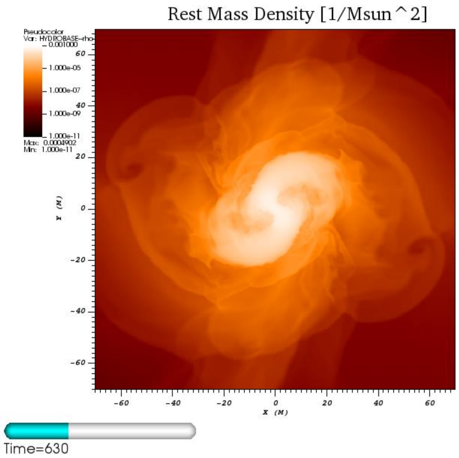
The third case consists of two neutron stars with each containing the minimum mass of 1.40 $M_{\odot}$ starting at a separation of 40 km. This case gives a better understanding of the evolution of the system with a longer run-time, allowing for a better look at the end product.

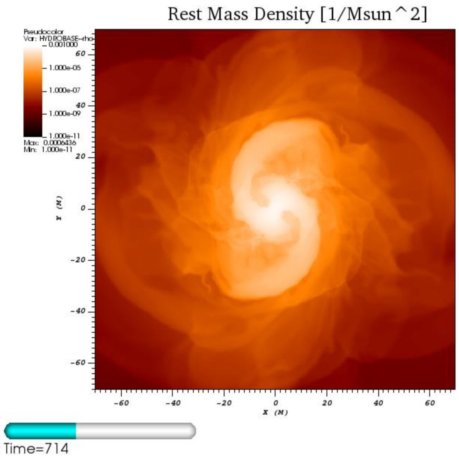
3.3.1. Matter Distribution

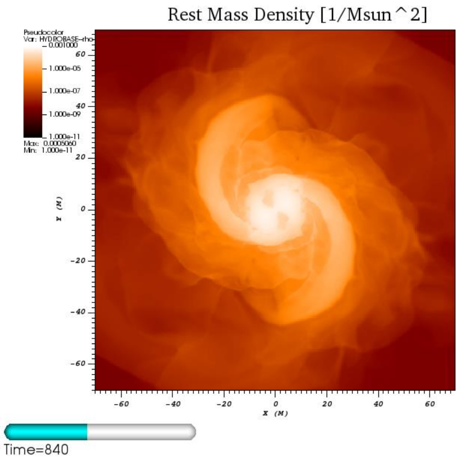
During the simulation of case 3, the neutron stars can be seen to start at their initial points and orbit around each other several times until they eventually merge. This simulation was able to run for a long enough time to give a good understanding of the evolution of this system as well as the final product that was produced during the merger.

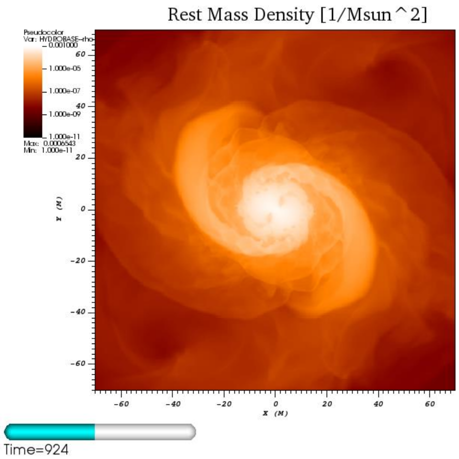
Figure 6: Density distribution during case 3 merger

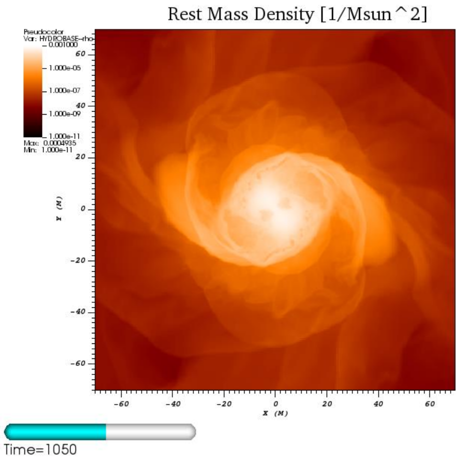


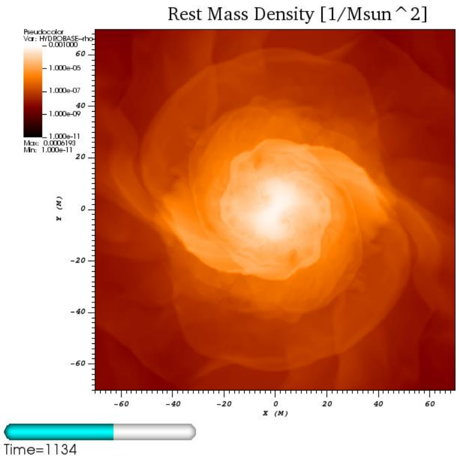

Rest Mass Density [1/Msun^2]
Pseudocolor
Var: HYDROBASE--rho
0.001000
1.000e-05
1.000e-07
1.000e-09
1.000e-11
Max: 0.0004902
Min: 1.000e-11
Y (M)
60
40
20
0
-20
-40
-60
X (M)
-60
-40
-20
0
20
40
60
Time=630


Rest Mass Density [1/Msun^2]
Pseudocolor
Var: HYDROBASE--rho
0.001000
1.000e-05
1.000e-07
1.000e-09
1.000e-11
Max: 0.0006436
Min: 1.000e-11
Y (M)
60
40
20
0
-20
-40
-60
X (M)
-60
-40
-20
0
20
40
60
Time=714

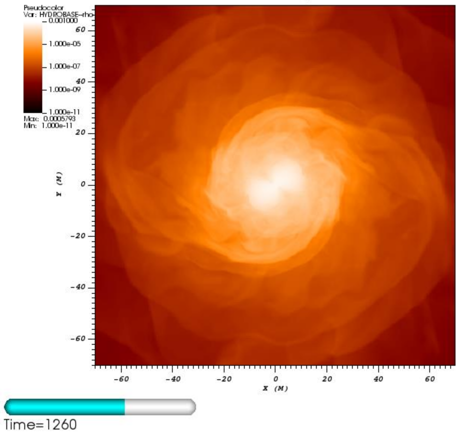

Rest Mass Density [1/Msun^2]
Pseudocolor
Var: HYDROBASE--rho
0.001000
1.000e-05
1.000e-07
1.000e-09
1.000e-11
Max: 0.0005060
Min: 1.000e-11
Y (M)
60
40
20
0
-20
-40
-60
X (M)
-60
-40
-20
0
20
40
60
Time=840


Rest Mass Density [1/Msun^2]
Pseudocolor
Var: HYDROBASE--rho
0.001000
1.000e-05
1.000e-07
1.000e-09
1.000e-11
Max: 0.0006543
Min: 1.000e-11
Y (M)
60
40
20
0
-20
-40
-60
X (M)
-60
-40
-20
0
20
40
60
Time=924

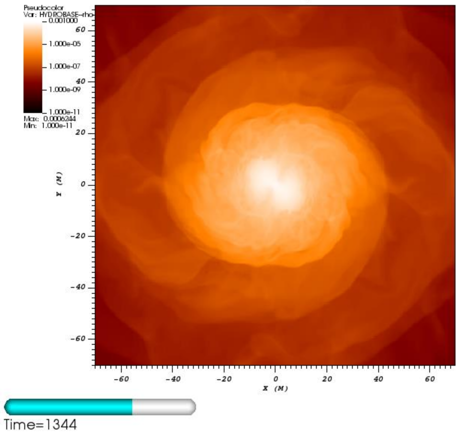

Rest Mass Density [1/Msun^2]
Pseudocolor
Var: HYDROBASE--rho
0.001000
1.000e-05
1.000e-07
1.000e-09
1.000e-11
Max: 0.0004935
Min: 1.000e-11
Y (M)
60
40
20
0
-20
-40
-60
X (M)
-60
-40
-20
0
20
40
60
Time=1050


Rest Mass Density [1/Msun^2]
Pseudocolor
Var: HYDROBASE--rho
0.001000
1.000e-05
1.000e-07
1.000e-09
1.000e-11
Max: 0.0006193
Min: 1.000e-11
Y (M)
60
40
20
0
-20
-40
-60
X (M)
-60
-40
-20
0
20
40
60
Time=1134

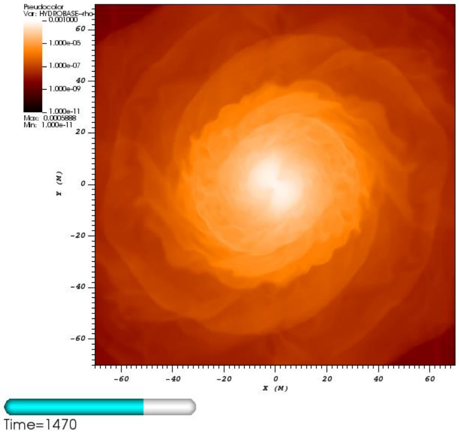
Rest Mass Density [1/Msun^2]



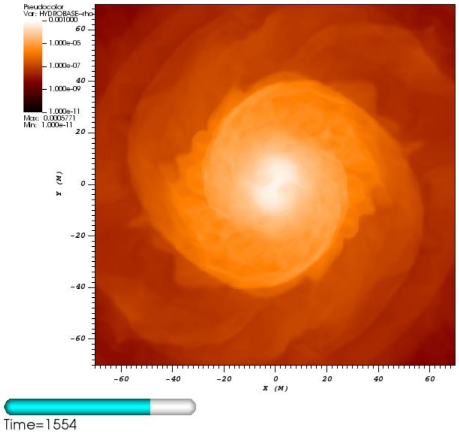
Rest Mass Density [1/Msun^2]

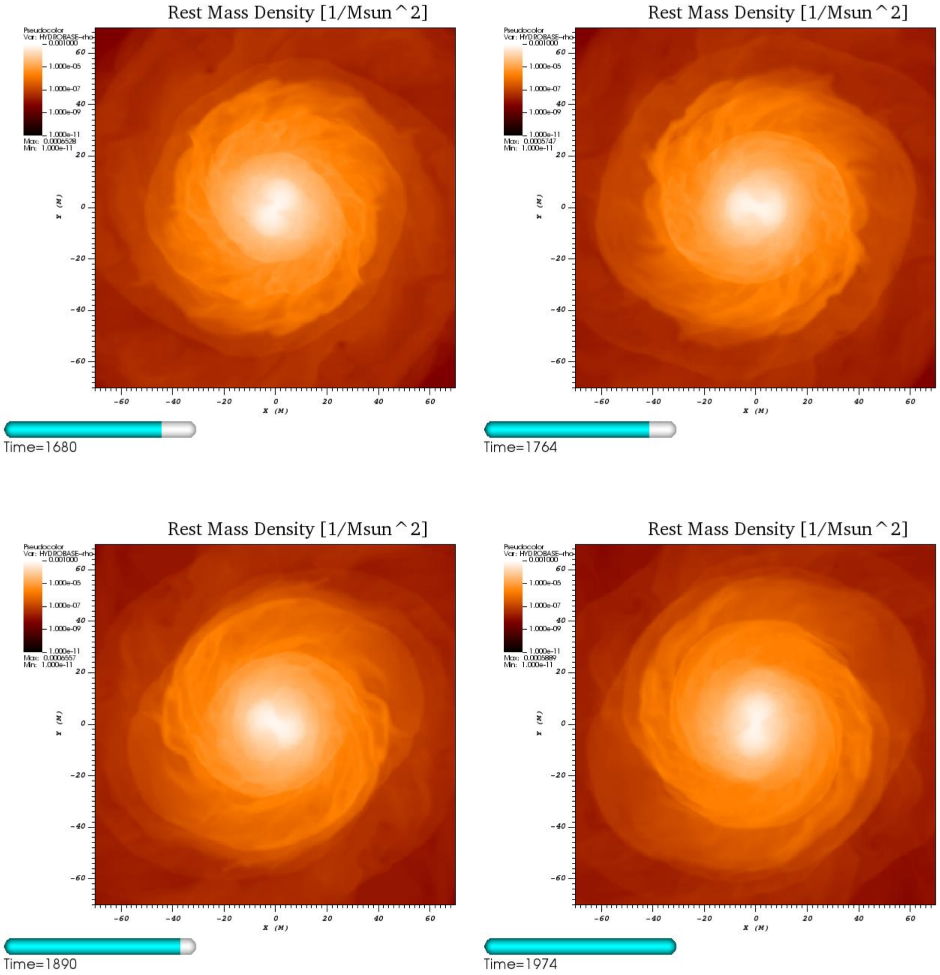


Rest Mass Density [1/Msun^2]



Rest Mass Density [1/Msun^2]



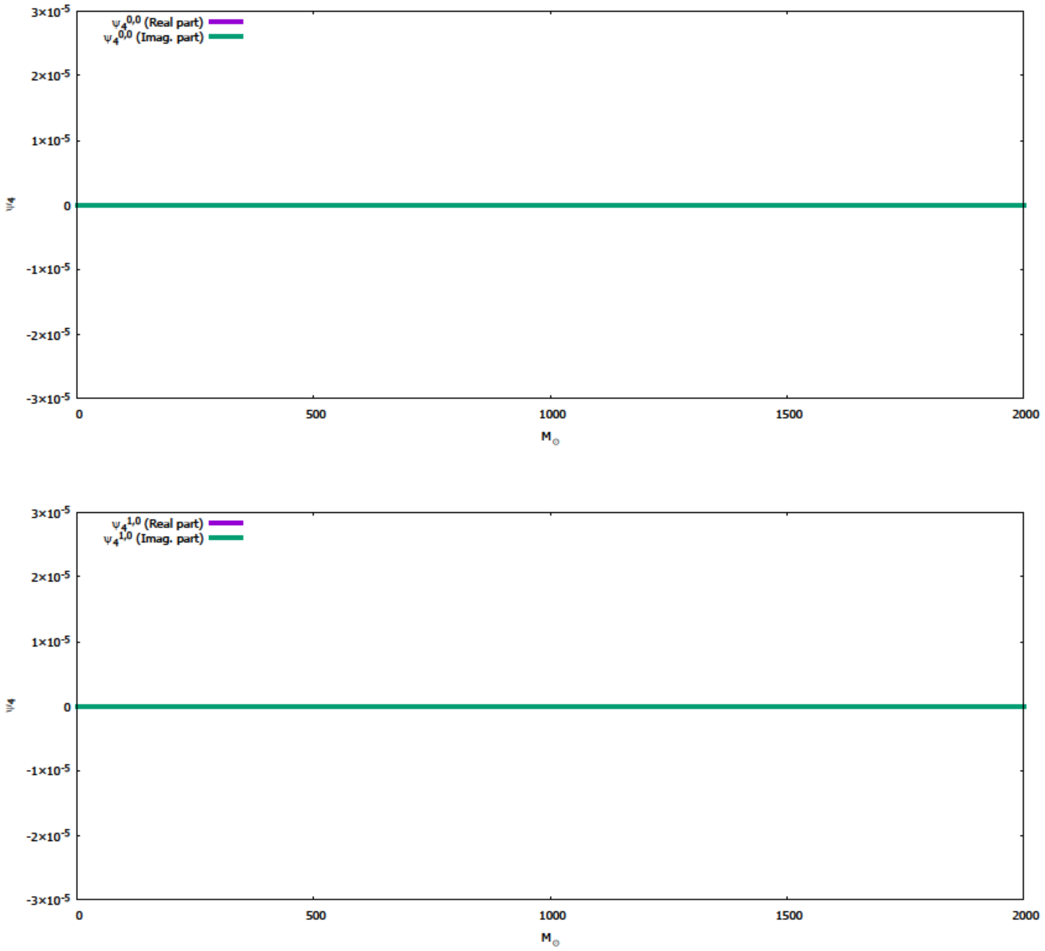


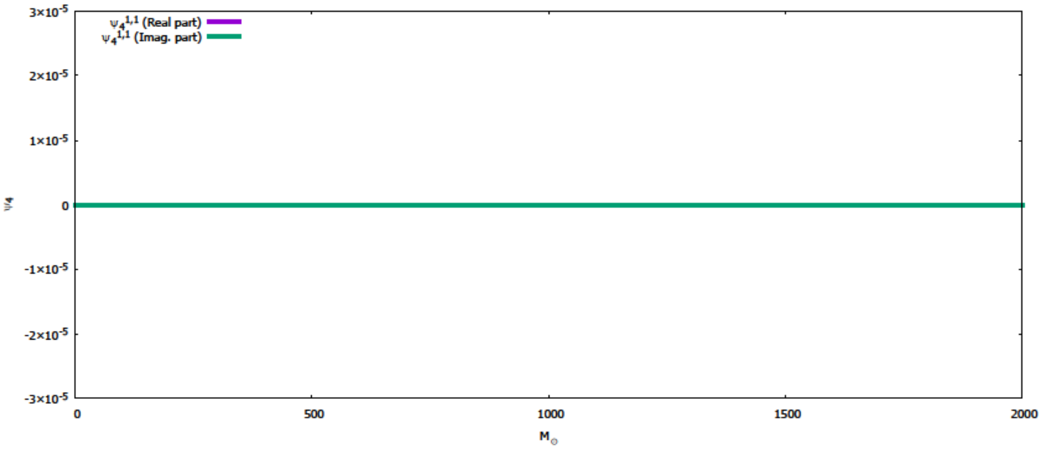
This case resulted in a longer running simulation, 1974 $M_\odot$ or about $9.73 \times 10^{-3}\ s$, with the formation of a hyper-massive neutron star (Ciolfi et al. 2017). A hyper-massive neutron star is a neutron star that has a greater mass than the theoretically allowed mass, the TOV mass (Sarracino & Eccles 1996), because it has enough angular momentum to counteract the collapse of the system into a black hole. This system will remain stable until it dissipates enough of the angular momentum through gravitational waves to collapse into a black hole. The lifetime of a hyper-massive neutron star is on the scale of about 1 s (Ciolfi et al. 2017).

