SENR/NRPy+: Numerical relativity in singular curvilinear coordinate systems

Ian Ruchlin,[1] Zachariah B. Etienne,[1, 2] and Thomas W. Baumgarte[3]

[1] *Department of Mathematics, 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 and Astronomy, Bowdoin College, Brunswick, Maine 04011, USA*

(Dated: March 30, 2018)

We report on a new open-source, user-friendly numerical relativity code package called SENR/NRPy+. Our code extends previous implementations of the BSSN reference-metric formulation to a much broader class of curvilinear coordinate systems, making it ideally suited to modeling physical configurations with approximate or exact symmetries. In the context of modeling black hole dynamics, it is orders of magnitude more efficient than other widely used open-source numerical relativity codes. NRPy+ provides a Python-based interface in which equations are written in natural tensorial form and output at arbitrary finite difference order as highly efficient C code, putting complex tensorial equations at the scientist's fingertips without the need for an expensive software license. SENR provides the algorithmic framework that combines the C codes generated by NRPy+ into a functioning numerical relativity code. We validate against two other established, state-of-the-art codes, and achieve excellent agreement. For the first time—in the context of moving puncture black hole evolutions—we demonstrate nearly exponential convergence of constraint violation and gravitational waveform errors to zero as the order of spatial finite difference derivatives is increased, while fixing the numerical grids at moderate resolution in a singular coordinate system. Such behavior outside the horizons is remarkable, as numerical errors do not converge to zero near punctures, and all points along the polar axis are coordinate singularities. The formulation addresses such coordinate singularities via cell-centered grids and a simple change of basis that analytically regularizes tensor components with respect to the coordinates. Future plans include extending this formulation to allow dynamical coordinate grids and bispherical-like distribution of points to efficiently capture orbiting compact binary dynamics.

I. INTRODUCTION

The Laser Interferometer Gravitational-Wave Observatory (LIGO) and Virgo Scientific Collaboration's direct detections of six gravitational wave events—five black hole binary mergers [1–5] and one neutron star binary merger [6, 7]—have ushered in the age of gravitational wave astrophysics and multimessenger astronomy. As the signal-to-noise ratios of gravitational wave detections grow with increased interferometer sensitivity, the need to continually improve our theoretical models of these phenomena is critical, as new physics may otherwise be missed.

Just over a decade ago, breakthroughs in numerical relativity [8–10] opened the door to simulating the inspiral, merger, and ringdown phases of black hole binaries in vacuum. Black hole simulations—and, indeed, all compact binary simulations—span many orders of magnitude both in length scale and timescale. Making them computationally tractable for reliable gravitational wave predictions requires that the underlying numerical grid structure be tuned to optimally sample the space. Ordered meshes that map to Cartesian grids are most practical, as this greatly simplifies algorithms for high-order approximations of spatial derivatives in Einstein's equations.

There are currently two basic approaches in numerical relativity to setting up numerical grids for compact binary simulations. The most popular is to apply adaptive mesh refinement (AMR), where the grids consist of nested Cartesian coordinate boxes at different, discrete numerical resolutions. This enables the application of highest numerical resolution where it is most needed: the strongly curved spacetime fields inside and around compact objects. The most widely used AMR infrastructure is provided by the open-source Cactus/Carpet code [11–14] within the Einstein Toolkit [15, 16] (ETK). One downside to this approach is that the compact objects of interest are typically round, not rectangular, which results in a highly inefficient distribution of grid points inside and near compact objects (see, e.g., [17] for a more detailed analysis). In addition, the sudden change in grid resolution at the refinement boundaries can produce spurious reflections in sharp gauge modes or high-frequency gravitational wave features when using the moving puncture formalism, resulting in poor convergence in gravitational waveforms extracted from these numerical models [18, 19].

The current alternative to adaptive mesh refinement in numerical relativity codes, pioneered by the Simulating eXtreme Spacetimes (SXS) Collaboration [20] in their Spectral Einstein Code (SpEC [21], and its in-development successor SpECTRE [22]), is to smoothly juxtapose a large number of curvilinear three-dimensional grid patches, each with a smooth one-to-one mapping to a Cartesian grid. Evaluating derivatives with these grids requires computation of Jacobians, which, alongside managing the dynamics of the grid structure itself, remains a significant contributor to their codes' computational costs for compact binary inspirals that are aimed at generating gravitational wave predictions.

Both approaches involve control systems that adjust the grids to track compact objects as they orbit, although the SXS control system is far more complex because their formalism [23] additionally requires for stability that black hole interiors be very carefully excised from the computational domain. Negative side effects of the algorithmic complexity for both methods include a steep learning curve for new users and difficulty in interpreting numerical errors.

We address these drawbacks by developing a new code that builds on an innovative rescaling approach for solving Einstein's equations in spherical coordinates. The approach is designed to take full advantage of symmetries in the underlying configuration, requiring as few numerical grid points as possible and unlocking the desktop as a powerful tool for numerical relativity. Our work builds on strategies that, in the context of spherical coordinates, rescale tensors component by component so that the detrimental effects of coordinate singularities on numerics are completely removed [24–29]. Treating all coordinate singularities analytically, the equations can then be integrated numerically without encountering instabilities. We generalize this approach to a much broader class of static orthogonal coordinate systems by absorbing the coordinate singularities out of the tensor components and into a noncoordinate basis. The rescaling strategy is implemented in the context of the BSSN reference-metric formulation to enable highly efficient puncture black hole evolutions using the moving puncture approach [9, 10, 30] in a broad class of spherical-, cylindrical-, and Cartesian-like coordinate systems without special integration methods or introducing gaps in the numerical grid.

We implement this approach within a new, open-source code package called SENR/NRPy+ [17]. At its core, SENR/NRPy+ aims to be as algorithmically simple and user friendly as possible, all while being highly efficient. In short, SENR/NRPy+ aims to minimize both human and computational expense while maximizing science outcomes.

SENR/NRPy+ is built upon the philosophy that the distribution of points on the numerical grid should take maximum advantage of approximate symmetries in the physical system. Compact object systems of interest in gravitational wave astronomy typically possess a high degree of angular symmetry, making spherical- and, more generally, cylindrical-like coordinate systems ideal candidates for efficient sampling.

NRPy+ ("Python-based code generation for numerical relativity and beyond") is designed to convert the BSSN reference-metric formulation of the Einstein equations, in a broad class of orthogonal coordinate systems, from Einstein-like notation directly into C code. It operates without the need for expensive, proprietary computer algebra systems like Mathematica or Maple. As its name suggests, NRPy+ is based entirely in Python and depends only on the standard Python computer algebra package SymPy [31] for symbolic algebra, which is widely available on supercomputing clusters.

SENR ("the Simple, Efficient Numerical Relativity code") incorporates C codes generated by NRPy+ to form a complete, OpenMP-parallelized [32] numerical relativity code. Its skeletal structure makes the algorithmic underpinnings of numerical relativity codes transparent to the user.

We verify SENR/NRPy+ by direct comparisons with two other numerical relativity codes that are both well established in the literature. In the context of strongly perturbed Minkowski spacetime (a version of the robust stability test [33–36]), we achieve roundoff-level agreement with the Baumgarte *et al.* [29] code, which evolves the BSSN equations in spherical coordinates at fixed fourth-order finite difference accuracy. We also demonstrate excellent agreement between the results of SENR/NRPy+ and the ETK in the context of simulating a single, dynamical black hole. Then, we perform simulations of single- and double-black-hole spacetimes, demonstrating that the finite difference truncation error converges to zero with increasing grid resolution at the expected rate.

Perhaps most importantly, we show for the first time that—in the context of moving puncture evolutions—the truncation error in our finite differencing scheme converges to zero nearly exponentially[1] outside puncture black hole horizons with linear increase in the finite difference order, keeping the numerical grids fixed at moderate resolution.

The fact that we observe nearly exponential convergence is remarkable for two reasons. First, the numerical grids chosen for these simulations possess coordinate singularities at all points where $r\sin(\theta) = 0$. Therefore, near-exponential convergence in regions within causal contact of these singularities demonstrates that our tensor rescaling algorithm and cell-centered grids completely eliminate convergence problems related to these coordinate singularities. Second, a puncture black hole exhibits nonsmooth fields at the site of the puncture, meaning we should have no *a priori* expectation of near-exponential numerical convergence outside the black hole, either. We attribute the observed convergence to the fact that the characteristics of the physical (nongauge) fields, in the vicinity of the puncture, point towards the puncture.

[1] When finite difference truncation error dominates, we expect numerical error to scale approximately as $\left|C_n u^{(n+1)}(\xi)\right| (\Delta x)^n$, where $n = N_{\rm FD}$ is the finite difference order, ξ is in the neighborhood of the point at which we evaluate the derivative of the function u, and $C_n \sim 1/4^n$ for a centered stencil on a numerical grid with uniform spacing Δx. In the case that $\left|u^{(n+1)}(\xi)\right|$ is bounded (e.g., $u(r) = \sin r$), pure exponential convergence of the finite difference derivative error is observed as n is increased and Δx is held fixed (again, assuming that truncation error dominates). When simulating gravitational fields, $\left|u^{(n+1)}(\xi)\right|$ can grow as $n!$ (e.g., $u(r) = 1/(1-r)$, $r \neq 1$), reducing the rate of exponential convergence at the finite difference orders we typically choose ($N_{\rm FD} \in \{2, 4, 6, 8, 10\}$). We refer to this behavior as *nearly exponential convergence*.

With sufficient resolution inside of the horizon, the errors resulting from finite differencing across the puncture singularity become trapped near the puncture, and are not able to escape and contaminate the simulation at large [37, 38].

Future SENR projects will involve further extending the formalism to handle dynamical, bispherical-like coordinate systems, so that compact binary dynamics may be modeled with minimum computational expense. In this work we demonstrate near-exponential convergence of gravitational waves with increased finite differencing order in the context of head-on collisions of puncture black holes, and will build on this success to tackle the orbital black hole binary problem as our next step.

This paper is organized as follows. In Sec. II, we present the reference-metric formulation of the BSSN and gauge evolution equations. In Sec. III, we outline the tensor component rescaling procedure that makes it possible to evolve gravitational fields in singular coordinate systems. In Sec. IV, we describe the structure of the SENR/NRPy+ code, including the implementation of grid structures, coordinate system options, diagnostics, and boundary conditions. In Sec. V, we first demonstrate that SENR/NRPy+ agrees with the results of other established numerical relativity codes. Then, we show that numerical errors converge to zero at the expected rates for a nonspinning black hole with varying grid resolution or finite difference order, and in different coordinate systems and gauges. Finally, in the context of head-on collisions of two nonspinning puncture black holes, we demonstrate near-exponential convergence of the gravitational waveforms with increasing finite difference order. We conclude and present plans for future work in Sec. VI.

Throughout this paper geometrized units are adopted, in which $c = 1$ and $G = 1$. Latin indices $(i, j, k, \ldots)$ denote spatial degrees of freedom and obey the Einstein summation convention.

II. BSSN AND GAUGE EVOLUTION EQUATIONS

In this section, we describe our strategy for solving Einstein's equations, based on the tensor-weight-zero BSSN [39–41] formulation of Brown [26]. Our numerical implementation extends the rescaling approach, developed by Baumgarte *et al.* [42] for spherical coordinates, to a broader class of singular curvilinear coordinate systems. In this paper, we focus on spherical-, cylindrical-, and Cartesian-like coordinate systems, though the method can be easily extended to many others. For any of our coordinate grids, the reference metric $\hat{\gamma}_{ij}$ represents the flat space metric components expressed in a coordinate basis. In Sec. III, we use $\hat{\gamma}_{ij}$ in the rescaling procedure to define a noncoordinate basis, in terms of which which all tensor components are explicitly free of coordinate singularities. We assume that the background is independent of the coordinate time t, so that $\partial_t \hat{\gamma}_{ij} = 0$. All hatted quantities are associated with the reference metric.

The reference metric is used to decompose the conformal metric $\bar{\gamma}_{ij}$ into a correction about the flat background

$$\bar{\gamma}_{ij} = \hat{\gamma}_{ij} + \varepsilon_{ij} \ , \tag{1}$$

where the components in ε_{ij} are not necessarily small and contain the metric fields that are evolved on our numerical grids. The conformal metric is related to the physical spatial metric γ_{ij} though a conformal rescaling

$$\gamma_{ij} = e^{4\phi} \bar{\gamma}_{ij} \ . \tag{2}$$

Taking the determinant of (2) we observe that the conformal factor e^{ϕ} can be expressed as

$$e^{\phi} = \left(\frac{\gamma}{\bar{\gamma}}\right)^{1/12} \ , \tag{3}$$

where $\gamma \equiv \det(\gamma_{ij})$ and $\bar{\gamma} \equiv \det(\bar{\gamma}_{ij})$ are the metric determinants. Quantities associated with the conformal metric $\bar{\gamma}_{ij}$ are barred. For example, the conformal covariant derivative operator $\bar{D}_i$ is defined with respect to the conformal metric. The inverse conformal metric $\bar{\gamma}^{ij}$ is defined to satisfy

$$\bar{\gamma}^{ik} \bar{\gamma}_{kj} = \delta^i_j \ , \tag{4}$$

where δ^i_j is the Kronecker delta tensor.

The conformal rescaling (2) is not yet unique. In contrast to the original BSSN formulation, in which $\bar{\gamma}$ was set to unity, we adopt Brown's "Lagrangian" choice [26]

$$\partial_t \bar{\gamma} = 0 \ , \tag{5}$$

so that $\bar{\gamma}$ remains equal to its initial value. In particular, this implies that both γ and $\bar{\gamma}$ are allowed to transform as determinants, i.e., as scalar densities of weight two. According to (3) the conformal factor e^{ϕ} then transforms as a scalar, rather than a scalar density, and all other tensorial objects in our formalism similarly transform as tensors of weight zero (see also [24, 26]). We choose $\bar{\gamma} = \hat{\gamma} \equiv \det(\hat{\gamma}_{ij})$ in the initial data for all applications in this paper. Since both determinants remain independent of time, they remain equal to each other throughout the evolution.

It is well known that Christoffel symbols do not transform covariantly between coordinate systems. However, the difference of two sets of Christoffel symbols is tensorial. We define the tensor

$$\Delta^i_{jk} \equiv \bar{\Gamma}^i_{jk} - \hat{\Gamma}^i_{jk} \ , \tag{6}$$

whose indices are raised and lowered with the conformal metric. It is useful to construct a vector by taking the trace

$$\Delta^i \equiv \bar{\gamma}^{jk} \Delta^i_{jk} \ . \tag{7}$$

In addition, the tensor-weight-zero conformal connection coefficient three-vector $\bar{\Lambda}^i$ is evolved independently, and satisfies the initial constraint

$$\mathcal{C}^i \equiv \bar{\Lambda}^i - \Delta^i = 0 \ . \tag{8}$$

The conformal, trace-free part of the extrinsic curvature is denoted

$$\bar{A}_{ij} = e^{-4\phi}\left(K_{ij} - \frac{1}{3}\gamma_{ij}K\right) \ , \tag{9}$$

where K_{ij} is the physical extrinsic curvature and $K = \gamma^{ij}K_{ij}$ is the mean curvature.

