BOWEN-TYPE INITIAL DATA FOR SIMULATIONS OF NEUTRON STARS IN BINARY SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

Michael Christopher Clark

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Physics

Georgia Institute of Technology
May 2016

BOWEN-TYPE INITIAL DATA FOR SIMULATIONS OF NEUTRON STARS IN BINARY SYSTEMS

Approved by:

John Wise, Committee Chair
School of Physics
Georgia Institute of Technology

Pablo Laguna, Advisor
School of Physics
Georgia Institute of Technology

Laura Cadonati
School of Physics
Georgia Institute of Technology

Deirdre Shoemaker
School of Physics
Georgia Institute of Technology

David Bader
School of Computational Science and Engineering
Georgia Institute of Technology

Date Approved: March 31, 2016

To my parents,

John and Kimberly,

who have supported me unceasingly and with love.

ACKNOWLEDGEMENTS

My thanks to my advisor, Pablo Laguna, for his advice and guidance, and to Tanja Bode, whose efforts to train me while she was a postdoctoral fellow substantially shaped my approach to computational science and physics. I also give thanks to my colleagues, Katharaine Tallaksen and Matt Kinsey, who have helped make many days tolerable through their willingness to share frustrations and listen.

I thank my friends distant and near: Victoria and Justin Heeney, Stacey Criswell, Sue Mackey, Patricia Adams, Michelle Rhodes, Dawn Trapp, and many others. You all have helped me have something to look forward to beyond my work, and for that, I am truly thankful.

And of course, I thank my parents, John and Kimberly, for their coninued support. No man could ask for two more dedicated people in his corner.

The work presented here was supposed by NSF grants 1205864, 1212433, and 1333360. Computations were performed through XSEDE allocation TG-PHY120016 on the Stampede cluster. Additional computations were performed on the Cygnus cluster at Georgia Tech.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

${}^{(3)}R_{\alpha\beta\mu\nu}$	Intrinsic Riemann tensor for a spatial hypersurface.
α	Lapse function.
$\bar{A}_{ij}$	Conformal traceless extrinsic curvature.
β^{μ}	Shift vector.
c	Speed of light.
D_{μ}	Covariant derivative.
$\eta_{\mu\nu}$	Flat spacetime metric.
$g_{\mu\nu}$	Spacetime metric.
G	Newton's gravitational constant.
$\gamma_{\mu\nu}$	Intrinsic metric for a spatial hypersurface; "spatial metric".
γ_{ij}	Conformal spatial metric.
$h_{\mu\nu}$	Spacetime metric perturbation.
h_{+}	"Plus" polarization of metric perturbation (gravitational wave).
h	Enthalpy.
$h_{\times}$	"Cross" polarization of metric perturbation (gravitational wave).
$K_{\mu\nu}$	Extrinsic curvature.
$M_{\odot}$	Solar mass.
n^{μ}	Normal vector to a hypersurface.
p	Pressure.
ψ	Conformal factor.
Ψ_4	Fourth Weyl scalar, related to second time derivatives of h_{+} and $h_{\times}$.
$R_{\mu\nu}$	Ricci tensor.
R	Ricci scalar (or "scalar curvature").
$R_{\alpha\beta\gamma\delta}$	Riemann curvature tensor.
ρ	Energy density.

ρ_0	Rest mass density.
S^μ	Momentum density.
$T_{\mu\nu}$	Stress-energy tensor.
W	Lorentz factor.

SUMMARY

A new method for generating initial data for simulations of neutron stars in binary systems. The construction of physically relevant initial data is crucial to accurate assessment of gravitational wave signals relative to theoretical predictions. This method builds upon the Bowen-York curvature for puncture black holes. This data is evolved and compared against simulations in the literature with respect to orbital eccentricity, merger and collapse times, and emitted energy and angular momentum. The data exhibits some defects, including large central density oscillations in stars and center of mass drift in unequal-mass systems. Some approaches for improvements in potential future work are discussed.

CHAPTER I

INTRODUCTION

Gravitational waves are now observational reality. Having been predicted by Einstein in 1916 [29], gravitational waves were indirectly observed by Hulse and Taylor in 1974 [34] [58]. They discovered a binary pulsar and Taylor, along with Fowler and McCulloch, verified it had a gradually contracting orbit, consistent with the energy losses predicted to escape as gravitational waves, and for this discovery, Hulse and Taylor were awarded the 1993 Nobel Prize in Physics, but direct proof of gravitational waves remained elusive until the Laser Interferometer Graviataional-Wave Observatory (LIGO) detected a coalescing binary black hole system on September 14, 2015 [1].

Key to this discovery was the role of numerical relativity—the use of computer simulations and algorithms to solving Einstein's equations—to developing templates of binary black hole systems' graviataional wave emissions. A thorough bank of templates allowed the LIGO collaborators to match the observed signal to that of a binary black hole simulation between two objects of masses 36 and 29 solar masses ($M_\odot$), as well as to identify the black holes' spins, the final merged black hole spin, and the total amount of energy emitted [59].

In the future, LIGO may detect binary systems involving neutron stars. Templates for these systems will prove essential to distinguishing black hole binaries from those with neutron stars, as well as to discern the properties of those neutron stars.

This thesis deals with one aspect of neutron star simulations: the construction of physically-relevant initial data. Einstein's equations impose various constraint equations that must be satisfied at the initial timestep of a simulation. We discuss the

techniques used in the literature to make these equations tractable, both from a theoretical and numerical standpoint, as well as how we build off of previous approaches for the procedure presented here.

CHAPTER II

GENERAL RELATIVITY, EINSTEIN'S EQUATIONS, AND GRAVITATIONAL WAVES

We begin with an overview of gravitational waves, starting with how they can be derived from Einstein's eqeuations.

2.1. A brief introduction to general relativity

The core mathematical model for general relativity is that of *pseudo-Riemannian geometry*. Broadly, this is the geometry of curved spaces, but it has a crucial twist: the lengths of nonzero vectors may be positive, negative, or zero. Given a coordinate system, this information is encoded in the spacetime metric, written $g_{\mu\nu}$ in the conventional index notation[1]. The metric contains all the inner products of the basis vectors, and as such, it can be used to compute inner products for all vectors, but again, these inner products need not be positive-definite.

The derivatives of the metric allow us to define various useful measures of curvature: the Riemann curvature tensor, the Ricci curvature tensor, and the scalar curvature (or "Ricci scalar"). In particular, the Ricci curvature $R_{\mu\nu}$ and the Ricci scalar R enter into Einstein's famous equation:

$$R_{\mu\nu} + \frac{1}{2} g_{\mu\nu} R = 8\pi T_{\mu\nu}, \tag{2.1}$$

[1]Here Greek characters denote indices ranging over all spacetime dimensions, and Latin characters will denote those ranging over only spatial dimensions. See Appendix A for further details on notation and conventions.

where $T_{\mu\nu}$ is the energy-momentum or *stress-energy tensor*[2], containing information about the energy density, momentum density, pressure, and stress in spacetime. This is one of the main pieces of physical content in general relativity: curvature, encoded in the Ricci tensor and scalar—which is, in turn, related to derivatives of the metric–is directly equal to quantities derived from the presence of matter or mass-energy.

2.2. General relativity and gravitational waves

Remarkably, even in regions of seemingly empty space, spacetime itself can carry away energy through gravitational waves.

Consider a metric $g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}$, where $\eta_{\mu\nu}$ is the flat spacetime metric and $h_{\mu\nu}$ is some suitably small perturbation[3].

With suitably chosen conditions, the Riemann tensor $R_{\alpha\beta\gamma\delta}$ reduces in part to

$$R_{i0j0} = -\frac{1}{2}\frac{d^2 h_{ij}}{dt^2}. \tag{2.2}$$

A metric perturbation obeying this equation is said to be in the *transverse-traceless* gauge. There is always freedom to restrict a perturbation into this gauge.

The required conditions are

$$D_\mu h^{\mu\nu} = 0, \quad h_{\mu 0} = 0, \quad h^\mu{}_\mu = 0 \tag{2.3}$$

These say that the perturbation has zero divergence, that it has no time-time or time-space components, and that it has no trace. These conditions cast Einstein's equations into a convenient wave equation, ensure that all effects of the perturbation

[2]Factors of G and c have been set equal to 1 as per common usage—these are so-called "geometricized units". For more information, consult Appendix A.

[3]This is typically done in Cartesian coordinates, and the determinant of $h_{\mu\nu}$ should be much smaller in magnitude than 1.

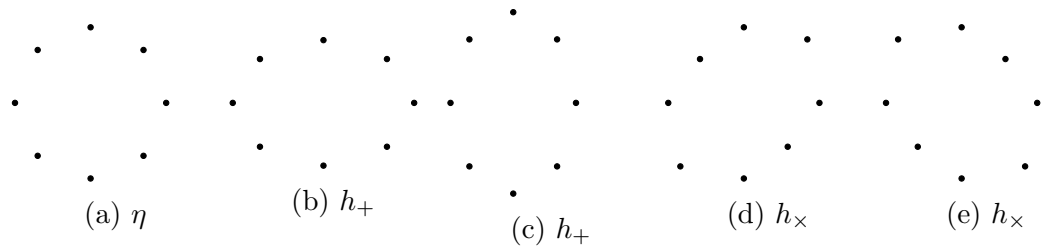


Figure 1: Plus and cross polarizations. From left to right: (a) a ring of test particles, (b) and (c) the same ring under the influences of a gravitational wave with only h_+ polarization, (d) and (e) the ring under influence of a wave with only $h_\times$ polarization. In (b)(c)(d)(e), the ring oscillates between the two corresponding states.

are purely spatial in nature, and allow us to write all these equations in terms of h and not its trace-reversed counterpart $\bar{h}$.

Another net effect of these conditions is that eight out of ten independent components of h are now determined, leaving only two degrees of freedom remaining. These correpond to two types of polarization states, plus and cross, demonstrated in Figure 1. The metric perturbation manifests itself as a linear combination of these polarizations, with amplitudes h_+ and $h_\times$.

Test particles caught in the presence of this metric perturbation follow the equation of motion

$$\frac{d^2x^i}{dt^2} = \frac{1}{2}\eta^{ij}\frac{d^2h_{jk}}{dt^2}x^k, \tag{2.4}$$

and these particles therefore accelerate directly according to the second time derivative of the perturbation.

Thus we see that a small metric perturbation can cause motion of particles in spacetime. The differences in this motion across large regions or distances is how LIGO detected its first graviataional wave event: LIGO's laser interferometers are sensitive to the momentary stretching and compression of spatial distances due to gravitational waves, as this affects the transit time of light through the instrument. Light traverses two paths at right angles, and differential compression or expansion

of these paths due to gravitational waves results in a different interference pattern than would otherwise be expected by the ordinary lengths of the arms. The effect on Earth is tiny—distortions in distances of order 10^{-21}—but over the 4 kilometer length of a LIGO instrument's arms, it proved enough to substantiate a detection.

In the case of an outgoing gravitational wave from a localized source, we can specialize to spherical coordinates. Let r be the radial distance, θ the polar angle, and ϕ the azimuthal angle. The h_+ polarization is then associated with $h_+ = h_{\hat{\theta}\hat{\theta}} = -h_{\hat{\phi}\hat{\phi}}$, where $\hat{\theta}$ and $\hat{\phi}$ denote components with respect to a local orthonormal frame. Similarly, $h_\times = h_{\hat{\theta}\hat{\phi}}$, which is also equal to $h_{\hat{\phi}\hat{\theta}}$ by symmetry.

Now, recall we can measure $\ddot{h}_+$ and $\ddot{h}_\times$, the second time derivatives of h_+ and $h_\times$, using the Riemann tensor, as in Equation 2.2. Hence, we can measure the Riemann tensor in a simulation and use this to extract the second time derivatives of the strain polarizations. We do this far from the source of our gravitational waves—as far as is computationally feasible—to reduce any error inherent in the perturbative approach.

Finally, it is common in the field to combine $\ddot{h}_+$ and $\ddot{h}_\times$ into a single complex quantity Ψ_4, equal to $\ddot{h}_+ - i\ddot{h}_\times$.[4]

When it is not necessary to talk about strain directly, we will state many results about gravitational waves using Ψ_4, or its decomposition into spin-weighted spherical harmonic modes with spin weight $s = 2$. In particular, the $(\ell, m) = (2, 2)$ mode is often dominant for binary systems, and for many of our results, we will confine ourselves to discussion of only this mode.

[4]The use of Ψ_4 and complex numbers derives from Newman-Penrose formalism [46], which uses a tetrad of complex, null vectors. These are suitable to extracting components of the Riemann tensor, which in vacuum is described by 10 independent reals, or 5 complex components—numbered Ψ_0 to Ψ_4. Different Newman-Penrose scalars pertain to different types of gravitatational radiation, including ingoing radiation for example. Nevertheless, the information from Newman-Penrose formalism can always be presented equivalently using manifestly real tensors, and indeed, in one implementation, the real and imaginary parts of Ψ_4 are computed separately, using entirely real tensor manipulations. Hence, we do not deem it necessary to go into detail about deriving Ψ_4 here.

CHAPTER III

NUMERICAL RELATIVITY: SOLVING EINSTEIN'S EQUATIONS WITH COMPUTERS

In practice, the task of solving Einstein's equations often involves—or even demands—computer aid. While this includes the use of computer algebra or numerical implementations of perturbative treatments—for example, post-Newtonian approximations—the direct solution of Einstein's equations numerically is what comprises the field of *numerical relativity*.

The first task to making Einstein's equations amenable to numerical integration is the separation of space and time.

3.1. The 3+1 decomposition

In a neighborhood in which three coordinate funtions are spacelike and one timelike[1], spacetime can be foliated into a continuous family of *spatial hypersurfaces*. This is demonstrated in Figure 2. These hypersurfaces are level sets of the time coordinate, but in general, any given hypersurface has an associated normal (co)vector, and this normal vector need not be parallel to the basis vector associated with the time coordinate.

Using the normal vector n^μ, we can define a metric tensor intrinsic to a given hypersurface. This is denoted $\gamma_{\mu\nu}$, and it is equal to

$$\gamma_{\mu\nu} = g_{\mu\nu} + n_\mu n_\nu. \tag{3.1}$$

Because the normal vector obeys $n_\mu n^\mu = -1$, this tensor merely projects out the

[1]That is, in which each coordinate's associated basis vector is spacelike or timelike.

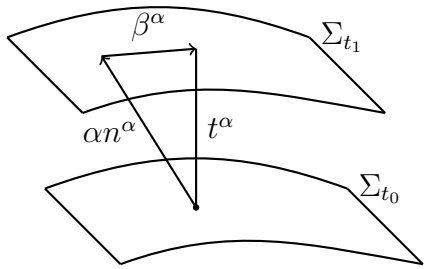


Figure 2: Standard 3+1 decomposition of spacetime: hypersurfaces Σ_{t_0} and Σ_{t_1} correspond to infinitesimally close coordinate times t_0, t_1 respectively. Each hypersurface has an intrinsic metric $\gamma_{\mu\nu}$ and a normal vector n^μ that might not coincide with the direction of time t^α, which is decomposed into a lapse function α and shift vector β^μ.

normal components of any given vector. For this reason, the spatial metric is also referred to as a projection tensor, as it can be used to take a general (four-)vector and yield that vector's projection into the hypersurface. Like the spacetime metric, the spatial metric of a hypersurface is symmetric. We also note that, while we used spacetime indices in this expression, it should be clear that $\gamma_{0\nu} = 0$—the spatial metric is purely spatial, and for this reason, we will have the flexibility to use it with purely spatial indices i, j, k and so on later.

If the spatial metric is specified on a particular hypersurface, how can we compute the spatial metric on some nearby hypersurface? This demands an evolution equation, and indeed, the 3+1 decomposition offers one in terms of the Lie derivative in the normal direction, $\mathcal{L}_n$:

$$\mathcal{L}_n \gamma_{\alpha\beta} = 2\gamma_\alpha{}^\mu \gamma_\beta{}^\nu D_{(\mu} n_{\nu)} = -2K_{\alpha\beta} \tag{3.2}$$

The Lie derivative in the normal direction is a natural derivative operator for evolution equations: it incorporates both pure time derivatives along with changes in the direction of time, and using it here gives us the tensor K, the *extrinsic curvature*, which can be expressed in terms of directional derivatives of the normal vector. Indeed, the use of derivatives of the normal vector immediately suggests a means to derive an evolution equation for $K_{\mu\nu}$: such covariant derivatives appear in the

Riemann tensor. Indeed, the requisite contraction is

$$\mathcal{L}_n K_{\mu\nu} = \gamma_\mu{}^\xi \gamma_\nu{}^\phi n_\xi n_\phi - K_{\mu\lambda} K_\nu{}^\lambda + \frac{1}{\alpha} D_\mu D_\nu \alpha, \tag{3.3}$$

where $\alpha = g_{tt}$ is the *lapse function*, and D_μ is the spatial covariant derivative.

With evolution equations for the spatial metric and its associated extrinsic curvature, we are closer to a fully-determined scheme. However, thus far we have used only 6 of the 20 independent components of the Riemann tensor—and in particular, we've only contracted the Riemann tensor with two normal directions. There are remaining equations that must be obeyed, and these form the basis for constraint equations:

$$\gamma_\alpha{}^\delta \gamma_\beta{}^\epsilon \gamma_\mu{}^\zeta \gamma_\nu{}^\theta R_{\delta\epsilon\zeta\theta} =^{(3)} R_{\alpha\beta\mu\nu} + K_{\alpha\mu} K_{\beta\nu} - K_{\alpha\nu} K_{\beta\mu} \tag{3.4}$$

$$\gamma_\alpha{}^\delta \gamma_\beta{}^\epsilon \gamma_\mu{}^\lambda n^\nu R_{\delta\epsilon\lambda\nu} = D_\beta K_{\alpha\mu} - D_\alpha K_{\beta\mu} \tag{3.5}$$

$$\tag{3.6}$$

where ${}^{(3)}R_{\alpha\beta\mu\nu}$ denotes the spatially intrinsic Riemann tensor within the hypersurface. Neither of these constraints involves time derivatives; they are true constraints that should be obeyed on every hypersurface.

To this point, we have expressed the evolution and constraint equations wholly in terms of the geometry of spacetime. We have not yet used Einstein's equations to couple these equations to the stress-energy tensor. Doing so (by contracting the equations as needed to write equations in terms of the Ricci tensor) yields the canonical forms of the *Hamiltonian constraint* and *momentum constraints*, respectively:

$${}^{(3)}R + K^2 - K_{\mu\nu} K^{\mu\nu} = 16\pi n^\mu n^\nu T_{\mu\nu} = 16\pi\rho \tag{3.7}$$

$$D_\mu (K^{\mu\nu} - \gamma^{\mu\nu} K) = -8\pi \gamma^{\nu\xi} n^\phi T_{\xi\phi} = 8\pi S^\nu \tag{3.8}$$

where K is the trace of the extrinsic curvature, or $K = K_\alpha{}^\alpha = g_{\alpha\beta} K^{\alpha\beta}$. Again, there are no time derivatives in these equations—they are true spatial constraints,

relating matter, energy, and stress to the extrinsic curvature, spatial Ricci tensor, and derivatives thereof. Finally, for the rest of this document, we will use the shorthands ρ, S^μ to denote the particular components of the stress-energy tensor used here.

Similarly, we can do the same for the evolution equation for the extrinsic curvature. Let $S_{\mu\nu} = \gamma_\mu{}^\alpha \gamma_\nu{}^\beta T_{\alpha\beta}$ with S its trace, and we have

$$\mathcal{L}_n K_{ij} = -\alpha^{-1} D_i D_j \alpha + \left[{}^{(3)}R_{ij} + K K_{ij} - 2K_{ik} K_j{}^k + 4\pi[\gamma_{ij}(S-\rho) - 2S_{ij}\right] \quad (3.9)$$

The equations 3.2, 3.7, 3.8, and 3.9 form the modern Anowitt-Deser-Misner (ADM) equations [64][2].

However, in itself the modern ADM equations are only a weak basis for a numerical approach to solving Einstien's equations. This can be proved with an analysis of hyperbolicity.

3.2. Hyperbolicity of PDEs

To ensure that a solution to a differential equation remains bounded by an exponential, let us impose that, for a vector u of evolution variables, we have

$$u(x,t) = A(\vec{k},t) e^{i\vec{k}\cdot\vec{x}}, \quad (3.10)$$

or in the general case, a solution vector might be a superposition or integral of such functions over the wavevector $\vec{k}$. For now, however, we consider the wavevector $\vec{k}$ fixed and confine ourselves to analysis of a single mode.

Suppose there is a linear function H such that $H(\vec{k})$ is a Hermitian, positive-definite matrix. We can use this to define a norm $|\cdot|$ such that

[2]The original ADM equations [4] were derived to cast general relativity in a Hamiltonian formulation. Hence, those equations are written in terms of the metric's canonical conjugate momentum. The transformation between the conjugate momentum and K_{ij} is trivial, but the two sets of equations do not have identical solution sets, only identical solutions corresponding to physical systems.

$$|v|^2 = v^\dagger H(\vec{k})v, \tag{3.11}$$

for any vector (of evolution variables) v.

Let u obey a hyperbolic PDE of the form

$$\partial_t u + M^i \partial_i u = 0, \tag{3.12}$$

where M^i are matrices[3]. Now, impose the condition that $\partial_t |u|^2 = 0$, and we see that

$$\begin{aligned} 0 &= \partial_t |u|^2 && (3.13)\\ &= (\partial_t u^\dagger) H(\vec{k}) u + u^\dagger H(\vec{k})(\partial_t u) && (3.14)\\ &= [iM^j k_j u_0]^\dagger H(\vec{k}) u + u^\dagger H(\vec{k})[iM^j k_j u_0] && (3.15)\\ & && (3.16) \end{aligned}$$

where $u_0 = u(\vec{x}, 0)$ and $v^\dagger$ is the conjugate-transpose of v for any v.

Now, let P denote the *principal symbol*, such that $P(\vec{k}) = M_i k^i$, and we see that our condition reduces to

$$P^\dagger H - HP = 0 \tag{3.17}$$

where, again, P, H are still both functions of the wavevector $\vec{k}$. We can now refer to H as the *symmetrizer*, and we can deduce whether such a function exists by examining the properties of the principal symbol. The cases of interest are

1. P is independent of direction. H is therefore independent of $\vec{k}$ completely. This

[3]We do not treat the case with a nonzero source term or matrices M^i that vary in time, space, or solution vector here. The analysis is similar, however, and some of the differences are discussed in [2].

case can be identified by all the M^i being Hermitian, and the system is said to be *symmetric hyperbolic*.

2. $P(\vec{k})$ always has real eigenvalues, which are distinct from one another, and a complete set of eigenvectors. The system is *strictly hyperbolic*.

3. $P(\vec{k})$ has real eigenvalues that are not necessarily distinct, and a complete set of eigenvectors. The system is *strongly hyperbolic*.

4. $P(\vec{k})$ has real eigenvalues but an incomplete set of eigenvectors. The system is *weakly hyperbolic*.

Crucially, only for some of these categories is a hyperbolic PDE *well-posed*. In particular, a solution is well-posed if its norm $|u|$ is bounded by $Ce^{bt}|u_0|$ for some positive constants C, b. Such does not hold for the weakly hyperbolic case, and the modern ADM equations, cast in a first-order formalism, are merely weakly hyperbolic [38].