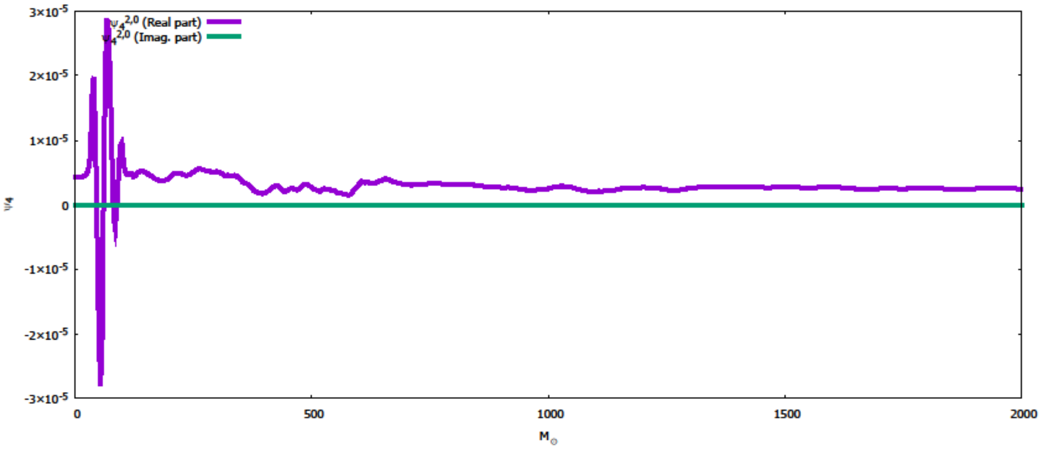
3.3.2. Gravitational Waves

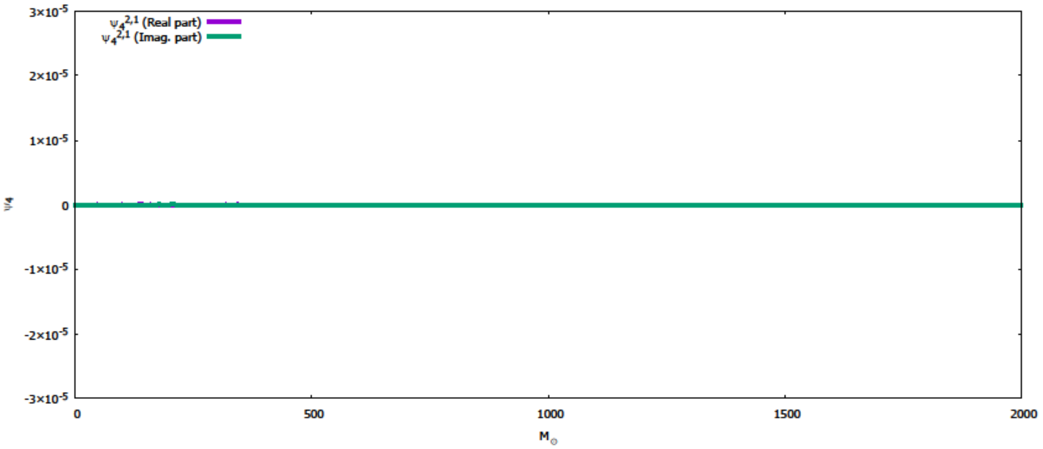
During the process of the inspiral of the BNS, gravitational waves of various modes are produced. What follows are the various modes of gravitational waves as a radius of 45 $M_{\odot}$ or approximately 67 km.

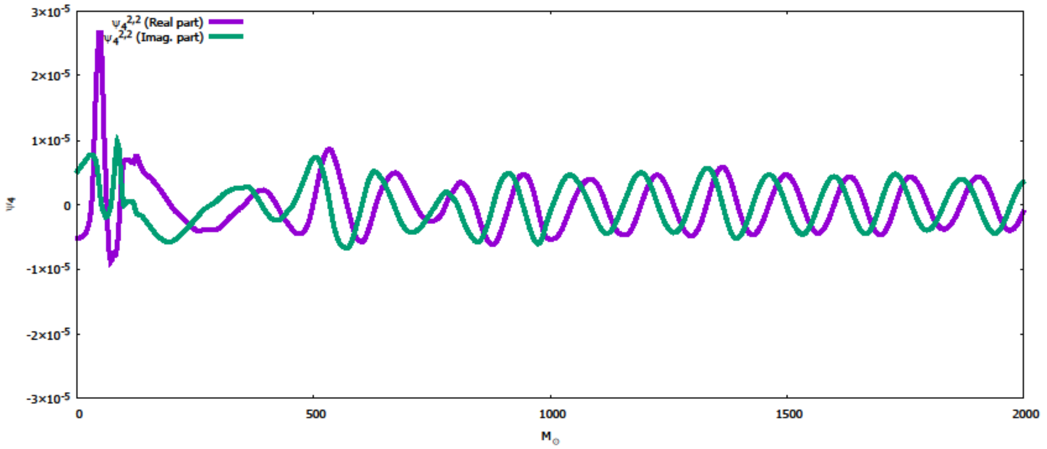
Figure 7: Gravitational wave strength of different rotation modes during case 4 merger

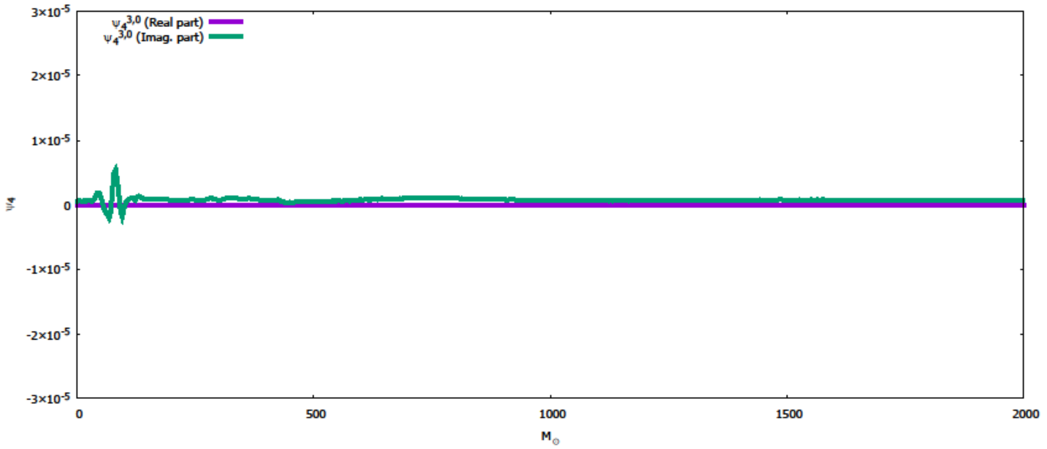


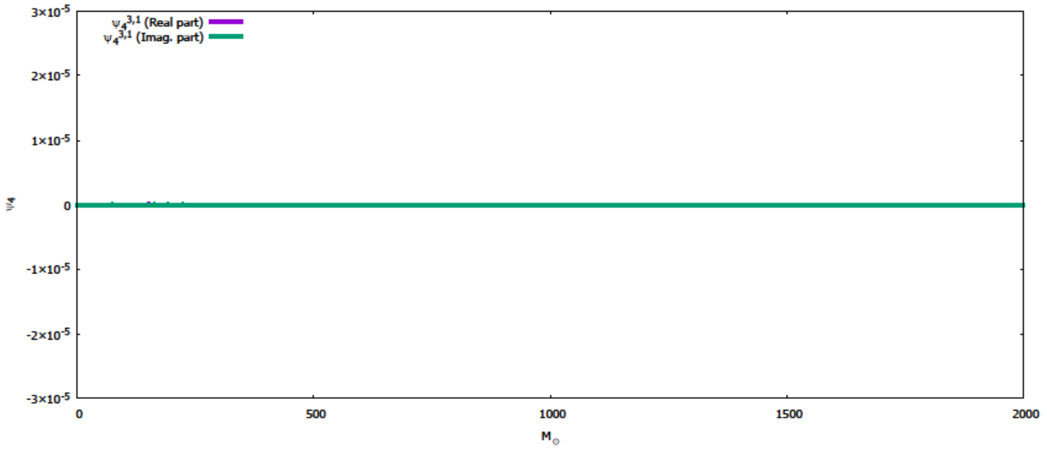

$\psi_4^{1,1}$ (Real part)
$\psi_4^{1,1}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

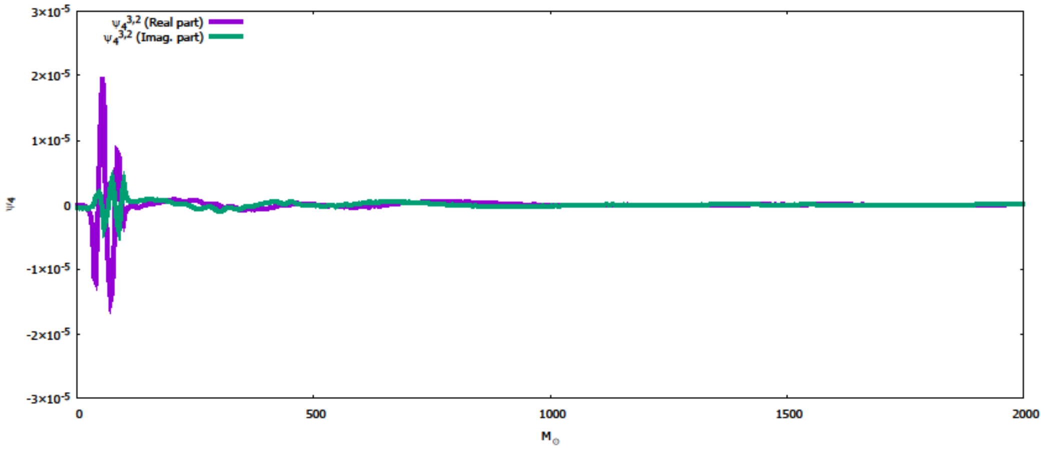

$\psi_4^{2,0}$ (Real part)
$\psi_4^{2,0}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

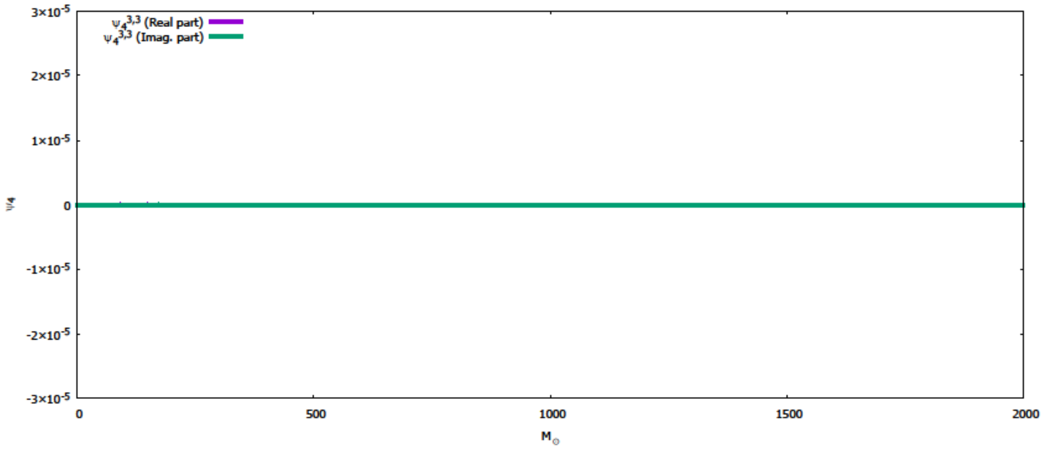

$\psi_4^{2,1}$ (Real part)
$\psi_4^{2,1}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

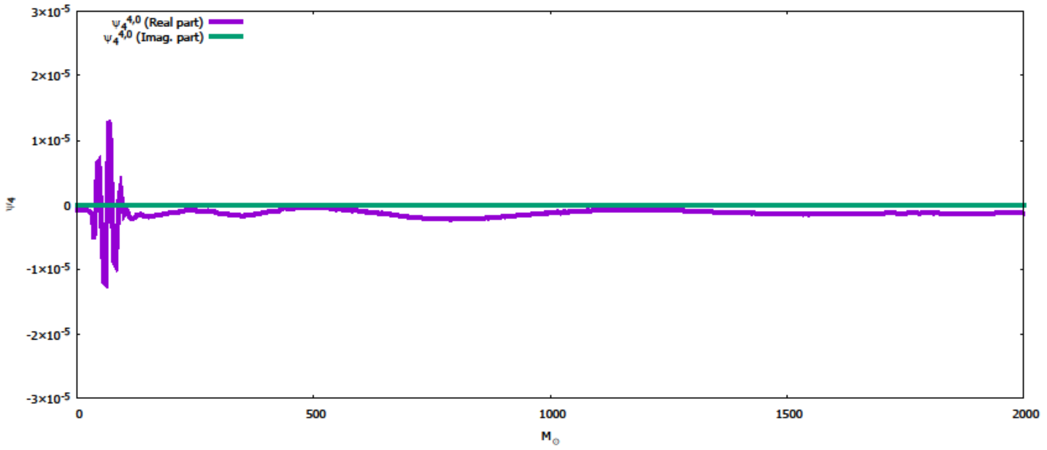

$\psi_4^{2,2}$ (Real part)
$\psi_4^{2,2}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

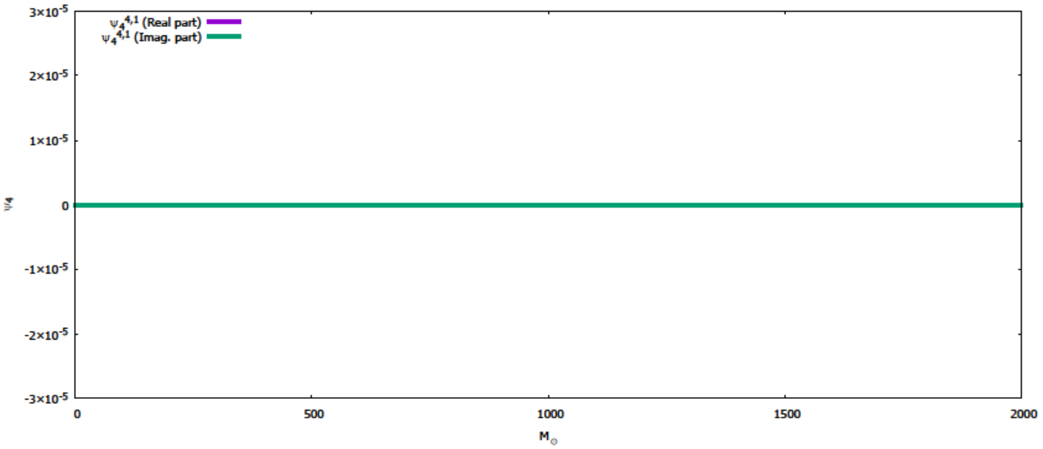

$\psi_4^{3,0}$ (Real part)
$\psi_4^{3,0}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

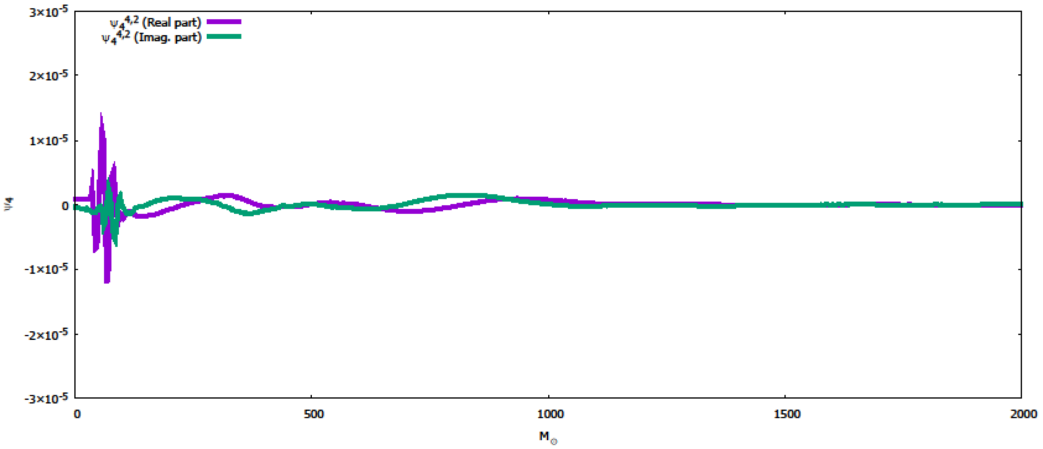

$\psi_4^{3,1}$ (Real part)
$\psi_4^{3,1}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

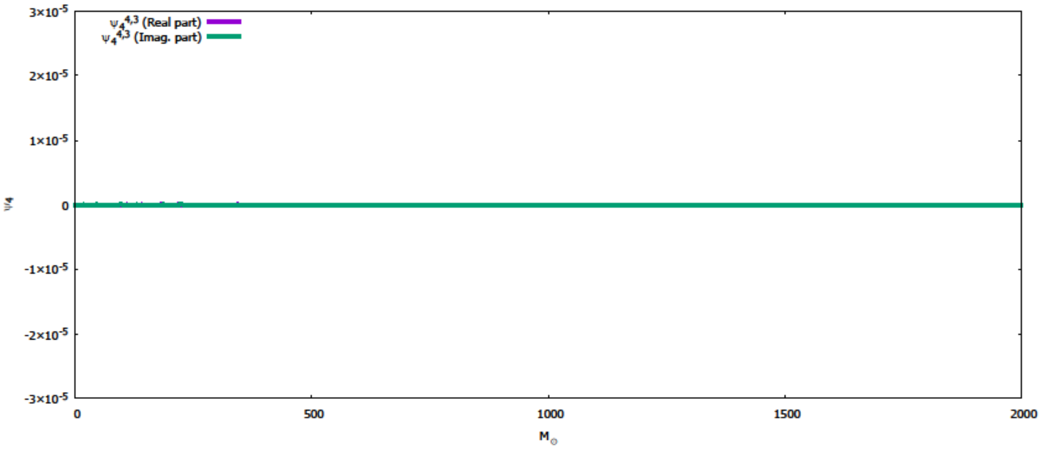

$\psi_4^{3,2}$ (Real part)
$\psi_4^{3,2}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

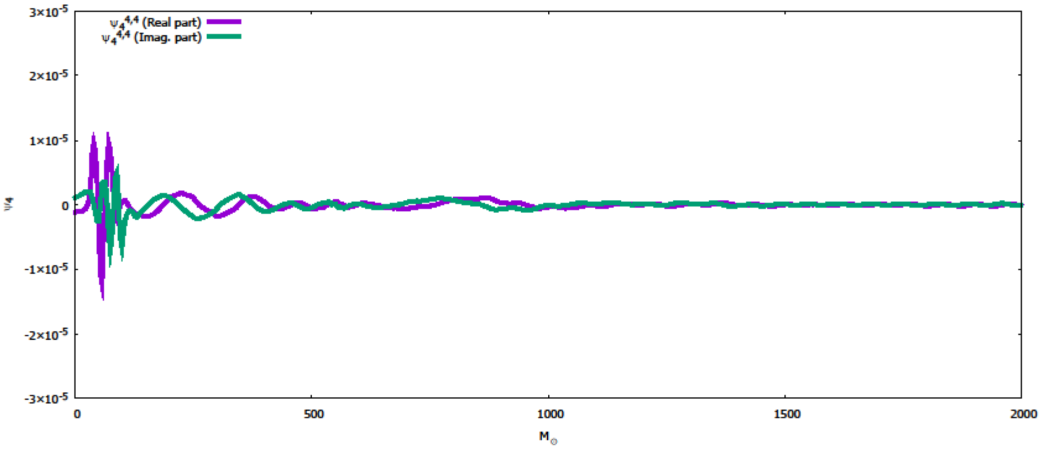

$\psi_4^{3,3}$ (Real part)
$\psi_4^{3,3}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

