

Fast hyperbolic relaxation elliptic solver for numerical relativity: Conformally flat, binary puncture initial data

Thiago Assumpção^{1,2,*} Leonardo R. Werneck^{3,1,2,†}
Terrence Pierre Jacques^{1,2,‡} and Zachariah B. Etienne^{3,1,2,§}

¹*Department of Physics and Astronomy, West Virginia University,
Morgantown, West Virginia 26506, USA*

²*Center for Gravitational Waves and Cosmology, West Virginia University,
Chestnut Ridge Research Building, Morgantown, West Virginia 26505, USA*

³*Department of Physics, University of Idaho, Moscow, Idaho 83843, USA*



(Received 3 November 2021; accepted 24 March 2022; published 18 May 2022)

We introduce `NRPyElliptic`, an elliptic solver for numerical relativity (NR) built within the `NRPy+` framework. As its first application, `NRPyElliptic` sets up conformally flat, binary black hole (BBH) puncture initial data (ID) on a single numerical domain, similar to the widely used `TwoPunctures` code. Unlike `TwoPunctures`, `NRPyElliptic` employs a hyperbolic relaxation scheme, whereby arbitrary elliptic partial differential equations (PDEs) are trivially transformed into a hyperbolic system of PDEs. As consumers of NR ID generally already possess expertise in solving hyperbolic PDEs, they will generally find `NRPyElliptic` easier to tweak and extend than other NR elliptic solvers. When evolved forward in (pseudo)time, the hyperbolic system exponentially reaches a steady state that solves the elliptic PDEs. Notably `NRPyElliptic` accelerates the relaxation waves, which makes it many orders of magnitude faster than the usual constant wave speed approach. While it is still $\sim 12x$ slower than `TwoPunctures` at setting up full-3D BBH ID, `NRPyElliptic` requires only $\approx 0.3\%$ of the runtime for a full BBH simulation in the Einstein Toolkit. Future work will focus on improving performance and generating other types of ID, such as binary neutron stars.

DOI: [10.1103/PhysRevD.105.104037](https://doi.org/10.1103/PhysRevD.105.104037)

I. INTRODUCTION

To date the LIGO/Virgo gravitational wave (GW) observatories have detected dozens of binary black hole (BBH) mergers [1], and numerical relativity (NR) BBH simulations form a cornerstone of the ensuing data analyses. Such simulations build from formulations of the general relativistic (GR) field equations [2–10] that decompose GR into an initial value problem in $(3 + 1)$ dimensions. These formulations generally rewrite the GR field equations as a set of time evolution and constraint equations, similar to Maxwell’s equations in differential form [11]. Thus, so long as one is provided initial data (ID) that satisfy the Einstein constraints (elliptic PDEs), the evolution equations (hyperbolic PDEs) can propagate them forward in time to construct the spacetime.

Construction of NR ID for realistic astrophysical scenarios is typically a complex and highly specialized task (see [12–15] for excellent reviews). A wide range of approaches have been employed by different groups to set up ID for realistic astrophysical scenarios, including the

use of finite difference [16–21], spectral [22–36], and Galerkin [37,38] methods, which are then combined with special numerical techniques to solve the associated elliptic PDEs. Notably, this generally requires a different skill set than those associated with solving the (hyperbolic) evolution equations, so experts in setting up ID for NR are rarely experts in solving the evolution equations, and vice versa.

`NRPyElliptic` is a new, extensible elliptic solver that sets up initial data for numerical relativity using the same numerical methods employed for solving hyperbolic evolution equations. Specifically, `NRPyElliptic` implements the hyperbolic relaxation method of [39] to solve complex nonlinear elliptic PDEs for NR ID. The hyperbolic PDEs are evolved forward in (pseudo)time, resulting in an exponential relaxation of the arbitrary initial guess to a steady state that coincides with the solution of the elliptic system. `NRPyElliptic` solves these equations on highly efficient numerical grids exploiting underlying symmetries in the physical scenario. To this end, `NRPyElliptic` is built within the SymPy-based [40] `NRPy+` code-generation framework [41,42], which facilitates the solution of hyperbolic PDEs on Cartesian-like, spherical-like, cylindrical-like, or bispherical-like numerical grids. For the purposes of setting up BBH puncture ID, `NRPyElliptic` makes use of the latter.

*ta0082@mix.wvu.edu

†leonardo@uidaho.edu

‡tp0052@mix.wvu.edu

§zetienne@uidaho.edu

The choice of appropriate numerical grids is critically important, as setting up binary compact object ID requires numerically resolving many orders of magnitude in length scale, from the sharp gravitational fields near each compact object, to the nearly flat fields far away. If a constant wave speed is chosen in the hyperbolic relaxation method (as in [39]), the Courant-Friedrichs-Lewy (CFL)-constrained global timestep for the relaxation will be many orders of magnitude smaller than the time required for a relaxation wave to cross the numerical domain. This poses a significant problem as hyperbolic relaxation methods must propagate the relaxation waves across the entire numerical domain several times to reach convergence. Thus hyperbolic relaxation solvers are generally far slower than elliptic solvers based on specialized numerical methods.

`NRPyElliptic` solves the hyperbolic relaxation PDEs on a single bispherical-like domain, enabling us to increase the local relaxation wave speed in proportion to the local grid spacing without violating the CFL condition. As grid spacing in our coordinate system grows exponentially away from the two-coordinate foci and toward the outer boundary, the relaxation waves accelerate exponentially toward the outer boundary, increasing the solver’s overall speedup over a constant wave speed implementation by many orders of magnitude. In fact the resulting performance boost enables `NRPyElliptic` to be useful for setting up high-quality, full-3D BBH puncture ID, though it is still 12x slower than the widely used pseudospectral `TwoPunctures` [31] BBH puncture ID solver.¹

Like `TwoPunctures`, `NRPyElliptic` adopts the conformal transverse-traceless decomposition [12,43–45] to construct puncture ID for two BHs. In this paper we present both 2D and full 3D validation tests, which demonstrate that `NRPyElliptic` yields identical results to `TwoPunctures` as numerical resolution is increased in both codes.

We also embed `NRPyElliptic` into an Einstein Toolkit [46–48] module (“thorn”), called `NRPyEllipticET`, which enables the generated ID to be interpolated onto Cartesian adaptive mesh refinement (AMR) grids within the Einstein Toolkit. To demonstrate that `NRPyElliptic` ID are of high fidelity, we first generate 3D BBH puncture ID with both `NRPyEllipticET` and the `TwoPunctures` Einstein Toolkit thorns at comparable accuracy, then evolve the ID forward in time through inspiral, merger, and ringdown using the Einstein Toolkit infrastructure, and finally show that the results of these simulations are virtually indistinguishable.

While existing algorithms for elliptic equations can be used to solve a general class of problems, their numerical implementations can be rather problem specific (see, e.g., Ref. [49] and references therein). For instance, the `TwoPunctures` code employs the biconjugate gradient stabilized (BiCSTAB) method, which requires a specific

preconditioner for numerical efficiency [31]. In contrast, hyperbolic relaxation solvers can be trivially used for other elliptic problems by simply replacing the right-hand sides of the resulting hyperbolic PDEs while preserving other aspects of the numerical implementation. In fact, the generality of the hyperbolic relaxation method has already been demonstrated in [39], where it was used to produce ID for many different scenarios of interest, such as scalar fields, Tolman-Oppenheimer-Volkoff (TOV) stars, and binary neutron stars (BNSs). In this work we will focus our discussion on BBH puncture ID, and further evidence of the extensibility of `NRPyElliptic` will be presented in forthcoming papers to generate e.g., BNS ID.

The remainder of this paper is organized as follows. Section II introduces the puncture ID formalism, the hyperbolic relaxation method, and our implementation of Sommerfeld (radiation) boundary conditions. In Sec. III we discuss the details of our numerical implementation, including choice of coordinate system and implementation of a grid spacing-dependent wave speed. We present 2D (axisymmetric) and full 3D validation tests, as well as results from a BBH evolution of our full 3D ID in Sec. IV. We conclude in Sec. V and discuss future work.

II. BASIC EQUATIONS

Throughout this paper we adopt geometrized units, in which $G = c = 1$, and Einstein summation convention such that repeated Latin (Greek) indices imply a sum over all three spatial (all four spacetime) components.

Consider the $(3 + 1)$ decomposition of the spacetime metric, with line element

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt). \quad (1)$$

Here, α is the lapse function, β^i is the shift vector, and γ_{ij} is the 3-metric.

It is useful to define a conformally related 3-metric $\tilde{\gamma}_{ij}$ via

$$\gamma_{ij} = \psi^4 \tilde{\gamma}_{ij}, \quad (2)$$

where the scalar function ψ is known as the conformal factor. We adorn geometric quantities associated with $\tilde{\gamma}_{ij}$ with a tilde diacritic. For instance, the Christoffel symbols associated with $\tilde{\gamma}_{ij}$ are computed using

$$\tilde{\Gamma}^k{}_{ij} = \frac{1}{2} \tilde{\gamma}^{lk} (\tilde{\gamma}_{li,j} + \tilde{\gamma}_{lj,i} - \tilde{\gamma}_{ij,l}). \quad (3)$$

Likewise, $\tilde{\nabla}_i$ is the associated conformal covariant derivative and \tilde{R}_{ij} the Ricci tensor. All geometric quantities compatible with the physical 3-metric γ_{ij} are written without tildes.

In the limit of vacuum (e.g., BBH) spacetimes, the Hamiltonian and momentum constraint equations can be written as [12]

¹Benchmark tests performed using an Intel Xeon Gold 6230 20-Core CPU for initial data of comparable quality.

$$\mathcal{H} \equiv R + K^2 - K_{ij}K^{ij} = 0, \quad (4)$$

$$\mathcal{M}^i \equiv \nabla_j(K^{ij} - \gamma^{ij}K) = 0, \quad (5)$$

where K_{ij} is the extrinsic curvature and $K \equiv \gamma^{ij}K_{ij}$ is the mean curvature. Setting up ID for vacuum spacetimes in numerical relativity generally involves solving these constraints, which exist as second-order nonlinear elliptic PDEs.

For the purposes of this paper, we will focus on the puncture ID formalism, in which a set of simplifying assumptions is applied to these constraints, known as the conformal transverse-traceless (CTT) decomposition (see e.g., [12]). For completeness we next apply the CTT approach to Eqs. (4) and (5) to derive the constraint equations solved in this paper by NRPyElliptic.

