Title

Primitive Variable Determination in Conservative Relativistic Magnetohydrodynamic Simulations

Journal

SIAM Journal on Scientific Computing, 36(4)

ISSN

1064-8275

Authors

Newman, William I
Hamlin, Nathaniel D

Publication Date

2014

DOI

10.1137/140956749

SIAM J. SCI. COMPUT.
Vol. 36, No. 4, pp. B661–B683

PRIMITIVE VARIABLE DETERMINATION IN CONSERVATIVE RELATIVISTIC MAGNETOHYDRODYNAMIC SIMULATIONS*

WILLIAM I. NEWMAN† AND NATHANIEL D. HAMLIN‡

Abstract. In nonrelativistic hydrodynamics and magnetohydrodynamics, conservative integration schemes for the fluid equations of motion are generally employed. The computed quantities, namely, the mass density, (vector) momentum density, and energy density, can readily be converted back into the primitive variables that define the problem, namely, the mass density, (vector) velocity, and thermal pressure. In practical terms, the primitive variables can be "peeled away" from the computed variables. In relativistic problems, however, the appearance of the Lorentz factor in the computed quantities dramatically complicates the problem owing to its near-singular dependence upon relativistic velocities. Conservative integration schemes for the hyperbolic partial differential equations of special relativistic magnetohydrodynamics (RMHD) yield estimates of the five conserved quantities that are related in a highly nonlinear way to the five primitive variables. We also observe that an equivalent set of five nonlinear equations emerges in describing the general relativistic magnetohydrodynamics problem. Conversion from the conserved quantities to primitive variables is the most computationally intensive part of the simulation, consuming almost all the computational effort. This paper presents a new algorithm for accurately and rapidly addressing this problem. We provide an analytic representation for this nonlinear system as a *single* equation in a *single* unknown. This equation lends itself to an iterative approach, emerging from its underlying physical and analytic properties, one whose convergence is rapid and sufficiently close to geometric that the Aitken acceleration scheme renders the iterations quadratically convergent. The new algorithm enjoys robust convergence properties that render it well-suited to parallel computing architectures. We show how our new scheme facilitates the rapid and accurate computation of solutions to the RMHD equations for moderate Lorentz factors as well as extreme relativistic situations that are observed in high-energy astrophysical objects, where the Lorentz factor can exceed 10^4 and magnetic fields can exceed 10^{13} Gauss, and in laboratory plasmas associated with fusion research where very high velocities and magnetic fields are introduced.

Key words. special relativity, hydrodynamics, magnetohydrodynamics, astrophysics, plasma physics, general relativity

AMS subject classifications. 83Axx, 83Cxx, 76Wxx, 85A30 and 82D10

DOI. 10.1137/140956749

1. Introduction.

Over the past 30 years and especially during the current decade, astrophysicists have discovered many different kinds of highly energetic astrophysical phenomena, including pulsars, neutron stars, quasars, black holes, soft gamma ray bursters, blazars, and magnetars—see, for example, Newman and Newman [31] for a treatment of the latter. Recently, Hamlin and Newman [16] explored the role of the relativistic Kelvin–Helmholtz instability in the presence of strong magnetic fields, a situation commonly encountered in laboratory plasmas, employed in fusion-related experiments, as well as in astrophysical environments. In these environments, especially the highly energetic recently discovered astrophysical ones, an explosive event can propel matter at a velocity $\vec{v}$ remarkably close to the speed of light c. The observed Lorentz factor $[1-(v/c)^2]^{-1/2}$ can exceed 100, corresponding

*Submitted to the journal's Computational Methods in Science and Engineering section February 18, 2014; accepted for publication (in revised form) April 29, 2014; published electronically July 29, 2014.

http://www.siam.org/journals/sisc/36-4/95674.html

†Departments of Earth, Planetary, and Space Sciences, Physics and Astronomy, and Mathematics, University of California, Los Angeles, CA 90095 (win@ucla.edu).

‡Laboratory of Plasma Studies, Cornell University, Ithaca, NY 14853 (nh322@cornell.edu).

to a velocity in excess of 99.99% of the speed of light. Moreover, due to the ultrahigh temperatures involved (10^{10} K and above), all matter becomes ionized. The equation of state (EOS) for such a plasma is simple to calculate but is fundamentally relativistic. The extremely rapid rotation rate of some of these astrophysical objects—e.g., pulsars, stars slightly more massive than our sun that have collapsed to around 10 km in radius—is a by-product of angular momentum conservation. Pulsars rotate as rapidly as 700 times *per second* in contrast with our sun's rotation period of 27 days. Dipolar magnetic field lines, associated with the progenitor star prior to its collapse, are violently wound up, conserving the magnetic flux and producing fields in excess of 10^{13} Gauss in the recently discovered magnetars. In contrast, the Earth's magnetic field is a fraction of a Gauss. Very high magnetic fields are also evident in jet-like features that are associated with many of these superenergetic objects. As a measure of interest generated by these objects, several thousand papers have been published during the last two decades dealing with them. At the same time laboratory experiments associated with magnetic inertial confinement of plasmas in the development of thermonuclear fusion employ increasingly higher velocities, magnetic fields, and temperatures.

This paper focuses on developing a new algorithm for converting the computed conserved quantities into the primitive variables which form the basis of the underlying physics and their associated partial differential equations. One critical issue here is that the conserved quantities depend on the Lorentz factor, which results in the relativistic dilation of time and contraction of length. The Lorentz factor depends very sensitively upon the flow velocity, which can be very close to that of light. In the case of extreme relativistic flows, roundoff errors associated with the computation of the Lorentz factor pose an impediment to convergence of all algorithms. This sensitivity, in turn, renders the conversion problem to be extraordinarily nonlinear and ill-conditioned. An equally critical issue is that the primitive variable recovery algorithm must be absolutely and rapidly convergent at every spatial grid point. The failure of the algorithm to converge at even a single grid point causes immense computational problems, particularly in parallel computing architectures. To effectively address this impediment to parallel computing architectures, it is essential that some ad hoc accommodations be introduced, such as interpolating the value of primitive variables from surrounding cells. While such accommodations are available, their implementation necessarily increases the overall computational cost. Accordingly, this paper addresses what is evidently one of the most challenging technical problems associated with performing accurate and rapid computational simulations of these high-energy astrophysical phenomena.

Having presented various physical phenomena that motivate the use of relativistic magnetohydrodynamics (RMHD) single-fluid modeling, we structure the remainder of our paper as follows. In section 2, we present the RMHD single-fluid equations in conservative form and describe in more detail the problem of recovering the primitive variables from the conserved quantities. We then present an overview of the history of primitive variable recovery algorithms and describe the technical challenges they have encountered. We then describe how our algorithm meets these challenges. In section 3, we review some fundamental relativistic identities. In section 4, we discuss the method of Komissarov [24] for recovering primitive variables, along with difficulties associated with using this method. In section 5, we present our iterative method for recovering primitive variables. In section 6, we discuss the convergence properties of our method, including its performance in the moderately relativistic RMHD simulations from Hamlin and Newman [16] and then in the highly relativistic cases presented

in Komissarov [24]. In section 7, we discuss our use of the Aitken accelerant to render the convergence of our scheme quadratic, which makes quadratic the speed of convergence. In section 8, we present the step-by-step implementation of our algorithm. In section 9, we compare our algorithm with existing algorithms for primitive variable recovery. We then discuss the use of our algorithm in conjunction with the latest generation of shock-capturing schemes as a direction of future research, including the RMHD modeling of highly relativistic shocks, which is one type of phenomenon to which we believe our algorithm is particularly well-suited.

2. Computational background and technical challenges. In nonrelativistic and relativistic MHD, the associated initial value problem is specified by a set of well-posed partial differential equations and the primitive variables density ρ, pressure P, the velocity vector $\vec{v}$, and the magnetic field vector $\vec{B}$. For convenience, we will refer to the set of eight primitive variables as the eight-vector $\vec{\mathcal{P}}$. Conservative integration schemes can be expressed in the form

$$\frac{\partial \vec{\mathcal{Q}}}{\partial t} + \frac{\partial \vec{\mathcal{F}}^x}{\partial x} + \frac{\partial \vec{\mathcal{F}}^y}{\partial y} + \frac{\partial \vec{\mathcal{F}}^z}{\partial z} = 0. \tag{2.1}$$

Here, the three "flux vectors" $\vec{\mathcal{F}}^x$, $\vec{\mathcal{F}}^y$, and $\vec{\mathcal{F}}^z$ are functions of the primitive variables $\vec{\mathcal{P}}$, as are the conserved quantities $\vec{\mathcal{Q}}$. Since the fundamental investigations for hyperbolic conservation laws by Lax [25] were performed, many numerical schemes with higher order accuracy have emerged. At each time step and at every point in the grid, we must take our estimate of the conserved quantities $\vec{\mathcal{Q}}$ and determine from them the primitive variables $\vec{\mathcal{P}}$ in order to proceed. In nonrelativistic MHD, the conversion from $\vec{\mathcal{Q}}$ to $\vec{\mathcal{P}}$ is explicit and poses no difficulties—see Van Putten [35] and Dai and Woodward [9] for reviews.

In conservative integration schemes for RMHD, the conversion is not direct—apart from the magnetic field components—and a set of five nonlinear algebraic equations must be solved to extract the remaining five primitive variables. While the underlying equations are invariant under a Lorentz transformation, this feature does not provide a mechanism for converting the conserved quantities into their underlying primitive variables. Typically, this has been done using multidimensional Newton methods or other multivariate iteration schemes—see, for example, Balsara [3] and other sources cited below—and this deceptively simple conversion problem generally consumes most of the computation time used in solving RMHD problems [24]. Thus, a task that would otherwise be relegated to a technical detail has become a serious limiting factor in our ability to simulate astrophysical and laboratory RMHD problems on the computer.

We now describe some of the underlying history of the associated computational problem, i.e., that of recovering the primitive variables in RMHD simulations, and the challenge that emerges. We then discuss the technical challenges faced by these algorithms and the manner in which our algorithm overcomes these challenges. As we will discuss, our algorithm, unlike previous ones, exhibits rapid and accurate convergence at every grid point in a simulation, without the need to reduce the time step, and thus it can be readily adapted to parallel computing architectures.

