Non-oscillatory central schemes for one- and two-dimensional MHD equations: I ☆

Jorge Balbás [a], Eitan Tadmor [b,*], Cheng-Chin Wu [c]

[a] *Department of Mathematics, University of California, Los Angeles, CA 90095, USA*
[b] *Department of Mathematics, Center of Scientific Computation and Mathematical Modeling (CSCAMM) and Institute for Physical Science and Technology (IPST), University of Maryland, MD 20742-4015, USA*
[c] *Department of Physics and Astronomy, University of California, Los Angeles, CA 90095, USA*

Abstract

The computations reported in this paper demonstrate the remarkable versatility of central schemes as black-box, Jacobian-free solvers for ideal magnetohydrodynamics (MHD) equations. Here we utilize a family of high-resolution, non-oscillatory central schemes for the approximate solution of the one- and two-dimensional MHD equations. We present simulations based on staggered grids of several MHD prototype problems. Solution of one-dimensional shock-tube problems is carried out using second- and third-order central schemes [Numer. Math. 79 (1998) 397; J. Comput. Phys. 87 (2) (1990) 408], and we use the second-order central scheme [SIAM J. Sci Comput. 19 (6) (1998) 1892] which is adapted for the solution of the two-dimensional Kelvin–Helmholtz and Orszag–Tang problems. A qualitative comparison reveals an excellent agreement with previous results based on upwind schemes. Central schemes, however, require little knowledge about the eigenstructure of the problem – in fact, we even avoid an explicit evaluation of the corresponding Jacobians, while at the same time they eliminate the need for dimensional splitting.
© 2004 Elsevier Inc. All rights reserved.

AMS: 76M20; 76W05; 65M06; 65M12
Keywords: Multidimensional conservation laws; Ideal magnetohydrodynamics (MHD) equations; High-resolution central schemes; Non-oscillatory reconstructions; Jacobian-free form

1. Introduction

In this paper we present second- and third-order non-oscillatory central schemes for the approximate solution of the equations of ideal magnetohydrodynamics (MHD)

$$\rho_t = -\nabla \cdot (\rho \mathbf{v}), \tag{1.1}$$

$$(\rho \mathbf{v})_t = -\nabla \cdot \left[\rho \mathbf{v} \mathbf{v}^T + \left(p + \frac{1}{2} B^2 \right) I_{3\times 3} - \mathbf{B}\mathbf{B}^T \right], \tag{1.2}$$

$$\mathbf{B}_t = \nabla \times (\mathbf{v} \times \mathbf{B}), \tag{1.3}$$

$$e_t = -\nabla \cdot \left[\left(\frac{\gamma}{\gamma - 1} p + \frac{1}{2} \rho v^2 \right) \mathbf{v} - (\mathbf{v} \times \mathbf{B}) \times \mathbf{B} \right]. \tag{1.4}$$

Here, ρ and e are scalar quantities representing, respectively, the mass density and the total internal energy, $\mathbf{v} = (v_x, v_y, v_z)^T$ is the velocity field with L^2-norm $v^2 := |\mathbf{v}|^2$, and $\mathbf{B} = (B_x, B_y, B_z)^T$ and $B^2 := |\mathbf{B}|^2$ represent the magnetic field and its L^2-norm. Finally, the pressure p is coupled to the internal energy, $e = (1/2)\rho v^2 + (1/2)B^2 + p/(\gamma - 1)$, where γ is the (fixed) ratio of specific heats. The system is augmented by the solenoidal constraint $\nabla \cdot \mathbf{B} = 0$; that is, if the condition $\nabla \cdot \mathbf{B} = 0$ is satisfied initially at $t = 0$, then by (1.3) it remains invariant in time.

The intrinsic complexity of these MHD equations suggests the class of central schemes as an efficient alternative for the class of upwind schemes, for computing approximate solutions the MHD model problems (1.1)–(1.4). The central schemes we use in this paper are based on the evolution of cell averages over *staggered* grids. The staggered versions of central schemes introduced in [12,19,21] are presented in Sections 2 and 3. Central schemes eliminate the need for a detailed knowledge of the eigenstructure of the Jacobian matrices. Instead of (approximate) Riemann solvers as building blocks for upwind schemes, simple quadrature formulae are used for the time evolution of central schemes. This approach not only saves the costly characteristic decomposition of the Jacobians, but in fact, it allows us to completely avoid the costly evaluations of 7×7 and 8×8 Jacobian matrices in one- and two-space dimensions. Moreover, central schemes eliminate the need for dimensional splitting which is particularly relevant for the multidimensional MHD system. Indeed, it is known that dimensional splitting may face difficulties with the propagation of other than genuinely nonlinear waves, e.g., in the weakly hyperbolic cases reported in [9,12,14]; we recall that the MHD equations consist of such waves according to a main observation of [3]. The resulting central schemes are black-box, Jacobian-free MHD solvers whose sole input is the computed MHD fluxes. The fact that despite their simplicity these central solvers are able to resolve accurately the complexity of one- and in particular two-dimensional MHD waves is the main issue of this paper. We demonstrate this point with a series of numerical simulations.

In this paper we focus on five prototype MHD problems. We use second- and third-order non-oscillatory central schemes to compute the approximate solution of the one-dimensional Brio–Wu shock tube models [3]. Two different MHD shock tube problems are studied in Sections 4.1 and 4.2. The second-order Jiang–Tadmor central scheme [12] is implemented for the approximate solution of three MHD models in two dimensions. In Section 5.1 we consider the Kelvin–Helmholtz transverse instability problem in both periodic and convective formulations studied earlier by e.g. [13,25], and in Section 5.2 we study the Orszag–Tang MHD vortex system [22,23]. Unlike the Brio–Wu and Kelvin–Helmholtz problems, the numerical solution of the Orszag–Tang vortex system does not necessarily preserve the divergence constraint, $\nabla \cdot \mathbf{B} = 0$. To enforce the latter, a Leray projection corrector is implemented at the end of each time-step, replacing the computed magnetic field with its divergence-free projection.

Our results are found to be in excellent agreement with previous simulations of the same problems carried out with upwind-type schemes [3,6,13,25], and complement the results of Wu and Chang [27], and the more recent results for relativistic MHD flows obtained with central-upwind schemes by Del Zanna et al. [7,8]. These results demonstrate the ability of central schemes to detect and resolve the discontinuous solutions that characterize these models, while retaining efficiency and simplicity. Indeed, the one- and two-dimensional results reported in this paper demonstrate the remarkable versatility of central schemes as black-box, Jacobian-free solvers for MHD computations. We conclude this paper with a two-page Appendix A which provides the complete 2D central code for the convective Kelvin–Helmholtz problem.

2. One-dimensional central schemes

We approximate the solution of (1.1)–(1.4) using predictor–corrector central schemes which are implemented over staggered grids. The schemes have two main ingredients: #1. A non-oscillatory piecewise polynomial reconstruction of pointvalues from their cell averages; followed by #2. Realizing the evolution of these reconstructed polynomials in terms of their staggered cell averages. Following is a description of the one-dimensional schemes that we use for the calculations reported in Section 4.

2.1. Non-oscillatory reconstructions

The system of Eqs. (1.1)–(1.4) admits the general conservation form

$$u_t + f(u)_x = 0 \tag{2.1}$$

(see Section 4 for the explicit form of u and $f(u)$ in the case of one-dimensional MHD equations). Starting with this general form, we follow Godunov's seminal idea for the approximation of discontinuous solutions of conservation laws (see Fig. 1). Using the formulation in [19] we consider the sliding averages, $\bar{u}(x,t) := (1/\Delta x) \int_{x-\Delta x/2}^{x+\Delta x/2} u(\xi,t)\,\mathrm{d}\xi$; the conservation law (2.1) then reads
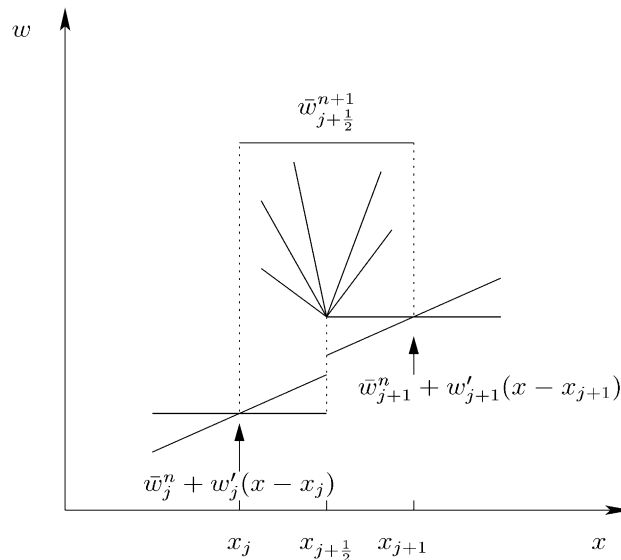

Fig. 1. Central differencing by Godunov-type scheme.

$$\bar{u}_t(x,t) + \frac{1}{\Delta x}\left[f(u(x+\frac{\Delta x}{2},t)) - f(u(x-\frac{\Delta x}{2},t))\right] = 0. \tag{2.2}$$

Introducing a small time step Δt, and integrating over the slab $t < \tau < t + \Delta t$ we arrive at

$$\bar{u}(x,t+\Delta t) = \bar{u}(x,t) - \frac{1}{\Delta x}\int_t^{t+\Delta t}\left[f\left(u\left(x+\frac{\Delta x}{2},\tau\right)\right) - f\left(u\left(x-\frac{\Delta x}{2},\tau\right)\right)\right]d\tau. \tag{2.3}$$

So far, (2.3) is exact. The solution to (2.3) is now realized at the discrete time level $t^n = n\Delta t$ by a piecewise polynomial approximation, $w(x,t^n) \sim u(x,t^n)$, which takes the form

$$w(x,t^n) = \sum p_j(x)\chi_j(x), \quad \chi_j(x) := 1_{I_j}. \tag{2.4}$$

Here, $p_j(x)$ are algebraic polynomials supported on the discrete cells $I_j = I_{x_j} = [x_{j+\frac{1}{2}}, x_{j+\frac{1}{2}}]$ with interfacing breakpoints at the half-integers grid points, $x_{j\pm\frac{1}{2}} = (j \pm \frac{1}{2})\Delta x$. Sampling (2.3) at $x = x_{j+\frac{1}{2}}$, we arrive at the new staggered cell averages, $\bar{w}_{j+\frac{1}{2}}^{n+1}$, centered at $I_{j+\frac{1}{2}} = I_{x_{j+\frac{1}{2}}}$,

$$\bar{w}_{j+\frac{1}{2}}^{n+1} = \frac{1}{\Delta x}\int_{I_{j+\frac{1}{2}}} w(x,t^n)\,dx - \frac{1}{\Delta x}\left[\int_{t^n}^{t^{n+1}} f(w(x_{j+1},t))\,dt - \int_{t^n}^{t^{n+1}} f(w(x_j,t))\,dt\right]. \tag{2.5}$$

The evaluation of the expressions on the right of (2.5) proceeds in two steps, which will occupy the rest of this section.

   The first step starts with the known cell averages, $\{\bar{w}_j^n\}_j$, which are used to reconstruct a non-oscillatory piecewise polynomial approximation $w(x,t^n) = \sum_j p_j(x)\chi_j(x)$. We use piecewise linear functions in the second-order case,

$$p_j(x) = \bar{w}_j^n + w_j'\left(\frac{x-x_j}{\Delta x}\right), \tag{2.6}$$

and piecewise quadratic functions in the third-order case,

$$p_j(x) = w_j^n + w_j'\left(\frac{x-x_j}{\Delta x}\right) + \frac{1}{2}w_j''\left(\frac{x-x_j}{\Delta x}\right)^2. \tag{2.7}$$

Here, $w_j^n, w_j'/\Delta x$, and $w_j''/(\Delta x)^2$ are the approximate point values and the first and second derivatives of $w(x,t^n)$ at $x = x_j$, which are *reconstructed* from the given cell averages, $\{\bar{w}_j^n\}_j$. Several approximations for these *numerical derivatives* are available within the accuracy constraints of the schemes. It should be noted that the procedure for reconstruction of point values and couple of numerical derivatives from the given cell averages is at the heart of high-resolution, non-oscillatory central schemes. In particular, such reconstructions should satisfy the following three essential properties:

- $\mathscr{P}_1$ – Conservation of cell averages: $\bar{p}_j(x)_{|x=x_j} = \bar{w}_j^n$
- $\mathscr{P}_2$ – Accuracy: $w(x,t^n) = u(x,t^n) + \mathcal{O}((\Delta x)^r)$ for r-order accurate method, wherever $u(\cdot,t)$ is sufficiently smooth.
- $\mathscr{P}_3$ – Non-oscillatory behavior of $\sum_j p_j(x)\chi_j(x)$ which is characterized in different ways for different reconstructions.