A. Puncture initial data formalism

To arrive at the CTT decomposition, we first rewrite the extrinsic curvature as

$$K_{ij} = A_{ij} + \frac{1}{3}\gamma_{ij}K, \quad (6)$$

where A_{ij} is the trace-free part of K_{ij} . The conformal counterpart of A_{ij} is defined through the relation

$$A_{ij} \equiv \psi^{-2}\tilde{A}_{ij}. \quad (7)$$

The CTT decomposition splits \tilde{A}_{ij} into a symmetric trace-free part \tilde{M}^{ij} and a longitudinal part $(\tilde{\mathbb{L}}V)^{ij}$,

$$\tilde{A}^{ij} = (\tilde{\mathbb{L}}V)^{ij} + \tilde{M}^{ij}, \quad (8)$$

where the longitudinal operator $\tilde{\mathbb{L}}$ is defined via

$$(\tilde{\mathbb{L}}V)^{ij} \equiv \tilde{\nabla}^i V^j + \tilde{\nabla}^j V^i - \frac{2}{3}\tilde{\gamma}^{ij}\tilde{\nabla}_l V^l. \quad (9)$$

Inserting these CTT quantities into the constraint equations [Eqs. (4) and (5)] yields the generic CTT Hamiltonian and momentum constraint equations (see, e.g., [45,50] and references therein)

$$\tilde{\nabla}^2\psi - \frac{1}{8}\psi\tilde{R} - \frac{1}{12}\psi^5K^2 + \frac{1}{8}\psi^{-7}\tilde{A}_{ij}\tilde{A}^{ij} = 0, \quad (10)$$

$$\tilde{\Delta}_{\mathbb{L}}V^i - \frac{2}{3}\psi^6\tilde{\nabla}^i K + \tilde{\nabla}_j\tilde{M}^{ij} = 0, \quad (11)$$

where \tilde{R} is the conformal Ricci scalar and the operator $\tilde{\Delta}_{\mathbb{L}}$ is defined as

$$\tilde{\Delta}_{\mathbb{L}}V^i \equiv \tilde{\nabla}_j(\tilde{\mathbb{L}}V)^{ij} = \tilde{\nabla}^2V^i + \frac{1}{3}\tilde{\nabla}^i(\tilde{\nabla}_jV^j) + \tilde{R}^i_jV^j. \quad (12)$$

The degrees of freedom in this formulation include the choice of \tilde{M}^{ij} , K , and $\tilde{\gamma}_{ij}$. Here we consider puncture ID, which assumes maximal slicing ($K = 0$), asymptotic flatness ($\psi|_{r \rightarrow \infty} = 1$), and conformal flatness

$$\tilde{\gamma}_{ij} = \hat{\gamma}_{ij}, \quad (13)$$

where $\hat{\gamma}_{ij}$ is the flat spatial metric. In addition the assumption $\tilde{M}^{ij} = 0$ is made, yielding Hamiltonian and momentum constraint equations of the form [31]

$$\hat{\nabla}^2\psi + \frac{1}{8}\psi^{-7}\tilde{A}_{ij}\tilde{A}^{ij} = 0, \quad (14)$$

$$\hat{\nabla}^2V^i + \frac{1}{3}\hat{\nabla}^i(\hat{\nabla}_jV^j) + \hat{R}^i_jV^j = 0, \quad (15)$$

where $\hat{\nabla}_i$ is the covariant derivative compatible with $\hat{\gamma}_{ij}$. Bowen and York [51] showed that the momentum constraint is solved for a set of N_p punctures with a closed-form expression for the extrinsic curvature. This expression can be written in terms of \vec{V} as follows:

$$\vec{V} = \sum_{n=1}^{N_p} \left(-\frac{7}{4|\vec{x}_n|}\vec{P}_n - \frac{\vec{x}_n \cdot \vec{P}_n}{4|\vec{x}_n|^3}\vec{x}_n + \frac{1}{|\vec{x}_n|^3}\vec{x}_n \times \vec{S}_n \right), \quad (16)$$

where $\vec{x}_n = (x_n - x, y_n - y, z_n - z)$, \vec{P}_n , and \vec{S}_n are the displacement relative to the origin [i.e., $(x, y, z) = (0, 0, 0)$], linear momentum, and spin angular momentum of puncture n , respectively.

The Hamiltonian constraint equation [Eq. (14)] must be solved numerically, but ψ becoming singular at the location of each puncture could spoil the numerical solution. Early attempts excised the singular terms from the computational domain (see, e.g., [12]), but modern approaches generally follow [52] in splitting the conformal factor into a singular and a nonsingular piece,

$$\psi = \psi_{\text{singular}} + u \equiv 1 + \sum_{n=1}^{N_p} \frac{m_n}{2|\vec{x}_n|} + u, \quad (17)$$

where m_n is the bare mass of the n th puncture. The Hamiltonian constraint equation, which can then be solved for the nonsingular part u , reads

$$\hat{\nabla}^2u + \frac{1}{8}\tilde{A}_{ij}\tilde{A}^{ij}(\psi_{\text{singular}} + u)^{-7} = 0, \quad (18)$$

since the Laplacian of the singular piece vanishes.

B. Hyperbolic relaxation method

We now describe the basic hyperbolic relaxation method of [39]. Consider the system of elliptic equations

$$\mathcal{L}_E \vec{u} - \vec{\rho} = 0, \quad (19)$$

where \mathcal{L}_E is an elliptic operator, \vec{u} is the vector of unknowns, and $\vec{\rho}$ is the vector of source terms. The hyperbolic relaxation method replaces Eq. (19) with the hyperbolic system of equations

$$\partial_t^2 \vec{u} + \eta \partial_t \vec{u} = c^2 (\mathcal{L}_E \vec{u} - \vec{\rho}), \quad (20)$$

where η is an exponential damping parameter (with units of $1/t$ [53]) and c is the wave speed. The variable t behaves as a time variable in this hyperbolic system of equations and is referred to as a relaxation (as opposed to physical) time. As noted in [39], the damping parameter η that maximizes dissipation is dictated by the length scale of the grid domain when the wave speed is constant. On the other hand, a spatially varying wave speed, as introduced in Sec. III B, gives rise to a new scale to the problem—the relaxation-wave crossing time, T_{RC} (refer to Appendix B for details). Through numerical experimentation, we found that the choice η that minimizes the required relaxation time follows a power law given by Eq. (36).

If appropriate boundary conditions are chosen, when Eq. (20) is evolved forward in (pseudo)time, the damping ensures that a steady state is eventually reached exponentially fast such that $\partial_t u \rightarrow 0$ and $\partial_t^2 u \rightarrow 0$. Thus u relaxes to a solution to the original elliptic problem. To this end, we adopt Sommerfeld (outgoing radiation) boundary conditions (BCs) for spatial boundaries, as described in Sec. II C; what remains is a choice of initial conditions. As this is a relaxation method, any smooth choice should suffice. For simplicity, in this work we set trivial initial conditions $\vec{u} = \partial_t \vec{u} = \vec{0}$.

To complete our expression of these equations in preparation for a full numerical implementation, we rewrite Eq. (20) as a set of two first-order (in time) PDEs

$$\begin{aligned} \partial_t \vec{u} &= \vec{v} - \eta \vec{u}, \\ \partial_t \vec{v} &= c^2 (\mathcal{L}_E \vec{u} - \vec{\rho}), \end{aligned} \quad (21)$$

so that the method of lines (Sec. III) can be immediately used to propagate the solution forward in (pseudo)time until a convergence criterion has been triggered (indicating numerical errors associated with the solution to the elliptic equation are satisfactorily small).

As a simple example, consider Poisson's equation, for which $\mathcal{L}_E \vec{u} = \mathcal{L}_E u = \nabla^2 u = u^i{}_{,i}$. This PDE can be easily made covariant ("comma goes to semicolon rule"),

$$\hat{\nabla}^2 u = u^i{}_{;i} = \rho, \quad (22)$$

where $\hat{\nabla}_i$ is the covariant derivative compatible with $\hat{\gamma}_{ij}$. In this way, the Laplace operator is expanded as

$$\hat{\nabla}^i \hat{\nabla}_i u = \hat{\gamma}^{ij} \hat{\nabla}_i \hat{\nabla}_j u = \hat{\gamma}^{ij} (\partial_i \partial_j u - \hat{\Gamma}^k{}_{ij} \partial_k u), \quad (23)$$

with $\hat{\Gamma}^k{}_{ij}$ the Christoffel symbols associated with $\hat{\gamma}_{ij}$. Poisson's equation is then written as the system

$$\begin{aligned} \partial_t u &= v - \eta u, \\ \partial_t v &= c^2 (\hat{\nabla}^2 u - \rho). \end{aligned} \quad (24)$$

Writing the PDEs covariantly enables the hyperbolic relaxation method to be applied in coordinate systems that properly exploit near symmetries. For this purpose we adopt a reference metric $\hat{\gamma}_{ij}$, which is chosen to be the flat spatial metric in the given coordinate system we are using. In this way, single compact object ID can be solved in spherical or cylindrical coordinates (using spherical or cylindrical reference metrics respectively), and binary ID can be solved in bispherical-like coordinates. Once the appropriate coordinate system is chosen, all covariant derivatives are expanded in terms of partial derivatives and the Christoffel symbols as prescribed in Eq. (23).

Truly the power of the hyperbolic relaxation method is its easy and immediate extension to complex, nonlinear elliptic PDEs by simply modifying the right-hand sides of Eq. (24). Case in point: ID for two punctures are constructed by solving Eq. (18) for u . This elliptic PDE is nonlinear, but is trivially embedded within the hyperbolic relaxation prescription via²

$$\begin{aligned} \partial_t u &= v - \eta u, \\ \partial_t v &= c^2 \left[\hat{\nabla}^2 u + \frac{1}{8} \tilde{A}_{ij} \tilde{A}^{ij} (\psi_{\text{singular}} + u)^{-7} \right]. \end{aligned} \quad (25)$$

Note that just like in the case of Poisson's equation, $\hat{\nabla}^2 u$ is expanded as in Eq. (23).

C. Boundary conditions

Similar to both the hyperbolic relaxation method implemented in [39] and the Einstein Toolkit BC driver NewRad [46,47,54], spatial BCs are applied to the time derivatives of the evolved fields instead of the fields directly. Consequently the desired BC is only satisfied by the steady state solution.

For example, assume that at $\partial\Omega$, the boundary of our numerical domain, we wish to impose Dirichlet BCs of the form

²Recall that in the previous section we adopted the tilde for the conformal metric and the hat for the flat metric, consistent with the general convention in the literature. Due to the choice of conformal flatness, both can be used interchangeably here, $\tilde{\gamma}_{ij} = \hat{\gamma}_{ij}$.

$$\begin{aligned}\vec{u}|_{\partial\Omega} &= \vec{a}, \\ \vec{v}|_{\partial\Omega} &= \vec{b},\end{aligned}\quad (26)$$

for some constant vectors \vec{a} and \vec{b} . In our implementation, these would be imposed as

$$\begin{aligned}\partial_t \vec{u}|_{\partial\Omega} &= \vec{u} - \vec{a}, \\ \partial_t \vec{v}|_{\partial\Omega} &= \vec{v} - \vec{b}.\end{aligned}\quad (27)$$

Upon reaching the steady state, $\partial_t u|_{\partial\Omega} = 0 = \partial_t v|_{\partial\Omega}$, and we recover the desired BCs.

When applying the hyperbolic relaxation method to solve the Einstein constraint equations, outgoing radiation BCs are most appropriate, as they allow the outgoing relaxation wave fronts to pass through the boundaries of the numerical domain with minimal reflection.

Radiation (Sommerfeld) BCs generally assume that near the boundary each field f behaves as an outgoing spherical wave, and our implementation follows the implementation within `NewRad`, building upon the ansatz,

$$f = f_0 + \frac{w(r-ct)}{r} + \frac{C}{r^2}, \quad (28)$$

where $f_0 = \lim_{r \rightarrow \infty} f$, $w(r-ct)/r$ satisfies the spherical wave equation for an outgoing spherical wave, and C/r^2 models higher-order radial corrections.

Just as in the case of Dirichlet BCs, we apply Sommerfeld BCs to the time derivative of the fields. Appendix A walks through the full derivation for applying Sommerfeld BCs to any field $\partial_t f$, as well as its numerical implementation. Based on Eq. (A10), Sommerfeld BCs for a generic hyperbolic relaxation of solution vector \vec{u} takes the form

$$\begin{aligned}\partial_t \vec{u}|_{\partial\Omega} &= -\frac{c}{r} [r \partial_r \vec{u} + (\vec{u} - \vec{u}_0)] + \frac{\vec{k}_u}{r^3}, \\ \partial_t \vec{v}|_{\partial\Omega} &= -\frac{c}{r} [r \partial_r \vec{v} + (\vec{v} - \vec{v}_0)] + \frac{\vec{k}_v}{r^3},\end{aligned}\quad (29)$$

where \vec{k}_u and \vec{k}_v are constant vectors computed at each boundary point for each field within the \vec{u} and \vec{v} vectors using Eq. (A15).

