Commun. Comput. Phys.
doi: 10.4208/cicp.2009.08.154

Vol. 7, No. 3, pp. 473-509
March 2010

Finite Volume Methods for Wave Propagation in Stratified Magneto-Atmospheres

Franz Georg Fuchs[1], Andrew D. McMurry[1], Siddhartha Mishra[1,*], Nils Henrik Risebro[1] and Knut Waagan[2]

[1] *Centre of Mathematics for Applications (CMA), University of Oslo, P.O. Box 1053, Blindern, N-0316 Oslo, Norway.*
[2] *High Altitude Observatory (HAO), National Center of Atmospheric Research, P.O. Box 3000, Boulder, 80307-3000 Colorado, USA.*

Received 26 September 2008; Accepted (in revised version) 7 May 2009

Communicated by Chi-Wang Shu

Available online 1 September 2009

Abstract. We present a model for simulating wave propagation in stratified magneto-atmospheres. The model is based on equations of ideal MHD together with gravitational source terms. In addition, we present suitable boundary data and steady states to model wave propagation. A finite volume framework is developed to simulate the waves. The framework is based on HLL and Roe type approximate Riemann solvers for numerical fluxes, a positivity preserving fractional steps method for discretizing the source and modified characteristic and Neumann type numerical boundary conditions. Second-order spatial and temporal accuracy is obtained by using an ENO piecewise linear reconstruction and a stability preserving Runge-Kutta method respectively. The boundary closures are suitably modified to ensure mass balance. The numerical framework is tested on a variety of test problems both for hydrodynamic as well as magnetohydrodynamic configurations. It is observed that only suitable choices of HLL solvers for the numerical fluxes and balanced Neumann type boundary closures yield stable results for numerical wave propagation in the presence of complex magnetic fields.

AMS subject classifications: 65M06, 35L45, 8508

Key words: Solar atmospheres, MHD equations, finite volume methods, non-reflecting boundary conditions.

*Corresponding author. *Email addresses:* `franzf@math.uio.no` (F. G. Fuchs), `a.d.mcmurry@ifi.uio.no` (A. D. McMurry), `siddharm@cma.uio.no` (S. Mishra), `nilshr@math.uio.no` (N. H. Risebro), `knutwa@ucar.edu` (K. Waagan)

1 Introduction

There is considerable interest in the astrophysics community regarding the problem of wave propagation in magnetized stellar atmospheres. The main theme of this research is to determine how convection generated waves transport and deposit energy in the overlying chromospheric and coronal plasmas. Models aim to explain the observed energy distribution in interesting astrophysical objects like the sun. The mathematical description of the underlying physical processes in realistic magneto-atmospheres is extremely complicated. The models for this wave heating problem include the equations of ideal magneto-hydrodynamics (MHD) together with complicated source and diffusion terms. In addition, radiative transfer and non-equilibrium thermodynamics also play leading roles. These models are described by systems of nonlinear partial differential and integral equations in three dimensions, coupled with realistic initial and boundary data. The issue of determining model parameters along with proper initial and boundary data requires considerable observational work.

It is not possible to obtain analytical solutions for the full model or even extremely simplified versions of it. Also for these equations, theoretical results concerning existence, uniqueness and qualitative behavior are currently unavailable. Therefore, in order to investigate these models, one must resort to numerical methods. Even this task faces formidable difficulties due to nonlinearity and sheer computational complexity. A detailed account of the physical processes involved in wave propagation along with extensive references to the corresponding astrophysics literature can be found in [1, 2].

In [1,2], the authors consider a relatively simple model for wave propagation in the solar atmosphere. This model takes into account the equations of compressible ideal MHD along with gravitational source terms, supplemented by a description of the underlying steady states. Waves in the "solar" atmosphere are modeled by inducing perturbations of these steady states. We adopt the modeling framework of the above papers as a starting point of this work and develop a class of schemes of the finite volume type to simulate this model. A complete description of these schemes involves suitable approximate Riemann solvers for the ideal MHD equations, proper treatment of the gravitational source term and an appropriate implementation of boundary conditions.

The core of the model we consider consists of the equations of ideal MHD. Consequently, most of the computational effort is directed at MHD solvers. The MHD equations are an example of a system of non-linear hyperbolic conservation laws. Solutions of these equations develop discontinuities such as shock waves and contact discontinuities even for smooth initial data. Furthermore, the MHD equations are not strictly hyperbolic and contain a large number of waves. Some of the characteristic fields are not convex (i.e. genuinely nonlinear except in some subset of state space), and the resulting solutions can have intermediate and compound shocks. All these issues have to be addressed in order to design efficient numerical methods for ideal MHD.

Finite volume methods are a popular type of numerical framework for approximating solutions to conservation laws. These methods are based on approximating the integral form of the conservation law on each cell or control volume. Numerical fluxes at each cell interface are based on either exact or approximate solutions of Riemann problems at the interface. Higher order of accuracy in space can be obtained by using non-oscillatory ENO/WENO type piecewise polynomial functions in each cell. Higher order accuracy in time is obtained by using suitable Runge-Kutta solvers. A detailed account of these methods is presented in [3].

In this paper, we will consider approximate Riemann solvers of the HLL type (see [3]) when constructing numerical fluxes. These solvers are based on approximating the solution of Riemann problems by a piecewise constant function containing fewer discontinuities than there are waves in the exact solution. A complete description of the solver is provided by specifying the wave speeds and intermediate states. The ideal MHD equations have seven waves (in one space dimension). Typical HLL solvers for the MHD equations involving 2, 3, 5 and 7 waves have been developed. Among these are the 3 wave solvers of [4, 5], the 5 wave solver of [6] and the 3, 5 and 7 wave solvers of [7, 8]. The latter ones are based on an extended Suliciu type relaxation system.

Another alternative is to use linearized approximate Riemann solvers. In this case, the exact solution of Riemann problems at each cell interface is replaced by an exact solution of a suitable linearization of the non-linear flux function across the interface. The resulting solvers are denoted as Roe type approximate Riemann solvers for systems of conservation laws (see [3] for details). These solvers are known to have high accuracy, in particular they resolve isolated discontinuities exactly. Roe type solvers for ideal MHD have been developed in [10–12] and other references therein.

When choosing an approximate Riemann solver, the key issues are computational cost, accuracy and stability. One essential stability criteria for ideal MHD is maintaining positive density and pressure. Another desirable property for a solver is that a discrete entropy condition should hold, in which case the solver is said to be entropy stable. In one space dimension, the standard HLL two wave solver, three wave solver of [5], five wave solver of [6] and the relaxation based solvers of [7, 8] are provably positivity preserving[†]. The three wave solver of [4] has not been proven to be positivity preserving. It is well known that Roe type solvers are not positivity preserving. The solvers of [7, 8] are also entropy stable. However, for MHD all these results are valid in only one space dimension.

The above schemes are only first-order accurate in both space and time. We use standard piecewise linear ENO [13] reconstruction to obtain second-order spatial accuracy. Second-order accuracy in time is obtained by using the strong stability preserving (SSP) Runge-Kutta method [14].

When simulating stratified atmospheres, the numerical treatment of the gravitational source term has to be taken into account. The presence of this source term leads to interesting steady states that need to be preserved. Furthermore, waves are modeled as perturbations of these steady states. The pressure and density at a steady state decay exponentially in the vertical direction due to the presence of gravity, which leads to very

[†] We define a positivity preserving scheme as a scheme that ensures that density and pressure remains positive under a suitable CFL-condition.

low densities and pressures at the top of the computational domain. Since preserving positivity is a challenge for any solver, this problem is particularly acute in stratified atmospheres. We treat the source term by using the method of fractional steps and identify suitable stability conditions and discretizations which still keep the HLL solvers positivity preserving. Second-order accurate fractional steps is obtained by using Strang splitting [3].

The next issue that needs to be addressed for the simulation of waves is that of boundary conditions. For simplicity, consider a two-dimensional model with x- and z-directions being associated with the horizontal and vertical directions respectively. Since we are interested in simulating a small part of the solar atmosphere, we use periodic boundary conditions at the x-boundary. At the bottom z-boundary, we have to prescribe boundary conditions in order to model incoming waves. At the top z-boundary, we need to implement numerical boundary conditions that minimize reflections when waves (from the bottom of the domain) reach this boundary. Furthermore, this boundary should not generate any waves or numerical artifacts that affect the quality of the solution in the interior of the domain. The vertical boundary conditions have to retain the mass balance and not lead to conservation errors. Conservation errors will force solutions away from the steady state. Since the solutions have very low pressures at the top boundary, conservation errors can easily lead to negative pressures and densities.

We adapt the non-reflecting characteristic boundaries developed in [15–17] to this problem. The adaptation is complicated on account of the complex eigenstructure and the non-strict hyperbolicity of the equations. We also introduce an alternative implementation of the non-reflecting boundary conditions of [15, 16]. Additionally, we have to adapt the boundary conditions in order to ensure mass balance. Another alternative implementation of boundary conditions is to modify simple Neumann type numerical boundary conditions to enforce mass balance. This choice might lead to greater amount of reflections. Furthermore, we propose a novel set of boundary conditions based on extrapolating (by an exponential profile) the pressure and the density from the interior of the domain into the ghost cells together with Neumann type conditions for the velocity and magnetic fields. We compare all the boundary conditions in numerical experiments.

The aim of the paper is to develop a finite volume based numerical framework that simulates wave propagation in an idealized magneto-atmosphere. We implement different first- and second-order accurate HLL type solvers as well as Roe solvers for the finite volume fluxes and compare them in a series of experiments. The sources are discretized by using a method of fractional steps that keeps pressures and densities positive. The model is completed by boundary conditions of the characteristic type as well as Neumann type that are modified to ensure mass balance in the domain. All the above ingredients are combined together to obtain a finite volume framework that can simulate waves in stratified magneto-atmospheres. We emphasize that a successful simulation of wave propagation requires proper choices of all the above ingredients and we present them in this paper.

The rest of the paper is organized as follows: in Section 2, we present the model of the idealized solar atmosphere and specify the detailed eigenstructure of ideal MHD equations. The approximate Riemann solvers are described and compared in a series of numerical experiments in Section 3. In Section 4, we explain the numerical approximation of the gravity source term and provide a complete description of the boundary conditions in Section 5. Numerical experiments for wave propagation in the idealized solar atmosphere are provided in Section 6.

2 The model

In this section, we present the equations and initial and boundary conditions modeling an idealized solar atmosphere. The basic equations of the model are the equations of ideal MHD along with source terms due to gravity given by

$$\rho_t+\mathrm{div}(\rho\mathbf{u})=0, \tag{2.1a}$$

$$(\rho\mathbf{u})_t+\mathrm{div}\Big(\rho\mathbf{u}\otimes\mathbf{u}+(p+\frac{1}{2}|\mathbf{B}|^2)I-\mathbf{B}\otimes\mathbf{B}\Big)=-\rho g\mathbf{e}_3, \tag{2.1b}$$

$$\mathbf{B}_t+\mathrm{div}(\mathbf{u}\otimes\mathbf{B}-\mathbf{B}\otimes\mathbf{u})=0, \tag{2.1c}$$

$$E_t+\mathrm{div}\Big((E+p+\frac{1}{2}|\mathbf{B}|^2)\mathbf{u}-(\mathbf{u}\cdot\mathbf{B})\mathbf{B}\Big)=-\rho g(\mathbf{u}\cdot\mathbf{e}_3), \tag{2.1d}$$

$$\mathrm{div}(\mathbf{B})=0, \tag{2.1e}$$

where ρ is the density, $\mathbf{u}=\{u^1,u^2,u^3\}$ and $\mathbf{B}=\{B^1,B^2,B^3\}$ are the velocity and magnetic fields respectively, p is the thermal pressure, g is constant acceleration due to gravity , $\mathbf{e}_3=\{0,0,1\}$, E is the total energy determined by an ideal gas equation of state of the form,

$$E=\frac{p}{\gamma-1}+\frac{1}{2}\rho|\mathbf{u}|^2+\frac{1}{2}|\mathbf{B}|^2, \tag{2.2}$$

where γ is the adiabatic gas constant. The above equations describe the conservation of mass, momentum and energy and the evolution of the magnetic field due to the velocity. In addition, magnetic monopoles have not been observed in nature and this fact is modeled by the constraint that the divergence of the magnetic field remains zero during the evolution.

In condensed form, the above equations (2.1) can be written as a system of balance laws of the form,

$$U_t+(f(U))_x+(g(U))_y+(h(U))_z=S(U), \tag{2.3}$$

where U is the vector of conserved variables, f, g and h are the directional fluxes and S is the source.

For simplicity, we consider the equations in two dimensions. The x coordinate represents the horizontal direction and the z coordinate the vertical direction. In particular this means that no variable depends on y, i.e., $\partial_y\equiv 0$ in (2.1). We consider (2.1) in the domain

$[0,X]\times[0,Z]$ where X and Z are positive numbers. Next, we specify steady states (stationary solutions) that are of interest as they will serve as a background for the propagation of waves.