2.1.1. Second-order reconstruction

   For the second-order results presented in Section 4, the numerical derivative in (2.6), w_j', is given by [1]

$$w_j' = \text{MinMod}(\alpha\Delta_+\bar{w}_j, \Delta_0\bar{w}_j, \alpha\Delta_-\bar{w}_j), \quad 1 \leqslant \alpha < 4. \tag{2.8}$$

---

[1] Where $\Delta_\pm$ and Δ_0 stand for the usual differences, $\Delta_\pm w_j = \pm(w_{j\pm 1} - w_j)$, and $\Delta_0 = \frac{1}{2}(\Delta_+ + \Delta_-)$.

Here, MinMod stands for van-Leer's limiter, cf. [17], where MinMod(a,b,c)=sign(a)min$(|a|,|b|,|c|)$ if sign(a) = sign(b) = sign(c), and it vanishes otherwise. It follows that this reconstruction procedure is non-oscillatory in the sense of satisfying a maximum principle, $\sup_x | \sum_j p_j(x)\chi_j(x) | \leqslant \sup_x | \sum_j \bar{w}_j^n \chi_j(x) |$. Moreover, for a restricted set of α-values, this MinMod-based reconstruction is total-variation diminishing (TVD)

$$\| \sum_j p_j^n(x)\chi_j(x)|_{TV} \leqslant \| \sum_j \bar{w}_j^n \chi_j(x)|_{TV},$$

and hence the corresponding central scheme is TVD [21].

2.1.2. Third-order reconstruction

In order to guarantee properties $\mathscr{P}_1$–$\mathscr{P}_3$ in the case of the piecewise quadratic reconstruction (2.7), we first construct a polynomial using the *non-limited* version of the point values and numerical derivatives that we obtain from the cell averages $\{\bar{w}_j^n\}_j$,

$$\tilde{p}_j(x) = \bar{w}_j^n - \frac{1}{24}\Delta_+\Delta_-\bar{w}_j^n + \Delta_0\bar{w}_j^n\left(\frac{x-x_j}{\Delta x}\right) + \frac{1}{2}\Delta_+\Delta_-\bar{w}_j^n\left(\frac{x-x_j}{\Delta x}\right)^2 \tag{2.9}$$

(we shall point out that starting with third-order, the point value w_j^n no longer coincides with the cell average $\bar{w}_j^n$). This polynomial, (2.9), satisfies properties $\mathscr{P}_1$ and $\mathscr{P}_2$, and an also important *shape preserving* property that guarantees that $\tilde{p}_j$ has the same shape as $\sum_{i=j-1}^{i=j-1}\bar{w}_i^n\chi_i$; and therefore no new extrema is created within the cell I_j. However, *interface jumps* appear when sign$(\tilde{p}_j + 1(x_{j+\frac{1}{2}}) - \tilde{p}_j(x_{j+\frac{1}{2}})) \neq$ sign$(\bar{w}_{j+1}^n - \bar{w}_j^n)$, resulting on the onset of spurious oscillations. To prevent such interface jumps, we seek a third-order limiter, θ_j (with $0 < \theta_j \leqslant 1$), along the lines of those specified in [19]. The limiters are calculated in terms of the cell quantities

$$M_j = \max_{x\in I_j}\tilde{p}_j(x), \quad m_j = \min_{x\in I_j}\tilde{p}_j(x), \tag{2.10}$$

and

$$\begin{cases} M_{j\pm\frac{1}{2}} = \max\left\{\frac{1}{2}(\bar{w}_j^n + \bar{w}_{j\pm1}^n), \tilde{p}_{j\pm1}(x_{j\pm\frac{1}{2}})\right\}, \\ m_{j\pm\frac{1}{2}} = \min\left\{\frac{1}{2}(\bar{w}_j^n + \bar{w}_{j\pm1}^n), \tilde{p}_{j\pm1}(x_{j\pm\frac{1}{2}})\right\}. \end{cases} \tag{2.11}$$

The limiter θ_j is then given by

$$\theta_j = \begin{cases} \min\left\{\frac{M_{j+\frac{1}{2}}-\bar{w}_j^n}{M_j-\bar{w}_j^n}, \frac{m_{j-\frac{1}{2}}-\bar{w}_j^n}{m_j-\bar{w}_j^n}, 1\right\} & \text{if } \bar{w}_{j-1}^n < \bar{w}_j^n < \bar{w}_{j+1}^n, \\ \min\left\{\frac{M_{j-\frac{1}{2}}-\bar{w}_j^n}{M_j-\bar{w}_j^n}, \frac{m_{j+\frac{1}{2}}-\bar{w}_j^n}{m_j-\bar{w}_j^n}, 1\right\} & \text{if } \bar{w}_{j-1}^n > \bar{w}_j^n > \bar{w}_{j+1}^n, \\ 0 & \text{otherwise (if } \Delta_+\bar{w}_j^n \cdot \Delta_-\bar{w}_j^n < 0). \end{cases} \tag{2.12}$$

The new piecewise quadratic polynomials are then written as the convex combination

$$p_j(x) = \bar{w}_j^n + \theta_j(\tilde{p}_j(x) - \bar{w}_j^n), \tag{2.13}$$

which still satisfies properties $\mathscr{P}_1$ and $\mathscr{P}_2$ and the shape preserving property mentioned above, and avoids the spurious interface extrema, i.e., sign$(p_{j+1}(x_{j+\frac{1}{2}}) - p_j(x_{j+\frac{1}{2}})) =$ sign$(\bar{w}_{j+1}^n - \bar{w}_j^n)$. This is known as number of extrema diminishing (NED) property, and characterizes, together with the shape preserving property, the non-oscillatory behavior of the scheme, i.e. property $\mathscr{P}_3$ [19,20].

Remarks

1. The values M_j and m_j in (2.10) do not need to be calculated explicitly as they only enter the calculation of the limiter, θ_j, when the sequence $\{\bar{w}_i^n\}_{i=j-1}^{i=j+1}$ is monotone and they coincide, respectively, with the interface values $\tilde{p}_j(x_{j\pm\frac{1}{2}})$ in the increasing case and $\tilde{p}_j(x_{j\pm\frac{1}{2}})$ in the decreasing case.
2. The monotonicity of $\{\bar{w}_i^n\}_{i=j-1}^{i=j+1}$ also guarantees that the differences $M_{j\pm\frac{1}{2}} - M_j$ and $m_{j\mp\frac{1}{2}} - m_j$ are of order $\mathcal{O}((\Delta x)^3)$ and, consequently θ_j is a third-order limiter and $1 - \theta_j = \mathcal{O}((\Delta x)^3)$.
3. In the case of MHD equations, the value of θ_j in the extrema cells (i.e., $\Delta_+\bar{w}_j^n \cdot \Delta_-\bar{w}_j^n < 0$) needs to be set to 0 to produce the so-called *clipping phenomena* that avoids the onset of spurious oscillations at the expense of reducing to the first order version of Lax–Friedrichs [19].
4. The polynomials (2.13) are equivalent to (2.7) with the point values and numerical derivatives given by

$$w_j' = \theta_j \Delta_0 \bar{w}_j^n, \tag{2.14}$$

$$w_j'' = \theta_j \Delta_+ \Delta_- \bar{w}_j^n, \tag{2.15}$$

$$w_j^n = \bar{w}_j^n - \frac{w_j''}{24}. \tag{2.16}$$

Once $w(x,t^n)$ is realized as a piecewise polynomial, $w(x,t^n) = \sum p_j(x) \chi_j(x)$, it is integrated exactly over the interval $I_{j+\frac{1}{2}}$ to compute the staggered cell averages on the right of (2.5)

$$\frac{1}{\Delta x} \int_{I_{j+\frac{1}{2}}} w(x,t^n)\, \mathrm{d}x = \frac{1}{\Delta x} \left[\int_{x_j}^{x_{j+\frac{1}{2}}} p_j(x)\, \mathrm{d}x + \int_{x_{j+\frac{1}{2}}}^{x_{j+1}} p_{j+1}(x)\, \mathrm{d}x \right] = \frac{1}{2}[\bar{w}_j^n + \bar{w}_{j+1}^n] + \frac{1}{8}[w_j' - w_{j+1}']. \tag{2.17}$$

Remark. Higher order non-oscillatory extensions of these reconstruction algorithms are offered by the Essentially non-oscillatory (ENO) reconstructions of Harten et al. e.g. [10], and their implementation with central schemes can be found in e.g. [18].

2.2. Staggered evolution

We turn to the second step in the construction of the central scheme. Here we follow the evolution of the reconstructed point values, $\{w(x_j, \tau \geqslant t^n)\}_j$, along the midcells, $x = x_j$, which are governed by

$$w_t + f(w)_x = 0, \quad \tau \geqslant t^n, \quad w(x,t^n) = p_j(x), \quad x \in I_j. \tag{2.18}$$

If $\{a_k(u)\}_k$ are the eigenvalues of the Jacobian $A(u) := \partial f / \partial u$, then by hyperbolicity, information regarding the interfacing discontinuities at $(x_{j+\frac{1}{2}}, t^n)$ propagates no faster than $\max_k |a_k(u)|$. Hence, the midcell values, $\{w(x_j, \tau \geqslant t^n)\}_j$, remain free of discontinuities as long as the CFL condition, $\Delta t \leqslant \frac{1}{2} \cdot \max_k | a_k(u) |$, is met. Therefore, the flux integrals in the right of (2.5) involve only smooth integrands and can be evaluated with appropriate quadrature rules to any desired degree of accuracy. In particular, the second-order Nessyahu–Tadmor scheme [21] makes use of the midpoint rule

$$\int_{t^n}^{t^{n+1}} f(w(x_j, \tau))\, \mathrm{d}\tau \approx \Delta t f(w_j^{n+\frac{1}{2}}),$$

while the third-order Liu–Tadmor scheme, [19], makes use of Simpson's rule

$$\int_{t^n}^{t^{n+1}} f(w(x_j, \tau))\, \mathrm{d}\tau \approx \frac{\Delta t}{6} \left[f(w_j^n) + 4f(w_j^{n+\frac{1}{2}}) + f(w_j^{n+1}) \right].$$

These quadrature formulae require the computation of the intermediate point values $w_j^{n+\beta}$, $\beta = 0, 1/2, 1$. A natural approach for computing these point values employs Taylor's expansion and the differential equation (2.1), $w_t = -f(w)_x$. The resulting predictor step in the second-order case reads

$$w_j^{n+\frac{1}{2}} = \bar{w}_j^n - \frac{\lambda}{2} f_j', \quad f_j = f(w_j^n). \tag{2.19}$$

Here $\lambda = \Delta t / \Delta x$ is the fixed mesh ratio and f_j' is an approximate numerical derivative – an approximation to the spatial derivative $f_j'/\Delta x \approx f(w_j^n)_x$. Different recipes for numerical derivatives are available within the accuracy constraints of our calculations. We shall mention two of them which retain the overall second-order accuracy, consult [21]. In the first approach we utilize the chain rule, $f(w)_x = A(w)w_x$. With $A(w_j^n)$ standing for the computed Jacobian and w_j' denoting the reconstructed numerical derivative, $w_j'/\Delta x \approx w(x_j, t^n)_x$, we have

$$f_j' = A(w_j^n)w_j', \quad w_j' = \mathrm{MinMod}(\alpha \Delta_+ \bar{w}_j, \Delta_0 \bar{w}_j, \alpha \Delta_- \bar{w}_j). \tag{2.20}$$

Alternatively, we can implement a spatial MinMod limiter directly for numerical differentiation of the grid function $\{f_j = f(w_j^n)\}_j$,

$$f_j' = \mathrm{MinMod}(\alpha \Delta_+ f_j, \Delta_0 f_j, \alpha \Delta_- f_j), \quad f_j = f(w_j^n). \tag{2.21}$$

Remark. The resulting predictor-step (2.19) and (2.21) completely avoids the evaluation of the Jacobian $A(w_j^n)$.