III. NUMERICAL IMPLEMENTATION

`NRPyElliptic` exists as both a standalone code and an Einstein Toolkit module (“thorn”), `NRPyEllipticET`. `NRPyEllipticET` incorporates the standalone code into the Einstein Toolkit, solving the elliptic PDE entirely within `NRPyElliptic`’s `NRPy+`-based infrastructure. Once the solution has been found, `NRPyEllipticET` uses the Einstein Toolkit’s built-in (third-order Hermite) interpolation infrastructure to interpolate the

solution from its native, bispherical-like grids to the Cartesian AMR grids used by the Einstein Toolkit. From there, the data can be evolved forward in time using any of the various BSSN or CCZ4 Einstein Toolkit thorns. Both standalone and Einstein Toolkit thorn versions of the `NRPyElliptic` code are fully documented in pedagogical JUPYTER notebooks. Henceforth, we will describe our implementation of the standalone version.

Our implementation of Eq. (25) within `NRPyElliptic` leverages the `NRPy+` framework [41,42] to convert these expressions, written symbolically using `NRPy+`’s Einstein-like notation, into highly optimized C-code kernels (`SymPy` [40] serves as `NRPy+`’s computer algebra system backend). Notably `NRPy+` supports the generation of such kernels with single instruction, multiple data (SIMD) intrinsics and common subexpression elimination (CSE). Like `TwoPunctures`, `NRPyElliptic` currently supports OpenMP parallelization [55] and both codes run on single computational nodes. Further `NRPy+` supports arbitrary-order finite-difference kernel generation, and we use 10th order to approximate all spatial derivatives in this work. The time evolution is performed using the method of lines infrastructure within `NRPy+`, choosing its fourth-order (explicit) Runge-Kutta implementation (RK4).

`NRPy+` supports a plethora of different reference metrics, enabling us to solve our covariant hyperbolic PDEs (Eq. 25) in a large variety of Cartesian-like, spherical-like, cylindrical-like, or bispherical-like coordinate systems. This in turn enables the user to fully take advantage of symmetries or near symmetries of any given problem. For example, for problems involving near-spherical symmetry we have used spherical-like coordinates (e.g., log-radial spherical coordinates). In this work, we make use of the prolate spheroidal-like (i.e., bispherical-like) coordinate system in `NRPy+` called `SinhSymTP`, described in detail in Sec. III A. This allows us to solve the elliptic problem for two puncture black hole initial data within a single domain, similar to the `TwoPunctures` code.

Note that the wave speed c appearing e.g., in Eq. (25), need not be constant. In curvilinear coordinates where the grid spacing is not constant, the CFL stability criterion remains satisfied if the wave speed is adjusted in proportion to the local grid spacing. As the grid spacing in the `SinhSymTP` coordinates adopted here grows exponentially with distance from the strong-field region, the wave speed grows exponentially as well. As a result, relaxation waves accelerate exponentially to the outer boundary, significantly speeding up the convergence to the solution of the elliptic PDE. Our implementation of this technique is detailed in Sec. III B.

A. Coordinate system

Like `TwoPunctures`, `NRPyElliptic` adopts a modified version of prolate spheroidal (PS) coordinates when setting up two-puncture ID. However, these coordinate systems are

distinct both from each other and from PS coordinates. Here we elucidate the differences and similarities.

Consider first PS coordinates (μ, ν, φ) , which are related to Cartesian coordinates (x, y, z) via [56]

$$\begin{aligned} x &= a \sinh \mu \sin \nu \cos \varphi, \\ y &= a \sinh \mu \sin \nu \sin \varphi, \\ z &= (a^2 \sinh^2 \mu + a^2)^{1/2} \cos \nu. \end{aligned} \quad (30)$$

Here, $\mu \in [0, \infty)$, $\nu \in [0, \pi)$, $\varphi \in [0, 2\pi)$, and the two foci of the coordinate system are located at $z = \pm a$.

TwoPunctures [31] adopts a PS-like coordinate system, which is written in terms of coordinate variables $A \in [0, 1)$, $B \in [-1, 1]$, and $\varphi \in [0, 2\pi)$. TwoPunctures coordinates are related to Cartesian via³

$$\begin{aligned} x &= b \frac{2A}{1-A^2} \frac{1-B^2}{1+B^2} \sin \varphi, \\ y &= b \frac{2A}{1-A^2} \frac{1-B^2}{1+B^2} \cos \varphi, \\ z &= b \frac{A^2+1}{A^2-1} \frac{2B}{1+B^2}. \end{aligned} \quad (31)$$

The two foci of TwoPunctures coordinates are situated at $z = \pm b$. Of note, the coordinate A is compactified with $|x|, |y|, |z| \rightarrow \infty$ as $A \rightarrow 1$. Similar to the term $\sin \nu$ (where $\nu \in [0, \pi)$) in PS coordinates, the $(1-B^2)/(1+B^2)$ (where $B \in [-1, 1]$) term is a concave down curve with a maximum of 1 at the midpoint of the range of B and zeroes at the endpoints $B = \pm 1$. Unlike PS coordinates, however, the coordinate system is not periodic in the variable B .

NRPyElliptic adopts the NRPy+ PS-like coordinate system SinhSymTP (x_1, x_2, x_3) with $x_1 \in [0, 1]$, $x_2 \in [0, \pi]$, and $x_3 \in [-\pi, \pi]$. These are related to PS coordinates via

$$\begin{aligned} a \sinh \mu &= \tilde{r} \equiv \mathcal{A} \frac{\sinh(x_1/w)}{\sinh(1/w)}, \\ \nu &= x_2, \\ \varphi &= x_3, \end{aligned} \quad (32)$$

with $\tilde{r} \in [0, \mathcal{A}]$. Introducing the parameter b , we obtain

$$\begin{aligned} x &= \tilde{r} \sin(x_2) \cos(x_3), \\ y &= \tilde{r} \sin(x_2) \sin(x_3), \\ z &= (\tilde{r}^2 + b^2)^{1/2} \cos(x_2). \end{aligned} \quad (33)$$

Given inputs for w , \mathcal{A} (domain size), and the foci parameter b , NRPy+ samples coordinates (x_1, x_2, x_3) uniformly when setting up numerical grids. The foci exist at $z = \pm b$, and

³We swap the x and z coordinates of [31] to simplify comparison.

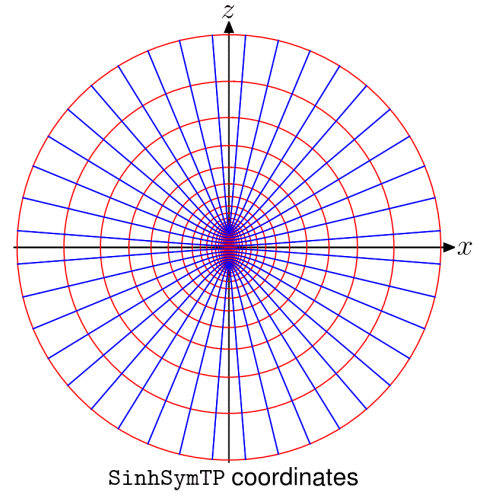


FIG. 1. Curves of constant x_1 (red) and constant x_2 (blue) using a cell-centered grid structure with $x_3 = 0$.

grid point density near the foci can be increased or decreased by decreasing or increasing w , respectively. Contrast this to PS coordinates, in which there is no such parameter to adjust the focusing of gridpoints near foci. In fact when the foci separation is increased in PS coordinates, the density of gridpoints decreases in proportion.

As illustrated in Fig. 1, like PS and TwoPunctures coordinates, SinhSymTP coordinates become spherical in the region far from the foci. Note also that regular spherical coordinates (with a nonuniform radial coordinate) are fully recovered by setting $b = 0$. As with TwoPunctures, NRPyEllipticET possesses the option to rotate the coordinate system, to situate the punctures on either the x -axis ($x = \pm b$) or the z -axis ($z = \pm b$).

For all cases considered here, the outer boundary is set to 10^6 (i.e., $\mathcal{A} = 10^6$), and the grid point spacing parameter w is set to 0.07. We set b so that the foci match the punctures' positions and, when interpolating the ID to the Einstein Toolkit grids, we adjust the origin of the coordinate system to coincide with the center of mass of the punctures, as is conventional.

B. Wave speed

To propagate the hyperbolic system of equations forward in (pseudo)time from the chosen initial conditions, we make use of NRPy+'s method of lines implementation. Specifically we choose the explicit fourth-order Runge-Kutta (RK4) method. As this is an explicit method, and we use three dimensions in space, the steps in time Δt are constrained by the CFL inequality,

$$\frac{c \Delta t}{\Delta s_{\min}} \leq \mathcal{C}_0, \quad (34)$$

where c is the local wave speed, and \mathcal{C}_0 is the CFL factor, which depends on the explicit time stepping method and

the dimensionality of the problem. Empirically we find that $C_0 = 0.7$ ensures both stability and large time steps for both 2D and 3D cases presented in this paper, with one exception: when the 3D case is pushed to very high resolution. In the single highest-resolution 3D case in this work, we find C_0 must be lowered to 0.55 for stability.

Further, Δs_{\min} is the minimum proper distance between neighboring points in our curvilinear coordinate system,

$$\Delta s_{\min} = \min(h_1 \Delta x_1, h_2 \Delta x_2, h_3 \Delta x_3), \quad (35)$$

where h_i and Δx_i are the i th scale factor and grid spacing of the flat space metric, respectively.⁴

The global relaxation time step Δt_{glob} is given by the minimum value of Δs_{\min} on our numerical grid. As we adopt prolate spheroidal-like coordinates, the global Δs_{\min} occurs precisely at the foci of the coordinate system. At this point, for simplicity we set the wave speed $c = 1$. As this is merely a relaxation (as opposed to a physical) wave speed, we increase c in proportion to the local Δs_{\min} , which grows exponentially away from the foci. In this way we maintain satisfaction of the CFL inequality while greatly improving the performance of the relaxation method. Details and implications of our implementation are described in Appendix B.

C. Choice of damping parameter η

Sweeping through values of η in early tests of our code (using different grid parameters than those adopted here), we found that setting $\eta = 12.5/M$ was optimal for minimizing the time to convergence of the relaxation scheme. After completing the tests, we later found that this is value of η is not optimal for the grids and resolutions we have chosen in this work. As a result, our core benchmark in Sec. IV B below, comparing the performance of NRPyElliptic with TwoPunctures, is slower by roughly a factor of 2 as compared to the optimum η for that grid choice, which would be closer to $\eta = 18/M$. We note that this suboptimal choice does not influence our relaxed solution, as every choice of η that leads to a stable evolution yields that same relaxed solution.

To find the optimal value of η , we perform a linear sweep through possible values in a range $\eta = 1$ through 20, in intervals of 0.25 (recall η has units of inverse mass, $1/M$, and here when quoting values of η we always choose $M = 1$). In the steady-state regime, the L^2 -norm of the relative error, E (see Sec. IV for details), becomes constant. Therefore, once we have chosen a trial value of η , we perform the relaxation until $|dE/dt| < \delta$ for n consecutive iterations, where $\delta \sim 10^{-4} - 10^{-6}$ and $n \sim 10-100$ (i.e., we relax until the L^2 -norm of the relative error has reached its

⁴In orthogonal, curvilinear coordinate systems, such as the ones supported by NRPy+, we have $h_i = \sqrt{\gamma_{ii}}$ (no summation implied).

minimum, constant value). The optimum value of η is that value that results in the fewest relaxation iterations before the L^2 -norm of the relative error has plateaued.

