Conservative finite volume scheme for first-order viscous relativistic hydrodynamics

Alex Pandya,[1,2,*] Elias R. Most,[2,3,4,†] and Frans Pretorius[1,2,‡]

[1]*Department of Physics, Princeton University, Princeton, New Jersey 08544, USA*
[2]*Princeton Gravity Initiative, Princeton University, Princeton, New Jersey 08544, USA*
[3]*Princeton Center for Theoretical Science, Princeton University, Princeton, New Jersey 08544, USA*
[4]*School of Natural Sciences, Institute for Advanced Study, Princeton, New Jersey 08540, USA*

(Received 10 February 2022; accepted 13 May 2022; published 1 June 2022)

We present the first conservative finite volume numerical scheme for the causal, stable relativistic Navier-Stokes equations developed by Bemfica, Disconzi, Noronha, and Kovtun (BDNK). BDNK theory has arisen very recently as a promising means of incorporating entropy-generating effects (viscosity, heat conduction) into relativistic fluid models, appearing as a possible alternative to the so-called Müller-Israel-Stewart (MIS) theory successfully used to model quark-gluon plasma. The major difference between the two lies in the structure of the system of partial differential equations (PDEs): BDNK theory only has a set of conservation laws, whereas MIS also includes a set of evolution equations for its dissipative degrees of freedom. The simpler structure of the BDNK PDEs in this respect allows for rigorous proofs of stability, causality, and hyperbolicity in full generality which have as yet been impossible for MIS. To capitalize on these advantages, we present the first fully conservative multidimensional fluid solver for the BDNK equations suitable for physical applications. The scheme includes a flux-conservative discretization, nonoscillatory reconstruction, and a central-upwind numerical flux and is designed to smoothly transition to a high-resolution shock-capturing perfect fluid solver in the inviscid limit. We assess the robustness of our new method in a series of flat-spacetime tests for a conformal fluid and provide a detailed comparison with previous approaches of Pandya and Pretorius [Phys. Rev. D **104**, 023015 (2021)].

DOI: 10.1103/PhysRevD.105.123001

I. INTRODUCTION

Relativistic hydrodynamics is a general framework based on the notion that many substances, even if governed by vastly different physics on small spatiotemporal scales, may be well understood on sufficiently large scales by appealing to thermodynamics and conservation laws for the local energy, momentum, and baryon number [1]. This framework has resulted in the development of successful fluid models of even exotic substances, such as the quark-gluon plasma (QGP) produced in collisions of heavy ions, black hole accretion flows, and the matter composing neutron stars. Experimental breakthroughs in studying the QGP, in particular, have spurred significant growth in the theoretical understanding of relativistic fluids, as the Relativistic Heavy Ion Collider is now sufficiently sensitive to observe phenomena beyond the scope of ideal (non-dissipative) hydrodynamics [2]. Likewise, there are indications that similar phenomena may become relevant in modeling astrophysical sources for the next generation of telescopes and gravitational-wave observatories [3,4], further motivating theoretical focus on extending relativistic hydrodynamics beyond thermodynamic equilibrium.

The modern interpretation of hydrodynamics views it as the expansion of a given microscopic physical theory (such as, for example, kinetic theory) for small values of the Knudsen number, $\mathrm{Kn} \equiv \ell/L \ll 1$, which gives the ratio of the microscopic (ℓ) and macroscopic (L) length scales characterizing the system of interest. The primary advantage a fluid model provides lies in the notion that many of the microscopic degrees of freedom average out when $\ell \ll L$, often resulting in a vast simplification of the equations of motion and corresponding dynamics.

Though hydrodynamics is thought of in terms of the coarse graining of some microscopic theory, it is often the case that the microphysics of interest is not known and the fluid model is being used to make predictions in its stead. In these cases, hydrodynamics is thought of as a long-wavelength effective field theory for a system with $\mathrm{Kn} \ll 1$. Such a theory is typically defined by prescribing a definition for a set of conserved currents (the stress-energy tensor and a baryon current) in terms of "hydrodynamic variables" drawn from equilibrium thermodynamics, common examples of which include the temperature and

[*] apandya@princeton.edu
[†] emost@princeton.edu
[‡] fpretori@princeton.edu

particle density. This can be done by reference to the so-called gradient expansion, which assumes (1) that the system is sufficiently near equilibrium such that it may be parametrized using variables drawn from equilibrium thermodynamics and (2) that gradients of these parameters may be treated as small quantities. The latter assumption may be thought of as morally tantamount to the assumption that $\mathrm{Kn} \ll 1$, since the natural way to incorporate said derivative terms is through quantities of the form $\ell\nabla \sim \ell/L \equiv \mathrm{Kn}$.

In the gradient expansion approach, ideal (perfect fluid) hydrodynamics appears as the simplest such theory since all possible gradient terms are dropped from the definitions of the conserved currents. In this respect, ideal hydrodynamics is zeroth order in gradients, and perfect fluids are always in local thermodynamic equilibrium. Effects arising outside of equilibrium (e.g., viscosity, heat conduction) are neglected at zeroth order, but can be incorporated by adding to the perfect fluid conserved currents the set of first derivative corrections, defining what is known as first-order relativistic hydrodynamics.

In order to capture the nonequilibrium effects of viscosity and heat conduction, Eckart [5] and Landau and Lifshitz [6] independently put forth their namesake fluid theories, each of which arises at first order in the gradient expansion. Long before these theories could be applied, however, they were shown to be pathological [7,8], possessing acausal characteristics and unstable equilibrium states. At the time, these issues were incorrectly attributed to first-order hydrodynamics in general, leading to the widespread adoption of an alternative formulation known as Müller-Israel-Stewart (MIS) theory [9–11] which includes both first- and second-order gradient terms.[1] Though the original version of MIS theory has the same issues with its first-order terms (related to the so-called hydrodynamic frame, as will be discussed later), the problems with stability and causality were fixed by promoting the dissipative corrections to independent degrees of freedom complete with their own evolution equations. This additional structure allowed for proofs of causality and stability for the linearized theory [8], and MIS-type theories became the standard to model relativistic dissipative fluids for decades to come [16–19].

Inspired by a series of works due to Van and Biro [20,21] and Freistühler and Temple [22–24], Bemfica *et al.* [25,26], Kovtun [27], and Hoult and Kovtun [28] put forth a general first-order theory (which we will refer to here as "BDNK theory") free of the pathologies of the theories of Eckart and Landau-Lifshitz. The key insight of these works is that the choice of coefficients weighting the gradient terms—the so-called hydrodynamic frame[2]—must be made carefully: "good" choices of frame lead to causal, stable, strongly hyperbolic theories, and "bad" choices (such as those of Eckart and Landau-Lifshitz) result in unphysical ones. The success of MIS theory in this regard derives from the "relaxation" form taken by the evolution equations for the dissipative degrees of freedom, which allows them to temporarily deviate from their (potentially acausal or unstable) Navier-Stokes values [2]. However, MIS-type theories should not be viewed as entirely distinct from the BDNK approach. In fact, it can be shown that extensions of MIS theories to generalized hydrodynamic frames reduce to BDNK theory instead of (acausal) Navier-Stokes equations in the first-order limit [29]. More work toward understanding this connection [30] and in deriving fluid models from microphysical theories [31] is needed to clarify this connection.

BDNK theory possesses a number of features which may be viewed as advantages over MIS theory. Most apparent of these is that BDNK theory does not possess the additional (dissipative) degrees of freedom and corresponding evolution equations which characterize MIS-type approaches. This partial differential equation (PDE) structure allows for rigorous proofs of strong hyperbolicity, causality, and the stability of equilibrium states for the class of BDNK theories where the transport coefficients satisfy a set of frame constraints [26]. Similar proofs do not exist in full generality for MIS theories, and the few that do[3] give constraints which are functions of both the transport coefficients *and* the dissipative degrees of freedom. The latter are "dynamical" in the sense that they depend on the state of the fluid, and this is an important distinction compared to BDNK, which only has state-independent constraints (as such these constraints can be considered to be part of the microphysical description of the fluid, since that determines the transport coefficients). For MIS this juxtaposition implies constraints have to be monitored within a simulation to ensure they are not dynamically violated (a step which is often omitted in the literature, leading to, for example, a number of studies which evolved fluids in regimes where the characteristics of the equations had superluminal speeds [34]).

Frame complications also cause problems for physical scenarios relevant to astrophysics: MIS breaks down when sufficiently strong (high Mach number) shock waves form [35], whereas arbitrarily strong shock wave solutions exist in BDNK theory for well-chosen frames [36,37]. On the

[1] MIS-like theories were specifically adopted to deal with the problems of Eckart and Landau and Lifshitz theories, not because second-order terms were thought to be relevant; in fact, most studies (e.g., [12,13] in astrophysics, and [14,15] in nuclear physics) drop many of these terms to simplify computations.

[2] Here we will use the term "frame" to refer to hydrodynamic frame; when specifying a Lorentz (coordinate) frame, we will use the terms "reference frame" or "rest frame."

[3] MIS theory has been proven to be causal subject to a set of dynamical constraints, locally well-posed for Gevrey initial data when heat conduction and particle diffusion are neglected [32], and hyperbolic when all dissipative effects except bulk viscosity are dropped [33].

other hand, astrophysical problems, by and large, require the consistent inclusion of magnetic fields into the evolution, which is currently only known for MIS-type dissipative theories [38–42]. Despite the lack of causality constraints, such theories have been successfully used in astrophysical studies [43,44].

Furthermore, MIS theory has the benefit of a strong base of existing numerical infrastructure developed to model heavy-ion collisions [15,45,46]. The existence of evolution equations for the dissipative corrections leaves the principal part of the conservation laws unchanged, allowing one to solve these equations in largely the same way as in ideal hydrodynamics. The equations used to evolve the dissipative corrections require novel methods, however, these problems were thoroughly addressed in the heavy-ion literature mentioned previously. BDNK theory, on the other hand, has no additional degrees of freedom beyond ideal hydrodynamics and hence only has the stress-energy and particle current conservation laws; however, these equations are not amenable to the numerical methods of ideal hydrodynamics due to the presence of first derivative terms in the conserved currents. Thus novel approaches are required to apply BDNK theory in numerical models of physical systems.

In this study, we build upon the exploratory work of [36] to provide a BDNK evolution scheme with enhanced stability for flows with high Lorentz factors, strong shock waves, and near-vacuum states, all of which arise generically in many physical systems of interest. The scheme is based on a flux-conservative finite volume discretization complete with nonoscillatory primitive variable reconstruction and a central-upwind numerical flux function. We also address problems unique to the BDNK equations and present a novel algorithm for primitive variable recovery capable of handling the numerically difficult inviscid limit, wherein the BDNK equations reduce to the relativistic Euler equations. In said limit, our scheme reduces exactly to a high-resolution shock-capturing (HRSC) finite volume perfect fluid solver.

We structure the remainder of the study as follows: In Sec. II, we begin with a brief overview of relativistic hydrodynamics and define the BDNK conserved currents. To clarify our presentation, we then specialize to a fluid with simple microphysics (namely conformal symmetry) and trivial spacetime geometry (4D Minkowski spacetime). In Sec. III, we outline our numerical method. More precisely, we review the finite volume method and explain how it is applied first in ideal hydrodynamics, then how we adapt it to a BDNK fluid. The new code's performance is evaluated in a set of problems with variation in one and two spatial dimensions, designed to test constraint preservation, stability for high-velocity flows with shock waves, and the behavior of solutions approaching the inviscid limit; these tests are presented in Sec. IV. In Sec. V we conclude with avenues for future work. In the Appendixes we list the BDNK conserved currents in our chosen coordinate basis, give an overview of how to generalize the scheme beyond conformal symmetry, review the weighted essentially nonoscillatory (WENO) algorithm, and show examples of our scheme's convergence properties.

II. EQUATIONS OF MOTION

Relativistic fluid models are typically defined through two conserved currents: the stress-energy tensor T^{ab} and a baryon current J^a, each of which must obey

$$\nabla_a T^{ab} = 0, \tag{1}$$

$$\nabla_a J^a = 0. \tag{2}$$

Hydrodynamics models a substance's long-wavelength behavior by asserting that all of the microphysical degrees of freedom average out at the scales of interest, such that in d spacetime dimensions the $d+1$ constraints (1) and (2) are sufficient to specify its macroscopic evolution. These $d+1$ constraints are then interpreted as a set of evolution equations for $d+1$ state variables, which are typically drawn from the laws of thermodynamics, leaving the fluid theory largely agnostic of the microphysics it is approximating. Typical choices of these hydrodynamic variables include the local flow four-velocity u^a (assumed to be timelike), as well as a set of scalar variables which are related to other, similar quantities by the laws of thermodynamics. Here we will use the energy density ϵ and the baryon number density n, though these are occasionally replaced by the temperature T and chemical potential μ in the literature.

Asserting that an observer comoving with the fluid sees energy density ϵ, an isotropic pressure P, and a number density n, gives the "perfect fluid" (ideal) stress-energy tensor and particle current

$$T_0^{ab} = \epsilon u^a u^b + P\Delta^{ab}, \tag{3}$$

$$J_0^a = n u^a, \tag{4}$$

where the tensor

$$\Delta^{ab} = g^{ab} + u^a u^b \tag{5}$$

projects onto the space orthogonal to u^a. Combining (3) and (4) with (1) and (2) yields the relativistic Euler equations. Note that (3) and (4) have $d+2$ parameters, rather than $d+1$—this implies that an additional closure relation, $P(\epsilon, n)$, is required; the definition of $P(\epsilon, n)$ is known as the equation of state.

One can better understand the thermodynamic properties of the perfect fluid if one takes the projection of the relativistic Euler stress-energy conservation equation (1) and (3) along u_b,

$$\nabla_a[(\epsilon + P)u^a] = u^a \nabla_a P. \tag{6}$$

Applying the thermodynamic relation $dP = sdT + nd\mu$, where s is the entropy density, then adding $\mu\nabla_a J_0^a = 0$ to the right-hand side results in

$$\nabla_a(\epsilon + P - \mu n)u^a = u^a s \nabla_a T. \tag{7}$$

Using the first law of thermodynamics in intrinsic form, $\epsilon + P - \mu n = Ts$, one arrives at the result

$$\nabla_a(su^a) = 0, \tag{8}$$

which may be interpreted as a conservation law for the entropy current, implying that *entropy is conserved in ideal hydrodynamics*.[4] Hence a different theory is required to incorporate entropy-producing dissipative processes such as heat conduction (diffusion due to thermal gradients) or viscosity (momentum transfer due to velocity gradients).

A natural place to start when constructing a dissipative fluid theory, then, is by including gradient terms in T^{ab} and J^a. Formally, this procedure may be thought of as an expansion about thermodynamic equilibrium. Since local fluctuations in the hydrodynamic variables are expected to be small, their gradients are assumed to be small as well, allowing one to stratify gradient corrections to the conserved currents by the order of derivative terms (or by the exponent of such terms, if they appear nonlinearly). Explicitly, this approach asserts that the "true" conserved currents (from the full microphysical theory, whatever that may be) can be written in a gradient expansion[5]

$$\begin{aligned} T^{ab} &= T_0^{ab} + O(\nabla) + O(\nabla^2) + \cdots, \\ J^a &= J_0^a + O(\nabla) + O(\nabla^2) + \cdots, \end{aligned} \tag{9}$$

where the $O(\nabla)$ terms include only first gradients, the $O(\nabla^2)$ terms include second gradients and products of first gradients, and so on. In practice, it is impossible to construct T^{ab}, J^a up to infinite order in derivative terms, so one typically truncates them after including all such terms up to a given order[6] n, denoted here with a subscript,

$$\begin{aligned} T_n^{ab} &= T_0^{ab} + O(\nabla) + \cdots + O(\nabla^n), \\ J_n^a &= J_0^a + O(\nabla) + \cdots + O(\nabla^n). \end{aligned} \tag{10}$$

By construction, the perfect fluid stress-energy tensor and particle current are recovered when all derivative corrections to T^{ab}, J^a are dropped; in other words, the perfect fluid arises from truncating the gradient expansion at zeroth order, keeping only T_0^{ab}, J_0^a.