Similar recipes are available for the third-order predictor,

$$w_j^{n+\beta} = w_j^n - \lambda\beta \left\{ f\left(w_j^n - \frac{\lambda\beta}{2} f_j'\right) \right\}', \quad \beta = \frac{1}{2}, 1. \tag{2.22}$$

As before, the evaluation of the expression on the right of (2.22) can utilize the chain rule

$$f\left(w - \frac{\lambda\beta}{2} f(w)_x\right)_x = f\left(w - \frac{\lambda\beta}{2} A(w)w_x\right)_x = A(w)w_x - \frac{\lambda\beta}{2}\left(A^2(w)w_x\right)_x + \mathcal{O}(\lambda)^2.$$

Using the third-order accurate numerical derivative in (2.14), we end up with two possible recipes for computing the third-order accurate predicted mid-values, $\{f(w_j^n - (\lambda\beta/2)f_j')\}'$ in (2.22)

$$\left\{ f(w_j^n - \frac{\lambda\beta}{2} f_j') \right\}' = \theta_j^{\beta-\frac{1}{2}} \Delta_0 \left\{ f(w_j^n - \frac{\lambda\beta}{2} A(w_j^n)w_j') \right\}, \quad \beta = \frac{1}{2}, 1, \ w_j' = \theta_j^{\beta-\frac{1}{2}} \Delta_0 w_j, \tag{2.23}$$

$$\left\{ f(w_j^n - \frac{\lambda\beta}{2} f_j') \right\}' = A(w_j^n)w_j' - \frac{\lambda\beta}{2} \theta_j^{\beta-\frac{1}{2}} \Delta_0 \left\{ A(w_j^n)^2 w_j' \right\}, \quad \beta = \frac{1}{2}, 1, \ w_j' = \theta_j^{\beta-\frac{1}{2}} \Delta_0 w_j. \tag{2.24}$$

A third possibility, a so-called Jacobian-free form (JFF) alternative is provided by

$$\left\{ f(w_j^n - \frac{\lambda\beta}{2} f_j') \right\}' = \theta_j^{\beta-\frac{1}{2}} \Delta_0 \left(f\left(w_j^n - \frac{\lambda\beta}{2} \Delta_0(f(w_j^n))\right) \right), \quad \beta = \frac{1}{2}, 1, \tag{2.25}$$

where the superscript $\beta - \frac{1}{2}$ in the θ_j limiter indicates that it is calculated using the grid function $\{\bar{w}_j^{n+\beta-\frac{1}{2}}\}$. We notice, however, that although the results presented in Section 4 include this intermediate update of the limiter, calculations performed with a single limiter calculated at the beginning of each time step do not differ significantly from those reported below. Again, the computation of the intermediate values $w_j^{n+\beta}$ in (2.25) avoids the explicit evaluation of the Jacobians, $A(w_j^n)$. The high-resolution of these Jacobian-free versions (2.21) and (2.25) is evident in the numerical results reported in Section 4. The intermediate values $w_j^{n+\beta}$ can also be approximated by Runge–Kutta solvers of the ODE $w_\tau = \hat{f}_x|_{x=x_j}$, $w(x_j, 0) = w_j^n$, $\tau > t^n$, where $\hat{f}_x$

stands for the numerical derivative of f. This approach offers yet another avenue for the time evolution of central schemes: in particular, we mention in this context the efficient evolution procedure based on higher order Runge–Kutta natural continuous extensions, e.g. [4,18], which significantly reduces the number of computations per grid point.

Equipped with the predicted point values, we are set to evaluate the integrals on the right-hand side of (2.5). They are approximated by the midpoint rule in the second-order case

$$\int_{t^n}^{t^{n+1}} f(w(x_j,\tau))\,\mathrm{d}\tau \approx \Delta t f(w_j^{n+\frac{1}{2}}) =: \Delta t f_j^{n+\frac{1}{2}},\tag{2.26}$$

and by Simpson's rule in the third-order case

$$\int_{t^n}^{t^{n+1}} f(w(x_j,\tau))\,\mathrm{d}\tau \approx \frac{\Delta t}{6}[f(w_j^n) + 4f(w_j^{n+\frac{1}{2}}) + f(w_j^{n+1})] =: \Delta t f_j^{n+\frac{1}{2}}.\tag{2.27}$$

Denoting these approximate values of the flux integrals by $f_j^{n+\frac{1}{2}}$, the corrector step for both the second- and third-order central schemes amount to the same statement

$$\bar{w}_{j+\frac{1}{2}}^{n+1} = \frac{1}{2}[\bar{w}_j^n + \bar{w}_{j+1}^n] + \frac{1}{8}[w_j' - w_{j+1}'] - \lambda\left[f_{j+1}^{n+\frac{1}{2}} - f_j^{n+\frac{1}{2}}\right].\tag{2.28}$$

3. Two-dimensional central schemes

As in the one-dimensional case, we begin the description of the two-dimensional scheme by writing (1.1)–(1.4) in conservation form

$$u_t + f(u)_x + g(u)_z = 0.\tag{3.1}$$

Introducing space scales Δx and Δz, we consider the sliding averages of (3.1) over two-dimensional cells $[x - \frac{\Delta x}{2}, x + \frac{\Delta x}{2}] \times [z - \frac{\Delta z}{2}, z + \frac{\Delta z}{2}]$

$$\bar{u}_t(x,z,t) + \frac{1}{\Delta x \Delta z}\int_{z-\frac{\Delta z}{2}}^{z+\frac{\Delta z}{2}}\left[f\left(u\left(x+\frac{\Delta x}{2},\eta,t\right)\right) - f\left(u\left(x-\frac{\Delta x}{2},\eta,t\right)\right)\right]\mathrm{d}\eta$$

$$+ \frac{1}{\Delta x \Delta z}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta z}{2}}\left[g\left(u\left(\xi,z+\frac{\Delta z}{2},t\right)\right) - g\left(u\left(\xi,z-\frac{\Delta z}{2},t\right)\right)\right]\mathrm{d}\xi = 0.\tag{3.2}$$

We proceed along the same lines of the one-dimensional case; integration over the slab $t \leqslant \tau \leqslant t + \Delta t$ is performed, and the solution of the resulting equation is approximated at time level $t^n = n\Delta t$ by a piecewise bilinear polynomial $w(x,z,t^n) = \sum p_{jk}(x,z)\chi_{jk}(x,z)$, where $\chi_{jk}(x,z) = 1_{I_{x_j} \times I_{z_k}}$ is the characteristic function of the control volume $I_{x_j} \times I_{z_k}$. Choosing $x = x_{j+\frac{1}{2}}$, and $z = z_{k+\frac{1}{2}}$ as the interfacing boundaries, the two-dimensional version of Eq. (2.5) reads (see Fig. 2)

$$\bar{w}_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1} = \bar{w}_{j+\frac{1}{2},k+\frac{1}{2}}^n - \frac{1}{\Delta x \Delta z}\int_{t^n}^{t^{n+1}}\int_{z_k}^{z_{k+1}}[f(w(x_{j+1},z,t)) - f(w(x_j,z,t))]\,\mathrm{d}z\,\mathrm{d}t$$

$$- \frac{1}{\Delta x \Delta z}\int_{t^n}^{t^{n+1}}\int_{x_j}^{x_{j+1}}[g(w(x,z_{k+1},t)) - g(w(x,z_k,t))]\,\mathrm{d}x\,\mathrm{d}t.\tag{3.3}$$
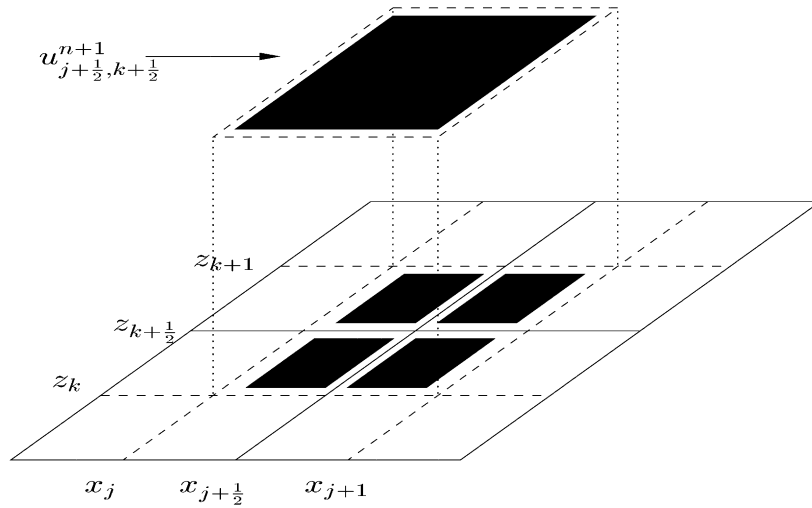
Fig. 2. Two-dimensional floor-plan of central staggered stencil.

The reconstruction of the point values $w(x,z,t^n)$ is carried out using piecewise bilinear functions $p_{jk}(x,z) = \bar{w}_{jk}^n + w'_{jk}\left(\frac{x-x_j}{\Delta x}\right) + \grave{w}_{jk}\left(\frac{z-z_k}{\Delta z}\right)$, with the approximate numerical derivatives in the x and z directions given by

$$w'_{jk} = \text{MinMod}(\alpha\Delta_{+x}\bar{w}_{jk}^n, \Delta_{0x}\bar{w}_{jk}^n, \alpha\Delta_{-x}\bar{w}_{jk}^n), \tag{3.4}$$

$$\grave{w}_{jk} = \text{MinMod}(\alpha\Delta_{+z}\bar{w}_{jk}^n, \Delta_{0z}\bar{w}_{jk}^n, \alpha\Delta_{-z}\bar{w}_{jk}^n). \tag{3.5}$$

These piecewise polynomials admit the following cell averages on the right of (3.3):

$$\begin{aligned}
\bar{w}_{j+\frac{1}{2},k+\frac{1}{2}}^n &= \frac{1}{\Delta x \Delta z}\int_{x_j}^{x_{j+\frac{1}{2}}}\int_{z_k}^{z_{k+\frac{1}{2}}} p_{jk}(x,z)\,\mathrm{d}z\,\mathrm{d}x + \int_{x_{j+\frac{1}{2}}}^{x_{j+1}}\int_{z_k}^{z_{k+\frac{1}{2}}} p_{j+1,k}(x,z)\,\mathrm{d}z\,\mathrm{d}x \\
&\quad + \frac{1}{\Delta x \Delta z}\int_{x_j}^{x_{j+\frac{1}{2}}}\int_{z_{k+\frac{1}{2}}}^{z_{k+1}} p_{j,k+1}(x,z)\,\mathrm{d}z\,\mathrm{d}x + \int_{x_{j+\frac{1}{2}}}^{x_{j+1}}\int_{z_{k+\frac{1}{2}}}^{z_{k+1}} p_{j+1,k+1}(x,z)\,\mathrm{d}z\,\mathrm{d}x \\
&= \frac{1}{4}\left(\bar{w}_{jk}^n + \bar{w}_{j+1,k}^n + \bar{w}_{j,k+1}^n + \bar{w}_{j+1,k+1}^n\right) \\
&\quad + \frac{1}{16}\left[(w'_{jk} - w'_{j+1,k}) + (w'_{j,k+1} - w'_{j+1,k+1}) + (\grave{w}_{jk} - \grave{w}_{j,k+1}) + (\grave{w}_{j+1,k} - \grave{w}_{j+1,k+1})\right]. \tag{3.6}
\end{aligned}$$

The reconstruction procedure described above enjoys the conservation, accuracy and non-oscillatory properties, $\mathscr{P}_1$–$\mathscr{P}_3$, analogous to those listed for one-dimensional schemes in Section 2. In particular, the non-oscillatory property in this case is characterized by the scalar maximum principle, consult [12].