As η has units of $1/t$, it is natural to inquire whether the optimum damping parameter, η_{optimum} , is inversely proportional to the relaxation-wave crossing time, T_{RC} . We applied the aforementioned optimum η search to a variety of grids, each with its own T_{RC} [in the range $T_{\text{RC}} \in (0.24, 0.99)$], finding that the optimum η obeys the following power law,⁵

$$M\eta_{\text{optimum}} \approx \frac{6.9}{(T_{\text{RC}}/M)^{0.79}} - 0.7. \quad (36)$$

IV. RESULTS

Validation of NRPyElliptic is performed in two stages. First, we generate initial data (ID) for a given physical scenario with the widely used TwoPunctures [31] code, increasing resolution on the TwoPunctures grids until roundoff error dominates its numerical solution of Eq. (18), u . We refer to this high-resolution result as the trusted solution. Second, we generate the same ID with NRPyElliptic, and demonstrate that its results approach the trusted solution at the expected convergence rate.

We repeat this procedure twice; first for an axisymmetric case of two equal-mass BHs with spin vectors collinear with their separation vector, and second for a full 3D case involving a GW150914-like unequal-mass, quasicircular, spinning BBH system. To demonstrate the fidelity of the latter case, we first generate NRPyElliptic and TwoPunctures ID at similar levels of accuracy. Then, using the Einstein Toolkit [46–48] we evolve the ID through inspiral, merger, and ringdown, and compare the results. Finally, we note that $M = M_+ + M_-$ is defined as the sum of individual ADM masses of the punctures—see Eq. (83) of Ref. [31].

A. Axisymmetric initial data

In this test, we generate initial data (ID) for a scenario symmetric about the z -axis; two equal-mass punctures at rest, with spin components $\vec{S}_{\pm} = \pm 0.4M_{\pm}^2 \hat{z}$ and initial positions $z_{\pm} = \pm 6M$.⁶

We start by constructing the trusted solution u , against which we will compare results from NRPyElliptic. The trusted solution is generated by increasing the resolution of the TwoPunctures numerical grids until roundoff errors dominate. As we adopt double-precision arithmetic, this occurs when

⁵For this fit, we generated three-dimensional initial data with physical parameters as described in Sec. IV B (with $M = 1$ at resolutions ranging from $32^2 \times 16$ to $128^2 \times 16$). As the wave speed depends on grid spacing, the relaxation-light crossing time is affected by the resolution.

⁶The bare mass of each puncture is $m = 0.456428$ and the total ADM mass is $M_{\text{ADM}} = 0.979989M$.

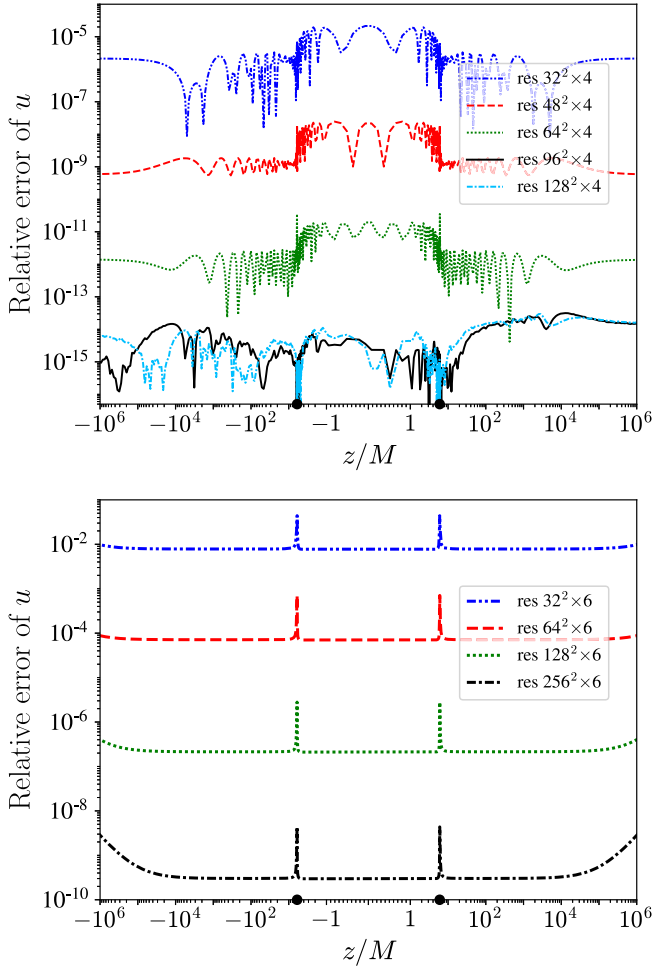


FIG. 2. Calibration of TwoPunctures and NRPyElliptic solutions, axisymmetric ID study. Top: Relative errors between a super-high resolution ($196^2 \times 4$) and lower-resolution results from the TwoPunctures code. Bottom: Calibration of NRPyElliptic solutions at various resolutions ($N_1 \times N_2 \times 6$), comparing against the trusted solution (i.e., the TwoPunctures result at $96^2 \times 4$ resolution). The horizontal axis is logarithmic in the range $|z| > 1$ and linear otherwise. Further, the black dots on the horizontal axes denote the puncture BH positions (also the locations of the coordinate foci). TwoPunctures data were rotated so that the punctures and the foci lie on the z axis.

relative errors reach levels of roughly 10^{-14} . The top panel of Fig. 2 indicates that this occurs at a TwoPunctures grid resolution of $N_A \times N_B \times N_\phi = 96^2 \times 4$.⁷

After obtaining this $96^2 \times 4$ trusted solution, we next compare it against NRPyElliptic at various resolutions in the bottom panel of the figure. To allow for a pointwise

⁷To ensure we reach the level where roundoff errors dominate, the maximum number of iterations of the Newton-Raphson method (Newton_maxit parameter) is set to 10, and the tolerance (Newton_tol parameter) is to 1×10^{-16} , for the axisymmetric case. Also we fix $N_\phi = 4$, following an axisymmetric example in Ref. [31].

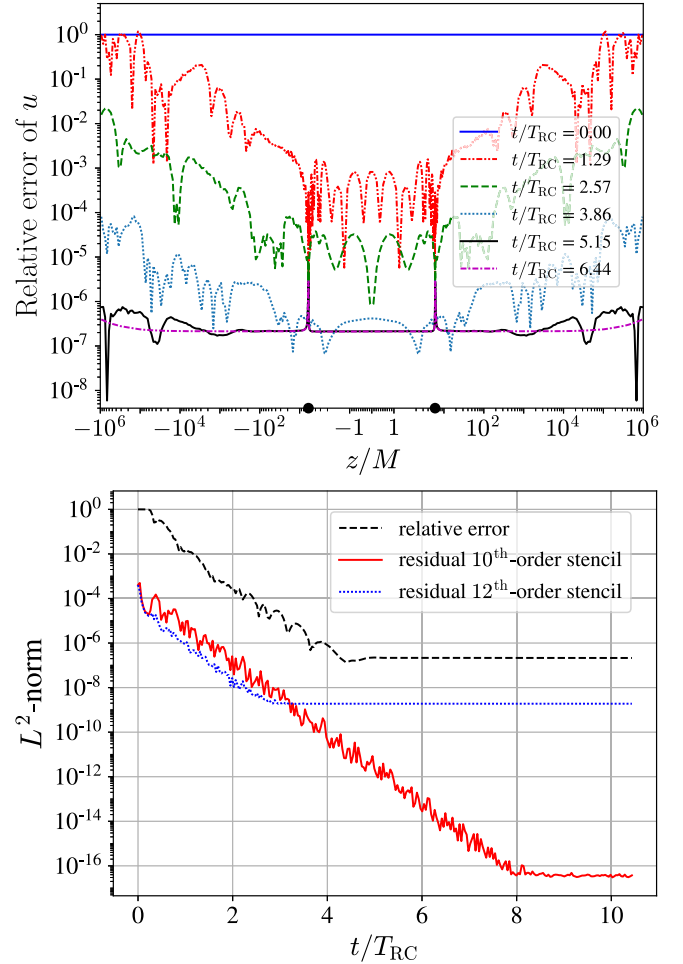


FIG. 3. Axisymmetric ID study. Top: Snapshots of the relative error between the NRPyElliptic solution at $128^2 \times 4$ resolution and the trusted solution at different times. TwoPunctures data were rotated so that the punctures and the foci lie on the z axis. Bottom: Black dashed curve: L^2 -norm of the same relative error as the top panel, computed within a sphere with radius $r = 100M$ and as a function of time. Blue (dotted) and red (solid) curves: L^2 -norms of residuals [second term in Eq. (18)] computed within the same sphere at two different finite-difference orders as a function of time.

comparison, we use the spectral interpolator in TwoPunctures [57] to evaluate the TwoPunctures solution at every point on our grid. Notice that the numerical errors drop by roughly 2^9 each time the resolution is doubled, indicating that numerical convergence is dominated by our choice of tenth-order finite-difference (FD) stencils. That perfect tenth-order convergence is not obtained is unsurprising, as our outer boundary condition is approximate and implements its own sixth-order FD stencils.⁸

⁸NRPy currently requires the minimum number of grid points in any given direction to be even and larger than $\text{FDorder}/2$, where FD_order order of the finite-difference scheme. Thus we set $N_3 = 6$ despite the system being axisymmetric. This requirement may be relaxed in the future.

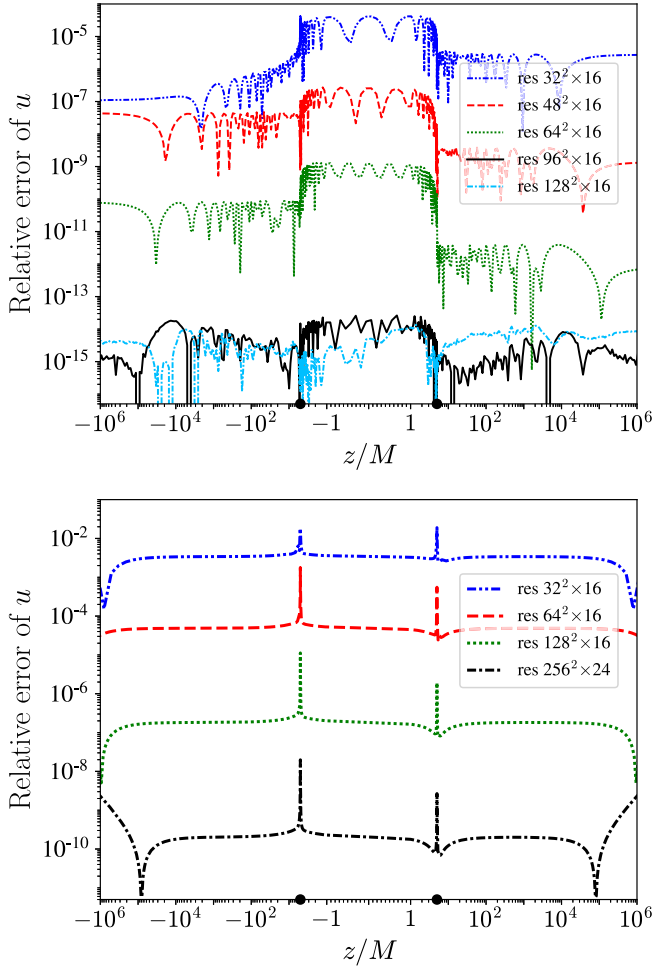


FIG. 4. Same as Fig. 2, but for GW150914-like, full-3D BBH initial data.