For this reason, we must modify the modern ADM equations—at minimum, by recasting them in terms of more convenient variables, or by adding or subtracting constraint equations—to arrive at a well-posed scheme. This is exactly what the BSSN formulation does.

3.3. The BSSN formulation

Though there are several strongly or symmetric hyperbolic formulations of Einstein's equations, the basis for our project is the BSSN formulation[4] [45] [56] [11].

In BSSN, we decompose the spatial metric into $\gamma_{ij} = e^{4\phi}\bar{\gamma}_{ij}$, with $\bar{\gamma}$, the determinant of γ_{ij}, set equal to 1. We will call γ_{ij} the *conformal (spatial) metric*.

[4]Though commonly called BSSN for "Baumgarte-Shapiro-Shibata-Nakamura," Oohara and Kojima contributed to the original paper [45] as well.

Similarly, we write the extrinsic curvature as $K_{ij} = e^{4\phi}\bar{A}_{ij} + \frac{1}{3}\gamma_{ij}K$, so that $\bar{A}_{ij}$ is conformally related to the traceless part of the extrinsic curvature.

Finally, we use derivatives of the conformal spatial metric to define

$$\bar{\Gamma}^i = -\partial_j \gamma^{ij}. \tag{3.18}$$

With these fundamental variables, we arrive at the following evolution equations:

$$\partial_t \phi = -\frac{1}{6}\alpha K + \beta^i \partial_i \phi + \frac{1}{6}\partial_i \beta^i \tag{3.19}$$

$$\partial_t \bar{\gamma}_{ij} = -2\alpha \bar{A}_{ij} + \beta^k \partial_k \bar{\gamma}_{ij} + \bar{\gamma}_{k(i}\partial_{j)}\beta^k - \frac{2}{3}\gamma_{ij}\partial_k \beta^k \tag{3.20}$$

$$\partial_t K = -\gamma^{ij} D_j D_i \alpha + \alpha\left(\bar{A}_{ij}\bar{A}^{ij} + \frac{1}{3}K^2\right) + 4\pi\alpha(\rho + S) + \beta^i \partial_i K \tag{3.21}$$

$$\partial_t A_{ij} = e^{-4\phi}\left[-(D_{\{i}D_{j\}}\alpha + \alpha({}^{(3)}R_{\{ij\}} - 8\pi S_{\{ij\}})\right] + \alpha(K\bar{A}_{ij} - 2\bar{A} + i\ell\bar{A}^{\ell}_{j} \tag{3.22}$$

$$+ \beta^k \partial_k \bar{A}_{ij} + \bar{A}_{k(i}\partial_{j)}\beta^k - \frac{2}{3}\bar{A}_{ij}\partial_k \beta^k \tag{3.23}$$

$$\partial_t \bar{\Gamma}^i = -2\bar{A}^{ij}\partial_j \alpha + 2\alpha\left(\bar{\Gamma}^i_{jk}\bar{A}^{kj} - \frac{2}{3}\bar{\gamma}^{ij}\partial_j K - 8\pi\bar{\gamma}^{ij}S_j + 6\bar{A}^{ij}\partial_j \phi\right) \tag{3.24}$$

$$+ \beta^j \partial_j \bar{\Gamma}^i - \bar{\Gamma}^j \partial_j \beta^i + \frac{2}{3}\Gamma^i \partial_j \beta^j + \frac{1}{3}\bar{\gamma}^{i\ell}\partial_\ell \partial_j \beta^j + \bar{\gamma}^{\ell j}\partial_j \partial_\ell \beta^i \tag{3.25}$$

where $X_{(ij)} = (X_{ij} + X_{ji})/2$ for any X, and $X_{\{ij\}} = X_{(ij)} - \gamma_{ij}X/3$, with X without indices denoting the trace. Here, ${}^{(3)}R_{ij}$ can be expressed in terms of the BSSN variables as

$${}^{(3)}R_{ij} = {}^{(3)}\bar{R}_{ij} - 2(\bar{D}_i\bar{D}_j\phi + \bar{\gamma}_{ij}\bar{\gamma}^{\ell m}\bar{D}_\ell D_m \phi) + 4(\bar{D}_i\phi\bar{D}_j\phi - \bar{\gamma}_{ij}\bar{\gamma}^{\ell m}\bar{D}_\ell\phi\bar{D}_m\phi) \tag{3.26}$$

BSSN relies on the concept of conformal rescaling. Many of the fundamental variables are physical quantities multiplied by some conformal factor, or powers thereof. While one interpretation may be that these conformal quantities are mere conveniences, it is nevertheless common to hear talk of "conformal space" as opposed to "physical" space or spacetime. This is especially true when the conformal spatial

metric is a flat metric, as is often chosen to be the case at initial data. Conformal quantities' transformation laws are not fixed but rather flexible and chosen to suit convenience.

BSSN has been shown to be strongly hyperbolic [53] and while strong hyperbolicity in itself is not enough to guarantee that a numerical scheme is suitable for simulations, the BSSN formalism is used today to the point that, outside of specific comparisons with other formalisms, its use is well-accepted at this point: for instance, [57] refers to at least three other papers by the same group all with similar numerical setup, which includes the use of BSSN for their spacetime evolution.

With a well-posed and high-performing numerical scheme in hand, there are only a few remaining problems for a practical implementation: the computation of spatial derivatives, integration in time, choice of gauge conditions, hydrodynamics, and initial data.

3.4. Computation of spatial derivatives through finite differencing, and the use of adaptive mesh refinement

The BSSN scheme requires second-order spatial derivatives. One way these can be estimated is through finite differencing. For example, consider three points: $p - h, p, p+h$. A second-derivative of a function f at 0 can be estimated through judicious use of Taylor expansions:

$$f(p \pm h) = f(p) \pm hf'(p) + \frac{1}{2}h^2 f''(p) \pm \frac{1}{6}h^3 f'''(p) + O(h^4) \tag{3.27}$$

Using this expansion, we can estimate the second derivative at an arbitrary point:

$$f''(p) \approx \frac{f(p+h) + f(p-h) - 2f(p)}{h^2} \tag{3.28}$$

The error in this approximation is of order h^2.

Similar formulas exist for derivative operators in multiple dimensions. In practice, we often use fourth-order accurate approximations or even better.

The use of a mesh of points for finite differencing demands that we cover our computational domain with points, storing the values of all functions on these points. Uniformly resolved grids for this purpose would be prohibitively expensive.

For this reason, we use mesh refinement: the use of a series of nested meshes, each with higher spatial resolution than its parent (typically, by a factor of 2, so that every other point coincides with a point on the parent mesh). These meshes are more refined in areas of interest—the immediate vicinity around a black hole horizon or a neutron star—and less refined at longer distances, particularly where distance is needed to reduce near-field influences on gravitational waveforms.

Fixed mesh refinement involves a static, unchanging set of nested meshes. While this may be reasonable for some problems, we can imagine some binary systems in which covering both objects with only fixed, unmoving boxes would adversely affect our ability to resolve the objects. Instead, we use adaptive mesh refinement [12] implemented in Carpet[24][54].

3.5. Time integration with the method of lines

The method of lines is one way to complete the time integration of the BSSN equations. With the spatial derivatives computed using finite differencing, we can attack the evolution equations of BSSN as though they are ordinary differential equations. This opens up a suite of integration methods.

Suppose we have a PDE of the form

$$\partial_t u = s + Du \tag{3.29}$$

where u, s are vectors of variables, and D is some spatial differentiation operator.

In the method of lines (MoL), we compute or estimate the spatial derivatives

first. We might do this with finite differencing and, in doing so, arrive at a set of coupled ODEs. For instance, if we let D be a first derivative in the x-direction, we can compute it with a centered difference, and arrive at the equation

$$\partial_t u_i = s_i + \frac{u_{i+1} - u_{i-1}}{2\Delta x}. \tag{3.30}$$

As there are no other explicit derivatives left, we can eliminate the time derivative through ODE integration. In principle, the particular ODE integrator used is arbitrary. Fourth-order Runge-Kutta is a common choice.

The method of lines affords us solutions in time as though it were a continuous variable. In practice, numerical ODE integrators still generate solutions in time on a discrete lattice of times—though the time spacing between lattice points can be much smaller than for spatial points in finite differencing for the same computational cost. In practice, the value for a full timestep must obey the Courant-Friedrichs-Lewy (CFL, or Courant) condition [27]:

$$C = v\frac{\Delta t}{\Delta x} \leq C_{\max} \tag{3.31}$$

where C is the Courant factor, v is the speed of propagation, δt is the timestep, δx the spatial grid spacing (in one dimension; analogous forms of this equation exist for several dimensions), and $C_{\max}$ is a maximum value dictated by the numerical scheme—typically, we take it to be $1/2$. Since disturbances propagate at the speed of light in spacetime, $v = c = 1$ in geometricized units, and our condition reduces to $\Delta t/\Delta x \leq 1/2$.

With the method of lines and finite differencing, we are very close to being able to evolve the fundamental variables of the BSSN formulation. However, the gauge quantities α, β^i remain.

3.5.1 Choosing the lapse and shift: the $1 + \log$ and gamma driver conditions, and the moving puncture gauge

The BSSN formalism does not prescribe the gauge functions α, β^i. Indeed, some of the more obvious choices prove unsuitable for evolution but may be useful for initial data.

A common condition for the lapse is *maximal slicing*, the condition that $K = 0$ and $\partial_t K = 0$, which generates an elliptic equation

$$D^2\alpha = \alpha(K^{ij}K_{ij} + 4\pi[\rho + S]). \tag{3.32}$$

Various factors of constraint equations could be added to this, but regardless, the maximal slicing condition maximizes the volume of a spatial hypersurface [64]. This can be a useful property in black hole spacetimes, in which we might want to minimize the loss of space flowing into a horizon, which is a property of *geodesic slicing* ($\alpha = 1$).

A similar condition on the shift is called *minimal distortion*. We will only state the condition and its properties:

$$(\Delta_L\beta)^i = 2A^{ij}D_j\alpha + \frac{4}{3}\alpha\gamma^{ij}D_jK + 16\pi\alpha S^i, \tag{3.33}$$

which minimizes the shear deformation of an ideal spheroid, with respect to our coordinates, as it evolves in time [64].

The maximal slicing and minimal distortion conditions have very desirable properties, but they pose elliptic equations that would have to be enforced in time, and for this reason, they are not amenable to the hyperbolic evolution offered by the use of the method of lines. Instead, one approach is to construct evolution equations for these functions—evolution equations that do not alter the overall hyperbolicity of the scheme—and reduce the problem of choosing lapse and shift to that of initial data. The $1 + \log$ and "gamma driver" conditions do just that.

The 1+log slicing condition [15] dictates the lapse function α through an evolution equation[5]:

$$\dot{\alpha} = -\alpha^2 f(\alpha) K. \tag{3.34}$$

There are two common choices for $f(\alpha)$: N/α, with N a constant, or 1. The case $N = 2$ yields a solution of the form $\alpha = 1 + \log \gamma$, and this is the historical basis for the name "$1 + \log$" slicing, of which the equation above is a more generalized version. This lapse condition has proven to be very robust, sharing some of the well-behaved properties of more expensive slicing conditions like *maximal slicing*, in which α obeys an elliptic PDE. Hyperbolic slicing conditions like this one are much cheaper by comparison, and they can be solved alongside the rest of the scheme using method of lines.

As with the lapse, there is a family of shift conditions that are hyperbolic in nature, similarly easy to solve alongside the rest of the scheme, and that show good numerical properties. This is the *gamma driver* shift condition [22]:

$$\partial_t \beta^i = B^i \tag{3.35}$$

$$\partial_t B^i = \frac{3}{4} \chi \partial_t \bar{\Gamma}^i - \eta B^i \tag{3.36}$$

We use a modified version of the gamma driver shift condition that has better behavior for moving punctures [61], in part through the introduction of advection terms:

$$\mathcal{L}_n \beta^i = \frac{3}{4} \frac{B^i}{\alpha} \tag{3.37}$$

$$\mathcal{L}_n B^i = \mathcal{L}_n \bar{\Gamma}^i - \eta \frac{B^i}{\alpha} \tag{3.38}$$

[5] [15] actually uses a generalized equation with the shift present, but the form we use in our simulations is reflected here, without shift involved.

With conditions for the lapse and shift, our system of equations is now closed, and we can evolve the conformal spatial metric and the other fundamental variables of the BSSN formalism, recovering physical variables as needed, to simulate vacuum spacetimes—such as binary black hole systems to predict their gravitational waves.

For systems with neutron stars, however, we must have a means to evolve the stress-energy tensor.

3.6. General relativistic hydrodynamics

For our purposes, matter like that in a neutron star can be modeled as a perfect fluid[6]. In our geometricized units, the stress energy tensor $T^{\mu\nu}$ of such a fluid can be written in terms of the rest mass density ρ_0, the enthalpy h, the pressure p, the four-velocity u^α, and the metric as

$$T^{\mu\nu} = \rho_0 h u^\mu u^\nu + p g^{\mu\nu} \tag{3.39}$$

The equations governing the evolution of these fluid variables are the continuity equation and the Bianchi identities:

$$D_\mu(\rho_0 u^\mu) = 0 \tag{3.40}$$

$$D_\mu T^{\mu\nu} = 0 \tag{3.41}$$

These equations guarantee that mass and non-gravitational energy and momentum are conserved.

The *Valencia formulation* [8] of general relativistic hydrodynamics is a scheme using various "conservative" variables that are functions or combinations of the previous "primitive" variables. With the Lorentz factor W given by $W = -u^\alpha n_\alpha$ and $v^i = u^i/W + \beta^i/\alpha$, the conservative variables are

[6] A more complete description of a neutron star—internal structure, radiative effects, etc. would be well below the resolution we can afford.

$$D = \rho_0 W \tag{3.42}$$

$$S_i = \rho_0 h W^2 \tag{3.43}$$

$$E = \rho_0 h W^2 - p - D, \tag{3.44}$$

with D the rest mass density with respect to the simulation frame (ρ_0 is the rest mass density in the fluid's own rest frame), S_i the momentum density, and E the enrgy. The resulting evolution equations are

$$\mathcal{L}_n D = KD - \frac{1}{\alpha} D_k(\alpha D v^k) \tag{3.45}$$

$$\mathcal{L}_n S^i = -\frac{1}{\alpha}(E+D)D^i\alpha + KS^i - \frac{1}{\alpha} D_k[\alpha(S^i v^k + \gamma^{ik} p)] \tag{3.46}$$

$$\mathcal{L}_n E = (E+p+D)\left[v^m v^n K_{mn} - \frac{1}{\alpha} v^m \partial_m \alpha\right] + K(E+p) - \frac{1}{\alpha} D_k[\alpha v^k (E+p)], \tag{3.47}$$

where D_k is a covariant derivative in the spatial hypersurface. An equation of state is required to close the system. Two common equations of state are the ideal fluid equation of state and the polytropic equation of state.

The ideal fluid equation of state is

$$p = (\Gamma - 1)\rho_0 \epsilon, \tag{3.48}$$

where Γ here is the *adiabatic index* (not related to the Christoffel symbols), and ϵ is the internal energy density, $\epsilon = \rho_0 h - p - 1$ in our units.

The polytropic equation of state has instead,

$$p = K\rho_0^{1+1/N} \tag{3.49}$$

with N some positive constant.

Curiously, the ideal fluid equation of state can be integrated to yield a similar equation, $p = K\rho_0^\Gamma$, but this holds only while the assumptions of an ideal fluid hold: in particular, that changes to the fluid are adiabatic—slow compared to the timescale for the fluid to settle into equilibrium. Any hydrodynamical shock violates this condition, for example. For this reason, the polytropic and ideal fluid equations of state are similar, but not quite the same.

With an equation of state available, we can convert freely between the conservative and primitive variables as needed, and often we must do so, as the primitive variables are easier to relate to physical quantities while the conservative variables are more convenient for use with the method of lines.

The practical computation of hydrodynamic flows still requires handling of shocks: physical discontinuities in the flow of material. To do this, we describe briefly the method used by the `Whisky` code [6] [63], which we use for our simultaions.

Without going into great detail, `Whisky` computes a spatial average $\tilde{q}$ of the conservative variables q over Cartesian cells. This spatial average is no longer discontinuous, and thus it is $\tilde{q}$ that is directly evolved using the method of lines.

This shifts the problem of shock capturing to those of "reconstruction" and the "Riemann problem":

First, there is the computation of q from the average $\tilde{q}$. For this, we use piecewise parabolic method [25] as implemented in [6]. As the name suggests, this method involves interpolation with a quadratic polynomial, but we will not discuss the finer details here.

Second, we must solve a "Riemann problem" after reconstruction. Riemann problems in general deal with discontinuous data, and that is precisely what we have here: we recontruct the conservative variables q from their spatial averages $\tilde{q}$ on cells. We do this on the faces of the cells, for that is what is required to compute the fluxes. As a result, reconstruction gives us *two* values: q_1 from one of the cells and q_2 from the

other. We use these as discontinuous initial data in the evolution equations q near the face, and once equipped with a Riemann solver to evolve that data in time, we can then estimate the fluxes as a function of time, giving us the information needed to carry out the evolution of q with the method of lines.

3.7. The Cactus framework

The Cactus code offers a common set of code to manage and develop simulations.

Common code, called the *flesh*, defines and manages how various modules—called *thorns*—can interact and when their routines are run.

The base model of a Cactus simulation involves the scheduling of routines. Each thorn must specify routines that the Cactus infrastructure will call directly. These *scheduled functions* then have access to information about the simulation domain and variables on the grid—the sum total of which is called the *grid hierarchy*. Scheduled routines may require access to each individual component of the grid or might be global in scope, requiring only sporadic information.

Cactus offers a basic structure for the schedule to divide a simulation step into distinct phases. Some of these *timebins* include bins for initial data, evolution (that is, the computation of time derivatives and then the updating of variables according to numerical integration), and analysis.

Thorns must specify not only a schedule but an *interface*—describing functions that should be available as well as functions provided for other thorns to use—and a set of *parameters* that should be provided at runtime, along with default values, or that the thorn might use from other thorns, or that the thorn might extend, providing additional allowed values. Parameters could be integers, real numbers, arbitrary strings, a restricted set of keywords, or boolean values.

Thorns handle a wide range of functionality—from offering packaged versions of libraries like LAPACK to providing grid structure and adaptive mesh refinement in

Carpet to generating initial data in TwoPunctures [3] to computing gravitational wave emissions in WeylScal4 [7].

It is within this common framework for simulations that we develop a new prescription for initial data: by writing new initial data thorns to fit within the Cactus framework, as well as by modifying existing thorns that aid in this process.

Now, we can discuss the mathematics of initial data in general, as well as specific decompositions used, in preparation for explaining our initial data.

3.8. Putting it all together: the Einstein Toolkit and Maya

Throughout this chapter, we have taken care to point out that, while much of the mathematics and algorithms is general, we are also describing the particular implementations and tools we use for our numerical relativity simulations. For example, we said our hydrodynamics code ultimately stems from Whisky and that we use Carpet for adaptive mesh refinement. These disparate codes can be used together through the Cactus framework.

One project to contribute thorns specifically for numerical relativity simulations is the Einstein Toolkit project [42], which performs continued development for gravitational waveform extraction, as well as their own hydrodynamics code GRHydro [44], descended from Whisky, and their own spacetime evolution code, McLachlan[43] [21].

While we at Georgia Tech use many EinsteinToolkit thorns, we maintain our own spacetime and hydrodynamics evolution codes. Together with our in-house analysis thorns, these are called Maya. The evolution codes are nevertheless quite similar: our spacetime evolution uses Mathematica through Kranc [36] [39] to generate C code with tensor operations unrolled, just as McLachlan does, and our hydrodynamics code is also based off Whisky.

At this point, we have covered elements of how a numerical relativity computation

is approached in practice, but one crucial aspect remains: the generation of initial data for our simulations—data that should be as close as possible to the physical system we wish to model and that obeys the constraints on the initial timeslice.

CHAPTER IV

INITIAL DATA FOR NUMERICAL RELATIVITY SIMULATIONS

Regardless of the particular scheme, a physical simulation here consists of a set of evolution and constraint equations. The evolution equations dictate the time derivatives of quantities: if we know the values of all quantities in the scheme on some initial spatial hypersurface, we can update those quantities to another nearby hypersurface (another member of the family of spatial hypersurfaces described by coordinate time) through the evolution equations.

However, the task of computing the values on that initial hypersurface is what we concern ourselves with here. Whatever values we wish to populate the initial hypersurface with, those values should obey the constraint equations. We will describe some methods that are applicable to or inform the task of generating initial data for neutron stars in binaries here.

4.1. Properties of binary systems in general relativity

A general consideration is that we expect reasonable data for binary systems to have low eccentricity: gravitational radiation emission reduces eccentricity [49]. While some eccentric binary systems are considered in the literature, we will spend the rest of our time here discussing initial data for binary systems in quasicircular orbits—as fully circular orbits are impossible without a continual influx of gravitational radiation to replace what the system emits.

4.2. Quasiequilibrium data for binary systems

First, we will discuss one approach to initial data: the use of the conformal thin-sandwich decomposition to generate sequences of quasiequilibrium systems—that is, sequences of configurations that are in approximate equilibrium, or such that the orbital timescale is far longer than that required for objects to achieve equilibrium even as energy and angular momentum are lost through gravitational radiation.

To understand quasiequilibrium sequences, we must discuss the *conformal thin sandwich decomposition* that it is based upon.

4.2.1 The conformal thin sandwich (CTS) decomposition

The conformal thin sandwich decomposition [65][51] relies on specifying the time derivative of the spatial metric as part of the initial data. This has some advantages: time evolution in a neighborhood near the initial hypersurface can be controlled or dictated, and approximate symmetries of the system can be incorporated into the initial data, solving for those time derivatives that remain.

The full CTS scheme[1] allows us to specify $\bar{\gamma}_{ij}$, K, and their time derivatives $\bar{u}_{ij} = \partial_t \bar{\gamma}_{ij}$ and $\partial_t K$.

Some variables used include the traceless extrinsic curvature, $A^{ij} = K^{ij} - \gamma^{ij} K/3$, and its conformal counterpart $\bar{A}_{ij}$, given by $\bar{A}^{ij} = A^{ij}\psi^{10}$. The differential operator $\bar{L}$ obeys $(\bar{L}X)^{ij} = \bar{D}^i X^j + \bar{D}^j X^i - 2\bar{\gamma}^{ij} D_k X^k/3$, where $\bar{D}_i$ denotes a covariant derivative with respect to the conformal spatial metric. Finally, it is conventional to use $\bar{\alpha} = \alpha/\psi^6$. The extended CTS equations then result from the following:

- By substitution into the momentum constraint—Equation 3.8—yielding Equation 4.1

[1] We present only the extended CTS scheme here, which specifies $\partial_t K$ instead of the densitized lapse $\bar{\alpha}$, as this is more popular for quasiequilibrium data.

