Fast Ejecta from Binary Neutron Star Mergers

by

Coleman Dean

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Physics
University of Alberta

Abstract

We investigate the production of fast ejecta from the contact interface in coalescing binary neutron stars. This ejecta expands rapidly enough (velocities $\gtrsim 0.6c$) that the r-process freezes out, generating an ultraviolet precursor to the kilonova powered by the decay of free neutrons. Previous work using grid-based simulations has reached inconclusive results about the amount of this fast ejecta produced in the merger, partly due to an inability to numerically resolve the surface layers of the colliding stars. These results stand in contrast to those from simulations using particle-based hydrodynamic methods (SPH), which yield much higher values of free neutron ejecta. Here we report the development and preliminary results of axisymmetric grid-based merger simulations that include the dominant physics in approximate form in order to examine the production of these ejecta at high resolutions not achievable in three dimensional models. At the resolutions studied thus far ({281.3, 140.6, 70.3} m equivalent to {2.2, 1.1, 0.56} % of the stellar radius, or ∼{70.3, 35.2, 17.6} times the pressure scale height at 99% of the neutron star radius), we find that the mass of fast ejecta has only a weak resolution dependence, suggesting that simulations of this ejecta component may be close to converging. Our results therefore cannot yet resolve the tension in ejecta masses produced by

SPH and grid-based binary neutron star merger simulations as a pure numerical resolution effect.

Preface

This thesis is original work by Coleman Dean. The FLASH simulation was built starting from an existing setup originally developed by Rodrigo Fernández to collide white dwarfs head-on. We have a journal manuscript in preparation, some text (mostly in the methods section, particularly Section 3.2) has been taken from the manuscript and adapted to this thesis format.

"the only difference between screwing around and science is writing it down."

Adam Savage

Acknowledgements

I would like to thank Rodrigo Fernández for his guidance and mentor-ship in conducting this research. I would like to thank Steven Fahlman, and Mario Ivanov for helpful conversations.

This research was supported by the Alberta Government through the Queen Elizabeth II Graduate Scholarship (2018/2019), and the Alberta Graduate Excellence Scholarship (2019/2020), as well as the National Sciences and Engineering Research Council of Canada (NSERC) through Discovery Grant RGPIN-2017-04286, and by the Faculty of Science at the University of Alberta. The software used in this work was in part developed by the DOE NNSA-ASC OASCR Flash Center at the University of Chicago. This research was enabled in part by support provided by WestGrid (`www.westgrid.ca`), the Shared Hierarchical Academic Research Computing Network (SHARCNET, `www.sharcnet.ca`), Calcul Québec (`www.calculquebec.ca`), SciNet (`https://www.scinethpc.ca/`), the Niagara cluster (`https://www.scinethpc.ca/niagara/`), and Compute Canada (`www.computecanada.ca`). Graphics were developed with `matplotlib` (Hunter, 2007), and `VisIt` (Childs et al., 2012).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Second only to black holes, neutron stars are the densest objects in the universe. The first neutron star, discovered as a radio pulsar (PSR B1919+21), a regularly repeating radio source, was found by Hewish et al. (1968). However, pulsars were later found to emit in other wavelengths as well. Pulsar emission is generally understood as arising from neutron stars with a dipole magnetic field inclined relative to their axis of rotation. Particles accelerated by this dipole magnetic field emit syncrotron radiation in a cone, in the direction of the dipolar axis (Karttunen et al., 2017). Neutron star masses can be measured from binary pulsars through the use of pulse timing, thereby measuring orbital parameters of the binary system to derive the individual masses (e.g., Hulse and Taylor 1975).

The equation of state of dense matter, particularly at the extreme densities present within a neutron star (which exceed the density of atomic nuclei), is largely unknown due to the non-trivial nuclear physics at work (Shapiro and Teukolsky, 1983). Similarly, due to their extreme density, neutron stars exist in the general relativistic regime with implications on their structure. Numerous theoretical models for this equation of state have been proposed in order to describe the relation between pressure, density, and temperature within a neutron star. Plausible models can be constrained by both theoretical limits from physical effects (e.g., causality of the speed of sound, $c_s < c$; Read et al. 2009), and empirical limits imposed by observations of neutron stars.

The coalescence of binary neutron stars is the natural outcome of the orbital decay due to the emission of gravitational waves, as in the Hulse-Taylor pulsar. The timescale for decay is long, however, with most galactic binaries taking longer than a Hubble time to merge. Nevertheless, the immense volume of space to which laser interferometers such as `advanced LIGO` or `advanced Virgo` are sensitive to gravitational waves from binary neutron star mergers from distant galaxies, leads to a non-negligible detectable binary neutron star merger rate ($290-2810\ \mathrm{Gpc}^{-3}\mathrm{yr}^{-1}$, by Abbott et al. 2020). Despite being a rare occurrence, these mergers were long predicted to be astrophysically interesting: as a source for the production of heavy elements (Lattimer and Schramm, 1974) and a progenitor of short Gamma Ray Bursts (Paczynski 1986 and Eichler et al. 1989).

With the advent of `advanced LIGO`, and `advanced Virgo`, gravitational waves have been detected directly. Initially, this was only the case for binary black hole mergers (Abbott et al., 2016), but on August 17th, 2017 the first binary neutron star merger `GW170817` was observed. After the detection alert, electromagnetic follow-up observations confirmed that neutron star mergers are both progenitors of sGRBs and produce r-process elements via the detection of an associated high energy emission (`GRB170817`) and kilonova (`AT 2017 gfo`) (Abbott et al. 2017a, Abbott et al. 2017b, and references therein).

The material ejected from merging binary neutron stars, which is hot expanding matter rich in neutrons, produces an ideal environment for the production of elements with atomic number $Z > 30$ via the rapid neutron capture process (r-process) (Cowan et al., 2019). The majority of the ejected material moves at sub-relativistic velocities ($v < 0.3c$). Most of the isotopes initially produced in this process are unstable and undergo beta-decay, thereby powering an optical / infrared transient commonly known as a kilonova (Li and

Paczyński 1998, and Metzger et al. 2010).

1.1 Fast Ejecta and the Neutron Precursor

In addition to the kilonova, as first envisioned by Kulkarni (2005), and worked out by Metzger et al. (2015), material ejected rapidly enough (expansion velocity $v \gtrsim 0.6\ c$) can generate a precursor to the kilonova powered by the decay of free neutrons on a timescale of hours after the merger. These neutrons are expected to travel fast enough at the forefront of the expanding ejecta that neutron capture is inefficient, thereby freezing out the r-process. Ejection of fast material in a shock breakout was witnessed initially in simulations by Sekiguchi et al. (2011), Paschalidis et al. (2012), and Bauswein et al. (2013). The r-process freeze-out in these fast travelling neutrons seen in Bauswein et al. (2013), was further discussed by Goriely et al. (2014), and the neutron precursor was predicted to peak in the Ultra-Violet on a timescale of hours after merger by Metzger et al. (2015). In the case of `GW170817`, the first optical counterpart was detected 10 hours after the merger due to the timing of the event and the telescopes available to observe it, thus a neutron-powered precursor, if existing, would not have been seen. Further, Arcavi (2018) found that the uncertainty in the ultraviolet flux from the first observations of `GW170817` was too large to constrain early emission mechanisms or free neutron ejecta mass. More recent observing runs of `LIGO/Virgo` issue public alerts on a timescale of seconds, so optical follow up on a timescale of hours is technically feasible.

Other mechanisms have been suggested to generate an Ultra-Violet precursor such as cocoon emission, where a relativistic jet punches through previous ejected material creating a heated, shocked cocoon (Piro and Kollmeier, 2018), or reheating such that the merger remnant injects energy into ejected material via a magnetized wind (Metzger et al., 2018). Here, however, we focus on the fast ejecta from the neutron star contact interface. While this fast ejecta has previously been seen in sufficient amounts to produce a detectable precursor in Smoothed Particle Hydrodynamics (SPH) simulations (Bauswein et al. 2013), grid based simulations have shown reduced ejection of fast material in tension with the SPH finding (Kiuchi et al. 2017, Radice et al. 2018b, and Ishii et al. 2018). It has been suggested by Kyutoku et al. (2014) that in order to resolve the fastest components of the ejecta from the contact interface (a shell of thickness $\lesssim R_{crust}/30$, where the crust of the neutron star has thickness $R_{crust} \approx 1km$), <10m spatial resolution would be required, to accurately capture the dynamics. This is currently unfeasible with a three dimensional simulation.

Here we develop and carry out axisymmetric grid-based merger simulations that include, in approximate form, the dominant physics necessary for the ejection of material in the contact plane, in order to examine the production of this ejecta at high resolutions not achievable in three dimensional models. We present preliminary results of our study at resolutions of {281.3, 140.6, 70.3} m equivalent to {2.2, 1.1, 0.56} % of the stellar radius, or ∼{70.3, 35.2, 17.6} times the pressure scale height at 99% R_{NS}.

The structure of this thesis is the following. Chapter 2 discusses the design of our numerical experiment, the coordinate system chosen, orbital motion, gravitational wave emission, and inertial forces. In Chapter 3, the computational method is discussed. This includes the equation of state chosen, physical approximations implemented, initial set-up, hydrodynamics, models evolved, and how data is extracted from `FLASH`. Chapter 4 covers the results of the simulations run, Chapter 5 discusses conclusions, and Appendix A discusses optimizations implemented into the code.

Chapter 2

Numerical Experiment Design

In this chapter we describe the physical assumptions that enter our simulations. We model the collision in two dimensional cylindrical coordinates, using Newtonian hydrodynamics, and a piece-wise polytropic equation of state with approximate thermal effects in order to investigate whether the ejecta velocity distribution (particularly the high velocity tail) has a spatial resolution dependence, reducing the computational workload relative to a 3D simulation while maintaining the hydrodynamic forces responsible for this contact plane ejecta. In the following sections I will discuss the implemented geometry and physics used.

2.1 2D Cylindrical Coordinates

We set up our simulation in two-dimensional cylindrical coordinates on the $\rho - z$ plane, with ρ the cylindrical radius and z the azimuthal symmetry axis, such that our domain is rectangular (twice as long in the z direction as it is in the ρ direction). Two neutron stars are placed symmetrically about the equatorial plane, such that the star centers are at $(0, z_{center})$, and $(0, -z_{center})$. While the neutron stars appear semi-circular in shape with rotational symmetry about the z-axis, if we consider the φ-direction the full spherical neutron star is being modeled as illustrated in Figure 2.1.

We account for orbital motion by employing a co-rotating frame with constant angular velocity ω about the cylindrical radial axis (ρ). We choose this angular frequency to match that of the neutron stars at $t = 0$. Due to gravitational radiation losses that will be discussed in Section 2.3, however, the stars after time $t = 0$ will not rotate as a rigid body with angular velocity ω, thus the stellar fluid rotates relative to the frame for $t > 0$. This co-rotational reference frame was chosen as it is the only way to model an orbiting binary system in two dimensions, given that the axis-symmetry reproduces the stars spherical shape, and each star orbits the ρ-axis. A Keplerian orbit test, maintaining orbital seperation without the emission of gravitational waves confirms this.

2.2 Orbital Motion

Orbital motion is accounted for by keeping track of the specific angular momentum of stellar material around the orbital axis. Given the assumed symmetries, this quantity is a scalar. The stars orbit around the ρ axis, and due to the use of an initially co-rotating reference frame, the stars are assumed to remain approximately in the simulation plane throughout the short inspiral process. Thus the inspiral of the binary neutron star system in the co-rotating reference frame appears as velocity of each star toward the origin along the z-axis.

Due to the emission of gravitational waves, the stars inspiral together. The rate of change of the orbital separation d for two point masses with respect to time is given by (Padmanabhan, 2010):

$$\left\langle \frac{\mathrm{d}d_{GW}}{\mathrm{d}t} \right\rangle = -\frac{64G^3}{5c^5} \frac{m_1 m_2 (m_1 + m_2)}{d^3 (1 - e^2)^{7/2}} \left(1 + \frac{73}{24} e^2 + \frac{37}{96} e^4 \right), \tag{2.1}$$

where the angle brackets denote the average rate of decrease due to the emission of gravitational waves, $m_{1,2}$ are the masses of the two neutron stars, G is the gravitational constant, c is the speed of light, and e is the eccentricity of the

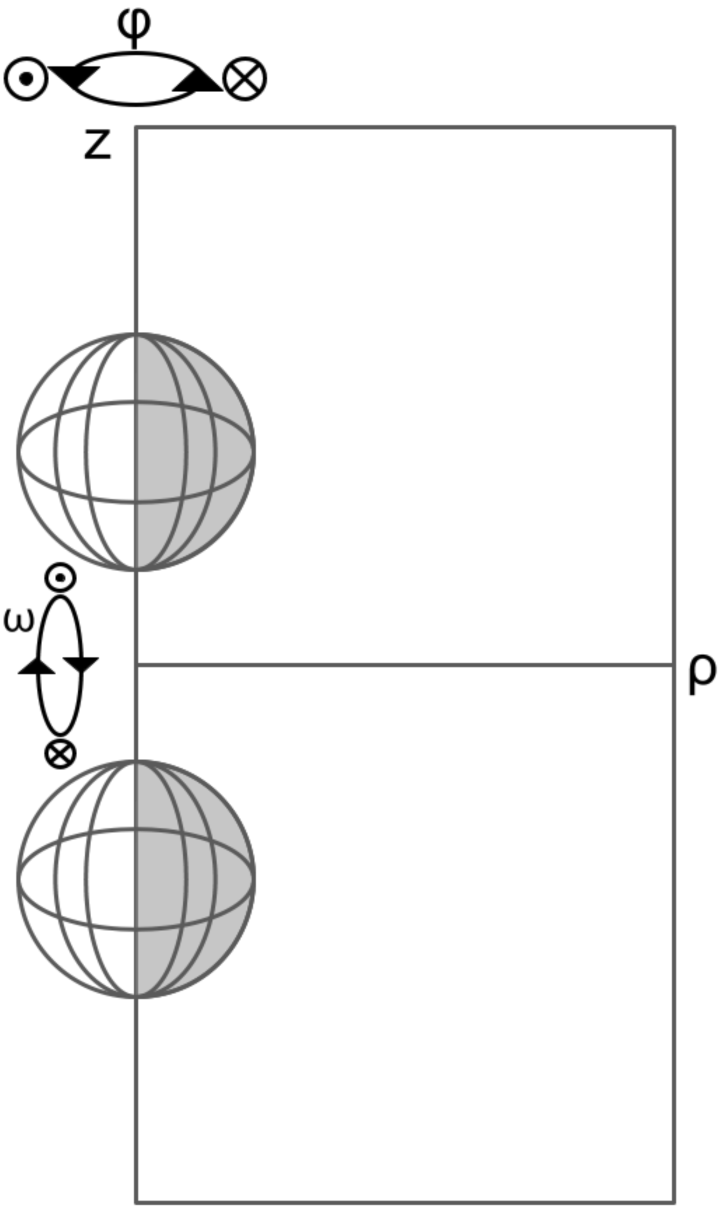

Figure 2.1: Schematic of two dimensional cylindrical geometry of the merger simulation with rotation of the frame indicated by the rotation rate ω, and its direction. The axis of cylindrical symmetry is z, with azimuthal angle φ. Spherical geometry of the neutron stars is illustrated representing the effect of cylindrical symmetry.

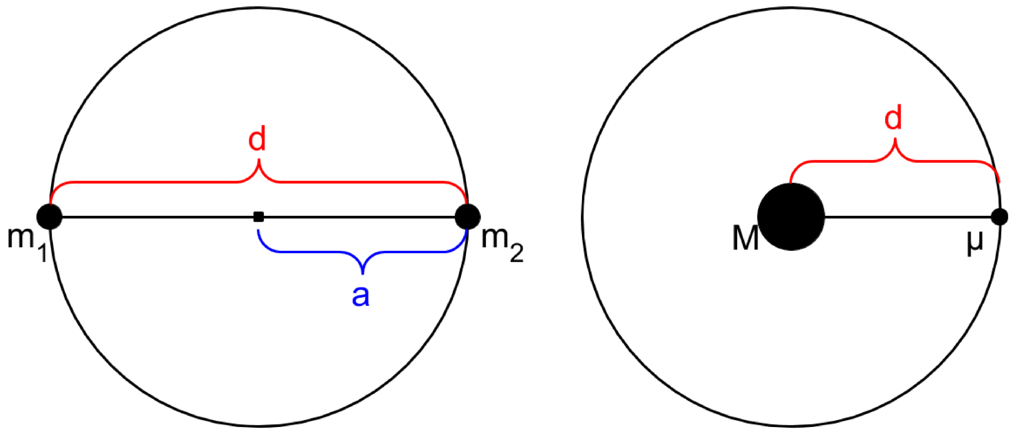

Figure 2.2: Left: Schematic of the orbital configuration in the laboratory frame, viewed from above. Solid black circles denote the neutron star positions, the centre of mass of the system for an equal mass binary is denoted by a black square. Right: Same as the left pane but in the centre of mass frame. Reduced mass μ orbits a central potential of mass M at a distance equal to the full binary orbital separation d.

orbit. Here we define the orbital separation between the orbiting neutron stars to be d, and the semi-major axis (half the orbital separation) for a circular orbit to be a as shown in Figure 2.2 left pane. In the limiting case of a non-eccentric orbit ($e = 0$) and using the definitions:

$$\mu = \frac{m_1 m_2}{m_1 + m_2}, \tag{2.2}$$

and

$$M = m_1 + m_2, \tag{2.3}$$

where μ is the reduced mass of the system, and M is the total mass of both neutron stars, we can rewrite Equation 2.1 as:

$$\left\langle \frac{\mathrm{d}d_{GW}}{\mathrm{d}t} \right\rangle = -\frac{64G^3}{5c^5}\frac{\mu M^2}{d^3}. \tag{2.4}$$

In particular, we limit ourselves to an equal mass binary such that $M = 2m$ and $\mu = m/2$.

Since we are modelling an inspiral, we know the initial orbital speed will be smaller than the Keplerian orbital velocity for a given initial orbital separation d,

$$v_{orb} < v_K = \frac{1}{2}\sqrt{\frac{GM}{d}}, \tag{2.5}$$

where v_K is the initial Keplerian velocity of each star in the laboratory frame. Thus our initial orbital parameters for each of the neutron stars are:

$$\begin{cases} |a_j| & = z_{center} \\ |v_{z,j}| & = \frac{1}{2}\left|\dot{d}_{GW}\right| = \frac{1}{2}\frac{64}{5}\frac{G^3}{c^5}\frac{\mu M^2}{d^3} \\ v_{orb,j} & < v_K = \frac{1}{2}\sqrt{\frac{GM}{d}}. \end{cases} \tag{2.6}$$

Where a_j is positive (negative), and $v_{z,j}$ is negative (positive) for the neutron star above (below) the ρ-axis. Here $j = 1, 2$ denotes which neutron star we are referring to, $v_{z,j}$ is the initial inward radial velocity for neutron star j due to the emission of gravitational waves, in our case this velocity is directed along the $z-$axis, and $v_{orb,j}$ is the initial orbital velocity of neutron star j, which defines the angular rotation rate of the co-rotating reference frame ω. To derive the initial orbital velocity we can use a conservation of energy argument in the centre of mass frame (Figure 2.2, right pane) comparing the energy of a binary at infinite separation ($d \to \infty$ limit) E_∞, to our initial condition ($d = 2 \cdot z_{center}$) E_i, see Figure 2.3.

$$U_{G,\infty} + E_{K,\infty} = U_{G,i} + E_{K,i} + |E_{loss,GW}|, \tag{2.7}$$

where U_G is the gravitational potential energy, E_K is the kinetic energy, the ∞, and i labels represent the two states (infinite separation ∞, and initial separation i), and $|E_{loss,GW}|$ is the absolute value of the total energy that is radiated away from the binary system due to gravitational wave emission to

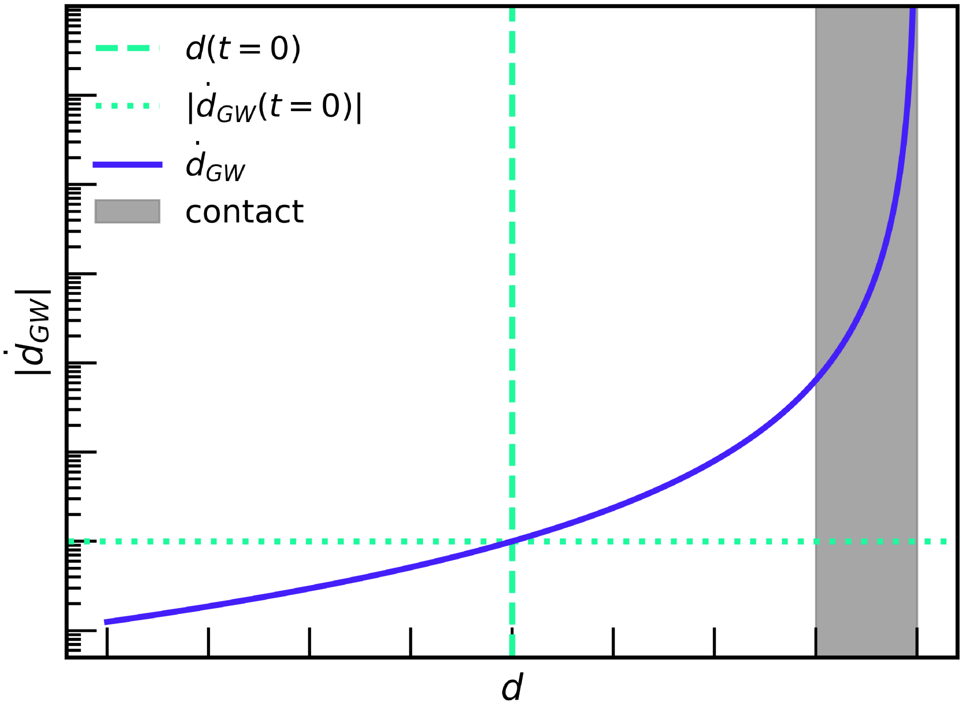

Figure 2.3: Schematic evolution of the rate of change of the orbital separation $\dot{d}_{GW}$ with respect to orbital separation over the relevant timescale for the simulation. The grey highlighted area denotes the collision phase of the inspiral, while the vertical dashed line and horizontal dotted lines represent the initial separation and initial rate of change of orbital separation respectively.

bring it from infinite separation to the initial separation considered in this simulation (thus $U_{G,i}$ is more negative than it would be without GW emission). We can rewrite this equation in terms of energy per unit reduced mass,

$$u_{G,\infty} + e_{K,\infty} = u_{G,i} + e_{K,i} + \frac{1}{\mu}|E_{loss,GW}|, \tag{2.8}$$

where $u_G = U_G/\mu$, and $e_K = E_K/\mu$. The average rate of energy loss due to the emission of gravitational waves $\left\langle \frac{\mathrm{d}E}{\mathrm{d}t} \right\rangle$ is given by (Padmanabhan, 2010):

$$\left\langle \frac{\mathrm{d}E}{\mathrm{d}t} \right\rangle = -\frac{32G^4}{5c^5}\frac{m_1^2 m_2^2 (m_1+m_2)}{d^5(1-e^2)^{7/2}}\left(1+\frac{73}{24}e^2+\frac{37}{96}e^4\right). \tag{2.9}$$

Once again considering the limiting case of $e = 0$, and taking advantage of Equations 2.2 & 2.3 we can rewrite Equation 2.9 as

$$\left\langle \frac{\mathrm{d}E}{\mathrm{d}t} \right\rangle = -\frac{32G^4}{5c^5}\frac{\mu^2 M^3}{d^5}. \tag{2.10}$$

This energy loss rate can then be used to calculate the total energy loss due to the emission of gravitational waves between our two energy states (infinite separation, and initial separation):

$$\begin{aligned}
E_{loss,GW} &= \int_0^{t_i} -\frac{32}{5}\frac{G^4}{c^5}\frac{\mu^2 M^3}{d^5}\mathrm{d}t \\
E_{loss,GW} &= -\frac{32}{5}\frac{G^4}{c^5}\mu^2 M^3 \int_0^{t_i} \frac{1}{d^5}\mathrm{d}t \\
E_{loss,GW} &= -\frac{32}{5}\frac{G^4}{c^5}\mu^2 M^3 \int_{\infty}^{2\cdot z_{center}} \frac{1}{d^5}\frac{\mathrm{d}d}{\frac{\mathrm{d}d}{\mathrm{d}t}} \\
E_{loss,GW} &= -\frac{32}{5}\frac{G^4}{c^5}\mu^2 M^3 \int_{\infty}^{2\cdot z_{center}} \frac{1}{d^5}\frac{\mathrm{d}d}{-\frac{64}{5}\frac{G^3}{c^5}\frac{\mu M^2}{d^3}} \qquad (2.11) \\
E_{loss,GW} &= \frac{G}{2}\mu M \int_{\infty}^{2\cdot z_{center}} \frac{1}{d^2}\mathrm{d}d \\
E_{loss,GW} &= \frac{G}{2}\mu M \Big[-\frac{1}{d}\Big]_{\infty}^{2\cdot z_{center}} \\
E_{loss,GW} &= -\frac{G}{2}\frac{\mu M}{2\cdot z_{center}} = -\frac{G}{2}\frac{\mu M}{2\cdot a} = -\frac{G}{2}\frac{\mu M}{d}.
\end{aligned}$$

Replacing this result into Equation 2.7 we get,

$$\cancelto{0}{u_{G,\infty}} + \cancelto{0}{e_{K,\infty}} = u_{G,i} + e_{K,i} + \frac{1}{\mu}\frac{G}{2}\frac{\mu M}{d} \qquad (2.12)$$

Where we choose to anchor our potential energy such that $\lim_{a\to\infty} u_G \to 0$, and the kinetic energy at infinity is neglected due to its small magnitude. Substituting into Equation 2.12:

$$\begin{aligned}
0 &= u_{G,i} + e_{K,i} + \frac{G}{2}\frac{M}{d} \\
0 &= -\frac{GM}{d} + \frac{1}{2}v_{rm}^2 + \frac{1}{2}\frac{GM}{d} \qquad (2.13) \\
-\frac{1}{2}\frac{GM}{d} &= -\frac{GM}{d} + \frac{1}{2}v_{rm}^2,
\end{aligned}$$

where v_{rm} is the total speed of the reduced mass, and we can see that the total energy in the initial state is different from zero due to the energy radiated away as gravitational waves. Solving for v_{rm}^2:

$$v_{rm}^2 = \frac{GM}{d} \tag{2.14}$$

The initial speed of the reduced mass v_{rm} can be broken down further into orbital and radial components:

$$v_{rm} = \sqrt{v_{orb,rm}^2 + v_{z,rm}^2}, \tag{2.15}$$

substituting this into Equation 2.13 we get

$$\begin{aligned} v_{orb,rm}^2 + v_{z,rm}^2 &= \frac{GM}{d} \\ v_{orb,rm}^2 + \dot{d}_{GW}^2 &= \frac{GM}{d} \\ v_{orb,rm} &= \sqrt{\frac{GM}{d} - \dot{d}_{GW}^2} \end{aligned} \tag{2.16}$$

Transforming to the laboratory frame to recover the individual neutron star initial orbital velocities in the code, we get:

$$\boxed{|v_{orb,j}| = \frac{1}{2}\sqrt{\frac{GM}{d} - \dot{d}_{GW}^2}}, \tag{2.17}$$

where $\dot{d}_{GW}$ is given by equation 2.4. Similarly the initial z velocity in the laboratory frame is given by Equation 2.6. The co-rotating reference frame angular rotation rate is obtained from the orbital velocity of the reduced mass:

$$\omega = \frac{1}{d}\sqrt{\frac{GM}{d} - \dot{d}_{GW}^2}. \tag{2.18}$$

With our initial orbital parameters derived for the simulation, we can now discuss how these are implemented in `FLASH`. Since we are using a co-rotating reference frame, orbital motion occurs perpendicular to the ρ-z plane. Motion is tracked with a scalar specific angular momentum variable, initially defined as:

$$
\begin{aligned}
\mathbf{L}_{orb,j} &= m_j(\mathbf{r_j} \times \mathbf{v_j}) \\
\mathbf{L}_{orb,j} &= m_j||\mathbf{r_j}||||\mathbf{v_j}||\cancelto{1}{sin(\theta)}\hat{n} \\
\boldsymbol{\ell}_{orb,j} &= v_{orb,j}a_j\hat{\rho} \\
\ell_{orb,j} &= ||\boldsymbol{\ell}_{orb,j}|| = v_{orb,j}a_j
\end{aligned}
\tag{2.19}
$$

Where $\mathbf{L}_{orb}$ is the orbital angular momentum, $\boldsymbol{\ell}_{orb} = \mathbf{L}_{orb}/m$, is the orbital angular momentum per unit mass, θ is the angle between $\mathbf{r}$ and $\mathbf{v}$ ($\sin(\theta) \rightarrow 1$ for a nearly circular orbit) , where $||\mathbf{r}||$ is the distance from the rotation axis, and $\mathbf{v}$, the orbital velocity vector ($-v_{orb,i} \cdot \mathrm{sgn(z)}\hat{\varphi}$). Lastly, $\hat{n}$ is the unit vector perpendicular to $\hat{r}$ and $\hat{v}$, $\hat{\rho} = \hat{z} \times (-\hat{\varphi})$ for $z > 0$, and $\hat{\rho} = (-\hat{z}) \times \hat{\varphi}$ for $z < 0$. The initial specific angular momentum scalar of the system is shown in Figure 2.4.