The exponential convergence in time of the hyperbolic relaxation method can be readily seen in Fig. 3. In the top panel snapshots of the relative error at resolution $128^2 \times 6$ are shown as a function of relaxation-wave crossing times (RCTs) (i.e., the time required for a relaxation wave to cross the numerical domain; see Appendix B). During the first RCT, the errors decrease (increase) in the region near (away from) the punctures and subsequently drop exponentially in time. Numerical errors are consistently smaller in the region near the origin due to the extreme grid focusing there. Notice that after ~ 4 RCTs the relative errors between the punctures have already reached the same order of magnitude as that of the fully relaxed solution, and the remaining relaxation time (40% of the total) is spent decreasing the relative errors away from the punctures.

Coincident with the exponential relaxation along the z -axis, the L^2 -norm of both the relative errors and residual [second term in the left-hand side of Eq. (18)], computed inside a sphere of radius $100M$ around the origin (situated at the center of mass), displays a steady exponential drop,

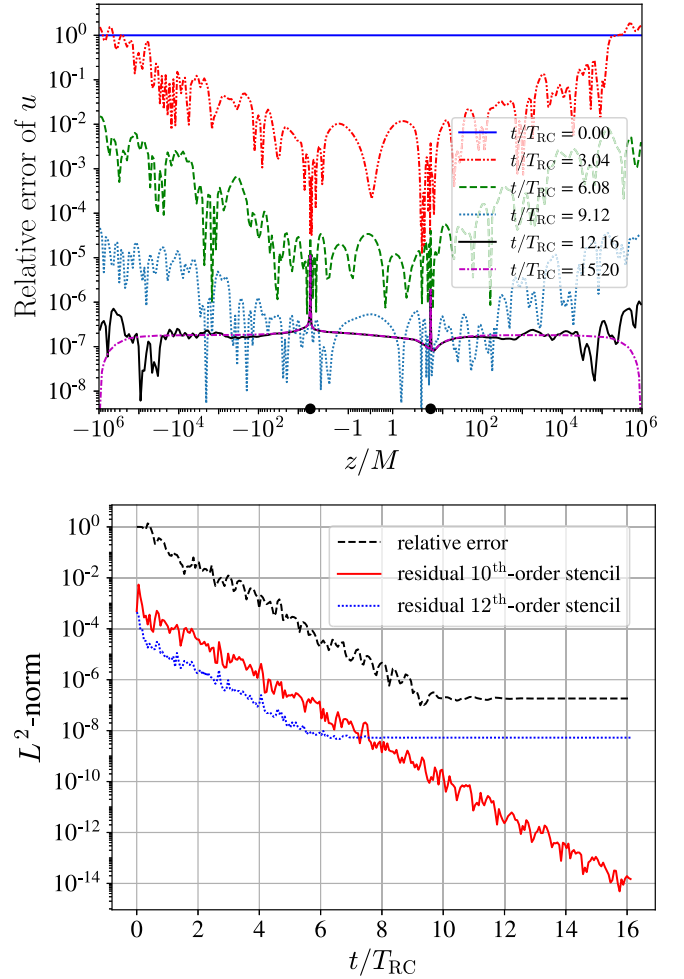


FIG. 5. Same as Fig. 3, but for GW150914-like, full-3D BBH initial data, with NRPyElliptic resolution of $128^2 \times 16$.

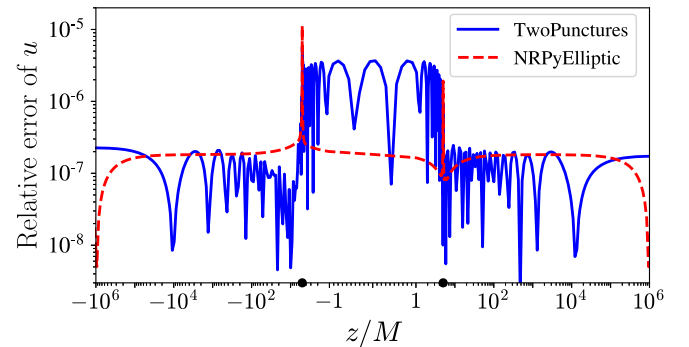


FIG. 6. Initial data quality of the GW150914-like BBH 3D scenario evolved with the Einstein Toolkit, showing the relative error of u against the trusted TwoPunctures solution. Here TwoPunctures and NRPyElliptic ID used resolutions of $38^2 \times 16$ and $128^2 \times 16$, respectively. Note that TwoPunctures data were rotated so that the punctures are positioned on the z axis.

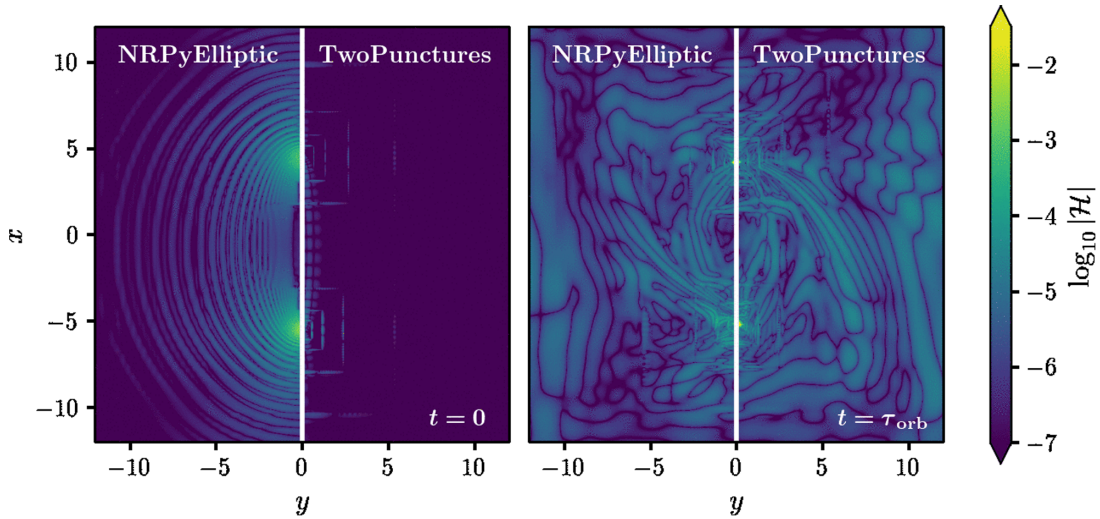


FIG. 7. Hamiltonian constraint violation $\log_{10} |\mathcal{H}|$, as defined in Eq. (4), in orbital plane after interpolation to AMR grids at $t = 0$ (left) and after one orbit (right). Note that NRPyElliptic data were rotated so that the punctures are positioned on the x axis.

as illustrated in the bottom panel of Fig. 3. We find that when the FD order for computing the residual matches the one used by the relaxation algorithm, the L^2 -norm of the residual continues to drop even after the steady state has been reached, until roundoff errors dominate this measure of residual. This indicates that the residual has become overspecialized to the approximation adopted for the FD operators. To ameliorate this, we also compute the residual using twelfth-order stencils, finding this measure of the residual to have more consistent behavior with the true error (i.e., the relative error against the trusted solution). We repeated this analysis at eighth and fourteenth finite-differencing order and found the same qualitative behavior as twelfth order.

B. Full 3D initial data

The second test consists of two punctures with mass ratio $q = 36/29$, in a quasicircular orbit emulating the GW150914 event [58]. The punctures are located at $z = \pm 5M$, with spin components $\vec{S}_+ = 0.31M_+^2 \hat{y}$ and $\vec{S}_- = -0.46M_-^2 \hat{y}$.⁹

The validation procedure for both TwoPunctures and NRPyElliptic follows the prescription used in the axisymmetric case. Since the full-3D system does not possess axial symmetry, the number of grid points along the azimuthal direction is no longer minimal. As indicated in the top panel of Fig. 4, the trusted TwoPunctures solution has resolution $96^2 \times 16$, and we found that using $N_\phi > 16$ does not lower

the relative errors further. In NRPyElliptic (bottom panel of Fig. 4), the optimal resolution in the azimuthal direction was also found to be $N_\phi = 16$, except for when the highest resolution ($256^2 \times 24$) was chosen. We find this solution to have smaller relative errors when compared to the $256^2 \times 16$ resolution data (not shown), and requires a lower CFL-factor of $\mathcal{C}_0 = 0.55$ for numerical stability. With truncation errors dominated by sampling in the x_1 and x_2 directions, we find roughly 9th-order convergence for this case as well.

As expected, the relaxation to the steady state has exponential convergence in (pseudo)time as depicted in both panels of Fig. 5. Although the relaxation requires a larger number of RCTs in the full-3D case, its qualitative behavior is the same.

Next, to demonstrate the fidelity of full-3D initial data generated by NRPyElliptic, we first generate ID for the GW150914-like BBH scenario using both NRPyElliptic and TwoPunctures. These ID are chosen to be of comparable quality, as shown in Fig. 6. Precise numerical parameters for the ID exactly follow the GW150914 Einstein Toolkit gallery example [31,46,59–67], for which complete results are freely available online.¹⁰ Notably the TwoPunctures code required slightly less than a minute to generate the ID (we input the bare masses directly), while NRPyElliptic required about 10 minutes, for a difference in performance of $\approx 12\times$.¹¹

After generation, the ID from both codes are interpolated onto the evolution grids, which consist of a set of

⁹The bare (local ADM) masses of the punctures are $m_+ = 0.518419$ ($M_+ = 0.553846$) and $m_- = 0.391936$. ($M_- = 0.446154$). The total ADM mass is $M_{\text{ADM}} = 0.989946M$. To elicit a quasicircular orbit, linear momenta are set to $\vec{P}_\pm = \pm P_\phi \hat{x} \pm P_r \hat{z}$, where $P_\phi = 9.53 \times 10^{-2}$ and $P_r = -8.45 \times 10^{-4}$, in code units.

¹⁰See <https://einstein toolkit.org/gallery/bbh/index.html> for more details.

¹¹Benchmarks were performed on a 16-core AMD Ryzen 9 3950X CPU; the speed-up factor was found to be very similar on other CPUs as well. Moreover, the damping parameter was not set to the optimum value of $\eta = 18/M$, and if it were the difference in performance would drop to only $\approx 6\times$.

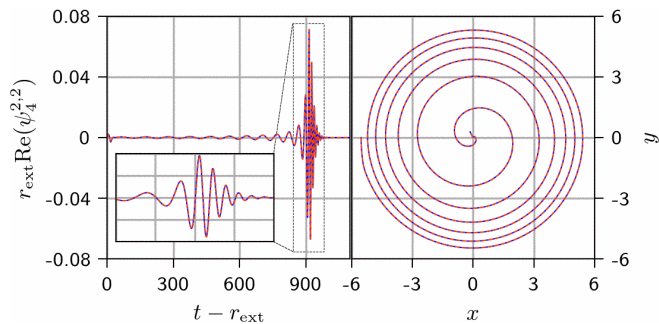


FIG. 8. Simulation results with ID by TwoPunctures (solid, blue) and NRPyElliptic (dashed, orange-red). Left: Dominant mode ($\ell = 2$, $m = 2$) of the Weyl scalar ψ_4 at extraction radius $r_{\text{ext}} = 500M$. Right: Trajectory of the less-massive BH.

CARPET [61]-managed Cartesian AMR patches in the strong field region, with resolution around the punctures of $\approx M/52$. These AMR patches are surrounded by a LLAMA [62]-managed cubed spheres grid in the weak-field region. When interpolating the ID onto these grids, TwoPunctures adopts a high-accuracy spectral interpolator (see e.g., [57]), and NRPyElliptic uses the third-order Hermite polynomial interpolator from the AEILocalInterp thorn. As our choice of Hermite interpolation order results in far less accuracy, it introduces error to the Hamiltonian constraint violation at time $t = 0$ on the orbital plane, as shown in the left panel of Fig. 7.

We then evolve these ID forward in time using the McLachlan BSSN thorn [65]. Notice that after the first orbital period the Hamiltonian constraint violations on the orbital plane are essentially identical (right panel of Fig. 7), indicating that numerical errors associated with the evolution quickly dominate ID errors.