2.1 Hydrodynamic steady state

To begin with, we assume that the magnetic field $\mathbf{B}$ is set to zero implying that the model is driven by ideal compressible hydrodynamics. In addition, the atmosphere is assumed to be steady by setting the velocity field $\mathbf{u}$ to zero. With this ansatz the pressure and the density have to satisfy the following ordinary differential equation

$$\frac{\partial p}{\partial z}=-\rho g. \tag{2.4}$$

We look for solutions of (2.4) satisfying $p(x,z)=c\rho(x,z)$ for some constant c and for all x and z, which amounts to assuming an isothermal atmosphere. This is a reasonable approximation since we are interested in simulating the region around the lower chromosphere of the sun where the temperature remains approximately constant. Substituting this into (2.4) leads to the following hydrodynamic steady state,

$$\mathbf{u}=0, \quad \mathbf{B}=0, \quad \rho(x,z)=\rho_0 e^{-z/H}, \quad p(x,z)=p_0 e^{-z/H}, \tag{2.5}$$

where the scale height H is given by $H=p_0/(g\rho_0)$ and p_0 and ρ_0 are the values of the pressure and density at the bottom boundary of the domain. Observe that the pressure and density decay exponentially with height, giving very low values near the top of the computational domain. As a consequence, when we are performing numerical calculations of small perturbations of this state, retaining positivity of pressure and density (particularly at the top of the computational domain) is going to be a key difficulty.

2.2 Magnetic steady states

Any realistic description of the solar atmosphere has to include magnetic fields. Hence, we need to calculate steady states of (2.1) with non-trivial magnetic fields. Momentum balance in (2.1) can also be written as

$$(\rho\mathbf{u})_t+\operatorname{div}(\rho\mathbf{u}\otimes\mathbf{u}+pI)=\operatorname{curl}(\mathbf{B})\times\mathbf{B}-\rho g\mathbf{e}_3.$$

This form makes the role of gravity and the Lorentz force explicit. The magnetic steady state is a stationary solution of (2.1) with the additional ansatz that $p=c\rho$, $\operatorname{curl}(\mathbf{B})=0$ and $\mathbf{u}=0$. This corresponds to stationary and Lorentz-force free fields. Substituting the above ansatz into (2.1), we obtain that the density and the pressure is given by (2.5). Furthermore, since the magnetic field $\mathbf{B}$ is assumed to be such that $\operatorname{curl}(\mathbf{B})\equiv 0$ and $\operatorname{div}(\mathbf{B})\equiv 0$, it can be expressed in terms of vector harmonic functions. As we consider a small part of the solar atmosphere and assume periodic boundary conditions in the horizontal direction, we choose to express the magnetic field in terms of a finite number of modes in a Fourier expansion. A resulting steady state is given by

$$\begin{aligned} &\mathbf{u}=0, \quad B_2=0, \quad p(x,z)=p_0 e^{-z/H}, \quad \rho(x,z)=\rho_0 e^{-z/H},\\ &B^1(x,z)=\sum_{k=0}^{M} f_k \sin\left(\frac{2k\pi x}{X}\right) e^{-2\pi k z/X},\\ &B^3(x,z)=\sum_{k=0}^{M} f_k \cos\left(\frac{2k\pi x}{X}\right) e^{-2\pi k z/X}, \end{aligned} \tag{2.6}$$

where f_k's are the Fourier coefficients corresponding to the data $B^1(x,0)$ and $B^3(x,0)$ at the bottom boundary and M is the total number of Fourier modes for the boundary data. We choose $B_2 \equiv 0$ as the resulting magnetic field is planar. This is done for simplicity.

A simple calculation shows that (2.6) is indeed a steady state of (2.1). Furthermore, the pressure and density have an exponential decay along the vertical direction. The magnetic field is quite complicated and leads to a genuinely multi-dimensional description of the model. These factors complicate design of numerical schemes.

2.3 The characteristic structure of ideal MHD

For the sake of completeness we give some details regarding the eigensystem of the MHD equations. Consider Eq. (2.1) in the x-direction without gravity i.e., $g=0$. The divergence constraint in one space dimension forces the magnetic field in x direction, B^1, to be constant in space and time, and thus act only as a parameter in the equations. Defining the vector of primitive variables,

$$V=\{\rho,u^1,u^2,u^3,B^2,B^3,p\},$$

the system (2.1) reduces to the following quasilinear form in one dimension,

$$V_t + A_1(V)V_x = 0. \tag{2.7}$$

For the precise expression for the Jacobian matrix A_1, see [18]. Defining the speeds

$$\begin{aligned} &a^2=\frac{\gamma p}{\rho}, \quad b_{1,2,3}=\frac{B^{1,2,3}}{\sqrt{\rho}},\\ &b^2=b_1^2+b_2^2+b_3^2, \quad b_\perp^2=b_2^2+b_3^2, \end{aligned}$$

the eigenvalues of A_1 read

$$\begin{aligned} &\lambda_1=u_1-c_f, \quad \lambda_2=u_1-b_1, \quad \lambda_3=u_1-c_s, \quad \lambda_4=u_1,\\ &\lambda_5=u_1+c_s, \quad \lambda_6=u_1+b_1, \quad \lambda_7=u_1+c_f, \end{aligned}$$

where c_f and c_s are given by

$$\begin{aligned} c_f^2 &= \frac{1}{2}\left(a^2+b^2+\sqrt{(a^2+b^2)^2-4a^2b_1^2}\right),\\ c_s^2 &= \frac{1}{2}\left(a^2+b^2-\sqrt{(a^2+b^2)^2-4a^2b_1^2}\right). \end{aligned} \tag{2.8}$$

The waves corresponding to λ_1 and λ_7 are called fast waves, the ones corresponding to λ_3 and λ_5 slow waves, those corresponding to λ_2 and λ_6 Alfvén waves and the wave associated with λ_4 is called a contact wave. As the above eigenvalues are real, the system is hyperbolic. But the eigenvalues are not always distinct, and the system is not strictly hyperbolic. This non-strict hyperbolicity is a formidable obstacle to the development of mathematical theory and numerical methods for MHD.

It is well known that the eigenvectors of (2.7) have to be scaled properly in order to be well-defined. We now present the orthonormal set of eigenvectors first described in [18]. The right and left eigenvectors corresponding to the contact wave λ_4 are given by,

$$r_e=(1,0,0,0,0,0,0)^T, \quad l_e=\frac{1}{a^2}\left(a^2,0,0,0,0,0,-1\right).$$

Define $\beta_{2,3}=b_{2,3}/b_\perp$. Then, the eigenvectors corresponding to the Alfvén waves λ_2 and λ_6 are given by

$$\begin{aligned} r_A^\pm &= \left(0,0,\pm\beta_3,\mp\beta_2,-\sqrt{\rho}\beta_3\mathrm{sign}(b_1),\sqrt{\rho}\beta_2\mathrm{sign}(b_1),0\right)^T,\\ l_A^\pm &= \frac{1}{2}\left(0,0,\pm\beta_3,\mp\beta_2,-\frac{\beta_3}{\sqrt{\rho}}\mathrm{sign}(b_1),\frac{\beta_2}{\sqrt{\rho}}\mathrm{sign}(b_1),0\right), \end{aligned}$$

where r_A^+ and l_A^+ correspond to λ_2.

As in [18], we introduce the following normalizing factors,

$$\alpha_f^2=\frac{a^2-c_s^2}{c_f^2-c_s^2}, \quad \alpha_s^2=\frac{c_f^2-a^2}{c_f^2-c_s^2}.$$

Note that $\alpha_f^2+\alpha_s^2=1$. The eigenvectors corresponding to the fast and slow waves read,

$$r_f^\pm=\begin{pmatrix} \alpha_f\rho\\ \pm\alpha_f c_f\\ \mp\alpha_s c_s\beta_2\mathrm{sign}(b_1)\\ \mp\alpha_s c_s\beta_3\mathrm{sign}(b_1)\\ \alpha_s\sqrt{\rho}a\beta_2\\ \alpha_s\sqrt{\rho}a\beta_3\\ \alpha_f\rho a^2 \end{pmatrix}, \quad r_s^\pm=\begin{pmatrix} \alpha_s\rho\\ \pm\alpha_s c_s\\ \mp\alpha_f c_f\beta_2\mathrm{sign}(b_1)\\ \mp\alpha_f c_f\beta_3\mathrm{sign}(b_1)\\ \alpha_f\sqrt{\rho}a\beta_2\\ \alpha_f\sqrt{\rho}a\beta_3\\ \alpha_s\rho a^2 \end{pmatrix},$$

$$l_f^{\pm}=\frac{1}{2a^2}\left(0,\pm\alpha_f c_f,\mp\alpha_s c_s\beta_2\mathrm{sign}(b_1),\mp\alpha_s c_s\beta_3\mathrm{sign}(b_1),\alpha_s a\frac{\beta_2}{\sqrt{\rho}},\alpha_s a\frac{\beta_3}{\sqrt{\rho}},\frac{\alpha_f}{\rho}\right),$$
$$l_s^{\pm}=\frac{1}{2a^2}\left(0,\pm\alpha_s c_s,\mp\alpha_f c_f\beta_2\mathrm{sign}(b_1),\mp\alpha_f c_f\beta_3\mathrm{sign}(b_1),\alpha_f a\frac{\beta_2}{\sqrt{\rho}},\alpha_f a\frac{\beta_3}{\sqrt{\rho}},\frac{\alpha_s}{\rho}\right).$$

The normalization factors α_f and α_s are not well-defined at the triple point where $b_1=a$ and $b_\perp=0$. In this case, we use the fact that $\alpha_f^2+\alpha_s^2=1,\beta_2^2+\beta_3^2=1$ and define $\alpha_f=\alpha_s=\beta_2=\beta_3=1/\sqrt{2}$.

3 Approximate Riemann solvers

In this section we review the different approximate Riemann solvers that we want to test. Consider the ideal MHD equations (2.1) with $g=0$ in the domain $[0,X]\times[0,Z]$. We divide this into $I\times K$ cells (or control volumes), and define $x_i=-\Delta x/2+i\Delta x$, $i=0,\cdots,I+1$, and $z_k=-\Delta z/2+k\Delta z$, $k=0,\cdots,K+1$, so that $z_{k+1/2}=k\Delta z$. Let $I_{i,k}$ denote the control volume $[x_{i-1/2},x_{i+1/2})\times[z_{k-1/2},z_{k+1/2})$. The cell average of the vector of conserved variables at time t^n over the cell $I_{i,k}$ is given by $U_{i,k}^n$. Let $\Delta=\min\{\Delta x,\Delta z\}$, then the time step Δt^n is chosen by the CFL condition,

$$\max_{i,k}\left\{\left|u_{i,k}^{1,n}\right|+c_{i,k}^{f1,n},\left|u_{i,k}^{3,n}\right|+c_{i,k}^{f3,n}\right\}\frac{\Delta t^n}{\Delta}\le 1, \tag{3.1}$$

where $c_{i,k}^{fj,n}$ is the fast speed in the j-th direction in the cell $I_{i,k}$. Then, a general finite volume scheme is written,

$$\begin{aligned}U_{i,k}^{n+1}&=\mathcal{F}\left(U_{i-1,k}^n,U_{i,k-1}^n,U_{i,k}^n,U_{i+1,k}^n,U_{i,k+1}^n\right)\\&=U_{i,k}^n-\frac{\Delta t^n}{\Delta x}\left(F_{i+1/2,k}^n-F_{i-1/2,k}^n\right)-\frac{\Delta t^n}{\Delta z}\left(H_{i,k+1/2}^n-H_{i,k-1/2}^n\right),\end{aligned} \tag{3.2}$$

where the numerical fluxes are functions of the neighboring cell averages, i.e.,

$$F_{i+1/2,k}^n=F\left(U_{i,k}^n,U_{i+1,k}^n\right),\quad H_{i,k+1/2}^n=H\left(U_{i,k}^n,U_{i,k+1}^n\right).$$

These numerical fluxes should be such that $F(A,B)$ is an approximation to the solution at $x=0$ of the Riemann problem in the x direction for (2.1). To be concrete, if U satisfies

$$U_t+f(U)_x=0,\quad U(x,0)=\begin{cases}U_L & x<0,\\ U_R & x>0,\end{cases}$$

then $F(U_L,U_R)\approx f(U(0,t))$. The numerical flux $H(U_L,U_R)$ is defined analogously.

We will use approximate Riemann solvers of the HLL type, see [3], in order to define the numerical fluxes.

3.1 HLL2 solver

The number "2" denoting this approximate Riemann solver means that we approximate the full Riemann solution using two moving discontinuities (waves). Let $U_{L,R}$ and $f_{L,R}$ denote the left and right state and flux respectively. If the speed of the two waves are s_L and s_R then the constant state in between these waves, U_* will be determined from conservation by

$$f_R - f_* = s_R(U_R - U_*), \quad f_* - f_L = s_L(U_* - U_L),$$

where f_* denotes the numerical flux between these two waves. Solving the above equations yields

$$U_* = \frac{f_R - f_L - s_R U_R + s_L U_L}{s_L - s_R}, \quad f_* = \frac{s_R f_L - s_L f_R + s_L s_R (U_R - U_L)}{s_R - s_L}. \tag{3.3}$$

Then, the numerical flux can be written as

$$F^{\text{hll2}}(U_L, U_R) = \begin{cases} f_L & \text{if } s_L \geq 0, \\ f_* & \text{if } s_L < 0 < s_R, \\ f_R & \text{if } s_R \leq 0. \end{cases} \tag{3.4}$$

It remains to define the waves speeds s_L and s_R, and we do this as in [19]. Let $\bar{U} = (U_L + U_R)/2$, then the wave speeds are given by

$$s_L = \min\left\{u_L^1 - c_{fL}, \bar{u}_1 - \bar{c}_f\right\}, \quad s_R = \max\left\{u_R^1 + c_{fR}, \bar{u}_1 + \bar{c}_f\right\}, \tag{3.5}$$

where $\bar{u}_1$ and $\bar{c}_f$ are the normal velocity and the fast wave speed of $A(\bar{U})$ respectively. This solver only approximates the outermost (fast) waves of the Riemann solution and can be very dissipative, particularly at contact discontinuities.

