PHYSICAL REVIEW D **76,** 124029 (2007)

Framework for large-scale relativistic simulations in the characteristic approach

Roberto Gómez[*]
Pittsburgh Supercomputing Center, 300 S. Craig Street, Pittsburgh, Pennsylvania 15213, USA
and Department of Physics, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA

Willians Barreto[†]
Centro de Física Fundamental, Facultad de Ciencias, Universidad de Los Andes, Mérida, Venezuela

Simonetta Frittelli[‡]
Department of Physics, Duquesne University, Pittsburgh, Pennsylvania 15282, USA
and Department of Physics and Astronomy, University of Pittsburgh, Pittsburgh, Pennsylvania 15260, USA
(Received 2 August 2007; published 26 December 2007)

We present a new computational framework (LEO) that enables us to carry out the very first large-scale, high-resolution computations in the context of the characteristic approach in numerical relativity. At the analytic level, our approach is based on a new implementation of the eth formalism, using a nonstandard representation of the spin-raising and spin-lowering angular operators in terms of nonconformal coordinates on the sphere; we couple this formalism to a partially first-order reduction (in the angular variables) of the Einstein equations. The numerical implementation of our approach supplies the basic building blocks for a highly parallel, easily extensible numerical code. We demonstrate the adaptability and excellent scaling of our numerical code by solving, within our numerical framework, for a scalar field minimally coupled to gravity (the Einstein-Klein-Gordon problem) in 3 dimensions. The nonlinear code is globally second-order convergent, and has been extensively tested using as a reference a calibrated code with the same initial and boundary data and radial marching algorithm. In this context, we show how accurately we can follow quasinormal mode ringing. In the linear regime, we show energy conservation for a number of initial data sets with varying angular structure. A striking result that arises in this context is the saturation of the flow of energy through the Schwarzschild radius. As a final calibration check, we perform a large simulation with resolution never achieved before.

DOI: 10.1103/PhysRevD.76.124029 PACS numbers: 04.25.Dm, 04.30.Db, 04.70.Bw, 04.20.Ex

I. INTRODUCTION

The characteristic approach has been used successfully to carry out numerical simulations of space-times with and without sources [1–8]. A significant computational effort is nevertheless necessary to extend its range of applicability to the simulation of astrophysically relevant sources of gravitational radiation, such as the black hole–neutron star binary problem, where the approach can be most useful. As work on the characteristic formulation to date illustrates [6,9–11], it is clear that most three-dimensional characteristic simulations, even vacuum simulations [9], are resolution limited. This is particularly true of three-dimensional simulations of systems containing compact matter sources [10], even when the matter source is an extended one. In general, all these simulations have been limited in resolution primarily because of the time required to integrate the equations numerically. For instance, at the finest resolution simulation considered in [10], tracking a neutron star in (close) orbit around a black hole requires approximately 1.5 months even on one of the fastest processors currently available. This is so even though the grid in question (81×81 angular points, 123 points radially) is fairly moderate by today's standards. To a lesser extent, characteristic simulations are also limited because of memory requirements, although the characteristic scheme is particularly economic in this regard. Even though it is feasible to equip a single-processor workstation with the 1.4 gigabytes of memory required by that moderate grid size, the time required for the numerical solution on even the fastest processor would make such serial simulations highly impractical.

Most of the past code development in the characteristic approach [3,6] has been geared towards vector or single-processor machines. The computational platforms available today require instead a parallel programming approach in order to perform large resolution simulations in a reasonable time; thus a parallel version of the characteristic code is clearly needed. In the present work we show how, with a well thought out yet modest programming effort, it is not only possible to produce an efficient, highly scalable parallel implementation of characteristic codes, but to do so in such a way that it becomes straightforward to extend our parallel implementation to new physical models.

We aim for our numerical implementation to be particularly useful for longtime simulations of sources of astrophysical interest, which are very demanding in terms of the

[*]gomez@psc.edu
[†]wbarreto@ula.ve
[‡]simo@mayu.physics.duq.edu

number of grid points on which to advance the solution, and thus on the number of floating point operations required. With that end in mind, our numerical code must scale well in platforms with a large number of processors. We show here that our implementation meets this goal, making it a potentially valuable tool when applied to some of the most interesting astrophysical applications of numerical relativity, such as the study of black hole–neutron star binary systems, in the close orbit regime up to the tidal disruption of the companion star.

The first astrophysical application we have in mind is a characteristic simulation of boson stars in orbit about a black hole. For that purpose we calibrate the current code for a massless scalar field minimally coupled with gravitation. This lets us, beyond numerical tests, compare and calibrate our code with linear versions of it, radial codes, and analytic (perturbative) results reported in the literature about quasinormal modes. The extension to a black hole–boson star system is straightforward but not trivial, deserving a detailed study of its own, as well as enormous computational resources. We want to stress that the production of gravitational waves by the scattering of scalar waves has many mathematical features in common with the production of gravitational waves by the motion of fluid bodies.

The article is organized as follows: In Sec. II we review briefly the standard numerical implementation of the *eth* approach [1], and discuss some of its drawbacks when applied to high-resolution simulations in the characteristic approach to numerical relativity. In Sec. III we present an implementation of the $\eth$ and $\bar{\eth}$ operators which combines their standard description in terms of stereographic coordinates with their numerical representation on nonconformal grid coordinates on the sphere. As this approach differs significantly from our previous work, we provide further motivation for this departure. Section IV provides a detailed description of the numerical implementation of the approach outlined in Sec. III. In Sec. V we illustrate how the parallel, scalable characteristic code framework (LEO) that we have developed can be used to implement the model problem of a scalar field minimally coupled to gravity in three dimensions. Section VI expands on additional numerical considerations specific to the hypersurface and evolution equations, and to the boundary conditions, and presents convergence tests of our numerical implementation. In Sec. VII, the viability of the approach is demonstrated by clearly resolving several problems which could not be tackled previously. We close in Sec. VIII with concluding remarks and an outline of future work.

II. THE STANDARD *ETH* APPROACH IN CHARACTERISTIC NUMERICAL RELATIVITY

The characteristic approach to numerical relativity is based on null coordinates $x^\alpha = (u, r, x^A)$, with u the retarded time, r a luminosity distance, and x^A coordinates on the sphere [1–8,12,13]. In its three-dimensional implementation, the angular coordinates chosen are stereographic coordinates $x^A = (\zeta, \bar{\zeta})$ on the sphere, which is covered with two stereographic coordinate patches, as first presented in Ref. [1]. We summarize here the salient aspects of the approach to provide the motivation for (and highlight the differences with) the implementation described in this article. The standard *eth* approach [1] is a straightforward numerical implementation in stereographic coordinates $x^A = (\zeta, \bar{\zeta})$ of the $\eth$, $\bar{\eth}$ operators introduced by Newman and Penrose [14,15]. Two stereographic patches are used to cover the unit sphere, with coordinates $\zeta_N = \tan(\theta/2)e^{i\phi}$ in the north patch and $\zeta_S = 1/\zeta_N$ in the south patch, respectively, where (θ, ϕ) are standard angular coordinates.

In terms of the dyad $q^A = P(1, i)$, where $P = 1 + \zeta\bar{\zeta}$, vectors U^A on the sphere are represented by a spin-weight-1 field $U = q^A U_A$ (or alternatively, by a spin-weight-1 field $\bar{U} = \bar{q}^A U_A$). This treatment generalizes to tensors on the sphere $T_{A\ldots N+M}$, which are represented in terms of spin-weighted functions obtained by contracting them with the dyad q^A and its complex conjugate $\bar{q}^A$, i.e.

$$\Psi = q^{A_1} \ldots q^{A_N} \bar{q}^{A_{N+1}} \ldots \bar{q}^{A_{N+M}} \times T_{A_1 \ldots A_N A_{N+1} \ldots A_{N+M}}. \quad (1)$$

The spin of the resulting scalar function is $s = N - M$. Angular derivatives of tensor fields are represented by the action of the spin-raising and spin-lowering operators $\eth$ and $\bar{\eth}$. For example, the angular derivatives of a vector field $\nabla_A U_B$, where ∇_A are the derivatives compatible with the flat sphere metric in the coordinates x^A, are represented by the spin-2 field $\eth U$ and spin-0 field $\bar{\eth} U$ given by

$$\eth U = q^A q^B \nabla_A U_B, \qquad \bar{\eth} U = \bar{q}^A q^B \nabla_A U_A. \quad (2)$$

The $\eth$ and $\bar{\eth}$ operators acting on a spin-weight s function Ψ are equivalently defined by

$$\eth \Psi = P^{1-s} \partial_{\bar{\zeta}} (P^s \Psi) = (1 + \zeta\bar{\zeta}) \partial_{\bar{\zeta}} \Psi + s\zeta\Psi, \quad (3a)$$

$$\bar{\eth} \Psi = P^{1+s} \partial_{\zeta} (P^{-s} \Psi) = (1 + \zeta\bar{\zeta}) \partial_{\zeta} \Psi - s\bar{\zeta}\Psi, \quad (3b)$$

where, in terms of the (real) coordinates (q, p), $\zeta = q + ip$, $\partial_\zeta = \partial_q - i\partial_p$, $\partial_{\bar{\zeta}} = \partial_q + i\partial_p$. Functions on the sphere with spin-weight s transform between patches according to

$$\Psi_N = \left(-\frac{\bar{\zeta}_S}{\zeta_S} \right)^s \Psi_S. \quad (4)$$

We implement this numerically by laying down a two-dimensional grid on each patch, with coordinates (q_m, p_n), $\zeta_{m,n} = q_m + ip_n$, such that $q_m = -1 + (m - 3)\Delta$, $p_n = -1 + (n - 3)\Delta$, $\Delta = 2/(N - 5)$, and with the grid point indices in the range $m, n = 1 \ldots N$. This grid covers the coordinate range $-1 - 2\Delta \leq (q, p) \leq 1 + 2\Delta$. Ghost zones are used on each side of the grid for the discretization of angular derivatives by centered, second-order-accurate,

finite-difference stencils. Function values at these ghost zones are obtained by interpolation from the function values on the opposite patch. In order to compute second angular derivatives to second-order accuracy, the interpolations must be evaluated to fourth-order accuracy, which can be readily attained using a 16-point stencil in two dimensions [1], provided the grid covers no less than the range indicated above. For some applications [9], we find it necessary to extend the grid to a finite overlap, i.e. $|q| \leq 1 + \epsilon$, with $\epsilon \geq 2\Delta$.

The set of ghost zones required for the north patch maps onto the south patch (and vice versa), as per the transformation $\zeta_N = 1/\zeta_S$, into a cloverleaf shape. Regardless of the number of grid points (or of ghost zones), there is a finite overlap between the north and south patches. While the points in the overlap area of each grid are redundant, we carry them all because we find it more efficient to work with rectangular grids. A potential problem of doing so is the development of two different numerical solutions in the overlap area of each patch, only loosely coupled at the stereographic patch edge.

We now discuss briefly some of the drawbacks of the standard ''eth'' approach, when it is applied to high-resolution simulations in the characteristic approach to numerical relativity, and the motivation for the changes that we propose in the next section.

Parallelization of existing characteristic codes

The first objection that we encounter is in the process of parallelizing our characteristic codes. Because of the radial march implicit in the radial integration of the hypersurface equations, the natural way to parallelize a characteristic simulation is to distribute the angular grid among processors, which would integrate the equations along a ''pencil'' of null rays. In the computational eth approach, this means assigning the computation of the solution over a subset of each stereographic patch to a given processor. A similar arrangement, in the context of axisymmetric simulations, was explored earlier by Bishop *et al.* [16].

Thus, given $M \times M$ processors, we can simply partition the $N \times N$ stereographic grid on each patch, assigning equal square subgrids of extent N/M on each direction to each processor. Load balancing (the requirement that all processors in a parallel computation do approximately the same amount of work) would, in principle, be achieved, so long as we restricted ourselves to explicit methods, thus guaranteeing that the time spent per subgrid remains constant. The communication pattern imposed by the two stereographic patches implementation of the eth approach [1] does present an obstacle to effective scaling. The mapping of ghost zones at the edge of the grid to grid points in the opposing patch is not restricted to nearest neighbors. Providing values for these ghost zones requires data from a set of grid points whose values are scattered among processors in an irregular pattern (in the sense that, depending on its location on the angular grid, the ghost zones may be obtained from one, two, or more processors). This procedure is not only cumbersome to program, but intrinsically inefficient.

If the data required for these ghost zones could be obtained just from grid points at the edge of an adjoining grid, the procedure would simplify considerably. The time spent in communication would be substantially reduced and remain constant over the set of processors, with a significant impact on the scalability and overall efficiency of a code. Unfortunately, such an arrangement is not possible with a stereographic grid. In addition, as we point out at the end of Sec. II, because of the fixed overlap between patches, a significant portion of the grid is wasted. While this might be acceptable in small scale simulations [6,9–11], it needs to be addressed in the context of large-scale computations, as in that case it translates into a serious waste of computational resources.

Yet another problem that arises from the angular grid layout is that of highly nonuniform angular resolution, as a direct consequence of using a stereographic grid. Considering the expression for the area element in stereographic coordinates,

$$ds^2 = \frac{4}{(1 + \zeta\bar{\zeta})^2} d\zeta d\bar{\zeta}, \tag{5}$$

it can be seen that there is a marked disparity between the resolutions at grid points at various places on the sphere. Considerably better resolution is attained near the equator than at the poles, with the largest disparity between a grid zone at the pole ($\zeta\bar{\zeta} = 0$) and a point at a corner of the grid ($\zeta\bar{\zeta} = 2$) where the respective area elements have a ratio of 9:1. For some situations, such as the case of a matter source in equatorial orbit around a black hole, this feature works to our advantage. Conversely, for a matter source in a polar orbit, the matter source would be resolved 3 times less accurately when it lies along the z axis ($q = p = 0$) compared to the resolution obtained when it crosses the equator, which in a second-order accurate code translates into a 9-fold increase in the intrinsic error in the numerical solution. One possible correction for this effect would be to maintain the standard stereographic grid as the basic computational grid, but to introduce a physical grid related to the computational grid by a fish-eye stretch in the angular coordinates, and allow the refined portion of the grid to follow the compact object.

A related issue is that of proper resolution of angular features, which is critical for characteristic simulations of compact objects in orbit around a black hole. In spherical coordinates, angular resolution decreases with distance to the center, as pointed out in [10]. There is then a limit to the distance at which we can initialize a characteristic simulation, a limit which is not dictated by physical considerations, such as the need to avoid the formation of caustics, which would lead to a breakdown of the coordinate system.

We are also limited by the number of points which can, in practice, be devoted to resolve a companion object.

A proposed extension to the characteristic approach [10] would include an adaptive mesh refinement (AMR) strategy, which indirectly would address both grid resolution issues mentioned. While this extension may prove necessary in simulations of the last stages of capture or disruption of a companion star by a black hole, the necessary technology has not yet been developed in the context of the characteristic approach; to our knowledge, applications of AMR in the characteristic framework have been made only in simplified one-dimensional models [17]. The intermediate stages of the black hole–neutron star problem, where the companion remains approximately contained in a finite region, could, at least in principle, be equally well resolved with fixed mesh refinement, an approach successfully used, for example, in [18–20]. The introduction of AMR carries with it a whole new set of issues, not the least of which is the problem of load balancing in a massively parallel computer. Current characteristic codes present serious obstacles to the implementation of an AMR strategy by the nature of the angular grid alone. To be effective, an AMR implementation would have to be capable of dealing with refined meshes in multiple coordinate patches. To our knowledge, few existing implementations have this capability [21], and none have been applied in numerical relativity. In the present work we present a parallel implementation on a single distributed grid, and we defer the discussion of possible techniques for fixed and adaptive mesh refinement for future work.