BDNK theory [26] arises when truncating the gradient expansion at first order,[7] and defines T_1^{ab}, J_1^a by taking linear combinations of all allowed one-derivative terms, weighted by zero-derivative transport coefficients. In [26], the authors also provide a set of conditions on these transport coefficients which guarantee the theory be strongly hyperbolic, causal, consistent with the second law of thermodynamics within the regime of validity of the gradient expansion [e.g., $\nabla_a(su^a) \geq 0 + O(\nabla^3)$, cf. (8)], and have stable equilibrium states.

The BDNK conserved currents are

$$T_1^{ab} = (\epsilon + \mathcal{A})u^a u^b + (P + \Pi)\Delta^{ab} + \mathcal{Q}^a u^b + \mathcal{Q}^b u^a - 2\eta\sigma^{ab} \tag{11}$$

and

$$J_1^a = \mathcal{N} u^a + \mathcal{J}^a, \tag{12}$$

where each has dissipative contributions linear in gradients of the hydrodynamic variables. These gradient corrections to T^{ab} are defined to be

$$\mathcal{A} = \tau_\epsilon[u^c \nabla_c \epsilon + (\epsilon + P)\nabla_c u^c], \tag{13}$$

$$\Pi = -\zeta \nabla_c u^c + \tau_P[u^c \nabla_c \epsilon + (\epsilon + P)\nabla_c u^c], \tag{14}$$

$$\mathcal{Q}^a = \tau_Q(\epsilon + P)u^c \nabla_c u^a + \beta_\epsilon \Delta^{ac} \nabla_c \epsilon + \beta_n \Delta^{ac} \nabla_c n, \tag{15}$$

and

[4] Entropy can increase when shocks are present, though. In these cases, the physical solution is given by the weak formulation of the equations. Since the weak solution may not be unique, the physical one is that which satisfies the second law of thermodynamics [47].

[5] Though in most cases it is unknown if the gradient expansion converges, there are many known examples where it does not [48–51]. Remarkably, in some cases even beginning with far from equilibrium and varied initial conditions, solutions still approach a similar ideal hydrodynamic evolution at late times, a phenomenon typically attributed to the existence of a universal attractor solution [52,53].

[6] Though one can certainly write down higher-order gradient theories (see, e.g., the third-order theory of [54]), the gradient expansion written in (10) is not meant to imply that higher-order theories are always strictly "better" than lower-order ones. In fact, it has been shown that inconsistencies arise already at second order in gradients, both in nonrelativistic contexts (super-Burnett theory) [55] and relativistic contexts (MIS theory) [56].

[7] The aforementioned MIS theory may be interpreted as arising from a truncation at second order in gradients, though it was not originally derived by reference to a gradient expansion; see [16,36].

$$\sigma^{ab} = \frac{1}{2}\left(\Delta^{ac}\Delta^{bd}\nabla_c u_d + \Delta^{ac}\Delta^{bd}\nabla_d u_c - \frac{2}{3}\Delta^{ab}\Delta^{cd}\nabla_c u_d\right), \tag{16}$$

which are the correction to the energy density, the bulk viscous pressure, the heat flow vector, and the shear tensor, respectively. It turns out that one may drop all gradient corrections to the particle current without compromising the hyperbolicity, causality, or thermodynamic stability properties of the resulting PDEs [26]. We make such a choice here and set

$$\mathcal{N} = n, \qquad \mathcal{J}^a = 0. \tag{17}$$

Note that the particle current (12) and (17) now takes the same form as in ideal hydrodynamics, and that one recovers the perfect fluid stress-energy tensor (3) from (11) if the gradient corrections $\mathcal{A}, \Pi, \mathcal{Q}^a, \eta\sigma^{ab}$ all vanish.

Each of the gradient terms is linear in one of the transport coefficients, which themselves are free of derivatives and are derived from the thermodynamics of the specific substance being modeled. Inspection of the terms above shows that these coefficients fall into three categories. The first are thermal transport coefficients,

$$\beta_\epsilon = \tau_Q\left(\frac{\partial P}{\partial \epsilon}\right)_n + \frac{\sigma T(\epsilon+P)}{n}\left(\frac{\partial(\mu/T)}{\partial\epsilon}\right)_n,$$
$$\beta_n = \tau_Q\left(\frac{\partial P}{\partial n}\right)_\epsilon + \frac{\sigma T(\epsilon+P)}{n}\left(\frac{\partial(\mu/T)}{\partial n}\right)_\epsilon, \tag{18}$$

which depend on derivatives of the equation of state $P(\epsilon, n)$ and chemical potential divided by the temperature, $\frac{\mu}{T}(\epsilon, n)$, which must be computed via the laws of thermodynamics. Next are the transport coefficients corresponding to well-known dissipative effects, namely the shear viscosity η, bulk viscosity ζ, and thermal conductivity σ (which appears in β_ϵ, β_n). Finally there are a set of three relaxation times τ_ϵ, τ_Q, τ_P which set the dissipative timescales.

Here, as in [36], for the sake of simplicity we specialize to a fluid with an underlying conformal symmetry (which requires $g_{ab}T^{ab} = 0$) and we set $\mu = 0$. Together these conditions[8] imply

$$P(\epsilon, n) = \frac{\epsilon}{3}, \qquad \Pi = \frac{\mathcal{A}}{3}, \qquad \zeta = 0,$$
$$\beta_\epsilon = \frac{\tau_Q}{3}, \qquad \beta_n = 0, \qquad \tau_P = \frac{\tau_\epsilon}{3}, \tag{19}$$

leaving us with only the shear viscosity η and the two relaxation times τ_ϵ, τ_Q. Note that the stress-energy tensor is now free of n, and hence J_1^a (12) and T_1^{ab} (11) decouple. In [36], this fact is used to neglect the evolution of the particle current; here we choose to evolve J_1^a so that we may use the number density n as an intuitive marker of the fluid's behavior in the tests below.

For the remaining three transport coefficients we follow the same prescription as in [36], adopting natural units and writing them as

$$\eta \equiv \eta_0\epsilon^{3/4},$$
$$\tau_\epsilon = \frac{3}{4\epsilon}\chi \equiv \frac{3}{4\epsilon}\chi_0\epsilon^{3/4},$$
$$\tau_Q = \frac{3}{4\epsilon}\lambda \equiv \frac{3}{4\epsilon}\lambda_0\epsilon^{3/4}, \tag{20}$$

where τ_ϵ, τ_Q are exchanged for χ, λ to parallel the notation of [25], and then are written with the ϵ dependence pulled out. Writing them in this way allows us to use the dimensionful constant η_0 as a free parameter controlling the amount of viscosity in the model, and the remaining two constants (χ_0, λ_0) determine the so-called hydrodynamic frame. In accordance with [36], we choose the frame

$$(\chi_0, \lambda_0) = \left(\frac{25}{4}\eta_0, \frac{25}{7}\eta_0\right), \tag{21}$$

which fixes the characteristic speeds to be exactly unity. This choice is consistent with the conditions of [25] which establish existence and uniqueness of solutions, causality, and linear stability about thermodynamic equilibrium, and those of [37] establishing the existence of smooth strong shock solutions.

With the transport coefficients written in the form (21), the limit $\eta_0 \to 0$ results in χ_0, $\lambda_0 \to 0$, and all dissipative corrections in the BDNK stress-energy tensor (13)–(16) vanish, reducing the BDNK conserved currents exactly to those of the perfect fluid (3) and (4). We refer to $\eta_0 \to 0$ as the "inviscid limit" of BDNK theory.

In the work that follows, we further specialize to a fluid in 4D Minkowski spacetime and adopt a Cartesian coordinate system $x^a = (t, x, y, z)^T$. To limit computational cost, we only consider test problems with variation in one spatial dimension (t, x) or two spatial dimensions (t, x, y). The components of T_1^{ab}, J_1^a in these coordinates are rather long, so we relegate them to Appendix A.

In the following section we review the finite volume method and how it is typically applied to the perfect fluid equations of motion (1)–(4), before adapting it to the BDNK equations (1), (2), (11)–(17).

III. NUMERICAL ALGORITHM

In this section we outline the finite volume method, then describe how one casts the relativistic Euler and BDNK equations into the conservative form required for its

[8] A conformal fluid with $\mu = 0$ is often used as a simple toy model for the QGP produced in heavy-ion collisions; for more on conformal fluids, see [16].

application. We then detail the steps in the finite volume algorithm as they are applied to the BDNK equations in 4D Minkowski spacetime, assuming one is only interested in problems with variation in two spatial dimensions (though the methods straightforwardly generalize to higher-dimensional problems).

In Sec. III C we discuss primitive variable recovery, which is trivial for BDNK theory, as the BDNK stress-energy tensor is linear in its primitive variables. Care is required to apply this analytic solution for small viscosities, however, as the solution breaks down in cells where the viscous terms are unresolved (smaller in magnitude than truncation error). We detail an adaptive algorithm which applies the perfect fluid's primitive variable solver in unresolved cells, allowing for a stable evolution at such "low" resolutions.

Section III D outlines the reconstruction of primitive variables, for which we use the WENO method [57]. For BDNK one must also compute spatial derivatives of primitive variables prior to reconstruction, and for that we use a method based on the central-WENO approach [58]. For the numerical fluxes, we use the Kurganov-Tadmor flux function [59] and set the maximum local speed $a = 1$, which is the exact local characteristic speed for the BDNK equations in the chosen hydrodynamic frame (21).

We conclude this section with a brief discussion of the time integration algorithm used (Heun's method) and comment on the choice of boundary conditions.

A. Finite volume method

Though it is not yet clear whether the BDNK equations possess sensible discontinuous shock wave solutions, it is well known that such solutions are essential in modeling perfect fluids, which arise in the inviscid limit of BDNK theory. As a result, it would be preferable to develop a numerical method for the BDNK equations which can stably evolve solutions with sharp gradients. For this reason, we adopt a "finite volume" discretization of the BDNK PDEs. Before doing so, however, we will first briefly review finite volume methods in the context of relativistic fluid mechanics.

Finite volume methods are adapted to the solution of hyperbolic conservation laws, which in general may be written in the form

$$\frac{\partial}{\partial t}\boldsymbol{q}(\boldsymbol{p}) + \frac{\partial}{\partial x^i}\boldsymbol{f}^i(\boldsymbol{p}) = \boldsymbol{\psi}(\boldsymbol{p}), \tag{22}$$

where $\boldsymbol{q}$ is a vector of conservative variables, $\boldsymbol{f}$ is the flux tensor, $\boldsymbol{\psi}$ is a vector of sources, and each of the aforementioned terms is a function of the vector of primitive variables $\boldsymbol{p}$.

Assuming one wants to solve the conservation law (22) over a spatial domain $\mathcal{D}$ (which we will take to be two-dimensional), one divides $\mathcal{D}$ into subdomains $\mathcal{S}_{i,j}$, which we will define to be rectangular with extent $|\mathcal{S}_{i,j}| = \Delta x \Delta y$, centered at the point (x, y). One may integrate (22) inside a given subdomain to get

$$\frac{\partial}{\partial t}\bar{\boldsymbol{q}}_{i,j} + \frac{\langle \boldsymbol{f}^x_{i+\frac{1}{2},j}\rangle - \langle \boldsymbol{f}^x_{i-\frac{1}{2},j}\rangle}{\Delta x} + \frac{\langle \boldsymbol{f}^y_{i,j+\frac{1}{2}}\rangle - \langle \boldsymbol{f}^y_{i,j-\frac{1}{2}}\rangle}{\Delta y} = \bar{\boldsymbol{\psi}}_{i,j}, \tag{23}$$

where we have introduced the shorthand

$$\bar{X}_i = \frac{1}{\Delta x \Delta y}\int_{\mathcal{S}_{i,j}} X dx dy, \tag{24}$$

$$\langle X^k \rangle = \int_{\partial \mathcal{S}_{i,j}} X^k dx^l, \quad k \neq l, \tag{25}$$

so an overbar denotes a cell-averaged quantity, and angle brackets denote a flux in direction k averaged over the face of the cell at constant coordinate x^k.

As written, (23) may be interpreted as a semidiscrete evolution equation for the cell averages $\bar{\boldsymbol{q}}_{i,j}$ after choosing suitable discrete approximation to the integrals (24) and (25) and reinterpreting all variables as discrete grid functions defined on the cells $\mathcal{S}_{i,j}$.

By discretizing the integral form of the conservation law (23) rather than the PDE (22), the finite volume method enjoys a number of advantages over other methods (such as the finite-difference method[9] used in [36]). Most important of these for our purposes is the ability to stably evolve shock wave solutions, which are generically discontinuous for inviscid flows. Such solutions satisfy the weak (integral) form of the equations (23) but not the continuum PDE (22), and hence may be recovered by virtue of discretizing (23) rather than (22). It is important to note, though, that the precise choice of discretization for the numerical fluxes $\boldsymbol{F}$ approximating $\langle \boldsymbol{f} \rangle$ has a significant impact on stability as well as constraint preservation (namely conservation of $\boldsymbol{q}$, modulo sources $\boldsymbol{\psi}$, across the simulation domain, and preservation of the irrotational nature of the spatial gradients $\partial_i \boldsymbol{p}$); these topics are discussed further in Sec. III E.

Since the conservation law (23) is discretized over a finite-sized cell (rather than, say, a zero-volume point, as is the case for the finite-difference method), there are a number of additional considerations which appear when solving the discrete equations. Consider taking a time step of the latter beginning at time level index n (either from initial data specified then or after a prior successful time step); at this point, the cell-averaged conservative variables $\bar{\boldsymbol{q}}^n_{i,j}$ are known at time level n for all of the spatial grid points indexed with i, j. To use the discrete version of (23)

[9] It is also possible to construct a conservative finite difference scheme, provided one defines the flux derivative term by reference to integrals of the flux; see [57,60].

to find the conservative variables at the next time level $\bar{\boldsymbol{q}}^{n+1}_{i,j}$, one has to take the following steps:

(1) To solve (23), one needs to compute the flux terms $\langle \boldsymbol{f} \rangle$ and the source term $\bar{\boldsymbol{\psi}}$, which are functions of the primitive variables $\boldsymbol{p}$. This is done by inverting the definitions of the (known) conservative variables $\boldsymbol{q}(\boldsymbol{p})$ to find $\bar{\boldsymbol{p}}^n_{i,j}$. This step is known as "primitive variable recovery."

(2) Once the primitive variables are known, the source term $\bar{\boldsymbol{\psi}}^n_{i,j}$ can be trivially computed. Computing the flux terms is not so straightforward, however, since these are averaged over cell faces (25) and the primitive variables we have computed are cell averages $\bar{\boldsymbol{p}}^n_{i,j}$. Hence one must interpolate the primitive variables from the cell average $\bar{\boldsymbol{p}}^n_{i,j}$ to the interfaces $\langle \boldsymbol{p}^n_{i,j} \rangle$ in a step known as "reconstruction."

(3) Using the interface-averaged primitive variables $\langle \boldsymbol{p}^n_{i,j} \rangle$ one can finally compute the numerical fluxes approximating $\langle \boldsymbol{f} \rangle$. The discretization (23) may now be solved for the cell-averaged conservative variables at the next time level $\bar{\boldsymbol{q}}^{n+1}_{i,j}$.

In the following sections we explain in detail how each of the above steps is carried out, first for the relativistic Euler equations, and then for the BDNK equations. We begin by casting both sets of equations into conservative form (22), then address primitive variable recovery, reconstruction, and numerical flux computation in successive subsections. We conclude the section with a brief discussion of the time integration algorithm and the types of boundary conditions implemented for the numerical tests which follow.