- By substitution into the Hamiltonian constraint—Equation 3.7—yielding Equation 4.2
- By identfiying αK with $\partial_t \sqrt{g}$ (with g the determinant of the full spacetime metric) and taking a determinant of the full Einstein equations, yielding Equation 4.3
- By definition of the extrinsic curvature as $K_{ij} = \partial_t \gamma_{ij}$ and substitution of the conformal quantities, yielding Equation 4.4

Those equations are now presented below:

$$0 = (\bar{\Delta}_L \beta)^i - (\bar{L}\beta)^{ij} \bar{D}_j \log \bar{\alpha} - \bar{\alpha} \bar{D}_j (\bar{\alpha}^{-1} \bar{u}^{ij}) - \frac{4}{3} \alpha \bar{D}^i K - 16\pi \bar{\alpha} \psi^{10} S^i \tag{4.1}$$

$$0 = \bar{D}^2 \psi - \frac{1}{8} \psi \bar{R} - \frac{1}{12} \psi^5 K^2 + \frac{1}{8} \psi^{-7} \bar{A}^{ij} \bar{A}_{ij} + 2\pi \psi^5 \rho \tag{4.2}$$

$$\begin{aligned} 0 = \bar{D}^2 (\alpha\psi) - \alpha\psi \left(\frac{7}{8} \psi^{-8} \bar{A}^{ij} \bar{A}_{ij} + \frac{5}{12} \psi^4 K^2 + \frac{1}{8} \bar{R} + 2\pi \psi^4 [\rho + 2S] \right) \\ + \psi^5 \partial_t K - \psi^5 \beta^i \bar{D}_i K \end{aligned} \tag{4.3}$$

$$\bar{A}^{ij} = \frac{1}{2\bar{\alpha}} \left[(\bar{L}\beta)^{ij} - \bar{u}^{ij} \right] \tag{4.4}$$

$$\tag{4.5}$$

where ψ is the *conformal factor* and obeys $\psi = e^{-\phi}$, where ϕ is the BSSN variable evolved by Equation 3.19.

The CTS equations put some constraints on the lapse and shift, consistent with the idea that, by specifying a time derivative of the conformal spatial metric, we are constraining ourselves to coordinates consistent with that derivative.

In common application for binary initial data, we choose $\bar{u}_{ij} = \partial_t K = K = 0$, reducing the free data to just the conformal spatial metric, which we choose to be $\bar{\gamma}_{ij} = \delta_{ij}$ for Cartesian coordinates.

Nevertheless, even with these simplifications, the CTS scheme reduces to three coupled, elliptic PDEs. These are, not surprisingly, difficult to solve. One library

available for such a problem is the LORENE code [16], which uses multi-domain spectral methods for solving PDEs.

Regardless of difficulty, the CTS system is well suited to quasiequilibrium data because we can identify the basic vector corresponding to our time coordinate with an approximate Killing vector. Killing vectors correspond to symmetries, and that's exactly the condition we would like to impose here: approximate time-translational symmetry.

4.2.2 Use of CTS in binary system initial data

There are several remaining issues in need of resolution to leverage CTS data into a practical model for initial data for quasicircular binary systems. We will not go into great detail on these, only summarizing some results:

Ensuring quasiequilibrium evolution of apparent horizons and neutron stars. The CTS equations with an approximate helical Killing vector let us, essentially, evolve a binary system in a corotating coordinate frame. Nevertheless, the size and shape of apparent horizons[2] or the size and shape of neutron stars might not be guaranteed to be momentarily in equilibrium.

For black holes, the evolution of an apparent horizon can be enforced using boundary conditions on the horizon. For neutron stars, we must solve the hydrodynamics equations for hydrostatic equilibrium. This admits two common solutions: the corotating star, which rotates along with the corotating coordinates, appearing not to rotate in that frame; or the irrotational star, which is somewhat more complicated as a solution but more astrophysically relevant. Either way, the use of multidomain solvers is powerful here: hydrodynamics equations involving the matter variables but not the spacetime variables will be nontrivial within the stars and trivial outside, and

[2]The determination of an *event horizon* requires global characterization of spacetime, but *apparent horizons* are characterized by more local conditions and therefore the horizons of choice for characterizing black holes in numerical relativity.

indeed, this is a capability offered by LORENE.

Regardless of the method used to solve them, the resulting equations couple a velocity potential Φ, enthalpy h, and rest mass density ρ_0 to the lapse α, and for this reason, the equations must be solved alongside with the CTS equations.

Ensuring quasicircularity of orbits. Whether a system involves black holes, neutron stars, or a combination of the two, ensuring approximate circularity of an orbital system relies on a correct choice of orbital angular velocity, which enters the equations through the approximate helical Killing vector $k^\alpha_{\rm hel} = \partial^\alpha_t + \Omega_{\rm orb}\partial^\alpha_\phi$. For spacetimes that admit a timelike Killing vector, there is a corresponding mass quantity—the Komar mass M_K. One way to solve for the correct orbital angular velocity of two black holes with a given separation is to ensure that the ADM mass and Komar mass are identical [31]. One must be careful to do this in an inertial frame (not the corotating frame that quasiequilibrium sequence simulations are performed in).

A quasiequilibrium sequence. Once we can construct initial data for a particular separation, we can do so for several separations. Provided that these separations are all, at the least, outside the innermost stable circular orbit, then we can generate quasiequilibrium sequences of initial data for binary systems, and in doing so, we can study the relationships between binding energy and orbital angular frequency, or other such properties of initial data.

QE data has been used over the past decade and a half for binary black holes [31] as well as binaries involving neutron stars [28].

4.3. Puncture initial data for black holes

Another approach to binary black hole initial data is the use of punctured domains—the computational domain consists of all but the black hole singularity points. While this method is of no use to neutron stars specifically, we can still use a puncture for

a black hole companion in a binary system, and we will use many of the techniques used for puncture initial data in our procedure as well.

4.3.1 The conformal transverse-traceless (CTT) decomposition

The conformal transverse-traceless decomposition[3] breaks up the traceless extrinsic curvature $\bar{A}^{ij}$ into transverse-traceless $\bar{A}^{ij}_{TT}$ and longitudinal $\bar{A}^{ij}_{L}$ terms that obey

$$\bar{A}^{ij} = \bar{A}^{ij}_{TT} + \bar{A}^{ij}_{L} \tag{4.6}$$

$$\bar{D}_j \bar{A}^{ij}_{TT} = 0 \tag{4.7}$$

$$\bar{A}^{ij}_{L} = \bar{D}^i W^j + \bar{D}^j W^i - \frac{2}{3}\bar{\gamma}^{ij}\bar{D}_k W^k, \tag{4.8}$$

where W is some vector potential. A simplifying result of this decomposition is that the divergence of $\bar{A}$ becomes a vector Laplacian of W: that is, $\bar{D}_j = (\bar{\Delta}W)^i$. This allows us to rewrite the momentum constraint in terms of W as

$$(\bar{\Delta}W)^i - \frac{2}{3}\psi^6 \bar{\gamma}^{ij}\bar{D}_j K = 8\pi\psi^{10} S^i, \tag{4.9}$$

and in doing so, along with the Hamiltonian constraint, we can choose to specify $\bar{\gamma}_{ij}$, $\bar{A}^{ij}_{TT}$, and K and use the constraints to solve for ψ, W^i. The stress-energy-tensor terms from ρ, S^i will themselves come from whatever sources we specify as well.

In itself, the CTT decomposition helps us eliminate derivatives of all the $\bar{A}$ components in favor of second derivatives of W, but we can go one step further: the structure of the momentum constraint now admits analytic solutions, so we can avoid attacking it numerically altogether.

[3]The history of the CTT decomposition is somewhat unclear. Even in [17], which uses CTT extensively, primary references to the decomposition are to non-journal articles, as well as to [66], which is far more general in scope and only cited in [17] with respect to the decomposition of $\bar{A}^{ij}_{L}$ into derivatives of W^i. Nevertheless, the decomposition is treated in major texts, including in [2].

4.3.2 Bowen-York curvature for black holes

Bowen-York curvature [19] is one family of these analytic solutions to the momentum constraint Equation 4.9. In these solutions, $\rho = 0$ and $S^i = 0$, as we are in vacuum[4]. We choose two further simplifying assumptions: that $K = 0$ on the initial timeslice, and $\bar{\gamma}_{ij} = \delta_{ij}$. These conditions are known as *maximal slicing* and *conformal flatness*, respectively. In such a choice, the covariant derivatives reduce to common ones, and the derivative of K term vanishes, so the momentum constraint reduces to

$$\bar{\Delta} W^i = 0 = \partial^j \partial_j W^i + \frac{1}{3} \partial^i \partial_j W^j \tag{4.10}$$

One family of solutions to this equation has the form

$$W^i = -\frac{1}{4r}[7P^i + \ell^i P^j \ell_j] + \frac{1}{r^2}[\epsilon^{ijk} \ell_j J_k] \tag{4.11}$$

The vectors P, J are constant vectors, and ℓ is the unit radial vector, with r the radial coordinate (or related to Cartesian coordinates in the usual way). As parts of a family of solutions, the vectors P, J have no intrinsic meaning. However, constructing the ADM integral for linear momentum yields

$$P^i_{\text{ADM}} = \frac{1}{8\pi} \lim_{r \to \infty} \oint (K^i{}_j - \delta^i{}_j K) \ell^j \, dS = P^i \tag{4.12}$$

A similar equation holds for J—the P, J vectors in the Bowen-York solutions are identical to the total ADM linear and angular momentum, respectively, of the spacetime they generate. Moreover, since the momentum constraints are linear now, any number of Bowen-York solutions can be superposed to yield another solution: that is, we can generate initial data for any number of black holes with arbitrary spins and momenta. All we need do is solve the Hamiltonian constraint at the end to

[4]Black hole spacetimes are universally referred to as vacuum spacetimes in practice, for any nonzero stress-energy tensor should be confined to the singularity itself.

compute the conformal factor, which we can do with an explicit form for the extrinsic curvature. The longitudinal part of the conformal, traceless extrinsic curvature is

$$\bar{A}^{ij}_L = \frac{3}{2r^2}[P^i\ell^j + P^j\ell^i - (\eta^{ij} - \ell^i\ell^j)(P^k\ell_k)], \tag{4.13}$$

$$\bar{A}^{ij}_L = \frac{3}{r^3}[\epsilon^{ik\ell}J_k\ell_\ell\ell^j + \epsilon^{jk\ell}J_k\ell_\ell\ell^i] \tag{4.14}$$

where again, we choose $K = 0$ and $\bar{A}^{ij}_{TT} = 0$ as well, fully specifying the extrinsic curvature.

What remains, then, is to solve for the conformal factor when there are obvious physical singularities. This is the reason for the puncture method.

4.3.3 Black holes as punctures

The method of Brandt and Brügmann [20] considers black holes as punctures—isolated points not part of the domain of the conformal factor—and shows reduces the solving of the conformal factor to an equation for a related quantity u that is entirely regular even in the neighborhood of the puncture locations.

In particular, a conformal factor solution for stationary, nonspinning black hole is $\psi_{0,m} = 1 + m/2r$, with m the *bare mass* an arbitrary parameter[5]. This solution has desirable properties at infinity: the conformal factor falls off to 1, so that asymptotically, the spacetime geometry approaches that of a flat spacetime.

Brandt and Brügmann showed that, using the Bowen-York curvature for an arbitrary number of black holes, one can decompose the conformal factor into

$$\psi = \left(\sum_i \psi^i_{0,m_i}\right) + u \tag{4.15}$$

[5]The bare mass coincides with the ADM mass in this particular case, but in other cases, we will use conformal factors of this form as initial guesses for numerical algorithms, and the bare mass will have no physical meaning.

Since each of the ψ_0 terms satisfies $\nabla^2\psi_0 = 0$, we can eliminate them from the Hamiltonian constraint freely, even though they are not regular at the puncture locations, and arrive at the following equation:

$$\nabla^2 u + \frac{1}{8}\psi^{-7}\bar{A}^{ij}\bar{A}_{ij} + 2\pi\rho\psi^5 = 0 \tag{4.16}$$

An analysis of the limiting behavior of the extrinsic curvature terms shows they are suppressed faster than ${r_i}^3$ near the puncture locations, where r_i is the distance to the ith puncture. Hence, u is completely regular everywhere, even at the punctures, and we can solve the Hamiltonian constraint by solving for u from Equation 4.16 and then adding on the nonregular parts of the conformal factor afterward.

4.3.4 Solving the Hamiltonian constraint with TwoPunctures

TwoPunctures [3] is the name of the original standalone code that implemented a Hamiltonian constraint solver in accordance with the puncture method, and specifically for the case of up to two distinct punctures. This name carried over to the EinsteinToolkit, where TwoPunctures lives on as one of several initial data thorns.

TwoPunctures uses a spectral coordinate system, with coordinates (A, B, ϕ). This compactifies the infinite spatial hypersurface into a finite region of spectral coordinate space, allowing the solver to enforce boundary conditions at infinity by evaluating at a finite point in spectral space. The spectral method is, essentially, a decomposition of the function u into Chebyshev basis functions in the A, B coordinates and Fourier basis functions in the ϕ coordinate.

TwoPunctures then uses a Newton-Raphson iterative algorithm for root-finding: the constraint equation is written in the form $f(u) = 0$, and Newton-Raphson finds the value of u (in actuality, the vector of coefficients for the Chebyshev-Fourier basis functions) that satisfies the equation, up to a specified tolerance. In the Einstein

TOOLKIT, the solution for u is then used to find the conformal factor at the grid points of the finite-differencing mesh, and the user has the option to evaluate the full Chebyshev-Fourier solution directly or approximate the value with Taylor expansion from the nearest spectral gridpoint.

The original TWOPUNCTURES code, as well as the version used in the EINSTEIN TOOLKIT, handles only black holes. Additional terms from matter variables—from hydrodynamics—require modifications to the solver, sometimes in nontrivial ways.

4.3.5 Building a quasicircular black hole binary with CTT/Bowen-York data

Unlike with CTS data, we have no approximate helical Killing vector to guarantee quasicircularity. We discuss two approaches to constructing such a system with puncture-based CTT data here.

One method involves computing an effective potential [26] [9]. Define the *binding energy* E_b as

$$E_b = M_{\rm ADM} - M_1 - M_2, \tag{4.17}$$

where M_1, M_2 are the total masses of the invidiual black holes, which can be computed either using the spin J_i and irreducible mass A_i of each hole, or in the puncture approach, using the value of the regular part of the conformal factor u_i and the bare masses m_i by

$$M_i = m_i \left(1 + u_i + \sum_{i \neq j} \frac{m_j}{2|\vec{r}_i - \vec{r}_j|} \right). \tag{4.18}$$

Under the effective potential method, we set the momenta P_i equal in magnitude and opposite in direction for each object, and orthogonal to the line of separation of the two objects. Then, we vary the separation distance ℓ to find a local minimum in the binding energy: that is, to find where $\partial E_b / \partial \ell = 0$, while holding the two masses

fixed.

Even with this method, however, the may be unacceptable levels of eccentricity. The reason is simple: an inspiral involves a gradual loss of separation, whereas with momenta orthogonal to separation, there is no instantaneous loss of separation. One method suggested to counter this is to evolve a binary through a few orbits, measure the eccentricity, and supply inward radial momentum to counter this [50]. Such a procedure might need to be repeated a few times to converge on a low eccentricity setup, which increases computational expense.

Instead, an alternative—which we will use in our simulations—is to compute the momenta through post-Newtonian approximations of many long-distance orbits, up to a desired separation [35].

In brief, a post-Newtonian expansion is a series expansion. For example, the Lorentz factor W explicitly obeys

$$W = \frac{1}{\sqrt{1 - v^2/c^2}}, \tag{4.19}$$

where here, temporarily, we will leave factors of c explicit. A post-Newtonian expansion of W in powers of v/c looks like

$$W \approx 1 + \frac{1}{2}\frac{v^2}{c^2} + O(v^4), \tag{4.20}$$

where $O(v^4)$ means that higher-order terms are at least of order v^4. A post-Newtonian expansion in this manner expands in terms of quantities that should be small (like v/c in this example) and neglects higher-order terms as a result.

The use of post-Newtonian approximation to simulate the long-distance orbits and generate initial momenta for simulations with full GR reduces measured eccentricity by upwards of a factor of 5 in [35] compared to naïve quasicircular data. Given an initial separation, mass ratio, and total mass, post-Newtonian integration allows us to generate momenta for the system.

4.3.6 Adding hydro to TwoPunctures' Newton-Raphson solver

The problem of adding hydrodynamics terms to the Hamiltonian constraint has long been known, even well before the implementation of a solver in TwoPunctures. York [64] argued that any terms proportional to ψ^n, with $n > 1$, would make the second-order PDE unsolvable, as no algorithm could be guaranteed to converge in the face of small perturbations.

Instead, one approach to this problem is to perform conformal rescaling on the source term ρ from the stress-energy tensor. The choice $\bar{\rho} = \rho\psi^8$ is common, as the rescaling that is common for the momentum constraint—$\bar{S}^i = S^i\psi^{10}$—can be used with this rescaling to compute an energy quantity that is conformally invariant. We will later see that this choice is convenient for other reasons.

With this rescaling in mind, we can now use a TwoPunctures-style solver even with hydrodynamics, solving the equation

$$\bar{\Delta}u + \frac{1}{8}\psi^{-7}\bar{A}^{ij}\bar{A}^{ij} + 2\pi\bar{\rho}\psi^{-3} = 0 \tag{4.21}$$

4.3.7 Recovering the matter primitives after conformal rescaling

When we provide the conformal energy density $\bar{\rho}$, we have to recover the rest mass density ρ_0 and four-velocity u^α after solving the Hamiltonian constraint. We can reconstitute these variables using the following relations:

$$\rho_H = \rho_0 h W^2 - p = \bar{\rho}\psi^{-8} \tag{4.22}$$

$$S^i = \rho_0 h W^2 v^i = \bar{S}^i\psi^{-10} \tag{4.23}$$

$$\tag{4.24}$$

Once we have solved the Hamiltonian constraint, the values of ρ_H, S^i are known; we consider them fixed. We construct a function of f such that $f(\rho_0)$ yet ρ_H, S^i

maintain these fixed values. That function is[6]

$$f(\rho_0) = [\rho_H + p(\rho_0)]^2 - [\rho_H + p(\rho_0)]\rho_0 h(\rho_0) - \gamma_{ij} S^i S^j \tag{4.25}$$

where p, h are functions of ρ_0 by way of the known equation of state, and $\gamma_{ij} = \delta_{ij}\psi^4$—though at any given point $\gamma_{ij} S^i S^j$ can also be considered a fixed quantity. Substituting the expressions from Equation 4.22 and Equation 4.23 yields a function that is manifestly zero for any physically consistent set of variables. All that remains is to find the value of ρ_0 that is indeed consistent with these equations.

For this problem, we use Newton's method with Aitken acceleration—combined, these comprise "Steffensen's method". Given a sequence of Newton's method values x_i, we compute the accelerated sequence $\hat{x}_i$ which obeys

$$\hat{x}_i = x_i - \frac{(x_{i+1} - x_i)^2}{x_{i+2} - 2x_{i+1} + x_i} \tag{4.26}$$

We use Steffensen's method to achieve faster convergence toward a solution than the traditional Newton's method[7].

The algorithm in practice recovers ρ_0, and from there, other matter primitives follow. In particular, the Lorentz factor W obeys

$$W^2 = \frac{1}{2} + \frac{\sqrt{1 + 4\gamma_{ij} S^i S^j / (\rho_0 h)^2}}{2}, \tag{4.27}$$

which follows from the expression for $\gamma_{ij} S^i S^j$ and the quadratic formula. The rest of the fluid primitives follow from the equation of state (determines p, h, etc.) , from normalization of the four-velocity (determines the magnitude of v^i), or from the direction of the momentum density (determines the individual components of v^i up

[6]We're not aware of a prior publication that uses this approach. We attribute the implementation of this method to Roland Haas.

[7]We credit the implementation of this technique for this purpose to Reid Priedhorsky and Brian Gough, though the Newton's method iteration itself they implemented through calls to the GNU Scientific Library (GSL).

to a function of W, which is already known).

CHAPTER V

BOWEN-TYPE EXTRINSIC CURVATURE FOR SPHERICALLY-SYMMETRIC SOURCES

The primary basis for this work is a generalization of Bowen-York curvature for spherically symmetric sources of momentum [17] [47]. This generalized Bowen-type curvature uses the same CTT decomposition as in the case of puncture black holes using Bowen-York curvature, but now generalized to extended, spherically symmetric sources.

5.1. Bowen's original generalization

As Bowen-York curvature offers an analytic solution to Equation 4.10, the spherically symmetric Bowen-type curvature offers an analytic solution to

$$\bar{\Delta} W^i = 8\pi \bar{S}^i = \partial^j \partial_j W^i + \frac{1}{3} \partial^i \partial_j W^j, \tag{5.1}$$

where again $\bar{S}^i = S^i \psi^{10}$, with $S^\alpha = -\gamma^{\alpha\beta} n^\gamma T_{\beta\gamma}$ being the momentum density.

Bowen decomposes W^i into two functions: another vector potential V^i and a scalar potential θ, with the two obeying

$$W^i = V^i - \frac{1}{4} \partial^i \theta. \tag{5.2}$$

When substituted into the momentum constraint, these result in a coupled set of second-order PDEs:

$$\partial_i \partial^i V^j = 8\pi \bar{S}^j \tag{5.3}$$

$$\partial_i \partial^i \theta = \partial_i V^i. \tag{5.4}$$

This decomposition is precisely that which is used for the linear momentum part of the Bowen-York curvature, with $\bar{S}^j$ replaced by a suitable delta function or solved everywhere other than a puncture—either way, the result is the same.

In the more general case, however, we substitute $\bar{S}^i = P^i \sigma$, with σ being a radial function only and P^i an "arbitrary" vector[1].

The function σ plays the role of a radial momentum density, and it is normalized so that

$$P^i = \int_M \bar{S}^i(r)\, dV = \int_M P^i \sigma\, dV = P^i \int_M \sigma\, dV, \tag{5.5}$$

or, more plainly,

$$\int_M \sigma\, dV = 1, \tag{5.6}$$

where M is the extent of the compact source. Solving Equation 5.3 and Equation 5.4 together with these assumptions yields

$$V^i = -2P^i F \tag{5.7}$$

$$\theta = -2P_i x^i H, \tag{5.8}$$

where the functions F, H are radial and obey[2]

[1] We will later show that this is, in fact, the ADM linear momentum of the spacetime.

[2] We assume that the underlying source has compact support—in this case, a limiting radius r_0. We do not write r_0 into the integrals here, as the contribution from σ is zero outside the source, and hence, doing so is unnecessary.

$$F = \frac{1}{r}Q + J \tag{5.9}$$

$$Q(r) = \int_0^r 4\pi\sigma(s)s^2\,ds \tag{5.10}$$

$$J(r) = \int_r^\infty 4\pi\sigma(s)s\,ds \tag{5.11}$$

$$H(r) = \frac{1}{r^3}\int_0^r F(r)s^2\,ds. \tag{5.12}$$

The resulting form of the vector potential W^i is

$$W^i = -2P^iF + \frac{1}{2}P^iH + \frac{1}{2}\ell^i x^j P_j H'. \tag{5.13}$$

The corresponding extrinsic curvature in the exterior of the source is given in [17] as

$$\bar{A}^{ij}_L = \frac{3}{2r^2}[P^i\ell^j + P^j\ell^i - (\delta^{ij} - \ell^i\ell^j)P^k\ell_k] + \frac{3C_{\max}}{2r^4}[P^i\ell^j + P^j\ell^i + (\delta^{ij} - 5\ell^i\ell^j)P^k\ell_k], \tag{5.14}$$

with the constant $C_{\max}$ given by

$$C_{\max} = \frac{1}{2\pi}\int \frac{1}{s} - F(s)\,dV. \tag{5.15}$$

Notice that the first term in Equation 5.14 is precisely the Bowen-York curvature for a black hole with linear momentum. The constant $C_{\max}$ therefore quantifies the deviation of the source from a point source.

