The Astrophysical Journal Supplement Series, 187:119–134, 2010 March doi:10.1088/0067-0049/187/1/119

ON THE RELIABILITY OF *ZEUS-3D*

David A. Clarke
Institute for Computational Astrophysics, Department of Astronomy & Physics, Saint Mary's University, Halifax, Nova Scotia B3H 3C3, Canada;
dclarke@ap.smu.ca

Received 2009 January 23; accepted 2010 January 15; published 2010 February 26

ABSTRACT

Recent and not-so-recent critiques of the widely used magnetohydrodynamics (MHD) code, *ZEUS-3D*, challenge its reliability and efficiency suggesting that its MHD algorithm is capable of "significant errors" in some simple one-dimensional shock-tube problems. I show that these concerns are either inapplicable in multi-dimensional astrophysical applications, or result from a misuse of the code rather than "flaws" in its design. I also describe a few multi-dimensional test problems including one for super-Alfvénic turbulence, and highlight some recent innovations and improvements to the code now available online.

Key words: hydrodynamics – magnetohydrodynamics (MHD) – methods: numerical

1. INTRODUCTION

Reports by Balsara (2001), Falle (2002; hereafter F02) and, more recently, Tasker et al. (2008; hereafter T08) have questioned the reliability of the *ZEUS* family of codes (Stone & Norman 1992a, 1992b, 1992c; Clarke 1996) and, in particular, suggest that its magnetohydrodynamics (MHD) algorithms can result in significant shock and rarefaction errors. This paper seeks to address these concerns. Indeed, as *ZEUS-3D* is probably the most widely used MHD code in astrophysics with hundreds of publications based on its results, this paper is perhaps long overdue.

Part of the problem is that since its first release in 1992, numerous versions of the code have been developed, including Michael Norman's *ZEUSMP* and the *ZEUS* module in *ENZO*,[1] Jim Stone's *ZEUS-2D* and his version of *ZEUS-3D*,[2] and a myriad of other versions modified by users some of which are available online. None of these codes are accountable to another, and it is difficult to determine whether a reported problem is peculiar to a particular version, or general to the underlying staggered-grid approach that defines the *ZEUS* family of codes.

The results presented herein were computed by my own version, `dzeus35`[3], where the "d" indicates "double precision" and the "35" indicates "version 3.5." This is a direct descendant of version 3.2 that Michael Norman and I released 18 years ago and from which he and Robert Fiedler developed *ZEUSMP*, still probably the most widely used member of the *ZEUS* family of codes today. `dzeus35` is a static-grid code (no Adaptive Mesh Refinement; AMR) and parallelized for *OpenMP*, not *MPI* (Message Passing Interface). A version incorporating AMR and *MPI* is currently under development. As a matter of notation, I shall use "*ZEUS*" in discussions about the codes in general, and "`dzeus35`" when I wish to refer to my own brand of the code.

I suggest that "rarefaction shocks" reported in F02 arising in some one-dimensional (1D) Riemann test problems are exceedingly unlikely to arise in actual multi-dimensional astrophysical applications and, even if they did, the physics is left unharmed. The so-called shock errors reported in one dimension by F02 and in three dimensions by T08 stem from comparing the results of a *non-conservative* (of mechanical energy) algorithm with analytical solutions of the *conservative* equations, and nothing to do with the integrity of the underlying (M)HD methods. Indeed, when the conservative algorithm suggested by Clarke (1996) to solve the total energy equation is used, these so-called shock errors disappear. I present conservative `dzeus35` solutions to the full suite of 1D test problems given in Ryu & Jones (1995, hereafter RJ95; some of these problems first appeared in Dai & Woodward 1994); the first time any such *ZEUS* solutions have appeared publicly. For reasons unrelated to published criticism of the code, all versions of *ZEUS* of which I am aware—except `dzeus35`—give an oscillatory solution to two of these 1D test problems, and I outline the simple fix that may be retrofitted to any version of *ZEUS* in current use.

I describe a few selected multi-dimensional test problems including a 2D test designed to demonstrate the utility of the MHD boundary conditions, two of the 2D tests suggested by Gardiner & Stone (2005), the 3D Sedov blast wave, and 3D super-Alfvénic turbulence.

Finally, the claim made by F02 that *ZEUS* is somehow *slower* than a fully upwinded code is also shown to be unfounded, particularly when compared with a Godunov-type code that is as accurate in 3D as it is in 1D (e.g., Stone et al. 2008). Indeed, the advantages of the *ZEUS* algorithm have been and remain its speed, its robustness in multi-dimensions, and its ability to accommodate additional physics such as viscosity, radiation, self-gravity, etc., without adversely affecting its underlying (M)HD algorithm.

2. "RAREFACTION SHOCKS"

The *ZEUS* family of codes is *upwinded* in the entropy and Alfvén waves, and *stabilized* in the compressional (fast and slow) waves, with stabilization accomplished by the use of two flavors of von Neumann–Richtmyer artificial viscosity (von Neumann & Richtmyer 1950). The so-called quadratic term, controlled by the *ZEUS* parameter `qcon`, is applied only to regions of strong compression and effectively captures shocks, typically within `qcon+2` zones. The "linear term," controlled by the *ZEUS* parameter `qlin`, is applied throughout the grid and stabilizes continuous structures such as magnetoacoustical and rarefaction waves. Typical values for `qcon` are 1–2 in all applications, and 0.1–0.2 for `qlin` in 1D shock-tube problems. For multi-dimensional applications, `qlin` is often set to zero relying on grid diffusion for stabilization.

[1] http://lca.ucsd.edu/portal/software/enzo
[2] http://www.astro.princeton.edu/~jstone
[3] http://www.ica.smu.ca/zeus3d

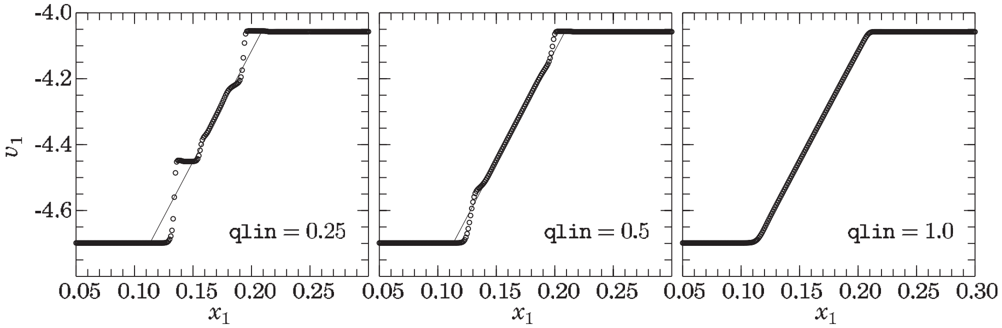

Figure 1. Fast rarefaction at $t = 0.1$ from the Riemann problem with left and right states $(\rho, v_1, v_2, v_3, B_2, B_3, p) = (1, -4.6985, -1.085146, 0, 1.9680, 0, 0.2327)$ and $(0.7270, -4.0577, -0.8349, 0, 1.355, 0, 0.1368)$, $B_1 = -0.7$ and $\gamma = 5/3$. The grid domain of [0, 1] is resolved with 1000 zones, although only the portion of the grid containing the rarefaction (resolved with ∼100 zones) is shown. The variable shown is v_1 for three different values of `qlin`.

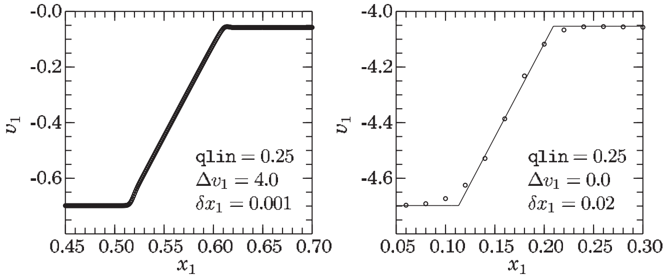

Figure 2. Same as Figure 1 except (left) with a Galilean transformation of $\Delta v_1 = 4.0$ applied, and (right) with the rarefaction resolved with only five zones ($\delta x_1 = 0.02$) and with no Galilean transformation applied. Both are run with `qlin = 0.25`.

Figures 1 and 2 show the fast rarefaction in Figure 2 of F02. The "rarefaction shocks" (the discontinuities in the left panel of Figure 1) are, according to F02, a result of the operator-split momentum equation in *ZEUS*, resulting in second-order accuracy in space but only first order in time. While it is possible to recast the momentum equation in `dzeus35` (or any other code in the *ZEUS* family) to an unsplit form, this rather invasive design change has yet to be undertaken. Thus, "rarefaction shocks" remain a *technical* problem for `dzeus35`, but not necessarily a *practical* one. For example, Figures 1 and 2 allow us to draw the following observations, some of which were pointed out in F02:

1. "Rarefaction shocks" can be eliminated with an appropriate (but sometimes undesirably high) linear viscosity term (Figure 1).
2. "Rarefaction shocks" are typically tripped in the rather contrived situation shown where the difference between the upwind and downwind speeds is small compared to the average speed across the fan. Performing a Galilean transformation toward the comoving frame of the fan (e.g., $\Delta v_1 = 4.0$, the left panel of Figure 2) eliminates the "rarefaction shocks" without increasing the linear viscosity.
3. *In multi-dimensional applications, it is highly unlikely that one would have 100 zones to resolve a rarefaction fan!* At more typical resolutions of, say, five zones (e.g., the right panel of Figure 2), "rarefaction shocks" disappear without increasing the linear viscosity and without performing a Galilean transformation.
4. Regardless of whether "rarefaction shocks" appear within the rarefaction region, the upstream state remains unaffected.

Of the numerous papers based on calculations performed by one form of *ZEUS* or another, F02 is the first I am aware of to exhibit "rarefaction shocks." These features seem to be reserved to a few pedagogical cases in highly resolved 1D test problems and are unlikely to pose any significant concern to the multi-physics, multi-dimensional simulations for which *ZEUS* has become renowned.

3. THE INTERNAL VERSUS TOTAL ENERGY EQUATIONS AND "SHOCK ERRORS"

Most codes bearing the *ZEUS* moniker solve only the internal energy equation, namely,

$$\partial_t e + \nabla \cdot (e\vec{v}) = -p\, \nabla \cdot \vec{v} - \mathsf{Q} : \nabla\vec{v}, \tag{1}$$

where Q is the von Neumann–Richtmyer artificial viscous stress tensor (see Equation (A3) in Appendix A) and where all other symbols have their usual meanings. Without the artificial viscous term, Equation (1) is formally isentropic. Thus, in addition to stabilizing MHD waves, the artificial viscous term helps ensure the correct entropy jump across shocks.

The principal advantage of Equation (1) is that in its differenced form, one can show that the Courant condition ($\mathcal{C} \equiv v\, \delta t/\delta x < 1$) is sufficient to guarantee a *positive-definite* e. In multi-dimensions, positive-definiteness is also guaranteed so long as the integrations are directionally split; in an unsplit scheme, one requires $\mathcal{C} < 1/2$ in 2D, and $\mathcal{C} < 1/3$ in 3D.

However, Equation (1) is not in *conservative* form, since the right-hand side is an imperfect divergence. As a result, algorithmic truncation errors introduce an effectively non-conservative term which, in some situations, can lead to significant deviations from the analytical conservative solutions, such as the 1D shock-tube examples shown in Figures 5 and 6 of F02, and in the Sedov blast wave described in T08.

Clarke (1996) shows how *ZEUS* can be fit with the *total energy equation*. It would seem, however, that F02 and T08 only considered the internal energy equation in performing their tests. As currently solved in `dzeus35`, the total energy equation is

$$\partial_t e_{\rm T} + \nabla \cdot (\vec{v}\,(e_{\rm H} + p) + \vec{v} \cdot \mathsf{Q} + \vec{S}) = 0, \tag{2}$$

where $e_{\rm H} = e + \frac{1}{2}\rho v^2$ is the total hydrodynamical energy density,[4] $e_{\rm T} = e_{\rm H} + B^2/2$ is the total energy density,[5] and

[4] In fact, `dzeus35` includes two more terms in $e_{\rm H}$, namely e_2 for the second (diffusive) fluid and $\rho\phi$ for the gravitational potential. However and for simplicity, discussion here remains limited to a single fluid with no gravity.

[5] The magnetic field, B, is in units where $\mu_0 = 1$.

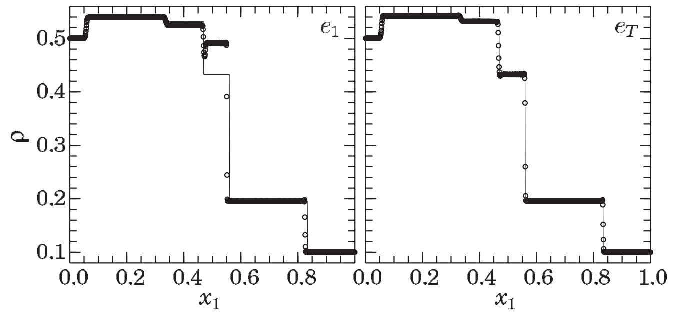

Figure 3. Density at $t = 0.06$ from the Riemann problem with left and right states $(\rho, v_1, v_2, v_3, B_2, B_3, p) = (0.5, 0, 2.0, 0, 2.5, 0, 10)$ and $(0.1, -10, 0, 0, 2, 0, 0.1)$ with $B_1 = 2$ and $\gamma = 5/3$. The grid domain [0, 1] is resolved with 1000 zones, with the original discontinuity at $x_1 = 0.5$. Open circles are dzeus35 solutions using the internal energy equation (left) and the total energy equation (right). In both cases, $\mathcal{C} = 0.75$, qcon = 2, and qlin = 0.2. Lines are the analytical solution using the Riemann solver described by RJ95. From left to right, the features are (1) fast shock, (2) slow rarefaction ($x_1 \sim 0.33$), (3) contact discontinuity ($x_1 \sim 0.47$), (4) slow shock ($x_1 \sim 0.56$), and (5) fast shock.

$\vec{S} = \vec{E} \times \vec{B}$ is the Poynting vector (where $\vec{E} = -\vec{v} \times \vec{B}$ is the induced electric field). Details of how this is implemented in dzeus35, which has evolved somewhat since Clarke (1996), are given in Appendix A.

Equation (2) is in conservative form, and renders *ZEUS* as much a conservative code as the zone-centered Godunov methods since the continuity and momentum equations (in Cartesian coordinates) have always been solved by *ZEUS* in conservative form. However, unlike the internal energy equation, solving the total energy equation does *not* guarantee a positive-definite e, and thus pressure, $p = (\gamma - 1)e$. This problem is rarely apparent in 1D test problems, but can be rampant in multi-dimensional applications.