3.2 HLL3 solvers

This type of solver uses three moving discontinuities to approximate the solution of the Riemann problem. Let the $U_{L,R}$ and $f_{L,R}$ be as before. With this notation a HLL3 solver gives the approximate solution

$$U^{\text{hll3}} = \begin{cases} U_L & \text{if } s_L \geq 0, \\ U_L^* & \text{if } s_L < 0 < s_M, \\ U_R^* & \text{if } s_M < 0 < s_R, \\ U_R & \text{if } s_R \leq 0, \end{cases} \quad F^{\text{hll3}}(U_L, U_R) \; = \begin{cases} f_L & \text{if } s_L \geq 0, \\ f_L^* & \text{if } s_L < 0 < s_M, \\ f_R^* & \text{if } s_M < 0 < s_R, \\ f_R & \text{if } s_R \leq 0, \end{cases} \tag{3.6}$$

where the outer wave speeds s_L and s_R are given by (3.5) and the middle wave speed is given by s_M. Local conservation implies,

$$s_L U_L^* - f_L^* = s_L U_L - f_L, \quad s_M U_R^* - f_R^* = s_M U_L^* - f_L^*, \quad s_R U_R - f_R = s_R U_R^* - f_R^*. \tag{3.7}$$

From this we obtain the following expressions for f_L^* and f_R^*

$$f_L^* = f_* - \frac{s_L(s_R - s_M)}{s_R - s_L}\Delta U^*, \quad f_R^* = f_* + \frac{s_R(s_M - s_L)}{s_R - s_L}\Delta U^*, \tag{3.8}$$

where $\Delta U^* = U_R^* - U_L^*$ and f_* is given by (3.3).

3.2.1 HLL3L solver

In [4], the author determines the two middle states by making the following *ansatz*,

$$\Delta U^* = \alpha(U_R - U_L),$$

where α is a parameter to be determined. If we set $\alpha = 0$ the middle wave disappears and we get the $HLL2$ solver. When $\alpha = 1$, an HLL3 solver will give a single discontinuity approximating the contact wave.

Since the middle wave is supposed to model the contact discontinuity, we can choose the middle wave speed as the corresponding velocity of the Jacobian $A(\bar{U})$, i.e., $s_M = \bar{u}^1$. Set $c_* = |\bar{c}_f - s_M|$ and

$$s = \frac{|\Delta f - s_M \Delta U|}{|\Delta U|},$$

where $\Delta U = U_R - U_L$ and $\Delta f = f_R - f_L$, the interpolation factor α is then defined as

$$\alpha = \max\left\{0, 1 - \frac{s}{c_*}\right\}.$$

This heuristic choice of α, which was suggested in [4], is motivated by the following argument. If the true solution of

$$V_t + A(\bar{U})V_x = 0, \quad V(x,0) = \begin{cases} U_L & x < 0, \\ U_R & x > 0, \end{cases}$$

consisted of a single contact discontinuity, then $\Delta f = s_M \Delta U$, which gives $s = 0$ and $\alpha = 1$. Similarly, if the solution consisted of a single fast shock with speed $s_M + \bar{c}_f$ (a particular form of the HLL2 solution) $s = c_*$, which gives $\alpha = 0$.

Although the pressure and density computed by this solver are not provably positive, the HLL3L solver gives satisfactory numerical results on many test problems (see Section 6).

3.2.2 HLL3G solver

A positivity preserving HLL three wave solver was proposed by Gurski in [5]. We use the notation of the previous section and the fluxes are defined in (3.6). We choose s_L and s_R as the Einfeldt speeds of (3.5) and the middle speed s_M as the normal Roe velocity as in the HLL3L solver. Then from the conservation equations (3.7), we can express the middle states in terms of s_M,

$$\begin{aligned}
&\rho_L^* = \rho_L \frac{(s_L - u_L^1)}{(s_L - s_M)}, \quad \rho_R^* = \rho_R \frac{(s_R - u_R^1)}{(s_R - s_M)}, \\
&p^* = p_L + \rho_L (u_L^1 - s_L)(u_L^1 - s_M) + \frac{|\mathbf{B}_L|^2}{2}.
\end{aligned} \tag{3.9}$$

Note that the pressure does not jump across the contact discontinuity modeled by the middle wave. Since we are considering (2.1) in one dimension, B^1 is taken to be a constant. The tangential components of the velocity and the magnetic field do not jump across the contact discontinuity and hence

$$u_L^{2,*} = u_R^{2,*} = u^{2,*}, \quad B_L^{2,*} = B_R^{2,*} = B^{2,*}.$$

Identical expressions hold for u^3 and B^3, and we substitute them into the conservation equations in order to get (in the case when $B^1 \neq 0$),

$$\begin{aligned}
u^{2,*} &= \frac{\rho_L u_L^2 (u_L^1 - s_L) - \rho_R u_R^2 (u_R^1 - s_R) + 0.5(B^1 (B_R^2 - B_L^2)}{\rho_L (u_L^1 - s_L) - \rho_R (u_R^1 - s_R)}, \\
u^{3,*} &= \frac{\rho_L u_L^3 (u_L^1 - s_L) - \rho_R u_R^3 (u_R^1 - s_R) + 0.5(B^1 (B_R^3 - B_L^3)}{\rho_L (u_L^1 - s_L) - \rho_R (u_R^1 - s_R)}, \\
B^{2,*} &= \frac{B_L^2 (u_L^1 - s_L) - B_R^2 (u_R^1 - s_R) + B^1 (u_R^2 - u_L^2)}{s_R - s_L}, \\
B^{3,*} &= \frac{B_L^3 (u_L^1 - s_L) - B_R^3 (u_R^1 - s_R) + B^1 (u_R^3 - u_L^3)}{s_R - s_L}, \\
E_L^* &= \frac{0.5(|\mathbf{B}_L|^2 u_L^1) + (p_L u_L^1 + E_L (u_L^1 - s_L) - s_M p^*) + B^1 (\mathbf{B}_L^* \cdot \mathbf{u}_L^* - \mathbf{B}_L \cdot \mathbf{u}_L)}{s_M - s_L}, \\
E_R^* &= \frac{0.5(|\mathbf{B}_R|^2 u_R^1) + (p_R u_R^1 + E_R (u_R^1 - s_R) - s_M p^*) + B^1 (\mathbf{B}_R^* \cdot \mathbf{u}_R^* - \mathbf{B}_R \cdot \mathbf{u}_R)}{s_M - s_R}.
\end{aligned}$$

If $B = 0$ we can find the relevant formulas in [5]. The pressure and density computed using this solver are always positive. In [5], the author noted that a modified version of this solver resolves Alfvén and slow waves better, but we found that this modification might lead to negative pressures in some of our simulations, and hence we use the original version of this solver.

3.2.3 HLL3R solver

In [7,8], an MHD 3-wave solver is derived from a relaxation system as an extension of the work for hydrodynamics in [9]. This solver is positivity preserving, and it also satisfies a discrete entropy inequality. We refer to [7,8] for justification, and only describe the solver here. First assume that the outer wave speeds $s_L = u_L - c_L/\rho_L$ and $s_R = u_R + c_R/\rho_R$ are given for some positive c_L, c_R. Define the relaxation pressure

$$\text{ß}_{L,R} = \left(p + \frac{1}{2}\mathbf{B}^2 - (B^1)^2, -B^1 B^2, -B^1 B^3 \right)_{L,R}.$$

Then define the intermediate states

$$\mathbf{u}_* = \mathbf{u}_*^L = \mathbf{u}_*^R = \frac{c_L \mathbf{u}_L + c_R \mathbf{u}_R + \pi_L - \pi_R}{c_L + c_R},$$
$$\pi_* = \pi_*^L = \pi_*^R = \frac{c_R \pi_L + c_L \pi_R + c_L c_R (\mathbf{u}_L - \mathbf{u}_R)}{c_L + c_R}.$$

These are constant across the middle wave, so we can define the middle wave speed $s_M = u_*^1$. The intermediate state values for ρ, $B^{2,3}$ and E are given by B^1 being constant, and

$$\frac{\mathbf{B}_{L,R}}{\rho_{L,R}} - B^1 \frac{\pi_{L,R}}{c_{L,R}^2} = \frac{\mathbf{B}_{L,R}^*}{\rho_{L,R}^*} - B^1 \frac{\pi_*}{c_{L,R}^2},$$
$$e_{L,R} + \frac{(\mathbf{B}_{L,R})^2}{2\rho_{L,R}} - \frac{\pi_{L,R}^2}{2c_{L,R}^2} = e_{L,R}^* + \frac{(\mathbf{B}_{L,R}^*)^2}{2\rho_{L,R}^*} - \frac{\pi_*^2}{2c_{L,R}^2}.$$

Finally, adopting the notation of (3.6), the numerical flux is given by

$$f_{L,R}^* = \left(\rho u^1, \rho u^1 \mathbf{u} + \pi, u^1 \mathbf{B} - B^1 \mathbf{u}, E u^1 + \pi \cdot \mathbf{u} \right)_{L,R}^*.$$

The coefficients c_L and c_R are given by

$$c_L = \rho_L \left(a_L^0 + \frac{1}{2}(\gamma+1) X_L \right), \quad c_R = \rho_R \left(a_R^0 + \frac{1}{2}(\gamma+1) X_R \right)$$

with γ as in the equation of state (2.2),

$$X_L = (u_L^1 - u_R^1)^+ + \frac{(\pi_R^1 - \pi_L^1)^+}{\sqrt{\gamma \rho_L p_L} + \rho_R c_f^R}, \quad X_R = (u_L^1 - u_R^1)^+ + \frac{(\pi_L^1 - \pi_R^1)^+}{\sqrt{\gamma \rho_R p_R} + \rho_L c_f^L},$$

and $a_{L,R}^0$ given by (using the notation of (2.8))

$$(a_{L,R}^0)^2 = \frac{1}{2} \left(a^2 + \frac{b^2}{\xi} + \sqrt{\left(a^2 + \frac{b^2}{\xi} \right)^2 - 4a^2 \frac{b_1^2}{\xi}} \right)_{L,R},$$

with

$$\xi_{L,R} = \left(\frac{c_f + \frac{1}{2}(\gamma-1) X}{c_f + \frac{1}{2}(\gamma+1) X} \right)_{L,R}.$$

3.3 Roe Solver

Although our focus in this paper is on the approximate Riemann solvers of the HLL type, we briefly describe the Roe solver in order to compare it with the HLL solvers as well as to use it in our implementation of non-reflecting characteristic boundary conditions.

Given two states U_L,U_R across an interface, let $\hat{A}$ (see [11]) be the Roe matrix associated with U_L,U_R (the simple average $A(\frac{U_L+U_R}{2})$ also suffices for most practical purposes). Let $\hat{R},\hat{L}$ be the right and left eigenvector matrices associated with $\hat{A}$. We can use the Roe-Balsara eigensystem used in section 2 for the eigen-system decomposition, then the Roe solver [20] is given by,

$$F^{\rm Roe}(U_L,U_R)=\frac{1}{2}\Big(f_L+f_R-\hat{R}|\hat{\Lambda}|\hat{L}(U_R-U_L)\Big), \tag{3.10}$$

where $|\hat{\Lambda}|={\rm diag}\{|\hat{\lambda}_1|,\cdots,|\hat{\lambda}_7|\}$ with $\hat{\lambda}$'s being the eigenvalues of the Roe-matrix. This flux needs to be augmented with some standard entropy fix like the Harten [21] or Harten-Hyman [22] entropy fix in order to comply with the entropy condition. We use the Harten fix [21] in the numerical experiments presented later. It is well known that the Roe solver is a linearized, low dissipative solver that resolves isolated shocks exactly. Furthermore, it doesn't necessarily preserve positivity leading to problems when it is implemented in our solar atmosphere model.

Remark 3.1. The above HLL and Roe solvers are based on the one-dimensional form of the equations and require that the normal magnetic field B^i for $i=1,3$ is constant in each direction. However, in multi dimensional computations, the normal magnetic field is no longer constant. Hence, we need to suitably modify the solvers for multidimensional problems.

We use a simple modification, in the i-th direction, we use $B^i=(B^i_L+B^i_R)/2$ in our formulas to define the numerical fluxes in each direction. This simple modification does not guarantee that the solver remains positive but works reasonably well in all the numerical experiments that we performed. A careful modification of the solvers to make them handle genuinely multidimensional data requires a suitable discretization of the Powell source term, see [26] and, for HLL3R, [8].

3.4 Numerical experiments

We will compare all the above solvers in a series of numerical experiments. Furthermore, for all numerical examples in this paper we use $\gamma=5/3$.

Regarding the measurement of errors, if we have a reference solution available, then we define the relative error as

$$100\times\frac{||\alpha-\alpha_{\rm ref}||}{||\alpha_{\rm ref}||},$$

where α is (a component of) the numerical approximation and $\alpha_{\rm ref}$ is (the same component of) the reference solution, and $||\cdot||$ is some norm.