Defining the hypersurface-normal derivative operator

$$\partial_\perp \equiv \partial_t - \mathcal{L}_\beta \ , \tag{10}$$

where $\mathcal{L}_\beta$ is the Lie derivative along the shift vector β^i, the BSSN evolution system in vacuum is written as [26]

$$\partial_\perp \varepsilon_{ij} = \frac{2}{3}\bar{\gamma}_{ij}\left(\alpha\bar{A}^k_k - \bar{D}_k\beta^k\right) + 2\hat{D}_{(i}\beta_{j)} - 2\alpha\bar{A}_{ij} \ , \tag{11a}$$

$$\begin{aligned}\partial_\perp \bar{A}_{ij} = &-\frac{2}{3}\bar{A}_{ij}\bar{D}_k\beta^k - 2\alpha\bar{A}_{ik}\bar{A}^k_j + \alpha\bar{A}_{ij}K \\ &+ e^{-4\phi}\left\{-2\alpha\bar{D}_i\bar{D}_j\phi + 4\alpha\bar{D}_i\phi\bar{D}_j\phi \right. \\ &\left. +4\bar{D}_{(i}\alpha\bar{D}_{j)}\phi - \bar{D}_i\bar{D}_j\alpha + \alpha\bar{R}_{ij}\right\}^{\rm TF} \ ,\end{aligned} \tag{11b}$$

$$\partial_\perp W = -\frac{1}{3}W\left(\bar{D}_k\beta^k - \alpha K\right) \ , \tag{11c}$$

$$\begin{aligned}\partial_\perp K = &\frac{1}{3}\alpha K^2 + \alpha\bar{A}_{ij}\bar{A}^{ij} \\ &- e^{-4\phi}\left(\bar{D}_i\bar{D}^i\alpha + 2\bar{D}^i\alpha\bar{D}_i\phi\right) \ ,\end{aligned} \tag{11d}$$

$$\begin{aligned}\partial_\perp \bar{\Lambda}^i = &\bar{\gamma}^{jk}\hat{D}_j\hat{D}_k\beta^i + \frac{2}{3}\Delta^i\bar{D}_j\beta^j + \frac{1}{3}\bar{D}^i\bar{D}_j\beta^j \\ &- 2\bar{A}^{ij}\left(\partial_j\alpha - 6\partial_j\phi\right) + 2\bar{A}^{jk}\Delta^i_{jk} \\ &- \frac{4}{3}\alpha\bar{\gamma}^{ij}\partial_j K \ .\end{aligned} \tag{11e}$$

In the above, "TF" denotes the trace-free part of the expression in brackets, the conformal factor is evolved as $W = e^{-2\phi}$ (following, e.g., [28, 43] to ensure smoother spacetime fields near puncture black holes), and the components of the conformal Ricci tensor are calculated by

$$\begin{aligned}\bar{R}_{ij} = &-\frac{1}{2}\bar{\gamma}^{kl}\hat{D}_k\hat{D}_l\bar{\gamma}_{ij} + \bar{\gamma}_{k(i}\hat{D}_{j)}\bar{\Lambda}^k + \Delta^k\Delta_{(ij)k} \\ &+ \bar{\gamma}^{kl}\left(2\Delta^m_{k(i}\Delta_{j)ml} + \Delta^m_{ik}\Delta_{mjl}\right) \ .\end{aligned} \tag{12}$$

The trace-free condition $\bar{\gamma}^{ij}\bar{A}_{ij} = 0$, which can be violated by numerical error, is enforced dynamically by the term proportional to $\bar{A}^i_i$ in Eq. (11a) [26]. In this paper, we restrict ourselves to vacuum spacetimes so that all of the matter source terms vanish.

The evolution system is completed with specification of the gauge: the lapse function α and the shift vector β^i. Unless otherwise stated, we employ the advective 1+log lapse condition [44]

$$\partial_0\alpha = -2\alpha K \tag{13}$$

and the advective Gamma-driver shift condition [30]

$$\partial_0\beta^i = B^i \ , \tag{14a}$$

$$\partial_0 B^i = \frac{3}{4}\partial_0\bar{\Lambda}^i - \eta B^i \ . \tag{14b}$$

Here, B^i is an auxiliary vector, η is a (dimensionful) damping parameter [45], and the (noncovariant) advective derivative operator is defined as (see Case No. 8 in Table I of [46])

$$\partial_0 \equiv \partial_t - \beta^j\partial_j \ . \tag{15}$$

The BSSN equations coupled to these gauge conditions are known together as the moving puncture approach [9, 10, 30]. A total of 24 fields are evolved.

III. TENSOR RESCALING

In the previous section, we described the reference-metric formulation of the BSSN evolution equations in the moving puncture approach (11), (13), and (14). These equations form the foundation for solving Einstein's equations in any coordinate system we like, and we leverage this coordinate freedom to take maximum advantage of near symmetries in physical systems. While, from an analytic perspective, there is no problem solving Einstein's equations in arbitrary coordinate systems, numerical solutions diverge if the chosen coordinate system possesses a coordinate singularity.

Well-known examples of coordinate singularities include the points at $\rho = 0$ in cylindrical coordinates or at $r\sin(\theta) = 0$ in spherical coordinates. Tensor components that are regular everywhere in a Cartesian basis will inherit these singularities during the change of basis to those coordinates. The fact that these singularities are a consequence of the coordinates themselves, and not of the underlying tensor fields, means that they can in principle be scaled out of the tensor components in a way that is consistent with the adopted reference-metric formulation of BSSN.

The goal of the tensor rescaling is to analytically absorb singular terms into the noncoordinate basis. Tensor components are naturally regular with respect to the coordinates when expressed in terms of the noncoordinate basis.

In Sec. III A, we show a simple example of rescaling a rank-1 tensor with one vector component and a rank-2 tensor with one two-dual-vector component in the case of ordinary spherical coordinates. We generalize the rescaling procedure to arbitrary coordinate distributions in Sec. III B.

A. Spherical rescaling examples

In this section, we refer to objects in Cartesian coordinates with indices i and j, and to objects in spherical coordinates with indices a and b.

Consider a finite vector field $\boldsymbol{V}$ with components V^i in Cartesian coordinates $y^{(i)} = (x, y, z)$ and a Cartesian coordinate basis $\frac{\boldsymbol{\partial}}{\boldsymbol{\partial} y^{(i)}}$

$$\boldsymbol{V} = V^i \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} y^{(i)}} \equiv V^x \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} x} + V^y \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} y} + V^z \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} z} \; . \tag{16}$$

An index in parentheses labels the individual coordinate functions or basis vectors, not vector components. For simplicity, and without loss of generality, we consider a vector that possesses only one nontrivial component ($V^y = V^z = 0$)

$$\boldsymbol{V} = V^x \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} x} \; . \tag{17}$$

The component V^x is a smooth, finite function of the Cartesian coordinate values.

In Cartesian coordinates, the natural set of noncoordinate basis vectors $\boldsymbol{e}_{(i)}$ coincides with the coordinate basis, and therefore

$$\boldsymbol{V} = V^x \boldsymbol{e}_{(x)} \; . \tag{18}$$

Now, we transform the vector to ordinary, uniform spherical coordinates $r^{(a)} = (r, \theta, \varphi)$, related to the Cartesian coordinates by

$$x = r \sin(\theta) \cos(\varphi) \; , \tag{19a}$$

$$y = r \sin(\theta) \sin(\varphi) \; , \tag{19b}$$

$$z = r \cos(\theta) \; . \tag{19c}$$

This coordinate relationship characterizes the familiar Jacobian matrix with components

$$\frac{\partial y^i}{\partial r^a} = \begin{pmatrix} \sin(\theta)\cos(\varphi) & r\cos(\theta)\cos(\varphi) & -r\sin(\theta)\sin(\varphi) \\ \sin(\theta)\sin(\varphi) & r\cos(\theta)\sin(\varphi) & r\sin(\theta)\cos(\varphi) \\ \cos(\theta) & -r\sin(\theta) & 0 \end{pmatrix} \tag{20}$$

and the inverse Jacobian matrix $\frac{\partial r^a}{\partial y^i}$ satisfies

$$\frac{\partial y^i}{\partial r^a} \frac{\partial r^a}{\partial y^j} = \delta^i_j \; , \quad \frac{\partial y^i}{\partial r^b} \frac{\partial r^a}{\partial y^i} = \delta^a_b \; . \tag{21}$$

An application of the ordinary derivative chain rule yields the transformation formula for the coordinate basis vectors

$$\frac{\boldsymbol{\partial}}{\boldsymbol{\partial} y^{(i)}} = \frac{\partial r^a}{\partial y^i} \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} r^{(a)}} \; . \tag{22}$$

The Cartesian components V^i are transformed from the spherical coordinate components V^a using the inverse Jacobian

$$V^i = V^a \frac{\partial y^i}{\partial r^a} \; . \tag{23}$$

It is this fact, that a tensor's components transform in a way that is inverse to the basis, which preserves the geometric meaning of a tensor field represented in any coordinate system. The resulting vector components in the spherical coordinate basis are

$$\begin{aligned} \boldsymbol{V} &= V^a \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} r^{(a)}} = V^x \frac{\partial r^a}{\partial x} \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} r^{(a)}} \\ &= V^x \sin(\theta)\cos(\varphi) \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} r} + \frac{V^x \cos(\theta)\cos(\varphi)}{r} \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} \theta} \\ &\quad - \frac{V^x \sin(\varphi)}{r \sin(\theta)} \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} \varphi} \; . \end{aligned} \tag{24}$$

The coordinate singularities that have been introduced by the Jacobian become obvious when the components are evaluated at the origin $r = 0$ and along the polar axis $\sin(\theta) = 0$.

We absorb this undesirable behavior into the basis vectors as follows. For the noncoordinate spherical basis $\boldsymbol{e}_{(a)}$, choose orthogonal vectors

$$\boldsymbol{e}_{(r)} = \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} r} \; , \tag{25a}$$

$$\boldsymbol{e}_{(\theta)} = \frac{1}{r} \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} \theta} \; , \tag{25b}$$

$$\boldsymbol{e}_{(\varphi)} = \frac{1}{r \sin(\theta)} \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} \varphi} \; . \tag{25c}$$

In terms of these, the vector components become

$$\begin{aligned} \boldsymbol{V} &= V^x \sin(\theta)\cos(\varphi) \boldsymbol{e}_{(r)} + V^x \cos(\theta)\cos(\varphi) \boldsymbol{e}_{(\theta)} \\ &\quad - V^x \sin(\varphi) \boldsymbol{e}_{(\varphi)} \; , \end{aligned} \tag{26}$$

which are manifestly regular over the entire domain.

To demonstrate that this strategy is more generally applicable, consider a rank-2 tensor $\boldsymbol{W}$ with only one nontrivial component in a Cartesian coordinate two-dual-vector basis

$$\boldsymbol{W} = W_{xx} \, \mathbf{d}x \otimes \mathbf{d}x \; , \tag{27}$$

where $\otimes$ is the tensor product and $\mathbf{d}$ is the exterior derivative operator acting on the scalar coordinate functions to produce one-forms $\mathbf{d}y^{(i)}$ (i.e. dual vectors). The transformation rule dual to (22) is

$$\mathbf{d}y^{(i)} = \frac{\partial y^i}{\partial r^a} \mathbf{d}r^{(a)} \; . \tag{28}$$

Two contractions with the inverse Jacobian matrix transforms the tensor components to the spherical coordinate basis

$$\begin{aligned} \boldsymbol{W} &= W_{xx} \sin^2(\theta)\cos^2(\varphi) \, \mathbf{d}r \otimes \mathbf{d}r \\ &\quad + 2W_{xx} r \sin(\theta)\cos(\theta)\cos^2(\varphi) \, \mathbf{d}r \otimes \mathbf{d}\theta \\ &\quad - 2W_{xx} r \sin^2(\theta)\sin(\varphi)\cos(\varphi) \, \mathbf{d}r \otimes \mathbf{d}\varphi \\ &\quad + W_{xx} r^2 \cos^2(\theta)\cos^2(\varphi) \, \mathbf{d}\theta \otimes \mathbf{d}\theta \\ &\quad - 2W_{xx} r^2 \sin(\theta)\cos(\theta)\sin(\varphi)\cos(\varphi) \, \mathbf{d}\theta \otimes \mathbf{d}\varphi \\ &\quad + W_{xx} r^2 \sin^2(\theta)\sin^2(\varphi) \, \mathbf{d}\varphi \otimes \mathbf{d}\varphi \; . \end{aligned} \tag{29}$$

Notice that the components of $\boldsymbol{W}$ vanish when evaluated at, say, $r = 0$ and $\theta = \pi$, which amounts to a coordinate singularity that destroys information about the tensor's value in other bases (e.g., Cartesian) at these points. This singular behavior is dual to that seen above, where the component values of $\boldsymbol{V}$ in the spherical coordinate basis became unbounded at certain locations.

Again, this is ameliorated by an alternative choice of basis. The noncoordinate spherical basis one-forms are defined as the dual to the basis vectors (25)

$$\boldsymbol{e}^{(r)} = \mathbf{d}r \ , \tag{30a}$$

$$\boldsymbol{e}^{(\theta)} = r\,\mathbf{d}\theta \ , \tag{30b}$$

$$\boldsymbol{e}^{(\varphi)} = r\sin(\theta)\,\mathbf{d}\varphi \ . \tag{30c}$$

In this noncoordinate basis, the tensor components become

$$\begin{aligned}\boldsymbol{W} = {} & W_{xx}\sin^2(\theta)\cos^2(\varphi)\boldsymbol{e}^{(r)}\otimes\boldsymbol{e}^{(r)} \\ & + 2W_{xx}\sin(\theta)\cos(\theta)\cos^2(\varphi)\boldsymbol{e}^{(r)}\otimes\boldsymbol{e}^{(\theta)} \\ & - 2W_{xx}\sin(\theta)\sin(\varphi)\cos(\varphi)\boldsymbol{e}^{(r)}\otimes\boldsymbol{e}^{(\varphi)} \\ & + W_{xx}\cos^2(\theta)\cos^2(\varphi)\boldsymbol{e}^{(\theta)}\otimes\boldsymbol{e}^{(\theta)} \\ & - 2W_{xx}\cos(\theta)\sin(\varphi)\cos(\varphi)\boldsymbol{e}^{(\theta)}\otimes\boldsymbol{e}^{(\varphi)} \\ & + W_{xx}\sin^2(\varphi)\boldsymbol{e}^{(\varphi)}\otimes\boldsymbol{e}^{(\varphi)} \ .\end{aligned} \tag{31}$$

Now, for any point in the spherical domain, it can be shown that all of the above components vanish simultaneously if and only if $W_{xx} = 0$.

For both $\boldsymbol{V}$ and $\boldsymbol{W}$—as well as higher rank tensors in general—these arguments extend to arbitrary Cartesian tensors, allowing all components to be nontrivial.

B. The general rescaling procedure

The rescaling examples reviewed in the previous section can be generalized by treating the noncoordinate basis as a collection of matrix operators (and the dual basis as the associated inverse operators), and using them to project the singularities out of tensor components represented in a coordinate basis. Since the difference between bases is only a coordinate transformation, and the BSSN formulation we adopt is covariant, we are free to apply this strategy to remove coordinate singularities from tensorial expressions in the formulation. This section reviews the general procedure.

We denote the noncoordinate basis vectors by $e^i_{(j)}$, where the index (j) lists the individual basis vectors and i labels the components of a particular vector with respect to the underlying coordinate basis. By definition, the set of basis vectors $\{\boldsymbol{e}_{(i)}\}$ is linearly independent and spans the tangent space at every point in the spatial hypersurface. We restrict our consideration to time-independent, orthonormal bases. There exists a dual basis $e^{(i)}_j$ satisfying

$$e^{(i)}_k e^k_{(j)} = e^{(k)}_j e^i_{(k)} = \delta^i_j \ . \tag{32}$$

The components of the flat background reference metric, represented in a coordinate basis, are related algebraically to the basis dual vectors via

$$\hat{\gamma}_{ij} = \delta_{kl} e^{(k)}_i e^{(l)}_j \ . \tag{33}$$

In this way, the reference metric is constructed as the "product" of basis vectors, or, equivalently, that the basis constitutes the "square root" of the reference metric [47]. We treat this relationship as the definition of the noncoordinate basis components in terms of the known flat space reference-metric components in the corresponding coordinate basis. The noncoordinate basis, defined as such, is an orthonormal basis. The components $e^{(i)}_j$ are sometimes referred to as the "scale factors" of the reference metric, and they contain the singularities associated with the coordinates. The components of other tensors are made regular with respect to the coordinates by factoring out the scale factors in the appropriate way. For example (1)

$$\varepsilon_{ij} = h_{kl} e^{(k)}_i e^{(l)}_j \tag{34}$$

and (8)

$$\bar{\Lambda}^i = \bar{\lambda}^j e^i_{(j)} \ . \tag{35}$$

The rescaled components are recovered by the inverse relationships

$$h_{ij} = \varepsilon_{kl} e^k_{(i)} e^l_{(j)} \tag{36}$$

and

$$\bar{\lambda}^i = \bar{\Lambda}^j e^{(i)}_j \ . \tag{37}$$

The rescaled tensor components h_{ij}, $\bar{\lambda}^i$, and so on, are regular with respect to the coordinate singularities. (Note that h_{ij} and $\bar{\lambda}^i$ in spherical coordinates correspond directly to the functions of the same name in [29].) The BSSN evolution equations (11) are tensorial, and are therefore independent of the choice of basis.

The rescaled fields are those that are integrated and differentiated numerically on the coordinate grid, described next, in Sec. IV. This rescaling procedure enables us to achieve stable and convergent solutions in a broad class of singular coordinate systems, as we demonstrate in Sec. V.

