

Computational Magnetohydrodynamics

Gianfranco Grillo

Cavendish Laboratory, Department of Physics, J J Thomson Avenue, Cambridge. CB3 0HE

Abstract

We present a two-dimensional Riemann-based numerical solver for the compressible, unsteady, and non-relativistic ideal magnetohydrodynamics (MHD) equations that achieves second order in space and time accuracy via the MUSCL-Hancock scheme. The solver minimizes deviations from the divergence constraint by using the hyperbolic divergence cleaning technique. Validation is performed using standard MHD tests from the numerical MHD literature. We discuss test results and possible solver improvements.

1. Introduction

The numerical solution of the magnetohydrodynamics (MHD) equations has historically been an area of extensive research, primarily with the purpose to model plasma behavior. Although similar in many ways to a gas, the motion of particles in a plasma is dominated by electric and magnetic fields, as opposed to the motion of gas particles which is driven by pressure. Modelling the behavior of plasma has applications in astrophysics, atmospheric science, and nuclear fusion energy, among other fields; this wide range of applicability, as well as the continuously increasing availability of processing power, has sustained research interest in numerical plasma simulations. Although modern solvers differ in a number of ways, many tackle the problem using a finite volume approach coupled with a Riemann problem based method of calculating intercell fluxes, a spatial reconstruction technique to achieve higher order accuracy, and an additional algorithm to enforce the divergence free constraint on the magnetic field.

In this work, we present a solver for the equations of ideal, non-relativistic MHD that is based on these principles. We employ an HLLC Riemann solver originally presented by Toro [1] and modified for use in MHD by Li [2] in conjunction with the MUSCL-Hancock scheme [3] and the hyperbolic divergence cleaning technique of Dedner [4] to construct an ideal MHD solver that is second order accurate in space and time and minimizes deviations from the divergence constraint for the magnetic field. The article is structured as follows. In Section 2, we provide a brief introduction to the plasma theory and a short literature review of previous research

in computational MHD. In Section 3 we describe the mathematical equations of ideal MHD and present and discuss the numerical techniques used in our implementation of a two-dimensional ideal MHD solver. In Section 4, we validate our solver implementation via a series of tests sourced from the numerical MHD literature and discuss our algorithm's accuracy. Finally, in Section 5 we summarize our findings and suggest possible ways of improving our solver.

2. Plasma Theory and Literature Review

2.1. Plasma Physics

A plasma is a gas that is significantly ionized, and has a low enough density such that it behaves classically [5]. Like fluids, plasmas can be described macroscopically in terms of an average density, velocity, and pressure; unlike fluids, the motions of plasma particles are dominated by electric and magnetic fields instead of pressure, given that a plasma's constituent particles are electrically charged. This means that magnetic and electric fields are coupled to the plasma's hydrodynamic behavior [6]. Nevertheless, this hydrodynamic behavior is still present, and much can be learned by modelling the plasma by directly combining Euler's equations of compressible fluid dynamics with Maxwell's equations for electrodynamics, especially at lower densities where the plasma's ions and electrons are held together by electrostatic forces and the plasma can be regarded as a conducting fluid. At higher densities, ions and electrons are no longer held together by the Coulomb force and instead can be modelled as two interacting fluids with

different properties. Even higher densities require treatments based on statistical mechanics and kinetic theory [5]. In this work, we restrict ourselves to modelling plasma that can be well described by the equations of ideal, non-relativistic MHD. This is an appropriate description for a low density plasma that has negligible resistivity, is fully ionized, and whose constituent particles move at speeds much lower than the speed of light. This can be an appropriate model for astrophysical applications that happen on large enough scales and the centre of a fusion reactor [6].

2.2. Literature Review

Research into solving the equations of MHD has been historically associated to that of the solution of the compressible hydrodynamics equations, given the mathematical similarities between these two systems. For instance, the equations of MHD reduce to the Euler equations when the magnetic field \mathbf{B} is set to zero, and both systems are (non-strictly) hyperbolic and can be written in conservative form. Therefore, advances in the numerical solution of the MHD equations have often been achieved by extending to MHD the solution methods that were originally applied to the Euler equations, although the presence of additional waves in the MHD system introduces certain subtleties. An additional complication is the need to enforce the divergence condition $\nabla \cdot \mathbf{B} = 0$, which is a problem exclusive to MHD.

An early attempt to numerically simulate the evolution of an ideal MHD system was that of Brio and Wu [7], who derived a one dimensional upwind differencing scheme by linearizing the equations of ideal MHD using a procedure first introduced by Roe [8]. They also introduced the now standard Brio and Wu shock tube test, to be analyzed in subsection 4.1. The result of their simulations were of limited utility given that their scheme was restricted to a single value of the ideal adiabatic index, $\gamma = 2$. Subsequently, Zachary and Colella [9] applied a finite volume algorithm first developed by Bell, Colella, and Trangenstein [10] to compute inter-cell fluxes in accordance with Godunov's scheme [11], described in more detail in subsection 3.3. Zachary and Colella's method achieved higher order accuracy than Brio and Wu and its applicability was not restricted to the case $\gamma = 2$. Dai and Woodward [12] used a Lagrangian approach to construct a nonlinear approximate Riemann solver appropriate for dealing with strong nonlinear waves. The same authors extended their approach to higher order and multiple dimensions in [13].

Additional work has focused on adapting approximate Riemann solvers popular in the hydrodynam-

ics community like HLLC to multidimensional MHD [14, 2, 15] and enforcing the divergence free condition [16, 4, 17, 18]. In later years, a number of research groups have developed and published sophisticated software packages that incorporate aspects of the research approaches mentioned previously. These include ZEUS [19], FLASH [20], RAMSES [21], PLUTO [22], HERACLES [23], Athena [24], AstroBEAR [25], and Enzo [26], among others.

3. Equations of MHD and Numerical Methods of Solution

3.1. Equations of ideal MHD

By combining the Euler equations of compressible hydrodynamics and Maxwell's equations of electromagnetism, and in the non-relativistic limit with negligible resistivity, one derives the equations of ideal MHD [6, 5]. In conservation form and independently of the geometry of the coordinate system, these are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \quad (1)$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot \left[\rho \mathbf{v} \otimes \mathbf{v} + \left(p + \frac{1}{2} B^2 \right) \mathbf{I} - \mathbf{B} \otimes \mathbf{B} \right] = 0, \quad (2)$$

$$\frac{\partial U}{\partial t} + \nabla \cdot \left[\left(U + p + \frac{1}{2} B^2 \right) \mathbf{v} - (\mathbf{v} \cdot \mathbf{B}) \mathbf{B} \right] = 0, \quad (3)$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{B} \otimes \mathbf{v} - \mathbf{v} \otimes \mathbf{B}) = 0, \quad (4)$$

where ρ is the plasma's density, $\mathbf{v} = (v_x, v_y, v_z)^T$ its velocity, $\mathbf{B} = (B_x, B_y, B_z)^T$ the magnetic field, U the total energy, and p the pressure; \mathbf{I} is the identity matrix. The total energy can be expressed as a sum of the internal, kinetic, and magnetic energies:

$$U = \rho \varepsilon + \frac{1}{2} \rho v^2 + \frac{1}{2} B^2. \quad (5)$$

In ideal MHD, one assumes that the plasma is fully ionized. Fully ionized plasma is well described by an ideal gas equation of state, which relates p to $\rho \varepsilon$ via

$$p = \rho \varepsilon (\gamma - 1), \quad (6)$$

where the adiabatic constant $\gamma = 5/3$ for most ideal MHD applications as a result of the thermodynamics of non-interacting monatomic mixtures [6], although we note that not all of the validation tests in section 4 use this value of γ . Finally, the magnetic field needs to satisfy the divergence constraint

$$\nabla \cdot \mathbf{B} = 0. \quad (7)$$

Mathematically, any initial state that satisfies 7 will continue to satisfy the constraint after being evolved by equations 1-4; as a result of discretization, however, this is not necessarily the case with MHD simulations in more than one dimension. This can lead to an accumulation of errors that can cause the simulation to fail and/or give incorrect results, as a consequence, several numerical schemes have been constructed in which this condition is satisfied. We detail one such scheme in subsection 3.6.

The simulations in this work are restricted to one and two dimensions, which leads to a simplification of the evolution equations because the z -derivative (and in the 1D case, the y -derivative) vanish. For numerical purposes, it is useful to write the system 1-4 as a single vector equation in conservation form. For the two dimensional case, and in Cartesian coordinates, the general conservation form is

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{u})}{\partial y} = 0. \quad (8)$$

In our formulation of ideal MHD, the vector of conserved variables has eight components, $\mathbf{u} = (\rho, \rho v_x, \rho v_y, \rho v_z, U, B_x, B_y, B_z)^T$. The corresponding flux vectors are given by

$$\mathbf{f}(\mathbf{u}) = \begin{bmatrix} \rho v_x \\ \rho v_x^2 + p_T - B_x^2 \\ \rho v_x v_y - B_x B_y \\ \rho v_x v_z - B_x B_z \\ (U + p_T) v_x - B_x (\mathbf{v} \cdot \mathbf{B}) \\ 0 \\ v_x B_y - B_x v_y \\ v_x B_z - B_x v_z \end{bmatrix}, \quad (9)$$

$$\mathbf{g}(\mathbf{u}) = \begin{bmatrix} \rho v_y \\ \rho v_x v_y - B_x B_y \\ \rho v_y^2 + p_T - B_y^2 \\ \rho v_y v_z - B_y B_z \\ (U + p_T) v_y - B_y (\mathbf{v} \cdot \mathbf{B}) \\ v_y B_x - B_y v_x \\ 0 \\ v_y B_z - B_y v_z \end{bmatrix}, \quad (10)$$

where we have introduced the total pressure $p_T = p + \frac{1}{2} B^2$.

3.2. Eigenstructure of the Equations

One way of verifying that the conservative system 8 is hyperbolic is by writing it in terms of the primitive variable vector $\mathbf{w} = (\rho, v_x, v_y, v_z, p, B_x, B_y, B_z)^T$ and in

quasilinear form,

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{A}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial x} + \mathbf{B}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial y} = 0. \quad (11)$$

Without loss of generality, we can simplify this further by considering the 1D version of the above along the x -direction [4, 17]. In this case, the divergence constraint requires $\partial B_x / \partial x = 0$, so B_x is a constant parameter and vanishes from the vector of primitive variables, $\mathbf{w}' = (\rho, v_x, v_y, v_z, p, B_y, B_z)^T$. Then the evolution equation is

$$\frac{\partial \mathbf{w}'}{\partial t} + \mathbf{A}'(\mathbf{w}') \frac{\partial \mathbf{w}'}{\partial x} = 0, \quad (12)$$

with the matrix \mathbf{A}' given by

$$\mathbf{A}'(\mathbf{w}') = \begin{bmatrix} v_x & \rho & 0 & 0 & 0 & 0 & 0 \\ 0 & v_x & 0 & 0 & 1/\rho & B_y/\rho & B_z/\rho \\ 0 & 0 & v_x & 0 & 0 & -B_x/\rho & 0 \\ 0 & 0 & 0 & v_x & 0 & 0 & -B_x/\rho \\ 0 & \rho c_s^2 & 0 & 0 & v_x & 0 & 0 \\ 0 & B_y & -B_x & 0 & 0 & v_x & 0 \\ 0 & B_z & 0 & -B_x & 0 & 0 & v_x \end{bmatrix}, \quad (13)$$

where $c_s = \sqrt{\gamma p / \rho}$ is the acoustic sound speed. This matrix has seven eigenvalues corresponding to seven different waves: one entropy wave traveling with speed v_x , two Alfvén waves traveling with speeds $v_x \pm c_a$, two fast magneto-acoustic waves traveling with speeds $v_x \pm c_f$, and two slow magneto-acoustic waves with speeds $v_x \pm c_{sl}$. The Alfvén speed c_a , fast speed c_f , and slow speed c_{sl} , $c_{sl} < c_a < c_f$, are given by the expressions

$$c_a = \frac{|B_x|}{\sqrt{\rho}}, \quad (14)$$

$$c_{f,sl} = \sqrt{\frac{1}{2} \left[c_s^2 + c_a^2 \pm \sqrt{(c_s^2 + c_a^2)^2 - 4 \frac{c_s^2 B_x^2}{\rho}} \right]}. \quad (15)$$

We can repeat the analysis along the y -direction, with the only difference in the resulting eigenvalues and definitions for the wave speeds being that $v_x \rightarrow v_y$ and $B_x \rightarrow B_y$. Thus, the system 8 is hyperbolic; however, because the eigenvalues are not always distinct, the system is not strictly hyperbolic.