B. Relativistic fluid equations in conservative form

Both the relativistic Euler equations (1)–(4) and the BDNK equations (1), (2), and (11)–(17) can be cast into the form (22) in the same way. Combining the different components of the equation into vectors, one can write (1) as

$$\boldsymbol{q} = \begin{pmatrix} T^{tt} \\ T^{tx} \\ T^{ty} \end{pmatrix}, \quad \boldsymbol{f}^x = \begin{pmatrix} T^{tx} \\ T^{xx} \\ T^{yx} \end{pmatrix}, \quad \boldsymbol{f}^y = \begin{pmatrix} T^{ty} \\ T^{xy} \\ T^{yy} \end{pmatrix}, \quad \boldsymbol{\psi} = \boldsymbol{0}, \qquad (26)$$

where each equation comes from a row of the vectors above. For example, the first equation is $T^{tt}_{,t} + T^{tx}_{,x} + T^{ty}_{,y} = 0$.

The particle current conservation law (2) is a scalar equation and may be written

$$q = J^t, \qquad f^x = J^x, \qquad f^y = J^y, \qquad \psi = 0; \qquad (27)$$

as mentioned before, the particle current is identical between the relativistic Euler and BDNK equations. As a result, one may evolve the particle current (and hence n) forward through time using standard methods used to solve the equations of ideal hydrodynamics. In the sections that follow, we focus on the methods used to solve (1). After defining these methods, we briefly summarize how they are applied to solve (2) in Sec. III F.

Though (26) appears to be essentially identical between the zeroth- and first-order theories, differences arise in the primitive variable recovery step (because each has a different set of primitive variables), in reconstruction, as well as in the computation of the flux terms. These differences will be described in the following three subsections.

C. Primitive variable recovery

For a conformal fluid in 4D Minkowski spacetime with variation in (t, x, y), the set of primitive variables for the perfect fluid (3) and (4) are

$$\boldsymbol{p}_0 = \begin{pmatrix} \epsilon \\ u^x \\ u^y \end{pmatrix}. \qquad (28)$$

The primitive variable solution $\boldsymbol{p}_0(\boldsymbol{q}_0)$ can be carried out analytically in this case and is given by

$$\begin{aligned} &\epsilon = -T^{tt} + \sqrt{6(T^{tt})^2 + 3[(T^{tt})^2 - (T^{tx})^2 - (T^{ty})^2]}, \\ &|v| = \frac{\sqrt{(T^{tx})^2 + (T^{ty})^2}}{T^{tt} + 3\epsilon}, \qquad u^t = \frac{1}{\sqrt{1 - |v|^2}}, \\ &u^x = \frac{3u^t T^{tx}}{3T^{tt} + \epsilon}, \qquad u^y = \frac{3u^t T^{ty}}{3T^{tt} + \epsilon}. \end{aligned} \qquad (29)$$

It is important to stress that, in general, the primitive variable solution analogous to (29) cannot be found analytically; the fact that it can be here is a result of conformal symmetry, the choice of Cartesian coordinates, and the flat-spacetime background.

To write the BDNK equations—which, unlike the relativistic Euler equations, are second-order PDEs—in conservative form, one must perform a first-order reduction, defining the BDNK primitive variables in terms of time derivatives of the hydrodynamic variables. Explicitly, one such choice would be to take $\boldsymbol{p}_1 = \dot{\boldsymbol{p}}_0 = (\dot{\epsilon}, \dot{u}^x, \dot{u}^y)$, where an overdot is shorthand for a time derivative, $\dot{X} \equiv \partial_t X$. Here, for improved stability[10] we evolve $\xi \equiv \ln(\epsilon)$ instead of ϵ, and hence we take the BDNK primitive variables to be

[10] We find that the primitive variables of [36], $\epsilon \in (0, \infty)$, $v^i \in (-1, 1)$ can reach unphysical values as a result of numerical error in the primitive variable recovery step. To avoid this problem, we instead evolve $\xi \equiv \ln(\epsilon)$ and u^i, whose values are physical for $\xi, u^i \in (-\infty, \infty)$.

$$\boldsymbol{p}_1 = \begin{pmatrix} \dot{\xi} \\ \dot{u}^x \\ \dot{u}^y \end{pmatrix}. \tag{30}$$

Performing the first-order reduction implies that the system (22) and (26) must be augmented with a set of trivial evolution equations used to update the hydrodynamic variables given their time derivatives; in this case, these equations are

$$\frac{\partial \xi}{\partial t} = \dot{\xi}, \qquad \frac{\partial u^x}{\partial t} = \dot{u}^x, \qquad \frac{\partial u^y}{\partial t} = \dot{u}^y. \tag{31}$$

Brief inspection of (11) would seem to imply that the primitive variable recovery would be very difficult for the BDNK equations, as the definition of the stress-energy tensor is much more complicated than it is in the perfect fluid case (3), where the primitive variable solution is generally impossible to perform analytically. It turns out, however, that since T_1^{ab} is linear in gradient terms by construction, (11) is actually of the form

$$\boldsymbol{q}_1 = \boldsymbol{q}_0(\boldsymbol{p}_0) + \eta_0[\boldsymbol{A}(\boldsymbol{p}_0)\cdot\boldsymbol{p}_1 + \boldsymbol{b}(\boldsymbol{p}_0, \partial_i\boldsymbol{p}_0)], \tag{32}$$

where we will use uppercase bold letters to denote matrices and lowercase bold letters for vectors. Written in the form (32), it is clear that

$$\boldsymbol{p}_1 = \boldsymbol{A}^{-1}\cdot\left[\frac{1}{\eta_0}(\boldsymbol{q}_1 - \boldsymbol{q}_0) - \boldsymbol{b}\right], \tag{33}$$

so the BDNK primitive variable solution can always[11] be obtained analytically. In this sense, primitive variable recovery is actually *simpler* for BDNK than it is for the relativistic Euler equations.

Though it is straightforward to derive the BDNK primitive variable solution, (33) cannot be naively applied in all cases of interest. In particular, the limit $\eta_0 \to 0$ causes significant problems in numerical simulations, where truncation error $\boldsymbol{\tau}$ is introduced and (32) becomes

$$\boldsymbol{q}_1 = \boldsymbol{q}_0(\boldsymbol{p}_0) + \eta_0[\boldsymbol{A}(\boldsymbol{p}_0)\cdot\boldsymbol{p}_1 + \boldsymbol{b}(\boldsymbol{p}_0, \partial_i\boldsymbol{p}_0)] + \boldsymbol{\tau}. \tag{34}$$

Note that truncation error appears as an additional correction to $\boldsymbol{q}_0$, much like the viscous term proportional to η_0; in this sense, $\boldsymbol{\tau}$ may be thought of as the contribution of "numerical viscosity" to the solution. Solving for $\boldsymbol{p}_1$ becomes difficult in cases where η_0 is so small that $\eta_0[\boldsymbol{A}\cdot\boldsymbol{p}_1 + \boldsymbol{b}] \lesssim \boldsymbol{\tau}$, as (34) effectively becomes inviscid up to truncation error,

$$\boldsymbol{q}_1 \approx \boldsymbol{q}_0(\boldsymbol{p}_0) + \boldsymbol{\tau}, \tag{35}$$

and naive application of (33) yields

$$\boldsymbol{p}_1 \approx \boldsymbol{A}^{-1}\cdot\left[\frac{\boldsymbol{\tau}}{\eta_0} - \boldsymbol{b}\right], \tag{36}$$

where the first term is numerical error amplified by the large factor η_0^{-1}. This problem may be stated succinctly as follows: the BDNK primitive variable solution (33) breaks down whenever the numerical viscosity is comparable to or larger than the physical viscosity.

In principle, one may be interested in solving the BDNK equations for arbitrarily small viscosities at finite grid resolution. Here we present an adaptive algorithm to handle such cases, where cells in which the physical viscosity is unresolved [cf. (35)] use the perfect fluid primitive variable solution, and those where it is resolved use a variant of (33). The criterion used to designate a cell as viscous or inviscid preferentially uses the former as resolution is increased, eventually using the viscous solution exclusively at sufficiently high resolution. This process should provide stable results at low resolution which converge to solutions of the continuum BDNK PDEs as the grid is refined.

To develop this adaptive scheme, we begin by examining the expected behavior of $\boldsymbol{p}_1$ at $\eta_0 = 0$, where (33) is indeterminate. When $\eta_0 = 0$, the time derivative terms $\boldsymbol{p}_1$ do not appear in the conservative variables, but instead in the equations of motion, which are linear in said terms and may be written (in nonconservative form) as

$$\boldsymbol{p}_1^{PF} = \boldsymbol{c}(\boldsymbol{p}_0, \partial_i\boldsymbol{p}_0), \tag{37}$$

where the superscript PF has been appended to denote that these variables are computed using the perfect fluid equations of motion. Ideally, one would want (33) to give $\boldsymbol{p}_1 \to \boldsymbol{p}_1^{PF}$ as $\eta_0 \to 0$; this can be done in practice by defining a new set of variables,[12]

$$\tilde{\boldsymbol{q}}_1 \equiv \boldsymbol{q}_1 - \boldsymbol{q}_1|_{\boldsymbol{p}_1\to\boldsymbol{p}_1^{PF}} = \eta_0\boldsymbol{A}\cdot(\boldsymbol{p}_1 - \boldsymbol{p}_1^{PF}), \tag{38}$$

where the second equality comes from applying (32). Inverting $\tilde{\boldsymbol{q}}_1(\boldsymbol{p}_1)$ yields

$$\boldsymbol{p}_1 = \frac{1}{\eta_0}\boldsymbol{A}^{-1}\cdot\tilde{\boldsymbol{q}}_1 + \boldsymbol{p}_1^{PF}. \tag{39}$$

As written, (39) suffers from the same problem as (33)—truncation error appearing in $\boldsymbol{A}^{-1}\cdot\tilde{\boldsymbol{q}}_1$ destabilizes the

[11] The primitive variable solution (33) requires $\boldsymbol{A}^{-1}$ to exist, which is always the case for physical values of the hydrodynamic variables in the chosen hydrodynamic frame.

[12] Note that $\tilde{\boldsymbol{q}}_1$ is not evolved; the standard conservative variables $\boldsymbol{q}_1$ are evolved via (1) and (2), and the shifted variables $\tilde{\boldsymbol{q}}_1$ are computed from $\boldsymbol{q}_1$ via (38) during the primitive variable recovery step.

scheme when η_0 is sufficiently small. To address this issue, we use (39) in the following algorithm:

(1) Compute an estimate for the numerical viscosity, which we use to define the "viscous tolerance" Δ_η.
(2) Compute $\tilde{\boldsymbol{q}}_1$ using (38).
(3) Compare $\tilde{\boldsymbol{q}}_1$ to Δ_η:
   (a) if $\tilde{\boldsymbol{q}}_1 \geq \Delta_\eta$, use (39) as-is to find $\boldsymbol{p}_1$. Update $\boldsymbol{p}_0$ terms using the trivial evolution equations (31).
   (b) if $\tilde{\boldsymbol{q}}_1 < \Delta_\eta$, use (39) with $\tilde{\boldsymbol{q}}_1 = 0$ to compute $\boldsymbol{p}_1$. Since the conservation law (1) decouples from (39) when $\tilde{\boldsymbol{q}}_1 = 0$, one must update $\boldsymbol{p}_0$ using the perfect fluid primitive variable solution (29). As a consequence, in this case (31) is no longer used.

As explained above, this algorithm is able to construct convergent solutions for arbitrarily small η_0 as long as the viscous tolerance Δ_η is lowered as the resolution is increased. Ideally, one would compute Δ_η using a method to estimate the local truncation error in the cell, perhaps using an approach based in Richardson extrapolation as is done in adaptive mesh refinement schemes [61]; here we adopt a simple empirical approach, tuning Δ_η on a problem-by-problem basis to be as small as possible without compromising the stability of the numerical solution. Tests illustrating the behavior and convergence properties of the scheme in the $\eta_0 \to 0$ limit are shown in Sec. IV C.

Though this section is specialized to primitive variable recovery for a conformal BDNK fluid, it generalizes to nonconformal fluids in a straightforward way—see Appendix B.

D. Reconstruction

As can be seen from (26) and (27), both the relativistic Euler and BDNK equations have fluxes which take roughly the same form. Both include the terms $\boldsymbol{p}_0 = (\epsilon, u^x, u^y)^T$, which must be reconstructed at the cell interfaces from their cell-averaged values $\bar{\boldsymbol{p}}_0$ after primitive variable recovery. Though there are many different reconstruction algorithms (see [47,62] for a review), we use the fifth-order WENO method [63,64]. We provide a review of WENO reconstruction in Appendix C. For the sake of simplicity, for the remainder of this section and in Appendix C we specialize to problems with variation in 1D, as the methods described generalize to higher dimensions by simple repeated application of the 1D algorithms.

The WENO procedure mentioned above may be used to reconstruct all of the variables present in the perfect fluid fluxes, $\boldsymbol{p}_0$. The same cannot be said for the BDNK fluxes, however, as they also include spatial derivative terms proportional to $\partial_i \boldsymbol{p}_0$ such as, e.g., $u^x_{,x}$; prior to reconstructing the values of these terms at the interfaces, one must first compute the needed derivatives. For smooth flows, it suffices to use standard finite-difference stencils to compute the derivative terms. For flows with sharp gradients, however, these finite differences result in the formation of spurious oscillations, which in turn produce unphysical fluid states that destabilize the primitive variable recovery step (39). To mitigate this instability, we instead compute the derivative terms using an adaptive procedure based in the central-WENO (CWENO) method of [58], whereby three different candidate stencils are combined to minimize spurious oscillations near sharp gradients.

To achieve this nonoscillatory property, CWENO produces an interpolation polynomial using a nonlinear weighted sum of essentially nonoscillatory (ENO) polynomials of the cell averages $\bar{p}_i$ as in WENO. Unlike WENO, however, CWENO uses ENO polynomial stencils which are centered about the interface rather than being left or right biased. To apply CWENO to compute derivatives, we take the CWENO interpolation polynomial $p_j(x^i)$ and we differentiate it with respect to x to get $p'_j(x^i)$. We can then evaluate this polynomial at the center of the cell of interest, which yields

$$\bar{p}'_i = \frac{\bar{p}_{i-2} - 4\bar{p}_{i-1} + 3\bar{p}_i}{2h} w_0 + \frac{\bar{p}_{i+1} - \bar{p}_{i-1}}{2h} w_1 + \frac{-3\bar{p}_i + 4\bar{p}_{i+1} - \bar{p}_{i+2}}{2h} w_2, \tag{40}$$

which is a weighted sum of the second-order backward, centered, and forward finite-difference stencils for a first derivative in x, where h is the grid spacing. The nonlinear weights w_k (C3) are defined the same way as in the WENO case with the same smoothness indicators, except the corresponding linear weights [which appear in (C3)] are modified to be

$$d_k = \left(\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\right) \tag{41}$$

and give fourth-order accuracy in the derivative (40).

Both the WENO reconstruction and CWENO derivative computation depend on a free parameter ϵ_W (C3) controlling the amount of sensitivity each step has to sharp gradients in one of the candidate stencils. In principle we can have different values for ϵ_W in these two steps, either to make the reconstruction algorithm more sensitive than the derivative algorithm, or vice versa. We find empirically that independently tuning the two parameters provides little to no advantage in the test cases we consider in Sec. IV, so for the remainder of this work we choose the same value of ϵ_W for both the WENO reconstruction and the CWENO derivative algorithms.