III. AN OPERATOR BASED ON NONCONFORMAL PROJECTIONS

A key consideration for the present work is that different numerical representations of the *eth* approach can be developed by laying down different types of grids on the sphere. We consider here an alternative to the standard eth approach, based on the ''cubed sphere'' or ''gnomic'' covering of the sphere introduced by Ronchi *et al.* [22], based on earlier work of Sadourny [23]. This approach is now in common use in global weather simulations, such as in the MIT General Circulation Model (MITgcm), for example [24]. It has also been applied in astrophysical simulations [25,26], in the study of wave propagation methods on the sphere [27], and more recently, in the evolution of scalar fields in a fixed background [28]. While preparing this manuscript, we learned [29] of work being carried out by Bishop *et al.* [30], on a similar grid arrangement, based on work by Thornburg [31].

In the cubed-sphere method, a covering of the unit sphere with six nonoverlapping patches results from projecting the sphere from its center onto the six faces of a circumscribing cube, whose edges have length two. For example, for points on the sphere with Cartesian coordinates (x, y, z) and angular coordinate $-\pi/4 \leq \theta \leq \pi/4$ (where $z = \cos\theta$), we project the point by tracing a line from the center of the sphere through the point (x, y, z) to the $z = 1$ face of the circumscribing cube, determining a point with Cartesian coordinates $(U, V, 1)$; we then label the point on the sphere according to the Cartesian coordinates (U, V) of its projection on the plane $z = 1$.

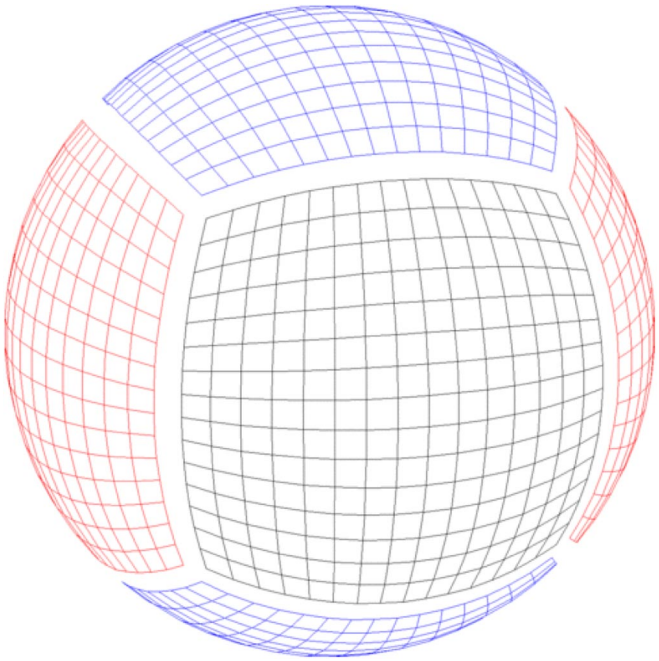

FIG. 1 (color online). The cubed sphere: covering of the sphere with six nonoverlapping gnomic patches.

In order to obtain a covering of the sphere with nearly uniform area, we label the points on the sphere by angular coordinates (α, β), where $U = \tan(\alpha)$, $V = \tan(\beta)$, introducing an equally spaced grid in the angular coordinates (α, β), i.e. $(\alpha_i, \beta_j) = (i\Delta_\zeta, j\Delta_\zeta)$, $i, j = -N \ldots N$, $\Delta_\zeta = \pi/(4N)$. Similar projections from the center of the sphere, as shown in Fig. 1, to the other faces of the cube provide a covering of the sphere with six patches.

For a finite-difference code, this arrangement is ideal, as the grid on each patch is equally spaced in the angular coordinates, and the same angular coordinate is used on any two patches in the direction perpendicular to a boundary. Thus any additional layers of ghost zones in the adjacent grid will fall on coordinate lines parallel to the boundary. Evaluating function values at those ghost zones requires only one-dimensional interpolation along coordinate lines parallel to the boundary. For example, an Nth-order centered stencil requires $N/2$ additional layers of ghost cells to be supplied along the edge of each spherical cap. The order of the interpolations used to supply these points can be selected so as to preserve the accuracy and maintain the desired dissipation properties of the numerical scheme. In the standard configuration [22], the spherical caps share common points along the edges of the grid, where two grids abut, and at the corners of each patch,

where three grids meet. These common points must have a unique value on each of the grids that share them. An approach advocated in the literature [27] is to replace the function values at these shared points by some form of weighted average. Here we dispense with this procedure, by carefully selecting the range of the angular coordinates in each patch, in a way that precludes the existence of points common to two (or three) cubed-sphere patches.

An advantage of a *gnomic* decomposition of the sphere, crucial to an efficient parallel implementation of the numerical ð approach, is that the patches can be laid out so that they are nonoverlapping. In contrast, in a stereographic covering, there is a finite overlap, which does not shrink in size as we increase the grid resolution. Although it is possible to construct a six-patch stereographic covering of the sphere, in which the amount of overlap is reduced with respect to the two-patch covering, the area of the overlap zone remains constant, regardless of grid size.

A. Nonconformal projections of the sphere

Nonconformal projections are based on nonorthogonal coordinates; thus there is no symmetry in some operators, such as the Laplacian on the sphere. It is still straightforward to couple a gnomic grid layout with the existing ð approach, i.e. while continuing to express the ð and $\bar{\text{ð}}$ operators of Ref. [1] on stereographic coordinates. In the following we detail how this is implemented by expressing the angular (stereographic) derivatives in terms of angular derivatives in gnomic coordinates. A gnomic covering of the sphere is given by six coordinate patches:

$$(x_1, y_1, z_1) = \frac{1}{D_1}(1, U_1, V_1), \tag{6a}$$

$$(x_2, y_2, z_2) = \frac{1}{D_2}(U_2, 1, V_2), \tag{6b}$$

$$(x_3, y_3, z_3) = \frac{1}{D_3}(-1, -U_3, V_3), \tag{6c}$$

$$(x_4, y_4, z_4) = \frac{1}{D_4}(U_4, -1, V_4), \tag{6d}$$

$$(x_5, y_5, z_5) = \frac{1}{D_5}(-V_5, U_5, 1), \tag{6e}$$

$$(x_6, y_6, z_6) = \frac{1}{D_6}(V_6, U_6, -1), \tag{6f}$$

where (x_i, y_i, z_i), $i = 1 \ldots 6$ are the Cartesian coordinates of the points, (U_i, V_i) are coordinates on the sphere in the range $-\sqrt{2} - 1 \le U_i, \quad V_i \le \sqrt{2} - 1$, and $D_i = \sqrt{1 + U_i^2 + V_i^2}$. [Numerical grid points can be set equally spaced in the coordinates (α, β), related to (U_i, V_i) by $U_i = \tan(\alpha_i), \quad V_i = \tan(\beta_i)$ with $-\pi/4 \le \alpha_i, \quad \beta_i \le \pi/4$]. The coordinate lines at $U_i = \text{const}$ ($V_i = \text{const}$) are great circles on the sphere which pass through the points where the Cartesian axis intersects the sphere. For instance, the lines at $U_3 = \text{const}$ are great circles spun around the x axis, while those of $V_3 = \text{const}$ are great circles rotated around the y axis, with α_3, β_3 the respective rotation angles. Similarly, on each patch, we define stereographic coordinates $\zeta_i = u_i + i v_i$, where the angular coordinates (u_i, v_i) on each patch are related to Cartesian coordinates by

$$(x_1, y_1, z_1) = \frac{1}{P_1}(2 - P_1, 2u_1, 2v_1), \tag{7a}$$

$$(x_2, y_2, z_2) = \frac{1}{P_2}(-2u_2, 2 - P_2, 2v_2), \tag{7b}$$

$$(x_3, y_3, z_3) = \frac{1}{P_3}(-2 + P_3, -2u_3, 2v_3), \tag{7c}$$

$$(x_4, y_4, z_4) = \frac{1}{P_4}(2u_4, -2 + P_4, 2v_4), \tag{7d}$$

$$(x_5, y_5, z_5) = \frac{1}{P_5}(-2v_5, 2u_5, 2 - P_5), \tag{7e}$$

$$(x_6, y_6, z_6) = \frac{1}{P_6}(2v_6, 2u_6 - 2 + P_6), \tag{7f}$$

with $P_i = 1 + u_i^2 + v_i^2$. The gnomic coordinates (U, V) and stereographic coordinates (u, v) on each patch are related by

$$U = \frac{2u}{1 - u^2 - v^2}, \qquad V = \frac{2v}{1 - u^2 - v^2}, \tag{8}$$

or, in more compact form, in terms of a complex gnomic coordinate $\xi = U + iV$,

$$\xi = \frac{2\zeta}{1 - \zeta\bar{\zeta}}, \qquad \zeta = \frac{\xi}{1 + \sqrt{1 + \xi\bar{\xi}}}. \tag{9}$$

B. The eth operator on the cubed sphere

We can express the angular derivatives in stereographic coordinates, ∂_ζ, $\partial_{\bar{\zeta}}$, which enter into the ð and $\bar{\text{ð}}$ operators in terms of angular derivatives (u, v) through the relations

$$\frac{\partial}{\partial\zeta} = \frac{1}{2}\left(\frac{\partial}{\partial u} - i\frac{\partial}{\partial v}\right), \qquad \frac{\partial}{\partial\bar{\zeta}} = \frac{1}{2}\left(\frac{\partial}{\partial u} + i\frac{\partial}{\partial v}\right), \tag{10}$$

and with the Jacobian

$$\begin{pmatrix} \frac{\partial U}{\partial u} & \frac{\partial U}{\partial v} \\ \frac{\partial V}{\partial u} & \frac{\partial V}{\partial v} \end{pmatrix} = \frac{2}{(1 - u^2 - v^2)^2} \times \begin{pmatrix} 1 + u^2 - v^2 & 2uv \\ 2uv & 1 - u^2 + v^2 \end{pmatrix} \tag{11}$$

we cast derivatives with respect to stereographic (u, v) in terms of derivatives with respect to gnomic coordinates (U, V), which are in turn related to derivatives with respect to gnomic coordinates (α, β) by

$$\frac{\partial}{\partial U} = \frac{1}{(1 + U^2)}\frac{\partial}{\partial\alpha}, \qquad \frac{\partial}{\partial V} = \frac{1}{(1 + V^2)}\frac{\partial}{\partial\beta}. \tag{12}$$

The spin-raising and spin-lowering operators $\eth$ and $\bar{\eth}$ [32] acting on a spin s function Ψ, Eq. (3), can be written as

$$\eth\Psi = (1+\zeta\bar{\zeta})\left(\frac{1}{1+\bar{\zeta}^2}\frac{\partial\Psi}{\partial\alpha} + \frac{i}{1-\bar{\zeta}^2}\frac{\partial\Psi}{\partial\beta}\right) + s\zeta\Psi, \tag{13}$$

$$\bar{\eth}\Psi = (1+\zeta\bar{\zeta})\left(\frac{1}{1+\zeta^2}\frac{\partial\Psi}{\partial\alpha} - \frac{i}{1-\zeta^2}\frac{\partial\Psi}{\partial\beta}\right) - s\bar{\zeta}\Psi, \tag{14}$$

where the values of $(\zeta, \bar{\zeta})$ for each grid point are computed from the respective $(\xi, \bar{\xi})$ values as per Eq. (9).

To complete the prescription of the $\eth$ operator on the cubed sphere, we must specify the transformation rule for spin-weighted functions on the sphere. The stereographic coordinates on the various patches transform according to

$$\zeta_1 = \frac{\zeta_4 - 1}{\zeta_4 + 1}, \qquad \zeta_2 = \frac{\zeta_1 - 1}{\zeta_1 + 1},$$
$$\zeta_3 = \frac{\zeta_2 - 1}{\zeta_2 + 1}, \qquad \zeta_4 = \frac{\zeta_3 - 1}{\zeta_3 + 1}, \tag{15a}$$

$$\zeta_1 = -i\frac{\zeta_5 + i}{\zeta_5 - i}, \qquad \zeta_2 = -i\frac{\zeta_5 - 1}{\zeta_5 + 1},$$
$$\zeta_3 = -i\frac{\zeta_5 - i}{\zeta_5 + i}, \qquad \zeta_4 = -i\frac{\zeta_5 + 1}{\zeta_5 - 1}, \tag{15b}$$

$$\zeta_1 = i\frac{\zeta_6 - i}{\zeta_6 + i}, \qquad \zeta_2 = i\frac{\zeta_6 - 1}{\zeta_6 + 1},$$
$$\zeta_3 = i\frac{\zeta_6 + i}{\zeta_6 - i}, \qquad \zeta_4 = i\frac{\zeta_6 + 1}{\zeta_6 - 1}, \tag{15c}$$

$$\zeta_6 = \frac{1}{\zeta_5}, \qquad \zeta_3 = -\frac{1}{\zeta_1},$$
$$\zeta_4 = -\frac{1}{\zeta_2}, \tag{15d}$$

where Eqs. (15a) relate neighboring equatorial patches, and Eqs. (15b) and (15c) supply the coordinate transformations between equatorial patches and the north and south patches, respectively. Equations (15d), which relate diametrically opposed patches on the sphere, are not strictly necessary for a numerical implementation and can be deduced from (15b) and (15c). Our convention for the gnomic parametrization follows Ref. [22], and it has the advantage that the orientation of the (u, v) axes on each patch is chosen so as to reduce the amount of bookkeeping needed to transfer information between patches. From Eq. (15), adjacent patches with coordinates ζ_i and ζ_j are related by $A\zeta_i = (B\zeta_j - 1)/(B\zeta_j + 1)$, where $A = 1, \pm i$ and $B = 1, \pm i$, while opposite patches are related by $\zeta_i = C/\zeta_j$, with $C = \pm 1$; in particular, we recover the coordinate transformation between "north" ($\zeta_N = \zeta_5$) and "south" ($\zeta_S = \zeta_6$) patches, $\zeta_N = 1/\zeta_S$, as in Ref. [1].

Given the dyad $q^A = P(1, i)$, $P(\zeta, \bar{\zeta}) = 1 + \zeta\bar{\zeta}$, and expressing $q = q^A\partial_A \equiv P(\zeta, \bar{\zeta})\partial_{\bar{\zeta}}$ in two coordinate systems $x^A = (\zeta, \bar{\zeta})$ and $x^{A'} = (\zeta', \bar{\zeta}')$, it can be seen that

$$q = W(\zeta, \bar{\zeta})q',$$

where

$$W(\zeta, \bar{\zeta}) = \frac{P(\zeta, \bar{\zeta})}{P(\zeta', \bar{\zeta}')}\frac{\partial\bar{\zeta}'}{\partial\bar{\zeta}}, \tag{16}$$

and where the substitution $\zeta' = \zeta'(\zeta)$ is understood in the right-hand side of (16). From this we deduce the transformation rule for spin-s functions on the sphere, given here only for the case of adjacent patches,

$$\Psi_1 = \left(\frac{\zeta_1 - 1}{\bar{\zeta}_1 - 1}\right)^s\Psi_4, \qquad \Psi_2 = \left(\frac{\zeta_2 - 1}{\bar{\zeta}_2 - 1}\right)^s\Psi_1,$$
$$\Psi_3 = \left(\frac{\zeta_3 - 1}{\bar{\zeta}_3 - 1}\right)^s\Psi_2, \qquad \Psi_4 = \left(\frac{\zeta_4 - 1}{\bar{\zeta}_4 - 1}\right)^s\Psi_3,$$
$$\Psi_1 = \left(-\frac{\zeta_1 + i}{\bar{\zeta}_1 - i}\right)^s\Psi_5, \qquad \Psi_2 = \left(i\frac{\zeta_2 + i}{\bar{\zeta}_2 - i}\right)^s\Psi_5,$$
$$\Psi_3 = \left(\frac{\zeta_3 + i}{\bar{\zeta}_3 - i}\right)^s\Psi_5, \qquad \Psi_4 = \left(-i\frac{\zeta_4 + i}{\bar{\zeta}_4 - i}\right)^s\Psi_5, \tag{17}$$
$$\Psi_1 = \left(-\frac{\zeta_1 - i}{\bar{\zeta}_1 + i}\right)^s\Psi_6, \qquad \Psi_2 = \left(-i\frac{\zeta_2 - i}{\bar{\zeta}_2 + i}\right)^s\Psi_6,$$
$$\Psi_3 = \left(\frac{\zeta_3 - i}{\bar{\zeta}_3 + i}\right)^s\Psi_6, \qquad \Psi_4 = \left(i\frac{\zeta_4 - i}{\bar{\zeta}_4 + i}\right)^s\Psi_6.$$

For the case of functions with spin-weight zero the transformations reduce to $\Psi_j(\zeta_j, \bar{\zeta}_j) = \Psi_i(\zeta_i, \bar{\zeta}_i)$.