3.3. Godunov's Scheme for Hyperbolic PDEs

A popular approach to numerically solving systems of non-linear hyperbolic PDEs in conservation form is Godunov's scheme [11], which commonly operates under a finite volume framework. In the one-dimensional,

Cartesian version of the scheme, the computational domain is discretized into a finite number of computational cells of width Δx , and the conserved variable vector is defined as an integral average at the center of cell i at time t , $\mathbf{u}(x, t) = \mathbf{u}_i^n$ is advanced to $\mathbf{u}(x, t + \Delta t) = \mathbf{u}_i^{n+1}$ per the formula

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t}{\Delta x} \left(\mathbf{f}_{i+\frac{1}{2}}^n - \mathbf{f}_{i-\frac{1}{2}}^n \right), \quad (16)$$

where $\mathbf{f}_{i\pm\frac{1}{2}}^n$ is the intercell numerical flux at the right (+) and left (-) cell interfaces at time t , and is given by

$$\mathbf{f}_{i\pm\frac{1}{2}} = \mathbf{f} \left(\mathbf{u}_{i\pm\frac{1}{2}}^n \right), \quad (17)$$

with $\mathbf{u}_{i\pm\frac{1}{2}}^n$ the solution along the ray $x/t = 0$ of the Riemann problem with initial conditions \mathbf{u}_i^n and $\mathbf{u}_{i\pm 1}^n$ centered at the appropriate cell interface. A Riemann problem is an initial value problem consisting of one or more equations in conservation form with initial conditions given by two constant states separated by a single discontinuity; in this case, the equations are those of ideal MHD, and the discontinuity occurs at the intercell boundary.

The basic Godunov's scheme solves a hyperbolic system of PDEs by solving a Riemann problem at each cell interface in the domain in order to compute the intercell flux, and assumes that the value of the conserved variables on each cell is piecewise constant inside the cell. The timestep of the simulation is restricted by the Courant-Friedrichs-Lewy condition,

$$\Delta t = \frac{c_{cfl} \Delta x}{a_{max}^n}, \quad c_{cfl} \leq 1, \quad (18)$$

where a_{max}^n is the maximum wave speed in the domain at time t^n and c_{cfl} is the dimensionless CFL constant. In the case of ideal MHD, a_{max}^n is given by

$$a_{max}^n = \max_i \left(|v_{x,i}^n| + c_{f,i}^n \right). \quad (19)$$

In practice, simulations usually have $c_{cfl} < 1$ in order to ensure that the scheme is stable even if S_{max}^n is not guaranteed to be completely accurate. Our solver uses a value of $c_{cfl} = 0.8$.

This scheme can be easily extended to two dimensions by solving an additional Riemann problem, now in the y -direction. This can be done in a dimensionally unsplit or a dimensionally split manner; our solver uses the dimensionally split approach because it does not lead to further restrictions on c_{cfl} [6, 27]. We first perform an update exclusively in the x -direction using a two dimensional version of equation 16, obtaining the

fluxes by solving the Riemann problems to at the left and right interfaces of each cell. For each cell, this results in an intermediate state $\bar{\mathbf{u}}_{i,j}^n$,

$$\bar{\mathbf{u}}_{i,j}^n = \mathbf{u}_{i,j}^n - \frac{\Delta t}{\Delta x} \left(\mathbf{f}_{i+\frac{1}{2},j}^n - \mathbf{f}_{i-\frac{1}{2},j}^n \right), \quad (20)$$

where $\mathbf{f}_{i\pm\frac{1}{2},j}^n$ is defined in an analogous way as in equation 17. The intermediate state is now updated in the y -direction by solving the Riemann problems centered at the top and bottom interfaces of each cell to obtain the states $\bar{\mathbf{u}}_{i,j\pm\frac{1}{2}}^n$ and calculating fluxes via $\mathbf{g} \left(\bar{\mathbf{u}}_{i,j\pm\frac{1}{2}}^n \right)$. This gives us the advanced state $\mathbf{u}_{i,j}^{n+1}$,

$$\mathbf{u}_{i,j}^{n+1} = \bar{\mathbf{u}}_{i,j}^n - \frac{\Delta t}{\Delta y} \left[\mathbf{g} \left(\bar{\mathbf{u}}_{i,j+\frac{1}{2}}^n \right) - \mathbf{g} \left(\bar{\mathbf{u}}_{i,j-\frac{1}{2}}^n \right) \right]. \quad (21)$$

The CFL condition is identical to that given in equation 18 except Δx is replaced by $\min(\Delta x, \Delta y)$ and $v_x \rightarrow v_y$ in the expression for a_{max}^n 19, which now also incorporates the additional index j . Our solver uses $\Delta x = \Delta y$ and deals with rectangular domains by using a different number of cells along the x -direction versus the y -direction.

The assumption that states \mathbf{u}_i^n are piecewise constant within cells restricts the basic Godunov method to first order accuracy in space, but several extensions exist that result in higher order schemes that still follow the same basic approach. In this work, we achieve second order accuracy by following the MUSCL-Hancock approach.

3.4. MUSCL-Hancock Scheme

3.4.1. Linear reconstruction

The simplest deviation from the piecewise constant state assumed by the first order Godunov method is that of a state that varies linearly within the cell. By linearly reconstructing the states in each cell in the computational domain, the MUSCL-Hancock scheme achieves second order accuracy in space. Higher order accuracy can be achieved by, for example, performing the reconstruction based on polynomials (e.g. the PPM method [28]) or by using higher order approximations to the derivatives used to perform the reconstruction (e.g. the ENO and WENO schemes [29]).

We illustrate the MUSCL-Hancock approach in the one-dimensional case; the method extends to multiple dimensions following the same dimensionally split approach as in the previous section. A state \mathbf{u}_i^n has a linear reconstruction $\mathbf{u}_i(x)$ given by

$$\mathbf{u}_i(x) = \mathbf{u}_i^n + (x - x_i) \frac{\Delta_i}{\Delta x}, \quad (22)$$

where Δ_i is a weighted average of the backwards and forward differences of \mathbf{u}_i^n ; $\Delta_{i-\frac{1}{2}} = \mathbf{u}_i^n - \mathbf{u}_{i-1}^n$ and $\Delta_{i+\frac{1}{2}} = \mathbf{u}_{i+1}^n - \mathbf{u}_i^n$. We have

$$\Delta_i = \frac{1}{2}(1 + \omega)\Delta_{i-\frac{1}{2}} + \frac{1}{2}(1 - \omega)\Delta_{i+\frac{1}{2}} \quad (23)$$

$$\omega \in [-1, 1].$$

For this application, we use $\omega = 0$, therefore expression 23 simplifies to the central difference, $\Delta_i = \frac{1}{2}(\mathbf{u}_{i+1}^n - \mathbf{u}_{i-1}^n)$. It follows that in this reconstructed state, the boundary values at the left and right interfaces of the cell, \mathbf{u}_i^L and \mathbf{u}_i^R are given by

$$\mathbf{u}_i^L = \mathbf{u}_i^n - \frac{1}{2}\Delta_i, \quad \mathbf{u}_i^R = \mathbf{u}_i^n + \frac{1}{2}\Delta_i. \quad (24)$$

3.4.2. Slope limiting

A well-known issue with this kind of reconstruction is that at discontinuities, the derivatives of some of the elements in \mathbf{u}_i^n are singular, and the resulting reconstructed state will have very steep slopes in those variables, leading to the generation of spurious oscillations that affect the stability and correctness of the simulation [30, 27]. In order to mitigate this issue, the MUSCL-Hancock scheme applies what is known as a slope limiter to the quantity Δ_i before reconstructing the state, reverting the scheme to first order accuracy at these discontinuities. In the general case, this slope limiter ξ is a vector function of \mathbf{r} , which measures the smoothness of the change in one or more variables of \mathbf{u}_i^n . One possible definition is

$$\mathbf{r} = \frac{\Delta_{i-\frac{1}{2}}}{\Delta_{i+\frac{1}{2}}} = \frac{\mathbf{u}_i^n - \mathbf{u}_{i-1}^n}{\mathbf{u}_{i+1}^n - \mathbf{u}_i^n}, \quad (25)$$

in which case the limiter acts in a component by component fashion. The slope limiter ξ can be defined in different ways, with some definitions limiting the slope more aggressively than others. In our tests, we use one of the more compressive versions, the minbee/minmod slope limiter, to ensure stability. We can define it in a component-wise fashion as [30]

$$\xi_k(r_k) = \begin{cases} 0 & r_k \leq 0 \\ r_k & 0 < r_k \leq 1 \\ \min\left(1, \frac{2}{1+r_k}\right) & r_k > 1 \end{cases} \quad (26)$$

where r_k are the components of the vector given by equation 25. We now define the slope limited versions of the reconstructed states at the left and right interfaces of cell i as

$$\bar{\mathbf{u}}_i^L(\mathbf{r}) = \mathbf{u}_i^n - \frac{1}{2}\xi(\mathbf{r}) \odot \Delta_i, \quad \bar{\mathbf{u}}_i^R(\mathbf{r}) = \mathbf{u}_i^n + \frac{1}{2}\xi(\mathbf{r}) \odot \Delta_i. \quad (27)$$

where \odot denotes component-wise multiplication.

Depending on the specific problem to be solved, it can be useful to construct versions of \mathbf{r} that lead to components in \mathbf{u}_i^n being limited based on other vector components. The simplest such approach is to limit the entire vector \mathbf{u}_i^n based on the change of a single component, which is suitably chosen as so to capture the discontinuous character of the solution. In some of our tests in section 4, we limit according to the change in the total energy U . In this scheme, \mathbf{r} becomes r , a scalar, defined by

$$r = \frac{U_i^n - U_{i-1}^n}{U_{i+1}^n - U_i^n}, \quad (28)$$

and thus $\xi(\mathbf{r}) \rightarrow \xi(r)$ is now a scalar function.

3.4.3. Local evolution and the Riemann problem

The MUSCL-Hancock scheme proceeds by locally evolving the boundary states to time $t^{n+\frac{1}{2}} = t + \frac{1}{2}\Delta t$ via

$$\begin{aligned} \bar{\mathbf{u}}_i^{L,n+\frac{1}{2}} &= \bar{\mathbf{u}}_i^L - \frac{1}{2} \frac{\Delta t}{\Delta x} [\mathbf{f}(\bar{\mathbf{u}}_i^R) - \mathbf{f}(\bar{\mathbf{u}}_i^L)], \\ \bar{\mathbf{u}}_i^{R,n+\frac{1}{2}} &= \bar{\mathbf{u}}_i^R - \frac{1}{2} \frac{\Delta t}{\Delta x} [\mathbf{f}(\bar{\mathbf{u}}_i^R) - \mathbf{f}(\bar{\mathbf{u}}_i^L)], \end{aligned} \quad (29)$$

which allows the method to achieve second order accuracy in time.