The angular momentum is implemented as a "mass scalar" variable in `FLASH`, initially set according to Equation 2.19, which is then allowed to advect with material as it moves throughout the domain. Besides advection, the specific angular momentum is modified by gravitational wave emission and torques due to the Coriolis force in an operator-split way. This will be discussed in more detail later in this chapter.

2.2.1 Fictitious Forces

The use of a co-rotating reference frame introduces fictitious forces. Here I will derive those forces, following Fetter and Walecka (2003), and discuss their implementation in `FLASH`.

Consider an inertial (non-accelerating) reference frame with a set of orthonormal coordinate axes $\hat{e}_i$ where an arbitrary vector $\mathbf{v}$ can be written as a linear combination of those 3 orthogonal coordinates:

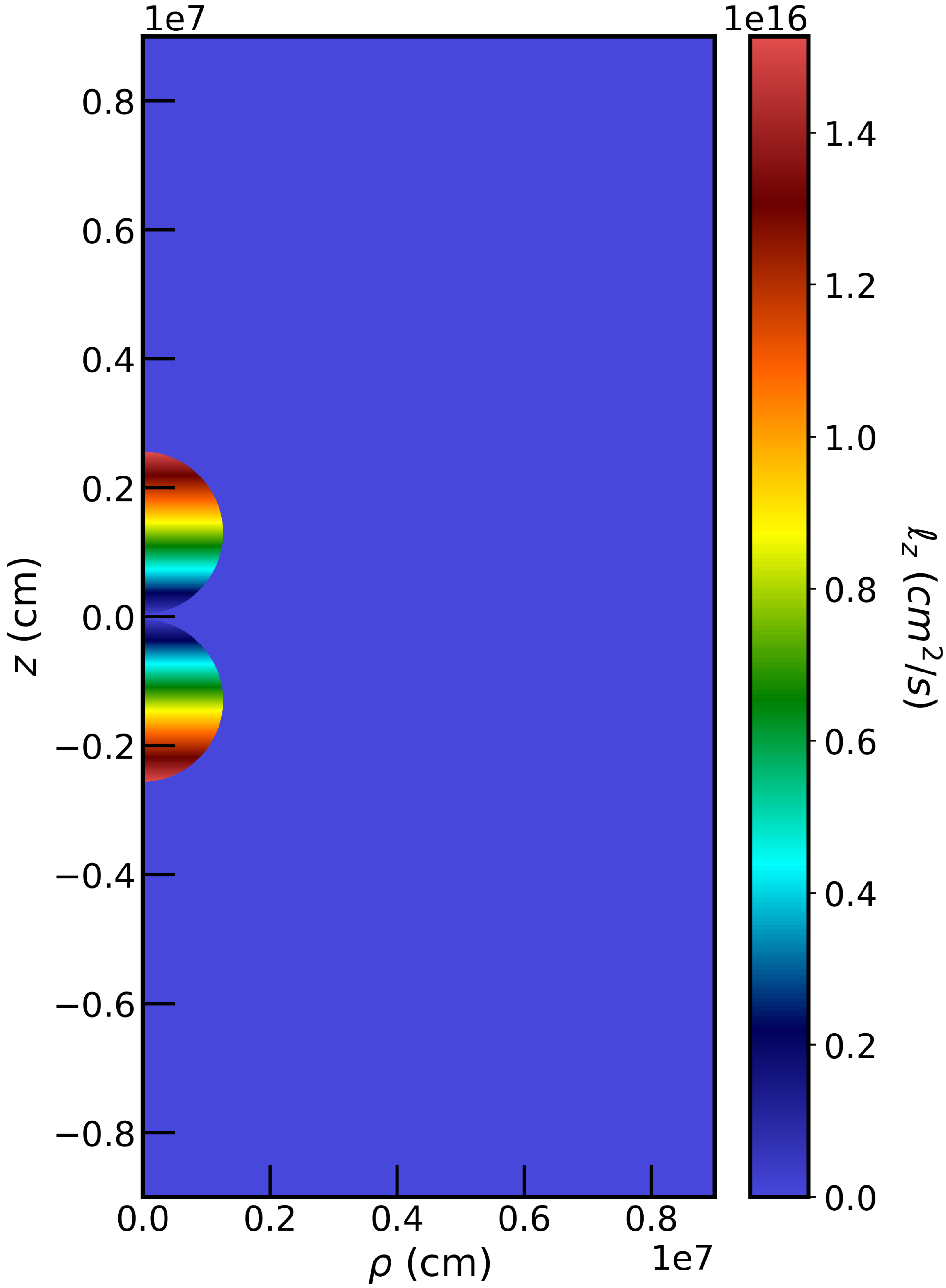

Figure 2.4: Colour map of initial specific angular momentum of the system.

$$\mathbf{v} = v^1\hat{e}_1 + v^2\hat{e}_2 + v^3\hat{e}_3 = v^i\hat{e}_i. \tag{2.20}$$

I use Einstein summation notation from here forward. Now consider a second set of orthogonal coordinate axes $\hat{u}_i$ that are rotating, and thus varying with time (Figure 2.5).

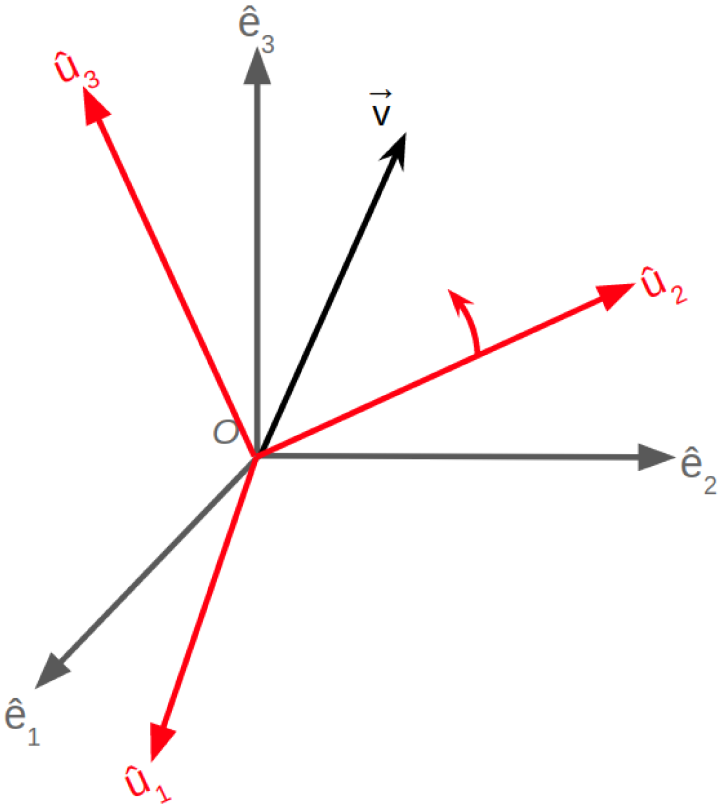

Figure 2.5: A stationary set $(\hat{e}_i)$, and a rotating set $(\hat{u}_i)$ of basis vectors in $\mathbb{R}^3$

The vector $\mathbf{v}$ can also be written as a linear combination of the rotating basis vectors:

$$\mathbf{v} = v_B^1 \hat{u}_1 + v_B^2 \hat{u}_2 + v_B^3 \hat{u}_3 = v_B^i \hat{u}_i \tag{2.21}$$

Where the subscript B represents the body frame, also known as the moving frame. If we project the vector $\mathbf{v}^i$ onto any of the basis vectors, we obtain the component of the vector in that direction:

$$\mathbf{v}^i \cdot \hat{e}_i = v^i, \tag{2.22}$$

and

$$\mathbf{v}^i \cdot \hat{u}_i = v_B^i. \tag{2.23}$$

We can also write out the derivative of vector $\mathbf{v}$ in terms of the individual bases:

$$\left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}\right)_{inertial} = \frac{\mathrm{d}v^i}{\mathrm{d}t}\hat{e}_i = \frac{\mathrm{d}v^1}{\mathrm{d}t}\hat{e}_1 + \frac{\mathrm{d}v^2}{\mathrm{d}t}\hat{e}_2 + \frac{\mathrm{d}v^3}{\mathrm{d}t}\hat{e}_3. \tag{2.24}$$

Similarly in terms of the rotating basis vectors:

$$\left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}\right)_{inertial} = \frac{\mathrm{d}v_B^i}{\mathrm{d}t}\hat{u}_i + v_B^i\frac{\mathrm{d}\hat{u}_i}{\mathrm{d}t}. \tag{2.25}$$

Where Equation 2.24 only has one term due to $\hat{e}_i$ being stationary with time, while the time derivative returns two terms in Equation 2.25 since $\hat{u}_i = \hat{u}_i(t)$. Notice in Equation 2.25, the first term is simply the rate of change of $\mathbf{v}$ in the body frame:

$$\left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}\right)_{body} = \frac{\mathrm{d}v_B^i}{\mathrm{d}t}\hat{u}_i. \tag{2.26}$$

Substituting this back into Equation 2.25 we obtain:

$$\boxed{\left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}\right)_{inertial} = \left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}\right)_{body} + v_B^i\frac{\mathrm{d}\hat{u}_i}{\mathrm{d}t}}, \tag{2.27}$$

where Equation 2.27 holds for any vector $\mathbf{v}$. If we consider $\hat{u}_i(t)$ to be a rotating coordinate system, we can break that rotation down into infinitesimal rotations. Time t and $t+dt$ have corresponding basis vectors $\hat{u}_i(t)$ and $\hat{u}_i(t+dt) = \hat{u}_i + d\hat{u}_i$ (Figure 2.6).

Keeping in mind the basis vectors are all orthonormal to one another, thus the identity:

$$\hat{u}_i \cdot \hat{u}_j = \delta_{ij}, \tag{2.28}$$

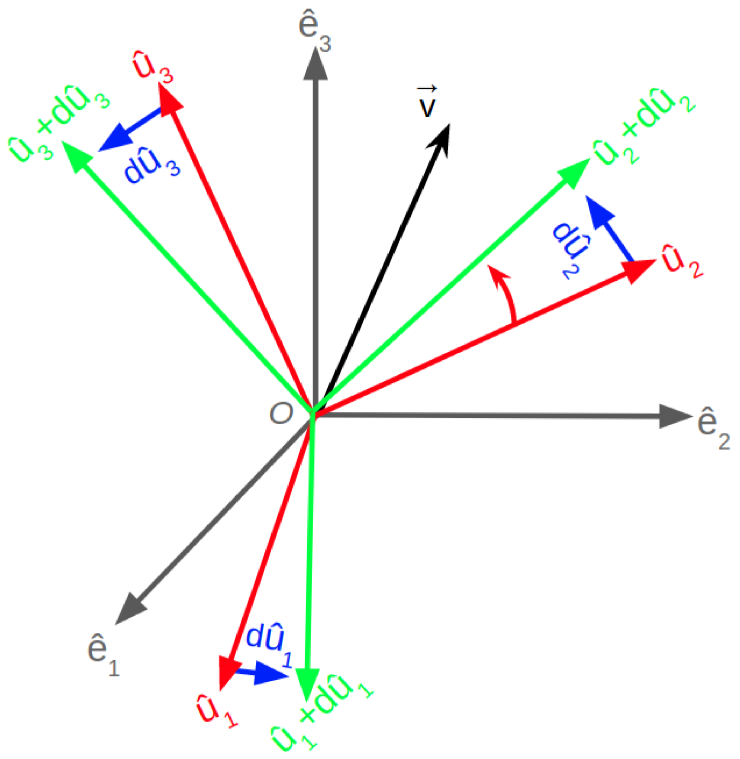

Figure 2.6: Demonstrates the infinitesimal rotation undergone by basis vectors $\hat{u}_i$.

where

$$\delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} . \tag{2.29}$$

Since $d\hat{u}_i$ lies perpendicular to the vector $\hat{u}_i$ as it represents the infinitesimal change in that vector and is thus pointed in the direction of rotation, we have

$$\hat{u}_i \cdot d\hat{u}_i = 0. \tag{2.30}$$

As the rotating basis vectors $\hat{u}_i$ form a complete set, we can write out the infinitesimal change vectors $d\hat{u}_i$ as linear combinations of $\hat{u}_i$:

$$d\hat{u}_i = d\Omega_i^j \hat{u}_j \tag{2.31}$$

where $d\Omega_i^j$ are infinitesimal coefficients for the linear combination. From Equation 2.30 we can see that all $d\Omega_i^i = 0$ for $i = 1, 2, 3$, and we can write in general,

$$d\hat{u}_i \cdot \hat{u}_j = d\Omega_i^j \tag{2.32}$$

as the coefficient is just the length of vector $d\hat{u}_i$ projected onto unit vector $\hat{u}_j$. In reality $d\Omega$ is a matrix of coefficients:

$$d\Omega = \begin{pmatrix} 0 & d\Omega_1^2 & d\Omega_1^3 \\ d\Omega_2^1 & 0 & d\Omega_2^3 \\ d\Omega_3^1 & d\Omega_3^2 & 0 \end{pmatrix} \tag{2.33}$$

where each element obeys Equation 2.32. So for example, $\Omega_1^2 = d\hat{u}_1 \cdot \hat{u}_2$ and $\Omega_2^1 = d\hat{u}_2 \cdot \hat{u}_1$. If we take the differential of Equation 2.28 we get:

$$d\hat{u}_i \cdot \hat{u}_j + \hat{u}_i \cdot d\hat{u}_j = 0, \tag{2.34}$$

which implies

$$d\Omega_{ij} = -d\Omega_{ji}. \tag{2.35}$$

This suggests that this rotational transformation is defined only by 3 independent coefficients $d\Omega_{12}$, $d\Omega_{23}$, and $d\Omega_{31}$ which from here on out we define as:

$$\begin{aligned} d\Omega_{12} &\equiv d\Omega_3 \\ d\Omega_{23} &\equiv d\Omega_1 \\ d\Omega_{31} &\equiv d\Omega_2 \end{aligned} \tag{2.36}$$

Thus we can write as a vector in terms of rotating basis $\hat{u}_i$:

$$d\mathbf{\Omega} = d\Omega_1 \hat{u}_1 + d\Omega_2 \hat{u}_2 + d\Omega_3 \hat{u}_3 \tag{2.37}$$

Following Equation 2.31 we can write out the infinitesimal basis vectors in terms of the infinitesimal rotation coefficients:

$$\begin{aligned} d\hat{u}_1 &= \cancelto{0}{d\Omega_1^1}\hat{u}_1 + d\Omega_1^2 \hat{u}_2 + d\Omega_1^3 \hat{u}_3 \\ d\hat{u}_2 &= d\Omega_2^1 \hat{u}_1 + \cancelto{0}{d\Omega_2^2}\hat{u}_2 + d\Omega_2^3 \hat{u}_3 \\ d\hat{u}_3 &= d\Omega_3^1 \hat{u}_1 + d\Omega_3^2 \hat{u}_2 + \cancelto{0}{d\Omega_3^3}\hat{u}_3 \end{aligned} \tag{2.38}$$

which written in terms of the coefficients $d\Omega_1$, $d\Omega_2$, and $d\Omega_3$ as:

$$\begin{aligned} d\hat{u}_1 &= d\Omega_3 \hat{u}_2 - d\Omega_2 \hat{u}_3 \\ d\hat{u}_2 &= -d\Omega_3 \hat{u}_1 + d\Omega_1 \hat{u}_3 \\ d\hat{u}_3 &= d\Omega_2 \hat{u}_1 - d\Omega_1 \hat{u}_2 \end{aligned} \tag{2.39}$$

which can be written more compactly as:

$$d\hat{u}_i = d\mathbf{\Omega} \times \hat{u}_i \tag{2.40}$$

as for example

$$d\hat{u}_1 = d\mathbf{\Omega} \times \hat{u}_i = \begin{vmatrix} \hat{u}_1 & \hat{u}_2 & \hat{u}_3 \\ d\Omega_1 & d\Omega_2 & d\Omega_3 \\ 1 & 0 & 0 \end{vmatrix} = 0\hat{u}_1 - (-d\Omega_3)\hat{u}_2 + (-d\Omega_2)\hat{u}_3 = d\Omega_3 \hat{u}_2 - d\Omega_2 \hat{u}_3 \tag{2.41}$$

Keep in mind $d\mathbf{\Omega}$ represents the infinitesimal rotation of the non-inertial reference frame $\hat{u}_i$, but as you can also write $d\mathbf{\Omega} = \sum_i d\Omega_i \hat{u}_i$ and thus it can be thought of as a set of individual infinitesimal rotations of magnitude $d\Omega_i$ around each of the axes $\hat{u}_i$.

Now consider an infinitesimal rotation about the $\hat{u}_1$ axis (Figure 2.7).

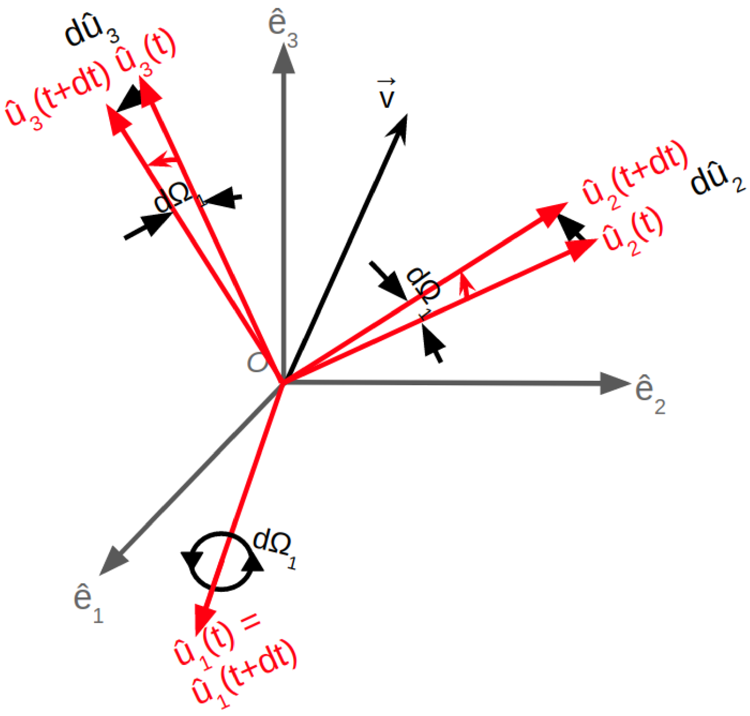

Figure 2.7: Infinitesimal rotation $d\mathbf{\Omega}_1 = d\Omega_1\hat{u}_1$ about the 1 axis

To define the infinitesimal change vectors $d\hat{u}_i$ in terms of the infinitesimal rotation $d\mathbf{\Omega}_1$ we can use of Equation 2.40 which is true for a general 3D rotation, and therefore is also true for the limiting case of a rotation along a single axis:

$$\begin{aligned} d\hat{u}_1 &= d\mathbf{\Omega}_1 \times \hat{u}_1 = 0 \\ d\hat{u}_2 &= d\mathbf{\Omega}_1 \times \hat{u}_2 = d\Omega_1\hat{u}_3 \\ d\hat{u}_3 &= d\mathbf{\Omega}_1 \times \hat{u}_3 = -d\Omega_1\hat{u}_2 \end{aligned} \tag{2.42}$$

Now with these different concepts covered we can return to Equation 2.27 to write out the rotation equation in terms of usable quantities such as rate of rotation. Over an infinitesimal change in time dt, the change in basis vector $\hat{u}_i$

is given by $d\hat{u}_i$:

$$d\hat{u}_i = \frac{\mathrm{d}\hat{u}_i}{\mathrm{d}t}dt \tag{2.43}$$

using Equation 2.40:

$$d\hat{u}_i = \frac{d\hat{u}_i}{dt}dt = \frac{\mathrm{d}\boldsymbol{\Omega}}{\mathrm{d}t} \times \hat{u}_i dt = \boldsymbol{\omega} \times \hat{u}_i dt \tag{2.44}$$

where $\boldsymbol{\omega}$ is the rate of rotation written as a vector with direction pointing in the direction of the axis of rotation, or in other words, the instantaneous angular velocity of the rotating frame, as seen in the inertial frame.

$$\boldsymbol{\omega} \equiv \frac{\mathrm{d}\boldsymbol{\Omega}}{\mathrm{d}t} \tag{2.45}$$

Substituting this identity into Equation 2.27 we get:

$$\begin{aligned} \left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}\right)_{inertial} &= \left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}\right)_{body} + v_B^i(\boldsymbol{\omega} \times \hat{u}_i) \\ &= \left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}\right)_{body} + \boldsymbol{\omega} \times \mathbf{v} \end{aligned} \tag{2.46}$$

Keeping in mind that $\boldsymbol{\omega}$ is a vector, like any other vector, $\boldsymbol{\omega}$ can be written in terms of any frame that consists of unit vectors making up a complete set. Thus substituting $\boldsymbol{\omega}$ into Equation 2.46 as our arbitrary vector $\mathbf{v}$ we find:

$$\begin{aligned} \left(\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t}\right)_{inertial} &= \left(\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t}\right)_{body} + \cancelto{0}{\boldsymbol{\omega} \times \boldsymbol{\omega}} \\ \left(\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t}\right)_{inertial} &= \left(\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t}\right)_{body} \end{aligned} \tag{2.47}$$

So whether in the inertial or body frame, the observers agree on the rate of change of the instantaneous angular velocity.

