

A robust moving mesh finite volume method applied to 1D hyperbolic conservation laws from magnetohydrodynamics

A. van Dam *, P.A. Zegeling

Mathematical Institute, Utrecht University, P.O. Box 80.010, 3508 TA Utrecht, The Netherlands

Received 13 July 2005; received in revised form 29 November 2005; accepted 21 December 2005

Available online 17 February 2006

Abstract

In this paper we describe a one-dimensional adaptive moving mesh method and its application to hyperbolic conservation laws from magnetohydrodynamics (MHD). The method is robust, because it employs automatic control of mesh adaptation when a new model is considered, without manually-set parameters. Adaptive meshes are a common tool for increasing the accuracy and reducing computational costs when solving time-dependent partial differential equations (PDEs). Mesh points are moved towards locations where they are needed the most. To obtain a time-dependent adaptive mesh, monitor functions are used to automatically ‘monitor’ the importance of the various parts of the domain, by assigning a ‘weight’-value to each location. Based on the equidistribution principle, all mesh points are distributed according to their assigned weights. We use a sophisticated monitor function that tracks both small, local phenomena as well as large shocks in the same solution. The combination of the moving mesh method and a high-resolution finite volume solver for hyperbolic PDEs yields a serious gain in accuracy at relatively no extra costs. The results of several numerical experiments including comparisons with h-refinement are presented, which cover many intriguing aspects typifying nonlinear magnetofluid dynamics, with higher accuracy than often seen in similar publications.

© 2006 Elsevier Inc. All rights reserved.

PACS: 02.70.Bf; 52.30.Cv; 52.35.Bj; 52.35.Tc; 52.65.Kj

MSC: 35L60; 35L65; 65M50; 76L05; 76M12; 76W05

Keywords: Moving mesh; Adaptive mesh refinement; Monitor function; Finite volumes; Conservation laws; Magnetohydrodynamics

1. Introduction

Adaptive techniques have become common use in many solvers for partial differential equations (PDEs) over the past decades. Finite volume methods are often enhanced with local mesh refinement. Moving mesh techniques have mainly been used in combination with finite element and finite difference methods in the past.

* Corresponding author. Tel.: +31 30 2531529; fax: +31 30 2518394.

E-mail addresses: dam@math.uu.nl (A. van Dam), zegeling@math.uu.nl (P.A. Zegeling).

We combine a high-resolution finite volume solver with a moving mesh method, further improved by a sophisticated monitor function. This results in a robust method for solving hyperbolic systems of nonlinear PDEs: little user input is needed, the solver automatically adapts itself to the considered problem. The method was successfully used on hyperbolic macroscopic traffic flow models, and gas dynamics [22]. In this paper we solve a range of problems from magnetohydrodynamics (MHD), including interesting, often ignored, physical aspects of the solutions.

Many interesting phenomena in plasma fluid dynamics can be described within the framework of magnetohydrodynamics (MHD). Numerical studies in plasma flows frequently involve simulations with highly varying spatial and temporal scales. As a consequence, numerical methods on uniform meshes may be inefficient to use, since a very large number of mesh points is needed to resolve the spatial structures, such as shocks, contact discontinuities, shear layers, or current sheets. For the efficient study of these phenomena, we need adaptive mesh methods which automatically track and spatially resolve all of these structures. The problems considered here come from previous work by Tóth et al. [21,20], Keppens [11], Torrilhon [19], and Zegeling and Keppens [27].

Research in adaptive methods has many aspects. Firstly, there are several different methods. Local mesh refinement, or h-refinement, adapts the mesh by locally adding or removing mesh points. This technique has gotten the most attention, as the refinement is easily prescribed, and error analysis is still carried out fairly easily. Moving mesh methods, or r-refinement, relocate mesh points to refine the mesh where needed. Although the governing equations for mesh adaptation are more complex, this method has distinct advantages over h-refinement. In principle, it is easy to implement; no mesh points are added or removed, so that administration is no issue here. In its uncoupled form, which we use here, it can be combined with any existing PDE solver without necessary changes. Finally, moving the mesh points to any location provides more freedom in adaptation than inserting new points at discrete locations.

Huang et al. [8] prescribe mesh movement by a moving mesh PDE (MMPDE), which is solved simultaneously with the physical PDEs for one-dimensional models. Although this avoids solution interpolation, the coupled system may be hard to solve due to differences in time scales and desired error tolerances. For two-dimensional models the MMPDE and physical PDEs are often decoupled and solved in an alternating way, as Huang and Russell [9] show. Stockie et al. [12] also use an MMPDE-based decoupled approach for solving one-dimensional hyperbolic systems of conservation laws. It is similarly based on the equidistribution principle that follows from a variational formulation of mesh energy minimization. Tang and Tang [15] extend this approach to two-dimensional domains, but use a stationary description for the mesh movement, hence a decoupled approach by definition. Their monitor function still needs parameterization for each new problem by hand, though. In this paper, we take the latter approach, with an improved monitor function. Zegeling et al. [26] have recently used a similar method for two-dimensional hydrodynamic problems. The smoothness of mesh distribution is important for decreasing interpolation errors in the decoupled approach. The most powerful means for this is a good choice of monitor function. Huang has done much research on different monitor functions [5, with Cao and Russell], monitor quality [6], and mesh quality [7]. Error analysis quickly becomes complicated for moving mesh methods, but Beckett and Mackenzie [1,2] have done some convergence studies for these methods. Tang [16] recently presented an interesting overview paper on moving mesh methods for computational fluid dynamics. Zegeling and Keppens [27] also employ a moving mesh method, but it is fully coupled and solved using the method of lines and an implicit time solver. Although their mesh movement is fairly sophisticated, ensuring mesh-consistency and smooth mesh movement, it still needs manually-set adaptivity parameters. Furthermore, an artificial diffusion term is added in order to handle discontinuities in the physical solution. To avoid these artificial terms, we use a high-resolution finite volume method with MUSCL-type flux-limiters as proposed by Van Leer [25].

The layout of this paper is as follows. In the next section, we present the full set of MHD equations and their physical meaning. In Section 3, we describe the adaptive moving mesh method, based on the equidistribution principle, including a conservative solution interpolation. This is followed by details on the high-resolution finite volume method. Special attention is also given to a more sophisticated monitor function. Numerical experiments are presented in Section 4. Not only accuracy is considered, but also computational efficiency, in comparison with uniform methods. Also, some experiments compare r-refinement with h-refinement. Besides, interesting physical aspects of MHD are studied, such as pseudo-convergence to incorrect crit-

ical solutions, propagation of Alfvén waves, and high speed magnetosonic effects. Section 5 presents conclusions and suggestions for improvement.

2. The equations of magnetohydrodynamics

The MHD equations govern the dynamics of a charge-neutral ionized gas, or ‘plasma’. Just as the conservative Euler equations provide a continuum description for a compressible gas, the MHD equations express the basic physical conservation laws a plasma must obey. Because plasma dynamics are influenced by magnetic fields through the Lorentz-force, the needed additions in going from hydrodynamic to magnetohydrodynamic behavior consist of a vector equation for the magnetic field evolution and extra terms in the Euler system that quantify the magnetic force and energy density.

