

Massively parallel simulations of binary black holes with adaptive wavelet multiresolution

Milinda Fernando^{1,*} David Neilsen^{2,†} Yosef Zlochower^{3,‡} Eric W. Hirschmann^{2,§} and Hari Sundar^{4,||}

¹*Oden Institute for Computational Engineering and Sciences, University of Texas at Austin, Austin, Texas 78712, USA*

²*Department of Physics and Astronomy, Brigham Young University, Provo, Utah 84602, USA*

³*Center for Computational Relativity and Gravitation, and School of Mathematical Sciences, Rochester Institute of Technology, 85 Lomb Memorial Drive, Rochester, New York 14623, USA*

⁴*School of Computing, University of Utah, Salt Lake City, Utah 84112, USA*



(Received 30 November 2022; accepted 9 February 2023; published 17 March 2023)

We present results from the new Dendro-GR code. These include simulations of binary black hole mergers for mass ratios up to $q = 16$. Dendro-GR uses wavelet adaptive multiresolution to generate an unstructured grid adapted to the spacetime geometry together with an octree-based data structure. We demonstrate good scaling, improved convergence properties, and efficient use of computational resources. We validate the code with comparisons to LazEv.

DOI: [10.1103/PhysRevD.107.064035](https://doi.org/10.1103/PhysRevD.107.064035)

I. INTRODUCTION

The gravitational wave detectors LIGO/Virgo have made a number of epochal discoveries [1,2]. These have given us a dramatically broader conception and understanding of the high-energy universe and some of its compact object constituents [3–5]. As these detectors continually improve [6–8] and are added to by new detectors, such as KAGRA [9], we can confidently expect an ongoing parade of additional discoveries.

The detection and analysis of gravitational waves (GW) uses a library of modeled waveforms for comparison with the detector output signal. Numerical relativity waveforms are computed using the full nonlinear, Einstein equations, and these waveforms span the evolution of the binary system from inspiral, through merger, and finally to ring-down. These waveforms may be used directly in the analysis of gravitational waves [10,11], or to inform and validate faster, approximate methods for generating waveforms, such as semianalytical and phenomenological methods (see, e.g., [12–19]). Numerical relativity can also probe certain astrophysical scenarios that are difficult to model with approximate methods. Examples of such scenarios include nonvacuum spacetimes, such as systems with neutron stars, accretion disks, and/or magnetic fields. Even some vacuum binary black hole systems can be

difficult to model with approximate techniques, such as binaries with large eccentricity, high spins, or large mass ratios. We use the mass ratio $q = m_1/m_2$, where m_1 is the mass of the primary with $m_1 \geq m_2$.

While there are advantages to using numerical relativity waveforms directly in gravitational wave analysis, there are significant challenges in calculating waveforms of sufficient quality. The waveforms must be sufficiently long, have errors bounded within known tolerances, and they must span a large region of the binary parameter space. The development of newer, more sensitive gravitational wave detectors significantly complicates the challenge. For example, recent work on requirements for third-generation (3G) detectors [20–22] and LISA [23] estimate that errors in numerical relativity waveforms need to be reduced by an order of magnitude [24]. Another study found that numerical resolutions of binary black hole (BBH) spacetimes will need to be increased by almost a factor of 10 in some cases [25]. Reducing the error in numerical waveforms to the level required by 3G detectors will require new algorithms and methods in numerical relativity.

The challenge of producing waveforms for future gravitational wave detectors will require highly scalable numerical relativity codes that are able to efficiently run on exascale supercomputers. Dendro-GR is a new code for relativistic astrophysics that is designed to meet some of the next-generation challenges in numerical relativity. Dendro-GR scales well on massively parallel supercomputers, and it uses fast, responsive wavelet adaptive multiresolution (WAMR). Importantly, Dendro-GR easily accommodates many well-tested numerical methods that have been developed in the relativity community, such as the

*milinda@oden.utexas.edu

†david.neilsen@byu.edu

‡yrzsma@rit.edu

§ehirsch@byu.edu

||hari@cs.utah.edu

evolution of Einstein equations in the BSSN formalism and high-resolution shock-capturing methods for relativistic fluid dynamics.

Several projects are currently being developed in the community that use modern adaptive-mesh infrastructures and sophisticated numerical algorithms to meet this computational challenge. Among these are GR-Athena++ [26], which uses the highly efficient octree AMR infrastructure of Athena++ for full numerical relativity simulations coupled to GRMHD, GRChombo [27], a fully modern AMR numerical relativity code allowing for complex grid configurations, and CarpetX, which is a new AMR driver for the Einstein Toolkit [28,29] that is built on the AMReX toolkit [30]. Pseudospectral and discontinuous Galerkin methods promise some advantages for massively parallel computing. SpECTRE [31] uses discontinuous Galerkin methods and a task-based parallelization scheme. Nmesh [32,33] and BAMPs [34] are other codes using discontinuous Galerkin methods. Simflowny [35] has a domain-specific language and a web-based development environment and graphical user interface. Simflowny can generate code for multiple platforms, such as SAMRAI [36]. Dendro-GR uses an efficient octree structure to store the grid elements similar in spirit to that used in GR-Athena++, while Dendro-GR's wavelet decomposition with an unstructured grid is similar in spirit to SpECTRE.

This paper presents results from some of the first binary black hole mergers performed with Dendro-GR. We study gravitational waves from binary black hole systems with mass ratios up to $q = 16$. We compare results with the well-known *LazEv* [37,38] code in some cases, and find that the solutions match in the convergence limit. We also present performance data for Dendro-GR.

II. METHODS

Dendro-GR has been built with the intention of tackling relativistic astrophysics problems involving merging compact objects. Its development uses and accommodates a number of standard techniques within numerical relativity as well as including some new approaches, all with an eye to improving the efficiency, scalability, and time to solution for still challenging problems such as large mass ratio binary black holes. Among the conventional and well-tested numerical methods used in Dendro-GR, we solve the Einstein equations using the Baumgarte-Shapiro-Shibata-Nakamura (BSSN) formulation together with typical coordinate conditions, initial data, and finite-differencing algorithms. Newer approaches used within Dendro-GR include some of the following and are discussed at greater length subsequently in this section. The code uses a dynamic grid which is constructed via an expansion of the grid functions in an interpolating wavelet basis. In this basis, terms in the wavelet expansion can be mapped to individual grid points. The resulting unstructured grid is naturally represented computationally as an octree. On

integrating the equations of motion in time, each node of this octree is separately unzipped (decompressed) into a local point representation on a uniform Cartesian grid. The integrated functions are then zipped (compressed) back to a sparse representation by thresholding the coefficients of the wavelet expansion. This sparse representation is compact and computationally efficient as it conserves computer memory and reduces parallel communication. This section describes some of these key components of Dendro-GR in more detail. We begin with a brief description of our formalism for solving the Einstein equations and setting initial data. We then describe the generation of the grid using WAMR and the process for integrating the equations.

A. Formalism

There is extensive literature on solving the BSSN equations in general relativity, including monographs such as [39–42]. This section briefly outlines our particular choices for solving the BSSN equations. We write the BSSN equations in terms of the conformal factor [38]

$$\chi^{-1} = \det(\gamma_{ij}). \quad (1)$$

For gauge conditions, we use the “1 + log” slicing condition and the Γ -driver shift as used in [43]

$$\partial_t \alpha = \beta^i \partial_i \alpha - 2\alpha K, \quad (2)$$

$$\partial_t \beta^i = \beta^j \partial_j \beta^i + \frac{3}{4} B^i, \quad (3)$$

$$\partial_t B^i = \beta^j \partial_j B^i + \partial_i \tilde{\Gamma}^i - \beta^j \partial_j \tilde{\Gamma}^i - \eta B^i. \quad (4)$$

Spatial derivatives are calculated using centered finite-difference operators that are $O(h^6)$ in the grid spacing, h .

The semidiscrete Einstein equations are integrated in time using explicit fourth-order Runge-Kutta with a time-step $\Delta t = 0.25 \Delta x_{\min}$, where $x_{\min}$ is the smallest distance between points on the grid. Kreiss-Oliger dissipation is added to the equations using a fifth-order operator

$$\Delta_x^5 u_m^n = (-u_{m+3}^n + 6u_{m+2}^n - 15u_{m+1}^n + 20u_m^n - 15u_{m-1}^n + 6u_{m-2}^n - u_{m-3}^n) / (64\Delta x) \quad (5)$$

with a tunable amplitude parameter σ , $0 \leq \sigma < 1$, which allows one to adjust the amount of dissipation [39]. As discussed below in Sec. IV A, we found best results with $\sigma = 0.4$. We make the common choice to enforce certain algebraic constraints and derivative definitions as described, for example, in [44]. Outgoing radiative boundary conditions are applied to the dynamical variables.

We extract gravitational waves from our simulations at five radii between $50 M \leq r \leq 100 M$ using the Penrose scalar, ψ_4 [39,45]. Here, M is the sum of the local masses of each black hole $M = m_1 + m_2$, and m_1 and m_2 are the ADM masses computed in the asymptotically flat region at

each puncture [46]. Their decomposition with respect to spin-weighted spherical harmonics (SWSH) is performed using the Lebedev quadrature [47]. To evaluate ψ_4 at each of the quadrature points on each 2-sphere, we perform an efficient search operation on the underlying grid, and SWSH projection coefficients are computed with a parallel reduction operation [48].

B. Initial data

Initial data for both Dendro-GR and LazEv are set using the TwoPunctures code [46] from the Einstein Toolkit [28,29]. For the initial values of shift, both codes set $\beta^i(t=0) = 0$. Initial values of the lapse in Dendro-GR use the *ad hoc* function $\alpha(t=0) = \tilde{\psi}^{-2}$, where $\tilde{\psi} = 1 + m_{p1}/(2r_1) + m_{p2}/(2r_2)$, r_i is the coordinate distance to the i th BH, and m_{pi} is the bare mass parameter of the i th BH. In LazEv the initial lapse is $\tilde{\psi} = 1 + 1/(4r_1) + 1/(4r_2)$.

In this paper, we evolve nonspinning black hole binaries with mass ratios $q = 1, 2, 4, 8$, and 16. We place the black holes initially on the x axis, with the binary's center of mass at the origin. The smaller black hole with mass m_2 is placed on the positive x axis, and the initial coordinate separation is fixed to $x_2 - x_1 = 8M$. Again, M is the sum of the local ADM masses of each black hole computed in the asymptotically flat region at each puncture. Initial data parameters for the $q = 1$ binary are *ad hoc* quasicircular parameters chosen to match previous work [49]. Parameters for all other cases were found using the low eccentricity post-Newtonian expressions reported in [50]. To simplify comparisons with LazEv, we set the TwoPunctures code to use the bare puncture masses and other parameters shown in Table I directly. Finally, we ran the $q = 1, 2$, and 4 cases with both Dendro-GR and LazEv, while the higher mass ratio simulations were only run with Dendro-GR.

C. Symbolic code generation

The evaluation of the BSSN equations at a given grid point is computationally expensive and can be challenging due to the large number of terms associated with the equations. Manually writing code to evaluate these equations can be

prone to error, difficult to debug, and challenging to perform architecture-specific optimizations. To address some of these issues, we have developed a SymPy-based code generation framework for Dendro-GR. This tool has some of the same capabilities as NrPy+ [51,52] but is more limited in scope. Using our symbolic framework, we compute the directed acyclic graph representing the underlying computations for the BSSN equations. We perform optimizations to reduce the overall number of operations as well as architecture-specific optimizations that improve our code's performance portability. The current implementation of the symbolic framework supports CPUs and GPUs [48,53].

D. Grid generation with WAMR

The computational complexity of the Einstein equations, together with the requirement of high accuracy across multiple spatial and temporal scales, motivates the use of grid adaptivity. Dendro-GR uses a wavelet-based approach which results in a representation of the underlying field variables on a sparse, adaptive mesh. We describe briefly here the fundamental aspects of this sparse representation. More complete details can be found in [48,53–55]. While we use the coefficients in a wavelet expansion to generate the computational grid, we store the grid functions only in the point representation. Thus, the wavelet coefficients are not used to integrate the equations of motion.

