Three-dimensional numerical general relativistic hydrodynamics: Formulations, methods, and code tests

José A. Font,[1] Mark Miller,[2],* Wai-Mo Suen,[2,3] and Malcolm Tobias[2]

[1]*Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, Schlaatzweg 1, D-14473, Potsdam, Germany*
[2]*McDonnell Center for the Space Sciences, Department of Physics, Washington University, St. Louis, Missouri 63130*
[3]*Physics Department, Chinese University of Hong Kong, Hong Kong*

This is the first in a series of papers on the construction and validation of a three-dimensional code for general relativistic hydrodynamics, and its application to general relativistic astrophysics. This paper studies the consistency and convergence of our general relativistic hydrodynamic treatment and its coupling to the spacetime evolutions described by the full set of Einstein equations with a perfect fluid source, complimenting a similar study of the (vacuum) spacetime part of the code. The numerical treatment of the general relativistic hydrodynamic equations is based on high resolution shock capturing schemes, specifically designed to solve non-linear hyperbolic systems of conservation laws. These schemes rely on the characteristic information of the system. A spectral decomposition for general relativistic hydrodynamics suitable for a general spacetime metric is presented. Evolutions based on different approximate Riemann solvers (flux-splitting, Roe, and Marquina) are studied and compared. The coupling between the hydrodynamics and the spacetime (the right and left hand side of the Einstein equations) is carried out in a treatment which is second order accurate in *both* space and time. The spacetime evolution allows for a choice of different formulations of the Einstein equations, and different numerical methods for each formulation. Together with the different hydrodynamical methods, there are twelve different combinations of spacetime and hydrodynamical evolutions. Convergence tests for all twelve combinations with a variety of test beds are studied, showing consistency with the differential equations and correct convergence properties. The test-beds examined include shock tubes, Friedmann-Robertson-Walker cosmology tests, evolutions of self-gravitating compact (TOV) stars, and evolutions of relativistically boosted TOV stars. Special attention is paid to the numerical evolution of strongly gravitating objects, e.g., neutron stars, in the full theory of general relativity, including a simple, yet effective treatment for the surface region of the star (where the rest mass density is abruptly dropping to zero).

PACS number(s): 04.25.Dm, 47.75.+f, 95.30.Sf, 97.60.Jd

I. INTRODUCTION

The field of computational astrophysics is entering an exciting and challenging era. The large amount of observational data involving general relativistic phenomena requires the integration of numerical relativity with the traditional tools of astrophysics, such as hydrodynamics, magneto-hydrodynamics, nuclear astrophysics, and radiation transport. General relativistic astrophysics — astrophysics involving gravitational fields so strong and dynamical that the full Einstein field equations are required for its accurate description, is quickly becoming a promising area of research.

As a first step in our study of ''computational general relativistic astrophysics,'' our collaboration (the NCSA–Potsdam–Wash University Numerical Relativity Collaboration) is building a code called ''CACTUS'' for solving the full set of Einstein field equations coupled to a perfect fluid source. Such a code will have many applications for astrophysical processes involving neutron stars and black holes. In this paper we present the formulation and methods of the 3D general relativistic hydrodynamic part of the code, and its coupling to the spacetime part of the code. We also present various tests for the validation of the code. The complimentary presentation on the (vacuum) spacetime evolution part of the code has been given in [1,2]. In the following we begin by discussing the background of our code development effort.

A. Motivation

Two of the major directions of astronomy in the next century are high energy astrophysics (x-ray and γ-ray astronomy) and gravitational wave astronomy. The former is driven by advanced x-ray and γ-ray satellite observations, e.g., CGRO, AXAF, GLAST [3], XMM, INTEGRAL, that are either current or planned in the next few years. High energy radiation is often emitted by highly relativistic events in regions of strong gravitational fields, e.g., near black holes (BHs) and neutron stars (NSs). One of the biggest mysteries of modern astronomy, γ-ray bursts, is likely related to processes involving interactions of compact binaries (BH-NS or NS-NS) or highly explosive collapse to a black hole (''hypernova'') (see, e.g., [4] and references therein). Such high energy astrophysical events often involve highly dynamical gravitational fields, strong gravitational wave emissions, and ejecta moving at ultrarelativistic speeds with relativistic Lorentz factors up to $10^3 - 10^4$. The modeling of such events can only be achieved by means of hydrodynamical simulations in the full theory of general relativity.

*Corresponding author.

61 044011-1

The second major direction, gravitational wave astronomy, involves the dynamical nature of spacetime in Einstein's theory of gravity. The tremendous recent interest in this frontier is driven by the gravitational wave observatories presently being built or planned in the U.S., Europe, and outer space, e.g., Laser Interferometric Gravitational Wave Observatory (LIGO), VIRGO, GEO600, Laser Interferometer Space Antenna (LISA), LAGOS [5], and the Lunar Outpost Astrophysics Program [5]. The American LIGO and its European counterparts VIRGO and GEO600 are scheduled to be on line in a few years [6], making gravitational wave astronomy a reality. The space detector LISA has been selected as one of the three ''cornerstone missions'' of the European Space Agency [7]. These observatories provide a completely new window on the universe: existing observations are mainly provided by the electromagnetic spectrum, emitted by individual electrons, atoms, or molecules, and are easily absorbed, scattered, and dispersed. Gravitational waves are produced by the coherent bulk motion of matter and travel nearly unscathed through space, coming to us carrying the information of the strong field regions where they were originally generated [8]. This new window will provide very different information about our universe that is either difficult or impossible to obtain by traditional means.

The numerical (theoretical) determination of gravitational wave forms is crucial for gravitational wave astronomy. Physical information in the data is to be extracted through template matching techniques [9], which *presupposes* that reliable wave forms are known. Accurate wave form detections are important both as probes of the fundamental nature of gravity and for the unique physical and astronomical information they carry, ranging from nuclear physics (the equation of state of NSs [9]) to cosmology (direct determination of the Hubble constant without going through the ''cosmic distance ladder'' [10]). In most situations, the wave form cannot be calculated without a numerical simulation based on the full theory of general relativity. This need for wave form templates is an important motivation of our effort.

In short, both of these frontiers of astronomy call for computational general relativistic astrophysics, i.e., the integration of numerical relativity with traditional tools of computational astrophysics, e.g., computational hydrodynamics, radiation transport, nuclear astrophysics, and magneto-hydrodynamics. If we are to fully understand the observational data generated by the non-linear and dynamical gravitational fields, detailed modeling taking dynamic general relativity into full account must be carried out.

B. Existing work in general relativistic hydrodynamics

We begin by briefly reviewing some of the significant existing investigations in the field of numerical general relativistic hydrodynamics (GR-Hydro in the following) to set the stage for the description of our own work. While there has been much effort in the study of relativistic hydrodynamics in *pre-determined* (fixed, or with its time evolution specified) background spacetimes, we focus on studies that are most relevant to *dynamical* spacetimes with the matter flows acting as sources to the Einstein equations.

The pioneering work dates back to the one-dimensional supernova core-collapse code by May and White [11]. It was based on a Lagrangian (i.e., coordinates co-moving with the fluid) finite difference scheme with artificial viscosity terms included in the equations to damp the spurious numerical oscillations caused by the presence of shock waves in the flow solution. Numerous astrophysical simulations were based on this approach. One drawback is that the Lagrangian character of the code makes it difficult to be extended to the multidimensional case.

The pioneering Eulerian (i.e., coordinates not co-moving with the fluid) finite difference GR-Hydro code was developed by Wilson [12] in the early 1970s. It used a combination of artificial viscosity (AV) and upwind techniques. It became the kernel of a large number of codes developed in the 1980s. Many different astrophysical scenarios were investigated with these codes, ranging from axisymmetric stellar core-collapse [13–15], to accretion onto compact objects [16,17], and to numerical cosmology [18]. In the following, we give a short overview of this large body of work, paying more attention to the numerical methods used than to the physical results obtained.

While there are a large number of numerical investigations in pre-determined background spacetimes based on the AV approach (e.g., [12,19,16,17,20]), we focus on those using a fully self-consistent treatment evolving the spacetime dynamically with the Einstein equations coupled to a hydrodynamic source. Although there is much recent interest in this direction, only the spherically symmetric case (1D) can be considered essentially solved [21–25]. In axisymmetry, i.e. 2D, only a few attempts have been made, with most of them devoted to the study of the gravitational collapse and bounce of rotating stellar cores and the subsequent emission of gravitational radiation [26,14,15,27]. Reference [26] was the first to calculate a general relativistic stellar core collapse. The computation succeeded in tracking the evolution of matter and the formation of a black hole but the numerical scheme was not accurate enough to compute the emitted gravitational radiation. The code in [14] used a radial gauge and a mixture of polar and maximal slicing. The GR-hydro equations were solved with standard finite difference methods with AV terms. In [15] the numerical scheme for the matter fields was more sophisticated, using monotonic upwind reconstruction procedures and flux limiters, with discontinuous solutions handled by adding AV terms in the equations. In [27], a numerical study of the stability of star clusters in axisymmetry was performed. In this investigation, the source of the gravitational field was assumed to be a configuration of collisionless (dust) particles, which reduces the hydrodynamic computation to a straightforward integration of the geodesic equations.

Three-dimensional extensions of these AV based GR-Hydro treatments have been attempted over the last few years. Wilson's original scheme has been applied to the study of NS binary coalescence in [28,29] under the assumption of a conformally flat spacetime, which leads to a considerable simplification of the gravitational field equations. A

code employing the *full* set of Einstein equations and self-gravitating matter fields is currently being developed [30]. In this work the complete set of the equations, spacetime and hydrodynamics, are finite differenced in a uniform Cartesian grid using van Leer's scheme [31] with total variation diminishing (TVD) flux limiters (see, e.g., [32] for definitions). Shock waves are spread out using a tensor AV algorithm. With this code they have studied the gravitational collapse of a rotating polytrope to a black hole (comparing to the original axisymmetric computation of Ref. [14]) and the coalescence of a binary NS system. Further work to achieve longer term stability is under way [30].

The success of the artificial viscosity approach is well known. However, it has inherent difficulties in handling the ultrarelativistic regime [33]. In Wilson's formulation of the GR-hydro equations, there are explicit spacetime derivatives of the pressure in the source terms. This breaks the conservative character of the system and introduces complications into the numerical treatment. This motivated, in recent years, the effort of extending to relativistic hydrodynamics high-resolution shock-capturing (HRSC) schemes originally developed in classical (Newtonian) computational fluid dynamics. Such schemes are based on the solution of local Riemann problems, exploiting the hyperbolicity of the hydrodynamic equations. To use such numerical treatments, the hydrodynamic equations are first cast into a first order (hyperbolic) system of conservation (or balance) laws. The characteristic fields of the system are then determined which allows the construction of numerical schemes which propagate the information along the fluid characteristics. We refer the reader to [32] for a review of these methods for general hyperbolic systems of conservation laws.

HRSC schemes were first introduced into GR-Hydro in [34], and applied in (spherical) dynamical spacetimes in [35] and [36]. The latter investigation focused on, among other problems, the study of supernova core collapse (including the infall epoch, bounce, and shock propagation). The numerical code was based on the radial-gauge and polar-slicing coordinate conditions [37]. In [38] the GR-Hydro equations were analyzed in the ''3+1'' formalism and the theoretical building blocks to construct a HRSC scheme in multidimensions were presented. Axisymmetric studies using HRSC schemes are currently being carried out in [39]. This investigation focussed on the study of accretion phenomena onto (dynamic) rotating black holes and the associated emission of gravitational radiation induced by the presence of the matter fields. Axisymmetric studies will also provide useful ''test beds'' in forthcoming investigations with the present 3D code discussed in this paper. As will be discussed in later sections of this paper, our present code is based on the same HRSC algorithmic machinery as in the aforementioned works. We extend the treatment to 3D, and develop a code that makes no assumptions on the nature of the spacetime, the form of the metric, or the slicing and spatial coordinates. We refer the interested reader to the above references for a first understanding of the numerical schemes used in our work.

We also want to mention a completely different approach for GR-Hydro based on pseudospectral methods [40]. These methods are well known for having extraordinary accuracy in smooth regions of the solution. The numerical error is evanescent, i.e., it decreases as e^{-N} with N being the number of coefficients in the spectral expansion. The main drawback of pseudospectral methods has been, traditionally, the inaccurate modeling of discontinuous solutions due to the appearance of the so-called Gibbs phenomenon. In the presence of discontinuities, the numerical approximation of the solution does not converge at the discontinuity and spurious oscillations appear. Recently, however, an innovative pseudospectral method based on a multidomain decomposition has been developed [41] which circumvents the Gibbs phenomenon. This new approach has already been shown to work remarkably well in the 3D numerical construction of McLaurin and Roche equilibrium models.

C. Issues of general relativistic hydrodynamics

In this subsection we discuss the main issues we considered in choosing our approach, building on the existing work discussed above. The main aim of our program is to study violent and highly-energetic astrophysical processes like NS-NS coalescence within the framework of general relativity. These scenarios involve strong gravitational fields, matter motion with (ultra) relativistic speeds and/or strong shock waves. These features make the numerical integration of the hydrodynamic equations a very demanding task. The difficulty is exacerbated by the intrinsic multidimensional character of these astrophysical systems, and by the inherent complexities in Einstein theory of gravity, e.g., coordinate degrees of freedom and the possible formation of curvature singularities (e.g., collapse of matter configurations to black holes). These complications call for the use of advanced numerical methodology, a flexible code construction which allows for the use of different treatments, and a large amount of careful testbed studies. In the following we discuss these issues in more detail.

Two major issues in GR-Hydro which are purely hydrodynamical in origin are the numerical modeling of flows with large Lorentz factors and strong shock waves. In [33] it was shown that the AV based schemes have difficulties in handling ultrarelativistic velocity flows with Lorentz factors ≥ 2. As a result, [33] proposed using implicit finite difference schemes to handle the GR-Hydro equations in the ultrarelativistic regime. However, investigations during the last decade have provided increasing evidence that the most appropriate schemes to deal with ultra-relativistic flow with strong shocks are those based on (approximate or exact) Riemann solvers, i.e., HRSC schemes. These methods have high accuracy (second order or more) in regions where the flow solution is smooth, and at the same time are able to resolve discontinuities in the solution (e.g., shock waves) with little smearing. They have been extensively tested and found to be applicable in the ultra-relativistic regime (see, e.g., [42] for a recent review).

While we believe HRSC schemes may be capable of providing the technology for treating the hydrodynamic part of the evolution, the field of computational GR-Hydro still contains many issues that are as yet unexplored, especially for

cases where the relativistic fluid is coupled to a dynamical spacetime. For a fully dynamical spacetime, one major issue is the handling of the gauge degrees of freedom. This problem is exacerbated in 3D simulations without symmetry assumptions. In a general 3D problem, there is no preferred choice of gauge to restrict the metric functions as in lower dimension simulations (e.g., radial gauge and polar slicing in spherically symmetric simulations). Lagrangian coordinate systems are inappropriate for complicated 3D flows. The inevitable lower resolution in 3D simulations also makes the problem more acute. Even in vacuum spacetime studies, the choice and implementation of appropriate gauge conditions for a general dynamical evolution is a largely unexplored territory [43]. How will the gauge choices be affected by the presence of relativistic fluid flows or by the existence of strong shocks which create sharp features in the sources of the metric evolution? For example, what will be a useful gauge condition for a process like the inspiral of a NS-NS binary? These are completely open issues. In order to provide the capability to investigate these problems, the code we construct here is designed to allow arbitrary gauge conditions, making no assumptions on the lapse function or the shift vector.

Another class of problems involves the connection of the numerical integration of the hydrodynamic equations to that of the spacetime equations. What is the best set of variables to use, locally measured quantities, coordinate variables, densitized quantities or some combination? With the spacetime metric an evolved variable, there are many choices. What is the best way to connect the hydrodynamics and the spacetime finite differencing steps to achieve not only a second order accurate scheme in both space and time, but also in a way that is suitable for long term evolutions? Even in Newtonian strong field evolutions, coupling the hydrodynamic integration to the gravitational potential calculation in different ways can yield different long term behavior [44]. As a consequence of the different character of the equations governing the geometry of the spacetime and the evolution of the matter fields, the numerical methods to handle them are drastically different. What are the effects of combining different methods, and is there a best combination for a particular class of problems? With the recent development of hyperbolic formulations in GR, an interesting possibility would be to consider all of the dynamical variables, both spacetime and matter fields, to be members of one master state vector. The entire system of equations could then be written as a single (vector) conservation (or balance) equation. One could then apply the same HRSC schemes to the entire system. What advantages would this bring? These are some of the issues that we have in mind in choosing our approach in developing the code as will be discussed next.