It is important to note that since the CWENO scheme computes the spatial derivative terms using an adaptive finite-difference stencil, the irrotational nature of the gradient of these terms ($\partial_i \boldsymbol{p}_0$, where i is a spatial index) is not exactly preserved [65]. Explicitly, consider the trivial constraint corresponding to $\partial_i \xi$; asserting that the curl of this gradient vanishes (and specializing to the type of problems considered here, in Minkowski spacetime with variation only in t, x, y) leads one to the constraint

$$0 = \partial_x \partial_y \xi - \partial_y \partial_x \xi. \tag{42}$$

It is straightforward to show that discretizations of (42) with fixed stencils, e.g., $\partial_x X \approx (X_{i+1,j} - X_{i-1,j})/(2h)$ and its analogy with $i \to j$ for $\partial_y X$, satisfy (42) exactly. For the CWENO scheme, however, constraints like (42) are only satisfied up to truncation error in the solution, here $O(h^2)$. That said, for large values of the WENO/CWENO parameter ϵ_W the derivatives approach those coming from a fixed stencil, and violations of (42) vanish; for a thorough exploration of curl-type constraint violation for the BDNK scheme, see Sec. IV A.

After the primitive variable recovery step of Sec. III C, we compute the spatial derivative terms $\partial_i \boldsymbol{p}_0$ across the entire grid using (40) and save them. We then treat them in the same way as the nonderivative terms $\boldsymbol{p}_0$, reconstructing their values at the cell interfaces using WENO (C4) and (C6) before feeding them into the numerical flux function.

E. Numerical flux

As was mentioned in Sec. III A, the choice of numerical flux function is critical to the stability of shock wave solutions in a HRSC finite volume scheme. The wide variety of these functions fall roughly into two categories: upwind methods and central methods. Upwind schemes treat the interface between two cells as a Riemann problem, which is solved by feeding information about the characteristics of the PDEs into a Riemann solver. This procedure allows such schemes to bias the required stencils such that they are upwind with respect to the flow, dramatically improving stability. Central schemes, on the other hand, eschew use of detailed characteristic information and Riemann solvers in favor of simple discretizations with stencils centered about cell interfaces. For a detailed assessment and discussion of central schemes in the context of astrophysical applications, see Ref. [66].

Both upwind and central schemes have been successfully applied to the relativistic Euler equations. For the BDNK equations, however, we find that computation of the characteristic information required for an upwind scheme—for example, computation of the linearized flux Jacobian $\frac{\partial f}{\partial q}$ required in a Roe scheme [67]—is difficult and yields a numerical flux which is computationally expensive to evaluate. Hence we opt for a Riemann-solver-free central scheme, specifically one based on the Kurganov-Tadmor numerical flux function [68] [using as an example the flux through the cell interface at $(x_{i+1/2}, y_j)$],

$$\boldsymbol{F}_{i+1/2,j} = \frac{1}{2}(\boldsymbol{f}(\boldsymbol{p}^-_{i+1/2,j}) + \boldsymbol{f}(\boldsymbol{p}^+_{i+1/2,j}) \\ - a[\boldsymbol{q}(\boldsymbol{p}^+_{i+1/2,j}) - \boldsymbol{q}(\boldsymbol{p}^-_{i+1/2,j})]). \tag{43}$$

The Kurganov-Tadmor flux requires only the primitive variables $\boldsymbol{p}_1$ computed at the cell interfaces (via WENO and CWENO, Sec. III D), the flux functions $\boldsymbol{f}$, and a the scalar quantity a, defined to be the maximum wave propagation speed. The value of a controls the amount of numerical diffusion applied at discontinuities and may be found empirically by changing $a \in [0, 1]$ until one strikes an acceptable balance between sharp shock resolution ($a \to 0$) and stability ($a \to 1$). For BDNK theory, though, since we have chosen a frame where the maximum propagation speed is equal to the speed of light, we know a exactly and set[13] $a = 1$. Since precise characteristic information is incorporated into the numerical flux calculation, the method applies aspects of both central and upwind schemes and is sometimes referred to as a central-upwind scheme [60].

It is important to note that the numerical flux [in our case (43)] is constructed such that it is symmetric in $\boldsymbol{p}^-$ and $\boldsymbol{p}^+$; this fact implies that the flux computed at the left side of the interface is equal to that computed on the right side. Physically, this implies that all of the flux of $\boldsymbol{q}$ out of a cell must flow into neighboring cells, and vise versa, such that the total quantity of $\boldsymbol{q}$ cannot change[14] in the absence of sources or boundaries [47]. Integrating (22) over a such a domain $\mathcal{D}$ (assumed to be 2D), one finds

$$\frac{\partial}{\partial t}\int_{\mathcal{D}} \boldsymbol{q}\, dxdy = 0, \tag{44}$$

implying that the total quantity of $\boldsymbol{q} = (T^{tt}, T^{tx}, T^{ty})^T$ in $\mathcal{D}$ is constant in time. The fact that finite volume schemes preserve (44) exactly is known as discrete conservation and is crucial to the success of such schemes in countless applications. In Sec. IV A, we check the conservation of $\boldsymbol{q}$ across the simulation domain and confirm that our scheme possesses the discrete conservation property.

We have constructed our BDNK scheme such that it reduces to a HRSC finite volume perfect fluid solver in the inviscid limit $\eta_0 \to 0$. For the sake of sharp comparisons between the viscous and inviscid cases in the tests that follow, we also use $a = 1$ for the relativistic Euler equations, even though their characteristic speeds are equal to the sound speed $|c_s| = \sqrt{\frac{\partial P}{\partial \epsilon}} = \frac{1}{\sqrt{3}}$. As described above, choosing a larger value of a results in slightly more numerical viscosity in the solution; this numerical viscosity converges away with resolution and is always orders of

[13] The Kurganov-Tadmor flux with $a = 1$ is equivalent to the so-called Harten-Lax-van Leer flux [69] as well as the local Lax-Friedrichs flux [70] when their respective maximum propagation speeds are set to unity.

[14] In a computer simulation, the use of finite precision floating point arithmetic results in round-off errors of order 10^{-16} at double precision; these errors are typically many orders of magnitude smaller than those due to truncation error in the solution, however.

magnitude smaller than the physical viscosities shown in Sec. IV.

F. Evolving J^a

Since we have assumed a fluid with an underlying conformal symmetry, the stress-energy tensor T^{ab} has no dependence on n, and (1) decouples from (2). This implies one has greater freedom in choosing a method to solve (2), since it cannot destabilize the solution to (1). In fact, if one is not interested in the evolution of n, one may forego solving (2) entirely and just solve (1) to evolve ϵ, u^a. For nonconformal fluids, T^{ab} will depend on n, however, and (1) and (2) will have to be solved as a coupled system of PDEs.

In this work, we choose to evolve both T^{ab} and J^a using a scheme based in the finite volume method. This entails applying the same steps described in the past three subsections—primitive variable recovery, reconstruction, and flux computation—to (2). Fortunately, the simple form of the particle current (4) [or equivalently (12) and (17)] simplifies this procedure significantly.

Primitive variable recovery is trivial for J^a, as the one conserved variable J^t (27) is linear in the one primitive variable n [which is the only possible choice of primitive variable, since u^a is being evolved with (1)]. The flux terms are functions only of n and u^a, and hence one may use the same reconstruction method as described above (here WENO, see Appendix C) to interpolate their values to cell interfaces. We use the same numerical flux function for (1) and (2), namely the Kurganov-Tadmor flux (43) with maximum local speed $a = 1$.

G. Time integration

Here, as in [36], we integrate the system of PDEs (23) in time using the total-variation-diminishing second-order Runge-Kutta algorithm known as Heun's method. Heun's method gives the conservative variables at the unknown advanced time level $\boldsymbol{q}^{n+1}$ by writing (23) as $\dot{\boldsymbol{q}} = \boldsymbol{H}(\boldsymbol{q})$ and applying the following procedure:

$$\hat{\boldsymbol{q}}^{n+1} = \boldsymbol{q}^n + \Delta t \boldsymbol{H}(\boldsymbol{q}^n),$$
$$\boldsymbol{q}^{n+1} = \boldsymbol{q}^n + \frac{\Delta t}{2}[\boldsymbol{H}(\boldsymbol{q}^n) + \boldsymbol{H}(\hat{\boldsymbol{q}}^{n+1})]. \tag{45}$$

Heun's method works by producing an estimate using a forward-Euler update step $\hat{\boldsymbol{q}}^{n+1}$, and then uses the known level ($\boldsymbol{q}^n$) and the estimate ($\hat{\boldsymbol{q}}^{n+1}$) to find the conservative variables at the unknown time level ($\boldsymbol{q}^{n+1}$).

H. Boundary conditions

In the tests that follow, we are exclusively interested in the dynamics in the interior of the simulation domain, and the boundaries have no physical interpretation. That said, finite computational resources dictate that boundaries are necessary, and we designate the outermost three grid cells in each direction as boundary cells. For most of the simulations described below, we define the boundary cells to be "ghost" cells, whereby the state in the cell is set to be the same as that in the nearest nonghost cell. Explicitly, at the boundaries at constant y (at constant values of the second index) we take

$$X_{k,j} \coloneqq X_{3,j}, \qquad k \in [0, 2],$$
$$X_{k,j} \coloneqq X_{N-4,j}, \qquad k \in [N-3, N-1], \tag{46}$$

for the hydrodynamic variables $\boldsymbol{p}_0$, and $A \coloneqq B$ is shorthand for "A is set equal to B." The boundaries at constant x (constant first index) are obtained from the above after switching the indices.

The use of ghost cells is common in numerical hydrodynamics, though in this case one must also determine how to handle the derivative terms, both spatial ($\partial_i \boldsymbol{p}_0$) and temporal ($\boldsymbol{p}_1$). It is clear that the choice for $\partial_i \boldsymbol{p}_0$ must be consistent with the choice for $\boldsymbol{p}_0$ in the ghost cells (46), but it is not so obvious how to treat $\boldsymbol{p}_1$. For all of the problems with ghost cells' boundaries considered here, we find no real difference between using (46) for $\boldsymbol{p}_1$ or setting $\boldsymbol{p}_1$ to zero in the ghost cell region; this is largely due to the design of the problems, however, as boundary conditions are not the main focus of this study. We will investigate boundary conditions for the BDNK equations more thoroughly in a future work.

We also consider a test with periodic boundaries, where opposite edges of the domain are identified. This is achieved numerically by identifying the three boundary cells on one side of the domain with the three nonboundary cells nearest to the other edge of the domain, for all four edges. Explicitly, this procedure sets all variables X along the boundary at constant y (second index) via

$$X_{0,j} \coloneqq X_{N-6,j}, \quad X_{1,j} \coloneqq X_{N-5,j}, \quad X_{2,j} \coloneqq X_{N-4,j},$$
$$X_{N-1,j} \coloneqq X_{5,j}, \quad X_{N-2,j} \coloneqq X_{4,j}, \quad X_{N-3,j} \coloneqq X_{3,j}, \tag{47}$$

where the variables along the boundaries at constant x are set in the same way as above except with the indices switched.

IV. NUMERICAL TESTS

In this section we present the results of a series of tests which compare the new BDNK scheme to the HRSC perfect fluid solver obtained in the inviscid limit $\eta_0 \to 0$, as well as to the semi-finite-difference[15] scheme of [36].

[15] In [36], the algorithm splits the stress-energy tensor into a perfect fluid piece and a dissipative correction, each of which has its own flux term. The former is discretized using a finite-volume approach with a Roe flux [67], and the latter with a nonconservative second-order finite-difference stencil. Since the approach of [36] is part finite volume and part finite difference, we refer to it as a "semi-finite-difference" scheme.

TABLE I. Maximum stable Courant factor $\lambda \equiv \frac{a \Delta t}{h}$ (where the local characteristic speed $a = 1$; $\lambda = 0.5$ is the maximum value satisfying the Courant-Friedrichs-Lewy condition for a 2D Cartesian grid [71]) and λ used to make figures for each of the sets of initial data considered here. Whenever the scheme of [36] is used for comparison, we take $\lambda = 0.1$ to improve stability of that scheme. Lower than maximum Courant numbers are used for the 2D viscous rotor test to minimize spurious reflections from the boundary. See Sec. IV B for a discussion of the stability of the 2D oblique shock wave test.

Initial data	Max. λ	λ used
1D Gaussian	0.5	0.1
2D viscous rotor	0.5	0.1
1D shock tube	0.5	0.1
2D oblique shock wave	0.1	0.1
1D steady-state shock wave	0.5	0.1
2D Kelvin-Helmholtz instability	0.5	0.5

The tests are performed on either a 1D or a 2D Cartesian grid, with variation in (t, x) or (t, x, y), respectively. We define a single grid scale h in both spatial directions, and we take the difference between time steps to be $\Delta t = \lambda h / a$, where $a = 1$ is the maximum local characteristic speed and the Courant factors $\lambda \in (0, 1)$ for the tests are reported in Table I.

In all of the simulations below, initial data are set by prescribing values for the hydrodynamic variables ϵ, n and u^a; viscous corrections are initialized to zero, so T_1^{tc} is set at $t = 0$ using T_0^{tc}.

All dimensionful quantities are given in code units,[16] which are the same as in [36]. It is important to note that the following simulations are tests designed to evaluate the performance of the algorithm, not attempts to model a known physical system. The scales chosen in these tests are arbitrary, and we choose the amount of viscosity used based on whether the dynamics are under- or overdamped, rather than by reference to a substance where the viscosity is known. In particular, none of the initial data we consider are particularly close to that relevant in modeling heavy-ion collisions, and hence it is not that meaningful to quantify viscosities via the entropy-normalized shear viscosity η / s as is typically done in the nuclear physics literature. Instead, we use the parameter η_0 defined in (20).

We order the set of tests into three categories: (1) tests of constraint preservation, (2) tests with sharp gradients, and (3) tests of the BDNK solutions approaching the inviscid limit. In each section we include results from both 1D and 2D simulations.

A. Tests of constraint preservation

1. 1D Gaussian test

We will first check the ability of the new scheme to preserve the spatial integral of the conservation law over the simulation domain in the absence of sources or significant boundary interactions (44). To do so, we first consider the simplest possible test, namely a 1D simulation starting from smooth initial data in x, as in [36]. Explicitly, at the initial time we take a stationary Gaussian profile in the energy density

$$\epsilon(t = 0, x) = A e^{-x^2 / w^2} + \delta, \qquad u^x(t = 0, x) = 0, \tag{48}$$

with parameter values $A = 1$, $w = 25$, and $\delta = 10^{-1}$, and we take the simulation domain to be $x \in [-L, L]$, where $L = 200$. For the viscosity, we choose $\eta_0 = 0.2$. Since it is smooth, the initial data (48) give results which are very similar to those given in [36]. The key difference, however, is that since the new scheme is conservative, the integrals of motion (44) are conserved to machine precision at times when no fluid is leaving the boundaries of the domain (44); the semi-finite-difference scheme of [36] conserves them only to truncation error, which is roughly 12 orders of magnitude larger—see Fig. 1.

2. 2D viscous rotor

In tests with two or more spatial dimensions, one must be careful to preserve "curl"-type constraints, of which (42) is an example. These constraints are satisfied exactly when

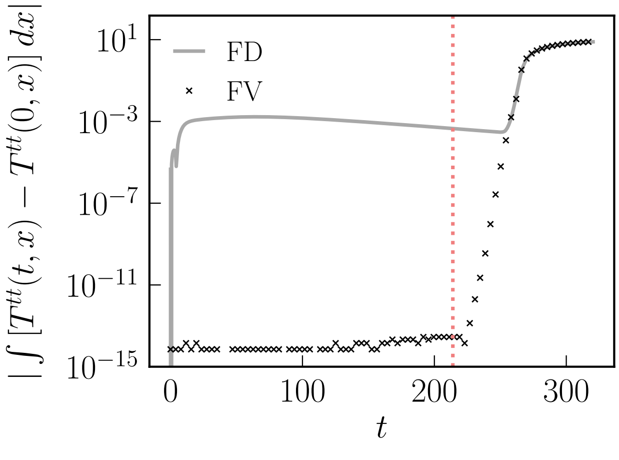

FIG. 1. Discrete conservation of T^{tt} across the spatial domain for a simulation starting from Gaussian initial data (48) with $\eta_0 = 0.2$ for the finite volume scheme presented here (FV) as well as the semi-finite-difference scheme of [36] (FD). As expected, the finite volume scheme conserves T^{tt} up to machine precision $\sim 10^{-15}$, until the fluid pulse reaches the boundary at the time marked by the light red dotted line. The semi-finite-difference scheme of [36] conserves T^{tt} only up to the level of truncation error, which in this case is $\sim 10^{-3}$.