Two essential ingredients in our approach are the notion of iterative interpolation [56] and the wavelet representation itself [57,58]. We demonstrate both of these in one dimension. The extension to multiple dimensions is straightforward and is accomplished by simply repeating the procedures we will describe in each additional dimension. To fix ideas, we first define a set of nested grids V_j , where

$$V_j = \{x^{j,k} : x^{j,k} = 2^{-j}k\Delta x\},$$

where j and k are nonnegative integers, and Δx is the spacing on the base grid (or level) and which is labeled with $j = 0$. This base grid V_0 is comprised of $N + 1$ grid points evenly spaced on a domain of length $L = N\Delta x$. Each finer

TABLE I. The initial configuration parameters for nonspinning binary black hole systems for increasing mass ratio. The presented numerical waveforms are based on these initial data parameters. The parameters for $q \geq 2$ were obtained using the expressions in [50]. The initial data are set using the bare mass parameters. The black holes are placed initially on the x axis at the locations x_1 and x_2 as given in the table. The linear momentum of the second black hole is given in the last two columns, $\mathbf{p}_2 = (p_x, p_y)$, and $\mathbf{p}_1 = -\mathbf{p}_2$.

Mass ratio	Puncture parameter		ADM mass		Total	x position		Momentum BH2	
$q = m_1/m_2$	m_{p2}	m_{p1}	m_2	m_1	ADM mass	x_2	x_1	p_x	p_y
1	0.48240	0.48240	0.50010	0.50010	0.98844	4	-4	0	0.1140
2	0.31715	0.65150	0.6667	0.33333	0.98931	5.3238	-2.6762	-0.0017777	0.10049
4	0.18805	0.78937	0.20000	0.80000	0.99237	6.3873	-1.6127	-0.0010647	0.072660
8	0.10362	0.88245	0.11111	0.88889	0.99534	7.1006	-0.89938	-0.00049937	0.045037
16	0.054585	0.93761	0.058824	0.94118	0.99740	7.5226	0.47741	-0.00019532	0.025319

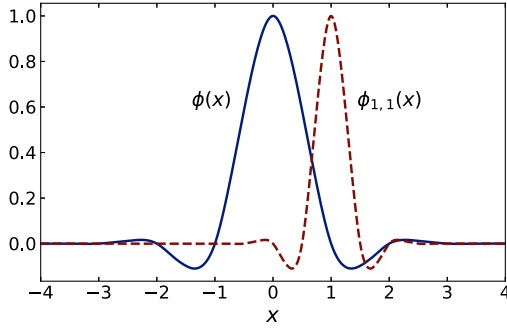


FIG. 1. This shows the fundamental solution of the iterated interpolation $\phi(x)$ (solid) and a basis element attached to the grid V_1 called $\phi_{1,1}(x)$ (dashed). All the basis functions are scaled, translated versions of the fundamental solution.

grid with $j > 0$ contains each point in every coarser grid. Values of a field u at level j are designated $u(x^{j,k}) \equiv u^{j,k}$. If known, these values are copied from coarser grids to all required fine grids. For example, to go from V_j to V_{j+1} , we take $u^{j+1,2k} = u^{j,k}$. A similar copy happens to all higher-level fine grids. Of course, on these finer grids, there will be points newly appearing. The field values on those grid points new to grid V_{j+1} are interpolated from the known values on the coarser grid V_j . We generally use Lagrange interpolation. In this manner, all fields at any refinement level can be had (see Fig. 2). This iterated interpolation continued to arbitrarily large levels produces continuous functions with compact support [57].

The process just described is the start of how our sparse grid will emerge. But it also produces a natural basis set with which we can represent our fields. This basis set is comprised of interpolating functions created via iteration from a sequence of zeros and a single value of one living on V_0 (sometimes referred to as a Kronecker sequence). More specifically, define a function $\phi_{0,k}(x)$ which takes values at the points $x^{0,l}$ (imagine on the base grid) of $\phi_{0,k}(x^{0,l}) = \delta_k^l$. Now interpolate as described above to find $\phi_{0,k}$ at other grid points and iterate. This can be repeated for $\phi_{j,k}$ with $j > 0$. The resulting iterated interpolating functions will have a number of properties, including compact support and a two scale relation given by

$$\phi_{j,k}(x) = \sum_l c_{j,k}^l \phi_{j+1,l}(x),$$

where the coefficients $c_{j,k}^l$ will depend on the order of the interpolation. Significantly, each of these iterated, interpolating functions are scaled, translated versions of a single, fundamental function $\phi(x)$ related to a particular limiting function of the above iterated interpolation. It is related to the Daubechies scaling function and shown in Fig. 1.

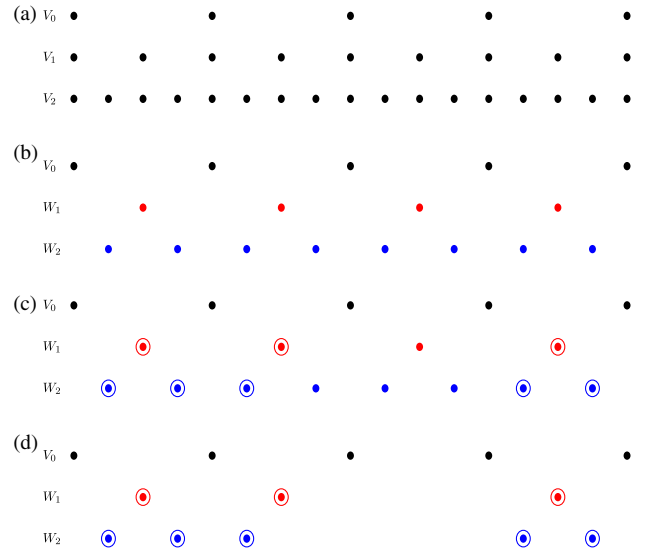


FIG. 2. This illustrates a simplified conception of how the grid is constructed. In (a), one-dimensional nested grids V_j (for $j = 0, 1, 2$) are shown. Note that every fine grid contains all the grid points on every coarser grid. This redundancy is removed in (b) on defining the complementary spaces, W_j (colored red and blue). We compute wavelet coefficients ($d^{j,k}$) as the difference between $u^{j,k}$ and the field as interpolated from level V_{j-1} . In (c), those grid points with wavelet coefficients larger than a pre-determined threshold ϵ are tagged (here with circles) as essential to the calculation. For those grid points with $|d^{j,k}| < \epsilon$, the corresponding terms in the interpolating wavelet expansion are ignored, and the grid points are discarded from the mesh, as illustrated in (d).

With these iterated, interpolating functions in hand, we can now return to and complete our wavelet representation. Note that at each point of each level, we have an associated scaling function

$$\phi_{j,k}(x) = \phi(2^j x / \Delta x - k),$$

which, when taken all together, constructs a basis at each level, j . However, across levels, the set of scaling functions is overdetermined and will not form a basis for the entire grid until we deal with the redundancy introduced by having common points in V_j and V_{j+1} . To this end, we consider the complementary space to V_j , which we call W_j , such that

$$W_j = \{x^{j,k} : x^{j,k} = 2^{-j}k\Delta x, k \text{ odd}\},$$

and is that set of points in V_j that are not in V_{j-1} [see Fig. 2(b)]. With this definition, we use the set of grids given by $\{V_0, W_j\}$ and thereby have a basis with respect to which we can define our fields u :

$$u(x) = \sum_{k \in S_0} u^{0,k} \phi_{0,k}(x) + \sum_{j=1}^{\infty} \sum_{k \in S_j} d^{j,k} \phi_{j,k}(x).$$

The coefficients $u^{0,k}$ and $d^{j,k}$ are expansion coefficients with $S_0 = 0, 1, \dots, N$ providing the index set for the base grid V_0 , and $S_j = 1, 3, \dots, 2^{j+1}N - 1$ being the index set for the fine grid given by W_j . This last expression is our interpolating wavelet expansion in which $u^{0,k}$ are just the values of the field on the base points and the coefficients $d^{j,k}$ referred to as wavelet coefficients are but the differences between the field values $u^{j,k}$ and the interpolated values at $x^{j,k}$ coming from the next lower level, $j - 1$. If we designate these interpolated values as $\tilde{u}^{j,k}$, the wavelet coefficients are then computed simply as

$$d^{j,k} = u^{j,k} - \tilde{u}^{j,k}.$$

We can think of that part of the expansion with the scaling functions as encoding the smooth part of the field $u(x)$, while the wavelet coefficients provide information about the function on fine scales. Because of the highly local nature of the wavelets used, this representation will have many wavelets in regions exhibiting strong spatial variations, while few will be necessary in regions where the field is changing slowly.

With the wavelet representation in hand, compression is now possible. More particularly, we can make the

representation sparse by choosing a threshold value ϵ , such that if the magnitude of the wavelet coefficients $|d^{j,k}|$ is smaller than ϵ , we truncate the expansion and discard the corresponding grid points from the grid itself. Doing so both reduces the grid size and provides an error bound on the representation of the field. In Figs. 2(c) and 2(d), we illustrate this approach to constructing the grid. As already mentioned, extending to multiple dimensions amounts to taking the basis functions to be products of the one-dimensional basis functions.

An example of the WAMR-constructed grid used to evolve binary black holes is shown in Fig. 3, which shows the grid for a $q = 16$ binary before merger. This computational grid is very efficient: The grid is sparse with refined regions that adapt to the small-scale features of the spacetime. The grid does not require refined regions to be rectangular on large scales, significantly saving on computational and memory costs. Moreover, large overlapping regions between refinement levels are not required.

E. Refinement functions

Interpolating wavelets are sensitive to any nondifferentiable or nonconvergent parts of a solution, triggering immediate refinement. This is important for resolving small-scale features in solutions. However, refinement can be triggered by uninteresting or unphysical features as well. In binary black hole spacetimes, we are primarily interested in resolving the binary at the center of the grid

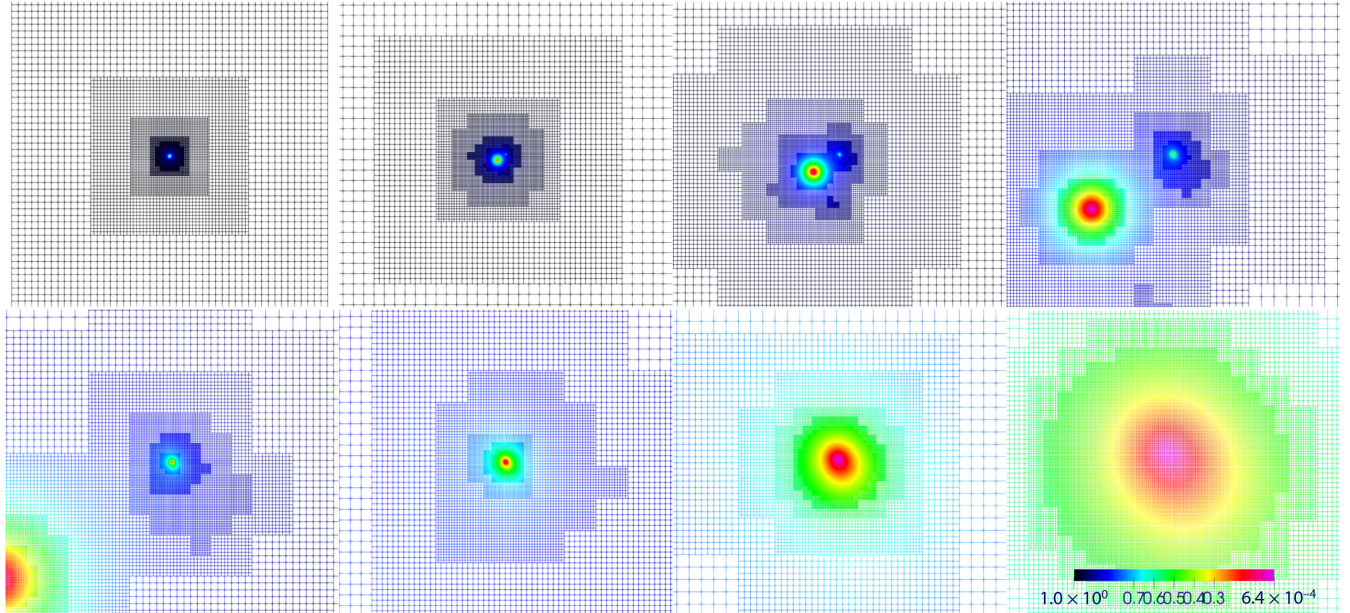


FIG. 3. This figure plots the lapse on the computational grid generated for a $q = 16$ black hole binary, after the system has evolved for two orbits. The top left frame shows the entire computational domain, and moving to the right, each frame successively moves toward the smaller of the two black holes. The computational grid is sparse, with refinement concentrated about the black holes, making it very computationally efficient. The 2:1 refinement constraint for constructing the grid, discussed in Sec. II F 2, is also apparent in the overall grid structure.