D. Outline of our approach

Our overall goal is to develop an efficient, flexible, computational tool for general relativistic astrophysics. Specifically for this paper, the aims are (1) to establish the formulation, including the spectral decomposition of the GR-Hydro equations, on which our code is based, (2) to validate the numerical code we constructed for solving the GR-Hydro equations, and (3) to compare the different numerical schemes we used.

The set of differential equations we are attempting to solve consists of very complicated, coupled partial differential equations involving thousands of terms. Considering the complexity and generality of the code, along with the fact that the solution space of the differential equations is largely unexplored, it is essential that any physical result produced by a 3D GR-Hydro code be preceded by a series of tests such as the ones we report here, in order to insure the fidelity of the discretization to the original differential equations. In fact, we consider the tests presented here to be a minimal set: *any* 3D GR-Hydro code should be able to reproduce these results. Further tests, especially those related to the long term stability of the code and detailed comparisons of 3D and 1D results will be presented in a forthcoming paper.

In exploring the very complex system of the GR-Hydro equations, it is also essential to have the capability to compare results based on different mathematical formulations and different numerical schemes. Our code is currently set up to allow two different formulations of the Einstein equations: the standard Arnowitt-Deser-Misner (ADM) formulation [45] and the Bona-Massó (BM) hyperbolic formulation [1] (other hyperbolic formulations will be included and reported later). The code allows for two different choices for finite differencing the ADM equations: a standard leapfrog scheme and an iterative Crank-Nicholson scheme. The BM equations are finite differenced using a Strang split to separate the source and flux updates. The latter are performed using a MacCormack method. As for the numerical treatment of the hydrodynamic equations, the code has the capability of using three different HRSC schemes: the first one is the flux split method, mainly chosen for its simplicity. The second is Roe's method [46] [note that contrary to [47], we do not use Roe's averaging but instead employ arithmetic averaging (see Sec. III below for details)]. The third scheme we use is the recently developed Marquina's method [48]. All three schemes are coupled to the spacetime evolution solver in a way which is second order accurate in both space and time.

In this code we also allow for arbitrary spacetime coordinate conditions. As mentioned previously, this enables the investigation of gauge choices in GR-Hydro and allows the use of different coordinate systems for different astrophysical simulations. This capability is built into our development, and we have carried out tests with non-trivial lapses and shifts in this paper. However, more investigation is needed in this direction.

E. Computational issues

As the aim of our program is to study *realistic* astrophysical systems, which often require full 3D simulations and involve many different time and length scales, it is important that the computer code we develop be capable of carrying out large scale simulations. This requires the use of massively parallel supercomputers. The ''CACTUS'' code was built with this in mind. Here we give a brief overview of the

computational infrastructure of the code and its performance. For a more extensive review, see [49].

The CACTUS code achieves parallelism through the MPI message passing interface [50]. This allows high performance portable parallelism using a distributed memory model. All major high performance parallel architectures, including the SGI Origin 2000, Cray T3E, HP/Convex Exemplar, and IBM SP-2 support this programming model. The MPI layer of Cactus also allows computing on clusters of networked workstations and PC's. Parallelism in Cactus is based on a generic domain decomposition, distributing uniform grid functions across multiple processors and providing ghost-zone based communications for a variety of stencil widths and grid staggerings. The code can also compile without MPI, allowing the same source code to be run on a single processor workstation and on massively parallel supercomputers. The platforms currently supported and tested include: the SGI Origin 2000 (up to 256 nodes), the Cray T3E (up to 1024 nodes), SGI O2 clusters, NT clusters, DEC alphas, and SGI workstations. We have recently benchmarked a version of the code (the ''GR3D'' version, constructed for the NASA Neutron Star Grand Challenge Project, see http://wugrav.wustl.edu/Relativ/nsgc.html) on a 1024 node T3E-1200, achieving over 140 GFlop/sec and a scaling efficiency of over 95% (for details of the benchmark, see http://wugrav.wustl.edu/Codes/GR3D/). Besides the floating point and scaling efficiency, it is also noteworthy that a relatively large grid size ($644 \times 644 \times 1284$ grid points for 32 bit accuracy, and $500 \times 500 \times 996$ grid points for 64 bit accuracy) were used for the benchmarked run on the T3E-1200. This is made possible by the efficient memory usage of the code. With the full set of the Einstein equations coupled to the relativistic hydrodynamics equations, a large number of 3D arrays are required to evolve the system. In order to have reasonable resolutions for realistic simulations, it is essential that the code make efficient use of available memory. It is also essential that the code be highly optimized in order for these large simulations to be carried out in a reasonable time.

During the code development, special attention was also given to software engineering problems, such as collaborative code development, maintenance, and management. The code was developed to be shared by the entire community for the investigation of general relativistic astrophysics. To minimize barriers associated with collaborative development, the code was constructed to have: (1) A modular code structure that allows new code modules to easily plug in as ''thorns'' to the core part of the code (the ''flesh''). The ''flesh'' contains the parallel domain decomposition software, I/O, and various utilities. (2) A consistency test suite library to make sure that new thorns will not conflict with other parts of the code. (3) Various code development tools, such as: documentation, elliptic solvers, and visualization tools, which provide a complete environment for code development, and testing. For detailed discussions of these and other features of collaborative infrastructure of the code, see [51,49]

These computational features of the code significantly enhance our effort in constructing a multi-purpose code for general relativistic astrophysics.

F. Organization of this paper

The organization of the paper is as follows: the formulation of the differential equations are given in Sec. II. A spectral decomposition of the GR-Hydro equations suitable for a general non-diagonal spatial metric is presented. The details of the discretization of the equations and of the coupling of the spacetime and hydrodynamics are given in Sec. III. Shock tube tests are performed in Sec. IV for shocks along the coordinate axes and along the diagonal. These test the hydrodynamic part of the code, with the background geometry held flat. We then go on to test the coupling of the hydrodynamics to curved and dynamical spacetimes. Section V is on tests using Friedmann-Robertson-Walker cosmologies with dust. Section VI contains tests using static spherical star solutions with a polytropic equation of state. We present a practical procedure which gives stable evolution of the surface region of the star. Section VII contains tests using the spherical star solutions described in Sec. VI but now relativistically boosted along the diagonal $\hat{x} + \hat{y} + \hat{z}$. This is a strong test of the fully coupled spacetime and hydrodynamics system, with all possible terms in the equations activated and with a non-trivial lapse and shift. Finally, Sec. VIII contains a brief summary.

All tests presented in Secs. V–VII contain convergence studies performed in the following way: errors are obtained by subtracting the exact solution at a specific time from the computed solution for a number of dynamical variables. These errors are produced at three different resolutions, Δx, $\Delta x/2$, and $\Delta x/4$. To demonstrate they have the correct convergence properties for a second order accurate discretization we check that each error function decreases by a factor of four for each factor of two increase in resolution. This is demonstrated by plotting the various error functions along 1D lines. These convergence tests are an essential part in validating the code.

II. FORMULATION

A. General relativistic hydrodynamic equations

In this subsection we present the hydrodynamic equations for a general curved spacetime in a form suitable for advanced numerical treatment. The equations for the evolution of the spacetime, including the hydrodynamic source, will be presented in a later subsection.

The general relativistic hydrodynamic equations, written in the standard covariant form, consist of the local conservation laws of the stress-energy, $T^{\mu\nu}$, and the matter current density, J^{μ}

$$\nabla_\mu T^{\mu\nu} = 0, \qquad (1)$$

$$\nabla_\mu J^{\mu} = 0, \qquad (2)$$

where $J^{\mu} = \rho u^{\mu}$, ρ is the rest mass density and u^{μ} the 4-velocity of the fluid. ∇_μ stands for the covariant derivative with respect to the 4-metric of the underlying spacetime. Throughout this paper we are using, unless otherwise stated, natural units ($G = c = 1$). Greek (Latin) indices run from 0 to

3 (1 to 3). In what follows we will neglect viscous effects, assuming the stress-energy tensor to be that of a perfect fluid

$$T^{\mu\nu} = \rho h u^{\mu} u^{\nu} + P g^{\mu\nu}, \tag{3}$$

where P is the fluid pressure and $g^{\mu\nu}$ is the 4-metric describing the spacetime. In addition, the relativistic specific enthalpy, h, is given by

$$h = 1 + \epsilon + P/\rho, \tag{4}$$

where ϵ is the rest frame specific internal energy density of the fluid.

The equations written in this covariant form are *not* suitable for the use of advanced numerical schemes. In order to carry out numerical hydrodynamic evolutions, and in particular to take advantage of the benefits of HRSC methods, the hydrodynamic equations after the $3+1$ split must be written as a hyperbolic system of first order flux conservative equations. We introduce coordinates $(x^0 = t, x^1, x^2, x^3)$ and write Eqs. (1) and (2) in terms of coordinate derivatives. We project Eq. (1) and Eq. (2) onto the basis $\{n^{\mu}, (\partial/\partial x^i)^{\mu}\}$, with n^{μ} being a timelike vector normal to a given hypersurface. A straightforward calculation yields the set of equations in the desired form

$$\partial_t \vec{\mathcal{U}} + \partial_i \vec{F}^i = \vec{S}, \tag{5}$$

where ∂_t denotes a partial derivative with respect to time and ∂_i indicates a partial derivative with respect to the spatial coordinate x^i.

The evolved state vector $\vec{\mathcal{U}}$ is given, in terms of the primitive variables (ρ, v^i, ϵ), as

$$\vec{\mathcal{U}} = \begin{bmatrix} \tilde{D} \\ \tilde{S}_j \\ \tilde{\tau} \end{bmatrix} = \begin{bmatrix} \sqrt{\gamma} W \rho \\ \sqrt{\gamma} \rho h W^2 v_j \\ \sqrt{\gamma}(\rho h W^2 - P - W\rho) \end{bmatrix}, \tag{6}$$

where γ is the determinant of the 3-metric γ_{ij}, v_j is the fluid 3-velocity, and W is the Lorentz factor, $W = \alpha u^0 = (1 - \gamma_{ij} v^i v^j)^{-1/2}$. Notice that the spatial components of the 4-velocity u^i are related to the 3-velocity by the following formula:

$$u^i = W(v^i - \beta^i/\alpha), \tag{7}$$

where α and β^i are, respectively, the lapse function and the shift vector of the spacetime. Also notice that we are using a slightly different set of variables as those used in [38]. We are now ''densitizing'' the evolved quantities, D, S_j and τ, with the factor $\sqrt{\gamma}$. The three flux vectors $\vec{F}^i$ are given by

$$\vec{F}^i = \begin{bmatrix} \alpha\left(v^i - \frac{1}{\alpha}\beta^i\right)\tilde{D} \\ \alpha\left(\left(v^i - \frac{1}{\alpha}\beta^i\right)\tilde{S}_j + \sqrt{\gamma}P\,\delta^i_j\right) \\ \alpha\left(\left(v^i - \frac{1}{\alpha}\beta^i\right)\tilde{\tau} + \sqrt{\gamma}v^i P\right) \end{bmatrix}. \tag{8}$$

Finally, the source vector $\vec{S}$ is given by

$$\vec{S} = \begin{bmatrix} 0 \\ \alpha\sqrt{\gamma}T^{\mu\nu}g_{\nu\sigma}\Gamma^{\sigma}_{\ \mu j} \\ \alpha\sqrt{\gamma}(T^{\mu 0}\partial_{\mu}\alpha - \alpha T^{\mu\nu}\Gamma^0_{\ \mu\nu}) \end{bmatrix}, \tag{9}$$

where $\Gamma^{\alpha}_{\ \mu\nu}$ is the 4-Christoffel symbol

$$\Gamma^{\alpha}_{\ \mu\nu} = \frac{1}{2}g^{\alpha\beta}(\partial_{\mu}g_{\nu\beta} + \partial_{\nu}g_{\mu\beta} - \partial_{\beta}g_{\mu\nu}). \tag{10}$$

A technical point must be included here. While the numerical code updates the state vector $\vec{\mathcal{U}}$ forward in time it makes use, internally, of the set of primitive variables defined above, (ρ, v^i, ϵ). Those are used throughout, e.g., in the computation of the characteristic fields (see below). These variables cannot be obtained from the evolved ones in a closed functional form. Instead, they must be recovered through some appropriate root-finding procedure (an example of this can be found in [52]).

B. Spectral decomposition and characteristic fields

The use of HRSC schemes, as will be presented in detail in the next section, depends crucially on the knowledge of the spectral decomposition of the Jacobian matrix of the system

$$\frac{\partial \vec{F}^i}{\partial \vec{\mathcal{U}}}. \tag{11}$$

The characteristic speeds (eigenvalues) and fields (eigenvectors) are the key ingredients of any HRSC scheme. The spectral decomposition of the Jacobian matrices of the general relativistic hydrodynamic equations with general equation of state was first reported in [38] (for polytropic EOS see [47]). However, we have found that the eigenvectors reported in [38] are correct only in the case of a diagonal spatial metric. In this section we display the full spectral decomposition valid for a generic spatial metric. We focus on the x direction, hence presenting the spectral decomposition of $(\partial \vec{F}^x / \partial \vec{\mathcal{U}})$, as the other two directions can be found by simple permutation of indices.

We start by considering an equation of state in which the pressure P is a function of ρ and ϵ, $P = P(\rho, \epsilon)$. The relativistic speed of sound in the fluid c_s is given by (see, e.g., [53])

$$c_s^2 = \left.\frac{\partial P}{\partial E}\right|_S = \frac{\chi}{h} + \frac{P}{\rho^2}\frac{\kappa}{h}, \tag{12}$$

where $\chi = \partial P/\partial\rho|_\epsilon$, $\kappa = \partial P/\partial\epsilon|_\rho$, $\mathcal{S}$ is the entropy per particle, and E is the total rest energy density which in our case is $E = \rho + \rho\epsilon$. We require a complete set of eigenvectors $[\vec{r}_i]$ and corresponding eigenvalues λ_i along the x direction, i.e.

$$\left[\frac{\partial \vec{F}^x}{\partial \vec{\mathcal{U}}}\right][\vec{r}_i] = \lambda_i[\vec{r}_i], \quad i = 1, \ldots, 5. \tag{13}$$

The solution contains a triply degenerate eigenvalue

$$\lambda_1 = \lambda_2 = \lambda_3 = \alpha v^x - \beta^x. \tag{14}$$

A set of linearly independent vectors that span this degenerate space is given by

$$\vec{r}_1 = \left[\frac{\kappa}{hW(\kappa - \rho c_s^2)}, v_x, v_y, v_z, 1 - \frac{\kappa}{hW(\kappa - \rho c_s^2)}\right]^T, \tag{15}$$

$$\vec{r}_2 = [Wv_y, h(\gamma_{xy} + 2W^2 v_x v_y), h(\gamma_{yy} + 2W^2 v_y v_y), h(\gamma_{yz} + 2W^2 v_y v_z), v_y W(2Wh - 1)]^T, \tag{16}$$

$$\vec{r}_3 = [Wv_z, h(\gamma_{xz} + 2W^2 v_x v_z), h(\gamma_{yz} + 2W^2 v_y v_z), h(\gamma_{zz} + 2W^2 v_z v_z), v_z W(2Wh - 1)]^T. \tag{17}$$

The superscript T denotes transpose. The other two eigenvalues are given by

$$\lambda_\pm = \frac{\alpha}{1 - v^2 c_s^2}\{v^x(1 - c_s^2) \pm \sqrt{c_s^2(1 - v^2)[\gamma^{xx}(1 - v^2 c_s^2) - v^x v^x(1 - c_s^2)]}\} - \beta^x, \tag{18}$$

with corresponding eigenvectors

$$\vec{r}_\pm = \left[1, hW\left(v_x - \frac{v^x - (\lambda_\pm + \beta^x)/\alpha}{\gamma^{xx} - v^x(\lambda_\pm + \beta^x)/\alpha}\right), hWv_y, hWv_z, \frac{hW(\gamma^{xx} - v^x v^x)}{\gamma^{xx} - v^x(\lambda_\pm + \beta^x)/\alpha} - 1\right]^T. \tag{19}$$

C. Equations for a dynamical spacetime with a hydrodynamic source

The dynamics of the gravitational field in general relativity theory is described by Einstein's field equations

$$G_{\mu\nu} = 8\pi T_{\mu\nu}, \tag{20}$$

which relate the (ten) metric components $g_{\mu\nu}(= g_{\nu\mu})$ of the spacetime to the stress energy tensor $T_{\mu\nu}$. Here, $G_{\mu\nu}$ is the Einstein tensor which involves second derivatives, in both space and time, of the dependent variables $g_{\mu\nu}$. A formulation of the Einstein equations suitable for numerical evolutions has been known for more than three decades [45]. In recent years, many new formulations have been proposed (for a review, see [54,55] and references therein) seeking to expose the hyperbolicity of the evolution components of the Einstein field equations.