Using the conservative variables density ρ , momentum density $\mathbf{m} \equiv \rho \mathbf{v}$ (with velocity \mathbf{v}), magnetic field \mathbf{B} , and total energy density e , the ideal MHD equations can be written as follows (cf. [4,21,20]):

Conservation of mass:

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

Conservation of momentum:

$$\frac{\partial \mathbf{m}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v} - \mathbf{B} \mathbf{B}) + \nabla p_{\text{tot}} = 0. \quad (2)$$

Magnetic field induction:

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{v} \mathbf{B} - \mathbf{B} \mathbf{v}) = 0. \quad (3)$$

Conservation of energy:

$$\frac{\partial e}{\partial t} + \nabla \cdot (e \mathbf{v} + \mathbf{v} p_{\text{tot}} - \mathbf{B} \mathbf{B} \cdot \mathbf{v}) = 0. \quad (4)$$

Hereafter, we will abstract from the above four quantities by introducing the solution vector $q(\mathbf{x}, t)$, where $\mathbf{x} \equiv [x, y]^T$ or $\mathbf{x} \equiv [x, y, z]^T$. In (2) and (4) the total pressure p_{tot} consists of both a thermal and a magnetic contribution, as given by:

$$p_{\text{tot}} = p + \frac{B^2}{2}, \quad \text{where } p = (\gamma - 1) \left(e - \rho \frac{v^2}{2} - \frac{B^2}{2} \right) \quad (5)$$

is the thermal pressure ($B^2 \equiv \mathbf{B} \cdot \mathbf{B}$). The adiabatic constant γ is the ratio of specific heats of the plasma. This set of equations must be solved in conjunction with an important condition on the magnetic field \mathbf{B} , namely the non-existence of magnetic ‘charge’ or monopoles. Mathematically, it is easily demonstrated that this property can be imposed as an initial condition alone, since

$$\nabla \cdot \mathbf{B}|_{t=0} = 0 \Rightarrow \nabla \cdot \mathbf{B}|_{t \geq 0} = 0. \quad (6)$$

In multi-dimensional numerical MHD, the combined spatio-temporal discretization may not always ensure this conservation of the solenoidal character of the vector magnetic field. Note that in our 1D applications this solenoidal property is satisfied automatically by construction (see below).

2.1. Derivation of 1.5D and 1.75D models

If we restrict the MHD model (1)–(6) to variations in one spatial dimension x , i.e., $\partial q / \partial y = 0$, with possibly non-vanishing y -components for the vector quantities, we obtain a 5-component PDE system in 1D, which is sometimes referred to as ‘1.5D’. If we also include possibly non-vanishing z -components of the vector quantities, but still keep $\partial / \partial z = 0$ for the flux, we obtain a 7-component PDE system in 1D, which is sometimes referred to as ‘1.75D’. This system is formally written as

$$\frac{\partial}{\partial t} q + \frac{\partial}{\partial x} f(q) = 0, \quad x \in [x_L, x_R], \quad t > 0. \quad (7)$$

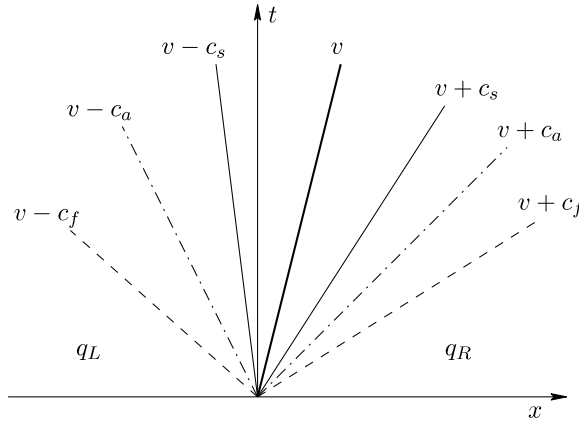


Fig. 1. Wave structure of a 1.75D MHD Riemann problem.

Here, $q = (\rho, m_1, m_2, m_3, B_2, B_3, e)^T$ is the vector of conserved variables (m_1, m_2, m_3 are now the x -, y - and z -components of the momentum vector and B_2 and B_3 denote the y - and z -component of the magnetic induction), with the flux-vector $f = (f_1, \dots, f_7)^T$ given by

$$\begin{aligned}
 f_1 &= m_1, \\
 f_2 &= \frac{m_1^2}{\rho} - \bar{B}_1^2 + (\gamma - 1)e - (\gamma - 1) \frac{m_1^2 + m_2^2 + m_3^2}{2\rho} + (2 - \gamma) \frac{\bar{B}_1^2 + B_2^2 + B_3^2}{2}, \\
 f_3 &= \frac{m_1 m_2}{\rho} - \bar{B}_1 B_2, \\
 f_4 &= \frac{m_1 m_3}{\rho} - \bar{B}_1 B_3, \\
 f_5 &= B_2 \frac{m_1}{\rho} - \bar{B}_1 \frac{m_2}{\rho}, \\
 f_6 &= B_3 \frac{m_1}{\rho} - \bar{B}_1 \frac{m_3}{\rho}, \\
 f_7 &= \frac{m_1}{\rho} \left(\gamma e - (\gamma - 1) \frac{m_1^2 + m_2^2 + m_3^2}{2\rho} + (2 - \gamma) \frac{\bar{B}_1^2 + B_2^2 + B_3^2}{2} \right) - \bar{B}_1 \left(\bar{B}_1 \frac{m_1}{\rho} + B_2 \frac{m_2}{\rho} + B_3 \frac{m_3}{\rho} \right).
 \end{aligned} \tag{8}$$

For notational convenience we do not use explicit vector notation for q and f : only for the physical MHD quantities we do so (e.g., \mathbf{m} and \mathbf{B}). The first component of the magnetic induction vector is kept at a constant value \bar{B}_1 . The vanishing divergence of the magnetic field is thereby trivially satisfied in this model situation. The remaining set of 7 PDEs given by (7) constitutes the physical model used for the ‘1.75D shock tube’ simulation found hereafter. Furthermore, several 1.5D simulations are shown; these are again described by (7), where f_4 and f_6 drop out of the flux formulae, as well as all terms involving m_3 and B_3 . Keppens [11] also derives these two models and solves two shock tube problems on uniform meshes.

We first indicate how this system is further manipulated and discretized to solve alternately for the adaptive mesh with its corresponding solution.

2.2. Eigen-structure for MHD

The eigenvalues of the flux Jacobian $f_q \equiv \partial f / \partial q$ represent the speeds at which the various waves of an MHD Riemann solution move. In 1.5D these are an entropy wave (v_1), two fast ($v_1 \pm c_f$) and two slow ($v_1 \pm c_s$) magnetosonic waves, where

$$c_{f,s}^2 = \frac{1}{2} \left(\frac{\gamma p + B^2}{\rho} \pm \sqrt{\left(\frac{\gamma p + B^2}{\rho} \right)^2 - 4 \frac{\gamma p}{\rho} \frac{\bar{B}_1^2}{\rho}} \right). \tag{9}$$

For the full system of MHD equations, i.e. 1.75D, two additional eigenvalues represent Alfvén waves with speed $v_1 \pm c_a$, where $c_a = B_1/\sqrt{\rho}$. In general, the following ordering holds: $c_s \leq c_a \leq c_f$. Fig. 1 shows the wave structure of a 1.75D MHD Riemann problem.