Finally, the flux at the left interface for cell i is calculated by solving the Riemann problem posed by the reconstructed state at the right interface of the cell to left of i , $\bar{\mathbf{u}}_{i-1}^{R,n+\frac{1}{2}}$ and the reconstructed state at the left interface of cell i , $\bar{\mathbf{u}}_i^{L,n+\frac{1}{2}}$. The solution to this Riemann problem, $\mathbf{u}_{i-\frac{1}{2}}^n = \mathbf{u}_{i-\frac{1}{2}}^n(\bar{\mathbf{u}}_{i-1}^{R,n+\frac{1}{2}}, \bar{\mathbf{u}}_i^{L,n+\frac{1}{2}})$, is then plugged into equation 17 in order to obtain the flux at the left edge of the cell. The flux at the right interface is calculated in an analogous manner, but the intermediate Riemann state is now a function of the left reconstructed state of cell $i+1$ and the right reconstructed state of cell i , $\mathbf{u}_{i+\frac{1}{2}}^n = \mathbf{u}_{i+\frac{1}{2}}^n(\bar{\mathbf{u}}_i^{R,n+\frac{1}{2}}, \bar{\mathbf{u}}_{i+1}^{L,n+\frac{1}{2}})$. The time update is subsequently executed according to equation 16.

3.4.4. Boundary conditions

From the analysis in the previous section, it can be seen that in order to compute the flux at a cell boundary, the MUSCL-Hancock scheme requires information from four different cells. Consider the state \mathbf{u}_1 at the left edge of a one dimensional computational domain. The flux at the left boundary $\mathbf{f}_{\frac{1}{2}}$ will be a function of the left reconstructed state of cell \mathbf{u}_1 , \mathbf{u}_1^L , and the right reconstructed state of cell \mathbf{u}_0 , \mathbf{u}_0^R . In order to perform these reconstructions, we require information from the cells

at both sides of \mathbf{u}_1 and \mathbf{u}_0 , as per equation 27, which means $\mathbf{u}_1^L = \mathbf{u}_1^L(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2)$, $\mathbf{u}_0^R = \mathbf{u}_0^R(\mathbf{u}_{-1}, \mathbf{u}_0, \mathbf{u}_1)$ and finally $\mathbf{f}_{\frac{1}{2}} = \mathbf{f}_{\frac{1}{2}}[\mathbf{u}_0^R(\mathbf{u}_{-1}, \mathbf{u}_0, \mathbf{u}_1), \mathbf{u}_1^L(\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2)]$. The same analysis can be performed on the right edge of the domain for state \mathbf{u}_N ; this means that the scheme needs at least two additional cells on both sides of the domain to work. In the two dimensional case, there will be two rows of these boundary cells at the top of the domain, another two at the bottom, and two columns each at the sides. These cells are not part of the computational domain itself, and are updated at every timestep using a different procedure than that used for updating the cells within the domain. Our solver uses fixed/Dirichlet boundary conditions, whereupon the values of the boundary cells are specified directly based on the values at the edge rows and columns of the computational domain.

We implemented three types of Dirichlet conditions: transmissive, reflective, and periodic. These can be applied to each dimension independently. In the transmissive case, the value of the boundary states at the beginning and end of the domain, respectively, are given by

$$\begin{aligned} \mathbf{u}_0^n &= \mathbf{u}_1^n, & \mathbf{u}_{N+1}^n &= \mathbf{u}_N^n, \\ \mathbf{u}_{-1}^n &= \mathbf{u}_2^n, & \mathbf{u}_{N+2}^n &= \mathbf{u}_{N-1}^n, \end{aligned} \quad (30)$$

where N is the size of the computational grid along a given dimension. The reflective case is similar except that vector components normal to the boundary are multiplied by -1 , as if the materials/fields are “bouncing” off the boundaries. Other vector components are unchanged with respect to the transmissive case.

Finally, periodic boundary conditions are defined as follows,

$$\begin{aligned} \mathbf{u}_0^n &= \mathbf{u}_N^n, & \mathbf{u}_{N+1}^n &= \mathbf{u}_1^n \\ \mathbf{u}_{-1}^n &= \mathbf{u}_{N-1}^n, & \mathbf{u}_{N+2}^n &= \mathbf{u}_2^n \end{aligned} \quad (31)$$

Conceptually, this represents a scenario in which material is leaving the domain through one side and reentering through the opposite one.

3.5. The HLLC Solver for MHD

We now turn to the solution of the Riemann problem itself. In this work, we implement the HLLC Riemann solver, an extension of the HLL solver for gas dynamics devised by Toro [1] and adapted to MHD by Li [2].

Unlike the earlier HLL solver[31], which employs a single intermediate state and can be overly diffusive, the HLLC solver allows for two intermediate states separated by a contact discontinuity, decreasing the level of

diffusion and enabling it to better capture isolated contact discontinuities. The scheme exploits the continuity of velocity and pressure across a contact discontinuity and the Rankine-Hugionot conditions in order to obtain algebraic expressions for the components of the intermediate star states \mathbf{u}_R^* and \mathbf{u}_L^* . This, coupled with appropriate estimates for the speeds of the left and right moving discontinuities, S_L and S_R , can be used to compute the flux across the interface separating the two initial states \mathbf{u}_R and \mathbf{u}_L . We give expressions for the one dimensional solver; as before, this can be extended to multiple dimensions in a dimensionally split fashion.

The starting point for the scheme is calculating v_x^* , the velocity of the contact discontinuity. There is no single way to do this; we follow Li and define it as

$$v_x^* = \frac{\rho_R v_{x,R} (S_R - v_{x,R}) - \rho_L v_{x,L} (S_L - v_{x,L}) - p_{T,R} + p_{T,L} + B_{x,R}^2 - B_{x,L}^2}{\rho_R (S_R - v_{x,R}) - \rho_L (S_L - v_{x,L})}, \quad (32)$$

which is also the expression for v_x in the intermediate state \mathbf{u}^{HLL} from the HLL scheme,

$$\mathbf{u}^{HLL} = \frac{S_R \mathbf{u}_R - S_L \mathbf{u}_L - \mathbf{f}_R + \mathbf{f}_L}{S_R - S_L}, \quad (33)$$

where $\mathbf{f}_K = \mathbf{f}(\mathbf{u}_K)$ and $K = L, R$. Application of the Rankine-Hugionot conditions allows us to derive an expression for the intermediate total pressure p_T^*

$$p_T^* = \rho_K (S_K - v_{x,K}) (v_x^* - v_{x,K}) + p_{T,K} - B_{x,K}^2 + (B_{x,K}^*)^2, \quad (34)$$

which satisfies $p_{T,L}^* = p_{T,R}^*$ only if $B_{x,L}^* = B_{x,R}^* = B_x^*$, implying that B_x^* is continuous across the contact discontinuity. The same procedure can be applied to density, energy, and the tangential components of the momentum to obtain

$$\rho_K^* = \rho_K \frac{S_K - v_{x,K}}{S_K - v_x^*}, \quad (35)$$

$$(\rho_K v_{y,K})^* = \rho_K^* v_{y,K} + \frac{B_{x,K} B_{y,K} - B_x^* B_{y,K}^*}{S_K - v_x^*}, \quad (36)$$

$$(\rho_K v_{z,K})^* = \rho_K^* v_{z,K} + \frac{B_{x,K} B_{z,K} - B_x^* B_{z,K}^*}{S_K - v_x^*}, \quad (37)$$

$$U_K^* = U_K \frac{\rho_K^*}{\rho_K} + \frac{p_T^* v_x^* - p_{T,K} v_{x,K} - B_x^* (\mathbf{B} \cdot \mathbf{v})^* + B_{x,K} (\mathbf{B} \cdot \mathbf{v})}{S_K - v_x^*}. \quad (38)$$

The Rankine-Hugionot conditions can also be used to derive expressions for $\mathbf{B}_K^* = (B_x^*, B_{y,K}^*, B_{z,K}^*)^T$ and $(\mathbf{B} \cdot \mathbf{v})^*$, however, as Li notes, the derived expressions are incompatible with the integral form of the MHD conservation laws. The solution to this problem is to set $\mathbf{B}_L^* = \mathbf{B}_R^* = \mathbf{B}^{HLL}$ and $(\mathbf{B} \cdot \mathbf{v})^* = \mathbf{B}^{HLL} \cdot \mathbf{v}^{HLL}$. With these definitions, \mathbf{u}_K^* will satisfy the integral form of the conservation laws, and the HLLC flux can be written in

the same manner as for the Euler equations, namely

$$\mathbf{f}^{HLLC} = \begin{cases} \mathbf{f}_L, & 0 \leq S_L \\ \mathbf{f}_L^* = \mathbf{f}_L + S_L (\mathbf{u}_L^* - \mathbf{u}_L) & S_L < 0 \leq v_x^* \\ \mathbf{f}_R^* = \mathbf{f}_R + S_R (\mathbf{u}_R^* - \mathbf{u}_R) & v_x^* < 0 \leq S_R \\ \mathbf{f}_R & S_R < 0 \end{cases}, \quad (39)$$

The final piece in the HLLC puzzle is estimating S_L and S_R . Again, there is no single way of doing this, and the subject continues to be an area of active research [32]. We find that for our test cases, setting the wave speeds to be the minimum and maximum possible for the material within the cell, as suggested in [6], works well,

$$\begin{aligned} S_L &= \min(v_{x,L}, v_{x,R}) - \max(c_{f,L}, c_{f,R}), \\ S_R &= \max(v_{x,L}, v_{x,R}) + \max(c_{f,L}, c_{f,R}). \end{aligned} \quad (40)$$

3.6. Hyperbolic Divergence Cleaning

As mentioned in previous sections, numerical solutions of the MHD equations are not guaranteed to satisfy the divergence constraint $\nabla \cdot \mathbf{B} = 0$. In one dimension, this is not an issue because the evolution equation for B_x vanishes, hence the quantity is guaranteed to remain constant throughout the simulation. This is not the case for multidimensional simulations.

Once $\nabla \cdot \mathbf{B} \neq 0$, the MHD system of equations is no longer conservative, and numerical simulations can become unstable [16, 17]. A number of ways have been proposed to resolve this problem, most of which are variations of the hyperbolic divergence cleaning method of Dedner et al [4] and the constrained transport method of Evans and Hawley [33]. Due to its relative simplicity, our solver implements the hyperbolic divergence cleaning approach, but we note that constrained transport can be a more effective method of enforcing the divergence constraint because it guarantees that $\frac{\partial}{\partial t} (\Delta \cdot \mathbf{B}) = 0$ with machine round-off error accuracy [17], which is not the case for the hyperbolic divergence cleaning approach.

Divergence cleaning works by replacing equation 4 with

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{B} \otimes \mathbf{v} - \mathbf{v} \otimes \mathbf{B}) + \nabla \psi = 0, \quad (41)$$

where ψ is known as a generalized Lagrangian multiplier (GLM), and satisfies

$$\mathcal{D}(\psi) + \nabla \cdot \mathbf{B} = 0. \quad (42)$$

\mathcal{D} is a linear differential operator whose choice will affect the effectiveness of the cleaning procedure. Our

solver implements two different operators: a purely hyperbolic one, given by

$$\mathcal{D}_H = \frac{1}{c_h^2} \frac{\partial \psi}{\partial t}, \quad (43)$$

and a mixed hyperbolic/parabolic operator with definition

$$\mathcal{D}_M(\psi) = \frac{1}{c_h^2} \frac{\partial \psi}{\partial t} + \frac{1}{c_p^2} \psi, \quad (44)$$

with $c_h > 0$, $c_p > 0$ constants. c_h determines the speed at which the local divergence errors are transported to the cell boundaries; we can select this speed to be the maximum possible wave speed within the computational domain at time t^n ,

$$c_h = c_h^n = a_{\max}^n = \max_{i,j,k} (|v_x^n| + c_{f,x}^n, |v_y^n| + c_{f,y}^n, |v_z^n| + c_{f,z}^n). \quad (45)$$

The optimal value of the parabolic damping term c_p is not obvious. We follow Dedner et al and fix the ratio $\frac{c_p^2}{c_h^2} = 0.18$, although as noted by Mignone and Tzeferacos [34], this is potentially problematic given that the ratio is not dimensionless.