We begin by reviewing some of the progress made during the last decade in developing algorithms for primitive variable recovery, which has been an unrelenting challenge to RMHD simulations. Balsara [3] developed a method for RMHD simulations in which he recovers primitive variables by solving a 5×5 system of equations for

the density, velocity, and pressure. Del Zanna, Bucciantini, and Londrillo [10] say that "this process is neither efficient nor stable when relativistic effects are strong." Noble et al. [32] performed a comparison of several primitive variable recovery schemes, one of them being the 5×5 scheme used by Balsara, saying in their conclusion that this scheme is "particularly slow and inaccurate." Komissarov [24] also performed an RMHD study, saying that in order to recover the primitive variables from conserved quantities "one has to solve numerically a system of rather complicated nonlinear algebraic equations." He later says that "these calculations are the most expensive part of our code and their optimization is therefore a matter of some importance." Komissarov [24] reduces the 5×5 system to three equations in three unknowns. These can be combined to yield a single eighth-degree polynomial, which we will explore in more detail in section 4. Noble et al. [32] present an algorithm designed to solve an eighth-degree polynomial but say that this method is "computationally expensive" and "can yield ambiguous results due to amplification of roundoff error, making it difficult to identify the correct solution." Del Zanna, Bucciantini, and Londrillo [10] improved upon these methods by reducing the problem to solving a seventh-degree polynomial in a single unknown. This method was used in moderately relativistic MHD simulations of pulsar wind nebulae [5, 4, 1] and relativistic winds from rotating neutron stars [7, 6]. Mignone and McKinney [28] developed a better method that solves a seventh-degree polynomial in a manner that exhibits improved behavior in the nonrelativistic and ultrarelativistic limits. The algorithm of Mignone and McKinney [28] was used in performing three-dimensional RMHD simulations of astrophysical jets [29]. An equivalent problem emerges in general relativistic magnetohydrodynamics (GRMHD) and is reviewed in Font [14], which also provides an overview of the techniques and codes in use. It is noteworthy that, in most instances, investigators employed iterative schemes to solve the nonlinear equations and not the associated polynomial.

Clearly, the recovery of primitive variables in RMHD simulations is a difficult though important problem, owing to the wide applicability of RMHD simulations in exploring a variety of astrophysical and laboratory plasma phenomena. However, one central restriction of the aforementioned algorithms is the required use of root-finding methods such as the Newton–Raphson method (also known as Newton's method) in solving high-order polynomials. Importantly, Newton's method is quadratically convergent when an accurate initial estimate of the solution is available. This is complicated further by the extraordinary sensitivity of the Lorentz factor to ultrarelativistic velocities, i.e., velocities that are very close to that of light. Thus, unless initial estimates of the solution are very accurate, the iteration often diverges. The accuracy of the initial estimate can be improved by reducing the time step much below physically motivated constraints associated with fluid stability requirements. However, this in turn makes the numerical integration of the RMHD equations prohibitively expensive. Our method converges rapidly to the correct root *for any nonnegative initial guess of the pressure*, without any need to artificially reduce the time step below physical constraints.

As a result, our primitive variable recovery scheme is readily adaptable to massively parallel computing architectures, which are essential given the computational demands present in an RMHD simulation. In a grid with the order of 1 million spatial points, the failure at even a single point of the conversion scheme produces substantial difficulties, particularly in parallel processing. In methods of primitive variable conversion that require a very accurate initial estimate of the solution, it often becomes necessary to substantially reduce the overall time step in order to guarantee the op-

erational success of the numerical integrations. Therefore, as we have observed, the inability of other conversion schemes to converge at a few isolated points can seemingly necessitate the use of a much reduced time step over the entire grid, thereby increasing the computational burden by orders of magnitude. Because our method converges rapidly and absolutely *without requiring an accurate initial estimate*, it will converge at all grid points without requiring a reduced time step or other algorithmic intervention and is therefore well-suited for parallelization. This property of our method is also especially beneficial for the use of relatively high-order in space and time finite difference schemes consistent with the Courant–Friedrichs–Lewy stability criteria, which makes it desirable to employ time steps that are as large as is practical. As we will observe in Figure 2, when our algorithm is implemented in our computational methodology, it has been consistently successful at addressing the remarkable degree of spatial heterogeneity that emerges in a nonlinear RMHD simulation. Furthermore, as we will discuss at the end of section 9, we expect that our algorithm will be a tremendous asset to the RMHD modeling of highly relativistic phenomena.

Before presenting our algorithm, we review some important steps made by Komissarov [24] in solving this problem. We will show that his approach is formally equivalent to the problem of solving an eighth-degree polynomial—complicated physically by the possibility of stagnation points and of flows that are locally orthogonal to the magnetic field. We then pursue an alternative route to solving this algebraic problem. In so doing, we obtain a cubic equation that must be solved—and this can be done explicitly with the appropriate branch cut selected by asymptotic matching. This gives rise both to a single equation in one unknown that has a unique solution (assuming that it exists) and, importantly, a new iterative method for its solution. We will show that this iterative scheme converges geometrically and that acceleration methods render quadratic the speed of convergence. In typical simulations [16], we have employed the algorithm that we detail below without failure literally many billions of times.

3. Review of fundamental relativistic identities. We begin by employing $\vec{\beta}$ to describe the velocity vector $\vec{v}$ normalized by the speed of light, i.e., $\vec{\beta} = \vec{v}/c$. Then, we will set u^0 to be the Lorentz factor and u^0 will be employed as the time-like component of the velocity four-vector

$$u^0 \equiv \frac{1}{\sqrt{1-\beta^2}}, \tag{3.1}$$

where $\beta^2 = \vec{\beta} \cdot \vec{\beta}$. Accordingly we can write

$$\beta^2 = \frac{\left(u^0\right)^2 - 1}{\left(u^0\right)^2}. \tag{3.2}$$

We now use

$$\vec{u} = \vec{\beta} u^0 \tag{3.3}$$

to provide the three spatial components u^i, $i = 1, 2, 3$, for the proper velocity. By convention, Greek indices will run from 0 through 3, while Latin ones will run from 1 through 3. The metric $g^{\mu\nu}$ is diagonal with signature +2, i.e.,

$$g^{\mu\nu} = \begin{cases} 0 & \text{if } \mu \neq \nu, \\ 1 & \text{if } \mu = \nu \neq 0, \\ -1 & \text{if } \mu = \nu = 0, \end{cases} \tag{3.4}$$

where the usual tensorial rules apply. Accordingly, the Einstein summation convention yields

$$u_\alpha u^\alpha = \vec{u}\cdot\vec{u} - \left(u^0\right)^2 = -1 \tag{3.5}$$

for the velocity four-vector.

Similarly, there is a four-vector analogue for the magnetic field $\vec{B}$ which we denote b^α. We define the time-like component according to

$$b^0 = \vec{B}\cdot\vec{u}. \tag{3.6}$$

Related to this quantity, we now define using (3.3)

$$\mathcal{S} = \vec{B}\cdot\vec{\beta} = b^0/u^0. \tag{3.7}$$

This quantity had an important role in previous algorithms designed to deal with RMHD, and we will return to it later in this discussion. Importantly, $\mathcal{S}$ vanishes at stagnation points—where the velocity $\vec{\beta}$ vanishes, and any location where the magnetic field is orthogonal to the flow. The space-like components b^i are defined according to

$$b^i = \frac{B^i + b^0 u^i}{u^0}. \tag{3.8}$$

Usually, b^α is derived from the electromagnetic field tensor—see Anile [2], for example. We will not reproduce its derivation here but will rely on the simple definitions given above. Inverting the latter expression, we can write

$$\vec{B} = \vec{b}u^0 - \vec{u}b^0, \tag{3.9}$$

whereby we can recover the standard magnetic field.

Finally, we state without proof two identities relating to b^α; their derivations are direct. First we have that

$$u_\alpha b^\alpha = 0, \tag{3.10}$$

showing that both four-vectors are mutually orthogonal. Second, we obtain that

$$\hat{b}^2\left(u^0\right)^2 = \mathcal{B}^2 + \left(b^0\right)^2, \tag{3.11}$$

where $\hat{b}^2$ is the square of the four-vector, i.e.,

$$\hat{b}^2 \equiv \vec{b}\cdot\vec{b} - \left(b^0\right)^2, \tag{3.12}$$

and $\mathcal{B}^2$ is the square of the three-vector $\vec{B}$. Note that in what follows the term $\hat{b}^2$ explicitly refers to the length squared of this four-vector and not to its second component b^2; we have introduced the caret $\hat{}$ to avoid confusion.

4. Komissarov's method of conversion. The focus of this paper emerges from the conversion of the conserved quantities $\vec{\mathcal{Q}}$ into primitive variables $\vec{\mathcal{P}}$ for RMHD. Clearly, $\vec{\beta}$ and $\vec{B}$, or their four-vector variants u^α and b^α, are fundamental quantities. In addition, the (rest) mass density ρ and the internal pressure P are also fundamental, just as they are in conventional, nonrelativistic, field-free gas dynamics. Associated with the density and pressure is the enthalpy $w(\rho, P)$ which

defines an associated EOS. Unfortunately, the establishment of the EOS for relativistic magnetized fluids is not without some controversy. We will, for the purpose of this paper, employ its most common representation, namely,

$$w(\rho, P) = \rho + \frac{\gamma}{\gamma - 1} P, \tag{4.1}$$

where $\gamma = 4/3$ is the (relativistic) ratio of specific heats. Taken together, the quantities $\vec{\beta}$, $\vec{B}$, ρ, and P constitute the primitive variables in RMHD, and we shall denote them by the eight-vector $\vec{\mathcal{P}}$. Explicitly, we can express the conserved eight-vector $\vec{\mathcal{Q}}$ as

$$\vec{\mathcal{Q}} = \begin{bmatrix} \left(w + \hat{b}^2\right) u^0 u^0 - \left(P + \frac{1}{2}\hat{b}^2\right) - b^0 b^0 \\ \left(w + \hat{b}^2\right) u^0 u^1 - b^0 b^1 \\ \left(w + \hat{b}^2\right) u^0 u^2 - b^0 b^2 \\ \left(w + \hat{b}^2\right) u^0 u^3 - b^0 b^3 \\ B^1 \\ B^2 \\ B^3 \\ \rho u^0 \end{bmatrix}. \tag{4.2}$$

While the determination of the magnetic field components is immediate, the remaining five quantities are highly nonlinear in their dependence upon the primitive variables.