In the present paper, we discuss the mathematical and algorithmic issues related to the coupling of the hydrodynamic equations to two different formulations of the Einstein equations. We start with the more commonly used ADM formulation [45]. Then we discuss the BM hyperbolic formulation of the Einstein equations [1].

1. Arnowitt-Deser-Misner formulation

In the ADM formulation [45], spacetime is foliated into a set of non-intersecting spacelike hypersurfaces. There are two kinematic variables which describe the evolution between these surfaces: the lapse function α, which describes the rate of advance of time along a timelike unit vector n^μ normal to a surface, and the spacelike shift vector β^i that describes the motion of coordinates. The line element is written as

$$ds^2 = -(\alpha^2 - \beta_i\beta^i)dt^2 + 2\beta_i dx^i dt + \gamma_{ij}dx^i dx^j. \tag{21}$$

The ADM formulation casts the Einstein equations into a first order (in time) quasi-linear [56] system of equations. The dependent variables are the 3-metric γ_{ij} and the extrinsic curvature K_{ij}. The evolution equations read

$$\partial_t\gamma_{ij} = -2\alpha K_{ij} + \nabla_i\beta_j + \nabla_j\beta_i, \tag{22}$$

$$\partial_t K_{ij} = -\nabla_i\nabla_j\alpha + \alpha\left[R_{ij} + K\ K_{ij} - 2K_{im}K_j^m\right.$$
$$\left. - 8\pi\left(S_{ij} - \frac{1}{2}\gamma_{ij}S\right) - 4\pi\rho_{ADM}\gamma_{ij}\right]$$
$$+ \beta^m\nabla_m K_{ij} + K_{im}\nabla_j\beta^m + K_{mj}\nabla_i\beta^m, \tag{23}$$

where ∇_i denotes a covariant derivative with respect to the 3-metric γ_{ij} and R_{ij} is the Ricci curvature of the 3-metric.

In addition to the evolution equations, $\gamma_{ij}, K_{ij}, \rho_{ADM}$, and j^i must satisfy the Hamiltonian constraint

$$^{(3)}R + K^2 - K_{ij}K^{ij} - 16\pi\rho_{ADM} = 0, \tag{24}$$

and the momentum constraints

$$\nabla_j K^{ij} - \gamma^{ij}\nabla_j K - 8\pi j^i = 0. \tag{25}$$

Here, $\rho_{ADM}, j^i, S_{ij}, S = \gamma^{ij}S_{ij}$ are the components of the stress energy tensor projected onto the 3D hypersurface (for a more detailed discussion, see [57]). In this paper we use the stress-energy tensor of a perfect fluid [Eq. (3)]. Hence, explicitly in terms of the primitive hydrodynamic variables,

$$\rho_{ADM} = \rho h W^2 - P, \tag{26}$$

$$j^i = \rho h W^2 v^i, \tag{27}$$

$$S_{ij} = \rho h W^2 v_i v_j + \gamma_{ij}P, \tag{28}$$

$$S = \rho h W^2 v_i v^i + 3P. \tag{29}$$

2. Bona-Massó hyperbolic formulation

In the BM hyperbolic formulation [58,1], the evolution equations are written as a first order balance law with, formally, the same mathematical structure as the hydrodynamic equations, Eq. (5). Now, the state vector, containing the evolved quantities for the spacetime, has the following 37 components:

$$\vec{\mathcal{U}} = (\gamma_{ij}, \alpha, K_{ij}, D_{kij}, A_k, V_k), \tag{30}$$

where

$$D_{kij} = \frac{1}{2}\partial_k\gamma_{ij}, \tag{31}$$

$$A_k = \partial_k\ln\alpha, \tag{32}$$

$$V_i = D^j_{ij} - D^j_{ji}. \tag{33}$$

We use the following notation to describe the fluxes; $F^k_-\gamma_{ij}$ denotes the k component of the flux in the evolution equation for γ_{ij}. Using this notation, the fluxes are

$$F^k_-\gamma_{ij} = 0, \tag{34}$$

$$F^k_-\alpha = 0, \tag{35}$$

$$F^k_-K_{ij} = -\beta^k K_{ij} + \alpha[D^k_{ij} - n/2V^k\gamma_{ij} + 1/2\delta^k_i(A_j + 2V_j$$
$$- D^r_{jr}) + 1/2\delta^k_j(A_i + 2V_i - D^r_{ir})], \tag{36}$$

$$F^k_-D_{kij} = -\beta^r D_{rij} + \alpha(K_{ij} - s_{ij}), \tag{37}$$

$$F^k_-A_k = -\beta^r A_r + \alpha Q, \tag{38}$$

$$F^k_-V_i = -\beta^k V_i + B^k_i - B^k_i, \tag{39}$$

where

$$B^i_k = \frac{1}{2}\partial_k\beta^i, \tag{40}$$

is calculated from the user supplied shift vector. Finally, the source terms read

$$S_-\gamma_{ij} = -2\alpha(K_{ij} - s_{ij}) + 2\beta^r D_{rij}, \tag{41}$$

$$S_-\alpha = -\alpha^2 Q + \alpha\beta^r A_r, \tag{42}$$

$$S_-K_{ij} = 2(K_{ir}B^r_j + K_{jr}B^r_i - K_{ij}B^r_r) + \alpha[-2K^k_i K_{kj} + \text{tr}\, KK_{ij} - \Gamma^k_{ri}\Gamma^r_{kj} + 2D^r_{ik}D^k_{rj} + 2D^r_{jk}D^k_{ri} + \Gamma^k_{kr}\Gamma^r_{ij} - (2D^k_{kr} - A_r)(D^r_{ij} + D^r_{ji})$$
$$+ A_i(V_j - 1/2D^k_{jk}) + A_j(V_i - 1/2D^k_{ik}) + A_j(V_i - 1/2D^k_{ik}) - nV^k D_{kij} - 4\pi(2S_{ij} - \gamma_{ij}[S + p + (n-1)\rho_{ADM}])]$$
$$+ n/4\alpha\gamma_{ij}[-D^{rs}_k\Gamma^k_{rs} + D^r_{kr}D^{ks}_s - 2V^k A_k + K^{rs}K_{rs} - (\text{tr}\, K)^2], \tag{43}$$

$$S_-D_{kij} = 0, \tag{44}$$

$$S_-A_k = 0, \tag{45}$$

$$S_-V_i = \alpha[8\pi j_i + A_r(K^r_i - \text{tr}\, K\delta^r_i) + K^r_s(D^s_{ir} - 2D^s_{ri}) - K^r_i(D^s_{rs} - 2D^s_{sr})] + 2(B^r_i - \delta^r_i\text{tr}\, B)V_r + 2(D^s_{ri} - \delta^s_i D^j_{jr})B^r_s, \tag{46}$$

where

$$s_{ij} = (B_{ij} + B_{ji})/\alpha, \tag{47}$$

is used to simplify the equations. The free parameter n allows one to select a specific evolution system (it is zero for the ''Ricci'' system and one for the ''Einstein'' system) as discussed in [58].

III. DISCRETIZATION OF THE EQUATIONS

A. Modern HRSC schemes for the GR-Hydro equations

As stated previously, the main aim of this work is to confirm the *consistency* of the coded finite difference equations with the partial differential equations and the *convergence* of three independent discretizations of the GR-Hydro equations. All three approaches are based on finite-difference

schemes employing HRSC schemes to account, explicitly, for the characteristic information of the equations. The methods considered are a flux split method [61], Roe's approximate Riemann solver [46], and Marquina's recently developed scheme [48].

To simplify the discussion, let us examine the update for the state vector $\vec{\mathcal{U}}$ for a flux in the x direction:

$$\frac{\partial \vec{\mathcal{U}}}{\partial t} + \frac{\partial \vec{F}^x}{\partial x} = 0. \tag{48}$$

The discretization of this equation takes the form

$$\frac{\partial \vec{\mathcal{U}}_i}{\partial t} + \frac{(\vec{f}^*)_{i+1/2} - (\vec{f}^*)_{i-1/2}}{\Delta x} = 0, \tag{49}$$

where $(\vec{f}^*)_{i\pm 1/2}$ is the "numerical flux" function calculated at the interfaces $i\pm 1/2$ of the spatial cell i. The different methods we are using simply differ in the way the numerical fluxes are calculated. The way the source terms are integrated in time is explained later in this section. We point out that, although we are interested in the specific application of the fluxes in Eq. (8), the methods presented here can be used with any advection equation, provided one has the spectral decomposition of the fluxes in hand.

1. Flux split method

The first scheme is a flux split method, where the flux is decomposed into the part contributing to the eigenfields with positive eigenvalues (fields moving to the right) and the part with negative eigenvalues (fields moving to the left). These fluxes are then discretized with one sided derivatives whose side depends on the sign of the particular eigenvalue.

For the flux split method, one makes the assumption that

$$\vec{F}^x(\sigma \vec{\mathcal{U}}) = \sigma \vec{F}^x(\vec{\mathcal{U}}), \tag{50}$$

for any constant σ. This is only true for the fluxes of Eq. (8) if the equation of state has the following functional form:

$$P = P(\rho, \epsilon) = \rho f(\epsilon), \tag{51}$$

for some function $f(\epsilon)$. For the flux split method we therefore assume the equation of state to be in the following form:

$$P = (\Gamma - 1)\rho \epsilon, \tag{52}$$

with Γ being the (constant) adiabatic index of the fluid. It is easy to show that, under the above assumptions, the flux vector $\vec{F}^x$ can be written

$$\vec{F}^x = \left(\frac{\partial \vec{F}^x}{\partial \vec{\mathcal{U}}}\right)\vec{\mathcal{U}}. \tag{53}$$

Using the spectral decomposition of Sec. II B one can express the Jacobian matrix $\partial \vec{F}^x / \partial \vec{\mathcal{U}}$ as

$$\frac{\partial \vec{F}^x}{\partial \vec{\mathcal{U}}} = M\Lambda M^{-1}, \tag{54}$$

where M is the matrix whose columns are the right eigenvectors of the system and Λ is a diagonal matrix constructed from the corresponding eigenvalues (see Sec. II B).

Given the characteristic information, we can now split the flux into the part that is moving to the right and the part that is moving to the left:

$$\vec{F}^x = (\vec{F}^x)^+ + (\vec{F}^x)^- = (M\Lambda^+ M^{-1})\vec{\mathcal{U}} + (M\Lambda^- M^{-1})\vec{\mathcal{U}}, \tag{55}$$

where $\Lambda^+ = \frac{1}{2}(\Lambda + |\Lambda|)$, and $\Lambda^- = \frac{1}{2}(\Lambda - |\Lambda|)$. The numerical flux which corresponds to an upwind method (first order in space) is then simply

$$(\vec{f}^*_{i+1/2})_{\text{first order}} = (\vec{F}^x)^+_i + (\vec{F}^x)^-_{i+1}. \tag{56}$$

One could attempt to construct a numerical flux based on one-sided derivatives that were second order accurate in space:

$$(\vec{f}^*_{i+1/2})_{\text{second order–non TVD}}$$

$$= (\vec{f}^*_{i+1/2})_{\text{first order}} + \frac{1}{2}(\vec{F}_i - (\vec{f}^*_{i-1/2})_{\text{first order}}) \tag{57}$$

$$+ \frac{1}{2}(\vec{F}_{i+1} - (\vec{f}^*_{i+3/2})_{\text{first order}}). \tag{58}$$

However, the method would not have the (numerically desirable) total variation diminishing (TVD) property (see, e.g. [32] for definition) unless flux limiters are used in front of the second order correction terms:

$$(\vec{f}^*_{i+1/2})_{\text{second order}}$$

$$= (\vec{f}^*_{i+1/2})_{\text{first order}} + \frac{1}{2}\psi^+_{i-1/2}(\vec{F}_i - (\vec{f}^*_{i-1/2})_{\text{first order}}) \tag{59}$$

$$+ \frac{1}{2}\psi^-_{i+3/2}(\vec{F}_{i+1} - (\vec{f}^*_{i+3/2})_{\text{first order}}), \tag{60}$$

with

$$\psi^+_{i+1/2} = \psi\left(\frac{\vec{F}^x_{i+2} - (\vec{f}^*_{i+3/2})}{\vec{F}^x_{i+1} - (\vec{f}^*_{i+1/2})}\right), \tag{61}$$

$$\psi^-_{i+1/2} = \psi\left(\frac{\vec{F}^x_{i-1} - (\vec{f}^*_{i-1/2})}{\vec{F}^x_i - (\vec{f}^*_{i+1/2})}\right), \tag{62}$$

where we are using van Leer flux limiters [59]

$$\psi(\sigma) = \frac{\sigma + |\sigma|}{1 + \sigma}. \tag{63}$$

2. Roe's method

The second scheme we use to integrate the hydrodynamic equations makes use of Roe's approximate Riemann solver [46]. This is undoubtedly the most established method for the accurate integration of non-linear hyperbolic systems of conservation laws. The suitability of Roe's method for the relativistic hydrodynamic equations have been shown in [60,47,38]. This method makes no assumption on the equation of state, and, in this respect, is more flexible than the flux split methods. As the method is well documented in the literature (see, e.g., [61]) it will only be briefly outlined here. As mentioned in the Introduction, all simulations reported in this paper using Roe's scheme are performed employing arithmetic averages. For the use of Roe's averaging in GR-Hydro see [47].

A monotonic piecewise-constant (piecewise-linear) reconstruction of the cell centered values of the primitive variables to the cell interfaces provides first-order (second-order) accuracy in space [59]. In order to get second-order convergence we have implemented a standard minmod piecewise-linear reconstruction algorithm [59]. The numerical fluxes across interfaces are calculated according to

$$(\vec{f}^*)_{i+1/2} = \frac{1}{2}\left[\vec{F}_R^x + \vec{F}_L^x - \sum_{m=1}^{5} \left|\tilde{\lambda}_m\right| \Delta\tilde{\omega}_m \tilde{\vec{r}}_m\right], \quad (64)$$

where R and L indicate the right and left sides of a cell interface. In addition, $\{\tilde{\lambda}_m, \tilde{\vec{r}}_m\}_{m=1,\ldots,5}$ are, respectively, the eigenvalues and right-eigenvectors of the Jacobian matrix of the system calculated at the cell interfaces as the arithmetic mean of the left and right reconstructed (interpolated) primitive variables. Averaged quantities in Eq. (64) are identified by a "tilde." Finally, the quantities $\{\Delta\tilde{\omega}_n\}_{n=1,\ldots,5}$, the jumps of the characteristic variables across each characteristic field, are obtained from

$$\vec{\mathcal{U}}_R - \vec{\mathcal{U}}_L = \sum_m \Delta\tilde{\omega}_m \tilde{\vec{r}}_m. \quad (65)$$

3. Marquina's Method

In [48] Donat and Marquina proposed a new flux formula to compute the numerical flux at a cell interface. The new flux formula has a clear flux splitting structure, and leads to an upstream scheme. The novelty of Marquina's approach lies in the extension of Shu and Osher's entropy-satisfying numerical flux [62] to systems of hyperbolic conservation laws. In this scheme there are no artificial intermediate states constructed at each cell interface. This implies that there are no Riemann solutions involved (either exact or approximate); moreover, the scheme has been proven to alleviate several numerical pathologies associated to the introduction of an averaged state (as Roe's method does) in the local diagonalization procedure (see [48,63]).

To compute the numerical flux at a given interface, separating the states $\vec{\mathcal{U}}_L$ and $\vec{\mathcal{U}}_R$, we compute first the sided local characteristic variables and fluxes:

$$\omega_l^p = \vec{l}^p(\vec{\mathcal{U}}_l) \cdot \vec{\mathcal{U}}_l, \quad \phi_l^p = \vec{l}^p(\vec{\mathcal{U}}_l) \cdot \vec{f}(\vec{\mathcal{U}}_l),$$

$$\omega_r^p = \vec{l}^p(\vec{\mathcal{U}}_r) \cdot \vec{\mathcal{U}}_r, \quad \phi_r^p = \vec{l}^p(\vec{\mathcal{U}}_r) \cdot \vec{f}(\vec{\mathcal{U}}_r), \quad (66)$$

for $p = 1,2\ldots,5$. Here $\vec{l}^p(\vec{\mathcal{U}}_l)$, $\vec{l}^p(\vec{\mathcal{U}}_r)$, are the (normalized) left eigenvectors of the Jacobian matrices of the system. Let $\lambda_1(\vec{\mathcal{U}}_l), \ldots, \lambda_5(\vec{\mathcal{U}}_l)$ and $\lambda_1(\vec{\mathcal{U}}_r), \ldots, \lambda_5(\vec{\mathcal{U}}_r)$ be their corresponding eigenvalues. For $k = 1, \ldots, 5$ the procedure is the following:

If $\lambda_k(\vec{\mathcal{U}})$ does not change sign in $[\vec{\mathcal{U}}_l, \vec{\mathcal{U}}_r]$, then the scheme is upwind.