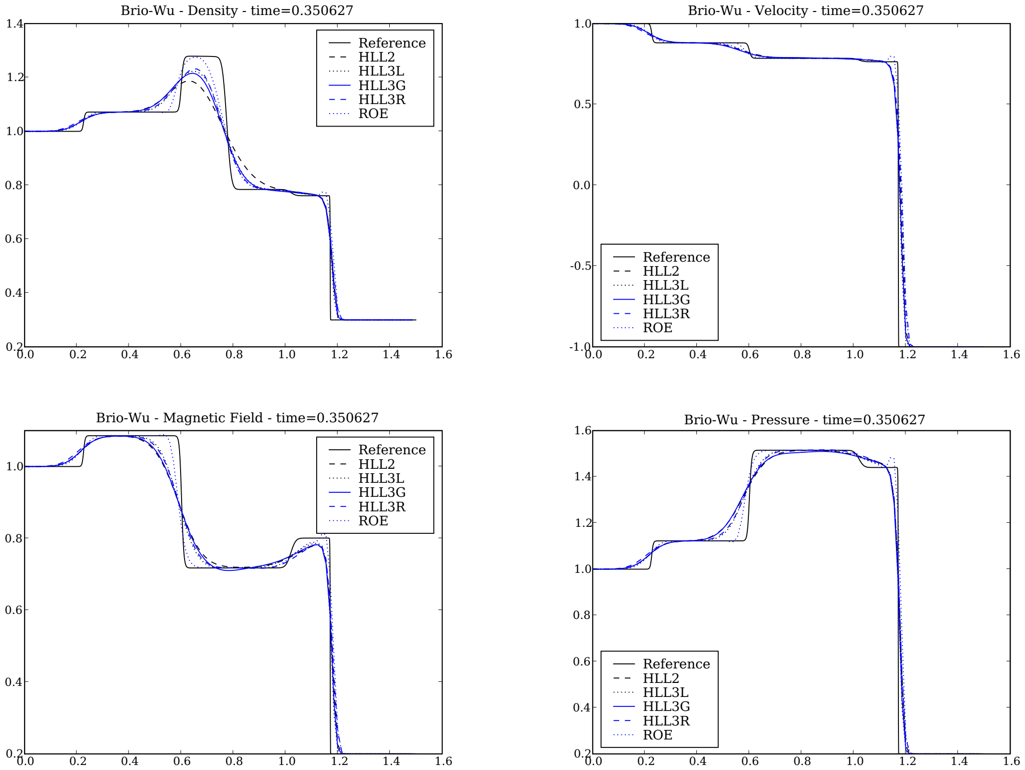


Figure 1: Results for the Brio-Wu shock tube for 200 mesh points at time $t=0.35$. Top left: ρ, top right: u^1, bottom left: B^2 and bottom right: P.

The Brio-Wu shock tube

We start with a one-dimensional test case proposed in [10]. The initial data are given by

$$\rho_0=\begin{cases}1.0 & \text{if } x<1.0,\\ 0.3 & \text{if } x\geq 1.0,\end{cases}\quad u_0^1=0,\quad u_0^2=0,\quad u_0^3=0,$$

$$B_0^1=0.7,\quad B_0^2=\begin{cases}0.0 & \text{if } x<1.0,\\ 1.0 & \text{if } x\geq 1.0,\end{cases}\quad B_0^3=0,\quad p_0=\begin{cases}1.0 & \text{if } x<1.0,\\ 0.1 & \text{if } x\geq 1.0.\end{cases}$$

The computational domain is $(x,t)\in[0,1.5]\times[0,0.35]$, and we use Neumann type artificial boundary conditions for $x=0$ and $x=1.5$. The numerical results for 200 mesh points at time $t=0.35$ are shown in Fig. 1 and are compared in Table 1. We calculate the reference solution using the HLL3L solver and 8000 mesh points. As seen from the error table as well as from Figure 1, the different solvers do quite well at resolving the complicated solution which has a large number of waves. The error table shows that the Roe-scheme has the best resolution, and the other schemes give very similar results. As expected,

Table 1: Relative errors in L^1 for the density and B^2 for the Brio-Wu shock tube using M mesh points.

	M	HLL	HLL3L	HLL3G	HLL3R	Roe
Relative errors in ρ.	100	4.42	3.35	3.71	3.49	2.64
	200	3.21	2.31	2.61	2.33	1.60
	400	1.92	1.25	1.50	1.30	0.72
	800	1.19	0.73	0.91	0.77	0.37
	1600	0.64	0.31	0.43	0.34	0.10

	M	HLL	HLL3L	HLL3G	HLL3R	Roe
Relative errors in B^2.	100	3.64	2.91	3.45	3.38	2.36
	200	2.68	2.16	2.59	2.34	1.43
	400	1.43	1.07	1.39	1.21	0.50
	800	0.88	0.67	0.88	0.76	0.29
	1600	0.43	0.28	0.43	0.34	0.11

all the schemes exhibit a convergence order roughly between $\frac{1}{2}$ and 1 as the number of mesh points increases. Among the HLL solvers, the HLL2 solver seems to be the most dissipative and least accurate.

An expansion problem

From the previous experiment, the Roe solver was found to be most accurate. However, it is well known that the Roe solver fails to be positivity preserving. We present a test case illustrating this phenomenon. Consider the initial data,

$$\rho_0 = p_0 \equiv 1, \quad u_0^2 = u_0^3 = B_0^1 = B_0^3 \equiv 0,$$
$$u_0^1 = \begin{cases} -4.0 & \text{if } x < 0.7, \\ 4.0 & \text{if } x \geq 0.7, \end{cases} \quad B_0^2 = \begin{cases} 1.0 & \text{if } x < 0.7, \\ -1.0 & \text{if } x \geq 0.7, \end{cases}$$

and the computational domain $[0,0.14] \times [0,0.12]$ and test all the solvers on this test case. The data is set up in a manner such that the exact solution loses mass in the center of the domain and the resulting pressure and density are quite close to zero. The numerical results for the pressure at 400 mesh points near the final time is shown in Fig. 2. From Fig. 2, the HLL solvers do well in this case in resolving very low pressures near the center of the domain. There are minor differences in the resolution between different HLL solvers. On the other hand, the Roe solver crashed on this test problem at time $t = 0.004$ i.e about 4 percent of the final time, showing that its high accuracy and resolution comes at a price i.e low stability particularly with respect to negative pressures. This example serves as a caution in using the Roe solver on problems involving stratified atmosphere, as low pressures are expected at the top boundary.

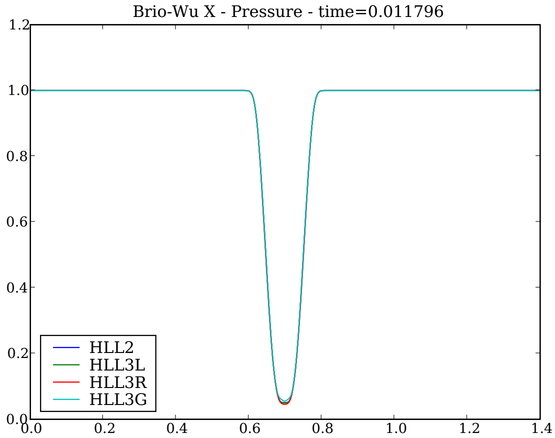


Figure 2: Results for the pressure in the expansion problem for 400 mesh points at time $t=0.12$ with all the HLL solvers.

The Orszag-Tang vortex

This is a commonly used two dimensional test problem, see, e.g., [25], with initial data given by

$$\left\{\rho_0,u_0^1,u_0^2,u_0^3,B_0^1,B_0^2,B_0^3,p_0\right\}=\left\{\gamma^2,-\sin(y),\sin(x),0,-\sin(y),\sin(2x),\gamma\right\}.$$

The computational domain is $(x,y,t)\in[0,2\pi]^2\times[0,2.85]$ with periodic boundary conditions in space. The initial data are smooth but shocks form even for small times. The solution also contains a vortex structure in the center of the domain, and it is a computational challenge to resolve the shock-vortex interaction. Note that $\operatorname{div}\mathbf{B}_0=0$. In Fig. 3, we show the pressure at time $t=2.85$ and compare the HLL3L, HLL3R, HLL3G and Roe solvers on a uniform 200×200 mesh. From the figure, it is clear that the Roe scheme is the least dissipative and that the HLL3G scheme is the most dissipative. The HLL3R has better resolution than the HLL3L solver for this particular problem, although the differences are minor.

For this problem a reference solution is not available, and it is common to use the pressure at the center of the domain as a measure of the accuracy of approximate solutions. In Table 2 we show the maximum pressure for the various solvers on a sequence of meshes. It is generally assumed that the maximum pressure correlates with the quality of the approximation. From Table 2, it is clear that the Roe solver is the most accurate. Among the HLL solvers, the HLL3R solver seems to have the best resolution. However, when we refine the mesh up to 800 mesh points in either direction, all the solvers except the HLL3R solver crash on account of negative pressures with the symbol $-$ denoting a crash of the solver. Even the HLL3R solver crashed on a 1600×1600 mesh.

One possible reason might be the discretization of the divergence constraint [24]. Since none of our solvers are divergence preserving, and although the divergence of $\mathbf{B}_0$

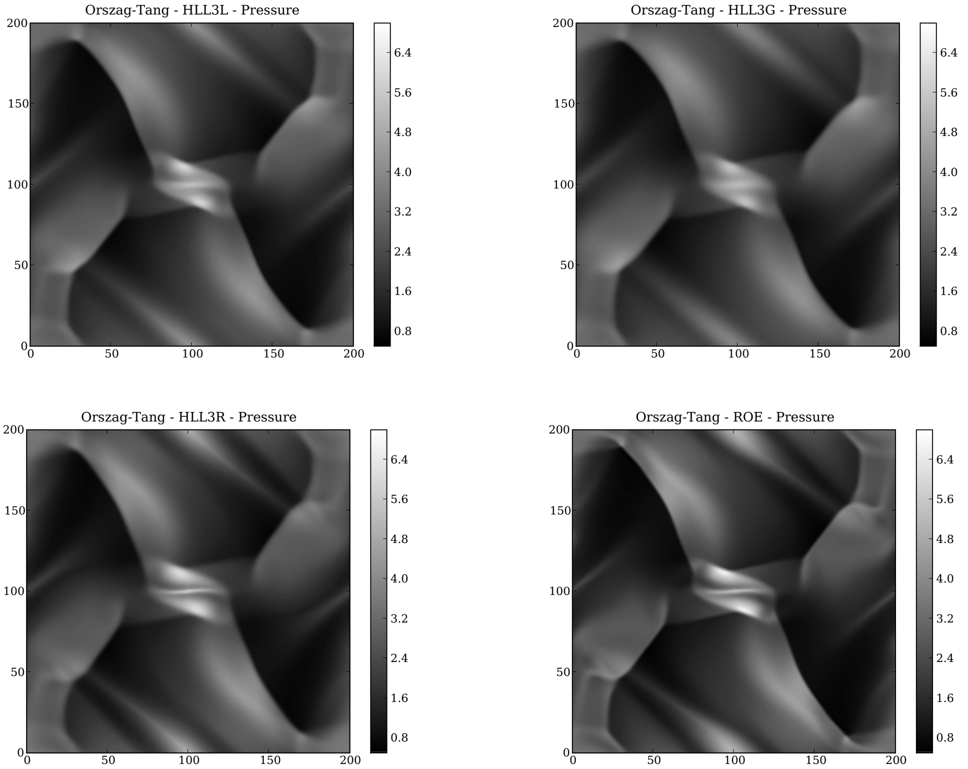


Figure 3: The pressure in the Orszag-Tang vortex at time $t=2.85$ on a 200×200 mesh. Top left: HLL3L, top right: HLL3G, bottom left: HLL3R and bottom right: Roe.

Table 2: Maximum pressure for the Orszag-Tang vertex on an $M\times M$ grid for $t=2.85$. A "-" indicates that the computation was stopped due to negative pressure or density.

M	HLL	HLL3L	HLL3G	HLL3R	Roe
100	4.00	4.41	4.01	4.90	5.27
200	4.74	4.94	4.73	5.38	5.39
400	5.11	5.21	5.09	5.59	5.88
800	-	-	-	5.97	-

is zero, for $t>0$ discretizations of the divergence may not necessarily remain small. It has been speculated that these divergence errors are the source of the instabilities and negative pressures and densities experienced with many solvers. Therefore we exhibit the L^1 norm of the central difference approximation to the divergence in Table 3. The divergence errors are quite large. Nevertheless, they do not seem to influence the quality of the solution (as measured by the maximum pressure). Furthermore, the HLL3R solver had the largest divergence errors, yet this was the only solver which managed to compute a solution on the 800×800 mesh. From the available data, it also seems that the divergence errors decrease with increasing mesh size. These preliminary observations indicate that

Table 3: The L^1 norm of the central discrete divergence for the Orszag-Tang vertex, calculated on a sequence of meshes.

M	HLL	HLL3L	HLL3G	HLL3R	Roe
100	1.92	2.81	1.86	9.66	7.77
200	1.77	2.93	1.71	8.28	6.94
400	1.47	2.60	1.44	6.95	5.64
800	-	-	-	6.03	-

the relationship between accuracy, positivity and divergence preservation is likely to be a complicated one. Similar features were observed in other two-dimensional numerical experiments like the Rotor problem [25].

Remark 3.2. We would like to mention that divergence cleaning for MHD is a very active research area and many methods to preserve discrete versions of the divergence constraint have been proposed. See [25] for a comparison of different divergence cleaning methods. We do not consider such methods here as it is difficult to design stable boundary closures for these methods and boundary conditions are key ingredient in our numerical recipe.

Another approach would be to add the Godunov-Powell source term (see [26]) that transports divergence errors out of the domain. However, this method requires a very careful "upwind" discretization of the source term. Furthermore, all divergence cleaning procedures suffer from stability problems, particularly for strict test cases [27]. Hence, we restrict ourselves to schemes without any special divergence cleaning in this paper.