For spinning stars, Oohara and Nakamura [47] extended Bowen's work to the case of a star with momentum density $\bar{S}^i = \epsilon^{ijk}J_j x_k \kappa$. κ plays the role of an angular momentum density the way σ does for the linear case. The corresponding normalization conditions, vector potential, and extrinsic curvature are

$$1 = \int_0^\infty \frac{8\pi}{3} s^4 \kappa \, ds W^i \quad = -\epsilon^{ijk} J_j x_k F F(r) = \frac{N(r)}{r^3} + U(r) \tag{5.16}$$

$$N(r) = \int_0^r \frac{8\pi}{3} s^4 \kappa \, ds U(r) \quad = \int_r^\infty \frac{8\pi}{3} s \kappa \, ds \bar{A}^L_{ij} = \frac{3N}{r^3} [\epsilon_{ik\ell} J^k \ell^\ell \ell_j + \epsilon_{jk\ell} J^k \ell^\ell \ell_i] \tag{5.17}$$

The extrinsic curvature here is near-identical to that of a spinning, Bowen-York black hole in Equation 4.14, save for the factor of N—which, by the normalization condition on κ, must go to 1 in the exterior of the star. Hence, in the exterior of a spinning source, both black holes and neutron stars have the same extrinsic curvature for the same J, the total ADM angular momentum.

We will now proceed to present our improvements and additions to the work of Bowen, Oohara, and Nakamura. As Oohara and Nakamura did for the spinning case, we derive a general form of the extrinsic curvature for a spherically symmetric source with linear momentum. We can then use such an expression to set the extrinsic curvature of a star with linear and angular momentum, and to superpose such solutions, for the purpose of constructing initial data of binary systems with neutron stars, or a mix of neutron stars and black holes.

5.2. The interior form of the extrinsic curvature for sources of linear momentum

At the end of last section, we saw that Bowen [17] had derived a natural extension of the Bowen-York curvature, one that provides a solution to the momentum constraint for arbitrary spherically symmetric sources of linear momentum.

For practical use in a numerical relativity code, however, this derivation omitted two crucial needs: first, it contained an explicit expression for the extrinsic curvature only in the exterior of the star, and second, it only considered linear momentum. While Oohara and Nakamura [47] stated a result for sources with spin, the explicit solution for $\bar{A}^{ij}_L$ everywhere in the linear momentum case remained to be shown. We

proceed to derive this now.

Using the vector potential equation Equation 5.1, we can derive a fully general expression for the extrinsic curvature in terms of the functions F, H in Equation 5.9 and Equation 5.12. The necessary relationship under the CTT decomposition is Equation 4.8, which we restate here:

$$\bar{A}_L^{ij} = \bar{D}^i \mathcal{W}^j + \bar{D}^j \mathcal{W}^i - \frac{2}{3}\bar{\gamma}^{ij}\nabla_k \mathcal{W}^k.$$

With conformal flatness, the conformal covariant derivatives $\bar{D}^i$ reduce to their flat space counterparts ∂^i, and in Cartesian coordinates, $\partial^i = \partial_i$.

We need only a few identities to efficiently simplify this expression. First, for any radial function X, $\partial^i X = X'\ell^i$, where again ℓ^i is the unit radial vector. Second, $\partial^i \ell^j = (\eta^{ij} - \ell^i\ell^j)/r$. The result is then

$$\bar{A}^{ij} = (-2F' + H')(P^i\ell^j + P^j\ell^i) + (rH'' - H')(P^k\ell_k)\ell^i\ell^j + \frac{1}{3}(4F' - rH'' - H')(P^k\ell_k)\eta^{ij}. \tag{5.18}$$

This result is still undesirable, however, as derivatives of H will still contain integrals of F. To avoid this, we decompose F into Bowen's Q and J functions as in equations Equation 5.10 and Equation 5.11, and we compute H through integration by parts:

$$\begin{aligned} H(r) &= \frac{1}{r^3}\int_0^r r'^2 \left[\frac{Q(r')}{r'} + J(r')\right] ds && (5.19)\\ &= \frac{1}{r^3}\left[\left(\frac{1}{2}Q(r')r'^2 + \frac{1}{3}J(r')\right)\Big|_0^r - \int_0^r \frac{1}{2}r'^2 Q'(r') + \frac{1}{3}r'^3 J'(r')\, dr'\right] && (5.20)\\ &= \frac{Q(r)}{2r} + J(r) - \frac{1}{r^3}\int_0^r \frac{2}{3}\pi r'^4 \sigma(r')\, dr', && (5.21) \end{aligned}$$

where in the last equation we use $Q'(r) = 4\pi r^2\sigma(r)$ and $J'(r) = -2\pi r^4\sigma(r)/3$.

Using this integration by parts approach, we define a function $C(r)$ such that

$$C(r) = \int_0^r \frac{2}{3}\pi r'^4 \sigma(r')\, dr' \tag{5.22}$$

$$H(r) = \frac{Q(r)}{2r} + J(r) - \frac{C(r)}{r^3}. \tag{5.23}$$

Defining the function C here introducing numerous convenient properties to the computation. In particular, none of the derivatives F', H', or H'' depend on J, nor explicitly on the derivatives of Q or C at all.

First, consider F':

$$F'(r) = \frac{Q'(r)}{r} - \frac{Q(r)}{2r^2} + J'(r) \tag{5.24}$$

$$= 4\pi r\sigma(r) - \frac{Q(r)}{r^2} - 4\pi r\sigma(r) \tag{5.25}$$

$$= -\frac{Q(r)}{r^2}. \tag{5.26}$$

The $Q'(r)/r$ and $J'(r)$ terms cancel.

Now, consider H' and H'':

$$H'(r) = \frac{Q'(r)}{2r} - \frac{Q(r)}{2r^2} + \frac{J'(r)}{3} - \frac{C'(r)}{r^3} + \frac{3C(r)}{r^4} \tag{5.27}$$

$$= 2\pi r\sigma(r) - \frac{Q(r)}{2r^2} - \frac{4}{3}\pi r\sigma(r) - \frac{2}{3}\pi r\sigma(r) + \frac{3C(r)}{r^4} \tag{5.28}$$

$$= -\frac{Q(r)}{2r^2} + \frac{3C(r)}{r^4} \tag{5.29}$$

$$H''(r) = -\frac{Q'(r)}{2r^2} + \frac{Q(r)}{r^3} + \frac{3C'(r)}{r^4} - \frac{12C(r)}{r^5} \tag{5.30}$$

$$= -2\pi\sigma(r) + \frac{Q(r)}{r^3} + 2\pi\sigma(r) - \frac{12C(r)}{r^5} \tag{5.31}$$

$$= \frac{Q(r)}{r^3} - \frac{12C(r)}{r^5}. \tag{5.32}$$

So armed with these expressions, we can now write the extrinsic curvature for a spherically symmetric object with linear momentum, solely in terms of that momentum, the radial direction, the distance from the center, and these Q and C functions,

which are simple one-dimensional integrals themselves and do not require repeated integrations.

Substituting the above into Equation 5.18, we collect like terms from multiples of Q and C to get

$$\bar{A}^{ij}_L = \frac{3Q}{2r^2}[P^i\ell^j + P^j\ell^i - (\eta^{ij} - \ell^i\ell^j)(P^k\ell_k)] + \frac{3C}{r^4}[P^i\ell^j + P^j\ell^i + (\eta^{ij} - 5\ell^i\ell^j)(P^k\ell_k)]. \quad (5.33)$$

Notice that Q in this expression multiplies the Bowen-York curvature for a black hole with linear momentum, as stated in Equation 4.13. That is, with $\bar{A}^{ij}$ written in terms of the Q and C functions, we can directly relate Bowen's curvature for spherically symmetric momentum sources to the simpler case of a black hole. Q is normalized to 1 in the exterior of the source, and C must also reduce to a constant in the exterior. At long distances, the C term is attenuated by the $1/r^4$ factor, and so, in a limiting process, the object will appear more and more like a black hole.

For this reason, in the accompanying code implementation that computes the Bowen curvature, we call the Q function the *momentum weight* and the C function the *point source deviation*, for in the case of a black hole, $Q = 1$ everywhere and $C = 0$ everywhere.

We have therefore reduced the problem of finding the extrinsic curvature of a spherically symmetric source of linear or angular momentum to finding the weighting functions σ, κ and the corresponding integrals Q, C, N that enter directly into the curvature.

5.3. The linear and angular momentum weighting functions for a neutron star

Neither Bowen nor Oohara and Nakamura's derivations prescribe the linear momentum weighting function σ from equation Equation 5.39 or the angular momentum

weighting function κ from equation Equation 5.41. These are arbitrary functions of distance from the center of the source.

In this section, we relate the weighting functions to the hydrodynamical variables and conformal factor of an isolated Tolman-Oppenheimer-Volkoff (TOV) star [48] in isotropic coordinates.

First, we consider the physical momentum density S^i in terms of the rest energy density ρ, the pressure p, the Lorentz factor W of fluid with respect to the normal vector, and the four-velocity u^i:

$$S^i = (\rho + p) W u^i \tag{5.34}$$

In the case of a spherically symmetric source of linear momentum, we can relate this expression to one using the total linear momentum P^i, the linear momentum weighting function σ, and the conformal factor Φ,

$$S^i = (\rho + p) W u^i = \bar{P}^i \sigma \Phi^{-10} \tag{5.35}$$

using the typical choice $\bar{S}^i = S^i \Phi^{10}$.

The notation here can be tricky: we're equating vectors (or components) in conformal space with those in physical space. $P^i = \bar{P}^i$ is an integrated quantity, not a local one, even though the right-hand side is local due to the density σ. Because of this, if we norm both sides of this equation, we must be careful to use a physical metric to contract with S^i while using a conformal metric to contract with $\bar{P}^i$. Doing so yields

$$|S^i|^2 = \gamma_{ij} S^i S^j = (\rho + p)^2 W^2 (W^2 - 1) \tag{5.36}$$

$$= \Phi^4 \bar{\gamma}_{ij} \bar{P}^i \bar{P}^j \sigma^2 \Phi^{-20} = |\bar{P}|^2 \sigma^2 \Phi^{-16} \tag{5.37}$$

This equation suggests defining $\bar{p} = p\Phi^8$ and $\bar{\rho} = \rho\Phi^8$ as conformal pressure and conformal energy density, respectively.

Even with such a choice made, σ is not yet determined. The most we yet know is that

$$|P|\sigma = (\bar{\rho} + \bar{p})W\sqrt{W^2 - 1} \tag{5.38}$$

Now, we make a choice: let σ be defined by

$$\sigma = (\bar{\rho} + \bar{p})/\mathcal{M} \tag{5.39}$$

for some constant $\mathcal{M}$.

This choice has an immediately desirable property: it makes W a constant over the source (again, in the case of linear momentum only). This can be verified by substituting the choice for σ into Equation 5.38 above, yielding $|P|/\mathcal{M} = W\sqrt{W^2 - 1}$. Since $|P|$ and $\mathcal{M}$ are constants over space, W must therefore be constant over space, taking the form

$$W^2 = \frac{1}{2}\left(1 + \sqrt{1 + \frac{4|P|^2}{\mathcal{M}^2}}\right) \tag{5.40}$$

This is precisely what would be expected if $\mathcal{M}$ were the mass of a point particle. So $\mathcal{M}$ plays the role of an effective mass, at least for the purposes of Bowen-type curvature.

In the angular momentum case, we proceed with a similar ansatz as from the linear case: we choose

$$\kappa = (\bar{\rho} + \bar{p})/\bar{I}. \tag{5.41}$$

The resulting form of W is

$$W^2 = \frac{1}{2}\left(1 + \sqrt{1 + \frac{4|J|^2 r^2 \sin^2\theta}{\mathcal{I}^2}}\right) \tag{5.42}$$

This is what would be expected of an object spinning with moment of inertia $\mathcal{I}$.

The constants $\mathcal{M}$ and $\mathcal{I}$ must be determined through the normalization conditions on σ and κ. These yield the following relations:

$$\int_0^{r_0} \sigma(r)\, dV = 1 \implies \frac{1}{\mathcal{M}} \int_0^{r_0} 4\pi(\bar{\rho} + \bar{p}) r^2\, dr = 1 \tag{5.43}$$

$$\int_0^{r_0} \frac{2}{3}\kappa(r) r^2\, dV = 1 \implies \frac{1}{\mathcal{I}} \int_0^{r_0} \frac{8\pi}{3}(\bar{\rho} + \bar{p}) r^4\, dr = 1 \tag{5.44}$$

from which concrete values for $\mathcal{M}$ and $\mathcal{I}$ follow, provided that $\bar{\rho}, \bar{p}$ are specified.

Finally, the similarity of these expressions for $\mathcal{M}$ and $\mathcal{I}$ to the Q and C functions of section 5.2 should be noted. In practice, we will reuse these integrals for the Q and C functions. It is, however, strange that the C function ends up related to that normalization constant for spin. This relationship may bear further study in the future.

CHAPTER VI

IMPLEMENTATION

We now present how we implemented a code to compute and use the Bowen-type curvature for spherically symmetric sources in the context of a numerical relativity simulation.

6.1. Existing code infrastructure and implementation outline

Georgia Tech already had a framework for NR simulations; Bowen-type curvature of this form would merely be one part or component in this existing code.

As stated in section 3.7, much infrastructure already exists for numerical relativity simulations. The problem of implementing this generalized Bowen-type curvature and integrating it into the rest of the simulation framework required the following pipeline:

- For each spherically symmetric object, specify the conformal energy density $\bar{\rho}$ and pressure $\bar{p}$, and the linear and angular momenta P, J.
- Compute the Q, C, N functions for each object, thereby specifying the extrinsic curvature and solving the momentum constraint.
- Solve the Hamiltonian constraint to compute the conformal factor Φ.
- Use Φ and $\bar{\rho}, \bar{p}$ to compute the hydrodynamical primitives used by Whisky: the rest mass density, three-velocity, pressure, internal energy density, and Lorentz factor.
- For a binary system, compute the mass of each object to check the total mass and mass ratio, as the momenta provided are computed from post-Newtonian

inspiral (as in subsection 4.3.5) of a particular system with specified mass ratio and total mass. If the masses have not converged within tolerance, use an iterative method to improve guesses for the black hole or star parameters and repeat.

By comparison, this is what we had available to us before starting this project:

- A post-Newtonian evolution code called PNEvo for point particles that computes the linear and angular momenta of binary companions after evolving them from a longer initial distance
- A Tolman-Oppenheimer-Volkoff (TOV) star solver, which computed the hydro variables as a function of Schwarzschild radius from the center of the star.
- TwoPunctures, a Hamiltonian constraint solver using Newton-Raphson iteration, modified to work with the Einstein Toolkit and to handle hydro source terms.
- A primitive variable solver using Steffensen's method, which computes the hydro variables for evolution given the conformal variables and the conformal factor.

The following sections detail how we repurposed or supplemented the existing code available or implemented new code where appropriate.

6.2. Specification of momenta

We chose to use the momenta resulting from PNEvo directly. For a given initial coordinate separation, mass ratio, and total mass, we would use the PNEvo-computed momenta directly. Future improvements to the pipeline could involve augmenting the PNEvo code with source terms that account for the actual extent of stars, accounting for tidal interactions for example.

6.3. Choice of conformal hydro variables

We chose the conformal hydrodynamical variables based on models of isolated TOV stars. For a single star, the variables take the form

$$\bar{\rho} = \rho_{\mathrm{TOV}} \psi_{\mathrm{TOV}}^8 \tag{6.1}$$

$$\bar{p} = p_{\mathrm{TOV}} \psi_{\mathrm{TOV}}^8. \tag{6.2}$$

When we simulate multiple stars (e.g. in a neutron star binary), we simply add the separate contributions. Since the stars are disjoint and have compact support, this does not have any effect on the value of the conformal variables locally: one star does not influence another in any meaningful way.

6.4. Computation of the Q, C, N functions and the isotropic TOV solver

With the conformal hydro variables set, the weighting functions σ, κ are determined by Equation 5.39 and Equation 5.41, respectively.

In practice, the conformal hydro variables are not known exactly because the TOV solution is not known exactly. Rather, the TOV solution is computed numerically. We had at our disposal a numerical solver using 4th order Runge-Kutta method (RK4) for ODE integration, producing the values of the hydro variables at radial points with a fixed interval, up until a termination condition (for example, a minimum pressure) was reached.

Our implementation modifies the preexisting RK4 solver to directly integrate Q', C', N'.

In addition, the preexisting solver used right-hand sides consistent with a Schwarzschild TOV solution instead of the isotropic coordinates used in puncture black hole simulations using Bowen-York data, so we modified the solver to isotropic coordinates as well.

The primary guide in converting to isotropic coordinates was a MATLAB code provided by Pablo Laguna. This code is also a TOV solver, and it also uses RK4. One of the key issues it illustrates is the inherent difference between the conformal factor used during integration and the final conformal factor that will be used for the rest of the computation.

The issue is the choice of the exterior conformal factor: for a single, isolated object at rest of mass M, the conformal factor at a distance r in the exterior is typically chosen to be $\psi(r) = 1 + M/2r$.

The RK4 integration cannot guarantee this *a priori*. The integration starts at the origin and works outward. Even if the mass and radius were known (or prescribed in advance), a radial integration inward would fail because the TOV equations are stiff.

Hence, in practice, we choose an arbitrary value for the conformal factor at the origin (typically, 1) and then rescale the conformal factor by a constant factor after the integration to meet the condition that $\psi(r) = 1 + M/2r$ in the exterior.

In the TOV model alone, this rescaling affects only the conformal factor itself (as well as the variable $\Theta = \alpha\psi$, with α the lapse, that is used in the isotropic solution), but to compute Q and C inside the same integration loop, powers of the conformal factor must be accounted for.

Let $\tilde{\psi}$ denote the conformal factor originally solved for in the ODE integration. Let $\tilde{r}$ be the radial coordinate for this integration, and suppose that $\tilde{\psi} = A\psi$ for some constant A, where $\psi(r) = 1 + M/2r$ in the exterior (M is a conformal invariant, so it can be computed using the ODE integrator without rescaling).

Now, let $\tilde{R}$ be the radius of the star in the $\tilde{r}$ coordinates, and R be the corresponding radius in the final r coordinates. Note that $A^2\tilde{r} = r$, as we're transforming between two radial coordinates of two sets of isotropic coordinates, and consider the quantity

$$\tilde{\psi}(\tilde{R}) + \tilde{\psi}'(\tilde{R})\tilde{R} = A\psi(R) + A\frac{dr}{d\tilde{r}}\psi'(R)A^{-2}R \tag{6.3}$$

$$= A[\psi(R) + \psi'(R)R] \tag{6.4}$$

$$= A\left[1 + \frac{M}{2R} - \frac{M}{2R^2}R\right] \tag{6.5}$$

$$= A, \tag{6.6}$$

where we used $dr/d\tilde{r} = A^2$ to simplify. This argument allows us to compute the value of A, the required factor for conformal rescaling, from the conformal factor computed with an arbitrary initial condition at the origin.

From here we can compute the necessary rescaling corrections for the Q, C functions. We write these functions first in terms of r, ψ and then in terms of $\tilde{r}, \tilde{\psi}, A$. Let $f(\tilde{r}) = A^2\tilde{r} = r$, and we get

$$Q(r) = \frac{1}{\mathcal{M}}\int_0^r 4\pi[\rho(r') + p(r')]\psi(r')^8 r'^2\, dr' \tag{6.7}$$

$$(Q \circ f)(\tilde{r}) = \frac{1}{\mathcal{M}}\int_0^{f(\tilde{r})} 4\pi[\rho(r') + p(r')]\psi(r')^8 r'^2\, dr' \tag{6.8}$$

$$= \frac{1}{\mathcal{M}}\int_0^{\tilde{r}} 4\pi[(\rho \circ f)(\tilde{r}') + (p \circ f)(\tilde{r}')](\psi \circ f)(\tilde{r}')^8 \tilde{r}'^2 \tag{6.9}$$

$$= \frac{1}{\mathcal{M}}\int_0^{\tilde{r}} 4\pi[(\rho \circ f)(\tilde{r}') + (p \circ f)(\tilde{r}')]A^{-8}\tilde{\psi}(\tilde{r}')^8 f(\tilde{r}')^2 \frac{dr}{d\tilde{r}}\, d\tilde{r}' \tag{6.10}$$

$$= \frac{1}{\mathcal{M}}\int_0^{\tilde{r}} 4\pi[(\rho \circ f)(\tilde{r}') + (p \circ f)(\tilde{r}')]A^{-8}\tilde{\psi}(\tilde{r}')^8 A^4 \tilde{r}'^2 A^2\, d\tilde{r}' \tag{6.11}$$

$$= \frac{1}{\mathcal{M}}A^2\int_0^{\tilde{r}} 4\pi[\tilde{\rho}(\tilde{r}') + \tilde{p}(\tilde{r}')]\tilde{\psi}(\tilde{r}')^8\, d\tilde{r}', \tag{6.12}$$

where $\tilde{\rho}$ is a function of $\tilde{r}$ and similarly for $\tilde{p}$, and the operator $\circ$ denotes function composition.

The procedure to compute Q then follows: we compute the derivative of $Q \circ f$ with respect to $\tilde{r}$ and integrate this in the ODE solver. Then, using the value of A computed from $\tilde{\psi}$ and its derivative, we enforce the condition $Q(R) = 1$. In practice,

this means integrating $q = \mathcal{M}A^2(Q \circ f)$ and then setting $q(\tilde{R}) = \mathcal{M}$.

We use the same general procedure for the C and N functions. Here, we present only the results:

$$(C \circ f)(\tilde{r}) = \frac{1}{\mathcal{M}} A^{-2} \int_0^{\tilde{r}} \frac{2\pi}{3} [\tilde{\rho}(\tilde{r}') + \tilde{p}(\tilde{r}')] \tilde{\psi}(\tilde{r}')^8 \, d\tilde{r}' \tag{6.13}$$

$$(N \circ f)(\tilde{r}) = \frac{1}{\mathcal{I}} A^{-2} \int_0^{\tilde{r}} \frac{8\pi}{3} [\tilde{\rho}(\tilde{r}') + \tilde{p}(\tilde{r}')] \tilde{\psi}(\tilde{r}')^8 \, d\tilde{r}' \tag{6.14}$$

Notice that C, N are proportional as a consequence of our choices for σ, κ. Our implementation computes them separately, but in principle, a more efficient implementation could compute N only (so as to more easily compute $\mathcal{I}$) and then compute $C = N\mathcal{I}/4\mathcal{M}$.

Furthermore, an alternative implementation might choose to leave the TOV solver untouched and compute Q, C, N using the resulting table of data instead—perhaps using Simpson's rule or another quadrature method. This would have an advantage in reusing the TOV solver as it stands more easily, and a table of ρ, p data could be a generic representation for spherically symmetric solutions on a discrete radial lattice. We are strongly considering such a change in the future, to further modularize the code.

Regardless, the rest of our modified, isotropic TOV solver involves ODE integration of the following equations, which can be derived from the conformal thin sandwich formalism (and hence, the TOV equations represent a spherically symmetric solution to the CTS equations with a timelike Killing vector):

$$\frac{1}{r^2}(r^2\psi')' = -2\pi\psi^5\rho \tag{6.15}$$

$$\frac{1}{r^2}(r^2\theta')' = -2\pi\theta\psi^4(\rho + 6p) \tag{6.16}$$

$$p' = -(\rho + p)\left(\frac{\theta'}{\theta} - \frac{\psi'}{\psi}\right) \tag{6.17}$$

where $\theta = \alpha\psi$ and primes denote radial derivatives. We rearrange these equations into a first-order system for ODE integration.