3. The moving mesh method

This section describes the moving mesh finite volume approach as introduced by Tang and Tang [15]. For increased robustness, we use a more sophisticated monitor function, originally proposed by Beckett et al. [3]. This combination yields a powerful solver that tracks and resolves both small, local and large solution gradients automatically. No parameter adaptation by hand using prior knowledge on the eventual shape of the solution is necessary. Hence, the solver can be quickly applied to problems from entirely different application areas.

The numerical algorithm is shown below. The symbol $Q_{j+1/2}$ represents the numerical solution for q , as will be introduced in (11). Each time step consists of a mesh moving step and a physical PDE solving step. The next two sections describe these separate steps. Finally, Section 3.3 deals with monitor functions in more detail.

Algorithm 1 MMFVSOLVE – 1D moving mesh finite volume PDE solver.

Generate an initial uniform mesh: $x_j^0 = x_L + j \cdot \frac{x_R - x_L}{N}$, $j = 0, \dots, N$.

Compute initial values $Q_{j+1/2}^0$ based on cell average of $q(x, 0)$.

while $t_n < T$ **do**

repeat

$v = 0$; $x_j^{[0]} = x_j^n$; $Q_{j+1/2}^{[0]} = Q_{j+1/2}^n$, $j = 0, \dots, N$.

 Move grid $\{x_j^{[v]}\}$ to $\{x_j^{[v+1]}\}$, using a Gauss–Seidel iteration (12).

 Compute the solution $\{Q_{j+1/2}^{[v+1]}\}$ on the new mesh, using (13).

until $v \geq v_{\max}$ or $\|x^{[v+1]} - x^{[v]}\| \leq \epsilon$

 Compute Q^{n+1} using high-resolution finite volumes (14).

end while

3.1. Mesh adaptation in 1D

The solution of the MHD equations, denoted by $q \in \mathbb{R}^m$, is defined on the physical domain $\Omega_p \equiv [x_L, x_R] \subset \mathbb{R}$ with coordinate x . Introducing a fixed computational domain $\Omega_c \equiv [0, 1] \subset \mathbb{R}$, with coordinate ξ , a coordinate transformation, or one-to-one mesh map, is defined by:

$$x = x(\xi), \quad \xi \in \Omega_c,$$

or its inverse

$$\xi = \xi(x), \quad x \in \Omega_p.$$

In a variational approach, finding the most appropriate mesh map $x(\xi)$ for some solution profile is equivalent to finding a ξ that minimizes a mesh energy functional $\mathcal{E}(\xi)$. A simple, but effective, mesh energy is:

$$\mathcal{E}(\xi) = \frac{1}{2} \int_{\Omega_p} \xi_x^2 \frac{1}{\omega} dx,$$

where $\omega > 0$ is a monitor function, which will be considered in more detail in Section 3.3. In general, ω is defined in terms of spatial derivatives of q . In a variational formulation (cf. [17]), minimization of the mesh energy yields the Euler–Lagrange equation:

$$\left(\frac{1}{\omega} \xi_x \right)_x = 0.$$

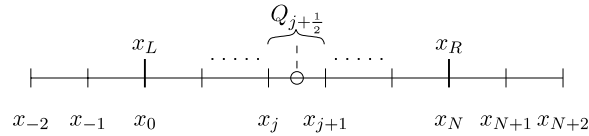


Fig. 2. The discretized spatial domain with ‘beyond-boundary’-points.

This is equivalent to the equidistribution principle in 1D, $\omega x_\xi = \text{constant}$, or:

$$(\omega x_\xi)_\xi = 0. \tag{10}$$

Now that the adaptive mesh is implicitly prescribed, a numerical algorithm can be set up, that determines the new mesh and updates the solution on it.

3.1.1. Domain discretization

To facilitate differential operators with stencils up to size 5, a domain discretization as depicted in Fig. 2 is used. The domain $\Omega_p \equiv [x_L, x_R]$ is discretized using $N + 1$ mesh points, with two additional mesh points on both sides outside Ω_p . The computational domain Ω_c is discretized with $N + 1$ uniform coordinates $\xi_j = j/N$ ($0 \leq j \leq N$).

As the finite volume solver uses cell averaged solution values, the discrete solution $Q_{j+1/2}$ is defined on the cell center:

$$Q_{j+1/2} \approx \frac{1}{\Delta x_{j+1/2}} \int_{x_j}^{x_{j+1}} q(x) dx, \quad 0 \leq j \leq N - 1, \tag{11}$$

where the local cell size, or mesh width is given by

$$\Delta x_{j+1/2} \equiv x_{j+1} - x_j.$$

3.1.2. Mesh redistribution

For every time $t > 0$, the new mesh should satisfy the redistribution Eq. (10). Using central differences for $(x_\xi)_{j+1/2}$, and inserting the current solution and monitor values yields a linear system in $[x_1, \dots, x_{N-1}]^T$, which is solved with a Gauss–Seidel (GS) iteration:

$$x_j^{[v+1]} = \frac{\omega(u_{j-1/2}^{[v]})x_{j-1}^{[v+1]} + \omega(u_{j+1/2}^{[v]})x_{j+1}^{[v]}}{\omega(u_{j-1/2}^{[v]}) + \omega(u_{j+1/2}^{[v]})}, \tag{12}$$

where $x_j^{[v+1]}$ ($v = 0, 1, \dots$) denotes the updated mesh point. Typically a mere three to five steps are performed before the mesh adaptation is considered appropriate ($v_{\max} = 3-5, \epsilon = 10^{-6}$). In most cases the v_{\max} -bound is reached before the ϵ -bound. Each mesh moving step also involves a solution interpolation, as described hereafter. The small number of GS steps keeps the costs of this interpolation low. Many, more advanced solvers exist, but accuracy of mesh movement is not the most critical aspect here. Tang and Tang give proof [15] of the preservation of monotonic order of $x^{[v]}$:

$$x_{j+1}^{[v]} > x_j^{[v]} \Rightarrow x_{j+1}^{[v+1]} > x_j^{[v+1]}, \quad 0 \leq j \leq N,$$

or, equivalently: x_ξ is strictly monotonically increasing. This is desirable, since otherwise mesh points might collapse and solution gradients could blow up.

3.1.3. Solution updating on the new mesh

In each redistribution step, mesh points x are moved to a new location \tilde{x} . Also, the solution Q needs to be updated on the new mesh, yielding \tilde{Q} . A conservative interpolation technique is used, to maintain physically correct solutions. Tang and Tang [15] introduce a conservative interpolation technique.