Equation 42 now reads

$$\frac{\partial \psi}{\partial t} + c_h^2 (\nabla \cdot \mathbf{B}) = \mathcal{S}(\psi), \quad (46)$$

where $\mathcal{S}(\psi) = 0$ in the purely hyperbolic case and $\mathcal{S}(\psi) = \frac{c_h^2}{c_p^2} \psi$ in the mixed correction case. In one dimension, equations 41 and 46 form a hyperbolic system (with a source term, when $\mathcal{D}(\psi) = \mathcal{D}_M(\psi)$), and decouple from the remaining MHD equations 1-3 [4, 17]. In the homogeneous case and for arbitrary left and right states $(B_{x,L}, \psi_L)^T$ and $(B_{x,R}, \psi_R)^T$, the Riemann problem can be solved exactly, with the solution $(\tilde{B}_x, \tilde{\psi})^T$ given by

$$\begin{bmatrix} \tilde{B}_x \\ \tilde{\psi} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} (B_{x,L} + B_{x,R}) - \frac{1}{2c_h} (\psi_R - \psi_L) \\ \frac{1}{2} (\psi_L + \psi_R) - \frac{c_h}{2} (B_{x,R} - B_{x,L}) \end{bmatrix}. \quad (47)$$

With the addition of ψ into the system, the vector of conserved quantities $\tilde{\mathbf{u}}$ and the updated flux $\tilde{\mathbf{f}}(\tilde{\mathbf{u}})$ are now given by

$$\begin{aligned} \tilde{\mathbf{u}} &= (\rho, \rho v_x, \rho v_y, \rho v_z, U, B_x, B_y, B_z, \psi)^T, \\ \tilde{\mathbf{f}}(\tilde{\mathbf{u}}) &= (\mathbf{f}^{HLLC}(\mathbf{u}_L, \mathbf{u}_R; \tilde{B}_x), 0)^T + (0, 0, 0, 0, 0, \tilde{\psi}, 0, 0, c_h^2 \tilde{B}_x)^T, \end{aligned} \quad (48)$$

where the first component in the sum for $\tilde{\mathbf{f}}(\tilde{\mathbf{u}})$ corresponds to the flux obtained from the solution of the Riemann problem between the original reconstructed eight component state vectors \mathbf{u}_L and \mathbf{u}_R with $B_x^L = B_x^R = \tilde{B}_x$ and an additional 0 appended at the end. When $\mathcal{S}(\psi) \neq$

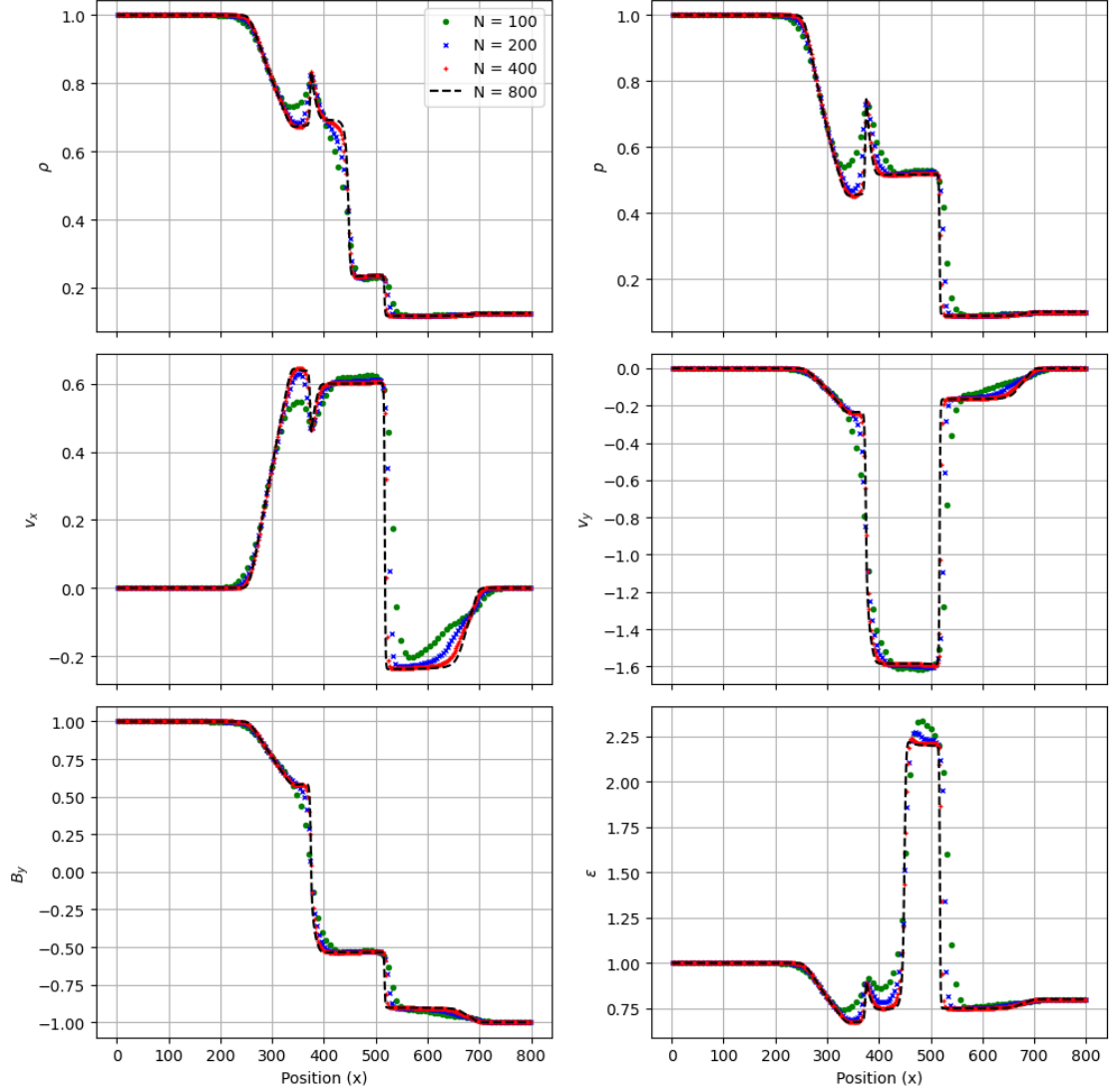


Figure 1: Results for the magnetic shock tube (Brio and Wu) test for density ρ , pressure p , x-velocity v_x , y-velocity v_y , y-magnetic field B_y , and specific internal energy $\varepsilon = p/(\gamma-1)$ using $N = 100$, $N = 200$, $N = 400$, and $N = 800$ cells.

0, we can use operator splitting. After executing the Godunov update based on the modified flux to obtain the intermediate state $\tilde{\mathbf{u}}^*$, we solve the initial value problem

$$\frac{d\psi}{dt} = \mathcal{S}(\psi), \quad \psi(0) = \psi^*. \quad (49)$$

In the mixed correction case, $\mathcal{S}(\psi) = \frac{c_h^2}{c_p^2}\psi$ and the above can be integrated exactly to yield $\psi^{n+1} = \psi^* \exp(-\Delta t c_h^2/c_p^2)$ for a given timestep Δt . The rest of the vector components are unchanged from the intermediate state. The boundary conditions for ψ are set exactly as they are for the other non-vector components of

$\tilde{\mathbf{u}}$, and we set $\psi^0 = 0$ for all cells at $t = 0$. We note that when we implement divergence cleaning in tandem with MUSCL-Hancock, we do not reconstruct either the component of \mathbf{B} normal to the interface or ψ ; doing otherwise leads to unphysical results.

4. Validation Tests

4.1. 1D Brio and Wu Magnetic Shock Tube Test

The first of the validation tests is the strictly one dimensional Brio and Wu test [7]. The test's initial conditions are that of a Riemann problem, with two

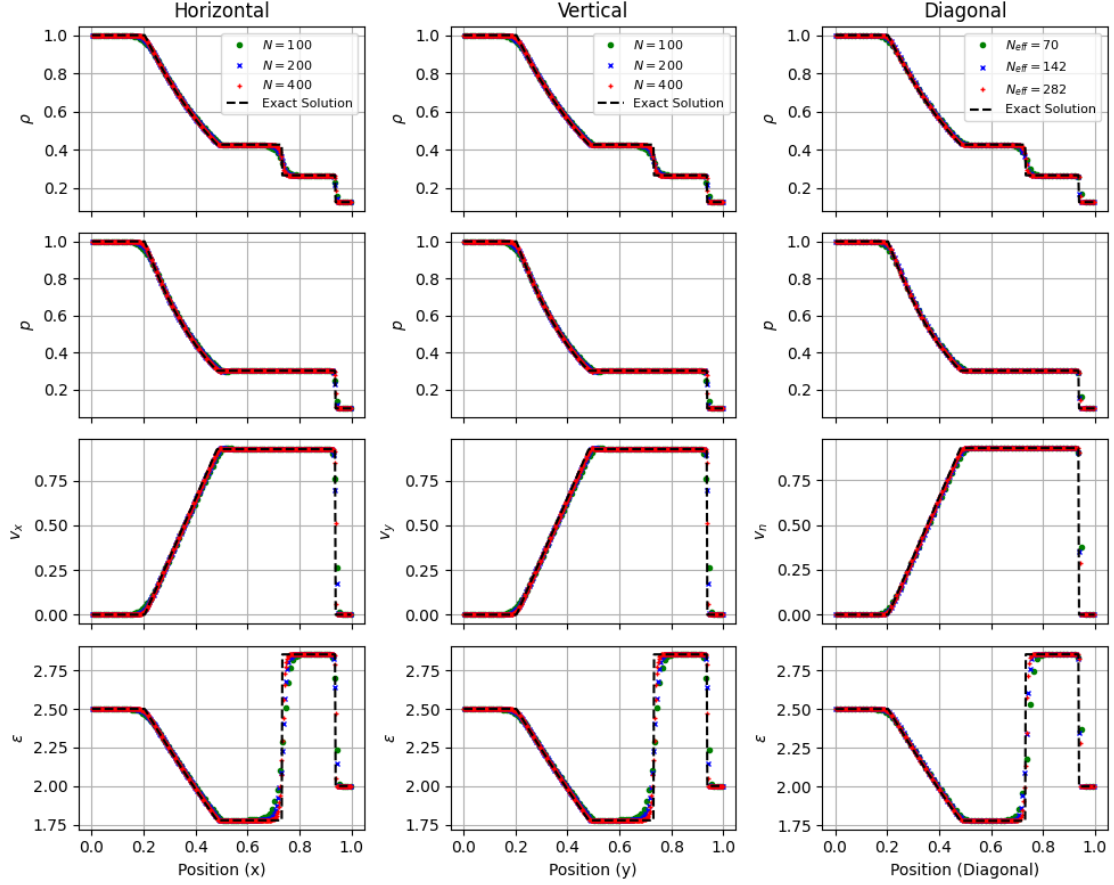


Figure 2: Results for the two-dimensional Sod test for density ρ , pressure p , normal velocity and specific internal energy $\varepsilon = p/(\rho(\gamma-1))$ using $N = 100$, $N = 200$, and $N = 400$ cells for three different orientations, together with the exact solutions. The plots correspond to slices along the lines in the $x - y$ plane that are perpendicular to the line separating the initial states \mathbf{w}_L and \mathbf{w}_R .

distinct uniform states on either side of the computational domain. In terms of primitive variables, $\mathbf{w} = (\rho, v_x, v_y, v_z, p, B_x, B_y, B_z)^T$, the initial conditions are

$$\mathbf{w}^0(x) = \begin{cases} \mathbf{w}_L = (1, 0, 0, 0, 1, 0.75, 1, 0)^T, & x \leq L/2, \\ \mathbf{w}_R = (0.125, 0, 0, 0, 0.1, 0.75, -1, 0)^T, & x > L/2, \end{cases} \quad (50)$$

where x goes from 0 to $L = 800$. We note that the initial conditions \mathbf{w}^0 are identical to that of the Sod test, which we present in the next subsection, except $\mathbf{B} \neq 0$. The value of the adiabatic constant $\gamma = 2$, and the test is run from $t = 0$ up to time $t = 80$. Boundary conditions are transmissive. We find that best results are obtained when we perform the slope limiting based on energy, thus the slope ratio r is defined according to equation 28.