Now we consider the effect of the rotations on accelerations, and thus the addition of any fictitious forces. If we choose our arbitrary vector **v** to be a position vector **r** we have:

$$\left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_{inertial} = \left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_{body} + \boldsymbol{\omega} \times \mathbf{r}. \tag{2.48}$$

Since Equation 2.48 holds true for any vector, time derivatives of any vector obeys the operator:

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}\right)_{inertial} = \left(\frac{\mathrm{d}}{\mathrm{d}t}\right)_{body} + \boldsymbol{\omega} \times . \tag{2.49}$$

Therefore to get the acceleration we simply apply this operator to **r** twice:

$$\begin{aligned}\left(\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2}\right)_{inertial} &= \left[\left(\frac{\mathrm{d}}{\mathrm{d}t}\right)_{body} + \boldsymbol{\omega} \times\right]\left[\left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_{body} + \boldsymbol{\omega} \times \mathbf{r}\right] \\ &= \left(\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2}\right)_{body} + \left(\frac{\mathrm{d}}{\mathrm{d}t}\right)_{body}(\boldsymbol{\omega} \times \mathbf{r}) + \boldsymbol{\omega} \times \left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_{body} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \\ &= \left(\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2}\right)_{body} + \left(\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t}\right)_{body} \times \mathbf{r} + \boldsymbol{\omega} \times \left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_{body} + \boldsymbol{\omega} \times \left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_{body} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \\ &= \left(\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2}\right)_{body} + 2\boldsymbol{\omega} \times \left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_{body} + \frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} \times \mathbf{r} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})\end{aligned} \tag{2.50}$$

Resulting in the fictitious acceleration formula:

$$\boxed{\left(\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2}\right)_{body} = \left(\frac{\mathrm{d}^2\mathbf{r}}{\mathrm{d}t^2}\right)_{inertial} \underbrace{-2\boldsymbol{\omega} \times \left(\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t}\right)_{body}}_{\mathbf{a}_{Coriolis}} \underbrace{-\frac{\mathrm{d}\boldsymbol{\omega}}{\mathrm{d}t} \times \mathbf{r}}_{\mathbf{a}_{Euler}} \underbrace{-\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})}_{\mathbf{a}_{Centrifugal}}} \tag{2.51}$$

For operational simplicity, we choose a coordinate system that rotates at constant angular velocity, while keeping track of variations of the rotation rate of each fluid element relative to this uniformly rotating frame via the angular momentum scalar. The Euler term above is therefore zero.

With our simulation setup in cylindrical coordinates following the geometry shown in Figure 2.1, we can derive the fictitious accelerations in terms of the simulation parameters. If material in a cell is located at position:

$$\mathbf{r} = \rho\hat{\rho} + z\hat{z}, \tag{2.52}$$

and has velocity in the co-rotating frame

$$\mathbf{v} = v_\rho\hat{\rho} + v_z\hat{z} \tag{2.53}$$

the centrifugal acceleration is given by:

$$a_{centrifugal} = -\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) \tag{2.54}$$

where $\boldsymbol{\omega} = \omega\hat{\rho}$. Using Equation 2.52,

$$\boldsymbol{\omega} \times \mathbf{r} = -\omega z\hat{\varphi} = \omega|z|\hat{\xi}. \tag{2.55}$$

Where $\hat{\xi}$ is the unit vector representing rotation about the ρ-axis (orbital direction) as seen in Figure 2.8. In terms of the standard cylindrical coordinates:

$$\hat{\xi} = \begin{cases} -\hat{\varphi} & z > 0 \\ \hat{\varphi} & z < 0 \end{cases} = -\text{sgn(z)}\hat{\varphi}. \tag{2.56}$$

It is also convenient to define $\hat{\zeta}$ as a unit vector pointing away from the rotation axis ρ in the simulation domain (Figure 2.8) such that it is defined as:

$$\hat{\zeta} = \begin{cases} \hat{z} & z > 0 \\ -\hat{z} & z < 0 \end{cases} = \text{sgn(z)}\hat{\text{z}}. \tag{2.57}$$

Similarly we can write $\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})$ as:

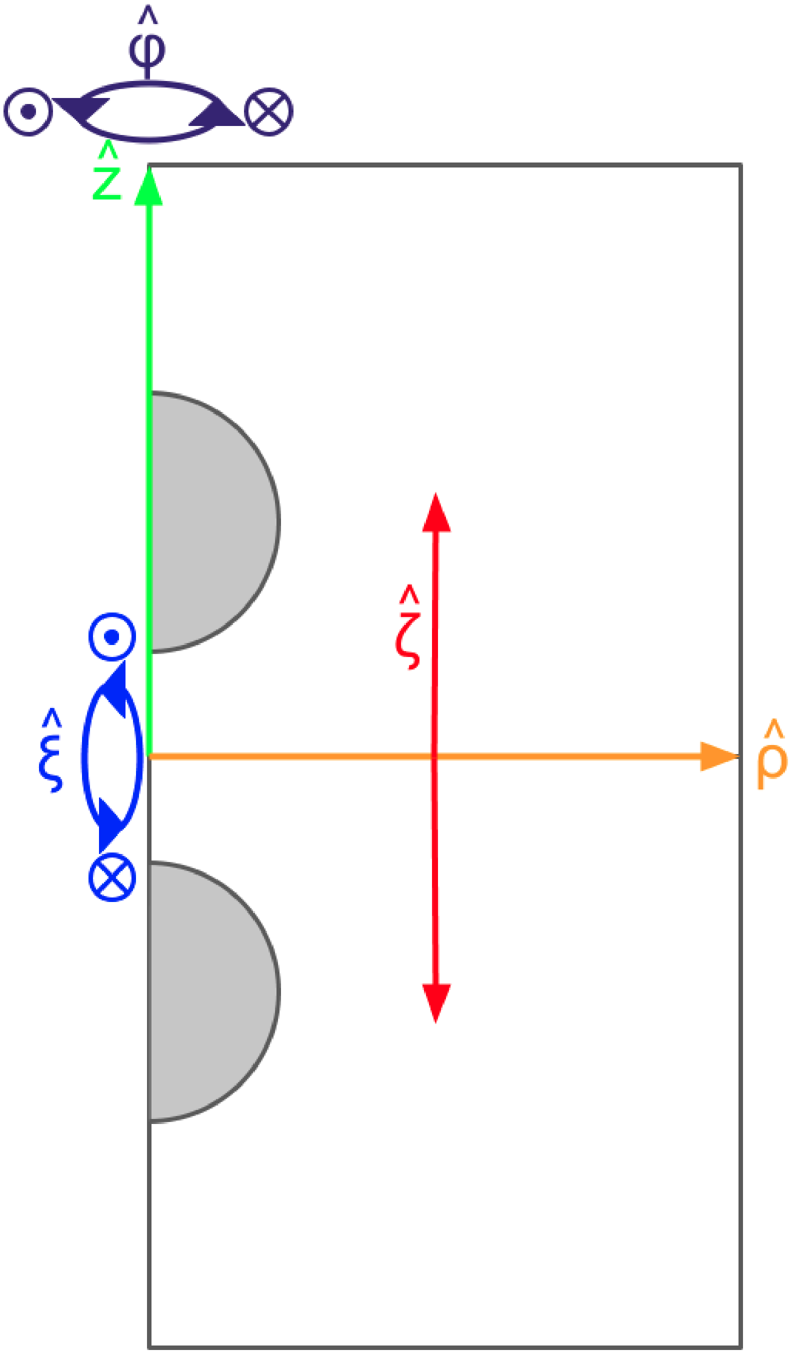

Figure 2.8: Unit vector directions in the simulation. It is convenient to define $\hat{\xi}$ as the orbital direction for rotation about the ρ-axis, and $\hat{\zeta}$ as the direction away from the rotation axis ρ.

$$\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) = -\omega^2 z\hat{z} = -\omega^2 |z| \hat{\zeta}. \tag{2.58}$$

Thus the Centrifugal force is given by:

$$\mathbf{F}_{centrifugal} = m\omega^2 z\hat{z} = m\omega^2 |z| \hat{\zeta}. \tag{2.59}$$

and thus

$$\boxed{\mathbf{a}_{centrifugal} = \omega^2 z\hat{z} = \omega^2 |z| \hat{\zeta}}. \tag{2.60}$$

The Centrifugal acceleration is implemented into the existing `FLASH` routine `hy_ppm_force` which adds the fictitious acceleration as a source term within the `Hydro` module during each time-step evolution. In the co-rotating frame, which is rotating at angular velocity ω, the centrifugal fictitious force per unit mass (acceleration) is added in the form:

$$\mathbf{f}_{centrifugal} = \omega^2 |z| \cdot \mathrm{sgn(z)} \cdot \hat{\mathrm{z}}, \tag{2.61}$$

where sgn(z) $\hat{\mathrm{z}}$ is pointing away from the ρ-axis.

The Coriolis force is given by:

$$\mathbf{F}_{coriolis} = -2m\boldsymbol{\omega} \times \left(\frac{d\mathbf{r}}{dt}\right)_{body}, \tag{2.62}$$

where,

$$\left(\frac{d\mathbf{r}}{dt}\right)_{body} = \mathbf{v}_{body} = v_\rho \hat{\rho} + v_z \hat{z} + v_{\xi,frame} \hat{\xi}, \tag{2.63}$$

is the fluid velocity in the co-rotating frame, and

$$v_{\xi,frame} = v_{\xi,lab} - \omega |z|, \tag{2.64}$$

is the rotational velocity in the co-rotating frame, with $v_{\xi,lab}$ the rotational velocity in the laboratory frame. For convenience, hereafter we refer to $v_{\xi,frame}$ as v_ξ. Therefore we can write,

$$\boldsymbol{\omega} \times \mathbf{v}_{body} = -\omega v_z \hat{\varphi} + \omega v_\varphi \hat{z} = \omega \ \mathrm{sgn(z)} \ (\mathrm{v_z}\hat{\xi} - \mathrm{v_\xi \hat{z}}) \tag{2.65}$$

where $v_\varphi \hat{\varphi} = v_\xi \hat{\xi}$, and $v_\varphi = -\mathrm{sgn(z)} \ \mathrm{v_\xi}$ (The sign factor of the velocity component and unit vector cancel out). So the Coriolis force is given by:

$$\mathbf{F}_{coriolis} = 2m\omega v_z \hat{\varphi} - 2m\omega v_\varphi \hat{z} = 2m\omega(v_z \hat{\varphi} - v_\varphi \hat{z}) = 2m\omega \ \mathrm{sgn(z)} \ (\mathrm{v_\xi \hat{z}} - \mathrm{v_z}\hat{\xi}) \tag{2.66}$$

and the Coriolis acceleration is given by:

$$\boxed{\mathbf{a}_{coriolis} = 2\omega(v_z \hat{\varphi} - v_\varphi \hat{z}) = 2\omega \ \mathrm{sgn(z)} \ (\mathrm{v_\xi \hat{z}} - \mathrm{v_z}\hat{\xi})} \tag{2.67}$$

The Coriolis force therefore acts both in the z-direction as well as in the ξ-direction. By acting in the orbital direction, a torque is introduced which modifies the angular momentum. This is accounted for by implementing a source term in the `Cooling` module of `FLASH` which modifies the angular momentum of material within individual cells in between calls to the hydrodynamic solver ("operator-split"). The z Coriolis force is added as a fictitious acceleration in the function `hy_ppm_force` which acts in addition to the centrifugal force, of the form:

$$\mathbf{f}_{coriolis,z} = 2\omega v_\xi \mathrm{sgn(z)\hat{z}}. \tag{2.68}$$

Where the sgn(z) factor accounts for how the orbital direction is represented in terms of φ (See Figure 2.1). For the $\hat{\xi}$ Coriolis force,

$$\mathbf{f}_{coriolis,\xi} = -2\omega v_z \mathrm{sgn(z)}\hat{\xi} \tag{2.69}$$

we convert this acceleration into a dimensionless source term to the angular momentum below. To do this we will evolve ℓ_{orb} as

$$\ell_{orb}^{n+1} = \ell_{orb}^{n}(1 + S_{coriolis}) = \ell_{orb}^{n}\left(1 + \frac{\Delta\ell_{orb}}{\ell_{orb}^{n}}\right) \tag{2.70}$$

where n labels the time step, and $S_{coriolis}$ is the source term to the angular momentum corresponding to the Coriolis force. Thus we need to calculate the torque per unit mass

$$\begin{aligned} \dot{\ell}_{orb} &= \frac{\tau}{m} = \mathbf{r}_{\perp} \times \mathbf{f}_{coriolis,\xi} \\ \frac{\Delta\ell_{orb}}{\Delta t} &= r_{\perp}\frac{\Delta v}{\Delta t} \\ \Delta\ell_{orb} &= \Delta v|z|. \end{aligned} \tag{2.71}$$

If we discretize the Coriolis acceleration as:

$$a_{coriolis} = \frac{\Delta v}{\Delta t} \tag{2.72}$$

thus

$$\Delta v = a_{coriolis}\Delta t = -2\omega v_z \mathrm{sgn(z)\Delta t}. \tag{2.73}$$

Substituting this into Equation 2.71, the source term can be written as:

$$S_{coriolis} = \frac{\Delta\ell_{orb}}{\ell_{orb}} = \frac{\Delta v|z|}{\ell_{orb}} = \frac{-2\omega v_z \mathrm{sgn(z)\Delta t|z|}}{\ell_{orb}}. \tag{2.74}$$

Writing ω in terms of ℓ_{orb} we have:

$$\omega = \frac{\ell_{orb}}{|z|^2} \tag{2.75}$$

So,

$$\boxed{S_{coriolis} = \frac{-2v_z \mathrm{sgn(z)} \Delta \mathrm{t}}{|z|}}. \tag{2.76}$$

As will be discussed in Section 2.3 the effect of gravitational wave emission is also added as a source term S_{GW} to the angular momentum, thus we have a total evolution equation in the `cooling` module of the form:

$$\ell_{orb}^{n+1} = \ell_{orb}^{n}(1 + [S_{GW} + S_{coriolis}]). \tag{2.77}$$

2.3 Gravitational Wave Emission

Due to the time varying quadrupole moment of the orbiting binary system, the binary neutron star emits gravitational waves. The emission of the gravitational waves removes angular momentum from the system, and is responsible for the inspiral. Similarly to how the Coriolis force is implemented, the emission of gravitational waves and its effect on the orbital separation in the simulation can be included as a source term to the angular momentum within the `Cooling` module in `FLASH`.

Neglecting the effects of higher order terms, we start by assuming the orbital velocity is approximately Keplerian. In the centre of mass frame (Figure 2.2, right pane) the orbital speed of the reduced mass for a circular orbit is:

$$v_{orb,i,rm} \approx v_{K,rm} = \sqrt{\frac{GM}{d}}. \tag{2.78}$$

The associated specific angular momentum of the system is:

$$\ell_{orb} = \sqrt{GMd}. \tag{2.79}$$

To calculate the source term used to model losses due to the emission of gravitational waves, we consider how the angular momentum changes with time $\dot{\ell}_{orb}$:

$$\begin{aligned}
\dot{\ell_{orb}} &= \frac{1}{2}\sqrt{\frac{GM}{d}}\dot{d}_{GW} \\
\dot{\ell_{orb}} &= \frac{1}{2}\frac{\ell_{orb}}{d}\dot{d}_{GW} \\
\dot{\ell_{orb}} &= \frac{1}{2}v_{K,rm}\dot{d}_{GW},
\end{aligned} \tag{2.80}$$

where $\dot{d}_{GW}$ is given by Equation 2.4. Therefore the dimensionless source term can be derived in a similar way to that for the Coriolis force:

$$\begin{aligned}
S_{GW} &= \frac{\Delta\ell_{orb}}{\ell_{orb}} \\
S_{GW} &= \frac{\dot{\ell}_{orb}}{\ell_{orb}}\Delta t,
\end{aligned} \tag{2.81}$$

substituting in Equation 2.80 we get:

$$\boxed{S_{GW} = \frac{1}{2}\frac{\dot{d}_{GW}}{d}\Delta t = S_{GW}} \tag{2.82}$$

Strictly speaking, as derived in Section 2.2 the initial orbital velocity is not exactly Keplerian as the binary is in inspiral:

$$\begin{aligned}
v_{orb,rm} &\neq v_{K,rm} \\
v_{orb,rm} &= \sqrt{\frac{GM}{d} - \dot{d}^2} \\
v_{orb,rm} &= \sqrt{\frac{GM}{d}}\left(1 - \frac{d\dot{d}^2}{GM}\right)^{1/2} \\
v_{orb,rm} &= \sqrt{\frac{GM}{d}}\left(1 - \frac{d}{GM}\left(-\frac{64G^3}{5c^5}\frac{\mu M^2}{d^3}\right)^2\right)^{1/2} \\
v_{orb,rm} &= \sqrt{\frac{GM}{d}}\left(1 - \frac{4096}{25}\frac{G^5}{c^{10}}\frac{\mu^2 M^3}{d^5}\right)^{1/2} \\
v_{orb,rm} &= v_K\left(1 - \frac{4096}{25}\frac{G^5}{c^{10}}\frac{\mu^2 M^3}{d^5}\right)^{1/2} \\
v_{orb,rm} &= v_K\left(1 - \frac{256}{25}\frac{G^5}{c^{10}}\frac{M^5}{d^5}\right)^{1/2} \\
v_{orb,rm} &= v_K\left(1 - \frac{256}{25}\left(\frac{v_K}{c}\right)^{10}\right)^{1/2}
\end{aligned} \tag{2.83}$$

Setting $M = 2 \times 1.4M_{\odot}$, $d = 2R_{NS}$, and $R_{NS} = 12.6km$, we obtain

$$v_{orb,rm} \approx 0.9994\ v_K \tag{2.84}$$

at contact. Given the small magnitude of this correction, we neglect it and employ the source term as shown in Equation 2.82.

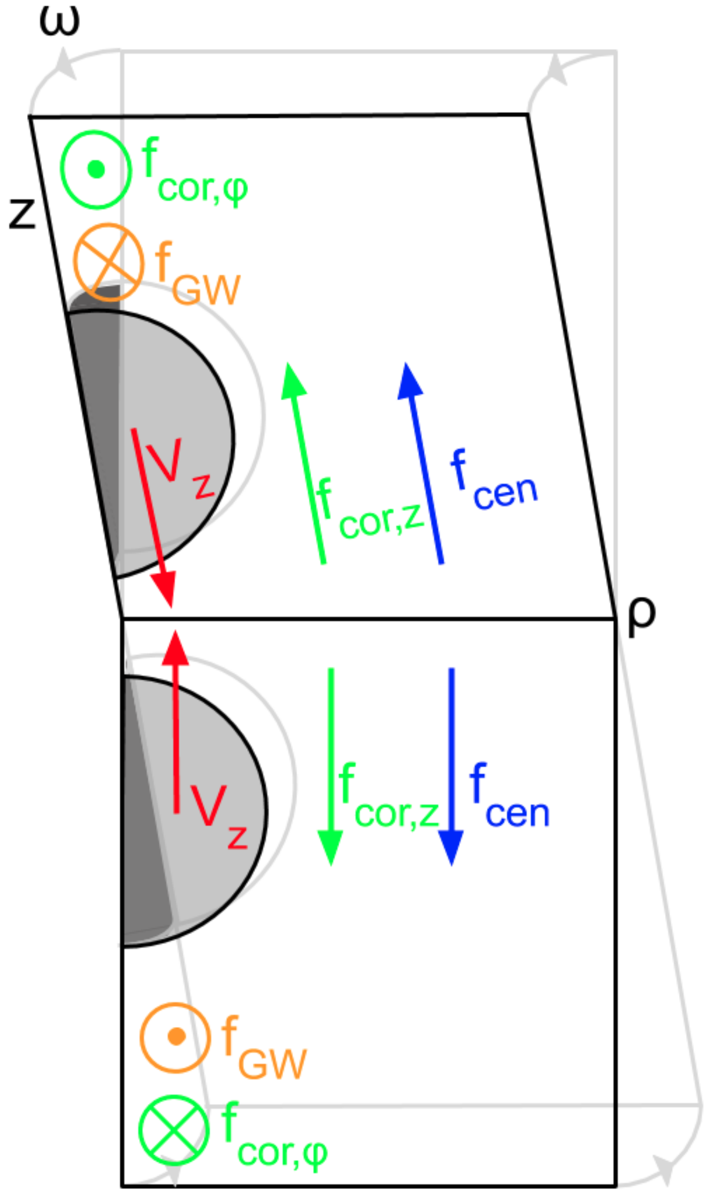

Figure 2.9: Schematic of inertial acceleration directions, centrifugal acceleration (Equation 2.61) in blue, Coriolis acceleration (Equations 2.68 and 2.69) in green due to the initial z-velocity (red), as well as the direction of the effective acceleration due to gravitational wave emission (Equation 2.80) in orange. The z-component of the Coriolis acceleration is non-zero only if there is azimuthal motion relative to the co-rotating frame: excess rotation generates acceleration away from the orbital axis in excess of the centrifugal contribution.

Chapter 3

Computational Method

This chapter will discuss the computational implementation in `FLASH`, the external programs written to produce the initial conditions, as well as those written for post-processing of the data produced in the simulation. In particular, I derive the equations I used to produce the initial neutron star profiles, which are used to initialize the simulation, modify the `FLASH` Equation of State module to run with a Piece-wise Polytropic Equation of State and approximate thermal effects (Bauswein et al., 2010), and track unbound material as it is ejected from the system. These derivations include writing the equations in the correct geometric form as they are implemented into the code.

3.1 Equation of State

The equation of state is a scalar thermodynamic function that relates pressure to internal energy and density, and is needed to close the system of two dimensional Newtonian hydrodynamics equations being solved by `FLASH` (Equations 3.41 - 3.44).

While `FLASH` provides several Equations of State (EOS) by default such as the Helmholtz and Gamma-law equation of state, in order to allow the parameter space relevant to neutron stars to be easily explored, I implemented a piece-wise polytropic equation of state as described in Read et al. (2009), with approximate thermal effects as described in Bauswein et al. (2010).

3.1.1 Piece-wise Polytropic EOS

The cold (zero-temperature) component of the equation of state is described in terms of a piece-wise polytropic (PWP) equation of state. This formulation from Read et al. (2009), allows most microphysical equations of state to be reproduced to good accuracy through the use of a four-piece piece-wise polytropic function of the form:

$$P_{cold} = K_{\mathrm{i}} \rho^{\Gamma_{cold,\mathrm{i}}}, \ \mathrm{i} = \{0, 1, 2, 3\}, \tag{3.1}$$

where P_{cold} is the zero-temperature component of the pressure, K_{i} are the scale factors, $\Gamma_{\mathrm{cold,i}}$ are the polytropic indices for the cold component of the pressure, and i is the index denoting which segment of the piece-wise polytropic EOS we are referring to, for densities higher than ρ_1 (Equation 3.7), and a `SLy` EOS (Douchin and Haensel, 2001) for lower densities (crust). This formulation is entirely determined by four parameters (P_1,Γ_1,Γ_2,Γ_3). P_1 is the dividing pressure between segments 1 and 2, and Γ_i are the polytropic indexes of segments 1, 2, and 3.

The dividing densities are set as:

$$\rho_{div} = \begin{cases} \rho_1 \\ 1.85\rho_{nuclear} \\ 3.70\rho_{nuclear}, \end{cases} \tag{3.2}$$

where $\rho_{nuclear} = 2.7 \times 10^{14}\ g/cm^3$, and ρ_1 is a calculated quantity. The fixed `SLy` crust is defined as a polytrope with scale factor $K_0 = 3.594 \times 10^{13}\ \frac{dyn}{cm^2} \left(\frac{g}{cm^3}\right)^{-\Gamma_0}$, and polytropic index $\Gamma_0 = 1.357$. Given the fit parameters (P_1,Γ_1,Γ_2,Γ_3), and the fixed parameters (ρ_2, ρ_3,K_0,Γ_0,a_0) where ρ_i is the ith dividing density, we

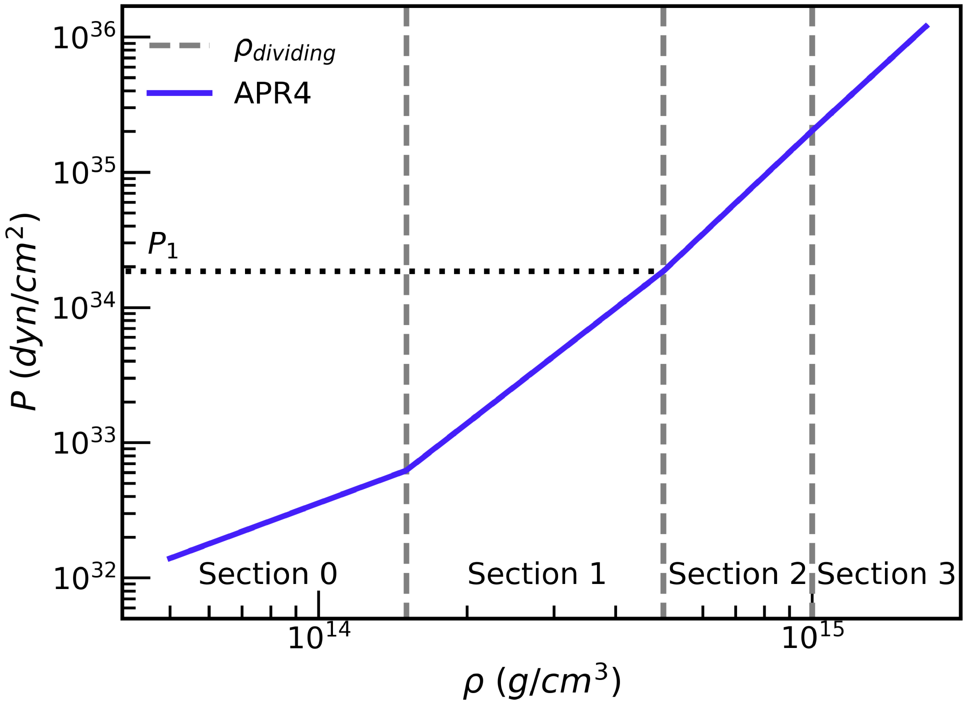

Figure 3.1: Schematic plot (adapted from Read et al. 2009 Figure 2) of pressure vs. density in the fiducial PWP Equation of State (APR4, Akmal et al. 1998) using the parameterization of Read et al. (2009). Grey dashed lines identify the dividing densities between segments. P_1 identifies the dividing pressure between section 1 and 2, and is one of the identifying parameters of the EOS. Section 0 represents the fixed SLy crust.

can solve for the remaining unknowns as follows:

$$K_1 = \frac{P_1}{\rho_2^{\Gamma_1}}, \tag{3.3}$$

$$K_2 = \frac{P_1}{\rho_2^{\Gamma_2}}, \tag{3.4}$$

$$K_3 = K_2 \rho_3^{(\Gamma_2 - \Gamma_3)}. \tag{3.5}$$

For this polytropic representation, with indicies $\Gamma_{\rm i}$ being unitless, the coefficients $K_{\rm i}$ have units:

$$K_{\rm i} \rightarrow \left[\frac{dyn}{cm^2}\left(\frac{g}{cm^3}\right)^{-\Gamma_i}\right], \tag{3.6}$$

to ensure the correct pressure units. Thus each $K_{\rm i}$ has different units. Continuing with the derivation:

$$\rho_1 = \left(\frac{K_0}{K_1}\right)^{1/(\Gamma_1-\Gamma_0)}, \tag{3.7}$$

$$\epsilon(\rho_1) = a_0 + \frac{K_0\rho_1^{\Gamma_0-1}}{\Gamma_0-1}, \tag{3.8}$$

$$a_1 = \epsilon(\rho_1) - \frac{K_1\rho_1^{\Gamma_1-1}}{\Gamma_1-1}, \tag{3.9}$$

$$\epsilon(\rho_2) = a_1 + \frac{K_1\rho_2^{\Gamma_1-1}}{\Gamma_1-1}, \tag{3.10}$$

$$a_2 = \epsilon(\rho_2) - \frac{K_2\rho_2^{\Gamma_2-1}}{\Gamma_2-1}, \tag{3.11}$$

$$\epsilon(\rho_3) = a_2 + \frac{K_2\rho_3^{\Gamma_2-1}}{\Gamma_2-1}, \tag{3.12}$$

and finally,

$$a_3 = \epsilon(\rho_3) - \frac{K_3\rho_3^{\Gamma_3-1}}{\Gamma_3-1}. \tag{3.13}$$

where ϵ is the specific internal energy and $a_0 = 0$ as derived in Section 3.1.2. Thus, through the use of Equations 3.3 - 3.13 we can use the equation of state parameters (P_1,Γ_1,Γ_2,Γ_3) provided in Read et al. (2009) Table III for a variety of neutron star equations of state, to solve for the remaining required parameters.