3.5 Second-order accurate scheme

The above finite volume (3.2) is first-order accurate in both space and time. Hence, the schemes were quite dissipative at least on coarse meshes in the numerical experiments presented above. Most realistic applications require the design of a scheme with higher-order spatial and temporal accuracy. The semi-discrete form of this scheme is given by

$$\begin{aligned}\frac{d}{dt}U_{i,k} &= \boldsymbol{\mathcal{F}}_{i,k}\\ &= -\frac{1}{\Delta x}\left(F_{i+1/2,k}-F_{i-1/2,k}\right)-\frac{1}{\Delta z}\left(H_{i,k+1/2}-H_{i,k-1/2}\right),\end{aligned} \tag{3.11}$$

where $U_{i,k}(t)$ is the cell-average of the unknown at time t. Note that we have dropped time dependence in (3.11) for notational convenience. We will define the numerical fluxes F,H below.

It is standard [3] to replace the cell averages $U_{i,k}$ by non-oscillatory piecewise linear reconstructions in-order to obtain second-order spatial accuracy. There are a variety of reconstructions including the popular TVD MUSCL limiters [3]. However, we will use second-order ENO reconstruction [13] as this procedure can be easily extended to obtain even higher-order schemes.

3.5.1 ENO Reconstruction

Given the cell averages $U_{i,k}$, define the ENO-gradients in each direction as

$$D^x U_{i,k} = \frac{1}{\Delta x} \begin{cases} U_{i+1,k} - U_{i,k}, & \text{if } \Gamma^x_{i,k} \leq 1, \\ U_{i,k} - U_{i-1,k}, & \text{otherwise,} \end{cases} \tag{3.12a}$$

$$D^z U_{i,j} = \frac{1}{\Delta z} \begin{cases} U_{i,k+1} - U_{i,k}, & \text{if } \Gamma^z_{i,k} \leq 1, \\ U_{i,k} - U_{i,k-1}, & \text{otherwise,} \end{cases} \tag{3.12b}$$

where

$$\Gamma^x_{i,k} = \frac{|\Psi(U_{i+1,k}) - \Psi(U_{i,k})|}{|\Psi(U_{i,k}) - \Psi(U_{i-1,k})|}, \quad \Gamma^z_{i,k} = \frac{|\Psi(U_{i,k+1}) - \Psi(U_{i,k})|}{|\Psi(U_{i,k}) - \Psi(U_{i,k-1})|},$$

and Ψ is a global smoothness indicator. We use $\Psi(U) = \rho + |\mathbf{B}|^2$ as the smoothness indicator as it provides adequate representation of all the discontinuities in the solution of the Riemann problem for the MHD equations. Other global smoothness indicators can also be used. Note that for piecewise linear reconstruction, the ENO procedure reduces to providing a limiter for the slopes in each direction. The reconstructed piecewise linear function is each cell is denoted by,

$$\overline{U}_{i,k}(x,z) = U_{i,k} + D^x U_{i,k}(x - x_i) + D^z U_{i,k}(y - y_k). \tag{3.13}$$

We define the following point values,

$$\begin{aligned} UE_{i,k} &= \overline{U}_{i,k}(x_{i+1/2}, z_k), \quad UW_{i,k} = \overline{U}_{i,k}(x_{i-1/2}, z_k), \\ UN_{i,k} &= \overline{U}_{i,k}(x_i, z_{k+1/2}), \quad US_{i,k} = \overline{U}_{i,k}(x_i, y_{k+1/2}). \end{aligned}$$

We use the above defined values to define the second-order numerical fluxes as

$$F_{i+1/2,k} = F(UE_{i,k}, UW_{i+1,k}), \quad H_{i,k+1/2} = H(UN_{i,k}, US_{i,k+1}),$$

where F,H are given by any of the HLL2, HLL3L, HLL3G, HLL3R and Roe solvers of the previous section. This completes the description of the second-order spatial accurate semi-discrete scheme (3.11).

3.5.2 Second-order time stepping

We augment the second-order spatially accurate scheme (3.11) with the following strong-stability preserving TVD second-order accurate Runge-Kutta time stepping [14],

$$U^*_{i,k} = U^n_{i,k} + \Delta t^n \mathcal{F}^n_{i,k}, \tag{3.14a}$$

$$U^{**}_{i,k} = U^*_{i,k} + \Delta t^n \mathcal{F}^*_{i,k}, \tag{3.14b}$$

$$U^{n+1}_{i,k} = \frac{1}{2}(U^n_{i,k} + U^{**}_{i,k}), \tag{3.14c}$$

where $\mathcal{F}$ is the residual in the semi-discrete scheme (3.11). The time step Δt^n is determined by a standard CFL condition.

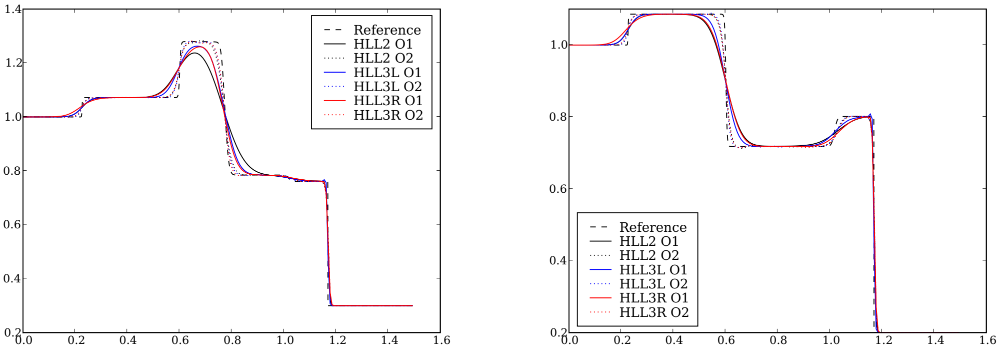


Figure 4: The density and the magnetic field B_2 in the Brio-Wu shock tube at time $t=0.35$ on a 200 mesh comparing first and second-order schemes.

3.6 Numerical experiments with second-order schemes

We present a one-dimensional numerical experiment to demonstrate the gain in accuracy for a second-order scheme. We choose the Brio-Wu shock tube of the previous subsection and show the results obtained with second-order schemes in Fig. 4. We show the density and B_2 computed on a mesh with 200 points. In-order to prevent cluttering in the figure, we present the results with just three schemes: HLL2, HLL3L and HLL3R in both the first-order (spatial and temporal) and second-order (spatial and temporal) versions. The results show that there is a large gain in accuracy and resolution by going to second-order. In particular, notice the sharp resolution of the contact discontinuity (even for the HLL2 solver). Furthermore, the differences between the solvers are far less pronounced in the second-order versions than in the first-order ones. A quantitative confirmation of the above observations is provided in the error Table 4 where we present the relative errors in pressure on a sequence of meshes for both the first- and second-order versions of the HLL-solvers. The results show that the second-order errors schemes have lower errors than the first-order schemes. The different second-order schemes with HLL2, HLL3L and HLL3R solvers are very similar in their numerical performance. Furthermore, the observed rate of convergence for the first-order schemes is around 0.7 and for the second-order schemes is around 1.05. This is expected as the presence of discontinuities in the solution erodes the expected convergence rates. The above test case serves to demonstrate the improvement obtained by using second-order accurate schemes. We will present two-dimensional second-order numerical results in the section on wave propagation.

4 Discretization of the source term

In order to complete the description of a finite volume scheme of the last section, it remains to include the gravity source term in (2.1). As stated before, a key issue that has

Table 4: Relative percentage errors for the pressure in L^1 at time $t=0.35$ and the order of convergence for the Brio-Wu shock tube for various mesh sizes M taking the HLL3L scheme with 6400 grid points as a reference solution.

M	HLL2		HLL2 (2nd)		HLL3L		HLL3L (2nd)	
50	4.2		3.2		3.4		3.0	
100	2.5	0.79	1.4	1.19	2.0	0.76	1.3	1.20
200	1.6	0.64	0.82	0.77	1.3	0.62	0.75	0.8
400	0.90	0.83	0.31	1.40	0.67	0.95	0.28	1.42
800	0.58	0.63	0.17	0.86	0.48	0.79	0.16	0.84

M	HLL3R		HLL3R (2nd)	
50	4.2		3.2	
100	2.5	0.69	1.5	1.1
200	1.6	0.8	0.82	0.87
400	0.92	0.75	0.32	1.35
800	0.57	0.64	0.17	0.91

to be considered is to discretize the source term in an appropriate manner so that the updated pressures and densities remain positive and the interesting steady states (2.5) and (2.6) are preserved to a sufficient degree of accuracy. The source term is included by the method of fractional steps or operator splitting. In the first step, we need to update the fluxes in the finite volume scheme.

In this step, we advance the approximate solution $U^n_{i,k}$ by a finite volume method using the approximate Riemann solvers of the previous section. This results in the following scheme,

$$U^{n+1/2}_{i,k}=\mathcal{F}\left(U^n_{i-1,k},U^n_{i,k-1},U^n_{i,k},U^n_{i+1,k},,U^n_{i,k+1}\right), \tag{4.1}$$

where $\mathcal{F}$ is the update function given in (3.2).

In the next step, we update U by solving the following ordinary differential equation,

$$U_t=S(U), \tag{4.2}$$

where S represents the right hand side of (2.1). This means that the cell average $U^{n+1}_{i,j}=U(\Delta t)$ where U is the solution of the differential equation

$$\frac{d}{dt}U(t)=S(U(t)), \quad U(0)=U^{n+1/2}_{i,k}.$$

We can solve this ODE by any suitable numerical method. However, due to the simple form of S, we are able to calculate the exact solution as

$$U^{n+1}_{i,k}=U^{n+1/2}_{i,k}+\Delta t\left(0,0,0,-g\rho^{n+1/2}_{i,k},0,0,0,-\rho^{n+1/2}_{i,k}u^{3,n+1/2}_{i,k}g+\frac{\rho^{n+1/2}_{i,k}g^2\Delta t^n}{2}\right). \tag{4.3}$$

Remark 4.1. If the update function $\mathcal{F}$ in (4.1) is positivity preserving, it is desirable that the scheme retains this positivity. If we use the exact update, (4.3), a straight-forward calculation shows that $p_{i,k}^{n+1} = p_{i,k}^{n+1/2}$, hence positivity still holds. If we use an implicit Euler method the resulting scheme is positive, while if we use an explicit Euler method, positivity holds for sufficiently small Δt.

If $\mathcal{W}$ denotes the solution operator of (4.2), then the resulting scheme reads

$$\begin{aligned} U_{i,k}^{n+1/2} &= \mathcal{F}\left(U_{i-1,k}^n, U_{i,k-1}^n, U_{i,k}^n, U_{i+1,k}^n,, U_{i,k+1}^n\right), \\ U_{i,k}^{n+1} &= \mathcal{W}\left(U_{i,k}^{n+1/2}\right). \end{aligned} \tag{4.4}$$

Remark 4.2. The above splitting is first-order accurate in time. It is straightforward to extend it to second-order accuracy by using the fairly standard Strang splitting procedure [3]. However, numerical experiments didn't show a big improvement using this procedure and we will show results only with the above splitting, even when second-order accurate spatial and temporal discretizations are employed for evolving the homogeneous part.

5 Non-reflecting characteristic boundary conditions

To complete the description of the numerical scheme (4.4) we now specify boundary values. As is standard for finite volume methods, this will be done by specifying values in ghost cells outside the computational domain.

Since we are aiming at modeling a small part of a stratified atmosphere we use periodic boundary values in the x direction. This amounts to setting $U_{0,k}^n = U_{I,k}^n$ and $U_{I+1,k}^n = U_{1,k}^n$ for $k = 1,\cdots,K$.

At the bottom boundary we specify the values in the ghost cells, i.e., we set $U_{i,0}^n$, $i = 1,\cdots,I$. This is supposed to model an inflow situation.

The top boundary has no physical significance, and its placement is a function of computational resources, most notably available storage and computing speed. The values in the ghost cells at the top of the computational domain are supposed to model the "rest of space". Therefore, waves should not be reflected from this numerical boundary, and we do not wish this boundary to generate any other numerical artifacts. Furthermore, we have to ensure that no mass leaks out via the top boundary. We also want these boundary conditions to result in a stable discretization which keeps density and pressure positive.

There is considerable amount of research on numerical boundary conditions. A very incomplete list includes [28, 29] and references therein. In [15], the author designed boundary conditions based on the characteristic decomposition for the Euler equations of gas dynamics. The "no-reflection" property is ensured by discretizing derivatives suitably. This scheme was extended to multiple space dimensions by a locally one-dimensional projection. Some further modifications were proposed by [17]. This characteristic based approach is very popular in the computational fluid dynamics community. They have been advocated as suitable boundary conditions for MHD equations and we present a version of this approach to the ideal MHD equations in this paper. Furthermore, we provide an alternative formulation of the characteristic boundary conditions of [15,16] in terms of linearized solvers. This alternative formulation is easier to implement with the finite volume procedure used in the interior of the computational domain. They also reveal that the resulting schemes may lead to negative pressures and densities. The resulting boundary conditions are designed to minimize reflections but are not well-balanced; i.e, don't ensure mass balance in the domain and lead to leakage of mass. We introduce a novel modification of the boundary conditions in order to ensure mass balance in the domain. This modification implies that there are some reflections from the boundary as information has to be propagated to the interior in order to preserve mass balance. Therefore this modification must be made in a such a way that the magnitude of reflections remains small. We begin with a description of the characteristic boundary conditions below.