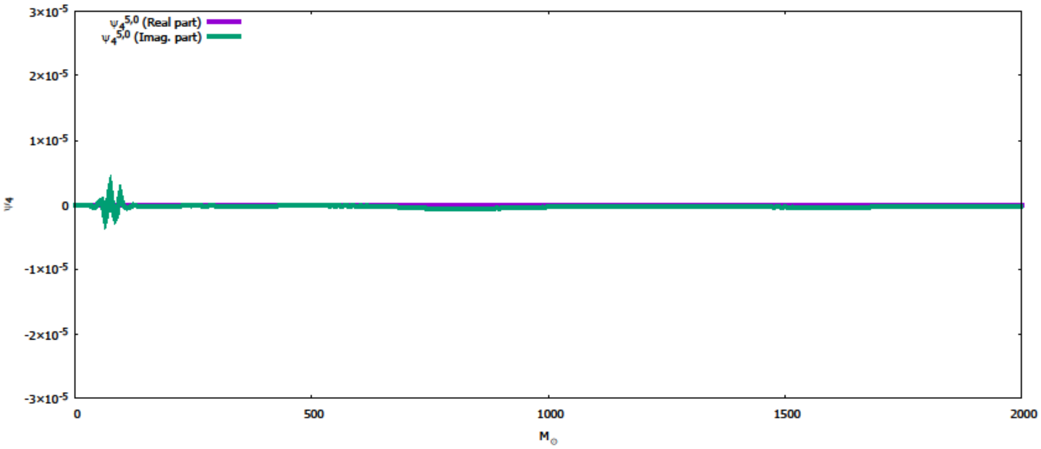

$\psi_4^{4,0}$ (Real part)
$\psi_4^{4,0}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

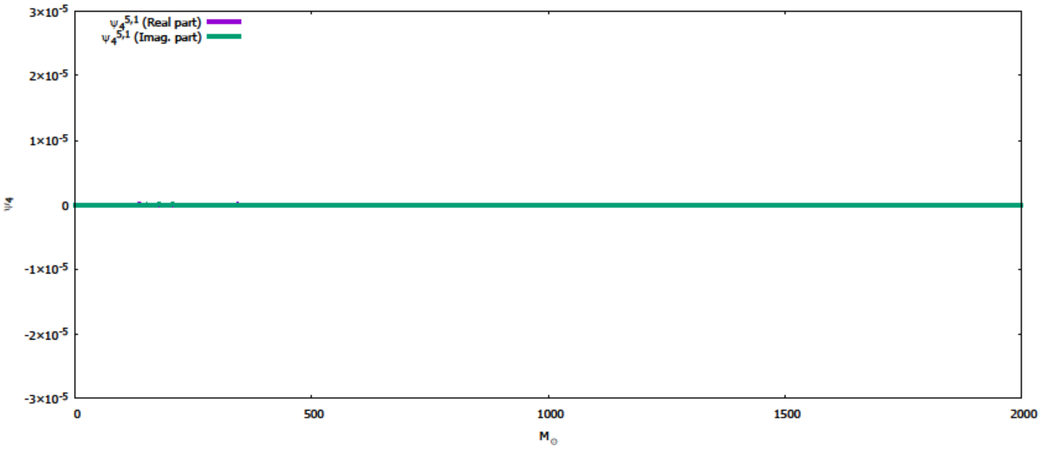

$\psi_4^{4,1}$ (Real part)
$\psi_4^{4,1}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

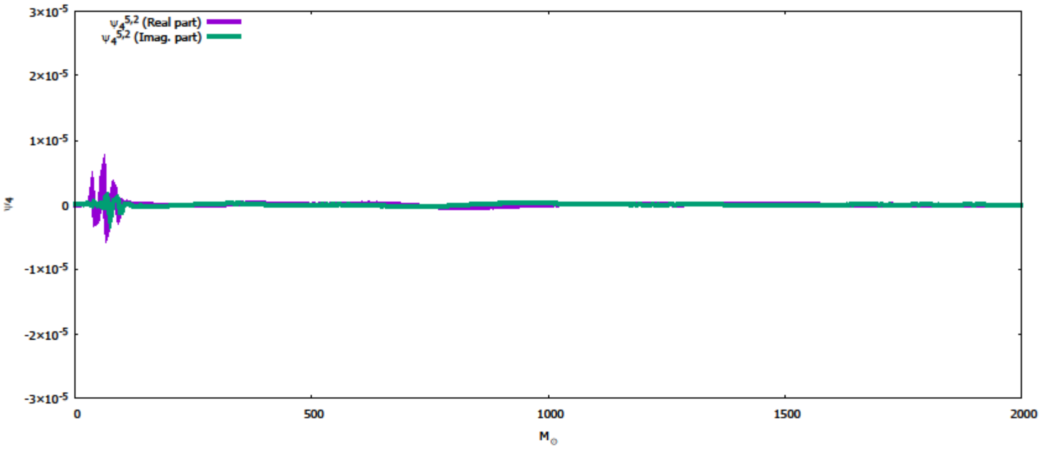

$\psi_4^{4,2}$ (Real part)
$\psi_4^{4,2}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$


$\psi_4^{4,3}$ (Real part)
$\psi_4^{4,3}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

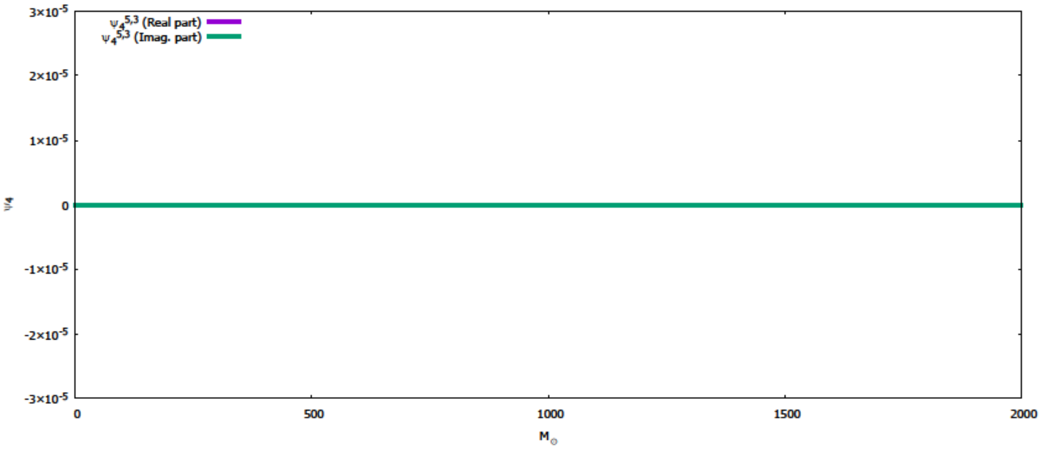

$\psi_4^{4,4}$ (Real part)
$\psi_4^{4,4}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$


$\psi_4^{5,0}$ (Real part)
$\psi_4^{5,0}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

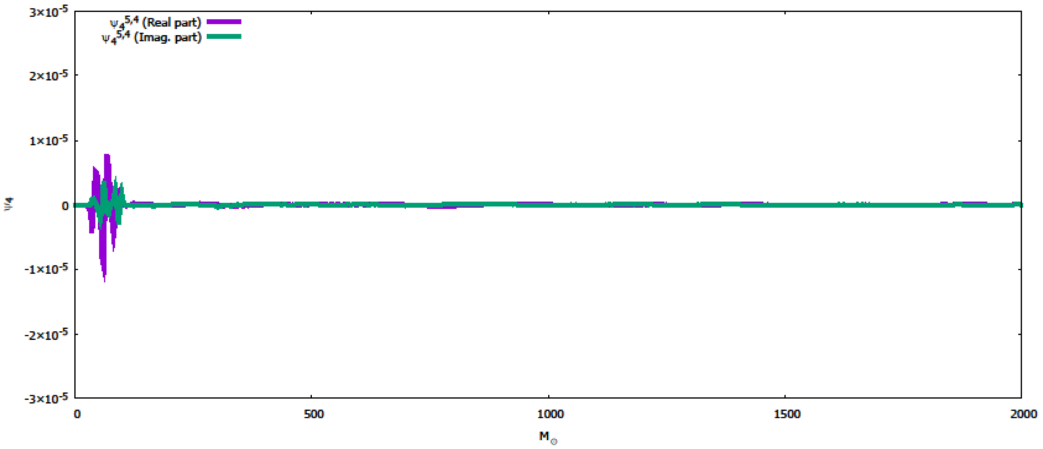

$\psi_4^{5,1}$ (Real part)
$\psi_4^{5,1}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$



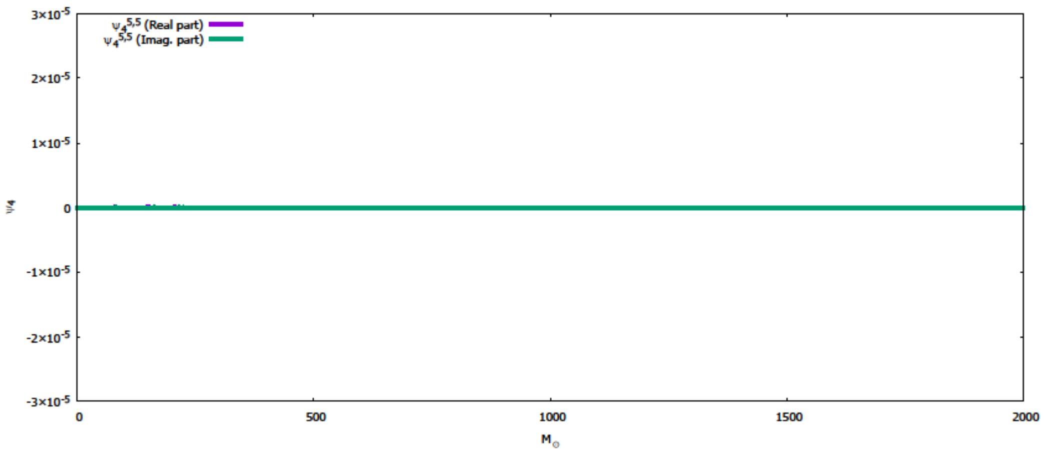
$\psi_4^{5,2}$ (Real part)
$\psi_4^{5,2}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

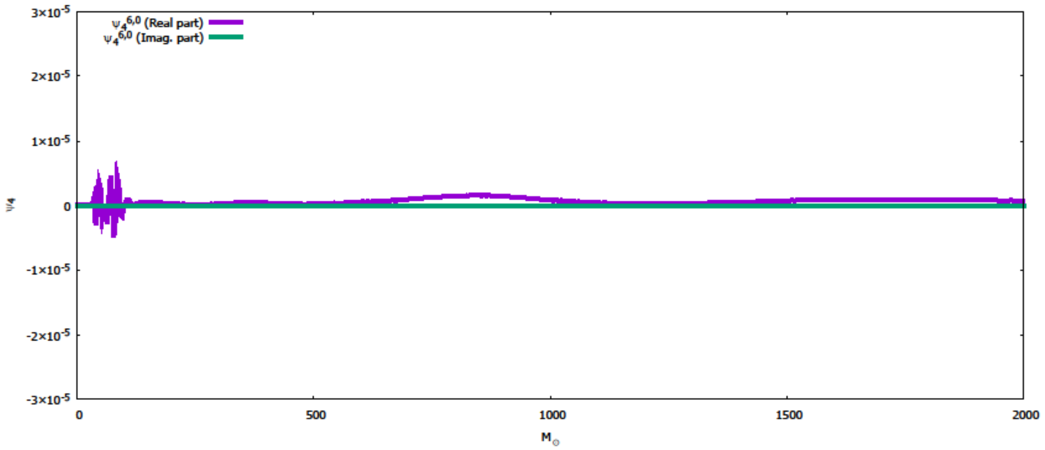


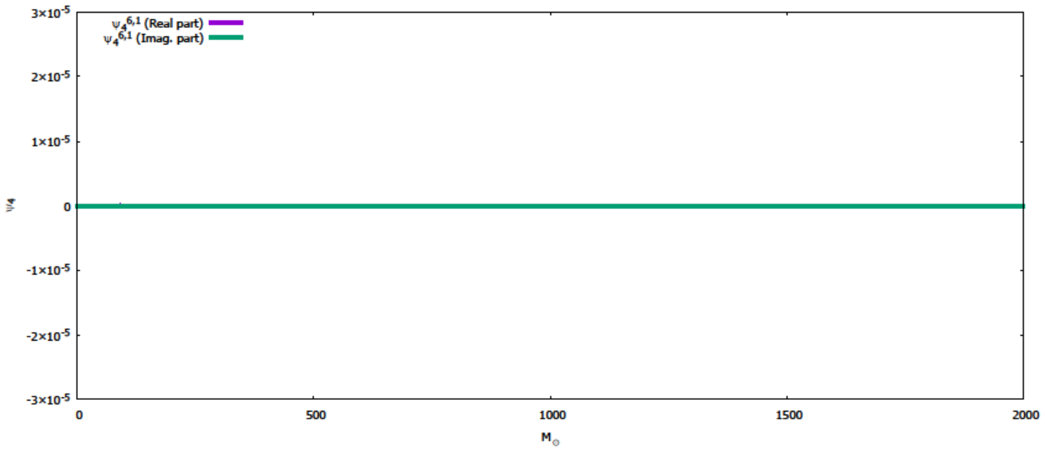
$\psi_4^{5,3}$ (Real part)
$\psi_4^{5,3}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

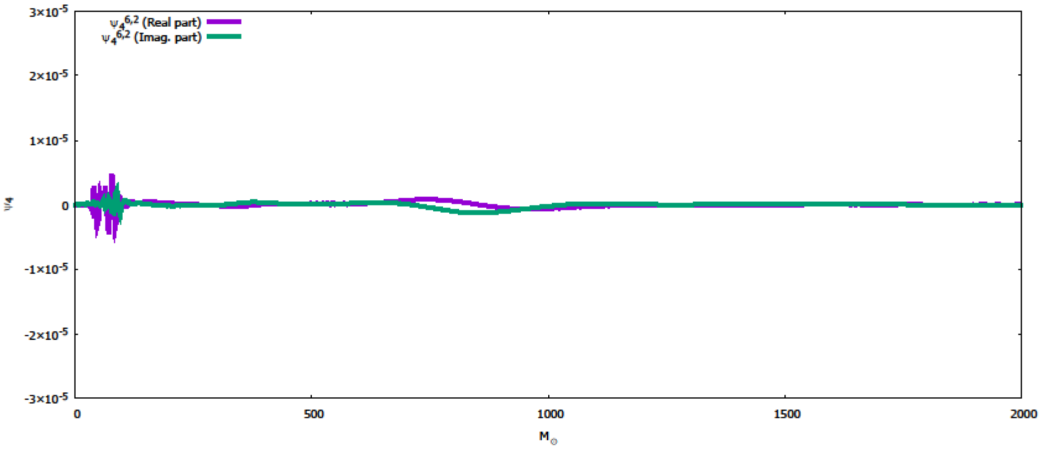


$\psi_4^{5,4}$ (Real part)
$\psi_4^{5,4}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

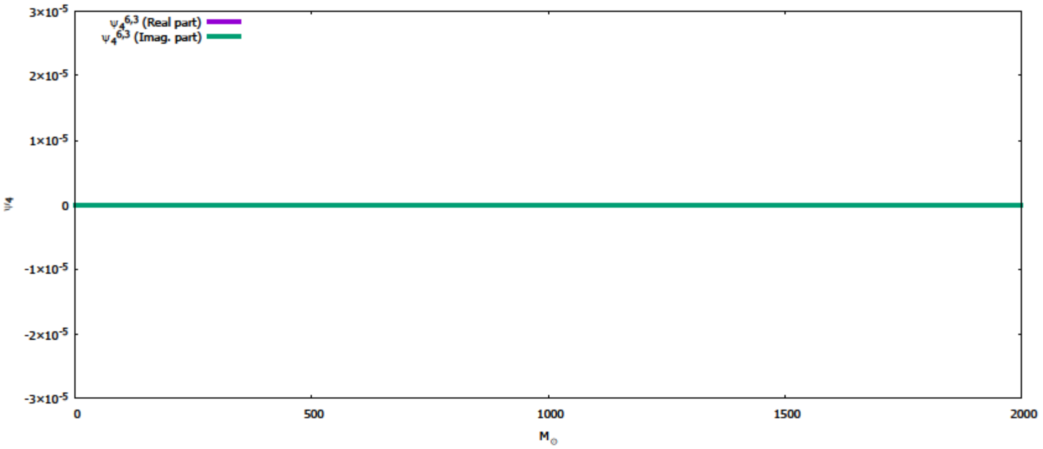

$\psi_4^{5,5}$ (Real part)
$\psi_4^{5,5}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$


$\psi_4^{6,0}$ (Real part)
$\psi_4^{6,0}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$

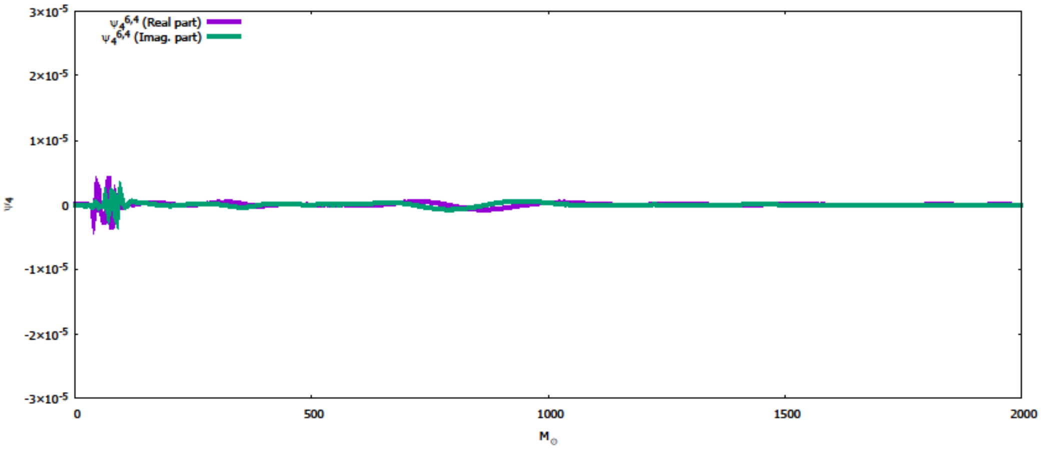

$\psi_4^{6,1}$ (Real part)
$\psi_4^{6,1}$ (Imag. part)
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
ψ_4
0
500
1000
1500
2000
$M_\odot$