3.1.2 Thermal Effects

The high density of material within neutron stars, is such that the temperature is low in comparison to the Fermi energy of that material (e.g., Bauswein et al. 2010), and thus in general a zero temperature approximation is valid for neutron stars (the pressure in a 'cold' neutron star is dominated by non-thermal nuclear forces). However, in the case of a merging binary neutron star the thermal energy generated at the time of merger via shocks can have a significant effect. Accurately including the effects of temperature in an EOS can be computationally expensive, and thus an approximate treatment allows the dominant thermal effects to be captured while only adding marginal computational expense. Here I use the approximate thermal treatment presented in Bauswein et al. (2010), and implement it into `FLASH`. Using this approximate thermal treatment, the total gas pressure can be split into a cold and thermal component:

$$P = P_{cold} + P_{th}. \tag{3.14}$$

Where P is the total gas pressure, P_{th} is the thermal contribution to pressure, and P_{cold} is the cold pressure provided by the piece-wise polytropic equation of state (that parameterizes the physical EOS) given by Equation 3.1. The total pressure can then be used to convert to total internal energy using the following:

$$P = (\gamma_e - 1)\rho\epsilon, \tag{3.15}$$

which can be used also to relate the individual internal energy components to their corresponding pressure components:

$$P_{cold} = (\Gamma_{cold} - 1)\rho\epsilon_{cold}, \tag{3.16}$$

for P_{cold} or as:

$$P_{th} = (\Gamma_{th} - 1)\rho\epsilon_{th}, \tag{3.17}$$

for P_{th}, where Γ_{th} is the thermal polytropic index. We adopt $\Gamma_{\rm th} = 5/3 \simeq 1.67$ as an intermediate value in the interval $[1.5, 2]$, which was found by Bauswein et al. (2010) to bracket the behavior of micro-physical, finite-temperature EOS's. In general, the EOS is used to obtain a third variable out of two, for example to obtain the internal energy given density and pressure as input during initialization. In what follows, I will derive the variables using the density-pressure mode.

In the density-pressure mode, where the equation of state is passed the evolved pressure and density, we begin by calculating the cold component of the pressure P_{cold} using Equation 3.1. Since the dividing densities are known, they are used to sort input densities into the relevant segment of the EOS. Next, to calculate the cold component of the internal energy, and ensure we recover the integration constant responsible for ensuring the internal energy is continuous, we begin with the first law of thermodynamics written in the differential form (Read et al., 2009):

$$d\frac{\varepsilon}{\rho} = -\frac{P}{c^2}\, d\frac{1}{\rho}, \tag{3.18}$$

where ε is the energy density (including the rest mass) divided by c^2 (units of mass density). Integrating the left side of the differential equation and substituting in Equation 3.1,

$$\begin{aligned}\int d\frac{\varepsilon}{\rho} &= -\frac{P}{c^2}d\frac{1}{\rho}\\ \frac{\varepsilon}{\rho} + A &= -\frac{K\rho^{\Gamma}}{c^2}d\frac{1}{\rho}\\ \frac{\varepsilon}{\rho} + A &= -\frac{K}{c^2}\left(\frac{1}{\rho}\right)^{-\Gamma}d\frac{1}{\rho},\end{aligned} \tag{3.19}$$

Integrating the right hand side,

$$\begin{aligned}\frac{\varepsilon}{\rho} + A &= -\frac{K}{c^2}\int\left(\frac{1}{\rho}\right)^{-\Gamma}d\frac{1}{\rho},\\ \frac{\varepsilon}{\rho} + B &= -\frac{K}{c^2}\frac{1}{1-\Gamma}\left(\frac{1}{\rho}\right)^{1-\Gamma}\\ \frac{\varepsilon}{\rho} + B &= \frac{1}{\Gamma-1}\frac{K}{c^2}\frac{\rho^{\Gamma}}{\rho}\\ \frac{\varepsilon}{\rho} + B &= \frac{1}{\Gamma-1}\frac{P}{\rho c^2},\\ \frac{\varepsilon}{\rho} &= -B + \frac{1}{\Gamma-1}\frac{P}{\rho c^2},\end{aligned} \tag{3.20}$$

where A and B are different constants. If we redefine the constant $B = -(1 + a/c^2)$ we have

$$\begin{aligned}\frac{\varepsilon}{\rho} &= 1 + a/c^2 + \frac{1}{\Gamma-1}\frac{P}{\rho c^2}\\ \frac{\varepsilon}{\rho} &= 1 + a/c^2 + \frac{1}{\Gamma-1}\frac{K\rho^{\Gamma-1}}{c^2}\end{aligned} \tag{3.21}$$

In our case where we are using the piece-wise polytrope defined in Equation 3.1 we can simply extend this to:

$$\frac{\varepsilon}{\rho} = 1 + a_{\mathrm{i}}/c^2 + \frac{1}{\Gamma_{\mathrm{i}}-1}\frac{K_{\mathrm{i}}\rho^{\Gamma_{\mathrm{i}}-1}}{c^2}. \tag{3.22}$$

Rearranging for a_{i},

$$a_{\mathrm{i}} = \left(\frac{\varepsilon}{\rho} - 1\right)c^2 - \frac{K_{\mathrm{i}}\rho^{\Gamma_{\mathrm{i}}-1}}{\Gamma_{\mathrm{i}} - 1}. \tag{3.23}$$

The energy density divided by c^2 (ε) can be related to the internal energy (ϵ) using the relation (Read et al., 2009):

$$\epsilon = \left(\frac{\varepsilon}{\rho} - 1\right)c^2. \tag{3.24}$$

We can enforce the limit that as $\rho \to 0$, $\epsilon \to 0$, and thus

$$\lim_{\rho\to 0} \frac{\varepsilon}{\rho} = 1. \tag{3.25}$$

Applying the limit 3.25 to Equation 3.23, we find $a_i = 0$ when $\rho \to 0$ therefore $a_0 = 0$, where the subscript 0 refers to the first polytropic segment (crust) corresponding to the lowest density regime of the Equation of State. Rewriting Equations 3.22 and 3.23 in terms of ϵ, and the dividing densities ρ_{i} (Equation 3.2) we have

$$a_{\mathrm{i}} = \epsilon(\rho_{\mathrm{i}}) - \frac{K_{\mathrm{i}}\rho_{\mathrm{i}}^{\Gamma_{\mathrm{i}}-1}}{\Gamma_{\mathrm{i}} - 1},\ \mathrm{i} = 1, 2, 3, \tag{3.26}$$

and

$$\epsilon(\rho_{\mathrm{i}}) = a_{\mathrm{i}-1} + \frac{K_{\mathrm{i}-1}\rho_{\mathrm{i}}^{\Gamma_{\mathrm{i}-1}-1}}{\Gamma_{\mathrm{i}-1} - 1},\ \mathrm{i} = 1, 2, 3. \tag{3.27}$$

We can solve for the remaining a_{i} by using Equation 3.26 and 3.27 in a leap-frog technique. For example, using a_0 to calculate $\epsilon(\rho_1)$ using Equation 3.27, then using the result to calculate a_1 using Equation 3.26, and so on. Thus for the piece-wise polytropic EOS, we have the specific internal energy given by:

$$\epsilon = \frac{P}{(\Gamma_{\mathrm{i}} - 1)\rho} + a_{\mathrm{i}}. \tag{3.28}$$

Next the cold component of the internal energy can be calculated using Equation 3.28:

$$\epsilon_{cold} = \frac{P_{cold}}{(\Gamma_{cold,i} - 1)\rho} + a_{\rm i}. \tag{3.29}$$

We now have enough information to solve for the total internal energy ϵ. Using a combination of:

$$P = P_{cold} + P_{th} = K_{\rm i}\rho^{\Gamma_{\rm i}} + (\Gamma_{th} - 1)\rho\epsilon_{th} \tag{3.30}$$

and

$$\epsilon_{th} = \epsilon - \epsilon_{cold}, \tag{3.31}$$

to solve for the internal energy:

$$\epsilon_{th} = \frac{(P - P_{cold})}{(\Gamma_{th} - 1)\rho}, \tag{3.32}$$

so the total specific internal energy ϵ is given by

$$\epsilon = \frac{(P - P_{cold})}{(\Gamma_{th} - 1)\rho} + \epsilon_{cold}. \tag{3.33}$$

Next the equation of state yields two adiabatic indices required for the hydrodynamic solver: γ_c, and γ_e:

$$P = (\gamma_e - 1)\rho\epsilon, \tag{3.34}$$

$$\gamma_e = \frac{P}{\rho\epsilon} + 1, \tag{3.35}$$

as defined in Fryxell et al. (2000). γ_c is defined as:

$$\begin{aligned}
\gamma_c &= \frac{\rho}{P}\left(\frac{\partial P}{\partial \rho}\right)_S \\
\gamma_c &= \frac{\rho}{P}\left(\frac{\partial P_{cold}}{\partial \rho} + \frac{\partial P_{th}}{\partial \rho}\right) \\
\gamma_c &= \frac{\rho}{P}\left(\Gamma_{cold}\frac{P_{cold}}{\rho} + \Gamma_{th}\frac{P_{cold}}{\rho}\right) \\
\gamma_c &= \frac{P_{cold}}{P}\Gamma_{cold} + \frac{P_{th}}{P}\Gamma_{th}.
\end{aligned} \tag{3.36}$$

3.1.3 Initial Neutron Star Profiles/Implementation in FLASH

To include neutron star models built with a given equation of state into the domain we must first generate a radial profile of the star over which pressure and density are defined $\{\rho(r), P(r)\}$. Given that the gravity in our simulations is Newtonian, we built initial stars in Newtonian hydrostatic equilibrium, using the equation:

$$\frac{dP}{dr} = -\frac{Gm(r)\rho(r)}{r^2}. \tag{3.37}$$

We also assume that the star is spherically symmetric, and thus

$$\frac{dm}{dr} = 4\pi r^2 \rho(r). \tag{3.38}$$

Finally the system of equations is closed with the Equation of State as described in Sections 3.1.1 - 3.1.2. Equation 3.1 can then be rewritten as a differential equation, for the density through the use of Equation 3.37:

$$
\begin{aligned}
\frac{dP}{dr} &= \Gamma \frac{P}{\rho} \frac{d\rho}{dr} \\
-\frac{Gm(r)\rho(r)}{r^2} &= \Gamma \frac{P}{\rho} \frac{d\rho}{dr} \\
\frac{d\rho}{dr} &= -\frac{Gm(r)\rho(r)^2}{\Gamma P r^2},
\end{aligned}
\tag{3.39}
$$

and we only use the cold part of the EOS. To solve for the neutron star profiles, the system of differential equations is solved by a fourth order accurate Runge-Kutta solver written in `python`. As the central density is an unknown initial condition for the solver, a first guess value is estimated, the solver produces a profile, then adjustments can be made to obtain the desired mass of the produced neutron star. In our case we limit our parameter space to the fiducial neutron star mass of $M_{NS} = 1.4 M_{\odot}$. Profiles are produced individually for each equation of state considered, at a chosen resolution. In this case we set the resolution of the profile to be finer than the resolution of the grid cells. We test the profiles by evolving this hydrostatic star for 10 dynamical times and verifying that it remains in near steady-state with low kinetic energy over several dynamical times (Figure 3.2). The dynamical time for a single neutron star τ_{dyn} is defined to be the characteristic length scale over the characteristic velocity:

$$
\begin{aligned}
\tau_{dyn} &= \frac{R}{v_{esc}} \\
\tau_{dyn} &= \frac{R}{\sqrt{\frac{2GM}{R}}} \\
\tau_{dyn} &= \sqrt{\frac{R^3}{2GM}},
\end{aligned}
\tag{3.40}
$$

where v_{esc} is the escape velocity. For our fiducial case (`APR4` EOS, $M = 1.4 M_{\odot}$, $R = 12.6$ km) $\tau_{dyn} = 7.3 \times 10^{-5}$ s.

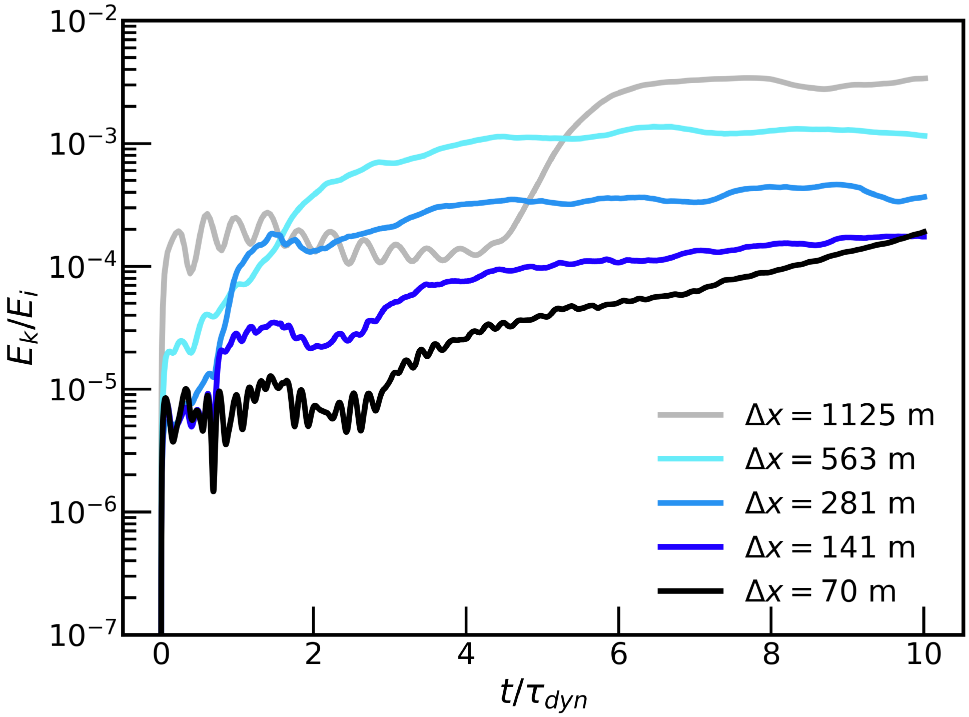

Figure 3.2: Hydrostatic equilibrium test, showing ratio of kinetic energy E_k in stellar material to internal energy E_i as a function of time, in units of the dynamical time of the star τ_dyn, for a $1.4 M_\odot$ neutron star constructed with the `APR4` EOS at the center of the computational domain. Different curves correspond to increasing resolution, as labeled with cell size in the legend.

In an effort to increase the initialization efficiency of the simulation at high resolution, I implemented a piece-wise polynomial fit to the initial profiles as described in Section A.4.

3.2 Numerical Hydrodynamics

We use `FLASH` version 4.5 (Fryxell et al., 2000; Dubey et al., 2009) to solve the equations of Newtonian hydrodynamics with self-gravity in 2D cylindrical coordinates:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{3.41}$$

$$\frac{D\mathbf{v}}{Dt} = -\frac{\nabla P}{\rho} - \nabla \Phi + \mathbf{f}_{\text{centrifugal}} + \mathbf{f}_{\text{coriolis,z}} \tag{3.42}$$

$$\frac{D\epsilon}{Dt} - \frac{P}{\rho^2}\frac{D\rho}{Dt} = 0 \tag{3.43}$$

$$\nabla^2 \Phi = 4\pi G \rho \tag{3.44}$$

$$\frac{d\ell_{orb}}{dt} = \frac{1}{2}\frac{\ell_{orb}}{|d|}\left(\frac{\mathrm{d}d}{\mathrm{d}t}\right)_{\text{gw}} + \mathbf{r}_{\perp} \times \mathbf{f}_{\text{coriolis},\xi} \tag{3.45}$$

where $D/Dt \equiv (\partial/\partial t + \mathbf{v}\cdot\nabla)$, where $\mathbf{v}$ is the poloidal velocity, ρ is the density, P is the total gas pressure, ϵ is the total specific internal energy, $\mathbf{v}$ is the velocity, Φ is the gravitational potential, G is the gravitational constant, t is time, and the inertial accelerations $\mathbf{f}_{\text{centrifugal}}$ and $\mathbf{f}_{\text{coriolis,z}}$ are defined in equations (2.61) and (2.68), $\mathbf{r}_{\perp}$ is the vector perpendicular to the ρ-axis, connecting the rotation axis and a given cell, and $\mathbf{f}_{\text{coriolis},\xi}$ is given by Equation 2.69.

The system of equations is solved with the dimensionally-split Piecewise-Parabolic Method (PPM; Colella and Woodward 1984), the multipole self-gravity solver of Couch et al. (2013), and is closed with a piece-wise polytropic equation of state (EOS) as discussed in Section 3.1.2.

While the gravitational wave losses are strictly valid only for point masses at the location of the centers of the stars (with equal masses and at equal distances from the center of mass at the origin), we apply this source term to all stellar material that has non-zero ℓ_{orb} (i.e., material ejected after the collision as described in Section 2.3).

The computational domain is discretized with a uniform grid. We choose our target resolution in relation to the pressure scale height $H_{\rm p}$ near the star's surface (Figure 3.3), such that the pressure scale height of the neutron star at 99% of the neutron star radius can be resolved with at least 10 cells. In order to ensure our simulation is generalizable, we consider three representative EOS that span the range of compactnesses present in the EOS parameter space, `APR1` (soft) (Akmal et al., 1998), `APR4` (intermediate) (Akmal et al., 1998), and `H3` (stiff) (Lackey et al., 2006). A scale height is the characteristic distance over which a variable changes:

$$|H_S| = \left| \frac{S(r)}{\frac{dS(r)}{dr}} \right|. \tag{3.46}$$

For exponentially varying quantities, this distance involves a change by a factor e. In the case of pressure:

$$\frac{\mathrm{d}P}{\mathrm{d}r} = -\frac{Gm(r)\rho(r)}{r^2}, \tag{3.47}$$

so the pressure scale height is

$$H_P = \left| \frac{P}{\frac{Gm(r)\rho(r)}{r^2}} \right| = \frac{P}{\rho|\nabla\Phi|}. \tag{3.48}$$

Given the numerical dissipation properties of PPM (e.g., Porter and Wood-

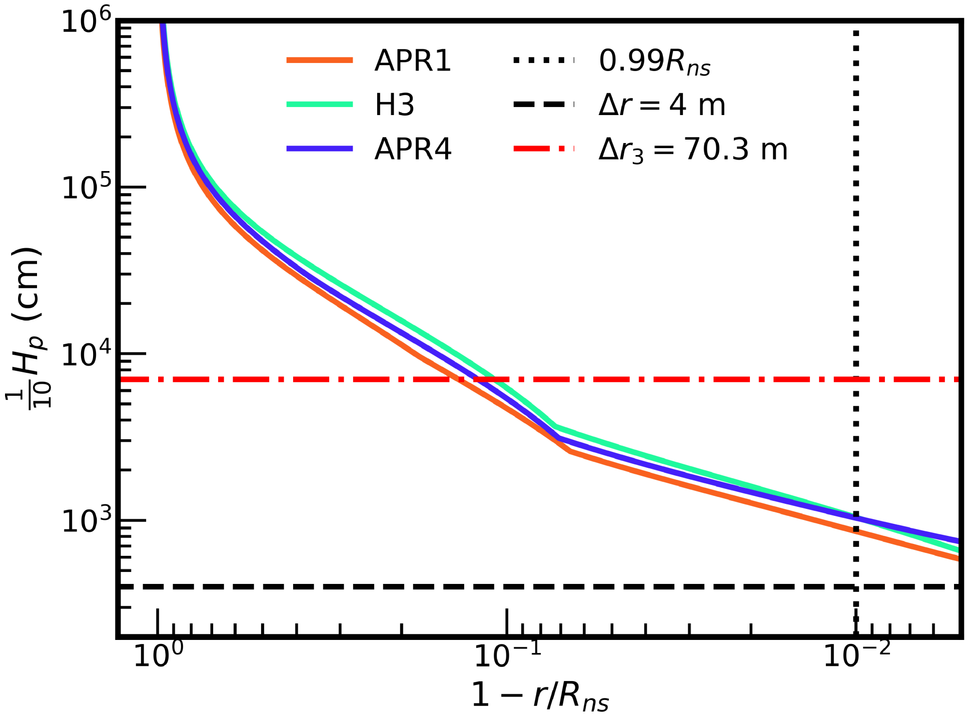

Figure 3.3: Plot of 1/10th the pressure scale height of neutron stars of three different representative Equations of State (`APR4`, `APR1`, and `H3`). The vertical dotted line marks 99% of the radius of the given neutron star, and the dashed line shows the maximum resolution desired in our study, resolving the neutron star of each equation of state to 99% its radius with 10 cells. $\Delta r_3 = \Delta x_3$ the maximum resolution of the simulations presented in this thesis (Section 3.2.2)

ward 1994), we consider a length scale "resolved" if we can cover it with at least 10 computational cells. Our target resolution is set so the pressure scale height near the stellar surface (out to 99% of $R_{\rm ns}$) is resolved with 10 cells. (Section 3.2.1). Thus all but the outermost $\sim 7 \times 10^{-4} M_\odot$ of each neutron star is well resolved at the desired maximum resolution $\Delta r = 400$ cm. For the highest resolution presented in this thesis, $\Delta x_3 = 70.3$ m, all but the outer $\sim 10^{-1} M_\odot$ is well resolved

The orbital stability of the configuration is tested by setting ω, the angular velocity of the binary system, to the Keplerian frequency and verifying that the stars maintain their initial positions in the absence of gravitational wave losses. This stationarity test, carried out with the star centers separated at 8 stellar radii, requires including at least 64 multipoles in the self-gravity solver. We use 128 multipoles by default in all of our simulations.

The domain is set to use a reflecting inner ρ-boundary, and outflow boundary condition preventing material from falling back into the domain at the outer ρ-boundary, as well as upper and lower z-boundaries. The variable floors (pressure, density, and internal energy) are set in the final version of the simulation to be constant (thus variables k_{in} and k_{out} are not used), and will be discussed further in Section 3.2.1.

3.2.1 Initial Conditions

Simulations are initialized with two identical stars placed along the z-axis, at a separation slightly larger than two stellar radii, and equidistant from the origin (Figure 2.9). The initial velocities are in the z-direction and toward the origin, with a magnitude equal to one half of the decay rate of the orbital seperation due to gravitational waves (Equation 2.4), typically $\approx 2 \times 10^8$ cm/s.

The stars are initially embedded in a low-density flat ambient medium. Density is set such that the ambient mass is negligible in comparison to the lower estimates of free neutron ejecta from Ishii et al. (2018) ($M_{fn} \sim 10^{-7}\ M_\odot$). The ambient pressure is set sufficiently low as to not limit the time step of the simulation, and the remaining thermodynamic variables determined by the EOS.

3.2.2 Models Evolved

Table 3.1 shows all models with their parameters. Our default case is a pair of stars with masses $M_{\rm ns} = 1.4 M_\odot$ each, built with the APR4 EOS (Akmal et al., 1998), yielding a radius $R_{\rm ns} = 12.6$ km, which is consistent with NICER radius measurements of J0030+0451 (Miller et al. 2019, and Riley et al. 2019), a nearby 1.4 $M_\odot$ millisecond pulsar, thus yielding a realistic compactness. This fiducial case is evolved for three preliminary resolutions $\Delta x_{1,2,3} = \{281.3, 140.6, 70.3\}$ m which are competitive with some of the finest resolutions used for grid based numerical relativity simulations of binary neutron star mergers run to date (123 m by Radice et al. (2018a), and $63 - 86$ m by Kiuchi et al. (2017)). The stars begin with a scalar specific angular momentum defined by Equation 2.19, which is evolved according to the emission of gravitational waves (Section 2.3). Source terms for the Centrifugal and Coriolis fictitious forces necessitated by the co-rotation of the simulation domain are calculated according to the formulae 2.61, 2.68 and 2.69. The initial z-velocity of each star is given by Equation 2.6, and the initial orbital velocity is given by Equation 2.17.