IV. THE SENR/NRPY+ CODE

Numerical relativity codes built to evolve 3+1 initial value formulations of Einstein's equations generally contain thousands of lines of code just to express the needed equations for initial data, time evolution, and diagnostics. Early incarnations were largely coded by hand, exacerbating the already laborious and time-consuming debugging process. Most numerical relativity groups have migrated to automatic code generation, typically relying on closed-source, proprietary computer algebra systems like Maple or Mathematica to directly convert tensorial expressions typed by hand directly in Einstein-like notation into, e.g., highly optimized C code. Kranc [48] is one example of a very nice open-source, Mathematica-based package for converting Einstein's equations—written in Einstein-like notation—into optimized C code.

Proprietary packages like Mathematica or Maple require expensive licenses that some users simply cannot afford, creating a barrier to entry for potential developers. Further, most numerical relativity simulations are performed with supercomputing systems, on which licenses for, e.g., Mathematica or Maple are not available—again due to the high licensing cost. As a workaround, numerical relativists will often generate their code locally, using Mathematica or Maple, and then transfer it to the supercomputing system—just another way that these licenses can inconvenience users.

NRPy+ ("Python-based code generation for numerical relativity and beyond") aims to address these issues. It is the first open-source [49, 50], non-Mathematica- or Maple-based code generation package for tensorial expressions written in Einstein notation. NRPy+ is written entirely in Python[2] and depends only on the standard Python computer algebra package SymPy [31] for symbolic algebra, which is widely available on supercomputing clusters.

If we wish to solve Einstein's equations in a new coordinate system with NRPy+, we need only define the corresponding reference metric in terms of its scale factors. Using these as input, NRPy+ generates Einstein's equations in these coordinates and outputs OpenMP-capable C code that is highly optimizable (SIMD vectorized) by compilers, resulting in a tremendous performance boost compared to simple serial implementations. NRPy+ also leverages SymPy's ability to eliminate common subexpressions from complicated algebraic expressions, minimizing the number of floating point operations per expression evaluation.

SENR ("the Simple, Efficient Numerical Relativity code") is a complete, OpenMP-parallelized [32] numerical relativity code, incorporating the C codes generated by NRPy+ wherever complicated tensorial expressions are needed. Its skeletal structure makes the algorithms on which numerical relativity codes are based transparent to the user.

The division of labor between SENR and NRPy+ is outlined in Table I, providing a convenient launching point for later subsections that expand on this structure.

A. Computational grid structures

Each of the 24 evolved fields defined in Sec. II are sampled by a discrete computational grid, represented as a numerical array storing the function value at each grid point. We define a uniformly sampled unit cube grid with coordinate labels $x^{(i)} = \left(x^1, x^2, x^3\right) \equiv (\mathtt{x1}, \mathtt{x2}, \mathtt{x3})$, where x^1 represents the first spatial degree of freedom in Einstein notation and `x1` represents the first coordinate as it appears in SENR/NRPy+, and so on for the other two coordinates. These coordinates correspond to the rescaled tensor basis, in which coordinate singularities have been removed. Thus we perform numerical integrations and finite difference operations on this grid using ordinary, uniformly spaced stencils. The uniform coordinates are mapped to the nonuniformly distributed Cartesian coordinates $y^{(i)} = y^{(i)}\left(x^{(j)}\right) \equiv (\mathtt{y1}, \mathtt{y2}, \mathtt{y3})$, chosen to exploit the near-symmetries of the physical system of interest. Tensors in the $y^{(i)}$ coordinates exist in a Cartesian coordinate basis with trivial symmetry and parity conditions (i.e., no inner boundaries).

The user specifies the number of grid points N_i dedicated to each coordinate direction x^i, fixing the uniform grid cell spacing

$$\Delta x^i = \frac{1}{N_i} \ . \tag{38}$$

The current method requires N_i to be even and $N_i \geq 2$. The user also chooses the finite difference order $N_{\rm FD}$ (see Sec. IV A 2), which determines the number of "ghost zone" points $N_{\rm G} = N_{\rm FD}/2{+}1$ on either side of the domain required to evaluate finite difference stencils that extend beyond the boundary. Most derivatives are computed using centered stencils, extending $N_{\rm FD}/2$ grid points symmetrically to either side of the point in question. Shift-advection derivatives (terms acted on by $\beta^i\partial_i$) are approximated by upwinded finite differences, which employ asymmetrical stencils with $N_{\rm FD}/2+1$ points on one side and $N_{\rm FD}/2-1$ on the other. Points in the ghost zone are not evolved directly, but depend entirely on the grid interior and are updated by the boundary condition routine (see Sec. IV E). Thus, the total number of points allocated to each coordinate is

$$N_{{\rm T}i} = N_i + 2N_{\rm G} \ . \tag{39}$$

The total number of points on the uniform grid is simply

$$N_{\rm T} = \prod_i N_{{\rm T}i} \ . \tag{40}$$

The grid points themselves are located at

$$x^{(i)}(j) = \Delta x^i\left(j - N_{\rm G} + \frac{1}{2}\right) \ , \tag{41}$$

where the grid index $j \in \{0, 1, \ldots, N_{{\rm T}i} - 1\}$. This produces a grid that guarantees functions are never evaluated on the coordinate singularities at, e.g., $r = 0$, $\theta = 0$,

[2] Both Python 2.7+ and 3.0+ are supported.

TABLE I. The division of labor between the SENR C code and the NRPy+ Python code, with a link to the relevant section detailing each task. To perform a simulation, NRPy+ is first run to automatically generate C files containing necessary initial data, evolution, and diagnostic equations, coupled to highly optimized finite difference codes as needed for spatial derivatives. SENR contains all of the infrastructure needed to make use of these C codes in the context of a full numerical simulation, complete with highly efficient time evolution algorithms, boundary conditions, diagnostics, and checkpointing capabilities.

Task	Description	Section
Coordinates	The curvilinear coordinates are defined in terms of the uniform coordinates within **NRPy+**. Only the scale factors (the square root of each diagonal reference-metric component) must be defined; all hatted quantities in Eq. (11) are evaluated directly from these scale factors. In addition, mappings from the chosen curvilinear coordinate basis to spherical and Cartesian coordinate bases are defined. The former is necessary for transforming initial data (currently expressed exclusively in spherical coordinates) to the desired uniform coordinates. The latter is necessary to transform from the chosen basis to evaluate the ADM integrals (expressed in Cartesian coordinates for convenience in interpreting the linear and angular momentum components).	IV A 1
Initial Data	Various initial configurations are included inside **NRPy+**, including multiple black hole Brill-Lindquist [51] initial data, conformally curved UIUC [52] initial data for single Kerr black holes, as well as analytical static trumpet initial data [53]. In addition, there are several choices for the initial gauge conditions. As described above, all initial data are currently written in a spherical basis, and converted to the desired curvilinear coordinate basis by **NRPy+** . **SENR** reads the initial data code generated by **NRPy+** to define each of the 24 BSSN fields evolved on our grids ("grid functions") at the initial time.	IV C
Boundary Conditions	**SENR** fills the ghost zone points with data from the grid interior for each of the evolved grid functions. Specialized boundary condition routines are written by hand for each type of boundary condition, whether they be inner (e.g., $\theta < 0$ in spherical-like coordinates) or outer boundary conditions.	IV E
Finite Differencing	Where spatial derivatives appear, **NRPy+** constructs finite difference stencils at the user-specified accuracy order on the uniform grid. Upwinded derivatives are enabled by default on all shift-advection terms (as is typical; see e.g., [30, 54, 55]).	IV A 2
Evolution Equations	**NRPy+** constructs and outputs the evolution equation right-hand-side C codes, expressing all spatial derivatives of the grid functions as finite differences. The C codes include reading all needed data from memory.	II
Diagnostics	**NRPy+** outputs the needed C codes for the BSSN constraints, the ADM integrands, and the spherically symmetric horizon finder. Additionally, **SENR** includes a code that interpolates grid function data onto a Cartesian grid to be evaluated using the large suite of diagnostic utilities within the ETK, such as generic horizon finding and gravitational wave extraction. Diagnostics are periodically evaluated by **SENR** and stored to disk.	IV B
Numerical Grids	**SENR** allocates memory for coordinate grids and evolved grid functions given the number of grid points and the coordinate definitions.	IV A
Time Integration	**SENR** computes the largest-allowed CFL time step from the proper distance, determined by the reference metric (output from **NRPy+**) and the chosen grid. It iterates the grid functions to the next time step, evaluating the evolution equation C codes. Generally, RK4 is used for time integrations.	IV D

or $\theta = \pi$ in spherical coordinates. Points for which all three coordinates satisfy $0 < x^i < 1$ are part of the grid interior, whereas points with any of the three $x^i < 0$ or $x^i > 1$ are in the ghost zone.

1. *Coordinate options*

In the present work, we demonstrate BSSN evolution in three different classes of coordinate system: Cartesian-like, cylindrical-like, and spherical-like. These are distinguished by the number and character of their inner boundary conditions. Boundary conditions must be applied on all faces of our numerical grid cube, whether the face maps to another face (e.g., in the case of the $\varphi = 0$ and $\varphi = 2\pi$ faces in spherical coordinates) or corresponds to the outer boundary (e.g., at $r = r_{\max}$ in spherical coordinates). The details of how to specify the inner and outer boundary ghost zone points are discussed in Sec. IV E.

Finite difference stencils are evaluated on the uniform $x^{(i)}$ grid, which possesses a one-to-one mapping to the nonuniformly sampled Cartesian coordinates $y^{(i)}$. For example, in spherical-like coordinates, the nonuniform grid is related to the uniform grid via

$$\texttt{y1} = x = f\left(\texttt{x1}\right)\sin\left(\pi\,\texttt{x2}\right)\cos\left(2\pi\,\texttt{x3}\right)\ , \tag{42a}$$

$$\texttt{y2} = y = f\left(\texttt{x1}\right)\sin\left(\pi\,\texttt{x2}\right)\sin\left(2\pi\,\texttt{x3}\right)\ , \tag{42b}$$

$$\texttt{y3} = z = f\left(\texttt{x1}\right)\cos\left(\pi\,\texttt{x2}\right)\ . \tag{42c}$$

The coordinate distribution is generalized by the function f which is required to be invertible, odd with respect to the origin, and at least twice differentiable. These properties of f determine the symmetry conditions for the inner boundaries of the grid, which are described in detail for spherical-like coordinates and the general case in Sec. IV E. The choice $f(\texttt{x1}) = r_{\max}\,\texttt{x1}$ reduces this to ordinary, uniform spherical coordinates extending out to radius $r_{\max}/M > 0$. It is often useful to adopt a logarithmically distributed radial coordinate of the form $f(\texttt{x1}) = A\sinh(\texttt{x1}/w)$, where $A, w > 0$ are free parameters. This allows the radial outer boundary to be pushed far away while maintaining high resolution near the origin. Another possibility is to take $f(\texttt{x1}) = a\,\texttt{x1} + b\,\texttt{x1}^3 + c\,\texttt{x1}^5$, where appropriate choices for the coefficients a, b, and c lead to increasing the relative coordinate density on a spherical shell, which is ideal for sampling a black hole horizon or neutron star crust.

The rescaling procedure developed in this paper also allows for the angular coordinates to be redistributed in a similar way, but we restrict our discussion to radial rescalings for the sake of simplicity. To be clear, by "radial," we refer to $\{\texttt{x1}, \texttt{x2}, \texttt{x3}\}$ in Cartesian-like, to $\{\texttt{x1}, \texttt{x3}\}$ (cylindrical radius and height) in cylindrical-like, and to $\{\texttt{x1}\}$ (radius) in spherical-like coordinates.

To construct the noncoordinate basis for use in the tensor component rescaling, we start with the flat metric in the $y^{(i)}$ Cartesian coordinate basis. Next, transform it to the coordinate basis of the uniform grid $x^{(i)}$ using the Jacobian matrix. We identify $\hat{\gamma}_{ij}$ with the flat metric in the $x^{(i)}$ basis. Then, the noncoordinate basis components are defined by Eq. (33). When the coordinate system is orthogonal, as is the case for all examples in Sec. V, then only three of the nine basis components are nonzero.

The coordinate system distributions and the corresponding basis components are summarized in Table II.

2. *Numerical representation of spatial derivatives*

The user-specified $n = N_{\rm FD}$ sets the order of the finite difference stencil approximation, so that the truncation error scales as

$$\mathcal{E}_{\rm FD} \sim \mathcal{O}\left(\Delta x^n \left|\partial_x^{n+1} u\right|\right) \tag{43}$$

for each of the 24 dynamical fields $u = \{\varepsilon_{ij}, \bar{A}_{ij}, W, K, \bar{\Lambda}^i, \alpha, \beta^i, B^i\}$. We demonstrate that finite-difference truncation errors converge to zero at the prescribed rates in Sec. V B 1. To calculate the finite difference stencil coefficients, NRPy+ inverts the corresponding linear system of Taylor series coefficients at user-specified order, akin to inverting the Vandermonde matrix for Lagrange polynomial interpolation [56]. Adopting a simple syntax, NRPy+ automatically replaces all spatial derivatives that appear in expressions with the appropriate finite difference approximation, at the desired order. Such spatial derivatives appear throughout the right-hand sides of the evolution system, Eqs. (11), (13), and (14), and diagnostics—including the BSSN constraint equations and ADM integrals (see Sec. IV B).

We use Kreiss-Oliger dissipation [57, 58] to diffuse unresolved, high-frequency modes that can reduce the convergence order. A standard high-order derivative operator $\mathcal{L}_{\rm KO}$ acting on the grid function as

$$\mathcal{L}_{\rm KO} u = -\epsilon_{\rm KO}\frac{(-1)^{n/2}}{2^n \Delta t}\left(\Delta x^i\right)^n \partial_i^n u \tag{44}$$

is added to the right-hand sides of the evolution equations (11), (13), and (14). Note that $\mathcal{L}_{\rm KO}$ is not a tensorial derivative in general, and its inclusion in the evolution equations violates spatial covariance. However, the coefficient of artificial dissipation is chosen such that the contribution of $\mathcal{L}_{\rm KO} u$ vanishes in the continuum limit. The dimensionless Kreiss-Oliger parameter $\epsilon_{\rm KO} = \epsilon_{\rm KO}\left(x^i\right)$ is allowed to vary smoothly over space, and typically approaches $\epsilon_{\rm KO} = 0.99$ in the weak field region. In particular, we often use a spherically symmetric transition function

$$\epsilon_{\rm KO}(r) = \frac{\epsilon_{\rm KO0}}{2}\left[\operatorname{erf}\left(\frac{r - r_{\rm KO}}{w_{\rm KO}}\right) + 1\right]\ , \tag{45}$$

where $\epsilon_{\rm KO0}, r_{\rm KO}, w_{\rm KO} > 0$ are constant parameters and r is a radial coordinate. We generally set this function to be less than 10^{-16} near the origin, so that its nonsmoothness at the origin is made irrelevant relative to nonsmoothness caused by roundoff error. In particular, we usually set $\epsilon_{\rm KO0} = 0.99$, $r_{\rm KO}/M = 2$, and $w_{\rm KO}/M = 0.17$.

B. Diagnostics

We employ a variety of diagnostics to monitor the accuracy of our calculations, as well as to probe the physical properties of the simulated spacetimes. Diagnostic routines fall broadly into two categories: diagnostics generated in NRPy+, and diagnostic routines within the ETK.

1. *Diagnostics generated by NRPy+*

The following describes the constraints, the ADM integrals, and a spherically symmetric horizon finder, which are the diagnostics written in Python in NRPy+. They contain spatial derivatives of the evolved fields, which are approximated by the automatically generated finite difference stencils. The resulting C code is evaluated by SENR during data output, after the time step iteration.

TABLE II. Summary of coordinate system choices. Finite difference operations take place on the uniformly sampled unit cube grid $x^{(i)} = (\texttt{x1}, \texttt{x2}, \texttt{x3})$, and are mapped to the nonuniformly sampled Cartesian grid $y^{(i)} = (\texttt{y1}, \texttt{y2}, \texttt{y3})$. The nontrivial scale factors constitute the basis, which is used to rescale tensor quantities. The function f allows the coordinates to be redistributed on the nonuniform grid, and the prime mark indicates differentiation with respect to the function argument. Although not shown here, our method allows for the more general case of a different redistribution function for each independent coordinate.