[16] We use natural units with energies measured in GeV, which implies velocities are dimensionless $[u^a] = 1$, coordinates have units of inverse energies $[x^a] = \mathrm{GeV}^{-1}$, and thus energy densities have unit $[\epsilon] = [T^{ab}] = \mathrm{GeV}^4$.

derivative terms are approximated using fixed stencils; such stencils are unstable about sharp gradients, however, so we opt for an adaptive scheme based upon the CWENO algorithm (see Sec. III D).

To evaluate the effectiveness of the new scheme at preserving constraints like (42), we study a set of initial data which correspond to a "viscous rotor," namely a fluid at constant pressure where a cylinder in the center of the domain is initially rotating at constant angular velocity ω. We implement the viscous rotor initial data on a grid with $x, y \in [-L, L]$ with $L = 3$, where at $t = 0$

$$\begin{aligned} \epsilon(x,y) &= 1, \\ v^x(x,y) &= -\omega\sqrt{x^2+y^2}\sin(\theta)D(d,\delta), \\ v^y(x,y) &= \omega\sqrt{x^2+y^2}\cos(\theta)D(d,\delta), \\ n(x,y) &= \frac{1}{2}(D(d,\delta)+1)+B(x,y), \end{aligned} \tag{49}$$

which gives a fluid at constant pressure $P = \frac{1}{3}$, with a circular region in the center initially rigidly rotating with angular velocity $\omega = 1$. This is implemented via functions

$$\begin{aligned} \theta &\equiv \mathrm{atan2}(y,x), \\ D(d,\delta) &\equiv \frac{1}{2}\left[1+\tanh\left(\frac{d}{\delta}\right)\right], \\ d &\equiv R-\sqrt{x^2+y^2}, \\ B(x,y) &= \begin{cases} 0.1 & \sqrt{x^2+y^2} \le R \ \& \ |y| < 0.1 \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \tag{50}$$

where atan2(y,x) is the two-argument arctangent, $D(d,\delta)$ is a function which is unity at the origin and decreases sharply but smoothly at radius $R = 0.5$, with the smoothness of the transition controlled by δ, which we take to be 0.05.$B(x,y)$ gives a raised bar oriented horizontally in the center of the rotating region which may be used to see how far the rotor has spun.

Figure 2 shows the evolution of this set of initial data as a function of time (columns) and viscosity (rows). From the figure, it is immediately clear that the viscosity has a significant effect on the late-time behavior of the solution—the inviscid case continues rigidly rotating for a while, leaving the bar of overdensity approximately straight up until the solution becomes Kelvin-Helmholtz unstable and forms vortices[17]; the intermediate viscosity case

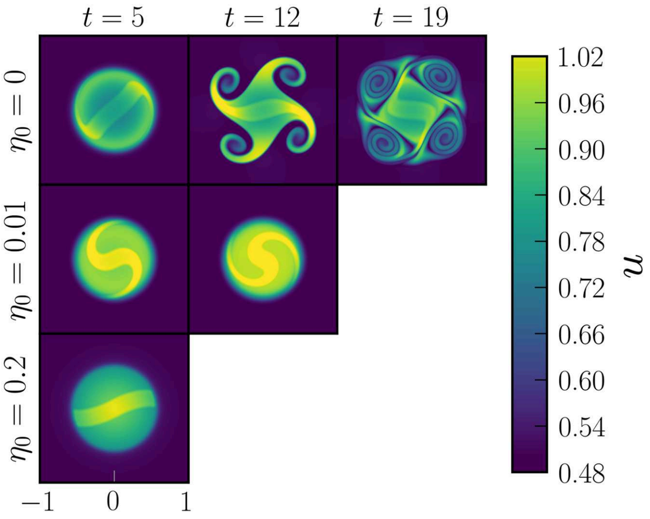

FIG. 2. Density (n) evolution of viscous rotor initial data (49) as a function of time (columns) for three different viscosities: $\eta_0 = 0$, 0.01, 0.2 in rows, from top to bottom. In the inviscid simulation, the cylinder of fluid is Kelvin-Helmholtz unstable and forms vortices which are not present in the viscous cases. At intermediate viscosity (middle row), the fluid experiences a shearing force which distorts the bar of overdensity present in the initial data, before the cylinder stops rotating entirely around $t \sim 12$. At the highest viscosity shown (bottom), the cylinder rotates only about 20° before stopping at $t \sim 5$.

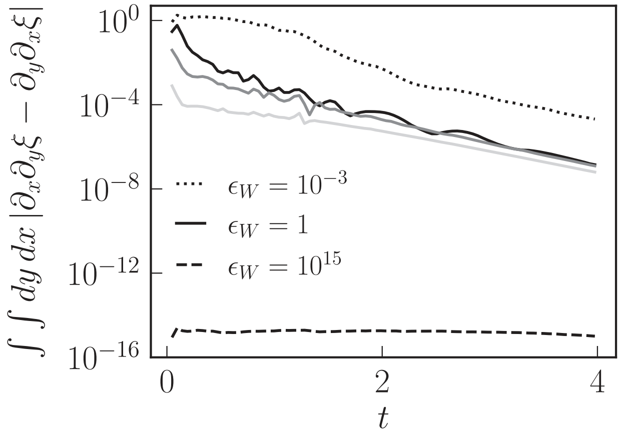

FIG. 3. Integral of the absolute value of the constraint (42) over the domain for viscous rotor initial data with $\eta_0 = 0.2$. The value of ϵ_W determines how strongly the smoothness of the candidate ENO stencils impacts the nonlinear weights; small values of ϵ_W imply strong sensitivity to nonsmoothness, and large values imply insensitivity (and as a result give a fixed fourth-order derivative stencil). Hence, for smaller ϵ_W one finds larger violations of the constraint (42), which converge away with resolution (the solid lines range over $N_x = 2^7, 2^8, 2^9$, with lighter shades representing higher resolutions). In the $\epsilon_W \to \infty$ limit, constraint violation approaches machine precision (cf. the $\epsilon_W = 10^{15}$ case).

[17] Circular symmetry is broken by the square grid, and the grid-scale bumps at the top, bottom, leftmost, and rightmost points on the circle each source the Kelvin-Helmholtz instability. Convergence is typically lost after these vortices form, as both the size of the perturbation and the numerical viscosity in the solution decrease as the grid is refined.

experiences a strong shearing force, distorting the bar into an "S" shape before stopping at $t \sim 12$; and the high-viscosity case stops almost immediately after $t \sim 5$.

Figure 3 shows violations of the constraint (42) for the case with $\eta_0 = 0.2$, where the viscous contribution to the fluxes is significant. Plotted in the figure are a set of lines with varying values of the WENO/CWENO parameter ϵ_W, which determines the amount of "stencil switching" that occurs during a simulation. As is described in Sec. III D and Appendix C, at low values of ϵ_W the CWENO algorithm adjusts the nonlinear weights to be significantly different from the linear ones, producing a nonuniform stencil and consequently violating the constraint (42); these violations converge away with numerical resolution, however (shown in the solid lines of varying shade for $\epsilon_W = 1$). In the limit $\epsilon_W \to \infty$ (approximated in the figure with $\epsilon_W = 10^{15}$), the CWENO algorithm gives a fixed, fourth-order centered finite difference stencil, and the violation of (42) drops to near machine precision.

B. Tests with sharp gradients

1. 1D shock tube

Though it remains unclear whether discontinuous solutions are sensible in BDNK theory,[18] one is still free to pose discontinuous initial data; such states may be interpreted as smooth solutions that are unresolved at the current grid resolution. It is essential that our algorithm be able to capture solutions with unresolved shock waves, as such features are prevalent in astrophysics applications.

To evaluate the performance of our algorithm for solutions with sharp gradients, we first consider the standard 1D shock tube test on a domain with $x \in [-L, L]$ with $L = 200$, where

$$\epsilon(t=0,x) = \begin{cases} 1 & x \le 0 \\ 0.1 & x > 0, \end{cases} \qquad u^x = 0 \tag{51}$$

and $\eta_0 = 0.2$, again following [36] except with a larger difference between the left and right states. This set of initial data highlights the advantages of a nonoscillatory, conservative discretization over the semi-finite-difference discretization of [36] in that the former gives a stable, convergent evolution, and the latter is plagued by spurious oscillations which do not quickly converge away (see Fig. 4).

2. 2D oblique shock wave

Outside of one spatial dimension, it is now possible for the fluid to possess sharp gradients which are not aligned with the numerical grid. To test this scenario we adopt the 2D oblique shock wave initial data of [72], whereby the simulation domain is divided into four regions,

$$(n, P, v^x, v^y) = \begin{cases} (0.5, 1, 0, 0) & x < 0, y < 0 \\ (0.1, 1, 0, 0.97) & x > 0, y < 0 \\ (0.1, 1, 0.97, 0) & x < 0, y > 0 \\ (0.1, 0.01, 0, 0) & x > 0, y > 0 \end{cases}. \tag{52}$$

Since the oblique shock wave forms dynamically during the simulation, we find it unnecessary to use discontinuous initial data, which may be ill-posed for BDNK theory. Hence we follow the pattern of tests described earlier and adopt a smoothed version of this set of initial data with tunable sharpness parameters. We use a grid with $x, y \in [-L, L]$ with $L = 200$ and define the initial data by

$$\begin{aligned} n &= 0.4D(d_n, \delta) + 0.1, \\ \epsilon &= 3 - 2.97D(d_\epsilon, \delta), \\ v^x &= 0.97D(d_{v^x}, \delta), \\ v^y &= 0.97D(d_{v^y}, \delta), \end{aligned} \tag{53}$$

with

$$\begin{aligned} d_n &= L - [(x+L)^\gamma + (y+L)^\gamma]^{\frac{1}{\gamma}}, \\ d_\epsilon &= L - [(x-L)^\gamma + (y-L)^\gamma]^{\frac{1}{\gamma}}, \\ d_{v^x} &= L - [(x+L)^\gamma + (y-L)^\gamma]^{\frac{1}{\gamma}}, \\ d_{v^y} &= L - [(x-L)^\gamma + (y+L)^\gamma]^{\frac{1}{\gamma}}, \end{aligned} \tag{54}$$

where δ controls the smoothness of the transitions and γ controls the squareness of each quadrant; we set $\gamma = \delta = 10$ here. This set of initial data is designed such that shortly after $t = 0$, high-velocity flows from the upper left and lower right quadrants meet the high-pressure flow from the lower left quadrant; the result is a high-pressure, high-velocity flow, bounded by an almond-shaped shock wave, which propagates through the low-pressure upper right quadrant.

The non-grid-aligned shock wave can be a significant source of spurious numerical oscillations; fortunately, these can be managed by careful choice of the WENO/CWENO parameter ϵ_W. For the case shown in Fig. 5, which has viscosity $\eta_0 = 0.2$, we use $\epsilon_W = 1$; significantly larger values of ϵ_W do not choose stencils which avoid the shock and are prone to oscillations, and significantly smaller values of ϵ_W switch stencils too frequently, introducing oscillations into the derivative terms found in the viscous

[18] It is well known that the weak formulation of the relativistic Euler equations possesses discontinuous solutions which are typically used to model shock waves. In BDNK theory, there is evidence that one should expect shock wave solutions to be smooth for good hydrodynamic frames [36,37], potentially eliminating the physical need for discontinuous solutions. Furthermore, the presence of derivative terms in the BDNK stress-energy tensor complicates the formulation of the Riemann problem, which has yet to be solved for BDNK theory.

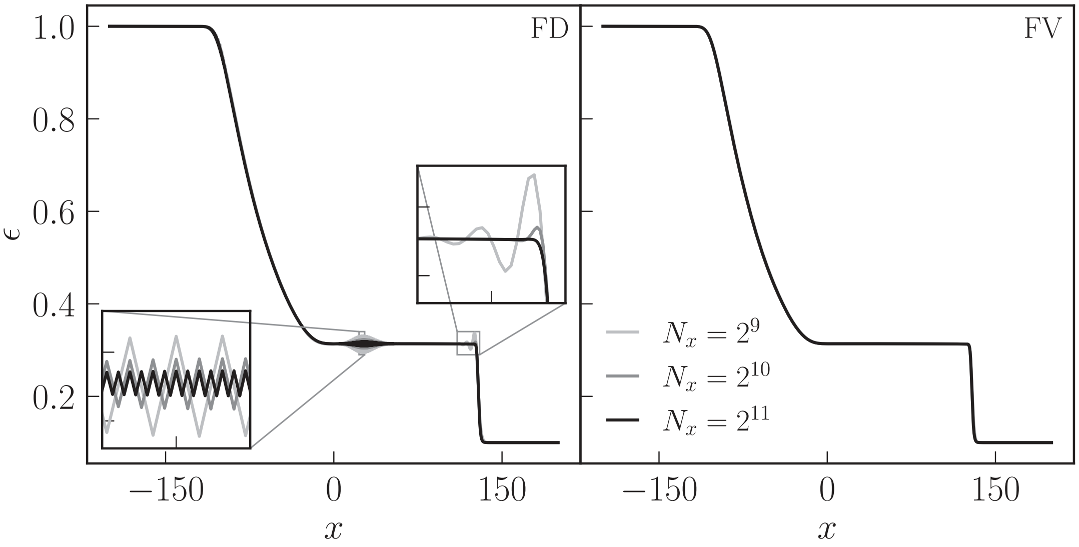

FIG. 4. Comparison of solutions for ϵ starting from shock tube initial data (51) at three successive resolutions for the semi-finite-difference scheme of [36] (left, FD) versus the finite volume scheme presented here (right, FV) at $t \sim 43$ for $\eta_0 = 0.2$. The FD scheme has oscillations near the shock front which quickly converge away with resolution, as well as grid-scale "sawtooth" oscillations that developed early on near the origin (the $t = 0$ location of the shock front) and do not converge away as rapidly with resolution. These features do not appear in the figures of [36] because the discontinuities there are smaller in amplitude, leading to oscillations small enough to be tamed by applying Kreiss-Oliger dissipation; said dissipation is not strong enough to remove the oscillations for the initial data (51), and we choose not to apply artificial dissipation in either scheme throughout this work. The FV solutions are free of noticeable oscillations, and the $N_x = 2^9, 2^{10}, 2^{11}$ curves all overlap at the resolution of the plot.

fluxes. That said, for the case shown in Fig. 5 the solution is largely free of oscillations.

This set of initial data (53) is posed as a challenging code test and as such it clearly illustrates the limitations of our current algorithm, which crashes for $v^x, v^y \gtrsim 0.98$. In these cases, the solution is stable until a while after the shock wave forms; the instability sets in near the "base of the almond," around the origin, where numerical error leads (39) to produce a complex result, crashing the code. Stability can be restored for higher initial velocities $v^x, v^y \sim 0.98$ by significantly reducing the Courant factor to $\lambda = 0.05$ or even 0.01; unfortunately, these values would likely be prohibitively expensive at higher resolutions or in 3D simulations. That said, the fact that the solutions are stabilized by reducing λ implies that the dominant error is coming from the time integration algorithm, and these simulations may be rendered stable by use of a higher-order time integration scheme in place of the second-order one used here.