We plot ρ , p , v_x , v_y , B_y , and specific internal energy $\varepsilon = p/(\rho(\gamma-1))$ in Figure 1 for four different resolutions.

These results agree with the results obtained by Brio and Wu and other authors [35, 36]. It is apparent that higher resolutions are able to capture the features of the test in a better way, especially because the test contains a larger number of discontinuities as a result of the complex interaction between the hydrodynamics variables and the magnetic field.

4.2. Sod Shock Tube Test

To initially validate the solver in two dimensions, we study three variations of the one dimensional Sod shock tube test [27]. This test is a purely hydrodynamical test because the initial conditions have all magnetic field components equal to zero. As in the Brio and Wu test, the initial conditions consist of two states, \mathbf{w}_L and \mathbf{w}_R , given by

$$\begin{aligned} \mathbf{w}_L &= (1, 0, 0, 0, 1, 0, 0, 0)^T, \\ \mathbf{w}_R &= (0.125, 0, 0, 0, 0.1, 0, 0, 0)^T, \end{aligned} \quad (51)$$

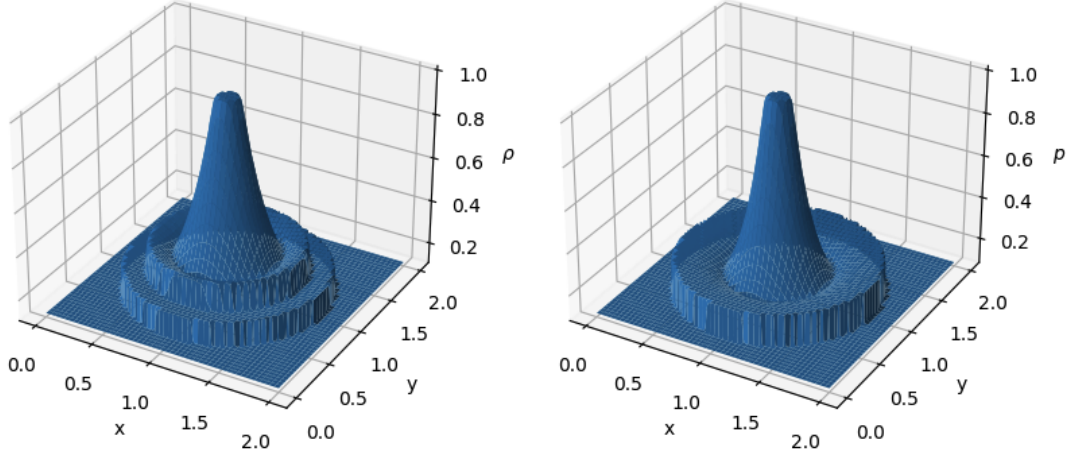


Figure 3: Surface plots for ρ and p in the cylindrical explosion test described in 4.3. The solution exhibits a complex wave pattern, with a circular shock wave travelling outwards from the center, a circular contact surface moving in the same direction, and a rarefaction travelling inwards toward the origin. The simulation has a resolution of 800×800 cells.

For this test, $x \in [0, 1]$, $y \in [0, 1]$, $\gamma = 1.4$, the simulation runs up to time $t = 0.25$, and the boundary conditions are transmissive along both axes. The three variations of the test differ only in their relative orientations with respect to the grid. For the first test $\mathbf{w}^0(x) = \mathbf{w}_L$ if $x < 0.5$, otherwise $\mathbf{w}^0(x) = \mathbf{w}_R$. For the second test, we perform a 180 degree rotation, and assign $\mathbf{w}^0(y) = \mathbf{w}_L$ if $y < 0.5$. Finally, for the third test we orient the states along the diagonal $y = 1 - x$, so $\mathbf{w}^0(x, y) = \mathbf{w}_L$ if $x + y > 1$. As before, our slope limiting is based on the energy variable.

We plot slices for ρ , p , normal velocity v_n , and ε along lines in the $x - y$ plane that are perpendicular to the line separating the states, with three different resolutions, as seen in Figure 2. As expected, there are no differences between the horizontal and vertical tests, and the results align with those of Toro [27]. In the diagonal test, the effective resolution is lowered due to the diagonal test orientation; in order to compare the results with those of the horizontal and vertical slices, we only use values within the domain of the circle with radius 0.5 centered at $x = y = 0.5$. This means that the resolution for the diagonal test is only 70% that of the horizontal and vertical tests for a given square computational grid of size $N \times N$.

4.3. Cylindrical Explosion Test

The next test is a genuine two dimensional test. It is a variation of the shock tube test from above with the initial states separated by a circle in the $x - y$ plane instead of straight lines, $\mathbf{w}_L \rightarrow \mathbf{w}_{in}$, $\mathbf{w}_R \rightarrow \mathbf{w}_{out}$, where \mathbf{w}_{in} is

the state inside the circle and \mathbf{w}_{out} is the state outside of it. These are the same as above,

$$\begin{aligned} \mathbf{w}_{in} &= (1, 0, 0, 0, 1, 0, 0, 0)^T, \\ \mathbf{w}_{out} &= (0.125, 0, 0, 0, 0.1, 0, 0, 0)^T, \end{aligned} \quad (52)$$

but now $x \in [0, 2]$, $y \in [0, 2]$ and $\mathbf{w}^0(x, y) = \mathbf{w}_{in}$ if $(x - x_0)^2 + (y - y_0)^2 < r^2$, otherwise $\mathbf{w}^0(x, y) = \mathbf{w}_{out}$. In this case, $x_0 = y_0 = 1$ and $r = 0.4$. The test runs up to a time $t = 0.25$, and $\gamma = 1.4$. We ran the simulation using three different resolutions: $N = 100$ cells, $N = 200$ cells, $N = 400$ cells along each axis. The slope limiting is again performed based on energy.

Figure 3 shows surface plots for the density ρ and the pressure p as a function of x and y at a resolution of 800×800 cells, and Figure 4 shows plots for ρ , p , x -velocity v_x and specific internal energy ε for $x \in [1, 2]$, where the slices are along the x -axis, $y = 0$. Boundary conditions are transmissive along both axes. Our results match those presented in [27].

4.4. 2D Brio and Wu Magnetic Shock Tube Test

To validate our 2D solver with a non-zero magnetic field, we extend the results from section 4.1 to two dimensions by running different orientations of the simulation in a square grid, as was done in section 4.2 for the one-dimensional Sod test. For the horizontal version of the test, the initial conditions are the same as those given in 50. For the vertical version of the test, the initial conditions are identical except we swap the x - and

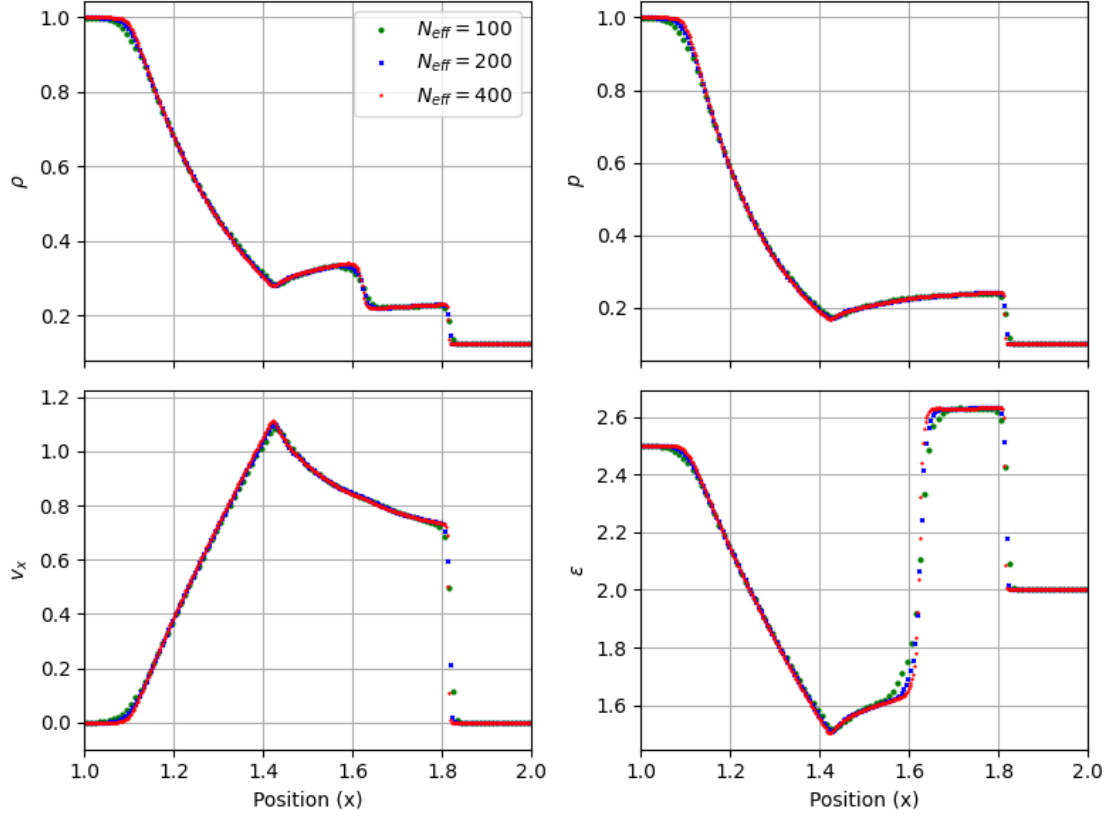


Figure 4: Results for the cylindrical explosion test for ρ , p , v_x and specific internal energy ε using $N = 200$, $N = 400$, and $N = 800$ cells along each axis. The plots correspond to slices along the line $y = 1$ and $x \in [1, 2]$. Note that $N_{eff} \neq N$ because the x -axis in the full simulation runs from 0 to 2; plotting the same quantities from $x = 0$ to $x = 1$ gives a mirror image of the plots above due to the symmetric nature of the problem.

y - components of the magnetic field in the initial states, and the states are now separated by the line $y = L/2$,

$$\mathbf{w}^0(y) = \begin{cases} \mathbf{w}_L = (1, 0, 0, 0, 1, 1, 0.75, 0)^T, & y \leq L/2 \\ \mathbf{w}_R = (0.125, 0, 0, 0, 0.1, -1, 0.75, 0)^T & y > L/2 \end{cases} \quad (53)$$

For the diagonally aligned test, we need to rotate the magnetic field orientation of the initial states by $\pi/4$, counter-clockwise, with respect to the orientation of the field in the horizontal version of the test. We now have

$$\mathbf{w}^0(x, y) = \begin{cases} \mathbf{w}_L = (1, 0, 0, 0, 1, -\sqrt{2}/8, 7\sqrt{2}/8, 0), & x + y \leq L \\ \mathbf{w}_R = (0.125, 0, 0, 0, 0.1, 7\sqrt{2}/8, -\sqrt{2}/8, 0) & x + y > L \end{cases} \quad (54)$$

Figure 5 shows slices for ρ , p , normal velocity, tangential velocity, and tangential magnetic field along lines in the $x - y$ plane perpendicular to the line separating the initial states, with three different resolutions for each orientation. It can be observed that the results agree with those presented in Figure 1, as expected. We note that for the diagonal test, the simulation blows up at resolutions beyond $N = 150$ cells; this is because our implementation of the boundary conditions leaves the

corner boundary cells in the domain undefined, which leads to numerical instabilities for high resolutions in this particular test. We make no attempt to correct this because this can be bypassed by rotating the coordinate system. As before, we observe that the resolution for the diagonal case is reduced versus the horizontal and vertical cases even if the size of the computational domain is the same.