In our default configuration, we simulate, for all three resolutions, a case with centrifugal acceleration (Equation 2.61), Coriolis acceleration (Equation 2.68) and gravitational wave losses (Equation 2.80), these models are denoted by $AF\{1, 2, 3\}$ for "All Forces". To estimate the range of uncertainty introduced by our approximations, and to connect with previous work, we model a "head-on" collision, where the stars are set to collide at the free-fall velocity, orbital angular momentum is zero initially, and assumed to stay small as the stars begin just before collision, thus these inertial forces are neglected in this case. This model is denoted by $FF\{2\}$ and is evolved at the Δx_2 resolution only. In this case the z-velocity of each star is set to be the Keplerian velocity (Equation 2.6).

We also consider an intermediate case, where the centrifugal and Coriolis forces are turned off, as well as the gravitational wave emission source term, and the angular momentum of the system is set to zero ($v_{orb} = 0$). The initial z-velocity is set by the semi-major axis decay rate at the initial separation distance d (Equation 2.6). This intermediate case is denoted $OR\{2\}$ for "On Rails", and is also evolved at resolution Δx_2 only.

Each model is evolved in a simulation domain of size 9×10^6 cm by 18×10^6

Table 3.1: Models evolved and results. Columns from left to right show model name, EOS, NS mass, NS radius, cell size, and unbound mass ejected. The latter is shown as total and broken up by radial velocity.

Model	EOS	$M_{\rm ns}$	$R_{\rm ns}$	Δx		$M_{\rm ej}$ $(M_\odot)$	
		$(M_\odot)$	(km)	(m)	total	$v > 0.6$c	$v < 0.6$c
AF1	APR4	1.4	12.6	281.3	6.3×10^{-5}	5.0×10^{-5}	1.3×10^{-5}
AF2	APR4	1.4	12.6	140.6	6.5×10^{-5}	5.3×10^{-5}	1.2×10^{-5}
AF3	APR4	1.4	12.6	70.3	6.4×10^{-5}	4.9×10^{-5}	1.5×10^{-5}
OR2	APR4	1.4	12.6	140.6	4.0×10^{-2}	1.8×10^{-2}	2.2×10^{-2}
FF2	APR4	1.4	12.6	140.6	3.0×10^{-2}	1.4×10^{-3}	2.9×10^{-2}

cm, with stars placed symmetrically about the ρ-axis at a distance of $z_{center} = 1.3 \times 10^6$ cm. Despite developing the capability to easily explore the neutron star EOS parameter space, for our purposes here we limit all runs to use the fiducial `APR4` EOS. Outside the two neutron stars, we fill the remainder of the simulation domain with ambient material. The initial ambient material has a constant density $\rho_{ambient} = 10^4$ g/cm^3, pressure $P_{ambient} = 10^{25}$ dyn/cm^2, and internal energy as determined by the EOS $\epsilon_{ambient} \approx 10^{21}$ ergs. We also implement a flat floor for each of the thermodynamic variables: $\rho_{small} = 10^3$ g/cm^3, $P_{small} = 10^{21}$ dyn/cm^2, and $\epsilon_{small} = 10^{16}$ ergs. The simulation is run for ten dynamical times, defined be the Keplerian orbital period of the stars at their initial separation:

$$t_{dyn} = \sqrt{\frac{4\pi^2}{G(m_1 + m_2)} \frac{1}{d^3}}. \tag{3.49}$$

Note that this dynamical time is different than that defined in Equation 3.40 for a single star. For the simulations presented here ($m_1 = m_2 = 1.4 M_\odot$, $d = 2z_{center} = 2(1.3 \times 10^6$ cm)), we have a dynamical time of $t_{dyn} = 1.6$ ms. 100 plot files are dumped during the simulation (over $10t_{dyn}$), discretizing the simulation into 0.1 t_{dyn} segments.

3.3 Data Extraction and Analysis

3.3.1 Extracting Data

FLASH outputs simulation data for analysis as HDF5 files at designated times. This file type allows many different types of data such as arrays of different sizes and dimensions to be stored together in an efficient package that is portable across platforms. In order to analyze the simulation data, and to ask questions such as how much mass is ejected, it is easiest to work with the data in a uniform grid format.

Adaptive Mesh Refinement

In order to extract the mass ejecta data from the FLASH output HDF5 files with a simulation run using Adaptive Mesh Refinement, as described in Section 3.3.2, we must first convert the data structure to uniform grid. This involves the use of either the python package yt, or my custom parallelised extraction code, each of which is discussed in Section A.1.

Uniform Grid

In the case of uniform grid, no use of a covering grid is necessary for data extraction, and thus we can jump straight to the data extraction method described in Section 3.3.2.

3.3.2 Mass Ejecta Measurement

Now that the data has been loaded as a uniform grid structure, we can track the mass ejection from the system. To do this, we consider the amount of unbound mass crossing a semi-circular threshold integrated over time. First we define unbound matter, as matter having a positive Bernoulli parameter:

$$B = \frac{1}{2}v^2 + e_{int} + \frac{P}{\rho} + \Phi, \tag{3.50}$$

where

$$\begin{cases} B > 0 & \text{unbound material} \\ B < 0 & \text{bound material} \\ B = 0 & \text{marginally bound material.} \end{cases} \tag{3.51}$$

To track the unbound material leaving the system, we begin by setting an outflow sampling radius r_{ej} through which to calculate the ejected mass. This is set to be slightly larger than the initial extent of the neutron stars ($\sim 2.4R_{NS}$) as seen in Figure 3.4 .

We then can say that the amount of mass leaving the system is:

$$\Delta M = \int \dot{M} dt, \quad B > 0 \tag{3.52}$$

This calculation can be discretized as mass flux through the r_{ej} surface over time. To do this we consider the fact that due to rotational symmetry about the z-axis, this semi-circular outflow threshold corresponds to a spherical surface in 3 dimensions. If we want to break the flux through this threshold down cell-by-cell we need to know the area of a "square" projected onto a sphere's surface, or in other words the square subtended by the spherical angle increments $d\theta$ and $d\varphi$, as this is the geometry of the interface between each cell along the threshold and the spherical threshold itself. The geometrical derivation of this area for infinitesimally small angles is shown in Figure 3.5, left pane.

The area of an infinitesimally small "square" spherical shell section is given by:

$$A = r_{ej}^2 \sin\theta d\theta d\varphi. \tag{3.53}$$

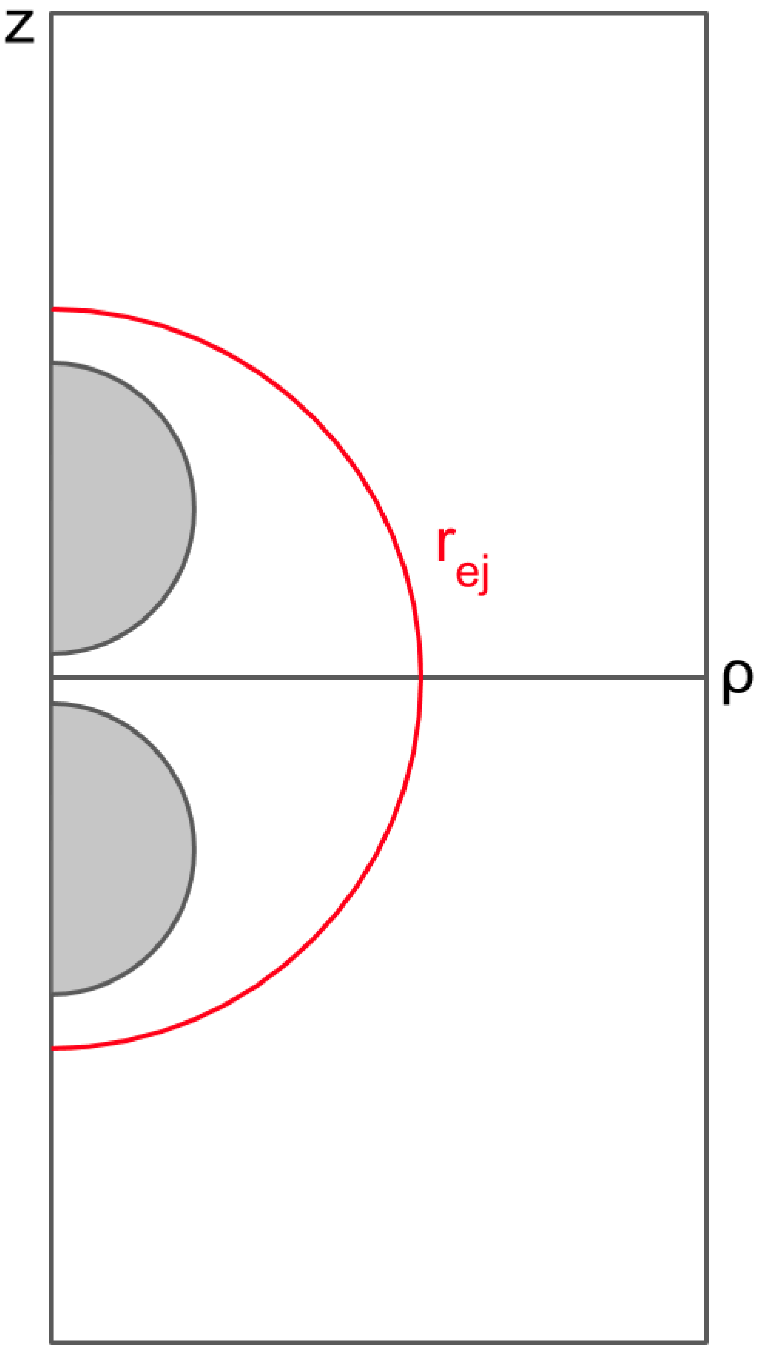

Figure 3.4: Schematic of mass ejecta sampling radius r_{ej} relative to the initial positions of the neutron stars.

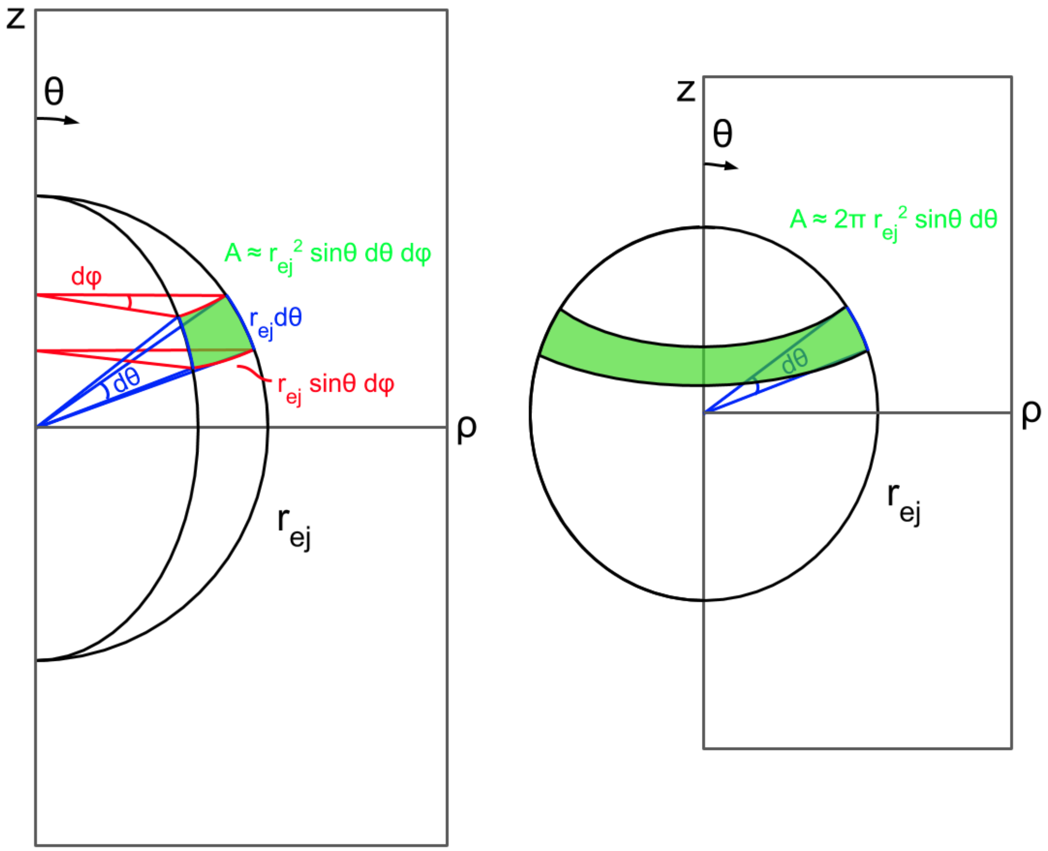

Figure 3.5: Left: Schematic of a square spherical shell subsection shown for the ejection threshold r_{ej} in the simulation domain. Right: Same as the left pane, but considering azimuthal symmetry.

If we consider cells which have a finite size despite being much smaller than the scale of the output threshold, the area is approximately given by:

$$\begin{aligned} A &\approx r_{ej}^2 \sin\theta \Delta\theta \Delta\varphi \\ A &\approx r_{ej}^2 \Delta\cos\theta \Delta\varphi \end{aligned} \tag{3.54}$$

Where $\Delta\varphi$ is the angular width of the cell in the φ-direction, which due to azimuthal symmetry this is set to be 2π, $\Delta\theta$ is the angular width of the cell in the θ-direction, and thus $\Delta\cos\theta$ is the difference in $\cos\theta$ between the two sides of the cell. Therefore we can integrate the outflow by summing the flux crossing the threshold of each cell that is intersected by the semi-circular outflow threshold. Each of which due z-axis-symmetry is represented by a slice of a spherical shell (Figure 3.5, right panel).

In this case where we are considering z-axis symmetry, the area of the spherical shell slice is given by:

$$A \approx 2\pi r_{ej}^2 \Delta \cos\theta. \tag{3.55}$$

Through the use of dimensional analysis, we can easily see mass flux $\mathcal{F}_M$ with units of $gcm^{-2}s^{-1}$ is given by:

$$\mathcal{F}_M = \rho v_r = \rho \mathbf{v} \cdot \hat{r} \tag{3.56}$$

where $\hat{r}$ is the spherical radial direction, or the normal to the outflow threshold. To get the mass crossing the boundary in a given amount of time we use:

$$\begin{aligned} \Delta M &= \mathcal{F}_M A \Delta t \\ \Delta M &= 2\pi r_{ej}^2 \Delta \cos\theta \rho v_r \Delta t \end{aligned} \tag{3.57}$$

So the amount of mass crossing the surface area of an angular bin in time Δt is given by:

$$\boxed{\Delta M = 2\pi r_{ej}^2 \Delta \cos\theta \rho |v_r| \Delta t \cdot mask(B > 0) \cdot mask(v_r > 0)}, \tag{3.58}$$

where

$$v_r = v_\rho \sin(\theta) + v_z \cos(\theta). \tag{3.59}$$

Here $mask(B > 0)$ is a mask applied so only gravitationally unbound outflow is considered to contribute to the "outflow mass", and $mask(v_r > 0)$ is applied so only out-flowing material is considered. Notice that we are calculating a density flux crossing the outflow threshold. Thus, this can be thought of as a constant increment of material at density ρ given by the density of the cell nearest the outflow threshold crossing the threshold in time Δt. Thus we only need to consider the cell closest to the threshold, no cells interior to that as while their mass may cross the threshold within the given time step, we are essentially doing a histogram type integration where we choose one constant value of the density for the given time step. Also, as we are discretizing this integration in terms of space (summing cell-by-cell), and in time considering time resolution Δt (set by the number of plot files dumped by the simulation). Thus the accuracy of this integration is limited by both of these factors, with an increase in temporal and spatial resolution resulting in a more accurate account of mass outflow.

Velocity and Angular Distribution

Using the method outlined in Section 3.3.2 we can quantify the total mass ejected from the system, as well as how it changes during the simulation time. However, if we want to consider the angular distribution of mass ejected, or how the ejecta is distributed in terms of velocity, we can bin our integration in terms of spherical polar angle θ as shown in Figure 3.5 left pane, or in terms of velocity.

This is achieved by applying a sorting system in which mass ejecta is sorted in terms of polar angle θ, and spherical radial velocity v_r during the integration. Further, as motivated by the physics of the merger geometry, the dynamical ejecta from tidal tails in the orbital plane can be distinguished from contact plane ejecta (plane where the two neutron stars first contact one another), mass ejecta binned in terms of velocity is also filtered by orbital plane vs. contact plane region as shown in Figure 3.6 using the arbitrarily chosen angle $\pi/6$.

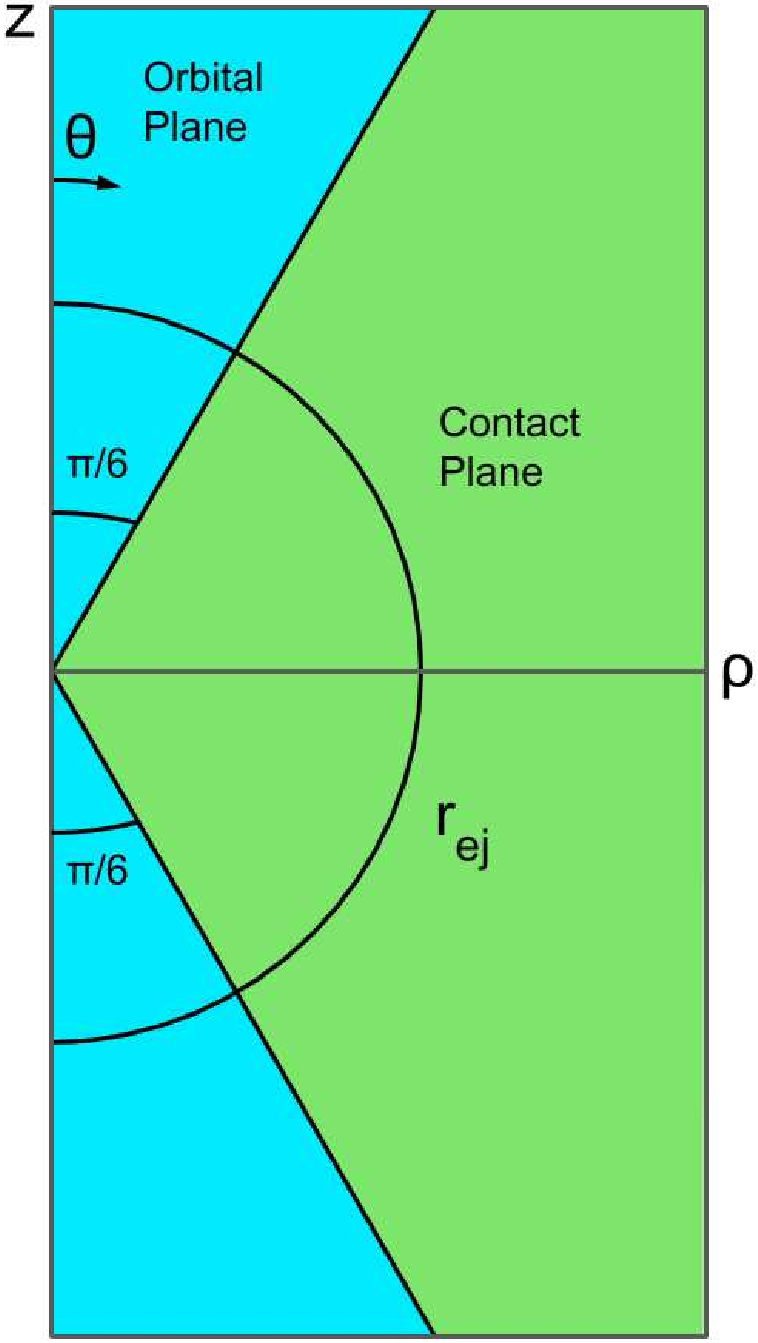

Figure 3.6: Schematic of how ejecta is divided in terms of orbital plane (blue), vs. contact plane (green).

Testing

To test my mass outflow post-processing module, I ran a simple code test of a spherical radial shell of material of constant pressure and density, advecting spherically outwards across r_{ej} as shown in Figure 3.7 left pane, and 3.7 right pane.

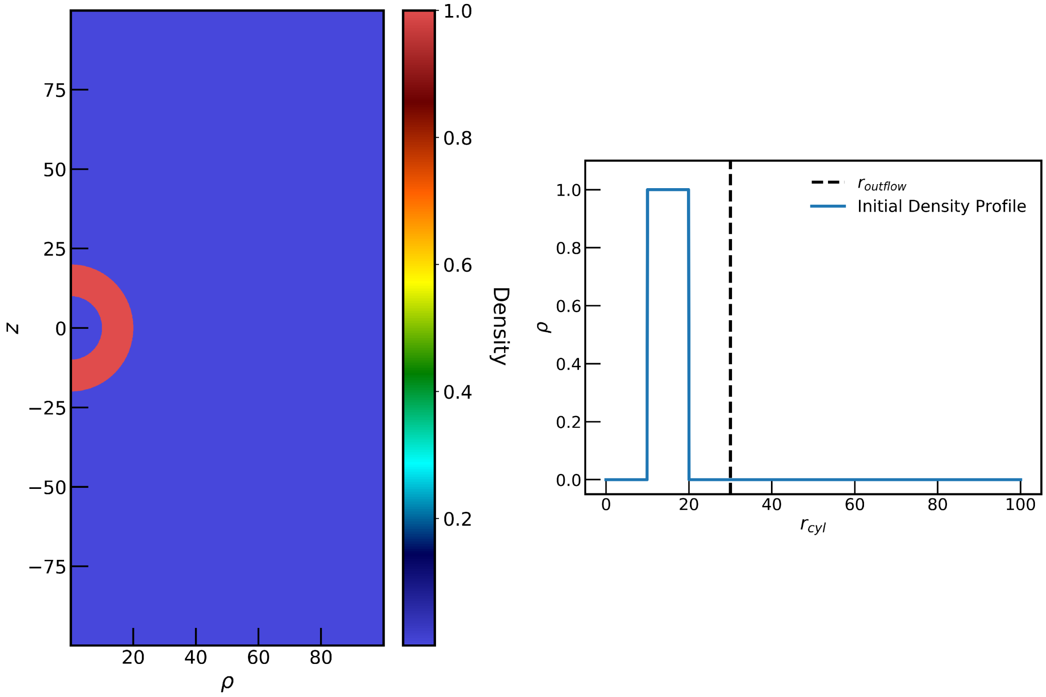

Figure 3.7: Left: Initial density colour plot of the mass outflow test. For simplicity, all variables are considered unit-less. Right: Initial radial profile of the density in the mass outflow test.

Setting the outflow threshold in terms of generic distance units to be $r_{ej} = 30$, and allowing the material to advect entirely across the outflow threshold, we can compare the theoretical mass outflow to the value obtained by our discretized integration method. Doing so we recover the ejected mass to $< 5\%$ error, where the source of error is introduced by the finite sampling in space and time of the numerical integration.

Chapter 4

Results

In this chapter we present the results of a preliminary set of simulations that, while not yet at our final target resolution, allow us to investigate the dependence of the contact plane ejecta on resolution.

4.1 Overview of Default Case (AF) and Resolution Dependence

In the AF models, the stars come into contact almost immediately at the beginning of the simulation, ejecting material in the contact plane as low density ejecta is launched from the surfaces of the neutron stars. Material from the contact plane begins to cross the sampling radius at a time $\sim 0.2\ t_{dyn}$ after contact, where $t_{dyn} = 1.6$ ms and the simulation is run for 10 t_{dyn} in total. On a slightly longer timescale of $\sim 0.4\ t_{dyn}$, material begins to cross the sampling threshold in the orbital plane. After 5 t_{dyn} the remnant remains distended in the orbital plane due to the the remaining rotational velocity v_ξ not being fully removed through the emission of gravitational waves. This sequence of events is illustrated in Figure 4.1.

To test the dependence of this ejecta on spatial resolution, I run the default scenario at 3 grid cell sizes: $\Delta x_{1,2,3} = \{281.3, 140.6, 70.3\}$ m, denoted by $AF\{1,2,3\}$, respectively. As the resolution is increased, the initial low density ejecta from the surface of the neutron star shows more small-scale features as it expands outward showing signs of Rayleigh-Taylor instabilities. This increase in resolution also produces an apparent increase in ejecta in the orbital plane, while the change in the contact plane is less noticeable.

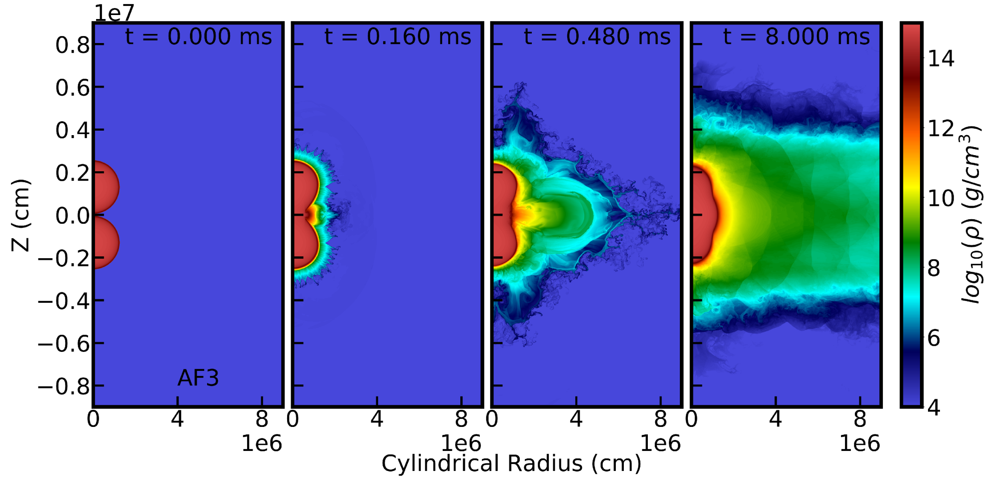

Figure 4.1: Pseudo-colour plot of the density of the simulation in the default “All forces” case with resolution $\Delta x_3 = 70.3$ m, at various times, as labeled. For reference, $1t_{dyn} = 1.6$ ms in this model (Equation 3.49).