Figure 3 shows essentially the same two plots in Figure 6 of F02. In the left panel, the solution to the Riemann problem using the non-conservative internal energy Equation (1) clearly disagrees with the analytical conservative solution (lines). Disagreements include both the levels attained in quiescent regions and distances traveled by shocks. On the other hand, the solution in the right panel using the total energy Equation (2) is at least as good as the solution in F02 using a fully upwinded, "shock-capturing" code. In fact, there should be no surprise that the solution obtained by solving a (numerically) lossy energy equation does not always agree with the analytical solution that assumes perfect conservation of mechanical energy. Indeed, the conclusions in F02 point more to an inappropriate use of the code than it does to any inherent "flaw" in its design and claims that results like these somehow make *ZEUS* "just about acceptable for pure gasdynamics [*sic*]" but "not satisfactory for adiabatic MHD" are unfounded.

Similarly, the finding in T08 that *ZEUS* (as manifest in *ENZO*; O'Shea et al. 2004) "fails" to get the shock speed right in a Sedov blast wave also stems from comparing the numerical solution using the lossy internal energy equation with the analytical conservative results. And while it is true that in 3D Cartesian coordinates and with an ill-advised setting of certain parameters, *ZEUS* will produce an anisotropic blast front particularly at early times, this has nothing to do with how far the blast wave propagates, as T08 claims.

To set up the Sedov blast wave, I initialize a quiescent region of gas with $\rho = 1$ and $p = 10^{-5}$ everywhere, and add 10^5 units of internal energy to a sphere of radius $r_0 = 0.0875$ centered at the origin of the grid.[6] Two problems are set up, each with qlin = 0.2, $\mathcal{C} = 0.75$, and second-order interpolation. The first is in spherical polar coordinates with ϑ- and φ-symmetry, and 5000 uniform radial zones in $0 \leqslant r \leqslant 5$. The second is in Cartesian coordinates with $0 \leqslant x, y, z \leqslant 5$ on a 200^3 grid with reflecting boundary conditions on the $x = 0$, $y = 0$, and $z = 0$ planes. In dzeus35, both reflecting boundary conditions and preservation of octal symmetry are good to machine round-off, and there is no need to do the entire $-5 \leqslant x, y, z \leqslant 5$ box. Note that the 200^3 octant reproduces the effective resolution of the AMR codes used by T08.

Table 1
Maximum Pressure and Shock Position of the Sedov Blast Front

	qcon	0	1	2	4	8
e:	$p_{\rm max}$	1740	3380	3900	4250	4580
	$r_{\rm sh}$	1.28	1.74	1.84	1.95	2.00
$e_{\rm T}$:	$p_{\rm max}$	3880	3850	3810	3770	3750
	$r_{\rm sh}$	1.81	1.81	1.82	1.82	1.82

Notes. The maximum pressure behind the Sedov shock front ($p_{\rm max}$) and its radius ($r_{\rm sh}$) at $t = 0.01$ tabulated against qcon for the 1D spherical polar dzeus35 solutions with 1250 zones using the internal (top) and total (bottom) energy equations. The analytical solution is $p_{\rm max} = 3960$ and $r_{\rm sh} = 1.83$, all in units described in the text. Note that with the internal energy equation, it is possible to get values both greater than and less than the analytical values, whereas with the total energy equation, results are largely independent of qcon.

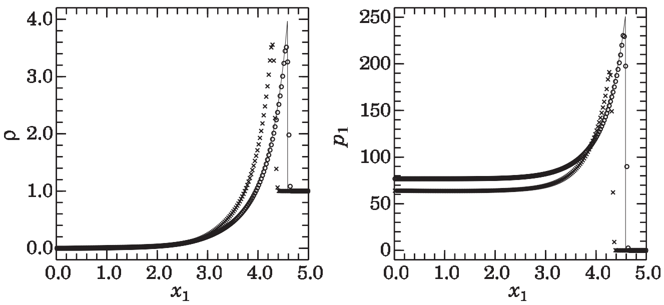

Figure 4. One-dimensional profiles of density (left) and pressure (right) at $t = 0.1$ for the Sedov blast wave described in the text. Lines are the "analytical" solution while circles (crosses) are 1D slices through the 3D Cartesian solutions using the total (internal) energy equation.

The choice of qcon warrants some comment. I find that when the total energy equation is used, the numerical solution—whether in 1D spherical polar coordinates or 3D Cartesian coordinates—is virtually independent of qcon, *including* qcon = 0. Because the time step is dictated by the exceedingly high temperature at the core, the shock takes many time steps to cross a single zone and the inherent *grid viscosity* is sufficient to stabilize it and achieve the correct entropy jump. On the other hand and as noted by T08, the nature of the solution (e.g., distance propagated by the shock and peak values behind the shock—but not shock stability) depends very heavily on the value of qcon when the internal energy equation is used. Illustrative examples for 1D spherical polar coordinates solutions at $t = 0.01$ are given in Table 1. Thus, I have chosen to use qcon = 0 for the total energy equation, and qcon = 2 for the internal energy equation, the latter corresponding to T08.

Figure 4 shows, at $t = 0.1$, the density and pressure profiles of the 1D spherically symmetric solution (lines, taken as the "analytic" solution) along with the 3D Cartesian solutions

[6] The initial pressure jump of 2.4×10^{12} means that machine accuracy may be a concern even with double precision.

using the total (circles) and internal (crosses) energy equations. Evidently, dzeus35 with the total energy equation reproduces the conservative analytical result as well as or better than any of the upwinded codes used by T08 (e.g., their Figure 6), while the internal energy equation clearly does not. In this problem, use of the total energy equation conserves the total energy to machine accuracy and, except for a very brief time at the beginning of the simulation where the blast front accelerates the ambient medium, the internal and kinetic energies remain constant as well. With the internal energy equation, about a third of the total energy is lost and both the internal and kinetic energies decline throughout the simulation. It is this loss of energy that results in the lower peak in the pressure profile and the slower advance of the shock.

Panels (a) and (b) of Figure 5 show density slices for the internal and total energy solutions at $t = 0.1$. Both exhibit highly spherical shock fronts whose diameter is perceptibly smaller for the internal energy equation (panel (a)), as already discussed. Departure from spherical symmetry is seen in the lowest contour levels (about 1% of peak), particularly in panel (a), though contrary to the conclusion drawn by T08, this has nothing to do with the loss of energy in the non-conservative solution.

Panels (c) and (d) of Figure 5 both show the total energy equation solution at $t = 0.01$ for qcon = 0 (c), and qcon = 2 ((d); as used by T08). *It is immediately apparent that the quadratic viscosity is largely responsible for the anisotropic contours described by T08 as an "asymmetrical diamond shape."* Most *ZEUS* codes, including dzeus35, use the diagonal form of the von Neumann–Richtmyer artificial viscosity which is known to behave anisotropically. Only after the blast front is well resolved (e.g., >50 zones) will the anisotropy begin to dissipate. This problem may be reduced by setting qcon as small as possible (e.g., Figure 5(c)), or by installing a proper *tensor artificial viscosity* into the code, such as those described in Shultz (1964), Richtmyer & Morton (1967) or, more recently, Campbell & Shashkov (2001). (See also Stone & Norman (1992a), who installed tensor artificial viscosity in *ZEUS-2D*.) However, the anisotropic symptoms of the diagonal viscosity are apparent only in spherically symmetric applications, in which case the simulation should be done as a 1D problem in (r, ϑ, φ) coordinates where the anisotropy is, of course, completely absent.

As a final comment on this Sedov blast test, the initial pressure jump of *12 orders of magnitude* is a *very* unusual initial condition, and I note that for more "normal" shock strengths, the internal energy equation does just fine, losing a percent or less of the total energy in many applications. Thus, one should not dismiss outright the use of the internal energy equation based on these very extreme results; they serve as a caution only.

So why would one choose one energy equation over the other? Where conservation of mechanical energy is critical, one must use the total energy equation. However, because it does not guarantee a positive-definite pressure particularly in multi-dimensions, this can pose a severe problem for some applications. (See Section A.1 for a discussion on how negative pressures are minimized in dzeus35.) In this case, one might have to consider using the internal energy equation, receiving solace in the fact that there are few astrophysical systems in which mechanical energy is strictly conserved.

4. A FULL SUITE OF 1D TEST PROBLEMS

Figures 6–17 present the conservative dzeus35 solutions to the complete set of 1D test problems given in RJ95. Each figure, including profiles for the total energy density (omitted here for space) and additional commentary, is also available online[3].

Each test problem was run with the total energy equation ($\gamma = 5/3$) using 512 zones and, unless otherwise noted, with $\mathcal{C} = 0.75$, qlin = 1.0, qlin = 0.2, and Colella & Woodward's (1984) third-order piecewise parabolic interpolation with the contact steepener engaged. The initial discontinuity is placed at $x_1 = 0.5$. Circles show the dzeus35 solution at the indicated time while lines are the analytical solution using the Riemann solver described in RJ95.

This is the first time a complete set of *ZEUS* solutions have been presented for this test suite, since no version of *ZEUS* previous to dzeus35 of which I am aware could solve the Riemann problems in Figures 7 and 9. The original *ZEUS* code (zeus04 developed in 1986) inherited a technique known as *Consistent Advection* (CA; Norman et al. 1980) that was applied to *all* hydrodynamical variables. CA was invented and originally applied to angular momentum transport to settle a numerical controversy in the late 1970s on whether self-gravitating, rotating adiabatic gas would collapse to form a thick accretion disk or a torus. It was so successful in solving this debate (in favor of disks) that by the time *ZEUS* was developed, it had been applied to the remaining variables, namely, the linear momentum components and energy. See Clarke (1996) for the most recent exposition of how CA is applied in most existing 3D versions of *ZEUS*.

However, applying CA to the energy equation in particular was never fully tested (M. L. Norman 2007, private communication), and no deleterious consequences were discovered until very recently. As the left panel of Figure 18 shows, *ZEUS* with CA applied to the energy equation (either internal or total) excites severe ringing in the v_1-profile (as well as most other

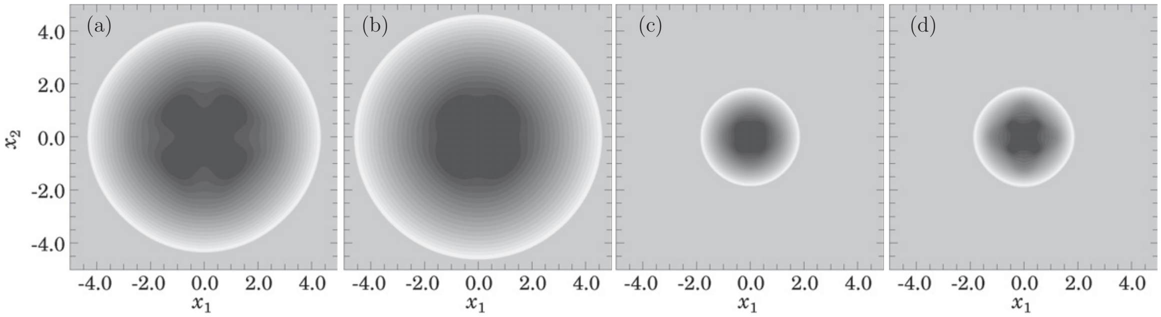

Figure 5. Two-dimensional slices of density for the Sedov blast wave described in the text. Panels (a) and (b) are solutions at $t = 0.1$ for the internal and total energy equations, respectively. Panels (c) and (d) are solutions at $t = 0.01$ using the total energy equation with qcon = 0 and 2, respectively.

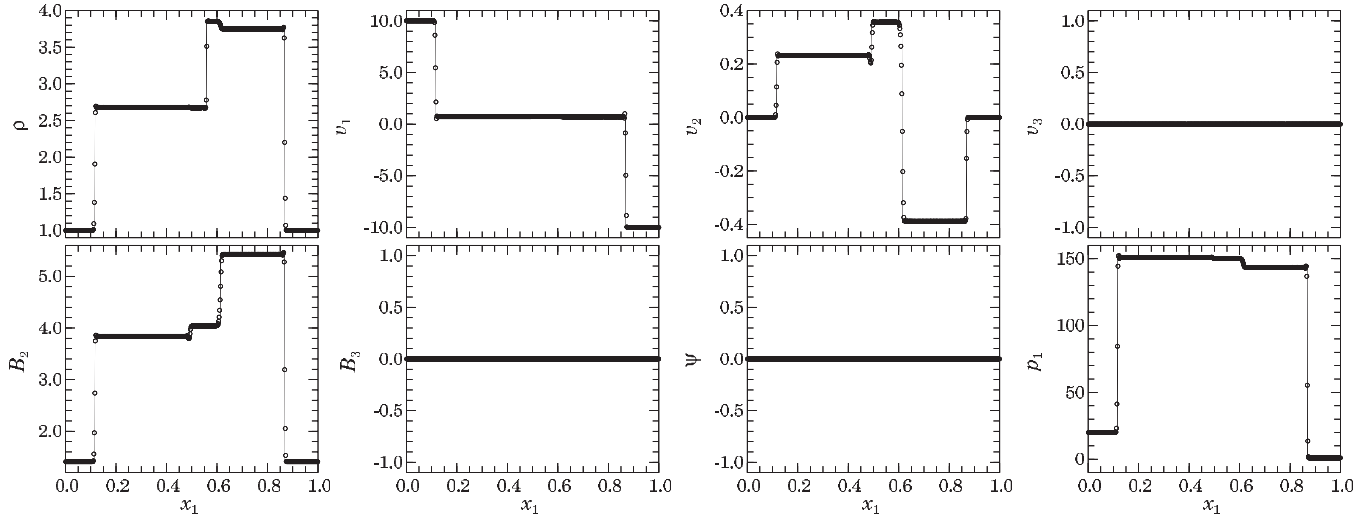

Figure 6. Solution to the Riemann problem at time $t = 0.08$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [1, 10, 0, 0, 5/(4\pi)^{1/2}, 0, 20]$ and $[1, -10, 0, 0, 5/(4\pi)^{1/2}, 0, 1]$ respectively, with $B_1 = 5/(4\pi)^{1/2}$ (Figure 1(a) from RJ95). Plots show from left to right: (1) fast shock, (2) slow rarefaction ($x_1 \sim 0.5$), (3) contact discontinuity ($x_1 \sim 0.55$), (4) slow shock ($x_1 \sim 0.62$), and (5) fast shock.