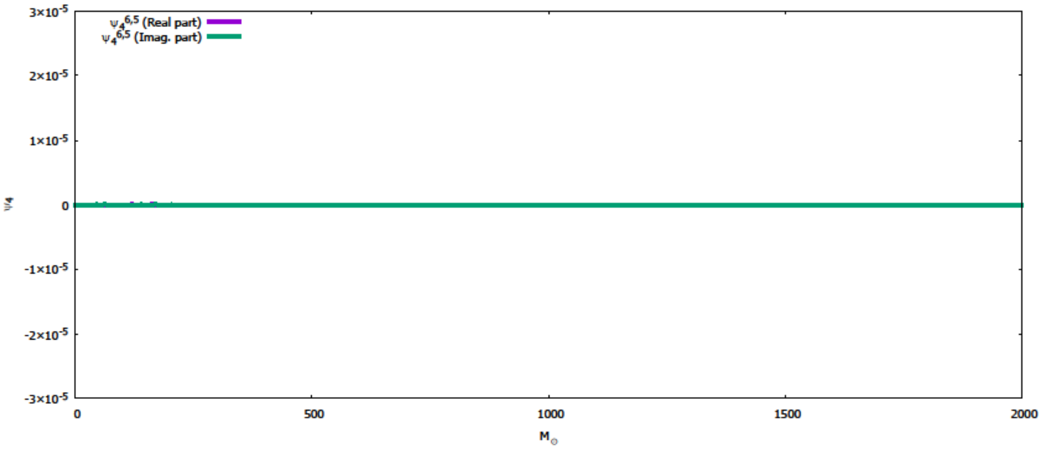
$\psi_4^{6,2}$ (Real part)
$\psi_4^{6,2}$ (Imag. part)
ψ_4
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
0
500
1000
1500
2000
$M_{\odot}$

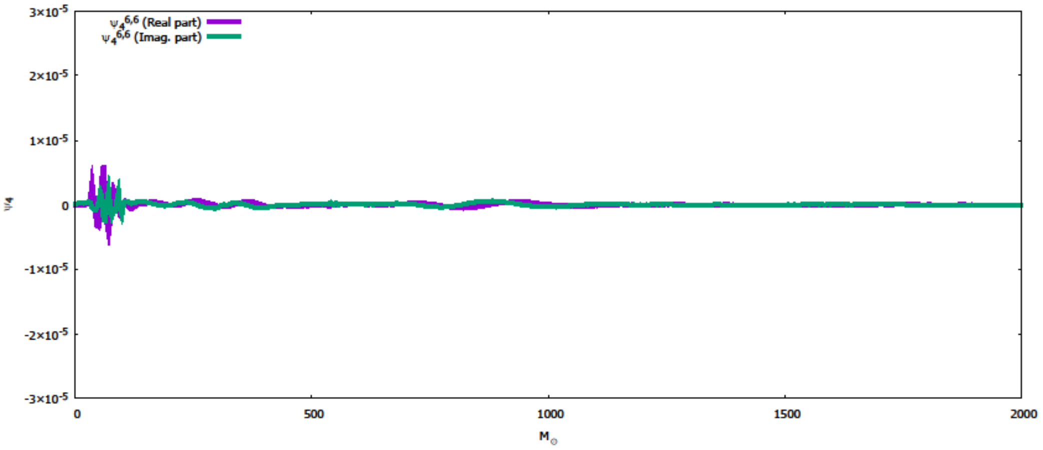


$\psi_4^{6,3}$ (Real part)
$\psi_4^{6,3}$ (Imag. part)
ψ_4
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
0
500
1000
1500
2000
$M_{\odot}$



$\psi_4^{6,4}$ (Real part)
$\psi_4^{6,4}$ (Imag. part)
ψ_4
3×10^{-5}
2×10^{-5}
1×10^{-5}
0
-1×10^{-5}
-2×10^{-5}
-3×10^{-5}
0
500
1000
1500
2000
$M_{\odot}$





As seen above, most of the gravitational modes are zero except for around the point where the metric becomes non-conformal, and from that point onward the prominent gravitational wave mode is $\Psi_4^{2,2}$.

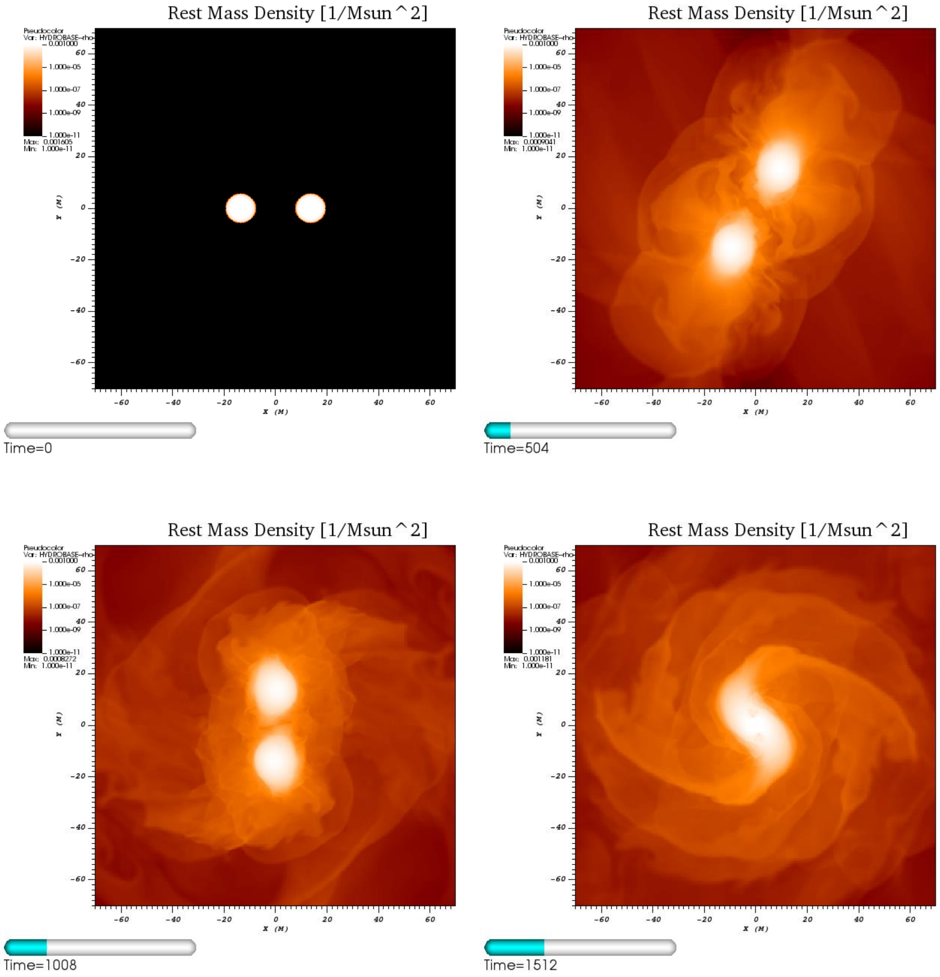
3.4. Case 4: 1.74 and 1.74 $M_\odot$

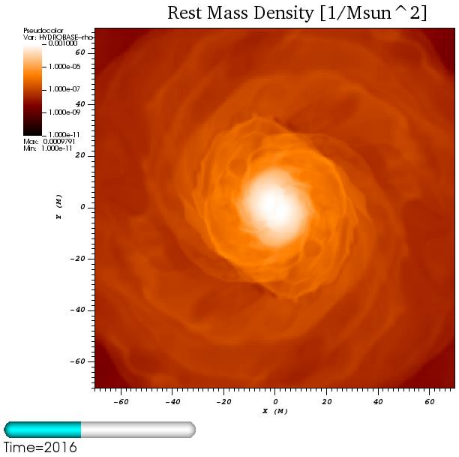
The simulation considered in case 4 is the case of two intermediate mass neutron stars each of 1.74 $M_\odot$ starting at a separation of 40 km. This being the longest run simulation, it gives the best view of gravitational waves.

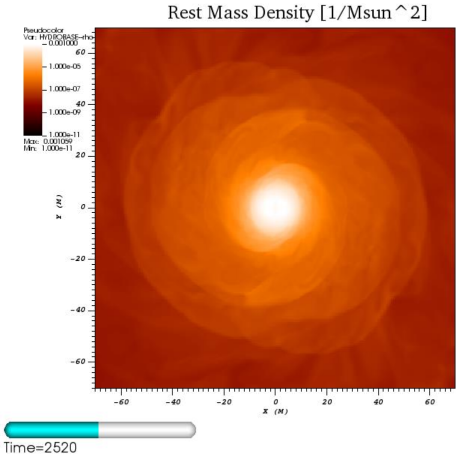
3.4.1. Matter Distribution

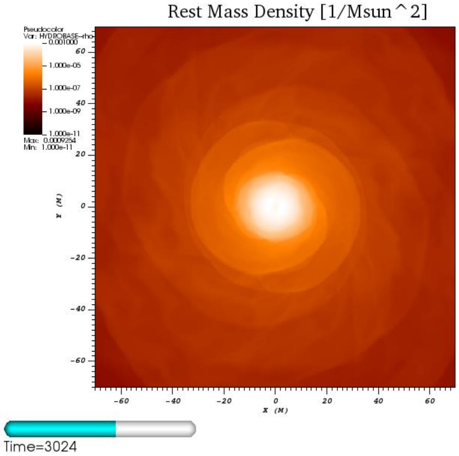
In this case, the neutron stars can be seen at their initial points and begin to orbit around each other and eventually coalescing into a single body. Not visible in the pictures is that the system orbited around each other approximately five times until they became close enough that they were indistinguishable by eye. It is difficult to conclude at what point the bodies become one elongated body versus two distinct stars barely touching surfaces.

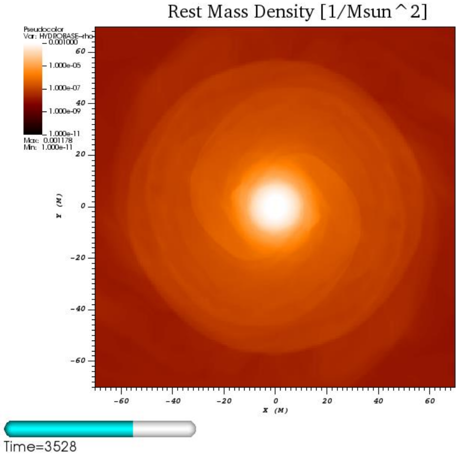
Figure 8: Density distribution during case 4 merger

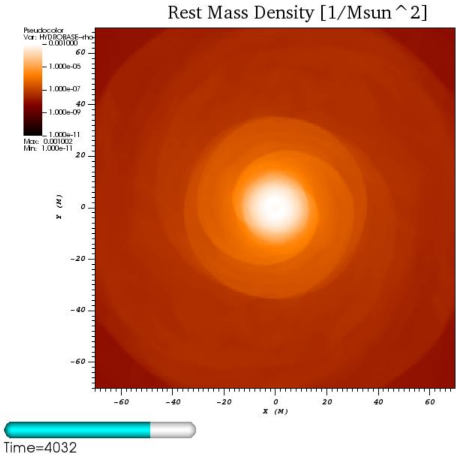


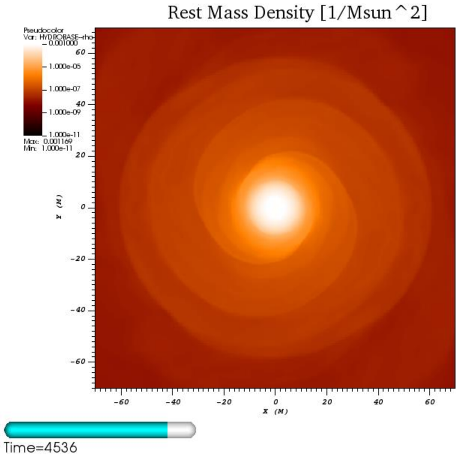

Rest Mass Density [1/Msun^2]
Pseudocolor
Var: HYDROBASE--rho
0.001000
1.000e-05
1.000e-07
1.000e-09
1.000e-11
Max: 0.0009791
Min: 1.000e-11
Y (M)
X (M)
60
40
20
0
-20
-40
-60
Time=2016

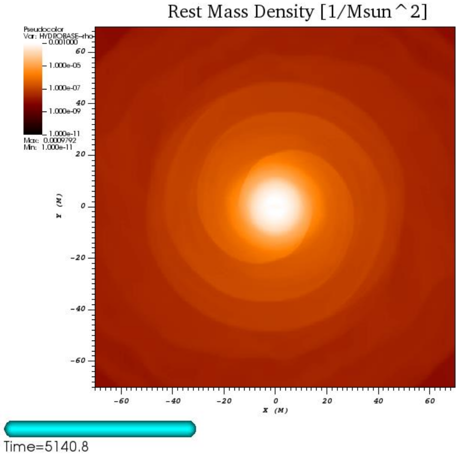

Rest Mass Density [1/Msun^2]
Pseudocolor
Var: HYDROBASE--rho
0.001000
1.000e-05
1.000e-07
1.000e-09
1.000e-11
Max: 0.001059
Min: 1.000e-11
Y (M)
X (M)
60
40
20
0
-20
-40
-60
Time=2520


Rest Mass Density [1/Msun^2]
Pseudocolor
Var: HYDROBASE--rho
0.001000
1.000e-05
1.000e-07
1.000e-09
1.000e-11
Max: 0.0009254
Min: 1.000e-11
Y (M)
X (M)
60
40
20
0
-20
-40
-60
Time=3024


Rest Mass Density [1/Msun^2]
Pseudocolor
Var: HYDROBASE--rho
0.001000
1.000e-05
1.000e-07
1.000e-09
1.000e-11
Max: 0.001178
Min: 1.000e-11
Y (M)
X (M)
60
40
20
0
-20
-40
-60
Time=3528





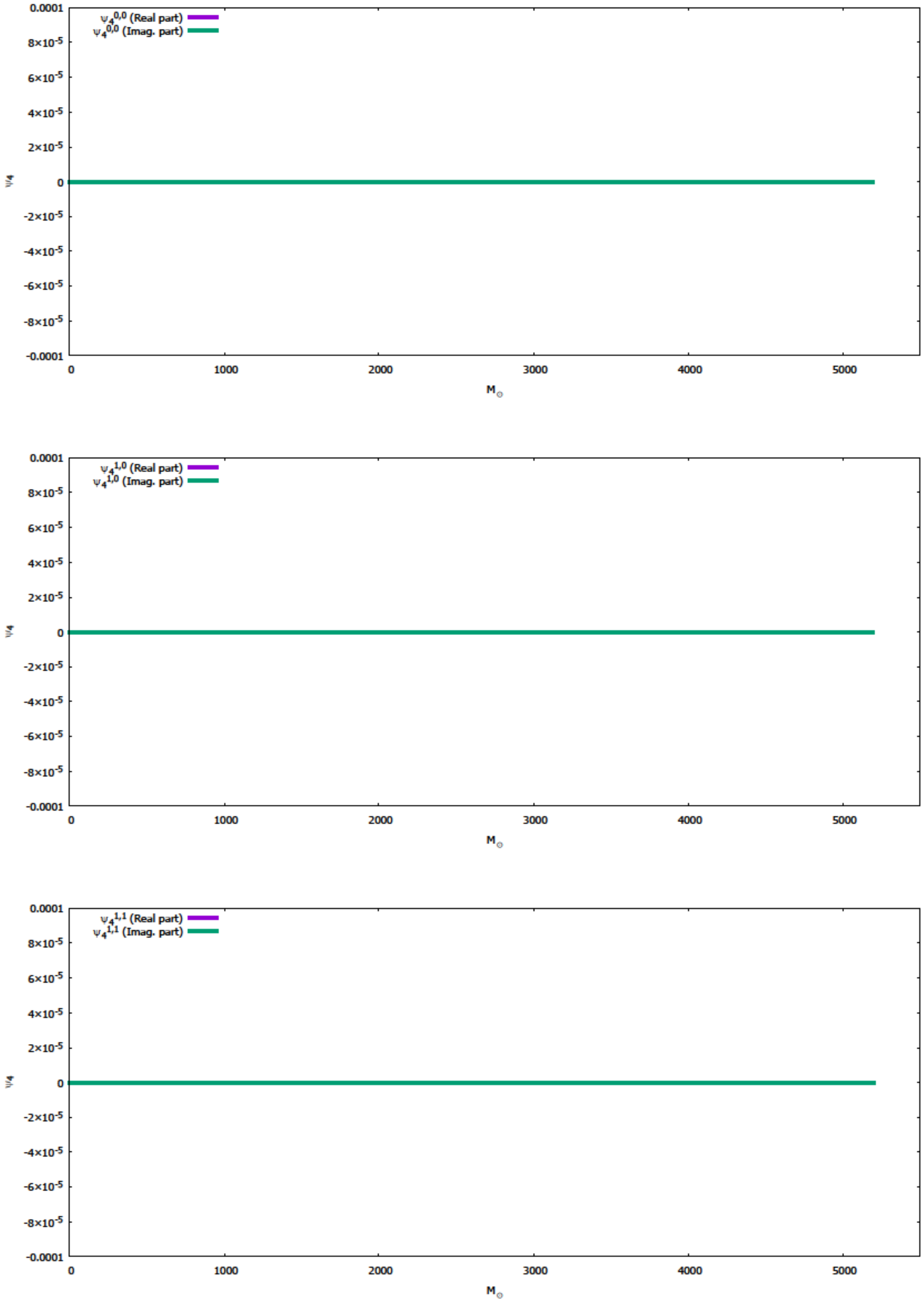


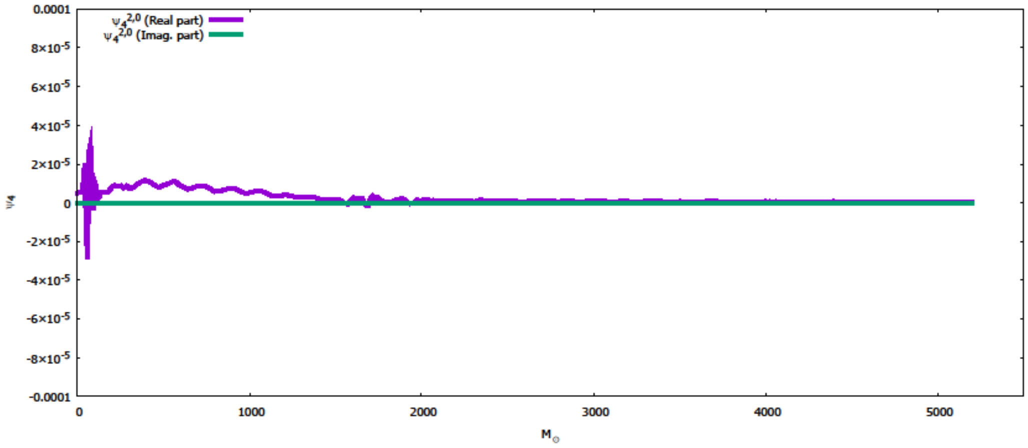
This is the longest simulation running for about $5141\ M_{\odot}$ or $2.53 \times 10^{-2}\ s$. The neutron stars in this simulation orbit each other about five time before they merge. Once they merge, they form an object that is still elongated and emitting gravitational waves.