Assuming that the difference $c(x)$ between the old mesh x and new mesh $\tilde{x} \equiv x - c(x)$ is small, using a perturbation method eventually yields the interpolation relation:

$$Q_{j+1/2}^{[v+1]} = \frac{(x_{j+1}^{[v]} - x_j^{[v]})Q_{j+1/2}^{[v]} - ((cQ)_{j+1}^{[v]} - (cQ)_j^{[v]})}{x_{j+1}^{[v+1]} - x_j^{[v+1]}}, \quad (13)$$

where the upwinding (cf. Van Leer [23, Eq. (12)]) numerical fluxes are approximated by:

$$(cQ)_j = \frac{c_j}{2} (Q_j^+ + Q_j^-) - \frac{|c_j|}{2} (Q_j^+ - Q_j^-).$$

This method uses the ‘wave speed’ $c_j^{[v]} = x_j^{[v]} - x_j^{[v+1]}$, and Q_j^+ and Q_j^- , which approximate Q_j at a cell edge, are defined by (18). The interpolation relation (13) is in a conservative flux-differencing form, hence the interpolation satisfies the following conservation property:

$$\sum_j \Delta \tilde{x}_{j+\frac{1}{2}} \tilde{Q}_{j+\frac{1}{2}} = \sum_j \Delta x_{j+\frac{1}{2}} Q_{j+\frac{1}{2}}.$$

The updating of the solution is preceded by a single mesh redistribution step; the combination of the two forms the body of the GS iteration.

3.2. Finite volume solver for physical PDEs

On the redistributed mesh, the physical PDEs can be solved by any PDE solver that accepts nonuniform discretizations. We use a second order finite volume method. In the following, the mesh at time t_n is given by $x^n := x^{[v+1]}$.

One-dimensional hyperbolic systems of conservation laws are described by the PDE system in (7). Integrating the PDE over the control volume $[t_n, t_{n+1}] \times [x_j^n, x_{j+1}^n]$ leads to the following explicit finite volume method (we improve the time integration in (20)):

$$Q_{j+1/2}^{n+1} = Q_{j+1/2}^n - \frac{t_{n+1} - t_n}{x_{j+1}^n - x_j^n} (F_{j+1}^n - F_j^n) \quad (14)$$

$$=: Q_{j+1/2}^n + \Delta t^n L_{j+1/2}(Q^n), \quad (15)$$

where the cell average $Q_{j+1/2}^{n+1}$ is defined in (11) and F_j^n is some numerical flux satisfying

$$F_j^n = F(Q_j^{n,-}, Q_j^{n,+}), \quad F(Q, Q) = f(Q). \quad (16)$$

We use a *local* Lax–Friedrichs (LF) flux

$$F(Q_a, Q_b) = \frac{1}{2} [f(Q_a) + f(Q_b) - \max_{Q \in \{Q_a, Q_b\}} \{|f_q|\} (Q_b - Q_a)], \quad (17)$$

where the largest absolute eigenvalues of the Jacobian $f_q \equiv \partial f / \partial q$ are used. Local LF is less diffusive than normal LF, since it locally limits the numerical viscosity instead of having a uniform viscosity on the entire domain.

To determine flux values at cell boundaries, the solution values Q_j are approximated using values from the cell centers at both the left and right side. In (16), $Q_j^{n,\pm}$ are defined using the initial reconstruction technique:

$$Q_j^{n,\pm} = Q_{j\pm 1/2}^n + \frac{1}{2} (x_j^n - x_{j\pm 1}^n) \tilde{S}_{j\pm 1/2}^n, \quad (18)$$

where $\tilde{S}_{j+1/2}^n$ is an approximation of the slope q_x at $x_{j+1/2}^n$, defined by:

$$\tilde{S}_{j+1/2}^n = \left(\text{sign}(\tilde{S}_{j+1/2}^{n,+}) + \text{sign}(\tilde{S}_{j+1/2}^{n,-}) \right) \frac{|\tilde{S}_{j+1/2}^{n,+} \tilde{S}_{j+1/2}^{n,-}|}{|\tilde{S}_{j+1/2}^{n,+}| + |\tilde{S}_{j+1/2}^{n,-}|}, \quad (19)$$

with

$$\tilde{S}_{j+1/2}^{n,+} = \frac{Q_{j+3/2}^n - Q_{j+1/2}^n}{x_{j+3/2}^n - x_{j+1/2}^n}, \quad \tilde{S}_{j+1/2}^{n,-} = \frac{Q_{j+1/2}^n - Q_{j-1/2}^n}{x_{j+1/2}^n - x_{j-1/2}^n}.$$

The above is a MUSCL-type method, where the slope approximation (19) uses a monotonicity preserving slope limiter as formulated by Van Leer [24, Eq. (67)].

To obtain a higher accuracy in the time range, the standard *one-step* finite volume formulation (15) is replaced by a second-order Runge–Kutta scheme:

$$\begin{aligned} Q_{j+1/2}^* &= Q_{j+1/2}^n + \Delta t^n L_{j+1/2}(Q^n), \\ Q_{j+1/2}^{n+1} &= \frac{1}{2} \left(Q_{j+1/2}^n + Q_{j+1/2}^* + \Delta t^n L_{j+1/2}(Q^*) \right). \end{aligned} \tag{20}$$

In a method with changing mesh widths, the stability criterion for the time step is extra important. The standard CFL limit reads

$$\left| \frac{f_q \Delta t}{\Delta x} \right| \leq 1 \quad \forall \Delta t, \Delta x, \quad \text{eigenvalues of } f_q. \tag{21}$$

To enforce higher accuracy, the Courant number will here be limited by a parameter \mathcal{C} , thereby limiting the time step to:

$$\Delta t^n \leq \mathcal{C} \min_j \frac{\Delta x_{j+1/2}}{|f_q(Q_{j+1/2}^n)|}, \tag{22}$$

where $0 < \mathcal{C} \leq 1$. Notice how we determine the limit on the time step locally, instead of using, e.g., $\Delta t^n \leq \min_j \Delta x_{j+1/2} / \max_j |f_q(Q_{j+1/2}^n)|$.

3.3. A sophisticated monitor function

An often seen, most basic choice for controlling adaptivity is the arc length-type monitor function:

$$\omega(q) = \sqrt{1 + \alpha(\partial q / \partial x)^2}, \tag{23}$$

where the adaptivity parameter α controls the amount of adaptivity. The value 1 is set as floor on the monitor function to prevent *all* points from concentrating in just the steep parts of the solution. This type of monitor has two problems. Firstly, α is problem dependent; a problem from gas dynamics might require an α of an entirely different order than a problem from hyperbolic macroscopic traffic flow models. Secondly, α is a constant, whereas the solution profile might change significantly through time. The chosen α based on the solution at the initial time may be far from optimal at some point of time $t > 0$.

From now on we will use the term ‘critical’ for parts of the domain where refinement is especially necessary. For the monitor function (23), ‘critical’ is equivalent to ‘steep’, because of the first-order derivative. In general, higher order derivatives may be used as well.