Following Komissarov [24], we identify the first quantity with the total energy density e,

$$e = \left(w + \hat{b}^2\right) u^0 u^0 - \left(P + \frac{1}{2}\hat{b}^2\right) - b^0 b^0, \tag{4.3}$$

the next three with the momentum density m^i,

$$m^i = \left(w + \hat{b}^2\right) u^0 u^i - b^0 b^i, \tag{4.4}$$

and the last quantity with the material mass density $\tilde{\rho}$,

$$\tilde{\rho} = \rho u^0. \tag{4.5}$$

As Komissarov [24] notes, the latter five equations—since (4.4) is vectorial—can be reduced to three scalar equations. This is achieved, first, by introducing (3.3), (3.7), and (3.8) into (4.4), whereupon we obtain

$$m^i = \left[w\left(u^0\right)^2 + \mathcal{B}^2\right]\beta^i - \mathcal{S} B^i. \tag{4.6}$$

Further, we introduce the definitions

$$\mathcal{M}^2 = m^i m_i \quad \text{and} \quad \mathcal{T} = m^i B_i. \tag{4.7}$$

In so doing, we obtain the following system of three equations for the three unknowns P, u^0, and $\mathcal{S}$, given the known quantities e, $\mathcal{M}^2$, and $\mathcal{T}$:

$$
\begin{aligned}
e &= w\left(u^0\right)^2 - P + \frac{1}{2}\mathcal{B}^2 + \frac{1}{2}\left(\mathcal{B}^2\beta^2 - \mathcal{S}^2\right), \\
\mathcal{M}^2 &= \left[w\left(u^0\right)^2\right]^2 \beta^2 + \left[\mathcal{B}^2 + 2w\left(u^0\right)^2\right]\left(\mathcal{B}^2\beta^2 - \mathcal{S}^2\right), \\
\mathcal{T} &= w\left(u^0\right)^2 \mathcal{S}.
\end{aligned} \tag{4.8}
$$

(A typographical error appears in Komissarov's paper, which is corrected here.) Komissarov comments simply that "this system is solved iteratively"—and that the u^i are then obtained from (4.6). We will now show that these three equations can be combined into a single eighth-degree polynomial in $\mathcal{S}$.

We observe that the second and third of (4.8) can be combined to give

$$
\beta^2 = \frac{\mathcal{M}^2\mathcal{S}^2 + \mathcal{S}^3\left(\mathcal{S}\mathcal{B}^2 + 2\mathcal{T}\right)}{\left(\mathcal{T} + \mathcal{S}\mathcal{B}^2\right)^2}. \tag{4.9}
$$

Thus, if $\mathcal{S}$ were known, then the latter would give us β (as a function of $\mathcal{S}$). Accordingly, (3.1) would give us u^0 and (4.5) would give us ρ (from $\tilde{\rho}$). The third of (4.8) would then yield w, while the first of these equations would provide P as a consequence of the EOS (4.1). Taken together, therefore, it follows that we should be able to identify a single equation in $\mathcal{S}$ describing the system of Komissarov [24].

We now introduce (4.9) for β^2 into the first of (4.8) for e; similarly, we obtain u^0, w, and P—as described above—and introduce them into this equation, whereupon we obtain

$$
\begin{aligned}
0 = {} & \left(\tilde{\rho}^2\mathcal{B}^6 + \mathcal{T}^2\mathcal{B}^4\right)\mathcal{S}^8 + \left(-8\mathcal{B}^6 e + 4\tilde{\rho}^2\mathcal{B}^4 + 4\mathcal{B}^8 + 4\mathcal{B}^4\mathcal{M}^2\right)\mathcal{T}\mathcal{S}^7 + \left(4\mathcal{B}^{12} + 4\mathcal{B}^4\mathcal{M}^4\right. \\
& -16e\mathcal{B}^{10} + 5\tilde{\rho}^2\mathcal{B}^2\mathcal{T}^2 + 8\mathcal{B}^8\mathcal{M}^2 - 16e\mathcal{B}^6\mathcal{M}^2 - \tilde{\rho}^2\mathcal{B}^8 + 14\mathcal{T}^2\mathcal{B}^6 + \tilde{\rho}^2\mathcal{B}^4\mathcal{M}^2 \\
& \left.-16\mathcal{T}^2\mathcal{B}^4 e + 16e^2\mathcal{B}^8 + 2\mathcal{T}^2\mathcal{M}^2\mathcal{B}^2\right)\mathcal{S}^6 + \left(28\mathcal{B}^{10} + 16\mathcal{B}^4\mathcal{T}^2 + 2\tilde{\rho}^2\mathcal{B}^2\mathcal{M}^2 + 2\tilde{\rho}^2\mathcal{T}^2\right. \\
& \left.- 8\mathcal{T}^2\mathcal{B}^2 e + 64e^2\mathcal{B}^6 - 88\mathcal{B}^8 e + 4\mathcal{M}^4\mathcal{B}^2 + 32\mathcal{B}^6\mathcal{M}^2 - 4\tilde{\rho}^2\mathcal{B}^6 - 40\mathcal{M}^2 e\mathcal{B}^4\right)\mathcal{T}\mathcal{S}^5 \\
& + \left(96e^2\mathcal{B}^4 - 6\tilde{\rho}^2\mathcal{B}^4 + \tilde{\rho}^2\mathcal{M}^2 + 81\mathcal{B}^8 + \mathcal{M}^4 + 6\mathcal{T}^2\mathcal{B}^2 - 192\mathcal{B}^6 e - 32\mathcal{M}^2 e\mathcal{B}^2\right. \\
& \left.+ 46\mathcal{B}^4\mathcal{M}^2\right)\mathcal{T}^2\mathcal{S}^4 + \left(28\mathcal{B}^2\mathcal{M}^2 - 4\tilde{\rho}^2\mathcal{B}^2 + 124\mathcal{B}^6 - 208e\mathcal{B}^4 - 8\mathcal{M}^2 e + 64e^2\mathcal{B}^2\right) \\
& \cdot\mathcal{T}^3\mathcal{S}^3 + \left(6\mathcal{M}^2 - \tilde{\rho}^2 + 106\mathcal{B}^4 - 112e\mathcal{B}^2 + 16e^2\right)\mathcal{T}^4\mathcal{S}^2 + \left(-24e + 48\mathcal{B}^2\right)\mathcal{T}^5\mathcal{S} \\
& + 9\mathcal{T}^6.
\end{aligned} \tag{4.10}
$$

This equation is not especially useful. Although we have been able to distill from our eight underlying equations an eighth-degree polynomial, it does not provide any practical advantage, owing to its inherent complexity. Descartes' rule [19], for example, offers us nothing since the coefficients are so unmanageable. While the coefficients of this polynomial in $\mathcal{S}$ are real-valued, the solutions can still be complex, and we are interested only in solutions which are real-valued. Further, viable solutions are restricted to those which result in β^2, determined from (4.9), between 0 and 1. Finally, (3.7) makes it necessary that $\mathcal{S}^2 \leq \mathcal{B}^2$. Were we able to find a solution $\mathcal{S}$ to (4.10) such that $\mathcal{S}$ was real-valued and satisfied the two resulting constraints mentioned, we would then take the emergent value of β^2 from (4.9) and compute u^0, ρ, w, and then P as described above. Polynomial root finding methods—see the review in Ralston

and Rabinowitz [33]—do not in general make this an optimal route to pursue. Computationally, this is complicated further when $\mathcal{S} = 0$, and therefore (4.9) becomes an indeterminate form. While this eventuality describes a set of measure zero, it corresponds to a physical stagnation point or, alternatively, locations where the magnetic field and velocity flow are mutually orthogonal, the latter being of potential interest in astrophysical applications. Therefore, we now seek another methodology for converting our conserved quantities into the primitive variables.

5. Iterative method for variable conversion. In order to avoid the problems associated with $\mathcal{S}$ vanishing, let us focus on the quantity $w\left(u^0\right)^2$ which should *always* be positive definite. For convenience, we define $\mathcal{L}$ by

$$\mathcal{L} \equiv w\left(u^0\right)^2, \tag{5.1}$$

whereby we can arrange terms in the second of (4.8) to obtain

$$\beta^2 = \frac{\mathcal{M}^2\mathcal{L}^2 + \mathcal{T}^2\left(\mathcal{B}^2 + 2\mathcal{L}\right)}{\mathcal{L}^2\left(\mathcal{B}^2 + \mathcal{L}\right)^2}. \tag{5.2}$$

Since $\mathcal{L} > 0$, this expression—unlike (4.9), which is ill-defined when $\mathcal{S} = 0$—is always well-posed. Using this expression plus some algebraic manipulation, we find that we can write

$$\left(\mathcal{B}^2\beta^2 - \frac{\mathcal{T}^2}{\mathcal{L}^2}\right) = \frac{\mathcal{M}^2\mathcal{B}^2 - \mathcal{T}^2}{\left(\mathcal{B}^2 + \mathcal{L}\right)^2}. \tag{5.3}$$

We note that the left-hand side of this equation is just the last term in the first of (4.8), allowing us to write

$$\left(e + P + \frac{\mathcal{B}^2}{2}\right) = \left(\mathcal{B}^2 + \mathcal{L}\right) + \frac{1}{2}\frac{\mathcal{M}^2\mathcal{B}^2 - \mathcal{T}^2}{\left(\mathcal{B}^2 + \mathcal{L}\right)^2}. \tag{5.4}$$

We recognize this as being a cubic polynomial in the quantity which we now define by

$$\mathcal{E} \equiv \mathcal{B}^2 + \mathcal{L}. \tag{5.5}$$

We will rewrite (5.4) in the form

$$a\mathcal{E}^2 = \mathcal{E}^3 + d, \tag{5.6}$$

where

$$\begin{aligned} a &= e + P + \frac{\mathcal{B}^2}{2}, \\ d &= \frac{1}{2}\left(\mathcal{M}^2\mathcal{B}^2 - \mathcal{T}^2\right). \end{aligned} \tag{5.7}$$

We shall now focus on establishing a relaxation method based on (5.6). In particular, we note that $d \geq 0$, owing to the definition of $\mathcal{T}$ in (3.1), is known and that $a > 0$ is uncertain only in its dependence upon the pressure P which, by a's definition, is but a small contributor to the numerical value of a.