IV. NUMERICAL IMPLEMENTATION

We give here a summary of the numerical techniques used so far in the LEO framework. It is worth noting that the framework is easily extensible, and thus we are not restricted, for instance, to the particular choice of radial grid made here, nor to the choice of radial or time integration schemes used in the present work. In the subsection on finite-difference operators on the sphere, for instance, we describe higher-order extensions that we have elected not to use in the example application considered here, as they are inconsistent with the radial and time integration schemes, which we have taken unchanged from [3].

A. Radial grid and finite-difference operators

Following [33], we take the computational radial grid to be equally spaced in the compactified coordinate $x = r/(R + r)$, restricted to the range $x_B \le x \le 1$, i.e. $x_k = x_B + (k-1)\Delta x$, $k = 1, \ldots, N_x$, $\Delta x = (1 - x_B)/(N_x - 1)$, with $x_B = r_B/(R + r_B)$. We express radial derivatives in terms of the compactified grid x_i, via the relation $\partial x/\partial r = (1-x)^2/R$, e.g.

$$f_{,r}|_{k+(1/2)} = \frac{(1 - x_{k+(1/2)})^2}{R}\frac{(f_{k+1} - f_k)}{\Delta x}, \tag{18}$$

$$f_{,r}|_k = \frac{(1-x_k)^2}{R}\,\frac{(f_{k+1}-f_{k-1})}{2\Delta x}. \tag{19}$$

B. Centered finite-difference operators on the sphere

We construct an equally spaced grid on the gnomic coordinates $x^A = (\alpha, \beta)$, with $\alpha_i = -\pi/4 + (i - \frac{1}{2})\Delta$, $\beta_j = -\pi/4 + (j - \frac{1}{2})\Delta$, and $\Delta = \pi/(2N_\xi)$, $i, j = 1, \ldots, N_\xi$. The useful part, exclusive of ghost zones, for each of the coordinates ranges from $-\pi/4 + \Delta/2$ to $\pi/4 - \Delta/2$. With this arrangement, the points with $|u| = \pi/4$ or $|v| = \pi/4$ are excluded, and thus we avoid storing double values for the points at the edges of each patch, and triple values for the points on the corners where three patches meet. Adding N_g ghost zones on each side of the grid allows us to evaluate derivatives to order $N = 2N_g$ with centered stencils of the form

$$\frac{\partial f}{\partial \alpha}|_{i,j} = \frac{1}{\Delta}\sum_{k=1}^{N/2} c_k (f_{i+k,j} - f_{i-k,j}). \tag{20}$$

The coefficients for the derivatives, up to eighth order, are given in Table I. We can verify that the coefficients of Table I for each of the derivatives are correct by noting that the numerical error of derivatives of order N is within the level of round-off when applied to a polynomial test function F of order N or lower. We have also verified the proper convergence rate of the numerical $\eth$ and $\bar{\eth}$ operators when applied to spin-weighted spherical harmonics [9]. Figure 2 shows the proper convergence rates of the $\eth$ operators constructed from derivatives of second, fourth, sixth, and eighth order when applied to the spin-2 spherical harmonic ${}_2Y_{43}$ on grid sizes ranging from $N_\zeta = 16$ to $N_\zeta = 128$.

TABLE I. Coefficients for centered angular derivatives.

N_g	c_1	c_2	c_3	c_4
1	1/2			
2	8/12	−1/12		
3	3/4	−3/20	1/60	
4	4/5	−1/5	4/105	−1/280

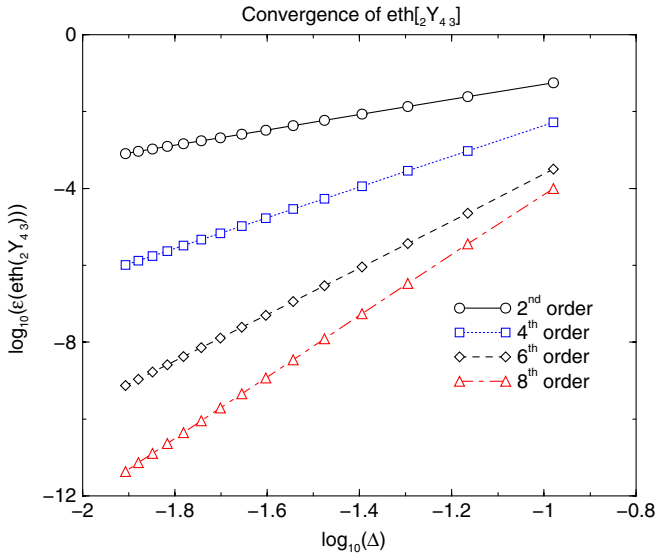

FIG. 2 (color online). Convergence rate of the $\eth$ operator, built upon angular derivatives of order 2, 4, 6, and 8 (indicated in the graph as circles, squares, diamonds, and triangles, respectively), acting on ${}_2Y_{43}$, and with grid sizes ranging from $N_\zeta = 16$ to $N_\zeta = 128$.

C. One-dimensional interpolation of ghost zones

Since the ghost zones required to evaluate derivatives fall on coordinate lines parallel to the boundary, we can obtain function values at these ghost zones with one-dimensional interpolations. We use standard Lagrangian interpolation formulas to Nth-order accuracy,

$$f(x) = \sum_{i=1}^{N} f_i \prod_{j\neq i} \frac{(x - x_j)}{(x_i - x_j)}, \tag{21}$$

adapted to equally spaced grids, i.e. $x_j = x_0 + j\Delta$. Figure 3 shows the calibration of the interpolation routines with a test function consisting of a polynomial of order 15, i.e.

$$P_N(\alpha) = \sum_{i=0}^{N} c_i \alpha^i, \qquad -\frac{\pi}{4} \le \alpha \le \frac{\pi}{4} \tag{22}$$

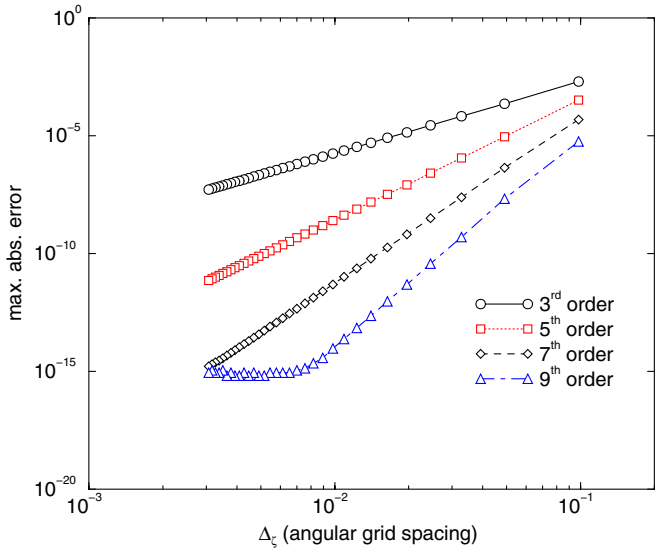

FIG. 3 (color online). Convergence rate of the various interpolation schemes used. Shown in the graph are the third-order (circles), fifth-order (squares), seventh-order (diamonds), and ninth-order (triangles) interpolators. For the highest order interpolator used (ninth order), the error goes down to double-precision round-off level ($\sim 10^{-16}$) when more than $N_\zeta = 100$ angular points per patch are used.

for $N = 15$, where the coefficients $c_i, i = 1 \ldots N$ are chosen randomly, subject to the condition $|c_i| \leq 1$. The interpolants display convergence to the correct order (third, fifth, seventh, and ninth order, respectively), for grid sizes in the range $8 \leq N_\zeta \leq 256$. For this range of values ($8 \leq N_\zeta \leq 256$), there are from 32 to 1024 points in the great circles determined by the intersection of the sphere with any of the Cartesian coordinate planes. As expected, for smooth data such as our test function, for sufficiently large grid sizes, the error goes down to round-off level when using the higher-order schemes. This saturation effect is already visible in the plot for the ninth-order interpolator when angular grid sizes reach approximately $N_\zeta = 100$. As indicated in Sec. III, we have chosen the range of the spherical coordinates (α, β) so that there are no overlapping points at the edge of each patch. This avoids the awkward procedure of averaging values from different patches to obtain a single-valued function throughout the computational grid.

The number of ghost zones, the order of the finite-difference approximations, and the order of interpolation, while related, are not directly tied to each other. One requirement is that we must have enough ghost zones (N_g) to compute the finite-difference approximation to the desired order, N_F; and since, in general, we want to use centered differences, the relation $N_F \leq 2N_g$ must hold. If we wish to maintain the symmetry of the interpolation stencils, $N_I \leq 2N_g + 1$ must also hold. For the cases we have considered, we find that our algorithms are stable if $N_I \geq N_F + 1$, with the inequality required only in the case of $N_F = 2$, the lowest order of finite differences that we considered. We are otherwise free to vary the number of ghost zones as dictated by efficiency considerations.

D. Integrals over the sphere and volume integrals

Integrals over the sphere and volume integrals arise naturally, in particular, when computing norms of various quantities. We evaluate integrals on the sphere to second-order accuracy by evaluating the area element in gnomic coordinates,

$$d\Omega = \frac{(1 + U^2)(1 + V^2)}{(1 + U^2 + V^2)^{3/2}} \Delta\alpha\Delta\beta, \tag{23}$$

evaluating the function value on grid cell centers, $f_{i,j} = f(U_i, V_j)$, and summing over grid cells,

$$\int_S f d\Omega = \sum_{i=1}^{N_\zeta} \sum_{j=1}^{N_\zeta} f_{i,j} \frac{(1 + U_i^2)(1 + V_j^2)}{(1 + U_i^2 + V_j^2)^{3/2}} \Delta^2, \tag{24}$$

where Δ stands for the grid spacing on both coordinates (α, β), which we have taken to be the same. Note that, since the spherical patches do not overlap, the integral over the sphere is just the sum of the integrals over the individual patches. Figure 4 shows the convergence of the integral of the area element itself to the correct answer of $\int d\Omega = 4\pi$ for grid sizes in the range of $N_\zeta = 8$ to $N_\zeta = 512$. The measured convergence rate is 2.0, in full agreement with the expected result.

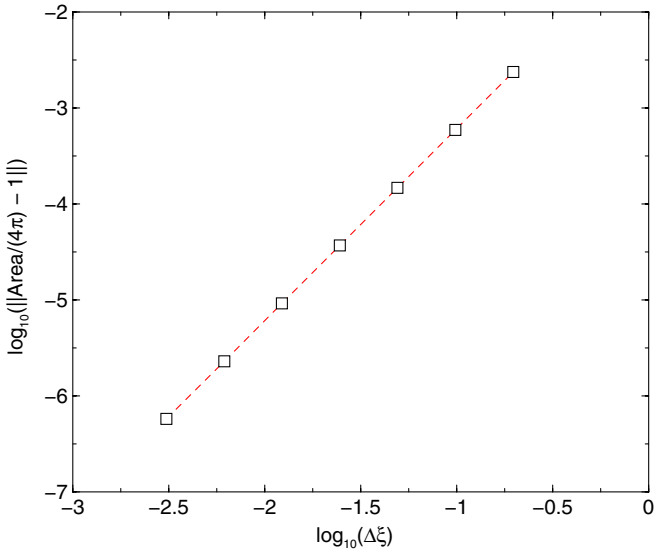

FIG. 4 (color online). Convergence rate of the integral of the area element over the sphere, for grid sizes ranging from $N_\zeta = 8$ to $N_\zeta = 512$. The markers indicate the error of the area element at the corresponding resolution; the line is the least-squares fit, yielding a convergence rate of 2.0.

Volume integrals are computed similarly to integrals over the sphere, but here we evaluate the spherical contributions midpoint in between radial points, i.e.

$$\begin{aligned}\int_S d\Omega \int_{r_m}^{r_n} f r^2 dr = &\sum_{i=1}^{N_\zeta} \sum_{j=1}^{N_\zeta} \frac{(1 + U_i^2)(1 + V_j^2)}{(1 + U_i^2 + V_j^2)^{3/2}} \Delta^2 \\ &\times \sum_{k=m}^{n-1} \frac{x_{k+(1/2)}^2}{(1 - x_{k+(1/2)})^4} \\ &\times \frac{(f_{i,j,k} + f_{i,j,k+1})}{2} \Delta x.\end{aligned} \tag{25}$$

We replace the flat volume element, $dV = r^2 dr d\Omega$, with the volume element corresponding to a Bondi metric, $dV = r^2 e^{2\beta} dr d\Omega$, when appropriate. To speed up the evaluation of integrals, we precompute the area element on the sphere, Eq. (23).

E. Accuracy of the spin-weighted spherical harmonic decomposition

We make use of spin-weighted spherical harmonics ${}_sY_{lm}$ throughout this paper, following the convention of [9]. In order to estimate the error introduced when we perform a spin-weighted spherical harmonic decomposition, we look at how well the orthonormality condition

$$\int_S {}_s Y_{lms} \bar{Y}_{lm} d\Omega = \delta_{l,l'} \delta_{m,m'} \tag{26}$$

is preserved (at the numerical level) for spherical harmonics with spin weight $s = 0$, 1 and 2, for a range of values of ℓ and m and angular grid sizes. As expected, the numerical value of the integral converges to the analytic result to second order on the grid spacing, since we have chosen to use a second-order integration algorithm. Figure 5 illustrates one instance, where we have taken $s = 0$, $\ell = 6$, with $m = -6 \ldots 6$, and varied the angular grid size from $N_\zeta = 32$ to $N_\zeta = 64$. We can also place an estimate on the accuracy of the projection of a spin-weight s function,

$$c_{lm}[F] = \int_S F_s \bar{Y}_{lm} d\Omega, \tag{27}$$

based on the magnitude of the off-diagonal values in (26) for a given grid size. When projecting the test functions $Y_{l'm'}$ into the spherical harmonics Y_{lm} for $l = 0 \ldots l_{\max}$, $m = -l \ldots l$, at the analytic level we would expect to obtain zero for all coefficients, except for $c_{l'm'}$, when it would evaluate to 1. We find that grid sizes of $N_\zeta = 64$ and larger are sufficient to keep the error in the coefficients to within one part in 10^4, which again is consistent with our integration scheme being second order in the angular discretization. Figure 6 shows the error in the coefficients computed for $Y_{l'm'}$, $l' = 6$, $m' = 3$ on a grid with $N_\zeta = 64$ points. We have omitted from the graph those coefficients for which the error is already at the level of round-off.