4.5. Orszag-Tang Vortex Test

The Orszag-Tang test is a stringent two-dimensional ideal MHD test in which the initial conditions lead to supersonic turbulent flows [37, 17]. The initial condi-

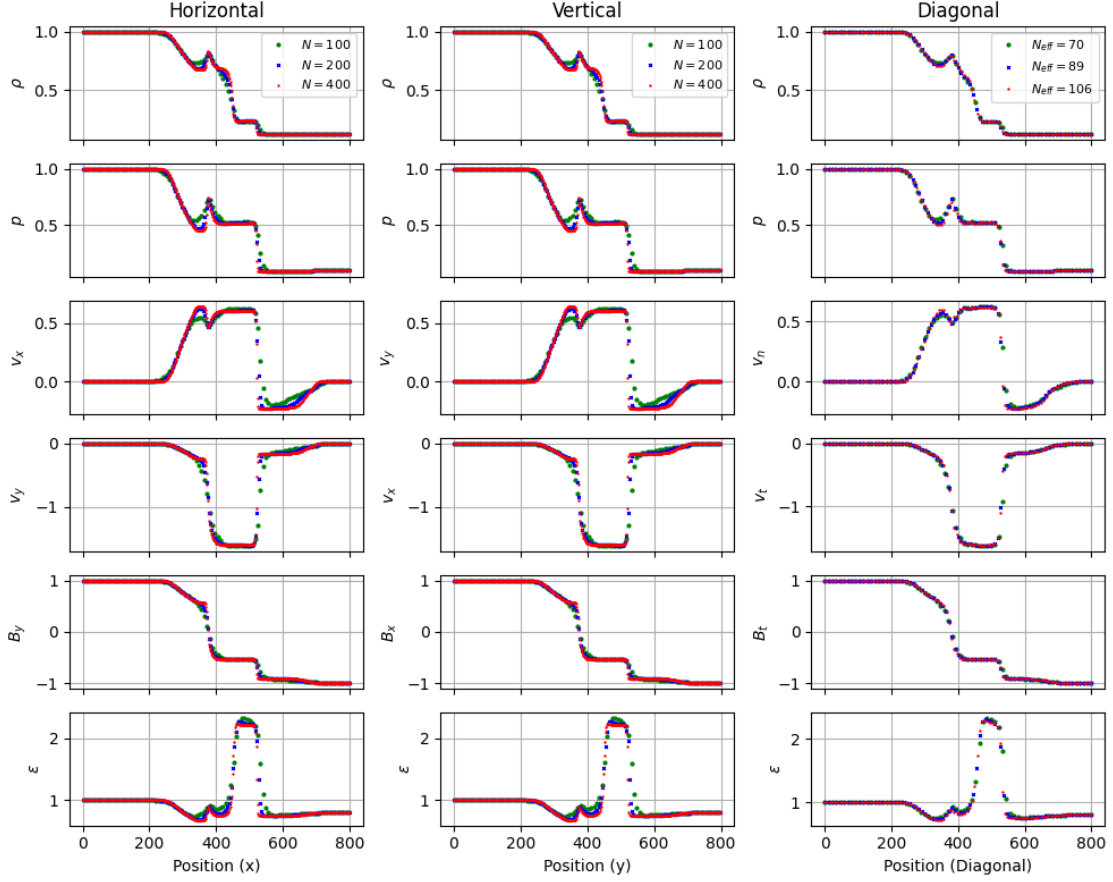


Figure 5: Results for the two-dimensional Brio and Wu test for ρ , p , normal velocity, tangential velocity, tangential magnetic field, and specific internal energy ε using $N = 200$, $N = 400$, and $N = 800$ cells along each axis for the horizontal and vertical orientations and $N = 100$, $N = 125$, and $N = 150$ cells along each axis for the diagonal orientation. The plots correspond to slices along the lines in the $x-y$ plane that are perpendicular to the line separating the initial states \mathbf{w}_L and \mathbf{w}_R .

tions are

$$\mathbf{w}^0(x, y) = \begin{bmatrix} \rho = \gamma^2 \\ v_x = -\sin(2\pi y) \\ v_y = \sin(2\pi x) \\ v_z = 0 \\ p = \gamma \\ B_x = -\sin(2\pi y) \\ B_y = \sin(4\pi x) \\ B_z = 0 \end{bmatrix}, \quad (55)$$

where $\gamma = 5/3$. The domain is square, with $x \in [0, 1]$ and $y \in [0, 1]$, and the boundary conditions are periodic along both axes. We run the simulation up to times $t = 0.5$ and $t = 1$ using a resolution of 256×256 cells, and compare results with no divergence cleaning, hyperbolic cleaning, and mixed cleaning. For this test, we find that performing the slope limiting on each variable independently provides better results, especially for the case in which no divergence cleaning is performed. This

means that the slope ratio is a vector quantity \mathbf{r} , defined by equation 25.

Figure 6 shows ρ as a function of x and y at both final times in all three cases. Although the results look similar, careful observation reveals that the uncorrected case presents some numerical artefacts as a result of the accumulation in divergence errors. This is further illustrated in Figure 7 which shows slices for ρ along the line $y = 0.314$ for the $t = 0.5$ simulation, where it can be readily seen that the absence of divergence cleaning leads to an artificially choppy curve.

Figure 8 shows explicitly how the divergence cleaning procedure reduces the divergence error significantly. We measure the divergence at a cell $(\nabla \cdot \mathbf{B})_{i,j}$ in our simulation using central differences,

$$(\nabla \cdot \mathbf{B})_{i,j} = \frac{(B_x)_{i+1,j} - (B_x)_{i-1,j}}{2\Delta x} + \frac{(B_y)_{i,j+1} - (B_y)_{i,j-1}}{2\Delta y}. \quad (56)$$

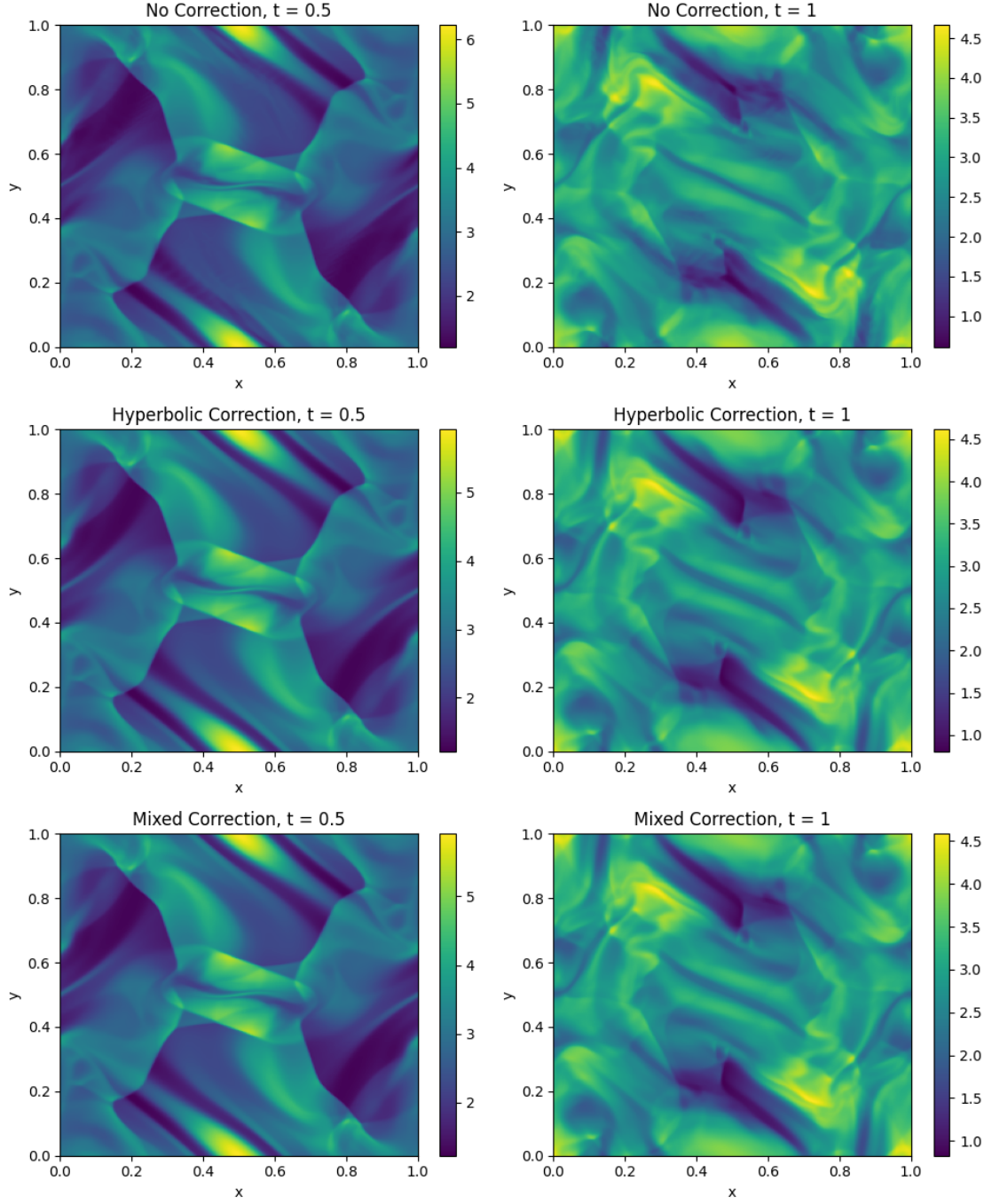


Figure 6: Snapshots of ρ for the Orszag-Tang vortex at $t = 0.5$ and $t = 1$ for simulations performed with no divergence cleaning (first row), hyperbolic divergence cleaning (second row) and mixed divergence cleaning (third row). For the $t = 1$ case especially, it can be seen that the accumulation of divergence errors results in the appearance of numerical artifacts that blur the pattern. Both types of divergence cleaning lead to much sharper images.