Suppose that we have a good approximation for P in the above—we will return to this issue later—and that we can assume that the value of a is known. (As a practical matter, a good estimate for the pressure at any point and time can be extrapolated from past values obtained by the integration scheme.) For convenience, we define a cubic polynomial

$$f(\mathcal{E}) = \mathcal{E}^3 - a\mathcal{E}^2 + d, \tag{5.8}$$

paralleling (5.6), where both a and d are positive. This polynomial has extrema at $\mathcal{E} = 0$ and $\mathcal{E} = 2a/3$. By definition (5.5), physically relevant situations must satisfy $\mathcal{E} > 0$. Accordingly, the negative root for $f(\mathcal{E})$ is physically irrelevant and we must focus on finding the appropriate positive-valued root. When we evaluate this polynomial at its other (positive) extremum, i.e., $f(2a/3)$, we observe that f must be nonpositive in order for there to exist at least one positive root. After some algebra, we find that a *sufficient* condition for the existence of a positive root is that

$$d \leq \frac{4}{27}a^3. \tag{5.9}$$

In numerical implementations, this inequality must be checked—if it is not satisfied, the initial value of P must be increased accordingly. (In our experience, this never became an issue.) For convenience—and for algebraic reasons that will become apparent later—we now define ϕ, with $0 \leq \phi \leq \pi/2$, such that

$$d \equiv \frac{4}{27}a^3 \cos^2(\phi). \tag{5.10}$$

It is now easy to verify that the three roots, which we will designate by $\mathcal{E}^{(\ell)}$, for $\ell = 1$, 2, and 3, of (5.8) can be expressed simply as

$$\mathcal{E}^{(\ell)} = \frac{1}{3}a - \frac{2}{3}a\cos\left(\frac{2}{3}\phi + \frac{2}{3}\ell\pi\right), \tag{5.11}$$

providing a hint as to why we employed the relation (5.10). It is easy to show that selecting $\ell = 1$ corresponds to the largest positive root, $\ell = 2$ corresponds to the smaller positive root, and $\ell = 3$ corresponds to the negative root. In the limit that $\phi \to \pi/2$, corresponding to $d = 0$, the smaller positive root and the negative root coalesce at $\mathcal{E} = 0$ while the largest positive root goes to $\mathcal{E} = a$. In this limit, it is evident that the appropriate root corresponds to $\ell = 1$. For larger values of d and, hence, smaller values of ϕ, it follows that this is also the appropriate branch cut to take for the cube root of (5.8). In the limit that $\phi \to 0$, corresponding to a satisfying the equality in (5.9), the two positive roots coalesce at $\mathcal{E} = 2a/3$—as expected from our discussion regarding the location of the extrema and the existence of a positive-valued root—while the negative root goes to $\mathcal{E} = -a/3$. The behavior of the roots as a function of ϕ, scaled to units of a, is shown in Figure 1.

The discussion above provides an algorithm for unambiguously obtaining the solution for $\mathcal{E}$. In particular, given d and our estimate of a, we find $0 \leq \phi \leq \pi/2$ which satisfies (5.10), and then we calculate $\mathcal{E}^{(1)}$ according to (5.11). Then, we calculate $\mathcal{L} = w\left(u^0\right)^2$ using (5.5). Equation (5.2) then yields β^2, while the Lorentz factor u^0 emerges from (3.1). Once $\left(u^0\right)^2$ is known, we then obtain the enthalpy w from (5.1). The last component of the conserved vector $\vec{\mathcal{Q}}$ corresponds to $\tilde{\rho} \equiv \rho u^0$ of (4.5), which

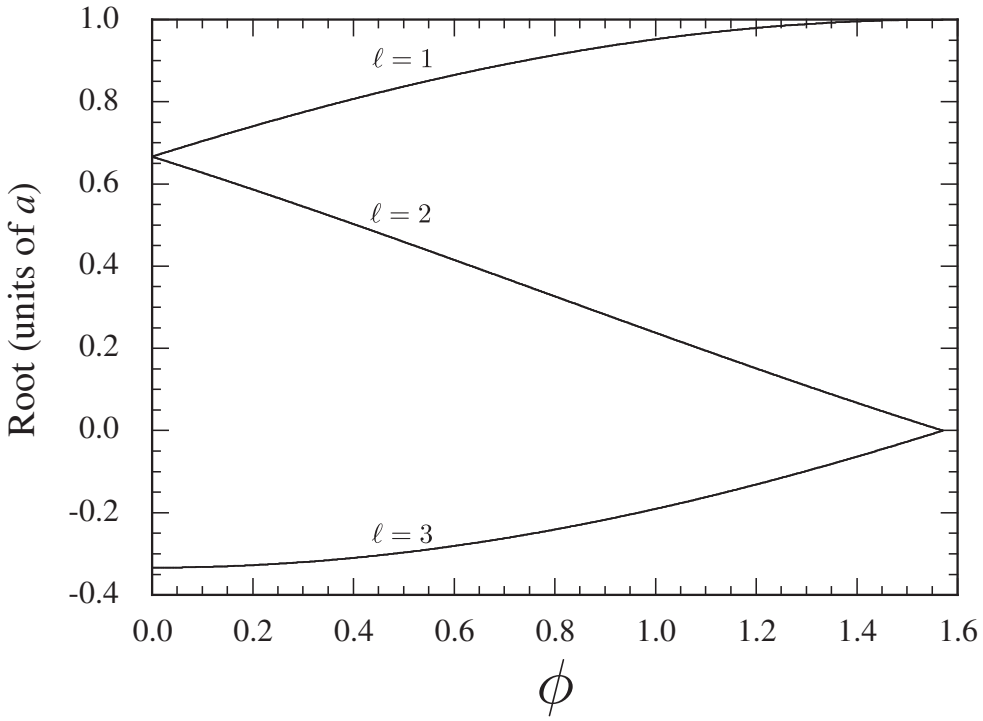

FIG. 1. *Roots of the cubic* (5.8) *as a function of* ϕ*, defined by* (5.11).

provides the density ρ, and finally the EOS (4.1) yields an improved estimate for the pressure P. We will revisit this procedure in the form of an algorithm after we discuss some of its convergence and uniqueness properties.

For convenience we will refer to this procedure, given an estimate P_n, $n = 0, 1, \ldots$, of obtaining an improved estimate P_{n+1}, as an iteration Π, namely,

$$P_{n+1} = \Pi\,(P_n)\,. \tag{5.12}$$

Here, the functional Π depends implicitly on a set of parameters, namely, e, $\mathcal{B}^2$, $\mathcal{M}^2$, $\mathcal{T}$, and $\tilde{\rho}$. Suppose that $\hat{P}$ is the solution to (5.12). Then, it follows that

$$P_{n+1} - \hat{P} = \Pi(P_n) - \Pi(\hat{P}) \approx \Pi'(\hat{P})[P_n - \hat{P}], \tag{5.13}$$

where Π' designates the derivative of the functional Π with respect to its argument. It follows, therefore, that convergence of this sequence is guaranteed if

$$R \equiv \left| \frac{P_{n+1} - \hat{P}}{P_n - \hat{P}} \right| \approx \left| \Pi'(\hat{P}) \right| < 1. \tag{5.14}$$

We refer to R as the Lipshitz parameter for this functional iteration; for a review of the properties of functional iteration, see Kincaid and Cheney [23].

6. Convergence properties of the iteration. In order to demonstrate the convergence properties of the iteration, we shall consider first moderately relativistic situations, such as those mentioned earlier, and then ultrarelativistic situations of the sort considered by Komissarov [24]. For moderate situations, we will consider one

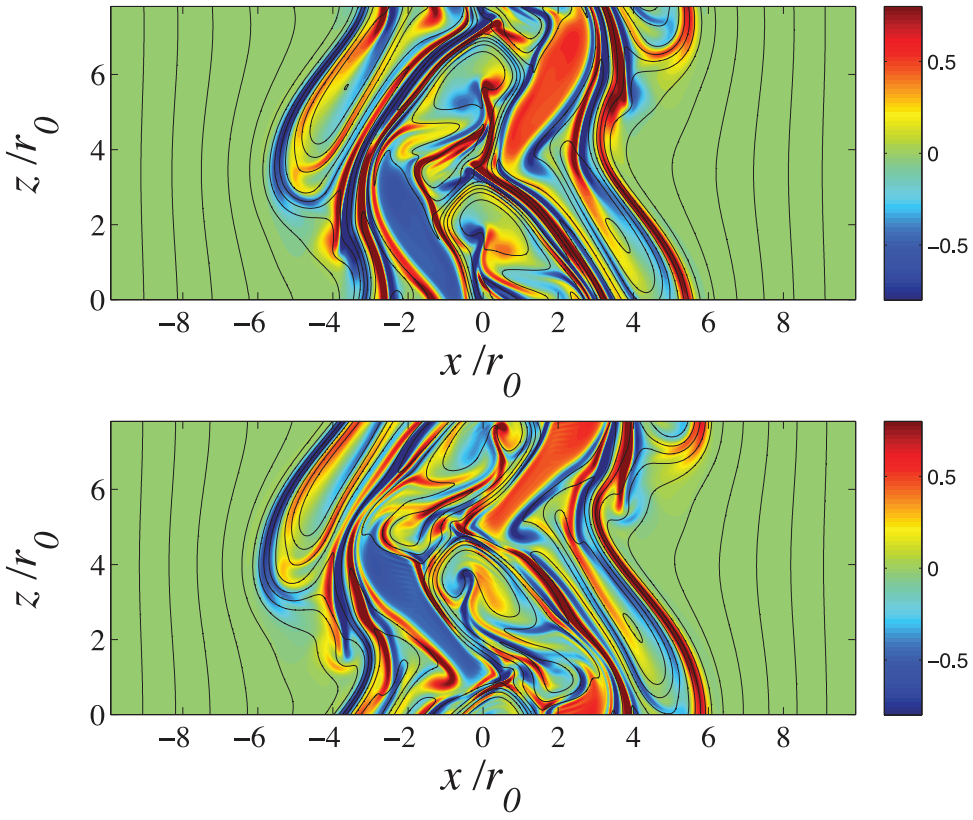

FIG. 2. *Vorticity plot showing magnetic field lines for two representative time steps in* [16]. *For simulations of this sort, our algorithm was called* $\approx 2 \times 10^{10}$ *times with no problems evident.*