To overcome the before-mentioned disadvantages, Beckett and Mackenzie [1] introduce a more sophisticated monitor function, which we schematically define as:

$$\omega(q) = \alpha(q) + \phi(q).$$

It has a solution dependent floor value $\alpha(q)$, where $\alpha(q)$ is defined as an average value of some function $\phi(q)$. Most often, ϕ will contain solution gradients. Huang [6] generalizes this monitor function with a parameter β that controls the ratio of points in critical parts. Here, we furthermore generalize to PDE *systems*, i.e. when q has $m > 1$ components. We define the monitor function $\omega(q) \in (\Omega_p \times \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^m) \rightarrow \mathbb{R}^{>0}$ as:

$$\omega(q) = \sum_{p=1}^m \left[(1 - \beta) \alpha_p(q) + \beta \left| \frac{\partial q_p}{\partial \xi} \right|^{1/2} \right], \tag{24}$$

where

$$\alpha_p(q) = \int_{\Omega_c} \left| \frac{\partial q_p}{\partial \xi} \right|^{1/2} d\xi. \tag{25}$$

The critical regions are now identified by the computational derivative $\partial q/\partial \xi$, which is smoother than the physical derivative $\partial q/\partial x$. The solution dependent $\alpha(q)$ averages this derivative for each component q_p separately. Finally, the m monitor values for all components are summed. Although β is still a user-defined parameter, we found $\beta = 0.8$ a suitable value for a range of different problems and keep it fixed at that for all numerical experiments in the following section.

Following the approach of Huang [6], it can be shown that for monitor (24), β is indeed the ratio of points in critical parts:

$$\beta = \frac{\int_{\Omega_p} \beta \phi \, dx}{\int_{\Omega_p} (1 - \beta) \langle \phi \rangle + \beta \phi \, dx}, \quad (26)$$

where $\langle \phi \rangle$ is the averaged ϕ , i.e., $\alpha(q)$. Hence, for our fixed choice of $\beta = 0.8$, approximately 80% of the mesh points is positioned in critical parts of the domain.

Another technique to prevent the mesh points from being moved too brusquely, when some local gradient changes rapidly, is to smoothen the monitor function. This is done by applying a low-pass filter, possibly multiple times:

$$\omega_{j+1/2}^{\text{smooth}} \leftarrow \frac{1}{4} \left(\omega_{j+3/2} + 2\omega_{j+1/2} + \omega_{j-1/2} \right), \quad (27)$$

where $\omega_{j+1/2} = \omega(Q_{j+1/2})$. Even with the sophisticated monitor (24) we found a single application of this smoothing operator to be beneficial and sufficient.

4. Numerical experiments

The moving mesh method is now used on a selection of problems from magnetohydrodynamics. Although still in one spatial dimension, these problems have five ($m = 5$) or even seven ($m = 7$) model equations, and consequently exhibit a range of shocks, rarefaction waves and contact discontinuities (at most m). Some problems were also used by Zegeling and Keppens [27] for testing their adaptive method of lines approach, which is a fully-coupled moving mesh method.

The numerical results are compared to a reference solution. Solutions to the shock tube problems (Sections 4.1–4.3) were obtained with the exact Riemann solver by Torrilhon [18]. The shear Alfvén problem (Section 4.4) is compared to a 2500 points adaptive solution. For all problems, solutions by the widely-used Versatile Advection Code [14] (VAC, see <http://www.phys.uu.nl/~toth>) are also considered.

A discrete L_1 norm is used for an error measure on adaptive meshes:

$$E_{L_1} = \sum_{j=1}^N \Delta x_j |Q_{j+1/2} - q(x_{j+1/2})|, \quad (28)$$

which is an approximation to the area between the numerical and exact solution profile. Note that E is still a vector in \mathbb{R}^m , error measures may later pick out single components from the solution, e.g., density, or sum them. In addition to observing the numerical errors (28), we have also checked some physical properties of the computed solution, such as conservation and positivity of solution components.

Most problems have homogeneous Neumann boundary conditions, unless stated otherwise. We expand solution values to the two ghost cells on the left and right by copying the first value inside the domain (i.e. $Q_{-3/2} = Q_{-1/2} = Q_{1/2}$ at the left). All experiments keep the CFL number at 0.5 for increased accuracy, although values up to 1.2 did not result in instability yet. We used a Pentium M 1.8 GHz notebook for all experiments.

4.1. MHD shock tube in 1.5D: computational efficiency

One-dimensional shock tube problems are Riemann problems, where an imaginary tube contains plasmas in two different states, separated by a diaphragm. At $t = 0$ the diaphragm opens and the left and right state start to interact. In hydrodynamics, Sod's shock tube is the best known example problem. Here, we consider the classical MHD shock tube in 1.5D, initially described by Brio and Wu [4], which now is widely considered a benchmark problem for MHD simulations.

The problem is set up in the domain $[0,1]$, with the discontinuity at $x=0.5$. We simulate for times $t \in [0,0.1]$. The plasma is initially at rest ($\mathbf{m} = \mathbf{0}$), with $\gamma = 2$ and $\bar{B}_1 = 0.75$. The problem is *co-planar*, i.e., $B_{2,L} = -B_{2,R}$. The difference in density and pressure between the two states is: $\rho_L = 8\rho_R = 1$, and $p_L = 10p_R = 1$. In conservation form, the initial conditions are:

$$\begin{aligned} [\rho, m_1, m_2, B_2, e]_L &= [1, 0, 0, 1, 1.78125] \quad \text{if } x \leq 0.5, \\ [\rho, m_1, m_2, B_2, e]_R &= [0.125, 0, 0, -1, 0.88125] \quad \text{if } x > 0.5, \end{aligned} \tag{29}$$

where subscripts L and R denote the left and right state. Homogeneous Neumann boundary conditions are used for all components.

4.1.1. Numerical results

First, we compare our moving mesh method to the same method with a uniform mesh. Also, a finite volume solver from the VAC package is considered; we use it with a similar TVD Lax–Friedrichs flux and Van Leer flux limiter for a fair comparison. We sum all five components of E_{L_1} to study the overall error. Fig. 3 shows the density and v_1 component of the velocity on the first row. The moving mesh and uniform solutions have $N = 250$ mesh points, and the uniform VAC solution has $N = 1500$ (both take equal running time approximately). The bottom left diagram focuses on the middle three waves. The uniform solution is quite diffusive, a known property of Lax–Friedrichs-type methods. VAC with 1500 points has a much higher resolution, especially at the compound wave. Our moving mesh result is slightly more accurate for all shocks. Its overall error