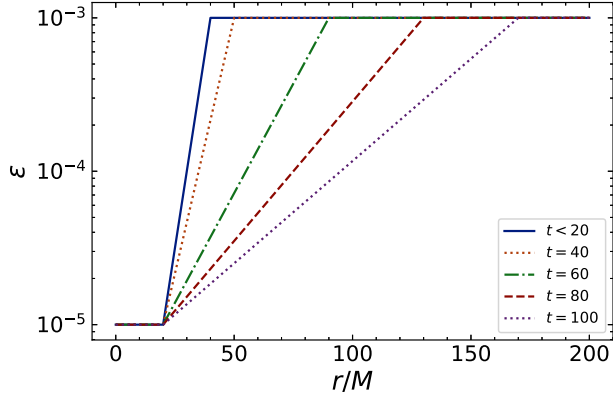


FIG. 4. This figure shows the wavelet tolerance $\epsilon(r)$ for refinement function RF3 at a few representative times. This refinement function is spherically symmetric, centered on the origin of the grid, and is independent of the black hole masses. The minimum wavelet tolerance is used over a relatively large region at the center of the grid. After $t = 20M$, the wavelet tolerance decreases in the GW extraction zone $50 < r/M < 100$, allowing the initial junk radiation to pass before triggering refinement in this region.

and following radiation out to the extraction region. To achieve computational efficiency, therefore, it is important to control where refinement occurs, focusing on physically interesting features in the solution.

One way to manage refinement in Dendro-GR is to set a maximum allowed level of refinement for the entire grid, $J_{\max}$. This limit is enforced globally for all times, and is chosen to allow for an expected minimum grid resolution. It is important to note that the spacetime at the black hole puncture is not smooth, and the WAMR grid will continue refining on this feature until the maximum level of refinement is reached. As we evolve binaries with large mass ratios, we need to prevent over-refinement of the more massive black hole in the binary. We modify the naive use of $J_{\max}$ by tracking the black hole locations and imposing a mass-dependent constraint on the maximum refinement level about each black hole. We refine a sphere expected to extend beyond the apparent horizon to the local maximum refinement level.

Recall that refinement in WAMR is controlled by the wavelet tolerance ϵ . Usually, ϵ is taken to be a constant. However, we have found that using a spatially dependent wavelet tolerance $\epsilon = \epsilon(r)$ allows us to focus refinement near the center of the grid and to reduce refinement beyond the wave extraction zone. We typically choose minimum and maximum values of ϵ , for the inner and outer regions of the grid, respectively, and let $\log \epsilon$ vary linearly between these limits.

Unfortunately, the situation is further complicated by junk radiation in the initial data and time-dependent gauge effects as the initial data relax onto the grid. This latter effect includes a fast moving gauge wave whose frequency becomes higher with increasing mass

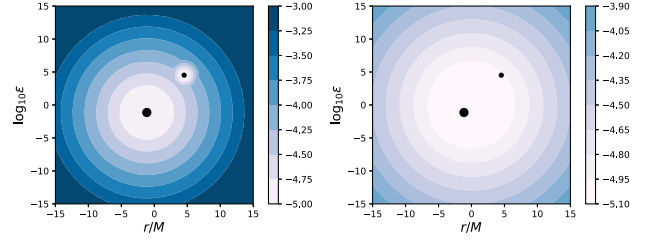


FIG. 5. This figure shows the wavelet tolerance $\epsilon(r)$ using refinement function RF4 for a $q = 4$ binary at two times, $t = 0$ and $t = 40M$. The black dots indicate representative positions and relative coordinate sizes for the two black holes, though not necessarily the physical horizons. This refinement function has the minimum tolerance centered about each black hole. After $t = 20M$, the refinement function becomes spherically symmetric and centered at the grid origin, with ϵ decreasing in the wave extraction region, similar to RF3.

ratio, q . These features trigger substantial refinement as q increases. Over-refining on this high-frequency radiation is a waste of computational resources. In order to limit over-refinement at early times, we have also found it beneficial to make ϵ a function of time near the beginning of the run.

For the purposes of this work, we define four refinement functions labeled RF2, RF3, RF4, and RF5. RF3 is time dependent, spherically symmetric, and linear in $\log \epsilon$, as shown in Fig. 4. RF2 is time independent, corresponding to RF3 for $t \geq 100M$. This refinement function works quite well for smaller values of q , such as $q \lesssim 5$. As q increases, however, this refinement function results in prohibitively expensive runs because of spurious waves originating around the smaller black hole. As a result, we introduce an additional spatial dependence to the refinement functions at early times RF4 and RF5 to more sharply focus refinement at early times around the individual black holes. Figures 5 and 6 show these two refinement functions at $t = 0$ and $t = 40M$. Beyond $t = 40M$, these refinement

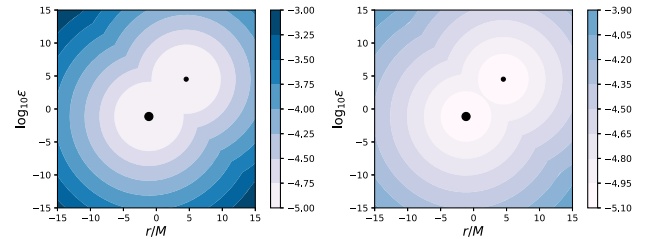


FIG. 6. This figure shows the wavelet tolerance $\epsilon(r)$ using refinement function RF5 for a $q = 4$ binary at two times, $t = 0$ and $t = 40M$. The black dots indicate representative positions and relative sizes for the two black holes, however, not the physical horizons. This refinement function has the minimum tolerance centered about each black hole. After $t = 20M$, the refinement function reduces to a functional form similar to RF3, but centered about each black hole, rather than the grid's origin.

functions become identical to RF3. Notice that refinement is concentrated in a region around the origin and the binary system, with ϵ increasing at larger radii. Both are also tuned in time to allow for sufficient resolution of the outgoing radiation in the extraction region, while limiting refinement on the initial burst of spurious radiation.

We note that the definitions of these refinement functions are *ad hoc* and tuned to the specific runs reported here through experimentation. When used with sufficient resolution, the refinement functions do not appear to interfere with or change significantly the convergence properties of Dendro-GR, as discussed in Sec. IV below, while significantly improving the computational efficiency of the runs. In future work, we will explore generalizations that could be more widely applicable.

F. Octree

1. Octree partitioning

Octree-based adaptive space discretizations (see Fig. 7) are commonly used in computational science applications [48,55,59–63]. Using octrees as the underlying data structure for spatial discretization is advantageous due to their simplicity, intrinsic hierarchical structure, and relative ease of use in designing scalable parallel algorithms.

In octree-based AMR applications, the local number of octants changes rapidly as the grid adapts and attempts to capture the spatially varying solution. This will create load imbalances between partitions that can reduce parallel performance. In order to maintain good load balancing, we need fast and efficient partitioning algorithms which, preferably, scale like $\mathcal{O}(n)$ where n is the number of octants. Doing so will also reduce the overall communication cost between partitions. To this end, we use space filling curves [64] with a flexible partitioning scheme [55]. Based on the order with which these curves traverse the octants, we are able to define a partial ordering operator on the octree domain, which, in turn, is used to sort the octree.

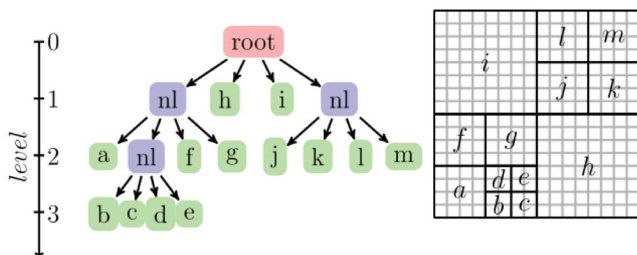


FIG. 7. A simple illustration of a 2D quadtree (in 3D, it would be an octree) as a data structure to represent a 2D adaptive grid. Note that we start from the root level and perform a hierarchical division of each dimension to generate spatially varying resolution on the computational domain. In terms of storage, we only store the leaf nodes of the tree since nonleaf nodes can be computed by performing a top-down or bottom-up traversal of the tree.

Once this happens, higher-dimensional partitioning reduces to a 1D problem along a curve.

2. Octree construction and balancing

Octree construction is the process of creating an adaptive octree discretization to capture a function $f: \Omega \rightarrow \mathcal{R}^n$ defined on a computational domain Ω . The wavelet expansion of f determines the adaptive structure for the user-specified tolerance function ϵ . Initially, we begin from the root level of the octree and continue refining if the computed wavelet coefficients are greater than ϵ . In our case, with the BSSN equations, the initial grid is generated based on TwoPuncture initial data (Sec. II B). All processes begin from the root level and continue refinement until at least p octants are produced (where p denotes the number of processes). These p octants are equally partitioned across processes. Further refinement occurs in an element-local fashion. As the number of octants increases with refinement, the octree is periodically repartitioned to ensure load balancing.

We enforce an additional constraint on the octree during refinement which we refer to as “2:1 grid balancing” [65]. This particular constraint enforces the condition that for a given octant in the octree, all of its geometric neighbors (faces, edges, and vertices) differ, at most, by a single level. Imposing this constraint ensures that the refinement structure varies smoothly through the entire grid. Moreover, we are guaranteed a correct interpolation stencil for points at level j from points at level $j - 1$. As a result, this simplifies the subsequent mesh generation process significantly.

3. Mesh generation

In order to perform numerical computations, the octree requires the notion of neighborhood information. The number of grid points placed in each octant depends on the degree of the finite-difference stencil or polynomial interpolant used. For d th-order finite differences, $(d + 1)^3$ points are placed on each octant. We refer to this representation as octant local. The wavelets are calculated via interpolations of the same order. As octants are shared through faces, edges, and vertices, neighboring octants will contain redundant information. These are efficiently identified and then removed in order to get the octant shared representation (see Fig. 8). We have two mappings between these two data representations which allow for finite-difference stencil computations of arbitrary order.

4. Evaluating the equations

All the field variables are defined in the compact octant shared, or zipped, representation. This zipped representation allows for efficient low overhead interprocess communication. However, to enable finite-difference (FD) computations, it is necessary to decompose the adaptive octree into smaller regular grid patches or blocks.

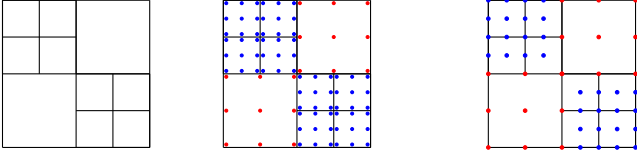


FIG. 8. This figure shows a 2D example of the octant local (center) and octant shared (right) nodal representation (with $d = 2$) of the adaptive quadtree shown on the left. Note that the octant local representation has grid points that are local to each octant and contains duplicate grid points in neighboring octants. By removing duplicate and hanging grid points, we get the octant shared representation. In this figure, grid points are color coded based on the octant level.

Following this decomposition from the octree to a block, we compute a padding region for which the width depends on the maximum FD stencil radius (see Fig. 9). The unzipped representation denotes the octant local representation together with the padding region constructed from the adaptive octree. This unzipped representation is purely local to each process and discarded after FD stencils are evaluated (see Fig. 9).

III. THE LazEv CODE

The LazEv code [37] was one of the two original codes to implement the moving puncture approach [38,66]. The current version uses the conformal function $W = \sqrt{\chi} = \exp(-2\phi)$ [67], eighth-order centered finite differencing in space [68], and a fourth-order Runge Kutta time integrator.

The LazEv code uses the Einstein Toolkit [28,29]/Cactus [69]/Carpet [70] infrastructure. The Carpet mesh refinement driver provides a “moving boxes” style of mesh refinement. In this approach, refined grids of fixed size are arranged about the coordinate centers of both holes. The Carpet code then moves these fine grids about the computational domain by following the trajectories of the two BHs.

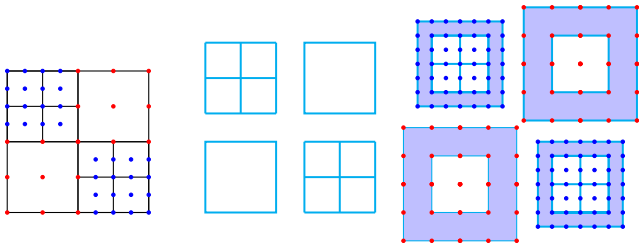


FIG. 9. This figure shows a simplified example of the octree to block decomposition and the unzip operation. The left figure shows the octant shared representation. The block decomposition is shown in the middle. Note that the given octree is decomposed into four regular blocks of different sizes. The right figure shows the decomposed blocks padded with values coming from neighboring octants with interpolated points when needed to give local uniform blocks.