of the many cases presented by Hamlin and Newman [16] for two-dimensional MHD flow with Lorentz factors of around 2. We employed the algorithm detailed here in all those cases but also performed comparison checks using several existing algorithms, including the algorithm developed by Mignone and McKinney [28]. These comparisons are discussed in more detail in section 9. Our algorithm invariably worked, while that developed by Mignone and McKinney [28] worked only when provided a sufficiently large initial estimate relative to the correct value of the unknown variable. A typical simulation of this sort required $\approx 2 \times 10^{10}$ applications of our algorithm, and it executed with no evident problems. In order to illustrate this problem, we present in Figure 2 a contour plot describing the vorticity of the flow where the color coding corresponds directly to the fluid vorticity. Details regarding the initial conditions, etc., for this simulation are given in Hamlin and Newman [16, section VIII A] (with an Alfvén Mach number $M_A = 41.4$) and in Hamlin's [15] dissertation (p. 362 as well as Table 1.2 on p. 3). Meanwhile, we also show the magnetic field that is providing some stabilization to the flow, albeit far from perfect. The solid lines correspond to the magnetic field and, as we readily observe, not only is the flow remarkably complicated, but the magnetic field lines are undergoing reconnection. The limiting factor in our simulation was not the conversion algorithm presented here but the need for accurate shock-capturing, an issue that demonstrated the limitations of our fourth-order in

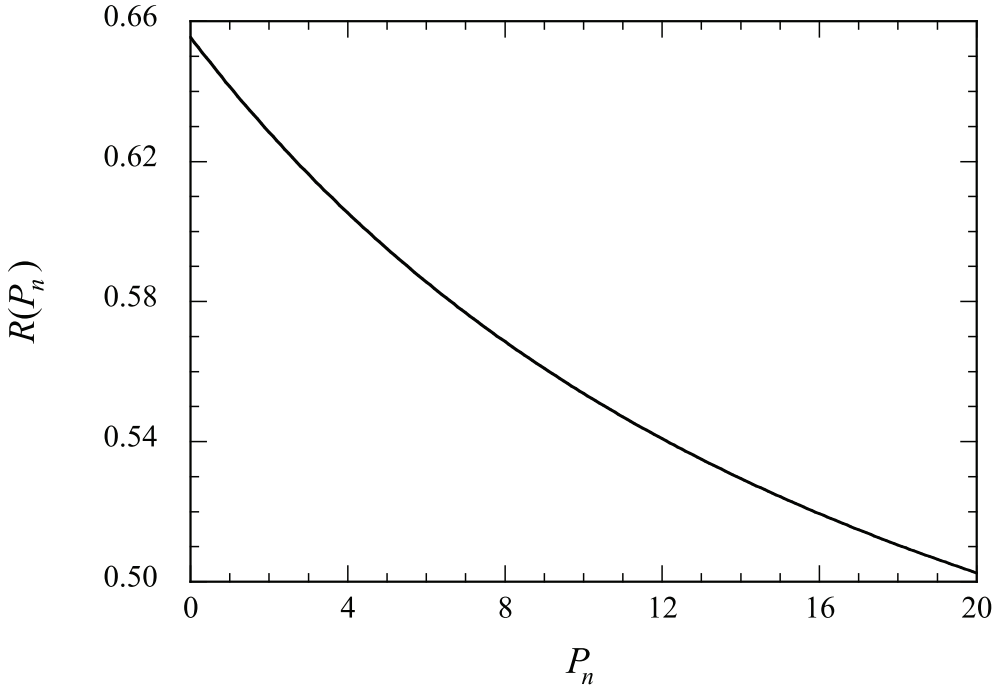

FIG. 3. *Estimate of the Lipshitz parameter as a function of P_n (in units of $\rho_0 c^2$, where ρ_0 is the unperturbed ambient density) for a representative moderately relativistic situation corresponding to the previous figure. Note that the Lipshitz parameter is very nearly constant, which renders our numerical method a good candidate for acceleration schemes.*

space in time method of lines algorithm [21, 34]. Ideally, a method such as those in [30, 27, 10] that employs shock-capturing would better address this issue.

While our algorithm provides a unique estimate of P_{n+1} given P_n, thanks to our ability to make an *ab initio* branch cut, the expression is algebraically complicated and defies the presentation of a simple formula. Nevertheless, using the algebraic manipulation software package `Maple`, we were able to explicitly compute the Lipshitz parameter $R(P_n)$, which we display in Figure 3. We observe that the Lipshitz parameter is always positive, is less than 1 in magnitude, and changes relatively little over a wide range of pressure estimates. The near constancy of the Lipshitz parameter ensures the smoothness of the nonlinear equation in the pressure that must be solved and, hence, guarantees that we have found the proper solution. While functional iteration [23] is a familiar tool employed by numerical analysts, it is less familiar to users such as astrophysicists and plasma physicists, and Figure 3 has been presented especially for their benefit. In situations where the Lipshitz parameter is constant, acceleration schemes as discussed below provide the exact result. Accordingly, we expect that our algorithm is a good candidate for acceleration schemes. With this insight, we wish to understand how the iteration, as proposed above, will proceed.

Figure 4 presents the outcome of the iteration. In this figure, the solid diagonal line shows the new pressure estimate as a function of the old pressure estimate, i.e., (5.12). Since the Lipshitz parameter is always less than 1 and is slowly varying, it appears to be a nearly straight line with a slope less than 1. The dashed line is

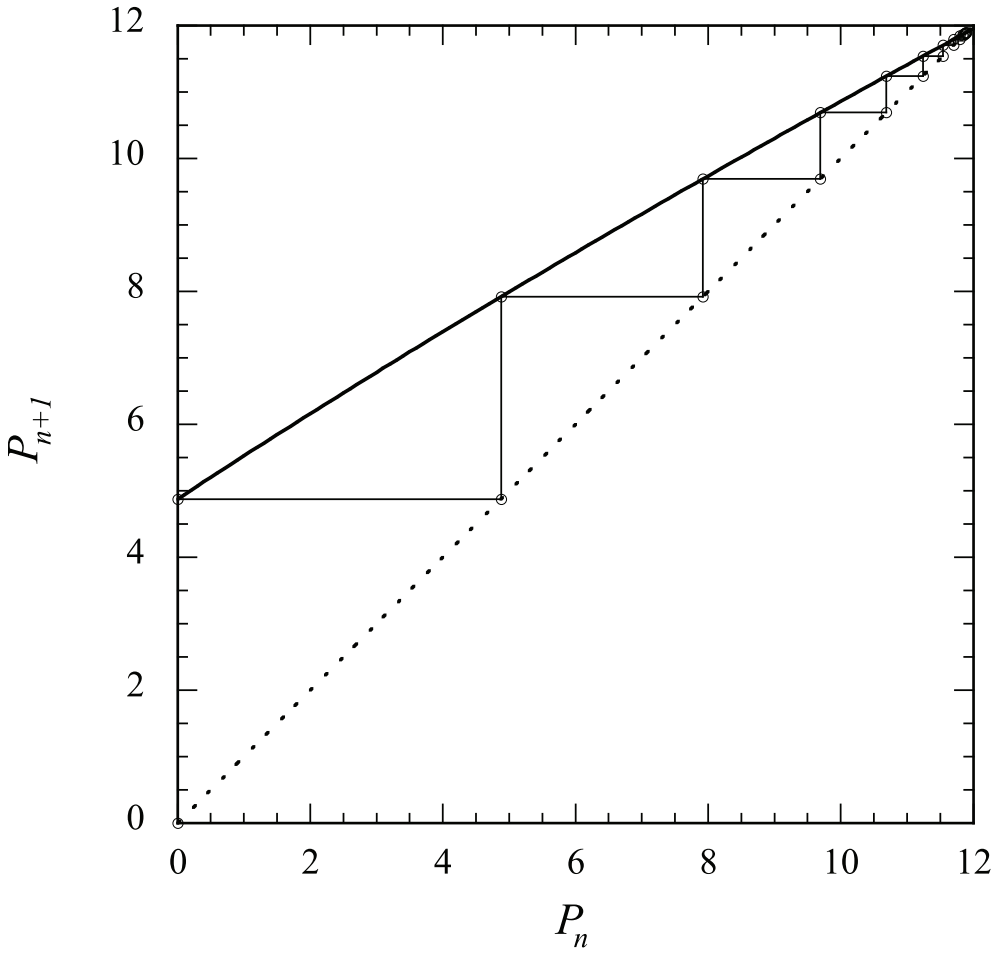

FIG. 4. *Representative sequence of estimates for the pressure P_n (in units of $\rho_0 c^2$) en route to convergence. The staircase evolution to convergence is a hallmark of the Lipshitz parameter being less than* 1.

simply the curve $P_{n+1} = P_n$, which has a slope of 1. Hence, the two curves, denoted by solid and dashed lines, respectively, intersect at only one point, establishing the uniqueness of our solution. The light line with open circles moves between the two curves—i.e., the solid diagonal line and the dashed line—and displays the iteration beginning from zero, i.e., $P_0 = 0$ on the bottom axis to $P_1 \approx 5$ on the vertical axis. The emergent light horizontal line depicts replacing $n = 0$ by $n = 1$, i.e., we now employ $P_1 \approx 5$ to estimate $P_2 \approx 8$, and so on. (The sequence of estimates obtained in this case was 0.0000, 4.8782, 7.9219, 9.6941, 10.6908, 11.2415, ..., 11.9016.) There is a clear pattern here with an emergent staircase moving between the solid line and the dashed line, homing in on the fixed point of the iteration. This staircase evolution is a hallmark of the positive Lipshitz parameter being less than 1, and remaining relatively unchanged, which guarantees a path toward geometric convergence. Our experience has been that during the evolution of the flow, the algorithm functioned well and without incident so long as the time step was chosen to be consistent with the Courant–Friedrichs–Lewy condition.