To approximate the flux integrals on the right of (3.3), we first predict the midpoint values

$$w_{jk}^{n+\frac{1}{2}} = \bar{w}_{jk}^n - \frac{\lambda}{2}f'_{jk} - \frac{\mu}{2}\grave{g}_{jk}, \tag{3.7}$$

where $\lambda = \Delta t/\Delta x$ and $\mu = \Delta t/\Delta z$ are the (fixed) mesh ratios in the x and z directions, and f'_{jk} and $\grave{g}_{jk}$ stand for the numerical slopes of $f(w)$ and $g(w)$. In this case, we choose the JFF

$$f'_{jk} = \text{MinMod}(\alpha\Delta_{+x}f_{jk}, \Delta_{0x}f_{jk}, \alpha\Delta_{-x}f_{jk}), \tag{3.8}$$

$$\grave{g}_{jk} = \text{MinMod}(\alpha\Delta_{+z}g_{jk}, \Delta_{0z}g_{jk}, \alpha\Delta_{-z}g_{jk}). \tag{3.9}$$

In the computations below we set the free parameter $\alpha = 1.4$. The flux integrals are then approximated by the midpoint rule in time, and by second-order trapezoidal quadrature in space

$$\int_{t^n}^{t^{n+1}} \int_{z_k}^{z_{k+1}} f(w(x_{j+1}, z, t)) \, \mathrm{d}z \, \mathrm{d}t \sim \frac{\Delta z \Delta t}{2} \left[f(w_{j+1,k}^{n+\frac{1}{2}}) + f(w_{j+1,k+1}^{n+\frac{1}{2}}) \right], \tag{3.10}$$

$$\int_{t^n}^{t^{n+1}} \int_{x_j}^{x_{j+1}} g(w(x, z_{k+1}, t)) \, \mathrm{d}x \, \mathrm{d}t \sim \frac{\Delta x \Delta t}{2} \left[g(w_{j+1,k+1}^{n+\frac{1}{2}}) + g(w_{j,k+1}^{n+\frac{1}{2}}) \right]. \tag{3.11}$$

The new staggered cell averages $\{\bar{w}_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1}\}_{j,k}$ read

$$\begin{aligned}
\bar{w}_{j+\frac{1}{2},k+\frac{1}{2}}^{n+1} &= \frac{1}{4} \left(\bar{w}_{jk}^n + \bar{w}_{j+1,k}^n + \bar{w}_{j,k+1}^n + \bar{w}_{j+1,k+1}^n \right) + \frac{1}{16} \left(w'_{jk} - w'_{j+1,k} \right) - \frac{\lambda}{2} \left[f(w_{j+1,k}^{n+\frac{1}{2}}) - f(w_{jk}^{n+\frac{1}{2}}) \right] \\
&\quad + \frac{1}{16} \left(w'_{j,k+1} - w'_{j+1,k+1} \right) - \frac{\lambda}{2} \left[f(w_{j+1,k+1}^{n+\frac{1}{2}}) - f(w_{j,k+1}^{n+\frac{1}{2}}) \right] + \frac{1}{16} \left(w`_{jk} - w`_{j,k+1} \right) \\
&\quad - \frac{\mu}{2} \left[g(w_{j,k+1}^{n+\frac{1}{2}}) - g(w_{jk}^{n+\frac{1}{2}}) \right] + \frac{1}{16} \left(w`_{j+1,k} - w`_{j+1,k+1} \right) - \frac{\mu}{2} \left[g(w_{j+1,k+1}^{n+\frac{1}{2}}) - g(w_{j+1,k}^{n+\frac{1}{2}}) \right].
\end{aligned} \tag{3.12}$$

In addition to reconstruction and evolution, our second-order two-dimensional scheme requires the correction of the magnetic field, **B**, in order to guarantee the solenoidal constraint $\nabla \cdot \mathbf{B} = 0$. We satisfy this constraint by updating the cell averages of the magnetic field at the end of each time step, replacing the computed **B**, with its divergence-free projection, $\mathbf{B}^c$. To this end the so-called Leray projection is carried out by solving the Poisson equation
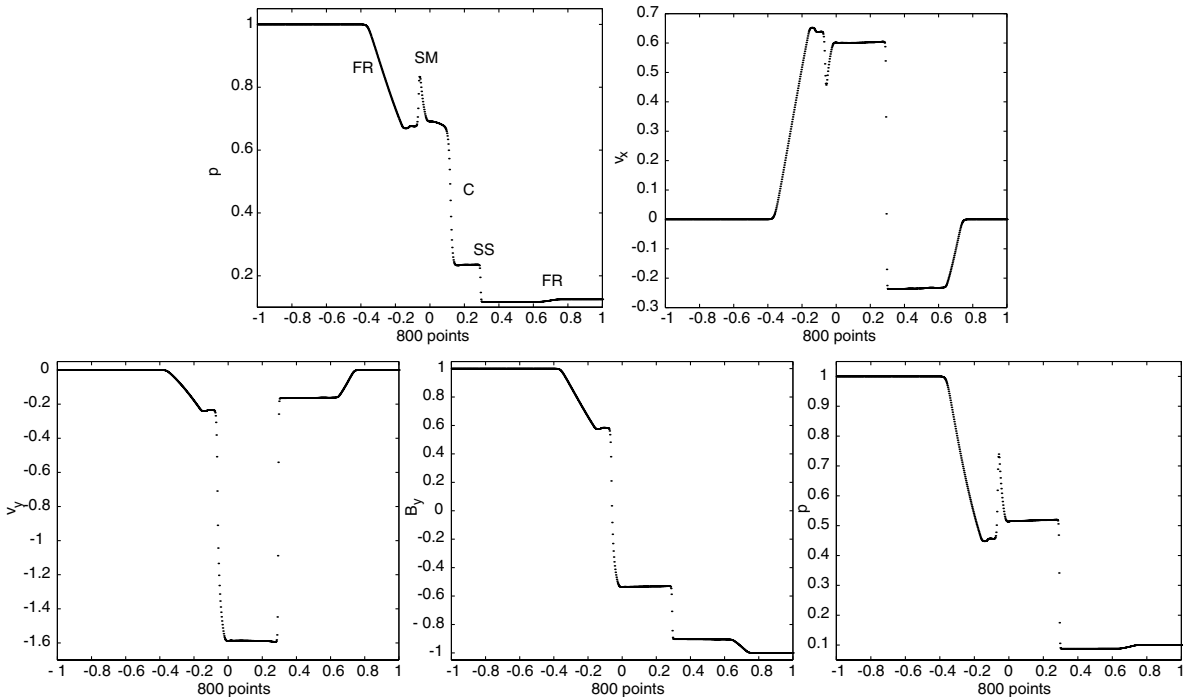
$$\Delta \phi = -\nabla \cdot \mathbf{B} \tag{3.13}$$

Fig. 3. Results of Brio–Wu shock tube problem at $t = 0.2$ computed with 800 grid points using the second-order central scheme. Here f'_j is computed component-wise with the MinMod limiter using the JFF version (2.21) with $\alpha = 1.4$.

with the appropriate boundary conditions. Since the coplanar structure of the problem guarantees $\partial B_y/\partial y = 0$, only the components B_x and B_z need to be updated. We use a fast Poisson solver – the Fishpack hwscrt code [1], for the standard five point discretization of the potential ϕ, and central differences for the divergence of the magnetic field, $\nabla \cdot \mathbf{B}$. The corrected divergence-free magnetic field, $\mathbf{B}^c$, is then realized as

$$\mathbf{B}^c = \mathbf{B} + \nabla\phi. \tag{3.14}$$

Applying the divergence operator $\nabla = (\partial/\partial x, \partial/\partial z)$ to both sides of (3.14), one can easily verify that $\nabla \cdot \mathbf{B}^c = 0$. We use here only one out of several methods to enforce the so-called 'constrained transport' and we refer the reader to [2,24] and the references therein for a general discussion and [11] for handling the solenoidal constraint in the context of MHD schemes over staggered grids.

4. One-dimensional numerical results

Eqs. (1.1)–(1.4) admit the conservative form (2.1) with

$$u = (\rho, \rho v_x, \rho v_y, \rho v_z, B_y, B_z, e)^{\mathrm{T}}, \tag{4.1}$$

$$\begin{aligned} f(u) = (\rho v_x, \rho v_x^2 + p^* - B_x^2, \rho v_x v_y - B_x B_y, \rho v_x v_z - B_x B_z, B_y v_x - B_x v_y, B_z v_x \\ - B_x v_z, (e + p^*)v_x - B_x(B_x v_x + B_y v_y + B_z v_z))^{\mathrm{T}}, \end{aligned} \tag{4.2}$$

where $p^* = p + (1/2)B^2$ stands for the total pressure (static plus magnetic).

In this section we present numerical simulations of the one-dimensional MHD equations (2.1)–(2.28), (3.1)–(3.14), (4.1), (4.2). The results were obtained using different versions of the second-order Ness-
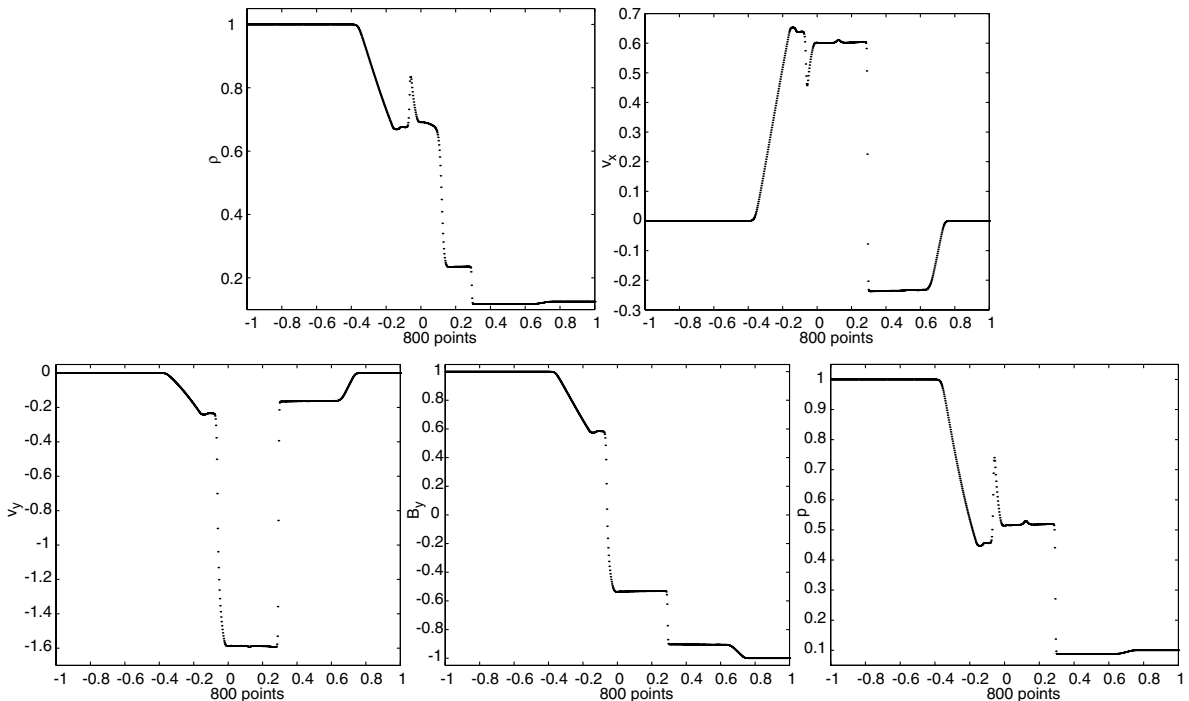

Fig. 4. Results of Brio–Wu shock tube problem at $t = 0.2$ computed with 800 grid points using the second-order central scheme. Here, the numerical flux, $f'_j = A(w_j^n)w'_j$, is evaluated using the MinMod limiter (2.20) with $\alpha = 1.4$.

yahu–Tadmor central scheme (2.19)–(2.21), (2.26),(2.28), based on the MinMod limiter (2.8), and third-order Liu–Tadmor central scheme, (2.22)–(2.25), (2.27), (2.28), based on the non-oscillatory limiter (2.14)–(2.16). The schemes are implemented for the approximate solution of two coplanar shock tube MHD models described by Brio and Wu [3]. We use a uniform grid in the space discretization, and in both cases we choose the time step dynamically with CFL restriction

$$\Delta t = 0.4 \frac{\Delta x}{\max_{k} | a_k(u) |}, \tag{4.3}$$

where $\{a_k(u)\}_k$ are the eigenvalues of the Jacobian matrix of $f(u)$.

4.1. Brio–Wu shock tube problem

The first one-dimensional Riemann problem we consider consists of a shock tube with two initial equilibrium states, u_l and u_r,

$$(\rho, v_x, v_y, v_z, B_y, B_z, p)^{\mathrm{T}} = \begin{cases} (1.0, 0, 0, 0, 1.0, 0, 1.0)^{\mathrm{T}} & \text{for } x < 0, \\ (0.125, 0, 0, 0, -1.0, 0, 0.1)^{\mathrm{T}} & \text{for } x > 0, \end{cases} \tag{4.4}$$

and complemented with the constant values of $B_x \equiv 0.75$ and $\gamma = 2$. The problem is solved for $x \in [-1,1]$ with 800 grid points, and numerical results are presented at $t = 0.2$. Figs. 3–6 show the density, the x- and y-velocity components, the y-magnetic field, and pressure profiles computed with different choices of numerical derivatives f'_j and u'_j.