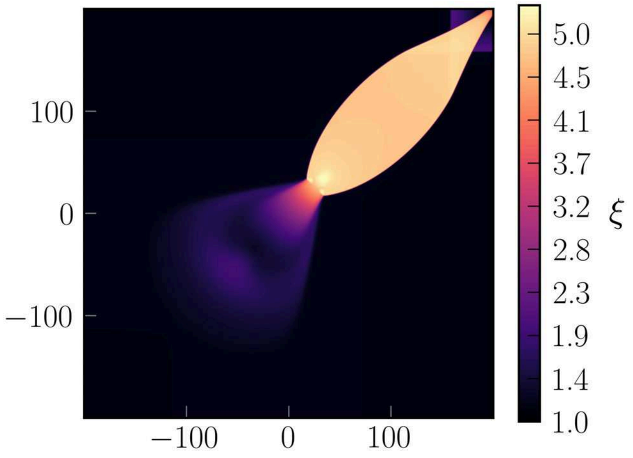

FIG. 5. Solution for the log of the energy density ξ for the 2D oblique shock wave initial data (53) at $t \sim 220$ for $\eta_0 = 0.2$. Note that the solution is nonoscillatory, even though there is an order unity jump in ξ (corresponding to a jump of ~50 in ϵ) which is not aligned with the numerical grid.

C. Tests of the inviscid limit

1. 1D steady-state shock wave

For a clear illustration of the behavior of our algorithm in the inviscid limit—which is designed to use the BDNK primitive variable solution only in regions where the physical viscosity is resolved—we now consider a case which has a clearly defined equilibrium region (where gradients are negligible and the perfect fluid approximation is valid) as well as a nonequilibrium region (where viscous corrections are appreciable). Specifically, we consider the case of a planar shock wave in its rest frame, as discussed in [36]. This solution is characterized by a central, smooth shock wave bridging the transition between two equilibrium states at $x \to \pm\infty$.

To model this shock wave, we choose a set of initial data which asymptotically (as $x \to \pm\infty$) satisfies the Rankine-Hugoniot conditions[19] for an ideal fluid shock wave in its rest frame,

$$\epsilon_R = \epsilon_L \frac{9v_L^2 - 1}{3(1 - v_L^2)},$$
$$v_R = \frac{1}{3v_L}, \tag{55}$$

where the equilibrium state to the left of the shock wave has parameters $\epsilon = \epsilon_L$, $v^x = v_L$, and the state on the right is defined by $\epsilon = \epsilon_R$, $v^x = v_R$. One should expect (55) to describe the analogous BDNK shock wave solution as well, provided one is sufficiently far outside of the shock.

Inside the shock, the viscous terms in the BDNK equations should produce a smooth profile transitioning between the two states. This precise profile is not known analytically, but we find that choosing a set of initial data which is sufficiently close to this profile leads to a solution which quickly settles down to the desired steady-state shock wave solution. For this initial data we choose a setup with left and right states given by (55) and the following smooth transition between the two states at $x = 0$ [here given by the Gaussian error function, $\mathrm{erf}(x)$],

$$\epsilon(t = 0, x) = \frac{\epsilon_R - \epsilon_L}{2}\left[\mathrm{erf}\left(\frac{x}{w}\right) + 1\right] + \epsilon_L,$$
$$v^x(t = 0, x) = \frac{v_R - v_L}{2}\left[\mathrm{erf}\left(\frac{x}{w}\right) + 1\right] + v_L, \tag{56}$$

where $w = 10$. We choose the left state to be given by $\epsilon_L = 1, v_L = 0.8$, and the right state is then computed using (55). The evolution quickly reaches the steady-state solution after a small blob of fluid propagates out of the domain, changing the shock profile from the erf function to one that satisfies the BDNK equations in the static limit (see [36] Appendix C).

The steady-state shock profile for $\eta_0 = 0.2$ is shown in the top panel of Fig. 6 as a dashed black line. At this viscosity and resolution, the BDNK primitive variable solution (39) is stable across the entire grid; we compare the results of the adaptive algorithm (Sec. III C) for various tolerances Δ_η against this solution. In the top panel, the region designated as "nonequilibrium" is highlighted in gray, where the shade is determined by the viscous tolerance Δ_η shown in the legend. For large Δ_η (darkest gray), the algorithm only sees regions with very steep gradients as nonequilibrium, using the perfect fluid primitive variable solution (29) over most of the grid. This results in sizable errors (bottom panel) when compared to the true solution, where (39) is used everywhere. Using small Δ_η results in more of the shock wave being designated as nonequilibrium, and the error is significantly reduced.

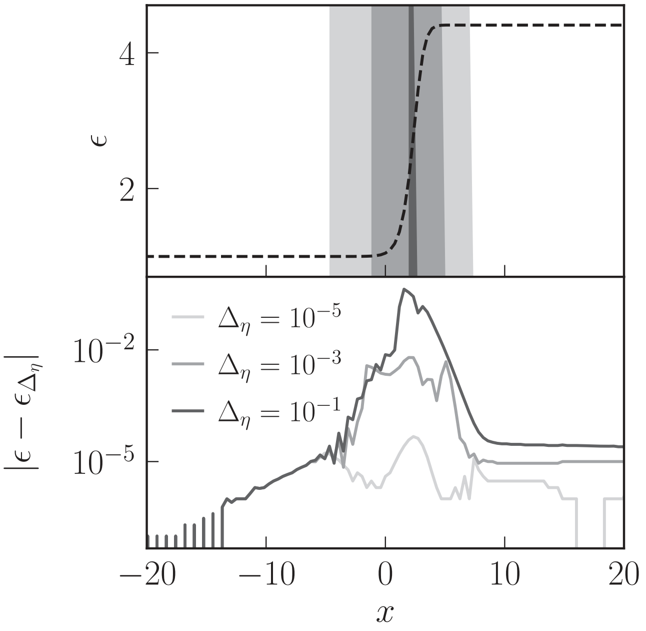

FIG. 6. Illustration of the algorithm used to capture the perfect fluid limit for steady-state shock wave initial data (56) at $\eta_0 = 0.2$. The dashed line (top) is the solution for this set of initial data at late times, constructed using the BDNK primitive variable solution (39) everywhere. When the adaptive primitive variable solver is used, (39) is only used in the gray region, where the shade of gray corresponds to the value of the viscous tolerance Δ_η shown in the legend. For large values of this tolerance, only regions with very steep gradients are identified as being nonequilibrium, and (39) is only used in a small sliver of the solution (and the perfect fluid primitive variable solution (29) is used elsewhere). This induces significant errors (bottom) when compared to the solution where only (39) is used. Shrinking the viscous tolerance Δ_η results in more of the nonequilibrium region being identified as such by the algorithm and gives successively smaller errors when compared to the dashed (BDNK-only) solution. For $\Delta_\eta \lesssim 10^{-7}$, the error drops to machine precision.

The behavior shown in Fig. 6 illustrates that the adaptive primitive variable algorithm is correctly identifying the equilibrium and nonequilibrium regions and demonstrates the effect of the tolerance Δ_η on the solution. That said, for the case shown one is best served by simply using the BDNK solution (39) everywhere, since it is stable; the next section shows an example where it is unstable, and one must use the adaptive algorithm to produce a solution at the given viscosity and numerical resolution.

2. 2D Kelvin-Helmholtz instability

In this section we consider a scenario in which one may be interested in physical viscosities which are unresolved at

[19] These conditions may be straightforwardly derived from the relativistic Euler equations by assuming a solution independent of time; see [36].

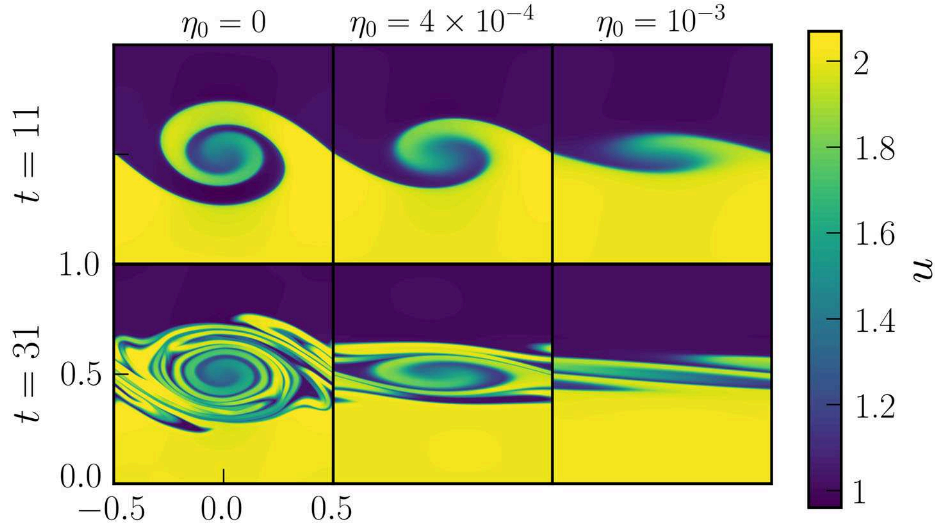

FIG. 7. Evolution of Kelvin-Helmholtz-unstable initial data (57) for the density n at three different viscosities in columns, from left to right: $\eta_0 = 0, 4 \times 10^{-4}, 10^{-3}$, at $t = 11$ (top row) and $t = 31$ (bottom row). Viscosity has a clear effect on both the early- and late-time state of the fluid; at $t = 11$ it determines the amount of growth of the perturbation of low-density fluid (dark blue) into the high-density (yellow) region. For the two lower-viscosity cases (left two columns), long-lived vortices form out of these perturbations. At high viscosity, no clear vortex has formed, instead the perturbation has been sheared into a long, thin mixed layer.

typical numerical resolutions, wherein the standard BDNK primitive variable solution (39) is unstable and one requires an alternative method. Specifically, we consider the set of initial data from [73],

$$
\begin{aligned}
\epsilon &= 1,\\
n &= 1 + \frac{1}{2}\left[\tanh\left(\frac{y-y_1}{a}\right) - \tanh\left(\frac{y-y_2}{a}\right)\right],\\
v^x &= u_{flow}\left[\tanh\left(\frac{y-y_1}{a}\right) - \tanh\left(\frac{y-y_2}{a}\right) - 1\right],\\
v^y &= A\sin(2\pi x)\left[\exp\left(-\left[\frac{y-y_1}{\sigma}\right]^2\right)\right.\\
&\quad \left. + \exp\left(-\left[\frac{y-y_2}{\sigma}\right]^2\right)\right],
\end{aligned}
\tag{57}
$$

where the domain is $x \in [-L, L], y \in [-2L, 2L]$, and $u_{\mathrm{flow}} = \frac{1}{4}c_s = \frac{1}{4\sqrt{3}}, A = 0.01, a = 0.05, \sigma = 0.2, y_1 = -0.5, y_2 = 0.5$. Since the domain is twice as large in the y direction, we double the numerical resolution in that direction, $N_y = 2N_x$. This set of initial data corresponds to a jet of high density passing through a region of lower ambient density, forming two interfaces. These interfaces are seeded with a small perturbation of low density into the jet region, which grows as a result of the Kelvin-Helmholtz instability, eventually forming vortices if the viscosity of the fluid is sufficiently small.

Figure 7 shows snapshots from the evolution of this set of initial data (57) for three resolutions, from left to right: $\eta_0 = 0, 4 \times 10^{-4}, 10^{-3}$, at two times (shown in columns). Since the initial data (57) has a reflect-and-shift symmetry [73] between the regions $y > 0$ and $y < 0$, only $y \geq 0$ is shown in the figure. The effect of viscosity is readily apparent at early times (top row), as the size to which the initial perturbation grows (roughly, the number of winds in the spiral) diminishes with increasing viscosity. At late times the behavior is markedly different between the left- and rightmost columns: a vortex persists for $\eta_0 = 0$, and for $\eta_0 = 10^{-3}$ the feature from the top panel gets sheared into a wide, thin layer. To investigate the transition between these two disparate behaviors, one must consider an intermediate viscosity, like that shown in the middle column of Fig. 7. There, the BDNK primitive variable solution is unstable for $N_x \lesssim 2^9$, so this case serves as a suitable test for the adaptive primitive variable solver of Sec. III C.

Figure 8 shows a set of screenshots at $t = 11$ of the $\eta_0 = 4 \times 10^{-4}$ simulations as a function of N_x. At the lower two resolutions, the BDNK primitive variable solver fails, and the solution can be stabilized using the adaptive algorithm with $\Delta_\eta = 10^{-3}, 10^{-4}$, respectively. In these cases, the perfect fluid primitive variable solution (29) is used over essentially the entire grid. Despite this fact, the solutions produced by increasing resolution and shrinking the viscous tolerance (Δ_η) still converge to the true BDNK viscous solution. We stress that most of the visible effect of viscosity is provided by the viscous fluxes, which are numerically well behaved in the inviscid limit; as a result, the top two (lower-resolution) panels of Fig. 8, despite using the perfect fluid primitive variable solution, still resemble (and converge to) the $\eta_0 = 4 \times 10^{-4}$ panel of Fig. 7 rather than the $\eta_0 = 0$ panel.

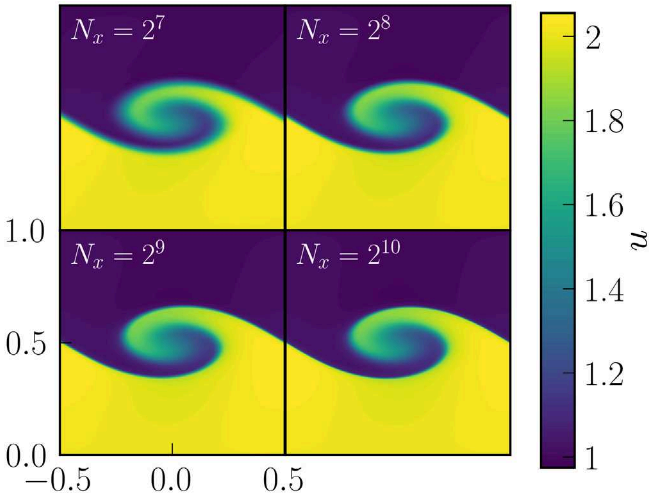

FIG. 8. Snapshot of the density n at $t = 11$ from the Kelvin-Helmholtz simulation for $\eta_0 = 4 \times 10^{-4}$, as a function of resolution. For the two lower-resolution panels (top row), the physical viscosity is small enough that the BDNK primitive variable solution is numerically unstable, and we use the adaptive primitive variable algorithm with tolerances $\Delta_\eta = 10^{-3}, 10^{-4}$, respectively. At these tolerances, the perfect fluid primitive variable solution is used across the entire grid for most of the simulation after the first few time steps; despite this, the solution still converges to the correct viscous solution and is noticeably different from the inviscid solution (top left of Fig. 7) because most of the dissipation comes from the viscous fluxes rather than the primitive variable solution.

V. CONCLUSION

In this work we have presented the first multidimensional finite volume scheme designed to solve the causal, stable relativistic Navier-Stokes equations of BDNK theory [26]. Our algorithm naturally extends traditional HRSC central schemes [66,68], but features distinct modifications to account for the different structure of the BDNK equations. In BDNK, the variables which must be reconstructed also include spatial derivatives of the hydrodynamic variables; to compute these in a nonoscillatory fashion, we use adaptive derivative stencils based in the CWENO method [58]. The most involved difference between a finite volume perfect fluid solver and our BDNK scheme comes in solving for the primitive variables (see, e.g., [74] for a discussion). It turns out that the BDNK primitive variable solution may generically be carried out analytically, though problems arise when viscous contributions are small enough to become unresolved at a given grid resolution. In these cases, we apply an adaptive algorithm which treats cells with unresolved viscosity as “effectively inviscid,” applying the perfect fluid primitive variable inversion. As resolution is increased, the viscous terms eventually become resolved in these cells, and the BDNK primitive variable solution is used. As a result, the adaptive approach produces solutions which converge to solutions of the continuum PDEs.

To evaluate the algorithm’s performance, we focus on the conformal fluid limit and apply it to several flat-spacetime test problems with variation in one and two spatial dimensions. We begin with a simple test of smooth initial data in 1D and confirm that the new scheme conserves the integrals of motion up to machine precision. A 1D shock tube test illustrates the improved stability of the new scheme over that of [36], and a 2D oblique shock wave test suggests a higher-order time integrator may be useful to stably evolve very high-velocity flows with sharp gradients. Steady-state shock wave solutions in 1D are used to illustrate the spatial dependence of the adaptive primitive variable scheme, and 2D simulations demonstrating the Kelvin-Helmholtz instability provide a case where the adaptive algorithm is necessary to produce convergent low-viscosity BDNK solutions at finite resolution.