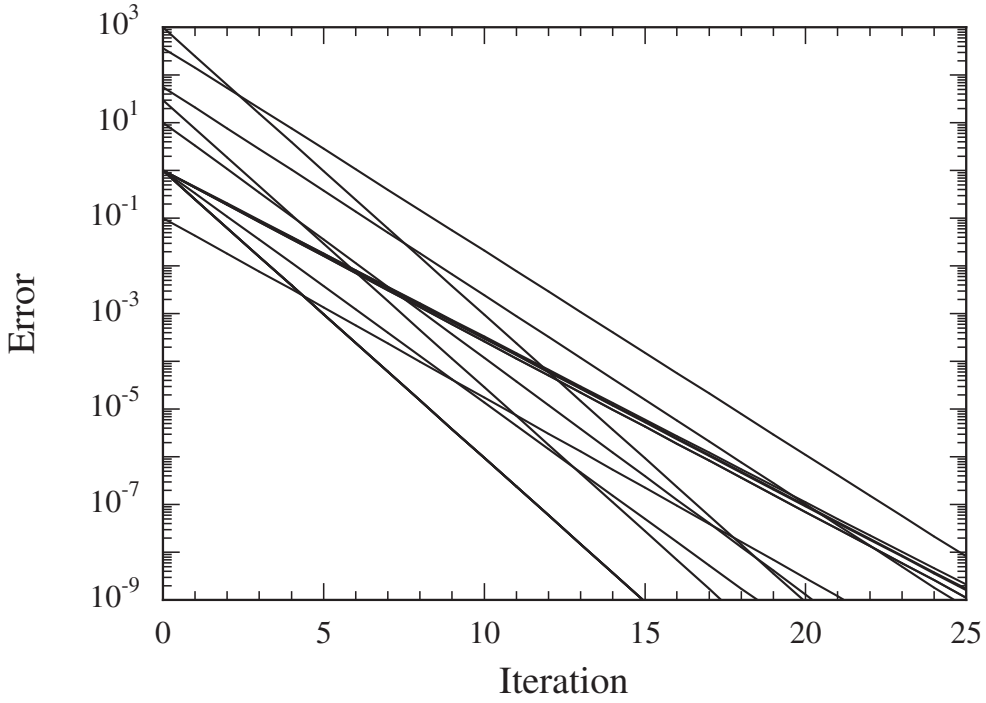

FIG. 5. *Convergence properties for the absolute error, in units of* $\rho_0 c^2$*, of our iterative technique for Komissarov's* 16 *test cases. Each curve is nearly a straight line, implying that the Lipshitz parameter for each situation remains remarkably unchanged.*

We turn our attention now to ultrarelativistic cases. Komissarov [24] included a comprehensive set of examples in his paper, including fast and slow shocks, switch-off fast rarefaction and switch-on slow rarefaction, Alfvén and compound waves, shock tubes, and "collisions." He employed eight sets of initial conditions, including both the left- and right-hand sides of the initial disturbance. We employ here these 16 cases as a test of our iterative procedure. We display results for the primitive variable recovery algorithm only for the initial conditions, although our algorithm worked well for the time evolution. As a worst-case scenario, we assume that we have no a priori knowledge or estimate for P and simply assume that $P_0 = 0$. (In practical cases, we can expect to have good estimates of the pressure at each point and time in a simulation.) Figure 5 shows, for each of these 16 cases, the convergence properties of the iteration. The identification of the different curves with the corresponding cases is immaterial—what matters here is that the convergence is robust, rapid, and remarkably consistent. Importantly, each curve is nearly a straight line, implying that the Lipshitz parameter for each situation remains remarkably unchanged and is related to the nearly constant slope, i.e., the convergence rate. By definition, the Lipshitz parameter describes the reduction in absolute error from one iteration to the next; on a semilogarithmic plot, the slope is approximately the logarithm of the Lipshitz parameter. The Lipshitz parameters are seen to vary from 0.25 to 0.45, values that are smaller than what we observed for mildly relativistic flows. We originally did not expect that ultrarelativistic cases would be better behaved than nominally moderately relativistic ones. Since these examples come from a wide array of physical situations, including strong shocks in extreme relativistic environments, we are optimistic that

this procedure will be reliable and efficient in treating extreme RMHD flows using shock-capturing methods. Given the remarkable constancy over each calculation of the Lipshitz parameter, we sought an acceleration scheme which could exploit this feature of the computation.

7. Aitken acceleration scheme. We chose the Aitken acceleration procedure because it renders the convergence of our recovery algorithm quadratic. We begin by presenting a brief survey of the literature in which this property of the Aitken accelerant is proved. Newton's method, the foundation underlying most other conversion schemes, is well-known to be quadratically convergent [23, 33, 22, 18]. Aitken's δ^2 acceleration method is commonly presented in standard numerical analysis textbooks; however, its convergence properties are rarely described. Isaacson and Keller [22] provide a comprehensive treatment of its convergence properties, as do more recent texts such as those by Kincaid and Cheney [23] and Dahlquist and Björck [8]. Henrici [18] provides a derivation of its properties, including an explicit demonstration of its quadratic convergence via an error expression that, interestingly, parallels that presented elsewhere for Newton's method. Henrici notes that "It is frequently... described by stating that the number of correct decimal places is *doubled* at each step," a feature that we observed in practice in the application of our algorithm. All authors cited above who addressed the Aitken method also noted the importance of using carefully chosen forms for the Aitken iteration to avoid deleterious roundoff effects. Higham [20], in his treatise on the accuracy and stability of numerical algorithms, leaves this as an exercise for the reader.

In our method, the evident geometric convergence of the sequence of estimates for the pressure P_n, $n = 0, 1, \ldots$, enables us to exploit the Aitken acceleration procedure, which, as discussed above, effectively renders convergence quadratic. In particular, given three such estimates for the pressure, namely, our initial guess P_0 and two iterations of the above procedure, we compute

$$\mathcal{R} \equiv \frac{P_2 - P_1}{P_1 - P_0}, \tag{7.1}$$

$$P_{\text{Aitken}} \equiv \frac{P_0 P_2 - P_1^2}{P_2 - 2P_1 + P_0}$$

as an improved estimate to P_2 and where $\mathcal{R}$ is an empirical estimate of the Lipshitz parameter or convergence rate R. In practice, we employ a slightly different, albeit equivalent, formula to mitigate arithmetic rounding issues, namely,

$$P_{\text{Aitken}} \equiv P_1 + \frac{P_2 - P_1}{1 - \mathcal{R}}. \tag{7.2}$$

If sufficient accuracy is not present, we employ P_{Aitken} as though it were our initial estimate P_0. With two more iterations of the preceding algorithm, this results in a much improved estimate of the pressure. Importantly, this procedure should be terminated soon after it produces apparently stationary values for the P_i since it will become susceptible to roundoff errors. In order to obtain 10 significant figure results, we used the Aitken procedure two or three times per grid point per time step, as detailed below.

We employ Komissarov's testbed of examples to illustrate the efficacy of this procedure. Table 1 shows the results obtained for the different cases. (In several instances, the lines of simulation results obtained were identical and are not repeated.

TABLE 1
Iterative estimates of pressure, including Aitken acceleration estimates, for Komissarov's test cases.

P_{true}	P_1	P_2	P_{Aitken}	P_1'	P_2'	P'_{Aitken}	$\mathcal{R}$
1.00	0.557	0.802	0.995	0.998	0.999	1.000	0.450
367.50	224.728	313.392	371.168	368.870	368.012	367.500	0.374
1.00	0.667	0.891	1.004	1.001	1.000	1.000	0.325
55.36	34.482	47.654	55.796	55.519	55.418	55.360	0.365
1.00	0.546	0.796	1.008	1.004	1.002	1.000	0.445
10.00	6.658	8.918	10.079	10.025	10.008	10.000	0.319
0.10	0.057	0.082	0.101	0.100	0.100	0.100	0.419
1.00	0.750	0.938	1.000				0.250
1.00	0.560	0.805	0.995	0.998	0.999	1.000	0.445
1000.00	750.000	937.500	1000.000				0.200
30.00	22.500	28.125	30.000				0.250
1.00	0.565	0.810	0.996	0.998	0.999	1.000	0.439

Also, some cells in the table are not occupied since the combination of our iterative procedure with the Aitken acceleration had yielded machine-accurate estimates.) The first column (P_{true}) provides the exact solution for the pressure. In this case, we assumed no prior knowledge of the pressure and took $P_0 \equiv 0$. The second column provides our first iteration or P_1 for each case, while the third column provides the second iteration or P_2. In the fourth column, we display P_{Aitken} as given in (7.2). We now employ P_{Aitken} as if it were a new P_0, so the fifth and sixth columns are new estimates of the pressure obtained from the iteration, denoted as P_1' and P_2', respectively, from which we derive a second Aitken estimate of P'_{Aitken}. Thus, we observe that with no prior information we are able to obtain at least six significant digits of accuracy with only four iterations of our new algorithm. Further refinement in most of these cases from an additional two iterations would have brought our solutions to nearly machine precision accuracy. (In practical terms, we terminated the iterations when no further improvement in accuracy was possible, owing to roundoff effects.) Indeed, an issue that must now be confronted is that of incorporating checks in our computations for convergence and the influence of roundoff error. We note also that the estimated Lipshitz parameter remains unchanged to three decimal places.

8. Algorithm for determining primitive variables. We now conclude this treatment by amalgamating our iterative procedure with the Aitken acceleration in the form of an algorithm.