The preceding description of the numerical implementation is complete but for one key aspect, namely, our

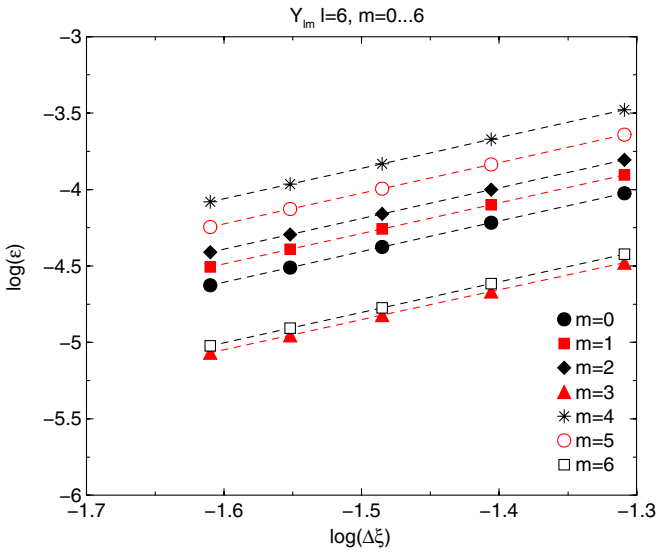

FIG. 5 (color online). Convergence of the orthonormality condition, illustrated here by computing the convergence rate of $\int_S {}_0 Y_{lm0} \bar{Y}_{lm} \equiv 1$, for the case $l = 6$, $m = 0 \ldots 6$, on grid sizes ranging from $N_\zeta = 32$ to $N_\zeta = 64$.

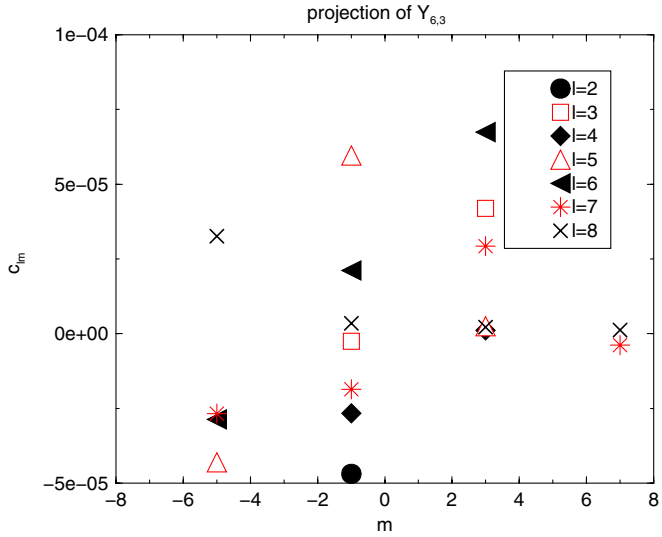

FIG. 6 (color online). Error in the coefficients c_{lm} when the function being projected is the spherical harmonic Y_{63}, on a grid of $N_\zeta = 64$ points. Coefficients whose errors are at round-off level are not shown.

parallelization strategy. In our framework, the six cubed-sphere grid patches are decomposed into *computational* subpatches, each with the same number of points on the angular directions, for efficiency reasons. These subpatches are distributed among processors, and the ghost-zone values required for the computation of angular derivatives are communicated by the use of message-passing calls [34]. The radial direction is not distributed, as the characteristic algorithm requires a radial march for the integration of the hypersurface equations as well as the evolution equations [33]. The computational complexity of a parallel implementation via message passing lies in that, knowing the location of its assigned grid subpatch on the global grid, each processor must determine which processors are its nearest neighbors, i.e. to which processes it must supply ghost-zone information (and also receive that information from). Due to the relative orientation of the cubed-sphere patches, we need to know whether the order in which the ghost zones are traversed must be reversed for subpatches on the edge of a cubed-sphere patch. Since the subpatch to processor mapping remains constant during a simulation, the relevant information needs to be computed only once, and at any rate, it incurs no measurable overhead in the computation involved in a simulation. An efficient, scalable implementation of the message passing itself requires only a small subset of the full message passing interface (MPI) functionality: a few calls to set up the appropriate groups of processors; sends, receives, and waits (for ghost-zone communication); some additional reduction operations (to accumulate integrated values); and some broadcasts (to propagate parameters). Exclusive of file access operations, only 14 MPI functions in all are invoked.

Having established that all the key computational aspects of the framework are in place, and have been correctly implemented, we proceed next to consider specific applications of the framework to systems of physical interest.

V. A THREE-DIMENSIONAL MASSLESS SCALAR FIELD SCATTERED OFF A SCHWARZSCHILD BLACK HOLE

We use the numerical formalism developed in the preceding sections to solve numerically a model problem consisting of a self-gravitating massless scalar field in three dimensions. Our starting points are Ref. [8] for a description of the vacuum problem, and Ref. [11] for the coupling of the scalar field to the gravitational metric fields. We use coordinates based upon a family of outgoing null hypersurfaces, and we let u label these hypersurfaces, x^A ($A = 2, 3$) label the null rays, and r be a surface area coordinate. In the resulting $x^\alpha = (u, r, x^A)$ coordinates, the metric takes the Bondi-Sachs form [35,36]

$$ds^2 = -[e^{2\beta}(1 + W/r) - r^2 h_{AB} U^A U^B]du^2 - 2e^{2\beta} du dr - 2r^2 h_{AB} U^B du dx^A + r^2 h_{AB} dx^A dx^B, \tag{28}$$

where W is related to the more usual Bondi-Sachs variable V by $V = r + W$, and where $h^{AB} h_{BC} = \delta^A_C$ and $\det(h_{AB}) = \det(q_{AB})$, with q_{AB} a unit sphere metric, given in terms of a complex dyad q_A satisfying $q^A q_A = 0$, $q^A \bar{q}_A = 2$, $q^A = q^{AB} q_B$, with $q^{AB} q_{BC} = \delta^A_C$ and $q_{AB} = \frac{1}{2}(q_A \bar{q}_B + \bar{q}_A q_B)$. We also use the intermediate variable $Q_A = r^2 e^{-2\beta} h_{AB} U^B_{,r}$. We represent tensors on the sphere by spin-weighted variables [1]. The conformal metric h_{AB} is represented by the complex function $J = h_{AB} q^A q^B/2$, and by the real function $K = h_{AB} q^A \bar{q}^B/2$, where $K^2 = 1 + J\bar{J}$. The metric functions U^A are similarly encoded in the complex function $U = U^A q_A$. Thus, it is necessary to introduce the intermediate spin-weighted variable $Q = Q_A q^A$, as well as the (complex differential) operators $\eth$ and $\bar{\eth}$ (see [1] for full details).

Treating the Einstein-Klein-Gordon model problem consistently within the LEO framework requires some modifications to [11], specifically to the wave equation for the scalar field ($\Box\phi = 0$) which is given by Eqs. (21)–(27) of [11]. We substitute all second-order angular derivatives of the metric fields in terms of $\eth$ and $\bar{\eth}$ operators acting on the additional fields $\nu = \bar{\eth} J$, $k = \eth K$, and $B = \eth\beta$ introduced in Ref. [8], whenever possible. A consistent treatment is obtained by introducing the additional variable

$$\psi = \eth\chi, \tag{29}$$

where $\chi = r\phi$, so that the scalar field equation is also in first-order differential form in the angular variables, on a par with the approach of [8] for the metric equations. The Bondi-Sachs hierarchy of hypersurface equations,

$$\nu_{,r} = \bar{\eth} J_{,r}, \tag{30}$$

$$k_{,r} = \eth K_{,r}, \tag{31}$$

$$\beta_{,r} = \frac{r}{8}(J_{,r}\bar{J}_{,r} - K^2_{,r}) + 2\pi r(\phi_{,r})^2, \tag{32}$$

$$B_{,r} = \eth\beta_{,r}, \tag{33}$$

$$\begin{aligned}(r^2 Q)_{,r} = r^2[&-K(k_{,r} + \nu_{,r}) + \bar{\nu} J_{,r} + \bar{J}\eth J_{,r} + \nu K_{,r} + J\bar{k}_{,r} \\ &- J_{,r}\bar{k}] + \frac{r^2}{2K^2}[\bar{\nu}(J_{,r} - J^2\bar{J}_{,r}) + \eth J(\bar{J}_{,r} - \bar{J}^2 J_{,r})] \\ &+ 2r^2 B_{,r} - 4rB + 16\pi r\phi_{,r}\psi,\end{aligned} \tag{34}$$

$$r^2 U_{,r} = e^{2\beta}(KQ - J\bar{Q}), \tag{35}$$

$$\begin{aligned}(r^2\tilde{W})_{,r} = \Re\Big\{&e^{2\beta}\Big(\frac{\mathcal{R}}{2} - K(\bar{\eth}B + B\bar{B}) + \bar{J}(\eth B + B^2) \\ &+ (\nu - k)\bar{B}\Big) - 1 + 2r\bar{\eth}U + \frac{r^2}{2}\bar{\eth}U_{,r} \\ &- e^{-2\beta}\frac{r^4}{4}\bar{U}_{,r}(KU_{,r} + J\bar{U}_{,r})\Big\} \\ &- 2\pi\frac{e^{2\beta}}{r^2}[2K\bar{\psi}\psi - J\bar{\psi}^2 - \bar{J}\psi^2],\end{aligned} \tag{36}$$

$$\psi_{,r} = \eth\chi_{,r}, \tag{37}$$

now includes an additional consistency condition, Eq. (37), and the equations for β, Q, and $\tilde{W} = W/r^2$ are modified to include the source terms as shown above. The evolution equation for the metric field J is given by

$$\begin{aligned}2(rJ)_{,ur} - (r^{-1}V(rJ)_{,r})_{,r} = &-K(r\eth U_{,r} + 2\eth U) \\ &+ \frac{2}{r}e^{2\beta}(\eth B + B^2) \\ &- (r\tilde{W}_{,r} + \tilde{W})J + J_H + JP_u \\ &+ \frac{8\pi}{r^3}e^{2\beta}\psi^2,\end{aligned} \tag{38}$$

with the quantities $\mathcal{R}$, J_H, and P_u as in Eqs. (24)–(26) of [8]. The scalar field evolution equation follows from Eq. (21) of [11],

$$2\chi_{,ur} - \left(\frac{V}{r}\chi_{,r}\right)_{,r} = -\left(\frac{W}{r}\right)_{,r}\frac{\chi}{r} + N_\phi. \tag{39}$$

The source term N_ϕ is

$$N_\phi = \frac{e^{2\beta}}{r}\Big[-\frac{1}{2r}(\bar{J}\eth\psi + J\bar{\eth}\,\bar{\psi}) + \frac{K}{r}\bar{\eth}\psi + \Big(K\bar{B} - \bar{J}B - \frac{1}{2}(K\bar{Q} - \bar{J}Q) - \frac{\bar{\nu}}{2} + \frac{1}{4K}(\bar{J}\nu + J\bar{\mu})\Big)\frac{\psi}{r} + \Big(KB - J\bar{B} - \frac{1}{2}(KQ - J\bar{Q}) - \frac{\nu}{2} + \frac{1}{4K}(J\bar{\nu} + \bar{J}\mu)\Big)\frac{\bar{\psi}}{r}\Big] - \frac{1}{r}(U\bar{\psi} + \bar{U}\psi) - \frac{r}{2}\phi_{,r}(\bar{\eth}U + \eth\bar{U}) - [U(\bar{\psi}_{,r} - \bar{\psi}) + \bar{U}(\psi_{,r} - \psi)]. \tag{40}$$

Following [13], we have used the shorthand $\mu = \eth J$, and eliminated the radial derivatives $U_{,r}$ and $\bar{U}_{,r}$ using Eq. (35),

$$Q = r^2 e^{-2\beta}(KU_{,r} + J\bar{U}_{,r}). \tag{41}$$

The data required on the initial null cone are the evolution variables J and ϕ. Given boundary values at a fixed value of r, the remaining variables (ν, k, β, B, Q, U, and $\tilde{W}$) can be determined on the initial null cone by explicit integration of the hypersurface equations (see [11] for details). The evolution equations (38) and (39) can then be used to find J and ϕ on the next null cone, and the process repeated to determine the spacetime to the future of the initial slice.

A. Scalar field on a fixed background

The above system of equations describes a self-gravitating scalar field. In the limit of small amplitudes, $|\phi| \ll 1$, the scalar field can be treated as a perturbation propagating on a fixed background. This considerably simpler model is contained in the fully nonlinear case, and is implemented in our code by integrating only Eqs. (37) and (39). For a Schwarzschild background, the metric fields J, β, U, ν, k, and B are zero, and $V = r - 2M$. The source term in Eq. (40) reduces to $N_\phi = \bar{\eth}\psi/r$, and we are left with the system

$$2\chi_{,ur} - \Big(\Big(1 - \frac{2M}{r}\Big)\chi_{,r}\Big)_{,r} = -\frac{2M\chi}{r^3} + \frac{\bar{\eth}\psi}{r}, \qquad \psi_{,r} = \eth\chi_{,r}. \tag{42}$$

For the simulations we discuss in the present work, we will be interested in solutions of the scalar field on a fixed background with definite angular dependence, as discussed in the next subsection.

B. Quasinormal modes in a Schwarzschild background

The linear equation for the scalar field on a fixed background, Eq. (42), is separable; i.e. its solutions can be written in the form

$$\phi(u, r, x^A) = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell}\chi_{\ell m}(u, r)\frac{Y_{\ell m}(x^A)}{r}, \tag{43}$$

with the x^A coordinates in the sphere, and where each of the $\chi_{\ell m}$ satisfies the one-dimensional wave equation in the plane (u, r),

$$2\chi_{,ur} - \Big(\Big(1 - \frac{2M}{r}\Big)\chi_{,r}\Big)_{,r} = -\Big(\frac{2M}{r^3} + \frac{\ell(\ell+1)}{r^2}\Big)\chi, \tag{44}$$

where we have used the property $\eth\bar{\eth}\chi = -\ell(\ell+1)\chi$ [37]. Equation (44) is the usual equation governing the scalar perturbations of a Schwarzschild black hole [38], written here in characteristic coordinates (u, r, x^A). It can be put in a more familiar form by writing it in the coordinates (t, r_*), with $u = t - r_*$, and where r_* is the usual "tortoise" coordinate, $r_* = r + 2M\ln(r/2M - 1)$.

$$\chi_{,tt} - \chi_{,r_*r_*} + \hat{V}(r)\chi = 0, \tag{45}$$

where $\chi = r\phi$ and the potential $\hat{V}(r)$ is given by

$$\hat{V}(r) = \Big(1 - \frac{2M}{r}\Big)\Big[\frac{2M}{r^3} + \frac{\ell(\ell+1)}{r^2}\Big]\chi, \tag{46}$$

and we have denoted it by $\hat{V}$ to avoid confusion with Bondi's V which we use throughout this paper. Equation (45) has been studied extensively [38–40], its most salient feature being the existence of quasinormal modes, whose frequencies have been tabulated; see for example [40]. Note that the right-hand side of Eq. (44) is the correct form of the potential in (u, r) coordinates. It differs by a factor of $(1 - 2M/r)$ from the potential as given in Eq. (45) (see [38]), because that factor is precisely the Jacobian of the coordinate transformation, $\partial_r/\partial_{r_*} = 1 - 2M/r$.

In the remainder of the present work we will use both the quasinormal mode equation, Eq. (44), and the linear system, Eq. (42), as tests of the validity of our numerical implementation. We do this in an incremental fashion, solving Eq. (44) for fixed values of ℓ, and comparing the effectiveness of the numerical integration scheme and of our boundary conditions in reproducing the quasinormal modes. To this end, we implement a purely radial code for Eq. (44) that employs the same numerical integration scheme that is used in the "linear" code [which solves Eqs. (42) and (44)], and in the full nonlinear code. Using this radial code, we can isolate the effects arising from the inner-boundary treatment at $r = 2M$ by implementing Eq. (44) as indicated, in outgoing null coordinates, using both a noncompactified coordinate r, with a simple extrapolative boundary condition at the outer boundary $r_{\rm out} > 2M$, and the compactified coordinate $x = r/(r + R)$, where the outer boundary lies at future null infinity. The use of a noncompactified coordinate allows us to isolate any effects that may arise due to the nonuniform coordinate

velocity introduced by the compactified coordinate x. Conversely, simulations using the compactified coordinate avoid the effects of placing the outer boundary at a finite distance.