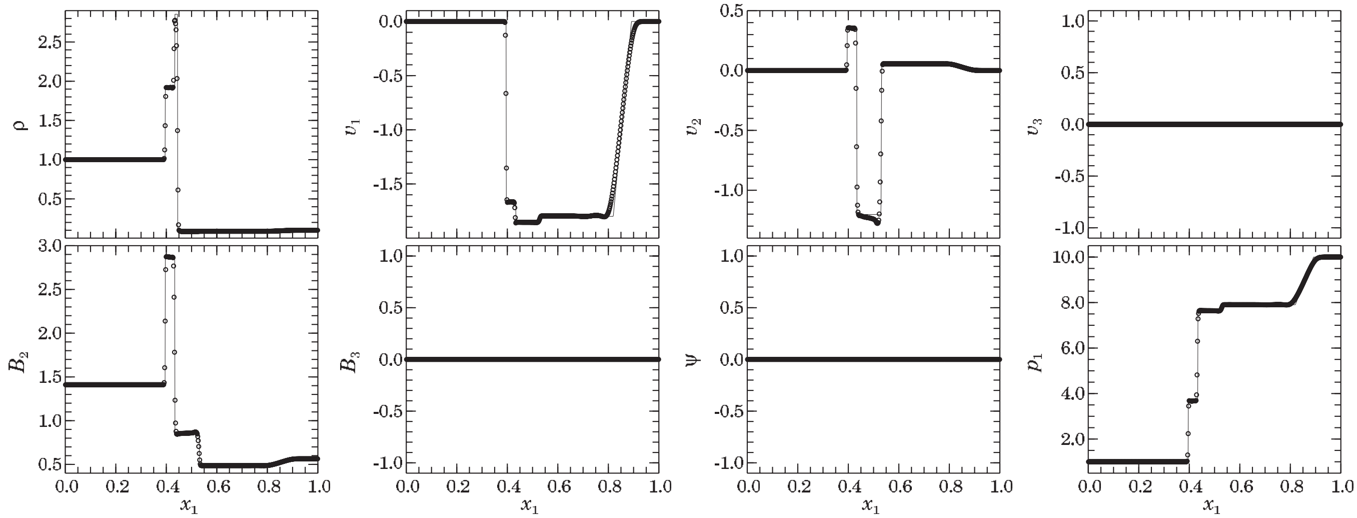

Figure 7. Solution to the Riemann problem at time $t = 0.03$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [1, 0, 0, 0, 5/(4\pi)^{1/2}, 0, 1]$ and $[0.1, 0, 0, 0, 2/(4\pi)^{1/2}, 0, 10]$, respectively, with $B_1 = 3/(4\pi)^{1/2}$ (Figure 1(b) from RJ95). Plots show from left to right: (1) fast shock, (2) slow shock ($x_1 \sim 0.43$), (3) contact discontinuity ($x_1 \sim 0.45$), (4) slow rarefaction ($x_1 \sim 0.53$), and (5) a fast rarefaction. The contact steepener is disengaged.

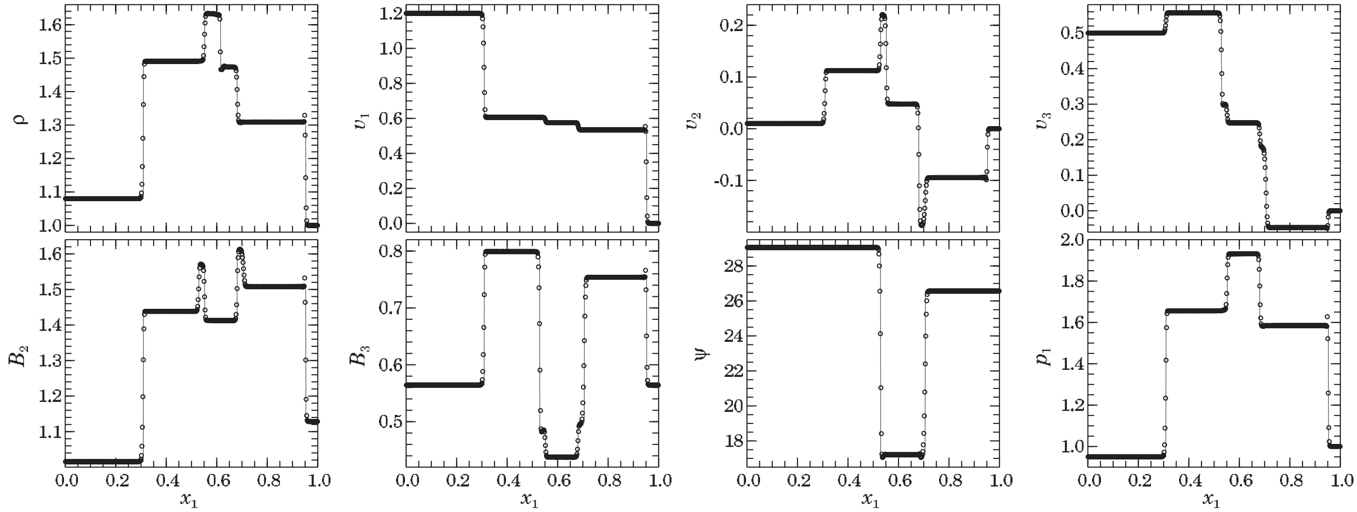

Figure 8. Solution to the Riemann problem at time $t = 0.2$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [1.08, 1.2, 0.01, 0.5, 3.6/(4\pi)^{1/2}, 2/(4\pi)^{1/2}, 0.95]$ and $[1, 0, 0, 0, 4/(4\pi)^{1/2}, 2/(4\pi)^{1/2}, 1]$, respectively, with $B_1 = 2/(4\pi)^{1/2}$ (Figure 2(a) from RJ95). Plots show from left to right: (1) fast shock, (2) rotational discontinuity ($x_1 \sim 0.53$), (3) slow shock ($x_1 \sim 0.55$), (4) contact discontinuity ($x_1 \sim 0.61$), (5) slow shock ($x_1 \sim 0.68$), (6) rotational discontinuity ($x_1 \sim 0.71$), and (7) fast shock.

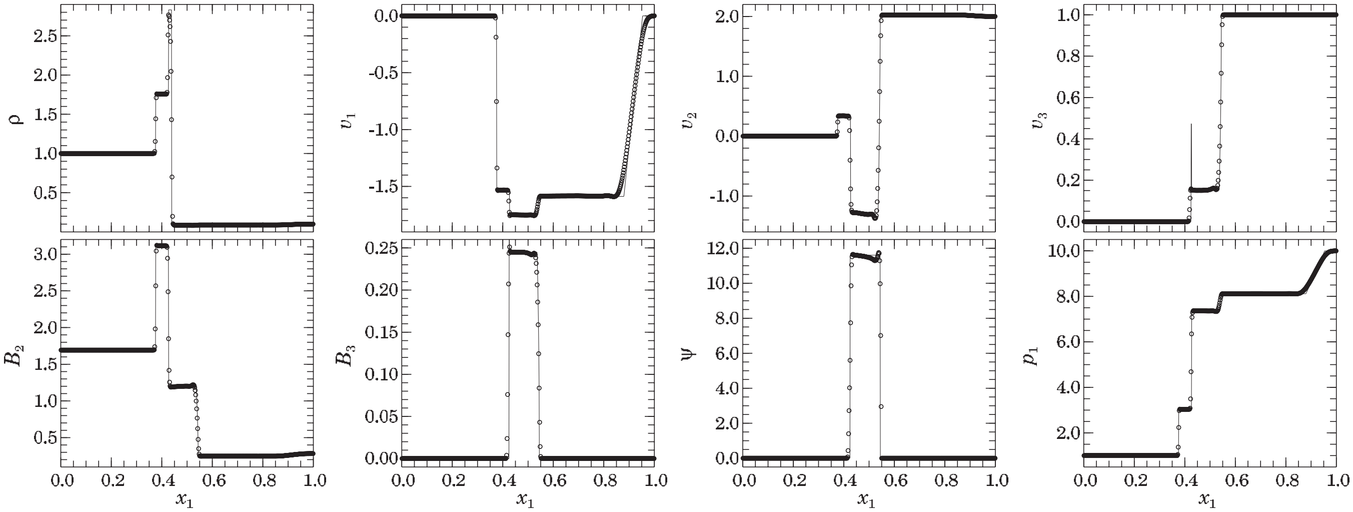

Figure 9. Solution to the Riemann problem at time $t = 0.035$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [1, 0, 0, 0, 6/(4\pi)^{1/2}, 0, 1]$ and $[0.1, 0, 2, 1, 1/(4\pi)^{1/2}, 0, 10]$, respectively, with $B_1 = 3/(4\pi)^{1/2}$ (Figure 2(b) from RJ95). Plots show from left to right: (1) fast shock, (2) rotational discontinuity ($x_1 \sim 0.425$), (3) slow shock ($x_1 \sim 0.426$), (4) contact discontinuity ($x_1 \sim 0.44$), (5) slow rarefaction ($x_1 \sim 0.54$), (6) rotational discontinuity ($x_1 \sim 0.55$), and (7) fast rarefaction. The contact steepener is disengaged.

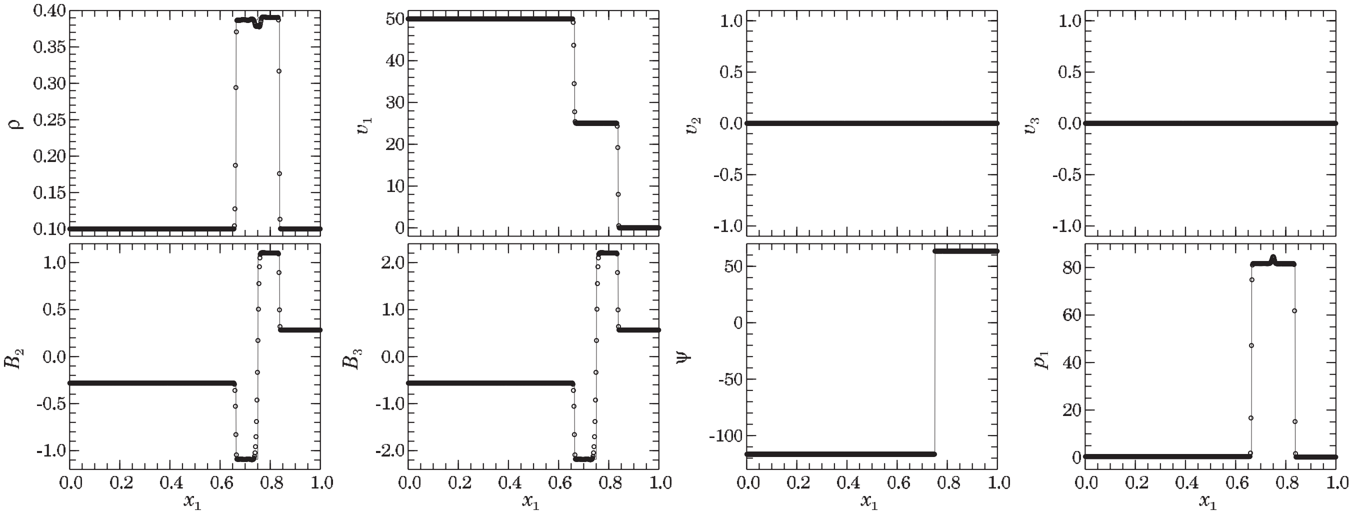

Figure 10. Solution to the Riemann problem at time $t = 0.01$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [0.1, 50, 0, 0, -1/(4\pi)^{1/2}, -2/(4\pi)^{1/2}, 0.4]$ and $[0.1, 0, 0, 0, 1/(4\pi)^{1/2}, 2/(4\pi)^{1/2}, 0.2]$, respectively, with $B_1 = 0$ (Figure 3(a) from RJ95). Plots show from left to right: (1) magnetoacoustical shock, (2) tangential discontinuity ($x_1 \sim 0.75$), and (3) magnetoacoustical shock. Second-order piecewise linear interpolations with no contact steepener (van Leer 1977) are used.

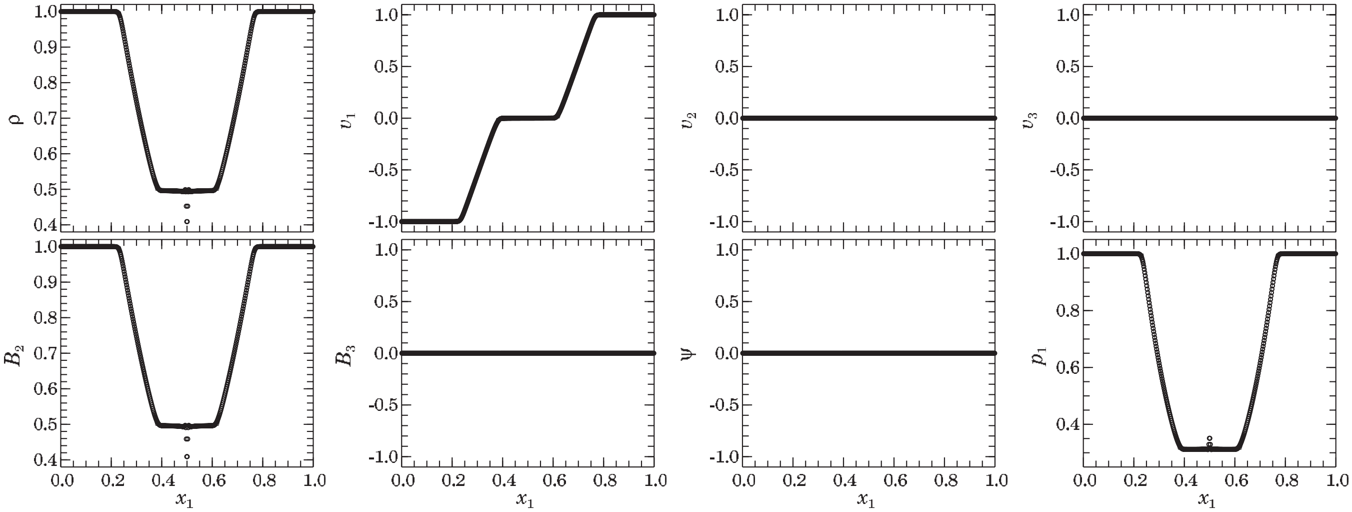

Figure 11. Solution to the Riemann problem at time $t = 0.1$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [1, -1, 0, 0, 1, 0, 1]$ and $[1, 1, 0, 0, 1, 0, 1]$, respectively, with $B_1 = 0$ (Figure 3(b) from RJ95). Plots show from left to right two oppositely moving magnetoacoustical ($B_1 = 0$) rarefactions. To suppress "rarefaction shocks," `qlin` = 0.4.