1. Collect e and $\tilde{\rho}$ and compute $\mathcal{M}^2$, $\mathcal{B}^2$, and $\mathcal{T}$. Construct an estimate for P, e.g., $P = 0$, and compute
$$a \leftarrow e + P + \mathcal{B}^2/2,$$
$$d \leftarrow \frac{1}{2}\left(\mathcal{M}^2\mathcal{B}^2 - \mathcal{T}^2\right),$$
making sure that P is selected so that $a = e + P + \mathcal{B}^2/2 \geq (27d/4)^{1/3}$.
2. Compute
$$\phi \leftarrow \cos^{-1}\left[\frac{1}{a}\sqrt{\frac{27d}{4a}}\right],$$
$$\mathcal{E}_1 \leftarrow \frac{a}{3} - \frac{2a}{3}\cos\left(\frac{2\phi}{3} + \frac{2\pi}{3}\right),$$

$$\begin{aligned}
\mathcal{L} &\leftarrow \mathcal{E}_1 - \mathcal{B}^2, \\
\beta^2 &\leftarrow \frac{\mathcal{M}^2\mathcal{L}^2 + \mathcal{T}^2\left(\mathcal{B}^2 + 2\mathcal{L}\right)}{\mathcal{L}^2\left(\mathcal{B}^2 + \mathcal{L}\right)^2}, \\
u^0 &\leftarrow \frac{1}{\sqrt{1-\beta^2}}, \\
w &\leftarrow \frac{\mathcal{L}}{\left(u^0\right)^2}, \\
\rho &\leftarrow \frac{\tilde{\rho}}{u^0}, \\
P &\leftarrow \frac{\gamma - 1}{\gamma}\left(w - \rho\right).
\end{aligned}$$

3. Perform a convergence check on P, i.e., check magnitude of difference between two successive estimates $|P_1 - P_0| < \epsilon$ for ϵ suitably small. Because of rounding considerations, ϵ must be very small, typically $< 10^{-12}$. If the convergence criterion is satisfied, then proceed to step 4; otherwise, repeat steps 1 and 2.
4. If three successive estimates of P are known—say, P_0, P_1, and P_2—then compute the approximate Lipshitz parameter $\mathcal{R}$ and the Aitken accelerant $P_{\rm Aitken}$, namely,

$$\begin{aligned}
\mathcal{R} &\leftarrow \frac{P_2 - P_1}{P_1 - P_0}, \\
P_{\rm Aitken} &\leftarrow P_1 + \frac{P_2 - P_1}{1 - \mathcal{R}}.
\end{aligned}$$

   Make certain that the sequence is converging, i.e., $|\mathcal{R}| < 1$. Once again, perform a convergence check on the P estimates, including $P_{\rm Aitken}$. If the convergence checks are satisfied, then proceed to step 5; otherwise, repeat steps 1 and 2.
5. Now that the algorithm has converged, the quantities P, ρ, w, β^2, u^0, and $\vec{B}$ are known—it may be necessary to repeat step 2 if the final estimate for P was obtained from the Aitken acceleration. To complete the calculation of *all* primitive variables, the following additional quantities must be computed for $i = 1, 2, 3$:

$$\begin{aligned}
\mathcal{S} &\leftarrow \frac{\mathcal{T}}{w\left(u^0\right)^2}, \\
\beta^i &\leftarrow \frac{m^i + \mathcal{S}B^i}{w\left(u^0\right)^2 + \mathcal{B}^2}, \\
u^i &\leftarrow \beta^i u^0, \\
b^0 &\leftarrow \mathcal{S}u^0, \\
b^i &\leftarrow \frac{B^i + b^0 u^i}{u^0}.
\end{aligned}$$

   With this last step, all primitive variables—including both three- and four-vector versions of velocity and magnetic field—have been determined.

These latter examples give strong support for the essential features of this algorithm, and our experience with this scheme in treating an array of ultrarelativistic and moderately relativistic astronomical problems has been problem-free. Indeed, we found that the speed of our computations was only modestly reduced from those of classical MHD and that the conversion of conserved quantities to primitive variables was not a significant issue.

9. Use of algorithm and comparison with existing schemes. Having presented our primitive variable recovery algorithm in detail, we now discuss a comparison with several existing primitive variable recovery schemes. We then discuss the future implementation of our method with shock-capturing schemes in order to model highly relativistic MHD phenomena.

We have compared our method to those described in section 2 for the moderately relativistic case discussed in section 6, namely, the two-dimensional nonlinear RMHD simulation of the Kelvin–Helmholtz instability from Hamlin and Newman [16]. These are the 5×5 method used by Balsara [3], the 3×3 method used by Komissarov [24], the method of Del Zanna, Bucciantini, and Londrillo [10], and the method of Mignone and McKinney [28]. We began by reading in a set of data at a given time step in the early nonlinear evolution (the same data for each test). In the case of Komissarov [24], we were unable to provide a sufficiently close initial guess for his algorithm to advance through a time step, as his algorithm fails at every point on the grid which is not close enough to this guess. The other three methods require less accurate initial estimates, and we were able to evolve each of these using any time step that satisfies the Courant–Friedrichs–Lewy stability criteria. After advancing through the first time step, at each spatial grid point we used the value from the previous time step as the new estimate. We achieved approximate machine accuracy with each algorithm over the spatial grid.

The main differences in algorithm performance were in the length of time each algorithm required in order to advance through 20 time steps. The time measurement here refers to the total clock time taken by the primitive variable recovery algorithm, not the total execution time of the code. Ours took 65.0 sec, the 5×5 algorithm used by Balsara [3] took 104.2 sec (60% longer than ours), Del Zanna's algorithm took 72.1 sec (11% longer than ours), and the Mignone–McKinney algorithm took 41.9 sec (35% shorter than ours). This test shows that even when the initial guess is sufficiently close to the correct value, the algorithms used by Balsara and by Del Zanna do not have the computational efficiency of our method.

The algorithm of Mignone and McKinney [28] performed about 35% more rapidly than ours in this moderately relativistic test case, namely, the simulation from Hamlin and Newman [16], because here the estimate from the previous time step is sufficiently close to the correct value and because we provided a sufficiently accurate initial estimate at the start of the simulation. However, we have found that their algorithm fails to converge to a solution whenever the initial guess underestimates the correct value by more than about 15%. We also tested the Mignone–McKinney algorithm in recovering the left and right states of several extreme relativistic test cases from Komissarov [24]. Recall from section 6 that our algorithm recovers machine-accurate solutions for these cases for any nonnegative initial pressure estimate. From Komissarov's testbed of examples, we identified circumstances where the Mignone and McKinney algorithm failed when the initial estimate was either below or more than 10% above the true value. These results indicate that in an extreme relativistic MHD simulation, their algorithm would fail at a significant percentage of grid points. In summary, while the

algorithm of Mignone and McKinney has a modest speed advantage in moderately relativistic cases, it sometimes encounters convergence problems in extreme relativistic MHD cases. Our algorithm overcomes these difficulties, because it recovers the correct solution for each of Komissarov's cases for *any nonnegative initial estimate of the pressure*. Only in extreme RMHD environments did we encounter any significant issues, and these appeared to be the outcome of rounding effects and the sensitivity of the Lorentz factor to the velocity. Therefore, when implemented in conjunction with shock-capturing schemes, we expect that our algorithm will improve the modeling of highly relativistic MHD phenomena, as we now discuss.

As a direction of future research, we expect that our scheme will be a tremendous asset to the latest generation of high-resolution shock-capturing (HRSC) schemes used in RMHD modeling. HRSC methods are designed specifically to solve conservative hyperbolic systems and are particularly well-suited to the modeling of flows with high Lorentz factors. Marquina et al. [26] presented HRSC methods for RHD modeling in the ultrarelativistic regime. Donat and Marquina [12] developed a flux formula for capturing shock reflections. Donat et al. [11] adapted this formula to a flux-split algorithm for relativistic flows. Duncan and Hughes [13] applied an HRSC scheme to the modeling of highly relativistic extragalactic jets in RHD.

Komissarov [24] and Balsara [3] made significant advances toward extending HRSC schemes to the simulation of RMHD fluids. However, their methods use Riemann solvers that require complicated diagonalization of the Jacobian relating the flux quantities to the conserved quantities. Their primitive variable recovery algorithms also require the solution of systems of equations. Del Zanna, Bucciantini, and Londrillo [10] applied what is known as the Harten–Lax–van Leer (HLL) scheme, originally presented in [17], to their RMHD modeling via HRSC. The HLL scheme has the advantage of requiring only the estimation of slow and fast wave speeds associated with the flux Jacobian, rather than its diagonalization. Mignone and Bodo [27] introduced an HLLC approximate Riemann solver that improves upon the HLL scheme by also resolving an intermediate wave associated with the contact discontinuity. Mignone, Ugliano, and Bodo [30] further improved upon their scheme with a so-called HLLD scheme that resolves still other characteristics in the RMHD flux Jacobian. Zrake and MacFadyen [36] used the HLLD Riemann solver in their simulations of RMHD turbulence. These RMHD studies only considered moderate Lorentz factors and used primitive variable recovery algorithms similar to that of Mignone and McKinney [28], which requires a reasonably accurate initial guess of the unknown variable.

Our primitive variable recovery algorithm, on the other hand, converges rapidly and accurately *for any initial estimate*. We therefore believe that our algorithm, used in conjunction with a recent HRSC scheme such as that of Mignone, Ugliano, and Bodo [30], will enable RMHD modeling at much higher Lorentz factors. One such application is the RMHD modeling of strong highly relativistic shocks, for example, in a supernova or gamma-ray burst, in which the Lorentz factor can increase by orders of magnitude over a few time steps at a given spatial point. These are points at which existing recovery algorithms tend to fail, mainly because they require an accurate initial estimate, usually the value from the previous time step. However, this value generally does *not* provide an accurate initial estimate when the Lorentz factor changes substantially from one time step to the next, particularly if the unknown variable depends on the Lorentz factor or Lorentz factor squared, as is the case with most existing recovery schemes, including that of Mignone and McKinney [28]. A problem of this type demands a recovery algorithm that converges rapidly

and accurately *without requiring an accurate initial guess*, which is precisely what our method offers.