Coordinates	Definitions			Scale Factors		
	$\texttt{y1}$	$\texttt{y2}$	$\texttt{y3}$	$e^{(\texttt{x1})}_{\texttt{x1}}$	$e^{(\texttt{x2})}_{\texttt{x2}}$	$e^{(\texttt{x3})}_{\texttt{x3}}$
Cartesian-like	$f\left(\texttt{x1}\right)$	$f\left(\texttt{x2}\right)$	$f\left(\texttt{x3}\right)$	$f'\left(\texttt{x1}\right)$	$f'\left(\texttt{x2}\right)$	$f'\left(\texttt{x3}\right)$
Cylindrical-like	$f\left(\texttt{x1}\right)\cos\left(2\pi\,\texttt{x2}\right)$	$f\left(\texttt{x1}\right)\sin\left(2\pi\,\texttt{x2}\right)$	$f\left(\texttt{x3}\right)$	$f'\left(\texttt{x1}\right)$	$f\left(\texttt{x1}\right)$	$f'\left(\texttt{x3}\right)$
Spherical-like	$f\left(\texttt{x1}\right)\sin\left(\pi\,\texttt{x2}\right)\cos\left(2\pi\,\texttt{x3}\right)$	$f\left(\texttt{x1}\right)\sin\left(\pi\,\texttt{x2}\right)\sin\left(2\pi\,\texttt{x3}\right)$	$f\left(\texttt{x1}\right)\cos\left(\pi\,\texttt{x2}\right)$	$f'\left(\texttt{x1}\right)$	$f\left(\texttt{x1}\right)$	$f\left(\texttt{x1}\right)\sin\left(\pi\,\texttt{x2}\right)$

In terms of the BSSN variables (see Sec II), the Hamiltonian constraint takes the form [29]

$$\begin{aligned}\mathcal{H} &\equiv \frac{2}{3}K^2 - \bar{A}_{ij}\bar{A}^{ij} + e^{-4\phi}\left(\bar{R} - 8\bar{D}^i\phi\bar{D}_i\phi - 8\bar{D}^2\phi\right) \\ &= 0\,, \end{aligned} \tag{46}$$

where $\bar{R} = \bar{\gamma}^{ij}\bar{R}_{ij}$, and the momentum constraint is[3]

$$\begin{aligned}\mathcal{M}^i &\equiv e^{-4\phi}\left(\hat{D}_j\bar{A}^{ij} + 2\bar{A}^{k(i}\Delta^{j)}_{jk} + 6\bar{A}^{ij}\partial_j\phi - \frac{2}{3}\bar{\gamma}^{ij}\partial_j K\right) \\ &= 0\,. \end{aligned} \tag{47}$$

The Hamiltonian, momentum, and conformal connection coefficient (8) constraints are monitored throughout the simulation as a measure of numerical accuracy. In addition, the ADM surface integrals for total mass $M_{\rm ADM}$, linear momentum $P^i_{\rm ADM}$, and angular momentum $J^i_{\rm ADM}$ also serve as diagnostics. The integrands are evaluated on a spherical surface on the boundary of the spatial hypersurface at spatial infinity. Numerically, the integrals are approximated by two-dimensional Riemann sums on a spherical surface which is near the outer boundary, ideally in the weak field region. Supposing that the spacetime is asymptotically flat, and that the spacetime metric $g_{\mu\nu}$ approaches the Minkowski metric $\eta_{\mu\nu}$ at least as fast as $g_{\mu\nu} - \eta_{\mu\nu} = \mathcal{O}(1/r)$ when $r \to \infty$, then the ADM integrals take the form [42]

$$M_{\rm ADM} = \lim_{r\to\infty}\frac{1}{16\pi}\oint\gamma^{ij}\left(\partial_i\gamma_{kj} - \partial_k\gamma_{ij}\right)\sqrt{\gamma}\,\mathrm{d}S^k\,, \tag{48a}$$

$$P^i_{\rm ADM} = \lim_{r\to\infty}\frac{1}{8\pi}\oint\left(K^{ij} - \gamma^{ij}K\right)\sqrt{\gamma}\,\mathrm{d}S_j\,, \tag{48b}$$

$$J^i_{\rm ADM} = \lim_{r\to\infty}\frac{[ijk]}{8\pi}\oint y_j\left(K_{kl} - \gamma_{kl}K\right)\sqrt{\gamma}\,\mathrm{d}S^l\,. \tag{48c}$$

The vector components $\mathrm{d}S^i$ play the role of the outward-oriented surface element induced at spatial infinity, y^i are the components of a Cartesian coordinate vector, and $[ijk]$ is the totally antisymmetric Levi-Civita symbol. Note that the ADM integrals are not covariant as written, and they must be evaluated in asymptotically Cartesian coordinates. This allows us to easily interpret the directionality of the $P^i_{\rm ADM}$ and $J^i_{\rm ADM}$ components.

In the special case of a spherically symmetric configuration, the expansion of outgoing null geodesics Θ takes the simplified form [59]

$$\Theta(r) = \frac{4\bar{\gamma}_{\theta\theta}\partial_r\phi + \partial_r\bar{\gamma}_{\theta\theta}}{e^{2\phi}\bar{\gamma}_{\theta\theta}\sqrt{\bar{\gamma}_{rr}}} - 2\frac{\bar{K}_{\theta\theta}}{\bar{\gamma}_{\theta\theta}}\,. \tag{49}$$

The coordinate radius of the apparent horizon $r_{\rm H}$ is defined to satisfy $\Theta(r_{\rm H}) = 0$. Numerically, SENR evaluates Θ at every grid point using finite difference stencil C code generated by NRPy+. Then, it searches for the pair of neighbors that straddle $\Theta = 0$, and linearly interpolates between those two points to approximate $r_{\rm H}$.

2. Diagnostics provided by the Einstein Toolkit

A translation layer for the ETK is implemented in SENR, where the fields $\bar{\gamma}_{ij}$, e^{ϕ}, $\bar{A}_{ij}$, and K are interpolated onto a Cartesian grid and then converted to the ADM quantities γ_{ij} and K_{ij} in the Cartesian basis. These data are fed into the ETK to unlock a wide variety of diagnostic tools [60–62], including the horizon finder thorn AHFinderDirect [63] and the Ψ_4 gravitational waveform extraction thorn WeylScal4 [64, 65]. The measured Ψ_4 contains information relating to the gravitational wave strain in the transverse-traceless gauge and the weak field region via [42]

$$\Psi_4 = \ddot{h}_+ - i\ddot{h}_\times\,, \tag{50}$$

where h_+ and $h_\times$ are the gravitational wave strain amplitudes of the "plus" and "cross" polarization states, respectively, and the dots denote time derivatives.

[3] The term $\bar{A}^{ik}\Delta^j_{jk}$ in Eq. (47) of this paper is missing from the momentum constraint in Eq. (17) of [26] and Eq. (14) of [29]. In the notation of their respective articles, the expression in [26] is corrected by the substitution $g \to g/\tilde{g}$, and in [29] by $\bar{\gamma} \to \bar{\gamma}/\hat{\gamma}$. Be mindful of a parenthesis size mismatch in Eq. (14) of [29].

C. Initial data

NRPy+ implements initial data for zero- ($\gamma_{ij} = \hat{\gamma}_{ij}$), one-, and two-black-hole spacetimes. Single Kerr black hole initial data are available in UIUC conformally curved coordinates [52], Schwarzschild trumpet coordinates [53], and boosted Schwarzschild black holes in isotropic coordinates [66]. Two-black-hole initial data take the form of initial black holes at rest (Brill-Lindquist [51]). All implemented initial data solve the Hamiltonian (46) and momentum (47) constraints exactly. Expressions for all initial data evolved in this work are presented alongside their results in Sec. V.

Typically, these initial data types are most naturally represented in the spherical coordinate basis. The Jacobian matrix is used to transform the initial data from uniform spherical coordinates to the desired uniformly sampled $x^{(i)}$ grid. Finally, rank-1 and -2 tensors are transformed to the noncoordinate basis, according to the procedure in Sec. III.

D. Time integration

The evolution equations (11), (13), and (14) are all first-order-in-time partial differential equations that may be written in the form

$$\partial_t u(t) = \mathcal{L}\left(u(t), t\right) \ , \tag{51}$$

where $u = \{\varepsilon_{ij}, \bar{A}_{ij}, W, K, \bar{\Lambda}^i, \alpha, \beta^i, B^i\}$ is a vector composed of the 24 evolved fields. As can be seen from Eqs. (11), (13), and (14), the differential operator $\mathcal{L}$ depends on multiple components of u, as well as their first and second spatial derivatives. All spatial derivatives of the evolved fields in $\mathcal{L}$ are calculated using finite differences on the uniformly sampled $x^{(i)}$ grid.

To advance the grid function in time $u(t) \to u(t + \Delta t)$, we adopt the fourth-order[4] Runge-Kutta (RK4) method [69]

$$k_1 = \mathcal{L}\left(u(t), t\right) \ , \tag{52a}$$

$$u_1 = u(t) + \frac{\Delta t}{2} k_1 \ , \tag{52b}$$

$$k_2 = \mathcal{L}\left(u_1, t + \frac{\Delta t}{2}\right) \ , \tag{52c}$$

$$u_2 = u(t) + \frac{\Delta t}{2} k_2 \ , \tag{52d}$$

$$k_3 = \mathcal{L}\left(u_2, t + \frac{\Delta t}{2}\right) \ , \tag{52e}$$

$$u_3 = u(t) + \Delta t k_3 \ , \tag{52f}$$

$$k_4 = \mathcal{L}\left(u_3, t + \Delta t\right) \ , \tag{52g}$$

$$u(t + \Delta t) = u(t) + \frac{\Delta t}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \ . \tag{52h}$$

Being fourth order means that the error associated with the time stepping, at a fixed time, scales as $\mathcal{E}_{\rm RK4} \sim \mathcal{O}\left(\Delta t^4\right)$. Immediately after evaluating the RK4 steps given by Eqs. (52b), (52d), and (52f), boundary conditions are applied to u_1, u_2, and u_3, respectively. If the boundary conditions are time dependent, then they are applied at $t + \Delta t/2$ on the first two substeps and at a full step $t + \Delta t$ on the third substep. At the end of the full RK4 iteration, boundary conditions are applied to u at time $t + \Delta t$. Finally, the algebraic correction

$$\bar{\gamma}_{ij} \to \left(\frac{\hat{\gamma}}{\bar{\gamma}}\right)^{1/3} \bar{\gamma}_{ij} \tag{53}$$

is applied to the metric components to enforce the Lagrangian specification constraint (5), where $\bar{\gamma} = \hat{\gamma}$ is the conformal metric determinant on the initial slice.

When applying this standard, explicit RK4 algorithm, the Courant-Friedrichs-Lewy (CFL) condition [69] must be satisfied. For a numerical grid with coordinates $x^{(i)}$, SENR finds the smallest proper distance $\Delta s_{\rm min}$ along each of the independent coordinate directions

$$\Delta s_{\rm min} = \min\left(\sqrt{\bar{\gamma}_{11}}\Delta\mathtt{x1}, \sqrt{\bar{\gamma}_{22}}\Delta\mathtt{x2}, \sqrt{\bar{\gamma}_{33}}\Delta\mathtt{x3}\right) \ , \tag{54}$$

where Δx^i is the uniform grid spacing between adjacent points in the x^i direction, and $\bar{\gamma}_{ii}$ is evaluated at x^i. Then the time step is

$$\Delta t = C \Delta s_{\rm min} \ , \tag{55}$$

where the Courant factor is set to $C = 0.5$ for all simulations presented here.

More explicitly, the CFL-limited time step varies with grid resolution in a nontrivial way depending on the coordinate choice. Suppose that we vary the number of grid points simultaneously in all three coordinates. Then for Cartesian coordinates $\Delta t \propto \Delta x^i$, for cylindrical coordinates $\Delta t \propto \left(\Delta x^i\right)^2$, and for spherical coordinates $\Delta t \propto \left(\Delta x^i\right)^3$. The higher-order dependence in the case of cylindrical and spherical coordinates is due to the focusing of grid points along the symmetry axis or near the

[4] See Ref. [67]. See also [68] for details on the stability properties of the fully explicit Runge-Kutta methods at various order. The widely used RK4 method is conditionally stable for this application, although PIRK4 might allow for similar accuracy with larger timesteps.

origin. This CFL restriction can be softened significantly by clever choice of coordinate redistribution function f (see Sec. IV A 1).

E. Boundary conditions

As described in Sec. IV A, SENR/NRPy+ maps a uniformly sampled unit cube grid to a nonuniformly distributed Cartesian coordinate grid, chosen to efficiently sample the space. Our grids are cell centered, so no points exist precisely on any of the faces of the unit cube. However, for the points that are nearest to the faces, but inside the cube, the finite difference stencils for spatial derivatives will reach outside of the cube. To ensure that these stencils correspond to valid data, we add a collection of points to a shell region exterior to the cube called the "ghost zone." If $N_{\rm G}$ ghost zone points are needed outside each boundary, then this would increase the total number of points in the grid from $N_{\tt x1} \times N_{\tt x2} \times N_{\tt x3}$ to $(N_{\tt x1} + 2N_{\rm G}) \times (N_{\tt x2} + 2N_{\rm G}) \times (N_{\tt x3} + 2N_{\rm G})$.

Prior to evaluation of the right-hand sides of the BSSN equations, these ghost zone points must be filled. Some ghost zone points, including those at $\varphi < 0$ or $\varphi > 2\pi$ on spherical- or cylindrical-like coordinate grids, map to points *inside* the cube; we call these *inner* boundaries. The remaining ghost zone points map back to points outside the interior of the cube (e.g., $r > r_{\rm max}$ in spherical-like coordinate grids); we call these *outer* boundaries.

Outer boundary ghost zone points may be filled in accordance with the desired outer boundary condition. Although the widely used Sommerfeld outer boundary condition is also implemented, we find the simple quadratic extrapolation condition to be quite effective on spherical-like coordinate grids

$$u(x) \approx 3u(x - \Delta x) - 3u(x - 2\Delta x) + u(x - 3\Delta x) \ , \quad (56)$$

where u is any of the evolved fields and x is the coordinate $x^{(i)}$ perpendicular to the boundary. As with any approximate boundary condition, this condition produces unwanted ingoing modes that contaminate the interior of the simulation. In practice, logarithmically spaced radial coordinates enable us to push the outer boundary out of causal contact with the origin for as long as we care to simulate.

Inner boundary conditions depend on the coordinate system, and must account for intrinsic periodic, axial, and radial symmetries. The coordinate redistribution function f (see Sec. IV A 1) is required to be odd, which ensures that ghost zone points across inner boundaries coincide exactly with other points on the grid interior, respecting the desired symmetries.

In the case of scalar functions, these symmetry conditions simply copy the appropriate values of the function from the grid interior to its ghost zone partner. Vectors and higher rank tensors, however, are sensitive to changes of sign in the basis when evaluated across inner boundaries. In that case, an appropriate change of sign must be copied into the ghost zone along with the function value itself. We refer to these changes of sign as *parity conditions*.

In the following, we will show by example the symmetry and parity conditions specific to the spherical coordinate topology. Then a generic algorithm for assigning ghost zone values in arbitrary coordinates is described.

1. Spherical boundary conditions example

To derive the boundary conditions appropriate for a coordinate system, first express those coordinates in terms of a system whose boundaries are well understood. To this end, we choose ordinary Cartesian coordinates on the domain $-\infty < x, y, z < \infty$. Since each coordinate is unbounded from both above and below, there are no inner boundaries. In addition, every point on the computational grid has a unique (x, y, z) label.

Now consider ordinary spherical coordinates, which are related to the Cartesian coordinates in the usual way [Eq. (19)]. The spherical coordinate domain is bounded by $0 < r < \infty$, $0 < \theta < \pi$, and $0 < \varphi < 2\pi$. In this case, there is only one outer boundary, corresponding to $r \to \infty$; what remains are the five inner boundaries.

To find the symmetry conditions for these inner boundaries, first recall that a scalar function g has a particular value at some location, regardless of the underlying coordinate choice. Next, evaluate the function in the spherical coordinate ghost zone, identify the corresponding Cartesian coordinate values, and then find the point in the spherical grid interior that corresponds to those same Cartesian coordinates. This links each point in the ghost zone to a unique point in the grid interior.