5.1 Characteristic boundary conditions

Consider the partially linearized equations (2.1) in the primitive form,

$$U_t + AU_z = \bar{S}(U), \tag{5.1}$$

where

$$\bar{S}(U) = S(U) - f(U)_x,$$

and A is the Jacobian $dh(U_0)$ evaluated at some constant U_0. We consider this equation for $z = Z$, i.e., at the top boundary. Let R and R^{-1} be an orthonormal eigensystem corresponding to A. Then we can diagonalize (5.1) by introducing the vector of characteristic variables $W = R^{-1}U$. The equations decouple to yield,

$$W_t + \Lambda W_z = R^{-1}\bar{S}, \tag{5.2}$$

where $\Lambda = \mathrm{diag}\{\lambda^1, \lambda^2, \cdots, \lambda^7\}$. We can use the Roe-Balsara eigensystem of Section 2.3 as the orthogonal eigensystem of A. Let $\hat{j}$ be such that $\lambda_j \le 0$ for $j \le \hat{j}$ and $\lambda_j > 0$ for $j > \hat{j}$. If $j \le \hat{j}$ the characteristics of the jth equation of (5.2) are moving in the negative z-direction, i.e., they are *incoming*. Similarly, for $j > \hat{j}$ the characteristics associated with the jth equation in (5.2) are *outgoing*. The basis of the characteristic boundary conditions of [15,16] is to discretize the z-derivative based on the direction of the characteristics. If the characteristic is outgoing, then we use an upwind derivative. This is possible as all the information is taken out of the domain. While if the characteristic is incoming, information is supposed to flow into the domain. However, we don't want any information to go into the domain, and we will set the derivative in the incoming characteristics to zero. This is the basis of the "no incoming wave" philosophy of the characteristic type boundary conditions. When implementing this in practice, we use a single row of ghost cells, located at $k = K+1$. This row is updated using a discretized version of (5.2) or (5.1), where we specify values for the characteristic variables in the ghost cells depending on the direction of the corresponding characteristics.

This update of the ghost cells is consistent with the direction of the flow and implies that there are no "incoming" waves into the domain and reflections are small. The algorithm for the update of the ghost cells reads

Algorithm 5.1:

Step 1 Given $U^n_{i,K}$ and $U^n_{i,K+1}$, choose a suitable Jacobian matrix $A^n_{i,K+1/2}$ determined by these two values. Two examples are the Jacobian evaluated at the average of $U^n_{i,K}$ and $U^n_{i,K+1}$ and the Roe matrix given in [11].

Step 2 Compute the Roe-Balsara eigensystem

$$R^n_{i,K+1/2} \text{ and } \left\{\lambda^{1,n}_{i,K+1/2},\cdots,\lambda^{7,n}_{i,K+1/2}\right\}$$

from $A^n_{i,K+1/2}$. Using this eigensystem, compute the characteristic variables

$$W^n_{i,K}=R^{-1,n}_{i,K+1/2}U^n_{i,K} \text{ and } W^n_{i,K+1}=R^{-1,n}_{i,K+1/2}U^n_{i,K+1}.$$

Step 3 Compute the vector $\mathbf{d}^n_{i,K+1/2}=\left\{d^{1,n}_{i,K+1/2},\cdots,d^{7,n}_{i,K+1/2}\right\}$ by

$$d^{j,n}_{i,K+1/2}=\begin{cases}\lambda^{j,n}_{i,K+1/2}\frac{W^{j,n}_{i,K+1}-W^{j,n}_{i,K}}{\Delta z} & \text{if } \lambda^{j,n}_{i,K+1/2}>0,\\ 0 & \text{otherwise,}\end{cases}$$

where $W^{j,n}_{i,k}$ is the characteristic weight in the (i,K) cell for the j-th characteristic field at the n-th timestep.

Step 4 The update of the ghost cells then reads

$$\begin{aligned}U^{n+1}_{i,K+1}=U^n_{i,K+1}-&\frac{\Delta t^n}{\Delta x}\left(F^n_{i+1/2,K+1}-F^n_{i-1/2,K+1}\right)\\ &-\Delta t^n R^n_{i,K+1/2}\mathbf{d}^n_{i,K+1/2}+\Delta t^n S\left(U^n_{i,K+1}\right).\end{aligned} \tag{5.3}$$

The above algorithm describes the implementation of the characteristic boundary conditions for the ideal MHD equations with the gravitational source term. The key step is step 3 above where the spatial derivative at the boundary is computed. Note that this is based on the "no incoming wave" philosophy of [15,16]. It turns out that this boundary condition can be formulated in an alternative manner. This alternative formulation is easier to use and more revealing about what happens when we use non-reflecting characteristic boundary conditions. We give the alternative formulation below.

Algorithm 5.2:

Step 1 Compute $\Lambda^n_{i,K+1/2}$ and $R^n_{i,K+1/2}$ as in Algorithm 5.1.

Step 2 Set

$$W^{j,n}_{i,K+1/2} = \begin{cases} W^{j,n}_{i,K} & \text{if } \lambda^{j,n}_{i,K+1/2} > 0, \\ W^{j,n}_{i,K+1} & \text{otherwise,} \end{cases}$$

and set $U^n_{i,K+1/2} = R^n_{i,K+1/2} W^n_{i,K+1/2}$.

Step 3 The update of the ghost cells then reads

$$\begin{aligned} U^{n+1}_{i,K+1} = U^n_{i,K+1} - \frac{\Delta t^n}{\Delta x}\left(F^n_{i+1/2,K+1} - F^n_{i-1/2,K+1}\right) \\ - \frac{\Delta t^n}{\Delta z} A^n_{i,K+1/2}\left(U^n_{i,K+1} - U^n_{i,K+1/2}\right) + \Delta t^n S\left(U^n_{i,K+1}\right). \end{aligned} \tag{5.4}$$

In this formulation we see that what we are doing is adding (yet) another ghost cell at $(i,K+2)$, setting the value at this ghost cell to $U^n_{i,K+1}$ and then updating $U^n_{i,K+1}$ via a Roe type solver.

Thus, the "non-reflecting" characteristic boundary conditions in this formulation are based on linearizing the equations at the boundary and using a Neumann-type boundary condition as the normal derivative is taken to be zero by putting the same value on the second ghost cell. In the lemma below we show that Algorithms 5.1 and 5.2 give the same result.

Lemma 5.1. *Let $U^n_{i,K}$ and $U^n_{i,K+1}$ be given and $A^n_{i,K+1/2}$ be the linearization defined above. Let $\mathbf{d}^n_{i,K+1}$ be calculated as in **Step** 3 of Algorithm 5.1 and $U^n_{i,K+1/2}$ be defined as in **Step** 2 of Algorithm 5.2. Then*

$$A^n_{i,K+1/2}(U^n_{i,K+1} - U^n_{i,K+1/2}) = \Delta z R^n_{i,K+1/2} \mathbf{d}^n_{i,K+1/2}, \tag{5.5}$$

and thus the two algorithms give the same result.

Proof. For clarity, we momentarily suppress "i" and "n" in our notation. Set

$$\hat{j} = \underset{j}{\operatorname{argmax}} \left\{ \lambda^j_{K+1/2} \leq 0 \right\},$$

and set $\hat{j} = 0$ if $\lambda^j_{K+1/2} > 0$ for all j. Then

$$W_{K+1} - W_{K+1/2} = \left\{ 0, \cdots, 0, W^{\hat{j}+1}_{K+1} - W^{\hat{j}+1}_{K}, \cdots, W^7_{K+1} - W^7_K \right\}.$$

From **Step 3** of Algorithm 5.1 we get

$$\Delta z \mathbf{d}_{K+1} = \left\{ 0, \cdots, 0, \lambda^{\hat{j}+1}_{K+1/2}\left(W^{\hat{j}+1}_{K+1} - W^{\hat{j}+1}_{K}\right), \cdots, \lambda^7_{K+1/2}\left(W^7_{K+1} - W^7_K\right) \right\}.$$

Therefore,

$$
\begin{aligned}
A_{K+1/2}(U_{K+1}-U_{K+1/2}) &= R_{K+1/2}\Lambda_{K+1/2}R^{-1}_{K+1/2}(U_{K+1}-U_{K+1/2})\\
&= R_{K+1/2}\Lambda_{K+1/2}(W_{K+1}-W_{K+1/2})\\
&= R_{K+1/2}\mathbf{d}_{K+1}.
\end{aligned}
$$

This completes the proof of the lemma. □

Hence, both the above formulations are equivalent. The second formulation based on linearization and zero Neumann boundary conditions is very easy to use with the finite volume framework of this paper. Numerical non-reflecting boundary conditions for finite volume methods often use zero Neumann boundary conditions in ghost cells (see [3]). The above lemma establishes that the characteristic boundary conditions of [15–17] are of the same spirit and one has to replace the finite volume solver at the boundary with a Roe solver to obtain the characteristic boundary conditions. This also paves the way for further analysis of these boundary conditions in the finite volume framework. Furthermore, it exposes a potential problem. As we discovered the Roe solver is not positivity preserving, so using it at the top boundary where the pressures are expected to be very low might lead to instabilities. Hence, characteristic boundary conditions might run into problems near low pressures. We will investigate this issue further in numerical experiments.

5.2 "Balanced" boundary conditions

When simulating stratified atmospheres we expect the density and pressure to be very low near the top boundary of the computational domain. Furthermore, we desire to preserve the steady states (2.5) and (2.6) at least approximately also in our numerical approximations. This means that mass should not "leak" from the top boundary, since such leakage is likely to lead to negative pressure or density. Hence, we have to balance the boundary conditions suitably.

The strategy to "balance" the boundary conditions at the top boundary is based on the following argument. The crucial step in Algorithm 5.1 is **Step 3** where the vector $\mathbf{d}^n_{i,K+1}$ is computed. If the j-th characteristic is outgoing, then using a upwind discretization is consistent with the direction of propagation.

If the j-th characteristic is incoming, we set $d^{n,j}_{i,k+1}=0$. The reasoning behind this is based on the "no incoming wave" philosophy. However, in order to balance the boundary conditions, we need to introduce some information from outside the domain. One reasonable way of doing so is to put conditions on the incoming characteristics. A (semi) discrete discretization of the jth equation of (5.2) reads

$$
\frac{dW^{n,j}_{i,K+1}}{dt}+d^{n,j}_{i,K+1}=\left(R^{n,-1}_{i,K+1/2}\bar{S}\right)^j.
$$

Since, we aim to preserve the steady states (2.5), (2.6), and calculate perturbations of them, it is reasonable to enforce the following steady state condition for the incoming characteristics,

$$\frac{dW^{n,j}_{i,K+1}}{dt}=0,$$

which is obtained by setting,

$$d^{n,j}_{i,K+1}=\left(R^{n,-1}_{i,K+1/2}\bar{S}\right)^j.$$

This modification ensures some form of mass balance across the boundary as the incoming characteristic variables are kept steady, and will only affect Algorithm 5.1 in **Step 3**, whose modified version now reads:

Step 3, balanced Set

$$\bar{S}^n_{i,K+1}=R^{n,-1}_{i,K+1/2}\left(S\left(U^n_{i,K+1}\right)-\frac{F^n_{i+1/2,K+1}-F^n_{i-1/2,K+1}}{\Delta z}\right).$$

Then

$$d^{j,n}_{i,K+1/2}=\begin{cases}\lambda^{j,n}_{i,K+1/2}\frac{W^{j,n}_{i,K+1}-W^{j,n}_{i,K}}{\Delta z} & \text{if } \lambda^{j,n}_{i,K+1/2}>0,\\ \bar{S}^{j,n}_{i,K+1} & \text{otherwise.}\end{cases}$$

Note that we enforce the steady state condition only on the incoming characteristic variables. No conditions are imposed on the outgoing characteristic variables. The non-zero values for the incoming characteristic variables ensure the mass-balance, and (unfortunately) this means that the "no-reflection" condition is violated and there will be some reflections from the top boundary.

Remark 5.1. The second-order version of the characteristic-boundary conditions is much more involved [17] and requires using second-order differences in Step 3 of Algorithm 5.1. We will not consider the second-version in the remaining part of the paper as it led to numerical instabilities.