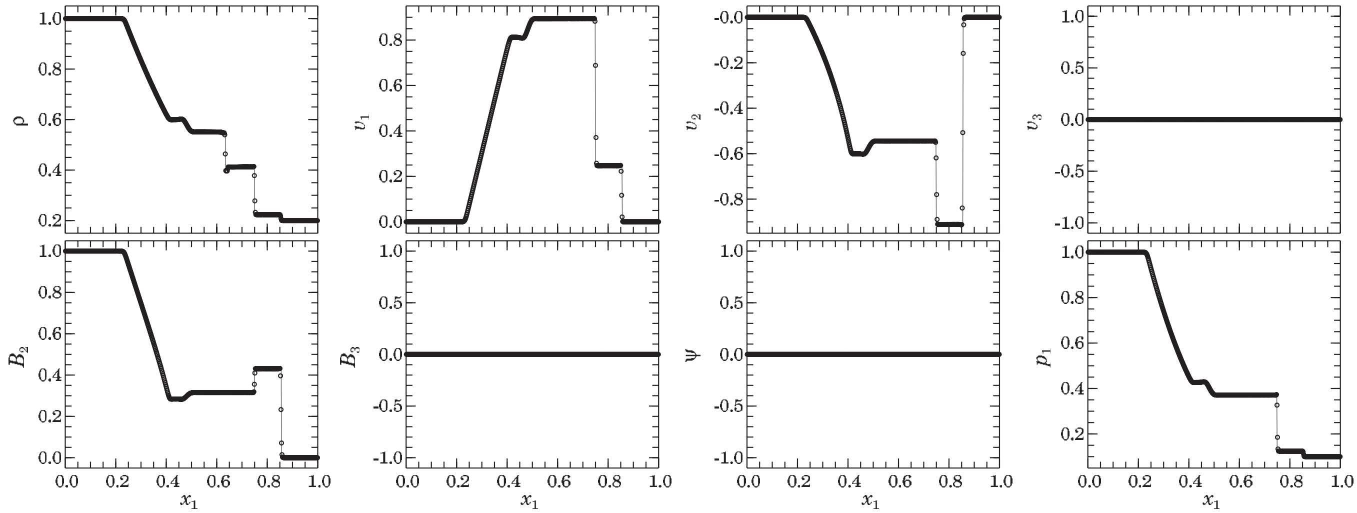

Figure 12. Solution to the Riemann problem at time $t = 0.15$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [1, 0, 0, 0, 1, 0, 1]$ and $[0.2, 0, 0, 0, 0, 0, 0.1]$, respectively, with $B_1 = 1$ (Figure 4(a) from RJ95). Plots show from left to right: (1) fast rarefaction, (2) slow rarefaction (at $x_1 \sim 0.45$), (3) contact discontinuity (at $x_1 \sim 0.64$), (4) slow shock (at $x_1 \sim 0.75$), and (5) "switch-on" fast shock. This is an example of a problem in which no linear viscosity is required for stabilization, and `qlin` = 0.

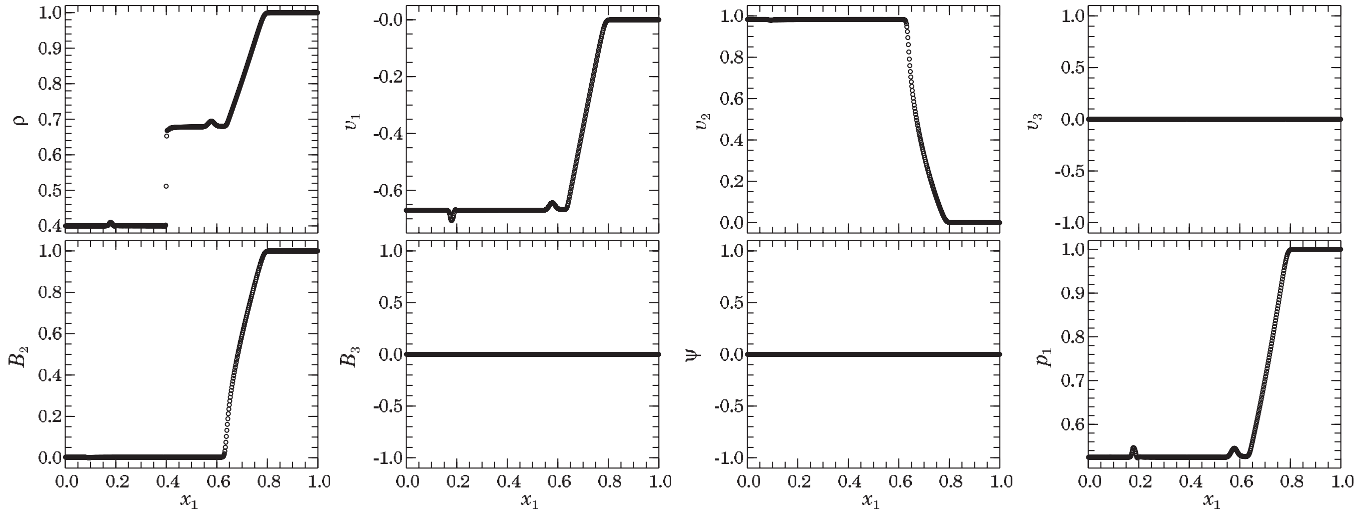

Figure 13. Solution to the Riemann problem at time $t = 0.15$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [0.4, -0.66991, 0.98263, 0, 0.0025293, 0, 0.52467]$ and $[1, 0, 0, 0, 1, 0, 1]$, respectively, with $B_1 = 1.3$ (Figure 4(b) from RJ95). Plots show from left to right: (1) contact discontinuity ($x_1 \sim 0.4$), and (2) "switch-off" fast rarefaction ($0.63 < x_1 < 0.78$). The Riemann solver failed to give an analytical solution for this problem.

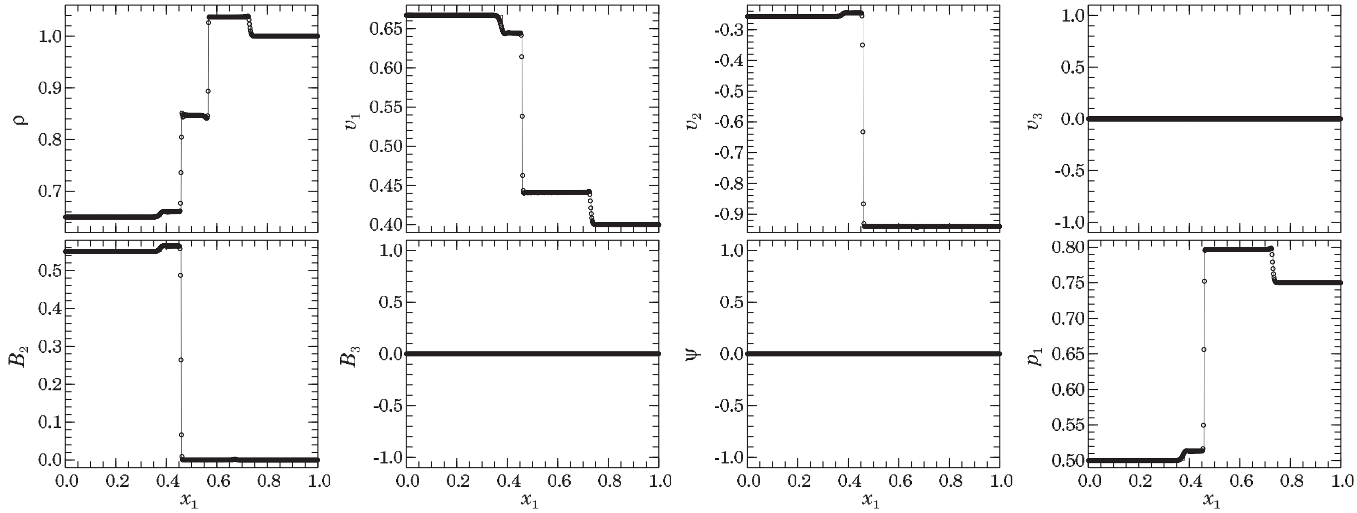

Figure 14. Solution to the Riemann problem at time $t = 0.15$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [0.65, 0.667, -0.257, 0, 0.55, 0, 0.5]$ and $[1, 0.4, -0.94, 0, 0, 0, 0.75]$, respectively, with $B_1 = 0.75$ (Figure 4(c) from RJ95). Plots show from left to right: (1) fast (weak) shock ($x_1 \sim 0.38$), (2) "switch-off" slow shock ($x_1 \sim 0.46$), (3) contact discontinuity ($x_1 \sim 0.56$), and (4) hydrodynamical shock ($x_1 \sim 0.73$).

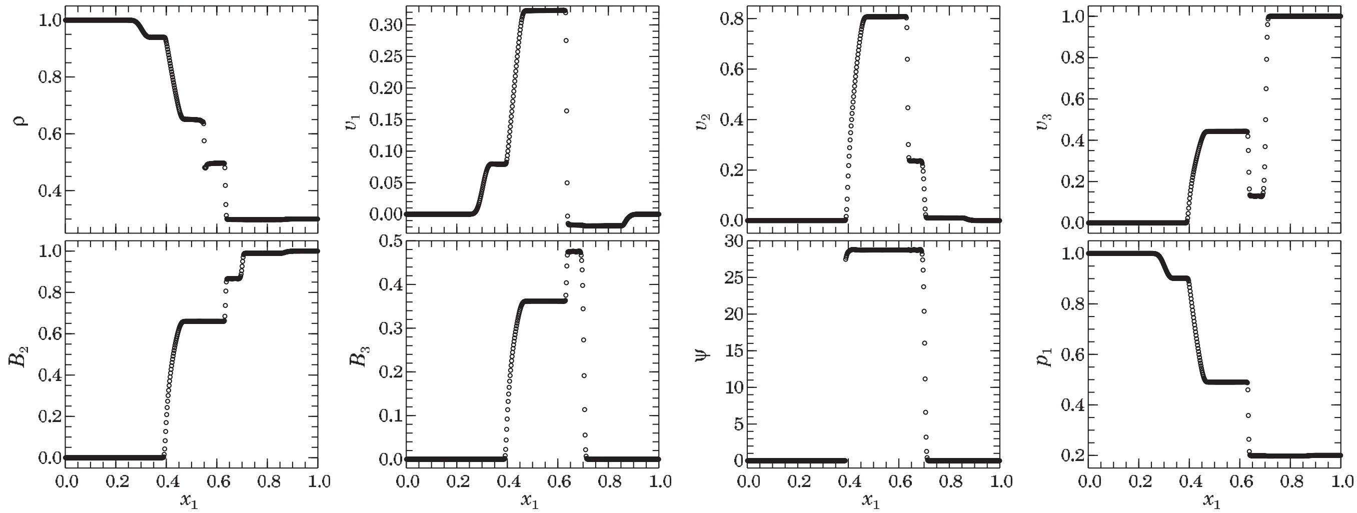

Figure 15. Solution to the Riemann problem at time $t = 0.16$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [1, 0, 0, 0, 0, 0, 1]$ and $[0.3, 0, 0, 1, 1, 0, 0.2]$, respectively, with $B_1 = 0.7$ (Figure 4(d) from RJ95). Plots show from left to right: (1) hydrodynamical rarefaction, (2) "switch-on" slow rarefaction ($0.4 < x_1 < 0.45$), (3) contact discontinuity ($x_1 \sim 0.55$), (4) slow shock ($x_1 \sim 0.64$), (5) rotational discontinuity ($x_1 \sim 0.7$), and (6) fast rarefaction. To suppress "rarefaction shocks" in feature (6), `qlin` = 0.4. The Riemann solver failed to give an analytical solution for this problem.

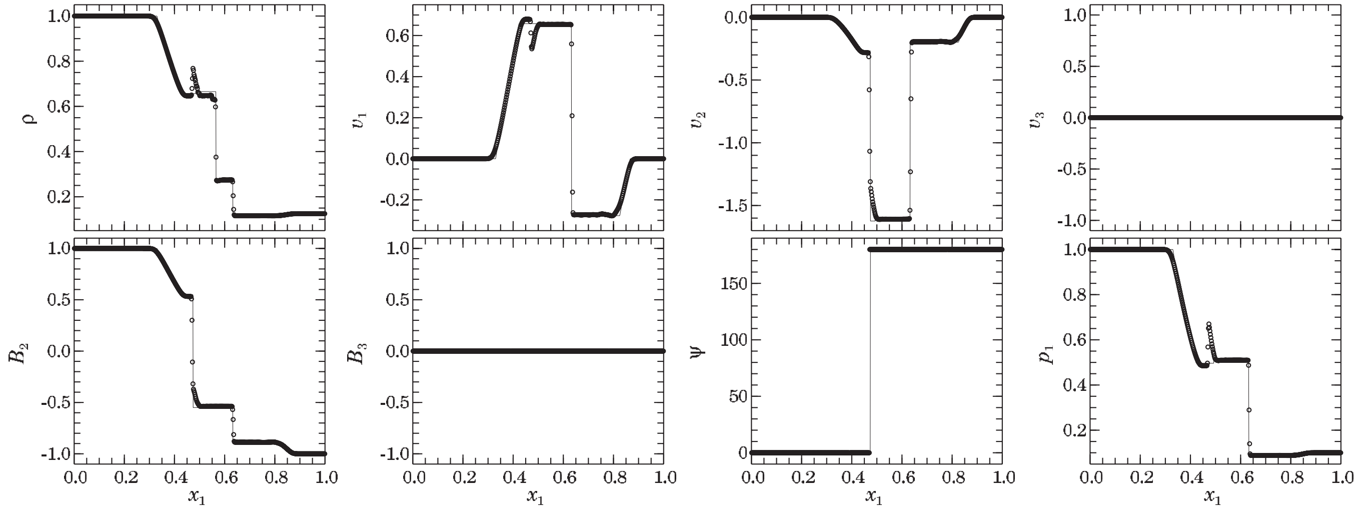

Figure 16. Solution to the Riemann problem at time $t = 0.1$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [1, 0, 0, 0, 1, 0, 1]$ and $[0.125, 0, 0, 0, -1, 0, 0.1]$, respectively, with $B_1 = 0.75$ (Figure 5(a) from RJ95). This is the Brio & Wu (1988) problem with $\gamma = 5/3$ instead of 2. Plots show from left to right: (1) fast rarefaction, (2) slow compound wave ($x_1 \sim 0.47$), (3) contact discontinuity ($x_1 \sim 0.56$), (4) slow shock ($x_1 \sim 0.63$), and (5) fast rarefaction. To stabilize the compound wave, `qlin` = 0.5.