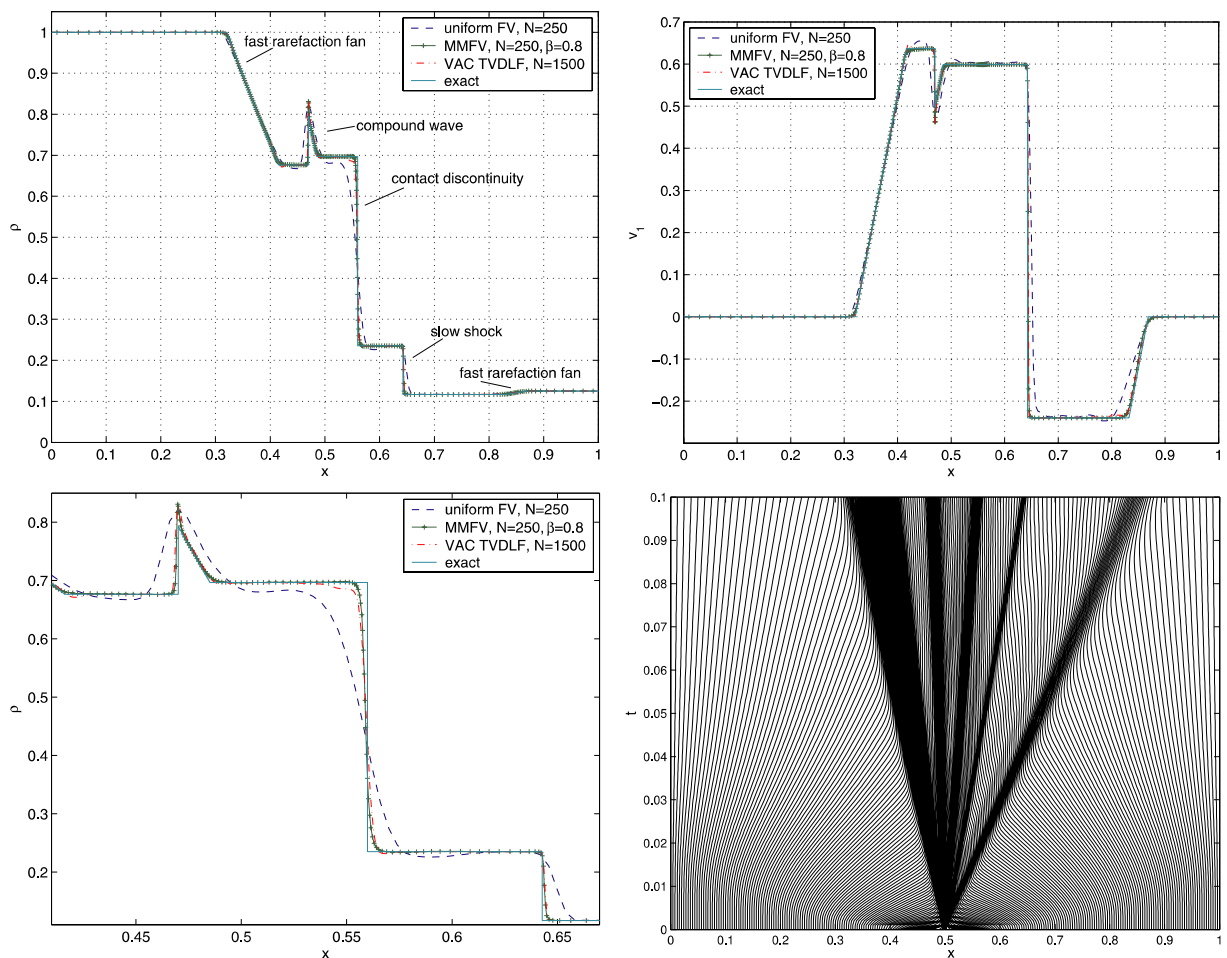


Fig. 3. Solutions to the Brio and Wu problem.

is 9.1×10^{-3} , and the VAC result has an overall error of 1.3×10^{-2} . The top right diagram shows the v_1 component of the velocity. It is very accurate and does not suffer from the dispersive effects observed by Zegeling and Keppens [27, Fig. 2]. Finally, the bottom right diagram shows the mesh movement through time. Note how the rightmost fast rarefaction fan is also properly detected.

4.1.1.1. Computational efficiency. The increased accuracy comes at a price: the mesh movement and conservative interpolation take about 50% of the total running time. The amount of mesh points needed is seriously smaller, though, so the moving mesh method should be more efficient on the whole. To test this, the Brio and Wu problem was solved with $N = 250, 500, 1000, 2000, 4000, 6000, 8000, 12,000, 24,000,$ and $48,000$. The error was determined using (28) and summing all five components, as we are interested in overall accuracy here.

The diagram in Fig. 4 sets out these errors on the horizontal axis and the running time on the vertical axis. The moving mesh method is used with monitor (24), and twice with the arc length-type monitor ($\alpha = 1$ and 10). Since VAC programs are in FORTRAN and our method runs in MATLAB, the VAC timings are normalized using

$$t_{\text{VAC,norm}} = t_{\text{VAC}} \cdot t_{\text{unif},N=N_{\text{norm}}} / t_{\text{VAC},N=N_{\text{norm}}},$$

where we normalized with the $N_{\text{norm}} = 8000$ measurements. The uniform lines for MATLAB and VAC now almost coincide for all N , which justifies the normalization. The arc length runs are never efficient enough to beat uniform solutions. With the advanced monitor function, the moving mesh method becomes a lot more efficient. The gain factor in running time, compared to uniform runs, is approximately 3. A possible improvement could be to adapt the mesh every $k > 1$ time steps. This will reduce the mesh movement and interpolation costs. In their study of efficiency of h-refinement, Keppens et al. have an average of 20% additional costs for mesh adaptation in 1D [10]. Also note how for the same amount of points (e.g., $N = 500$) the obtained accuracy differs (by almost a factor 10 between adaptive and uniform runs). When computer memory is an issue, adaptive methods can still compute accurate solutions with relatively small discrete solution vectors. This becomes a definite advantage in higher-dimensional simulations.

4.1.1.2. r-Refinement vs. h-refinement. In this research we perform mesh adaptation by points movement (r-refinement). An alternative is local mesh refinement (h-refinement), where mesh cells are split into smaller cells or merged again. An adaptive version of the previously used VAC package exists: AMRVAC [10]. It uses

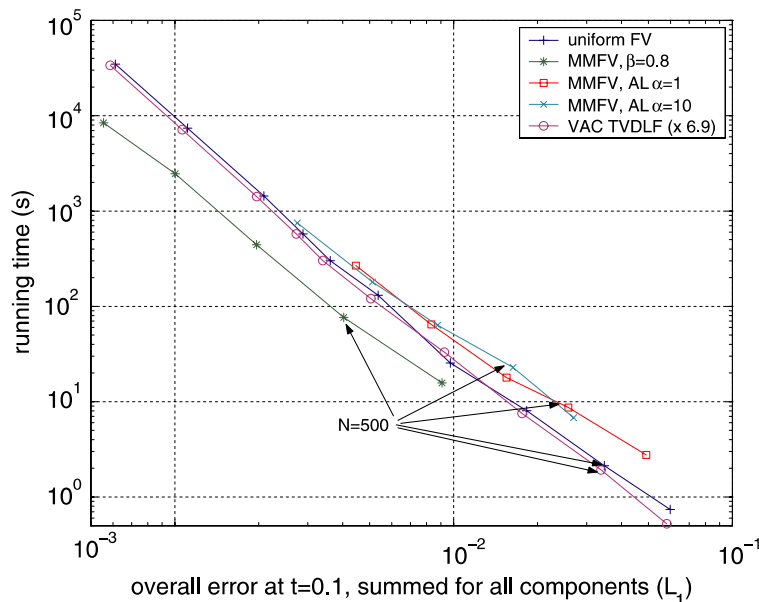


Fig. 4. Computational efficiency for the Brio and Wu problem.

L mesh levels, where level 1 is the initial uniform mesh. We used AMRVAC with a refinement ratio of 2 on each level (i.e. splitting a cell into two equal pieces), hence the maximal mesh refinement is 2^{L-1} .