6.5. Interpolation of TOV functions

The numerical integration of the TOV matter functions (as well as the curvature functions Q, C, N derived from them) yields values on an evenly-spaced radial lattice. This lattice cannot in general coincide with the grid points of the Cartesian mesh nor with the spectral grid points used by the TwoPunctures solver. The results must be interpolated.

As with the TOV solver's RK4 routine, the interpolation method we had available to us was a linear interpolation algorithm attributed to Roland Haas. We replaced this routine with piecewise third-order Lagrange interpolation: for any point, we used the four closest known data points to construct a third-order interpolating polynomial to interpolate to the target point.

While three continuous derivatives should not be necessary (the Hamiltonian constraint should only require two), the cubic interpolant requires two points on either side of the target and therefore does not require arbitrary asymmetry in the interpolation method.

6.6. Solving the Hamiltonian constraint with TwoPunctures

Marcus Ansorg and Erik Schnetter originally ported Ansorg's TwoPunctures standalone to the Cactus framework. Tanja Bode later modified this thorn to handle Teukolsky waves and, presumably, (though it is not clearly documented) matter sources in general, in accordance with [64]. Again, per discussion there (and covered in subsection 4.3.6), general matter sources require conformal rescaling for the Hamiltonian constraint PDE to be well-posed. For this reason, the matter source term in this "TwoPuncturesMatter" thorn has the form $2\pi\bar{\rho}_H\Phi^{-3}$.

We made minor structural changes to TwoPuncturesMatter for use with our generalized Bowen-type curvature:

First, we removed unnecessary code for solving the momentum constraint; our generalized Bowen-type curvature automatically solves this constraint.

Second, since all information about the distribution of matter, the initial conformal factor guesses, and the extrinsic curvature are computed in a thorn we wrote called BowenID, the TwoPunctures-like thorn that solves the Hamiltonian constraint does not need to compute any of these source terms on its own. Rather, it need only get them through a Cactus aliased function, which we named `getSourceInfo`, providing the conformal factor (singular parts and nonsingular), the conformal Hamiltonian energy density $\bar{\rho}$, the extrinsic curvature, and the conformal momentum density $\bar{S}$ (which will not be needed until we construct the primitive hydro variables).

Hence, our new TwoPuncturesSolver is literally just a Hamiltonian constraint solver using a spectral grid. It has no awareness of whether the underlying source terms involve black holes, TOV stars, or anything else. This increased separation fosters modularity, but it does shift the burden of computing masses to other thorns, which we will discuss later.

Finally, we separated the computation of TwoPuncturesSolver's solution from the setting of the metric and extrinsic curvature variables. This allows us to generate new solutions based on changing system parameters without looping over the entire mesh, and it is necessary for the iterative procedure to converge on the masses of the objects in a binary system.

6.7. Iterating on the parameters to converge on final masses

Once the hydro primitives are set, we have completely specified all that is required to start evolution—at least, for the parameters we specified. In practice, the TOV solution is governed by (among other things) a choice of central density, and the

puncture black holes are governed by bare mass parameters. There is no guarantee (yet) that these choices will correspond to the masses (total mass and mass ratio) that governed the choice of momenta for the extrinsic curvature.

The most straightforward way to solve this problem is to treat it as a two-dimensional root-finding problem.

We had available a thorn called Baum, written by Frank Herrmann and based off methods in [9]. This approach was originally meant to search for binding energy minima, as described in subsection 4.3.5. For our purposes, this method was outmoded: Baum uses a secant method on each parameter individually to converge on a solution. Such a method is not very well formulated, especially when the parameters differ significantly in relative size (as our black hole bare masses and TOV central densities often do, in the case of mixed binaries).

Rather than repurpose Baum, we wrote a new iterative solver based on Broyden's method, particularly the "bad" Broyden method [40]: given a vector function f, we generate a sequence of approximate inverse Jacobians H_i and approximate roots x_i according to

$$x_{i+1} = x_i - H_i[f(x_i)] \tag{6.18}$$

$$H_{i+1} = H_i - \frac{\Delta x_i - H_i[\Delta f_i]}{\Delta f_i^T H_i \Delta f_i} \Delta f_i^T \tag{6.19}$$

where a^T is the transpose of a, $f_i = f(x_i)$, and $\Delta b_i = b_i - b_{i-1}$ for any sequence b.

We felt this quasi-Newton method appropriate as computing the Jacobian for our system (essentially, a function from the TwoPuncturesSolver conformal factor solution to the masses) would've been prohibitively expensive, if not impossible. Instead, Broyden's method converges rapidly and seldom gets stuck, and it estimates the Jacobian reasonably well in just four TwoPuncturesSolver evaluations.

Our implementation of Broyden's method is intended to be modular: the core

routine is implemented as a higher-order function, accepting an arbitrary function of the required signature (essentially, the function we're finding the root of) as well as a function governing the termination of the iteration procedure (either because tolerance is met or a maximum number of iterations is reached).

That modularity goal is one reason why our Broyden's method code makes heavy use of monads and functors: to allow the core data being manipulated to sit alongside various other, less essential data (like a verbose log) or other similar transformations, with minimal impact on the core algorithm. We will expound on our use and implementation of monads in section 6.10.

6.8. Mass measures of TOV stars and black holes

To complete the initial data procedure, we need some measure of the compact objects' masses.

For black holes, we continue to use Equation 4.18.

For TOV stars, the situation is less settled. There are several candidates for the right mass quantity, but what we found ensures the best circularity is to use the isolated ADM mass of the star. In practice, this cannot be computed directly: unlike with puncture black holes, the isolated ADM mass of a star does not cleanly decouple from the ADM mass of the rest of the system.

Instead, we make one more assumption: that the "isolated ADM mass" and rest mass are both conserved throughout. This is a stronger assumption than that used in the QE literature, in which only the rest mass is presumed conserved (see, for example, [10]), but QE data has no need for a notion like the "isolated ADM mass" that we need here.

At any rate, the rest mass M_0 can be computed by

$$M_0 = \int_\Sigma \rho_0 \alpha \sqrt{\gamma} \alpha u^t \, d^3x, \tag{6.20}$$

where Σ is the full 3D hypersurface, α is the lapse, γ the determinant of the three-metric, and u^t the time component of the four-velocity of fluid. This integral has no interaction terms; individual stars' rest masses can be determined by integrals over their separate extents.

For a given equation of state, there should exist a locally invertible function between rest mass and isolated ADM mass. In practice, we presume that the ratio between rest mass and isolated ADM mass is approximately constant over the range of rest masses we encounter in producing initial data. Thus, for any given rest mass we find, we have an estimate of the isolated ADM mass merely from running the TOV solver once.

An alternative solution to this problem would be to find the rest mass corresponding to the target isolated ADM mass and then running the iterative procedure for initial data with a target rest mass. This would require several additional runs of the TOV solver, so we chose not to pursue this method.

6.9. Computation of the masses

To compute a black hole's mass, we evaluate the Chebyshev polynomial solution for the nonsingular part of the conformal factor, u, at the puncture location. Contributions from any other sources that may be singular somewhere else in the hypersurface (but not at this black hole's puncture) are also added, and then the Equation 4.18 is used.

For computation of TOV rest mass, we must evaluate the hydro primitives at least to the point of computing the rest mass density ρ_0 and Lorentz factor W. Hence, we use our PrimitiveSolver thorn to compute these primitives, given the conformal factor at a point and the conformal hydro variables derived from the TOV solution, as in subsection 4.3.7.

Then, we integrate according to Equation 6.20. Encapsulated in our thorn BasicIntegrator, our algorithm for computing this integral is extremely basic: we establish a rectangular (usually square) box around the star, divide the box into points, and use a composite rectangular rule.

Owing to the general challenges of 3D numerical integration, as well as the approximate nature of the relation between rest mass and isolated ADM mass, we felt a more sophisticated algorithm might introduce disproportionate complexity for comparatively small reductions in total error for the initial data. Nevertheless, any more sophisticated 3D quadrature method could be used here.

Finally, a remark on the computation of u: a faster method for computing u involves Taylor expansion from the nearest TwoPunctures collocation point. This may be suitable for early iterations on the masses and could potentially speed up the initial data procedure overall, with evaluation used only in the final step before evolution.

6.10. Functors, monads, and category theory: high-level manipulation of functions

Both BasicIntegrator and Broyden rely on the power of functors or monads for high-level computations [62], imported through a standalone template thorn we wrote, FunctionalToolkit.

Applied to computer science, a category-theoretic functor F is a map that specifies

1. For every data type A, an associated data type $F(A)$, and
2. For types A, B, a map for functions of signature $A \to B$ to functions of signature $F(A) \to F(B)$, in a manner preserving identity functions and function composition

In implementing functors in C++, a given functor takes the form of a templated

class, having a templated member function called `fmap` that transforms functions in the specified manner.

The basic structure of a functor allows us to avoid repeating ourselves in code with common computational patterns. For example, the `Maybe` functor allows us to deal with computations that may take a particular value or a default value. Code like this...

```
if(have_value)
{
  func(value);
}
else
{
  func(default_value);
}
```

can be replaced systematically by a method accomplishing the same thing:

```
template<typename A>
class Maybe
{
  private:
    bool have_value;
    A value;

  public:
    /* other methods excerpted */

    template<typename F>
```

```
    const auto maybe(const F& func, const A& default_value)
      -> decltype(func(default_value))
    {
      return have_value ? func(value) : func(default_value);
    }
};
```

Thus far, functors might not seem very useful. Indeed, the member function `fmap` would simply apply a function to `value` or, if `have_value` is `false`, return another `Maybe` with the same, but of the new type. A general `List` structure can be understood as a functor, with `fmap` corresponding boardly to `std::transform` for standard containers, but the power of functors is more apparent with their more specific cousins: monads.

Monads introduce an additional operation called `join`, which for a monad M and arbitrary type A has signature join : $M^2(A) \to M(A)$. `join` allows us to flatten nested functor data. This is most useful in combination with `fmap` on functions of signature $A \to M(B)$, defining another function called `bind`, which obeys

$$\begin{aligned} \text{bind} &: [A \to M(B)] \to [M(A) \to M(B)] \\ \text{bind}(f) &= (\text{join} \circ \text{fmap})(f) \end{aligned} \tag{6.21}$$

The usefulness of monads is the power of the tools `fmap` and `bind`. Some monads include `Writer`, which holds a string of output along with the main value of the computation, as well as `Either`, which holds a plain value or an alternative data value that, once introduced, propagates untouched through the rest of the computation until explicitly handled. These handle the common problems of verbose output or exceptions, respectively. `fmap` allows us to write functions that are blind to the monad that might be used in a computation and then promote the function, and `bind` does

a similar thing with functions that must produce errors or log output, and so on.

In `BasicIntegrator`, the vector of data values is encapsulated in a `List` monad. Most other functions are plain functions (e.g. from integers to reals, etc.) and promoted to act on `List` with `fmap`. `List` itself uses `std::vector` as its underlying data, with `std::transform`, `std::accumulate` and so on as backend functions.

In `Broyden`, the base Broyden algorithm is written to be insensitive to the monad used in the overall computation. In the actual initial data pipeline, we use a combination of monads: `Either` is used to pick up errors from `Cactus` infrastructure calls, and `IO` is used to encapsulate the stateful manipulations of `Cactus`.

The use of `IO` monad results in a proliferation of lambda functions being passed around, using `std::function` as a generic container for function objects of specified signature. While it may increase readability—the mere use of `IO` indicates that a function may no longer be referentially transparent, or a simple mathematical function of its inputs—the performance overhead of passing so many lambda functions may demand reconsideration in the future.

In `Broyden`, the `IO` and `Either` monads are combined using a monad transformer[41]: for any given monad M, a monad transformer T produces $T(M)$, which is also a monad. In doing so, the monad transformer also guarantees the existence of a function $\text{lift}_T : M(A) \to T(M)(A)$ for any type A. lift therefore promotes objects from a base monad to the transformed monad.

We implement a monad transformer in C++ using template-template parameters. Here is a skeletal outline of our `EitherT` monad transformer:

```
template<typename L, template<typename> class Md, typename A>
class EitherT
{
  private:
    Md<Either<L,A> > value;
```

```
  public:
    Md<Either<L,A> > run() const
    {
      return value;
    }

  ...
};
```

Here, `L` is the fixed error type of the `Either`, `Md` is the type of the base monad being transformed, and `A` is the base type of data that the monadic object will be associated with. In this case, we also provide a method called `run`, which we use to break out of the monad transformer into a monad with a monadic type as its base type. This is useful when we must use methods associated with the template `Md` in order to remove that layer of the "monadic onion".

The fmap and join operations for monad transformers are more complex than for plain monads. For example, the join operation for $\mathcal{E}^L$, the `EitherT` monad transformer with error type L, is

$$\text{join}_{\mathcal{E}^L(M)}(x) = \left[\text{unit}_{\mathcal{E}^L(M)} \circ \text{fmap}_M[\text{run}_{\mathcal{E}^L} \circ \{\text{either}_{E^L}(\text{left}_{E^L})(\text{id})\}] \circ \text{run}_{\mathcal{E}^L(M)}\right](x), \tag{6.22}$$

where Left_{E^L} denotes the constructor for `Either` data with error type L that holds a value of type L (i.e. is not data of the base data type; in typical usage, this would be an `Either` corresponding to an error message or exception type, not the expected data type of the computation).

CHAPTER VII

TESTS AND ANALYSIS

We now discuss various tests and analyses of the initial data pipeline.

7.1. Description of test cases

We considered four main systems or cases to analyze.

7.1.1 Common features of our test cases

For all cases, we use a polytropic equation of state for the TOV stellar model. For all binaries, we position the objects along the x-axis of the computational domain, approximately such that the center of mass is at the origin (to the extent that such a quantity may be defined). For all binaries, we set the momentum of one object (for its Bowen curvature) equal and opposite to the momentum of the other.

Each simulation's grid structure consists of nested boxes, each typically (but not absolutely) with the same number of points as the next coarser box immediately containing it, but with half the extent per dimension and therefore twice the resolution, and sharing the same center as the next coarser box. In practice, we often move boxes from their parents' centers to track black holes or neutron stars in motion: we fix some number of boxes to be immovable, and then the rest follow an object, steered by an algorithm to find the root of the shift (for black holes) or a search over a region for maximum rest mass density (for neutron stars). Finally, with two compact objects to consider, and without rotational symmetry of the system in the general case, a single coarse box might have two daughter sets of boxes, with each set tracking a different object, and with the separate sets of boxes ultimately merging into a single hierarchy as the objects coalesce.

In some of the coarser refinement levels, we may elect not to perform time refinement between meshes: that is, the time step for one refinement level is the same as that of the next coarser level. Otherwise, most levels use timesteps that are 1/2 that of the next coarser level. When levels differ in spatial resolution by a factor of 2 as well, this condition ensures that the Courant factor is preserved between levels.

In our binary systems, the parameters given are those we computed through our Broyden's method procedure. We use cubical boxes for integration of rest mass density, with 60 points per side length, which we set to the radius of the TOV model. We set target masses according to mass ratio and relative tolerances to 10^{-5}. For computation of the initial Jacobian, we estimate the derivatives through finite differencing about a range of $\pm 5\%$ about the initial parameters. We choose the dimensionless value $\rho_{0,c}K$—the product of the central rest mass density and the polytropic constant for our polytropic, $\Gamma = 2$ equation of state—to characterize TOV stars. For black holes, we use a bare mass equal to the target mass, even though we know that the correct bare mass will almost always be smaller.

The simulations are carried out using the Einstein Toolkit in conjunction with Georgia Tech's Maya code for spacetime evolution and Whisky code for hydrodynamics, and with the use of computational resources from XSEDE and the Stampede supercluster at the University of Texas.

Now, the systems we considered:

7.1.2 An isolated neutron star

This is a TOV stellar model using an initial polytropic equation of state with polytropic constant $K = 100 M_\odot^2$ and $\Gamma = 2$ and central rest mass density $\rho_{0,c} = .1249/K = 6.235 \times 10^{14}\,\mathrm{gcm}^{-3}$. The resulting star has radius 13.4 km and isolated ADM mass $M_{\mathrm{NS}} = 1.543 M_\odot$. We consider various momenta for this star, directed along the x-axis, up to $|P|/M = 0.4$. We carried out this simulation with code unit $M = M_\odot$.

Grid structures for this case vary, as we considered a variety of momenta, necessitating long, rectangular grids with more space in the x-direction. However, the finest box in each case is cubical, with 60 points per dimension (without symmetry) and an extent of $9.10335M$, entirely containing the star at the initial timestep. These simulations have 5 levels of refinement, and we enforce reflection symmetry over the xy-plane.

7.1.3 A black/hole neutron star ("mixed") binary

For the star, we set $\rho_{0,c}/K = 0.138492$ and $K = 123.641 M_\odot^2$, as this is a common value in the literature, and it allows us to match system M50.145b in [55], although we choose a coordinate separation approximately 10% larger than theirs.

We choose a mass ratio $q = M_{\rm BH}/M_{\rm NS} = 5$, and set the bare mass of the black hole, $m_{\rm BH} = .824968M$. These values are the result of our Broyden's method iterative procedure, combined with initial momenta of $(P^x_{\rm BH}, P^y_{\rm BH}) = (-0.000821827, 0.0592790)M$ (these values rounded) and initial separation $d = 8.78M$, with $M \approx 9.26 M_\odot$. In physical units, the star has gravitational mass $1.54 M_\odot$.

The grid structure for this case has 8 levels of refinement with cubical extent of $\pm 134.223M$ in the x- and y-directions. In the z-direction, the extent is $[0, 134.223]$ due to reflection symmetry. On the coarsest level, there are 64 points per x or y side. On all other levels, there are 60 points. The finest level has side length $.98308M$ and resolution $0.0327693M \approx .30 M_\odot$. The coarsest 5 levels have the same timestep size.

7.1.4 A double neutron star binary

This is an equal-mass, nonspinning binary. Each star has $\rho_{0,c}/K = 0.130576$. The code unit and total mass is $M = 3.14 M_\odot$. The stars' separation is $d = 12M$, and their momenta are $\pm(-0.000696566, 0.0851143)M$. These parameters allow us to consider similarities to the "1.62-45" system in [5], although we increase the initial coordinate system from 45 km to 54.6 km.

The grid has reflection symmetry about the orbital (xy) plane and rotational symmetry in this plane as well: hence, the actual computational domain is only the $+x, +z$ quadrant. The grid has 7 levels of refinement, with boundaries at $\pm 192.793M$ in the y-direction. The finest level has side length $2.91521M$ in the y-direction at time 0. The grid's five finest levels may all move to track the motion of one star's center in the computational domain (the other star's center will always be out of the domain due to symmetry). We performed this simulation with three different spatial resolutions, with 60, 88, or 120 points across the side length in the y-direction. Only the finest 2 levels have time refinement over the other 5.

7.1.5 A spinning double neutron star binary

This is also an equal-mass system, but with spin: each star has spin angular momentum $S = (0, 0, -0.0125)$, corresponding to dimensionless spin parameter $a = |J|/M_{\rm NS}^2 = 0.05$. Thus, the stars' spins are aligned, not only with each other but with the orbital axis of rotation as well.

Each star has $\rho_{0,c}K = 0.137266$, and their momenta are $\pm(0.000542612, -0.08096414)M$, with $M = 3.14M_\odot$. The initial separation is still $d = 12M$. We chose these parameters to be similar to system Γ^{--}_{050} in [13], though our separation is larger by an unclear amount.

Like with the nonspinning DNS, the grid has reflection and rotational symmetry, 7 levels of refinement, a maximum extent of $\pm 195.501M$ and a finest level extent of $2.86378M$. Otherwise, the details of the grid structure are substantially the same as that of the nonspinning case. We do, however, only perform this simulation with 60 points across the finest box.

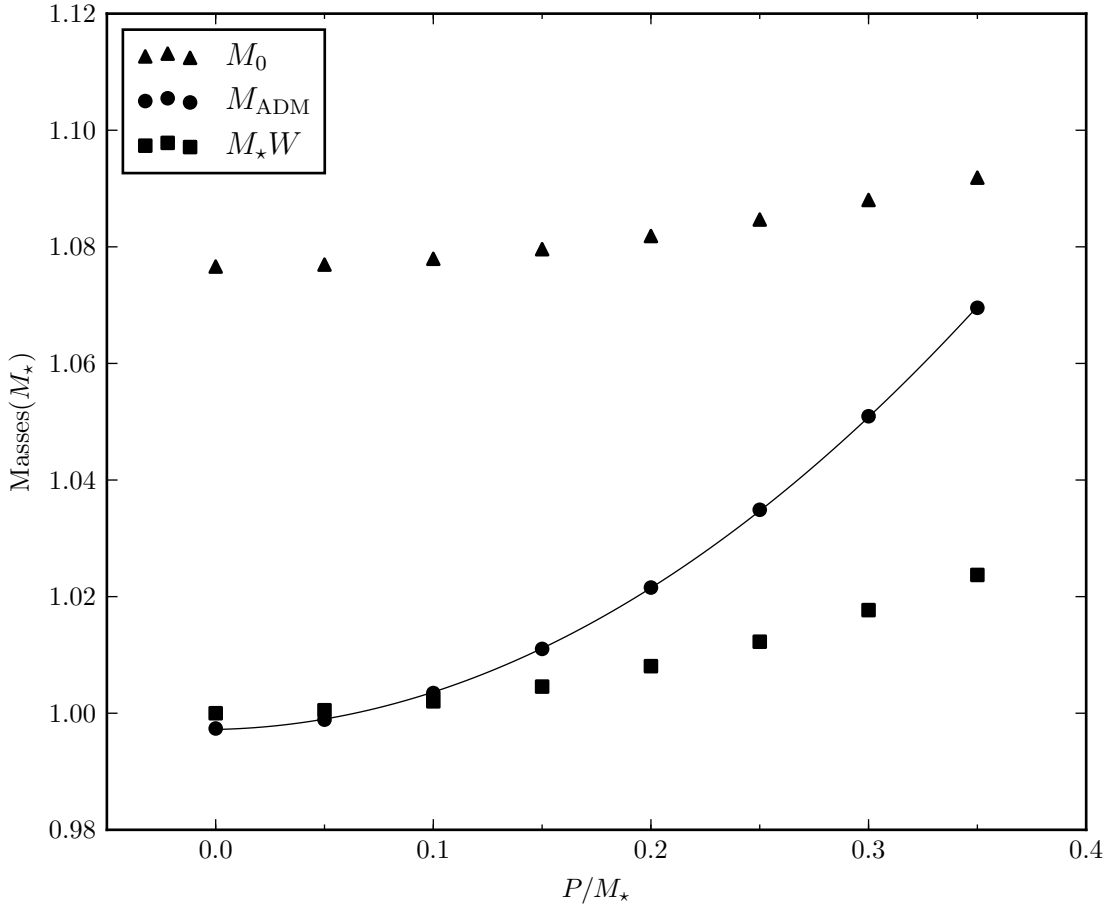


Figure 3: Comparison of various possible mass measures for a neutron star with momentum. M_0 is the rest mass, $M_{\rm ADM}$ is the ADM mass of the system, and $M_\star W$ is the isolated ADM mass of a stationary TOV model multiplied by the Lorentz factor W (which is uniform by construction). We fit the ADM mass to a quadratic in P, as it is expected to obey such a relation in the low momentum limit.