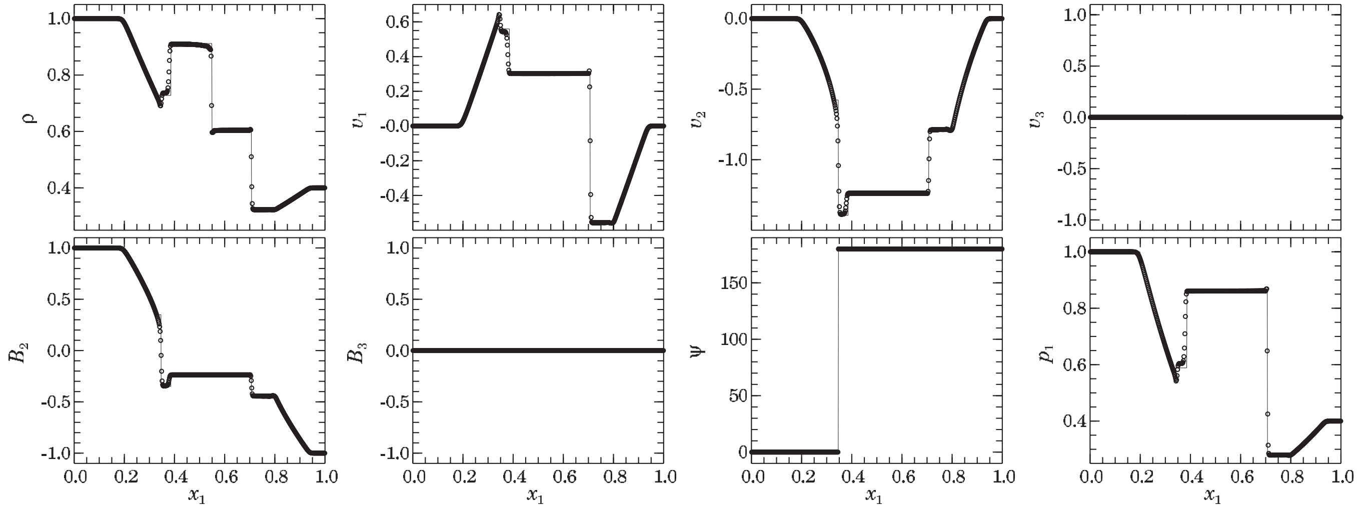

Figure 17. Solution to the Riemann problem at time $t = 0.16$ with initial left and right states given by $(\rho, v_1, v_2, v_3, B_2, B_3, p) = [1, 0, 0, 0, 1, 0, 1]$ and $[0.4, 0, 0, 0, -1, 0, 0.4]$, respectively, with $B_1 = 1.3$ (Figure 5(b) from RJ95). Plots show from left to right: (1) fast compound wave, (2) slow shock ($x_1 \sim 0.38$), (3) contact discontinuity ($x_1 \sim 0.55$), (4) slow shock ($x_1 \sim 0.7$), and (5) a fast rarefaction.

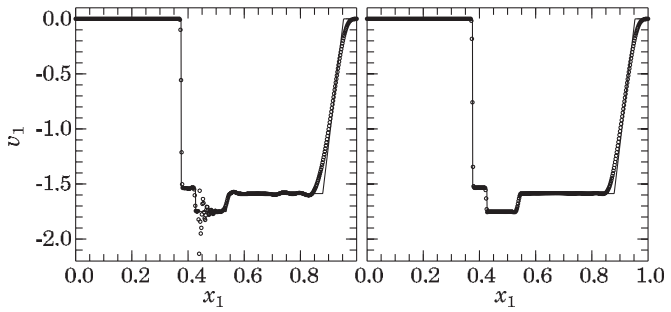

Figure 18. v_1-profile for the problem depicted in Figure 9 with CA applied to the total energy equation (left) and without (right). Solutions using the internal energy equation are qualitatively identical.

variables) between the slow waves in the problem shown in Figure 9. With CA removed from the energy equations (but still applied to the momenta), the right panel of Figure 18 shows the `dzeus35` solution with no oscillations at all.

CA fails in this pedagogical case because of a juxtaposition of two rather unusual events. In both the problems depicted in Figures 7 and 9, the specific energy density ($e_{\rm T}/\rho$)—which CA requires be interpolated rather than $e_{\rm T}$ directly—is practically continuous across the slow shock (at $x_1 \sim 0.43$) since both $e_{\rm T}$ and ρ rise by a factor of $\sim$1.5. With little or no discontinuity in the specific energy, the monotonizers are not engaged and any oscillations creeping into $e_{\rm T}$ are left undamped. Where do these oscillations come from? The huge jump in density across the contact at $x_1 \sim 0.45$—only several zones away from the shock—severely challenges the monotonizers in the interpolation routines because, while $e_{\rm T}$ drops by a modest factor of 1.6 across the contact, $e_{\rm T}/\rho$ jumps by a factor of well over 100. The resulting and unchecked oscillations in $e_{\rm T}$ persist and contaminate the remaining variables.

These tests do not necessarily mean that CA must be removed from the energy equation for *all* applications. One could, for example, imagine CA being useful in preventing an artificial redistribution of temperatures in a region where the internal energy density varies more strongly than the density. Thus, should it be suspected that an application might benefit from the removal of CA from the energy equation, Appendix B gives details on how this may be done.

Finally, a comment is given on the overall efficiency of *ZEUS*. By doing a simple count, `dzeus35` requires, in aggregate, ~ 140 zones to resolve the 34 discontinuities in the 12 test problems shown in Figures 6–17, while the conservative and upwinded TVD scheme used by RJ95—which does not use a contact steepener—requires $\sim$200. Figure 6 in F02 shows $\sim$16 zones in the four discontinuities, whereas `dzeus35` requires $\sim$20 (Figure 3). If the definition of a "shock-capturing" algorithm is based on the number of zones required to resolve and stabilize a discontinuity, `dzeus35` is as much a "shock-capturing" code as any fully upwinded scheme. Indeed, zone for zone, `dzeus35` can generally provide just as crisp a solution to a 1D Riemann problem as most "upwinded schemes" despite formal measures that show *ZEUS* converges to first order while upwinded schemes converge to second order. But even if *ZEUS* does require more zones than an upwinded code to obtain a given result to a similar accuracy, it can update those zones at a significantly faster rate than a typical Godunov scheme (e.g., Stone 2009) partially or completely offseting the cost of any additional resolution. Claims, therefore, that suggest *ZEUS* needs twice the resolution and thus, in 3D, 16 times the cpu time to compute the same problem to the same accuracy as an upwinded scheme (e.g., F02) are completely unproven, particularly when directionally split upwinded schemes reduce to first order in multi-dimensions anyway.

In practice and not including I/O, `dzeus35` requires about 1500, 2200, and 3000 floating point operations (FLOPs) per MHD zone update (no self-gravity) for 1D, 2D, and 3D simulations, respectively. Thus, a 256^3 simulation carried to 25,000 time steps (typical for the 3D turbulence run discussed in Section 6) requires about 10^{15} FLOPs to complete. On a 2 GHz chip with 50% efficiency, this means about 350 cpu hr or, with a 12.5 speed-up factor on a 16-way SMP,[7] 28 wall-clock hours.

5. A PARTIAL SUITE OF 2D TEST PROBLEMS

Three 2D test problems are described here. These and additional test problems, as they are prepared, will appear on the *ZEUS-3D* Web site[3].

5.1. Launching an Alfvén Wave from a Boundary

An MHD code is only as good as its boundary conditions, and a great deal of effort has been spent in ensuring the variety of MHD boundary conditions in `dzeus35` (reflecting, periodic, flow-in, and flow-out being the primary examples), are stable and preserve the solenoidal condition to machine round-off. Further, as many practical applications require more than one boundary type along the same boundary (e.g., launching a jet into a grid requires flow-in conditions in the jet orifice, but reflecting or flow-out conditions on the rest of the boundary), zones of "mixed boundary type" are unavoidable particularly on a staggered grid, and must be dealt with carefully.

Consider the problem of launching an Alfvén wave on an axisymmetric cylindrical grid from the $r > 1$ portion of the $z = 0$ boundary. Thus, the point $(z, r) = (0, 1)$ is a "mixed boundary point" where "flow-in" conditions for $r > 1$ abut with "reflecting" conditions for $r < 1$. Previous to `dzeus35`, no *ZEUS* code I am aware of (including versions of *ZEUS-2D* and *ZEUSMP*, as well as earlier versions of my own code) could do this problem without over-specifying the boundary conditions (since a zero in-flow speed is clearly sub-slow, one can only set four of the seven characteristics) and getting a completely incorrect solution on the grid.

Under the assumption of incompressibility, the φ components of the MHD equations (e.g., Clarke 1996) in axisymmetric cylindrical coordinates reduce to

$$\partial_t v_\varphi + v_z \partial_z v_\varphi - a_z \partial_z a_\varphi = 0,$$
$$\partial_t a_\varphi + v_z \partial_z a_\varphi - a_z \partial_z v_\varphi = 0,$$

where $a_i = B_i/\sqrt{\rho}$ is the Alfvén speed associated with B_i, $i = z, \varphi$. Adding and subtracting these equations in the comoving frame ($v_z = 0$) yield the usual characteristic equations:

$$D_t^{\pm}(v_\varphi \mp a_\varphi) = 0, \tag{3}$$

where the Lagrangian derivatives, $D_t^{\pm} = \partial_t \pm a_z \partial_z$ have been introduced. With the initial conditions,

$$v_\varphi(r, t = 0) = \begin{cases} v_{\varphi,0}(r), & z \leqslant 0, \\ 0, & z > 0, \end{cases} \quad a_\varphi(r, t = 0) = 0, \tag{4}$$

[7] `dzeus35` is written for *OpenMP* commands, but not *MPI*.

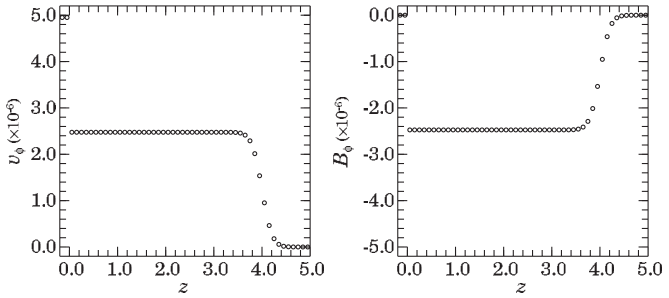

Figure 19. Profiles of v_φ (left) and B_φ (right) at $r = 5$ (last active zone before the outflow boundary conditions) and $t = 4$ for the Alfvén wave launched from the $z = 0$ boundary, as described in the text. Open circles are the `dzeus35` solutions, with the two left-most markers indicating the imposed boundary values.

and with these conditions *maintained* in $z < 0$, Equations (3) are differenced and solved for $v_\varphi(r, t)$ and $a_\varphi(r, t)$ to get

$$v_\varphi(r,t) = \begin{cases} v_{\varphi,0}(r), & z \leqslant 0, \\ \frac{1}{2} v_{\varphi,0}(r), & 0 < z < a_z t, \\ 0, & z \geqslant a_z t, \end{cases}$$

$$a_\varphi(r,t) = \begin{cases} 0, & z \leqslant 0, \\ -\frac{1}{2} v_{\varphi,0}(r), & 0 < z < a_z t, \\ 0, & z \geqslant a_z t. \end{cases}$$

This is the analytical solution for an Alfvén wave launched from the $z = 0$ boundary with the initial and boundary conditions described by Equations (4).

To test the ability of `dzeus35` to launch such an Alfvén wave, I set a 50 × 50 2D axisymmetric grid in cylindrical coordinates with $(z, r) = (0\!:\!5, 0\!:\!5)$, and initialize $\rho = p = B_z = 1$ with all remaining vector components zero. At the $z = 0$ boundary, I additionally set

$$v_{\varphi,0}(r) = \begin{cases} 0, & r < 1, \\ 10^{-6} r, & r \geqslant 1. \end{cases}$$

Such an azimuthal velocity will be a perturbation on the otherwise quiescent initial conditions, and thus the assumption of incompressibility will hold approximately. So that it holds exactly, one could reset all variables except v_φ and B_φ to their initial values during and at the end of each MHD step.

At $t = 4$, the Alfvén wave (with Alfvén speed 1) propagates to $z = 4$, as shown in Figure 19. Plotted are the z-profiles of v_φ and B_φ just inside $r = 5$, and thus the last active row of zones before the outflow conditions.

Commenting on the performance of the boundary conditions, I note the following:

1. The outflow boundary conditions at $r = 5$ have *zero* measurable effect on the solution shown in Figure 19;
2. The "mixed boundary zone" at $(z, r) = (0, 1)$ has *zero* measurable effect on the solution at $r = 1$ (not shown in Figure 19);
3. As seen in Figure 19, the Alfvén wave moves onto the grid with a "perfect" discontinuity (i.e., no diffusion) between the maintained boundary conditions and the first active zone on the grid. (Meanwhile, the leading edge of the Alfvén wave is spread over several zones, consistent with the second-order accuracy of the MHD algorithm.[8]);
4. The values for v_φ and B_φ immediately to the right of $z = 0$ differ from the analytical values by less than one part in 10^6, with the error decreasing quadratically in time.

5.2. 2D MHD Blast

Gardiner & Stone (2005) suggest a few 2D problems to test a code's bias along the grid coordinate directions. In particular, they point out that in a directionally split Godunov scheme, the omission of the *compressional magnetic terms* (i.e., $\partial_i B_i$, $i = x, y, z$) which is justified in 1D by the solenoidal condition is carried forward to the multi-dimensional algorithm where this justification no longer holds (see also Balsara & Spicer 1999). They demonstrate that the omission of compressional magnetic terms leads to directional biases in calculations such as an MHD blast wave, which amounts to a 2D Riemann problem in axisymmetric coordinates.

Shown in Figure 20 is the result of the 2D test, with the particulars given in the figure caption. In this case, the magnetic field is oriented at 45° relative to the grid, with no discernible differences from similar simulations done with the field aligned with one of the coordinate axes, or at any other angle. This is because *ZEUS*, not being dependent on 1D Godunov solvers, has

[8] With the hydrodynamics effectively squelched as in this problem, `dzeus35` is second order in both space and time.

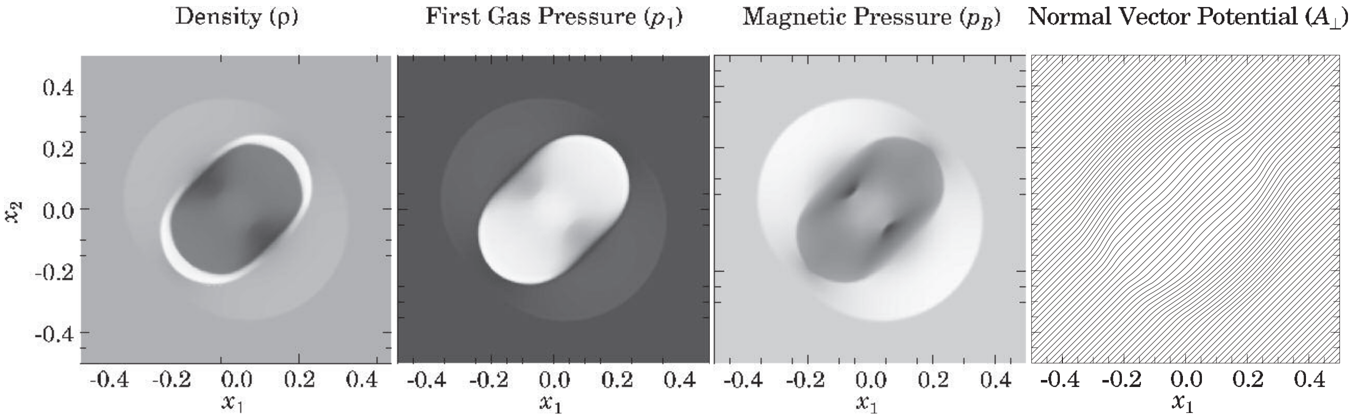

Figure 20. 200 × 200 zone 2D Cartesian grid with domain $(x_1, x_2) = (-0.5\!:\!0.5, -0.5\!:\!0.5)$ is initialized with $(\rho, \vec{v}, B_1, B_2, B_3) = (1, \vec{0}, 5\sqrt{2}, 5\sqrt{2}, 0)$ everywhere. A disk of radius $r = 0.125$ centered at the origin is over-pressured ($p = 100$) relative to the rest of the grid ($p = 1$). Shown from left to right are density, gas pressure, magnetic pressure, and magnetic field lines at $t = 0.02$ as solved by `dzeus35` using the total energy equation with `qcon` = 1.0, `qlin` = 0.1, $\mathcal{C} = 0.5$, and third-order interpolation with the contact steepener engaged. White (black) indicates high (low) values. Extrema of the variables at the epoch shown are $0.200 < \rho < 3.22$, $0.771 < p < 32.0$ and $24.9 < p_B < 76.0$ (cf. $0.192 < \rho < 3.31$, $1.00 < p < 32.1$ and $23.5 < p_B < 77.7$ for *ATHENA*).