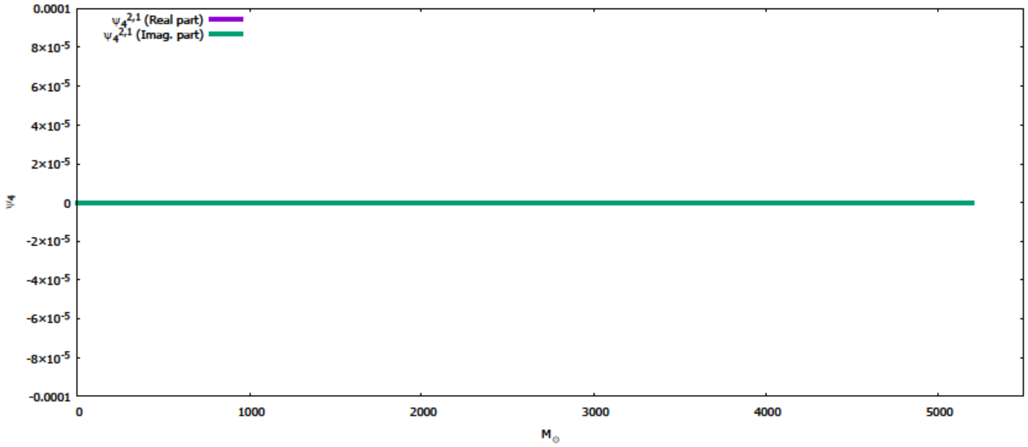
3.4.2. Gravitational Waves

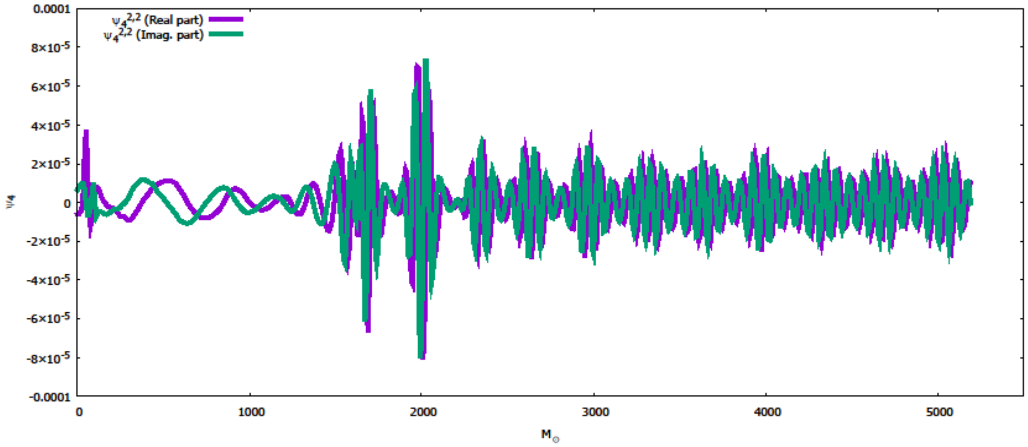
As the neutron stars in the binary inspiral to eventually merge into a final object, they emit gravitational that carry away energy and allow for the orbits to decay. What follows is gravitational waves of various modes during the inspiral of case 4 measured from $45\ M_{\odot}$ or approximately 67 km.

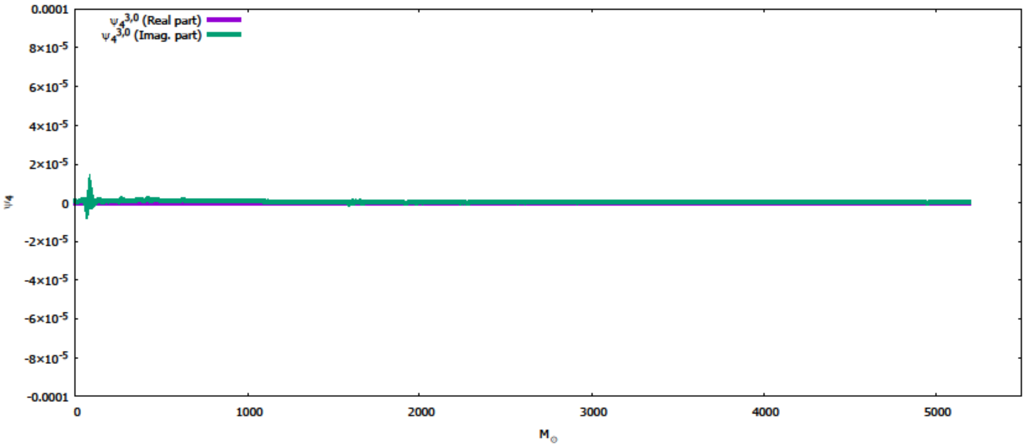
Figure 9: Gravitational wave strength of different rotation modes during case 4 merger

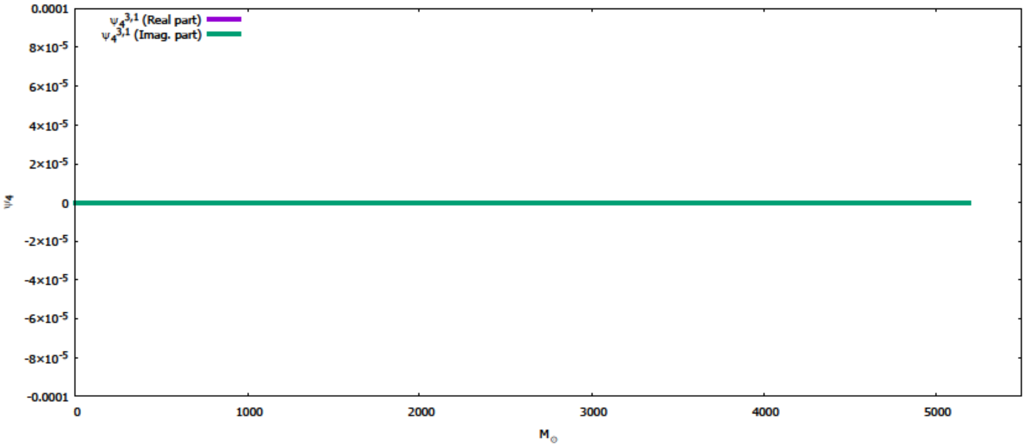


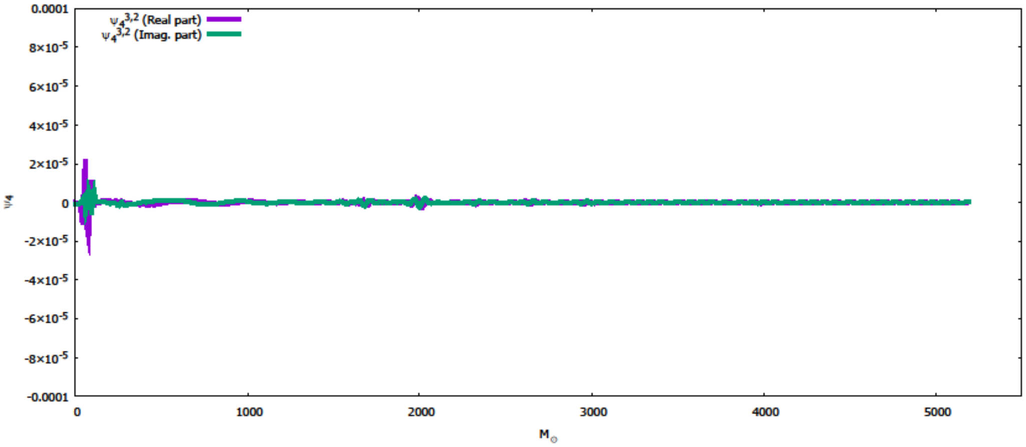

$\psi_4^{2,0}$ (Real part)
$\psi_4^{2,0}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

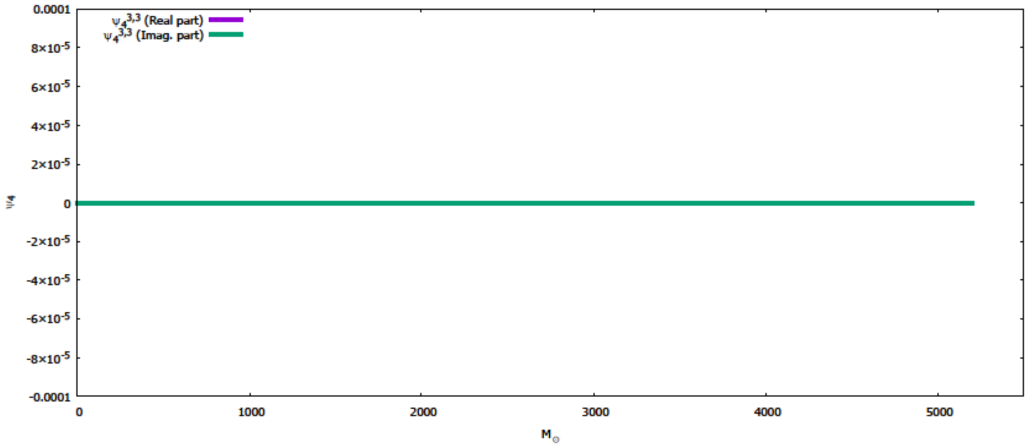

$\psi_4^{2,1}$ (Real part)
$\psi_4^{2,1}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

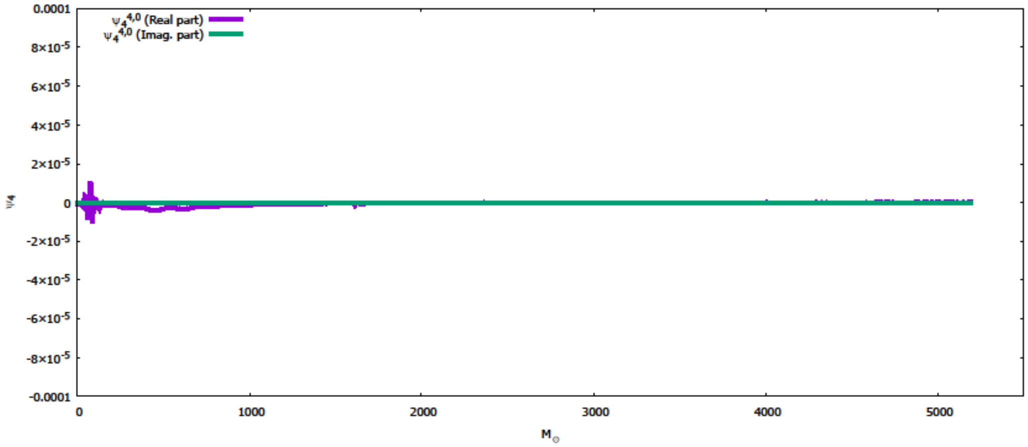

$\psi_4^{2,2}$ (Real part)
$\psi_4^{2,2}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

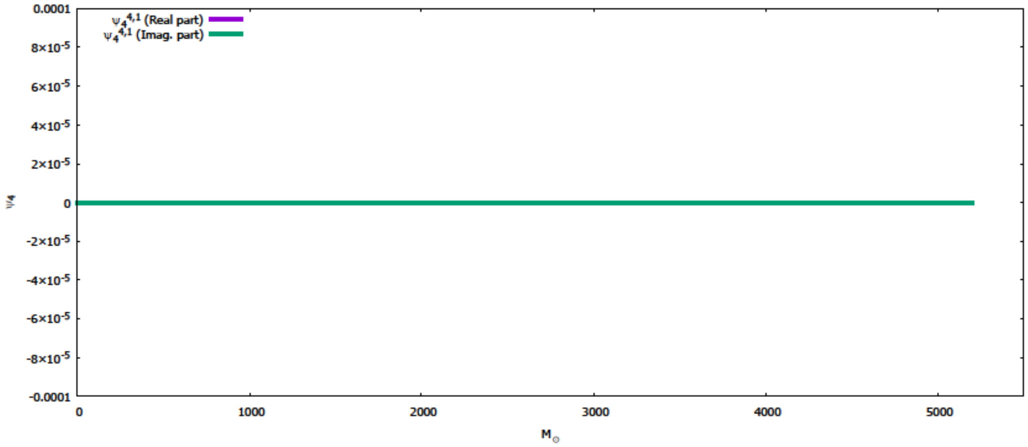

$\psi_4^{3,0}$ (Real part)
$\psi_4^{3,0}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

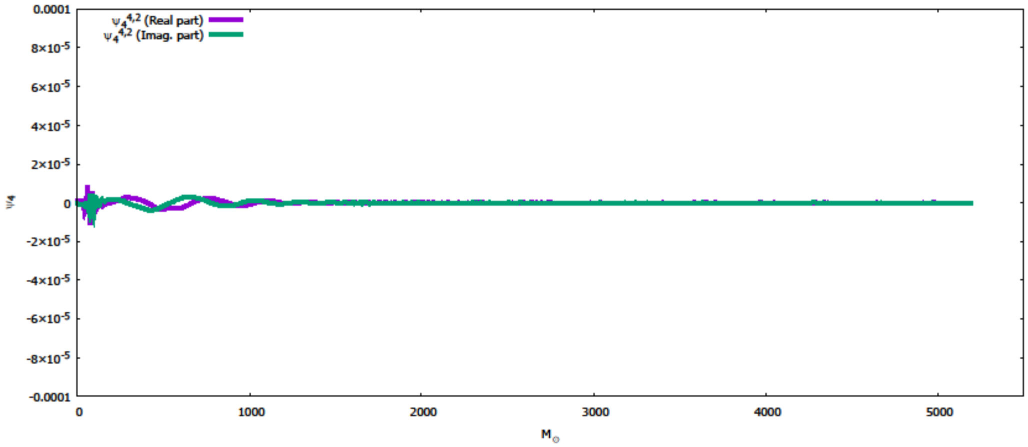

$\psi_4^{3,1}$ (Real part)
$\psi_4^{3,1}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

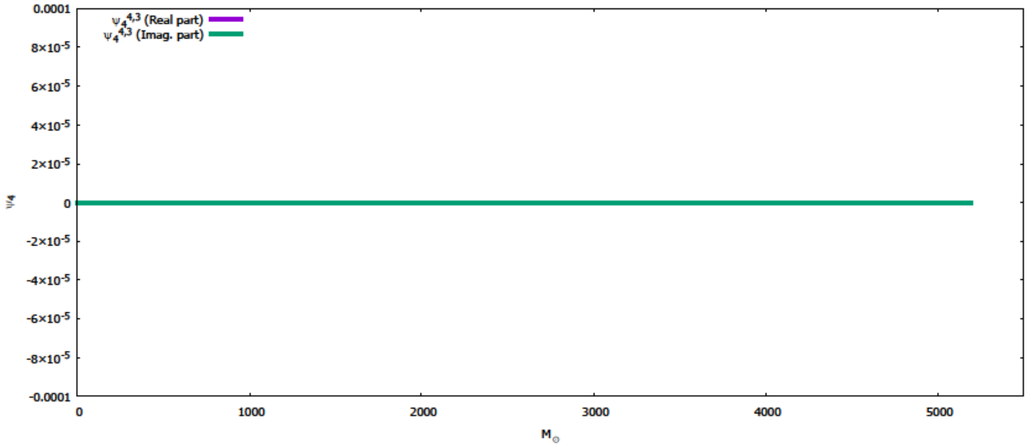

$\psi_4^{3,2}$ (Real part)
$\psi_4^{3,2}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

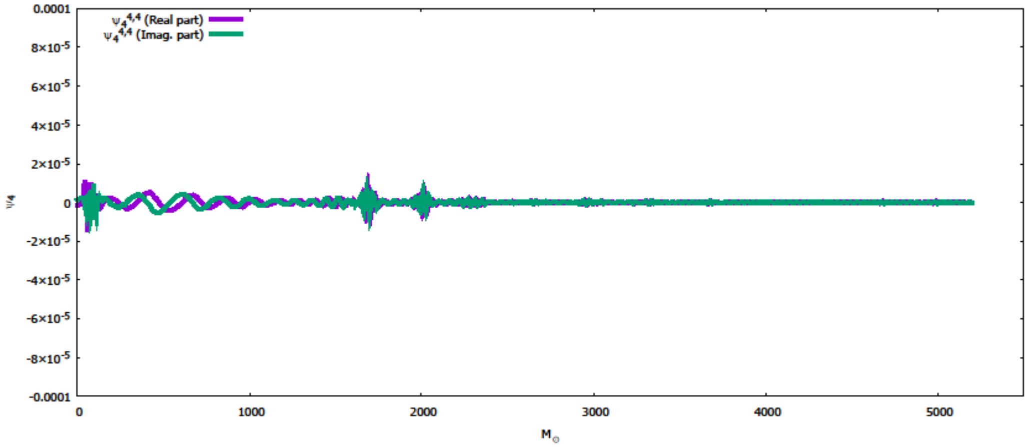

$\psi_4^{3,3}$ (Real part)
$\psi_4^{3,3}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

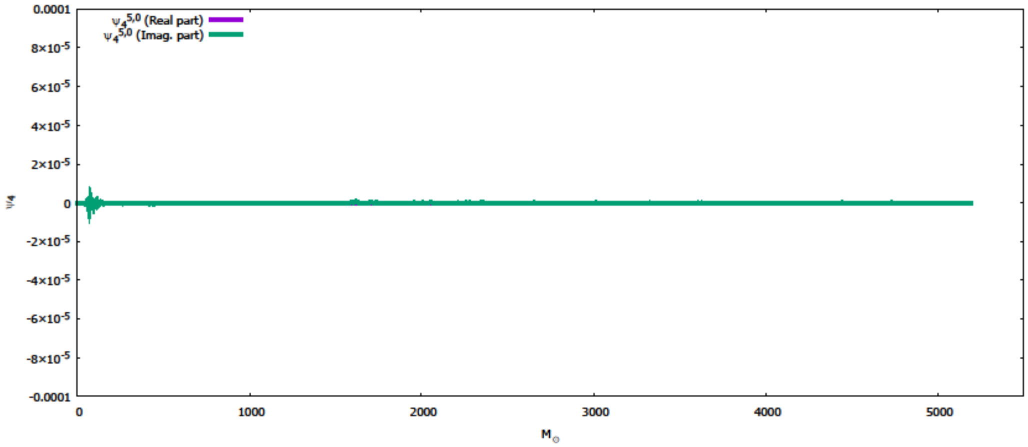

$\psi_4^{4,0}$ (Real part)
$\psi_4^{4,0}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