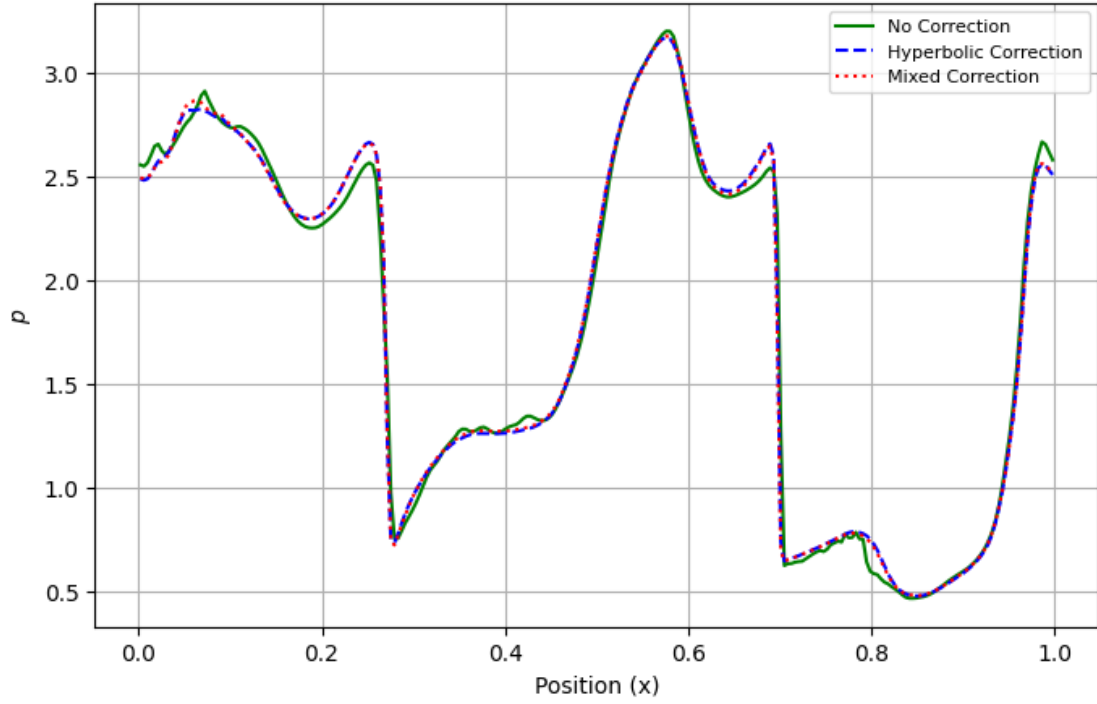


Figure 7: Slice of ρ along the line $y = 0.314$ at $t = 0.5$ for the Orszag-Tang vortex, for simulations using no divergence cleaning, hyperbolic cleaning, and mixed cleaning. It can be seen that the absence of divergence cleaning leads to an artificially choppy curve and that both types of cleaning give very similar results.

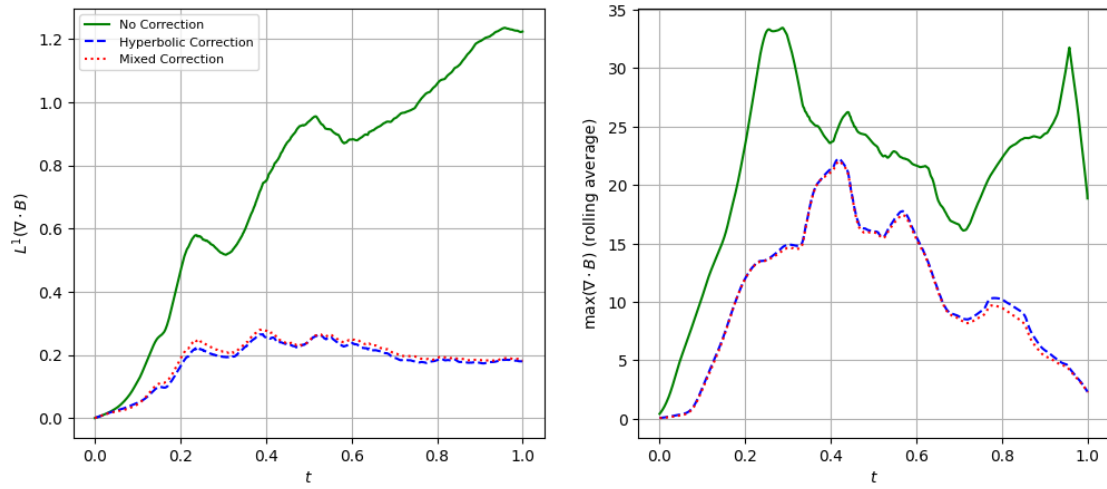


Figure 8: Evolution of the L_1 error and a rolling average of $\max(\nabla \cdot \mathbf{B})$ for the Orszag-Tang vortex. Without correction, divergence errors accumulate as the simulation goes on; these can be minimized using the divergence cleaning procedures. Both hyperbolic and mixed corrections yield similar reductions in error, up to a factor of $\sim 7\times$ in the case of the L_1 norm and $\sim 14\times$ in the case of $\max(\nabla \cdot \mathbf{B})$.

We then quantify the error in terms of the L^1 norm and $\max(\nabla \cdot \mathbf{B})_{i,j}$. The L^1 norm is defined as

$$\begin{aligned} L^1(\nabla \cdot \mathbf{B}) &= \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |(\nabla \cdot \mathbf{B})_{real} - (\nabla \cdot \mathbf{B})_{i,j}| \\ &= \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |(\nabla \cdot \mathbf{B})_{i,j}|, \end{aligned} \quad (57)$$

for an evenly spaced computational grid, where the second equality follows from the fact that $(\nabla \cdot \mathbf{B})_{real} = 0$ for all i, j , and N_x, N_y are the number of computational cells along the x - and y - axes, respectively. From the Figure, we can see that the hyperbolic and mixed corrections give very similar results, reducing the divergence error up to a factor of $\sim 7x$ in terms of the L^1 norm and $\sim 14x$ in terms of $\max(\nabla \cdot \mathbf{B})$.

Our simulation results are in general agreement with those from other references [34, 17], although Vides et al. find that in the absence of divergence cleaning, $\max(\nabla \cdot \mathbf{B})$ reaches higher levels than what we were able to observe. This might be related to their use of a seven-wave HLLD solver versus our implementation of the more diffusive HLLC solver.

4.6. Kelvin-Helmholtz Instability

Our final test is the Kelvin-Helmholtz instability described in [34, 17], which constitutes another example of highly nonlinear evolution. In this case, the initial state \mathbf{w}^0 is given by

$$\mathbf{w}^0(x, y) = \begin{bmatrix} \rho = 1 \\ v_x = \frac{1}{2} \tanh(20y) \\ v_y = 0.01 \sin(2\pi x) \exp(-100y^2) \\ v_z = 0 \\ p = 1/\gamma \\ B_x = 0.1 \cos(\pi/3) \\ B_y = 0 \\ B_z = 0.1 \sin(\pi/3) \end{bmatrix}, \quad (58)$$

where $\gamma = 5/3$. The domain is rectangular with $x \in [0, 1]$ and $y \in [-1, 1]$. The boundary conditions are reflective on top and bottom, and periodic on left and right. We run the simulation up to times $t = 5$, $t = 8$, and $t = 12$ using a resolution of 256×512 cells, which allows us to keep $\Delta x = \Delta y$. As was the case in the previous test, we perform the slope limiting on each variable independently. To compare with results obtained in previous works [37, 17], we plot the ratio of the poloidal field strength to the toroidal component strength, $\alpha = \sqrt{B_x^2 + B_y^2}/B_z$, and show results with

no divergence cleaning, hyperbolic divergence cleaning, and mixed divergence cleaning, as seen in Figure 9.

The Figure shows that the ratio α evolves from a relatively simple cat vortex shape at $t = 5$ into a highly complex turbulent pattern as a result of the appearance of magnetic instabilities that lead to magnetic reconnection events. At $t = 5$, the results for the simulations with and without divergence cleaning are remarkably similar, although the magnitude of α is larger in the uncorrected test. As the simulation evolves, the differences become more noticeable, and by $t = 12$ it is clear that the results are different depending on whether the simulation is performed with divergence cleaning or not. The most remarkable difference is that the uncorrected case is unable to preserve the test's initial rotational symmetry, while both the hyperbolic and mixed corrections do preserve it.

Figure 10 shows the evolution of the L_1 error and a rolling average of $\max(\nabla \cdot \mathbf{B})$ as a function of simulation time t . Both types of correction give very similar results, with a reduction of up to $\sim 14x$ for the L_1 error and $\sim 8x$ for the maximum error, which grows very quickly in the uncorrected case up to $t = 6$ and then oscillates.

Our results are consistent with those found by previous authors [37, 17], although these references used less diffusive approximate Riemann solvers and/or higher order reconstruction methods, so their results are somewhat different.

5. Conclusions

We have presented and validated a two-dimensional numerical solver that is able to solve the compressible Euler equations and the ideal, non-relativistic equations of magnetohydrodynamics with second order accuracy in both space and time. Our method uses Godunov's scheme [11] in conjunction with a modified version of the HLLC solver [2], the MUSCL-Hancock scheme [3], and divergence cleaning [4] to obtain results comparable to those of previous works [37, 17] for stringent one- and two-dimensional MHD tests such as the Brio and Wu test, the Orszag-Tang vortex test, and the Kelvin-Helmholtz instability test. We have demonstrated how in the absence of divergence cleaning, divergence errors can accumulate in a way that leads to the generation of numerical artifacts that affect the results of the simulations.

Our solver can be potentially extended in a number of ways. As a first step, a less diffusive Riemann solver such as HLLD [38] could be used. This solver was developed specifically for MHD and is able to explicitly capture all seven waves present in the one-dimensional

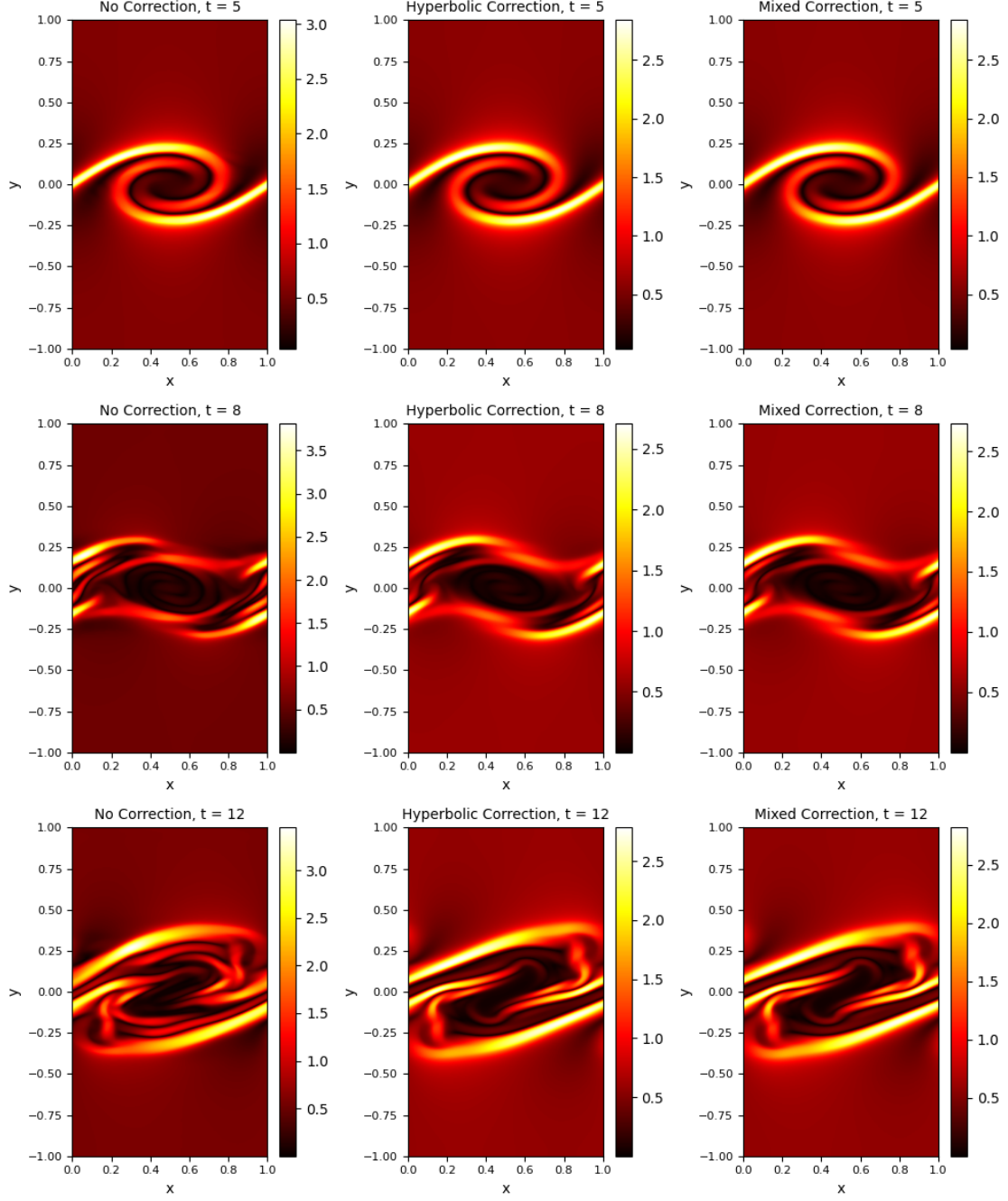


Figure 9: Snapshots of $\alpha = \sqrt{B_x^2 + B_y^2}/B_z$ for the Kelvin-Helmholtz instability at $t = 5$, $t = 8$, and $t = 12$ for simulations performed with no divergence cleaning (first column), hyperbolic divergence cleaning (second column) and mixed divergence cleaning (third column). At $t = 12$, it can be seen that the uncorrected simulation is unable to preserve the original rotational symmetry of the pattern, unlike the simulations using either type of divergence cleaning schemes.