By this procedure, the five symmetry conditions for the spherical coordinate inner boundaries are found to be

$$g(-r, \theta, \varphi) = g(r, \pi - \theta, \pi + \varphi) \ , \quad (57a)$$

$$g(r, -\theta, \varphi) = g(r, \theta, \pi + \varphi) \ , \quad (57b)$$

$$g(r, \pi + \theta, \varphi) = g(r, \pi - \theta, \pi + \varphi) \ , \quad (57c)$$

$$g(r, \theta, -\varphi) = g(r, \theta, 2\pi - \varphi) \ , \quad (57d)$$

$$g(r, \theta, 2\pi + \varphi) = g(r, \theta, \varphi) \ . \quad (57e)$$

These correspond to radial symmetry about the origin [Eq. (57a)], axial symmetry about the north [Eq. (57b)] and south [Eq. (57c)] poles, and periodic symmetry around the axis in the negative [Eq. (57d)] and positive [Eq. (57e)] orientations. For any point in the ghost zone, there exists a combination of the inner boundary symmetry rules (57) that maps that point to either the grid interior or the outer boundary. Note that these symmetries refer only to the coordinate distribution, and not the evolved fields, which are allowed to be completely asymmetrical.

As mentioned above, these symmetry conditions are sufficient for filling the inner boundary ghost zone points of a scalar function. For the case of vectors and higher rank tensors, however, parity conditions must also be taken into account.

The needed parity conditions are found by comparing the basis vectors in the ghost zone to their counterparts in the grid interior. Again, we express the basis vectors in terms of a basis in which the parity conditions are well understood. Since the Cartesian basis has no inner boundaries, the Cartesian basis has only the trivial parity conditions, which is to say that there are no changes of sign. Start with the noncoordinate spherical basis $\boldsymbol{e}_{(a)}$ (25) which is expressed in terms of the spherical coordinate basis $\frac{\boldsymbol{\partial}}{\boldsymbol{\partial} r^{(a)}}$. Then, transform the spherical coordinate basis to the Cartesian basis (22)

$$\boldsymbol{e}_{(r)} = \frac{\partial x^i}{\partial r} \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} x^{(i)}} \ , \tag{58a}$$

$$\boldsymbol{e}_{(\theta)} = \frac{1}{r} \frac{\partial x^i}{\partial \theta} \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} x^{(i)}} \ , \tag{58b}$$

$$\boldsymbol{e}_{(\varphi)} = \frac{1}{r\sin(\theta)} \frac{\partial x^i}{\partial \varphi} \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} x^{(i)}} \ . \tag{58c}$$

Remembering $\boldsymbol{e}_{(i)} = \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} x^{(i)}}$ in Cartesian coordinates and contracting with the inverse of the Jacobian matrix (20) results in

$$\begin{aligned} \boldsymbol{e}_{(r)}(r,\theta,\varphi) = {} & \sin(\theta)\cos(\varphi)\boldsymbol{e}_{(x)} + \sin(\theta)\sin(\varphi)\boldsymbol{e}_{(y)} \\ & + \cos(\theta)\boldsymbol{e}_{(z)} \ , \end{aligned} \tag{59a}$$

$$\begin{aligned} \boldsymbol{e}_{(\theta)}(r,\theta,\varphi) = {} & \cos(\theta)\cos(\varphi)\boldsymbol{e}_{(x)} + \cos(\theta)\sin(\varphi)\boldsymbol{e}_{(y)} \\ & - \sin(\theta)\boldsymbol{e}_{(z)} \ , \end{aligned} \tag{59b}$$

$$\boldsymbol{e}_{(\varphi)}(r,\theta,\varphi) = -\sin(\varphi)\boldsymbol{e}_{(x)} + \cos(\varphi)\boldsymbol{e}_{(y)} \ , \tag{59c}$$

where we indicate on the left-hand side the explicit functional dependence of the spherical noncoordinate basis vectors on (r,θ,φ). To find the parity conditions, compute the dot product between a basis vector in the ghost zone and its partner in the grid interior, according to Eq. (57). If the coordinate system is properly constructed, then this dot product should always evaluate to ± 1, where the negative case indicates that the basis changes sign in the ghost zone.

TABLE III. Cylindrical- and spherical-like parity conditions for rank-1 and rank-2 tensors. “Radial” refers to the parity across $r = 0$, “Axial” refers to the parity across $\rho = 0$ or $\sin(\theta) = 0$, and “Periodic” refers to the parity across $\varphi = 0$ or $\varphi = 2\pi$. Compare with Table I in [29].

Coordinates	Component	Radial	Axial	Periodic
Cylindrical-like	ρ		$-$	$+$
	φ		$-$	$+$
	z		$+$	$+$
	$\rho\rho$		$+$	$+$
	$\rho\varphi$		$+$	$+$
	ρz		$-$	$+$
	$\varphi\varphi$		$+$	$+$
	φz		$-$	$+$
	zz		$+$	$+$
Spherical-like	r	$-$	$+$	$+$
	θ	$+$	$-$	$+$
	φ	$-$	$-$	$+$
	rr	$+$	$+$	$+$
	$r\theta$	$-$	$-$	$+$
	$r\varphi$	$+$	$-$	$+$
	$\theta\theta$	$+$	$+$	$+$
	$\theta\varphi$	$-$	$+$	$+$
	$\varphi\varphi$	$+$	$+$	$+$

For the current example, we find

$$\boldsymbol{e}_{(r)}(-r,\theta,\varphi) \cdot \boldsymbol{e}_{(r)}(r,\pi-\theta,\pi+\varphi) = -1 \ , \tag{60a}$$

$$\boldsymbol{e}_{(\theta)}(-r,\theta,\varphi) \cdot \boldsymbol{e}_{(\theta)}(r,\pi-\theta,\pi+\varphi) = +1 \ , \tag{60b}$$

$$\boldsymbol{e}_{(\varphi)}(-r,\theta,\varphi) \cdot \boldsymbol{e}_{(\varphi)}(r,\pi-\theta,\pi+\varphi) = -1 \ , \tag{60c}$$

$$\boldsymbol{e}_{(r)}(r,-\theta,\varphi) \cdot \boldsymbol{e}_{(r)}(r,\theta,\pi+\varphi) = +1 \ , \tag{60d}$$

$$\boldsymbol{e}_{(\theta)}(r,-\theta,\varphi) \cdot \boldsymbol{e}_{(\theta)}(r,\theta,\pi+\varphi) = -1 \ , \tag{60e}$$

$$\boldsymbol{e}_{(\varphi)}(r,-\theta,\varphi) \cdot \boldsymbol{e}_{(\varphi)}(r,\theta,\pi+\varphi) = -1 \ , \tag{60f}$$

$$\boldsymbol{e}_{(r)}(r,\pi+\theta,\varphi) \cdot \boldsymbol{e}_{(r)}(r,\pi-\theta,\pi+\varphi) = +1 \ , \tag{60g}$$

$$\boldsymbol{e}_{(\theta)}(r,\pi+\theta,\varphi) \cdot \boldsymbol{e}_{(\theta)}(r,\pi-\theta,\pi+\varphi) = -1 \ , \tag{60h}$$

$$\boldsymbol{e}_{(\varphi)}(r,\pi+\theta,\varphi) \cdot \boldsymbol{e}_{(\varphi)}(r,\pi-\theta,\pi+\varphi) = -1 \ , \tag{60i}$$

$$\boldsymbol{e}_{(r)}(r,\theta,-\varphi) \cdot \boldsymbol{e}_{(r)}(r,\theta,2\pi-\varphi) = +1 \ , \tag{60j}$$

$$\boldsymbol{e}_{(\theta)}(r,\theta,-\varphi) \cdot \boldsymbol{e}_{(\theta)}(r,\theta,2\pi-\varphi) = +1 \ , \tag{60k}$$

$$\boldsymbol{e}_{(\varphi)}(r,\theta,-\varphi) \cdot \boldsymbol{e}_{(\varphi)}(r,\theta,2\pi-\varphi) = +1 \ , \tag{60l}$$

$$\boldsymbol{e}_{(r)}(r,\theta,2\pi+\varphi) \cdot \boldsymbol{e}_{(r)}(r,\theta,\varphi) = +1 \ , \tag{60m}$$

$$\boldsymbol{e}_{(\theta)}(r,\theta,2\pi+\varphi) \cdot \boldsymbol{e}_{(\theta)}(r,\theta,\varphi) = +1 \ , \tag{60n}$$

$$\boldsymbol{e}_{(\varphi)}(r,\theta,2\pi+\varphi) \cdot \boldsymbol{e}_{(\varphi)}(r,\theta,\varphi) = +1 \ . \tag{60o}$$

The same procedure can be used to determine the symmetry and parity conditions for cylindrical-like coordinates. The parity conditions for cylindrical- and spherical-like coordinates across each inner boundary are summarized in Table III.

Operationally, vector and tensor component values are copied to the ghost zone using the symmetry condition.

For every basis vector attached to each tensor component, compensate for any changes in sign in the basis by applying the parity condition.

2. *The general boundary condition procedure*

The procedure described in the previous section is generalized to construct an explicit routine for filling inner and outer boundary ghost zone points in other coordinate systems. Here we review the routine, which was developed for SENR/NRPy+ to validate the special, coordinate-specific inner and outer boundary condition routines developed for cylindrical- and spherical-like grids.

Starting with the mapping between Cartesian coordinates and the coordinates of interest [e.g., Eq. (19)], the routine automatically classifies the ghost zones as inner or outer boundaries, maps each inner ghost zone point to its partner on the grid interior, constructs the noncoordinate basis vectors, and determines the parity conditions. The symmetry and parity conditions are compiled into a list at the start of a simulation, and the routine later runs through the list to apply the boundary conditions as needed.

The ghost zone points are filled, in order, outward from the interior grid. This is of particular importance at the outer boundary, where the one-sided quadratic extrapolation stencil equation (56) requires that outer boundary ghost zone points be filled in an outward-going direction. In addition, for cases of high symmetry and large $N_{\rm FD}$, it is possible for the finite difference stencil to be wider than the grid interior, which means that a ghost zone point could be mapped to another ghost zone point. Assigning the boundary values in an outward fashion avoids attempts to copy to the ghost zone from uninitialized memory.

V. RESULTS

In this section we present a number of validation and verification tests performed on the SENR/NRPy+ code. In Sec. V A, we compare its results with two other numerical relativity codes, and demonstrate that, e.g., differences in results between SENR/NRPy+ and the code of Baumgarte *et al.* [29] are at the level of roundoff error in the case of ordinary spherical coordinate evolutions of a strongly perturbed Minkowski spacetime. In Sec. V B, we demonstrate in the contexts of a single, nonspinning black hole and a double black hole head-on collision that the finite difference truncation error converges to zero with increasing grid resolution at the expected rate, and that increasing finite difference order with fixed grid resolution results in near-exponential convergence of the error.

These tests also act to showcase the efficiency of SENR/NRPy+ against other open-source numerical relativity codes in the context these physical scenarios, as all tests were performed on desktop-scale computers, using at most only about 1.5 GB of RAM.

A. Code comparisons

Here we directly compare SENR/NRPy+ results with two other established BSSN evolution codes. These tests verify that all of the evolution equations and diagnostics are implemented correctly, and that the simulations do indeed contain black hole horizons with the expected properties.

1. *Robust stability test: Roundoff-level agreement between SENR/NRPy+ and BMCCM*

This section compares SENR/NRPy+ to the spherical-polar, spatially fourth-order finite differenced BSSN code of Baumgarte *et al.* [29] (hereafter BMCCM). BMCCM includes many features beyond the scope of this paper [70–74]; here we focus on initial data that represent a version of the robust stability test bed [33–36], which involves a strong random perturbation about flat spacetime. At each grid point, the random number generator `drand48` [75] is seeded with a unique (but constant) integer tied to the grid index. Then, each of the grid functions are populated, in turn, with Minkowski initial data ($e^{\phi} = 1$, $\varepsilon_{ij} = 0$, $\bar{A}_{ij} = 0$, $K = 0$, $\alpha = 1$, $\beta^i = 0$, and $B^i = 0$) plus a random value picked from the uniform distribution $[-0.02, 0.02]$. This produces repeatable initial data with no spatial correlation, and tests every aspect of the evolution and diagnostic algorithms.

Standard second-order Runge-Kutta (RK2) integration is unconditionally unstable in the presence of coordinate singularities. To circumvent this problem, BMCCM evolves the BSSN fields using the second-order partially implicit Runge-Kutta (PIRK2) integrator [76] (see [68] for details of the method and its higher-order generalizations). The PIRK2 method treats all of the regular terms in the evolution equations explicitly, as in RK2, but evolves the singular terms by an implicit step that depends on the updated values of the regular parts. This technique does not require any analytical or numerical inversion, and so its cost is comparable to that of fully explicit schemes. Although by default SENR implements RK4, which *is* stable in the presence of coordinate singularities [68], we also implement PIRK2 in SENR to directly compare results with the established BMCCM code.

We compare between SENR/NRPy+ and BMCCM the L_2 norm of the Hamiltonian violation over the entire grid, distilling an entire grid's worth of data down to a single number at each time. Double precision floating point arithmetic maintains approximately 16 digits of significance in any mathematical operation, limiting the extent to which it can be said that two algorithms are in numer-

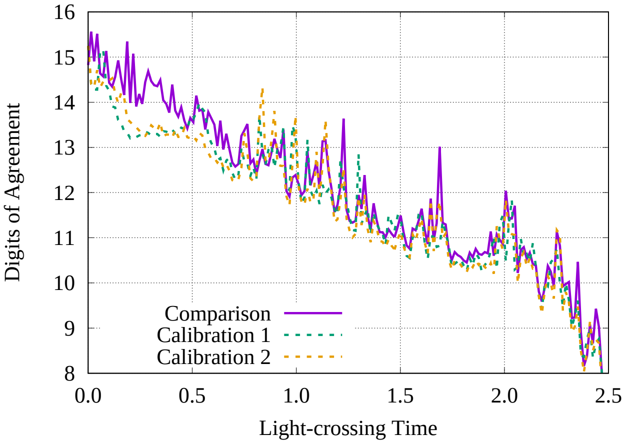

FIG. 1. Digits of agreement between SENR/NRPy+ and BMCCM evolving random initial data, measuring the L_2 norm of the Hamiltonian constraint over the entire grid. The initial data are a random perturbation about flat spacetime of maximum magnitude 0.02. The calibration runs compare BMCCM with itself randomly perturbed, initially, in the least significant digit. Both codes terminate due to NaNs at exactly the same iteration.

ical agreement. Over the course of a simulation, errors at the 16th significant digit will gradually rise—an unavoidable phenomenon when using finite-precision arithmetic known as roundoff error. If two codes are shown to produce results that agree to within roundoff error, they are functionally identical, so that if one code has been proven robust, then the other code possesses identical robustness.

The comparison is calibrated by first running BMCCM with the random initial data. Then, it is run again with identical initial data, except that the least significant digit at every point is reset to a random number. This tiny initial difference grows over time due to roundoff error. Given the strong, discontinuous nature of the random perturbations away from flat space, infinities and NaNs eventually develop, causing the simulations to terminate. For good measure, we perform the calibration a second time, starting with an identical perturbation of Minkowski, but re-randomizing the least significant digit for each function at every grid point. The results diverge from the base case in a similar fashion, shown as dashed lines in Fig 1.

Finally, we run the initial data of the BMCCM base case through SENR/NRPy+ on an identical grid. The resulting differences are illustrated as a solid line in Fig 1. SENR/NRPy+ maintains significance at least as well as the BMCCM calibrations, indicating the two codes agree to within roundoff error. Thus the results of the SENR/NRPy+ and BMCCM codes are numerically indistinguishable.

2. *SENR/NRPy+ and the Einstein Toolkit: Comparison of nonspinning puncture black hole evolutions*

In this section, results from SENR/NRPy+ are compared with the ETK in the context of nonspinning black hole evolutions, tracking the apparent horizon radii. The initial data represent a single wormhole slice of the Schwarzschild black hole spacetime in isotropic coordinates. These data are conformally flat ($\varepsilon_{ij} = 0$), maximally sliced ($K = 0$), and exist at a moment of time symmetry ($\bar{A}_{ij} = 0$). The conformal factor is the solution to the flat space Laplace equation when the asymptotic flatness condition $\lim_{r\to\infty} e^{\phi} = 1$ is imposed at infinity

$$e^{\phi} = 1 + \frac{M}{2r} \; , \tag{61}$$

where r is the isotropic radial coordinate distance to the puncture located at the origin. The constant of integration is chosen such that the total ADM energy (48a) of the slice is $M_{\rm ADM} = M$. As initial gauge conditions, we use the popular “pre-collapsed” lapse [30]

$$\alpha = e^{-2\phi} \tag{62}$$

and vanishing shift $\beta^i = 0$ and $B^i = 0$. The shift evolution damping parameter η (14b) influences the coordinate radius of the black hole apparent horizon. We found that our choice of $\eta = 0.25/M$ results in a horizon that quickly settles down to a static state.