Our algorithm can also be readily adapted to GRMHD simulations. The conserved energy and momentum have general relativistic corrections associated with the use of a curved rather than flat space-time metric [14]. However, our recovery algorithm makes use of the momentum-squared rather than the individual momentum components. As discussed on p. 27 of Font's [14] paper, when the system to be solved uses the momentum-squared, the general relativistic primitive variable recovery problem is fundamentally unchanged from the special relativistic problem.

10. Conclusions. Extreme relativistic environments continue to challenge investigators employing conservative integration schemes to numerically solve the RMHD equations. A major computational obstacle common to *all* such schemes is the conversion of the conserved quantities computed by these methods to the primitive variables required both for the ongoing simulation and physical interpretation of astrophysical models. In order to work well with large time steps and parallel computing architectures often required by the computational demands of an RMHD simulation, a primitive variable recovery scheme must be absolutely and rapidly convergent at all grid points in a simulation.

We have developed such a scheme by reducing the problem associated with solving five coupled nonlinear equations—or three using Komissarov's [24] method—to a single problem corresponding to the intersection of the two curves in Figure 4 utilizing a single equation in a single unknown. We then showed that this leads to a natural and well-behaved iterative procedure. Moreover, we derived its (approximate) convergence properties and adapted the Aitken acceleration scheme to render quadratic the speed of convergence. Existing primitive variable recovery schemes require an accurate initial estimate of the unknown variable, whereas our scheme has rapid and absolute convergence *for any nonnegative initial guess of the unknown pressure*. This enables our scheme to converge at every grid point in a simulation without the need to reduce the time step below physically motivated constraints and makes our scheme easily adaptable to massively parallel computer architectures.

We have found that our scheme works without fail in both the highly relativistic and moderately relativistic regimes. In the highly relativistic regime, we tested our scheme with Komissarov's testbed of examples. In the moderately relativistic regime, we used an RMHD simulation from Hamlin and Newman [16]. We found that our method worked without fail, and when we compared our method to existing methods, we found that only the method of Mignone and McKinney [28] worked, when applied with due care, at all grid points. In such cases, their method was about 35% faster than ours. Their method, however, worked only when provided an initial guess that was comparable to or greater than the actual solution, and did not converge when the initial guess underestimated the solution by more than about 15%. Our algorithm worked for *any* initial estimate. For this reason, we believe that when used in conjunction with HRSC schemes, our algorithm will enable the modeling of highly relativistic MHD phenomena such as the shocks in a supernova or gamma-ray burst.

There are many phenomena in astrophysics and laboratory plasma physics that motivate the use of relativistic MHD modeling, which has thus far been hampered by the immense technical challenges associated with recovering the primitive variables from the conserved quantities. We have developed an algorithm that overcomes these challenges in both moderately relativistic and highly relativistic regimes and which can be used in conjunction with the Aitken accelerant to achieve quadratic convergence.

Although the expressions involved in our method are too complicated to provide algebraic proofs of convergence, we have demonstrated the convergence properties quantitatively. Having overcome the technical difficulties that have long plagued primitive variable recovery in RMHD simulations, we look forward to employing our algorithm in relativistic environments that have thus far been prohibitively difficult to explore.

Acknowledgments. We wish to thank Philip Sharp, Richard Lovelace, Andrea Mignone, anonymous referees, and especially James M. Hyman for helpful comments.

REFERENCES

[1] E. Amato, L. del Zanna, and N. Bucciantini, *2D Relativistic MHD Simulations of PWNe: Preliminary Results*, in Young Neutron Stars and Their Environments, F. Camilo and B. M. Gaensler, eds., IAU Symposium 218, International Astronomical Union, Paris, 2004.

[2] A. M. Anile, *Relativistic Fluids and Magneto-fluids: With Applications in Astrophysics and Plasma Physics*, Cambridge University Press, Cambridge, UK, 1989.

[3] D. Balsara, *Total variation diminishing scheme for relativistic magnetohydrodynamics*, Astrophys. J. Supp., 132 (2001), pp. 83–101.

[4] N. Bucciantini, E. Amato, and L. Del Zanna, *Relativistic MHD simulations of pulsar bow-shock nebulae*, Astronom. Astrophys., 434 (2005), pp. 189–199.

[5] N. Bucciantini and L. D. Zanna, *Local Kelvin-Helmholtz instability and synchrotron modulation in Pulsar Wind Nebulae*, Astronom. Astrophys., 454 (2006), pp. 393–400.

[6] N. Bucciantini, T. A. Thompson, J. Arons, E. Quataert, and L. D. Zanna, *Relativistic magnetohydrodynamics winds from rotating neutron stars*, Monogr. Not. R. Astronom. Soc., 368 (2006), pp. 1717–1734.

[7] N. Bucciantini, T. A. Thompson, J. Arons, E. Quataert, and L. D. Zanna *Relativistic MHD winds from rotating neutron stars*, Adv. Space Research, 40 (2007), pp. 1441–1445.

[8] G. Dahlquist and Å. Björck, *Numerical Methods in Scientific Computing*, SIAM, Philadelphia, 2008.

[9] W. Dai and P. R. Woodward, *A high-order godunov-type scheme for shock interactions in ideal magnetohydrodynamics*, SIAM J. Sci. Comput., 18 (1997), pp. 957–981.

[10] L. Del Zanna, N. Bucciantini, and P. Londrillo, *An efficient shock-capturing central-type scheme for multidimensional relativistic flows.* II. *Magnetohydrodynamics*, Astronom. Astrophys., 400 (2003), pp. 397–413.

[11] R. Donat, J. A. Font, J. M. Ibánez, and A. Marquina, *A flux-split algorithm applied to relativistic flows*, J. Comput. Phys., 146 (1998), pp. 58–81.

[12] R. Donat and A. Marquina, *Capturing shock reflections: an improved ux formula*, J. Comput. Phys., 125 (1996), pp. 42–58.

[13] G. C. Duncan and P. A. Hughes, *Simulations of relativistic extragalactic jets*, Astrophys. J. Lett., 436 (1994), pp. L119–L122.

[14] J. A. Font, *Numerical hydrodynamics and magnetohydrodynamics in general relativity*, Living Rev. Relativity, 11 (2008).

[15] N. D. Hamlin, *The Role of the Kelvin-Helmholtz Instability in the Evolution of Magnetized Relativistic Sheared Plasma Flows*, Ph.D. dissertation, Department of Physics and Astronomy, University of California, Los Angeles, 2012; also available online from http://gradworks.umi.com/35/26/3526706.html.

[16] N. D. Hamlin and W. I. Newman, *Role of the Kelvin-Helmholtz instability in the evolution of magnetized relativistic sheared plasma flows*, Phys. Rev. E, 87 (2013), 043101.

[17] A. Harten, P. D. Lax, and B. van Leer, *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*, SIAM Rev., 25 (1983), pp. 35–61.

[18] P. Henrici, *Elements of Numerical Analysis*, Wiley, New York, 1964.

[19] P. Henrici, *Applied and computational complex analysis*, Pure and Appl. Math., Wiley, New York, 1986.

[20] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[21] J. M. Hyman, *A method of lines approach to the numerical solution of conservation laws*, in Advances in Computer Methods for Partial Differential Equations III, 1979, pp. 313–321.

[22] E. Isaacson and H. B. Keller, *Analysis of Numerical Methods*, Wiley, New York, 1966.

[23] D. Kincaid and E. W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, 2nd ed., Brooks/Cole, Pacific Grove, CA, 1996.
[24] S. S. Komissarov, *A Godunov-type scheme for relativistic magnetohydrodynamics*, Monogr. Not. R. Astronom. Soc., 303 (1999), pp. 343–366.
[25] P. D. Lax, *Weak solutions of nonlinear hyperbolic equations and their numerical computation*, Comm. Pure Appl. Math., 7 (1954), pp. 159–193.
[26] A. Marquina, J. M. Marti, J. M. Ibanez, J. A. Miralles, and R. Donat, *Ultrarelativistic hydrodynamics—High-resolution shock-capturing methods*, Astronom. Astrophys., 258 (1992), pp. 566–571.
[27] A. Mignone and G. Bodo, *An HLLC Riemann solver for relativistic flows* II. *Magnetohydrodynamics*, Monogr. Not. R. Astronom. Soc., 368 (2006), pp. 1040–1054.
[28] A. Mignone and J. C. McKinney, *Equation of state in relativistic magnetohydrodynamics: variable versus constant adiabatic index*, Monogr. Not. R. Astronom. Soc., 378 (2007), pp. 1118–1130.
[29] A. Mignone, P. Rossi, G. Bodo, A. Ferrari, and S. Massaglia, *High-resolution* 3*D relativistic MHD simulations of jets*, Monogr. Not. R. Astronom. Soc., 402 (2010), pp. 7–12.
[30] A. Mignone, M. Ugliano, and G. Bodo, *A five-wave Harten-Lax-van Leer Riemann solver for relativistic magnetohydrodynamics*, Monogr. Not. R. Astronom. Soc., 393 (2009), pp. 1141–1156.
[31] W. I. Newman and A. L. Newman, *The interaction of a strong magnetic field with a cold plasma: The evolution of a magnetic bubble*, Astrophys. J., 515 (1999), pp. 685–695.
[32] S. C. Noble, C. F. Gammie, J. C. McKinney, and L. D. Zanna, *Primitive variable solvers for conservative general relativistic magnetohydrodynamics*, Astrophys. J., 641 (2006), pp. 626–637.
[33] A. Ralston and P. Rabinowitz, *A First Course in Numerical Analysis*, 2nd ed., Internat. Ser. Pure and Appl. Math., McGraw-Hill, New York, 1978.
[34] W. E. Schiesser, *The Numerical Method of Lines: Integration of Partial Differential Equations*, Academic Press, New York, 1991.
[35] M. H. P. M. van Putten, *A numerical implementation of MHD in divergence form*, J. Comput. Phys., 105 (1993), pp. 339–353.
[36] J. Zrake and A. I. MacFadyen, *Numerical simulations of driven relativistic magnetohydrodynamic turbulence*, Astrophys. J., 744 (2012), pp. 32–41.