The LazEv code implements both the BSSN [71–73] and CCZ4 [74] evolution systems. For the tests here, we use the BSSN system. For the gauge conditions, we use a modified 1+log lapse and a modified Gamma-driver shift condition [38,75,76],

$$(\partial_t - \beta^i \partial_i) \alpha = -2\alpha K, \quad (6a)$$

$$\partial_t \beta^a = (3/4) \tilde{\Gamma}^a - \eta(\vec{x}) \beta^a. \quad (6b)$$

For the function η , we choose

$$\eta(\vec{r}) = (\eta_c - \eta_o) \exp(-(r/\eta_s)^4) + \eta_o, \quad (7)$$

where $\eta_c = 2.0/M$, $\eta_s = 40.0M$, and $\eta_o = 0.25/M$. With this choice, η is small in the outer zones. The magnitude of η limits how large the time step can be with $dt_{\max} \propto 1/\eta$ [77]. Because this limit is independent of spatial resolution, it is only significant in the very coarse outer zones where the standard CFL condition would otherwise lead to a large value for $dt_{\max}$.

We use AHFinderDirect [78] to locate apparent horizons. We measure the magnitude of the horizon spin using the isolated horizon algorithm [79]. Note that once we have the horizon spin, we can calculate the horizon mass via the Christodoulou formula

$$m_H = \sqrt{m_{\text{irr}}^2 + S_H^2/(4m_{\text{irr}}^2)}, \quad (8)$$

where $m_{\text{irr}} = \sqrt{A/(16\pi)}$, A is the surface area of the horizon, and S_H is the spin angular momentum of the BH (in units of M^2).

We calculate the radiation scalar ψ_4 using the Antenna thorn [80,81]. We then extrapolate the waveform to an infinite observer location using perturbative formulas from [82].

While we use eighth-order centered difference stencils, we use a fifth-order Kreiss-Oliger dissipation stencil and fifth-order spatial prolongation operator (prolongation in time is second order). We found that a rather large-dissipation coefficient of $\epsilon_{\text{dis}} = 0.4$ gave the best results.

IV. TESTS

In this section, we present some numerical results to demonstrate the overall accuracy and performance of the Dendro-GR framework. We first present results that suggest that the maximum amount of Kreiss-Oliger dissipation should be used when solving BSSN-like formulations of the Einstein equations. Higher amounts of Kreiss-Oliger dissipation increase the rate of convergence observed in our tests. Second, we study binary black hole mergers with mass ratios $1 \leq q \leq 16$ using Dendro-GR. We show that these results converge to equivalent solutions obtained using LazEv. Finally, we present results on the numerical

performance of Dendro-GR. We discuss some of the refinement challenges in binary black hole spacetimes, and show how different refinement strategies affect the overall computational cost of the solution.

A. Effects of Kreiss-Oliger dissipation on BBH mergers

Kreiss-Oliger dissipation is widely used in numerical relativity. This dissipation is explicitly added to the numerical scheme to eliminate high-frequency noise that can arise in the evolution, especially near the puncture, where spacetime variables are nondifferentiable and at refinement boundaries. A common expectation is that one should minimize the amount of explicit dissipation provided that high-frequency noise is well controlled. However, when performing the initial comparisons of results from binary black hole mergers with Dendro-GR and LazEv, we found the opposite to be true.

In our tests, the fifth-order Kreiss-Oliger dissipation operator in Eq. (5) is added to the rhs of the semidiscrete equations with the parameter σ , $0 \leq \sigma < 1$. We performed multiple binary black hole mergers for $q = 1$ with different values of σ using both LazEv and Dendro-GR. Results from LazEv are shown in Fig. 10, which plots the coordinate separation between the two black holes. This figure shows that the runs with small dissipation $\sigma = 0.04$ differ from those with large dissipation, $\sigma \sim 0.4$. Further, the solution with small dissipation converges toward those with large dissipation with increasing resolution. Curiously, for the runs with large dissipation, the order of the spatial finite derivatives (4, 6, 8) and the order of the Kreiss-Oliger dissipation operator (5, 9) were not as important as the amount of dissipation, i.e., the value of σ . Similar results were obtained with Dendro-GR.

This result is counterintuitive, and we are not aware of a similar discussion in the literature. The numerical noise in the $\sigma = 0.04$ runs was well controlled, and visual inspection of the solutions did not indicate potential problems. However, when solving the BSSN equations for black hole spacetimes, better solutions at lower resolutions are obtained using larger amounts of explicit numerical dissipation.

B. Convergence tests for Dendro-GR and LazEv

Convergence is an important test not only of the computational code, but it is also the only way to establish an estimate of the overall error in the waveform. To test the convergence of both codes, we evolved initial data for an equal mass ($q = 1$), nonspinning binary. The initial data parameters are shown in Table I. For the $q = 1$ binary, we ran the LazEv code at three resolutions, $\Delta x = h_0$, $h_0/1.2$, and $h_0/1.44$ with $h_0 = 3.3M$, on the coarsest grid with nine levels of refinement. As shown in the center panel of Fig. 11, the waveform is not initially convergent, as relatively small stochastic errors owing to reflections of

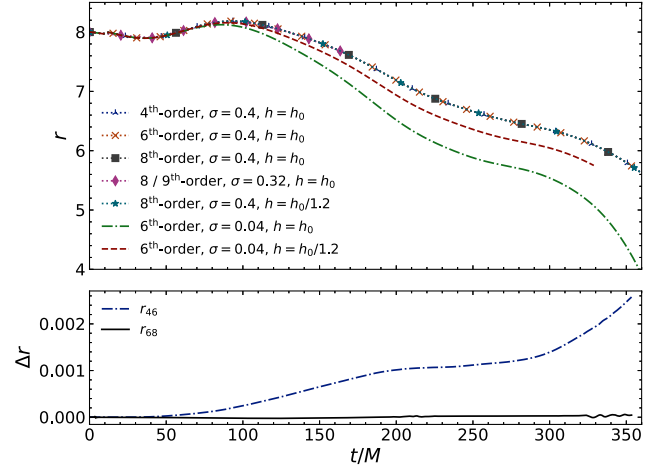


FIG. 10. This figure shows the effect of Kreiss-Oliger dissipation on the $q = 1$ binary BH merger using the LazEv code. Shown are results of the binary separation versus time using both different finite-difference orders and different dissipation amplitudes (σ). The top frame plots the coordinate separation (r) between the BHs as a function of the coordinate time for runs with large ($\sigma \sim 0.4$) and small ($\sigma = 0.04$) Kreiss-Oliger dissipation, different finite-difference orders (4, 6, 8), and at two resolutions. Fifth-order Kreiss-Oliger dissipation and fifth-order prolongation are used for all cases, except the one marked 8/9th order. The latter use eight-order finite differencing and both ninth-order dissipation and ninth-order prolongation. All of the high-dissipation cases computed with different FD orders are indistinguishable on this plot (dotted curves). The two low-dissipation-coefficient runs (sixth-order FD at two resolutions) converge to the higher-dissipation-coefficient results as $\mathcal{O}(h^5)$. The low-dissipation results, while convergent, show a systematically larger error than the higher-dissipation results. The lower panel plots the difference between the fourth- and sixth-order results ($r_{46} = r_{4\text{th}} - r_{6\text{th}}$) and the sixth- and eighth-order ($r_{68} = r_{6\text{th}} - r_{8\text{th}}$) results for $\sigma = 0.4$ (fifth-order dissipation). Note that even though the corresponding three curves in the top panel are indistinguishable, clear convergence with increasing order is seen in the lower panel. Note that h indicates the coarsest resolution of the AMR grid and $h_0 = 3.3M$.

high-frequency spurious radiation off the refinement boundaries dominate the error. As these high-frequency waves dissipate and the physical signal gets larger, convergence of the error becomes clear. The bottom panel shows that the waveform is convergent for the late inspiral at order 3.5. Note that Fig. 10 also shows convergence of the radial separation for the $q = 1$ case. The $q = 2$ binaries were run with base resolutions of $h_0/1.2$ and $h_0/1.4$, but added an additional level of refinement around the smaller BH. Similar convergence results were obtained for $q = 2$. Finally, for $q = 4$, we ran with a base resolution of $h_0/1.2$ and added two additional refinement levels (compared to $q = 1$) about the smaller black hole.

Dendro-GR uses an unstructured grid, and convergence is both more difficult to define and more challenging to demonstrate. Convergence depends both on the spatial

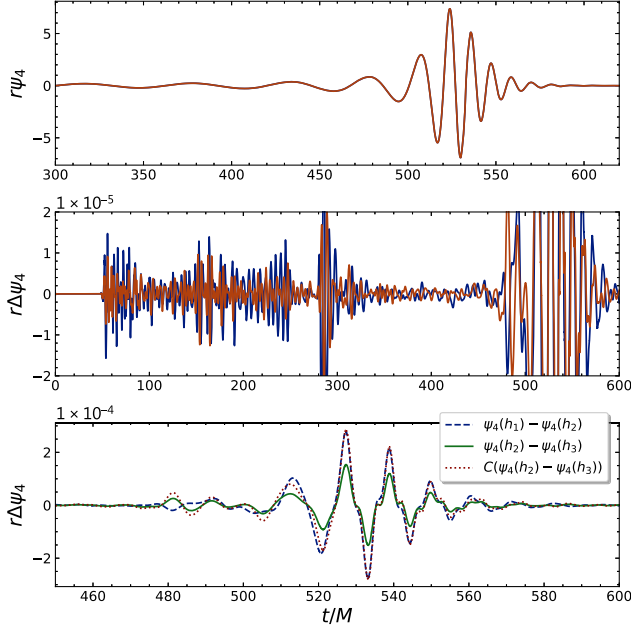


FIG. 11. Convergence test of a $q = 1$, nonspinning binary using LazEv. The top panel shows the low- and high-resolution waveforms. The middle panel shows the differences in the waveforms between the low and medium resolutions (in blue) and the medium and high resolutions (in red). Because the waveforms are of comparable size, initially there is small, but nonconvergent noise (at these resolutions). The bottom panel shows the differences rescaled, assuming 3.5-order convergence, at the peak of the waveform. At the peak, the stochastic AMR noise is smaller than the truncation error.

resolution Δx as well as the wavelet tolerance ϵ . Figure 12 shows the convergence of Dendro-GR solutions (for ψ_4) at two resolutions, labeled low (runs q2RF3l and q2RF4l) and medium (q2RF3m and q2RF4m), for $q = 2$ binaries. The highest-resolution LazEv ψ_4 is also plotted for comparison. With respect to changing Δx , the low- and medium-resolution runs converge to the LazEv solution.

As mentioned above, we choose the wavelet tolerance ϵ to be a function of both time and space in Dendro-GR. Thus, choosing different refinement functions can also potentially affect the solution. Figure 12 also shows this effect by plotting results from two different wavelet refinement functions, RF3 and RF4, for each resolution. In this case, the effect of changing the refinement function had a relatively small effect on the solution and the overall run-time; see Table II.

Figure 13 illustrates the effect of only varying ϵ on the solution. This figure compares the Dendro-GR waveforms for three values of $\epsilon_{\min} = \{10^{-3}, 10^{-5}, 10^{-6}\}$ with the highest-resolution LazEv waveform by plotting the difference. Clearly, the differences decrease with decreasing ϵ , as smaller values for ϵ trigger larger refined regions in the octree. While this is a form of convergence with respect to wavelet tolerance, the maximum refinement level $J_{\max}$ and

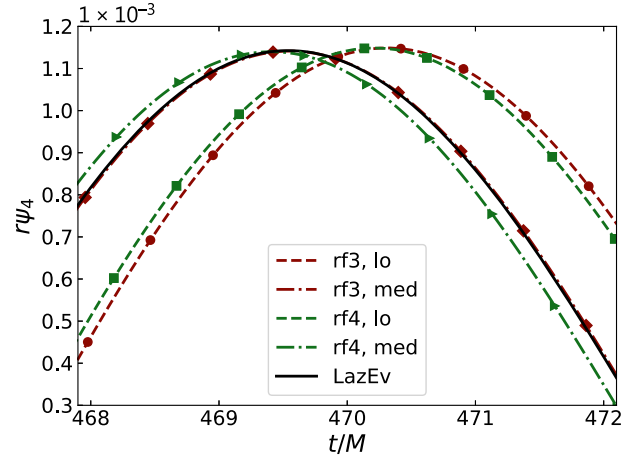


FIG. 12. This figure shows the convergence of Dendro-GR solutions with decreasing Δx . For $q = 2$, GW solutions were computed at two resolutions with two refinement functions with fixed $\epsilon_{\min}$. The low-resolution runs are plotted with dashed lines, with some representative points indicated with circles and squares. The higher-resolution runs are plotted with dashed dot lines, and representative points are indicated with triangles and diamonds. The RF3 solutions are in red and the RF4 in green. Both RF3 and RF4 solutions converge to the LazEv solution (solid black line) as the maximum refinement level is increased. The convergence is largely unaffected by the choice of refinement function.

the minimum resolution $\Delta x_{\min}$ are fixed, so this is not convergence in the Richardson sense of the term.