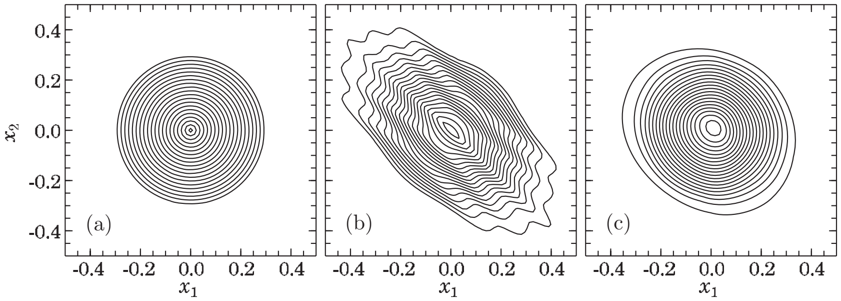

Figure 21. Contours of A_3 (and thus magnetic field lines) in the inner 64×64 half of a 128×64 periodic grid at $t = 2$ with: (a) $(v_1, v_2) = (0, 0)$, $\mathcal{C} = 0.5$; (b) $(v_1, v_2) = (2, 1)$, $\mathcal{C} = 0.5$; and (c) $(v_1, v_2) = (2, 1)$, $\mathcal{C} = 0.1$. The non-zero velocity moves the flux loop diagonally across the grid twice.

always accounted for the compressional magnetic terms. These solutions are qualitatively identical and quantitatively similar to those performed by *ATHENA*, an "unsplit" Godunov solver (Gardiner & Stone 2005).

5.3. A Difficult Problem for `dzeus35`*: Flux Loop Advection*

A well-known problem that smoothed particle hydrodynamics (SPH) can do much better than finite-volume codes is simple advection. Because of their Lagrangian design, SPH codes can move around "blobs" of material with effectively *zero* numerical dissipation, whereas finite-volume codes always suffer some dissipation as dynamically static features are moved across a stationary grid.

This problem can be even worse in MHD. Gardiner & Stone (2005) use the advection of a passive magnetic flux loop with negligible internal dynamics as a discriminator for a number of variations of their grid-based MHD module in *ATHENA*, and find only one is capable of both "holding onto" a static flux loop, and moving it diagonally across their 2D grid. Algorithms they rejected were either too diffusive and caused a standing flux loop to dissipate too rapidly, or not diffusive enough, in which case an advected flux loop broke up into stripes orthogonal to the direction of propagation.

To perform the passive flux loop test, I initialize a 128×64 zone 2D Cartesian grid with domain $(x_1, x_2) = (-1.0 : 1.0, -0.5 : 0.5)$ and periodic boundary conditions with $\rho = p = 1$ everywhere. The 3-component of the vector potential is set to

$$A_3 = \begin{cases} 10^{-3}(r_l - r), & r < r_l = 0.3, \\ 0, & r \geqslant r_l, \end{cases}$$

where r is the radial distance from the origin and r_l is the radius of the loop. This gives a uniform azimuthal (about the origin) magnetic field of 10^{-3}. Second-order interpolation is used and, since the dynamical effects are negligible, `qcon` and `qlin` are set to 0.

Figure 21(a) shows the static flux loop (zero advection velocity) after $t = 2$ and 350 time steps, using a Courant number $\mathcal{C} = 0.5$. Qualitatively, this image is identical to the initial conditions (not shown) while quantitatively, the total magnetic energy drops by 0.15% and the maximum magnetic field strength (right at the center of the loop) rises by 0.35% (because of the minute dynamical effects of the non-zero magnetic field).

With an advection velocity $(v_1, v_2) = (2, 1)$, Figure 21(b) shows the flux loop at $t = 2$ ($\mathcal{C} = 0.5$, 681 time steps) after it has been moved diagonally across the grid twice. Here, the flux loop has developed pronounced "stripes" orthogonal to $\vec{v}$, the magnetic energy density has fallen by 36%, and the peak magnetic field strength has risen by 4.3%.

A much improved result can be achieved with a smaller Courant number. Panel (c) of Figure 21 is the same as panel (b) except with $\mathcal{C} = 0.1$ (3325 time steps by $t = 2$). The striping is gone (indeed, there is no striping with $\mathcal{C} = 0.25$ either), although the entire flux loop is flattened slightly in a direction orthogonal to $\vec{v}$ (which can be reduced by further reducing $\mathcal{C}$). In this case, the total magnetic energy density has fallen by 20.5% and the peak magnetic field strength has risen by 1.1%.

There are numerous problems involving passive magnetic fields that `dzeus35` can handle just fine; indeed, the main algorithm in the code to compute the induced electric fields (the *Consistent Method of Characteristics* (*CMoC*); Clarke 1996) was designed specifically for problems in super-Alfvénic turbulence, as discussed in the following section. It would seem, however, that something as "simple" as advecting a passive magnetic flux loop is problematic for `dzeus35`. I note that the "striping instability" manifest in `dzeus35` does not seem to be a diffusion problem as was concluded in Gardiner & Stone (2005) for *ATHENA*, since one needs only to reduce the Courant number to obtain a stable solution without having to increase `qlin` or `qcon`. This, therefore, remains an area of investigation.

6. A 3D TEST PROBLEM: SUPER-ALFVÉNIC TURBULENCE

With few exceptions (notably, Padoan et al. 2004, and references therein, who also use a staggered-mesh code like *ZEUS*), 3D *super*-Alfvénic turbulence has proven to be a vexing problem for many MHD schemes, including the *Method of Characteristics* (*MoC*; Stone & Norman 1992b) available in *ZEUSMP* and `dzeus35`, as well as some recently developed "unsplit Godunov methods" (M. L. Norman 2007, private communication; R. I. Klein 2008, private communication). When MHD algorithms fail in such simulations, they seem to do so catastrophically exhibiting what can only be described as an "explosive instability" in either or both of $\vec{B}$ and $\vec{v}$.

In the case of *ZEUS*' *MoC* algorithm, a completely passive magnetic field can be boosted locally to dynamically important strengths *within a single time step*, thereby destroying the integrity of the simulation. This problem is completely cured by the *CMoC*[9] (Clarke 1996), found in some versions of *ZEUS* including `dzeus35`.

Following Mac Low (1999) (whose simulations are for *trans*- and *sub*-Alfvénic turbulence), a 3D Cartesian grid with a domain $(x_1, x_2, x_3) = (-1.0 : 1.0, -1.0 : 1.0, -1.0 : 1.0)$ and volume $V = 8$ is initialized with an isothermal *weakly* magnetized gas ($\rho = 1$, $c_s = 0.1$, $\vec{B} = 10^{-7}\hat{x}_1$), and a supersonic

[9] The so-called *HSMoC* algorithm described in Hawley & Stone (1995) also cures the explosive instability, though must be run with half the Courant number as *CMoC* to preserve stability in the transport of Alfvén waves.

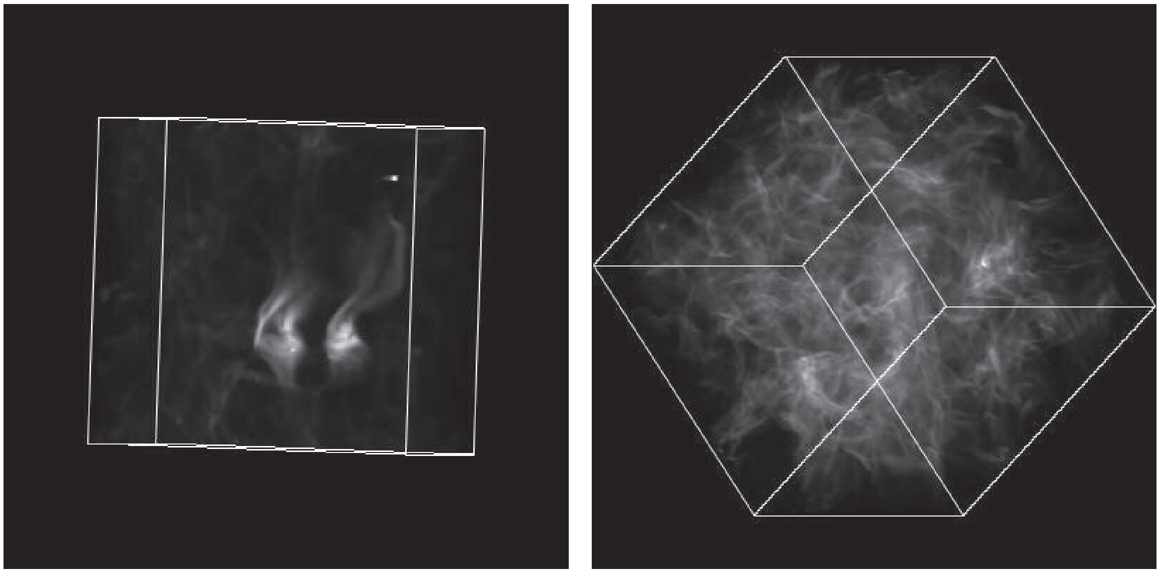

Figure 22. Left panel shows the magnetic pressure at $t = 0.7t_{c_s}$ of a 128^3 *MoC* run. The bright spot near the top right of the box is the "magnetic explosion" that brought the simulation to a halt. The right panel shows the 256^3 *CMoC* run at $t = 2t_{c_s}$ with no signs of the explosive instability.

ultra-Alfvénic turbulent velocity distribution whose initial rms velocity is 5 ($M = 50$, $M_A = 5 \times 10^7$). The turbulent velocity profile is set in Fourier space, where the independent variable is $\vec{k} = (k_x, k_y, k_z)$. Each component, k_i, is an integer $1 \leqslant k_i \leqslant n_i$, where n_i is the number of zones in the i-direction. If $V_i(\vec{k}) = \mathrm{FT}[v_i(\vec{r})]$ is the Fourier transform of the i-component of the velocity, it is a complex number whose real and imaginary parts are given by

$$V(\vec{k}) = \begin{cases} k^\alpha \mathcal{N}\big(\cos(2\pi\mathcal{R}),\ \sin(2\pi\mathcal{R})\big) & k_{\min} \leqslant k \leqslant k_{\max}, \\ (0,\ 0) & \text{otherwise}, \end{cases}$$

where $\alpha = 5/3$, $k = |\vec{k}|$, $0 < \mathcal{R} \leqslant 1$ is a random deviate, and $\mathcal{N}$ is a normal deviate with unit standard deviation. So that $\mathrm{FT}^{-1}[\mathrm{V_i}(\vec{\mathrm{k}})]$ is real, $V_i(-\vec{k}) = V_i^*(\vec{k})$ (complex conjugate) is enforced. Power is restricted to a specific range in k by $k_{\min}$ and $k_{\max}$, chosen here as 3 and 4, respectively. In this way, three arrays $V_i(\vec{k})$ are filled and their inverse FTs are taken to yield three real velocity components, $v_i(\vec{r})$, which are normalized to give the desired initial rms velocity of 5.

To maintain the turbulence, one counters the numerical dissipation of kinetic energy (modeled here as $e^{-t/\tau_{\mathrm{d}}}$, where τ_{d} is the e-folding time for numerical dissipation of kinetic energy) with a driving power, P_{dr} (chosen here to be 1), applied at each time step. One can then show that the kinetic energy asymptotes to $K_{\mathrm{asym}} = P_{\mathrm{dr}}\tau_{\mathrm{d}}$. The driving power is applied by adding to the velocity components a fraction of the initial velocity arrays at each time step, δt, so that $\delta K = P_{\mathrm{dr}}\,\delta t$. FORTRAN subroutines to initialize and drive the turbulence are available from the ZEUS-3D Web site[3].

The left panel of Figure 22 shows the magnetic energy density (integrated along the line of sight) in a 128^3 run after $t = 14 = 0.7t_{c_s}$ (where $t_{c_s} = 20$ is the sound-crossing time) using *ZEUS*' *MoC* algorithm. A 256^3 run crashed almost immediately, and is not shown. The bright spot in the top right of the image is a magnetic field "explosion" in which the local Alfvén Mach number, M_A, is brought from $\sim 10^6$ to unity *within a single time step*. This causes an immediate evacuation of the zone (total pressure is now much higher than in the neighboring zones in which the explosion did *not* happen), reducing the local Alfvén time step to near zero and bringing the simulation to a halt. Of course, such spikes in the magnetic field are completely unphysical.

The right panel of Figure 22 represents a 256^3 run at $t = 40 = 2t_{c_s}$ using *CMoC* with no evidence of the explosive instability whatever. Indeed, a 128^3 simulation taken to $t = 160 = 8t_{c_s}$ remains perfectly stable as well, and shows that the total magnetic energy grows *exponentially* until it saturates (Figure 23(a)), in this case at $t \sim 110 = 5.5t_{c_s}$. For the reason explained in Appendix C, saturation occurs when the rms Alfvén speed is comparable to the constant sound speed (Figure 23(b)), and *not* when the energies are in equipartition. Figure 23(c) shows the Fourier spectra of various variables for the 256^3 simulation at $t = 40 = 2t_{c_s}$, well before the magnetic energy density has saturated. The profiles of the density and kinetic energy are well approximated by $-3/2$ power laws (e.g., Maron & Goldreich 2001) inside a modest "inertial range" of $4 \leqslant k \leqslant 32$. True Kolmogorov $-5/3$ spectra are not normally observed until resolutions of 1024^3 (e.g., Müller & Biskamp 2000), 2048^3, and now 4096^3 (Kaneda & Ishihara 2006). Additional figures and animations of these simulations can be found at the *ZEUS-3D* Web site[3].