7.2. Isolated star tests

Our isolated star tests comprise various checks on the validity of the data for a range of momenta, down to the zero velocity limit:

System rest mass and ADM mass as a function of momentum. In Figure 3, we compare rest mass and ADM mass for an isolated star as a function of momentum, ranging from $|P|/M_{\rm TOV} = 0$ to 0.4. In general, the rest mass is larger than the ADM mass, but both exhibit quadratic scaling with the momentum, as

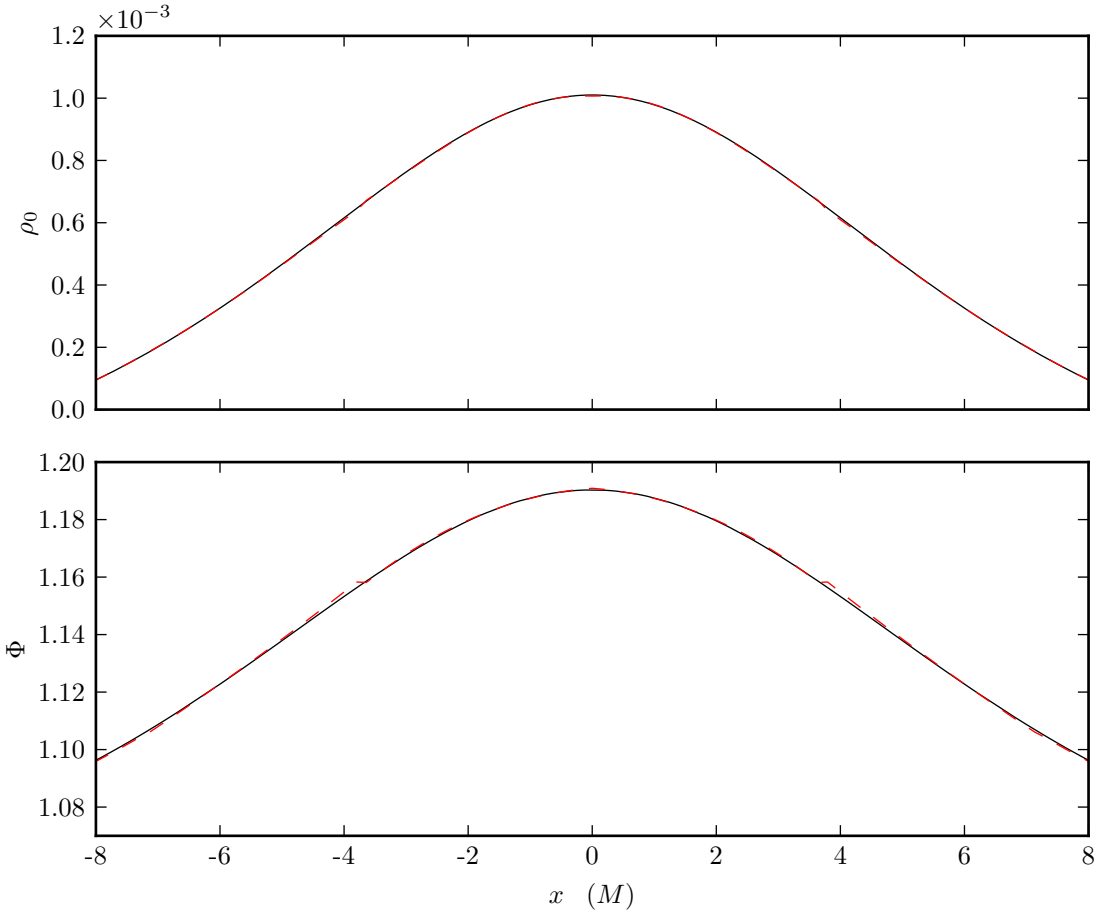


Figure 4: TOV star comparison of the conformal factor Φ and rest mass density ρ_0 before (black) and after (red dashed) passing through our modified TwoPunctures solver for solving the Hamiltonian constraint, as well as subsequent hydrodynamic variable reconstitution with PrimitiveSolver.

expected when expanding the Lorentz factor to its lowest order terms in P: that is, we expect that the ADM mass $M_{\rm ADM} = M_{\rm TOV}W$, where W is the Lorentz factor and obeys $W = (1 - |P|^2/M_{\rm TOV}^2)^{1/2}$. Expanding this in a Taylor series gives $M_{\rm ADM}(|P|) = M_{\rm ADM}(0) + C|P|^2$ for some constant C, which we have determined through a quadratic fit. Similar logic applies for the rest mass.

Comparison of TOV data before and after Hamiltonian constraint solving. A crucial test of correctness is that the TOV solution is not materially changed when we pass pure TOV data through the Hamiltonian constraint solver. That is, in

the presence of no momentum, our algorithm should not introduce substantial differences in the initial data from a TOV solution. Figure 4 demonstrates the agreement between the rest mass density and conformal factor before and after our version of TwoPunctures is used to solve the Hamiltonian constraint.

Evolution and stability of a star with momentum. We evolved a single star with $|P|/M_{\rm TOV} = 0.2$ to examine long-term stability. Some effects of this evolution are shown in Figure 5. The left panel shows how the density profile along the x-axis changes over time. As simulation time progresses, deformations along the leading edge become more prominent. We attribute this to the nonphysical nature of the perfectly spherical shape that Bowen curvature imposes.

Moreover, the effects of a nonphysical initial shape are evident in the longterm evolution of the maximum rest mass density, in the right panel of Figure 5. Here, the central density fluctuates upwards of 3% over time.

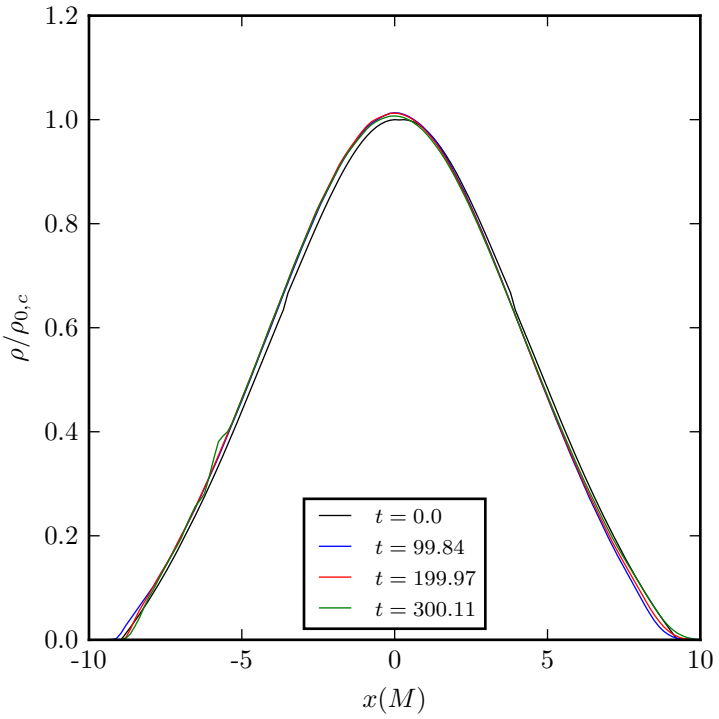
Gravitational radiation from a star with momentum. Finally, we consider the gravitational radiation emitted from a star with this data. For the same case as above, we consider a star with momentum and extract the $(l, m) = (2, 2)$ spherical harmonic mode of Ψ_4 at a distance $r = 150M_\odot$ from the origin. The characterisic spike of spurious radiation is similar to that from Bowen-York data for black holes and is a general property of the conformal flatness approximation used here.

7.3. Binary system tests

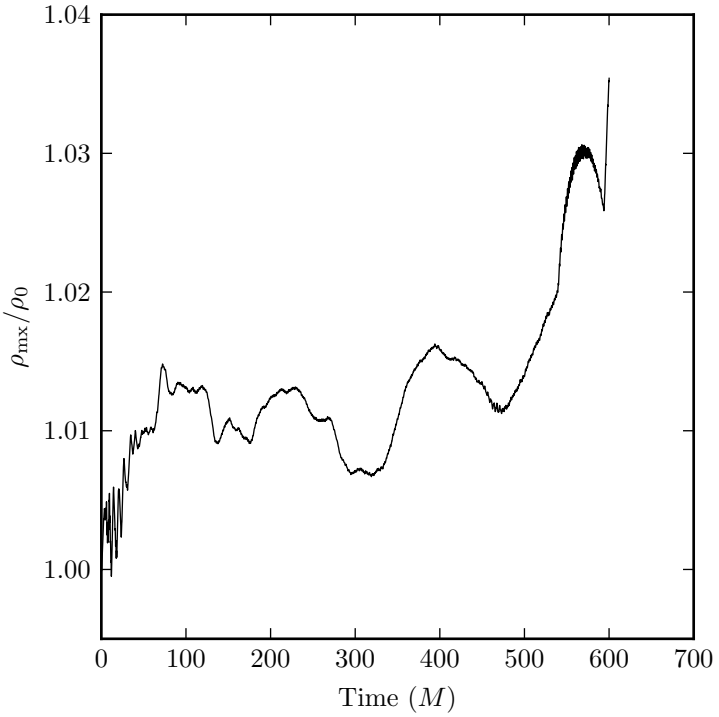
Here we discuss various features of our binary systems, including comparisons to systems in the literature.

7.3.1 Central density variation in stars

Our data exhibits significant variation in the central density of stars during a compact object binary simulation.



(a) Comparison of rest mass density profiles through evolution of a single star with momentum. Profiles are shifted to so that points of maximum rest mass density approximately coincide.



(b) Maximum rest mass density for a star with momentum.

Figure 5: Additional comparisons for a star with $|P|/M_{\mathrm{TOV}} = 0.2$.

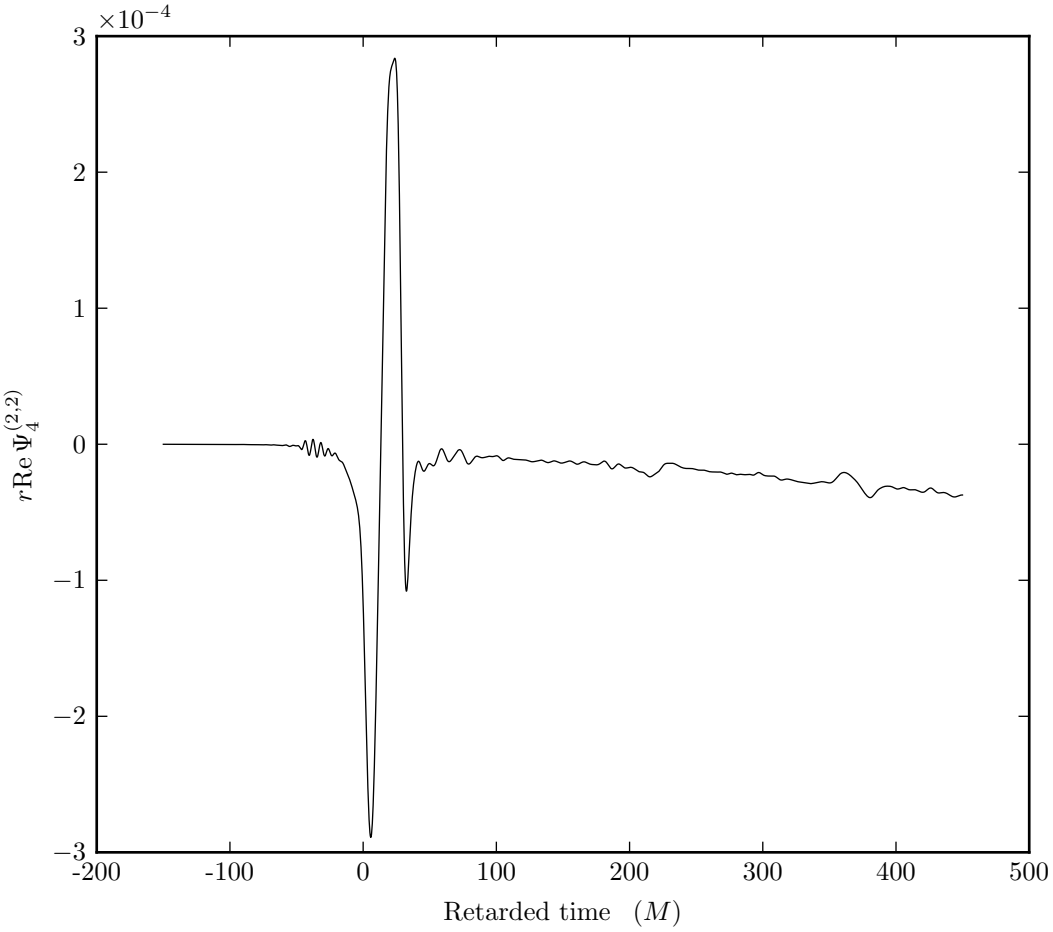


Figure 6: The real part of $\Psi_4^{(2,2)}$ extracted at $r = 150M$ versus retarded time for our single star with momentum $|P| = 0.2M_\star$ in the x-direction. Nonphysical radiation reaches the extraction surface prior to the initial burst of radiation expected from conformal flatness approximation. We attribute the slight decrease in signal over time to the off-center position of the star at late times relative to the fixed, origin-centered extraction sphere.

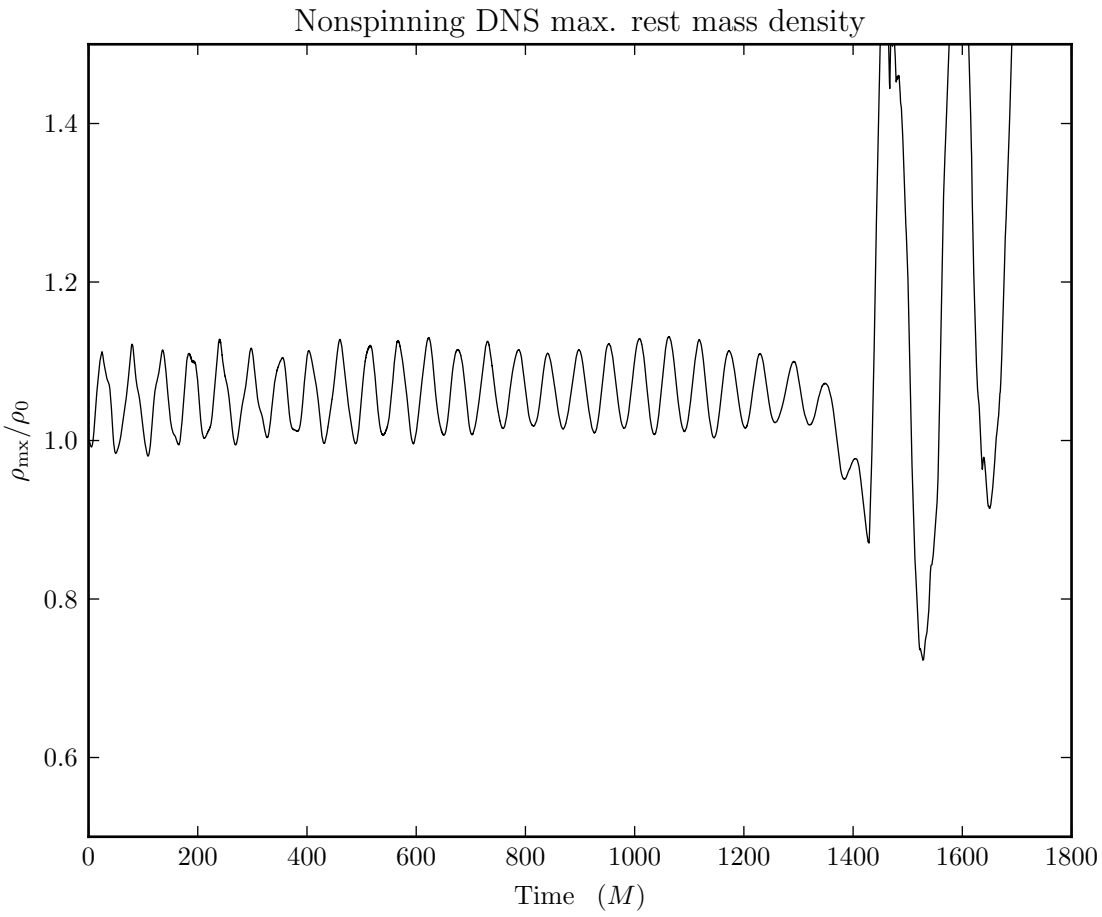


Figure 7: Ratio of the maximum ρ_0 to that on the initial timeslice for our nonspinning DNS binary. Density fluctuations are approximately 15% in the early inspiral, and these fluctuations manifest for the entire star as a continuous pulsation throughout.

As evident in Figure 7, our stars experience central density oscillations on the order of 10% during inspiral, and this phenomenon is consistent (though the amplitude decreases over time) in each of our three binary test cases. This is a defect not shared by quasiequlibrium data, which solves the hydrodynamical equations on the initial timeslice in a more expensive, but also more systematic manner.

Ultimately, we believe that the data will have to be improved to remove this defect as, among other reasons, it may introduce high frequency oscillations in the frequency of Ψ_4. One possible strategy to remedy this defect is to make the stars nonspherical. Bowen suggested a means to decompose sources into spherical harmonic modes (see subsection 8.1.2), but given that we expect stars with linear or angular momentum to take on spheroidal shapes, we may have to consider some spheroidal harmonic decomposition instead. This would drastically increase the complexity of the initial data, as well. We expand on this idea later.

7.3.2 Center of mass drift

In our mixed binary case, the center of mass of the system drifts significantly away from the origin, even during inspiral. We compute the center of mass $r^i_{\rm CM} = M_1 r^i_1 + M_2 r^i_2$, where r_1, r_2 are the Cartesian coordinate positions of objects 1 and 2, and similarly for the masses M_1 and M_2, which are presumed from the prescribed mass ratio and total mass of the system.

Figure 8 shows the progressive drift of the center of mass in this mixed binary system. The drift remains small or nonexistent during the first $100M$ of time evolution. While this drift remains unexplained, anecdotally we have noticed that, as we improved our initial data procedure (particularly the computation of masses to match the PN initial data), the amount of drift has decreased. This suggests that the measurement of NS mass can be further improved. Alternatively, there could be a better choice of gauge for shift evolution. We do not, at this time, believe that

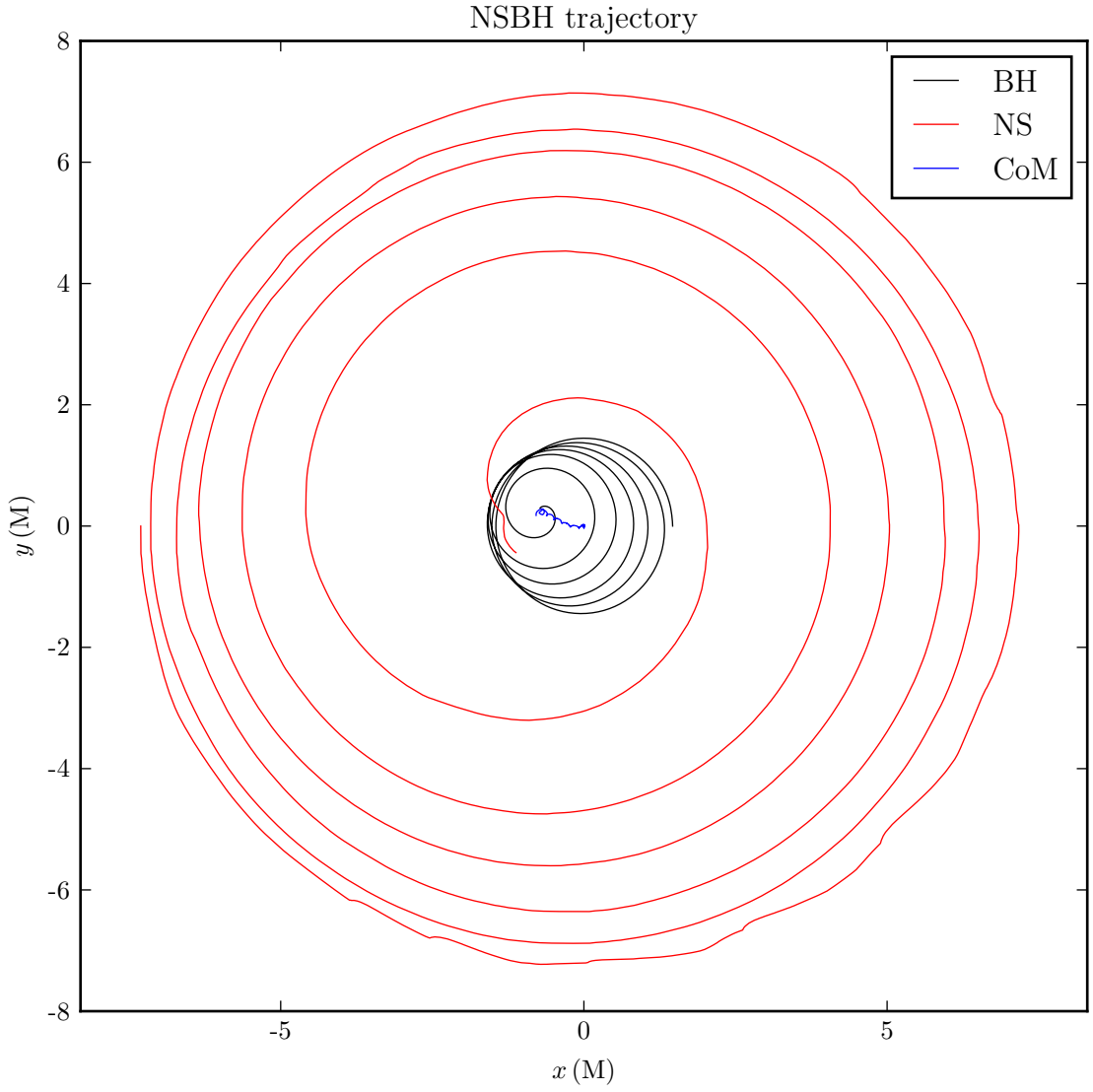


Figure 8: Orbital trajectories of objects in our mixed binary. The black hole is in black, and the neutron star in res, and the center of mass in blue.

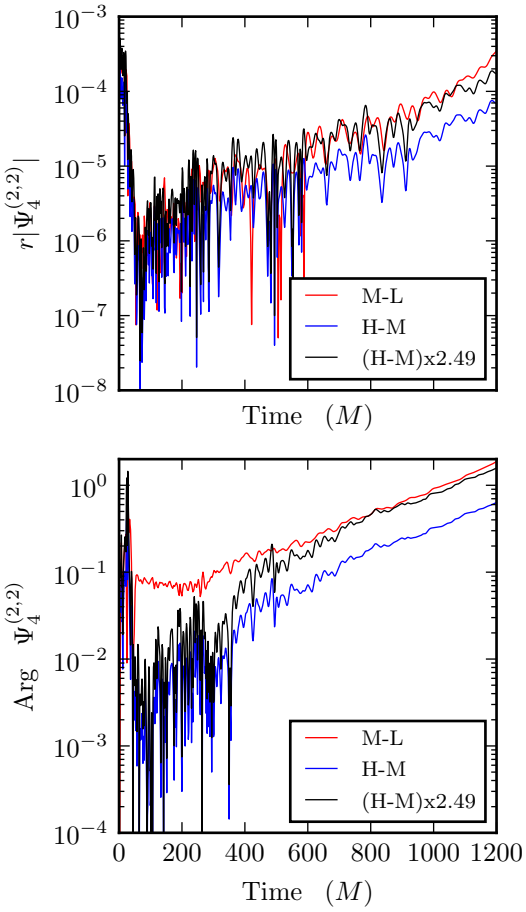


Figure 9: Convergence measurements for our nonspinning DNS simulation. Simulations at low (L), medium (M), and high (H) resolutions are used, and their differences are plotted. The residuals between high and medium (H-M) and medium to low (M-L) are compared with multiplying the larger residual by 2.49—a factor corresponding to second order convergence with our resolutions.

this drift represents actual kick velocity due to asymmetric emission of gravitational radiation, nor do we think this is a numerical effect from the position of refinement boundaries (which could, in principle, reflect gravitational waves back into the system and alter system dynamics).