We also implement the equivalent of Eq. (44) in *ingoing* null coordinates, (v, r), with $v = t + r_*$, namely,

$$2\chi_{,vr} + \left[\left(1 - \frac{2M}{r}\right)\chi_{,r}\right]_{,r} = \left[\frac{2M}{r^3} + \frac{\ell(\ell+1)}{r^2}\right]\chi. \quad (47)$$

In ingoing null coordinates, the slices at $v = \text{const}$ penetrate the event horizon $r = 2M$, effectively providing for an excision scheme, where evolution can be stopped at a finite number of points inside the boundary, because the behavior of the field inside the horizon does not affect the solution outside. Evolutions in ingoing coordinates are carried out on a noncompactified radial grid, for which boundary data are required at a fixed value of $r_{\text{out}} > 2M$. Because of the presence of this outer boundary, simulations in ingoing coordinates can only be run for a limited time, typically $u \sim 2r_{\text{out}}$, before outer boundary effects influence the signal extracted. A similar effect is seen when using outgoing, noncompactified null coordinates. When using compactified coordinates, no such effects are seen, as expected. A detailed comparison between ingoing and outgoing versions of characteristic systems of equations, in compactified as well as noncompactified coordinates, along with their relative advantages and disadvantages for specific applications, is worthwhile but lies outside the scope of the present work and will be reported elsewhere. We will refer only briefly to these issues in the remainder of this work.

C. Energy carried out by the scalar field

As a useful physical indicator we calculate the balance of the scalar field energy contained between the inner boundary and null infinity. The expressions we give here are valid in the linear case, where the background metric is the Schwarzschild metric. For a more general approach to this issue, the linkage integrals have to be calculated; specifically the asymptotic Killing vector field must be parallelly propagated from null infinity [41].

Restricted to the background case, then, given a Killing vector field ξ^ν of the metric $g_{\mu\nu}$, $\mathcal{L}_\xi g_{\mu\nu} = 0$, we can define the conserved quantity

$$\mathcal{C} = \int T^\mu_\nu \xi^\nu d\Sigma_\mu. \quad (48)$$

In particular, selecting the timelike Killing vector $\xi^\nu = \delta^\nu{}_u$, and for a surface of constant u, $\mathcal{C}$ is the energy contained on the surface,

$$E(u) = \int T^u_u dV, \quad (49)$$

where dV is the volume element of the surface at constant u. For a sphere at constant r, $\mathcal{C}$ represents the energy flux across the surface,

$$P(u) = \int T^r{}_u r^2 d\Omega, \quad (50)$$

with $d\Omega$ the solid angle element. The relevant components of the stress-energy tensor for a massless scalar field are

$$T^u{}_u = e^{-2\beta}\frac{V}{2r}(\phi_{,r})^2 + \frac{K}{2r^2}\eth\phi\bar{\eth}\phi - \frac{1}{4r^2}[\bar{J}(\eth\phi)^2 + J(\bar{\eth}\phi)^2] - \frac{1}{2}e^{-2\beta}\phi_{,r}(\bar{U}\eth\phi + U\bar{\eth}\phi), \quad (51)$$

$$T^r{}_u = e^{-2\beta}\phi_{,u}\left[\phi_{,u} - \frac{V}{r}\phi_{,r} + \frac{1}{2}(\bar{U}\eth\phi + U\bar{\eth}\phi)\right]. \quad (52)$$

In the case of a linear scalar perturbation on a Schwarzschild background, the energy content of a hypersurface at constant u is given by

$$E(u) = \frac{1}{2}\int\left[\left(1 - \frac{2M}{r}\right)(r\phi_{,r})^2 + \eth\phi\bar{\eth}\phi\right]dr d\Omega. \quad (53)$$

The power radiated at time u across a surface of constant r, such as the inner boundary, which in our simulations we place close enough to the Schwarzschild black hole, is

$$P_{in}(u) = \int \phi_{,u}\left[\phi_{,u} - \left(1 - \frac{2M}{r}\right)\phi_{,r}\right]r^2 d\Omega. \quad (54)$$

For the flux across the inner boundary, the integral as well as the spatial and time derivatives are to be taken as evaluated at $r = r_{\text{in}}$. Analogously, the power radiated at time u at null infinity is the limiting form (as $r \to \infty$) of the above expression, i.e.

$$P_{\text{out}}(u) = \int (r\phi_{,u})^2 d\Omega, \quad (55)$$

where we have used the behavior of the scalar field near null infinity $\mathcal{I}$ to simplify the expression. With these definitions, the following global energy conservation law holds,

$$\Sigma(u) = E(u) + \int_{u_0}^{u}[P_{\text{out}}(u') - P_{\text{in}}(u')]du' \equiv \text{const}. \quad (56)$$

Even though the expressions given above hold *only* in the limit in which ∂_t is a Killing vector of the metric, we expect them to hold in an approximate sense for our nonlinear evolutions, so we use them as a criterion for code testing.

As stated previously, we use the radial code to calibrate the fully nonlinear, three-dimensional LEO code in the linear regime. When computing the energy in the radial code, we make use of the property

$$\int \eth\phi\bar{\eth}\phi d\Omega = -\int \phi\eth\bar{\eth}\phi d\Omega \quad (57)$$

(see [37]). Since the data we pose are pure spherical harmonics, the integral in the right-hand side is proportional to the norm $\int \phi\bar{\phi}d\Omega$. Equation (57) allows us then to properly account for the contribution of the angular derivatives of the field to the energy (49) when using only the radial code.

VI. ADDITIONAL NUMERICAL CONSIDERATIONS

A. Hypersurface equations

The integration of the hypersurface equations does not present any inherent difficulty, as they are discretized at midpoint between grid points as per [8,11]. An important issue which arises because of the parallel implementation of our algorithm is that, after each step in the radial march, that is, after each hypersurface equation has been advanced radially one grid point, we must synchronize the variable which has just been integrated. By this we mean that we communicate the ghost-zone values to the processors carrying out the integration in neighboring patches. Since communication is an expensive operation even on the most tightly coupled parallel computers, we take the approach of explicitly synchronizing a variable only if an $\eth$ (or $\bar{\eth}$) operator will be applied to the variable in question. An alternative approach would be to incorporate the synchronization into the $\eth$ (and $\bar{\eth}$) operators. The first approach requires more bookkeeping on our part, whereas the second is more straightforward. Because of the number of $\eth$ (or $\bar{\eth}$) operations that appear in the full nonlinear equations, however, the performance difference between these two approaches is significant. For this reason we take the first approach, minimizing communication costs, with a substantial increase in performance.

B. Evolution equations

The evolution equation (38) for J is treated as reported in [11], except that the first two radial points are subject to the boundary condition explained below. The evolution equation for the scalar field is recast in terms of the two-dimensional wave operator

$$\Box^{(2)}\chi = e^{-2\beta}[2\chi_{,ru} - (r^{-1}V\chi_{,r})_{,r}], \tag{58}$$

where $\chi = r\phi$ and Eq. (39) reduces then to

$$e^{2\beta}\Box^{(2)}\chi = \mathcal{H}, \tag{59}$$

where

$$\mathcal{H} = -(W/r)_{,r}\chi/r + N_{\phi}. \tag{60}$$

Since all two-dimensional wave operators are conformally flat, with conformal weight -2, we can apply to (59) a flat-space identity relating the values of χ at the four corners P, Q, R, and S of a null parallelogram $\mathcal{A}$, with sides formed by incoming and outgoing radial characteristics. In terms of χ, this relation leads to an integral form of the evolution equation for the scalar field,

$$\chi_Q = \chi_P + \chi_S - \chi_R + \frac{1}{2}\int_{\mathcal{A}} dudr\mathcal{H}. \tag{61}$$

The corners of the null parallelogram cannot be chosen to lie exactly on radial grid points; thus the values of χ at the vertices of the parallelogram are approximated to second-order accuracy by linear interpolations between nearest neighbor grid points on the same outgoing characteristic. Approximating the integrand by its value at the center C of the parallelogram (evaluated using average values between the points P and S), we then have

$$\chi_Q = \chi_P + \chi_S - \chi_R + \frac{\Delta u}{4}(r_Q - r_P + r_S - r_R)\mathcal{H}_C. \tag{62}$$

The evolution algorithm for the metric function J follows the procedure outlined in [3,8,11]. As with the hypersurface equations, we synchronize the fields ϕ and J; i.e. we communicate the ghost-zone information from each patch to its neighbors, immediately after advancing radially these two fields with their respective evolution equations.

C. Boundary treatment for the evolved fields

For the ingoing formulation, we set the field values $\phi(v, r = r_{\text{out}}) = 0$, and we march inwards until a few points beyond the black hole horizon ($r = 2M$). Since the past light cones tilt outwards once inside the horizon, the values computed just inside the horizon can never affect those points of the grid that lie outside. This scheme then provides an extremely simple and effective form of excision, as discussed in [4,5,42] in the context of characteristic evolution, and in [43] in the context of $3 + 1$ simulations in the Bondi-Sachs gauge.

For the outgoing formulation on a noncompactified radial grid, we use simple extrapolative boundary conditions at the outermost point; i.e. the field χ at the last point is set equal to the value of χ at the point immediately before. This approximation is justified for sufficiently large r, as the field ϕ behaves, to leading order, as $\phi \sim O(1/r)$. Our treatment of the inner boundary is motivated by physical considerations that arise naturally in the study of quasinormal modes. It can be seen from Eq. (46) that when the potential $\hat{V}(r)$ goes to zero, as it does in the limits $r \to 2M$ and $r \to \infty$, the solutions to Eq. (45) are traveling waves, $\chi_- = F_L(t + r_*)$ and $\chi_+ = F_R(t - r_*)$. In the linear approximation then, it is consistent to apply an *open boundary* condition to the scalar field ϕ based on the assumption that, at the inner boundary, the field behaves as a left-traveling wave, $\chi = F_L(t + r_*)$. It follows also that the same condition must be applied to the spin-weight-2 metric field J. In the linear approximation, Eq. (38) reduces to Eq. (45), with the potential $\hat{V}(r)$ corresponding to that of a spin-weight-2 field; see [38]. This open boundary condition is equivalent to stating that the fields $\chi = r\phi$ and rJ

propagate towards the horizon along the incoming characteristics of the two-dimensional wave operator, Eq. (58). In practice, we implement this condition for the first two points of the radial grid, and use the evolution equations for χ and J elsewhere.

In the nonlinear case, the horizon can no longer be assumed to be static; rather, it is dynamically distorted and grows as the scalar field accretes into the black hole. Our boundary condition is applied always to the same set of points, which are subsequently enveloped by the growing horizon; thus any inaccuracy we might have introduced in those first two points cannot have any effect on the exterior spacetime.

Our approach suggests the following iterative method to treat the inner boundary, in a manner which is consistent with the *open boundary* condition: (1) as a first approximation, solve the homogeneous equation (45) for the first two radial points, i.e. assume the evolved fields propagate along incoming characteristics up to the retarded time $u + \Delta u$, and (2) with the values predicted for the fields at time $u + \Delta u$, correct the right-hand side of the full evolution equations.

D. Tests of second-order convergence

The simulations for the tests reported in the remainder of this section are conducted in compactified outgoing (retarded) null coordinates. To verify that the numerical algorithm is globally second-order convergent, we compute the L_2 norm of the relative residuals for three grid sizes, e.g.,

$$\begin{aligned} \mathcal{Q}_{cm} &= \int [\chi_c - \chi_m]^2 dx d\Omega, \\ \mathcal{Q}_{mf} &= \int [\chi_m - \chi_f]^2 dx d\Omega, \end{aligned} \tag{63}$$

where the c, m, and f subscripts denote the field as computed on coarse, medium, and fine grids, respectively. The field is evolved from an initial retarded time $u = 0$, and the integrals (63) are calculated at the same final retarded time u, using the same set of spatial grid points, obtained by appropriately subsampling from the fine and medium grids to the coarse grid. Here we take the angular (and radial) grids to be in a proportion of 1:3:5. Grids in these ratios have a common set of points that align directly, and thus do not require interpolating cell values from the finest to the coarser grids. In this case, given the values $\mathcal{Q}_{cm}$ and $\mathcal{Q}_{mf}$, it can be shown that the order of convergence $O(\Delta^n)$ of the algorithm can be read by solving the following equation for n:

$$(\mathcal{Q}_{cm}/\mathcal{Q}_{mf})^{1/2} = \frac{1 - 1/3^n}{1/3^n - 1/5^n}. \tag{64}$$

For this test we evolve the initial data

$$\chi(0, r, x^A) = \lambda e^{-(r-r_0)^2/\sigma^2} Y_{\ell m}, \tag{65}$$

with $\lambda = 10^{-4}$, whose radial profile is characterized by $r_0 = 3M$, $\sigma = \frac{1}{2}M$, and whose angular dependence is given by $\ell = 4$ and $m = 2$, from $u = 0$ up to $u = 1M$. We perform three simulations, on the angular grid sizes $N_\zeta = 10, 30, 50$ and the corresponding radial grid sizes $N_x = 501, 1501, 2501$, for which we take 152, 456, 760 time steps, respectively. From (64) we find that the measured order of convergence is $n = 2.05$, in excellent agreement with the expected second-order convergence. It should be noted that this procedure tests the Cauchy convergence of the code, providing a basic check of the consistency of the discretization. For low amplitudes (in the perturbative regime), and for a given value of ℓ, the scalar field profiles computed with the fully three-dimensional code match, to within second order, the profiles obtained with a purely radial code which solves Eq. (44), as expected.

We want to stress that the boundary conditions, the initial data, and the marching algorithm for the scalar field used in this numerical test are all the same as those which we have used to calibrate the radial code, the solutions of which we use here in place of an analytic solution. In fact, the convergence rate for the radial code is exactly 2.00 for the radial grid sizes of $N_x = 501, 1501, 2501$, measured at $u = 1M$ with its respective subsampling, as per Eqs. (63) and (64).

For sufficiently low values of (ℓ, m), the angular grid sizes N_ζ used in the convergence test are adequate. For a given angular grid size, it is also possible to reduce the angular error by increasing the order of the angular derivatives, for example, to fourth order or higher. The increased computational expense is offset by the increased accuracy obtained; in a parallel application there is also the potential for additional overhead because more ghost cells must be communicated. In practice, we observe that, for the smallest angular grid size considered ($N_\zeta = 10$), changing the discretization of the angular derivatives from second to fourth order increases the execution time by about 20%.

In the work reported here, since the radial and time integrations are carried out with a scheme that is second-order convergent, we have opted to use second-order-accurate angular derivatives, as with this choice the nonlinear code exhibits second-order-accurate Cauchy convergence.

For the initial data considered here, the radial resolution must be at least $N_x = 501$ to guarantee second-order convergence, as the radial features are the dominant source of numerical error. We note that if we repeat the test above using the same angular grid sizes, $N_\zeta = 10, 30, 50$, but using instead radial grids with fewer points, i.e. $N_x = 251, 751, 1250$, the measured convergence rate is lower, namely, $n = 1.56$.

For the numerical simulations we present in the remainder of this article, we have chosen grid sizes such that the numerical algorithm is always in the second-order conver-

gence regime. For more details on the convergence properties of the radial evolution algorithm, see [44].

VII. NUMERICAL RESULTS

A. Quasinormal modes

The simulations reported in this and subsequent sections are all carried out in compactified, outgoing (retarded) null coordinates, because these coordinates allow us to read off scalar radiation patterns at null infinity. (The treatment of the inner boundary is as described in Sec. VI C.) The quality of the waveforms extracted depends in part on the location of the inner boundary and other factors. We describe here the method used and analyze the sources of error. For the simulations in this section we use a grid with sizes $N_x = 1500$, $N_\zeta = 11$; the initial data correspond to Eq. (65), with $\lambda = 10^{-4}$, $r_0 = 3M$, $\sigma = \frac{1}{2}M$, and $M = 1$.

To extract the quasinormal modes we have used the free software package HARMINV [45], which employs a low-storage filter diagonalization method (FDM) for finding the quasinormal modes in a given frequency interval. This software package is based on the FDM algorithm described in [46,47]. The advantage of using HARMINV is that FDM methods provide better accuracy than what can be obtained with a fast Fourier transform (FFT) [9], and are more robust than the least-squares fit [48]. We find it surprising that this approach, to our knowledge, has not been used in the context of reading quasinormal modes in gravitational simulations.