C. Dendro-GR binaries with different mass ratios

Table II gives some refinement and performance information for the Dendro-GR runs reported in this paper. The refinement information includes the maximum allowed

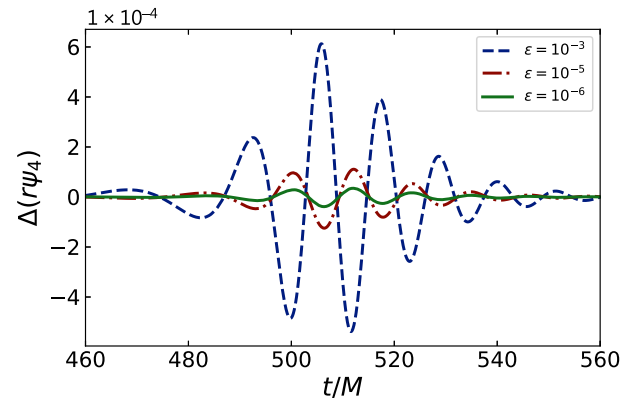


FIG. 13. This figure shows convergence with respect to ϵ given a fixed maximum refinement level. The plot shows the difference between the extracted GWs using the LazEv and Dendro-GR codes for the real part of the ψ_4 scalar with decreasing ϵ . The Dendro-GR solutions with decreasing ϵ converge to the LazEv waveforms. Refinement function RF3 was used here.

TABLE II. Some parameters and run-time information for the runs presented in this paper. All runs used wavelet tolerances $\epsilon_{\min} = 10^{-5}$ and $\epsilon_{\max} = 10^{-3}$. Runs were performed on Expanse at SDSC.

Run ID	Mass ratio $q = m_1/m_2$	$J_{\max}$ (BH2)	$\Delta x_{\min}$ (BH2)	$J_{\max}$ (BH1)	$\Delta x_{\min}$ (BH1)	RF	SUs ^a (cpu·h)	Wall time ^b (h)
q1RF2 ^c	1	15	4.069×10^{-3}	15	4.069×10^{-3}	2	...	...
q2RF3l	2	15	4.069×10^{-3}	14	8.138×10^{-3}	3	5 540	43
q2RF3m	2	16	2.034×10^{-3}	15	4.069×10^{-3}	3	41 170	80
q2RF4l	2	15	4.069×10^{-3}	14	8.138×10^{-3}	4	5 229	41
q2RF4m	2	16	2.034×10^{-3}	15	4.069×10^{-3}	4	39 521	77
q4RF2	4	16	2.034×10^{-3}	14	8.138×10^{-3}	2	22 717	89
q8RF3	8	18	5.086×10^{-4}	14	8.138×10^{-3}	3	485 810	483
q8RF4	8	18	5.086×10^{-4}	14	8.138×10^{-3}	4	101 915	318
q8RF5	8	18	5.086×10^{-4}	14	8.138×10^{-3}	5	64 477	263
q16RF4	16	19	2.543×10^{-4}	14	8.138×10^{-3}	4	799 590	1 149
q16RF5 ^c	16	19	2.543×10^{-4}	14	8.138×10^{-3}	5	...	...

^aHere, $SU = \sum_i c_i t_i$, where c_i is the number of CPUs used for a time t_i measured in hours, and i is an index that runs over all of the batch jobs used to complete the run. This measure of computational workload is not exact, as Dendro-GR regularly rebalances the workload, which may change the number of CPUs actually used in the simulation.

^bThis is also an imperfect measure of computational performance, as the wall-clock time depends on many factors, including the number of CPU cores available for the job, and the workload per core.

^cTiming information for these jobs is not available.

refinement level $J_{\max}$, the minimum resolution used in the run $\Delta_{\min}$, and the refinement function. The performance information provides an estimate for the total number of SUs, defined as the number of CPU · hours to complete the run. This number is approximated because Dendro-GR dynamically changes the number of active threads during a run. Finally, the table includes the total wall-clock time used to complete the run. While this information is valuable in providing a general view of Dendro-GR's performance, we caution that detailed conclusions cannot be drawn. First, the runs in this table were run over a long time period. During this time, code changes were made, and parameters were adjusted as we gained experience with the code. These changes impacted the computational costs of the runs. Second, wall-clock times depend on the number of cores used for each job, the final integration time, the workload per core, etc. For comparison, the LazEv $q = 1$ medium-resolution run used 27472 SUs, while the high-resolution run used 71651 SUs. The LazEv $q = 2$ medium-resolution run used 100766 SUs, while the high-resolution run used 228065 SUs. Finally, the LazEv $q = 4$ medium resolution used 169799 SUs, while the LazEv $q = 4$ high resolution used 474683 SUs. All the LazEv runs were performed on the same Intel Skylake cluster. Note that the LazEv runs were performed at relatively high resolution to ensure that the error in the LazEv simulations is small compared to the Dendro-GR simulations. These high-resolution runs are required because we will use the LazEv simulations to calibrate the accuracy of the Dendro-GR simulations.

Figures 14 and 15 show gravitational waveforms computed for nonspinning binaries with mass ratios up to $q = 16$. Parameters for the initial data are shown in

Table II, which also gives resolution and refinement function data, as well as the computational cost and time to solution. As noted in Sec. II B, the initial data for $q \geq 2$ are constructed from a single family of initial data [50], while data for $q = 1$ are constructed from *ad hoc* parameters. For $q = 1, 2$, and 4, the figures also show waveforms computed with LazEv. Because of the relatively long wall-clock time required, we chose not to complete the correspondingly high-resolution simulations for $q = 4$ simulations. Thus, the difference between the Dendro-GR and LazEv waveforms for $q = 4$ may only indicate that the LazEv simulation was underresolved. Importantly, due to its scaling, the Dendro simulations were obtained more quickly. The binaries with $q = 8$ and $q = 16$ were performed only with Dendro-GR. These figures show that Dendro-GR produces gravitational waveforms very similar to LazEv. The mismatch for these different waveforms are calculated below in Sec. IV D. Unfortunately, it is difficult to draw conclusions on the accuracy of Dendro-GR across different values of q , because these runs were performed over a long period of time with changing refinement strategies and a changing code base. Many of the code changes and new approaches were motivated, in fact, in the process of running these cases. We were not able to go back and rerun all cases with the same version of the code and consistent refinement criteria.

As discussed in Sec. II E, a gauge wave propagates across the computational domain at early times, as the coordinates transition from the Bowen-York gauge conditions used to calculate the initial data [46], to the puncture gauge conditions used in the evolution. The wavelength of the gauge wave is related to the black hole size, and thus the

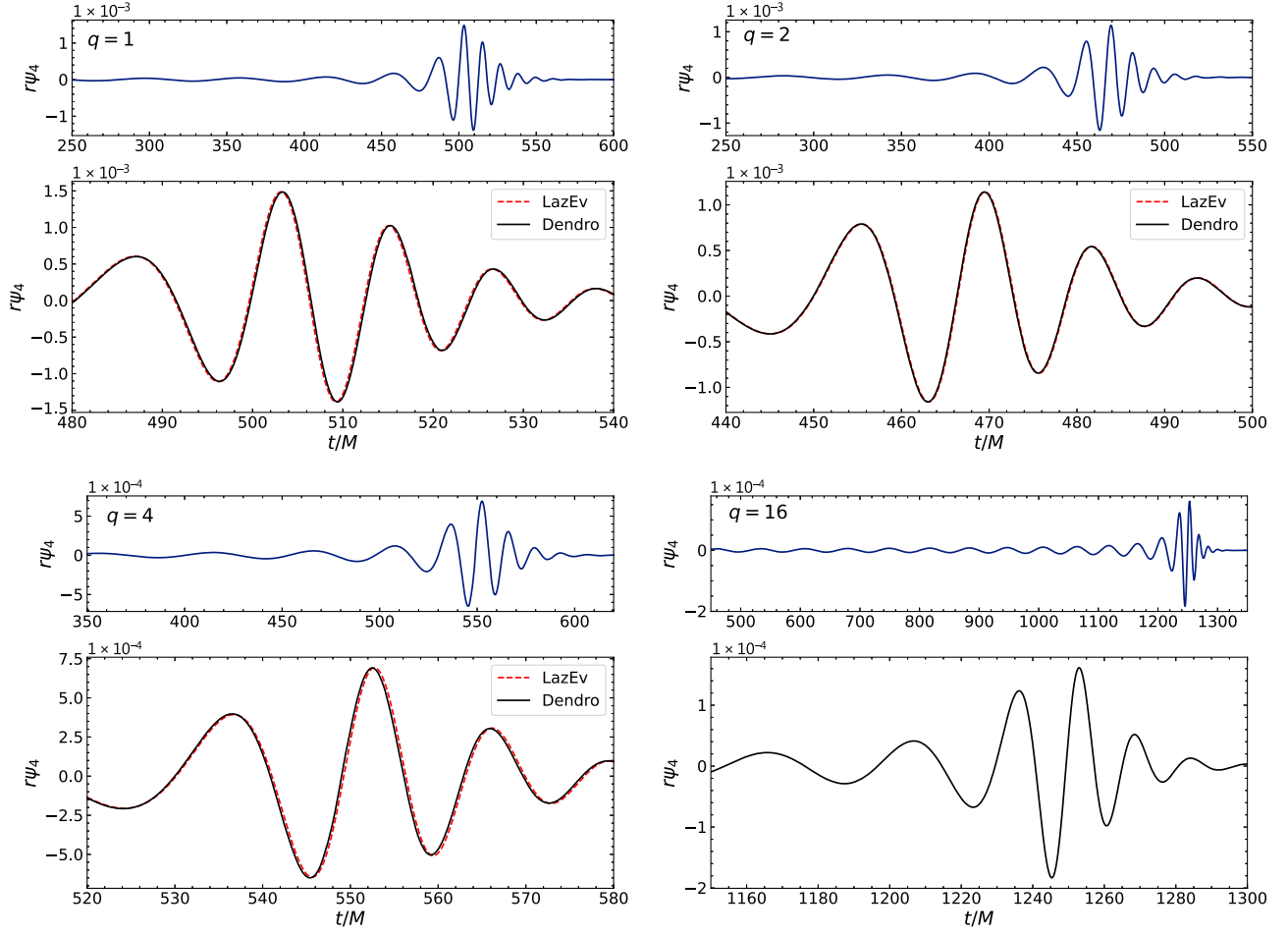


FIG. 14. Plots comparing the waveform for the $q = 1, 2, 4, 16$ cases. LazEv and Dendro-GR waveforms are given for all except $q = 16$, where only Dendro-GR results were produced. This figure plots the real part of ψ_4 for the highest-resolution Dendro-GR and LazEv (where available) runs.

frequency of this unphysical wave increases with mass ratio, q . The high-frequency wave triggered excessive refinement at the beginning of the higher mass ratio runs, particularly for $q \geq 8$, prompting our experimentation with different wavelet refinement functions. We ran simulations of the $q = 8$ binary with three different refinement functions, and plot the resulting waveforms in Fig. 15. RF3 uses $\epsilon_{\min}$ over a larger volume of the grid, while RF4 and RF5 allow for a larger wavelet tolerance over a larger region of the grid. Consequently, RF3 likely gives a more precise solution but at a greater computational cost as it may tend to over-refinement. While RF4 and RF5 are more computationally efficient, differences in the waveforms become noticeable. For $q = 16$, RF3 was too expensive, and this run was only done with RF4 and RF5. The RF5 run was not completed, as the differences in results for RF3 and RF5 for $q = 8$ were large. Interestingly, the differences in RF3 and RF4 occur only at the initial time. After $t = 40M$, both refinement functions are identical. The phase differences

seen in the figure seem to arise solely from small variations in the refinement at early times.