Finally, to demonstrate that NRPyElliptic's 3D data are of sufficiently high fidelity, we demonstrate that when they are evolved forward in time the results are indistinguishable from an evolution of TwoPunctures ID at comparable initial accuracy. For instance, in Fig. 8 we show the trajectory of the less-massive BH and the dominant mode ($\ell = 2$, $m = 2$) of the Weyl scalar ψ_4 , for both evolutions. We find that quantities extracted from the evolution of NRPyElliptic ID to be in excellent agreement with results from evolution of ID from the trusted TwoPunctures code. Although not shown, the trajectory of the more-massive BH for both simulations are also visually indistinguishable, as are other, higher-order ψ_4 modes.

V. CONCLUSIONS AND FUTURE WORK

NRPyElliptic is an extensible ID code for numerical relativity that recasts nonlinear elliptic PDEs as covariant, hyperbolic PDEs. To this end it adopts the hyperbolic relaxation method of [39], in which the (pseudo)time evolution of the hyperbolic PDEs exponentially relaxes to a steady state consistent with the solution to the elliptic

problem. That standard hyperbolic methods are used to solve the elliptic problem is beneficial, as consumers of numerical relativity ID generally already have expertise in solving hyperbolic PDEs. Thus the learning curve is significantly lowered for core users of NRPyElliptic, enabling them to build on existing expertise to modify and extend the solver.

NRPyElliptic leverages NRPy+'s reference metric infrastructure to solve the hyperbolic/elliptic PDEs in a wide variety of Cartesian-like, spherical-like, cylindrical-like, and bispherical-like coordinate systems. As its first application, in NRPyElliptic we solve a nonlinear elliptic PDE to set up two-puncture ID for numerical relativity. Similar to the TwoPunctures [31] code, NRPyElliptic solves the elliptic PDE for the Hamiltonian constraint in a prolate spheroidal-like coordinate system. But unlike the one-parameter coordinate system used in prolate-spheroidal coordinates or TwoPunctures coordinates, NRPyElliptic adopts NRPy+'s SinhSymTP three-parameter coordinate system, providing greater flexibility in setting up the numerical grid.

As the SinhSymTP numerical grid is not compactified, finite-radius boundary conditions must be applied. To address this, a new radiation boundary condition algorithm within NRPy+ has been developed, which is based on the widely used NewRad radiation boundary condition driver within the Einstein Toolkit. While NewRad implements a second-order approach, NRPyElliptic extends to fourth and sixth finite difference orders as well. This high-order boundary condition meshes well with the high (tenth) order finite-difference representation of the elliptic operators adopted in NRPyElliptic.

To greatly accelerate the relaxation, we set the wave speed of the hyperbolic PDEs to grow in proportion to the grid spacing. As the SinhSymTP grid spacing grows exponentially with distance from the central region of the coordinate system, so does the relaxation wavespeed. Thus this approach is many orders of magnitude faster than the traditional, constant wave speed choice, in fact making it fast enough to set up high-quality, full-3D BBH ID for numerical relativity.

Although NRPyElliptic currently requires only a tiny fraction of the total runtime of a typical NR BBH merger calculation, it is roughly 12x slower than TwoPunctures when setting up the full-3D BBH ID in this work. Efforts in the immediate future will in part focus on improving this performance. To this end, a couple of ideas come to mind. First, all ID generated in this work used the trivial ($u = v = 0$) initial guess. We plan to explore whether relaxations at lower-resolutions might be used to provide a superior initial guess on finer grids, in which convergence is accelerated.

Second, due to the CFL condition, the speed of NRPyElliptic is proportional to the smallest grid spacing, which occurs at the foci of our SinhSymTP coordinate system. Typical grid spacings at the foci are $\sim 10^{-4}M$ due to

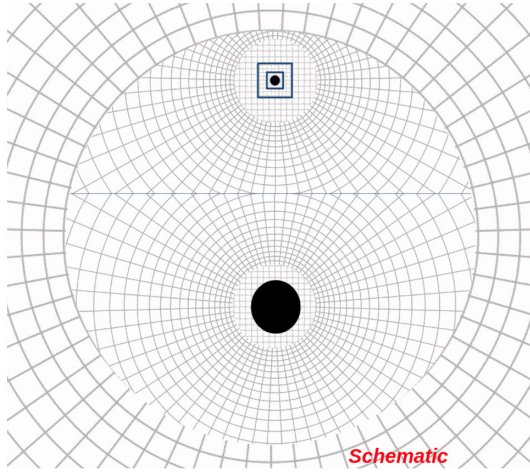


FIG. 9. Schematic of seven-grid BiSpheres numerical meshes used for BlackHoles@Home $\sim 6:1$ mass-ratio BBH simulations during the long inspiral phase. BHs are represented as black dots.

extreme grid focusing there, which in turn are $\sim 1/100$ those typically used in (near-equal-mass) binary puncture evolutions, indicating that a significant speed-up may be possible if superior grid structures are used.

To this end, we plan to adopt the same seven-grid bispheres grids infrastructure adopted by the NRPy+ -based BlackHoles@Home [68] project. As illustrated in Fig. 9, seven-grid BiSpheres consists of seven overlapping spherical-like and Cartesian-like grids. This approach places Cartesian-like AMR grid patches over regions where the spherical-like grids would otherwise experience extreme grid focusing ($r \rightarrow 0$), constraining the smallest grid spacings in the strong-field region to $\approx M/200$. Thus with such grids, accounting for needed inter-grid interpolations, we might expect roughly a $\sim 10x$ increase in speed—making NRPyElliptic comparable in performance to TwoPunctures for near-equal-mass-ratio systems—all while maintaining excellent resolution in the strong-field region. Further, the Cartesian-like AMR patches on these grids are centered precisely at the locations of the compact objects, making them efficiently tunable to higher mass ratios, unlike SinhSymTP or other prolate spheroidal-like coordinates mentioned in this work. Extending NRPyElliptic to higher mass ratios in this way, as well as to other types of NR ID, will be explored in forthcoming papers.

ACKNOWLEDGMENTS

The authors would like to thank S. Brandt and R. Haas for useful discussions and suggestions during the preparation of NRPyElliptic. Support for this work was provided by NSF Grants No. OAC-2004311, No. PHY-1806596, No. PHY-211035, as well as NASA Grants No. ISFM-80NSSC18K0538 and No. TCAN-80NSSC18K1488. Computational resources were in part provided by West Virginia University’s Thorny Flat high-performance

computing cluster, funded by NSF MRI Grant No. 1726534 and West Virginia University. T. P. J. acknowledges support from West Virginia University through the STEM Graduate Fellowship.

APPENDIX A: SOMMERFELD BOUNDARY CONDITIONS

Sommerfeld boundary conditions (BCs)—also referred to as radiation or transparent boundary conditions—aim to enable outgoing wave fronts to pass through the boundaries of a domain with minimal reflection. Sommerfeld BCs typically assume that for large values of r any given field $f = f(t, r)$ behaves as an outgoing spherical wave, with an asymptotic value f_0 as $r \rightarrow \infty$. Following NewRad [54], our ansatz for $f(t, r)$ on the boundary takes the form

$$f = f_0 + \frac{w(r-ct)}{r} + \frac{C}{r^n}, \quad (\text{A1})$$

where $w(r-ct)/r$ represents an outgoing wave that solves the wave equation in spherical symmetry,¹² and C is a constant. The $1/r^n$ correction term encapsulates higher-order corrections with $n > 1$ falloff.

We follow the hyperbolic relaxation method of [39] and NewRad, and apply Sommerfeld boundary conditions not to f directly, but to $\partial_t f$,

$$\partial_t f = -c \frac{w'(r-ct)}{r}. \quad (\text{A2})$$

To better understand the $w'(r-ct)$ term, we compute the radial partial derivative of f as well,

$$\partial_r f = \frac{w'(r-ct)}{r} - \frac{w(r-ct)}{r^2} - n \frac{C}{r^{n+1}}. \quad (\text{A3})$$

Solving Eq. (A3) for $w'(r-ct)$ and substituting into Eq. (A2) yields

$$\partial_t f = -c \left[\partial_r f + \frac{w(r-ct)}{r^2} + n \frac{C}{r^{n+1}} \right]. \quad (\text{A4})$$

To take care of the (as yet) unknown $w(r-ct)/r^2$ term, notice that our ansatz Eq. (A1) implies

$$\frac{w(r-ct)}{r^2} = \frac{f-f_0}{r} - \frac{C}{r^{n+1}}, \quad (\text{A5})$$

which when inserted into Eq. (A4) yields

$$\partial_t f = -c \left[\partial_r f + \frac{f-f_0}{r} \right] + \frac{k}{r^{n+1}}, \quad (\text{A6})$$

¹²That is, $\partial_t^2(rw) - c^2 \partial_r^2(rw) = 0$.

where $k = -cC(n-1)$ is just another constant. Thus we have derived the desired boundary condition

$$\partial_t f = -\frac{c}{r} [r \partial_r f + (f - f_0)] + \frac{k}{r^{n+1}}. \quad (\text{A7})$$

To generalize Eq. (A7) to arbitrary curvilinear coordinate systems x_{Curv}^i , we make use of the chain rule

$$\frac{\partial f(x_{\text{Curv}}^i)}{\partial r} = \left(\frac{\partial x_{\text{Curv}}^i}{\partial r} \right) \left(\frac{\partial f}{\partial x_{\text{Curv}}^i} \right), \quad (\text{A8})$$

which can be plugged into Eq. (A7) to give us Eq. (29)

$$\partial_t f = -\frac{c}{r} \left[r \frac{\partial x_{\text{Curv}}^i}{\partial r} \frac{\partial f}{\partial x_{\text{Curv}}^i} + (f - f_0) \right] + \frac{k}{r^{n+1}}. \quad (\text{A9})$$

Returning to the original ansatz [Eq. (A1)], we would generally expect the lowest-order correction to be one order higher than the dominant, $1/r$ falloff. As the correction term in Eq. (A1) has $1/r^n$ falloff, we therefore set $n = 2$ to obtain our final expression for imposing outgoing radiation boundary conditions any given field f ,

$$\partial_t f = -\frac{c}{r} \left[r \frac{\partial x_{\text{Curv}}^i}{\partial r} \frac{\partial f}{\partial x_{\text{Curv}}^i} + (f - f_0) \right] + \frac{k}{r^3}. \quad (\text{A10})$$

Regarding numerical implementation of this expression a couple of subtleties arise. First, note that $\partial x_{\text{Curv}}^i / \partial r$ may be impossible to compute analytically, as the spherical radius r is generally easy to write in terms of the curvilinear coordinates x_{Curv}^i , but not its inverse $x_{\text{Curv}}^i(r)$.

To address this, for all coordinate systems x_{Curv}^i implemented in NRPy+, the function

$$x_{\text{Sph}}^i = (r(x_{\text{Curv}}^i), \theta(x_{\text{Curv}}^i), \phi(x_{\text{Curv}}^i)), \quad (\text{A11})$$

is explicitly defined. If we define the Jacobian

$$J^j_i = \frac{\partial x_{\text{Sph}}^j}{\partial x_{\text{Curv}}^i}, \quad (\text{A12})$$

and use NRPy+ functions to invert this matrix, we obtain exact expressions for the inverse Jacobian matrix, which encodes $\partial x_{\text{Curv}}^i / \partial x_{\text{Sph}}^j$,

$$(J^{-1})^i_j = \frac{\partial x_{\text{Curv}}^i}{\partial x_{\text{Sph}}^j}. \quad (\text{A13})$$

From this we can express $\partial x_{\text{Curv}}^i / \partial r$ exactly for any curvilinear coordinate system implemented within NRPy+.