5.3 Simple boundary conditions

Lemma 5.1 provided a connection between characteristic boundary conditions and zero Neumann boundary conditions. Characteristic boundary conditions are equivalent to zero Neumann boundary conditions if the given finite volume solver is replaced at the boundary by a Roe solver. We can also use the zero Neumann boundary conditions by specifying the ghost cell value to be the same as the value in the last physical cell of the domain. However, this boundary implementation is not balanced. We can balance it by the following simple modification. We compute the ghost cell values as follows,

$$\begin{aligned}\rho^{n+1}_{i,K+1}&=\rho^0_{i,K+1}, & P^{n+1}_{i,K+1}&=P^0_{i,K+1},\\ \mathbf{B}^{n+1}_{i,K+1}&=\mathbf{B}^0_{i,K+1}, & \mathbf{u}^{n+1}_{i,K+1}&=\mathbf{u}^{n+1}_{i,K},\end{aligned}\tag{5.6}$$

where $U^0_{i,K+1}$ are the values of the variables initially in the ghost cell. These conditions imply that ρ, P and $\mathbf{B}$ in the ghost cells are set to their initial values. This step ensures mass and magnetic field balance across the boundary. Furthermore, we put Neumann boundary conditions for the velocity $\mathbf{u}$ which should minimize reflections when waves from the interior reach the top boundary. It is quite straight-forward to extend (5.6) to second-order accurate schemes. For second-order schemes, we need to specify values in two layers of ghost cells in each boundary. Periodic boundary conditions along the x-boundary are trivial to implement in the second-order version. The above simple boundary conditions lead to the following values in the outermost ghost layer at the top boundary,

$$\begin{aligned}&\rho^{n+1}_{i,K+2}=\rho^0_{i,K+2}, \quad P^{n+1}_{i,K+2}=P^0_{i,K+2},\\ &\mathbf{B}^{n+1}_{i,K+2}=\mathbf{B}^0_{i,K+2}, \quad \mathbf{u}^{n+1}_{i,K+2}=\mathbf{u}^{n+1}_{i,K-1}.\end{aligned} \tag{5.7}$$

5.4 Extrapolated Neumann boundary conditions

Numerical experiments will show that the simple Neumann boundary conditions (5.6) presented above lead to large reflections and the characteristic type boundary conditions are potentially unstable (due to the lack of positivity), particularly for magnetic fields. Hence, we design a different set of boundary conditions that are stable, preserve the mass balance and keep reflections at the top boundary low. These boundary conditions are inspired by the specific structure of the exponentially decaying steady state pressure and density profiles in (2.5) and (2.6). The first-order version of these extrapolated boundary conditions are of the form,

$$\begin{aligned}&\rho^{n+1}_{i,K+1}=\rho^{n+1}_{i,K}e^{-\frac{\Delta z}{H}}, \quad P^{n+1}_{i,K+1}=P^{n+1}_{i,K}e^{-\frac{\Delta z}{H}}, \quad \forall i,\\ &\mathbf{u}^{n+1}_{i,K+1}=\mathbf{u}^{n+1}_{i,K}, \quad \mathbf{B}^{n+1}_{i,K+1}=\mathbf{B}^{n+1}_{i,K}.\end{aligned} \tag{5.8}$$

The above boundary conditions extrapolate the pressure and density in the ghost cells based on the exponential decay profile in (2.5),(2.6). The velocity and magnetic fields are simply mirrored in the ghost cells. The differences between the extrapolated boundary conditions in (5.8) and the simple boundary conditions in (5.6) lies in the way the pressure and the density are extrapolated from the interior by using a exponential decay rather than keeping the pressure and density fixed to their initial values. Furthermore, a Neumann condition is used for the magnetic field rather than keeping it fixed to the initial value. The second-order version of these boundary conditions are based on specifying the following values at the outermost ghost layer of the top boundary,

$$\begin{aligned}&\rho^{n+1}_{i,K+2}=\rho^{n+1}_{i,K}e^{-\frac{2\Delta z}{H}}, \quad P^{n+1}_{i,K+2}=P^{n+1}_{i,K}e^{-\frac{2\Delta z}{H}}, \quad \forall i,\\ &\mathbf{u}^{n+1}_{i,K+2}=\mathbf{u}^{n+1}_{i,K-1}, \quad \mathbf{B}^{n+1}_{i,K+2}=\mathbf{B}^{n+1}_{i,K-1}.\end{aligned} \tag{5.9}$$

Note that the above boundary condition is very simple to implement.

6 Wave propagation: Numerical experiments

In this section, we use the complete scheme including approximate Riemann solvers for the numerical fluxes, proper discretization of source terms and boundary conditions to simulate wave propagation in an idealized solar atmosphere. We present four different sets of experiments, two without the magnetic field and two including the magnetic field.

6.1 Hydrodynamics: Steady state

We begin with simulations of the hydrodynamic steady state (2.5). The computational domain is $[0,4]\times[0,1]$. The initial conditions are given by the steady state (2.5) with scale height $H=0.158$, initial pressure $p_0=1.13$ and gravitational constant $g=2.74$. Periodic boundary conditions are used for the x-boundary. We test with the first- and second-order versions of all the schemes proposed in this paper and with all three boundary conditions at the top and bottom boundaries i.e, the characteristic type boundary conditions, the Neumann-type simple boundary conditions (5.6), (5.7) and the extrapolated Neumann boundary conditions (5.8), (5.9). The aim of this experiment is to see if the steady state is preserved to a reasonable degree of accuracy.

Table 5: Absolute errors in p and the rate of convergence for first- and second-order versions of the HLL3G and HLL3R schemes for the hydrodynamic steady state at $t=1.8$ on a $4M\times M$ mesh.

M	HLL3G		HLL3G (2nd)		HLL3R		HLL3R (2nd)	
50	$4.7e+1$		$2.1e-1$		$2.4e+1$		$1.6e-1$	
100	$6.2e-0$	2.9	$4.9e-2$	2.09	$4.0e-0$	2.58	$3.8e-2$	2.07
200	$1.6e-0$	1.9	$1.2e-2$	2.09	$1.2e-0$	1.73	$9.5e-3$	2.0
400	$6.2e-1$	1.36	$3.1e-3$	1.95	$4.9e-1$	1.29	$2.4e-3$	1.98

The exact form of boundary conditions didn't matter in this steady state experiment and different boundary conditions led to similar qualitative results. Hence, we present results only with the extrapolated boundary conditions in Table 5. In this table, we present absolute errors in L^1 for the pressure at the final time with the first- and second-order versions of the HLL3G and HLL3R solvers. The results show that the steady state is not preserved exactly. For the first-order schemes, the errors are quite large (of order one). The $HLL3R$ solver is slightly more accurate than the $HLL3G$ solver. However, the gain in accuracy obtained by using a second-order scheme is considerable. The errors are reduced by two or three orders of magnitude and are quite low when one uses the second-order schemes in this case. Also, the table demonstrates that the second-order schemes result in the expected rates of convergence 2 in this case. The rates of convergence with the first-order schemes show large variability (mostly due to the large errors) and the trend suggests that we get a rate of close to one by refining the mesh further. We are not presenting the corresponding results with the HLL2, HLL3L and Roe solvers as these solvers gave very similar results. The HLL2 solver was slightly less accurate than the HLL3G or HLL3R solvers and the HLL3L and Roe solvers were slightly more accurate. The differences were much less pronounced when using the second-order versions of these solvers. Similarly, the mass balance in the boundary conditions was absolutely essential. Using any of the boundary conditions without the balance led to errors, several orders of magnitude more than the errors with the balanced versions of the boundary conditions.

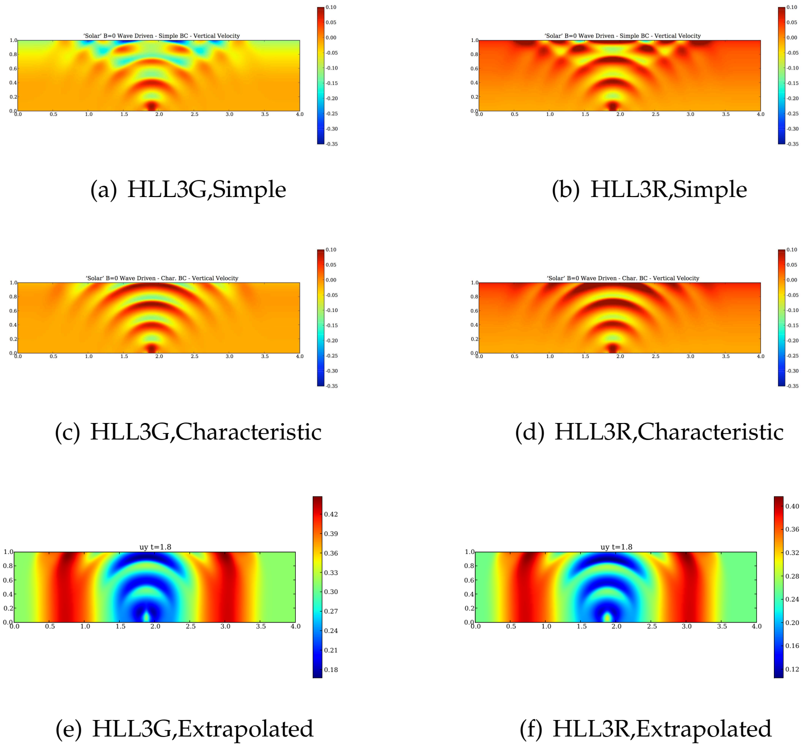
6.2 Hydrodynamics: Wave propagation

The next step is to numerically simulate a hydrodynamic wave propagation model. The computational domain is the same as in the previous experiment and we use a balanced form of the boundary conditions at the top boundary. The propagation of waves is initiated by sending in a sinusoidal (in time) sequence of waves from the bottom boundary and letting them propagate across the domain and (hopefully) exit at the top. The hydrodynamic steady state (2.5) serves as the background for the propagation of waves. At the bottom boundary, the pressure, density and tangential velocities are specified by a simple boundary condition analogous to (5.6). The waves are modeled by the following boundary conditions for the normal velocity at the bottom,

$$u_{i,-1}^{3,n+1} = c\sin\left(6\pi t^{n+1}\right)\chi_{[1.85,1.95]}. \tag{6.1}$$

Hence, we model the bottom boundary as a localized piston in the interval [1.85,1.95] sending in sinusoidal waves. These waves move up the domain and are modified by the flow equations. In Fig. 5 we show u_3 at $t=1.8$ computed using different solvers and boundary conditions. The results presented in Fig. 5 show that there is very little difference between the HLL3G and HLL3R solvers when the boundary condition is fixed. In fact all the five solvers showed very similar results for a given boundary condition and we choose to present the results with the HLL3G and HLL3R solvers. However, the differences in boundary conditions at the top boundary are much more pronounced. The simple boundary conditions (5.6) reflect the waves considerably and are quite unsuitable for simulating wave propagation. The reflection is reduced quite a bit by employing either the characteristic boundary conditions or the extrapolated boundary conditions (5.8). The results with the characteristic and extrapolated boundary conditions are a bit different, on account of differences in the bottom boundary conditions. It is difficult to decide which one is better in the current example although the extrapolated boundary conditions seem to be slightly better at reducing reflections.

The effect of using second-order schemes is shown in Fig. 6 in which we present u_3 at time $t=1.8$ with the first- and second-versions of the HLL3G and HLL3R solvers and the extrapolated Neumann type boundary conditions (5.8), (5.9). This figure shows that the second-order results are much less dissipative and the wave fronts are resolved quite sharply at this resolution. The reflections are also quite low indicating the robustness of the extrapolated boundary conditions. Summarizing, the hydrodynamic simulations

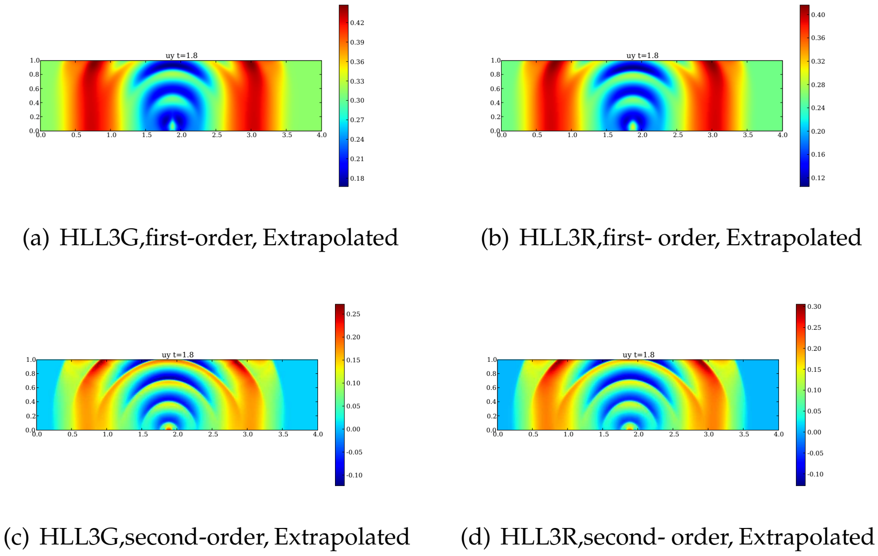


(a) HLL3G,Simple (b) HLL3R,Simple

(c) HLL3G,Characteristic (d) HLL3R,Characteristic

(e) HLL3G,Extrapolated (f) HLL3R,Extrapolated

Figure 5: $u^3(x,z,1.8)$ for the hydrodynamic wave propagation with the HLL3G and HLL3R schemes with different boundary conditions at the top boundary on a 400×100 mesh.



(a) HLL3G,first-order, Extrapolated (b) HLL3R,first- order, Extrapolated

(c) HLL3G,second-order, Extrapolated (d) HLL3R,second- order, Extrapolated

Figure 6: Results for u_3 at time $t=1.8$ for the hydrodynamic wave propagation on a 400×100 mesh with the first- and second-order versions of the HLL3G and HLL3R solvers with the extrapolated Neumann type boundary conditions.

suggest that the HLL solvers are quite robust and the steady state is preserved reasonably well, at least with the second-order accurate schemes. However, the boundary conditions at the top boundary have to chosen carefully. The simple Neumann type boundary conditions (5.6) preserve mass balance but lead to high reflections. The reflections are reduced somewhat for the characteristic-type boundary conditions but these are complicated to implement and expensive (at least for the second-order versions). The extrapolated boundary conditions (5.8), (5.9) are very simple to implement and have at least as low reflections as the characteristic boundary conditions. Furthermore, they allow robust and accurate simulations of waves when combined with second-order spatially and temporally accurate schemes.