If $\lambda_k(\vec{\mathcal{U}}_l) > 0$ then

$$\phi_+^k = \phi_l^k,$$

$$\phi_-^k = 0, \quad (67)$$

else

$$\phi_+^k = 0,$$

$$\phi_-^k = \phi_r^k. \quad (68)$$

Otherwise, the scheme is switched to the more viscous, entropy-satisfying, local-Lax-Friedrichs scheme

$$\alpha_k = \max|\lambda_k(\vec{\mathcal{U}})|, \quad \vec{\mathcal{U}} \in \Gamma(\vec{\mathcal{U}}_l, \vec{\mathcal{U}}_r),$$

$$\phi_+^k = .5(\phi_l^k + \alpha_k\omega_l^k),$$

$$\phi_-^k = .5(\phi_r^k - \alpha_k\omega_r^k), \quad (69)$$

$\Gamma(\vec{\mathcal{U}}_l, \vec{\mathcal{U}}_r)$ is a curve in phase space connecting $\vec{\mathcal{U}}_l$ and $\vec{\mathcal{U}}_r$. In addition, α_k can be determined as

$$\alpha_k = \max\{|\lambda_k(\vec{\mathcal{U}}_l)|, |\lambda_k(\vec{\mathcal{U}}_r)|\}. \quad (70)$$

Marquina's flux formula is then

$$(\vec{f}^*)_{i+1/2}^n = \sum_{p=1}^{m} (\phi_+^p \vec{r}^p(\vec{\mathcal{U}}_l) + \phi_-^p \vec{r}^p(\vec{\mathcal{U}}_r)), \quad (71)$$

where, $\vec{r}^p(\vec{\mathcal{U}}_l)$, $\vec{r}^p(\vec{\mathcal{U}}_r)$, are the right (normalized) eigenvectors of the system. For further technical information about this solver we refer the reader to [48]. The suitability of this scheme for the accurate integration of the hydrodynamic equations and many of its desirable properties can be found in [48] (Newtonian hydrodynamics) and [63,64] (relativistic hydrodynamics).

B. Discretization techniques for the spacetime and spacetime-hydrodynamics coupling

1. Spacetime discretization

In this section we outline the discretization techniques used in the vacuum spacetime part of the code. For a more detailed discussion we refer the reader to [1]. Here we give the essential formulas for completeness and discuss in detail

TABLE I. This table summarizes the abbreviations used for the various methods used for the spacetime and hydrodynamical evolutions.

Abbreviation	Spacetime formulation/evolution scheme	Hydrodynamics update method
ADMLEAP_ROE	ADM / leapfrog	Roe
ADMLEAP_FLUX	ADM / leapfrog	flux split
ADMLEAP_MAR	ADM / leapfrog	Marquina
ADMICN_ROE	ADM / iterative Crank-Nicholson	Roe
ADMICN_FLUX	ADM / iterative Crank-Nicholson	flux split
ADMICN_MAR	ADM / iterative Crank-Nicholson	Marquina
BMEIN_ROE	BM (Einstein) / MacCormack	Roe
BMEIN_FLUX	BM (Einstein) / MacCormack	flux split
BMEIN_MAR	BM (Einstein) / MacCormack	Marquina
BMRIC_ROE	BM (Ricci) / MacCormack	Roe
BMRIC_FLUX	BM (Ricci) / MacCormack	flux split
BMRIC_MAR	BM (Ricci) / MacCormack	Marquina

only the issues relevant to its coupling to hydrodynamics described in the next subsection.

The BM system uses the so-called Strang splitting [65] to separate Eq. (5) into two evolution steps. In the first step, only the source terms are used to update the variables

$$\partial_t \vec{\mathcal{U}} = \vec{S}, \tag{72}$$

while in the second step, only the flux terms are used for the update

$$\partial_t \vec{\mathcal{U}} + \partial_i \vec{F}^i = 0. \tag{73}$$

To ensure second order accuracy in both space and time, this is done by first evolving the source terms forward in time half a time step, then evolving with only the flux terms a full time step, and finally evolving with only the source terms another half time step. The source terms are evolved forward using a second order accurate predictor-corrector method, while the flux terms are evolved using a second order accurate MacCormack scheme. Specific details of these methods are discussed in [1].

The ADM system supports the use of several different numerical schemes. Currently, a leapfrog (non-staggered in time) and iterative Crank-Nicholson scheme have been coupled to the hydrodynamic solver.

The leapfrog method assumes that all variables exist on both the current time step n and the previous time step $n-1$. Variables are updated from $n-1$ to $n+1$ (future time) evaluating all terms in the evolution equations on the current time step n.

The iterative Crank-Nicholson solver first evolves the data from the current time step n to the future time step $n+1$ using a forward in time, centered in space (FTCS) first

order method. The solution at steps n and $n+1$ are then averaged to obtain the solution on the half time step $n+\frac{1}{2}$. This solution at the half time step $n+\frac{1}{2}$ is then used in a leapfrog step to re-update the solution at the final time step $n+1$. This process is then iterated. The error is defined as the difference between the current and previous solutions on the half time step $n+\frac{1}{2}$. This error is summed over all grid points and all evolved variables. This process is repeated until some desired tolerance is reached. Care is taken to make sure that at least two iterations are taken to make the process second order accurate.

2. Spacetime-hydrodynamics coupling

Our code evolves the spacetime geometry and the matter fields separately. This allows different methods to be used for each system (spacetime and hydrodynamics). The coupling of those different evolution algorithms in a way that is second order accurate in both space and time is highly method dependent. We will therefore discuss the coupling of each system, ADM or BM, with hydrodynamics, separately. A summary of the different combined schemes appears in Table I.

The coupling between the BM system (for both the ''Einstein'' (BMEIN) and ''Ricci'' (BMRIC) systems) with the hydrodynamic solver is fairly straightforward as both systems of equations take a similar (first-order flux-conservative) form. The steps involved in the coupling are outlined in Fig. 1. In step 1 we simultaneously update the spacetime and hydrodynamic variables with the source terms via a two-step predictor-corrector scheme (second order accurate in time) to the half-time step $n+1/2$. In step 2, the spacetime variables are updated with the flux terms using a MacCormack scheme (second order accurate in time) again to the half-time step $n+1/2$. In step 3, we update the hydro-
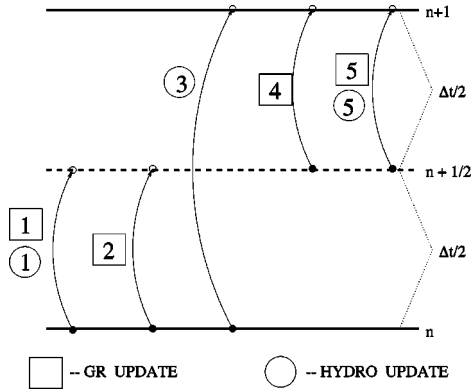
FIG. 1. This figure represents the coupling between the hydrodynamics and the MacCormack evolution scheme with the BM formulation of the field equations (either BMEIN or BMRIC). STEP 1: Simultaneous update of the spacetime and hydrodynamic variables with the source terms via a two-step predictor-corrector scheme to the half-time step $n+1/2$. STEP 2: Update of the spacetime variables with the flux terms via a MacCormack scheme to the half-time step $n+1/2$. STEP 3: Update of the hydrodynamic variables with the flux terms via a two-step predictor-corrector scheme to the $n+1$ step. STEP 4: Same as step 2 but the update is to the final time $n+1$. STEP 5: Same as step 1 but updating both sets of variables to the final $n+1$ step.
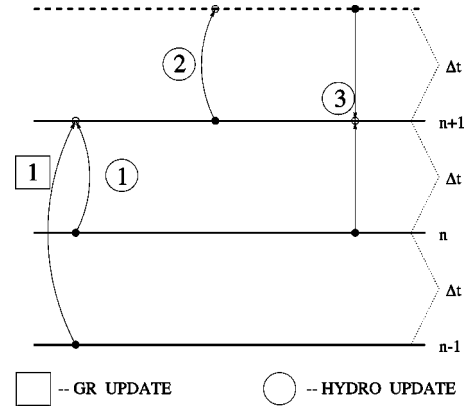

FIG. 2. This figure represents the coupling between the hydrodynamics and the leapfrog evolution scheme used for the ADM spacetime equations (ADMLEAP). STEP 1: Simultaneous update of the ADM equations via a leapfrog step (second order accurate in time) and of the hydrodynamic equations via a (first order accurate in time) Euler-predictor step using the method of lines. STEP 2: Update of the hydrodynamic equations to a virtual $n+2$ time step via a first order in time Euler-corrector step. STEP 3: A second order accurate in time update of the hydrodynamic variables is obtained by averaging the corrected quantities obtained in step 2 with the original state of step n.

dynamic variables with the flux terms via a two-step predictor-corrector scheme to the $n+1$ step. In step 4 the spacetime variables are updated with the flux terms via a MacCormack scheme to the $n+1$ step. Finally, in step 5, the spacetime and hydrodynamic variables are simultaneously updated with the source terms via a two-step predictor-corrector scheme to the final $n+1$ step.

In Fig. 2 we display the coupling between the ADM leapfrog evolution (ADMLEAP) and the hydrodynamical evolution. In step 1 we simultaneously update the ADM equations via a leapfrog step (second order accurate in time) and update the hydrodynamic equations with an Euler-predictor step (first order in time) using the method of lines. In step 2, we update the fluid variables to a virtual $n+2$ time step with a first order in time Euler-corrector step using the method of lines. Finally, in step 3, we obtain a second order accurate in time update of the hydrodynamic variables by averaging the corrected quantities obtained in step 2 with the original state of step n.

Our last combination appears plotted in Fig. 3. Here we display the coupling between the iterative Crank-Nicholson evolution scheme for the ADM equations (ADMICN) and the hydrodynamical evolution. First, in step 1, we simultaneously update the ADM and hydrodynamic equations using an Euler-predictor step, which is first-order order accurate in time, to the half time step $n+1/2$. In step 2 through M, we update the ADM equations via an iterative Crank-Nicholson scheme (second order in time) to the $n+1$ time step and average the $n+1$ and n states to produce a corrected $n+1/2$ state. The solution is guaranteed to be second order accurate (in time) for $M \geq 2$. In step M+1 we simultaneously update the ADM equations via a leapfrog step (second order in time) based on the n and $n+1/2$ states and the hydrody-

namic equations via the second half of the Euler-predictor step (first half applied in step 1) using a method of lines. In step M+2 the hydrodynamic equations are updated to a virtual $n+2$ time step via an Euler-corrector step using a method of lines. Finally, in step M+3 we obtain a (second order in time) hydrodynamics update by averaging the corrected variables obtained in step $M+2$ and the original state of step n.

IV. SHOCK TUBE TESTS

We start testing the code with one of the standard tests in fluid dynamics, the shock tube (see Refs. [52,33,66,63] for a sample of previous relativistic simulations). In this test, the fluid initially has two different thermodynamical states on either side of an interface. When this interface is removed, the fluid evolves in such a way that four states appear. Each state is separated by one of three elementary waves: a shock wave, a contact discontinuity, and a rarefaction wave. This time-dependent problem has an exact solution to which our numerical integration can be compared. This problem only checks the hydrodynamical part of the code, as it assumes a flat background metric. However, it provides a good test of the shock capturing properties of any HRSC scheme. The integration of the hydrodynamic equations in each of the three spatial directions can be tested independently by placing the initial discontinuity along each of the coordinate axes or in the fully multidimensional case when the interface is placed along the main diagonal of the computational domain.

The initial state of the fluid is specified by $P_L = 13.3$, $\rho_L = 10$ on the left side of the interface and $P_R = 0.66 \times 10^{-6}$, $\rho_R = 1$ on the right side. The local sound speed in the high density region is only moderately relativistic, $c_s = 0.72$. The fluid is assumed to be initially at rest on both sides of the
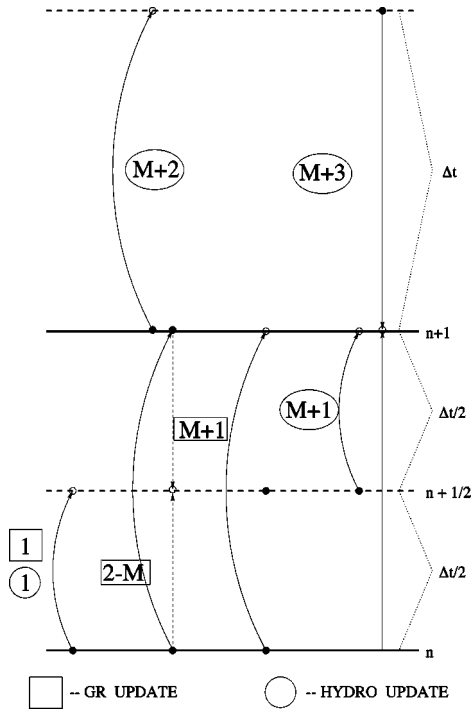
FIG. 3. This figure represents the coupling between the hydrodynamic evolution scheme and the iterative Crank-Nicholson method used for the integration of the ADM equations (ADMICN). STEP 1: Simultaneous update of the ADM and hydrodynamic equations via a Euler-predictor step (first order in time) to the half time step $n+1/2$. STEP 2 through M: Update of the ADM equations via an iterative Crank-Nicholson scheme (second order accurate in time) to the $n+1$ time step, then compute a corrected $n+1/2$ state by averaging the $n+1$ and n states. STEP M+1: Simultaneous update of the ADM equations via a leapfrog step (second order in time) based on the n and $n+1/2$ states and the hydrodynamics equations via the second half of the Euler-predictor step (first half applied in step 1) using a method of lines. STEP M+2: Update of the hydrodynamic equations to a virtual $n+2$ time step via a (first order in time) Euler-corrector step using method of lines. STEP M+3: A second order (in time) hydrodynamics update is obtained by averaging the corrected quantities of step $M+2$ and the original variables of step n.

interface. We use a perfect fluid equation of state, $P=(\Gamma-1)\rho\epsilon$ with $\Gamma=5/3$.

We have first tested the code for each direction, x, y, and z, separately. The integration domain extends from -0.5 to 0.5 and at $t=0$ the interface is placed at $x=0$ (similarly when testing in the y and z directions). We use a grid of 400 zones along the relevant direction. Figures showing one-dimensional shock tube tests with the three different hydrodynamical methods we are using can be found in the supplement to this paper [2]. The L1-norm errors of the different hydrodynamical quantities for each method, at a final evolution time $t=0.4$, are presented in Table II. From this table (and the accompanying figures of the supplement [2]) we see that Marquina's scheme gives better results than the other two schemes. The improvement with respect to the Roe solver is not too sensitive for high resolutions ($\Delta x=1/400$). It is however quite relevant for the coarse grid (Δx

$=1/200$) evolutions. Also noticeable are the large errors found with the flux split method for the coarse grid although they are drastically reduced when the resolution is doubled. All errors reported in Table II are measured taking into account the whole domain of integration, i.e., including the discontinuities. Obviously, the smooth parts of the solution, e.g., the rarefaction wave, have less numerical error.

We next test the code by placing the initial discontinuity along the diagonal of the computational domain. With this setup we are checking the finite-differencing of all three directions simultaneously. We consider a grid of 128^3 zones, spanning an interval of length $1/\sqrt{3}$ in every direction. The diagonal of the cube therefore has unit length. We evolve to the same time as in the 1D tests, $t=0.4$, using a CFL factor $\Delta t/\Delta x=2\sqrt{3}/25$. This corresponds to 640 update iterations. The results of the evolution for Marquina's method are depicted in Fig. 4 (additional figures showing the results for the other two methods appear in the supplement [2]). We find good agreement between the numerical and analytic results. Notice that, due to the lack of resolution, some features such as the constant intermediate state in density, are less resolved than in the one dimensional case. The errors of the hydrodynamic quantities for this run are also contained in Table II. By inspection of this table, we observe that, although the errors are very similar for all schemes, the flux split method is slightly more accurate than both Roe and Marquina's method. However, the existence of a ''kink'' in the leading part of the rarefaction wave (which is more clear in one-dimesional simulations; see the corresponding figure in the supplement [2]) is still noticeable in the flux split method.

We summarize this section by stressing the shock-capturing capabilities of the different numerical schemes we use to integrate the hydrodynamic equations. Such capabilities are essential to our final goal of performing accurate simulations of interesting astrophysical scenarios, such as coalescing NS binaries.