The angular distribution of ejecta is mostly isotropic, as shown in Figure 4.2, with some deficit in some directions in the lowest resolution model. In the orbital plane the bins covering polar angles $[0, \pi/10]$ and $[9\pi/10, \pi]$ show a monotonic resolution dependence, with increasing the resolution resulting in more ejecta escaping in these directions. However, towards the contact plane the results are mixed, and there is no clear correlation.

The velocity distribution of mass ejected from the system in the “all forces” model sequence extends up to $v \sim 0.7c$ for the majority of mass ejected, with most of the mass ejected at the highest velocities. Due to the limitation of the use of Newtonian physics there is a small amount ($\sim 10^{-11} M_\odot$) of superluminal ejecta in the highest resolution run ($AF3$). There does not appear to be a clear

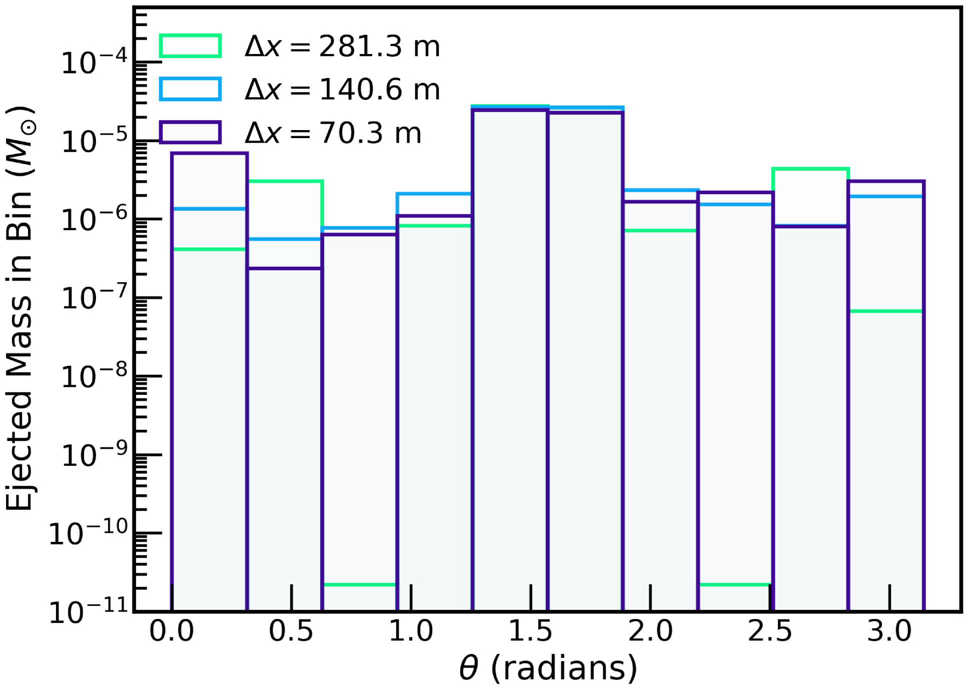

Figure 4.2: Mass ejection in the default "all forces" scenario binned by spherical polar angle θ. The distributions are shown for the three resolutions $\Delta x_1 = 281.3$ (green), $\Delta x_2 = 140.6$ m (blue), and $\Delta x_3 = 70.3$ m (purple).

trend with increasing resolution in the contact plane (Figure 4.3). However, in the orbital plane the increased resolution allows more ejecta to be resolved. In particular, these additional ejecta are most prominent at speeds $v \sim 0.3c$, however the additional ejecta is only on the order of $\sim 10^{-6} M_\odot$.

The vast majority of the mass ejected from the system does so on a timescale of less than a dynamical time in the contact plane (Figure 4.5). However, the majority of the slower velocity material ($0.1 < v/c < 0.3$), which is ejected in the orbital plane occurs on a longer timescale. This is notable as the remnant of the merger remains distended in the z-direction for the remainder of the simulation simply oscillating and ejecting additional mass. Episodic mass ejection associated with oscillations in the merger remnant are a common feature of numerical relativity simulations of binary neutron star mergers (e.g., Bauswein et al. 2013).

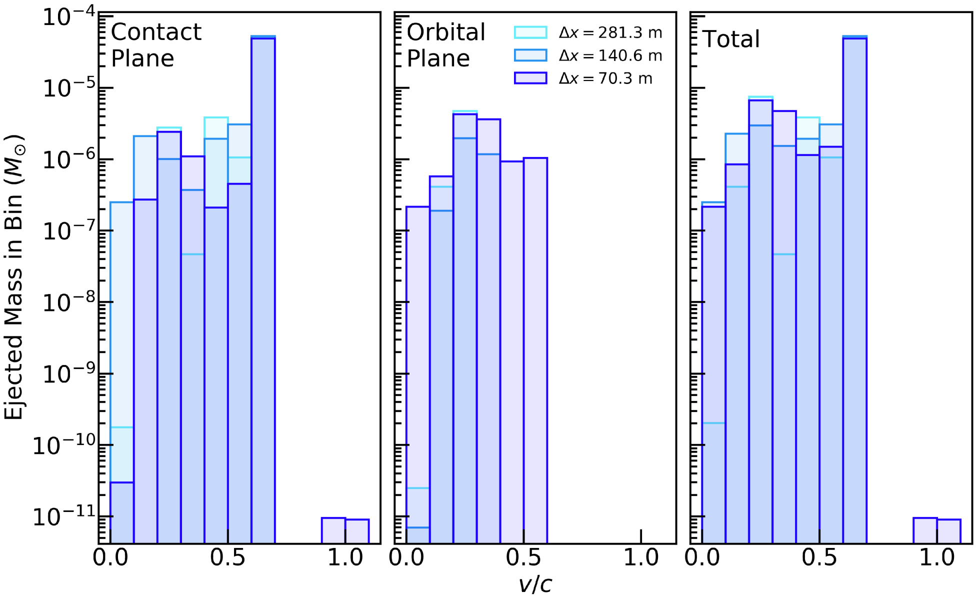

Figure 4.3: Velocity distribution of mass ejected from the system in the default sequence $AF\{1,2,3\}$. Mass ejecta is binned by velocity, at the three resolutions $\Delta x_1 = 281.3$ (teal), $\Delta x_2 = 140.6$ m (blue), and $\Delta x_3 = 70.3$ m (purple).

Plotting the two-dimensional histogram of ejecta mass binned in terms of both velocity and angle (Figure 4.4), the angular dependence of fast ejecta in the contact plane region compared to the slower ejecta in the orbital plane is made more apparent. Mass is ejected in the contact plane in greater quantities at higher velocities, while orbital plane ejecta is produced predominantly at slower velocities. The contact plane ejecta in angular bins closest to the rotation axis are launched within a narrow range of radial speeds around $0.6c$.

4.2 Models with Different Dynamic Components

As the inspiral of the merger is not perfectly reproduced in our simulation due to the use of Newtonian physics (for example, by the lack of a final plunge phase), we consider a range of force prescriptions (as described in Section 3.2.2) in order to estimate the uncertainty in our results.

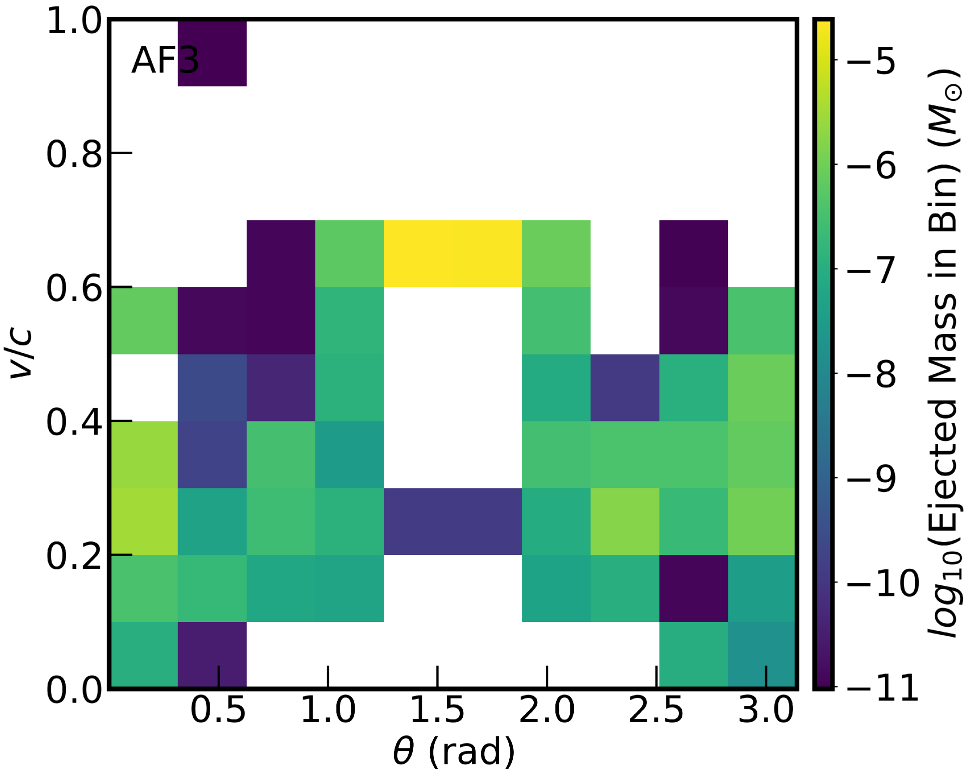

Figure 4.4: Two dimensional histogram of mass ejected in the AF model at highest resolution ($\Delta x_3 = 70.3$ m). The mass ejecta is binned by velocity on the y-axis and by angle on the x-axis.

Figure 4.6 demonstrates the differences in evolution between the "all forces" simulation, the intermediate "on rails" simulation, and the "head-on" merger simulation (See Section 3.2.2 for definitions). The different merger velocities are apparent from the left two columns, as by 0.1 t_{dyn} the "head-on" simulation has resulted in significantly more ejecta in the contact plane than the "on rails" and "all forces" simulations. In the case of the "head-on" and "on rails" simulations, the neutron stars have $v_{orb} = 0$ cm/s, and thus no orbital angular momentum, as well as no inertial forces. This allows the remnant to collapse into a more spherical, less distended geometry than the "all forces" scenario.

The angular distribution of mass ejected in the different force scenarios is shown in Figure 4.7. The $AF2$ scenario predictably produces the smallest amount of ejecta across the angular distribution. Notably however, across most

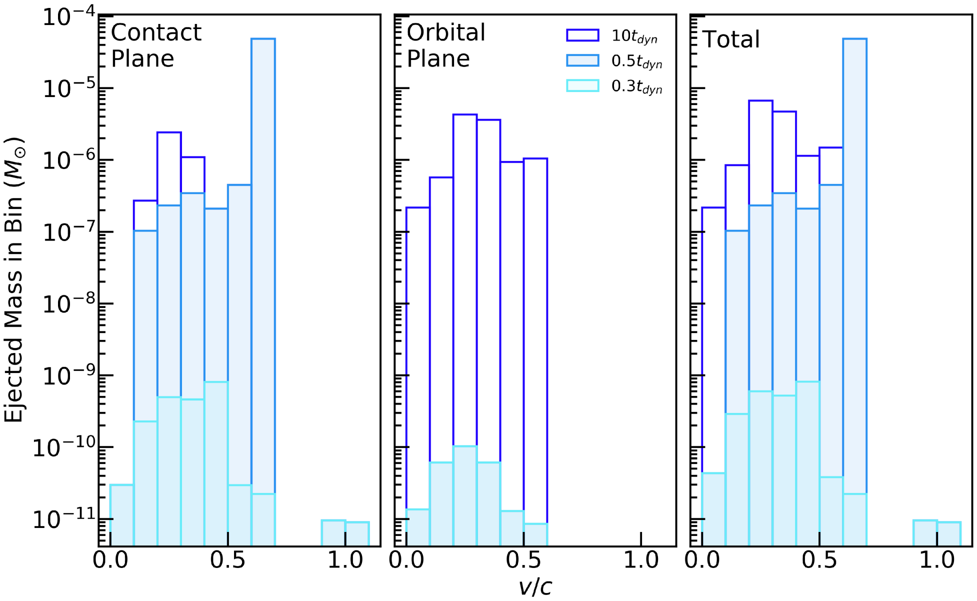

Figure 4.5: Time evolution of mass ejected in the all forces model at highest resolution ($\Delta x_3 = 70.3$ m). Mass ejecta is binned by velocity, with histograms corresponding to mass ejected up until 0.3 t_{dyn} (teal), 0.5 t_{dyn} (blue), and 10 t_{dyn} (purple) of simulation time. In the central pane, histograms $0.3t_{dyn}$ and $0.5t_{dyn}$ overlap entirely.

of the angular distribution the "on rails" simulation produces a comparable amount of ejecta to the "head-on" simulation, despite having a smaller initial collisional velocity ($v_{z,i} \approx 2 \times 10^8 cm/s$) than the "head-on" scenario ($v_{z,i} \approx 6 \times 10^9 cm/s$). With the slower collisional velocity, the "on rails" scenario produces less initial contact plane ejecta, but produces more orbital plane ejecta on the first oscillation of the remnant. This is illustrated in Figure 4.8, where initially the "head-on" scenario leads the three scenarios in mass ejected due to initial contact plane ejecta, before being surpassed by material ejected in the orbital plane in the "on rails" scenario.

The velocity distribution of mass ejected from the three scenarios `AF2`, `FF2`, and `OR2` are shown in Figure 4.9. Notably, the "on rails" scenario produces $\approx 10^{-3} - 10^{-2} M_\odot$ of high velocity ($0.6c < v < 0.9c$) ejecta in the orbital plane, which is not reproduced in the "head-on" scenario. As this material is ejected within the first dynamical time (Figure 4.11) this may be due to

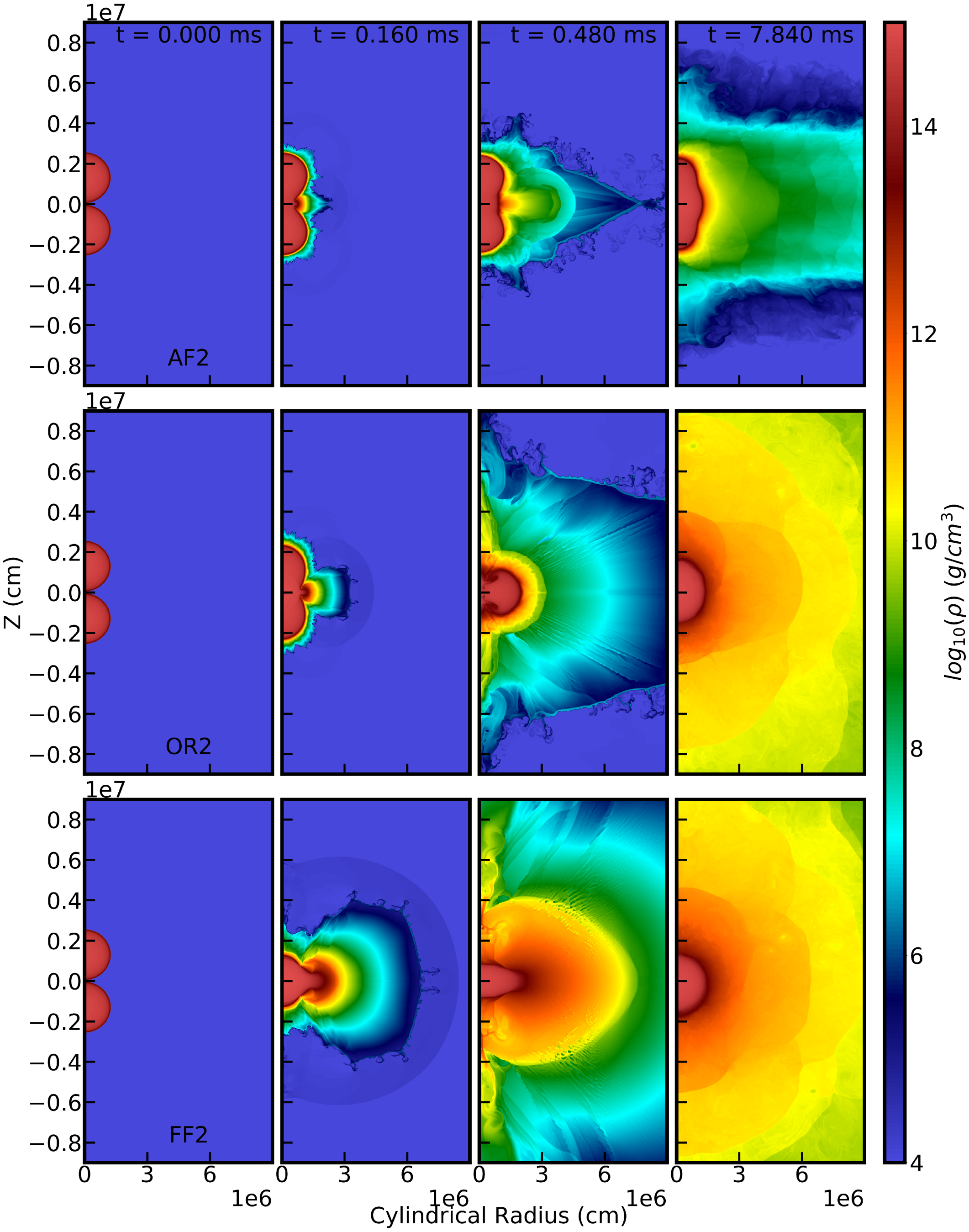

Figure 4.6: Same as Figure 4.1, but for the models $AF2$ (top), $OR2$ (middle), and $FF2$ (bottom) scenarios, each at a resolution of $\Delta x_2 = 140.6$ m.

a stronger first oscillation of the merger remnant in the "on rails" scenario where the collisional velocity v_z is smaller than the "head-on" scenario. The "head-on" scenario however, produces more low velocity ejecta ($v < 0.3c$) than

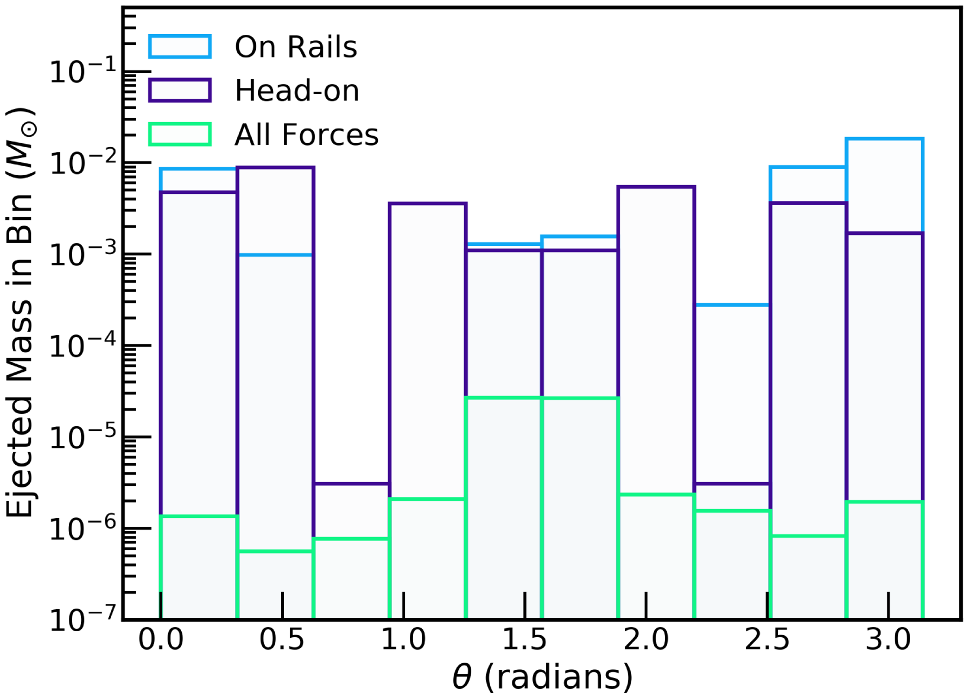

Figure 4.7: Mass ejection binned by spherical polar angle θ for the force regimes "all forces" (green), "on rails" (blue), and "head-on" (purple).

the "on rails" scenario. This material is predominantly launched over a longer timescale than the initial high velocity ejecta, and is spread across the entire domain suggesting that it is launched as a result of oscillations of the merger remnant over the remainder of the simulation time. Similarly to the "all forces" scenario, due to the limitation of the use of Newtonian physics, in the "on rails" and "head-on" simulations, a small amount ($\sim 10^{-7} M_\odot$) of ejecta is launched with superluminal velocity. Figure 4.9 includes all of the superluminal ejecta in the final bin. The velocity distribution of the superluminal ejected material can be seen in Figure 4.10.

The time evolution of the velocity distribution of ejecta in the "on rails" and "head-on" scenarios are shown in Figures 4.11 and 4.12 respectively. Each of which follow the general trend of early fast ejecta followed by longer term slow ejecta launched predominantly from the oscillations of the merger remnant.

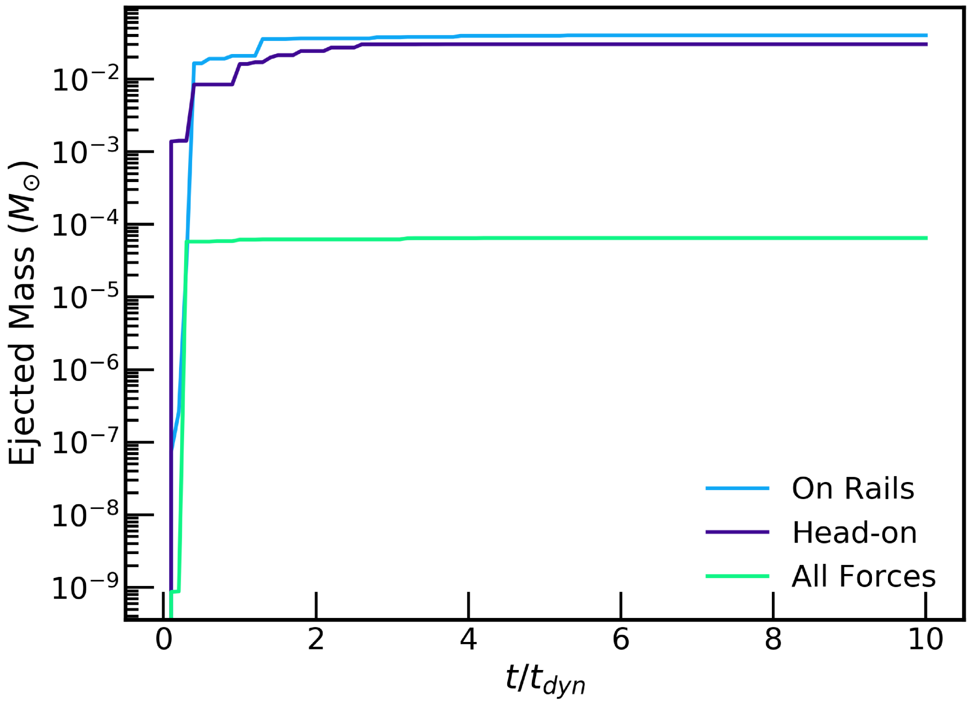

Figure 4.8: Mass ejection across the sampling threshold over time for the scenarios $OR2$ (blue), $FF2$ (purple), and $AF2$ (green).

4.3 Implications for Kilonova Precursors

The weak resolution dependence of our default "all forces" scenario (Figure 4.2 and 4.3) is not yet inconsistent with the discrepancy between the fast ejecta mass produced in SPH simulations as compared to that produced in grid based simulations. While the resolutions run here ($\Delta x_{1,2,3} = 281.3,\ 140.6,\ 70.3$ m) are competitive with some of the finest resolutions used for grid based 3D numerical relativity simulations of binary neutron star mergers to date (123 m Radice et al. (2018a), and $63 - 86$ m Kiuchi et al. (2017)), we have yet to reach our target resolution that captures the pressure scale height at 99% R_{NS} ($\sim$ 4 m for the selected equations of state). While the weak resolution dependence of the `AF1`, `AF2`, and `AF3` simulations suggests that we may be approaching convergence in fast ejecta mass already, further higher resolution tests are desired in order to ensure this is the case.