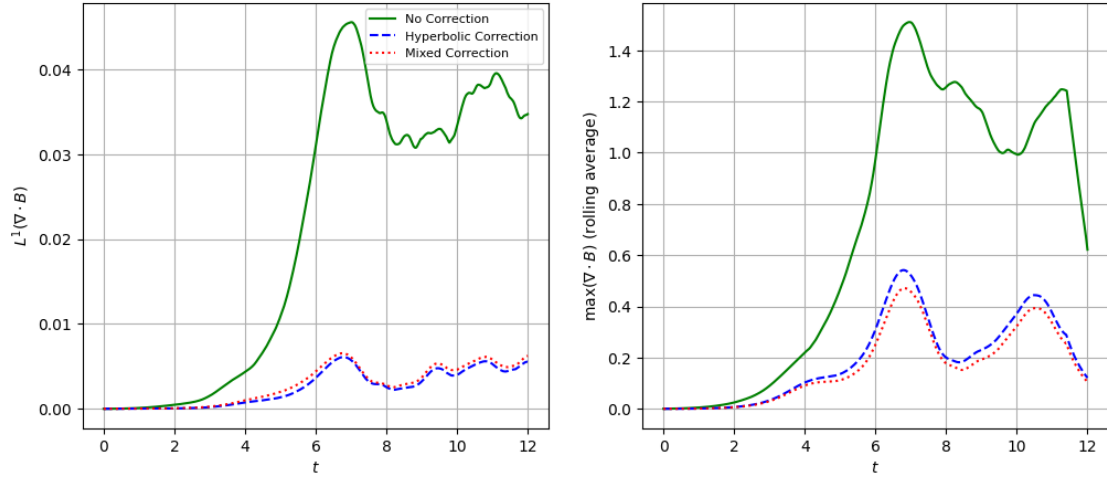


Figure 10: Evolution of the L_1 error and a rolling average of $\max(\nabla \cdot \mathbf{B})$ for the Kelvin-Helmholtz instability. As seen previously with the Orszag-Tang vortex (8) without correction, divergence errors accumulate as the simulation goes on, and using the divergence cleaning procedure minimizes the error. Both hyperbolic and mixed corrections yield similar reductions in error, up to a factor of $\sim 14\times$ in the case of the L_1 norm and $\sim 8\times$ in the case of $\max(\nabla \cdot \mathbf{B})$.

MHD equations, as opposed to HLLC which is a three-wave solver. Higher order accuracy can be obtained by using an adaptation of the PPM scheme [13, 39], or versions of the WENO scheme [37], among other methods. Additionally, the divergence free condition could be enforced by a thermodynamically consistent version of divergence cleaning [18] or via constrained transport [33], which is inherently divergence free but harder to implement.

References

- [1] E. F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the hll-riemann solver, *Shock waves* 4 (1994) 25–34.
- [2] S. Li, An hllc riemann solver for magneto-hydrodynamics, *Journal of computational physics* 203 (1) (2005) 344–357.
- [3] B. Van Leer, On the relation between the upwind-differencing schemes of godunov, engquist–osher and roe, *SIAM Journal on Scientific and statistical Computing* 5 (1) (1984) 1–20.
- [4] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, M. Wesenberg, Hyperbolic divergence cleaning for the mhd equations, *Journal of Computational Physics* 175 (2) (2002) 645–673.
- [5] K. S. Thorne, R. D. Blandford, *Modern classical physics: optics, fluids, plasmas, elasticity, relativity, and statistical physics*, Princeton University Press, 2017.
- [6] S. M. Millmore, *Advanced continuum modelling lecture notes* (2022).
- [7] M. Brio, C. C. Wu, An upwind differencing scheme for the equations of ideal magnetohydrodynamics, *Journal of computational physics* 75 (2) (1988) 400–422.
- [8] P. L. Roe, Approximate riemann solvers, parameter vectors, and difference schemes, *Journal of computational physics* 43 (2) (1981) 357–372.
- [9] A. L. Zachary, P. Colella, A higher-order godunov method for the equations of ideal magnetohydrodynamics, *Journal of Computational Physics* 99 (2) (1992) 341–347.
- [10] J. B. Bell, P. Colella, J. A. Trangenstein, Higher order godunov methods for general systems of hyperbolic conservation laws, *Journal of Computational Physics* 82 (2) (1989) 362–397.
- [11] S. K. Godunov, I. Bohachevsky, Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics, *Matematicheskij sbornik* 47 (3) (1959) 271–306.
- [12] W. Dai, P. R. Woodward, An approximate riemann solver for ideal magnetohydrodynamics, *Journal of Computational Physics* 111 (2) (1994) 354–372.
- [13] W. Dai, P. R. Woodward, Extension of the piecewise parabolic method to multidimensional ideal magnetohydrodynamics, *Journal of Computational Physics* 115 (2) (1994) 485–514.
- [14] K. F. Gurski, An hllc-type approximate riemann solver for ideal magnetohydrodynamics, *SIAM Journal on Scientific Computing* 25 (6) (2004) 2165–2187.
- [15] D. S. Balsara, A two-dimensional hllc riemann solver for conservation laws: Application to euler and magnetohydrodynamic flows, *Journal of Computational Physics* 231 (22) (2012) 7476–7503.
- [16] G. Toth, The $\nabla \cdot \mathbf{b} = 0$ constraint in shock-capturing magnetohydrodynamics codes, *Journal of Computational Physics* 161 (2) (2000) 605–652.
- [17] J. Vides, E. Audit, H. Guillard, B. Nkonga, Divergence-free mhd simulations with the heracles code, in: *ESAIM: Proceedings*, Vol. 43, EDP Sciences, 2013, pp. 180–194.
- [18] D. Derigs, A. R. Winters, G. J. Gassner, S. Walch, M. Böhm, Ideal glm-mhd: about the entropy consistent nine-wave magnetic field divergence diminishing ideal magnetohydrodynamics equations, *Journal of Computational Physics* 364 (2018) 420–467.
- [19] J. M. Stone, M. L. Norman, Zeus-2d: a radiation magnetohydrodynamics code for astrophysical flows in two space dimensions. ii. the magnetohydrodynamic algorithms and tests, *The Astrophysical Journal Supplement Series* 80 (1992) 791.

- [20] B. Fryxell, K. Olson, P. Ricker, F. Timmes, M. Zingale, D. Lamb, P. MacNeice, R. Rosner, J. Truran, H. Tufo, Flash: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes, *The Astrophysical Journal Supplement Series* 131 (1) (2000) 273.
- [21] S. Fromang, P. Hennebelle, R. Teyssier, A high order godunov scheme with constrained transport and adaptive mesh refinement for astrophysical magnetohydrodynamics, *Astronomy & Astrophysics* 457 (2) (2006) 371–384.
- [22] A. Mignone, G. Bodo, S. Massaglia, T. Matsakos, O. e. Tesileanu, C. Zanni, A. Ferrari, Pluto: A numerical code for computational astrophysics, *The Astrophysical Journal Supplement Series* 170 (1) (2007) 228.
- [23] M. Gonzalez, E. Audit, P. Huynh, Heracles: A three-dimensional radiation hydrodynamics code, *Astronomy & Astrophysics* 464 (2) (2007) 429–435.
- [24] J. M. Stone, T. A. Gardiner, P. Teuben, J. F. Hawley, J. B. Simon, Athena: A new code for astrophysical mhd, *The Astrophysical Journal Supplement Series* 178 (1) (2008) 137.
- [25] A. J. Cunningham, A. Frank, P. Varnière, S. Mitran, T. W. Jones, Simulating magnetohydrodynamical flow with constrained transport and adaptive mesh refinement: algorithms and tests of the astrobear code, *The Astrophysical Journal Supplement Series* 182 (2) (2009) 519.
- [26] G. L. Bryan, M. L. Norman, B. W. O’Shea, T. Abel, J. H. Wise, M. J. Turk, D. R. Reynolds, D. C. Collins, P. Wang, S. W. Skillman, et al., Enzo: An adaptive mesh refinement code for astrophysics, *The Astrophysical Journal Supplement Series* 211 (2) (2014) 19.
- [27] E. F. Toro, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*, Springer, 2009.
- [28] P. Colella, P. R. Woodward, The piecewise parabolic method (ppm) for gas-dynamical simulations, *Journal of computational physics* 54 (1) (1984) 174–201.
- [29] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, *Journal of computational physics* 115 (1) (1994) 200–212.
- [30] S. M. Millmore, N. Nikiforakis, L. Michael, *Computational continuum modelling lecture notes* (2022).
- [31] A. Harten, P. D. Lax, B. v. Leer, On upstream differencing and godunov-type schemes for hyperbolic conservation laws, *SIAM review* 25 (1) (1983) 35–61.
- [32] E. F. Toro, L. O. Müller, A. Siviglia, Bounds for wave speeds in the riemann problem: Direct theoretical estimates, *Computers & Fluids* 209 (2020) 104640.
- [33] C. R. Evans, J. F. Hawley, Simulation of magnetohydrodynamic flows—a constrained transport method, *The Astrophysical Journal* 332 (1988) 659–677.
- [34] A. Mignone, P. Tzeferacos, A second-order unsplit godunov scheme for cell-centered mhd: The ctu-glm scheme, *Journal of Computational Physics* 229 (6) (2010) 2117–2138.
- [35] J. M. Stone, J. F. Hawley, C. R. Evans, M. L. Norman, A test suite for magnetohydrodynamical simulations, *The Astrophysical Journal* 388 (1992) 415–437.
- [36] K. G. Powell, An approximate riemann solver for magnetohydrodynamics: (that works in more than one dimension), *Upwind and high-resolution schemes* (1997) 570–583.
- [37] A. Mignone, P. Tzeferacos, G. Bodo, High-order conservative finite difference glm–mhd schemes for cell-centered mhd, *Journal of Computational Physics* 229 (17) (2010) 5896–5920.
- [38] T. Miyoshi, K. Kusano, A multi-state hll approximate riemann solver for ideal magnetohydrodynamics, *Journal of Computational Physics* 208 (1) (2005) 315–344.
- [39] W. Dai, P. R. Woodward, A high-order godunov-type scheme for shock interactions in ideal magnetohydrodynamics, *SIAM*

Journal on Scientific Computing 18 (4) (1997) 957–981.