V. FRIEDMANN-ROBERTSON-WALKER COSMOLOGY TESTS

For the first testbed of the coupled GR-Hydro code with dynamical spacetimes, we use the Friedmann-Robertson-Walker (FRW) model of an expanding cosmology. We use the standard form of the FRW metric

$$ds^2=-dt^2+R^2(t)\left[\frac{dr^2}{1-kr^2}+r^2(d\theta^2+\sin^2\theta d\phi^2)\right],$$
$$(74)$$

corresponding to an open ($k=-1$), flat ($k=0$), or closed ($k=1$) universe, with scale factor $R(t)$. For the special case $k=0$, each constant time slice is spatially flat. For this case, all terms involving spatial derivatives drop out of both the spacetime and hydrodynamic evolution equations. Although we still use the fully general form of the evolution equations to evolve the initial data, this allows us to concentrate on the coupling of the two codes in time. We note that the case $k=0$ has been extensively used in the literature for calibration

TABLE II. L1-norm errors of different hydrodynamical quantities, density, velocity, and pressure for the shock tube tests. The results correspond to the three different schemes we employ to integrate the hydrodynamic equations. The waves along the axis use either 200 of 400 grid zones in the direction of propagation, and one zone in the remaining two directions. This allows our 3D code to be effectively run as a 1D code. The diagonal shock tube test is run with 128 grid zones in each direction. All three solvers are found to perform nicely in the multidimensional case.

Dimension	$\Delta x = \Delta y = \Delta z =$	Solver	$\|E(\rho)\|_1$	$\|E(v)\|_1$	$\|E(p)\|_1$
Along axis	$\frac{1}{200}$	Flux-Split	1.99×10^0	3.46×10^{-1}	2.88×10^0
		Roe	1.19×10^{-1}	1.36×10^{-2}	8.15×10^{-2}
		Marquina	7.65×10^{-2}	8.13×10^{-3}	4.60×10^{-2}
	$\frac{1}{400}$	Flux-Split	6.61×10^{-2}	6.67×10^{-3}	4.25×10^{-2}
		Roe	6.90×10^{-2}	7.72×10^{-3}	4.33×10^{-2}
		Marquina	4.65×10^{-2}	4.84×10^{-3}	2.41×10^{-2}
diagonal	$\frac{1/\sqrt{3}}{128}$	Flux-Split	7.95×10^{-2}	8.35×10^{-3}	6.62×10^{-2}
		Roe	9.12×10^{-2}	9.39×10^{-3}	7.53×10^{-2}
		Marquina	9.23×10^{-2}	9.66×10^{-3}	7.98×10^{-2}

of cosmological codes. We also study the non-spatially flat case $k=-1$, which involves non-trivial spatial derivatives.

We take the matter to be collisionless dust, $P=\epsilon=0$. Under these assumptions, the hydrodynamic evolution equations reduce to

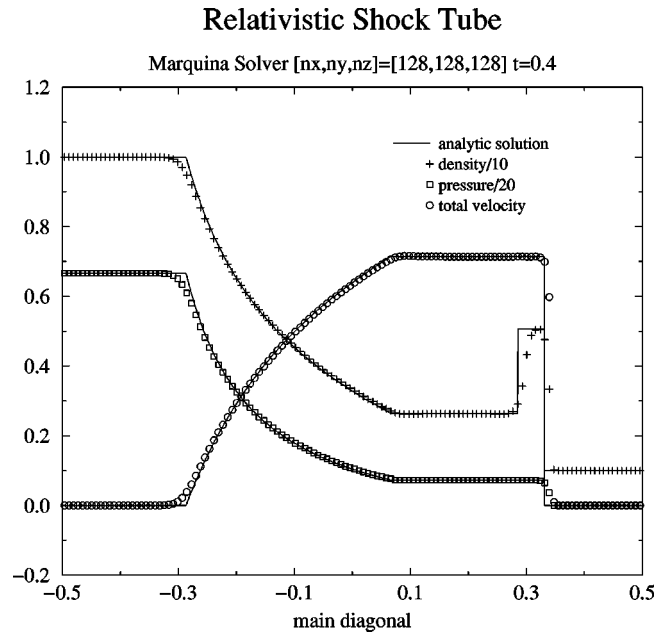$$\partial_t(\rho R^3)=0, \tag{75}$$

Relativistic Shock Tube

FIG. 4. Numerical (symbols) versus analytic (solid lines) results for the three-dimensional shock tube tests at $t=0.4$ using Marquina's method. Shown are normalized profiles of rest-mass density, ρ, pressure, P, and velocity, $v=(\gamma^{ij}v_iv_j)^{1/2}$, as functions of the coordinate distance along the main diagonal. A uniform Cartesian grid of 128^3 zones was used.

while the Einstein equations take the form

$$(\partial_t R)^2+k=\frac{8\pi}{3}\rho R^2 \tag{76}$$

(see, for example, [67]).

It is important to stress that we are using the *full* set of evolution equations to integrate the initial data. The only assumption we make is that the stress-energy tensor take the form of a perfect fluid with zero pressure and internal energy. We only use the simplified solutions given by Eqs. (75), (76) to check the validity of our numerical evolutions.

A. $k=0$ convergence tests

For the $k=0$ case we have a flat spatial geometry. Notice that Eq. (75) implies that the evolved ''densitized'' variable $\tilde{D}$ is a constant of motion

$$\tilde{D}=\sqrt{\gamma}W\rho=\rho R^3=\text{const.} \tag{77}$$

This, combined with the fact that no fields have any spatial dependence, results in the hydrodynamical finite difference equations becoming exact for our choice of variables. Therefore, we see no difference between the three methods which we use to integrate the hydrodynamical variables. Because of this, we only present results for the flux split hydrodynamics method, and note that the results for Roe and Marquina methods are equivalent. We concentrate on the spacetime evolution as driven by the hydrodynamic source. This is the first direct test of the spacetime-hydrodynamics coupling aspects of our code.

Since numerical error only comes from finite differencing errors in time, we could run the convergence tests by simply decreasing the CFL number. However, to minimize boundary effects, we run the convergence tests at three different

TABLE III. Different run parameters for the FRW $k=0$ convergence tests.

Resolution	No. of points in each direction	Δx	$c\dfrac{\Delta t}{\Delta x}$	No. of time steps
low	10	0.01	0.25	1
medium	20	0.05	0.25	2
high	40	0.025	0.25	4

resolutions: $40^3, 20^3, 10^3$ with a fixed CFL factor as summarized in Table III. For initial conditions we choose $R(t=0)=1$, and $\rho(t=0)=0.01$.

In Fig. 5 we display the Hamiltonian constraint violations for the BMEIN_FLUX scheme. We see classical second order convergence of the Hamiltonian constraint. Similar graphs for other schemes can be found in [2].

We turn next to evolutions with ADM based evolution schemes. In Fig. 6 we compare the Hamiltonian constraint for the ADMICN_FLUX (iterative Crank-Nicholson) and ADMLEAP_FLUX (leapfrog) systems. For the ADMICN_FLUX system we get second order convergence, with some ''noise'' at the boundaries. This noise propagates into the grid more quickly than with other methods, due to the iterative nature of the ICN scheme.

The ADMLEAP_FLUX system does *not* appear to be converging at second order. When we plot the root mean square (RMS) Hamiltonian as a function of time step for this system, Fig. 7, we observe an oscillatory behavior. This appears to be a ''loused'' solution to the finite difference equations, as described by New *et al.* [68], occurring when non-staggered leapfrog methods are used to evolve certain non-linear systems of equations. The ''loused'' solution is a nonphysical solution characterized by oscillations from time step
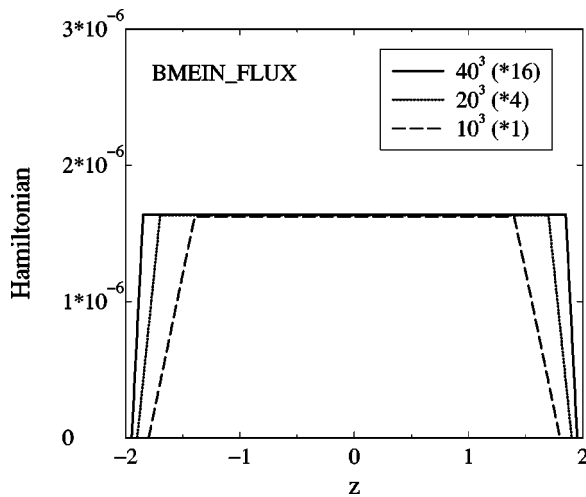

FIG. 5. The Hamiltonian constraint violation for the BMEIN_FLUX system. Since the hydrodynamic code gives the exact solution for the special case of $k=0$ initial data, there is no need to monitor the errors in the matter field evolutions, or to compare different hydrodynamic methods. We find that the BMEIN_FLUX system converges to second order.

to time step. We have seen evidence of this solution only when using the ADMLEAP system, both in vacuum evolutions, and when coupled to hydrodynamics. It does not appear with any other system. For comparison, see the RMS Hamiltonian for the BMEIN_FLUX evolution in Fig. 7. The degree to which this ''loused'' solution occurs depends strongly on the initial data and choice of gauge.

It is important to note that this ''loused'' solution is a solution to the finite difference equations, *not* to the differential equations. Thus, it must converge away with increased resolution. In Fig. 8 we show the RMS Hamiltonian at each iteration for three different resolutions: $40^3, 20^3$, and 10^3. Its value has been scaled by a factor of sixteen for the finest resolution grid, and a factor of four for the medium resolution grid, so that if the solution is converging to second order, the value graphed should remain constant. The iteration number refers to the finest grid. Since the ''loused'' solution oscillates between a maximum and minimum value every time step, at iteration number four the fine and medium grid correspond to a minimum (they have evolved an even number of time steps) while the coarse grid is at a maximum (having evolved an odd number of time steps). Hence, iteration eight is the first time when all three grids correspond to a minimum. At this point we see that the solution is indeed converging at second order.

To summarize this subsection, we have verified that the coupling between the spacetime and hydrodynamic methods, described in Sec. III, yield second order convergence in time. We have seen evidence of a ''loused'' solution in the ADMLEAP system. This ''loused'' solution produces a nonphysical oscillation in time. By comparing this oscillation at three resolutions, and at the same stage of oscillation (when all three resolutions are at a minimum), we see that it is converging away.

B. $k=-1$ convergence tests

Having confirmed the fidelity of the (second order accurate) coupling in time between the spacetime and hydrodynamical evolutions, we next test the spatial derivatives by studying the non-spatially flat $k=-1$ FRW solution. Since we now have to resolve spatial gradients, we increase the grid resolution using $160^3, 80^3$, and 40^3 zones. The different runs are summarized in Table IV. For initial conditions we choose $R(t=0)=1$, and $\rho(t=0)=0.01$.

To monitor the correctness of the spacetime evolution we cut off our grid at $x=y=z=\pm1$ and ignore errors caused by boundary effects. To analyze the spacetime, we look at the
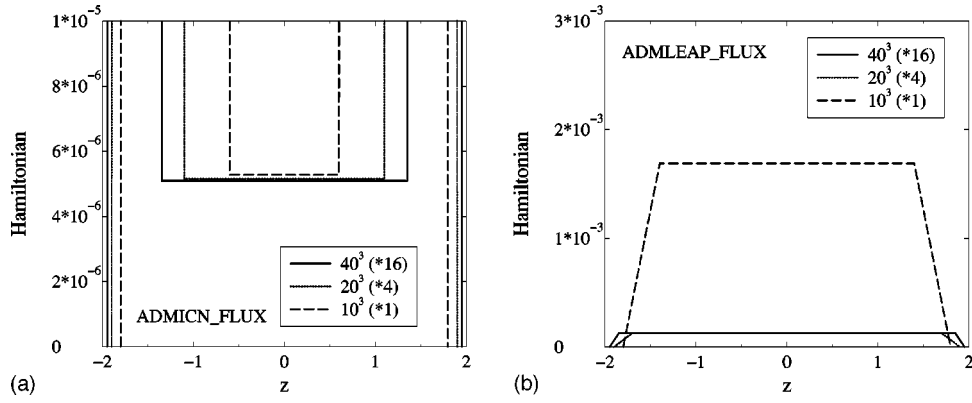
FIG. 6. Comparison of the Hamiltonian constraint values for the ADMICN_FLUX (left plot) and ADMLEAP_FLUX systems. For the ADMICN_FLUX system we obtain second order convergence apart from some numerical ''noise'' at the boundaries. The ADMLEAP_FLUX system does *not* appear to be converging.

Hamiltonian constraint, H, and the x component of the momentum constraint, M^x. Correspondingly, to address the accuracy of the hydrodynamic evolution we look at the conservation of ρR^3, $\Delta(\rho R^3) = \rho R^3|_t - \rho R^3|_{t=t_0}$. This quantity should remain constant for a matter dominated FRW solution.

In Fig. 9 we plot H, M^x, and the error in the conserved quantity ρR^3 ($\Delta(\rho R^3)$) for the ADMLEAP_ROE system. We find second order convergence for all variables, and for all systems [2]. We also find little difference between the various hydrodynamic methods for this initial data [2].

To summarize, we find second order convergence for all systems, even when non-trivial spatial gradients are present. The FRW $k = -1$ spacetime tests out many of the terms in the spacetime evolution, however, the assumption that the matter be composed of dust excludes many of the terms in the hydrodynamical equations. Due to this, we see little difference between the three different hydrodynamical methods when evolving this initial data. In Table V, we summarize the errors for all combinations of evolution systems obtained at the lowest resolution. We find that the BMEIN system is slightly more accurate for this initial data.

VI. TOLMAN-OPPENHEIMER-VOLKOFF TESTS

The FRW tests analyzed in the previous section assumed the matter fields to be dust, i.e., $P = \epsilon = 0$. We now turn to a case where the pressure gradients play a central role. We evolve a static star in the general relativistic setting, that is, a self-gravitating matter distribution satisfying the (equilibrium structure) Tolman-Oppenheimer-Volkoff (TOV) equations.

The ability to numerically evolve a compact, strongly gravitating object is crucial for our program of developing a general purpose code for general relativistic astrophysics. Although the TOV solution is static, we evolve it with the full set of evolution equations for the hydrodynamics and the spacetime. To maintain the static solution during the numerical evolution, the pressure gradient must exactly balance the gravitational force in the general relativistic setting where both the energy density and pressure are sources of gravity. Even though the extrinsic curvature and fluid velocities are initially zero, finite differencing errors will allow these quantities to evolve away from zero.

Besides the complexity of the full GR-Hydro equations, there are two other difficulties in numerically evolving this strongly gravitating stellar configuration: (1) the treatment of
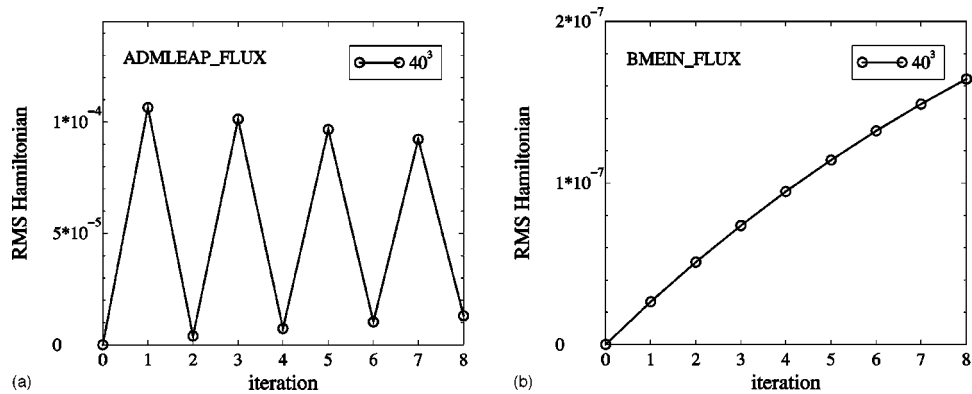

FIG. 7. Plot of the RMS Hamiltonian as a function of the time step for the ADMLEAP_FLUX (left) and BMEIN_FLUX systems. We observe that the Hamiltonian is oscillating only for the ADMLEAP_FLUX scheme. This has been reported in the literature as a non-physical ''loused'' solution. The degree to which the oscillations occur depends strongly on the initial data and gauge choice.
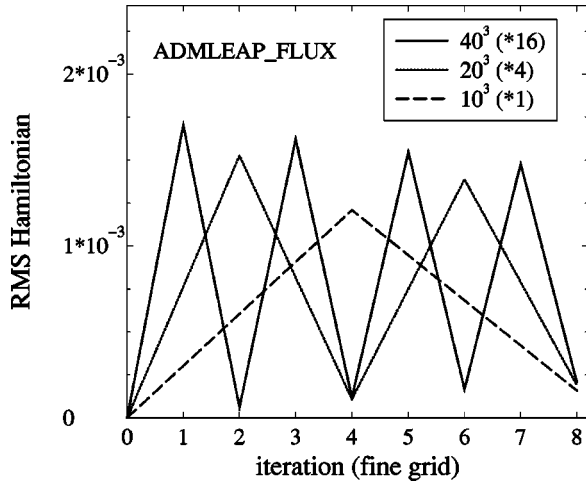
FIG. 8. The convergence of the ''loused'' oscillations appearing for the ADMLEAP_FLUX system is examined by plotting the RMS Hamiltonian at each iteration for three different resolutions: $40^3, 20^3$, and 10^3. The value is scaled by a factor of sixteen for the finest resolution grid, and a factor of four for the medium resolution grid. The iteration number refers to the finest grid. As explained in the text, iteration eight is the first time when all three grids coincide to a minimum value. At this point we see that the solution converges at second order.

the surface of the star, and (2) providing a coordinate condition which maintains the long term stability of the evolution. We will discuss the first difficulty, including our numerical treatment, later in this section. We will defer discussing the problem of long term stability to the second paper in this series.