The advantage of h-refinement is its simplicity. The disadvantage is that the eventual number of mesh points is unknown, which can lead to unexpectedly long running times. Besides, good results require proper knowledge of the parameters by which the user controls refinement (initial mesh size, number of refinement levels, refinement ratios, and the tolerance level for deciding on local refinement). Our method is virtually free of user-defined parameters and is problem independent. For proper choices of refinement levels and tolerance, AMRVAC also produces good results.

We ran the Brio and Wu problem (29) again. Fig. 5 shows the AMRVAC results. The left diagram shows the density, notice how all shocks are represented properly on the maximal refinement level. Also, both rarefaction waves are properly detected and refined. We used $N_{\text{initial}} = 100$ and six mesh levels, with tolerance $\epsilon_{\text{tol}} = 0.002$. All solution components are used equally in the error estimate for deciding on local refinement. The final mesh contains 314 mesh points and has an overall error of 1.4×10^{-2} . Running time is 5.1 s, which would roughly scale to 35 s in MATLAB. Our $N = 250$ result reaches a smaller error in 16 s.

The right diagram in Fig. 5 compares three AMRVAC results with our $N = 250$ result. The smallest mesh cell in our experiment is 13 times smaller than in the original uniform mesh. This can be compared to five refinement levels ($2^{5-1} = 16$). The biggest mesh cells are between two and five times larger than the initial uniform cells. As AMRVAC does not coarsen its initial mesh, we start AMRVAC also with smaller mesh sizes ($N = 50$ and 100). The diagram also shows the final number of points; only for $N = 50$ this is less than 250. Table 1 summarizes the results and lists the overall errors.

AMRVAC contains more powerful methods as well: MUSCL-type solvers that use problem-specific Riemann solvers, and more sophisticated limiters. We used TVDLF with a Van Leer limiter for an equal comparison with our solver.

4.2. MHD shock tube in 1.75D: physical energy loss

The 1.5D Brio and Wu shock tube of the previous section can be extended to 1.75D. Keppens [11] describes a problem where all seven MHD waves show up. The problem is set up in the domain $[0, 1]$, with the discontinuity at $x = 0.35$. We simulate for times $t \in [0, 0.08]$. The plasma has $\gamma = 5/3$ and $B_1 = 1$. In primitive form, the initial conditions are:

$$\begin{aligned} [\rho, v_1, v_2, v_3, B_2, B_3, p]_L &= [0.5, 0, 1, 0.1, 2.5, 0, 0.1] \quad \text{if } x \leq 0.35, \\ [\rho, v_1, v_2, v_3, B_2, B_3, p]_R &= [0.1, 0, 0, 0, 2, 0, 0] \quad \text{if } x > 0.35. \end{aligned} \tag{30}$$

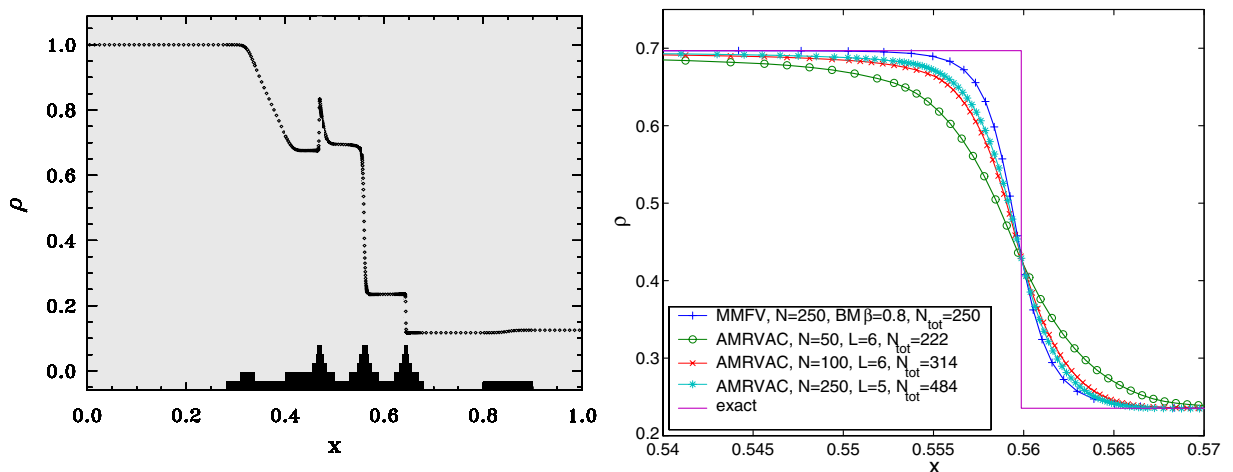


Fig. 5. The Brio and Wu problem solved using h-refinement. Left: AMRVAC solution for the density. Black bars represent mesh refinement levels. Right: Comparison of r-refinement with h-refinement, detail at contact discontinuity.

Table 1
Brio and Wu problem solved with r- and h-refinement

Method	Initial N	L	ϵ_{tot}	Final N	Running time (MATLAB equivalent)	Overall error
MMFV	250	–	–	250	16.5 (16.5)	0.0091
AMRVAC: TVDLF, Van Leer limiter	50	6	0.005	222	2.4 (16.5)	0.0254
	100	6	0.002	314	5.1 (35.2)	0.0135
	250	5	0.0005	484	7.2 (49.7)	0.0102

4.2.1. Numerical results

We use $N = 250$ mesh points again. The left diagram in Fig. 6 shows the density and the v_3 component of the velocity. Note how the Alfvén signals do not change the density. Similarly, the contact discontinuity, the fast rarefaction and the fast shock are not reflected in v_3 . Still, the monitor is based on *all* solution components. Indeed, the right diagram in Fig. 6 shows that the mesh movement captures all seven structures in a balanced way. The seven waves are directly related to the eigen-structure of the 1.75D MHD system, as depicted in Fig. 1.

Throughout rarefaction fans, the entropy $s = c_v \log(p/\rho^\gamma)$ should remain constant. We have verified that this is indeed the case here. Also, for increasing N , the second-order accuracy of the finite volume solver in smooth regions was confirmed.

A final physical check here is the conservation of solution components. Mass conservation is satisfied, but energy conservation is not. Between $t = 0$ and $t = 0.08$, a constant decrease of energy yields a total loss of 0.2. This is exactly right though: at the left boundary the only nonzero part of the energy flux (8) is $\bar{B}_1 B_2 v_2 = 2.5$, whereas at the right boundary it is zero. Integrated over time, this should indeed cause a total energy loss of $0.08 \times 2.5 = 0.2$.