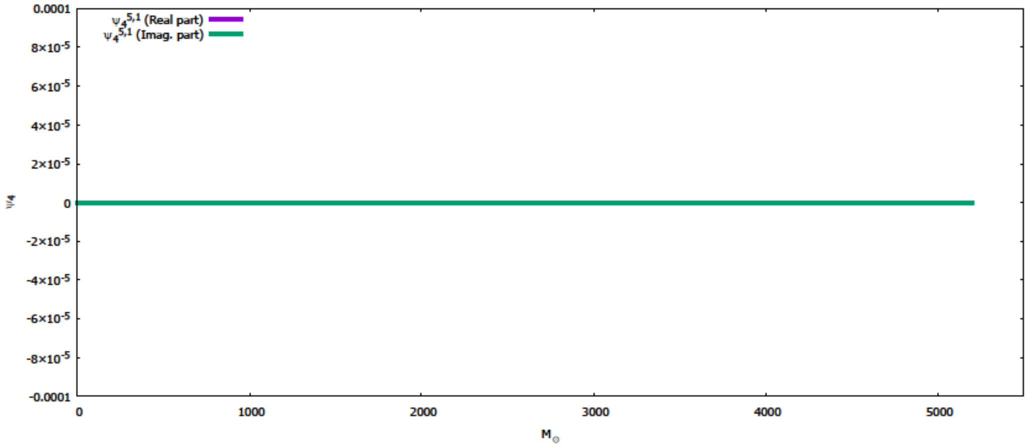

$\psi_4^{4,1}$ (Real part)
$\psi_4^{4,1}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

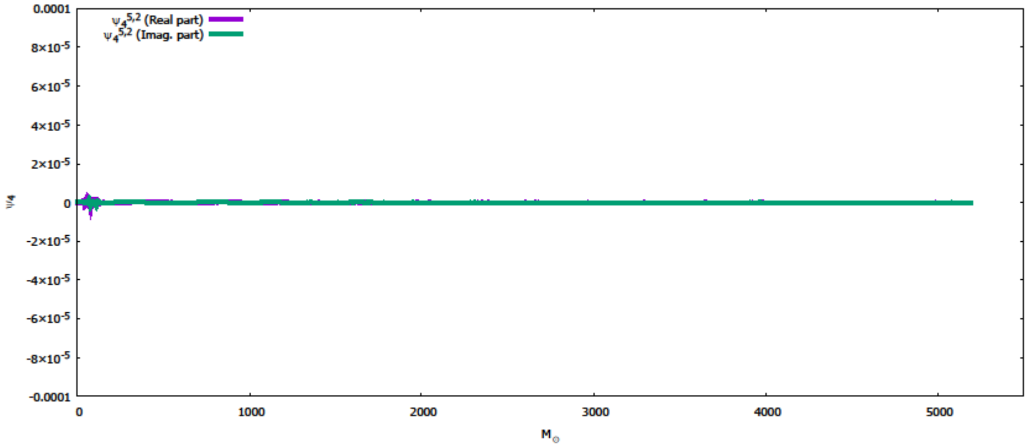

$\psi_4^{4,2}$ (Real part)
$\psi_4^{4,2}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

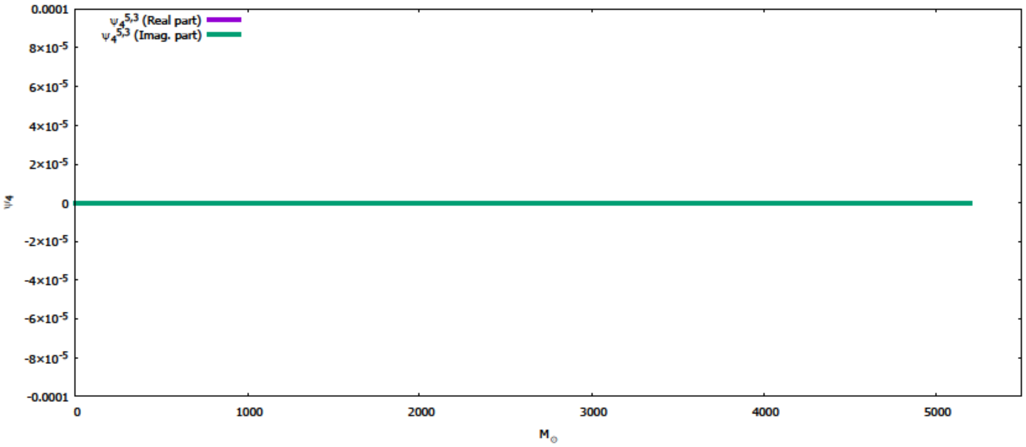

$\psi_4^{4,3}$ (Real part)
$\psi_4^{4,3}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

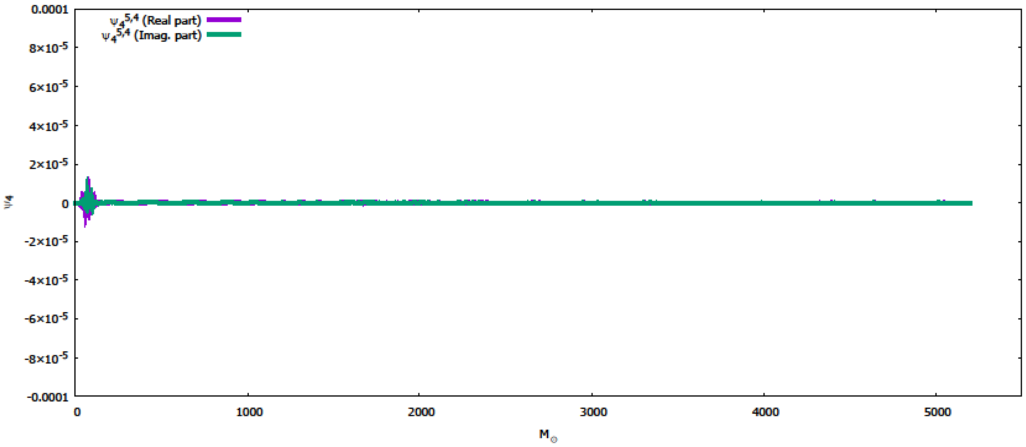

$\psi_4^{4,4}$ (Real part)
$\psi_4^{4,4}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

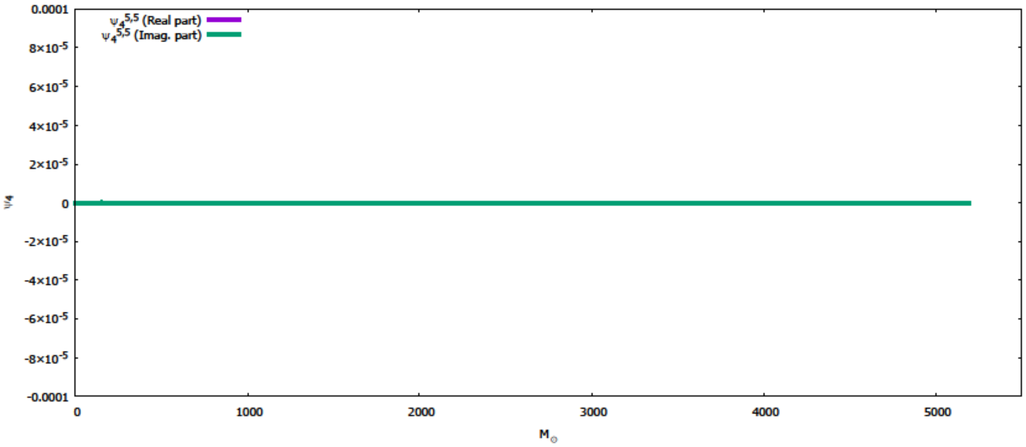

ψ4^5,0 (Real part)
ψ4^5,0 (Imag. part)
0.0001
8×10^-5
6×10^-5
4×10^-5
2×10^-5
0
-2×10^-5
-4×10^-5
-6×10^-5
-8×10^-5
-0.0001
ψ4
0
1000
2000
3000
4000
5000
M⊙

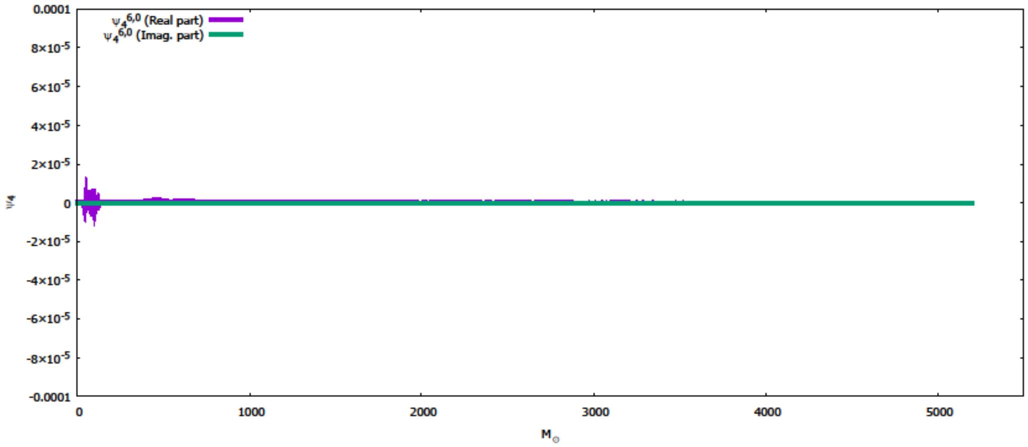

ψ4^5,1 (Real part)
ψ4^5,1 (Imag. part)
0.0001
8×10^-5
6×10^-5
4×10^-5
2×10^-5
0
-2×10^-5
-4×10^-5
-6×10^-5
-8×10^-5
-0.0001
ψ4
0
1000
2000
3000
4000
5000
M⊙

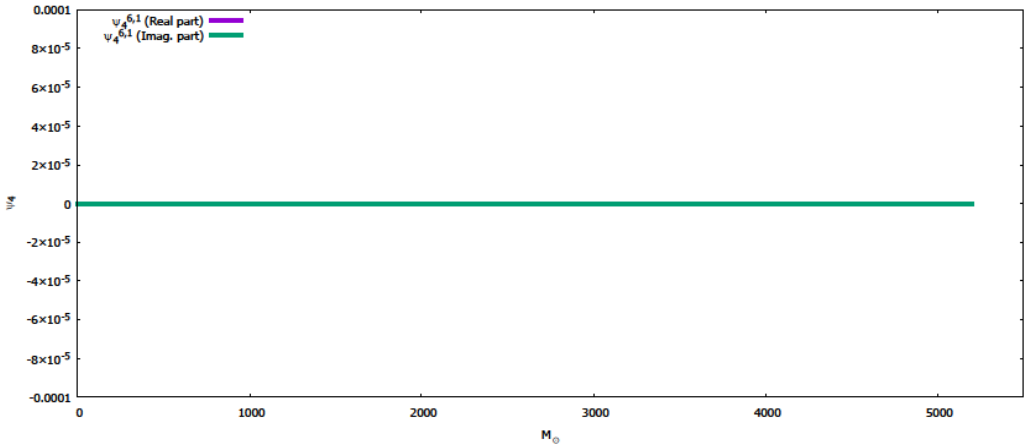

ψ4^5,2 (Real part)
ψ4^5,2 (Imag. part)
0.0001
8×10^-5
6×10^-5
4×10^-5
2×10^-5
0
-2×10^-5
-4×10^-5
-6×10^-5
-8×10^-5
-0.0001
ψ4
0
1000
2000
3000
4000
5000
M⊙

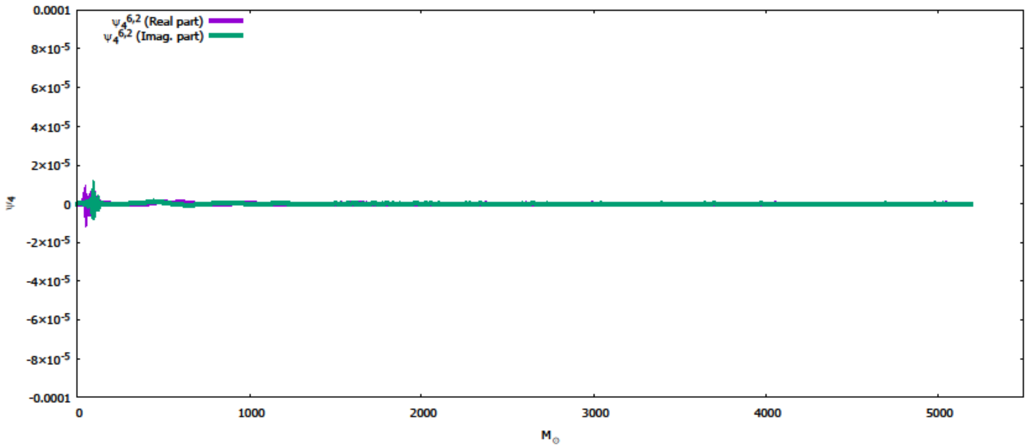

ψ4 5,3 (Real part)
ψ4 5,3 (Imag. part)
ψ4
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
0
1000
2000
3000
4000
5000
M⊙

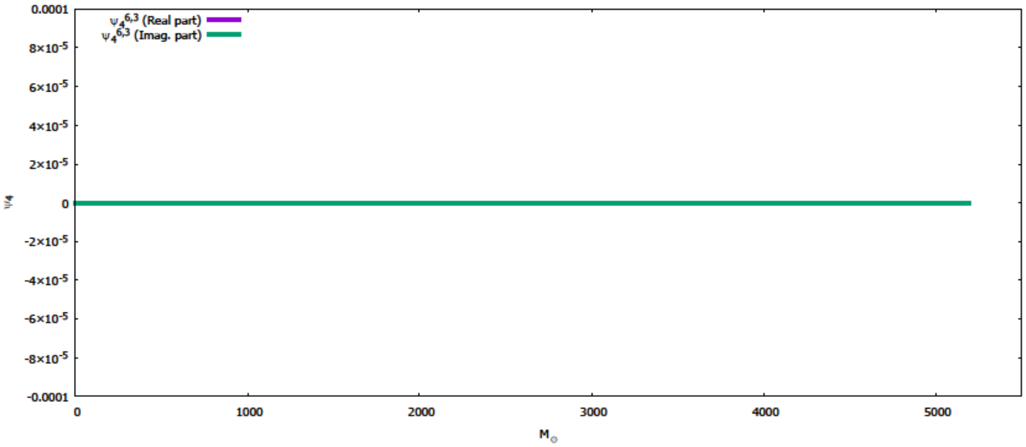

ψ4 5,4 (Real part)
ψ4 5,4 (Imag. part)
ψ4
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
0
1000
2000
3000
4000
5000
M⊙

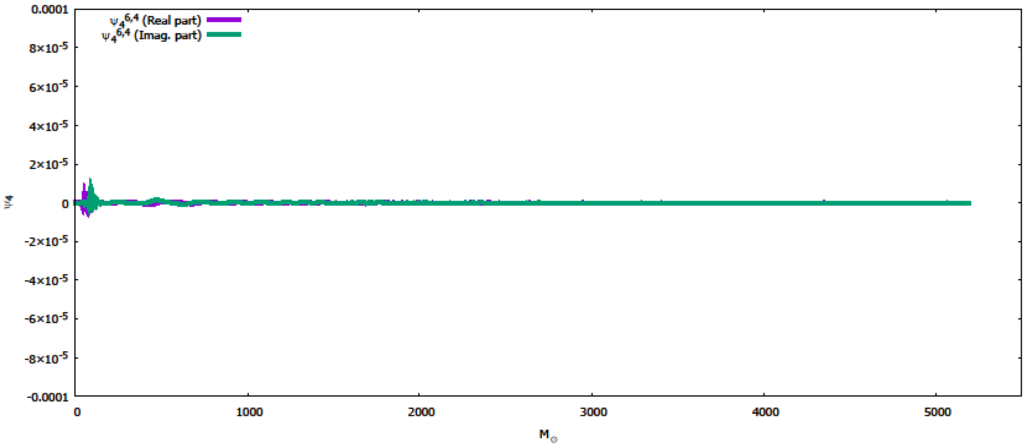

ψ4 5,5 (Real part)
ψ4 5,5 (Imag. part)
ψ4
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
0
1000
2000
3000
4000
5000
M⊙

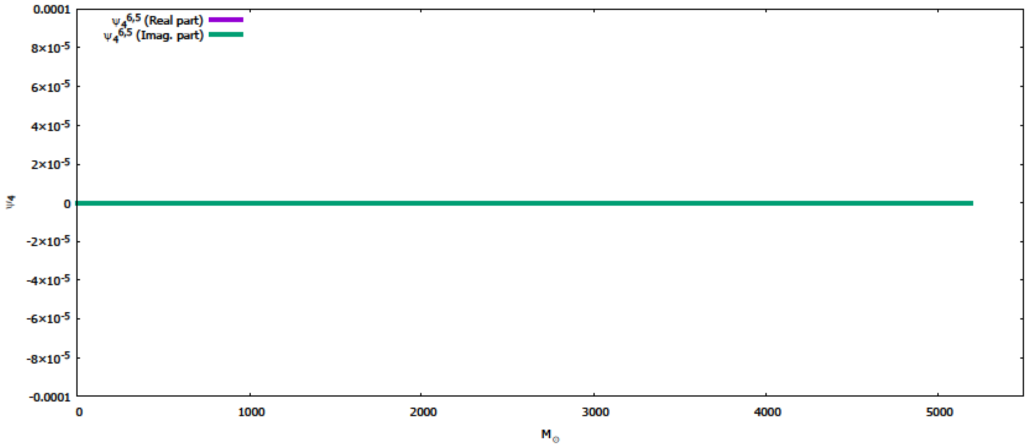

$\psi_4^{6,0}$ (Real part)
$\psi_4^{6,0}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$

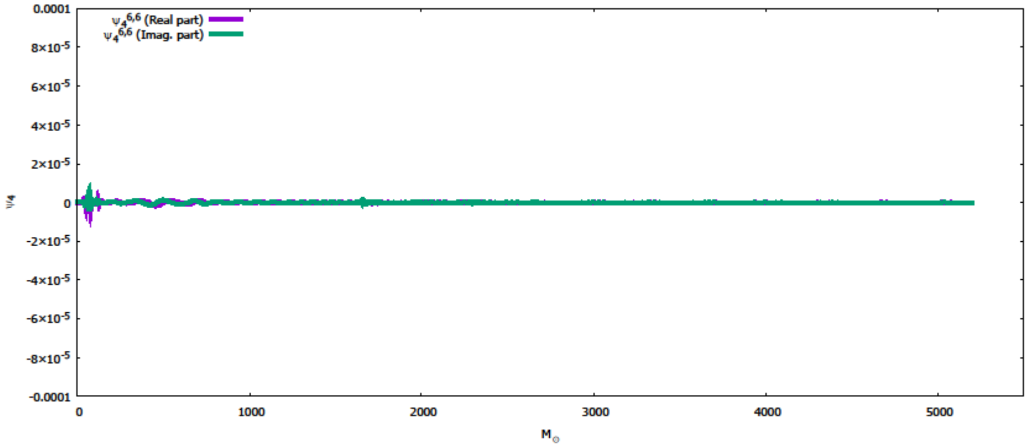

$\psi_4^{6,1}$ (Real part)
$\psi_4^{6,1}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$


$\psi_4^{6,2}$ (Real part)
$\psi_4^{6,2}$ (Imag. part)
0.0001
8×10-5
6×10-5
4×10-5
2×10-5
0
-2×10-5
-4×10-5
-6×10-5
-8×10-5
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$


$\psi_4^{6,3}$ (Real part)
$\psi_4^{6,3}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$


$\psi_4^{6,4}$ (Real part)
$\psi_4^{6,4}$ (Imag. part)
0.0001
8×10^{-5}
6×10^{-5}
4×10^{-5}
2×10^{-5}
0
-2×10^{-5}
-4×10^{-5}
-6×10^{-5}
-8×10^{-5}
-0.0001
ψ_4
0
1000
2000
3000
4000
5000
$M_\odot$





This being the longest run simulation allows for some good long-term behavior of gravitational waves to be observed. In this case, as with the other cases, most of the gravitational wave modes are zero with the $\Psi_4^{2,2}$ mode being the most prominent. The $\Psi_4^{2,2}$ mode exhibits some type of pulsing in addition to the expected gravitational waves. As can be seen from the $\Psi_4^{2,2}$ mode, the system is still not spherically symmetric, and has not fully collapsed into a black hole, although the mass should be above the point at which a black hole would form.