While we have specialized to conformal fluids in this work, future extensions will be equipped with more general microphysics, allowing for the study of effects such as bulk viscosity. A generalization of the approach presented here would also make possible a detailed comparison of BDNK and MIS-type theories constructed in general hydrodynamic frames [29], which could help elucidate the connections between the two theories. There are also a number of open numerical problems to be investigated—one example would be to perform a comparison of fully flux-conservative solvers for BDNK and MIS [44]; another would be to consider a full first-order reduction of the BDNK equations, wherein the spatial derivative terms are evolved using their own set of evolution equations.

The BDNK algorithm presented here should be sufficiently stable and accurate to be applied to a variety of relativistic hydrodynamics problems where first-order dissipation might be relevant. Among those would be the investigation of viscous effects in the inspiral [75] and merger [4] of binary neutron star systems. The guaranteed causality of the BDNK equations would also motivate the application of this numerical scheme to simulations of heavy-ion collisions, where current MIS approaches show acausal behavior [34]. While viscous effects might also be important in black hole accretion problems [76], the presence of magnetic fields introduces anisotropies in the dissipative sector presently unaccounted for in BDNK theory [38,44]. Although first-order formulations of dissipative magnetohydrodynamics have been proposed [77], their extension to general hydrodynamic frames has just started to be investigated [78].

ACKNOWLEDGMENTS

The authors thank Fabio Bemfica, Marcelo Disconzi, and Jorge Noronha for insightful discussions related to this work. This material is based upon work supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. DGE-1656466.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. F. P. acknowledges support from NSF Grant No. PHY-1912171, the Simons Foundation, and the Canadian Institute For Advanced Research (CIFAR). E. R. M. acknowledges support from postdoctoral fellowships at the Princeton Center for Theoretical Science, the Princeton Gravity Initiative, and the Institute for Advanced Study.

APPENDIX A: COORDINATE COMPONENTS OF THE CONSERVED CURRENTS

Here we present the Cartesian components of the conserved currents J_1^a, T_1^{ab} for a conformal fluid in 4D Minkowski spacetime. Since we only consider systems with translation invariance in the z direction, only the t, x, y components will be necessary. For 1D test problems (namely the 1D Gaussian, shock tube, and steady-state shock wave in Sec. IV) we take all fields to only be functions of t, x, and as a result $T_1^{cy} = u^y = 0$; these equations may also be found in [36].

Beginning with the particle current (which is the same for the perfect fluid and BDNK theory), the components of J_1^a are obtained immediately using (12) and the four-velocity

$$u^a = \left(\sqrt{1 + (u^x)^2 + (u^y)^2}, u^x, u^y, 0\right)^T. \tag{A1}$$

Combining the four-velocity with the perfect fluid stress-energy tensor (3) may be used compute T_0^{ab} as well.

The BDNK stress-energy tensor may be specified by defining the components in (11). The correction to the energy density is

$$\begin{aligned}\mathcal{A} = \frac{1}{4}\chi_0 e^{\frac{3}{4}\xi}\bigg[&u^x\left(\frac{4\dot{u}^x}{u^t} + 3\xi_{,x}\right) + u^y\left(\frac{4\dot{u}^y}{u^t} + 3\xi_{,y}\right)\\ &+ 3\dot{\xi}u^t + 4u^x_{,x} + 4u^y_{,y}\bigg].\end{aligned} \tag{A2}$$

The two independent components of the heat flux vector are

$$\begin{aligned}\mathcal{Q}^x = \frac{1}{4}\lambda_0 e^{\frac{3}{4}\xi}[&4\dot{u}^x u^t + u^x\dot{\xi}u^t + ((u^x)^2 + 1)\xi_{,x}\\ &+ 4u^x u^x_{,x} + u^y(u^x\xi_{,y} + 4u^x_{,y})]\end{aligned} \tag{A3}$$

and

$$\begin{aligned}\mathcal{Q}^y = \frac{1}{4}\lambda_0 e^{\frac{3}{4}\xi}[&u^y(u^t\dot{\xi} + u^x\xi_{,x} + u^y\xi_{,y} + 4u^y_{,y})\\ &+ 4u^t\dot{u}^y + 4u^x u^y_{,x} + \xi_{,y}],\end{aligned} \tag{A4}$$

which are related to the t component by the constraint $u_a\mathcal{Q}^a = 0$, which implies

$$\mathcal{Q}^t = \frac{1}{u^t}(u^x\mathcal{Q}^x + u^y\mathcal{Q}^y), \tag{A5}$$

and $\mathcal{Q}^z = 0$ due to the spatial symmetry we have assumed. The independent components of the shear term are the xx component

$$\begin{aligned}-2\eta\sigma^{xx} = \frac{2\eta_0 e^{\frac{3}{4}\xi}}{3u^t}[&u^t(-2((u^x)^2 + 1)u^x_{,x} + (u^x)^2 u^y_{,y}\\ &- 3u^x u^x_{,y}u^y + u^y_{,y}) - 2(u^x)^3\dot{u}^x + (u^x)^2 u^y\dot{u}^y\\ &- u^x\dot{u}^x(3(u^y)^2 + 2) + u^y\dot{u}^y],\end{aligned} \tag{A6}$$

the xy component

$$\begin{aligned}-2\eta\sigma^{xy} = -\frac{\eta_0 e^{\frac{3}{4}\xi}}{3u^t}[&u^t(3(u^x)^2 u^y_{,x} + u^x u^y(u^x_{,x} + u^y_{,y})\\ &+ 3u^x_{,y}((u^y)^2 + 1) + 3u^y_{,x})\\ &+ \dot{u}^x u^y((u^x)^2 + 3(u^y)^2 + 3)\\ &+ u^x\dot{u}^y(3(u^x)^2 + (u^y)^2 + 3)],\end{aligned} \tag{A7}$$

and the yy component

$$\begin{aligned}-2\eta\sigma^{yy} = \frac{2\eta_0 e^{\frac{3}{4}\xi}}{3u^t}[&-3u^t u^x u^y u^y_{,x} + u^t u^x_{,x}((u^y)^2 + 1),\\ &- 2u^t((u^y)^2 + 1)u^y_{,y} - 3(u^x)^2 u^y\dot{u}^y\\ &+ u^x\dot{u}^x((u^y)^2 + 1),\\ &- 2((u^y)^3 + u^y)\dot{u}^y].\end{aligned} \tag{A8}$$

The remaining required components may be found from the constraint $u_a\sigma^{ab} = 0$, which implies

$$\sigma^{tc} = \frac{1}{u^t}(u^x\sigma^{xc} + u^y\sigma^{yc}). \tag{A9}$$

APPENDIX B: PRIMITIVE VARIABLE RECOVERY FOR A NONCONFORMAL BDNK FLUID

For a nonconformal BDNK fluid, (34) generalizes to

$$\boldsymbol{q}_1 = \boldsymbol{q}_0(\boldsymbol{p}_0) + \boldsymbol{A}(\boldsymbol{p}_0)\cdot\boldsymbol{C}\cdot\boldsymbol{p}_1 + \boldsymbol{D}\cdot\boldsymbol{b}(\boldsymbol{p}_0, \partial_i\boldsymbol{p}_0) + \boldsymbol{\tau}, \tag{B1}$$

where the matrices $\boldsymbol{C}, \boldsymbol{D}$ are populated solely with transport coefficients and vanish in the inviscid limit. In the conformal limit, $\boldsymbol{C}, \boldsymbol{D} \to \eta_0\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix, recovering (34). The primitive variables may still be obtained analytically,

$$\boldsymbol{p}_1 = \boldsymbol{C}^{-1}\cdot\boldsymbol{A}^{-1}\cdot[(\boldsymbol{q}_1-\boldsymbol{q}_0)-\boldsymbol{D}\cdot\boldsymbol{b}-\boldsymbol{\tau}], \tag{B2}$$

though (B2) suffers the same problems as its conformal analog in the inviscid limit, and all terms vanish except for $\boldsymbol{C}^{-1}\cdot A^{-1}\cdot\boldsymbol{\tau}$, which diverges at finite grid resolution.

To stabilize the scheme in these cases, one may compute $\boldsymbol{p}_1^{PF}$ using (37) and compute a set of shifted variables $\tilde{\boldsymbol{q}}$, where (38) generalizes to

$$\tilde{\boldsymbol{q}}_1 \equiv \boldsymbol{q}_1 - \boldsymbol{q}_1|_{\boldsymbol{p}_1\to\boldsymbol{p}_1^{PF}} = \boldsymbol{A}\cdot\boldsymbol{C}\cdot(\boldsymbol{p}_1-\boldsymbol{p}_1^{PF}), \tag{B3}$$

implying the solution $\boldsymbol{p}_1(\tilde{\boldsymbol{q}}_1)$ is

$$\boldsymbol{p}_1 = \boldsymbol{C}^{-1}\cdot\boldsymbol{A}^{-1}\cdot\tilde{\boldsymbol{q}}_1 + \boldsymbol{p}_1^{PF}. \tag{B4}$$

Assuming one suitably modifies (31) to accommodate the choice of BDNK primitive variables, and one has a perfect fluid primitive variable solution for the case of interest[20] [to replace (29)], one may use (B3) and (B4) in place of (38) and (39) in the algorithm described in Sec. III C to obtain stable, convergent solutions to the BDNK equations in the inviscid limit.

APPENDIX C: REVIEW OF WENO RECONSTRUCTION

For the sake of simplicity, we will review the WENO reconstruction algorithm for a problem with variation only in one dimension; hence we will consider how the algorithm constructs the primitive variables $p^{\pm}_{i+1/2}$ at the right (+) and left (−) sides of the cell interface located at $x_{i+1/2} = x_i + \frac{1}{2}h$, where h is the grid spacing. Beginning with the reconstructed value at the right side of the interface, $p^{+}_{i+1/2}$, the WENO algorithm begins with the computation of three so-called ENO polynomials constructed from the cell averages,

$$\begin{aligned}
v^0_{i+1/2} &= -\frac{1}{6}\bar{p}_{i-2} + \frac{5}{6}\bar{p}_{i-1} + \frac{1}{3}\bar{p}_i,\\
v^1_{i+1/2} &= \frac{1}{3}\bar{p}_{i-1} + \frac{5}{6}\bar{p}_i - \frac{1}{6}\bar{p}_{i+1},\\
v^2_{i+1/2} &= \frac{11}{6}\bar{p}_i - \frac{7}{6}\bar{p}_{i+1} + \frac{1}{3}\bar{p}_{i+2}.
\end{aligned} \tag{C1}$$

Each of these stencils on its own constitutes an approximation to $p^{+}_{i+1/2}$ that is third-order accurate in the grid spacing h. WENO achieves the essentially nonoscillatory property by adaptively weighting how much of each stencil goes into the final approximation for $p^{+}_{i+1/2}$ using a set of smoothness indicators,

$$\begin{aligned}
\beta^0 &= \frac{1}{4}(3\bar{p}_i - 4\bar{p}_{i+1} + \bar{p}_{i+2})^2 + \frac{13}{12}(\bar{p}_i - 2\bar{p}_{i+1} + \bar{p}_{i+2})^2,\\
\beta^1 &= \frac{1}{4}(\bar{p}_{i+1} - \bar{p}_{i-1})^2 + \frac{13}{12}(\bar{p}_{i-1} - 2\bar{p}_i + \bar{p}_{i+1})^2,\\
\beta^2 &= \frac{1}{4}(\bar{p}_{i-2} - 4\bar{p}_{i-1} + 3\bar{p}_i)^2 + \frac{13}{12}(\bar{p}_{i-2} - 2\bar{p}_{i-1} + \bar{p}_i)^2,
\end{aligned} \tag{C2}$$

where β^k is large when the stencil $v^k_{i+1/2}$ contains a sharp gradient. Such stencils should have small weights in the final reconstructed primitive variable, then, which is achieved by writing the weights w_k as

$$w_k = \frac{\alpha_k}{\sum_l \alpha_l}, \quad \alpha_k = \frac{d_k}{(\epsilon_W + \beta^k)^2}, \quad d_k = \left(\frac{3}{10}, \frac{3}{5}, \frac{1}{10}\right), \tag{C3}$$

where the constant linear weights d_k are chosen such that the reconstructed solution attains the highest possible order of accuracy (fifth order) when the solution is smooth (β^k is small) in all three stencils.

The quantity ϵ_W is a free parameter which is inserted to prevent divide-by-zero errors when the smoothness indicators β^k vanish. The sensitivity of the WENO algorithm to sharp features in the solution depends strongly on the magnitude of ϵ_W. Cases where ϵ_W is small can yield w_k far from d_k in nonsmooth regions, resulting in significant differences between the stencils being used across the grid. Conversely, the limit $\epsilon_W \to \infty$ forces $w_k \to d_k$, recovering a fixed fifth-order reconstruction for $p^{\pm}_{i+1/2}$.

The final WENO approximation for $p^{+}_{i+1/2}$ is given by

$$p^{+}_{i+1/2} = w_0 v^0_{i+1/2} + w_1 v^1_{i+1/2} + w_2 v^2_{i+1/2}, \tag{C4}$$

which, again, gives the value of p at the right side of the interface at $x_{i+1/2}$. At the left side of the interface, the approximation is achieved by reflecting the stencils (C1) across the interface, which yields ENO polynomials

$$\begin{aligned}
u^0_{i+1/2} &= \frac{1}{3}\bar{p}_i + \frac{5}{6}\bar{p}_{i+1} - \frac{1}{6}\bar{p}_{i+2},\\
u^1_{i+1/2} &= -\frac{1}{6}\bar{p}_{i-1} + \frac{5}{6}\bar{p}_i + \frac{1}{3}\bar{p}_{i+1},\\
u^2_{i+1/2} &= \frac{1}{3}\bar{p}_{i-2} - \frac{7}{6}\bar{p}_{i-1} + \frac{11}{6}\bar{p}_i.
\end{aligned} \tag{C5}$$

The smoothness indicators and linear weights (C3) remain the same, giving the final approximation

$$p^{-}_{i+1/2} = w_0 u^0_{i+1/2} + w_1 u^1_{i+1/2} + w_2 u^2_{i+1/2}. \tag{C6}$$

[20] Note that in general the perfect fluid primitive variable solution is unobtainable analytically, so it is likely that a numerical solver will be necessary in cells where the physical viscosity is unresolved. In cells where it is resolved, however, the analytic BDNK primitive solution (B4) should be used.

For 2D simulations on uniform Cartesian grids like those considered here, WENO reconstruction is applied in the same way in both spatial directions; to obtain $p^{\pm}_{i,j+1/2}$, simply keep the index i constant and swap $i \to j$ in (C1)–(C6).

APPENDIX D: CONVERGENCE TESTS

To check the validity of the simulation results presented here, we have performed a number of different convergence tests. Principal among these is the computation of a set of independent residuals R_N, which are copies of the equations of motion (1) and (2) with different discretizations from those solved in the solution algorithm. For a second-order-accurate numerical scheme (as presented here), a second-order-accurate independent residual should yield $R_N \propto O(h^2)$, where the grid spacing h is related to the number of grid points N_x by $h = \frac{L_+ - L_-}{N_x}$, and $L_\pm$ are the left and right sides of the domain in either spatial coordinate (as we use equal grid spacing in x and y). Hence, a quantitative measure of convergence would be to compute the instantaneous convergence factor

$$Q_N(t) = \frac{||R_{N/2}||}{||R_N||}, \tag{D1}$$

where $||\cdot||$ is any vector norm. Here we use the 1-norm, defined for a vector X to be $||X||_1 \equiv \sum_i |X_i|$. For a second-order-accurate scheme, when the solution is sufficiently smooth, one can show that $Q_N(t) \to 4$ as $h \to 0$.