6.3 Magneto-hydrodynamics: Steady states

A realistic model of the solar atmosphere must account for the magnetic field. To test the boundary conditions of the last section on this more complicated problem, we first examined how the various methods manage to preserve the magnetohydrodynamic steady state (2.6). The hydrodynamic variables and the computational domain were the same as in the previous section, while the magnetic field was given in terms of its Fourier components, see (2.6). These are listed in (6.2). (The X in (2.6) is in this case equal to 4.)

$$\begin{aligned}\{f_0,f_1,\cdots,f_{14}\}=10^{-1}\{&0.21,-0.10,0.11,-0.11,0.10,-0.08,0.07,-0.05,\\&0.03,-0.02,0.01,-0.008,0.005,-0.002,0.001\}.\end{aligned} \tag{6.2}$$

The above magnetic field is both divergence and curl free and is complicated on account of the Fourier modes. However, the coefficients of the larger modes are very small and the initial modes dominate. We choose such a magnetic field to test whether the schemes can handle a realistic configuration. Our aim is to test the solvers (in both first- and second-order versions) and the boundary conditions. The results were very different from the hydrodynamic case, particularly with regards to stability.

First, the characteristic type boundary conditions were quite unstable and led to crashes due to oscillations and negative pressures. This is not entirely unexpected as the alternative characterization of the characteristic boundary conditions in Lemma 5.1 shows that these boundary conditions use the Roe solver at the boundary. The Roe solver is known to have problems with low pressures and densities, as is the case at the top boundary. Hence, the characteristic boundary condition (used with any of the solvers) is quite unstable with magnetic fields. The simple boundary conditions (5.6) are more stable but lead to large errors. The extrapolated Neumann boundary conditions (5.8) were found to the most stable among the three alternatives as well as the most accurate.

Furthermore, the HLL3L and Roe solvers fail to be stable with this magnetic field (even on the coarsest mesh) resulting in crashes due to negative pressures. This was independent of the boundary condition used. The HLL2 solver is stable but inaccurate. The best results were obtained with the HLL3G and HLL3R solvers, together with the

Table 6: Absolute errors in p and the rate of convergence for first- and second-order versions of the HLL3G and HLL3R schemes for the magneto-hydrodynamic steady state at $t=1.8$ on a sequence of meshes with the extrapolated Neumann boundary conditions.

Mesh	HLL3G		HLL3G (2nd)		HLL3R		HLL3R (2nd)	
200×50	$2.3e+1$		$1.5e-1$		$2.5e+1$		$1.6e-1$	
400×100	$4.2e+0$	2.45	$3.6e-2$	2.05	$4.0e+0$	2.64	$3.8e-2$	2.07
800×200	$1.5e+0$	1.48	$9.0e-3$	2.0	$1.3e+0$	1.62	$9.2e-3$	2.04

extrapolated boundary conditions (5.8), (5.9) and we show results obtained with them in Table 6.

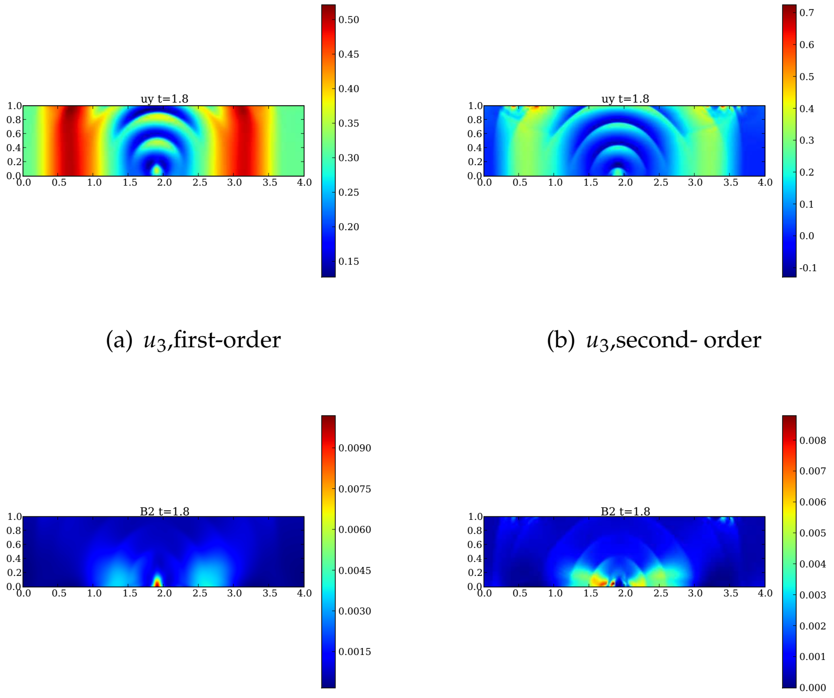
The pressure errors shown in Table 6 establish that the differences between the HLL3G and HLL3R solvers are minor in this case. However, the errors are large and of the order one for the first-order schemes. The convergence rates for first-order schemes are better than expected, probably on account of the large errors on these meshes. The second-order schemes are much more (two to three orders of magnitude) accurate in this case and the expected rates of convergence are obtained. Observe that the errors are very similar to those obtained for the hydrodynamic steady case (Table 5).

6.4 Magneto-hydrodynamics: Wave propagation

We use the same initial condition as the steady state computation (2.6) and introduce waves by sinusoidally perturbing the bottom boundary like in (6.1). The wave propagation results are presented by showing u_3 at time $t=1.8$ with the HLL3G solver (in both first- and second-order versions) and the extrapolated Neumann type boundary conditions (5.8), (5.9) in Fig. 7. The norm of the magnetic field $|\mathbf{B}|^2$ is also shown in the Fig. 7. We present the results only with the HLL3G solver as the second-order version of HLL3R solver crashed on some meshes in this case. From 7, the HLL3G solver and the extrapolated boundary conditions seem to be robust in simulating the waves. Observe that the magnetic field is perturbed on account of the waves. The second-order scheme is more accurate and resolves the wave-fronts sharply.

7 Conclusion

Summing up, we proposed a model for wave propagation in stratified magneto-atmospheres. The model was based on the ideal MHD equations with gravitational source term. The object of interest was to simulate waves by perturbing steady states. Both hydrodynamic and magneto-hydrodynamic steady states were considered and waves introduced by perturbing the bottom boundary. Numerical difficulties included employing appropriate approximate Riemann solvers, suitable discretizations of the gravity source term and design of numerical boundary conditions to maintain stability, mass balance and reduce reflections at the top boundary.



(a) u_3,first-order

(b) u_3,second- order

(c) $|\mathbf{B}|^2$,first-order

(d) $|\mathbf{B}|^2$,second- order

Figure 7: Results for u_3 and $|\mathbf{B}|^2$ at time $t\!=\!1.8$ for the magneto-hydrodynamic wave propagation on a 400×100 mesh with the first- and second-order versions of the HLL3G solver with the extrapolated Neumann type boundary conditions.

We implemented various approximate Riemann solvers of both the HLL-type and Roe-type. Second-order accuracy in space was obtained by using a ENO-type limiter. Temporal accuracy was increased to second-order by employing Runge-Kutta discretizations. No special form of divergence cleaning was used in this paper. The solvers were compared on some benchmark one-dimensional and two-dimensional test cases.

The gravity source term was discretized by using a fractional steps method that preserved positivity. The choice of source discretization didn't influence the results to a great extent. A critical issue was the use of numerical boundary conditions at the top boundary. A characteristic type boundary condition was used (together with modifications to ensure mass balance). This boundary condition was characterized in terms of a Roe solver, thus revealing a potential problem with preservation of positive densities and pressures. A simple Neumann type boundary condition (fixing density and magnetic fields at the boundary to their initial value) was also used. A novel extrapolated Neumann type boundary condition was proposed. This boundary condition extrapolated (by a hydrostatic profile) the values of the density and pressure along with Neumann type conditions for the velocity and magnetic fields.

Computations with the hydrodynamic steady state showed a considerable improvement in accuracy by using second-order schemes. Hydrodynamic wave propagation revealed that the extrapolated boundary condition was as good as the characteristic boundary condition with the added advantages of being cheaper to implement and faster to runs. The differences between the solvers for hydrodynamics were minor.

The magneto-hydrodynamic computations (for steady states as well as waves) were much more difficult on account of numerical instabilities. The HLL3L and Roe solvers crashed in most cases. Similarly, the characteristic boundary conditions were quite unstable. On the other hand, the HLL3G and HLL3R solvers, together with the extrapolated Neumann boundary conditions were quite robust, particularly at second-order. The errors were however larger than in the hydrodynamics cases. Another problem was instabilities when strong magnetic fields were used. All the solvers and boundary conditions led to instabilities in this case. These could be on account of divergence errors or the fact that the schemes didn't preserve a discrete version of the steady state exactly. These questions are going to be addressed in a forthcoming paper.

Acknowledgments

The National Center for Atmospheric Research is sponsored by the National Science Foundation.

References

[1] C. S. Rosenthal et al. Waves in the magnetized solar atmosphere I: Basic processes and internetwork oscillations. Astrophys. J., 564, 2002, 508-524.

[2] T. J. Bogdan et al. Waves in the magnetized solar atmosphere II: Waves from localized sources in magnetic flux concentrations. Astrophys. J., 599, 2003, 626-660.

[3] R. J. LeVeque. Finite Volume Methods for Hyperbolic Problems. Cambridge University Press, Cambridge, 2002.

[4] T. J. Linde. A Three Adaptive Multifluid MHD Model for the Heliosphere. Ph.D thesis, University of Michigan, Ann-Arbor, 1998.

[5] K. F. Gurski. An HLLC-type approximate Riemann solver for ideal Magneto-hydro dynamics. SIAM J. Sci. Comput., 25(6), 2004, 2165-2187.

[6] T. Miyoshi and K. Kusano. A multi-state HLL approximate Riemann solver for ideal magneto hydro dynamics. J. Comput. Phys. 208 (1), 2005, 315-344.

[7] F. Bouchut, C. Klingenberg and K. Waagan. A multi-wave HLL approximate Riemann solver for ideal MHD based on relaxation I – theoretical framework. Numer. Math. 108 (1), 7-42

[8] F. Bouchut, C. Klingenberg and K. Waagan. A multi-wave HLL approximate Riemann solver for ideal MHD based on relaxation II – Numerical implementation. To appear.

[9] F. Bouchut. Entropy satisfying flux vector splittings and kinetic BGK models. Numer. Math. 94 (4), (2003) 623-672

[10] M. Brio and C. C. Wu. An upwind differencing scheme for the equations of ideal MHD. J. Comput. Phys., 75 (2), 1988, 400-422.

[11] P. Cargo and G. Gallice. Roe matrices for ideal MHD and systematic construction of Roe matrices for systems of conservation laws. J. Comput. Phys, 136 (2), 1997, 446-466.

[12] D. S. Balsara. Linearized formulations of the Riemann problem for adiabatic and isothermal hydrodynamics. Astrophys. J. Suppl., 116, 119, 1998.

[13] A. Harten, B. Engquist, S. Osher and S. R. Chakravarty. Uniformly high order accurate essentially non-oscillatory schemes. J. Comput. Phys., 1987, 231-303.

[14] S. Gottlieb, C. W. Shu and E. Tadmor. High order time discretizations with strong stability property. SIAM. Review, 43, 2001, 89-112.

[15] K. W. Thompson. Time dependent boundary conditions for hyperbolic systems. J. Comput. Phys., 68, 1987, 1-24.

[16] K. W. Thompson. Time dependent boundary conditions for hyperbolic systems – II. J. Comput. Phys., 89, 1990, 439-461.

[17] T. J. Poinsot and S. K. Lele. Boundary conditions for direct simulations of compressible viscous flows. J. Comput. Phys. 101, 1992, 104-129.

[18] P. L. Roe and D. S. Balsara. Notes on the eigensystem of magnetohydrodynamics. SIAM. J. Appl. Math., 56 (1), 1996, 57-67.

[19] B. Einfeldt. On the Godunov type methods for gas dynamics. SIAM. J. Numer. Anal., 25 (2), 1988, 294-318.

[20] P. L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. J. Comput. Phys. 43, 1981, 357-372.

[21] A. Harten. High resolution schemes for hyperbolic conservation laws. J. Comput. Phys. 49, 1983, 357-393.

[22] A. Harten and J. M. Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. J. Comput. Phys., 50, 1983, 235-269.

[23] D.S. Balsara and D. Spicer. A Staggered Mesh Algorithm Using High Order Godunov Fluxes to Ensure Solenoidal Magnetic Fields in Magnetohydrodynamic Simulations. J. Comput. Phys, 149 (2), 1999, 270-292.

[24] D. S. Balsara. Total Variation diminshing algorithms for adiabatic and isothermal hydrodynamics. Astrophys. J., suppl., 116, 133, 1998.

[25] G. Toth. The div$\mathbf{B}=0$ constraint in shock capturing magnetohydrodynamics codes. J. Comput. Phys.,161:605-652, 2000.

[26] K. G. Powell. An approximate Riemann solver for magneto-hydro dynamics (that works in more than one space dimension). Technical report, 94 -24, ICASE, Langley, VA, 1994.

[27] D.S. Balsara and J. W. Kim. An intercomparison between divergence cleaning and staggered mesh formulations for numerical magneto-hydrodynamics. Astrophys. J., 602, 1079-1090, 2004.

[28] B. Engquist and A. Majda. Absorbing boundary conditions for the numerical simulation of waves. Math. Comp. 31, 1977, 629-651.

[29] S. Karni. Far field filtering operators for suppresion of reflections from artificial boundaries. SIAM J. Numer. Anal. 33 (3), 1996, 1014-1047.

[30] E. Audusse, F. Bouchut, M. O. Bristeau, R. Klien and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. SIAM. J. Sci. Comput., 25 (6), 2004, 2050-2065.