The number of computational cores used in the $q = 8$ and $q = 16$ runs are plotted in Figs. 16 and 17, respectively. Dendro-GR regularly repartitions the computational workload across the available cores. To balance the communication cost between cores, it will use fewer cores than the total number available if the workload per core drops below some threshold. In these runs, over-refinement on the high-frequency gauge wave and junk radiation remains a problem and causes the large increase in demand for computational resources at the beginning of the run. As this radiation moves beyond the gravitational wave extraction region, the grid is coarsened and the runs become much more efficient.

D. Overlaps

When using numerical waveforms for gravitational wave data analysis, numerical convergence provides an important estimation of the error in the numerical solution. Ideally, the

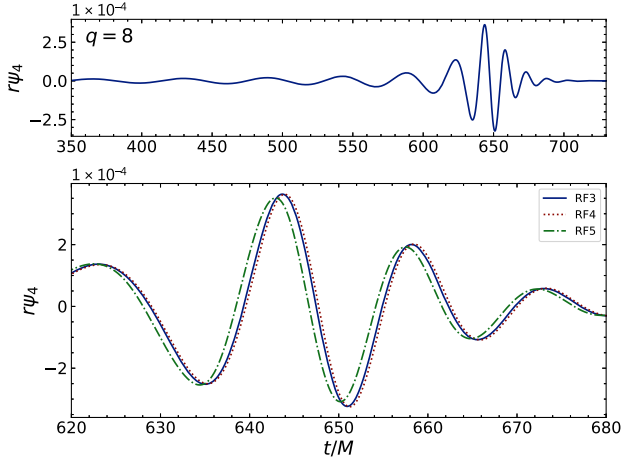


FIG. 15. This figure shows the real part of ψ_4 for a $q = 8$ black hole binary computed with Dendro-GR using three different refinement functions with the same minimum spatial resolution. RF3 has the smallest error tolerance at the center of the grid, but is very computationally expensive. The RF4 solution is quite similar to RF3, but at roughly one fifth of the computational cost (see Table II). The RF5 solution is the least expensive to compute, but for the parameters used here, the phase error is significant.

convergence error determined by comparing the solutions computed at two different resolutions is smaller than the other errors in the analysis. However, convergence testing overlooks the frequency response of a real-world detector. The overlap provides a way to compare two waveforms as

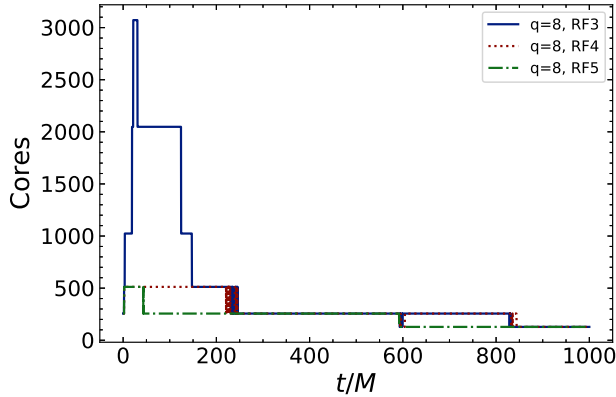


FIG. 16. This figure shows the effect of different refinement functions on the computational cost of evolving a $q = 8$ black hole binary with Dendro-GR by plotting the number of computational cores used during the simulation as a function of the computational time. The three refinement functions RF3, RF4, and RF5 differ only for $t \leq 40M$. Thus, the differences arise primarily in how the initial junk radiation and gauge waves are resolved. As shown in Fig. 15, the gravitational wave results from RF3 and RF4 are similar, although the maximum workload for RF4 was about 6 times smaller than the maximum workload for RF3. The results for RF5, while the most efficient run, show larger differences than the other two cases.

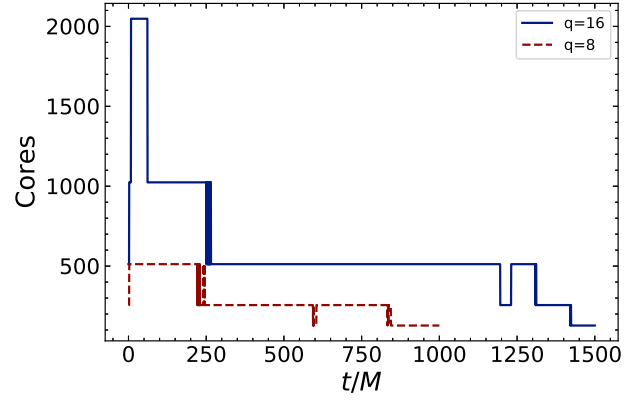


FIG. 17. This figure shows the number of computational cores used in $q = 8$ and $q = 16$ binary black hole merger simulations with RF4.

measured in a detector with a given frequency response. In addition, the overlap helps us to determine the computational resources required to simulate a particular configuration. This allows us to determine how similar two waveforms computed at different resolutions or with different codes are to one another.

For this analysis, we will measure the consistency of two waveforms using the `CreateCompatibleComplexOverlap` function in `LALSimUtils` (which is freely available) [83,84]. This function automatically optimizes over both time translations and phase shifts. Because of this, the mode-by-mode mismatch allows for the phase shifts of different modes to be inconsistent. That is, one expects each m mode to be shifted by $m\phi$.

Internally, this function uses the inner product

$$\langle h_1 | h_2 \rangle = 2 \int_{-\infty}^{\infty} \frac{h_1^*(f) h_2(f)}{S_n(f)} df, \quad (9)$$

where $h(f)$ is the Fourier transform of the complex waveform $h(t)$ and we use the Advanced-LIGO design sensitivity zero-detuned high-power noise curve [85] $S_n(f)$ with $f_{\min} = 20$ Hz and $f_{\max} = 2000$ Hz. This inner product is then further maximized over time and phase shifts as described in [86]

$$\langle h_1 | h_2 \rangle = \max_{t_0, \phi_0} \left[2 \left| \int_{-\infty}^{\infty} \frac{h_1^*(f) h_2(f)}{S_n(f)} df \right| \right]. \quad (10)$$

The overlap of two waveforms is then given by

$$\mathcal{O} = \frac{\langle h_1 | h_2 \rangle}{\sqrt{\langle h_1 | h_1 \rangle \langle h_2 | h_2 \rangle}}, \quad (11)$$

and the mismatch is given by

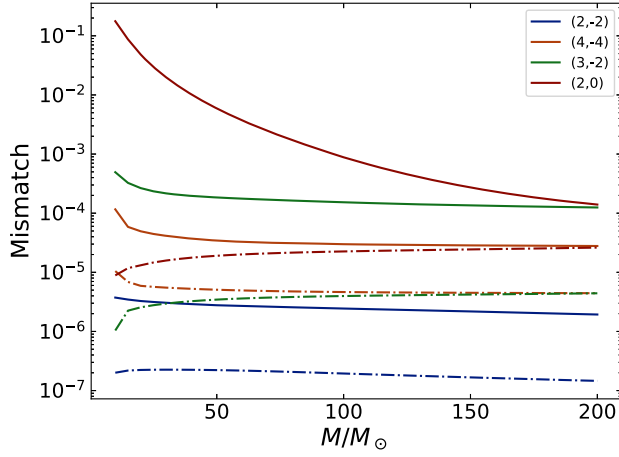


FIG. 18. This figure shows the mismatch in various (ℓ, m) gravitational waveform modes between the Dendro-GR and the high-resolution LazEv solutions for a $q = 1$ binary. The dot-dashed lines show the mismatch between the modes of the LazEv medium-resolution waveforms and the LazEv high-resolution waveforms. The solid lines show the overlaps of the Dendro-GR waveforms against the LazEv high-resolution waveforms. The modes are presented in order of decreasing amplitude. The $(2, 0)$ mode, which fails our accuracy goal of $\mathcal{M} < 0.005$, is subdominant to all the other modes.

$$\mathcal{M} = 1 - \mathcal{O}. \quad (12)$$

Because Eq. (9) directly involves the detector's noise sensitivity curve $S_n(f)$, the mismatch is a function of the actual frequency waveform and is not invariant under a

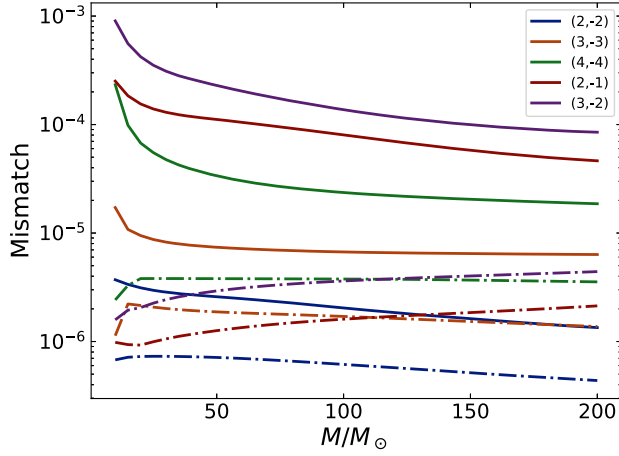


FIG. 19. This figure shows the mismatch in various (ℓ, m) gravitational waveform modes between the Dendro-GR and the high-resolution LazEv solutions for a $q = 2$ binary. The dot-dashed lines show the mismatch between the modes of the LazEv medium-resolution waveforms and the LazEv high-resolution waveforms. The solid lines show the overlaps of the Dendro-GR waveforms against the LazEv high-resolution waveforms. The modes are presented in order of decreasing amplitude [i.e., $(3, -3)$ is subdominant to $(2, -2)$, $(4, -4)$ to $(3, 3)$, etc.].

change in the total mass of the system. Depending on the total mass, only a portion of the waveform may be in the detector's sensitivity region. As an example, for systems with small total mass, LIGO is most sensitive to the low-frequency portion of the signal at early times. For systems with a large total mass, however, LIGO is most sensitive to the high-frequency merger and early ringdown portion of the waveforms.

We use the mismatch $\mathcal{M}$ to compare the Dendro-GR and LazEv waveforms. A mismatch of $\mathcal{M} < 0.005$ was determined in Ref. [87] to be minimally acceptable for Advanced LIGO analysis. Ideally, a mismatch $\mathcal{M} \ll 0.005$ is desired. However, this limit of < 0.005 is for the net mismatch in the observed waveforms (i.e., after summing all modes). Setting the mismatch tolerance to < 0.005 for all subdominant modes is therefore more restrictive than required. Here, we want to use the mismatch between the LazEv and Dendro-GR simulations to measure the truncation error in the Dendro-GR simulations. This is only true if the error in the LazEv simulations is much smaller than the Dendro-GR simulations. In an attempt to guarantee this, we require that the corresponding LazEv-to-LazEv mismatches (between the medium and high resolutions) are much smaller than the corresponding LazEv-to-Dendro-GR mismatches (we note that a small LazEv-to-LazEv mismatch may not account for all possible global errors). When this is not the case, the mismatch between the two codes is not a measure of the error of the Dendro-GR simulations. Figures 18–20 show the overlaps for different modes of ψ_4 computed with Dendro-GR and LazEv for the $q = 1$, $q = 2$, and $q = 4$ binaries, respectively. In particular, we use the

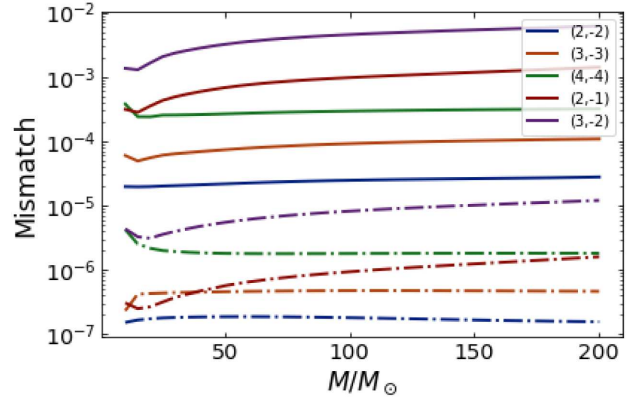


FIG. 20. This figure shows the mismatch in various (ℓ, m) gravitational waveform modes between the Dendro-GR and the high-resolution LazEv solutions for a $q = 4$ binary. The dot-dashed lines show the mismatch between the modes of the LazEv medium-resolution waveforms and the LazEv high-resolution waveforms. The solid lines show the overlaps of the Dendro-GR waveforms against the LazEv high-resolution waveforms. The modes are presented in order of decreasing amplitude [i.e., $(3, -3)$ is subdominant to $(2, -2)$, $(4, -4)$ to $(3, 3)$, etc.].