The second subtlety lies in formulating a way to approximate k . If the function f represented only an outgoing spherical wave, then it would exactly satisfy the advection equation

$$\left[\frac{\partial f}{\partial t} \right]_{\text{adv}} \equiv -\frac{c}{r} \left[r \frac{\partial x_{\text{Curv}}^i}{\partial r} \frac{\partial f}{\partial x_{\text{Curv}}^i} + (f - f_0) \right], \quad (\text{A14})$$

which is identical to Eq. (A10) but with $k = 0$.

Next, consider an interior point r_{int} directly adjacent to the outer boundary. Then, $f(r_{\text{int}})$ approximately satisfies both the time evolution equation [e.g., Eq. (25)], and the advection equation Eq. (A14). We compute $\partial_t f(r_{\text{int}})$ for a given field f directly from evaluating the corresponding right-hand side of Eq. (21), and $[\partial_t f]_{\text{adv}}(r_{\text{int}})$ from Eq. (A14). The difference of these two equations yields the departure from the expected purely outgoing wave behavior at that point k/r_{int}^{n+1} . From this we can immediately extract k ,

$$k = r_{\text{int}}^3 \left(\frac{\partial f}{\partial t} - \left[\frac{\partial f}{\partial t} \right]_{\text{adv}} \right)_{\text{int}}, \quad (\text{A15})$$

where again we impose $n = 2$.

Our numerical implementation of Sommerfeld BCs evaluates $\partial f / \partial x_{\text{Curv}}^i$ in Eq. (A14) using either centered or fully upwinded finite-difference derivatives as needed to ensure finite-difference stencils do not reach out of bounds. Unlike NewRad, which only implements second-order finite-difference derivatives for $\partial f / \partial x_{\text{Curv}}^i$, our implementation supports second, fourth, and sixth-order finite differences.

We validated this Sommerfeld boundary condition algorithm against NewRad for the case of a scalar wave propagating across a 3D Cartesian grid, choosing second-order finite-difference derivatives in our algorithm. We achieved roundoff-level agreement for the wave propagating toward each of the individual faces.

APPENDIX B: SPATIALLY DEPENDENT WAVE SPEED AND RELAXATION-WAVE CROSSING TIME

At every point in the domain we compute the smallest proper distance between neighboring points, ΔS_{min} , and define a local wave speed that is proportional to the grid spacing as

$$c(\vec{x}) = C_0 \frac{\Delta S_{\text{min}}(\vec{x})}{\Delta t}, \quad (\text{B1})$$

where Δt is the time step used by the time integrator, and C_0 is the CFL factor.

For the sake of readability, we repeat here the relationship between SinhSymTP coordinates and Cartesian coordinates,

$$\begin{aligned} x &= \tilde{r} \sin(x_2) \cos(x_3), \\ y &= \tilde{r} \sin(x_2) \sin(x_3), \\ z &= (\tilde{r}^2 + b^2)^{1/2} \cos(x_2). \end{aligned} \quad (\text{33})$$

To establish the time scales associated with the propagation of relaxation waves, we calculate the relaxation-wave-crossing

time along the x -axis— $(x_2, x_3) = (\pi/2, 0)$ in Eq. (33)—and the z -axis. NRPyElliptic adopts topologically Cartesian, cell-centered grids with intercell spacing of Δx_1 , Δx_2 , and Δx_3 in the x_1 , x_2 , and x_3 directions, respectively. Because of this, the closest points to the x -axis are obtained by setting $x_2 = (\pi + \Delta x_2)/2$ and $x_3 = \Delta x_3/2$. Likewise, the closest points to the z -axis are given by setting $x_2 = \Delta x_2/2$ and x_3 to any fixed value. The relaxation-wave crossing time along the x axis is given by

$$T_{\text{RC}}^{(x)} = \int \frac{dx}{c(x)} = \int_{x_1=0}^1 \frac{f(x_1) dx_1}{c^{(x)}(x_1)}, \quad (\text{B2})$$

where

$$f(x_1) = \partial_{x_1} x(x_1, x_2, x_3)|_{x_2=(\pi+\Delta x_2)/2, x_3=\Delta x_3/2}, \quad (\text{B3})$$

and $c^{(x)}$ is the wave speed along the x -axis. For the choice of grid parameters described in Sec. III, the smallest grid spacing along the x -axis is given by the proper distance in the direction of the x_2 coordinate, so that

$$c^{(x)}(x_1) = C_0 \frac{\Delta S^{(2)}(x_1)}{\Delta t} = \frac{C_0 \Delta x_2}{\Delta t} h^{(2)}(x_1), \quad (\text{B4})$$

where $h^{(2)}(x_1) = \sqrt{\tilde{r}^2(x_1) + b^2 \cos^2(\Delta x_2/2)}$ is the scale factor. Substituting Eq. (B4) in Eq. (B2) and integrating yields

$$T_{\text{RC}}^{(x)} = \frac{\Delta t}{C_0 \Delta x_2} \operatorname{arctanh} \left(\frac{\mathcal{A}}{\sqrt{\mathcal{A}^2 + b^2 \cos^2(\Delta x_2/2)}} \right) \times \sin(\Delta x_2/2) \cos(\Delta x_3/2). \quad (\text{B5})$$

Similarly, the wave speed along the z -axis is given by

$$c^{(z)}(x_1) = C_0 \frac{\Delta S^{(3)}(x_1)}{\Delta t} = \frac{C_0 \Delta x_3}{\Delta t} h^{(3)}(x_1), \quad (\text{B6})$$

where $h^{(3)}(x_1) = \tilde{r}(x_1) \sin(\Delta x_2/2)$ is the scale factor in the direction of x_3 . Thus, the relaxation-wave crossing time along the z -axis can be computed as

$$\begin{aligned} T_{\text{RC}}^{(z)} &= \int \frac{dz}{c(z)} = \int_{x_1=0}^1 \frac{g(x_1) dx_1}{c^{(z)}(x_1)} \\ &= \frac{\Delta t}{C_0 \Delta x_3} \operatorname{arctanh} \left(\frac{\mathcal{A}}{\sqrt{\mathcal{A}^2 + b^2}} \right) \cot(\Delta x_2/2), \end{aligned} \quad (\text{B7})$$

where we used

$$g(x_1) = \partial_{x_1} z(x_1, x_2, x_3)|_{x_2=\Delta x_2/2}. \quad (\text{B8})$$

Plugging in the values of the grid parameters \mathcal{A} , w , and step sizes, we find $T_{\text{RC}}^{(x)} = 0.248M$ ($0.574M$) and $T_{\text{RC}}^{(z)} = 1.28M$ ($1.26M$) for full-3D (axisymmetric) ID.

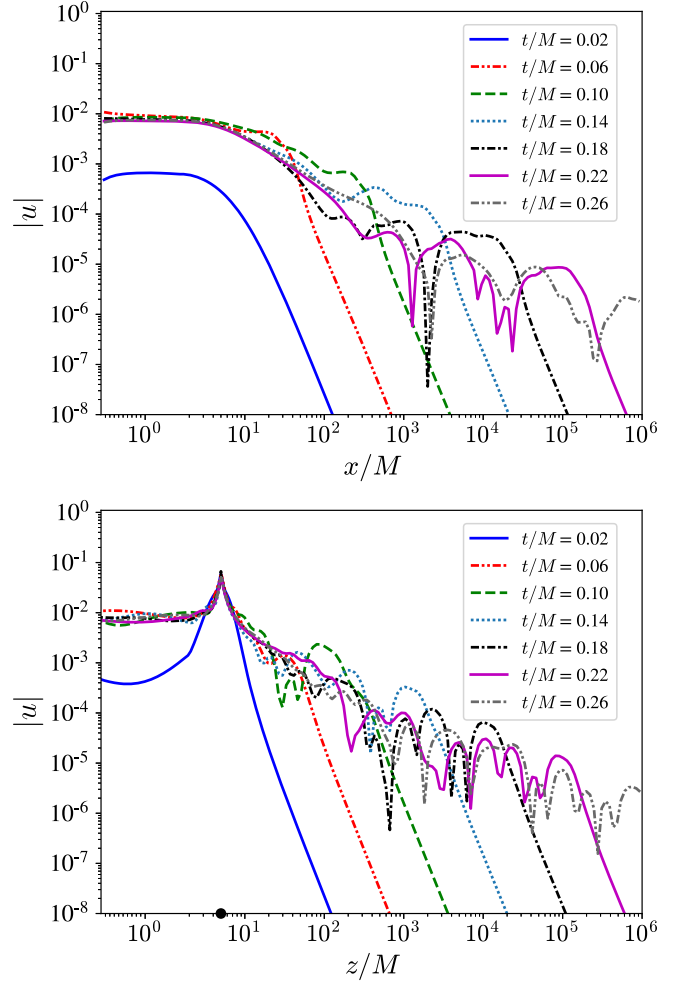


FIG. 10. Top: Propagation of u along gridpoints closest to the x -axis during the first relaxation-wave crossing time for the 3D ID. Bottom: Same as top panel, except for gridpoints closest to the z -axis.

From Eq. (B6) we find that for fixed values of x_1 and small angles x_2 the wave speed along the z -axis behaves as

$$c^{(z)} \sim \tilde{r}(x_1) x_2 \Delta x_3. \quad (\text{B9})$$

Our cell-centered grid has $x_{2i_2} = (i_2 + 1/2)\Delta x_2$ with $i_2 = 0, 1, 2, \dots, N_2$, and thus the wave speed quickly increases as we move away from the axis. Such rapidly moving signals slightly off the z -axis are not considered in our analytic estimates, yet influence the RCT so that in our full-3D simulations we observe the same value of $T_{\text{RC}} \approx 0.25M \approx T_{\text{RC}}^{(x)}$ along *both* axes, as shown in Fig. 10. Thus we use $T_{\text{RC}} = T_{\text{RC}}^{(x)}$ in all figures.