In performing a fit with HARMINV to the scalar field waveforms, we find it sometimes necessary to factor out, at least approximately, the exponential decay of the signal. This happens when the magnitude of the imaginary part of the frequency (the decay rate) is comparable to the real (oscillatory) part, where the FDM method fails to find a fitting frequency. In those cases, we premultiply the signal by an exponentially increasing function $f = \exp(|\omega_f|t)$, perform the fit with HARMINV, and adjust the frequency obtained accordingly. When an analytic value for the frequency is available, we take its imaginary part as the value for ω_f. In general, when the imaginary part of the frequency is not known, it suffices to use a rough estimate of the decay rate, which can obtained graphically. We also need to decide what range of values of u to use to extract this information. We do this by plotting the signal $\chi(u)$ and noting when the waveform is clearly periodic with an exponentially decaying envelope. For example, in Figs. 7 and 8, one can clearly see that this regime starts at about $u = 20M$. We take the end of the fitting interval when the signal no longer appears to be a damped sinusoidal waveform. For initial data of the form (65), with $\ell = 1$, we use HARMINV to extract the frequency, using $u = [20, 70]$ as the fitting interval. The measured frequency is $\omega = 0.3076\,(5\%) - 0.1064i\,(9\%)$. Here the values in parentheses indicate the percentage deviation from the value calculated in [40] via the WKB method to sixth order. A comparison between the signal computed and the quasinormal mode fit is shown in Figs. 7 and 8. The figures show the profiles computed with the three-dimensional code (solid line). These profiles are indistinguishable, at the resolution of the graph, from the profiles obtained by solving numerically the perturbative equation (44) for the same initial data; thus we have opted not to show the perturbative solution, as is customary. For comparison,

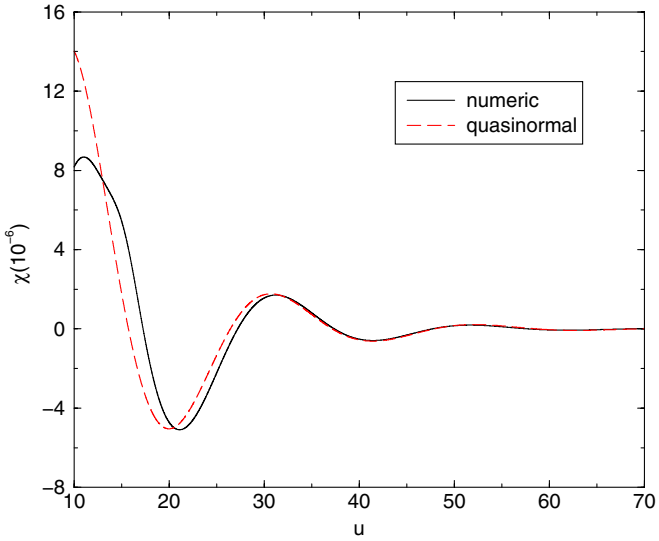

FIG. 7 (color online). The function $\chi(u)$ at $\mathcal{I}$ as a function of Bondi time, showing the quasinormal mode regime oscillations for $\ell = 1$, $m = 0$. The solid line is the output from LEO for $N_\zeta = 11$ and $N_x = 1001$, when the initial data and boundary conditions are given as for the convergence test; the dashed line is the quasinormal mode extracted from the data.

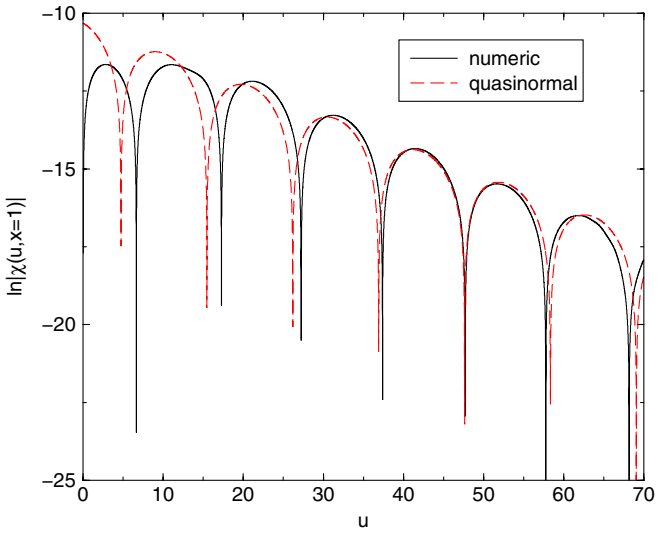

FIG. 8 (color online). Log of the absolute value of the function $\chi(u)$ at $\mathcal{I}$ as a function of Bondi time. Parameters and conditions are the same as in Fig. 7. The solid line is the output from LEO; the dashed line is the quasinormal mode extracted from the data.

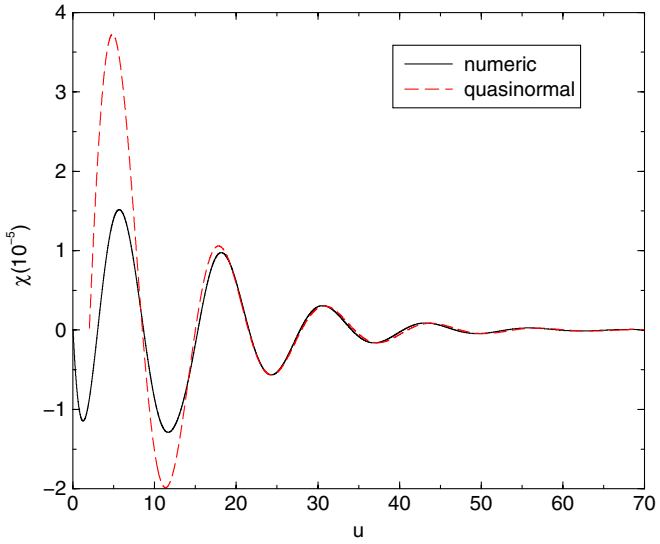

FIG. 9 (color online). The function χ at $\mathcal{I}$ as a function of Bondi time, showing the quasinormal mode regime oscillations for $\ell = 2$, $m = 0$. The solid line is the output from LEO for $N_\zeta = 11$ and $N_x = 1001$, when the initial data and boundary conditions are given as for the convergence test, except that $r_{\rm in} = 2.13M$; the dashed line is the quasinormal mode extracted from the data.

we have shown instead, in the same graph, the quasinormal mode $\chi = \exp\omega u$ (dashed line) extracted, i.e. the fit provided by HARMINV. There is some disagreement initially between the numerical solution and the fit, as the numerical solution settles into the dominant quasinormal mode, a process which takes from one to one and a half cycles of the quasinormal mode.

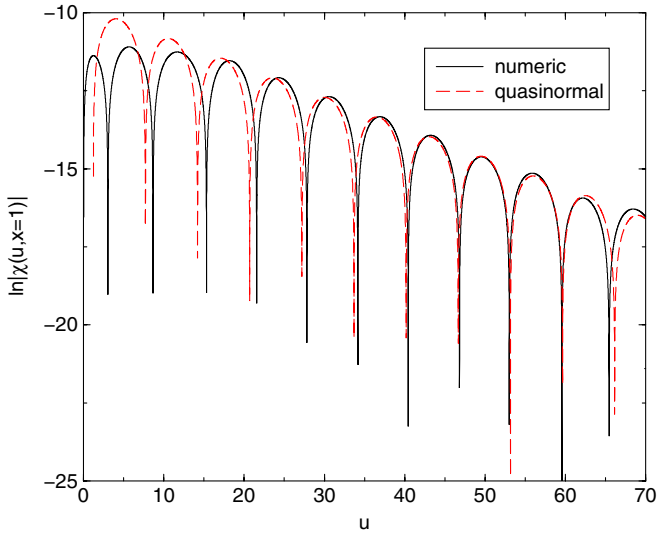

FIG. 10 (color online). Log of the absolute value of the function $\chi(u)$ at $\mathcal{I}$ as a function of Bondi time. Parameters and conditions are the same as in Fig. 9. The solid line is the output from LEO; the dashed line is the quasinormal mode extracted from the data.

For the same initial data, but with $\ell = 2$, we read a frequency $\omega = 0.4971\,(3\%) - 0.0992i\,(2\%)$, in the range $u = [40, 70]$, with the comparison between the computed signal and [40] shown in Figs. 9 and 10. We have observed that the relative percent error for the decay rate is larger for $\ell = 1$ because it depends strongly on the value selected for the location of the boundary, $r_{\rm in}$. Numerical experiments with the radial code confirm this and suggest that, by carefully tuning the location of the inner boundary, better accuracy can be achieved for any one value of ℓ. We have done this only partially in computing the frequency for the case $\ell = 2$. We want to emphasize that the dependence of the frequency on the boundary is not a numerical artifact of the code, but a consequence of the choice of outgoing null coordinates. This is confirmed by numerical experiments with the radial code in ingoing coordinates, in which case we find that the frequency can be read off with an error of less than 0.1% for the same initial data and grid sizes.

B. Energy conservation

For initial data of the form (65) with $\ell = 0, 1, 2$, Fig. 11 shows that energy is conserved in the linear regime. It is immediately clear from the graph that the energy contained on the initial slice increases with the value of ℓ. In all cases energy is clearly conserved; however, we have seen also that, if the resolution is not sufficient for a given ℓ, this fact shows up clearly in the graph of energy conservation. Thus, we can use energy conservation, as well as the results from

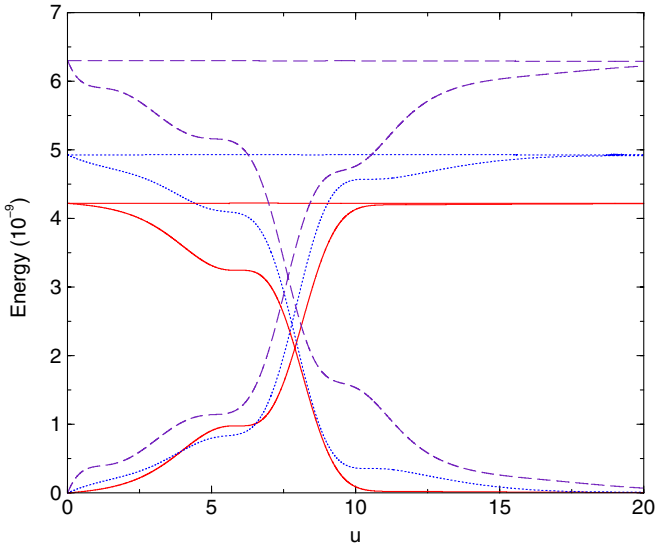

FIG. 11 (color online). Energy conservation as a function of Bondi time for $\ell = 0$ (solid line), $\ell = 1$ (dotted line), and $\ell = 2$ (long dashed line). This calculation was done using the same grid parameters as for Fig. 9 except for $r_{\rm in} = 2.3$. For each specific ℓ, the descending curve corresponds to energy given by Eq. (54). The ascending curve corresponds to the algebraic sum of $E_{\rm in} = -\int P_{\rm in} du$ and $E_{\rm out} = \int P_{\rm out} du$. Thus, in accordance with Eq. (56), the horizontal curve represents the global conservation of energy.

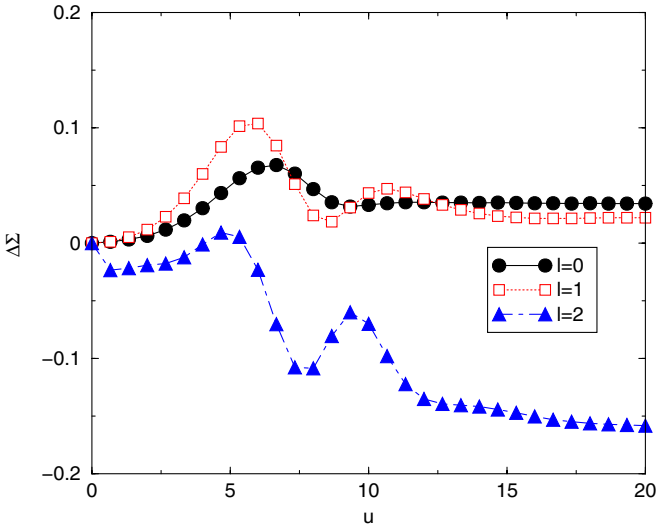

FIG. 12 (color online). Percentage variation in $\Sigma(u)$ with respect to $\Sigma(0)$ as a function of Bondi time for $\ell = 0$ (circles), $\ell = 1$ (squares), and $\ell = 2$ (triangles). The graph shows that energy is conserved to within less than 0.2% of the energy content of the initial surface.

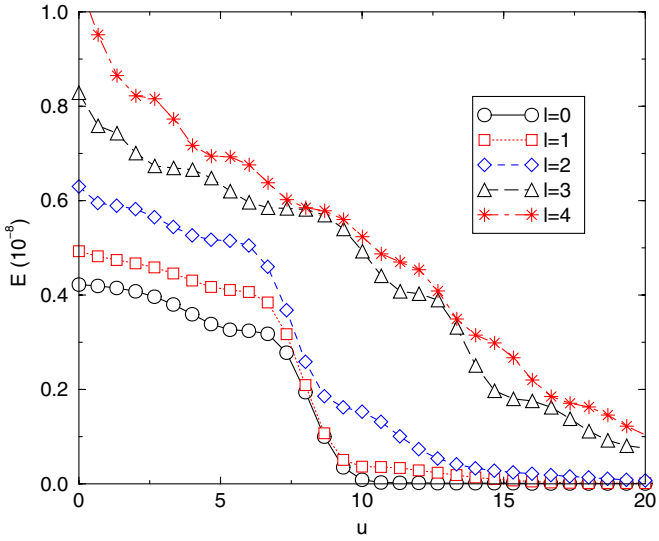

FIG. 13 (color online). Energy content $E(u)$ as a function of Bondi time for $\ell = 0$ (circles), $\ell = 1$ (squares), $\ell = 2$ (diamonds), $\ell = 3$ (triangles), and $\ell = 4$ (stars). This calculation was done using the grid parameters $N_\zeta = 11$ and $N_x = 1001$. The initial data and boundary conditions are the same as in the convergence test.

running the same initial data on the radial code, to debug and calibrate the nonlinear code, as well as to estimate the evolution time needed and its computational requirements. From Fig. 11 alone, the reader might be left to guess as to the extent of the deviation of the total energy from a straight line, since that deviation is clearly so small that it does not show up in the plot for any of the three simulations reported in Fig. 11. Figure 12 shows the variation in the energy balance $\Delta\Sigma(u)$, defined as the percentage variation in $\Sigma(u)$ relative to the initial value, $\Sigma(u_0)$, i.e.

$$\Delta\Sigma = (\Sigma(u)/\Sigma(0) - 1) \times 100. \tag{66}$$

It can be seen from Fig. 12 that the relative change $\Delta\Sigma(u)$ stays below 0.2% during the simulation. We will revisit energy conservation in the context of large resolution simulations in Sec. VII C.

Figure 13 shows the energy content $E(u)$ as a function of Bondi time u for a sequence of simulations with initial data (65) with varying values of ℓ. For lower values of ℓ ($\ell = 0, 1, 2$), the energy content $E(u)$ decays slowly at first, then drops rather sharply, and afterwards it decays again slowly, at a much lower rate. For higher values of ℓ ($\ell = 3, 4$), the energy decays approximately monotonically from the beginning of the simulation.