We note that the hydrodynamical data of this problem are the same as that in Sod's shock tube problem of gas dynamics. The variety of MHD waves, however, poses a considerable challenge for high-resolution
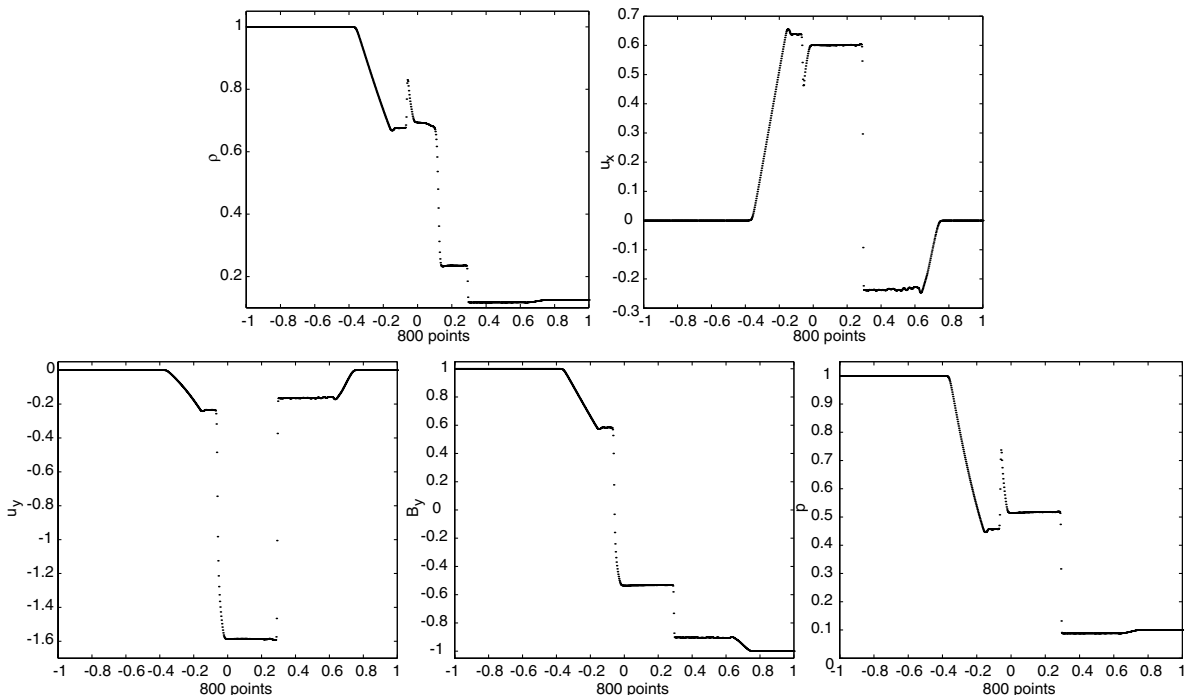

Fig. 5. Results of Brio–Wu shock tube problem at $t = 0.2$ computed with 800 grid points using the third-order central scheme. Here f'_j are computed component-wise using the JFF version (2.25).

methods such as the JFF central schemes described in this paper. The solution of this problem consists of a left-moving fast rarefaction wave (FR), a slow compound wave (SM) which results from an intermediate shock that changes B_y from 0.58 to -0.31 and a slow rarefaction that changes B_y from -0.31 -0.53, a contact discontinuity (C), a right-moving slow shock (SS), and a right-moving fast rarefaction wave (FR). Note that the solution to this problem is not unique if B_z and v_z are not identically zero.

Figs. 3–6 show the solutions calculated with different versions of the second- and third-order schemes. These results are comparable with the second-order upwind computations of Brio and Wu [3], and with the fifth-order WENO computations presented by Jiang and Wu [13]. Our numerical results demonstrate that central schemes – while avoiding any characteristic information other than an estimate of the maximal speed $\max_k |a_k(u)|$, they still capture the main features of the discontinuous MHD solutions.

In Figs. 5 and 6 we observe the third-order oscillations near the trailing edge of the fast rarefaction wave that are less evident with the second-order results. This is due to the higher order polynomial reconstruction; indeed, the same phenomenon is reported in [13] when comparing fifth-order WENO with second-order results. A final remark regarding the JFF, (2.21), (2.25): not only that they offer a more economical approach by avoiding costly matrix multiplication, but they also provide a better control of these oscillations as well as a better resolution of the contact discontinuity (C); in particular, they allow the expected jump in the density profile without disrupting the constant state of the remaining conserved quantities.

4.2. Brio–Wu high mach shock tube problem

The following shock tube model proposed by Brio and Wu [3] is used to check the robustness of the numerical schemes for high Mach number problems. The initial equilibrium states, u_l and u_r, are given by
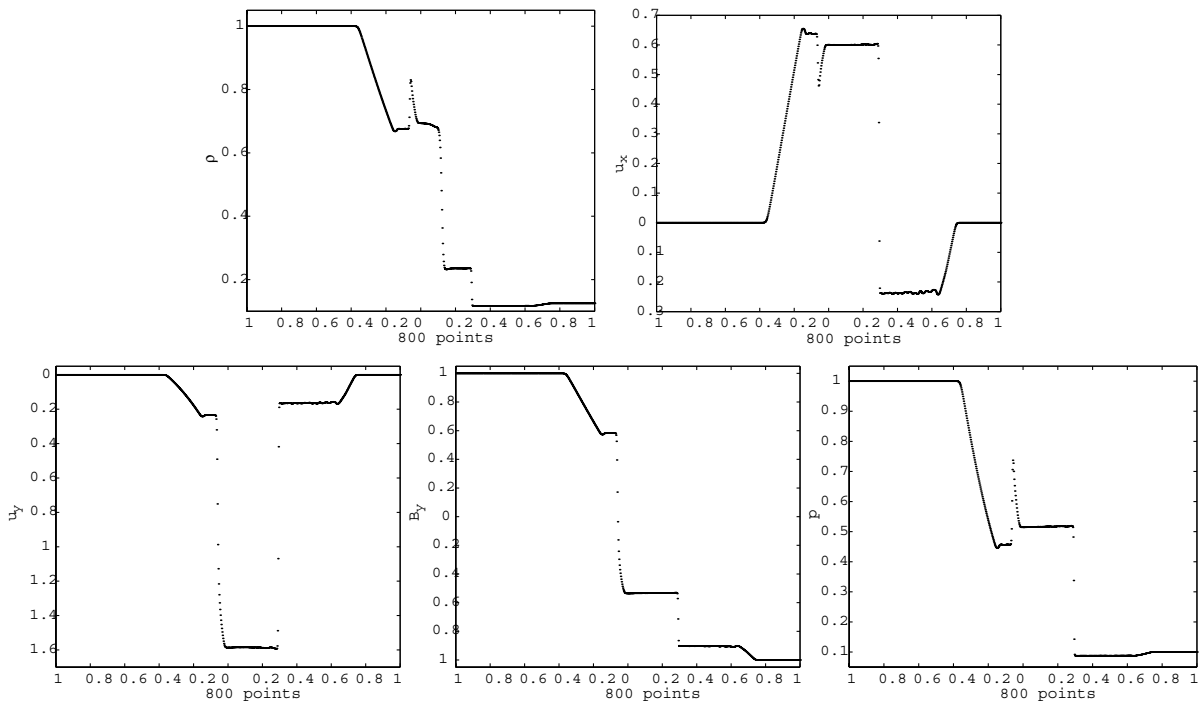

Fig. 6. Results of Brio–Wu shock tube problem at $t = 0.2$ computed with 800 grid points using the third-order central scheme. Here, the intermediate numerical fluxes, $\{f(w_j - \lambda\beta/2f'_j)\}'$, $\beta = 0, 1/2, 1$, are evaluated using the third-order limiter (2.23).
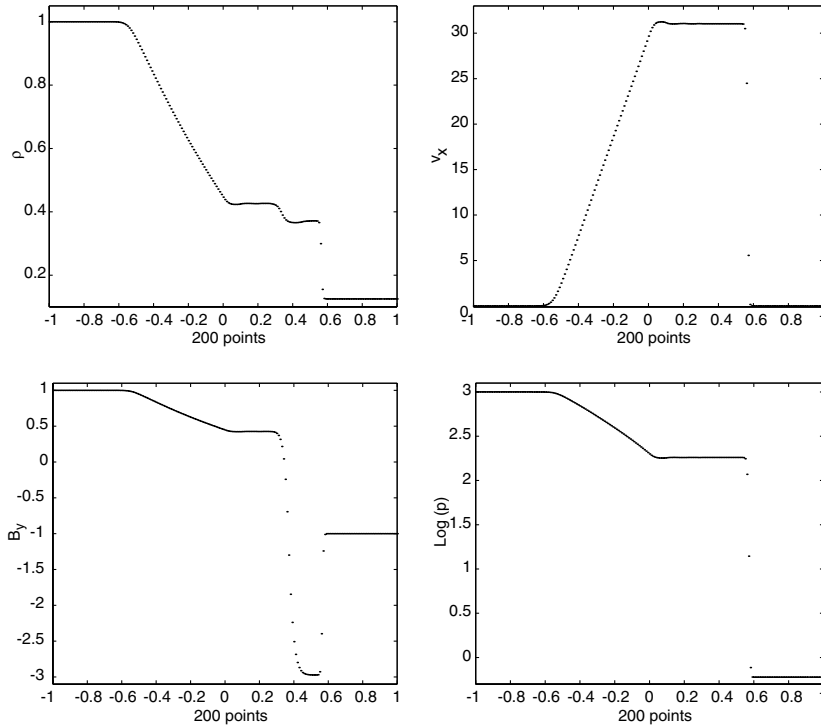
Fig. 7. Results of Brio–Wu high Mach problem at $t = 0.012$ computed with 200 grid points using second-order central scheme. Here f'_j are computed component-wise using the JFF (2.21) with $\alpha = 1.4$.

$$(\rho, v_x, v_y, v_z, B_y, B_z, p)^{\mathrm{T}} = \begin{cases} (1.0, 0, 0, 0, 1.0, 0, 1000)^{\mathrm{T}} & \text{for } x < 0, \\ (0.125, 0, 0, 0, -1.0, 0, 0.1)^{\mathrm{T}} & \text{for } x > 0 \end{cases} \tag{4.5}$$

complemented with the values of $B_x \equiv 0$, and $\gamma = 2$. The Mach number of the right-moving shock wave is 15.5. If the plasma pressure is replaced by the sum of the static and magnetic pressures – denoted by p^* above, the problem becomes a standard hydrodynamical Riemann problem. The solution is presented at $t = 0.012$, $x \in [-1,1]$, with 200 grid points and with CFL number 0.4, consult (4.3) (see Figs. 7 and 8).

The solution of this second Riemann problem consists of a left-moving fast rarefaction wave (FR), followed by a tangential discontinuity (TD), and a right moving fast shock (FS) with Mach number 15.5. Across the tangential discontinuity, the density, the magnetic field and the pressure can change, but both the fluid velocity and the total pressure are continuous.

As in the previous problem, our results are comparable to those obtained by Brio and Wu [3] with their second-order upwind method and the ones presented by Jiang and Wu [13], computed with a fifth-order WENO scheme. These results indicate that the methods described above are robust even in their 'greedy', Jacobian-free version.

5. Two-dimensional numerical results

In two space dimensions, Eqs. (1.1)–(1.4) admit the conservative form (3.1) with

$$u = (\rho, \rho v_x, \rho v_y, \rho v_z, B_x, B_y, B_z, e)^{\mathrm{T}}, \tag{5.1}$$
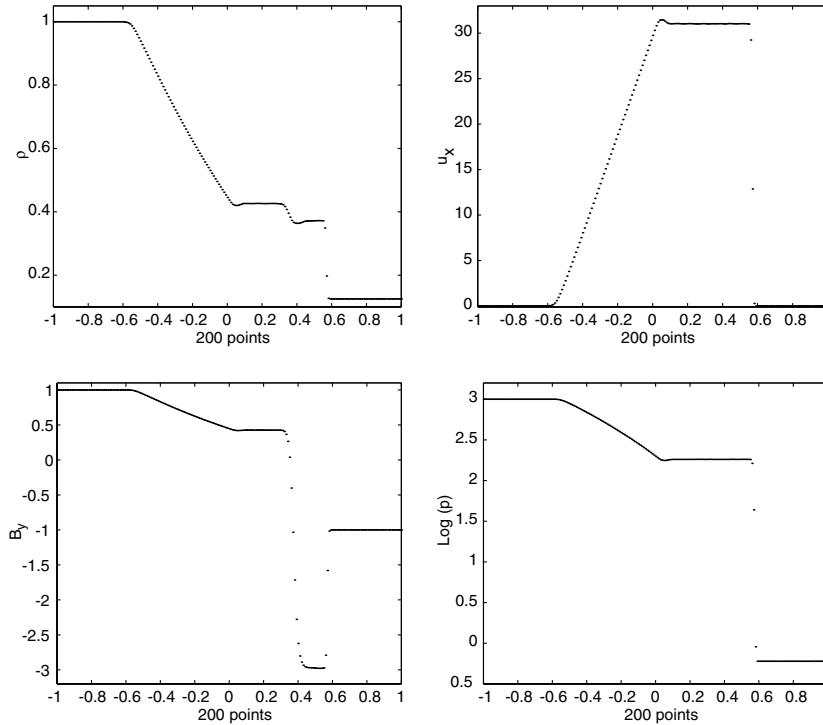
Fig. 8. Results of Brio–Wu high Mach problem at $t = 0.012$ computed with 200 grid points using third-order central scheme. Here, the intermediate numerical fluxes, $\{f(w_j - \lambda\beta/2f'_j)\}'$, $\beta = 0, 1/2, 1$, are evaluated using the third-order limiter (2.23).