4. Conclusion and Future Work

As seen, there is much that goes into the simulation of the merger of neutron stars or system of compact bodies. Starting from a foundation of general relativity that can be adapted into the ADM and BSSN formulations, numerical relativity can be used to solve problems that are far too complicated for an analytic solution to exist. This process is still computationally intensive with a steep learning curve with many failed simulations. The Einstein Toolkit, being an open-source collaborative program, lends itself to being a fantastic tool for numerical relativistic simulations. However, since it is community driven, the scope of the Toolkit is beyond any one person and at times it acts like a 'black box' in which initial data is input. There are many more ways to interpret the data that is produced during each of the simulations, with this thesis only covering the basics of the matter distribution and gravitational waves produced during the inspiral.

There are many areas that are still available to be explored in the field of gravitational astronomy. The field of gravitational wave astronomy is still in the beginning stages and has many possibilities within it. Refinement of the tools of numerical relativity will lead to better simulations that will allow for more accurate predictions. Improvement of gravitational wave detection tools, like Advanced LIGO, will allow for more refined measurements of observed gravitational waves.

In addition to the field of gravitational wave detection still at the beginning stage, I am personally still in the early phase of understanding numerical relativity simulations and gravitational waves. Our first goal is to complete the case 1 simulation successfully so that the complete merger of the 2.08 $M_{\odot}$ neutron stars may be studied. Additionally, we have started an effort to simulate the merger of binary black holes in order to reproduce the gravitation wave event GW150914. This type of research will be extended to use the high-performance computing facility available at the University of North Dakota.

References

Abbot B. P. et al., 2016, Phys Rev Lett, 116, 061102, DOI: 10.1103/PhysRevLett.116.061102

Allen G., Goodale T., Löffler F., Rideout D., Schnetter E., Seidel E., 2010, 11th IEEE/ACM International Conference on Grid Computing, 2010, pp. 359-368, DOI: 10.1109/GRID.2010.5698008.

Baiotti L., Hawke I., Montero P., 2007, https://einsteintoolkit.org/thornguide/EinsteinEvolve/GRHydro/documentation.html

Bode T., Löffler F., 2010, https://einsteintoolkit.org/thornguide/EinsteinBase/HydroBase/documentation.html

Ciolfi R., Kastaun W., Giacomazzo B., Endrizzi A., Siegel D., Perna R., 2017, Phys Rev D, 95, 063016, DOI: 10.1103/PhysRevD.95.063016

Colella P., Sekora M., 2008, Journal of Computaional Physics 227, 7069-7076, DOI: 10.1016/j.jcp.2008.03.034

Eddington A., 1922, Proc. R. Soc. Lond. A, 102, 268-282, DOI: 10.1098/rspa.1922.0085

Einstein A., 1905, Annalen Phys. 17, 891-921, 2005, Annalen Phys. 14, 194-224, DOI: 10.1002/andp.200590006

Lorentz, Einstein, Minkowski, 1913, Science, 39, 1017, DOI: 10.1126/science.39.1017.944.a

Etieene Z., Werneck L., 2021, https://einsteintoolkit.org/thornguide/WVUThorns_Diagnostics/VolumeIntegrals_GRMH

D/documentation.html

Fonseca E., et al., ApJL, 915:L12, DOI: 10.3847/2041-8213/ac03b8

Goodale T.,

https://einsteintoolkit.org/thornguide/EinsteinBase/ADMBase/documentation.html

Gourgoulhon E., Grandclément P., Taniguchi K., Marck J., Bonazzola S., 2001 Phys Rev D, 63, 064029, DOI: 10.1103/PhysRevD.63.064029

Grandclément P., Gourgoulhon E., Bonazzola S., 2002, Phys Rev D, 65, 044021, DOI: 10.1103/PhysRevD.65.044021

Grandclément P., Novak J., 2008, Living Rev. Relativity, 12, 1

Hahn S., Lindquist R., 1964, Annals of Physics, 29, 304-331

Hawke I., Rideout D.,

https://einsteintoolkit.org/thornguide/EinsteinBase/ADMCoupling/documentation.html

Lattimer J., Prakash M., 2004, Science, 304, 536, DOI: 10.1226/science.1090720

Lee W., Ramierez-Ruiz E., Van De Ven G., 2010, ApJ, 720:953-975, DOI: 10.1088/0004-637X/720/1/953

Löffler, F. 2022,

https://einsteintoolkit.org/thornguide/EinsteinAnalysis/Hydro_Analysis/documentation.html

Löffler F., et al., 2012, Class. Quantum Grav. 29, 115001, DOI: 10.1088/0264-9381/29/11/115001

Mazzali P., Röpke F., Benetti S., Hillebrandt W., 2007, Science, 315, 825-828

McCorquodale P., Colella P., 2011, Communications in Applied Mathematics and Computational Science, 6, 1

Misner, C. W., Wheeler, J. A., & Thorne, K. S. 2017, Gravitation (Second; Princeton

University Press)

Mösta P., et al., 2013, Class. Quantum Grav. 31, 015005, DOI: 10.1088/0264-9381/31/1/015005

Newman E., Penrose R., 1962, J. Math. Phys. 3, 566, DOI: 10.1063/1.1724257

Oppenheimer J., Volkoff G., 1939, Phys Rev, 55, 374-381

Ott C., Schnetter E., 2013, https://einsteintoolkit.org/thornguide/EinsteinEOS/EOS_Omni/documentation.html

Pe'er A., 2020, https://asafpeer2.ph.biu.ac.il/wp-content/uploads/2020/08/Einstein.pdf

Pian E., 2021, Frontiers in Astronomy and Space Sciences, 7:609460, DOI: 10.3389/fspas.2020.609460

Radke T., https://einsteintoolkit.org/thornguide/CactusUtils/NaNChecker/documentation.html

Read J., Lackey B., Owen B., Friedman J., 2009a, Phys Rev D, 79, 124032, DOI: 10.1103/PhysRevD.79.124032

Read J., Markakis C., Shibata M., Uryū K., Creighton J., Friedman J., 2009b, Phys Rev D, 79, 124033, DOI: 10.1103/PhysRevD.79.124033

Renzo M., et al., 2019, A&A, 624, A66, DOI: 10.1051/0004-6361/201833297

Sana H., et al., 2012, Science, 337, 444-446

Sarracino R., Eccles, M., 1996, Astrophys Space Sci, 111, 375

Schnetter, E., https://einsteintoolkit.org/arrangementguide/Carpet/documentation.html

Schnetter, E. 2007, https://einsteintoolkit.org/thornguide/EinsteinBase/TmunuBase/documentation.html

Schutz B., 1999, Class. Quantum Grav. 16 A131

Schwarzschild, K. 1916, Gen Relativ Gravit, 35, 951

Taylor J., Weisberg J., 1982, ApJ, 253:908-920

The International Astronomical Union. 2012, https://asa.hmnao.com/static/files/2016/Astronomical_Constants_2016.pdf

Thornburg, J., https://einsteintoolkit.org/thornguide/EinsteinAnalysis/AHFinderDirect/documentation.html

Tolman R., 1939 Phys Rev, 55, 364-373

Xue C., et al., 2020, National Science Review, 7:1803-1817, DOI: 10.1093/nsr/nwaa165

Ye C., Fong W., Kremer K., Rodriguez C., Chatterjee., Fragione G., Rasio F., 2020, ApJL, 888:L10 DOI: 10.3847/2041-8213/ab5dc5

Zilhão M., Löffler F. 2013, Int J Mod Phys A, 28, 1340014, DOI: 10.1142/S0217751X13400149

Appendix A – List of Abbreviations

ADM – Arnowitt Deser Misner

AMR – Adaptive Mesh Refinement

BNS – Binary Neutron Stars

BSSN – Baumgarte, Shapiro, Shibata, Nakamura

EOS – Equation of State

FMR – Fixed Mesh Refinement

GRMHD – General-Relativistic MagnetoHydroDynamics

LORENE – Langage Objet pour la RElativité NumériquE

LIGO –Laser Interferometer Gravitational-Wave Observatory

NaN – Not a Number

PDE – Partial Differential Equations

Appendix B – Unit Conversions

For the purpose of this thesis, the following definitions and constants of nature are defined as:

Constant	Value	Reference
c	$299\,792\,458\; m \cdot s^{-1}$	(The International Astronomical Union 2012)
G	$6.674\,28 \times 10^{-11}\; m^3 \cdot kg^{-1} \cdot s^{-1}$	(Xue et al. 2020)
$M_\odot$	$1.988\,92 \times 10^{30}\; kg$	(The International Astronomical Union 2012)