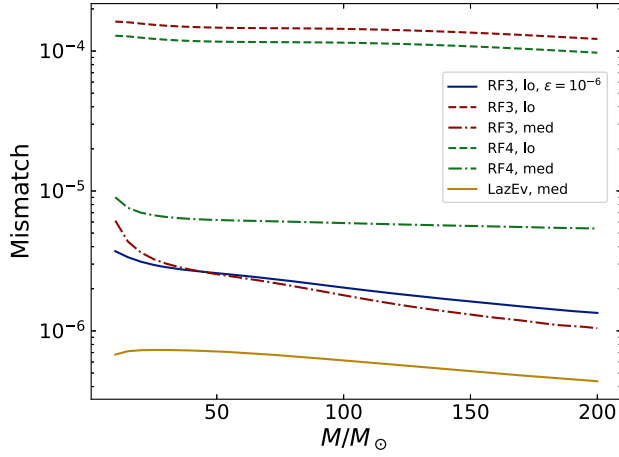


FIG. 21. This figure shows the overlaps in the $(\ell, m) = (2, -2)$ mode of ψ_4 computed with various Dendro-GR refinement criteria against the high-resolution LazEv solution for a $q = 2$ binary. The blue line is computed with $\epsilon = 10^{-6}$, and all other cases use $\epsilon = 10^{-5}$. As expected, the waveforms computed with the more expensive RF3 refinement function (red) have better overlaps to LazEv than those computed with RF4 (green). Finally, the tan solid line shows the overlap of the LazEv medium- and high-resolution waveforms.

high-resolution LazEv solutions as the base solutions for comparison with the Dendro-GR and medium-resolution LazEv waveforms. The figure shows the modes in order of decreasing amplitude. The $q = 1$ subdominant Dendro $(2,0)$ mode shows a significant mismatch with the corresponding LazEv mode. All modes with amplitude larger than the $(2,0)$ mode show much smaller mismatches between Dendro and LazEv. The $q = 2$ and $q = 4$ comparisons show similar behavior [here, more modes

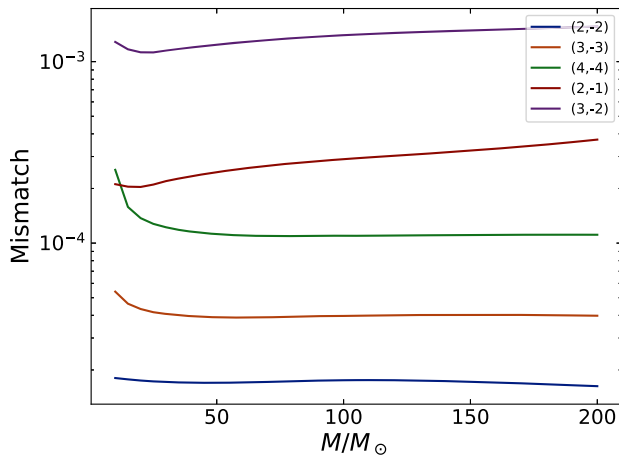


FIG. 22. This figure shows the overlaps in the Dendro-GR $q = 8$ gravitational waveforms computed with the RF3 and RF4 refinement functions. The waveforms are presented in order of decreasing amplitude. As expected, the dominant modes have smaller mismatches than the subdominant modes.

are nontrivial, and the $(2,0)$ mode is subdominant to all modes shown].

Figure 21 shows the mismatch in the $(\ell, m) = (2, -2)$ mode of ψ_4 between Dendro-GR waveforms computed with different refinement criteria and the high-resolution LazEv waveform. These mismatches compare solutions computed with different refinement functions RF3 and RF4 with a refinement tolerance $\epsilon = 10^{-5}$. The figure also shows the mismatch for a solution computed with RF3 and the refinement tolerance set to $\epsilon = 10^{-6}$. The main result from this figure is that higher resolution (more restrictive error tolerance) leads to a better agreement between LazEv and Dendro. Finally, the mismatch between the medium- and high-resolution LazEv solutions is shown. Consistent with the earlier convergence results, the Dendro-GR runs match the high-resolution LazEv solution well, and the RF3 solution is slightly closer. Finally, Fig. 22 shows the mismatch between the Dendro-GR solutions for the $q = 8$ binary computed with the RF3 and RF4 refinement functions.

V. DISCUSSION

This paper presented binary black hole evolutions performed with Dendro-GR for different mass ratios up to $q = 16$. We presented validation tests in comparison with results from LazEv, and we gave performance information for these runs.

While the focus of this paper is on evolving binary black holes with Dendro-GR, the first result that we presented has general applicability in the numerical relativity community. We found that in binary black hole evolutions with the BSSN formalism, the rate of convergence is increased when a large amount of Kreiss-Oliger dissipation is added to the solution. In our tests, runs with a dissipation parameter of $\sigma = 0.4$ had a better rate of convergence than runs with $\sigma = 0.04$, where σ is bounded by $\sigma < 1$ for numerical stability.

While the performance and scaling of Dendro-GR is very good, we are currently working on additional improvements. In particular, we are improving the unzip process, in which an octant of the tree is locally expanded to a uniform Cartesian grid, to reduce the communication overhead. As shown in this paper, we started exploring different ways to control the refinement algorithm, especially during the initial times of an evolution. We want to improve the computational performance of Dendro-GR, while not sacrificing the accuracy of the solutions. While we have had some initial success, much more work remains to be done. We continue working on a more general method to monitor errors in the evolution of black hole spacetimes. Finally, in an independent project, we are developing a version of Dendro-GR that runs primarily on GPUs.

The version of Dendro-GR used to produce the results in this paper is distributed subject to the MIT license in Ref. [88].

ACKNOWLEDGMENTS

This research was supported by NASA Grant No. 80NSSC20K0528 and NSF Grant No. PHY-1912883 (BYU). Y. Z. was also supported by NSF Grants No. OAC-2004044, No. PHY-2110338, and No. PHY-2207920. H. S. was also supported by NSF Grant No. PHY-1912930. This work used the Extreme Science and Engineering

Discovery Environment, which is supported by NSF Grant No. ACI-1548562, and the Frontera Pathways allocation, Grant No. PHY22019. Additional computational resources were provided by the GreenPrairies and WhiteLagoon clusters at the Rochester Institute of Technology, which were supported by NSF Grants No. PHY-2018420 and No. PHY-1726215.

-
- [1] R. Abbott *et al.* (LIGO Scientific, Virgo, and KAGRA Collaborations), GWTC-3: Compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run, [arXiv:2111.03606](#).
 - [2] R. Abbott *et al.* (LIGO Scientific, Virgo, and KAGRA Collaborations), The Population of Merging Compact Binaries Inferred Using Gravitational Waves through GWTC-3, [arXiv:2111.03634](#) [Phys. Rev. X (to be published)].
 - [3] B. P. Abbott *et al.* (Virgo and LIGO Scientific Collaborations), Observation of Gravitational Waves from a Binary Black Hole Merger, *Phys. Rev. Lett.* **116**, 061102 (2016).
 - [4] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral, *Phys. Rev. Lett.* **119**, 161101 (2017).
 - [5] B. P. Abbott, R. Abbott, T. D. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Properties of the Binary Neutron Star Merger GW170817, *Phys. Rev. X* **9**, 011001 (2019).
 - [6] J. Miller, L. Barsotti, S. Vitale, P. Fritschel, M. Evans, and D. Sigg, Prospects for doubling the range of Advanced LIGO, *Phys. Rev. D* **91**, 062005 (2015).
 - [7] L. Barsotti, L. McCuller, M. Evans, and P. Fritschel, The A+ design curve, Technical Report No. T1800042-v5, 2018.
 - [8] D. McClelland, M. Cavaglia, M. Evans, R. Schnabel, B. Lantz, V. Quetschke, and I. Martin, Instrument science white paper, Technical Report No. T1600119-v4, 2016.
 - [9] H. Abe *et al.*, The current status and future prospects of KAGRA, the large-scale cryogenic gravitational wave telescope built in the Kamioka underground, *Galaxies* **10**, 63 (2022).
 - [10] J. Lange *et al.*, Parameter estimation method that directly compares gravitational wave observations to numerical relativity, *Phys. Rev. D* **96**, 104041 (2017).
 - [11] B. P. Abbott *et al.* (LIGO Scientific and Virgo Collaborations), Directly comparing GW150914 with numerical solutions of Einstein's equations for binary black hole coalescence, *Phys. Rev. D* **94**, 064035 (2016).
 - [12] Y. Pan, A. Buonanno, L. T. Buchman, T. Chu, L. E. Kidder, H. P. Pfeiffer, and M. A. Scheel, Effective-one-body waveforms calibrated to numerical relativity simulations: coalescence of non-precessing, spinning, equal-mass black holes, *Phys. Rev. D* **81**, 084041 (2010).
 - [13] Y. Pan, A. Buonanno, M. Boyle, L. T. Buchman, L. E. Kidder, H. P. Pfeiffer, and M. A. Scheel, Inspiral-merger-ringdown multipolar waveforms of nonspinning black-hole binaries using the effective-one-body formalism, *Phys. Rev. D* **84**, 124052 (2011).
 - [14] A. Taracchini, Y. Pan, A. Buonanno, E. Barausse, M. Boyle, T. Chu, G. Lovelace, H. P. Pfeiffer, and M. A. Scheel, Prototype effective-one-body model for nonprecessing spinning inspiral-merger-ringdown waveforms, *Phys. Rev. D* **86**, 024011 (2012).
 - [15] Y. Pan, A. Buonanno, A. Taracchini, L. E. Kidder, A. H. Mroué, H. P. Pfeiffer, M. A. Scheel, and B. Szilágyi, Inspiral-merger-ringdown waveforms of spinning, precessing black-hole binaries in the effective-one-body formalism, *Phys. Rev. D* **89**, 084006 (2014).
 - [16] R. Cotesta, A. Buonanno, A. Bohé, A. Taracchini, I. Hinder, and S. Ossokine, Enriching the symphony of gravitational waves from binary black holes by tuning higher harmonics, *Phys. Rev. D* **98**, 084028 (2018).
 - [17] S. Husa, S. Khan, M. Hannam, M. Pürrer, F. Ohme, X. Jiménez Forteza, and A. Bohé, Frequency-domain gravitational waves from nonprecessing black-hole binaries. I. New numerical waveforms and anatomy of the signal, *Phys. Rev. D* **93**, 044006 (2016).
 - [18] L. London, S. Khan, E. Fauchon-Jones, C. García, M. Hannam, S. Husa, X. Jiménez-Forteza, C. Kalaghatgi, F. Ohme, and F. Pannarale, First Higher-Multipole Model of Gravitational Waves from Spinning and Coalescing Black-Hole Binaries, *Phys. Rev. Lett.* **120**, 161102 (2018).
 - [19] C. García-Quirós, M. Colleoni, S. Husa, H. Estellés, G. Pratten, A. Ramos-Buades, M. Mateu-Lucena, and R. Jaume, Multimode frequency-domain model for the gravitational wave signal from nonprecessing black-hole binaries, *Phys. Rev. D* **102**, 064002 (2020).
 - [20] M. Punturo *et al.*, The Einstein Telescope: A third-generation gravitational wave observatory, *Classical Quantum Gravity* **27**, 194002 (2010).
 - [21] D. Reitze *et al.*, Cosmic Explorer: The U.S. contribution to gravitational-wave astronomy beyond LIGO, *Bull. Am. Astron. Soc.* **51**, 035 (2019), [arXiv:1907.04833](#).
 - [22] S. Dwyer, D. Sigg, S. W. Ballmer, L. Barsotti, N. Mavalvala, and M. Evans, Gravitational wave detector with cosmological reach, *Phys. Rev. D* **91**, 082001 (2015).
 - [23] C. Cutler and M. Vallisneri, LISA detections of massive black hole inspirals: Parameter extraction errors due to inaccurate template waveforms, *Phys. Rev. D* **76**, 104018 (2007).