$$
\begin{aligned}
f(u) = &(\rho v_x, \rho v_x^2 + p^* - B_x^2, \rho v_x v_y - B_x B_y, \rho v_x v_z - B_x B_z, 0, B_y v_x - B_x v_y, B_z v_x \\
&- B_x v_z, (e + p^*) v_x - B_x(B_x v_x + B_y v_y + B_z v_z))^{\mathrm{T}},
\end{aligned}
\tag{5.2}
$$

$$
\begin{aligned}
g(u) = &(\rho v_z, \rho v_z v_x - B_z B_x, \rho v_z v_y - B_z B_y, \rho v_z^2 + p^* - B_z^2, B_x v_z - B_z v_x, B_y v_z \\
&- B_z v_y, 0, (e + p^*) v_z - B_z(B_x v_x + B_y v_y + B_z v_z))^{\mathrm{T}}.
\end{aligned}
\tag{5.3}
$$

In this section we present the numerical solutions of two prototype models of two-dimensional MHD equations. For the first problem – the Kelvin–Helmholtz instability with transverse magnetic field configuration, we consider two different sets of boundary conditions in the x-direction: periodic in the first case and a free outflow boundary in the second convective setup. The second Orszag–Tang problem introduced by Orszag and Tang [22] as a simple model to study MHD turbulence, and has become a standard model to validate numerical algorithms. In both cases, the time scale, Δt is determined dynamically with CFL=0.4,

$$
\Delta t = \frac{0.4}{(\max_k |a_k(u)| / \Delta x) + (\max_k |b_k(u)| / \Delta z)},
\tag{5.4}
$$

where $\{a_k(u)\}_k$ and $\{b_k(u)\}_k$ represent the eigenvalues of the Jacobian matrices of $f(u)$ and $g(u)$, respectively.

5.1. Transverse Kelvin–Helmholtz instability

The Kelvin–Helmholtz instability arises when two superposed fluids flow one over the other with a relative velocity. It models, for example, the important mechanism for the momentum transfer at the Earth's

magnetopause boundary, which separates the solar wind from the Earth's magnetosphere [13]. We apply the second-order central scheme of Jiang and Tadmor, (3.7)–(3.12), for the two-dimensional periodic and convective models with transverse magnetic field configuration.

In both cases, the governing equations are (3.1) with u, $f(u)$ and $g(u)$ given by (5.1), (5.2) and (5.3), respectively, and are subject to initial conditions

$$(\rho, v_x, v_y, v_z, B_x, B_y, B_z, p)^{\mathrm{T}} = (1.0, v_{x0} + \tilde{v}_{x0}, 0, 0, 0, 1.0, 0, 0.5)^{\mathrm{T}}, \tag{5.5}$$

where

$$v_{x0} = \frac{v_0}{2} \tanh\left(\frac{z}{a}\right), \tag{5.6}$$

and

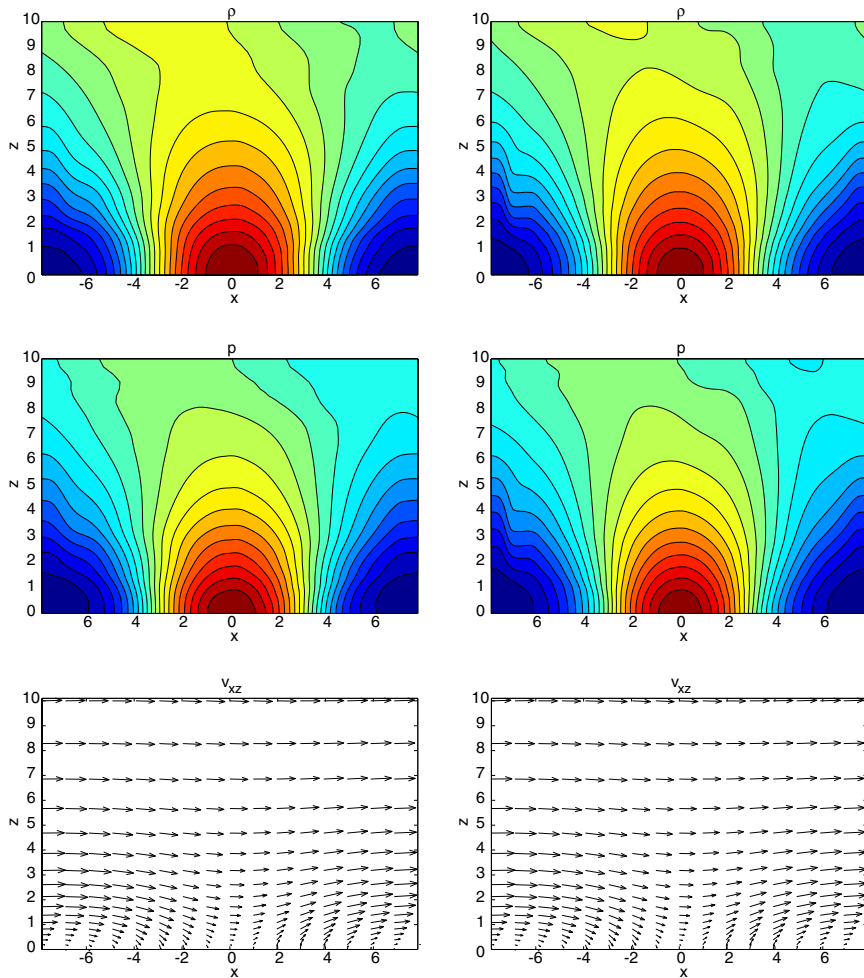
Fig. 9. Results of transverse Kelvin–Helmholtz instability with periodic x-boundary conditions. Left column uses 96×96 points and right column uses 192×192 grids, respectively. There are 20 contours for density and pressure. Red – high value, blue – low value. Density ranges from 0.79 to 1.2, pressure range from 0.32 to 0.71 and maximum value for the velocity is 1.25. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\tilde{v}_{x0} = \begin{cases} -\tilde{v}_0 \sin(\frac{2\pi x}{\lambda}) \frac{1}{1+z^2}, & \text{if } -\frac{\lambda}{2} < x < \frac{\lambda}{2}, \\ 0 & \text{otherwise}, \end{cases} \tag{5.7}$$

with $v_0 = 2$, $\tilde{v}_0 = -0.008$, $\lambda = 5\pi$ and $a = 1$. Also, the grids are stretched in the z-direction with a Roberts transformation (consult [13]),

$$z \leftarrow \frac{H \sinh(\tau z/2H)}{\sinh(\tau/2)}, \quad \tau = 6, \tag{5.8}$$

which renders a denser grid near $z = 0$ where the effect of the small initial perturbation $\tilde{v}_{x0}$ is more noticeable, and a coarser grid near $z = \pm H$, where little action takes place.

In the periodic case, the computational domain is $[-L/2, L/2] \times [0, H]$, with $L = 5\pi$ and $H = 10$. Reflection boundary conditions are applied for the outflow variables at the top boundary, $z = H$, and the bottom boundary values are recovered by symmetry, since ρ, p, and B_y are symmetric, and v_x and v_z are antisymmetric under the transformation $(x, z) \rightarrow (-x, -z)$. In Fig. 9, we present solutions at $t = 144$, with 96×96 and 192×192 grid points.
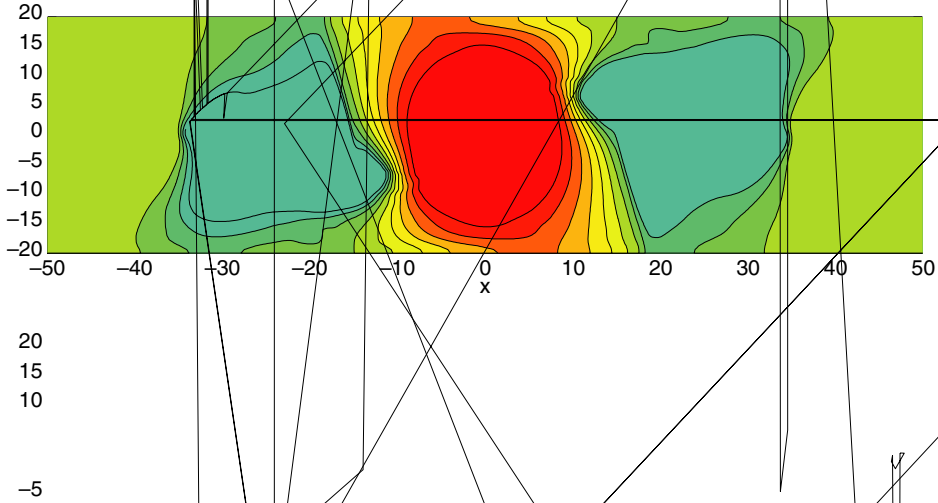
The resolution and accuracy of our results are comparable to those achieved by the second-order gas kinetic scheme of Tang and Xu [25] using a 200×200 uniform grid, and to those achieved by Jiang and Wu's fifth-order WENO scheme, [13], obtained with a coarser grid. It should be noted, however, that in the case of central schemes, the additional computational cost generated by thinner grids is compensated by the simplicity of the algorithm: no characteristic decompositions are computed, no Jacobians are required and dimensional splitting is avoided.

In the convective setup, the initial conditions and perturbation are the same as in the periodic setup, (5.5)–(5.7). Free outflow conditions are applied to all four boundaries of the computational domain $[-L/2,L/2]\times[-H,H]$. Here $H=20$ and $L=55\pi$, with $L\gg\lambda$ – so chosen to allow the excitation to convect freely without disturbing the x-boundaries. In this case, rather than using symmetry to reconstruct the solution in the bottom half of the domain, we test the symmetry preserving capabilities of our schemes by solving the problem in the entire domain.

Figs. 10 and 11 display the solution computed with 1056×192 grid points at $t=120$ and $t=145$, respectively. As in the periodic case, our results are comparable to those previously obtained with upwind
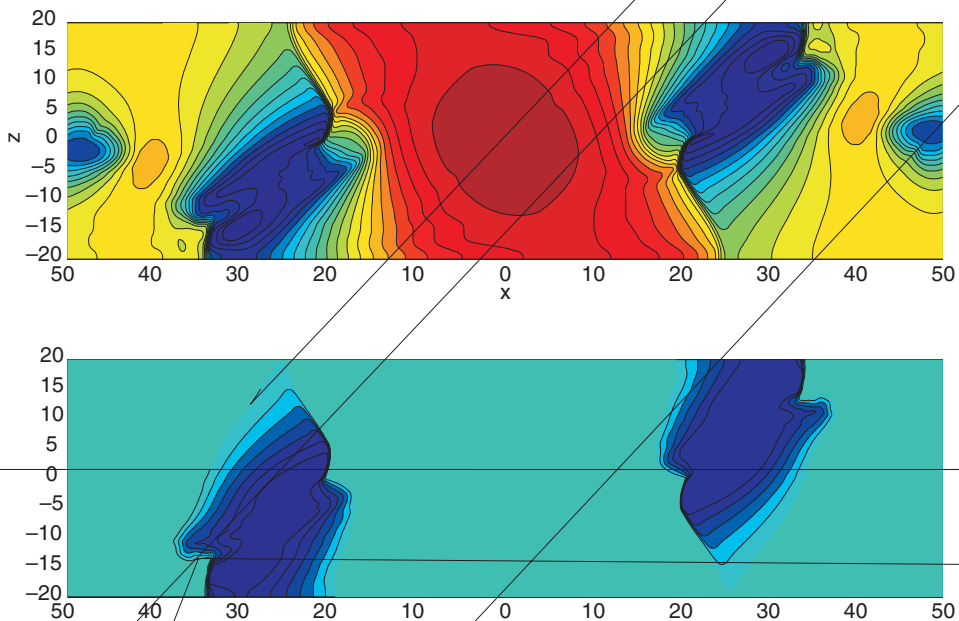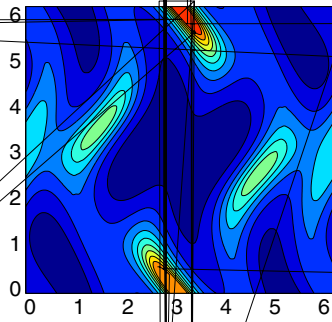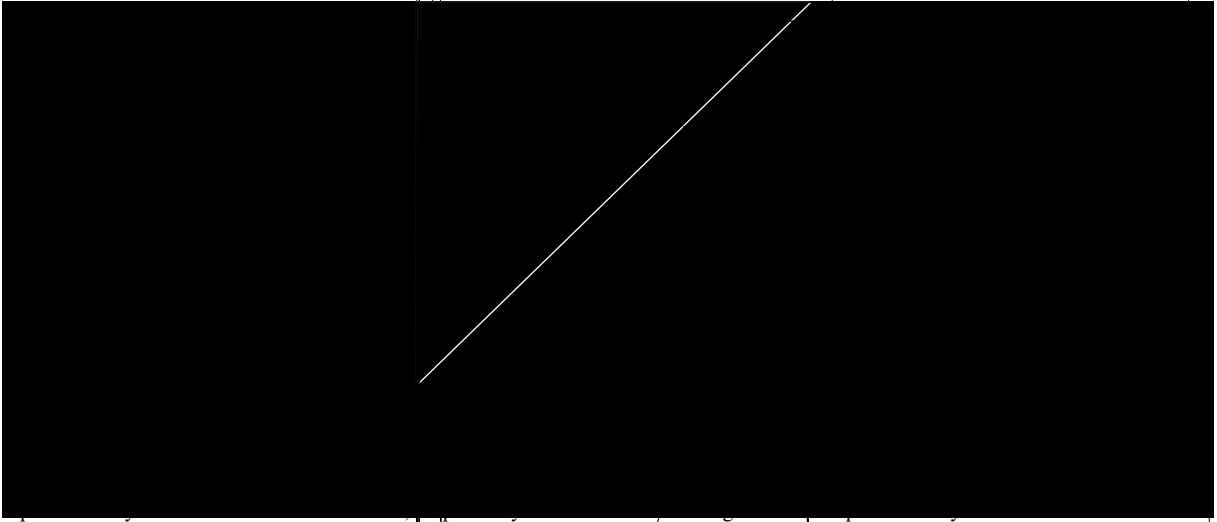
schemes, [13,25], and demonstrate the ability of central schemes to detect and resolve steep gradients without any detailed knowledge of the eigenstructure of the Jacobian matrices and to preserve the symmetry of the solution.