Table 5: Physical constants of nature

Additionally, the units for mass, length, time, and magnetic field then become

$$[M] = M_\odot,$$

$$[L] = [M]\frac{G}{c^2},$$

$$[T] = \frac{[L]}{c}.$$

This corresponds to a unit conversion of

$$[L] = 1\; M_\odot = 1.477 \times 10^3\; m,$$

$$[T] = 1\; M_\odot = 4.92673 \times 10^{-6}\; s.$$

Appendix C – Parameter File

```
# Carpet parameter file for binary Neutron star system
# physical ID is LORENE
#


#-----------------------------------------------------
# Cactus parameters:
#-----------------------------------------------------
Cactus::cctk_run_title     = "May23-
MagneticFieldVolumeCase1"
Cactus::cctk_full_warnings = "yes"
Cactus::highlight_warning_messages = "no"


Cactus::terminate       = "time"
Cactus::cctk_final_time = 2000.0


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


ActiveThorns = "Boundary CartGrid3D CoordBase Fortran
InitBase IOUtil LocalReduce SymBase Time"
ActiveThorns = "AEILocalInterp"
ActiveThorns = "MoL Slab SpaceMask SphericalSurface"
ActiveThorns = "Carpet CarpetInterp CarpetInterp2
CarpetIOASCII CarpetIOHDF5 CarpetIOScalar CarpetLib
CarpetIOBasic CarpetReduce CarpetRegrid2 CarpetSlab
CarpetTracker CarpetMask LoopControl"
ActiveThorns = "Formaline"
ActiveThorns = "NaNChecker TerminationTrigger
TimerReport"
```

```
ActiveThorns = "ADMbase ADMcoupling ADMmacros
CoordGauge StaticConformal"
ActiveThorns = "RotatingSymmetry180
ReflectionSymmetry"
ActiveThorns = "Constants TmunuBase HydroBase"
ActiveThorns = "QuasiLocalMeasures"
ActiveThorns = "EOS_Omni"
ActiveThorns = "GRHydro"
ActiveThorns = "SummationByParts"
ActiveThorns = "GenericFD NewRad"
ActiveThorns = "ML_BSSN ML_BSSN_Helper
ML_ADMConstraints"
ActiveThorns = "Hydro_Analysis NSTracker"
ActiveThorns = "Dissipation"
ActiveThorns = "SystemStatistics SystemTopology"
ActiveThorns = "VolumeIntegrals_GRMHD"
# Wave extraction (Psi4)
ActiveThorns = "WeylScal4 Multipole"
ActiveThorns = "AHFinderDirect"


#------------------------------------------------------
# Diagnostic parameters:
#------------------------------------------------------


Carpet::output_timers_every = 0
Carpet::storage_verbose   = "no"
Carpet::verbose           = "no"
Carpet::veryverbose       = "no"
Carpet::grid_structure_filename   = "carpet-grid-
structure"
Carpet::grid_coordinates_filename = "carpet-grid-
coordinates"
```

```
CarpetLib::output_bboxes  = "no"

CarpetMask::verbose    = "no"
CarpetReduce::verbose  = "no"
CarpetRegrid2::verbose = "no"
CarpetRegrid2::veryverbose    = "no"
CarpetTracker::verbose = "no"



TimerReport::out_every    = 4096
TimerReport::out_filename = "TimerReport"
TimerReport::output_all_timers          = "yes"
TimerReport::output_all_timers_together = "yes"
TimerReport::output_all_timers_readable = "yes"
TimerReport::n_top_timers               = 40



QuasiLocalMeasures::verbose   = "no"
SphericalSurface::verbose   = "no"

#------------------------------------------------------
# Utility parameters:
#------------------------------------------------------

NaNChecker::check_every    =  128 # twice for
every_coarse
NaNChecker::check_vars = "
            ADMBase::curv
            ADMBase::metric
            ADMBase::lapse
            ADMBase::shift
            HydroBase::rho
            HydroBase::eps
            HydroBase::press
            HydroBase::vel
"
NaNChecker::action_if_found   =  "terminate"
#NaNChecker::action_if_found = "just warn"
#"terminate", "just warn", "abort"


#------------------------------------------------------
# Run parameters:
#------------------------------------------------------


#------
# Grid:
#------


MoL::ODE_Method             = "rk4"
MoL::MoL_Intermediate_Steps = 4
MoL::MoL_Num_Scratch_Levels = 1
# use dt = 0.4 dx (works for core collapse)
Time::dtfac = 0.35


CoordBase::domainsize = "minmax"
CoordBase::xmin =    0.00
CoordBase::ymin = -960.00
CoordBase::zmin =    0.00
CoordBase::xmax = +960.00
```

```
CoordBase::ymax = +960.00
CoordBase::zmax = +960.00
CoordBase::dx   =  10.00
CoordBase::dy   =  10.00
CoordBase::dz   =  10.00

CoordBase::boundary_size_x_lower     = 3
CoordBase::boundary_size_y_lower     = 3
CoordBase::boundary_size_z_lower     = 3
CoordBase::boundary_size_x_upper     = 3
CoordBase::boundary_size_y_upper     = 3
CoordBase::boundary_size_z_upper     = 3

CoordBase::boundary_shiftout_x_lower = 1
CoordBase::boundary_shiftout_y_lower = 0
CoordBase::boundary_shiftout_z_lower = 1

reflectionsymmetry::avoid_origin_x       = no
reflectionsymmetry::avoid_origin_y       = no
reflectionsymmetry::avoid_origin_z       = no
reflectionsymmetry::reflection_x         = no
reflectionsymmetry::reflection_y         = no
reflectionsymmetry::reflection_z         = yes

CartGrid3D::type = "coordbase"
Carpet::domain_from_coordbase = "yes"

Driver::ghost_size                      = 3

# General Carpet parameters:
```

```
Carpet::enable_all_storage       = "no"
Carpet::use_buffer_zones         = "yes"
Carpet::schedule_barriers        = "no"

Carpet::poison_new_timelevels    = "yes"
Carpet::check_for_poison         = "no"

Carpet::init_3_timelevels        = "no"
Carpet::init_fill_timelevels     = "yes"

CarpetLib::poison_new_memory         = "yes"
CarpetLib::poison_value              = 114
CarpetLib::check_bboxes              = "no"
CarpetLib::interleave_communications = "yes"
CarpetLib::combine_sends             = "yes"

CarpetInterp::tree_search = "yes"
CarpetInterp::check_tree_search = "no"

CarpetRegrid2::freeze_unaligned_levels = "yes"
CarpetRegrid2::freeze_unaligned_parent_levels = "yes"
CarpetRegrid2::ensure_proper_nesting   = "yes"
CarpetRegrid2::snap_to_coarse          = "yes"
CarpetRegrid2::symmetry_rotating180    = "yes"

# System specific Carpet parameters:
Carpet::max_refinement_levels    = 8
Carpet::prolongation_order_space = 5
Carpet::prolongation_order_time  = 2
```

```
Carpet::refinement_centering     = "vertex"


CarpetRegrid2::regrid_every = 64 # as often as
required

CarpetRegrid2::num_centres  = 3

CarpetRegrid2::min_distance = 0


# box sizes are given by:

# * the stars seem to puff up to about 13M during the
initial phase of the evolution

# * I need 12 buffer points (RK4, 3 ghost zones)

# * need three coarse points for interpolation onto
last fine buffer point

# these boxes are minimal in this sense. The coarser
grid are completely

# covered by the finer grids and their buffers.

# add 4 coarse grid points in between to have some
leeway against roundoff

# grid step sizes are for coarsest anticipated
simulation dx = 1.5M


CarpetRegrid2::num_levels_1 = 7

CarpetRegrid2::position_x_1 = 0

CarpetRegrid2::radius_1[1]  = 960

CarpetRegrid2::radius_1[2]  = 228

CarpetRegrid2::radius_1[3]  = 114

CarpetRegrid2::radius_1[4]  = 66

CarpetRegrid2::radius_x_1[5]  = 35

carpetregrid2::radius_y_1[5]  = 35

carpetregrid2::radius_z_1[5]  = 24

CarpetRegrid2::radius_1[6]  = 13

CarpetRegrid2::radius_1[7]  = 6.5
```

```
CarpetRegrid2::num_levels_2 = 7
CarpetRegrid2::position_x_2 = -15
CarpetRegrid2::radius_2[1]  = 320
CarpetRegrid2::radius_2[2]  = 164
CarpetRegrid2::radius_2[3]  = 96
CarpetRegrid2::radius_2[4]  = 48
CarpetRegrid2::radius_2[5]  = 18
CarpetRegrid2::radius_2[6]  = 11
CarpetRegrid2::radius_2[7]  = 5.5

CarpetRegrid2::num_levels_3 = 7
CarpetRegrid2::radius_3[1]  = 320
CarpetRegrid2::radius_3[2]  = 164
CarpetRegrid2::radius_3[3]  = 96
CarpetRegrid2::radius_3[4]  = 48
CarpetRegrid2::radius_3[5]  = 18
CarpetRegrid2::radius_3[6]  = 11
CarpetRegrid2::radius_3[7]  =  5.5

carpetmask::excluded_surface[0]          = 0
carpetmask::excluded_surface[1]          = 1
carpetmask::excluded_surface_factor[0]   = 1
carpetmask::excluded_surface_factor[1]   = 1

CarpetTracker::surface_name[0] = "Righthand NS"
CarpetTracker::surface_name[1] = "Lefthand NS"

#------
# MODEL:
```

```
#------

ActiveThorns = "Meudon_Bin_NS"
HydroBase::initial_hydro         = "Meudon_Bin_NS"
ADMBase::initial_data            = "Meudon_Bin_NS"
ADMBase::initial_lapse           = "Meudon_Bin_NS"
ADMBase::initial_shift           = "zero"
ADMBase::initial_dtlapse         = "zero"
ADMBase::initial_dtshift         = "zero"

# change this to be the full path to the initial data
file
Meudon_Bin_NS::filename ="case208208.d"



#------------------EOS-----------------------------

EOS_Omni::poly_K = 123.613314525753

EOS_Omni::hybrid_gamma_th       = 1.8 #gamma thermal
from Hotokezaka et al. 2013 (arxiv.org/abs/1307.5888)

EOS_Omni::n_pieces              = 7 #3 for the core +
4 for the crust (Read et al 2009)

##k0=6.8011e-09 in cgs units and Kcu =
k0_cgs*(cu_to_cgs**(gamma-1)) where
cu_to_cgs=6.1762691458861658e+17 converts from CU to
g/cm^3

EOS_Omni::hybrid_k0             = 168.58190246577206
```

```
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_rho[0]         = 3.95160737e-11
EOS_Omni::hybrid_rho[1]         = 6.12595478e-07
EOS_Omni::hybrid_rho[2]         = 4.25474745e-06
EOS_Omni::hybrid_rho[3]         = 2.36741168e-04
EOS_Omni::hybrid_rho[4]         = 8.11472463e-04
EOS_Omni::hybrid_rho[5]         = 1.61910043e-03


#----------
# Numerics:
#----------

InitBase::initial_data_setup_method =
"init_some_levels"

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

HydroBase::timelevels            = 3
```

```
SpaceMask::use_mask      = "yes"

SphericalSurface::nsurfaces = 5
SphericalSurface::maxntheta = 39
SphericalSurface::maxnphi = 76

SphericalSurface::ntheta      [0] = 39
SphericalSurface::nphi        [0] = 76
SphericalSurface::nghoststheta[0] = 2
SphericalSurface::nghostsphi  [0] = 2
SphericalSurface::name        [0] = "Righthand NS"

SphericalSurface::ntheta      [1] = 39
SphericalSurface::nphi        [1] = 76
SphericalSurface::nghoststheta[1] = 2
SphericalSurface::nghostsphi  [1] = 2
SphericalSurface::name        [1] = "Lefthand NS"

SphericalSurface::ntheta      [3] = 39
SphericalSurface::nphi        [3] = 76
SphericalSurface::nghoststheta[3] = 2
SphericalSurface::nghostsphi  [3] = 2
SphericalSurface::set_spherical[3] = yes
SphericalSurface::radius      [3] = 100
SphericalSurface::name        [3] = "waveextract
surface at 100"

SphericalSurface::ntheta      [4] = 39
SphericalSurface::nphi        [4] = 76
```

```
SphericalSurface::nghoststheta[4] = 2
SphericalSurface::nghostsphi  [4] = 2
SphericalSurface::set_spherical[4] = yes
SphericalSurface::radius      [4] = 250
SphericalSurface::name        [4] = "waveextract
surface at 250"

#-----------
# Evolution:
#-----------

# test

HydroBase::initial_Bvec = "zero"
Hydrobase::Bvec_evolution_method   = "GRHydro"
GRHydro::transport_constraints = yes

HydroBase::evolution_method      = "GRHydro"

ADMMacros::spatial_order = 4
GRHydro::sources_spatial_order = 4

GRHydro::riemann_solver            = "HLLE"   #
Marquina is currently not supported by MP
GRHydro::recon_method              = "ppm"
GRHydro::GRHydro_stencil            = 3
GRHydro::bound                     = "flat"
GRHydro::rho_abs_min               = 1.e-11
GRHydro::GRHydro_atmo_tolerance    = 0.01

GRHydro::c2p_reset_pressure        = "yes"
```

```
GRHydro::GRHydro_eos_type           = "General"
GRHydro::GRHydro_eos_table          = "Ideal_Fluid"

# these can save some memory since they prevent MoL
from allocating unnecessary
# scratch space for saveandrestore variables
GRHydro::GRHydro_MaxNumSandRVars = 0

GRHydro::use_enhanced_ppm            = "yes"
# Parameters are defaults, which in turn are from
Colella & Sekora 2008 and
# McCorquodale & Colella 2011
GRHydro::sync_conserved_only     = "yes"
GRHydro::reconstruct_Wv          = "yes"
GRHydro::c2p_resort_to_bisection = "yes"
GRHydro::use_cxx_code            = "yes"

# 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::timelevels                     = 3

ML_BSSN::initial_boundary_condition  = "extrapolate-
gammas"
```

```
ML_BSSN::rhs_boundary_condition       = "NewRad"
Boundary::radpower                      = 2 # not
really needed I think but who knows what NewRad uses

ML_BSSN::harmonicN           = 1      # 1+log
ML_BSSN::harmonicF           = 2.0    # 1+log
ML_BSSN::ShiftGammaCoeff     = 0.75
ML_BSSN::AlphaDriver         = 0.0
ML_BSSN::BetaDriver          = 1.0
ML_BSSN::advectLapse         = 1.0
ML_BSSN::advectShift         = 1.0

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

ML_BSSN::UseSpatialBetaDriver = 1
ML_BSSN::SpatialBetaDriverRadius = 50

#ML_BSSN::apply_dissipation   = "never"

ML_BSSN::epsDiss              =0.0
```

```
Dissipation::epsdis = 0.1
Dissipation::order = 5
Dissipation::vars                       = "
        ML_BSSN::ML_log_confac
        ML_BSSN::ML_metric
        ML_BSSN::ML_trace_curv
        ML_BSSN::ML_curv
        ML_BSSN::ML_Gamma
        ML_BSSN::ML_lapse
        ML_BSSN::ML_shift
        ML_BSSN::ML_dtlapse
        ML_BSSN::ML_dtshift
"

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

IOBasic::outInfo_every = 1
IOBasic::outInfo_reductions = "maximum"
IOBasic::outInfo_vars  = "
 Carpet::physical_time_per_hour
 HydroBase::rho
 ML_ADMConstraints::ML_Ham
 SystemStatistics::maxrss_mb
 GRHydro::dens{reductions = 'sum maximum'}
 HydroBase::w_lorentz
"

IOScalar::outScalar_every      = 256 # every_coarse
```

```
IOScalar::all_reductions_in_one_file = "yes"
IOScalar::one_file_per_group   = "yes"
IOScalar::outScalar_reductions = "minimum maximum
average norm1 norm2"
IOScalar::outScalar_vars       = "
 ADMBase::lapse
 ADMBase::shift
 ADMBase::metric
 ADMBase::curv
 HydroBase::rho
 HydroBase::vel
 HydroBase::w_lorentz
 GRHydro::dens{reductions = 'minimum maximum average
norm1 norm2 sum'}
 SystemStatistics::process_memory_mb
 SphericalSurface::sf_radius
 ML_ADMConstraints::ML_Ham
"


IOASCII::one_file_per_group     = "yes"
IOASCII::compact_format  = "yes"


IOASCII::out0D_every     = 256 # every_coarse
IOASCII::out0D_vars      = "
 Carpet::timing
 QuasiLocalMeasures::qlm_scalars
 SphericalSurface::sf_active
 SphericalSurface::sf_valid
 SphericalSurface::sf_info
 SphericalSurface::sf_radius
 SphericalSurface::sf_origin
 SphericalSurface::sf_coordinate_descriptors
 Hydro_Analysis::Hydro_Analysis_rho_max_loc

Hydro_Analysis::Hydro_Analysis_rho_max_origin_distance
"


#Set these IOASCII options for initial data only:
IOASCII::out1D_every     = 0
IOASCII::out1D_d         = "no"
IOASCII::out1D_vars      = "
 HydroBase::rho
 HydroBase::vel
 ADMBase::lapse
 ADMBase::shift
 ADMBase::metric
 ADMBase::curv
 ML_ADMConstraints::ML_Ham
"


CarpetIOHDF5::one_file_per_group             = "no"
# this is required by multipatch
CarpetIOHDF5::open_one_input_file_at_a_time  = "yes"
CarpetIOHDF5::out2D_every                    = 1536
# 6*every coarse
CarpetIOHDF5::out2D_xy                       = "yes"
CarpetIOHDF5::out2D_xz                       = "no"
CarpetIOHDF5::out2D_yz                       = "no"
CarpetIOHDF5::out2D_xyplane_z                = 0.0
CarpetIOHDF5::out2D_vars      = "
  CarpetReduce::weight
  Grid::coordinates
  HydroBase::rho
  HydroBase::vel
  HydroBase::entropy
  HydroBase::press
  HydroBase::eps
  ADMBase::lapse
  ADMBase::shift
  ADMBase::metric
  ML_ADMConstraints::ML_Ham
 "

IOHDF5::out3D_every = 8192 # = 32*every_coarse
IOHDF5::out3D_vars  = "
 CarpetReduce::weight
 HydroBase::rho
 HydroBase::vel
 HydroBase::eps
 ADMBase::lapse
 ADMBase::shift
 ML_ADMConstraints::ML_Ham
 grid::coordinates
"

#------------------------------------------------------
# Analysis:
#------------------------------------------------------
Hydro_Analysis::Hydro_Analysis_comp_rho_max = "true"
Hydro_Analysis::Hydro_Analysis_rho_max_loc_only_positi
ve_x = "true"
Hydro_Analysis::Hydro_Analysis_comp_rho_max_origin_dis
tance = "yes"
```

```
Hydro_Analysis::Hydro_Analysis_average_multiple_maxima
_locations = "yes"
Hydro_Analysis::Hydro_Analysis_interpolator_name =
"Lagrange polynomial interpolation (tensor product)"


NSTracker::NSTracker_SF_Name          = "Righthand NS"
NSTracker::NSTracker_SF_Name_Opposite = "Lefthand NS"
NSTracker::NSTracker_max_distance = 10
NSTracker::NSTracker_verbose = "yes"
NSTracker::NSTracker_tracked_location =
"Hydro_Analysis::Hydro_Analysis_rho_max_loc"


QuasiLocalMeasures::num_surfaces   = 2
QuasiLocalMeasures::spatial_order  = 4
QuasiLocalMeasures::interpolator = "Lagrange
polynomial interpolation"
QuasiLocalMeasures::interpolator_options = "order=4"
QuasiLocalMeasures::surface_name[0] = "waveextract
surface at 100"
QuasiLocalMeasures::surface_name[1] = "waveextract
surface at 250"



######################################################
######################################################
# Wave extraction
######################################################
######################################################



WeylScal4::offset                  = 1e-8
WeylScal4::fd_order                = "4th"
```

```
WeylScal4::verbose                   = 0

Multipole::nradii = 8
Multipole::out_every = 128
Multipole::radius[0] = 45
Multipole::radius[1] = 70
Multipole::radius[2] = 100
Multipole::radius[3] = 125
Multipole::radius[4] = 150
Multipole::radius[5] = 200
Multipole::radius[6] = 250
Multipole::radius[7] = 300
Multipole::variables = "WeylScal4::Psi4r{sw=-2
cmplx='WeylScal4::Psi4i' name='Psi4'}"
Multipole::l_max = 6

#------------------------------------------------------
# Checkpoint/Recovery:
#------------------------------------------------------
IOHDF5::checkpoint                  = "yes"
IO::checkpoint_dir                  = "../checkpoint"
IO::checkpoint_ID                   = "yes"
IO::checkpoint_every_walltime_hours = 6.0
CarpetIOHDF5::checkpoint_every_divisor = 55552
IO::checkpoint_keep=2
IO::checkpoint_on_terminate         = "yes"

IO::recover     = "autoprobe"
IO::recover_dir = "../checkpoint"
```

```
#-------------------------------------------------------
# AHFinderDirect:
#-------------------------------------------------------


AHFinderDirect::find_every = 0


AHFinderDirect::run_at_CCTK_POST_RECOVER_VARIABLES =
no


AHFinderDirect::move_origins            = yes
AHFinderDirect::reshape_while_moving    = yes
AHFinderDirect::predict_origin_movement = yes


AHFinderDirect::geometry_interpolator_name = "Lagrange
polynomial interpolation"
AHFinderDirect::geometry_interpolator_pars = "order=4"
AHFinderDirect::surface_interpolator_name  = "Lagrange
polynomial interpolation"
AHFinderDirect::surface_interpolator_pars  = "order=4"


AHFinderDirect::output_h_every = 128


AHFinderDirect::N_horizons = 1


AHFinderDirect::which_surface_to_store_info
[1] = 0
AHFinderDirect::reset_horizon_after_not_finding
[1] = no
AHFinderDirect::initial_guess__coord_sphere__radius
[1] = 1.3528


#-------------------------------------------------------
```

```
# Control
#-------------------------------------------------------

ActiveThorns = "Trigger"

Trigger::Trigger_Number = 3

# Check for lapse < 0.5, and if so, set this trigger
to 'once'
Trigger::Trigger_Once            [0] = 0
Trigger::Trigger_Checked_Variable[0] = "ADMBase::alp"
Trigger::Trigger_Reduction       [0] = "minimum"
Trigger::Trigger_Relation        [0] = "<"
Trigger::Trigger_Checked_Value   [0] = 0.5
Trigger::Trigger_Reaction               [0] =
"steerparam"
Trigger::Trigger_Steered_Parameter_Thorn[0] =
"Trigger"
Trigger::Trigger_Steered_Parameter_Name [0] =
"Trigger_Once[0]"
Trigger::Trigger_Steered_Parameter_Value[0] = "1"

# if lapse < 0.2 enable AHFinder
Trigger::Trigger_Once            [1] = 1
Trigger::Trigger_Checked_Variable[1] = "ADMBase::alp"
Trigger::Trigger_Reduction       [1] = "minimum"
Trigger::Trigger_Relation        [1] = "<"
Trigger::Trigger_Checked_Value   [1] = 0.3
Trigger::Trigger_Reaction               [1] =
"steerparam"
Trigger::Trigger_Steered_Parameter_Thorn[1] =
"AHFinderDirect"
```

```
Trigger::Trigger_Steered_Parameter_Name [1] =
"find_every"
Trigger::Trigger_Steered_Parameter_Value[1] = "128"


# if trigger 0 was set 'once' (lapse < 0.5)
# add refinement level
Trigger::Trigger_Once                   [2] = 1
Trigger::Trigger_Checked_Variable       [2] = "param"
Trigger::Trigger_Checked_Parameter_Thorn[2] =
"Trigger"
Trigger::Trigger_Checked_Parameter_Name [2] =
"Trigger_Once[0]"
Trigger::Trigger_Relation               [2] = "=="
Trigger::Trigger_Checked_Value          [2] = 1
Trigger::Trigger_Reduction              [2] = ""
Trigger::Trigger_Reaction               [2] =
"steerscalar"
Trigger::Trigger_Steered_Scalar         [2] =
"CarpetRegrid2::num_levels[0]"
Trigger::Trigger_Steered_Scalar_Value   [2] = "7"


#-------VolumeIntegrals_GRMHD----------------
# We use this to track the NS movement
# Uncomment the following lines to have moving boxes
# also increase by one the value of
carpetregrid2::num_levels_2 and
carpetregrid2::num_levels_3


VolumeIntegrals_GRMHD::NumIntegrals = 6
VolumeIntegrals_GRMHD::VolIntegral_out_every = 64
VolumeIntegrals_GRMHD::enable_file_output = 1
VolumeIntegrals_GRMHD::outVolIntegral_dir =
"volume_integration"
```

```
VolumeIntegrals_GRMHD::verbose = 1


## The AMR centre will only track the first referenced
integration quantities that track said centre.

##   Thus, centeroflapse output will not feed back
into the AMR centre positions.


VolumeIntegrals_GRMHD::Integration_quantity_keyword[1]
= "one"

VolumeIntegrals_GRMHD::Integration_quantity_keyword[2]
= "centerofmass"

VolumeIntegrals_GRMHD::Integration_quantity_keyword[3]
= "one"

VolumeIntegrals_GRMHD::Integration_quantity_keyword[4]
= "centerofmass"

VolumeIntegrals_GRMHD::Integration_quantity_keyword[5]
= "one"

VolumeIntegrals_GRMHD::Integration_quantity_keyword[6]
= "restmass"


#Use output from volume integral to move AMR box
centre 2


VolumeIntegrals_GRMHD::volintegral_sphere__center_x_in
itial         [2] = -15.0

VolumeIntegrals_GRMHD::volintegral_inside_sphere__radi
us            [2] =  10.0

VolumeIntegrals_GRMHD::amr_centre__tracks__volintegral
_inside_sphere[2] =  1

VolumeIntegrals_GRMHD::volintegral_sphere__center_x_in
itial         [3] = -15.0

VolumeIntegrals_GRMHD::volintegral_inside_sphere__radi
us            [3] =  10.0
```

```
#Use output from volume integral to move AMR box
centre 3


VolumeIntegrals_GRMHD::volintegral_sphere__center_x_in
itial         [4] =  15.0

VolumeIntegrals_GRMHD::volintegral_inside_sphere__radi
us            [4] =  10.0

VolumeIntegrals_GRMHD::amr_centre__tracks__volintegral
_inside_sphere[4] =  2

VolumeIntegrals_GRMHD::volintegral_sphere__center_x_in
itial         [5] =  15.0

VolumeIntegrals_GRMHD::volintegral_inside_sphere__radi
us            [5] =  10.0
```