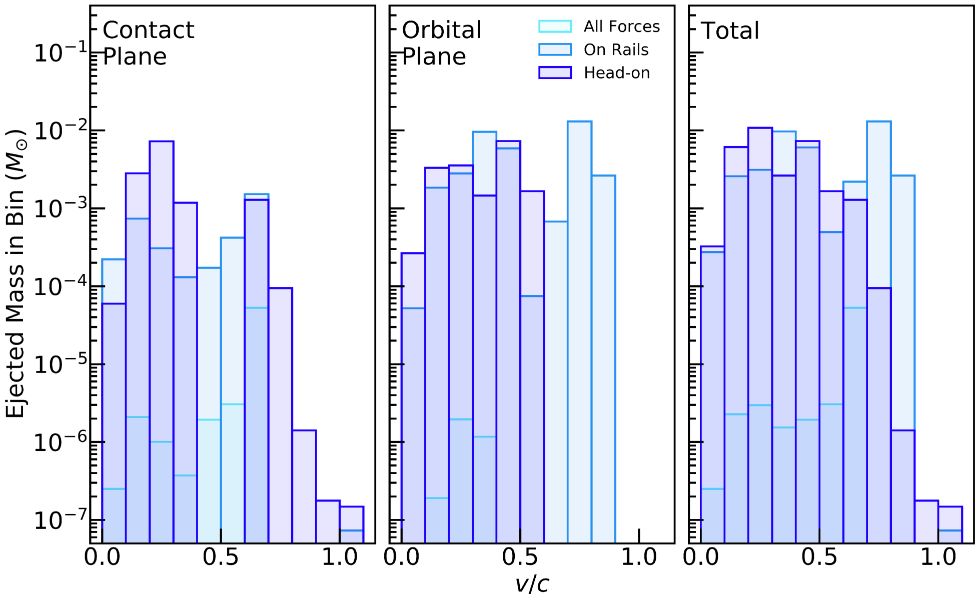

Figure 4.9: Mass ejection binned by velocity for the contact plane (left), orbital plane (middle), and entire domain (right). The different force regimes are marked as "all forces" (teal), "on rails" (blue), and "head-on" (purple). The final bin contains all mass ejected with $(v > c)$.

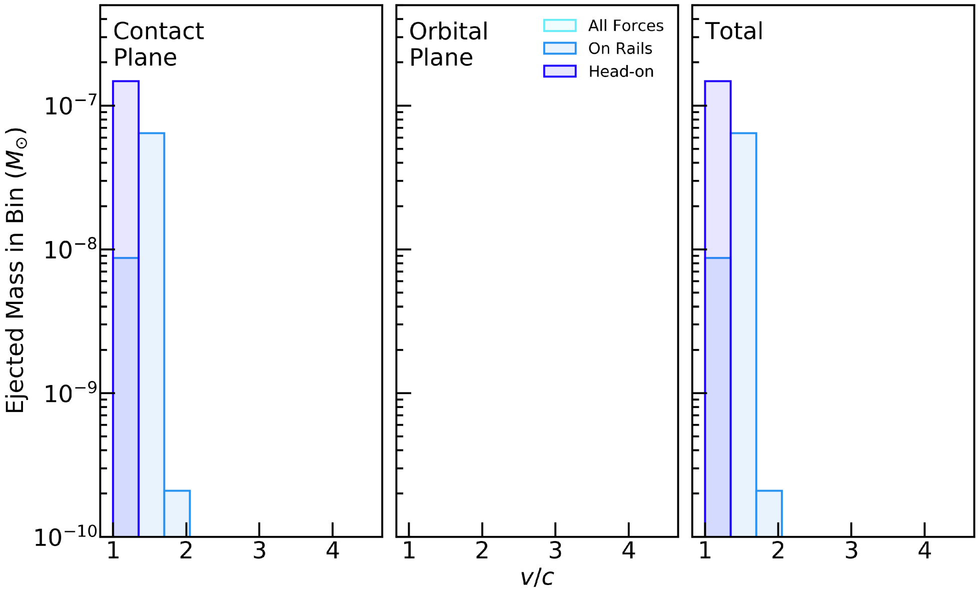

Figure 4.10: Same as Figure 4.9, but showing ejecta with $v > c$.

While the absolute value of the fast ejecta launched from any single one of our simulations of binary neutron star mergers is not reliable, due to the

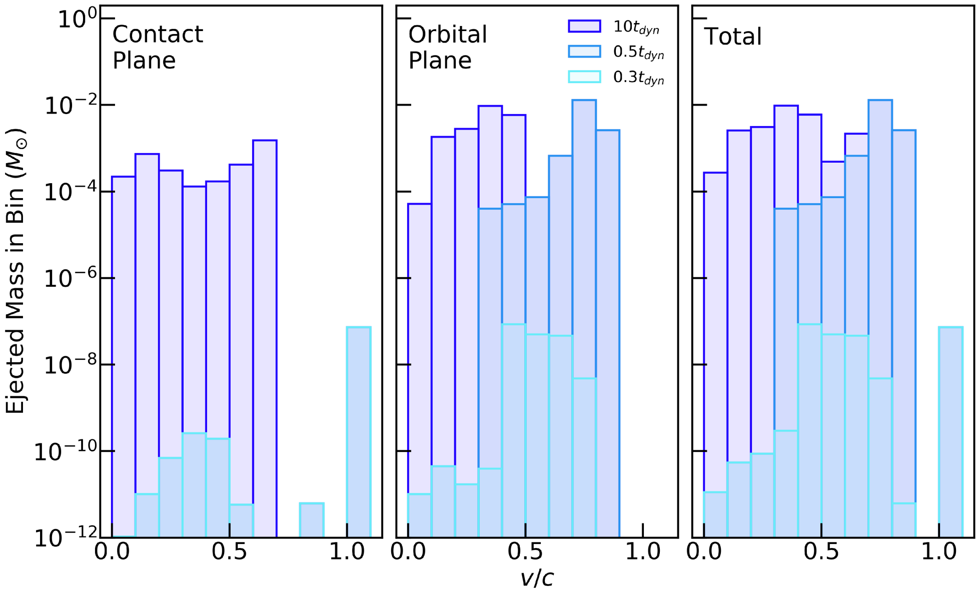

Figure 4.11: Velocity profile for mass ejected from the "on rails" simulation, with resolution $\Delta x_2 = 140.6$ m

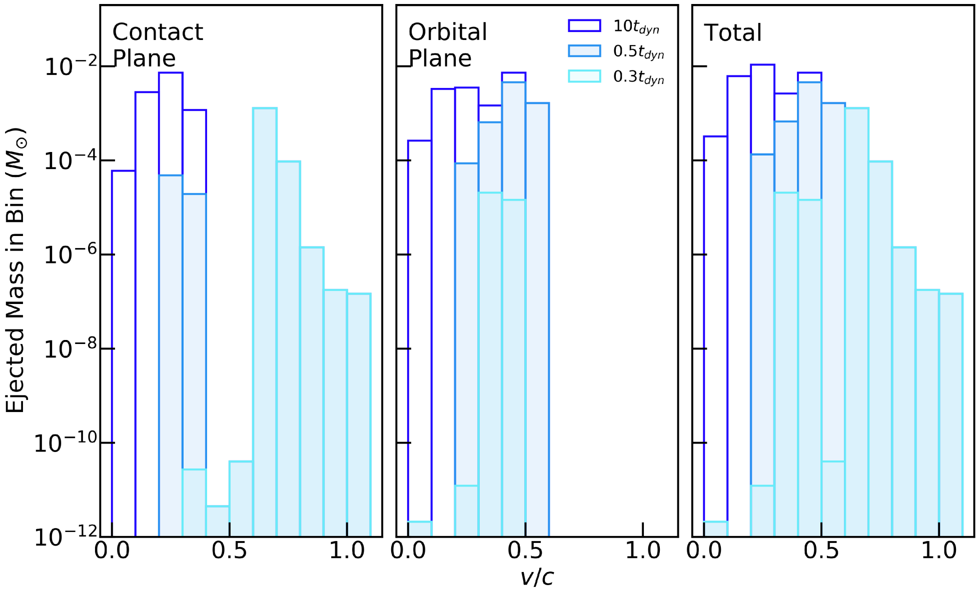

Figure 4.12: Same as Figure 4.11, except for the "head-on" simulation.

limitations of this computational experiment (Newtonian hydrodynamics, approximate thermal effects, etc.), we estimate $M_{ej,v>0.6c} \sim 10^{-5} - 10^{-2} M_\odot$ (Table 3.1) from the range of scenarios presented, which can bracket the true physical behaviour of the merging binary system. These values nonetheless overlap the range of fast ejecta masses obtained in both grid based simulations ($M_{fn} \sim 10^{-6} - 10^{-5} M_\odot$ Radice et al. (2018b)) and SPH simulations ($M_{fn} \sim 10^{-4} M_\odot$ Metzger et al. (2015)). More reliable however, is the lack of a strong resolution dependence of the fast contact plane ejecta launched from the `AF1`, `AF2`, and `AF3` simulations.

If this trend persists at higher resolution, it would mean that grid-based simulations have already converged in the amount of fast ejecta to order of magnitude, with the tension with SPH models being due to a systematic difference arising from the differing numerical methods.

Chapter 5

Conclusions

Motivated by the tension between the fast ejecta mass produced in published SPH and grid-based binary neutron star merger simulations, we investigate the resolution dependence of fast ejecta mass in grid-based models. This fast ejecta is thought to be made mostly of free-neutrons, which at sufficient velocity ($v > 0.6c$) freeze out the r-process, and can power an ultra-violet precursor to the kilonova. This precursor would peak on a timescale of hours after the merger, and would encode useful information that could be used to further constrain the neutron star equation of state and electromagnetic emission mechanisms. The question of whether or not this ultra-violet precursor is observable for a merger at a given distance largely depends on the mass of free-neutrons ejected, thus understanding the relationship between neutron star compactness and fast ejecta mass is crucial to understanding the observability of this precursor, as well as extracting useful information about the binary neutron star system.

Due to the extreme density and pressure gradient present near the surface of a neutron star, and fast ejecta largely being launched from the surface of the individual neutron stars, it is important to ensure the surface of the neutron stars are sufficiently resolved when tracking this fast ejecta with grid-based hydrodynamic methods. Previous numerical relativity simulations in three dimensions have reached a spatial resolution of $63 - 86$ m cells (Kiuchi et al., 2017). We argue however, that in order to resolve the pressure scale height out to 99% the radius of the neutron star, cells of size ~ 4 m would be required, therefore the under-representation of fast ejecta may be due to under-resolving the neutron star surface. In order to achieve such a resolution however (which is currently unfeasible in three dimensions), we perform merger simulations in two dimensional cylindrical coordinates, using the `FLASH` code. We use Newtonian hydrodynamics, and other microphysical simplifications such as a piece-wise polytropic equation of state, and approximate thermal effects. In this thesis, I outline the simulation set-up developed, micro-physics implemented into `FLASH`, as well as how post-processing of simulation data was done, and how the efficiency of the simulation was improved.

We present preliminary results of our study with resolutions competitive with the highest resolutions in existing numerical relativity simulations run to date. Given the approximations introduced to reduce the dimensionality, we vary the nature of the key physics driving the binary evolution (inertial forces and gravitational wave emission) to bracket the uncertainty introduced by these simplifications. We find only a weak resolution dependence of fast ejecta in the contact plane, with a somewhat stronger trend for orbital plane ejecta, with an increase on the order of $\sim 10^{-6}M_{\odot}$ from simulation resolution $\Delta x_1 = 281.3$ m to $\Delta x_3 = 70.3$ m. This weak resolution dependence suggests that we may be close to converging on the mass ejected, however higher resolution simulations are desired to ensure this result.

We also consider the scenarios of a head-on collision between the neutron stars, motivated by previous works, and a merger with initial velocity set by the gravitational wave emission orbital seperation decay rate of the binary system, neglecting inertial forces and explicit gravitational wave emission. As our simulation uses a variety of microphysical simplifications, and only Newtonian physics, these additional scenarios are intended to bracket the true physical behaviour of the merging system by varying the collisional velocity, and thus ejecta kinematics. We find that the mass of fast ejecta from our simulations fall between $M_{ej,v>0.6c} \sim 10^{-5} - 10^{-2} M_{\odot}$ (Table 3.1). These values are consistent with the range of fast ejecta masses obtained in both grid based simulations ($M_{fn} \sim 10^{-6} - 10^{-5} M_{\odot}$ Radice et al. (2018b)) and SPH simulations ($M_{fn} \sim 10^{-4} M_{\odot}$ Metzger et al. (2015)).

Therefore, while higher resolution simulations are still required in this study, the lack of a strong resolution dependence of contact plane ejecta suggests we may be close to convergence on the fast ejecta from the system. We do not yet have strong evidence to resolve the tension between the fast ejecta masses produced in SPH and grid-based merger simulations. But if this discrepancy persists at higher resolutions, we will be able to conclude that the discrepancy in ejecta masses can be attributed to systemic differences in the numerical methods.

Bibliography

B. P. Abbott et al. Observation of Gravitational Waves from a Binary Black Hole Merger. *PhRvL*, 116(6):061102, feb 2016. doi: 10.1103/PhysRevLett.116.061102.

B. P. Abbott et al. GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral. *PhRvL*, 119(16):161101, October 2017a. doi: 10.1103/PhysRevLett.119.161101.

B. P. Abbott et al. Multi-messenger Observations of a Binary Neutron Star Merger. *ApJL*, 848(2):L12, October 2017b. doi: 10.3847/2041-8213/aa91c9.

B. P. Abbott et al. GW190425: Observation of a Compact Binary Coalescence with Total Mass $\sim$ 3.4 M$_\odot$. *ApJL*, 892(1):L3, March 2020. doi: 10.3847/2041-8213/ab75f5.

A. Akmal, V. R. Pandharipande, and D. G. Ravenhall. Equation of state of nucleon matter and neutron star structure. *PhRvC*, 58(3):1804–1828, September 1998. doi: 10.1103/PhysRevC.58.1804.

Iair Arcavi. The First Hours of the GW170817 Kilonova and the Importance of Early Optical and Ultraviolet Observations for Constraining Emission Models. *ApJ*, 855(2):L23, Mar 2018. doi: 10.3847/2041-8213/aab267.

A. Bauswein, H.-T. Janka, and R. Oechslin. Testing approximations of thermal effects in neutron star merger simulations. *PRD*, 82(8):084043, October 2010. doi: 10.1103/PhysRevD.82.084043.

A. Bauswein, S. Goriely, and H. T. Janka. Systematics of Dynamical Mass Ejection, Nucleosynthesis, and Radioactively Powered Electromagnetic Signals from Neutron-star Mergers. *ApJ*, 773(1):78, Aug 2013. doi: 10.1088/0004-637X/773/1/78.

Hank Childs et al. VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data. In *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*, pages 357–372. Oct 2012.

P. Colella and P. R. Woodward. The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations. *JCP*, 54:174–201, September 1984.

Sean M. Couch, Carlo Graziani, and Norbert Flocke. An Improved Multipole Approximation for Self-gravity and Its Importance for Core-collapse Supernova Simulations. *ApJ*, 778(2):181, Dec 2013. doi: 10.1088/0004-637X/778/2/181.

John J. Cowan, Christopher Sneden, James E. Lawler, Ani Aprahamian, Michael Wiescher, Karlheinz Langanke, Gabriel Martnez-Pinedo, and Friedrich-Karl Thielemann. Origin of the heaviest elements: the rapid neutron-capture process, 2019.

F. Douchin and P. Haensel. A unified equation of state of dense matter and neutron star structure. *A&A*, 380:151–167, December 2001. doi: 10.1051/0004-6361:20011402.

Anshu Dubey, Katie Antypas, Murali K. Ganapathy, Lynn B. Reid, Katherine Riley, Dan Sheeler, Andrew Siegel, and Klaus Weide. Extensible component-based architecture for flash, a massively parallel, multiphysics simulation code. *J. Par. Comp.*, 35(10-11):512 – 522, 2009. ISSN 0167-8191. doi: DOI:10.1016/j.parco.2009.08.001.

David Eichler, Mario Livio, Tsvi Piran, and David N. Schramm. Nucleosynthesis, neutrino bursts and γ-rays from coalescing neutron stars. *Nature*, 340 (6229):126–128, July 1989. doi: 10.1038/340126a0.

Alexander L. Fetter and John Dirk Walecka. *Accelerated Coordinate Systems*, chapter 2. Dover Publications, 2003.

B. Fryxell, K. Olson, P. Ricker, F. X. Timmes, M. Zingale, D. Q. Lamb, P. MacNeice, R. Rosner, J. W. Truran, and H. Tufo. FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes. *ApJS*, 131:273–334, November 2000. doi: 10.1086/317361.

S. Goriely, A. Bauswein, H. T. Janka, J. L. Sida, J. F. Lemaître, S. Panebianco, N. Dubray, and S. Hilaire. The r-process nucleosynthesis during the decompression of neutron star crust material. In Sunchan Jeong, Nobuaki Imai, Hiroari Miyatake, and Toshitaka Kajino, editors, *American Institute of Physics Conference Series*, volume 1594 of *American Institute of Physics Conference Series*, pages 357–364, May 2014. doi: 10.1063/1.4874095.

A. Hewish, J. D. Bell, J. D. H. Pilkington, P. F. Scott, and R. A. Collins. Observation of a Rapidly Pulsating Radio Source. *Nature*, 217:709–713, Feb 1968. doi: 10.1038/217709a0.

R. A. Hulse and J. H. Taylor. Discovery of a pulsar in a binary system. *ApJL*, 195:L51–L53, January 1975. doi: 10.1086/181708.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

Ayako Ishii, Toshikazu Shigeyama, and Masaomi Tanaka. Free Neutron Ejection from Shock Breakout in Binary Neutron Star Mergers. *ApJ*, 861(1):25, Jul 2018. doi: 10.3847/1538-4357/aac385.

Hannu Karttunen, Pekka Kröger, Heikki Oja, Markku Poutanen, and Karl Johan Donner. *Fundamental Astronomy.* 2017. doi: 10.1007/978-3-662-53045-0.

Kenta Kiuchi, Kyohei Kawaguchi, Koutarou Kyutoku, Yuichiro Sekiguchi, Masaru Shibata, and Keisuke Taniguchi. Sub-radian-accuracy gravitational waveforms of coalescing binary neutron stars in numerical relativity. *PRD*, 96(8):084060, Oct 2017. doi: 10.1103/PhysRevD.96.084060.

S. R. Kulkarni. Modeling Supernova-like Explosions Associated with Gamma-ray Bursts with Short Durations. *preprint, astro-ph/0510256*, art. astro-ph/0510256, Oct 2005.

Koutarou Kyutoku, Kunihito Ioka, and Masaru Shibata. Ultrarelativistic electromagnetic counterpart to binary neutron star mergers. *MNRAS*, 437(1): L6–L10, Jan 2014. doi: 10.1093/mnrasl/slt128.

Benjamin D. Lackey, Mohit Nayyar, and Benjamin J. Owen. Observational constraints on hyperons in neutron stars. *PhRvD*, 73(2):024021, January 2006. doi: 10.1103/PhysRevD.73.024021.

J. M. Lattimer and D. N. Schramm. Black-Hole-Neutron-Star Collisions. *ApJL*, 192:L145, September 1974. doi: 10.1086/181612.

Li-Xin Li and Bohdan Paczyński. Transient Events from Neutron Star Mergers. *ApJL*, 507(1):L59–L62, November 1998. doi: 10.1086/311680.

B. D. Metzger, G. Martínez-Pinedo, S. Darbha, E. Quataert, A. Arcones, D. Kasen, R. Thomas, P. Nugent, I. V. Panov, and N. T. Zinner. Electromagnetic counterparts of compact object mergers powered by the radioactive decay of r-process nuclei. *MNRAS*, 406(4):2650–2662, August 2010. doi: 10.1111/j.1365-2966.2010.16864.x.

Brian D. Metzger, Andreas Bauswein, Stephane Goriely, and Daniel Kasen. Neutron-powered precursors of kilonovae. *MNRAS*, 446(1):1115–1120, Jan 2015. doi: 10.1093/mnras/stu2225.

Brian D. Metzger, Todd A. Thompson, and Eliot Quataert. A Magnetar Origin for the Kilonova Ejecta in GW170817. *ApJ*, 856(2):101, April 2018. doi: 10.3847/1538-4357/aab095.

M. C. Miller, F. K. Lamb, A. J. Dittmann, S. Bogdanov, Z. Arzoumanian, K. C. Gendreau, S. Guillot, A. K. Harding, W. C. G. Ho, J. M. Lattimer, R. M. Ludlam, S. Mahmoodifar, S. M. Morsink, P. S. Ray, T. E. Strohmayer, K. S. Wood, T. Enoto, R. Foster, T. Okajima, G. Prigozhin, and Y. Soong. PSR J0030+0451 Mass and Radius from NICER Data and Implications for the Properties of Neutron Star Matter. *ApJL*, 887(1):L24, December 2019. doi: 10.3847/2041-8213/ab50c5.

B. Paczynski. Gamma-ray bursters at cosmological distances. *ApJL*, 308:L43–L46, September 1986. doi: 10.1086/184740.

T. Padmanabhan. *Gravitation: Foundations and Frontiers.* Cambridge University Press, 2010.

Vasileios Paschalidis, Zachariah B. Etienne, and Stuart L. Shapiro. Importance of cooling in triggering the collapse of hypermassive neutron stars. *PRD*, 86 (6):064032, Sep 2012. doi: 10.1103/PhysRevD.86.064032.

Anthony L. Piro and Juna A. Kollmeier. Evidence for Cocoon Emission from the Early Light Curve of SSS17a. *ApJ*, 855(2):103, Mar 2018. doi: 10.3847/1538-4357/aaaab3.

David H. Porter and Paul R. Woodward. High-Resolution Simulations of Compressible Convection Using the Piecewise-parabolic Method. *ApJS*, 93:309, Jul 1994. doi: 10.1086/192057.

David Radice, Albino Perego, Kenta Hotokezaka, Sebastiano Bernuzzi, Steven A. Fromm, and Luke F. Roberts. Viscous-dynamical Ejecta from Binary Neutron Star Mergers. *ApJ*, 869(2):L35, Dec 2018a. doi: 10.3847/2041-8213/aaf053.

David Radice, Albino Perego, Kenta Hotokezaka, Steven A. Fromm, Sebastiano Bernuzzi, and Luke F. Roberts. Binary Neutron Star Mergers: Mass Ejection, Electromagnetic Counterparts, and Nucleosynthesis. *ApJ*, 869(2):130, Dec 2018b. doi: 10.3847/1538-4357/aaf054.

J. S. Read, B. D. Lackey, B. J. Owen, and J. L. Friedman. Constraints on a phenomenologically parametrized neutron-star equation of state. *PRD*, 79 (12):124032, June 2009. doi: 10.1103/PhysRevD.79.124032.

T. E. Riley, A. L. Watts, S. Bogdanov, P. S. Ray, R. M. Ludlam, S. Guillot, Z. Arzoumanian, C. L. Baker, A. V. Bilous, D. Chakrabarty, K. C. Gendreau, A. K. Harding, W. C. G. Ho, J. M. Lattimer, S. M. Morsink, and T. E. Strohmayer. A NICER View of PSR J0030+0451: Millisecond Pulsar Parameter Estimation. *ApJL*, 887(1):L21, December 2019. doi: 10.3847/2041-8213/ab481c.

Yuichiro Sekiguchi, Kenta Kiuchi, Koutarou Kyutoku, and Masaru Shibata. Gravitational Waves and Neutrino Emission from the Merger of Binary Neutron Stars. *PRL*, 107(5):051102, Jul 2011. doi: 10.1103/PhysRevLett.107.051102.

Stuart L. Shapiro and Saul A. Teukolsky. *Black Holes, White Dwarfs, and Neutron Stars: The Physics of Compact Objects*. Wiley-Interscience, 1983.

M. J. Turk, B. D. Smith, J. S. Oishi, S. Skory, S. W. Skillman, T. Abel, and M. L. Norman. yt: A Multi-code Analysis Toolkit for Astrophysical Simulation Data. *The Astrophysical Journal Supplement Series*, 192:9, January 2011. doi: 10.1088/0067-0049/192/1/9.

Appendix A

Code Optimization

A.1 Adaptive Mesh Refinement

The need to resolve the pressure scale height at the Neutron Star surface, while requiring a much lower resolution to resolve the dynamics of ejected material, at larger distances motivates the use of FLASH's Adaptive Mesh Refinement (AMR) capability. FLASH implements AMR via the PARAMESH library (MacNeice et al 1999). The use of AMR allows us to resolve the strongest gradients of density (or another selected variable), such as at the surface of the Neutron Star, with the finest resolution cells within the domain, while shallower density gradients can be evolved using lower resolution (Figure A.2, left pane). This lightens the work load of processors as the total number of cells to evolve is decreased relative to a Uniform Grid. AMR maintains conservation by matching fluxes (of mass, momentum, and energy) at AMR boundaries (Figure A.1). Fluxes crossing an AMR boundary from a finer resolution to a coarser resolution are added together when advecting into the neighbouring coarser cell.

AMR also allows for a time varying grid structure that is updated based on the location of the steepest density gradients throughout the material as it moves throughout the domain, ensuring that the highest resolution is employed where needed.

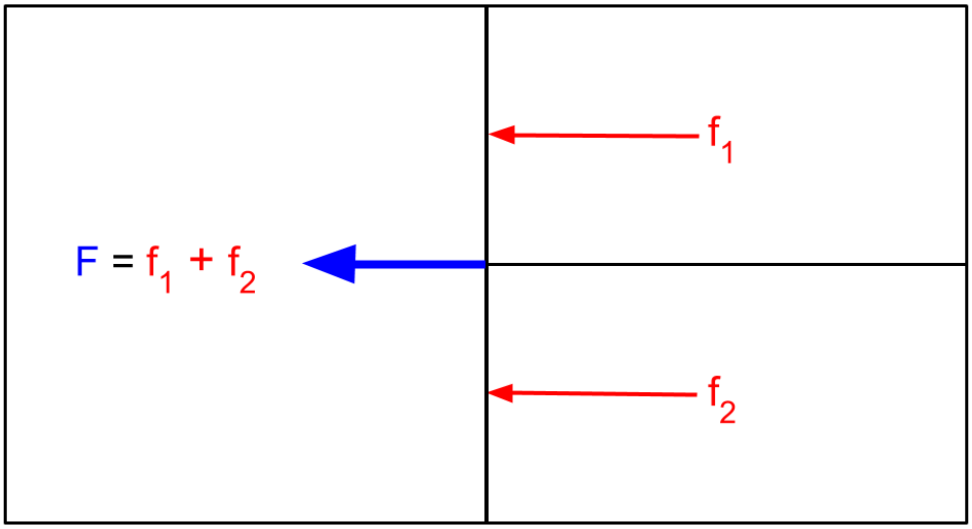

Figure A.1: Schematic of flux conservation at an AMR boundary. Flux into the coarser cell F is given by the sum of the fluxes exiting the two finer cells f_i. Figure adapted from Fryxell et al. (2000)

In testing the use of AMR for our setup, we encountered problems with the method. In particular, strong gradients in density such as the leading edge of the expanding ejecta meeting infalling ambient material would in some cases result in non-convergence of the Riemann solver. This problem was exacerbated as these strong gradients crossed AMR boundaries. In an attempt to subdue this issue, I reduced the more complex AMR grid structure (Figure A.2, left pane), with the limiting case of Static Mesh Refinement (SMR), preventing the grid structure from changing with time, thus ensuring the material ejected from the system would only have to cross the refinement boundaries once. I also limited the refinement structure to follow a simpler rectangular shape (Figure A.2, right pane). At the cost of a slight loss of efficiency, as more cells must be evolved, using this simple grid structure reduces the number of refinement boundaries that ejected material must cross as it advects out of the domain.