## 5.2. Orszag–Tang MHD turbulence problem

This model considers the evolution of a compressible Orszag–Tang vortex system. The evolution of the vortex system involves the interaction between several shock waves traveling at various speed regimes [5,13,26], which makes the problem especially attractive for numerical experiments.

The initial data are given by

$$\rho(x, z, 0) = \gamma^2, \quad v_x(x, z, 0) = -\sin z, \quad v_z(x, z, 0) = \sin x,$$
$$p(x, z, 0) = \gamma, \quad B_x(x, z, 0) = -\sin z, \quad B_z(x, z, 0) = \sin 2x,$$

where $\gamma = 5/3$. With this initial data, the root mean square values of the velocity and magnetic fields are both 1; the initial average Mach number is 1, and the average plasma beta is 10/3.

We solve the problem in $[0,2\pi] \times [0,2\pi]$, with periodic boundary conditions in both x- and z-directions using a uniform grid with 384×384 points.
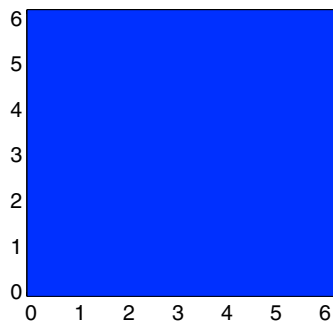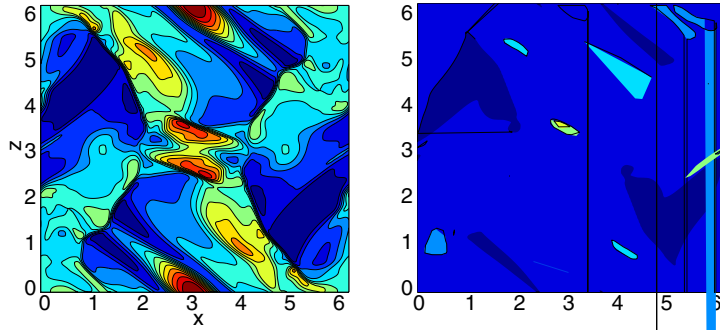
2   3   4   5   6

In this problem, the numerical scheme fails to satisfy the divergence-free constraint of the magnetic field, $\nabla\cdot\mathbf{B}=0$. In order to guarantee this condition and avoid numerical instability, we project the updated magnetic field $\mathbf{B}$ into its divergence-free component at the end of every time iteration by applying the correction (3.13) and (3.14).

Figs. 13–15 present the solution of the Orszag–Tang vortex system at $t = 0.5, t = 2$, and $t = 3$, respectively. Again, these results are comparable with those obtained by upwind schemes. The lower order of our methods require a thinner grid than Jiang and Wu's fifth-order WENO scheme for resolving accurately the various shocks that the vortex system develops. However, this relative loss of efficiency is still compensated by the simplicity in the implementation of the schemes (see Fig. 16).

6. Conclusions

The numerical results presented in this paper demonstrate the ability of central schemes to compute the discontinuous solutions of ideal MHD equations accurately. Our numerical tests are in excellent agreement with the results of Jiang and Wu [13], and they complement previous results of Wu and Chang [27] and the more recent results of Del Zanna et al. [7,8] where the efficiency of central schemes is demonstrated for other MHD models.

How one can quantify this efficiency? the answer depends on too many local features – computer code, hardware and database configurations, etc. which prevent a precise quantitative answer. We therefore report here on our 'subjective' results of CPU running time for solving the above Orszag–Tang problem, comparing the second-order staggered central scheme [12] vs. WENO scheme [13] (fifth-order in space and forth in time). The JFF of the second-order central schemes offered a speed-up factor of 25 in this case. We should point out, however, that the second-order results required a refinement of the spatial grid by a factor of ~ 2, in order to achieve a resolution similar to the fifth-order WENO. With the more restrictive CFL condition – the staggered central CFL condition is ~ 0.5 rather than 1, this increases the amount of 2D 'work' by a factor 8. Thus, the acceleration factor for a given resolution in this case is a factor of ~ 3. The additional gain offered by the black-box central solvers lies in their *simplicity*: neither characteristic decomposition nor dimensional splitting is required. The relative ease of implementation is highlighted by the 2D code in Appendix A, where the intricate eigensystem specified in [13, pp. 570–572] is completely avoided.

Finally, we observe that the lower resolution of our staggered central schemes necessitates further refinement of the spatial grids in order to achieve the same resolution as higher order upwind methods, resulting in a loss of efficiency. We will address this issue in a second part of our work on central schemes for MHD equations, where we use a semi-discrete version of the central schemes introduced in [15,16] and its higher-order extensions, consult [4,18]. This approach retains both the simplicity of implementation and the efficiency of Riemann-solver-free algorithms.