We also observe (see Fig. 14) that, in general, increasing ℓ corresponds to an increase of the energy radiated at $\mathcal{I}$. The oscillations observed in the profiles increase with the value of ℓ, as would be expected. The most interesting observation in the analysis of energy balance arises from Fig. 15, and is the following: for values of ℓ from 0 to 2, the total energy flux towards the black hole [as measured by $E_{\rm in}(u)$ as $u \to \infty$] increases with the value of ℓ; however, for values of $\ell \geq 2$, the total flux of energy towards the black hole diminishes with increasing values of ℓ. We have confirmed that this is the case with the radial code, so this is not a nonlinear effect. It is also clear that the sudden change of energy for $\ell \leq 2$ is due to the energy carried away by the scalar field as it falls into the black hole. At about $\ell = 2$, the radiation into the black hole saturates, and

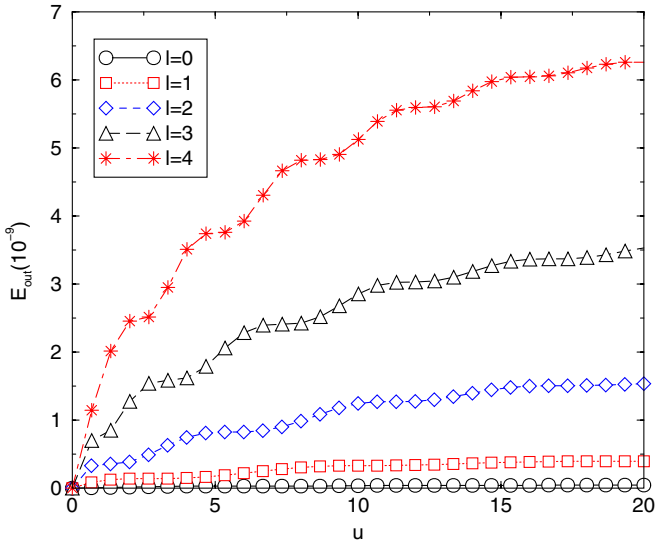

FIG. 14 (color online). Energy flow to infinity $E_{\rm out} = \int P_{\rm out}(u)du$ as a function of Bondi time for $\ell = 0$ (circles), $\ell = 1$ (squares), $\ell = 2$ (diamonds), $\ell = 3$ (triangles), and $\ell = 4$ (stars). This calculation was done using the same conditions as in Fig. 13.

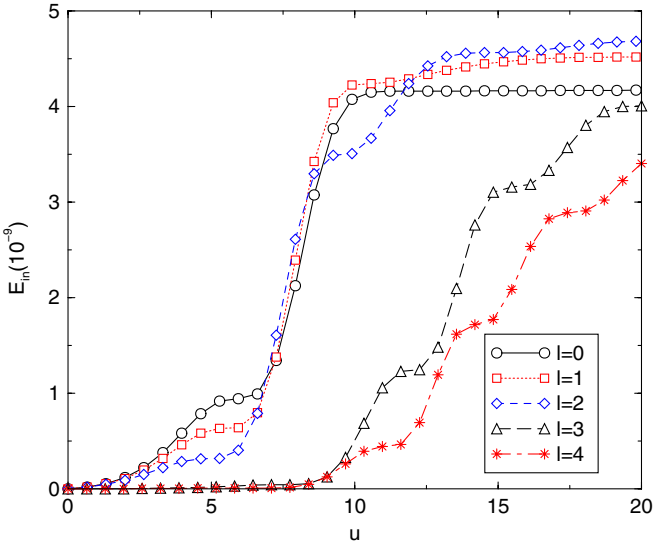

FIG. 15 (color online). Energy flow into the black hole, $E_{\rm in} = -\int P_{\rm in}(u)du$, as a function of Bondi time for $\ell = 0$ (circles), $\ell = 1$ (squares), $\ell = 2$ (diamonds), $\ell = 3$ (triangles), and $\ell = 4$ (stars). Both curves for $\ell = 3$ and $\ell = 4$ eventually saturate, without crossing, for for $u > 30$. This calculation was done using the same conditions as in Fig. 13.

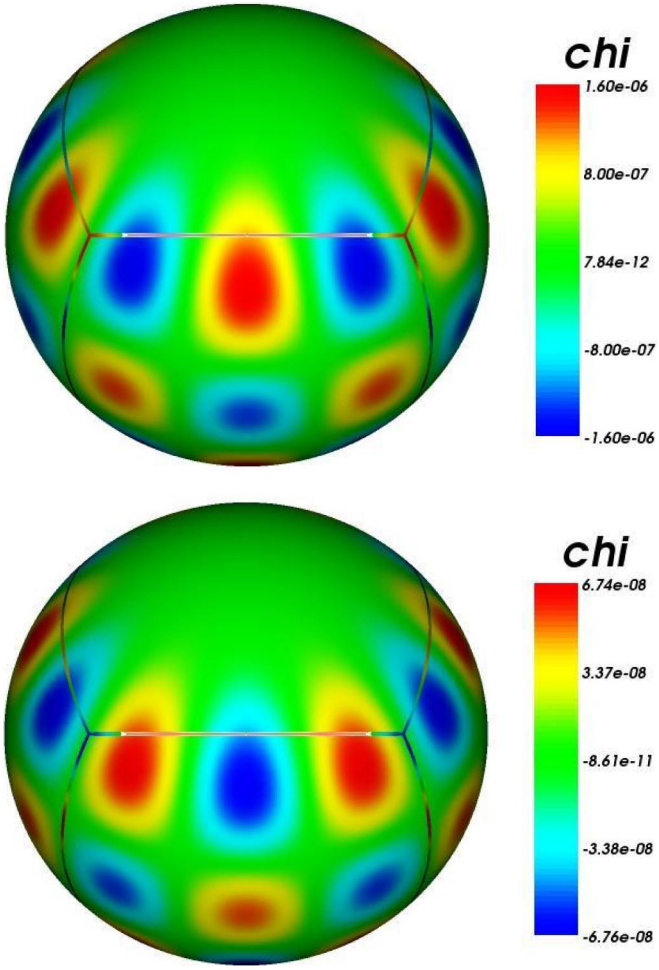

FIG. 16 (color online). Surface plots of χ at $\mathcal{I}$ for $u = 2.5M$ (top graph) and $u = 30M$ (bottom graph). The parameters of the initial data are $\lambda = 10^{-4}$, $r_0 = 3M$, $\sigma = 0.5M$, $\ell = 8$, $m = 6$. The grid size is $N_\zeta = 93$, $N_x = 1501$.

for higher values of ℓ, i.e. for $\ell > 2$, the centrifugal potential barrier prevents much of the field from falling into the black hole. Thus, for the same amplitude, configurations with higher angular momentum (larger ℓ values) carry more energy, most of which will be radiated away and less of which will fall into the black hole, so in that sense these configurations are proportionally more efficient at carrying energy out to $\mathcal{I}$.

C. Large resolution simulations

In order to get a first glimpse of the type of simulations that our framework enables us to perform, and to perform a final calibration check of the nonlinear code, we select initial data given by Eq. (65), with $\lambda = 10^{-4}$, $r_0 = 3M$, $\sigma = \frac{1}{2}M$, $\ell = 8$, $m = 6$, and evolve this configuration until $u = 30M$. This simulation is performed in compactified outgoing coordinates, with the treatment of the inner boundary as described in Sec. VI C. The plots shown are of quantities computed at null infinity, $\mathcal{I}$. The angular grid has size $N_\zeta = 93$, that is, there are 372 points on a great circle on the sphere, while the radial grid has $N_x = 1501$ points. This simulation requires 27 hours on 54 processors, for a total of 1458 processor hours, or the equivalent of two months of a single-processor run. It is by no means the largest simulation we could run with our framework: we have performed scaling studies that indicate the code scales linearly well into the 4000 + processor range, but it suffices as a demonstration of the resolution that can be achieved and the typical turnaround times. We assign no particular significance to the initial data selected, other than the fact that its angular complexity provides an excellent test of the code. On any large simulation, data analysis and visualization is always a challenge. In LEO, visualization is performed by having each processor write its own data set at run time; the individual data files are then post-processed, and graphs of the desired quantities are generated with PARAVIEW [49]. PARAVIEW allows us to easily generate graphs of slices at constant coordinate lines and of volumetric renderings of various fields. Of particular interest to us is the behavior of the various metric quantities at null infinity.

Figures 16–20 display the code variables ϕ, J, β, U, and W as functions on the sphere at null infinity, at $u = 2.5M$ and at $u = 30M$. In each case, the graphs show the corresponding field on the six cubed-sphere caps (the gap between the caps is the actual size of the spacing between grid cells). The north pole is at the top of the figure, rotated 45 degrees towards the observer. For those fields that are complex (and which have spin different from zero), i.e. J,

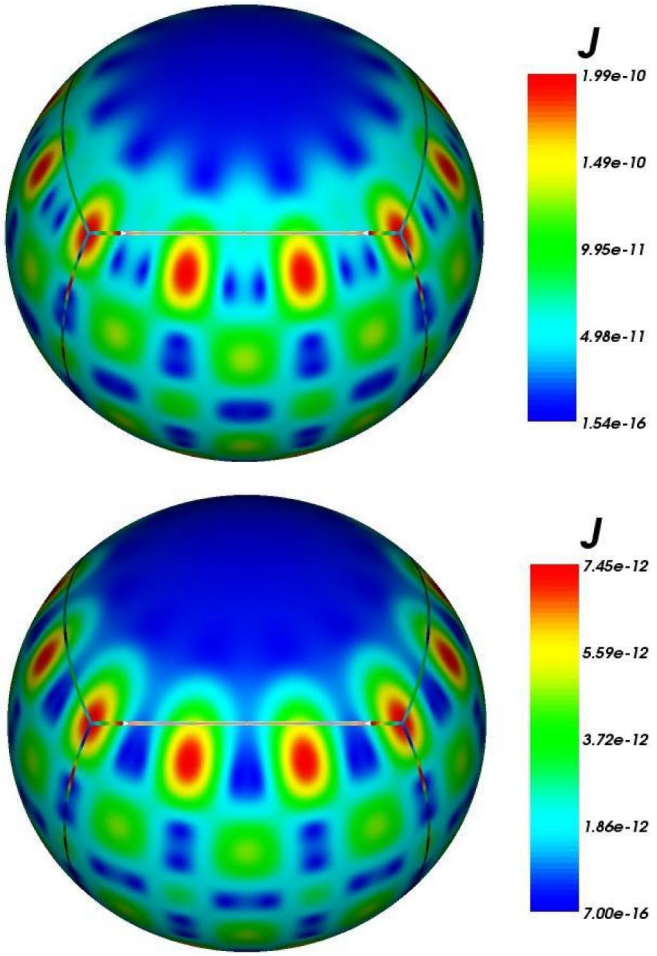

FIG. 17 (color online). Surface plots of $J\bar{J}$ at $\mathcal{I}$ for $u = 2.5M$ (top graph) and $u = 30M$ (bottom graph). The parameters of the initial data and grid size are the same as in Fig. 16.

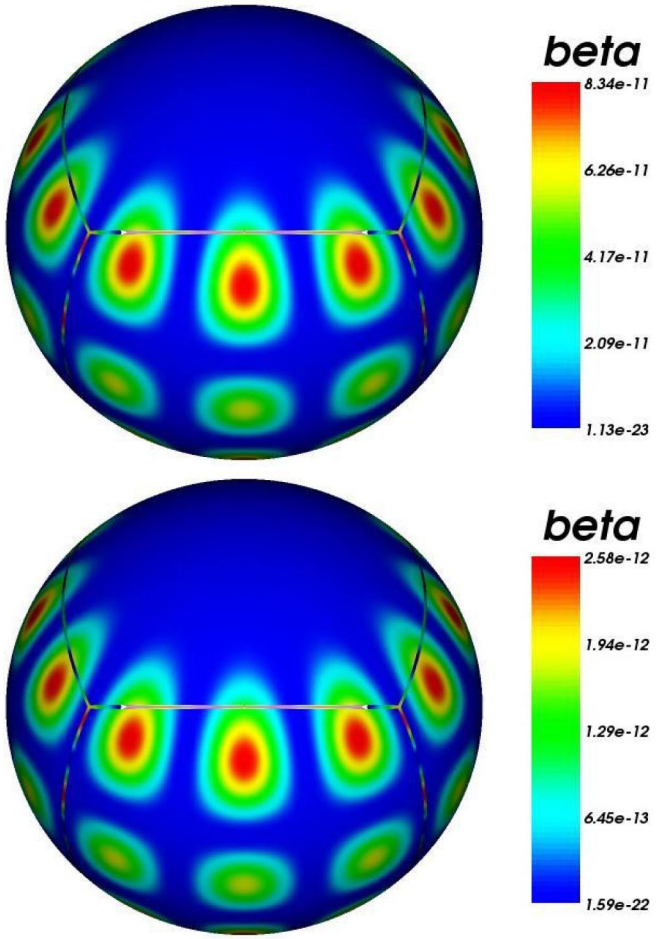

FIG. 18 (color online). Surface plots of β at $\mathcal{I}$ for $u = 2.5M$ (top graph) and $u = 30M$ (bottom graph). The parameters of the initial data and grid size are the same as in Fig. 16.

U, we display, for ease of visualization, the combinations $J\bar{J}$ and $U\bar{U}$, which are real and have spin zero. Clearly visible in Fig. 16 is the $m = 6$ azimuthal dependence of the field, marked by the presence of six maxima and minima. It is also apparent that χ oscillates in time, as the maxima and minima alternate between the top and bottom figures. The angular dependence is preserved by the evolution, as expected, as the only change between the two figures is in the overall amplitude (by a factor of ≈ 25 in between the two times shown). The graphs of $J\bar{J}$, Fig. 17, are clearly different in their angular dependence, showing the presence of various harmonics at an earlier time. This can be understood since in our initial data $J(r, x^A) = 0$; thus J develops from χ, i.e. $J \sim (\eth\phi)^2$, and it is not until later times that a definite profile for J has formed. The graph of β, Fig. 18, shows precisely the angular dependence resulting from the contribution $\beta_{,r} \sim (\phi_{,r})^2$ to the source term in Eq. (33), and remains constant throughout the simulation, up to an overall amplitude. The graphs of U and W, Figs. 19 and 20, show higher-order angular dependence arising from the angular derivatives of U, which in turn are essentially driven by the source term in Eq. (35), i.e. by $U \sim Q \sim \phi\eth\phi$.

Figure 21 shows again that energy is conserved during the entire simulation. The variation in the energy balance $\Sigma(u)$ is well below 1%; thus $\Sigma(u)$ is indistinguishable from a straight line at the resolution of the graph, as noted in Sec. VII B. To more fully appreciate to what extent energy is conserved, the graph insert in Fig. 21 shows the percentage variation of $\Sigma(u)$, normalized to its value at $u = 0$. From the graph insert we see that the total energy varies by at most 0.025% during the simulation. As we stated earlier, energy conservation is a requisite for accuracy in the waveforms; note that energy is conserved during this simulation to within tighter limits than in the simulations of Sec. VII A, which is due primarily to the increased angular resolution. For this simulation we took $N_\zeta = 93$ in order to accurately resolve the higher-order harmonic angular dependence, where in Sec. VII A we set $N_\zeta = 11$; this amounts to (approximately) an 8-fold increase in resolu-

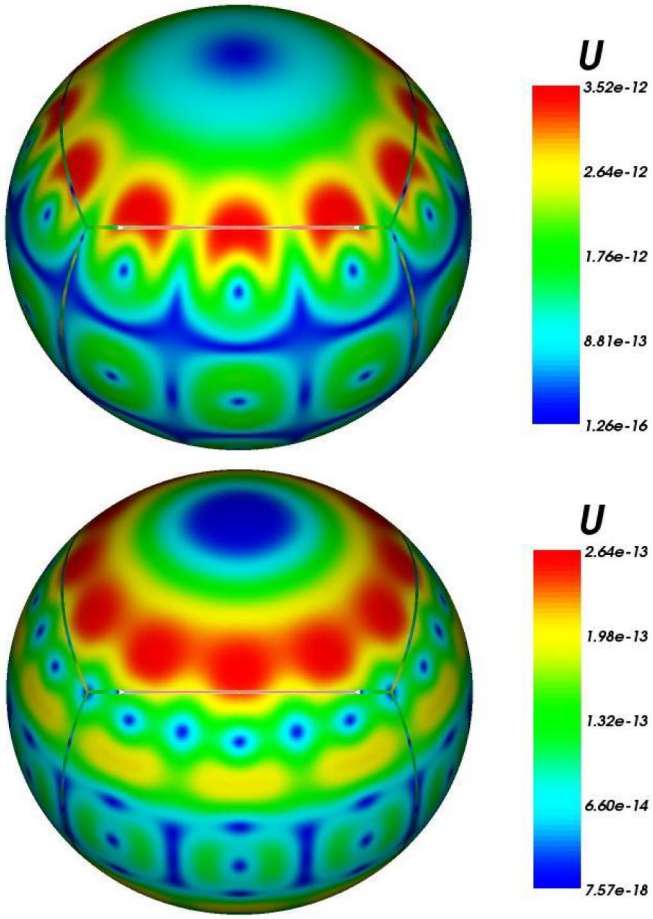

FIG. 19 (color online). Surface plots of $U\bar{U}$ at $\mathcal{I}$ for $u = 2.5M$ (top graph) and $u = 30M$ (bottom graph). The parameters of the initial data and grid size are the same as in Fig. 16.