We begin with a discussion of the initial data. The TOV equations [69,70] are the Einstein equations coupled to a perfect fluid stress-energy tensor under the assumptions that the solution is static and spherically symmetric. Specifically, the metric is given by

$$ds^2 = -\alpha(r)^2 dt^2 + \gamma_{rr}(r)dr^2 + r^2 d\theta^2 + r^2 \sin^2\theta d\phi^2, \tag{78}$$

where the metric functions, $\alpha(r)$ and $\gamma_{rr}(r)$, along with the primitive hydrodynamic variables, $\rho(r)$ and $\epsilon(r)$, are assumed to depend only on the circumferential radius r. (We note that this form of the metric is only used in computing the initial data, *not* in our dynamic evolution code, where a general form of the metric in a Cartesian coordinate system

TABLE IV. Run parameters for the FRW $k=-1$ convergence tests.

Resolution	No. of points in each direction	Δx	$c\frac{\Delta t}{\Delta x}$	No. of time steps
low	40	0.05	0.25	2
medium	80	0.025	0.25	4
high	160	0.0125	0.25	8

is used.) Under these assumptions, the Einstein equations, along with the hydrodynamic equations, reduce to

$$\frac{\partial P(r)}{\partial r} = -\frac{(\rho + \rho\epsilon + P)(m + 4\pi r^3 P)}{r(r-2m)}, \tag{79}$$

$$\frac{\partial(\ln \alpha(r))}{\partial r} = \frac{m + 4\pi r^3 p}{r(r-2m)}, \tag{80}$$

$$\frac{\partial m(r)}{\partial r} = 4\pi r^2(\rho + \rho\epsilon), \tag{81}$$

$$\gamma_{rr}(r) = \left(1 - \frac{2m(r)}{r}\right)^{-1}, \tag{82}$$

where $m = m(r)$ is the mass energy contained inside a sphere of circumferential radius r. To determine the initial data, the above coupled ordinary differential equations are integrated using a 4th order Runge-Kutta method, supplemented by a polytropic equation of state (EOS)

$$P = \mathcal{K}\rho^\Gamma. \tag{83}$$

We note that in the dynamical evolution we use the more generic EOS $P = (\Gamma-1)\rho\epsilon$. In the simulations shown in this section, we use $\Gamma = \frac{5}{3}$, $\mathcal{K} = 5.380 \times 10^9$ cm^4/g$^{2/3}$sec^2, and a central mass density of $\rho_c = 5 \times 10^{14}$ gcm^3. This configuration corresponds, roughly, to a neutron star [71]. Other initial parameters, showing essentially the same features as the one presented here, have also been tested. This choice of parameters leads to a TOV solution with a total ADM mass of $0.566 M_\odot$ and a circumferential coordinate radius of 14.9 km. The TOV solution is then matched at the star's surface to an exterior Schwarzschild spacetime with the appropriate mass and coordinates [72]. A coordinate transformation from spherical to Cartesian coordinates is then performed to obtain initial data for the 3D evolution code. The evolution is performed with ''harmonic slicing,'' that is $\dot{\alpha} = \alpha\sqrt{\gamma}\gamma^{ij}K_{ij}$. While this configuration represents a compactness ratio of $R/2M \approx 9$, we note that we have performed tests using TOV configurations with compactness ratios of $2.5 \leq R/2M < 100$. We have verified strict second order convergence for this range of compactness ratio for the TOV configurations and for the boosted TOV configurations of Sec. VII.

We now present convergence tests for the evolution code. These tests are performed with the parameters described in Table VI. As the initial data exhibits octant symmetry, only one octant is evolved (with appropriate boundary conditions used at the inner faces of the computational domain). This is an important capability of the code, in that it enables us to achieve a higher resolution and make more efficient use of available computational resources when allowed by the symmetry of the problem. The initial data is then evolved up to $t = 0.986$ μs. We calculate $\Delta\rho = \rho_{num} - \rho_{exact}$, the difference between the numerically computed mass density, ρ_{num}, and the exact solution, ρ_{exact}. We also monitor the Hamiltonian constraint, H, and the x-momentum constraint, M^x. The maximum values of these errors obtained at the lowest reso-

lution (excluding the region near the surface of the star) are summarized in Table VII for each evolution system. The order of convergence of the code, for various combinations of the spacetime and hydrodynamics integrations, is demonstrated as before.

For the TOV solution, we have tested the consistency of the flux-split, Roe, and Marquina methods both with and without the limiting functions. We note that the limiters result in a truncation error that is first order in Δx at points that attain minimum and maximum in the hydrodynamical variables, while all other points have truncation errors which are second order in Δx. For the convergence tests in this section we will only present results without the flux limiters. Specifically, we have built a switch in the code that will set $\psi = 1$ for the limiting of the flux split method, Eq. (63), and turn off the minmod function for the Roe and Marquina methods, so that we can perform tests with and without limiters to ensure that all components of our code have the expected convergence properties.

A typical convergence plot is shown in Fig. 10 (see [2] for convergence plots of all combinations of the different numerical methods). Notice that all quantities are converging as second order in Δx in the interior of the star, $r < 14.9$ km. The treatment of the surface of the star will be discussed in detail later.

Notice that the rest mass density for all systems using the flux-split hydrodynamical evolution scheme (FLUX) is converging at a rate that is higher than second order for the Δx used here. This is due to the fact that, for the resolutions used here, the truncation error terms that are proportional to Δx^3 have a magnitude comparable to the truncation error terms that are proportional to Δx^2. This causes the appearance of ''hyper-convergence'' as seen in Fig. 11. To confirm this point, we plot in Fig. 12 the results of a convergence test for
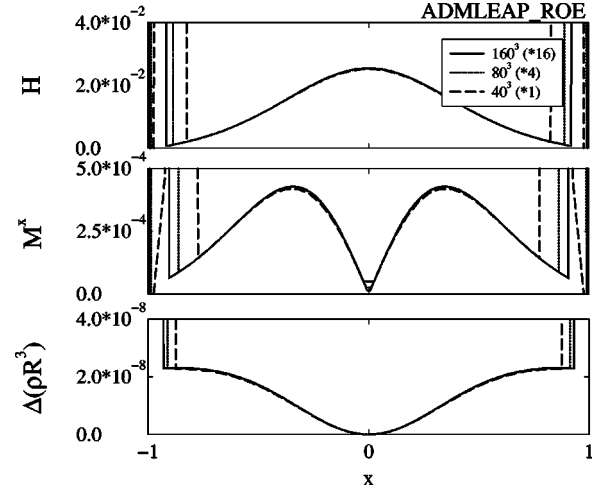

FIG. 9. Convergence plot of the Hamiltonian constraint (H), the x component of the momentum constraint (M^x), and error plot for the conserved quantity ρR^3 ($\Delta(\rho R^3)$) for the ADMLEAP_ROE system. All variables show second order convergence for every combination of systems [2]. We also find little difference between the various hydrodynamic methods for this initial data [2].

the ADMLEAP_FLUX system with two times the spatial resolution as described in Table VI. A clear indication of second order convergence is observed at this resolution.

We also note that for the same resolution, the BMRIC spacetime evolution scheme is slightly more accurate than the other spacetime evolution schemes (from a standpoint of the absolute value of the error functions plotted). Also, the Roe and Marquina schemes for evolving the hydrodynamics are more accurate (by an order of magnitude) than the flux-split method (see [2] for details).

Next we turn to the star's surface treatment. There are

TABLE V. This table summarizes the magnitudes of the errors for the different evolution systems when evolving the FRW $k=-1$ spacetime. These values correspond to the lowest resolution listed in Table IV.

Abbreviation	max(H)	max (M^x)	max ($\Delta(\rho R^3)$)
ADMLEAP_ROE	2×10^{-2}	4×10^{-4}	2×10^{-8}
ADMLEAP_FLUX	2×10^{-2}	4×10^{-4}	2×10^{-8}
ADMLEAP_MAR	2×10^{-2}	4×10^{-4}	2×10^{-8}
ADMICN_ROE	2×10^{-2}	4×10^{-4}	2×10^{-8}
ADMICN_FLUX	2×10^{-2}	4×10^{-4}	2×10^{-8}
ADMICN_MAR	2×10^{-2}	4×10^{-4}	2×10^{-8}
BMEIN_ROE	2×10^{-2}	5×10^{-5}	1×10^{-8}
BMEIN_FLUX	2×10^{-2}	5×10^{-5}	1×10^{-8}
BMEIN_MAR	2×10^{-2}	5×10^{-5}	1×10^{-8}
BMRIC_ROE	2×10^{-2}	1×10^{-4}	2×10^{-8}
BMRIC_FLUX	2×10^{-2}	1×10^{-4}	2×10^{-8}
BMRIC_MAR	2×10^{-2}	1×10^{-4}	2×10^{-8}

TABLE VI. Computational grid parameters for the TOV tests.

Resolution	No. of points in each coordinate direction	Δx (km)	$c\dfrac{\Delta t}{\Delta x}$	No. of time steps	Total evolved time (μs)
low	16	1.182	0.125	2	0.986
medium	32	0.591	0.125	4	0.986
high	64	0.2955	0.125	8	0.986

three related issues in the numerical evolution of the surface region of a compact self-gravitating object in general relativity. (1) At the surface of the star, the second normal derivatives of some of the hydrodynamic quantities, e.g., the density, are discontinuous for most equations of state. This discontinuity is also present in the TOV solution with $\Gamma = 5/3$ studied in this section. The Einstein equations imply that the curvature tensor (which contains second derivatives of the metric functions) has a kink at the stellar surface. The curvature tensor enters explicitly in the evolution equation of the extrinsic curvature. This makes the numerical treatment of the stellar surface considerably more difficult in the relativistic framework than in the Newtonian case. Accordingly, the numerical evolutions are less stable in the relativistic case. (2) In the exterior of the star there is vacuum and hence, the density must drop to zero. As the density approaches zero, the transformation from the evolved variables $(\tilde{D}, \tilde{S}_i, \tilde{\tau})$ to the primitive variables (ρ, v^i, ϵ) becomes singular. The ''standard'' treatment of this problem is to add an ad hoc ''atmosphere,'' with some choice of thermal properties in the exterior region. This atmosphere typically has a den-

sity orders of magnitude smaller than that of the interior of the star and should have a negligible effect on the dynamics of the system. In our simulations, we typically pick the atmosphere to be 10^{-4} to 10^{-5} of the central density ρ_c of the TOV star and with the same EOS as the star. This is sufficient to ensure that the GR-Hydro equations are neither singular nor degenerate in our treatment, while having negligible effect on the actual dynamical evolution of the star. (3) In regions of low density, especially in the atmosphere near the surface of the star, there are two related difficulties: (i) it is difficult to accurately recover the pressure (which is a power of the density) from the evolved variables $(\tilde{D}, \tilde{S}_i, \tilde{\tau})$, and (ii) it is easy to develop high velocity flows due to the strong gravitational field there. In particular, the atmosphere ($r > 14.9$ km) is not part of the equilibrium TOV initial data, and the gravitational field is driving it to collapse onto the surface of the star. Numerically, it is problematic to have the atmosphere colliding with the surface of the star, creating a shock, and leaving the specific internal energy density ϵ in the atmosphere behind plunging to zero. These difficulties eventually cause the code to crash. An explicit demonstration

TABLE VII. This table summarizes the magnitudes of the errors for the different evolution systems when evolving the TOV spacetime. These values correspond to the lowest resolution listed in Table VI, and errors near the surface of the star are ignored.

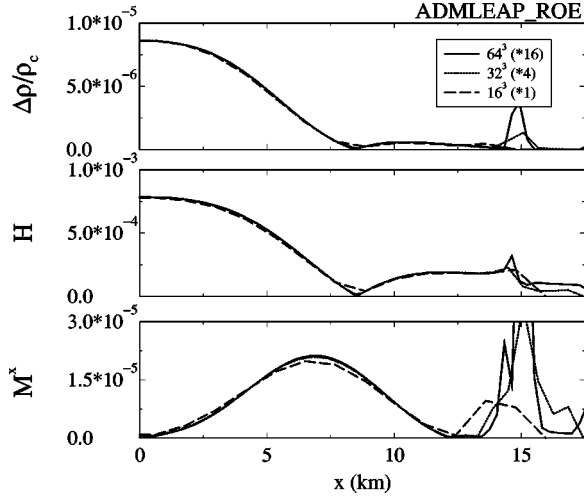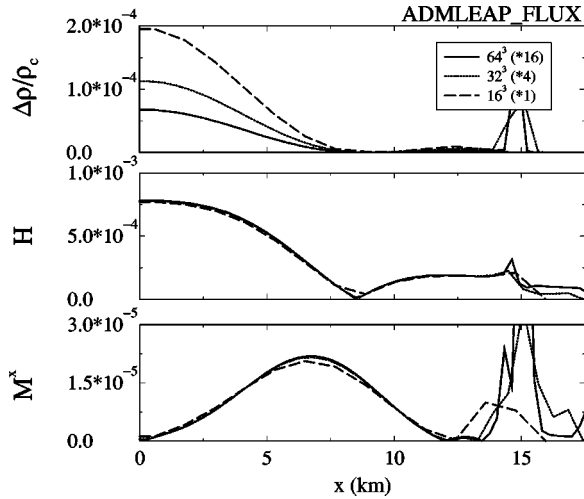| Abbreviation	$\max\left(\dfrac{\Delta\rho}{\rho_c}\right)$	$\max(H)$	$\max(M^x)$
ADMLEAP_ROE	8×10^{-6}	8×10^{-4}	2×10^{-5}
ADMLEAP_FLUX	2×10^{-4}	8×10^{-4}	2×10^{-5}
ADMLEAP_MAR	8×10^{-6}	8×10^{-4}	2×10^{-5}
ADMICN_ROE	8×10^{-6}	8×10^{-4}	2×10^{-5}
ADMICN_FLUX	2×10^{-4}	8×10^{-4}	2×10^{-5}
ADMICN_MAR	8×10^{-6}	8×10^{-4}	2×10^{-5}
BMEIN_ROE	5×10^{-6}	8×10^{-4}	5×10^{-6}
BMEIN_FLUX	2×10^{-4}	8×10^{-4}	5×10^{-6}
BMEIN_MAR	5×10^{-6}	8×10^{-4}	5×10^{-6}
BMRIC_ROE	1×10^{-6}	8×10^{-4}	1×10^{-5}
BMRIC_FLUX	2×10^{-4}	8×10^{-4}	1×10^{-5}
BMRIC_MAR	1×10^{-6}	8×10^{-4}	1×10^{-5}

FIG. 10. We demonstrate the convergence of the ADMLEAP_ROE evolution system for three different error functions: the difference between the analytic and computed rest mass density (normalized by the central rest mass density ρ_c) $\Delta\rho/\rho_c$, the Hamiltonian constraint H, and the x-momentum constraint M^x. In each of the three cases, we multiply the high resolution result by sixteen and the medium resolution by four to show second order convergence. All results are shown at $t=0.986$ μs which corresponds to eight iterations at the highest resolution. The graphs are taken along the x axis (results on the y and z axes are identical, and results on the diagonal axis are similar).