These simulations are not presented to study super-Alfvénic turbulence at a physically realistic resolution, but as a 3D test problem. As such, the exponential growth rate of the magnetic energy is a practical comparator among algorithms. One can see from Figure 23(a) that while the magnetic field remains passive, $E_B(t) \sim e^{t/\tau_B}$, where τ_B is the e-folding time of the magnetic energy (see Appendix C for a theoretical explanation for the exponential growth of a weak field). Measured directly from Figure 23(a), $\tau_B \sim 4.7$. At other resolutions (not shown), $\tau_B \sim 15$ (32^3) and 4.9 (64^3). The 256^3 simulation was not taken to saturation, and thus no value for τ_B is quoted from it. Clearly, a threshold was passed between the 32^3 and 64^3 simulations, where the former is under-resolved in some fundamental way. The fact that the e-folding times for the 64^3 and 128^3 simulations agree ($\sim 5 = 0.25t_{c_s}$) suggests that convergence to a physical solution has begun, although these resolutions are insufficient to establish a credible inertial range.

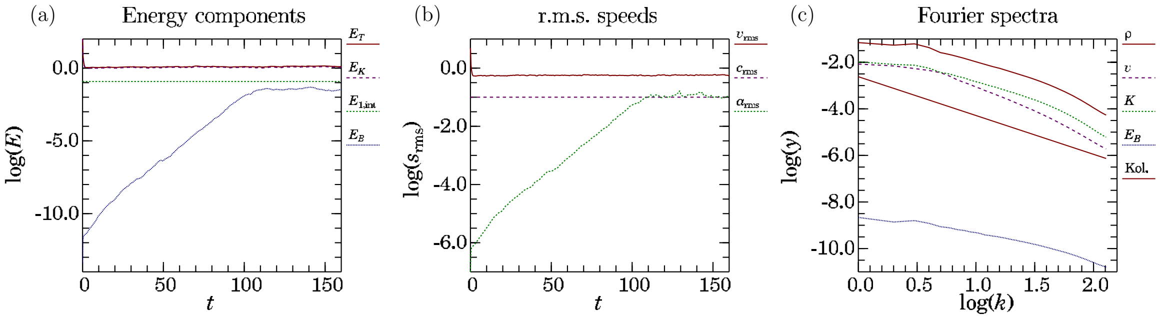

Figure 23. (a) Energy density integrated over the grid as a function of time. Solid line is total energy, long dashed line is kinetic energy (∼1), short dashed line is the constant thermal energy (0.12), and dotted line is magnetic energy (saturating at ∼0.038, up from its initial value of 4×10^{-14}). (b) rms velocities as a function of time. Solid line is flow speed (∼0.5), long dashed line is the constant sound speed (0.1), and short dashed line is the rms Alfvén speed (starting at 10^{-7}, saturating at ∼0.1). (c) Fourier spectra for the 256^3 simulation at $t = 40$. While the turbulence is driven in the range $3 \leqslant k \leqslant 4$, spectra approximate power laws in $4 < k < 32$ ("inertial range"). Spectral indices as measured directly from the figure are: ρ (solid fine line): $-3/2$; v (long dashed line): -2; K (kinetic energy density, short-dashed line): $-3/2$; E_B (magnetic energy density): -1. The heavy solid line is the Kolmogorov $-5/3$ spectrum included for comparison. Note that at $t = 40$, E_B has another 6 orders of magnitude to grow before reaching saturation.

(A color version of this figure is available in the online journal.)

Other practical comparators include the saturation level of the magnetic energy and τ_d. When the rms Alfvén speed reaches the sound speed, the magnetic energy saturates at $\log E_B \sim -1.6$ for the 64^3 simulation, and ~ -1.4 for the 128^3 simulation. Since $K_{\rm asym} \sim 1$, the *e*-folding time for numerical dissipation is $\tau_d = K_{\rm asym}/P_{\rm dr} \sim 1$. Finally, and even though a Kolmogorov spectrum was not actually obtained, the spectra given in Figure 23(c) with the power indices in the caption may also serve as useful comparators.

7. DISCUSSION

Contrary to claims made elsewhere, I have shown that—used as designed—*ZEUS* can obtain satisfactory results to virtually any 1D Riemann MHD test problem including the Sedov blast wave in spherical coordinates. Discontinuities are tracked with similar accuracy (propagation speed, upwind levels, number of zones required for capture) as any published results for fully upwinded schemes. "Rarefaction shocks" remain a technical problem for the algorithm and appear in some pedagogical test problems, but I have argued that these are very unlikely to cause any concern for published or future results generated by *ZEUS*.

These test results were achieved, in part, because of two algorithmic augmentations for *ZEUS* which I describe in detail. These include a modification to CA (Norman et al. 1980) in which CA is no longer applied to the energy equation, and an algorithm for the total energy equation that has been modified somewhat from the original scheme described in Clarke (1996).

Multi-dimensional tests are also presented which show that *ZEUS* does not suffer from the grid biases exhibited by some upwinded schemes that ignore compressional magnetic terms (Gardiner & Stone 2005). *ZEUS* satisfies the solenoidal condition trivially, and can be (largely) inoculated against negative pressures even when the conservative total energy equation is solved. Finally, *ZEUS* can be used successfully for the important problem of super-Alfvénic turbulence which has vexed many upwinded schemes.

One of *ZEUS*' great strengths is its ability to incorporate additional physics (e.g., radiation, self-gravity, viscosity, etc.) without revising the underlying MHD scheme or compromising its accuracy. Still, each algorithm including *ZEUS*, however robust and stable it may be in the realm in which it is developed, can be "broken" by a sufficiently well-aimed counterexample. The important question is, are such counterexamples relevant to the intended astrophysical application? More particularly, is mechanical energy conservation or a positive-definite pressure paramount? Is multi-dimensional stability important, or will simulations be performed in only one or possibly two dimensions? Is the order of convergence of the algorithm critical and is the stated order of convergence in 1D test problems the same as that delivered in multi-dimensional applications? Answers to these questions will help determine which code, among the many available, is best-suited to a particular problem.

I thank Jim Stone and Elizabeth Tasker for useful comments on the manuscript. Development of *ZEUS* and, in particular, `dzeus35` is supported, in part, by the Natural Sciences and Engineering Research Council (NSERC). Some of the simulations were performed using the facilities at ACEnet, funded by the Canada Foundation for Innovation (CFI), the Atlantic Canada Opportunities Agency (ACOA), and the provinces of Nova Scotia, Newfoundland and Labrador, and New Brunswick.

APPENDIX A

DIFFERENCING THE TOTAL ENERGY EQUATION

Figure 24 reminds the reader of the variable locations on a staggered grid. I use a "generic" set of coordinates, $(x_1(i), x_2(j), x_3(k))$, which, in `dzeus35`, could be Cartesian, cylindrical, or spherical polar. For simplicity, I omit the metric factors that account for the curvilinear coordinates, and thus the discussion here is peculiar to Cartesian coordinates.[10] The indices (i, j, k) label the zone center as well as all zone faces, edges, and corners that are half a zone closer in one or more directions to the (left, bottom, back) corner of the grid.

In `dzeus35`, the total energy equation is *unsplit*, at least in 1D. In multi-dimensions, the fluxes are accounted for in a directionally and planar-split fashion, as is the design of *CMoC*. In difference form and considering only the 1-derivatives, Equation (2) becomes

$$e_{\rm T}^{n+1}(i,j,k) = e_{\rm T}^{n}(i,j,k) - \frac{\delta t^n}{\delta x_1}\bigg(\mathcal{F}_e^{n+\frac{1}{2}}(i+1,j,k) - \mathcal{F}_e^{n+\frac{1}{2}}(i,j,k) + \mathcal{G}_e^{n+\frac{1}{2}}(i+1,j,k) - \mathcal{G}_e^{n+\frac{1}{2}}(i,j,k)\bigg),$$

[10] The reader may download the code (see footnote 3) or refer to Stone & Norman (1992a) to see how the metric factors are incorporated.

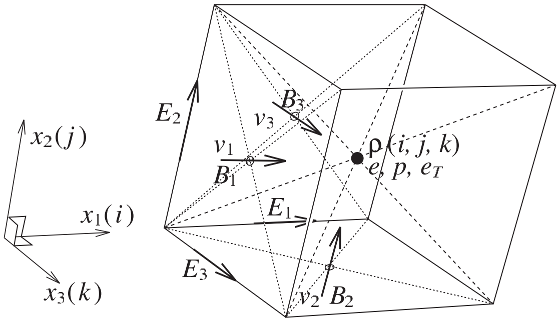

Figure 24. (i, j, k)th zone on a staggered grid, showing in each case the locations of the (i, j, k)th element of the zone-centered scalars $(\rho, e, p, e_{\rm T})$, l-face-centered l-components of the primary vectors $(\vec{v}, \vec{B})$, and the l-edge-centered l-components of the secondary vectors $(\vec{E} = \vec{v} \times \vec{B})$, where $l = 1, 2, 3$. The dashed lines diagonally through the zone and dotted lines across the i-, j-, and k-faces are there to guide the eye.

where the superscript n indicates the nth time step, and where

$$\mathcal{F}_e^{n+\frac{1}{2}}(i, j, k) = \left[\left(\overline{e_{\rm T}}^i + \overline{p}^i - \overline{p_B}^i + \overline{Q_{11}}^i\right) v_1\right]_{(i,j,k)}, \quad \text{(A1)}$$

$$\mathcal{G}_e^{n+\frac{1}{2}}(i, j, k) = \left[\langle E_2\rangle_3 \langle \overline{B_3}^i\rangle_3 - \langle E_3\rangle_2 \langle \overline{B_2}^i\rangle_2\right]_{(i,j,k)}, \quad \text{(A2)}$$

are the *compressional* energy fluxes ($\mathcal{F}_e$) and *transverse* (Poynting) energy fluxes ($\mathcal{G}_e$) respectively[11]. The superscript $n + \frac{1}{2}$ indicates time-centering, as required for a finite-*volume* code. The artificial viscosity, Q, is a diagonal tensor whose $(1, 1)$-element is given by

$$Q_{11}(i, j, k) = \rho(i, j, k)\, \delta_1 v_1 (q_2 \min(0, \delta_1 v_1) - q_1\, c_{\rm s}(i, j, k)), \quad \text{(A3)}$$

where $\delta_1 v_1 = v_1(i+1, j, k) - v_1(i, j, k)$, $c_{\rm s} = \sqrt{\gamma p/\rho}$ is the sound speed, and where q_1 and q_2 are coefficients for the linear and quadratic viscous terms corresponding, respectively, to the `dzeus35` parameters `qlin` and `qcon` defined in Section 2. Thus, Q_{11} is a cleanly zone-centered quantity, with no averages necessary to make it so.

For the compressional fluxes in Equation (A1), "overbars" with leading indices ($\overline{}^i$) indicate monotonic upwinded (in the flow velocity) time-centered interpolations of the quantity in the i-direction to the face center. Thus, $\overline{p}^i$ is the i-interpolation of the zone-centered p to the 1-face (Figure 24), where it is needed to construct the 1-flux; see Clarke (1996) for details on how the interpolations are performed. Note that the total energy density ($e_{\rm T}$), pressure (p), magnetic pressure ($p_B = B^2/2$), and the artificial viscous stresses (Q_{11}) are separately interpolated before they are added and multiplied with v_1 (which is already centered at the 1-face and thus needs no interpolation). The decision to sum the interpolations (including treating $e_{\rm T} - p_B$ as two separate terms instead of combining them as $e_{\rm H}$; see Equation (2)) rather than interpolate the sums was arrived at by direct experimentation. While the latter is computationally more economical, it causes undue ringing in many of the 1D tests problems in Section 4. Conversely, the former—in which the monotonizers can work directly on $e_{\rm T}$—results in the clean solutions presented.

For the transverse fluxes in Equation (A2), the induced electric field components are given by

$$E_2(i, j, k) = \overline{v_1}^k\, \overline{B_3}^i - \overline{v_3}^i\, \overline{B_1}^k,$$
$$E_3(i, j, k) = \overline{v_2}^i\, \overline{B_1}^j - \overline{v_1}^j\, \overline{B_2}^i, \quad \text{(A4)}$$

where the overbars in Equations (A2) and (A4) indicate monotonic upwinded (in both Alfvén characteristics) time-centered interpolations of the quantity in the indicated direction (i, j, or k) to the edge center. Thus, $\overline{v_3}^i$ is the i-interpolation of the 3-face-centered v_3 to the 2-edge (Figure 24), where it is needed to construct E_2. To get face-centered fluxes then requires the additional step of two-point averages in the appropriate direction, as indicated by the $\langle\,\rangle_l$ notation in Equation (A2), where $l = 2, 3$. Note that I have explicitly used the product of the averages (e.g., $\langle E_2\rangle_3 \langle \overline{B_3}^i\rangle_3$) rather than the equally plausible average of the product (e.g., $\langle E_2\, \overline{B_3}^i\rangle_3$). I know of no discriminator to choose between these possibilities—1D tests give identical results for both since the averaging is moot. In `dzeus35`, I have arbitrarily chosen the former.

In evaluating E_2 in Equation (A4), i-interpolations of v_3 and B_3 and k-interpolations of v_1 and B_1 are required. Unique to the *CMoC* algorithm, these four interpolations are performed simultaneously and implicitly so that the bases of the characteristics at which the quantities are interpolated are placed using characteristic speeds computed from the same interpolated values. This step, necessary to prevent the magnetic field explosions discussed in Section 6, is technically rather complex. It is described at length in Clarke (1996) and rendered in FORTRAN in the `CMOC*` (`*`=1,2,3) routines in `dzeus35`.

The algorithm described is for 1-fluxes only. A completely analogous algorithm can be constructed for the 2-fluxes by permuting $(i, j, k) \rightarrow (j, k, i)$ and $(1, 2, 3) \rightarrow (2, 3, 1)$, and then for the 3-fluxes by permuting $(j, k, i) \rightarrow (k, i, j)$ and $(2, 3, 1) \rightarrow (3, 1, 2)$. Whether *CMoC* or some other edge-centering technique is used to evaluate the induced electric field, $\vec{E}$, and the edge-centered velocity and magnetic field components used to compute it, these quantities can be used for numerous purposes in a staggered-grid algorithm, including the following:

1. evaluating the Poynting flux in the total energy equation, as described above;
2. updating the magnetic field with the induction equation ($\partial_t \vec{B} + \nabla \times \vec{E} = 0$);
3. evaluating the *transverse* component of the Lorentz force ($\vec{J} \times \vec{B} = (\nabla \times \vec{B}) \times \vec{B} = -\nabla(B^2/2) + (\vec{B} \cdot \nabla)\vec{B}$, where the second term is the transverse component);
4. providing the interpolated velocities necessary to perform the *transverse* momentum transport (e.g., $\partial_1 s_2$) steps.