The convergence factor $Q_N(t)$ implicitly assumes that the solution is smooth everywhere, which is not the case here when unresolved shock waves are present. In these cases, we instead directly compute the independent

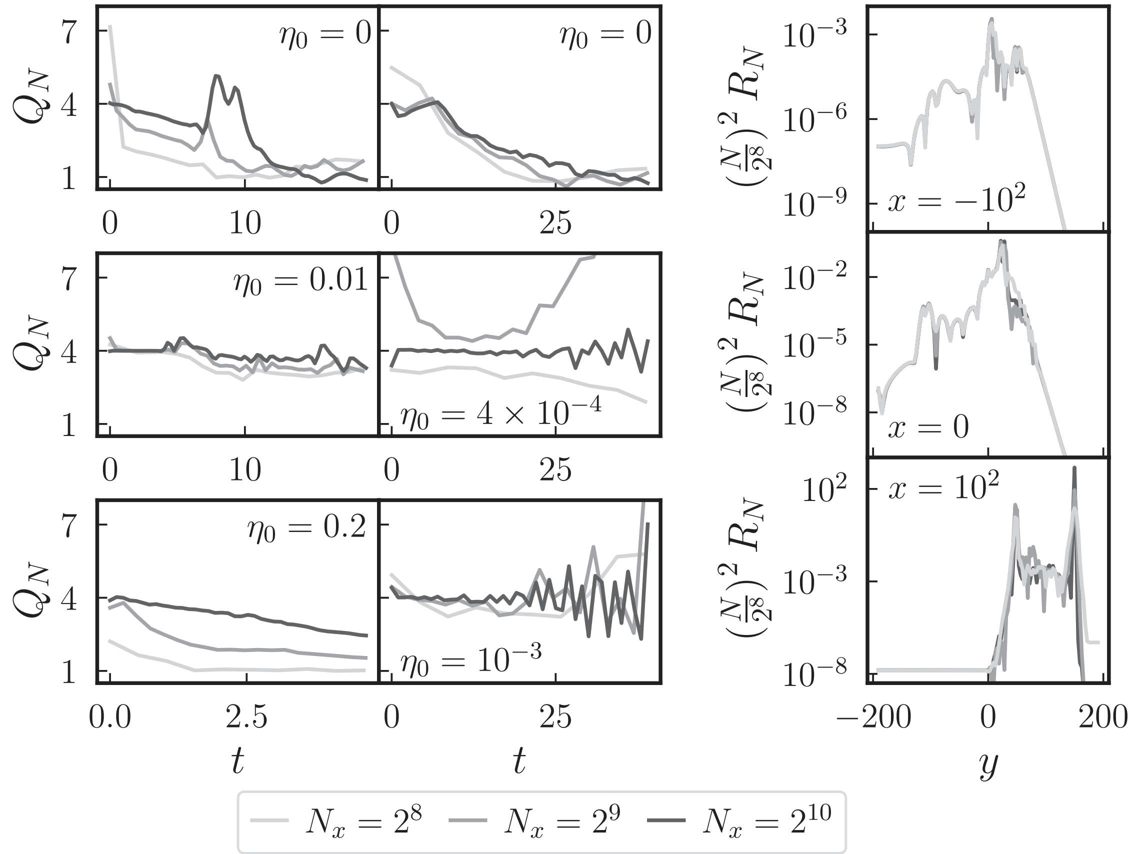

FIG. 9. Convergence plots corresponding to an independent (Crank-Nicolson second-order finite-difference) discretization of the x component of (1), $\nabla_c T^{cx} = 0$, for the 2D simulations shown above. Leftmost column: plots of $Q_N(t)$ for the viscous rotor simulations showing that the curves approach second-order convergence as resolution increases (shown with finer grids in successively darker colors, corresponding to resolutions $N_x = 2^8, 2^9, 2^{10}$). Middle column: similar plots of $Q_N(t)$ for the Kelvin-Helmholtz-unstable initial data. Rightmost panel: plots of the residual $\nabla_c T^{cx}$ for the 2D oblique shock initial data, at the time shown in Fig. 5, with the same color coding by resolution, scaled such that all curves should overlap if they are converging at second order, e.g., the $N_x = 2^9$ curve is multiplied by 4 and the $N_x = 2^{10}$ curve is multiplied by 16. The three panels show successive slices through the domain at constant x, and the top two show convergence at the expected order (all curves overlap). The bottom curve shows a slice through the shock wave and converges roughly at the expected order everywhere outside the spikes which appear at the shock fronts. Increasing resolution should produce taller, thinner spikes at the shock wave until it is finally resolved and the solution begins converging there at second order.

residuals R_N over the spatial grid and confirm that these obey $R_N \propto O(h^2)$ in smooth regions. An unresolved shock wave appears as step function discontinuity, which in the independent residual (which typically involves derivatives across the discontinuity) generates a delta-function-like peak which grows taller and narrower with resolution. We confirm that this behavior occurs at the unresolved shock waves present in our simulations.

Figure 9 shows the convergence of an independent residual—namely, a second-order Crank-Nicolson finite-difference discretization of the x component of (1)—for the viscous rotor, Kelvin-Helmholtz, and 2D oblique shock wave simulations in columns from left to right. For the left two columns, the solutions are free of unresolved shock waves and $Q_N(t)$ is a good measure of convergence; for each of the cases shown (which differ in viscosity), the solutions approach $Q_N(t) = 4$ with increasing resolution (which is denoted with lines of increasing darkness). Note that the inviscid simulations lose convergence at late times; this is because they begin forming features at the grid scale, which are unresolved at lower resolution. The time at which convergence is lost, however, is pushed later and later as resolution increases, as expected. The third column of the figure shows the independent residual R_N on constant-x slices at $t = 220$ as in Fig. 5, as a function of y, scaled by the expected order of convergence such that the three lines should overlap if the scheme is converging at second order. One can see that the top two slices exhibit the expected order of convergence, and all three resolutions lie on top of each other; in the bottom plot (which passes through the shock waves), delta-function-like spikes form at the two shock fronts and grow taller and narrower with resolution, as expected. Elsewhere the solutions roughly converge at the expected rate, though the sharp gradients in this region produce some numerical “noise” as well.

Similar trends to those shown in Fig. 9 appear in independent residuals of the other components of (1); computations of the self-convergence of the hydrodynamic variables $\{\xi, n, u^x, u^y\}$ are even better behaved and converge at second order as well.

[1] P. Kovtun, J. Phys. A **45**, 473001 (2012).

[2] P. Romatschke and U. Romatschke, arXiv:1712.05815.

[3] M. G. Alford, L. Bovard, M. Hanauske, L. Rezzolla, and K. Schwenzer, Phys. Rev. Lett. **120**, 041101 (2018).

[4] E. R. Most, S. P. Harris, C. Plumberg, M. G. Alford, J. Noronha, J. Noronha-Hostler, F. Pretorius, H. Witek, and N. Yunes, Mon. Not. R. Astron. Soc. **509**, 1096 (2021).

[5] C. Eckart, Phys. Rev. **58**, 919 (1940).

[6] L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, edited by L. D. Landau and E. M. Lifshitz, Course of Theoretical Physics, Vol. 6, 2nd ed. (Butterworth-Heinemann, London, 1987).

[7] W. A. Hiscock and L. Lindblom, Ann. Phys. (N.Y.) **151**, 466 (1983).

[8] W. A. Hiscock and L. Lindblom, Phys. Rev. D **31**, 725 (1985).

[9] I. Müller, Z. Phys. **198**, 329 (1967).

[10] W. Israel, Ann. Phys. (N.Y.) **100**, 310 (1976).

[11] W. Israel and J. Stewart, Ann. Phys. (N.Y.) **118**, 341 (1979).

[12] H. Saida, R. Takahashi, and H. Nagakura, arXiv:1002.0187.

[13] M. Shibata, K. Kiuchi, and Y.-i. Sekiguchi, Phys. Rev. D **95**, 083005 (2017).

[14] B. Schenke, S. Jeon, and C. Gale, Phys. Rev. Lett. **106**, 042301 (2011).

[15] B. Schenke, S. Jeon, and C. Gale, Phys. Rev. C **85**, 024901 (2012).

[16] R. Baier, P. Romatschke, D. T. Son, A. O. Starinets, and M. A. Stephanov, J. High Energy Phys. 04 (2008) 100.

[17] G. S. Denicol, H. Niemi, E. Molnar, and D. H. Rischke, Phys. Rev. D **85**, 114047 (2012); **91**, 039902(E) (2015).

[18] M. Martinez and M. Strickland, Nucl. Phys. **A848**, 183 (2010).

[19] W. Florkowski and R. Ryblewski, Phys. Rev. C **83**, 034907 (2011).

[20] P. Van and T. S. Biro, Eur. Phys. J. Special Topics **155**, 201 (2008).

[21] P. Van and T. S. Biro, Phys. Lett. B **709**, 106 (2012).

[22] H. Freistühler and B. Temple, Proc. R. Soc. A **470**, 20140055 (2014).

[23] H. Freistühler and B. Temple, Proc. R. Soc. A **473**, 20160729 (2017).

[24] H. Freistühler and B. Temple, J. Math. Phys. (N.Y.) **59**, 063101 (2018).

[25] F. S. Bemfica, M. M. Disconzi, and J. Noronha, Phys. Rev. D **98**, 104064 (2018).

[26] F. S. Bemfica, M. M. Disconzi, and J. Noronha, arXiv:2009.11388.

[27] P. Kovtun, J. High Energy Phys. 10 (2019) 034.

[28] R. E. Hoult and P. Kovtun, J. High Energy Phys. 06 (2020) 067.

[29] J. Noronha, M. Spaliński, and E. Speranza, arXiv:2105.01034.

[30] T. Dore, L. Gavassino, D. Montenegro, M. Shokri, and G. Torrieri, arXiv:2109.06389.

[31] G. S. Rocha and G. S. Denicol, Phys. Rev. D **104**, 096016 (2021).

[32] F. S. Bemfica, M. M. Disconzi, V. Hoang, J. Noronha, and M. Radosz, Phys. Rev. Lett. **126**, 222301 (2021).

[33] F. S. Bemfica, M. M. Disconzi, and J. Noronha, Phys. Rev. Lett. **122**, 221602 (2019).

[34] C. Plumberg, D. Almaalol, T. Dore, J. Noronha, and J. Noronha-Hostler, arXiv:2103.15889.

[35] T. S. Olson and W. A. Hiscock, Ann. Phys. (N.Y.) **204**, 331 (1990).

[36] A. Pandya and F. Pretorius, Phys. Rev. D **104**, 023015 (2021).

[37] H. Freistuhler, Phys. Rev. D **103**, 124045 (2021).

[38] M. Chandra, C. F. Gammie, F. Foucart, and E. Quataert, Astrophys. J. **810**, 162 (2015).

[39] G. S. Denicol, X.-G. Huang, E. Molnár, G. M. Monteiro, H. Niemi, J. Noronha, D. H. Rischke, and Q. Wang, Phys. Rev. D **98**, 076009 (2018).

[40] G. S. Denicol, E. Molnár, H. Niemi, and D. H. Rischke, Phys. Rev. D **99**, 056017 (2019).

[41] A. K. Panda, A. Dash, R. Biswas, and V. Roy, J. High Energy Phys. 03 (2021) 216.

[42] A. K. Panda, A. Dash, R. Biswas, and V. Roy, Phys. Rev. D **104**, 054004 (2021).

[43] M. Chandra, F. Foucart, and C. F. Gammie, Astrophys. J. **837**, 92 (2017).

[44] E. R. Most and J. Noronha, Phys. Rev. D **104**, 103028 (2021).

[45] M. Takamoto and S.-i. Inutsuka, J. Comput. Phys. **230**, 7002 (2011).

[46] K. Okamoto and C. Nonaka, Eur. Phys. J. C **77**, 383 (2017).

[47] R. J. LeVeque, *Numerical Methods for Conservation Laws* (Birkhauser, Basel, Switzerland, 2006).

[48] M. P. Heller, R. A. Janik, and P. Witaszczyk, Phys. Rev. Lett. **110**, 211602 (2013).

[49] A. Buchel, M. P. Heller, and J. Noronha, Phys. Rev. D **94**, 106011 (2016).

[50] G. S. Denicol and J. Noronha, arXiv:1608.07869.

[51] M. P. Heller, A. Kurkela, M. Spaliński, and V. Svensson, Phys. Rev. D **97**, 091503 (2018).

[52] M. P. Heller and M. Spalinski, Phys. Rev. Lett. **115**, 072501 (2015).

[53] P. Romatschke, Phys. Rev. Lett. **120**, 012301 (2018).

[54] S. Grozdanov and N. Kaplis, Phys. Rev. D **93**, 066012 (2016).

[55] I. De Schepper, H. Van Beyeren, and M. Ernst, Physica (Amsterdam) **75**, 1 (1974).

[56] P. Kovtun, G. D. Moore, and P. Romatschke, Phys. Rev. D **84**, 025006 (2011).

[57] C.-W. Shu, Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws, in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations: Lectures Given at the 2nd Session of the Centro Internazionale Matematico Estivo (C.I.M.E.) held in Cetraro, Italy, 1997*, edited by A. Quarteroni (Springer, Berlin, Heidelberg, 1998), pp. 325–432.

[58] D. Levy, G. Puppo, and G. Russo, Math. Model. Numer. Anal. **33**, 547 (1999).

[59] A. Kurganov and E. Tadmor, J. Comput. Phys. **160**, 241 (2000).

[60] L. Rezzolla and O. Zanotti, *Relativistic Hydrodynamics* (Oxford University Press, New York, 2013), 10.1093/acprof:oso/9780198528906.001.0001.

[61] M. J. Berger and J. Oliger, J. Comput. Phys. **53**, 484 (1984).

[62] J. A. Font, Living Rev. Relativity **3**, 2 (2000).

[63] X.-D. Liu, S. Osher, and T. Chan, J. Comput. Phys. **115**, 200 (1994).

[64] G.-S. Jiang and C.-W. Shu, J. Comput. Phys. **126**, 202 (1996).

[65] M. Dumbser, S. Chiocchetti, and I. Peshkov, in *Continuum Mechanics, Applied Mathematics and Scientific Computing: Godunov's Legacy* (Springer, New York, 2020), pp. 125–134.

[66] A. Lucas-Serrano, J. A. Font, J. M. Ibanez, and J. M. Marti, Astron. Astrophys. **428**, 703 (2004).

[67] P. Roe, J. Comput. Phys. **135**, 250 (1997).

[68] A. Kurganov and E. Tadmor, J. Comput. Phys. **160**, 241 (2000).

[69] A. Harten, P. D. Lax, and B. v. Leer, SIAM Rev. **25**, 35 (1983).

[70] P. D. Lax, Commun. Pure Appl. Math. **7**, 159 (1954).

[71] V. A. Titarev and E. F. Toro, J. Comput. Phys. **201**, 238 (2004).

[72] W. E. East, F. Pretorius, and B. C. Stephens, Phys. Rev. D **85**, 124010 (2012).

[73] D. Lecoanet, M. McCourt, E. Quataert, K. J. Burns, G. M. Vasil, J. S. Oishi, B. P. Brown, J. M. Stone, and R. M. O'Leary, Mon. Not. R. Astron. Soc. **455**, 4274 (2016).

[74] S. C. Noble, C. F. Gammie, J. C. McKinney, and L. Del Zanna, Astrophys. J. **641**, 626 (2006).

[75] P. Arras and N. N. Weinberg, Mon. Not. R. Astron. Soc. **486**, 1424 (2019).

[76] F. Foucart, M. Chandra, C. F. Gammie, E. Quataert, and A. Tchekhovskoy, Mon. Not. R. Astron. Soc. **470**, 2240 (2017).

[77] V. A. Dommes, M. E. Gusakov, and P. S. Shternin, Phys. Rev. D **101**, 103020 (2020).

[78] J. Armas and F. Camilloni, arXiv:2201.06847.