In the ETK simulation, we make use of the open-source Cactus/Carpet [13, 14] AMR infrastructure to place a single wormhole black hole at the origin, surrounded by 8 levels of mesh refinement. The AMR grids are Cartesian and adopt a Cartesian basis. Each refinement level contains the same number of points (excluding ghost and AMR buffer zones), and the grid spacing doubles each time a refinement boundary is crossed (starting from the origin and moving outward). The grid outer boundary is set to $r_{\rm max}/M = 128$. There are 32 uniformly spaced grid points across each refinement level in each coordinate direction, giving 16 points across the horizon when at its smallest (on the initial slice). On the finest refinement level, the grid spacing along each of the coordinate directions is $\Delta r_{\rm min}/M = 0.03125$. RK4 time evolution (the method of lines) and sixth-order finite difference stencils are adopted, with upwinding on the shift-advection terms. The BSSN equation C code in the McLachlan BSSN thorn [15] is automatically generated using the Mathematica-based Kranc code [48]. The black hole apparent horizon radius is measured from the evolved fields using the AHFinderDirect thorn [63].

In SENR/NRPy+, we adopt the same initial data, but place it on a single spherical grid with coordinate redistribution function

$$f(\texttt{x1}) = r_{\rm max} \frac{\sinh\left(\texttt{x1}/w\right)}{\sinh\left(1/w\right)} \; , \tag{63}$$

with all tensorial variables expressed in the spherical basis. The grid parameters are tuned to match both the outer boundary of the ETK simulation, as well as the minimum grid spacing at the black hole. The ETK AMR grid has $32 \times 8 = 256$ (192 nonoverlapping) points along each Cartesian coordinate direction. Along the diagonal, the resolution is effectively reduced to $256/\sqrt{3} \approx 148$ points. For the SENR grid, we allocate $N_{\texttt{x1}} = 148$ and $N_{\texttt{x2}} = N_{\texttt{x3}} = 2$ (the minimum number of angular grid points required by our boundary condition module), let $w = 0.173435$, and set the outer boundary $r_{\rm max}/M = 128$. This results in $\Delta r_{\rm min}/M \approx 0.03123$, agreeing well with the ETK grid. The black hole apparent horizon radius is measured from the evolved fields by hunting for the root of the null expansion (see Sec. IV B 1 for details). We adopt $N_{\rm FD} = 6$ and RK4 time integration in SENR/NRPy+ to match ETK.

Despite tuning the grid resolutions and basic numerical evolution strategies to be consistent across codes, the chosen shift condition in both codes is not covariant [note the partial derivatives appearing in Eqs. (14) and (15), and see discussion in Brown [26] for how to make this shift condition covariant]. Thus we should not in general expect results that agree between the codes, as they do not adopt the same coordinate system. For spherically symmetric spacetimes, however, the partial derivatives ∂_i in Eqs. (14) and (15) can be replaced with the covariant derivative $\hat{D}_i$ in both spherical and Cartesian coordinates, showing that, in this special case, the gauge conditions are geometrically identical.

In Fig. 2, we monitor the coordinate radius of a puncture black hole as an indicator of the spacetime field and shift dynamics (obviously, the shift condition directly influences the coordinate radius of a puncture black hole). Notice that the apparent horizon radii measured by SENR/NRPy+ and ETK *match extremely well over time*, starting from the expected initial coordinate radius of the wormhole throat $r_{\rm H}/M = 0.5$ and equilibrating to a trumpet coordinate radius of $r_{\rm H}/M \approx 0.883$.

We find that the constraint violation in SENR/NRPy+ versus radius is typically below the level observed in the ETK simulation by about 2 orders of magnitude, and at worst the two share the same level of violation.

Even though results between the two codes in the strong-field region agree extremely well (Fig. 2), and the magnitude of constraint violations is significantly smaller in SENR/NRPy+, the ETK simulation requires approximately 10 GB of RAM, whereas the SENR/NRPy+ simulation needs only 28 MB—about 0.28% of the ETK simulation. Of course this is due to the ability of SENR/NRPy+ to take advantage of the spherical symmetry in the spherical basis, while the ETK simulation models the black hole on nested Cartesian AMR grids. Note that in spite of the spherical symmetry, this is still a full 3+1 simulation for SENR, just with very few points sampling the angular directions.

Having demonstrated excellent agreement with the BMCCM and ETK codes, we next turn our attention to code validation tests, in which numerical errors in SENR/NRPy+ are demonstrated to converge to zero as expected.

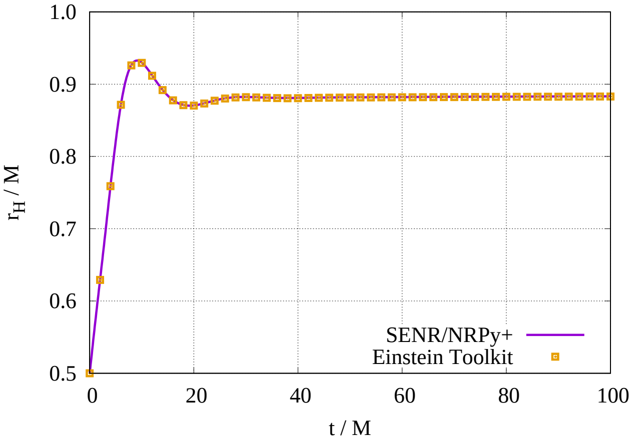

FIG. 2. Single black hole wormhole initial data evolved in the standard gauge with sixth-order finite differencing: comparison between SENR/NRPy+ and the ETK. Evolution of the apparent horizon coordinate radius as measured in SENR/NRPy+ by finding the root of the null expansion (49), and in the ETK using the AHFinderDirect thorn [63].

B. Convergence tests

Section V B 1 shows that finite difference truncation errors converge to zero with increasing grid resolution at the expected rates in SENR/NRPy+. We show in Sec. V B 2 that the truncation error converges to zero nearly exponentially with linear increase in the finite difference order, keeping the numerical grids held fixed. We explore the convergence behaviors of physical quantities extracted from the evolved fields. Then, in Sec. V B 3 we evolve the dynamical head-on collision from rest of two nonspinning black holes, and confirm that the ringdown of the merged black hole gravitational waveform matches the analytical prediction both in frequency and amplitude.

1. *Convergence of puncture black hole evolutions*

The BSSN equations are solved on the uniformly sampled $x^{(i)}$ grid, which directly maps to the solution on the nonuniformly sampled, Cartesian $y^{(i)}$ grid. In particular, we use the linear coordinate redistribution function

$$f(\texttt{x1}) = r_{\rm max}\,\texttt{x1}\,, \tag{64}$$

and similarly for `x2` or `x3`, depending on the choice of coordinates. Numerical errors in solving these equations stem largely from the finite difference representation of spatial derivatives (i.e., truncation errors of these derivatives are typically dominant). Finite difference operators effectively fit a polynomial to a function sampled at a fixed number of neighboring points, so that the derivative of the polynomial acts as an approximation to the exact derivative. Truncation error—i.e., the error caused by approximating functions with finite polynomials of degree D—drops as the sample rate to some power $N_{\rm FD}$ that is related to D. In our finite difference schemes, $N_{\rm FD} = D$.

In this section, we confirm that in fact the error drops in proportion to our underlying uniform grid spacing $\mathcal{E}_{\rm FD} \sim \mathcal{O}\left(\Delta x^{N_{\rm FD}}\right)$, in the context of a single, nonspinning puncture black hole evolved using the BSSN formalism, in which $N_{\rm FD} \in \{2, 4, 6\}$. We monitor the Hamiltonian constraint violation. With uniform resolution, it becomes very expensive to simultaneously resolve the apparent horizon and push the outer boundary far from the puncture. This limits the total run time before error from the outer boundary contaminates the horizon.

As an alternative approach, we perform convergence tests in which the grid resolution is held fixed and the finite difference order is allowed to vary. In this way, we increase the degree of the finite difference polynomial that is fit to each function at each point. Therefore, we expect the convergence rate to be approximately exponential, provided roundoff error is sufficiently small and the underlying functions are smooth (see, e.g., [77] for additional discussion of “exponential” convergence).

These tests adopt the same isotropic wormhole initial data and initial gauge conditions as in Sec. V A 2. We set the outer boundary to $r_{\rm max}/M = 10$. The fastest waves on the grid (related to the 1+log lapse condition; see, e.g., [19, 46]) propagate, with speed $\sqrt{2\alpha}$, inward from the outer boundary and outward from the puncture. These simulations end at $t/M \approx 5$.

In Fig. 3, we show that the finite difference truncation errors converge at the expected rates in all three coordinate system classes by rescaling the higher resolution data by a constant factor $(N_i/200)^{N_{\rm FD}}$, where $N_i \in \{200, 254, 322, 410\}$ represents the number of grid points (in the nonangular coordinate directions) in each run. We let $N_{\rm FD} = 2$ in the Cartesian case, $N_{\rm FD} = 4$ in the cylindrical case, and $N_{\rm FD} = 6$ in the spherical case. The lack of convergence at $r/M \gtrsim 6$ is due to the approximate outer boundary conditions; in practice we push outer boundaries out of contact from the physical system of interest via a simple logarithmic radial rescaling of the underlying coordinate system.

Remarkably, in Fig. 4, we find that while the Hamiltonian constraint violation is *anticonvergent* inside the horizon ($r/M \lesssim 0.5$), we maintain approximately exponential convergence in a sizable region *outside* the horizon, hitting roundoff error at eighth order. We have observed nonconvergent behavior propagating outside the horizon only in cases when the total number of points inside the horizon are set to be so small that finite difference stencils outside the horizon touch grid points immediately surrounding the puncture.

More alarming than the puncture itself, the sharp lapse wave (a “gauge shock” [78, 79]) that propagates

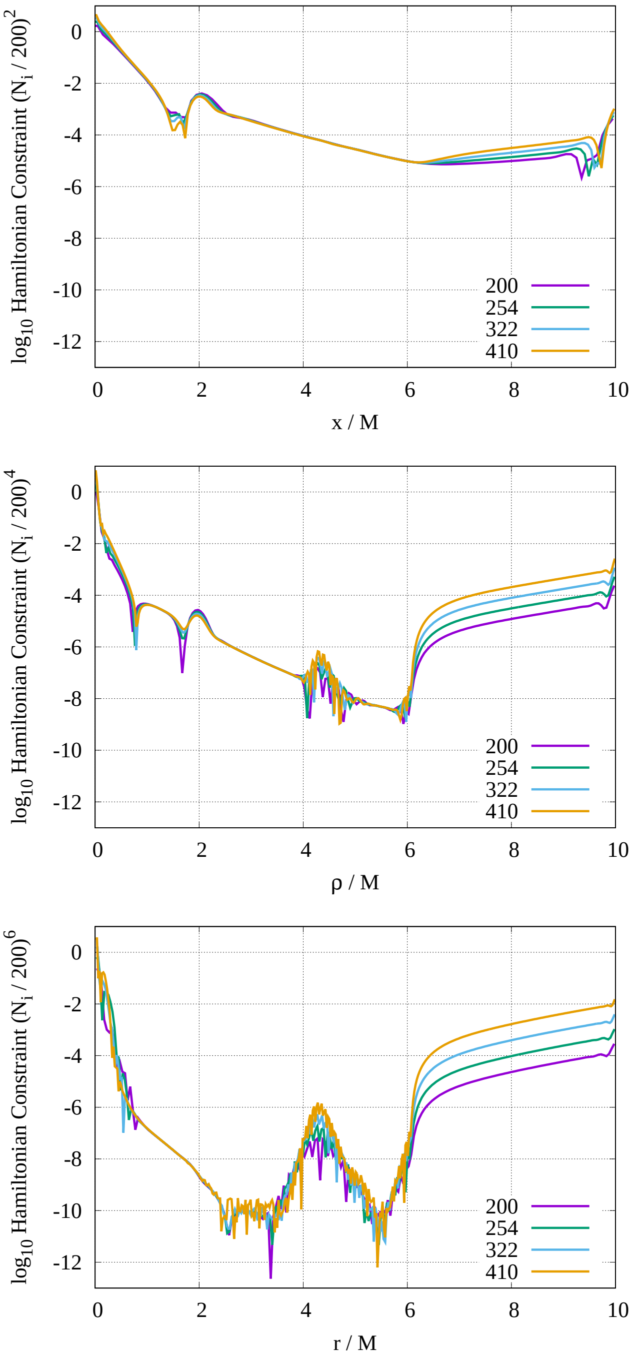

FIG. 3. Single black hole wormhole initial data evolved in the standard gauge: convergence of truncation errors to zero of Hamiltonian constraint violation in linearly distributed Cartesian (**top panel**), cylindrical (**middle panel**), and spherical coordinates (**bottom panel**). The legends indicate the number of grid points N_i in each (nonangular) direction. The data shown here are scaled by the factor $(N_i/200)^{N_{\rm FD}}$. Data are measured along a radial line at $t/M \approx 5$. Cartesian, cylindrical, and spherical coordinate evolutions are performed with $N_{\rm FD} = 2$, 4, and 6, respectively. We adopt the linear redistribution function parameter $r_{\rm max}/M = 10$ [Eq. (64)].

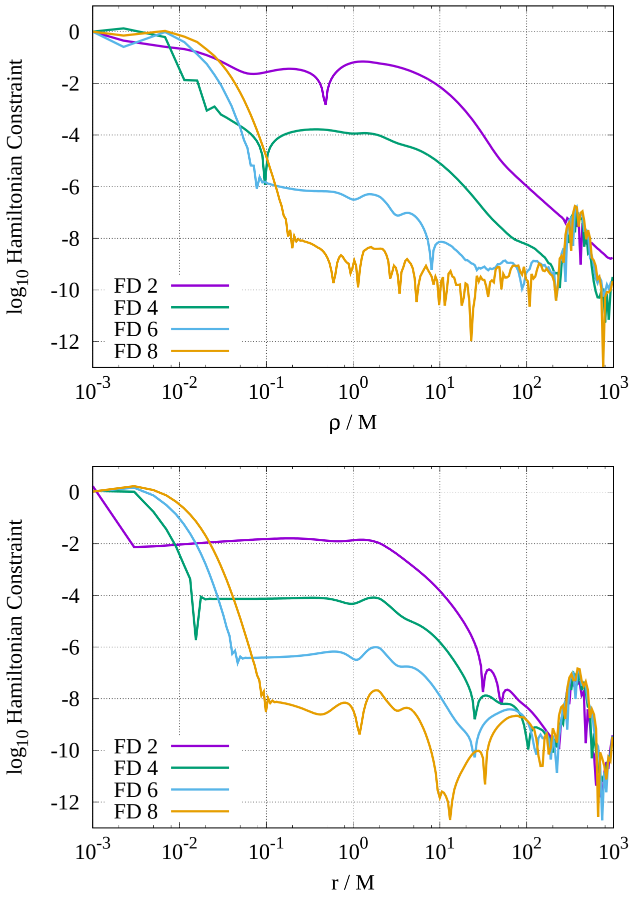

FIG. 4. Single black hole wormhole initial data evolved in the standard gauge: convergence of truncation errors to zero of Hamiltonian constraint violation in sinh-cylindrical (**top panel**) and sinh-spherical (**bottom panel**) coordinates. Numerical grids are held fixed at moderate resolution, and only finite difference order is increased. Hamiltonian constraint violation is measured along a radial line at $t/M \approx 300$. As the nonconvergent gauge wave pulse propagates outward from the puncture, exponential convergence of the Hamiltonian is restored in its wake. For these simulations, we take $\eta = 2/M$. We adopt redistribution parameters $r_{\rm max}/M = 1000$ and $w = 0.0916845$ [Eq. (63)]. The spherical-like grid uses $N_{\rm x1} = 200$ and $N_{\rm x2} = N_{\rm x3} = 2$ points, and the cylindrical-like grid uses $N_{\rm x1} = 200$, $N_{\rm x2} = 2$, and $N_{\rm x3} = 400$.

outward from the horizon in moving puncture simulations is another source of nonconvergence. Remarkably, exponential-like convergence is restored in the wake of this gauge wave pulse; in Fig 4, the pulse has reached $r/M \approx 425$ at the time of measurement. Restoration of convergence after the gauge pulse is very difficult to achieve on Cartesian AMR grids, as sharp outgoing waves are partially reflected off of refinement boundaries. (See, e.g., [19] for discussion of how this problem might be mitigated.)