This is how `dzeus35` is designed. Thus, while *CMoC* may account for half of the cpu used per MHD cycle, its products are well-used and renders `dzeus35` an efficient and fully Eulerian code. By comparison, some versions of *ZEUS* still use the (*HS*)*MoC* scheme in which a hybrid Eulerian–Lagrangian step is used to update the magnetic field and Lorentz forces, and a separate step is then taken to perform the transverse momentum transport (Stone & Norman 1992b; Hawley & Stone 1995). My own tests show that despite its algorithmic complexity, the fully Eulerian *CMoC* step is significantly *less* computationally expensive than (*HS*)*MoC*.

[11] *Compressional* terms contain derivatives of the form $\partial_i f$ or $\partial_i w_i$, where f is a scalar and w_i is a vector component. *Transverse* terms include cross-derivatives ($\partial_i w_j$, $i \neq j$) only.

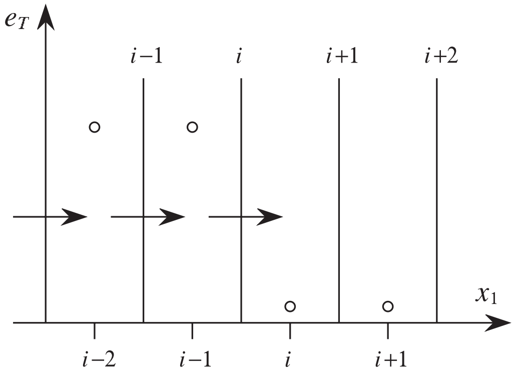

Figure 25. Depicted are the few zones surrounding a 1D shock propagating into a cold and quiescent medium (essentially zero $e_{\rm T}$), where the shock has reached face *i* but not zone-center *i*. Arrows represent velocity vectors while the level of the open circles indicate total energy density.

A.1. Curtailing Negative Pressures

One of the first problems encountered when differencing the total energy equation is evaluating the thermal pressure, given by

$$p = (\gamma - 1)e = (\gamma - 1)\left(e_{\rm T} - \tfrac{1}{2}\rho v^2 - \tfrac{1}{2}B^2\right).$$

Both $e_{\rm T}$ and p are zone-centered, and thus so must be the kinetic and magnetic energy densities ($e_{\rm K}$ and e_B). It is therefore necessary to construct $e_{\rm K}$, for example, from face-centered vector components, and how one does this can have enormous effects on the quality of the algorithm.

An Occam's razor approach might be to take two-point averages. Thus, for v_1,

$$\langle v_1\rangle_i = \tfrac{1}{2}(v_1(i+1,j,k) + v_1(i,j,k)), \qquad \text{(A5)}$$

with the other vector components done likewise. For the 200^3 Sedov blast wave problem described in Section 3, averaging the velocities to construct the kinetic energies caused some 350 *million* negative pressures, or about 1 negative pressure per 100 zone updates.

The origin of the negative pressures can be understood from Figure 25, which depicts the few zones across a 1D shock moving through a quiescent medium. The levels of the open circles represent the values of the zone-centered $e_{\rm T}$, and the lengths of the arrows represent the magnitudes of the face-centered v_1. Here, I have depicted the shock as having just reached face i (and thus $v_1(i)$ is non-zero), but not having reached zone center i (and thus $e_{\rm T}(i)$ remains at the floor level). *By averaging v_1, the zone-centered estimate of $e_{\rm K}(i)$ includes the shock-accelerated velocity, $v_1(i)$, and yet $e_{\rm T}(i)$ itself has not yet been affected by the shock.* Thus, $e_{\rm T}(i) - e_{\rm K}(i)$ becomes negative at the base of strong shocks where shock acceleration boosts the kinetic energy over the quiescent internal energy.

The fix is clear. Considering data only from the current time step, zone center i cannot yet be aware of the *upwind* velocity, $v_1(i)$. On the other hand, the *downwind* velocity, $v_1(i+1)$ (depicted here as 0), is representative of what v_1 must have been at zone-center i in the very recent past. Thus, a zone-centered estimate of the velocity that obeys the principle of causality is a *downwinded* interpolation whose order of accuracy can be controlled much like the *upwinded* interpolations used to construct the time-centered fluxes in the transport step (e.g., Clarke 1996). Thus, a "donor-cell" downwinded interpolation is given by

$$^{i}\overline{v_1} = \begin{cases} v_1(i,j,k), & \langle v_1\rangle_i \leqslant 0, \\ v_1(i+1,j,k), & \langle v_1\rangle_i > 0, \end{cases} \qquad \text{(A6)}$$

where $\langle v_1\rangle_i$ is given by Equation (A5), and where "overbars" with trailing indices ($\overline{}^{i}$) denote downwinded interpolation. A second- and third-order downwind interpolation can be constructed from piecewise linear (van Leer 1977) or quadratic (Colella & Woodward 1984) interpolation functions, and details are omitted for brevity. `dzeus35` uses second-order downwinded interpolations to estimate zone-centered velocities[12] when computing the pressure from the total energy density. For the 200^3 Sedov blast wave problem, this reduces the number of negative pressures from 350 million to *zero*.

Still, this does not prove that downwinding renders the total energy equation positive-definite, and one must be prepared to deal with negative pressures should they arise. In `dzeus35`, I simply reset them to a small positive quantity, keep track of where and when such resets occur, and report the accumulated internal energy added to the grid[13] at the end of the simulation. An alternate approach, known by some as the *dual-energy method*, solves *both* the internal and total energy equations, and computes the pressure from the internal energy equation whenever the total energy equation yields a negative pressure (e.g., *ENZO*; O'Shea et al. 2004). Such a scheme still effectively adds an arbitrary amount of internal energy density to troublesome zones and, unlike simply setting negative pressures to a small positive value, can lead to square wave pressure profiles over time, as the pressure switches back and forth between the near-zero values from the total energy equation, and the possibly not-so-small values from the internal energy equation.

It is worth noting that the 1D shock-tube tests in Section 4 cannot distinguish between averaging and downwinding the velocities; both sets of solutions are equal in quality with no negative pressures in either case. However, the "other" option, namely, *upwinding* (as one might naïvely jump to given its role in evaluating fluxes) is clearly ruled out by the 1D tests, as it excites severe ringing in several of the problems.

Finally, while downwinding avoids negative pressures much better than averaging in the Sedov blast wave problem, there seems to be no measurable difference in the two solutions otherwise. Resetting all 350 million negative pressures back to essentially zero required adding the equivalent of ∼1.8% of the internal energy density on the grid and created a final state indistinguishable in any other way from the downwinded solution (e.g., advance of shock, isotropy, peak levels behind shock, etc.). Resetting a negative pressure back to zero is a stabilizing act. It can only *reduce* the local pressure gradient, thereby accelerating the fluid *less*, rendering the next estimate of the kinetic energy *lower* and thus making it *less* likely for a negative pressure to reoccur. Thus, while it is clearly well to avoid negative pressures where one can, it is not necessarily a bad thing simply to reset them to zero when one must.

[12] Magnetic fields are still averaged to the zone-centers since downwinding them relative to $\vec{v}$ causes negative pressures in the MHD blast wave (Section 5.2). It is likely $\vec{B}$ needs to be downwinded relative to the Alfvén speeds.

[13] Note that this addition is made to the pressure only, and not to the total energy. Thus, the effects are local, not global.

APPENDIX B

REMOVING CONSISTENT ADVECTION FROM THE ENERGY EQUATION

In this Appendix, I outline a simple procedure to existing *ZEUS* codes to disengage CA from the (internal) energy equation. Discussion is necessarily specific to the structure and coding conventions of most *ZEUS* codes.

In the "transport" step, and in particular in routine `TRANX1`, the energy is updated by statements resembling:

```
mflx1(i+1,j,k) = v1    (i+1,j,k) * dtwid1(i+1,j,k) * dt
dflx1(i+1    ) = mflx1(i+1,j,k) * dar1b (i+1)
eflx1(i+1    ) = dflx1(i+1    ) * etwid1(i+1,j,k)
e    (i  ,j,k) = e(i,j,k) + ( eflx1(i) - eflx1(i+1) ) / dvl1a(i)
```

It is the use of the mass flux (`mflx1`) to construct the 1-fluxes of momentum and energy that is the embodiment of CA in *ZEUS*. In this snippet of coding, the mass flux, with units *mass per area*, is the product of v_1, the interpolation of density to the 1-face (`dtwid1`), and the time step. The density flux (`dflx1`) is then the mass flux times the area of the 1-face (`dar1b`), with units *mass*. Finally, the energy flux (`eflx1`) is the product of the density flux and the 1-interpolation of the *specific energy*, e/ρ (and *not* e), giving a quantity with units *energy*. The difference in energy fluxes on either side of the zone divided by the zone volume (`dvl1a`) then gives the change in energy density, e, resulting from 1-transport during the time step.

To remove CA from the algorithm, one must change the third line evaluating `eflx1` to:

```
eflx1(i+1) = v1(i+1,j,k) * etwid1(i+1,j,k) * dt * dar1b(i+1)
```

where `etwid1` is now the 1-interpolation of the *energy density*, e (and *not* e/ρ). This last step is critical, and will require another small change earlier in the routine. Before the triple do-loop (double do-loop for *ZEUS-2D*) containing this piece of coding, there should be a call to the interpolator, e.g., `X1ZC3D`, which will include in its argument list the *specific energy*. Change this to the *energy density*, e.

Similar modifications to the other transport routines, `TRANX2` and `TRANX3`, are necessary to complete the task for 3D. Note that these changes do nothing to interfere with CA from being applied to the momentum fluxes (by use of the 3D arrays `mflx1`, etc.) which was its original intent and for which there is ample evidence of benefit.

APPENDIX C

EXPONENTIAL GROWTH OF A WEAK MAGNETIC FIELD

To understand why the magnetic energy, $E_B = \int \frac{1}{2} B^2\, dV$, should undergo exponential growth in a sustained super-Alfvénic turbulent medium, start with the induction equation:

$$\begin{aligned}\partial_t \vec{B} &= \nabla \times (\vec{v} \times \vec{B}) = (\vec{B} \cdot \nabla)\vec{v} - (\vec{v} \cdot \nabla)\vec{B} - \vec{B}(\nabla \cdot \vec{v}) \\ &\Rightarrow d_t \vec{B} = (\vec{B} \cdot \nabla)\vec{v} - \vec{B}(\nabla \cdot \vec{v}),\end{aligned}$$

where $\nabla \cdot \vec{B} = 0$ has been explicitly assumed, and where $d_t = \partial_t + \vec{v} \cdot \nabla$. Thus,

$$\vec{B} \cdot d_t \vec{B} = \tfrac{1}{2} d_t B^2 = \vec{B} \cdot [(\vec{B} \cdot \nabla)\vec{v}] - B^2(\nabla \cdot \vec{v}). \qquad \text{(C1)}$$

To get an evolution equation for E_B, we integrate Equation (C1) over the volume of the grid, V. Doing this, the first term on the right-hand side vanishes to within statistical noise since, in a turbulent medium, the integrand is equally likely to be positive or negative with no bias from the sign to its magnitude. The second term, however, is different. When integrated over V, it represents a grid sum of $\nabla \cdot \vec{v}$ weighted by B^2 which, for a passive magnetic field, is typically greater in regions of compression (where $\nabla \cdot \vec{v} < 0$) than in regions of expansion (where $\nabla \cdot \vec{v} > 0$). Thus, the volume-integrated right-hand side of Equation (C1) is positive and proportional to E_B, yielding the observed exponential growth of E_B with the e-folding time, τ_B, varying inversely with the weighted sum of $\nabla \cdot \vec{v}$. Therefore, the higher the rms velocity of the turbulence, the greater the average magnitude of $\nabla \cdot \vec{v}$, and the shorter τ_B.

When the plasma beta reaches order unity (i.e., $a_{\rm rms} \sim c_{\rm s}$), the magnetic field ceases to be slave to the hydrodynamics, and the correlation between B^2 and the sign of $\nabla \cdot \vec{v}$ disappears. Thus, the volume integral of the second term in Equation (C1) becomes zero (to within statistical noise) as well, rendering E_B independent of time thereafter. This is precisely the behavior observed in Figures 23(a) and (b).

REFERENCES

Balsara, D. S. 2001, J. Comput. Phys., 174, 614
Balsara, D. S., & Spicer, D. S. 1999, J. Comput. Phys., 149, 270
Brio, M., & Wu, C. C. 1988, J. Comput. Phys., 75, 400
Campbell, J., & Shashkov, M. 2001, J. Comput. Phys., 172, 739
Clarke, D. A. 1996, ApJ, 457, 291
Colella, P., & Woodward, P. R. 1984, J. Comput. Phys., 54, 174
Dai, W., & Woodward, P. R. 1994, J. Comput. Phys., 75, 400
Falle, S. A. E. G. 2002, ApJ, 577, L123 (F02)
Fryxell, B., et al. 2000, ApJS, 131, 273
Gardiner, T. A., & Stone, J. M. 2005, Comput. Phys. Commun., 89, 127
Hawley, J. F., & Stone, J. M. 1995, J. Comput. Phys., 205, 509
Kaneda, Y., & Ishihara, T. 2006, J. Turb., 7, 1
Mac Low, M.-M. 1999, ApJ, 524, 169
Maron, J., & Goldreich, P. 2001, ApJ, 554, 1175
Müller, W.-C., & Biskamp, D. 2000, Phys. Rev. Lett., 84, 475
Norman, M. L., Wilson, J. R., & Barton, R. 1980, ApJ, 239, 968
O'Shea, B. W., Bryan, G., Bordner, J., Norman, M. L., Abel, T., & Harkness, R. 2004, in Springer Lecture Notes on Computational Science and Engineering, Adaptive Mesh Refinement–Theory and Applications, ed. A Kritsuk, T. Plewa, T. Linde, & V. G. Weirs (Berlin: Springer)
Padoan, P., Jimenez, R., Nordlund, Å., & Boldyrev, S. 2004, Phys. Rev. Lett., 92, 1
Richtmyer, R. D., & Morton, K. W. 1995, Difference Methods for Initial Value Problems (New York: Wiley)
Ryu, D., & Jones, T. W. 1995, ApJ, 442, 228 (RJ95)
Shultz, W. D. 1964, J. Math. Phys., 5, 133
Stone, J. M. 2009, in ASP Conf. Ser. 406, Numerical Modeling of Space Plasma Flows: AstroNUM 2008, ed. N. V. Pogorelov et al. (San Francisco, CA: ASP), 277
Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon, J. B. 2008, ApJS, 178, 137
Stone, J. M., & Norman, M. L. 1992a, ApJS, 80, 753
Stone, J. M., & Norman, M. L. 1992b, ApJS, 80, 791
Stone, J. M., & Norman, M. L. 1992c, ApJS, 80, 819
Tasker, E. J., Brunino, R., Mitchell, N. L., Michielson, D., Hopton, S., Pearce, F. R., Bryan, G. L., & Theuns, T. 2008, MNRAS, 390, 1267 (T08)
van Leer, B. 1977, J. Comput. Phys., 23, 276
von Neumann, J., & Richtmyer, R. D. 1950, J. Appl. Phys., 21, 232