FIG. 11. We demonstrate the convergence of the ADMLEAP_FLUX evolution system for three different error functions: the difference between the analytic and computed rest mass density (normalized by the central rest mass density ρ_c) $\Delta\rho/\rho_c$, the Hamiltonian constraint H, and the x-momentum constraint M^x. In each of the three cases, we multiply the high resolution result by sixteen and the medium resolution by four to show second order convergence. Note that the rest mass density (top frame) is converging faster than second order in Δx (see text for explanation). All results are shown at $t=0.986$ μs which corresponds to eight iterations at the highest resolution. The graphs are taken along the x axis (results on the y and z are identical, and results on the diagonal axis are similar).
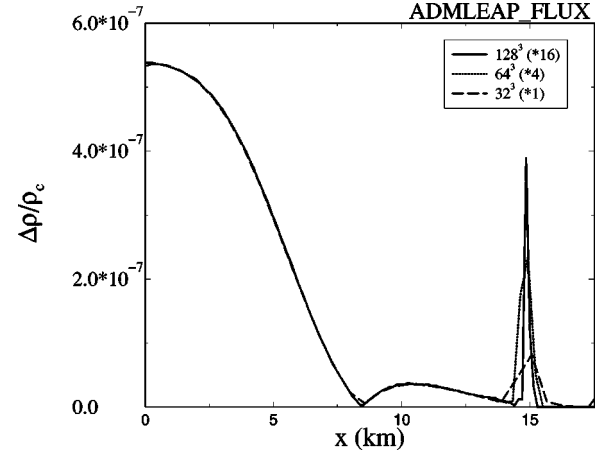
FIG. 12. We demonstrate the convergence of the evolved rest mass density with the ADMLEAP_FLUX evolution system for a set of grids with twice the resolution as that displayed in Table VI. All results are shown at $t=0.493$ μs which corresponds to eight iterations at the highest resolution. The graphs are taken along the x axis (results on the y and z axes are identical, and results on the diagonal axis are similar).

of these difficulties is exhibited in Figs. 13 and 14. For these simple tests, we evolve a TOV configuration as described above. We implement an atmosphere with density $\rho_{atmos} = 10^{-5}$ ρ_c. The resulting atmosphere has a specific internal energy of $\epsilon_{atmos} \approx 4.6 \times 10^{-4}$ ϵ_c. We then evolve the configuration with the ADMLEAP_FLUX evolution scheme. We first evolve this configuration *without* implementing the surface treatment described below (which *is* implemented for all other runs in this paper). The code crashes after only 0.043 ms, due to the specific internal energy dropping to zero near the surface of the star. Figure 13 shows a 3D isosurface plot of ϵ corresponding to a value of $\epsilon = 0.87\epsilon_{atmos}$ at time $t=0.04$ ms. These regions indicate where the specific internal energy is dropping significantly in this short time interval. Figure 14 shows a 1D plot of ϵ/ϵ_c. Clearly, the specific internal energy is dropping to zero rapidly in an unstable fashion near the surface of the star $r=14.9$ km.

To circumvent these problems, we have found a simple, yet effective treatment for the stable numerical evolution of low density regions. Again, since this scheme is enacted only for very low density flow, it has a negligible effect on the dynamics of the system. It is important to note that the first indication of problems is in the recovery of the primitive variables (ρ, v^i, ϵ) from the evolved variables $(\widetilde{D}, \widetilde{S}_i, \widetilde{\tau})$, given by Eq. (6). In regions where the rest mass density ρ is less than some specified minimum density ρ_{min} (typically some fraction of the atmosphere rest mass density), if the recovery of the primitive variables (ρ, v^i, ϵ) from the evolved variables $(\widetilde{D}, \widetilde{S}_i, \widetilde{\tau})$ results in a negative specific energy density ϵ, then the primitive variables are solved again, with the condition for adiabatic flow

$$P = \mathcal{K}\rho^\Gamma, \tag{84}$$

replacing the definition of $\widetilde{\tau}$ [the fifth component of Eq. (6)].
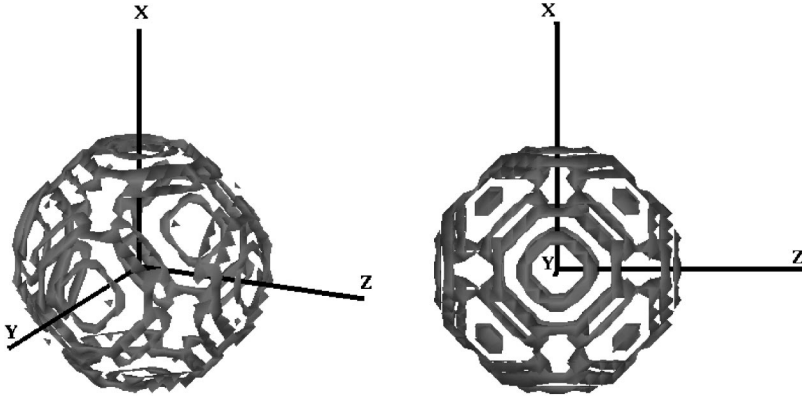
FIG. 13. A 3D isosurface of constant internal specific energy density at time $t = 0.04$ ms is shown. This value of $\epsilon = 4.1 \times 10^{-4}$ ϵ_c corresponds to 87% of the atmosphere specific internal energy density. Notice that the troublesome regions are not along the coordinate or diagonal axes and hence would not be observable in 1D plots along these axes.

To demonstrate the stability of this scheme, we show in Fig. 15 the final rest mass density profile for numerical evolutions of the TOV configuration described above. The three different hydrodynamics evolutions schemes (flux-split, Roe, and Marquina) are used to evolve the same TOV configuration to 1.0 ms (approximately 8000 time steps). Note that for Roe's method, the final mass density configuration is indistinguishable from the initial configuration profile.

VII. BOOSTED TOV TESTS

The final test case we present is the most stringent. The boosted TOV solution effectively tests many of the features one requires of a general relativistic hydrodynamical spacetime code: relativistic fluid motion, strong gravitational fields, and a non-trivial coordinate condition for both the lapse and shift. In addition, this test has an analytic solution to compare against. We simulate a neutron star moving with a high velocity along the $\hat{x} + \hat{y} + \hat{z}$ diagonal of the computational domain. We find that in such a simulation, every term in both the Einstein equations and the GR-Hydro equations is activated; each evolved variable is a nontrivial function of space and time. We obtain an analytic solution of a boosted neutron star by applying a coordinate transformation that corresponds to a Lorentz boost at spatial infinity on a solution to the TOV equations. Specifically, if the solution to the TOV equations (see Sec. VI) is expressed in Cartesian (t,x,y,z) coordinates, we transform to another set of coordinates (t',x',y',z') via the transformation

$$
\begin{bmatrix} t' \\ x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \gamma_b & \xi_x \gamma_b & \xi_y \gamma_b & \xi_z \gamma_b \\ \xi_x \gamma_b & \left(1 + \dfrac{(\gamma_b - 1)\xi_x^2}{\xi^2}\right) & \left(\dfrac{(\gamma_b - 1)\xi_x\xi_y}{\xi^2}\right) & \left(\dfrac{(\gamma_b - 1)\xi_x\xi_z}{\xi^2}\right) \\ \xi_y \gamma_b & \left(\dfrac{(\gamma_b - 1)\xi_x\xi_y}{\xi^2}\right) & \left(1 + \dfrac{(\gamma_b - 1)\xi_y^2}{\xi^2}\right) & \left(\dfrac{(\gamma_b - 1)\xi_y\xi_z}{\xi^2}\right) \\ \xi_z \gamma_b & \left(\dfrac{(\gamma_b - 1)\xi_x\xi_z}{\xi^2}\right) & \left(\dfrac{(\gamma_b - 1)\xi_y\xi_z}{\xi^2}\right) & \left(1 + \dfrac{(\gamma_b - 1)\xi_z^2}{\xi^2}\right) \end{bmatrix} \begin{bmatrix} t \\ x \\ y \\ z \end{bmatrix}, \tag{85}
$$

where $\xi^2 = \xi_x^2 + \xi_y^2 + \xi_z^2$ and $\gamma_b = (1 - \xi^2)^{-1/2}$. The resulting metric and stress-energy tensor in the primed frame,

$$
g_{\mu'\nu'}(t',x',y',z') \equiv \frac{\partial x^\alpha}{\partial x^{\mu'}} \frac{\partial x^\beta}{\partial x^{\nu'}} g_{\alpha\beta}, \tag{86}
$$

$$
T_{\mu'\nu'}(t',x',y',z') \equiv \frac{\partial x^\alpha}{\partial x^{\mu'}} \frac{\partial x^\beta}{\partial x^{\nu'}} T_{\alpha\beta}, \tag{87}
$$

are also solutions to the Einstein equations coupled to the GR-Hydro equations. Notice that the shift in the boosted (primed) coordinates is non-zero. For example, the x component of the shift vector is given by

$$\beta_{x'}(t',x',y',z') = g_{t'x'}(t',x',y',z')$$

$$= \gamma_b^2 \xi_x \alpha^2 - \gamma_b \left(1 + \frac{(\gamma_b - 1)\xi_x^2}{\xi^2} \right)(\xi_x \gamma_{xx} + \xi_y \gamma_{xy} + \xi_z \gamma_{xz}) - \gamma_b \frac{(\gamma_b - 1)\xi_x \xi_y}{\xi^2}(\xi_x \gamma_{xy} + \xi_y \gamma_{yy}$$

$$+ \xi_z \gamma_{yz}) - \gamma_b \frac{(\gamma_b - 1)\xi_x \xi_z}{\xi^2}(\xi_x \gamma_{xz} + \xi_y \gamma_{yz} + \xi_z \gamma_{zz}), \qquad (88)$$

where α and γ_{ij} are the lapse and 3-metric, respectively, computed in the rest frame (unprimed coordinates) evaluated at coordinates (t,x,y,z) that correspond to the primed coordinates (t',x',y',z'). In the tests performed in this section, we specify the lapse and shift to be given by these values and check that all evolved variables converge to the analytic solution. For the pre-boosted TOV solution, we use the same configuration used in the previous section.

In Fig. 16 we plot the evolution of the rest mass density along the direction of boost ($\hat{x} + \hat{y} + \hat{z}$ diagonal) as a function of time. The boosted star is evolved for 0.4 ms, and is boosted with a velocity of $v/c = 0.3$. As can be seen, the neutron star has traversed approximately 36 km during the 0.4 ms, maintaining its original profile.

For completeness, we also present convergence tests for the boosted star. The grid parameters for the tests are given in Table VIII. The initial data is evolved with the three different resolutions. To test the code in the highly relativistic regime, we use the boost parameters $\xi_x = \xi_y = \xi_z = 0.5$ giving a Lorentz factor $\gamma_b = 2$. These boost parameters correspond to a neutron star moving in the $\hat{x} + \hat{y} + \hat{z}$ diagonal direction with a velocity of $v/c = 0.87$.

Figure 17 shows convergence plots of various evolved variables for a typical numerical scheme (see [2] for convergence plots for all of the different numerical methods). The
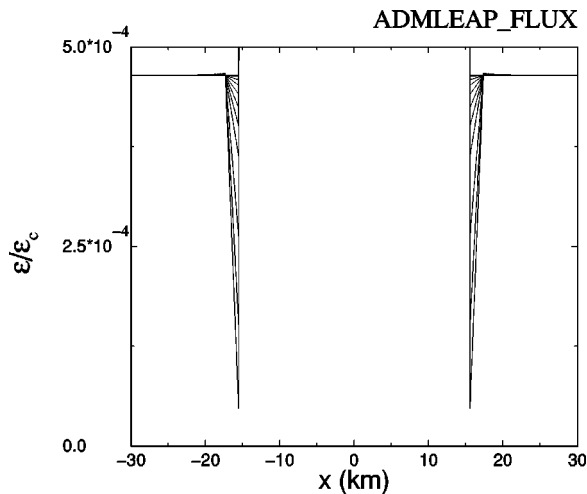
left panel of the figure contains plots of the difference between the numerically evolved rest mass density and the analytic solution, normalized by the central density ($\Delta\rho/\rho_c$), the Hamiltonian constraint (H), and the x component of the momentum constraint (M^x). The right panel of the figure contains plots of the difference between the numerically evolved specific energy density and the analytic solution (normalized by the central specific energy density), $\Delta\epsilon/\epsilon_c$, the difference between the xx component of the extrinsic curvature and the analytic solution, ΔK_{xx}, and the difference between the x component of the momentum and the analytic solution, ΔS^x.

To provide adequate resolution for the convergence tests, we move the boundaries of our computational domain inside of the star. We ignore errors caused by the boundary, and focus on the convergence properties of the interior solution.

In comparing the absolute value of the errors for the different schemes, we notice no significant difference between the two spacetime evolution schemes (ADMLEAP, ADMICN). However, there is clearly a difference between the hydrodynamical evolution schemes, where the Roe and Marquina methods (ROE,MAR) are equally more accurate than the flux-split method (FLUX) for the resolutions used. The maximum values of the errors are summarized in Table IX.

FIG. 14. 1D plot of ϵ/ϵ_c for the ADMLEAP_FLUX system evolving a TOV configuration. The lines are plotted in the x direction at time intervals of 0.002 ms, and correspond to coordinate values $z = 4.6$ km, $y = -6.4$ km. The final time is $t = 0.042$ ms. The code crashes shortly afterwards.
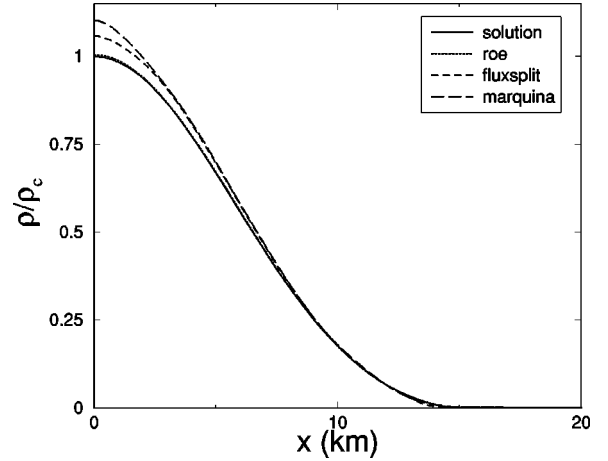

FIG. 15. Long term (1.0 ms) evolution of TOV initial data. The final rest mass density is plotted for the ADMLEAP_ROE, ADMLEAP_FLUX, and ADMLEAP_MAR systems, along with the static analytic solution. The evolution with the ADMLEAP_ROE system is indistinguishable from the analytic solution. The resolution used for these runs correspond to $\Delta x = 0.2954$ km, with $\Delta t/\Delta x = 0.125/c$. 8000 time steps were required to evolve to 1.0 ms.
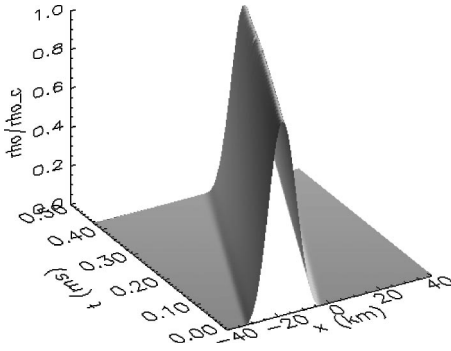
FIG. 16. A plot of the evolution of the rest mass density ρ, scaled by the central rest mass density ρ_c along the diagonal as a function of time. The initial data corresponds to the TOV configuration from Sec. VI boosted in the $\hat{x}+\hat{y}+\hat{z}$ diagonal direction with velocity $v/c=0.3$. The star is evolved for 0.4 ms. The spatial resolution corresponds to approximately 30 points across the star.

VIII. CONCLUSIONS

In this paper we present a new three-dimensional, Eulerian, general relativistic hydrodynamical code constructed for general relativistic astrophysics. This code is capable of evolving the coupled system of the Einstein and hydrodynamic equations. The code is constructed for a completely general spacetime metric based on a Cartesian coordinate system, with arbitrarily specifiable lapse and shift conditions. This paper discussed the general relativistic hydrodynamics part of the code, and its coupling to the spacetime code, in parallel to the presentation of the spacetime (vacuum) part of the code in [1].

We have derived a spectral decomposition for the GR-Hydro equations valid for general spatial metrics, generalizing the results of [38] which were only valid for the case of a diagonal metric. Based on this spectral decomposition, three different approximate (linearized) Riemann solvers, flux-split, Roe and Marquina, were used to integrate the relativistic hydrodynamic equations. We tested these methods individually and compared the results against one another. While we found all methods converging to second order in the discretization parameter, we also compared the absolute values of errors of the different methods.

Which method produced the smallest absolute error, and whether the spacetime or hydrodynamical evolution was the dominant source of error, depends on the initial data being evolved. For the shocktube problem, only the hydrodynamical evolution was relevant since the evolution took place on a flat background metric. For an evolution along a coordinate axis, the Roe and Marquina methods were superior to the flux split method. For an evolution where the shockfront is along the diagonal, the flux split method was slightly more accurate than both the Roe and Marquina method. For the FRW evolutions, the spacetime evolution is the main source of error. The BM system tends to be more accurate than the ADM system. For the TOV tests, we find that the Roe and Marquina methods are more accurate than the flux split method, and the BM system is more accurate than the ADM system. For the boosted TOV test, the Roe and Marquina methods are again superior to flux split. We caution that these statements could depend on the resolution used and the duration of evolution.