Acknowledgement

Appendix A. A central code for two-dimensional ideal MHD equations

The following C++ code was used to compute the solution of the convective Kelvin–Helmholtz Instability, Figs. 10–12. We refer the reader to http://www.math.ucla.edu/~jbalbas/MHD/code for this two-dimensional MHD code and additional software required for its implementation.

```
////////////////////////////////////////////////////////
// 2nd order Central Scheme for 2-d MHD Equations:
//
// Transverse Kelvin-Helmholtz instability--convective set-up.
//
// Jacobian Free form of Staggered LxF with MUSCL
// type reconstruction
//
// Input: grid size J x K, limiter parameter alpha, CFL number.
//
// Output: density, velocity field, magnetic field, pressure.
////////////////////////////////////////////////////////

int main()
{
////////////////////////////////////////////////////
// Grid
////////////////////////////////////////////////////
  double x_init, x_final, z_init, z_final, t_init, t_final, t;
  double dx, dz_eq, dt, lambda;
  double alpha, pi;
  double gamma=2.0;
  double cfl;
  double rho_doub, p_doub, A, B, max_eigx, max_eigz;
  double e_trans;
  long J, K;
  long j, k, j1, j2, l;
  long mid, s=0;
  bool odd=true;

  ifstream InFile;
  InFile.open("input", ios::in);
  if (!InFile)
  {
      cerr<<"Error in opening input file";
      exit(1);
  }

  InFile>>J;
  InFile>>K;
  InFile>>cfl;
  InFile>>alpha;

  CAMdoubleVector x(J+1),z(K+1),dz(K+1),nu(K+1);

  pi=4.0*atan2(1.0,1.0);
  x_init=-27.5*pi;
  x_final=27.5*pi;
  z_init=-20.0;
  z_final=20.0;
  t_init=0;
  t_final=145;
  dx=(x_final - x_init)/J;
  dz_eq=(z_final-z_init)/K;

  mid=long(ceil(K/2.0));

  x(1)=x_init;

  for (j=2; j<=J+1; j++)
      x(j)=x(j-1)+dx;

  z(1)=z_init;
  z_beg=z(1)-dz_eq;

  for (k=2; k<=K+1; k++)
      z(k)=z(k-1)+dz_eq;

  z_end=z(K+1)+dz_eq;

  z_beg=20*sinh(z_beg*3/20)/sinh(3);

  for (k=1; k<= K+1; k++)
      z(k)=20*sinh(z(k)*3/20)/sinh(3);

  z_end=20*sinh(z_end*3/20)/sinh(3);

  dz(1)=.5*(z(2)-z_beg);

  for (k=2; k<=K; k++)
   dz(k)=.5*(z(k+1)-z(k-1));

  dz(K+1)=.5*(z_end-z(K));
```

```
///////////////////////////////////////
// Solution Variables
///////////////////////////////////////
  CAMdoubleArray v_init(J+1,K+1,8);
  CAMdoubleArray vn(J+1,K+1,8);
  CAMdoubleArray f(J+1,K+1,8);
  CAMdoubleArray g(J+1,K+1,8);
  CAMdoubleArray f_p(J+1,K+1,8);
  CAMdoubleArray g_q(J+1,K+1,8);
  CAMdoubleArray v_star(J+1,K+1,8);
  CAMdoubleArray v_p(J+1,K+1,8);
  CAMdoubleArray v_q(J+1,K+1,8);
  CAMdoubleArray f_star(J+1,K+1,8);
  CAMdoubleArray g_star(J+1,K+1,8);
  CAMdoubleArray vnp1(J+1,K+1,8);

  double vx, vy, vz, cfx, cfz;
  CAMdoubleMatrix eigx(J+1,K+1), eigz(J+1,K+1);

  CAMdoubleVector vn_vector(8),fluxx(8),fluxz(8),v_star_vec(8);
  CAMdoubleVector e_int(42000), time(42000);

  double fu, fv, fw, gu, gv, gw, u, v, w;
  double vx_init;
  double I1, I2, I3 ,I4;

///////////////////////////////////////
// Initial Conditions
///////////////////////////////////////
  for (j=1; j<=J+1; j++){
  for (k=1; k<=K+1; k++){
   vx_init=tanh(z(k));
   v_init(j,k,1)=1.0;
   v_init(j,k,2)=vx_init;
   v_init(j,k,3)=0;
   v_init(j,k,4)=0;
   v_init(j,k,5)=0;
   v_init(j,k,6)=1.0;
   v_init(j,k,7)=0;}}

  j1=long(ceil(25*pi/dx)+1);
  j2=long(ceil(30*pi/dx)+1);

  for (j=j1; j<=j2; j++){
  for(k=1; k<=K+1; k++)
   v_init(j,k,2)=v_init(j,k,2)
   -.008*(sin(.4*x(j)))*(1.0/(1.0+pow(z(k),2.0)));}

  for (j=1; j<=J+1; j++){
  for (k=1; k<=K+1; k++)
   v_init(j,k,8)=1.0+.5*pow(v_init(j,k,2),2.0);}

///////////////////////////////////////////////////
// time loop
///////////////////////////////////////////////////
  vn=v_init;

  double sum_t=0;
  double dt_cpu=0;
  double t_start=realtime();

  do{
    for (j=1; j<=J+1; j++){
    for(k=1; k<=K+1; k++){
      rho_doub=vn(j,k,1);
      vx=vn(j,k,2)/rho_doub;
      vy=vn(j,k,3)/rho_doub;
      vz=vn(j,k,4)/rho_doub;
      p_doub=vn(j,k,8)-.5*((pow(vx,2) + pow(vy,2)
      +pow(vz,2))/rho_doub)-.5*(pow(vn(j,k,5),2)
      +pow(vn(j,k,6),2)+pow(vn(j,k,7),2));
      A=gamma*p_doub/rho_doub;
      B=(pow(vn(j,k,5),2)+pow(vn(j,k,6),2)
      +pow(vn(j,k,7),2))/rho_doub;
      cfx=sqrt(.5*(A+B+sqrt(pow(A+B,2)
      -4*A*pow(vn(j,k,5),2)/rho_doub)));
      cfz=sqrt(.5*(A+B+sqrt(pow(A+B,2)
      -4*A*pow(vn(j,k,7),2)/rho_doub)));
      eigx(j,k)=fabs(vx)+cfx;
      eigz(j,k)=fabs(vz)+cfz;}}

    max_eigx=eigx.max();
    max_eigz=eigz.max();
    dt=cfl/((max_eigx/dx)+(max_eigz/dz(mid)));
```

```
  lambda=dt/dx;                                        I3=.0625*(v_q(j,k,l)-v_q(j,k+1,l))
  t=t+dt;                                                -.5*nu(k)*(g_star(j,k+1,l)-g_star(j,k,l));
                                                       I4=.0625*(v_q(j+1,k,l)-v_q(j+1,k+1,l))
  for(k=1; k<=K+1; k++)                                  -.5*nu(k)*(g_star(j+1,k+1,l)-g_star(j+1,k,l));
    nu(k)=dt/dz(k);                                    vnp1(j,k,l)=.25*(vn(j+1,k,l)+vn(j,k,l)
                                                         +vn(j+1,k+1,l)+vn(j,k+1,l)) +I1+I2+I3+I4;}}}
  for (j=1; j<=J+1; j++){
  for (k=1; k<=K+1; k++){                             for (k=1; k<=K; k++){
  for (l=1; l<=8; l++)                                for(l=1; l<=8; l++){
     vn_vector(l)=vn(j,k,l);                            I3=.125*(v_q(J+1,k,l)-v_q(J+1,k+1,l))
    flux_x(vn_vector, fluxx);                            -nu(k)*(g_star(J+1,k+1,l)-g_star(J+1,k,l));
    flux_z(vn_vector, fluxz);                           vnp1(J+1,k,l)=.5*(vn(J+1,k,l)+vn(J+1,k+1,l)) +I3;}}

  for (l=1; l<=8; l++){                               for (j=1; j<=J; j++){
      f(j,k,l)=fluxx(l);                               for(l=1; l<=8; l++){
      g(j,k,l)=fluxz(l);}}}                             I1=.125*(v_p(j,K+1,l)-v_p(j+1,K+1,l))
                                                         -lambda*(f_star(j+1,K+1,l)-f_star(j,K+1,l));
  for (l=1; l<=8; l++){                                 vnp1(j,K+1,l)=.5*(vn(j+1,K+1,l)+vn(j,K+1,l)) +I1;}}
  for (k=1; k<=K+1; k++){
  for (j=2; j<=J; j++){                                for (l=1; l<=8; l++)
      fu=alpha*(f(j+1,k,l)-f(j,k,l));                     vnp1(J+1,K+1,l)=vn(J+1,K+1,l);
      fv=.5*(f(j+1,k,l)-f(j-1,k,l));
      fw=alpha*(f(j,k,l)-f(j-1,k,l));                  vn=vnp1;
      u=alpha*(vn(j+1,k,l)-vn(j,k,l));                 odd=false;}
      v=.5*(vn(j+1,k,l)-vn(j-1,k,l));
      w=alpha*(vn(j,k,l)-vn(j-1,k,l));              else{
      f_p(j,k,l)=minmod3(fu, fv, fw);                  for (j=2; j<=J+1; j++){
      v_p(j,k,l)=minmod3(u, v, w);}                    for (k=2; k<=K+1; k++){
                                                       for (l=1; l<=8; l++){
    f_p(1,k,l)=0;                                        I1=.0625*(v_p(j-1,k-1,l)-v_p(j,k-1,l))
    f_p(J+1,k,l)=0;                                        -.5*lambda*(f_star(j,k-1,l)-f_star(j-1,k-1,l));
    v_p(1,k,l)=0;                                       I2=.0625*(v_p(j-1,k,l)-v_p(j,k,l))
    v_p(J+1,k,l)=0;}                                      -.5*lambda*(f_star(j,k,l)-f_star(j-1,k,l));
                                                         I3=.0625*(v_q(j-1,k-1,l)-v_q(j-1,k,l))
  for (j=1; j<=J+1; j++){                                 -.5*nu(k)*(g_star(j-1,k,l)-g_star(j-1,k-1,l));
  for (k=2; k<=K; k++){                                  I4=.0625*(v_q(j,k-1,l)-v_q(j,k,l))
      gu=alpha*(g(j,k+1,l)-g(j,k,l));                     -.5*nu(k)*(g_star(j,k,l)-g_star(j,k-1,l));
      gv=.5*(g(j,k+1,l)-g(j,k-1,l));                      vnp1(j,k,l)=.25*(vn(j,k-1,l)+vn(j-1,k-1,l)
      gw=alpha*(g(j,k,l)-g(j,k-1,l));                      +vn(j,k,l)+ vn(j-1,k,l)) + I1+I2+I3+I4;}}}
      u=alpha*(vn(j,k+1,l)-vn(j,k,l));
      v=.5*(vn(j,k+1,l)-vn(j,k-1,l));                  for (k=2; k<=K+1; k++){
      w=alpha*(vn(j,k,l)-vn(j,k-1,l));                 for (l=1; l<=8; l++){
      g_q(j,k,l)=minmod3(gu, gv, gw);                    I3=.125*(v_q(1,k-1,l)-v_q(1,k,l))
      v_q(j,k,l)=minmod3(u, v, w);}                       -nu(k)*(g_star(1,k,l)-g_star(1,k-1,l));
                                                         vnp1(1,k,l)=.5*(vn(1,k-1,l)+vn(1,k,l))+I3;}}
    g_q(j,1,l)=0;
    g_q(j,K+1,l)=0;                                    for (j=2; j<=J+1; j++){
    v_q(j,1,l)=0;                                      for (l=1; l<=8; l++){
    v_q(j,K+1,l)=0;}}                                    I1=.125*(v_p(j-1,1,l)-v_p(j,1,l))
                                                           -lambda*(f_star(j,1,l)-f_star(j-1,1,l));
//////////////////////////////////////                  vnp1(j,1,l)=.5*(vn(j-1,1,l)+vn(j,1,l))+I1;}}
//predicted values
//////////////////////////////////////                for (l=1; l<=8; l++)
  for (j=1; j<=J+1; j++){                                 vnp1(1,1,l)=vn(1,1,l);
  for (k=1; k<=K+1; k++){
  for(l=1; l<=8; l++){                                 vn=vnp1;
      v_star(j,k,l)=vn(j,k,l)-.5*lambda*f_p(j,k,l)     odd=true;}
      -.5*nu(k)*g_q(j,k,l);
      v_star_vec(l)=v_star(j,k,l);}                  e_int(s)=0;
                                                     for(j=1; j<=J+1; j++){
    flux_x(v_star_vec, fluxx);                       for(k=1; k<=K+1; k++){
    flux_z(v_star_vec, fluxz);                           e_trans=.5*(pow(vn(j,k,4),2.0)/vn(j,k,1))*dx*dz(k);
                                                         e_int(s)=e_int(s)+e_trans;}}
  for (l=1; l<=8; l++){
      f_star(j,k,l)=fluxx(l);                        time(s)=t;
      g_star(j,k,l)=fluxz(l);}}}                      s++;

//////////////////////////////////////               dt_cpu=realtime()-t_start;
//corrected values                                    sum_t=sum_t+dt_cpu;
//////////////////////////////////////                t_start=realtime();
  if(odd){
      for (j=1; j<=J; j++){                         }while(t<t_final);
      for (k=1; k<=K; k++){
      for (l=1; l<=8; l++){                         writeout(vn,x,z,J,K);
          I1=.0625*(v_p(j,k,l)-v_p(j+1,k,l))
          -.5*lambda*(f_star(j+1,k,l)-f_star(j,k,l));  return 0;
          I2=.0625*(v_p(j,k+1,l)-v_p(j+1,k+1,l))    }
          -.5*lambda*(f_star(j+1,k+1,l)-f_star(j,k+1,l)); //////////////////////////////////////
```

References

[1] J. Adams, P. Swarztrauber, R. Sweet, Fishpack. Available from <http://www.scd.ucar.edu/css/software/fishpack/>.

[2] J.U. Brackbill, J.C. Barnes, The effect of Nonzero div B on the numerical solution of the magnetohydrodynamic equations, J. Comput. Phys. 35 (1980) 426.

[3] M. Brio, C.C. Wu, An upwind differencing scheme for the equations of magnetohydrodynamics, J. Comput. Phys. 75 (2) (1988) 400–422.

[4] F. Bianco, G. Puppo, G. Russo, High order central schemes for hyperbolic systems of conservation laws, SIAM J. Sci. Comput. 21 (1) (1999) 294–322.

[5] R.B. Dahlburg, J.M. Picone, Evolution of the Orszag–Tang vortex system in a compressible medium. I . Initial average subsonic flow, Phys. Fluids B 1 (1989) 2153.

[6] W. Dai, P.R. Woodward, A simple finite difference scheme for multidimensional magnetohydrodynamical equations, J. Comput. Phys. 142 (1998) 331–369.

[7] L. Del Zanna, N. Bucciantini, P. Londrillo, A third order shock-capturing code for relativistic 3-D MHD, Mem. Soc. Astron. Italiana 1 (2002).

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

[9] B. Engquist, O. Runborg, Multi-phase computations in geometrical optics, J. Comput. Appl. Math. 74 (1996) 175–192.

[10] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes III, J. Comput. Phys. 71 (1987) 231–303.

[11] D.R. van der Heul, C. Vuik, P. Wesseling, A conservative pressure-correction method for the Euler and ideal MHD equations at all speeds, Int. J. Numer. Meth. Fluids 40 (2002) 521–529.

[12] G.-S. Jiang, E. Tadmor, Nonoscillatory central schemes for multidimensional hyperbolic conservation laws, SIAM J. Sci Comput. 19 (6) (1998) 1892–1917.

[13] G.-S. Jiang, C.C. Wu, A high-order WENO finite difference scheme for the equations of ideal magnetohydrodynamics, J. Comput. Phys. 150 (1999) 561–594.

[14] S. Jin, Z.-P. Xin, Numerical passage from systems of conservation Laws to Hamilton–Jacobi equations and relaxation schemes, SIAM J. Numer. Anal. 35 (1994) 2385–2404.

[15] A. Kurganov, E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations, J. Comput. Phys. 160 (2000) 214–282.

[16] A. Kurganov, S. Noelle, G. Petrova, Semi-discrete central-upwind schemes for hyperbolic conservation laws and Hamilton–Jacobi equations, SIAM J. Sci. Comput. 23 (2000) 707–740.

[17] B. van Leer, Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method , J. Comput. Phys. 32 (1979) 101–136.

[18] D. Levy, G. Puppo, G. Russo, A fourth order central WENO scheme for multidimensional hyperbolic systems of conservation laws, SIAM J. Sci. Comput. (24) (2002) 480–506.

[19] X.-D. Liu, E. Tadmor, Third order nonoscillatory central scheme for hyperbolic conservation laws, Numer. Math. 79 (1998) 397–425.

[20] X.-D. Liu, S. Osher, Nonoscillatory third order accurate self-similar maximum principle satisfying shock capturing schemes. I , Sinum 33 (1996) 760–779.

[21] H. Nessayahu, E. Tadmor, Non-oscillatory central differencing for hyperbolic conservation laws, J. Comput. Phys. 87 (2) (1990) 408–463.

[22] S.A. Orszag, C.M. Tang, Small-scale structure of two-dimensional magnetohydrodynamic turbulence, J. Fluid Mech. 90 (1979) 129.

[23] J.M. Picone, R.B. Dahlburg, Evolution of the Orszag–Tang vortex system in a compressible medium II. Supersonic flow, Phys. Fluids B 3 (1991) 29.

[24] K.G. Powell, P.L. Roe, R.S. Myong, T. Gombosi, D. deZeeuw, An upwind scheme for magnetohydrodynamics, in: AIAA 12th Computational Fluid Dynamics Conference, San Diego, CA, June 19–22, 1995, pp. 661–674.

[25] H.-Z. Tang, K. Xu, A high-order gas-kinetic method for multidimensional ideal magnetohydrodynamics, J. Comput. Phys. 65 (2000) 69–88.

[26] C.C. Wu, Formation, structure, and stability of MHD intermediate shocks, J. Geophys. Res. 95 (A6) (1990) 8149–8175.

[27] C.C. Wu, T. Chang, Further study of the dynamics of two-dimensional MHD coherent structures – a large scale simulation, J. Atmos. Solar Terrestr. Phys. 95 (2001) 1447–1453.