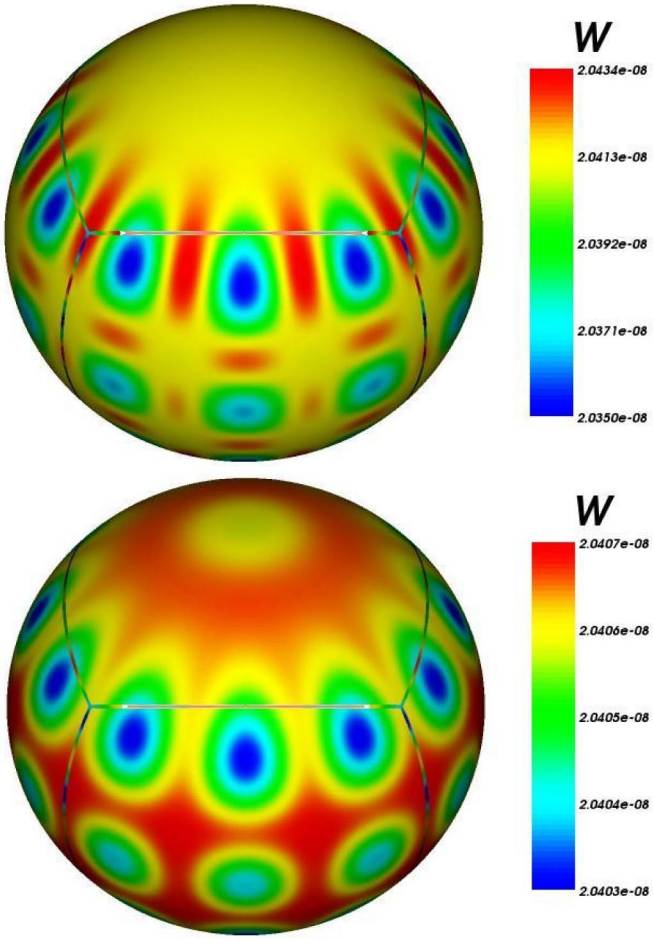

FIG. 20 (color online). Surface plots of W at $\mathcal{I}$ for $u = 2.5M$ (top graph) and $u = 30M$ (bottom graph). The parameters of the initial data and grid size are the same as in Fig. 16.

tion, and for the same radial resolution, a 64-fold increase on the computing resources required.

We would be remiss if we did not discuss, at least briefly, the performance characteristics of our code. As part of our calibration and testing, we have performed detailed profiling studies, which we will not go into detail about here. Suffice it to say that, in its current configuration, the code performs at approximately 20% of peak on the Cray XT3. Its weak scaling is linear (that is, its performance solving progressively larger configurations, while keeping the load per processor constant, scales linearly with the number of cores), while running on up to 4056 CPU cores on the Cray XT3 at the Pittsburgh Supercomputing Center.

VIII. CONCLUDING REMARKS AND OUTLOOK OF FUTURE WORK

We have presented a new computational framework (LEO), which we can use to perform large-scale, high-resolution calculations in the context of the characteristic approach in numerical relativity. This highly parallel and easily extensible implementation has been used to solve the model problem of a massless scalar field minimally coupled to gravity (the three-dimensional Einstein-Klein-Gordon problem). We have shown that our nonlinear code is globally second-order convergent, and how accurately we can follow quasinormal mode ringing. We have studied the balance of energy for a number of initial data sets with different angular structure. Aside from the interesting result of energy flow saturation through the Schwarzschild horizon, the LEO framework offers a good prospect to study new configurations beyond the linear regime and the grid sizes used in this work.

Future directions we are currently exploring include the application of the LEO framework to a consistent, quasi-linear, fully first-order formalism derived from [13], and the extension of the model problem considered here to massive scalar fields. The latter case is particularly important because it will allow us to simulate a boson star orbiting a black hole. We will compare the performance of the characteristic framework in ingoing versus outgoing null coordinates in the extraction of quasinormal modes, and in the study of nonlinear effects in the neighborhood of

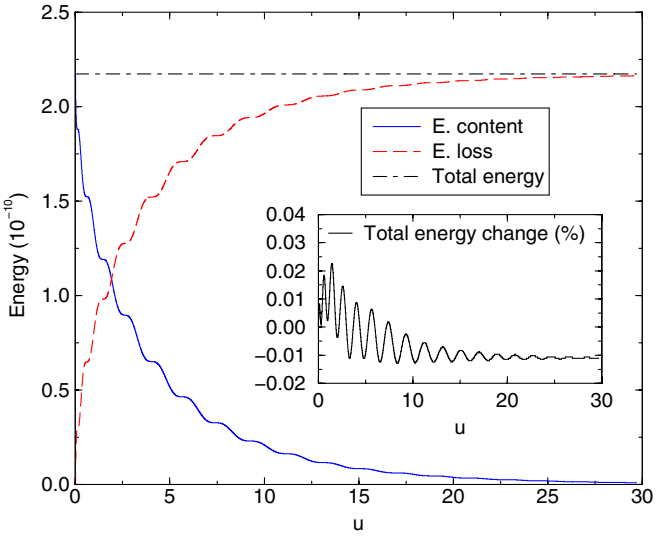

FIG. 21 (color online). Energy conservation for the simulation which generated the results shown in Figs. 16–20. The solid line corresponds to the energy content $E(u)$ at successive times, the dashed line to the sum of the energy radiated through the inner $[E_{\rm in}(u)]$ and outer $[E_{\rm out}(u)]$ boundaries, and the dot-dashed line to the sum, $\Sigma(u) = E(u) + E_{\rm in}(u) + E_{\rm out}(u)$, respectively. The insert graph shows the percentage variation in $\Sigma(u)$, relative to its final value at $u = 30$.

the central black hole. The underlying framework can be applied equally well to 3 + 1 formulations of the Einstein equations in spherical coordinates, in particular, to a generalization to three dimensions of the Bondi-Sachs gauge of [43], and finally to matched 3 + 1 and characteristic evolutions [42,50,51].

We have not addressed, in the present work, some outstanding problems with the calculation of the Bondi news [30], some of which arise from second angular derivatives of the metric fields at $\mathcal{I}$ entering in the computation of the news, a feature which can lead to substantial propagation of errors. We have observed, during our simulations, that the metric fields computed at $\mathcal{I}$ are smooth, as evident in Figs. 16–20 (and so are those fields which represent their angular derivatives, although these are not shown here). It should be noted that, in our formulation [8], the first angular derivatives of some fields have been promoted to auxiliary variables, for which a hypersurface equation is integrated radially. (We do this for those fields whose second angular derivatives enter in the computation of the news). In practice, this means that only first-order angular derivatives of any of the fields we evolve need to be computed to calculate the news. It is possible that the cubed-sphere approach leads to substantial improvements in the computed waveforms, and this issue remains to be addressed. Although of potential importance for the accuracy of gravitational waveforms, such a study lies outside the scope of the present work.

ACKNOWLEDGMENTS

R. G. wishes to thank Raghurama Reddy for many enlightening discussions on the parallelization of the characteristic code, and acknowledges the hospitality of the Universidad de Los Andes, Mérida, Venezuela, and the Kavli Institute for Theoretical Physics at Santa Barbara, where portions of this project were carried out. W. B. thanks the Pittsburgh Supercomputing Center for hospitality. This work was supported in part by the National Science Foundation under Grant No. PHY-0135390 to Carnegie Mellon University, and No. PHY-0244752 and No. PHY-0555218 to Duquesne University; by FONACIT under Grant No. S1-98003270 and No. F2002000426; and by CDCHT-ULA under Grant No. C-1267-04-05-A. Code development and simulations were carried out on the Cray XT3 at the Pittsburgh Supercomputing Center, under Grant No. PHY060004P and No. PHY070022N. Additional computer time was provided by the Centro Nacional de Cálculo Científico, Universidad de Los Andes (CeCalcULA).

[1] R. Gómez, L. Lehner, P. Papadopoulos, and J. Winicour, Classical Quantum Gravity **14**, 977 (1997).
[2] N. T. Bishop, R. Gómez, L. Lehner, and J. Winicour, Phys. Rev. D **54**, 6153 (1996).
[3] N. T. Bishop, R. Gómez, L. Lehner, M. Maharaj, and J. Winicour, Phys. Rev. D **56**, 6298 (1997).
[4] R. Gómez, L. Lehner, R. L. Marsa, and J. Winicour, Phys. Rev. D **57**, 4778 (1998).
[5] R. Gómez, L. Lehner, R. L. Marsa, J. Winicour, A. M. Abrahams, A. Anderson, P. Anninos, T. W. Baumgarte, N. T. Bishop, and S. R. Brandt *et al.*, Phys. Rev. Lett. **80**, 3915 (1998).
[6] N. T. Bishop, R. Gómez, L. Lehner, M. Maharaj, and J. Winicour, Phys. Rev. D **60**, 024005 (1999).
[7] R. Gómez, S. Husa, and J. Winicour, Phys. Rev. D **64**, 024010 (2001).
[8] R. Gómez, Phys. Rev. D **64**, 024007 (2001).
[9] Y. Zlochower, R. Gómez, S. Husa, L. Lehner, and J. Winicour, Phys. Rev. D **68**, 084014 (2003).
[10] N. T. Bishop, R. Gómez, L. Lehner, M. Maharaj, and J. Winicour, Phys. Rev. D **72**, 024002 (2005).
[11] W. Barreto, A. da Silva, R. Gómez, L. Lehner, L. Rosales, and J. Winicour, Phys. Rev. D **71**, 064028 (2005).
[12] R. Gómez, S. Husa, L. Lehner, and J. Winicour, Phys. Rev. D **66**, 064019 (2002).
[13] R. Gómez and S. Frittelli, Phys. Rev. D **68**, 084013 (2003).

[14] E. Newman and R. Penrose, J. Math. Phys. (N.Y.) **7**, 863 (1966).
[15] J. N. Goldberg *et al.*, J. Math. Phys. (N.Y.) **8**, 2155 (1967).
[16] N. T. Bishop, C. J. S. Clarke, and R. A. d'Inverno, Classical Quantum Gravity **7**, L23 (1990).
[17] F. Pretorius and L. Lehner, J. Comput. Phys. **198**, 10 (2004).
[18] E. Schnetter, S. H. Hawley, and I. Hawke, Classical Quantum Gravity **21**, 1465 (2004).
[19] U. Sperhake, B. Kelly, P. Laguna, K. L. Smith, and E. Schnetter, Phys. Rev. D **71**, 124042 (2005).
[20] B. Imbiriba, J. Baker, D.-I. Choi, J. Centrella, D. R. Fiske, J. D. Brown, J. R. van Meter, and K. Olson, Phys. Rev. D **70**, 124025 (2004).
[21] D. L. Brown, W. D. Henshaw, and D. J. Quinlan, in *Proceedings of the Scientific Computing in Object-Oriented Parallel Environments (ISCOPE '97)* (Springer-Verlag, London, England, 1997), pp. 177–184, ISBN 3-540-63827-X.
[22] C. Ronchi, R. Iacono, and P. S. Paolucci, J. Comput. Phys. **124**, 93 (1996).
[23] R. Sadourny, Mon. Weather Rev. **100**, 136 (1972).
[24] A. Adcroft, J.-M. Campin, C. Hill, and J. Marshall, Mon. Weather Rev. **132(12)**, 2845 (2004); see also MIT's Climate Modeling Initiative, http://paoc.mit.edu/cmi/.
[25] A. V. Koldoba, M. M. Romanova, G. V. Ustyugova, and R. V. E. Lovelace, Astrophys. J. **576**, L53 (2002).
[26] M. M. Romanova, G. V. Ustyugova, A. V. Koldoba, J. V. Wick, and R. V. E. Lovelace, Astrophys. J. **595**, 1009 (2003).
[27] J. A. Rossmanith, J. Comput. Phys. **213**, 629 (2006).
[28] L. Lehner, O. Reula, and M. Tiglio, Classical Quantum Gravity **22**, 5283 (2005).
[29] N. T. Bishop (private communication).
[30] C. Reisswig, N. T. Bishop, C. W. Lai, J. Thornburg, and B. Szilagyi, Classical Quantum Gravity **24**, S327 (2007).
[31] J. Thornburg, Classical Quantum Gravity **21**, 3665 (2004).
[32] J. Stewart, *Advanced General Relativity* (Cambridge University Press, Cambridge, England, 1993).
[33] R. Gomez, J. Winicour, and R. Isaacson, J. Comput. Phys. **98**, 11 (1992).
[34] W. Gropp, L. Ewing, and A. Skjellum, *Using MPI—Portable Parallel Programming with the Message-Passing Interface* (MIT Press, Cambridge, MA, 1999), 2nd ed.
[35] H. Bondi, M. G. J. van der Burg, and A. W. K. Metzner, Proc. R. Soc. A **269**, 21 (1962).
[36] R. K. Sachs, Proc. R. Soc. A **270**, 103 (1962).
[37] E. T. Newman and R. Penrose, J. Math. Phys. (N.Y.) **7**, 863 (1966).
[38] H.-P. Nollert, Classical Quantum Gravity **16**, R159 (1999).
[39] H.-P. Nollert and B. G. Schmidt, Phys. Rev. D **45**, 2617 (1992).
[40] R. Konoplya, J. Phys. Stud. **8**, 93 (2004).
[41] J. Winicour and L. Tamburino, Phys. Rev. Lett. **15**, 601 (1965).
[42] R. Gómez, R. L. Marsa, and J. Winicour, Phys. Rev. D **56**, 6310 (1997).
[43] S. Frittelli and R. Gómez, Phys. Rev. D **75**, 044021 (2007).
[44] L. Lehner, J. Comput. Phys. **149**, 59 (1999).
[45] HARMINV: a program to solve the harmonic inversion problem via the filter diagonalization method (FDM), developed by S. Johnson, http://ab-initio.mit.edu/wiki/index.php/Harminv.
[46] V. A. Mandelshtam and H. S. Taylor, J. Chem. Phys. **107**, 6756 (1997).
[47] V. A. Mandelshtam and H. S. Taylor, J. Chem. Phys. **109**, 4128 (1998).
[48] E. N. Dorband, E. Berti, P. Diener, E. Schnetter, and M. Tiglio, Phys. Rev. D **74**, 084028 (2006).
[49] PARAVIEW is an open-source, multiplatform application designed to visualize data sets of size varying from small to very large; http://www.paraview.org/.
[50] N. T. Bishop, R. Gomez, P. R. Holvorcem, R. A. Matzner, P. Papadopoulos, and J. Winicour, Phys. Rev. Lett. **76**, 4303 (1996).
[51] R. Gómez, P. Laguna, P. Papadopoulos, and J. Winicour, Phys. Rev. D **54**, 4719 (1996).