The hydrodynamic evolution is coupled to the spacetime evolution in a manner which is second order accurate in *both* space and time. The coupled code was subjected to a series of convergence tests, with different combinations of the spacetime and hydrodynamics finite differencing schemes, demonstrating the consistency of the discrete equations with the differential equations [73]. The extensive convergence tests performed are important not only for the validation of the code, but have also been important debugging tools during the code development process. We consider the tests presented to be a minimal set that any 3D GR-Hydro code should pass before actual applications. The test-beds that we report on in this paper include: special relativistic shock tubes, Friedmann-Robertson-Walker cosmology tests, evolution of equilibrium configurations of compact stars (solutions to the Tolman-Oppenheimer-Volkoff equations), and the evolution of relativistically boosted TOV stars transversing diagonally across the computational domain. The degree of complexity presented in these tests increases from purely special relativistic flows in flat backgrounds to fully general relativistic flows in dynamical spacetimes. In particular, the last test-bed (the boosted star) involves *all* possible terms in the coupled set of GR-Hydro evolution equations and were carried out with a non-trivial lapse and shift vector.

We found a simple, yet effective treatment for handling the surface region of a general relativistic self-gravitating compact object. The key idea is to replace the energy equation update by the condition of adiabatic flow in regions of low density. While the surface region is not changing the overall dynamics of the star, numerical instabilities there could halt the numerical evolution if uncontrolled. The capability to handle the surface region in a stable fashion is important for the application of the code to the study of neutron star astrophysics. We have demonstrated this capability in the equilibrium and boosted star test-beds. Refinement of this

TABLE VIII. Computational grid parameters for boosted TOV tests.

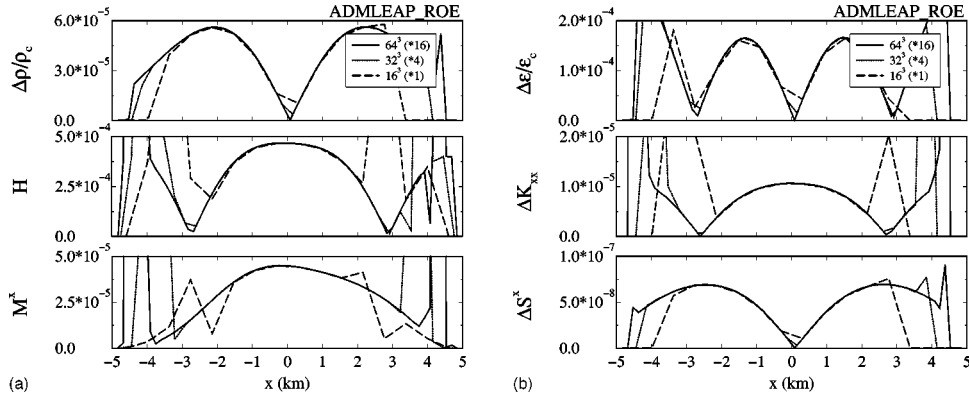| Resolution	No. of points in each coordinate direction	Δx (km)	$c\dfrac{\Delta t}{\Delta x}$	No. of time steps	Total evolved time (μs)
low	16	0.3545	0.125	2	0.296
medium	32	0.1772	0.125	4	0.296
high	64	0.0886	0.125	8	0.296

FIG. 17. We demonstrate the convergence of the ADMLEAP_ROE evolution system for six different error functions. On the left panel, we plot the difference between the analytic and computed rest mass density (normalized by the central rest mass density ρ_c) $\Delta\rho/\rho_c$, the Hamiltonian constraint H, and the x-momentum constraint M^x. On the right panel, we plot the difference between the analytic and computed specific energy density (normalized by the central specific energy density) $\Delta\epsilon/\epsilon_c$, the difference between the xx component of the extrinsic curvature and the analytic solution ΔK_{xx}, and the difference between the x component of the momentum and the analytic solution ΔS^x. In each case, we multiply the high resolution result by sixteen and the medium resolution by four to show second order convergence. All results are shown at $t=0.296$ μs which corresponds to eight iterations at the highest resolution. The graphs are taken along the $\hat{x}+\hat{y}+\hat{z}$ diagonal axis [results on the coordinate axis (x, y,z) are similar].

treatment for long term stability is presently being investigated.

Additional code calibrations that are underway include long-term stability analysis of single neutron stars, comparisons of wave forms from perturbed neutron stars, and comparisons with one-dimensional and axisymmetric (2D) independent GR-Hydro codes that we (together with our collaborators) constructed [74,39]. Those will be reported in later papers in this series.

The formulation of the coupled set of equations and the numerical code reported in this paper were used for the construction of the milestone code ''GR3D'' for the NASA Neutron Star Grand Challenge project (for a description of the project, see http://wugrav.wustl.edu/Relativ/nsgc.html). The goal of this project is to develop a code for general relativistic astrophysics, and in particular, one that is capable of simulating the inspiral coalescence of a neutron star binary system. The coalescences of neutron star binaries are expected to be important sources of gravitational waves for

interferometric detectors. The strongest signal will come from the highly dynamic ''plunge'' during the final phase of the inspiral; a fully general relativistic code provides the only way to calculate this portion of the wave form. A version of the code which passed the milestone requirement of the NASA Grand Challenge project, has recently been released to the community [75]. This code has been benchmarked at over 140 GFlop/sec on a 1024 node Cray T3E with a scaling efficiency of over 95%, showing the potential for large scale 3D simulations of realistic astrophysical systems. Further development of our general relativistic code, and its application to the specific study of the neutron star coalescence scenario, will be described in later papers in this series.

To summarize, this paper presents the first (and necessary) steps towards constructing an accurate and reliable tool for the numerical study of astrophysical phenomena involving matter at relativistic speeds and strong gravitational fields.

TABLE IX. This table summarizes the magnitudes of the errors for the different evolution systems used in the boosted NS test.

Abbreviation	$\max\left(\dfrac{\Delta\rho}{\rho_c}\right)$	$\max(H)$	$\max(M^x)$	$\max\left(\dfrac{\Delta\epsilon}{\epsilon_c}\right)$	$\max(\Delta K_{xx})$	$\max(\Delta S^x)$
ADMLEAP_ROE	6×10^{-5}	5×10^{-4}	5×10^{-5}	2×10^{-4}	1×10^{-5}	6×10^{-8}
ADMLEAP_FLUX	1×10^{-4}	5×10^{-4}	5×10^{-5}	1×10^{-3}	1×10^{-5}	1×10^{-7}
ADMLEAP_MAR	6×10^{-5}	5×10^{-4}	5×10^{-5}	2×10^{-4}	1×10^{-5}	6×10^{-8}
ADMICN_ROE	6×10^{-5}	5×10^{-4}	5×10^{-5}	2×10^{-4}	1×10^{-5}	6×10^{-8}
ADMICN_FLUX	1×10^{-4}	5×10^{-4}	5×10^{-5}	1×10^{-3}	1×10^{-5}	1×10^{-7}
ADMICN_MAR	6×10^{-5}	5×10^{-4}	5×10^{-5}	2×10^{-4}	1×10^{-5}	6×10^{-8}

[1] C. Bona, J. Massó, E. Seidel, and P. Walker, gr-qc/9804052.

[2] See AIP Document No. EPAPS: E-PRVDAQ-61-029004 for a comprehensive set of convergence tests for *every* combination of numerical methods used for evolving the hydrodynamics and spacetime. These sets of convergence tests are presented for every model mentioned in this paper. This document may be retrieved via the EPAPS homepage (http://www.aip.org/pubservs/epaps.html) or from ftp.aip.org in the directory /epaps/. See the EPAPS homepage for more information.

[3] Technical report (unpublished) for a summary of the present and planned high energy missions, see, e.g., http://heasarc.gsfc.nasa.gov/docs/heasarc/missions.html

[4] B. Paczynski, *Fourth Huntsville Gamma-Ray Burst Symposium* (American Institute of Physics, College Park, Maryland, 1997).

[5] Technical report (unpublished), NASA publication NP-152; Report on the ad hoc committee on gravitation physics and astrophysics, NASA, 1990.

[6] A. A. Abramovici *et al.*, Science **256**, 325 (1992).

[7] Report, Max Planck Institut fur quantenoptik, MPQ 177, 1993 (unpublished).

[8] K. Thorne, in *Proceedings of the Snowmass 95 Summer Study on Particle and Nuclear Astrophysics and Cosmology*, edited by E. W. Kolb and R. Peccei (World Scientific, Singapore, 1996).

[9] C. Cutler *et al.*, Phys. Rev. Lett. **70**, 2984 (1993).

[10] B. F. Schutz, Nature (London) **323**, 310 (1986).

[11] M. M. May and R. H. White, Phys. Rev. **141**, 1232 (1966).

[12] J. R. Wilson, Astrophys. J. **173**, 431 (1972).

[13] T. Nakamura, Prog. Theor. Phys. **65**, 1876 (1981).

[14] R. F. Stark and T. Piran, Phys. Rev. Lett. **55**, 891 (1985).

[15] C. Evans, in *Dynamical Spacetimes and Numerical Relativity*, edited by J. Centrella (Cambridge University Press, Cambridge, England, 1986), pp. 3–39.

[16] J. Hawley, L. Smarr, and J. Wilson, Astrophys. J. **277**, 296 (1984).

[17] L. I. Petrich, S. L. Shapiro, R. F. Stark, and S. A. Teukolsky, Astrophys. J. **336**, 313 (1989).

[18] J. Centrella and J. Wilson, Astrophys. J., Suppl. Ser. **54**, 229 (1984).

[19] J. R. Wilson, in *Sources of Gravitational Radiation*, edited by L. Smarr (Cambridge University Press, Cambridge, England, 1979), p. 275.

[20] J. Hawley, Astrophys. J. **381**, 496 (1991).

[21] P. G. Dykema, Ph.D. thesis, University of Texas at Austin, 1980.

[22] S. L. Shapiro and S. A. Teukolsky, Astrophys. J. **235**, 199 (1980).

[23] C. R. Evans, Ph.D. thesis, University of Texas at Austin, 1984.

[24] P. J. Schinder, S. A. Bludmann, and T. Piran, Phys. Rev. D **37**, 2722 (1988).

[25] A. Mezzacappa and R. A. Matzner, Astrophys. J. **343**, 853 (1989).

[26] T. Nakamura and M. Sasaki, Phys. Lett. **106B**, 69 (1981).

[27] A. M. Abrahams, G. B. Cook, S. L. Shapiro, and S. A. Teukolsky, Phys. Rev. D **49**, 5153 (1994).

[28] J. R. Wilson and G. J. Mathews, Phys. Rev. Lett. **75**, 4161 (1995).

[29] J. R. Wilson, G. J. Mathews, and P. Marronetti, Phys. Rev. D **54**, 1317 (1996).

[30] K.-I. Oohara and T. Nakamura, in *Relativistic Gravitation and Gravitational Radiation*, edited by J.-P. Lasota and J.-A. Marck (Cambridge University Press, Cambridge, England, 1997).

[31] B. J. van Leer, J. Comput. Phys. **23**, 276 (1977).

[32] R. J. Leveque, *Numerical Methods for Conservation Laws* (Birkhauser Verlag, Basel, 1992).

[33] M. L. Norman and K.-H. A. Winkler, in *Astrophysical Radiation Hydrodynamics*, edited by M. L. Norman and K.-H. A. Winkler (Reidel, Dordrecht, Holland, 1986).

[34] J. M. Martí, J. M. Ibáñez, and J. A. Miralles, Phys. Rev. D **43**, 3794 (1991).

[35] C. Bona, J. Ibáñez, J. Martí, and J. Massó, in *Gravitation and General Relativity: Rotating Bodies and Other Topics*, edited by F. Chinea, Lecture Notes in Physics Vol. 423 (Springer-

Verlag, New York, 1993), Chap. Shock Capturing Methods in 1D Numerical Relativity.

[36] J. V. Romero, J. M. Ibáñez, J. M. Martí, and J. A. Miralles, Astrophys. J. **462**, 839 (1996).

[37] J. M. Bardeen and T. Piran, Phys. Rep. **196**, 205 (1983).

[38] F. Banyuls *et al.*, Astrophys. J. **476**, 221 (1997).

[39] S. Brandt *et al.*, gr-qc/9807017.

[40] E. Gourgoulhon, Astron. Astrophys. **252**, 651 (1991).

[41] S. Bonazzola, E. Gourgoulhon, and J.-A. Marck, Phys. Rev. D **58**, 104020 (1998).

[42] J. Ibáñez and J. Martí, J. Comput. Appl. Math. **109**, 173 (1999).

[43] J. Balakrishna *et al.*, Class. Quantum Grav. **13**, L135 (1996).

[44] A. C. Calder, F. D. Swesty, and E. Y. M. Wang, in the proceedings of the Second Oak Ridge Symposium on Atomic and Nuclear Astrophysics, astro-ph/9806019.

[45] R. Arnowitt, S. Deser, and C. W. Misner, in *Gravitation: An Introduction to Current Research*, edited by L. Witten (John Wiley, New York, 1962), pp. 227–265.

[46] P. L. Roe, J. Comput. Phys. **43**, 357 (1981).

[47] F. Eulderink and G. Mellema, Astron. Astrophys. **284**, 652 (1994).

[48] R. Donat and A. Marquina, J. Comput. Phys. **125**, 42 (1996).

[49] E. Seidel and W.-M. Suen, J. Comput. Appl. Math. (to be published).

[50] The Message Passing Interface (MPI) standard, http://www.ncs.anl.gov/mpi/

[51] J. Massó and P. Walker (unpublished).

[52] J. M. Martí and E. Müller, J. Comput. Phys. **123**, 1 (1996).

[53] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics* (Pergamon, Oxford, 1987).

[54] C. Bona, in *Relativity and Scientific Computing*, edited by F. Hehl (Springer-Verlag, Berlin, 1996).

[55] O. Reula, Living Reviews in Relativity **1** (1998).

[56] R. D. Richtmyer and K. Morton, *Difference Methods for Initial Value Problems* (Interscience Publishers, New York, 1967).

[57] J. York, in *Sources of Gravitational Radiation*, edited by L. Smarr (Cambridge University Press, Cambridge, England, 1979).

[58] C. Bona, J. Massó, E. Seidel, and J. Stela, Phys. Rev. D **56**, 3405 (1997).

[59] B. J. van Leer, J. Comput. Phys. **32**, 101 (1979).

[60] J. Font, J. Ibáñez, J. Martí, and A. Marquina, Astron. Astrophys. **282**, 304 (1994).

[61] C. Hirsch, in *Numerical Computation of Internal and External Flows*, edited by R. Gallagher and O. Zienkiewicz (Wiley-Interscience, New York, 1992).

[62] C. W. Shu and S. J. Osher, J. Comput. Phys. **83**, 32 (1989).

[63] R. Donat, J. A. Font, J. M. Ibáñez, and A. Marquina, J. Comput. Phys. **146**, 58 (1998).

[64] J. M. Martí *et al.*, Astrophys. J. **479**, 151 (1997).

[65] G. Strang, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **5**, 506 (1968).

[66] V. Schneider *et al.*, J. Comput. Phys. **105**, 92 (1990).

[67] B. Schutz, *A First Course in General Relativity* (Cambridge University Press, Cambridge, England, 1985).

[68] K. C. B. New, K. Watt, C. W. Misner, and J. M. Centrella, Phys. Rev. D **58**, 064022 (1998).

[69] R. C. Tolman, Phys. Rev. **55**, 364 (1939).

[70] J. R. Oppenheimer and G. Volkoff, Phys. Rev. **55**, 374 (1939).

[71] S. L. Shapiro and S. A. Teukolsky, in *Dynamical Spacetimes and Numerical Relativity*, edited by J. M. Centrella (Cambridge University Press, Cambridge, England, 1986), pp. 74–100.

[72] C. W. Misner, K. S. Thorne, and J. A. Wheeler, *Gravitation* (Freeman, San Francisco, 1973).

[73] P. Lax, Commun. Pure Appl. Math. **9**, 135 (1956).

[74] J. A. Font and J. Massó, *Proceedings of the 8th Marcel Grossmann Meeting on General Relativity* (World Scientific, Singapore, 1997).

[75] The source and documentation of the released code can be downloaded at http://wugrav.wustl.edu/Codes/GR3D. For credit of the code development, see the document http://wugrav.wustl.edu/Codes/GR3D/nasa_ms2.ps