- [24] M. Pürrer and C.-J. Haster, Gravitational waveform accuracy requirements for future ground-based detectors, *Phys. Rev. Res.* **2**, 023151 (2020).
- [25] D. Ferguson, K. Jani, P. Laguna, and D. Shoemaker, Assessing the readiness of numerical relativity for LISA and 3G detectors, *Phys. Rev. D* **104**, 044037 (2021).
- [26] B. Daszuta, F. Zappa, W. Cook, D. Radice, S. Bernuzzi, and V. Morozova, GR-Athena++: Puncture evolutions on vertex-centered oct-tree adaptive mesh refinement, *Astrophys. J. Suppl. Ser.* **257**, 25 (2021).
- [27] T. Andrade *et al.*, GRChombo: An adaptable numerical relativity code for fundamental physics, *J. Open Source Software* **6**, 3703 (2021).
- [28] S. R. Brandt *et al.*, The Einstein Toolkit, 2021; to find out more, visit <http://einstein toolkit.org>.
- [29] F. Löffler *et al.*, The Einstein Toolkit: A community computational infrastructure for relativistic astrophysics, *Classical Quantum Gravity* **29**, 115001 (2012).
- [30] AMReX Development Team, AMReX: A software framework for massively parallel, block-structured adaptive mesh refinement (AMR) applications, <https://amrex-codes.github.io/amrex/> (accessed October 27, 2022).
- [31] L. E. Kidder *et al.*, SPECTRE: A task-based discontinuous Galerkin code for relativistic astrophysics, *J. Comput. Phys.* **335**, 84 (2017).
- [32] W. Tichy, A. Adhikari, and L. Ji, Numerical relativity with the new Nmesh code, *Bull. Am. Phys. Soc.* **65**, D15.00004 (2020), <https://meetings.aps.org/Meeting/APR20/Session/D15.4>.
- [33] W. Tichy, L. Ji, A. Adhikari, and A. Rashti, The current status of the Nmesh code, *Bull. Am. Phys. Soc.* **67**, B16.00008 (2022), <https://meetings.aps.org/Meeting/APR22/Session/B16.8>.
- [34] D. Hilditch, A. Weyhausen, and B. Brügmann, Pseudo-spectral method for gravitational wave collapse, *Phys. Rev. D* **93**, 063006 (2016).
- [35] C. Palenzuela, B. Miñano, A. Arbona, C. Bona-Casas, C. Bona, and J. Massó, simflowny 3: An upgraded platform for scientific modeling and simulation, *Comput. Phys. Commun.* **259**, 107675 (2021).
- [36] B. T. N. Gunney and R. W. Anderson, Advances in patch-based adaptive mesh refinement scalability, *J. Parallel Distrib. Comput.* **89**, 65 (2016).
- [37] Y. Zlochower, J. G. Baker, M. Campanelli, and C. O. Lousto, Accurate black hole evolutions by fourth-order numerical relativity, *Phys. Rev. D* **72**, 024021 (2005).
- [38] M. Campanelli, C. O. Lousto, P. Marronetti, and Y. Zlochower, Accurate Evolutions of Orbiting Black-Hole Binaries Without Excision, *Phys. Rev. Lett.* **96**, 111101 (2006).
- [39] M. Alcubierre, *Introduction to 3+1 Numerical Relativity*, International Series of Monographs on Physics (Oxford University Press, Oxford, 2008).
- [40] T. W. Baumgarte and S. L. Shapiro, *Numerical Relativity: Solving Einstein's Equations on the Computer* (Cambridge University Press, Cambridge, England, 2010).
- [41] M. Shibata, *Numerical Relativity* (World Scientific Publishing Co., Inc., River Edge, NJ, 2015).
- [42] L. Rezzolla and O. Zanotti, *Relativistic Hydrodynamics* (Oxford University Press, Oxford, 2013).
- [43] D. Neilsen, S. L. Liebling, M. Anderson, L. Lehner, E. O'Connor, and C. Palenzuela, Magnetized neutron stars with realistic equations of state and neutrino cooling, *Phys. Rev. D* **89**, 104029 (2014).
- [44] B. Bruegmann, J. A. Gonzalez, M. Hannam, S. Husa, U. Sperhake, and W. Tichy, Calibration of moving puncture simulations, *Phys. Rev. D* **77**, 024027 (2008).
- [45] N. T. Bishop and L. Rezzolla, Extraction of gravitational waves in numerical relativity, *Living Rev. Relativity* **19**, 2 (2016).
- [46] M. Ansorg, B. Brügmann, and W. Tichy, A single-domain spectral method for black hole puncture data, *Phys. Rev. D* **70**, 064011 (2004).
- [47] V. I. Lebedev, Spherical quadrature formulas exact to orders 25–29, *Sib. Math. J.* **18**, 99 (1977).
- [48] M. Fernando, D. Neilsen, H. Lim, E. Hirschmann, and H. Sundar, Massively parallel simulations of binary black hole intermediate-mass-ratio inspirals, *SIAM J. Sci. Comput.* **41**, C97 (2019).
- [49] D. Neilsen, L. Lehner, C. Palenzuela, E. W. Hirschmann, S. L. Liebling, P. M. Motl, and T. Garret, Boosting jet power in black hole spacetimes, *Proc. Natl. Acad. Sci. U.S.A.* **108**, 12641 (2011).
- [50] J. Healy, C. O. Lousto, H. Nakano, and Y. Zlochower, Post-Newtonian quasicircular initial orbits for numerical relativity, *Classical Quantum Gravity* **34**, 145011 (2017).
- [51] Z. B. Etienne, NRPyTUTORIAL, <https://github.com/zachetienne/nrpytutorial>, 2022.
- [52] I. Ruchlin, Z. B. Etienne, and T. W. Baumgarte, SENR/NRPy+: Numerical relativity in singular curvilinear coordinate systems, *Phys. Rev. D* **97**, 064036 (2018).
- [53] M. Fernando, D. Neilsen, E. W. Hirschmann, and H. Sundar, A scalable framework for adaptive computational general relativity on heterogeneous clusters, in *Proceedings of the ACM International Conference on Supercomputing* (Association for Computing Machinery, New York, NY, 2019), pp. 1–12, [10.1145/3330345.3330346](https://doi.org/10.1145/3330345.3330346).
- [54] J. DeBuhr, B. Zhang, M. Anderson, D. Neilsen, and E. W. Hirschmann, Relativistic hydrodynamics with wavelets, *Astrophys. J.* **867**, 112 (2018).
- [55] M. Fernando, D. Duplyakin, and H. Sundar, Machine and application aware partitioning for adaptive mesh refinement applications, in *Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing* (Association for Computing Machinery, New York, NY, 2017), pp. 231–242, [10.1145/3078597.3078610](https://doi.org/10.1145/3078597.3078610).
- [56] G. Deslauriers and S. Dubuc, Symmetric iterative interpolation processes, *Constr. Approx.* **5**, 49 (1989).
- [57] D. L. Donoho, Interpolating wavelet transforms, Technical Report No. 408, Department of Statistics, Stanford University (1992).
- [58] M. Holmström, Solving hyperbolic PDEs using interpolating wavelets, *SIAM J. Sci. Comput.* **21**, 405 (1999).
- [59] C. Burstedde, L. C. Wilcox, and O. Ghattas, p4EST: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees, *SIAM J. Sci. Comput.* **33**, 1103 (2011).

- [60] R. S. Sampath, S. S. Adavani, H. Sundar, I. Lashuk, and G. Biros, Dendro: Parallel algorithms for multigrid and AMR methods on 2:1 balanced octrees, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis* (ACM/IEEE, 2008), [10.1109/SC.2008.5218558](#).
- [61] A. Ahimian, I. Lashuk, S. Veerapaneni, C. Aparna, D. Malhotra, I. Moon, R. Sampath, A. Shringarpure, J. Vetter, R. Vuduc, D. Zorin, and G. Biros, Petascale direct numerical simulation of blood flow on 200k cores and heterogeneous architectures, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis* (ACM/IEEE, 2010), [10.1109/SC.2010.42](#).
- [62] T. Weinzierl, The Peano software—Parallel, automaton-based, dynamically adaptive grid traversals, *ACM Trans. Math. Softw.* **45**, 14 (2019).
- [63] J. Bédorf, E. Gaburov, and S. Portegies Zwart, Bonsai: A GPU tree-code, [arXiv:1204.2280](#).
- [64] L. Valgaerts, Space-filling curves an introduction, *Joint Advanced Student School (JASS)*, Course 2: The Interplay of Mathematical Modeling and Numerical Simulation (2005), http://www.mayr.informatik.tu-muenchen.de/konferenzen/Jass05/courses/2/Valgaerts/Valgaerts_paper.pdf.
- [65] H. Sundar, R. Sampath, and G. Biros, Bottom-up construction and 2:1 balance refinement of linear octrees in parallel, *SIAM J. Sci. Comput.* **30**, 2675 (2008).
- [66] J. G. Baker, J. Centrella, D.-I. Choi, M. Koppitz, and J. van Meter, Gravitational Wave Extraction from an Inspiral Configuration of Merging Black Holes, *Phys. Rev. Lett.* **96**, 111102 (2006).
- [67] P. Marronetti, W. Tichy, B. Bruegmann, J. Gonzalez, and U. Sperhake, High-spin binary black hole mergers, *Phys. Rev. D* **77**, 064010 (2008).
- [68] C. O. Lousto and Y. Zlochower, Foundations of multiple black hole evolutions, *Phys. Rev. D* **77**, 024034 (2008).
- [69] Cactus Computational Toolkit, <https://cactuscode.org>.
- [70] E. Schnetter, S. H. Hawley, and I. Hawke, Evolutions in 3D numerical relativity using fixed mesh refinement, *Classical Quantum Gravity* **21**, 1465 (2004).
- [71] T. Nakamura, K. Oohara, and Y. Kojima, General relativistic collapse to black holes and gravitational waves from black holes, *Prog. Theor. Phys. Suppl.* **90**, 1 (1987).
- [72] M. Shibata and T. Nakamura, Evolution of three-dimensional gravitational waves: Harmonic slicing case, *Phys. Rev. D* **52**, 5428 (1995).
- [73] T. W. Baumgarte and S. L. Shapiro, Numerical integration of Einstein's field equations, *Phys. Rev. D* **59**, 024007 (1998).
- [74] D. Alic, C. Bona-Casas, C. Bona, L. Rezzolla, and C. Palenzuela, Conformal and covariant formulation of the Z4 system with constraint-violation damping, *Phys. Rev. D* **85**, 064040 (2012).
- [75] M. Alcubierre, B. Brügmann, P. Diener, M. Koppitz, D. Pollney, E. Seidel, and R. Takahashi, Gauge conditions for long-term numerical black hole evolutions without excision, *Phys. Rev. D* **67**, 084023 (2003).
- [76] J. R. van Meter, J. G. Baker, M. Koppitz, and D.-I. Choi, How to move a black hole without excision: Gauge conditions for the numerical evolution of a moving puncture, *Phys. Rev. D* **73**, 124011 (2006).
- [77] E. Schnetter, Time step size limitation introduced by the BSSN gamma driver, *Classical Quantum Gravity* **27**, 167001 (2010).
- [78] J. Thornburg, A fast apparent-horizon finder for 3-dimensional Cartesian grids in numerical relativity, *Classical Quantum Gravity* **21**, 743 (2004).
- [79] O. Dreyer, B. Krishnan, D. Shoemaker, and E. Schnetter, Introduction to isolated horizons in numerical relativity, *Phys. Rev. D* **67**, 024018 (2003).
- [80] M. Campanelli, B. J. Kelly, and C. O. Lousto, The Lazarus project. II. Space-like extraction with the quasi-Kinnersley tetrad, *Phys. Rev. D* **73**, 064005 (2006).
- [81] J. G. Baker, M. Campanelli, and C. O. Lousto, The Lazarus project: A pragmatic approach to binary black hole evolutions, *Phys. Rev. D* **65**, 044001 (2002).
- [82] H. Nakano, J. Healy, C. O. Lousto, and Y. Zlochower, Perturbative extraction of gravitational waveforms generated with numerical relativity, *Phys. Rev. D* **91**, 104022 (2015).
- [83] C. Pankow, P. Brady, E. Ochsner, and R. O'Shaughnessy, Novel scheme for rapid parallel parameter estimation of gravitational waves from compact binary coalescences, *Phys. Rev. D* **92**, 023002 (2015).
- [84] J. Lange, R. O'Shaughnessy, and M. Rizzo, Rapid and accurate parameter inference for coalescing, precessing compact binaries, [arXiv:1805.10457](#).
- [85] LIGO Scientific Collaboration, Advanced LIGO anticipated sensitivity curves, Report No. T0900288-v3, 2011.
- [86] F. Ohme, M. Hannam, and S. Husa, Reliability of complete gravitational waveform models for compact binary coalescences, *Phys. Rev. D* **84**, 064029 (2011).
- [87] L. Lindblom, B. J. Owen, and D. A. Brown, Model waveform accuracy standards for gravitational wave data analysis, *Phys. Rev. D* **78**, 124020 (2008).
- [88] <https://github.com/paralab/Dendro-GR>.