7.3.3 Waveform convergence

We verify here that our gravitational wave signals converge with increasing resolution and extraction radius.

First, we examine effects of increasing resolution. We performed our nonspinning DNS simulation with 60, 88, and 120 points per dimension (ignoring symmetries) on the finest mesh. We compare the absolute value of residuals between the gravitational waveforms produced in Figure 9.

We presume that, for a resolution (grid spacing) Δ, a numerically computed function f obeys

$$f(\Delta) = f(0) + u(\Delta), \tag{7.1}$$

where $f(0)$ is the exact solution to the differential equation. According to Richardson [52], we can expand u as a Taylor series:

$$u(\Delta) = u_1\Delta + \frac{1}{2}u_2\Delta^2 + \frac{1}{3!}u_3\Delta^3 + \dots \tag{7.2}$$

Then, for our resolutions that are $h/60, h/88, h/120$ (with h a fixed value that will ultimately cancel), we obtain two equations:

$$f\left(\frac{h}{60}\right) - f\left(\frac{h}{88}\right) \approx u_1\frac{h}{240} + 7.432 \times 10^{-5}u_2h^2 + \dots \tag{7.3}$$

$$f\left(\frac{h}{88}\right) - f\left(\frac{h}{120}\right) \approx u_1\frac{h}{330} + 2.984 \times 10^{-5}u_2h^2 + \dots \tag{7.4}$$

Experimentally, we find that the differences in our measured waveforms closely matches the ratio of the second order terms, or approximately $2.491 : 1$, as shown in Figure 9. This suggests $u_1 = 0$ and that the finite differencing error is dominated by those terms second-order in h. Though this analysis uses the gravitational waveform, this behavior is broadly consistent with previous studies of the Whisky code for GR hydrodynamics [14], which also showed approximately second-order convergence.

We also present an estimate of the error in our high-resolution using Richardson expansion. In brief, we expand $u = r\Psi_4^{(2,2)}$ in a power series in the resolution Δ. For an order n scheme, we have

$$u(\Delta) = u(0) + E_n \Delta^n + O(\Delta^{n+1}) = u(0) + \epsilon(\Delta), \tag{7.5}$$

where $\epsilon(\Delta)$ is the error with respect to the exact solution $u(0)$ as a funciton of the resolution. We can then estimate the error for any given resolution given a lower resolution solution:

$$u(\Delta_1) - u(\Delta_2) = E_n(\Delta_1^n - \Delta_2^n) + O(\Delta^{n+1}) \tag{7.6}$$

$$= E_n \Delta_2^n \left(\frac{\Delta_1^n}{\Delta_2^n} - 1 \right) + O(\Delta^{n+1}) \tag{7.7}$$

$$= \epsilon(\Delta_2) \left(\frac{\Delta_1^2}{\Delta_2^2} - 1 \right) + O(\Delta^{n+1}) \tag{7.8}$$

Since we measure $u(\Delta_1), u(\Delta_2)$ directly with our simulations, and we know that our waveforms exhibit approximately second-order convergence ($n = 2$), we solve these equations for the high-resolution error $\epsilon(\Delta_2)$. We performed this analysis on the magnitude and phase of $r\Psi_4^{(2,2)}$, from a time after the first orbit to just before the final inspiral/merger phase.

We find that the relative error in the magnitude is under 10% on this interval, and the absolute error in the phase is under 0.4 radians in the late stages of the inspiral.

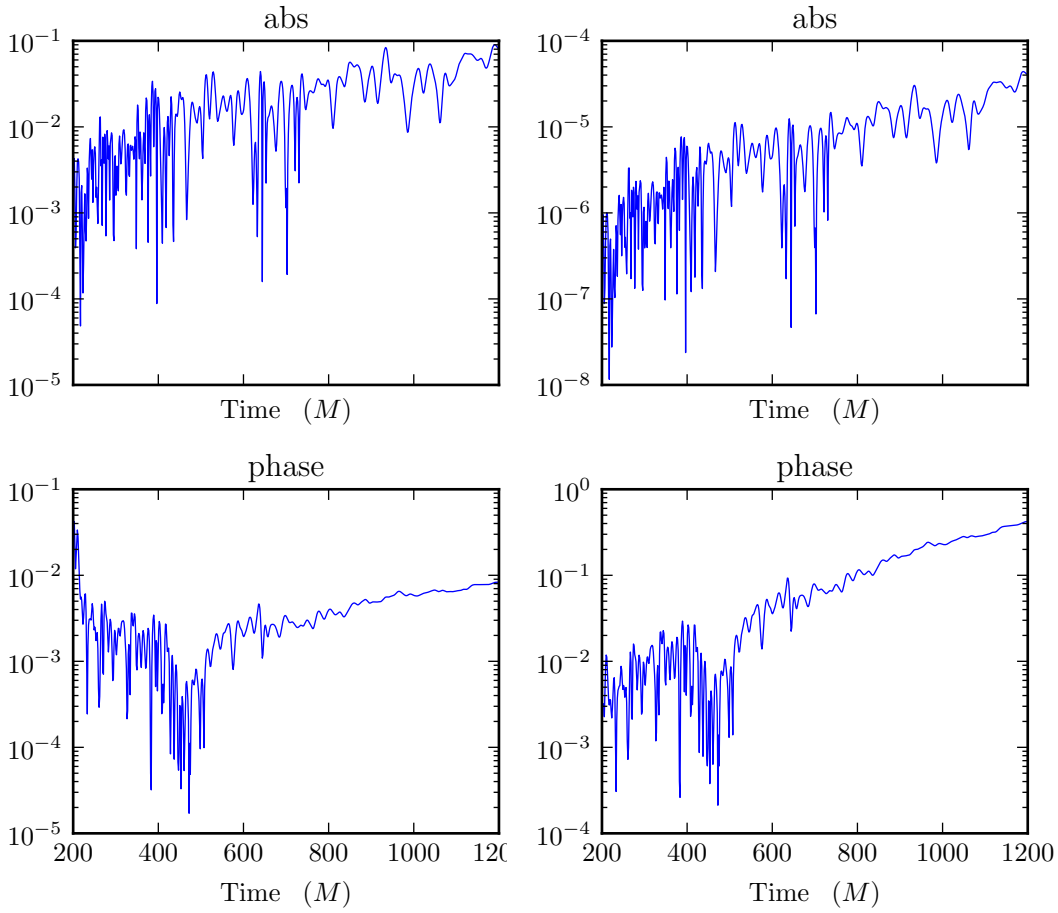
7.3.4 Radiated energy and angular momentum

Gravitational waves carry away energy and angular momentum from their sources.

Let $\ddot{H} = -\psi_4$, so that $H = h^+ + ih^\times$ is a "complex strain". The energy radiated is then [23]

$$\dot{E} = \lim_{r\to\infty} \frac{r^2}{16\pi} \oint |\dot{H}|^2 \, d\Omega \tag{7.9}$$

$$\dot{P}^i == \lim_{r\to\infty} \oint \ell_i |\dot{H}|^2 \, d\Omega, \tag{7.10}$$



(a) Relative errors

(b) Absolute errors

Figure 10: Richardson errors for the absolute value and phase of $r|\Psi_4^{(2,2)}|$ in our nonspinning DNS system.

where ℓ^i is the unit radial vector, once again. The angular momentum radiated is [60]

$$\dot{J}_z = -\lim_{r\to\infty} \frac{r^2}{16\pi} \Re \oint \partial_p hi H \dot{H}^\dagger \, d\Omega. \tag{7.11}$$

In practice, we decompose ψ_4 into spin-weighted spherical harmonics with spin weight $s = 2$, and in some cases, we only consider the $(s, \ell, m) = (2, 2, \pm 2)$ mode of ψ_4, which we will refer to consistently as $\psi_4^{(2,\pm 2)}$.

In both our mixed binary and spinning DNS binary, we find a shortfall of gravitational radiation, both in terms of energy and angular momentum radiated through the $(\ell, m) = (2, 2)$ mode. For our spinning DNS binary, the energy and angular momentum radiated through the $(2, 2)$ mode are .7% and 16%, respectively, compared to 1.2% and 18% in [13]. Similarly, for our mixed binary, we find the energy radiated to be .7% versus 1% in [55].

One factor in common for both systems is that we use a larger initial separation, which naïvely would lead us to expect even more energy and momentum radiated—though most energy and momentum should be radiated from merger, and less so from inspiral.

Instead, that we see a consistent deficit may suggest a defect in our merger phase dynamics. The low resolutions of both of these simulations could be a factor, but [33] suggests that resolution should have only a weak effect on energy and momentum radiated beyond 60 points per dimension. Since our reference system for our non-spinning DNS has no information about energy or momentum radiated, this remains an unsolved problem for the data.

7.3.5 Orbital eccentricity

A key test of circularity is to measure orbital eccentricity. We follow [35] in fitting the coordinate separation to a function D_c of the form

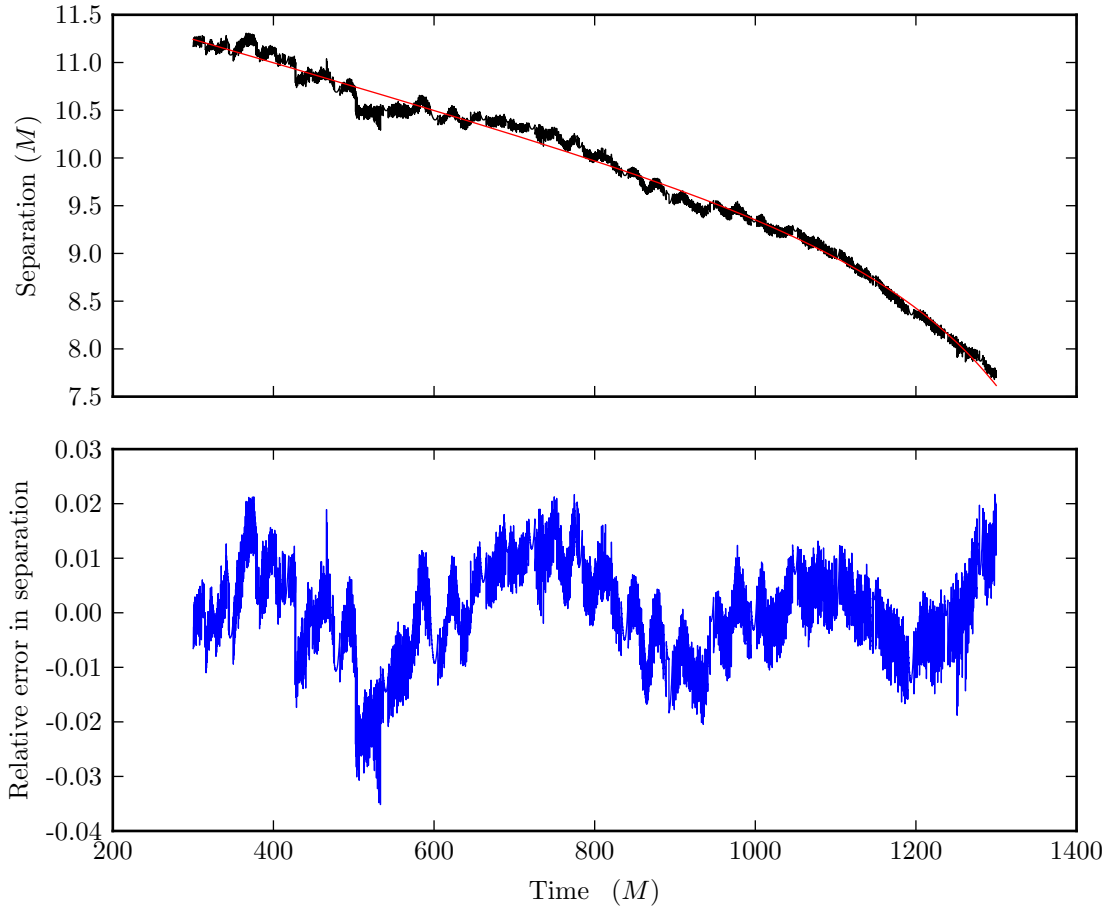


Figure 11: Nonspinning DNS coordinate separation and eccentricity plots. Top panel: Coordinate separation (black) and corresponding fit (red), computed according to Equation 7.12 and [35]. Bottom panel: relative error $(D - D_c)/D_c$, with D the raw coordinate separation and D_c the fit. Eccentricity is approximately the magnitude of the extrema of relative error. We omit some period of time at the beginning of the simulation and near merger.

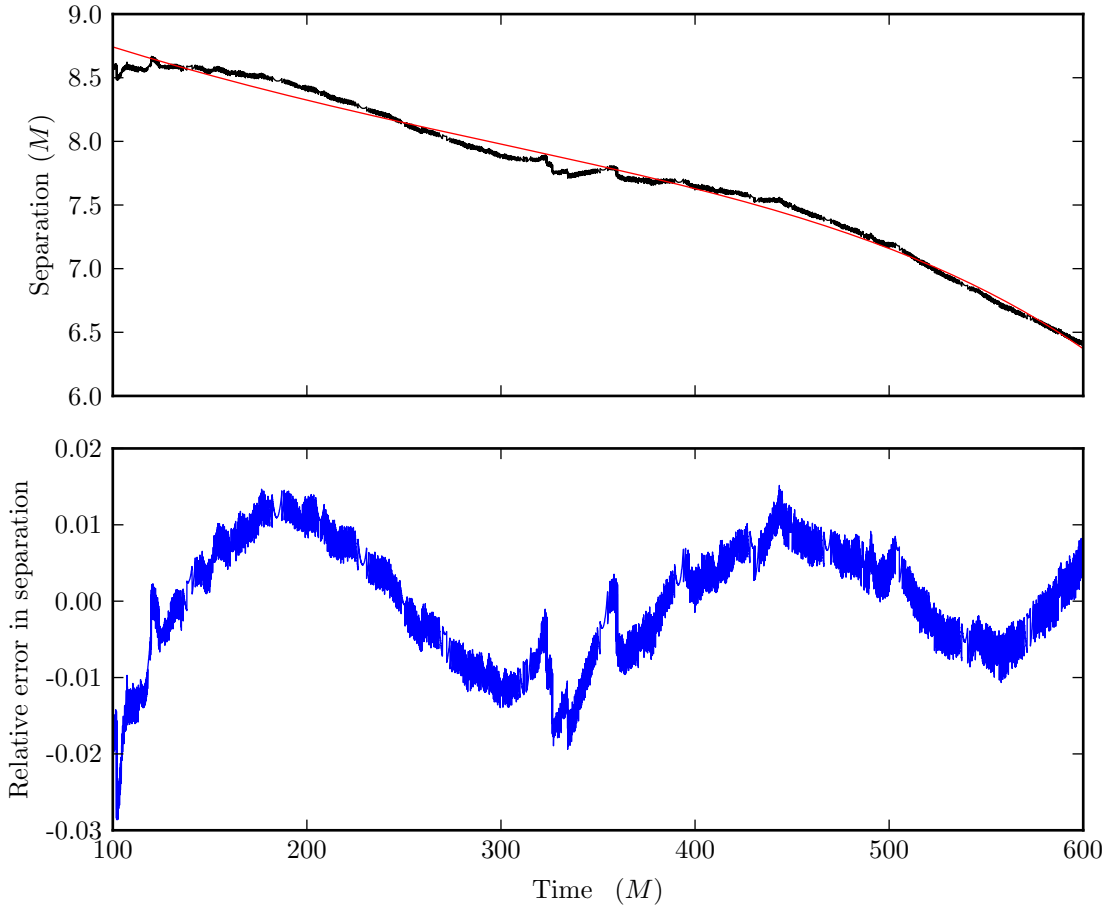


Figure 12: Like Figure 11, but for our mixed binary system.

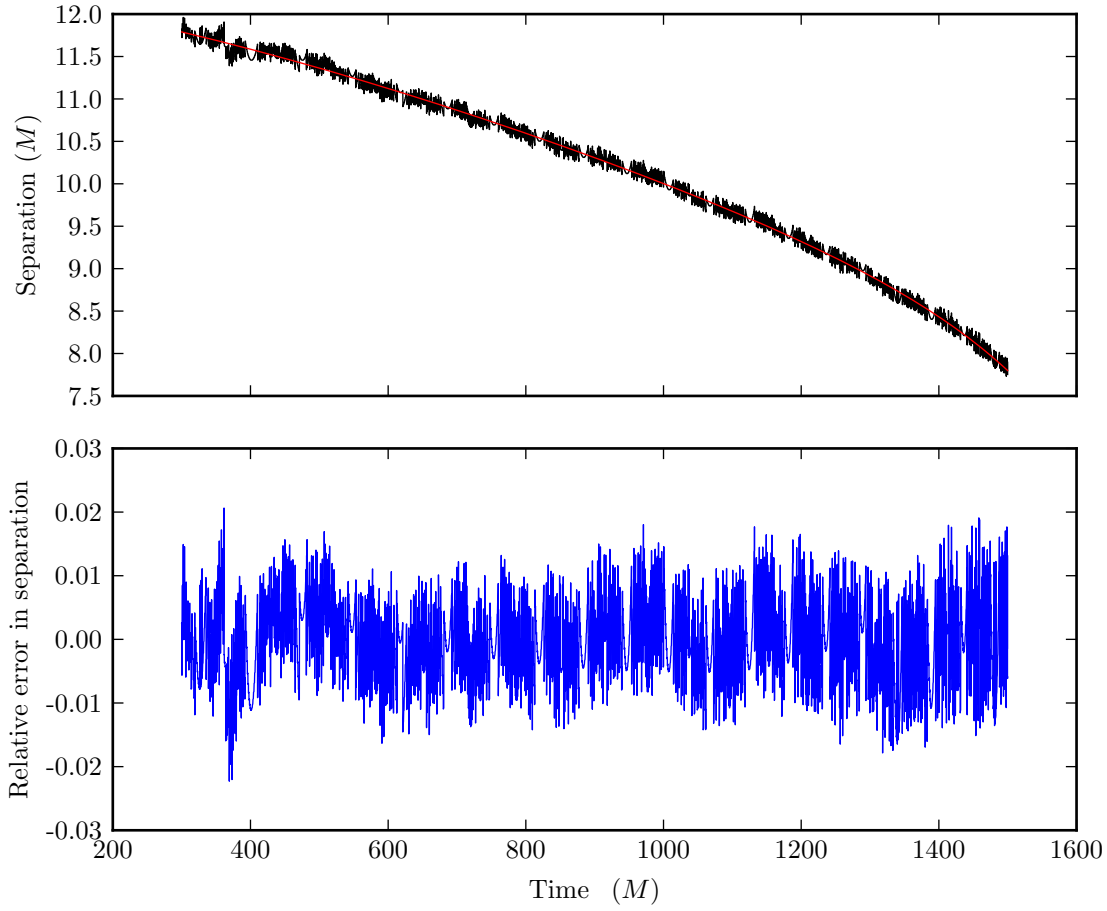


Figure 13: Like Figure 11, but for our spinning DNS system.

$$D_c(t) = A_1\sqrt{t-T} + A_2(t-T) + A_3(t-T)^{3/2} + A_4(t-T)^4 \qquad (7.12)$$

where T is the merger time, and the constants A_i are the fit parameters to be determined. The eccentricity can then be read off as the magnitude of the extrema. However, our data is quite noisy, which we attribute to our tracking of the star (which searches for the maximum rest mass density in a region, and will therefore only report positions that are grid points[1]) as well as the oscilations of central density, which may account for some of the higher frequency oscillations seen in the residuals.

In any event, our results are exhibited in Figures 11, 12, and 13. We measure eccentricity near 3% for each case. This is rather large compared to that resulting from the eccentricity reduction techniques of [50], for example, which are typically an order of magnitude lower.

7.3.6 Merger and collapse timescales

Our simulations show inconsistent differences in merger and collapse timescales. While some of this can be attributed to resolution, this may speak to a general deficiency due to spurious central density oscillations or merely differences due to gauge conditions.

Our best data comes from our nonspinning DNS simulations, for we have data from three different resolutions. Here, we see a difference in merger timescales vary by roughly $100M_{\rm tot}$ (or about 1.5 ms) and collapse timescales vary by $200M_{\rm tot}$ (3 ms) between our lowest and highest resolutions.

By comparison, the system in [5] shows a shorter merger timescale (6 ms compared to 10 ms) from 45 km coordinate separation. We believe fundamental differences in gauge could be at issue here and that comparison according to proper separation or orbital angular velocity might reduce the differences in these numbers, if such can be precisely quantified.

[1] [**?**] considered finding the center of rest mass instead and found that this reduced such noise considerably. This remains a promising option in the future.

In contrast, our collapse timescale is much shorter. The hypermassive neutron star (HMNS) formed by the merger of the two stars collapses to a black hole in 5 ms for our highest resolution simulation versus 10 ms for the reference system in [**?**]. Consequently, the reference system exhibits more cycles of gravitational waves from the HMNS compared to our system. The differences here could be attributed to the unphysical oscillations in density that are present in inspiral; we believe these may fundamentally alter the dynamics of merger.

We make no statements about comparing our spinning DNS system to the reference system in this respect, as the reference states no initial coordinate separation (only proper separation) and no particular initial orbital angular velocity (which we could, in principle, compare against). Hence, there is no basis for comparison of merger time, and in both their system and ours, the HMNS collapses almost imediately, due to the spin of the HMNS working in opposition of orbital angular momentum.

CHAPTER VIII

FUTURE WORK AND CONCLUSIONS

The work presented thus far is an early attempt to generate initial data for binary systems with neutron stars in a manner using the conformal transverse-traceless (CTT) decomposition, like in the use of Bowen-York data and puncture black holes. While this method shares in the simplicity of that approach on several levels—particularly the convenience of analytic solutions to the momentum constraint—our experiments have shown several lingering issues that will have to be improved upon in the future—if the method is to be competitive with the state of the art of quasiequilibrium initial data.

8.1. Central density oscillations and the nature of equilibrium

Perhaps the biggest challenge for this data going forward is the inherent difficulty of ensuring approximate equilibrium of neutron stars.

The conformal thin sandwich formalism is suited to constructing quasiequilibrium data due to the conditions it can impose on derivatives, and the approximate Killing vector can be used to define a time coordinate and time direction that is natural for a simulation. In our data, hydrostatic equilibrium cannot be guaranteed, and moreover, we neglect several effects that should distort the shape of a star under the assumption of spherical symmetry—in particular, the natural oblate shape a star should assume when rotating or moving with linear momentum, or the tidal forces exerted when under the influence of a binary companion. In particular, our measures of density fluctuation in single stars versus stars in binary systems indicate that tidal influences are significant: the density fluctuation goes from 3% to 10% or more.