- [1] R. Abbott *et al.* (LIGO Scientific, VIRGO Collaborations), GWTC-2.1: Deep extended catalog of compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run, [arXiv:2108.01045](https://arxiv.org/abs/2108.01045).
- [2] T. Nakamura, K.-I. Oohara, and Y. Kojima, General relativistic collapse to black holes and gravitational waves from black holes, *Prog. Theor. Phys. Suppl.* **90**, 1 (1987).
- [3] M. Shibata and T. Nakamura, Evolution of three-dimensional gravitational waves: Harmonic slicing case, *Phys. Rev. D* **52**, 5428 (1995).
- [4] T. W. Baumgarte and S. L. Shapiro, On the numerical integration of Einstein's field equations, *Phys. Rev. D* **59**, 024007 (1998).
- [5] F. Pretorius, Numerical relativity using a generalized harmonic decomposition, *Classical Quantum Gravity* **22**, 425 (2005).
- [6] C. Bona, T. Ledvinka, C. Palenzuela, and M. Zacek, General covariant evolution formalism for numerical relativity, *Phys. Rev. D* **67**, 104005 (2003).
- [7] S. Bernuzzi and D. Hilditch, Constraint violation in free evolution schemes: Comparing BSSNOK with a conformal decomposition of Z4 formulation, *Phys. Rev. D* **81**, 084003 (2010).
- [8] D. Alic, C. Bona-Casas, C. Bona, L. Rezzolla, and C. Palenzuela, Conformal and covariant formulation of the Z4 system with constraint-violation damping, *Phys. Rev. D* **85**, 064040 (2012).
- [9] D. Alic, W. Kastaun, and L. Rezzolla, Constraint damping of the conformal and covariant formulation of the Z4 system in simulations of binary neutron stars, *Phys. Rev. D* **88**, 064049 (2013).
- [10] N. Sanchis-Gual, P. J. Montero, J. A. Font, E. Müller, and T. W. Baumgarte, Fully covariant and conformal formulation of the Z4 system in a reference-metric approach: Comparison with the BSSN formulation in spherical symmetry, *Phys. Rev. D* **89**, 104033 (2014).
- [11] A. M. Knapp, E. J. Walker, and T. W. Baumgarte, Illustrating stability properties of numerical relativity in electrodynamics, *Phys. Rev. D* **65**, 064031 (2002).
- [12] G. B. Cook, Initial data for numerical relativity, *Living Rev. Relativity* **3**, 5 (2000).
- [13] H. P. Pfeiffer, The initial value problem in numerical relativity, *J. Hyperbol. Differ. Equations* **02**, 497 (2005).
- [14] W. Tichy, The initial value problem as it relates to numerical relativity, *Rep. Prog. Phys.* **80**, 026901 (2017).
- [15] E. Gourgoulhon, Construction of initial data for 3 + 1 numerical relativity, *J. Phys. Conf. Ser.* **91**, 012001 (2007).
- [16] A. A. Tsokaros and K. Uryū, Numerical method for binary black hole/neutron star initial data: Code test, *Phys. Rev. D* **75**, 044026 (2007).
- [17] K. Uryū, F. Limousin, J. L. Friedman, E. Gourgoulhon, and M. Shibata, Nonconformally flat initial data for binary compact objects, *Phys. Rev. D* **80**, 124004 (2009).
- [18] A. Tsokaros, K. Uryū, and L. Rezzolla, New code for quasiequilibrium initial data of binary neutron stars: Corotating, irrotational, and slowly spinning systems, *Phys. Rev. D* **91**, 104030 (2015).
- [19] K. Uryū and A. Tsokaros, New code for equilibria and quasiequilibrium initial data of compact objects, *Phys. Rev. D* **85**, 064014 (2012).
- [20] W. E. East, F. M. Ramazanoğlu, and F. Pretorius, Conformal thin-sandwich solver for generic initial data, *Phys. Rev. D* **86**, 104053 (2012).
- [21] N. Moldenhauer, C. M. Markakis, N. K. Johnson-McDaniel, W. Tichy, and B. Brügmann, Initial data for binary neutron stars with adjustable eccentricity, *Phys. Rev. D* **90**, 084043 (2014).
- [22] P. Grandclément, Accurate and realistic initial data for black hole neutron star binaries, *Phys. Rev. D* **74**, 124002 (2006).
- [23] E. Gourgoulhon, P. Grandclément, K. Taniguchi, J.-A. Marck, and S. Bonazzola, Quasiequilibrium sequences of synchronized and irrotational binary neutron stars in general relativity: Method and tests, *Phys. Rev. D* **63**, 064029 (2001).
- [24] H. P. Pfeiffer, L. E. Kidder, M. A. Scheel, and S. A. Teukolsky, A multidomain spectral method for solving elliptic equations, *Comput. Phys. Commun.* **152**, 253 (2003).
- [25] S. Ossokine, F. Foucart, H. P. Pfeiffer, M. Boyle, and B. Szilágyi, Improvements to the construction of binary black hole initial data, *Classical Quantum Gravity* **32**, 245010 (2015).
- [26] F. Foucart, L. E. Kidder, H. P. Pfeiffer, and S. A. Teukolsky, Initial data for black hole neutron star binaries: A flexible, high-accuracy spectral method, *Phys. Rev. D* **77**, 124051 (2008).
- [27] N. Tacik, F. Foucart, H. P. Pfeiffer, R. Haas, S. Ossokine, J. Kaplan, C. Muhlberger, M. D. Duez, L. E. Kidder, M. A. Scheel, and B. Szilágyi, Binary neutron stars with arbitrary spins in numerical relativity, *Phys. Rev. D* **92**, 124012 (2015).
- [28] N. Tacik, F. Foucart, H. P. Pfeiffer, C. Muhlberger, L. E. Kidder, M. A. Scheel, and B. Szilágyi, Initial data for black hole-neutron star binaries, with rotating stars, *Classical Quantum Gravity* **33**, 225012 (2016).
- [29] K. Taniguchi, E. Gourgoulhon, and S. Bonazzola, Quasiequilibrium sequences of synchronized and irrotational binary neutron stars in general relativity. II. Newtonian limits, *Phys. Rev. D* **64**, 064012 (2001).
- [30] K. Taniguchi and E. Gourgoulhon, Quasiequilibrium sequences of synchronized and irrotational binary neutron stars in general relativity. III. Identical and different mass stars with $\gamma = 2$, *Phys. Rev. D* **66**, 104019 (2002).
- [31] M. Ansorg, B. Brügmann, and W. Tichy, A Single-domain spectral method for black hole puncture data, *Phys. Rev. D* **70**, 064011 (2004).
- [32] T. Dietrich, N. Moldenhauer, N. K. Johnson-McDaniel, S. Bernuzzi, C. M. Markakis, B. Brügmann, and W. Tichy, Binary neutron stars with generic spin, eccentricity, mass ratio, and compactness: Quasi-equilibrium sequences and first evolutions, *Phys. Rev. D* **92**, 124007 (2015).
- [33] W. Tichy, A. Rashti, T. Dietrich, R. Dudi, and B. Brügmann, Constructing binary neutron star initial data with high spins, high compactnesses, and high mass ratios, *Phys. Rev. D* **100**, 124046 (2019).
- [34] P. Grandclément, KADATH: A spectral solver for theoretical physics, *J. Comput. Phys.* **229**, 3334 (2010).
- [35] L. J. Papenfort, S. D. Tootle, P. Grandclément, E. R. Most, and L. Rezzolla, New public code for initial data of unequal-mass, spinning compact-object binaries, *Phys. Rev. D* **104**, 024057 (2021).

- [36] A. Rashti, F. M. Fabbri, B. Brüggmann, S. V. Chaurasia, T. Dietrich, M. Ujevic, and W. Tichy, Elliptica: a new pseudo-spectral code for the construction of initial data, [arXiv:2109.14511](https://arxiv.org/abs/2109.14511).
- [37] W. Barreto, P. C. M. Clemente, H. P. de Oliveira, and B. Rodriguez-Mueller, RIO: a new computational framework for accurate initial data of binary black holes, *Gen. Relativ. Gravit.* **50**, 71 (2018).
- [38] N. L. Fischer *et al.*, A scalable elliptic solver with task-based parallelism for the SPECTRE numerical relativity code, [arXiv:2111.06767](https://arxiv.org/abs/2111.06767).
- [39] H. R. Rüter, D. Hilditch, M. Bugner, and B. Brüggmann, Hyperbolic relaxation method for elliptic equations, *Phys. Rev. D* **98**, 084044 (2018).
- [40] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh *et al.*, sympy: symbolic computing in PYTHON, *PeerJ Comput. Sci.* **3**, e103 (2017).
- [41] I. Ruchlin, Z. B. Etienne, and T. W. Baumgarte, SENR/NRPy+: Numerical relativity in singular curvilinear coordinate systems, *Phys. Rev. D* **97**, 064036 (2018).
- [42] NRPy+'s webpage, <http://nrpyplus.net/>.
- [43] J. W. York, Jr., Conformal "Thin-Sandwich" Data for the Initial-Value Problem of General Relativity, *Phys. Rev. Lett.* **82**, 1350 (1999).
- [44] G. Lovelace, M. Boyle, M. A. Scheel, and B. Szilagy, Accurate gravitational waveforms for binary-black-hole mergers with nearly extremal spins, *Classical Quantum Gravity* **29**, 045003 (2012).
- [45] M. Alcubierre, *Introduction to 3+1 Numerical Relativity* (Oxford University Press, Oxford, U.K., 2008).
- [46] F. Löffler *et al.*, The Einstein toolkit: A community computational infrastructure for relativistic astrophysics, *Classical Quantum Gravity* **29**, 115001 (2012).
- [47] Einstein toolkit home page, <http://einstein-toolkit.org>.
- [48] Cactus web site, <http://www.cactuscode.org>.
- [49] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*, 2nd ed. (Cambridge University Press, New York, 1992).
- [50] T. W. Baumgarte and S. L. Shapiro, *Numerical Relativity: Solving Einstein's Equations on the Computer* (Cambridge University Press, Cambridge, U.K., 2010).
- [51] J. M. Bowen and J. W. York, Jr., Time-asymmetric initial data for black holes and black-hole collisions, *Phys. Rev. D* **21**, 2047 (1980).
- [52] S. Brandt and B. Brueggmann, A Simple Construction of Initial Data for Multiple Black Holes, *Phys. Rev. Lett.* **78**, 3606 (1997).
- [53] E. Schnetter, Time step size limitation introduced by the BSSN gamma driver, *Classical Quantum Gravity* **27**, 167001 (2010).
- [54] M. Alcubierre, B. Brueggmann, P. Diener, M. Koppitz, D. Pollney, E. Seidel, and R. Takahashi, Gauge conditions for long-term numerical black hole evolutions without excision, *Phys. Rev. D* **67**, 084023 (2003).
- [55] OpenMP home page, <https://www.openmp.org>.
- [56] P. Moon and D. E. Spencer, *Field Theory Handbook: Including Coordinate Systems, Differential Equations and Their Solutions* (Springer, New York, 2012).
- [57] V. Paschalidis, Z. B. Etienne, R. Gold, and S. L. Shapiro, An efficient spectral interpolation routine for the TwoPunctures code, [arXiv:1304.0457](https://arxiv.org/abs/1304.0457).
- [58] B. P. Abbott *et al.* (LIGO Scientific, Virgo Collaborations), Observation of Gravitational Waves from a Binary Black Hole Merger, *Phys. Rev. Lett.* **116**, 061102 (2016).
- [59] B. Wardell, I. Hinder, and E. Bentivegna, *Simulation of GW150914 binary black hole merger using the Einstein Toolkit* (2016), <https://www.einsteintoolkit.org/gallery/bbh/index.html>.
- [60] D. Pollney, C. Reisswig, E. Schnetter, N. Dorband, and P. Diener, High accuracy binary black hole simulations with an extended wave zone, *Phys. Rev. D* **83**, 044045 (2011).
- [61] E. Schnetter, S. H. Hawley, and I. Hawke, Evolutions in 3-D numerical relativity using fixed mesh refinement, *Classical Quantum Gravity* **21**, 1465 (2004).
- [62] J. Thornburg, A Fast apparent horizon finder for three-dimensional Cartesian grids in numerical relativity, *Classical Quantum Gravity* **21**, 743 (2004).
- [63] O. Dreyer, B. Krishnan, D. Shoemaker, and E. Schnetter, Introduction to isolated horizons in numerical relativity, *Phys. Rev. D* **67**, 024018 (2003).
- [64] T. Goodale, G. D. Allen, G. Lanfermann, J. Massó, T. Radke, E. Seidel, and J. Shalf, The cactus framework and toolkit: Design and applications, in *Vector and Parallel Processing—VECPAR'2002, 5th International Conference, Lecture Notes in Computer Science* (Springer, Berlin, 2003).
- [65] J. D. Brown, P. Diener, O. Sarbach, E. Schnetter, and M. Tiglio, Turduckening black holes: An analytical and computational study, *Phys. Rev. D* **79**, 044023 (2009).
- [66] S. Husa, I. Hinder, and C. Lechner, Kranc: A Mathematica application to generate numerical codes for tensorial evolution equations, *Comput. Phys. Commun.* **174**, 983 (2006).
- [67] M. Thomas and E. Schnetter, Simulation factory: Taming application configuration and workflow on high-end resources, in *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on (2010)*, [arXiv:1008.4571](https://arxiv.org/abs/1008.4571).
- [68] Blackholes@home home page, <http://blackholesathome.net/>.