Based on these results, we anticipate much cleaner exponential-like convergence in black hole evolutions for which this gauge pulse does not exist. Next we explore just such a case: evolutions of trumpet black hole initial data.

2. *Convergence of evolved static trumpet initial data*

The trumpet solution represents a time-independent slicing of the Schwarzschild spacetime [53]. In this section, we adopt the trumpet solution to show that the numerical evolution outside the horizon is completely dominated by truncation error.

The trumpet data are conformally flat ($\varepsilon_{ij} = 0$), and describe a single black hole with mass M. With the choice $f_1 = R_0 = M$ in [53], the trumpet conformal factor is

$$e^{\phi} = \sqrt{1 + \frac{M}{r}} \tag{65}$$

and the nonvanishing extrinsic curvature terms are

$$K = \frac{M}{(M+r)^2} \ , \tag{66a}$$

$$\bar{A}_{rr} = -\frac{4M}{3(M+r)^2} \ , \tag{66b}$$

$$\bar{A}_{\theta\theta} = \frac{\bar{A}_{\varphi\varphi}}{\sin^2(\theta)} = \frac{2Mr^2}{3(M+r)^2} \ . \tag{66c}$$

Using an alternative to the standard 1+log condition given by Eq. (13), the lapse is evolved according to a condition consistent with staticity

$$\partial_0 \alpha = -\alpha(1-\alpha)K \ . \tag{67}$$

For the shift vector evolution equation, we desire only that the right-hand sides vanish analytically (although numerical error is expected to result in specious evolution). To this end, we adopt the nonadvecting Gamma-driver condition

$$\partial_t \beta^i = B^i \ , \tag{68a}$$

$$\partial_t B^i = \frac{3}{4}\partial_t \bar{\Lambda}^i - \eta B^i \ . \tag{68b}$$

The initial lapse and shift take on the forms

$$\alpha = \frac{r}{M+r} \ , \tag{69a}$$

$$\beta^r = \frac{Mr}{(M+r)^2} \ . \tag{69b}$$

We use damping parameter $\eta = 0.25/M$, although the results are not very sensitive to its particular value because the evolution begins and remains in a quasistatic state.

Analytically, the trumpet solution with these gauge conditions is static, but numerical errors result in unwanted evolution of the fields away from the initial data.

We perform numerical evolutions of these data on fixed numerical grids, subject to these gauge evolution equations, to confirm that evolution away from the initial data disappears nearly exponentially with increased finite difference order.

We choose a fixed, spherical-like coordinate grid with $N_{\mathtt{x1}} = 128$ and $N_{\mathtt{x2}} = N_{\mathtt{x3}} = 2$, where the radial points are distributed according to Eq. (63) with $w = 0.0747$ and $r_{\max}/M = 1000$ (the location of the outer boundary). We perform numerical evolutions on these fixed grids at finite difference orders $N_{\rm FD} \in \{2, 4, 6, 8\}$.

After $t/M \approx 100$, the freely evolved conformal factor W, and the gauge functions α and β^r, are compared with their initial values. The relative differences are shown in Fig. 5 for varying $N_{\rm FD}$. As in the puncture evolution of the previous section, we observe nonconvergent numerical errors inside the horizon ($r/M \lesssim 0.5$), which leads to additional spurious dynamics in the black hole with increased finite difference order. Unlike in the 1+log evolutions, however, no sharp lapse wave exists in the trumpet solution, so that exponential convergence is maintained over the entire numerical grid outside the horizon and inside the region influenced by the approximate outer boundary conditions. In fact, at high finite difference order, all plotted quantities drop to roundoff-level agreement somewhere in the region between $r/M = 10$ and 100.

3. *Gravitational waves from a head-on collision*

In Sec. V B 1, we demonstrated that nearly exponential convergence outside a puncture black hole horizon (and inside the region causally influenced by the outer boundary) is restored in the wake of a sharp gauge wave. In [18, 19], it is posited that this sharp wave causes nonconvergent errors in moving puncture black hole binary simulations on AMR grids, due to reflections off refinement boundaries. These nonconvergent errors have a direct impact on the convergence of the gravitational waves in these simulations.

In this section, we explore the convergence of gravitational wave signals from a head-on Brill-Lindquist black hole collision on spherical-like coordinate grids, keeping the numerical grids fixed at a moderate resolution and varying only finite difference derivative order, choosing $N_{\rm FD} \in \{2, 4, 6, 8, 10\}$.

Brill-Lindquist initial data [51] represent nonspinning black holes starting from rest. The initial data are constructed from a superposition of isotropic wormhole slices of the Schwarzschild spacetime. Though the formulation holds for an arbitrary number of black holes, here we use only two.

The wormhole slice of the Schwarzschild spacetime with isotropic radial coordinate r is conformally flat ($\varepsilon_{ij} = 0$) on the initial Cauchy surface. This hypersurface is maximal ($K = 0$) and exists at a moment of time symmetry ($\bar{A}_{ij} = 0$). The conformal factor is the solution to the flat-space Laplace equation, which allows for a direct superposition of single wormhole conformal factors given in Eq. (61)

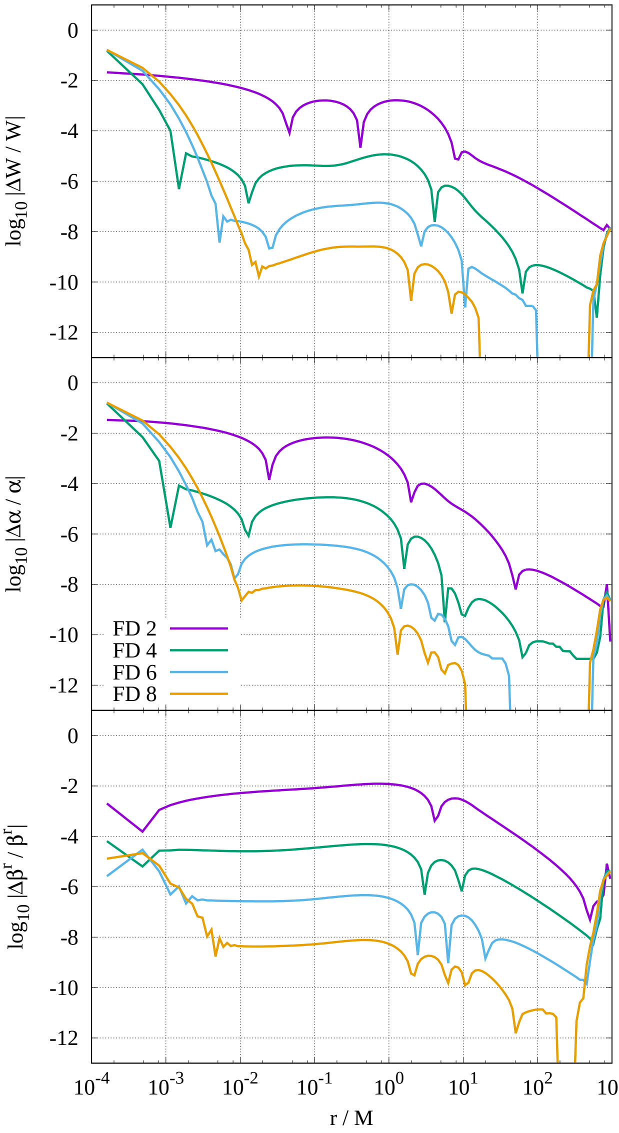

FIG. 5. Trumpet black hole initial data evolved for $t/M \approx 100$ in the static trumpet gauge: relative difference between evolved and initial conformal factor W (**top panel**), lapse α (**middle panel**), and radial component of the shift β^r (**bottom panel**), versus distance from the origin. The trumpet black hole is centered on the origin.

$$e^{\phi} = 1 + \frac{M_+}{2r_+} + \frac{M_-}{2r_-} \,, \tag{70}$$

where

$$r_\pm = \sqrt{r^2 + b^2 \mp 2br\cos(\theta)} \tag{71}$$

is the isotropic radial coordinate distance from the coordinate origin to the puncture with mass parameter $M_\pm$

located along the spherical-polar axis $\pm b/M$ above ($+$) or below ($-$) the origin. This configuration is axisymmetric with respect to the polar axis, though, as with all of the other cases, we perform our simulation in full 3+1 dimensions.

For the runs presented here, we focus on an equal-mass case in which $M_{\pm} = M/2$, so that the ADM mass integral (48a) gives $M_{\rm ADM} = M$. We confirm that the ADM momentum (48b) and angular momentum (48c) integrals vanish, as expected. We also chose $b/M = 0.5$.

We use Eq. (62) for the initial lapse, and $\beta^i = 0$ and $B^i = 0$ for the initial shift. The shift evolution damping parameter is set to $\eta = 2/M$, which results in an equilibrium remnant horizon radius of $r_{\rm H}/M \approx 1.4$. We use a spherical-like grid with redistribution function equation (63), so that `x1` corresponds to a radial coordinate. We choose $r_{\rm max}/M = 1000$ and $w = 0.125$ with $N_{\tt x1} = 400$, $N_{\tt x2} = 64$, and $N_{\tt x3} = 2$ (where $N_{\tt x3}$ is the axis of symmetry for the collision). Evolving wormhole initial data with the ordinary 1+log lapse condition Eq. (13) results in a sharp, nonconvergent gauge pulse that propagates outward from the puncture. We find, however, that excellent convergence is restored in the wake of the pulse as it moves towards the outer boundary.

The black holes are both nonspinning and are released from rest, so they remain on the polar axis during infall and collide head-on. They merge to form a single, strongly perturbed black hole, which quickly rings down to a stationary state as the time-changing quadrupole in the horizon is radiated away in the form of gravitational waves. The result is that the waveform after merger behaves as an exponentially damped harmonic oscillator. In spin-weight-two spheroidal harmonics, the fundamental gravitational wave mode in the $\ell = 2$ harmonic has complex frequency $\omega M \approx 0.3737 - 0.0890i$ [80]. With only a constant amplitude $A_{\rm f}$ and phase offset $\phi_{\rm f}$ acting as fitting parameters, the magnitude of the expected ringdown signal

$$\Re\left(\Psi_4^{\rm rd}\right) = A_{\rm f} \exp(-0.0890\, t) \cos(0.3737\, t + \phi_{\rm f}) \qquad (72)$$

is plotted in the upper panel of Fig. 6, atop the amplitude of the real part of the dominant $\ell = 2$ mode of Ψ_4 measured in our simulation. Excellent agreement is observed between the expected ringdown signal and the results from our simulation *over more than six decades* in amplitude. Further, the symmetry of the head-on collision is expected to result in gravitational waves that are in a pure + polarization state, which we confirm by measuring the imaginary part of Ψ_4 to be zero to roundoff error [see Eq. (50)].

The bottom panel of Fig. 6 demonstrates that differences in waveforms at adjacent finite difference orders (keeping the spherical-like coordinate grids fixed at moderate resolution) converge nearly exponentially with increased finite difference order. Notice that after the peak gravitational wave signal has passed, $|\text{FD } 6 - \text{FD } 8|$ and $|\text{FD } 8 - \text{FD } 10|$ are at times influenced by roundoff error, as evidenced by their suddenly stochastic behavior. We confirmed this feature by repeating the simulations with `long double` (80-bit) floating-point precision.

The peak amplitude occurs at retarded time of approximately $t_{\rm ret}/M \approx 18$, when the differences between waveforms at adjacent finite difference orders are near their peaks. At this time, this mode of Ψ_4 gains about an order of magnitude more precision with each increment of finite difference order. The particular rate of exponential convergence depends on the grid spacing.

The wave carries away energy from the black hole, so to what degree is convergence in the waveform reflected in the Hamiltonian (energy) constraint? Figure 7 plots the Hamiltonian constraint at the time in which the peak gravitational wave signal crosses the gravitational wave extraction radius ($t/M \approx 62.9$ and $r_{\rm ext}/M \approx 44.8$). As with the single puncture black hole, the Hamiltonian constraint restores its exponential convergence in the wake of the outgoing gauge pulse. Notice the Hamiltonian constraint violation in the eighth- and tenth-order finite difference cases is nearly indistinguishable in the range $r/M \gtrsim 7$ due to roundoff error. If this Hamiltonian constraint violation were to influence the gravitational waveform convergence, violations at this radius should impact the gravitational waves at retarded times $t_{\rm ret}/M \approx 0$ through 52. However, in that range the $|\text{FD } 8 - \text{FD } 10|$ and $|\text{FD } 6 - \text{FD } 8|$ curves are easily distinguishable in Fig. 6. We conclude that even if the Hamiltonian constraint is dominated by roundoff error, the gravitational waveforms may still be convergent.

Based on these results, we conclude that SENR/NRPy+ would be an excellent tool for studying perturbed black holes.

VI. CONCLUSION

In this work, we extend the reference-metric formulation of the BSSN equations pioneered by [24, 26, 28, 29] to handle Cartesian-, cylindrical-, and spherical-like numerical grids. At the heart of this strategy, a noncoordinate basis is adopted to remove from tensorial variables all coordinate singularities that arise from the choice of certain bases. Treating these singularities analytically, we successfully evolve black holes using the moving punctures approach [9, 10, 30] without resorting to special integration techniques, and without encountering numerical instabilities.

We announce a new numerical relativity code package called SENR/NRPy+, which implements this approach. It is fully open source, open development, and nonproprietary. NRPy+, written entirely in Python, converts tensorial expressions and their derivatives in Einstein notation to optimized C code, representing derivatives with suitable finite difference approximations. Our current implementation supports Cartesian-like, spherical-like and cylindrical-like coordinates, but our methods can be generalized easily for other orthogonal coordinate systems. SENR contains all of the basic numerical al-

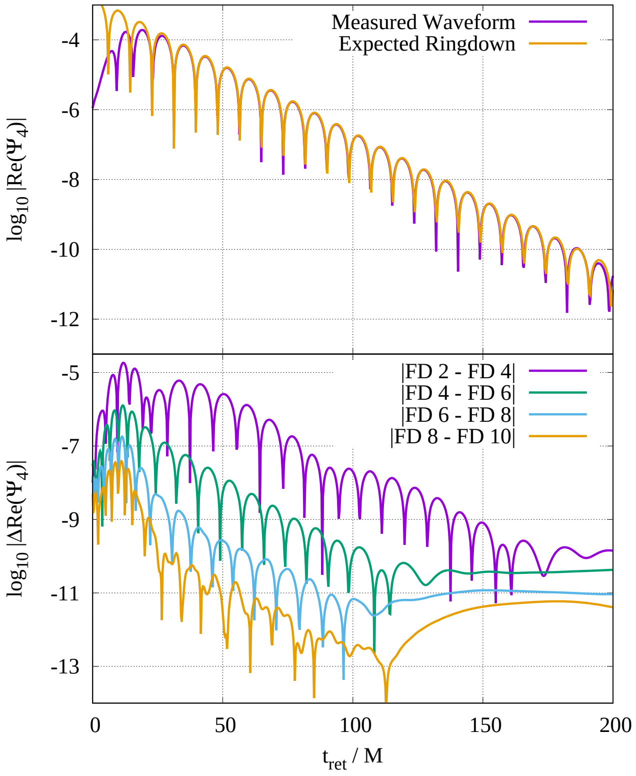

FIG. 6. Analysis of gravitational waves from the head-on collision of two puncture black holes resulting from Brill-Lindquist initial data evolved with the standard gauge conditions. The common axis shows the retarded time $t_{\rm ret} \equiv t - r_{\rm ext}$, where the gravitational wave extraction radius is $r_{\rm ext}/M \approx 44.8$. (**Top panel**) Dominant ($\ell = 2, m = 0$) mode of Ψ_4, $|\Re(\Psi_4)|_{\ell=2,m=0}$, plotted atop $\left|\Psi_4^{\rm rd}\right|$, the quasinormal mode ringdown amplitude and frequency expected from black hole perturbation theory [Eq. (72)]. Numerical results are plotted at finite difference order $N_{\rm FD} = 10$. (**Bottom panel**) Convergence of absolute differences in $|\Re(\Psi_4)|_{\ell=2,m=0}$ at adjacent finite differencing orders, keeping the spherical grid fixed at moderate resolution.