While tracking mass ejected from the merging binary system in post-processing, we found that AMR and SMR introduce another problem due to how grid data is stored. As an AMR grid is non-uniform, data is stored in a one dimensional

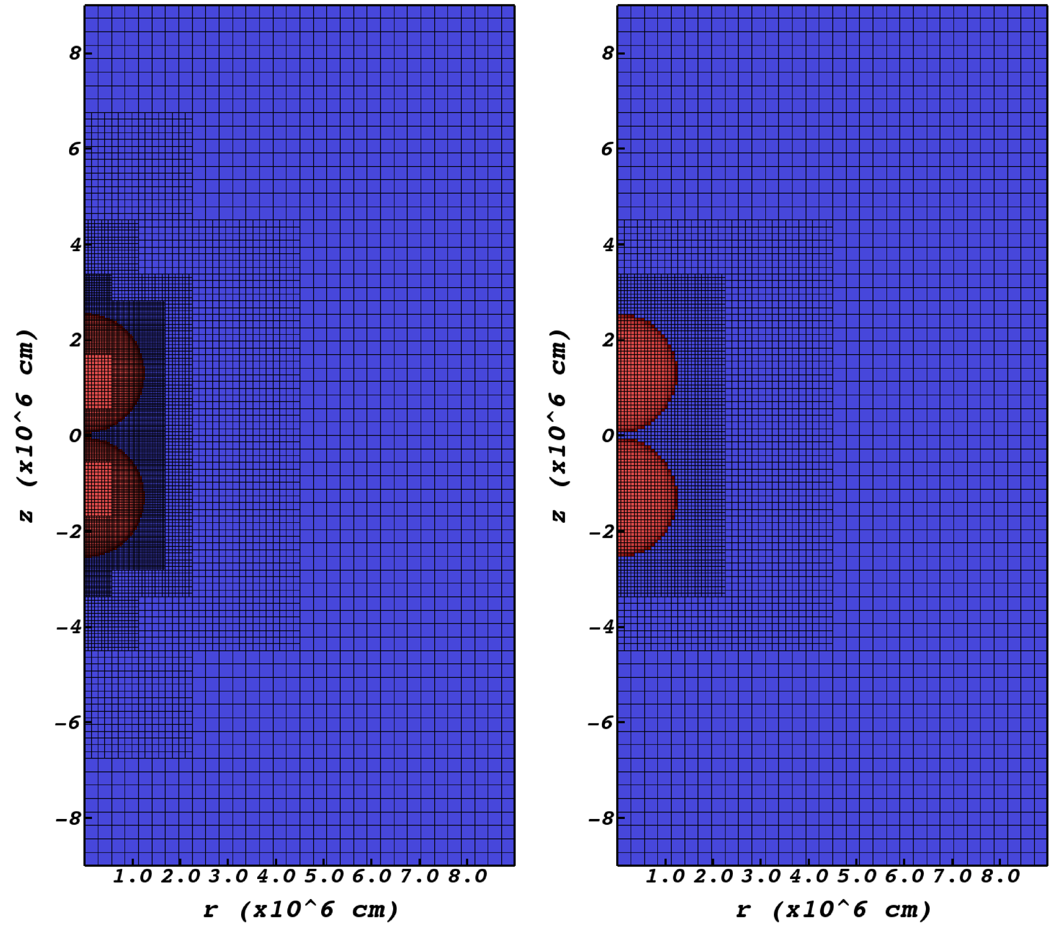

Figure A.2: Left: Initial Adaptive Mesh Refinement grid structure overlaying a density colour map plot. Here density varies from the highest density (red) corresponding to $\sim 7 \times 10^{14}$ g/cm^3 at the center of the Neutron Stars, to the ambient value (blue) corresponding to 10^4 g/cm^3. Right: Same as Left but now showing a rectangular Static Mesh Refinement grid structure.

sequence defined by a Morton Curve, which is a curve defining the order of cells such that a two or three dimensional AMR grid may be flattened into a one dimensional data vector (See Figure A.3). This way of storing the data complicates mass tracking of the ejecta. The simplest solution to this problem is in post processing to convert the AMR data to a uniform grid data structure of the highest resolution present in the AMR grid. This can be done with the "Covering Grid" functionality within the python package `yt` (Turk et al., 2011). This algorithm covers a non-uniform grid structure with a uniform grid of specified resolution. This Covering Grid is set to the maximum resolution within the AMR structure, extracting an equivalent uniform grid data structure.

Extracting the AMR data in this way revealed a few problems: first being that this extraction process is computationally expensive at even a fraction of the finest resolution targeted in this research. Extracting data in this way is also problematic, as the data at the highest resolution even for just one physical variable, is larger than available memory. A double precision number used to store each variable in each cell at the highest resolution would require:

$$\frac{Memory}{Variable} = \frac{8\ bytes}{variable \cdot cell} \times 22,500\ xcells \times 45,000\ ycells = 8.1\ GB/variable \tag{A.1}$$

Thus with `FLASH` outputting 35 variables in its current state, at full resolution on the order of 100 GB of memory would be required to simply load the data for the entire domain at full resolution. As the python module `yt` is built in such a way that customization of the loading routine `covering grid` would involve modification of the source code, I decided to write my own parallelised python module consisting of functions that extract the data from the `FLASH` output HDF5 file, recursively interpreting the cell neighbour, parent, and child cell information to map the solution data onto a uniform grid. Writing this script from scratch allowed it to be run in parallel, to reduce the overall computation time necessary to convert a series of data files, as well as see the inner mechanics of the loading method. This allowed the domain to be loaded in a piece-by-piece manner, loading a fraction of the domain to memory, converting to a uniform grid, then dumping the data to the hard drive and clearing the memory before loading another section of the domain thus avoiding the larger than memory issue.

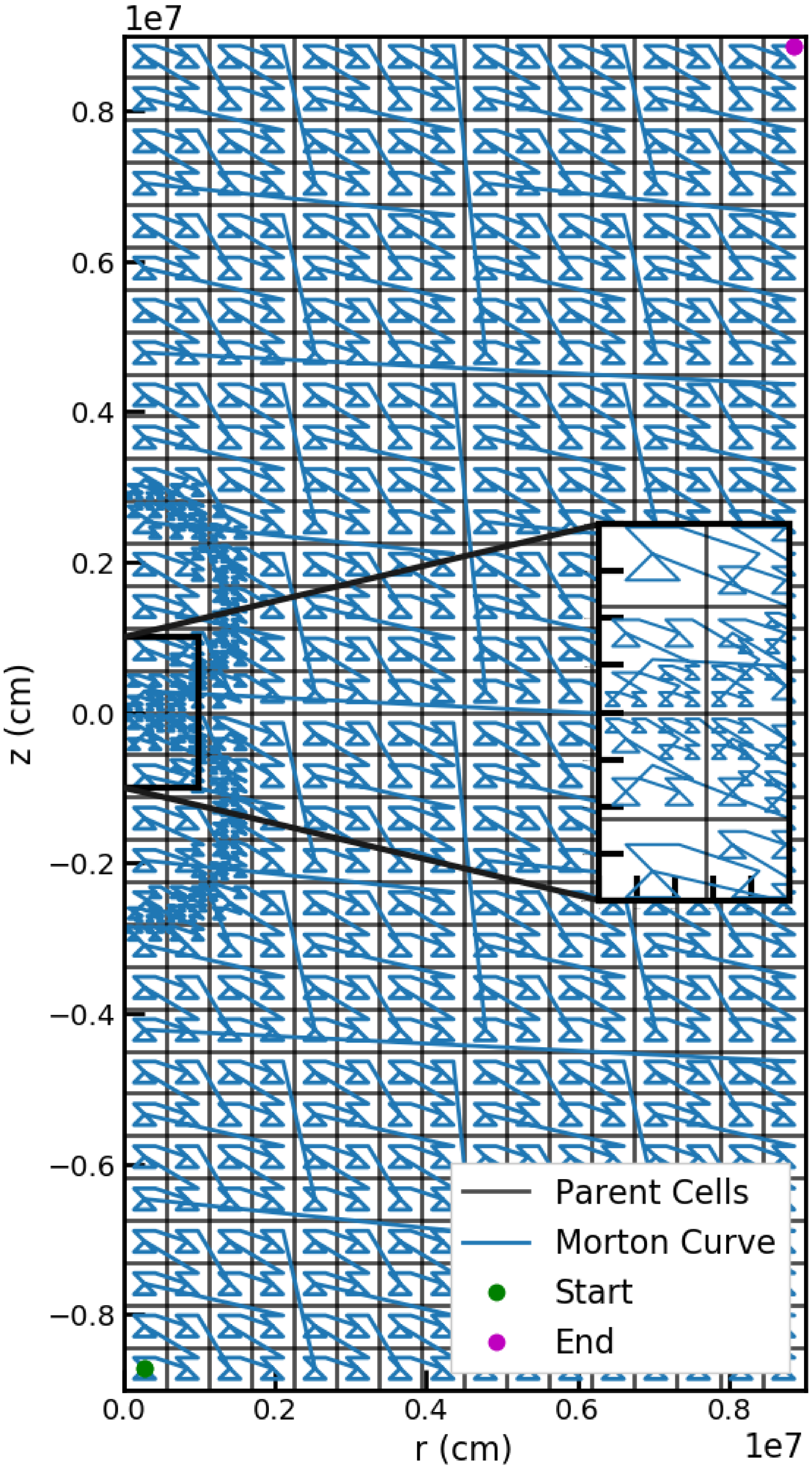

Figure A.3: Initial Morton Curve for the merger simulation using an Adaptive Mesh Refinement grid structure.

A.2 Uniform Grid

Weighing the benefits and costs of running the simulations using the simplified SMR grid structure, given that the post processing was a significant computational drain that must be run after a simulation is finished, in addition to the issues introduced by strong density gradients crossing AMR boundaries, I decided to use the simpler method of running using a Uniform Grid structure, as the benefits to the stability of the Riemann solver, and removal of the need to post process the data, and restructure to a Uniform Grid outweighed the additional computational cost. The final grid structure used in the simulation was a uniform grid structure in two dimensional cylindrical geometry (ρ,z) as shown in Figure A.4.

A.3 Power-law Pressure Floor

By default FLASH sets a floor which prevents the pressure of simulated material from becoming negative and thereby crashing the hydrodynamic solver. This is implemented in the form:

$$P = max(P, P_{small}), \tag{A.2}$$

where P_{small} is the value of the pressure floor parameter, to be specified by the user. However, the default implementation only allows for a single floor value to be applied across the entire domain. In the case where the ambient material is set to have density and pressure that follow a power-law with radius, so that the total mass of the ambient material remains low while reducing the pressure and density jump between the edge of the Neutron Stars and the ambient material, a power-law pressure floor that more closely tracks the initial ambient is desired. Similarly this is also useful near the stars when the ejecta first begins to expand, with a gradually decreasing floor as the material expands

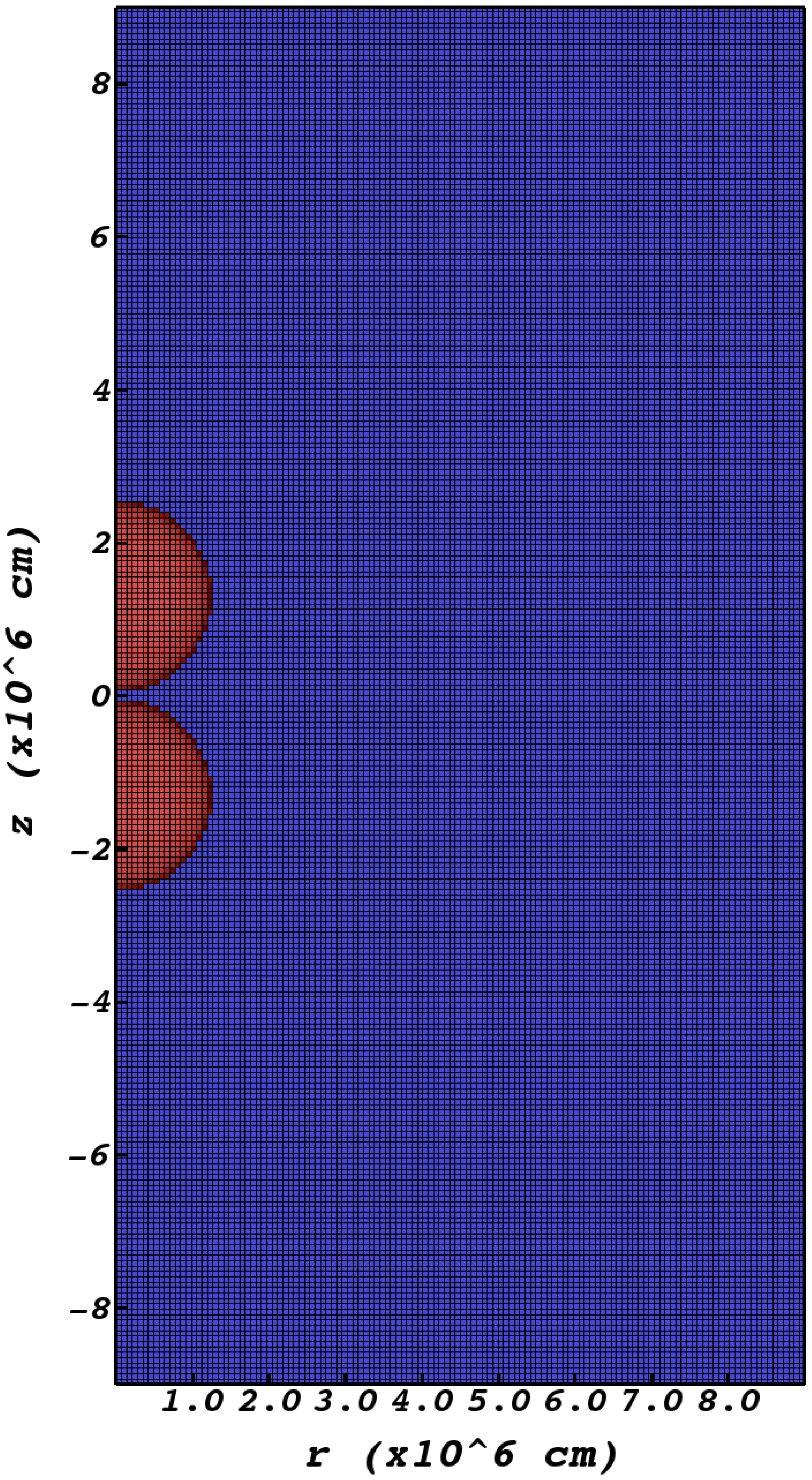

Figure A.4: Same as Figure A.2, but now showing a Uniform Grid structure.

outward. To address these issues, I implemented a Power-law pressure floor of the form:

$$P_{floor}(r) = \begin{cases} P_{small}\left(\frac{R_{trans}}{R_{start}}\right)^{k_{in}} & r \leq R_{start} \\ P_{small}\left(\frac{R_{trans}}{r}\right)^{k_{in}} & R_{start} < r \leq R_{trans} \\ P_{small}\left(\frac{R_{trans}}{r}\right)^{k_{out}} & r > R_{trans} \end{cases} \tag{A.3}$$

where P_{small} is the pressure floor value at the transition radius R_{trans}, $r = \sqrt{\rho^2 + z^2}$ is the spherical radius, k_{in}, and k_{out} are the inner and outer power-law slopes of the broken power-law, and R_{start} is the radius within which the pressure floor is flat, avoiding the divergence of a power-law at $r = 0$. While these parameters can be set manually, the additions to the code also allow the parameters to be set automatically such that the pressure floor remains a constant factor below ambient pressure if it is also defined as a power-law.

A.4 Computational Efficiency of Initialization

Another issue exposed by running simulations at extremely high resolution was the inefficiency of the initialization process. This included loading in the Neutron Star profile, interpolating from that profile to define the pressure and density throughout the Neutron Stars in the computational domain, defining the pressure and density for the cells filled with ambient material, calling upon the Equation of state to fill in other thermodynamic variables, and defining the initial kinematics of the material. Despite the initialization being parallelised by default in `FLASH`, each processor being responsible for initializing its own block of cells, the initialization at very high resolutions was of considerable computational cost in comparison to the cost of evolving the hydrodynamic variables throughout the simulation. In particular, initialization of the simulation was taking on the order of hours to tens of hours at full resolution. Thus I decided to spend some time increasing the efficiency of the initialization method. Ultimately, a line of legacy code no longer used was responsible for most of the slowdown in the initialization at high resolution. Nevertheless, prior to uncovering this I investigated several modifications to increase the initialization efficiency, which on their own resulted in a reduction of initialization time by 25%. Below I describe these additional modifications.

A.4.1 Profile Interpolation

As discussed in Section 3.1.3, to define the density and pressure of cells within the Neutron Stars, we interpolate from a density/pressure profile of the Neutron Star. This involves looping through the profile for each cell within either of the stars, to determine the nearest radial distance included in the profile, then interpolating to the cells radial distance. While this is a straight-forward approach, it is not particularly efficient.

Some efficiency improvements can be made using block information. The entire domain is subdivided into blocks each of which is initialized and evolved by a particular processor. Use of the block limits to determine the nearest point along the blocks perimeter to the star center, allows the interpolation index search to be truncated by a minimum search index. Similarly, once the nearest index has been found, an exit statement can be used to truncate the index search on the back end. This approach avoids the search over the entire profile for each grid point inside the star

A.4.2 Piece-wise Polynomial Fit Profile

In an attempt to remove the index search entirely, I implemented an alternative piece-wise polynomial fit to the Neutron Star profile to further increase the efficiency of the initialization. As discussed in Section 3.1.1, the simulation runs using a Piece-wise Polytropic Equation of state, a method used in Read et al. (2009) to reproduce a variety of different Neutron Star equations of state with a four piece piece-wise polytrope (with a fixed Sly EOS crust) that can be completely described via four parameters (P_1,Γ_1,Γ_2,Γ_3). P_1 is the dividing pressure between sections 1 and 2, and Γ_i is the polytropic index of section i. Thus the Neutron Star profile discussed in Section 3.1.3 acts as a continuous density/pressure curve with respect to radius with a discontinuous derivative corresponding to shifts between sections of the polytropic EOS. In order to remove the need for an index search we can simply fit the profile with a piecewise polynomial, with divisions corresponding to the density division between EOS pieces. Thus given the cells spherical radial position with respect to the initial center of the Neutron Star we can simply compute the density and pressure values rather than interpolating from the full profile. The results of fitting a 6th order polynomial to each of the four sections of the `APR4` EOS can be seen below for density in Figure A.5, and for pressure in Figure A.6. This fit reproduces the profile well, limited in accuracy only near the surface where the fractional difference is largest. Even in this region, the profile is reproduced to better than 1% accuracy, which is sufficient given the limitations introduced by other assumptions made for this simulation.

A.4.3 Profile Loading

Another area where efficiency could be gained while initializing the simulation is loading the profile to memory. In the case of using the full profile, initially the file was read in by one processor, before being broadcast to the remaining processors. While efficient in terms of work done by the processors, this takes additional time as the processors must sit idle while waiting for the primary processor to finish reading the profile before broadcasting can occur. In this case we can reduce the total amount of initialization by reading the profile in individually by each processor, thus eliminating the need for broadcasting.

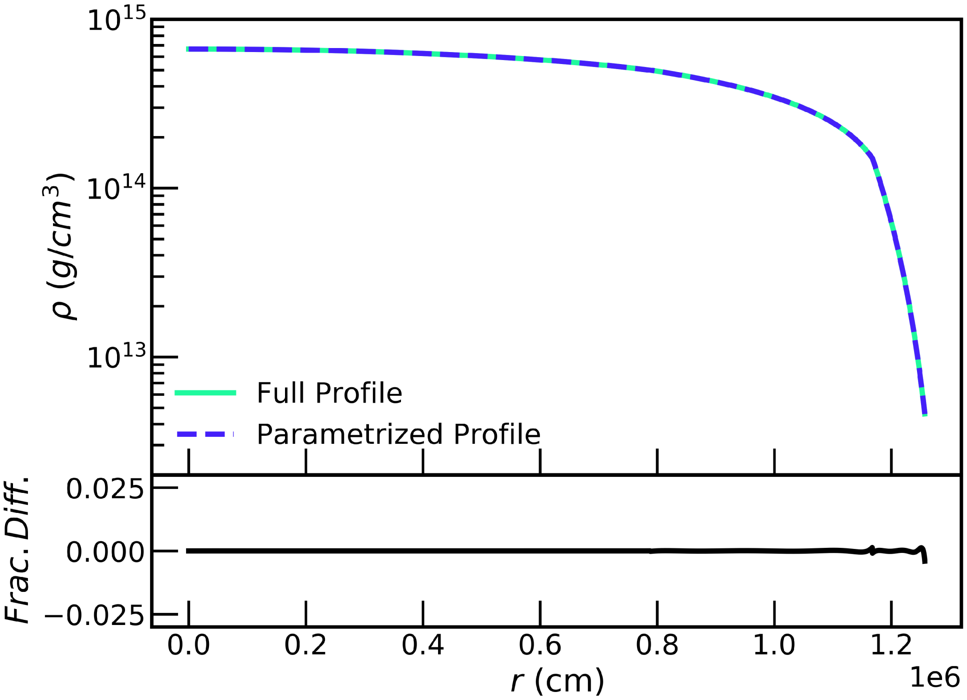

Figure A.5: Top: Density profile of a 1.4 $M_\odot$ Neutron Star constructed with the APR4 EOS. The solid line represents the full profile, and the dashed line represents the parameterized fit profile. Bottom: Fractional difference between the two curves in the top pane.

A.4.4 Piece-wise Polytropic Equation of State Sorting

The EOS subroutine in FLASH is called slice-by-slice, as the directionally split hydrodynamic solver sweeps through the domain in a particular direction, data is fed into the equation of state as an array representing the cells in a row parallel to the sweep direction. Thus the EOS subroutine is passed an array of cells for which the equation of state is used to calculate the remaining hydrodynamic variables at each time-step after density and internal energy have been evolved in time. As the density of material in a given cell must first be sorted into the correct section of the piece-wise polytropic equation of state, the initial method of looping through this array to sort by density can be improved in efficiency by removing the for loop. This can be achieved by using the Fortran MERGE function. This allows an array of indexes to be returned based on a mask. If a

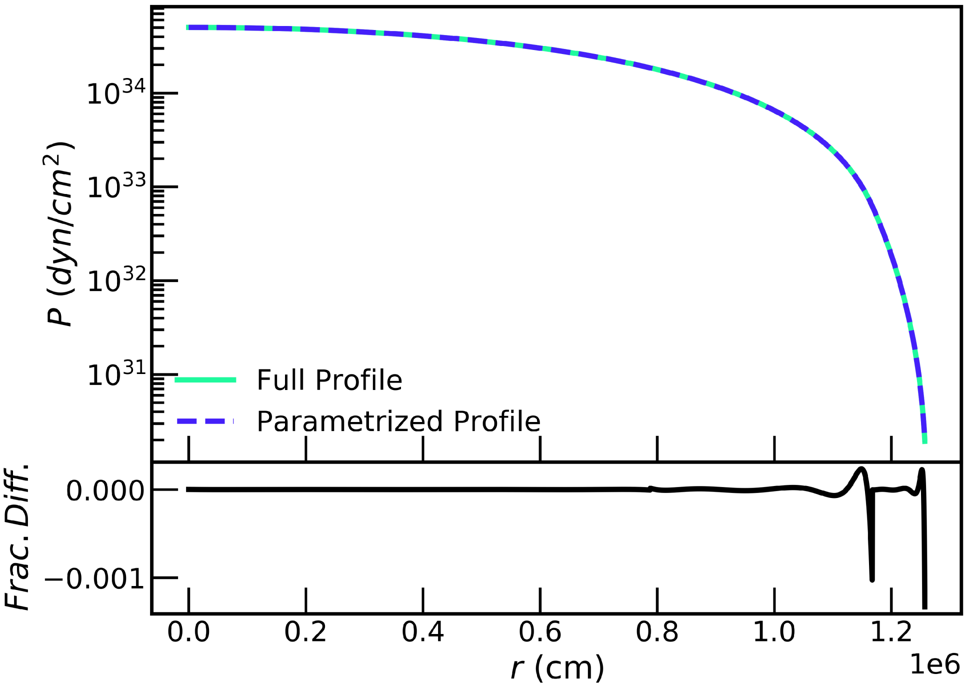

Figure A.6: Pressure profile and parameterized fit for the same star as in Figure A.5.

mask is made for each section of the piece-wise polytrope, the `MERGE` function can be used for each, to return an array of indexes indicating which section of the piece-wise polytrope to use. The removal of this `for` loop used for sorting each array is a significant efficiency improvement as the EOS is called multiple times per time-step for each row of cells, thus adding up to a significant efficiency gain in high resolution simulations run for many time-steps.