There are a few possibilities for attacking this problem:

8.1.1 Use of post-Newtonian fluid solutions, or other models for the conformal momentum densities

Perhaps the most promising approach—in that it requires minimal modification to the overall method—is the use of a post-Newtonian fluid model from which to construct the conformal linear and angular momentum densities $\bar{\sigma}, \bar{\kappa}$.

Our approach to black hole binaries with Bowen-York data relies on post-Newtonian computations to evolve point particles to the initial separation from which we start simulation in full general relativity. A similar post-Newtonian estimate of the hydrodynamical variables—or even a direct estimate of the conformal momentum densities—could provide a far better density to base our computations upon. While the momentum densities for our generalized Bowen-type curvature are necessarily spherically symmetric about the centers of the sources, the conformal factor need not be, and this could be considered the ultimate source of distortions.

Whether our conformal momentum densities come from post-Newtonian approximations or other computational approaches, such would relax the conditions we used in Equation 5.40 and Equation 5.42—in which we chose conformal momentum densities to enforce constant linear or angular velocity for isolated stars.

Perhaps a more radical change in approach would be to consider general, asymmetric sources of linear and angular momentum. We discuss generating the extrinsic curvature for such sources in subsection 8.1.2, but we will tackle a broader question right now: is a post-Newtonian type of method feasible for simulating neutron stars—or neutron stars and black holes—to generate both their momenta at some close separation as well as a matter profile that could be used for extrinsic curvature?

We believe the answer is *yes*, for there is at least one example in the literature: a smoothed particle hydrodynamics, post-Newtonian code from Faber and Rasio [30]. While interest in this method seems to have petered out in recent years, the core idea

would be a great match for our method, and we would only need to simulate the long inspiral phase, rather than the more complicated dynamics of merger.

An open question, however, would be whether such a method—the use of a PN hydrodynamical simulation for inspiral, generating initial data for an NR simulation—would compare well with schemes to generate QE data.

8.1.2 Use of asymmetric sources

Bowen [18] derived a more general form of curvature for sources with no particular symmetries. This decomposes the source into conventional spherical harmonics, with radial functions describing those modes' amplitudes. Coupled with a post-Newtonian estimate of the hydrodynamical variables at some particular separation, a scheme could decompose a PN-based solution into modes and proceed from there.

Without going over Bowen's full derivation, we present merely Bowen's results: $\bar{A}^{ij}_L$ can be written in terms of two potentials—a vector potential V^i and a scalar potential θ such that

$$\bar{A}^{ij}_L = \partial^i V^j + \partial^j V^i - \frac{1}{2}\delta^{ij}\partial_k V^k - \frac{1}{2}\partial^i\partial^j\theta. \tag{8.1}$$

The two potentials can be written in terms of the following multipole tensors[1]:

$$Q^{i_0 i_1 \dots i_\ell} = \frac{(2\ell-1)!!}{\ell!}\int_M \bar{S}^{i_0} x^{\{i_1} x^{i_2} \dots x^{i_\ell\}}\, dV \tag{8.2}$$

$$C^{i_0 i_1 \dots i_\ell} = \frac{(2\ell-1)!!}{\ell!}\int_M r^2 \bar{S}^{i_0} x^{\{i_1} x^{i_2} \dots x^{i_\ell\}}\, dV, \tag{8.3}$$

where curly braces denote the symetric, traceless part of a tensor only. That is, $B_{\{ij\}} = (B_{ij} + B_{ji})/2 - B\delta_{ij}/3$.

The forms of these tensors Q, C unifies the linear and angular momentum descriptions: for a star with no linear momentum but nonzero total angular momentum, Q^i

[1] We will not reproduce the full forms of V, θ here.

evaluates to zero, and only C^i survives.

The resulting extrinsic curvature would then be flexible enough for arbitrary sources—albeit only to the level of multiple expansion tolerable for practical implementation.

However, Bowen himself only provided the exterior solutions for the potentials V^i, θ (though deriving interior solutions should not be difficult and would help prove the regularity of the extrinsic curvature at the center of a star, or at whatever coordinate origin might be used).

Furthermore, while this curvature can handle arbitrarily-shaped sources, it can give no guidance on what those sources should be to be in equilibrium or to be physically relevant. Nevertheless, the forms of the extrinsic curvature derived from this multipole decomposition would be more compatible with PN-based descriptions of stars.

8.1.3 Choice of gauge conditions

A remaining freedom we have is the choice of initial gauge conditions. For all our simulations, we used the UTB lapse $\alpha = \psi^{-2}$ and zero shift for the initial lapse and shift, respectively. More sophisticated gauge choices could yield better stability for the stars.

In point of fact, since the gauge conditions for QE data are exactly those of maximal slicing and minimal distortion (lapse and shift, respectively), it is possible to implement these conditions alongside CTT data, but only at the cost of coupling the gauge conditions to the rest of the initial data. This would render the convenience of an exact solution to the momentum constraint all but pointless.

A more subtle discussion involves the difference between (extended) CTS gauge conditions and the gauge conditions we imposed on our simulations.

Our simulations use the initial gauge condition $\alpha = 1/\psi^2$ and $\beta^i = 0$. These differ

substantially from extended CTS gauge conditions, and one area to explore going forward is how close we can get to CTS gauge conditions without spoiling the overall structure of the CTT equations.

A point of interest is that, for a suitable choice of lapse, the non-extended CTS equations in vacuum can reduce to the Bowen-York solution [32]. This solution relies solely on the choice of $\bar{\alpha} = 1/2$, or $\alpha = \psi^6/2$. Should a similar solution relating the Bowen-type curvature here and the CTS equations be derived, that would suggest a means to convert between CTS-style QE data and Bowen-type data. That being said, the gauge choice $\alpha = \psi^6/2$ is numerically unstable for Bowen-type data, as the lapse would then be infinite at the punctures. This may make the relationship between Bowen-type data and CTS data more of a curiousity than a useful tool.

Another avenue of investigation would be the difference between the lapse associated with our CTS TOV model and that used in simulations. If we could construct a lapse function that reduces to the TOV lapse in the large-separation, low-momentum, and low-spin limit, that may also help reduce spurious oscillations.

One choice might be to consider the densitized lapse a conformal quantity, just as $\bar{\rho}$ and $\bar{S}^i$ are.

8.2. Center of mass drift

The center of mass drift seen in our mixed binary simulations could be explained by inaccuracy in the method for estimating neutron star masses. In such a case, a larger mass ratio is more sensitive to errors in neutron star mass. A subsequent test of this idea would be an examination of binaries at several mass ratios, including unequal mass ratio double neutron star binaries.

Alternatively, this drift could be some kind of gauge effect and, therefore, the previous discussion about gauge choices could also apply here.

8.3. Comparison of initial data with quasuequilibrium methods

One difficulty in direct comparisons of initial data is the direct role the orbital angular velocity plays in quasiequilibrium data versus the lack of a direct measure of orbital angular velocity in CTT-based data. This creates a significant obstacle in comparing our initial data to QE data, particularly in comparing binding energy vs. orbital angular frequency. Orbital angular frequency is a preferable parameter to describe binary systems of a given mass ratio and composition (pair of objects) because it is a gauge invariant.

8.4. Eccentricity reduction

Our analysis of eccentricity in subsection 7.3.5 considers only the eccentricity resulting from the scheme presented, but alternative schemes often include explicit eccentricity-reduction steps. For example, the method of [50] uses short evolutions to estimate eccentricity, considering the inward radial velocities of the objects to be tunable parameters for QE data. Though that method is based in CTS formalism, the core idea could be applied to the inward radial momenta here.

Alternatively, the concept of a better post-Newtonian solver for generating initial momenta and/or matter distributions could be more in line with [35], which sought to avoid repeated evolutions to reduce eccentricity, and which is the basis for the binary black hole approach we initially sought to extend.

8.5. Final conclusions and remarks

We have offered a new method for constructing initial data for neutron stars in binary systems. The method is still in need of refining: central density oscillations have no obvious solution, eccentricity is at least an order of magnitude higher than would be competitive with other methods, and center of mass drift in unequal-mass binaries poses a significant obstacle to reliable gravitational wave extraction.

Nevertheless, it is our hope that the method proposed can be improved in the future, whether with some of the suggestions and ideas put forward here or by other means.

APPENDIX A

NOTATION AND CONVENTIONS

Throughout this document we commonly use Einstein summation over indices: for example, $a_\mu b^\mu \equiv \sum_{\mu=0}^{3} a_\mu b^\mu$. The summation range 0–3 applies for Greek indices; the range 1–3 applies for Latin indices. For those reasons, we may also refer to such indices as "spacetime" and "spatial" indices, respectively.

It's common in numerical relativity to adopt units such that $G = c = 1$. Use of these "geometricized units" leaves only one scale factor common to mass, length, and time. We will commonly use M to denote this scale factor, measuring lengths, durations, and masses all in M. This is convenient in numerical code, as physical quantities can be converted without need to keep track of factors of G or c. In recovering metric units, one can convert multiply a quantity measured in M by GM/c^2 to recover length, or GM/c^3 to recover a time, and the units used for G and c will generate kilometers or seconds or whatever other units desired. Finally, it is common to measure M itself in terms of solar masses, $M_\odot$.

REFERENCES

[1] Abbott, B. P., Abbott, R., Abbott, T. D., Abernathy, M. R., Acernese, F., Ackley, K., Adams, C., Adams, T., Addesso, P., Adhikari, R. X., and et al., "Observation of Gravitational Waves from a Binary Black Hole Merger," *Physical Review Letters*, vol. 116, no. 6, p. 061102, 2016.

[2] Alcubierre, M., *Introduction to 3+1 Numerical Relativity*. Oxford University Press, UK, 2008.

[3] Ansorg, M., Brügmann, B., and Tichy, W., "A single-domain spectral method for black hole puncture data," *Phys. Rev. D*, vol. 70, p. 064011, 2004.

[4] Arnowitt, R., Deser, S., and Misner, C. W., "Republication of: The dynamics of general relativity," *General Relativity and Gravitation*, vol. 40, no. 9, pp. 1997–2027, 2008.

[5] Baiotti, L., Giacomazzo, B., and Rezzolla, L., "Accurate evolutions of inspiralling neutron-star binaries: Prompt and delayed collapse to a black hole," Phys. Rev. D, vol. 78, p. 084033, Oct. 2008.

[6] Baiotti, L., Hawke, I., Montero, P. J., Löffler, F., Rezzolla, L., Stergioulas, N., Font, J. A., and Seidel, E., "Three-dimensional relativistic simulations of rotating neutron star collapse to a Kerr black hole," *Phys. Rev. D*, vol. 71, p. 024035, 2005.

[7] Baker, J. G., Campanelli, M., and Lousto, C. O., "The Lazarus project: A Pragmatic approach to binary black hole evolutions," *Phys. Rev. D*, vol. 65, p. 044001, 2002.

[8] Banyuls, F., Font, J. A., Ibáñez, J. M. l., Martí, J. M. l., and Miralles, J. A., "Numerical {3 + 1} General Relativistic Hydrodynamics: A Local Characteristic Approach," ApJ, vol. 476, pp. 221–231, Feb. 1997.

[9] Baumgarte, T. W., "Innermost stable circular orbit of binary black holes," Phys. Rev. D, vol. 62, p. 024018, July 2000.

[10] Baumgarte, T. W., Cook, G. B., Scheel, M. A., Shapiro, S. L., and Teukolsky, S. A., "Binary neutron stars in general relativity: Quasiequilibrium models," *Phys. Rev. Lett.*, vol. 79, pp. 1182–1185, Aug 1997.

[11] Baumgarte, T. W. and Shapiro, S. L., "Numerical integration of Einstein's field equations," Phys. Rev. D, vol. 59, p. 024007, Jan. 1999.

[12] Berger, M. J. and Oliger, J., "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of Computational Physics*, vol. 53, no. 3, pp. 484 – 512, 1984.

[13] Bernuzzi, S., Dietrich, T., Tichy, W., and Brügmann, B., "Mergers of binary neutron stars with realistic spin," Phys. Rev. D, vol. 89, p. 104021, May 2014.

[14] Bernuzzi, S., Thierfelder, M., and Brügmann, B., "Accuracy of numerical relativity waveforms from binary neutron star mergers and their comparison with post-newtonian waveforms," *Phys. Rev. D*, vol. 85, p. 104030, May 2012.

[15] Bona, C., Massó, J., Seidel, E., and Stela, J., "New Formalism for Numerical Relativity," *Physical Review Letters*, vol. 75, pp. 600–603, July 1995.

[16] Bonazzola, S., Gourgoulhon, E., and Marck, J.-A., "Numerical approach for high precision 3D relativistic star models," Phys. Rev. D, vol. 58, p. 104020, Nov. 1998.

[17] Bowen, J. M., "General form for the longitudinal momentum of a spherically symmetric source.," *General Relativity and Gravitation*, vol. 11, pp. 227–231, Oct. 1979.

[18] Bowen, J. M., "General solution for flat-space longitudinal momentum," *General Relativity and Gravitation*, vol. 14, no. 12, pp. 1183–1191, 1982.

[19] Bowen, J. M. and York, J. W., "Time-asymmetric initial data for black holes and black-hole collisions," *Phys. Rev. D*, vol. 21, pp. 2047–2056, Apr 1980.

[20] Brandt, S. and Brügmann, B., "A Simple Construction of Initial Data for Multiple Black Holes," *Physical Review Letters*, vol. 78, pp. 3606–3609, May 1997.

[21] Brown, J. D., Diener, P., Sarbach, O., Schnetter, E., and Tiglio, M., "Turduckening black holes: an analytical and computational study," *Phys. Rev. D*, vol. 79, p. 044023, 2009.

[22] Campanelli, M., Lousto, C. O., Marronetti, P., and Zlochower, Y., "Accurate Evolutions of Orbiting Black-Hole Binaries without Excision," *Physical Review Letters*, vol. 96, p. 111101, Mar. 2006.

[23] Campanelli, M. and Lousto, C. O., "Second order gauge invariant gravitational perturbations of a kerr black hole," *Phys. Rev. D*, vol. 59, p. 124022, May 1999.

[24] Carpet: Adaptive Mesh Refinement for the Cactus Framework.

[25] Colella, P. and Woodward, P. R., "The Piecewise Parabolic Method (PPM) for Gas Dynamical Simulations," *J. Comp. Phys.*, vol. 54, pp. 174–201, 1984.

[26] Cook, G. B., "Three-dimensional initial data for the collision of two black holes. ii. quasicircular orbits for equal-mass black holes," *Phys. Rev. D*, vol. 50, pp. 5025–5032, Oct 1994.

[27] Courant, R., Friedrichs, K., and Lewy, H., "Über die partiellen differenzengleichungen der mathematischen physik," *Mathematische Annalen*, vol. 100, no. 1, pp. 32–74, 1928.

[28] Dietrich, T., Moldenhauer, N., Johnson-McDaniel, N. K., Bernuzzi, S., Markakis, C. M., Brügmann, B., and Tichy, W., "Binary neutron stars with generic spin, eccentricity, mass ratio, and compactness: Quasi-equilibrium sequences and first evolutions," Phys. Rev. D, vol. 92, p. 124007, Dec. 2015.

[29] Einstein, A., "Approximative Integration of the Field Equations of Gravitation," *Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.)*, vol. 1916, pp. 688–696, 1916.

[30] Faber, J. A. and Rasio, F. A., "Post-newtonian sph calculations of binary neutron star coalescence: Method and first results," *Phys. Rev. D*, vol. 62, p. 064012, Aug 2000.

[31] Grandclément, P., Gourgoulhon, E., and Bonazzola, S., "Binary black holes in circular orbits. II. Numerical methods and first results," Phys. Rev. D, vol. 65, p. 044021, Feb. 2002.

[32] Hannam, M., Husa, S., Brügmann, B., González, J. A., and Sperhake, U., "Beyond the Bowen York extrinsic curvature for spinning black holes," *Classical and Quantum Gravity*, vol. 24, pp. S15–S24, June 2007.

[33] Hotokezaka, K., Kiuchi, K., Kyutoku, K., Muranushi, T., Sekiguchi, Y.-i., Shibata, M., and Taniguchi, K., "Remnant massive neutron stars of binary neutron star mergers: Evolution process and gravitational waveform," *Phys. Rev. D*, vol. 88, p. 044026, Aug 2013.

[34] Hulse, R. A. and Taylor, J. H., "Discovery of a pulsar in a binary system," ApJ, vol. 195, pp. L51–L53, Jan. 1975.

[35] Husa, S., Hannam, M., González, J. A., Sperhake, U., and Brügmann, B., "Reducing eccentricity in black-hole binary evolutions with initial parameters from post-Newtonian inspiral," Phys. Rev. D, vol. 77, p. 044037, Feb. 2008.

[36] Husa, S., Hinder, I., and Lechner, C., "Kranc: a Mathematica application to generate numerical codes for tensorial evolution equations," *Comput. Phys. Commun.*, vol. 174, pp. 983–1004, 2006.

[37] Kidder, L. E., Scheel, M. A., Teukolsky, S. A., Carlson, E. D., and Cook, G. B., "Black hole evolution by spectral methods," Phys. Rev. D, vol. 62, p. 084032, Oct. 2000.

[38] Kidder, L. E., Scheel, M. A., and Teukolsky, S. A., "Extending the lifetime of 3d black hole computations with a new hyperbolic system of evolution equations," *Phys. Rev. D*, vol. 64, p. 064017, Aug 2001.

[39] "Kranc: Kranc assembles numerical code."

[40] Kvaalen, E., "A faster broyden method," *BIT Numerical Mathematics*, vol. 31, no. 2, pp. 369–372, 1991.

[41] Liang, S., Hudak, P., and Jones, M., "Monad transformers and modular interpreters," in *In Proceedings of the 22nd ACM Symposium on Principles of Programming Languages. ACMPress*, 1995.

[42] Löffler, F., Faber, J., Bentivegna, E., Bode, T., Diener, P., Haas, R., Hinder, I., Mundim, B. C., Ott, C. D., Schnetter, E., Allen, G., Campanelli, M., and Laguna, P., "The Einstein Toolkit: A Community Computational Infrastructure for Relativistic Astrophysics," *Class. Quantum Grav.*, vol. 29, no. 11, p. 115001, 2012.

[43] "McLachlan, a public BSSN code."

[44] Mösta, P., Mundim, B. C., Faber, J. A., Haas, R., Noble, S. C., Bode, T., Löffler, F., Ott, C. D., Reisswig, C., and Schnetter, E., "GRHydro: A new open source general-relativistic magnetohydrodynamics code for the Einstein Toolkit," *Classical and Quantum Gravity*, vol. 31, no. 1, p. 015005, 2014.

[45] Nakamura, T., Oohara, K., and Kojima, Y., "General relativistic collapse to black holes and gravitational waves from black holes," *Progress of Theoretical Physics Supplement*, vol. 90, pp. 1–218, 1987.

[46] Newman, E. and Penrose, R., "An approach to gravitational radiation by a method of spin coefficients," *Journal of Mathematical Physics*, vol. 3, no. 3, pp. 566–578, 1962.

[47] Oohara, K.-i. and Nakamura, T., "Three-dimensional initial data of colliding neutron stars," *Progress of Theoretical Physics*, vol. 81, no. 2, pp. 360–369, 1989.

[48] Oppenheimer, J. R. and Volkoff, G. M., "On massive neutron cores," *Phys. Rev.*, vol. 55, pp. 374–381, Feb 1939.

[49] Peters, P. C., "Gravitational radiation and the motion of two point masses," *Phys. Rev.*, vol. 136, pp. B1224–B1232, Nov 1964.

[50] Pfeiffer, H. P., Brown, D. A., Kidder, L. E., Lindblom, L., Lovelace, G., and Scheel, M. A., "Reducing orbital eccentricity in binary black hole simulations," *Classical and Quantum Gravity*, vol. 24, pp. S59–S81, June 2007.

[51] Pfeiffer, H. P. and York, J. W., "Extrinsic curvature and the einstein constraints," *Phys. Rev. D*, vol. 67, p. 044022, Feb 2003.

[52] Richardson, L. F., "The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 210, no. 459-470, pp. 307–357, 1911.

[53] Sarbach, O., Calabrese, G., Pullin, J., and Tiglio, M., "Hyperbolicity of the baumgarte-shapiro-shibata-nakamura system of einstein evolution equations," *Phys. Rev. D*, vol. 66, p. 064002, Sep 2002.

[54] Schnetter, E., Hawley, S. H., and Hawke, I., "Evolutions in 3-D numerical relativity using fixed mesh refinement," *Class. Quantum Grav.*, vol. 21, pp. 1465–1488, 2004.

[55] Shibata, M., Kyutoku, K., Yamamoto, T., and Taniguchi, K., "Gravitational waves from black hole-neutron star binaries: Classification of waveforms," *Phys. Rev. D*, vol. 79, p. 044030, Feb 2009.

[56] Shibata, M. and Nakamura, T., "Evolution of three-dimensional gravitational waves: Harmonic slicing case," *Phys. Rev. D*, vol. 52, pp. 5428–5444, Nov 1995.

[57] Takami, K., Rezzolla, L., and Baiotti, L., "Spectral properties of the post-merger gravitational-wave signal from binary neutron stars," Phys. Rev. D, vol. 91, p. 064001, Mar. 2015.

[58] Taylor, J. H., Fowler, L. A., and McCulloch, P. M., "Measurements of general relativistic effects in the binary pulsar PSR 1913+16," Nature, vol. 277, pp. 437–440, Feb. 1979.

[59] The LIGO Scientific Collaboration and the Virgo Collaboration, "GW150914: First results from the search for binary black hole coalescence with Advanced LIGO," *ArXiv e-prints*, Feb. 2016.

[60] Thorne, K. S., "Multipole expansions of gravitational radiation," *Rev. Mod. Phys.*, vol. 52, pp. 299–339, Apr 1980.

[61] van Meter, J. R., Baker, J. G., Koppitz, M., and Choi, D.-I., "How to move a black hole without excision: Gauge conditions for the numerical evolution of a moving puncture," *Phys. Rev. D*, vol. 73, p. 124011, Jun 2006.

[62] Wadler, P., "Monads for functional programming," in *Advanced Functional Programming, First International Spring School on Advanced Functional Programming Techniques-Tutorial Text*, (London, UK, UK), pp. 24–52, Springer-Verlag, 1995.

[63] "Whisky – relativistic hydrodynamics and magnetohydrodynamics."

[64] York, Jr., J. W., "Kinematics and dynamics of general relativity," in *Sources of Gravitational Radiation* (Smarr, L. L., ed.), pp. 83–126, 1979.

[65] York, Jr., J. W., "Conformal "Thin-Sandwich" Data for the Initial-Value Problem of General Relativity," *Physical Review Letters*, vol. 82, pp. 1350–1353, Feb. 1999.

[66] York, J. W., "Conformally invariant orthogonal decomposition of symmetric tensors on riemannian manifolds and the initialvalue problem of general relativity," *Journal of Mathematical Physics*, vol. 14, no. 4, pp. 456–464, 1973.

VITA

Michael Christopher Clark was born in Marietta, GA on April 23, 1988. An only child, Michael is the son of John and Kimberly. He attended Georgia Tech for his bachelor's degree in physics, working primarily with Edward Conrad on x-ray diffraction of graphene, which included a visit to Argonne National Labs, and received his degree in the spring of 2010. He continued at Georgia Tech for his graduate work, transitioning to numerical relativity under Pablo Laguna in early 2011.