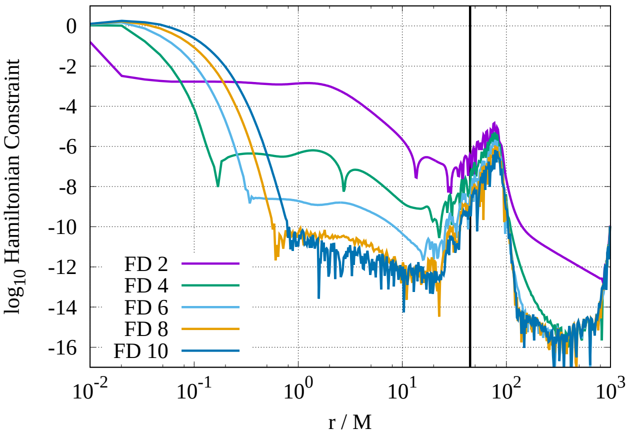

FIG. 7. Brill-Lindquist head-on collision of two puncture black holes: Hamiltonian constraint violation at $t/M \approx 62.9$, such that the peak gravitational wave amplitude passes through $r_{\rm ext}/M \approx 44.8$, the radius at which gravitational waves are extracted in Fig. 6. Spherical grids are fixed at moderate resolution; only finite differencing order is varied. The black vertical line denotes the approximate location of the peak gravitational wave strain.

gorithms needed for a numerical relativity code, making use of the C codes generated by NRPy+ where complex tensorial expressions are required. To the best of our knowledge, this is the first open-source numerical relativity code that lets the user select from a broad range of curvilinear coordinate systems.

The SENR/NRPy+ implementation of the BSSN equations is validated against two other well established numerical relativity codes. In the context of a Minkowski spacetime with strong random perturbations, we achieve roundoff-level agreement with the BMCCM code, which evolves the BSSN equations in spherical coordinates at fixed fourth-order finite difference accuracy. We similarly observe excellent agreement between the results of SENR/NRPy+ and the ETK's McLachlan BSSN thorn [15] in the context of a single puncture black hole evolution. We also show that, for both single and double black hole spacetimes, the finite difference truncation error converges with increasing grid resolution at the expected rate. In addition, we demonstrate exponential convergence of the error with increasing finite difference order, keeping the grid resolution constant.

A number of physical diagnostic quantities are implemented in SENR/NRPy+, including constraint violations and ADM integrals. For additional tools, we developed an ETK compatibility layer within SENR that interpolates quantities in the chosen curvilinear coordinate basis to the Cartesian basis, and onto a Cartesian grid. In this way, SENR can take direct advantage of the large suite of ETK-based diagnostic utilities, including, e.g., apparent horizon finders and gravitational wave diagnostics. These diagnostics are applied to head-on collisions of two nonspinning black holes to show that the absolute difference between gravitational waveforms converges exponentially at successive finite difference order.

We conclude that our extended formalism for BSSN on arbitrary coordinate grids as implemented SENR/NRPy+ provides an outstanding tool for analyzing perturbed black holes without approximation. Further, the spherical-like coordinate systems adopted are ideal for gravitational wave extraction and analysis. We next plan to add coordinate system dynamics and explore bispherical-like coordinate geometries, so that black hole binaries may be modeled with extreme efficiency. All simulations displayed in this work can be performed on aging desktop computers (except for a few of the high resolution Cartesian runs, which were executed on a desktop with additional RAM), and given the ability of these coordinate systems to exploit near-symmetries near compact objects, we anticipate that SENR/NRPy+ may be the first code to unlock the desktop as a powerful tool for fully general relativistic gravitational wave astrophysics.

ACKNOWLEDGMENTS

We wish to thank Vassilios Mewes, Manuela Campanelli, and Yosef Zlochower for many useful discussions. We thank Isabel Cordero-Carrión and Pedro Cerdá-Durán for their correspondence regarding the stability of Runge-Kutta integration. We also thank Saul Teukolsky for conversations concerning the expected rates of convergence of our finite difference approximations. Funding for computer equipment was provided in part by NSF EPSCoR Grant No. OIA-1458952. Some computations were performed on single nodes of West Virginia University's Spruce Knob high-performance computing cluster, funded in part by NSF EPSCoR Research Infrastructure Improvement Cooperative Agreement No. 1003907, the state of West Virginia (WVEPSCoR via the Higher Education Policy Commission), and West Virginia University. Some computational resources were also provided by the Extreme Science and Engineering Discovery Environment (XSEDE) Jetstream cloud computing infrastructure at Indiana University, via the XSEDE Campus Champion for West Virginia University account TG-TRA140024. This work was also supported by NSF Grants No. 1402780 and No. 1707526 to Bowdoin College.

[1] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, and et al., Physical Review Letters **116**, 061102 (2016), arXiv:1602.03837 [gr-qc].

[2] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, and et al., Physical Review Letters **116**, 241103 (2016), arXiv:1606.04855 [gr-qc].

[3] The LIGO Scientific Collaboration, the Virgo Collaboration, B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, and et al., Phys. Rev. Lett. **118**, 221101 (2017), arXiv:1706.01812 [gr-qc].

[4] B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya, and et al., Physical Review Letters **119**, 141101 (2017), arXiv:1709.09660 [gr-qc].

[5] The LIGO Scientific Collaboration, the Virgo Collaboration, B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, and et al., ArXiv e-prints (2017), arXiv:1711.05578 [astro-ph.HE].

[6] B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya, and et al., Physical Review Letters **119**, 161101 (2017), arXiv:1710.05832 [gr-qc].

[7] B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya, and et al., Astrophysical Journal Letters **848**, L12 (2017), arXiv:1710.05833 [astro-ph.HE].

[8] F. Pretorius, Physical Review Letters **95**, 121101 (2005), gr-qc/0507014.

[9] M. Campanelli, C. O. Lousto, P. Marronetti, and Y. Zlochower, Physical Review Letters **96**, 111101 (2006), gr-qc/0511048.

[10] J. G. Baker, J. Centrella, D.-I. Choi, M. Koppitz, and J. van Meter, Physical Review Letters **96**, 111102 (2006), gr-qc/0511103.

[11] T. Goodale, G. Allen, G. Lanfermann, J. Massó, T. Radke, E. Seidel, and J. Shalf, in *Proceedings of the 5th International Conference on High Performanc* VECPAR'02 (Springer-Verlag, Berlin, Heidelberg, 2003) pp. 197–227.

[12] E. Schnetter, S. H. Hawley, and I. Hawke, Classical and Quantum Gravity **21**, 1465 (2004), gr-qc/0310042.

[13] "Cactus Computational Toolkit," `http://www.cactuscode.org`.

[14] "Carpet: Adaptive mesh refinement for the Cactus framework," `http://www.carpetcode.org`.

[15] F. Löffler, J. Faber, E. Bentivegna, T. Bode, P. Diener, R. Haas, I. Hinder, B. C. Mundim, C. D. Ott, E. Schnetter, G. Allen, M. Campanelli, and P. Laguna, Classical and Quantum Gravity **29**, 115001 (2012), arXiv:1111.3344 [gr-qc].

[16] "Einstein Toolkit Consortium Homepage," `http://www.einsteintoolkit.org`.

[17] "SENR/NRPy+ website," `http://math.wvu.edu/~zetienne/SENR/`.

[18] Y. Zlochower, M. Ponce, and C. O. Lousto, Phys. Rev. D **86**, 104056 (2012), arXiv:1208.5494 [gr-qc].

[19] Z. B. Etienne, J. G. Baker, V. Paschalidis, B. J. Kelly, and S. L. Shapiro, Phys. Rev. D **90**, 064032 (2014), arXiv:1404.6523 [astro-ph.HE].

[20] "SpEC: Spectral Einstein Code website," `https://black-holes.org`.

[21] B. Szilágyi, International Journal of Modern Physics D **23**, 1430014 (arXiv:1405.3693 [gr-qc].

[22] L. E. Kidder, S. E. Field, F. Foucart, E. Schnetter, S. A. Teukolsky, A. Bohn, N. Deppe, P. Diener, F. Hébert, J. Lippuner, J. Miller, C. D. Ott, M. A. Scheel, and T. Vincent, Journal of Computational Physics **335**, 84 (2017), arXiv:1609.00098 [astro-ph.HE].

[23] F. Pretorius, Classical and Quantum Gravity **22**, 425 (2005), gr-qc/0407110.

[24] S. Bonazzola, E. Gourgoulhon, P. Grandclément, and J. Novak, Phys. Rev. D **70**, 104007 (2004), gr-qc/0307082.

[25] M. Shibata, K. Uryū, and J. L. Friedman, Phys. Rev. D **70**, 044044 (2004), gr-qc/0407036.

[26] J. D. Brown, Phys. Rev. D **79**, 104029 (2009), arXiv:0902.3652 [gr-qc].

[27] E. Gourgoulhon, *3+1 Formalism and Bases of Numerical Relativity* (Springer-Verlag Berlin Heidelberg, 2012).

[28] P. J. Montero and I. Cordero-Carrión, Phys. Rev. D **85**, 124037 (2012), arXiv:1204.5377 [gr-qc].

[29] T. W. Baumgarte, P. J. Montero, I. Cordero-Carrión, and E. Müller, Phys. Rev. D **87**, 044026 (2013).

[30] M. Alcubierre, B. Brügmann, P. Diener, M. Koppitz, D. Pollney, E. Seidel, and R. Takahashi, Phys. Rev. D **67**, 084023 (2003), gr-qc/0206072.

[31] A. Meurer, C. P. Smith, M. Paprocki, O. Čertík, S. B. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. K. Moore, S. Singh, T. Rathnayake, S. Vig, B. E. Granger, R. P. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. J. Curry, A. R. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz, PeerJ Computer Science **3**, e103 (2017).

[32] L. Dagum and R. Menon, IEEE Comput. Sci. Eng. **5**, 46 (1998).

[33] B. Szilágyi, R. Gómez, N. T. Bishop, and J. Winicour, Phys. Rev. D **62**, 104006 (2000), gr-qc/9912030.

[34] B. Szilágyi, B. Schmidt, and J. Winicour, Phys. Rev. D **65**, 064015 (2002), gr-qc/0106026.

[35] M. Alcubierre, G. Allen, C. Bona, D. Fiske, T. Goodale, F. S. Guzmán, I. Hawke, S. H. Hawley, S. Husa, M. Koppitz, C. Lechner, D. Pollney, D. Rideout, M. Salgado, E. Schnetter, E. Seidel, H.-a. Shinkai, D. Shoemaker, B. Szilágyi, R. Takahashi, and J. Winicour, Classical and Quantum Gravity **21**, 589 (2004), gr-qc/0305023.

[36] "Apples With Apples: Robust Stability Test," `http://www.appleswithapples.org/TestMethods/Tests/robust/robust.html`.

[37] J. D. Brown, Phys. Rev. D **77**, 044018 (2008), arXiv:0705.1359 [gr-qc].

[38] J. D. Brown, Phys. Rev. D **80**, 084042 (2009), arXiv:0908.3814 [gr-qc].

[39] T. Nakamura, K. Oohara, and Y. Kojima, Progress of Theoretical Physics Supplement **90**, 1 (1987).

[40] M. Shibata and T. Nakamura, Phys. Rev. D **52**, 5428 (1995).

[41] T. W. Baumgarte and S. L. Shapiro, Phys. Rev. D **59**, 024007 (1999), gr-qc/9810065.

[42] T. W. Baumgarte and S. L. Shapiro, *Numerical Relativity: Solving Einstein's Equations on the Computer* (Cambridge University Press, New York, NY, USA, 2010).

[43] P. Marronetti, W. Tichy, B. Brügmann, J. González, and U. Sperhake, Phys. Rev. D **77**, 064010 (2008), arXiv:0709.2160 [gr-qc].

[44] C. Bona, J. Massó, E. Seidel, and J. Stela, Physical Review Letters **75**, 600 (1995), gr-qc/9412071.

[45] E. Schnetter, Classical and Quantum Gravity **27**, 167001 (2010), arXiv:1003.0859 [gr-qc].

[46] J. R. van Meter, J. G. Baker, M. Koppitz, and D.-I. Choi, Phys. Rev. D **73**, 124011 (2006), gr-qc/0605030.

[47] S. Carroll, *Spacetime and Geometry: An Introduction to General Relativity* (Addison Wesley, 2004).

[48] S. Husa, I. Hinder, and C. Lechner, Computer Physics Communications **174**, 983 (2006), gr-qc/0404023.

[49] "The 2-Clause BSD License," `https://opensource.org/licenses/BSD-2-Clause`.

[50] "The GNU General Public License v3.0," `https://www.gnu.org/licenses/gpl-3.0.en.html`.

[51] D. R. Brill and R. W. Lindquist, Phys. Rev. **131**, 471 (1963).

[52] Y. T. Liu, Z. B. Etienne, and S. L. Shapiro, Phys. Rev. D **80**, 121503 (2009).

[53] K. A. Dennison and T. W. Baumgarte, Classical and Quantum Gravity **31**, 117001 (2014), arXiv:1403.5484 [gr-qc].

[54] J. Thornburg, Classical and Quantum Gravity **21**, 3665 (2004), gr-qc/0404059.

[55] M. Chirvasa and S. Husa, ArXiv e-prints (2008), arXiv:0812.3752 [gr-qc].

[56] N. Macon and A. Spitzbart, The American Mathematical Monthly **65**, 95 (1958).

[57] H. Kreiss and J. Oliger, *Methods for the Approximate Solution of Time Dependent Problems*, Global Atmospheric Research Programme (GARP): GARP Publication Series, Vol. 10 (GARP Publication, 1973).

[58] M. Alcubierre, *Introduction to 3+1 Numerical Relativity*, International series of monographs on physics (Oxford Univ. Press, Oxford, 2008).

[59] J. Thornburg, Phys. Rev. D **59**, 104007 (1999), gr-qc/9801087.

[60] O. Dreyer, B. Krishnan, D. Shoemaker, and E. Schnetter, Phys. Rev. D **67**, 024018 (2003), gr-qc/0206008.

[61] L. B. Szabados, Living Reviews in Relativity **7**, 4 (2004).

[62] E. Schnetter, B. Krishnan, and F. Beyer, Phys. Rev. D **74**, 024028 (2006), gr-qc/0604015.

[63] J. Thornburg, Classical and Quantum Gravity **21**, 743 (2004), gr-qc/0306056.

[64] J. Baker, M. Campanelli, and C. O. Lousto, Phys. Rev. D **65**, 044001 (2002), gr-qc/0104063.

[65] D. Pollney, C. Reisswig, E. Schnetter, N. Dorband, and P. Diener, Phys. Rev. **D83**, 044045 (2011), arXiv:0910.3803 [gr-qc].

[66] I. Ruchlin, J. Healy, C. O. Lousto, and Y. Zlochower, Phys. Rev. D **95**, 024033 (2017).

[67] I. Cordero-Carrión, P. Cerdá-Durán, and V. Mewes, private communication.

[68] I. Cordero-Carrión and P. Cerdá-Durán, ArXiv e-prints (2012), arXiv:1211.5930 [math-ph].

[69] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, New York, NY, USA, 2007).

[70] T. W. Baumgarte, P. J. Montero, and E. Müller, Phys. Rev. D **91**, 064035 (2015), arXiv:1501.05259 [gr-qc].

[71] T. W. Baumgarte and P. J. Montero, Phys. Rev. D **92**, 124065 (2015), arXiv:1509.08730 [gr-qc].

[72] T. W. Baumgarte and C. Gundlach, Physical Review Letters **116**, 221103 (2016), arXiv:1603.04373 [gr-qc].

[73] C. Gundlach and T. W. Baumgarte, Phys. Rev. D **94**, 084012 (2016), arXiv:1608.00491 [gr-qc].

[74] A. J. Miller and T. W. Baumgarte, Classical and Quantum Gravity **34**, 035007 (2017), arXiv:1607.03047 [gr-qc].

[75] C. S. Roberts, The Bell System Technical Journal **61**, 2053 (1982).

[76] I. Cordero-Carrión and P. Cerdá-Durán, SEMA SIMAI Springer Series, **4** (2014).

[77] J. Boyd, *Chebyshev and Fourier Spectral Methods: Second Revised Edition*, Dover Books on Mathematics (Dover Publications, 2001).

[78] M. Alcubierre, Classical and Quantum Gravity **20**, 607 (2003), gr-qc/0210050.

[79] M. Alcubierre, Classical and Quantum Gravity **22**, 4071 (2005), gr-qc/0503030.

[80] E. Berti, V. Cardoso, and A. O. Starinets, Classical and Quantum Gravity **26**, 163001 (2009), arXiv:0905.2975 [gr-qc].