4.3. Regular and critical solutions

We now consider a more general 1.75D shock tube problem described by Torrilhon [19] to investigate multiple possible solutions. The problem is set up in the domain $x \in [-1, 1.5]$ with the discontinuity at $x = 0$. We simulate for times $t \in [0, 0.4]$. In primitive form, the initial conditions are:

$$\begin{aligned} [\rho, v_1, v_2, v_3, B_2, B_3, p]_{\text{L}} &= [1, 0, 0, 0, 1, 0, 1] \quad \text{if } x \leq 0, \\ [\rho, v_1, v_2, v_3, B_2, B_3, p]_{\text{R}} &= [0.2, 0, 0, 0, \cos \theta, \sin \theta, 0.2] \quad \text{if } x > 0. \end{aligned} \quad (31)$$

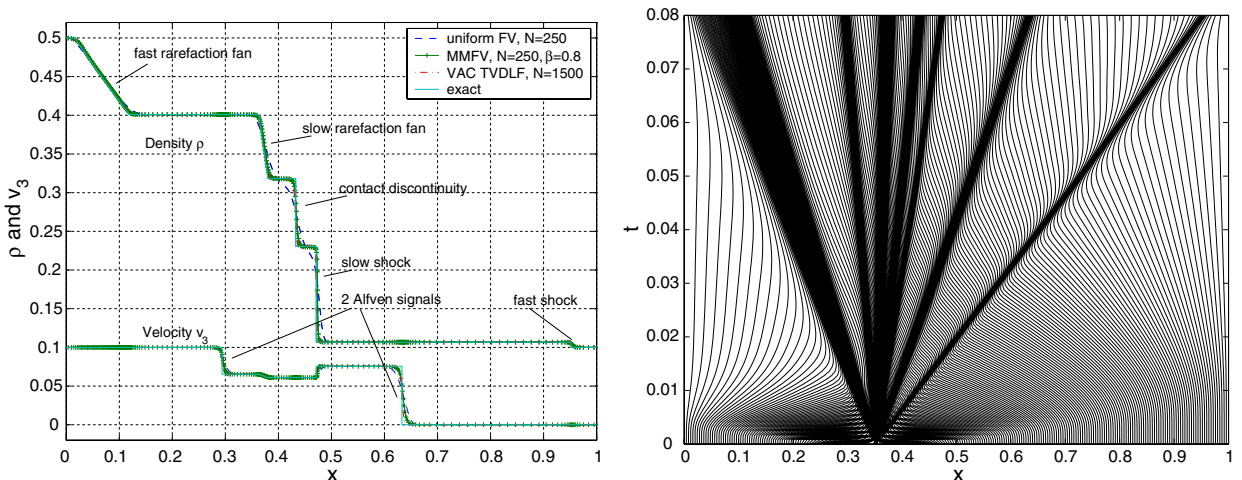


Fig. 6. Solution to Keppens' 1.75D shock tube problem. Left: density and v_3 component of the velocity. Right: mesh history.

The problem is *non-planar* if the angle θ between the transversal parts (i.e. $[B_2, B_3]^T$) of \mathbf{B}_L and \mathbf{B}_R is not a multiple of π . Torrilhon describes how θ affects the possibility of multiple solutions. Regular r -solutions consist only of shocks or contact discontinuities, whereas critical c -solutions can also contain non-regular waves, such as compound waves. For critical choices of θ , both an r - and c -solution are analytically correct simultaneously; $\theta = \pi$ is such a choice. In the Brio and Wu example indeed the irregular compound wave from the c -solution showed up. It depends on the amount of numerical diffusion whether a PDE solver will converge to the r -solution.

4.3.1. Numerical results

We now consider the *almost* co-planar case $\theta = 3$. Analytically, this has only one r -solution. However, the numerical solution is attracted towards the nearby critical solution for $\theta = \pi$. Fig. 7 shows the density and the B_2 component of the magnetic field. The solutions resemble the one to the Brio and Wu problem (a c -solution), but are clearly different from the correct r -solution here.

Increasing the number of mesh points results in smaller mesh cells, hence less numerical diffusion. We study, for increasing N , the convergence of our numerical solution towards the correct r -solution, just as Torrilhon [19, Section 4.2.1] does. Fig. 8 shows the density and the B_2 component of the magnetic field at $[-0.35, -0.1]$ for N up to 2500. The dashed line represents the co-planar c -solution to which the $N = 100$ solution clearly is attracted. For larger values of N , the solutions converge towards the solid black line of the correct r -solution. At $N = 1000$ the solution is about as good as the uniform $N = 20,000$ solution by Torrilhon: a considerable improvement.

4.4. Shear Alfvén waves in 1.5D

This test problem was described by Stone and Norman [13] and also used by Tóth and Odstrčil [21] for their evaluation of different discretization schemes. A homogeneous, uniformly magnetized plasma state is perturbed with a localized velocity pulse transverse ($v_2 := (m_2/\rho) \neq 0$) to the horizontal (x -direction) magnetic field. This evolves into two oppositely traveling Alfvén waves that have associated $v_2 := m_2/\rho$ and B_2 perturbations.

The problem is set up in the domain $x \in [0, 3]$, with the velocity pulse on $x \in [1, 2]$. We simulate for times $t \in [0, 0.8]$. The plasma has an adiabatic constant $\gamma = 1.4$, and $\bar{B}_1 = 1$. In conservation form, the initial conditions are:

$$\begin{aligned} [\rho, m_1, m_2, B_2, e] &= [1, 0, 10^{-3}, 0, 0.5000005025] && \text{for } x \in [1, 2], \\ [\rho, m_1, m_2, B_2, e] &= [1, 0, 0, 0, 0.5000000025] && \text{elsewhere,} \end{aligned} \tag{32}$$

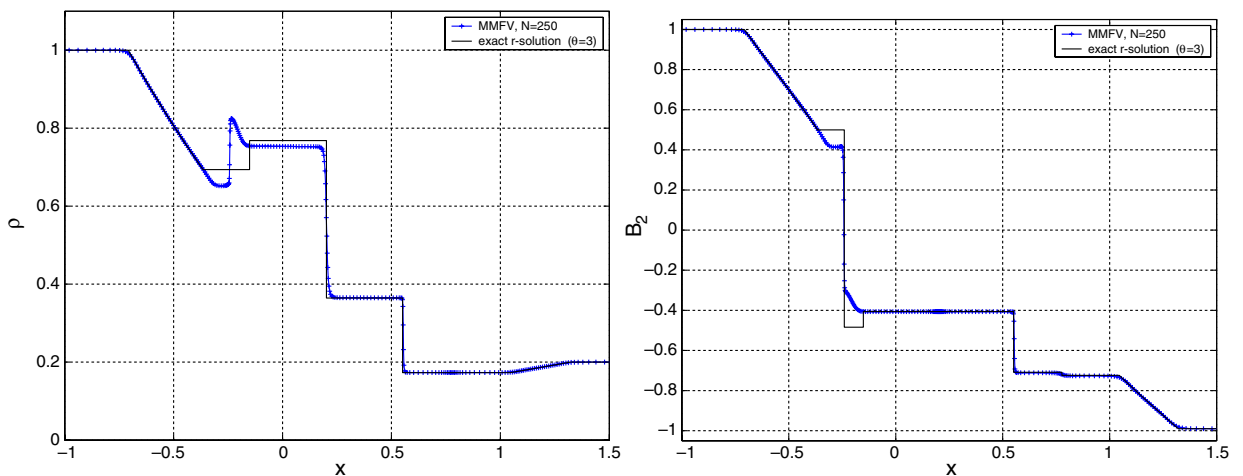


Fig. 7. Solution to the almost co-planar problem (31) with $\theta = 3$. The adaptive solution uses $N = 250$ mesh points, and suffers from pseudo-convergence towards an incorrect c -solution.

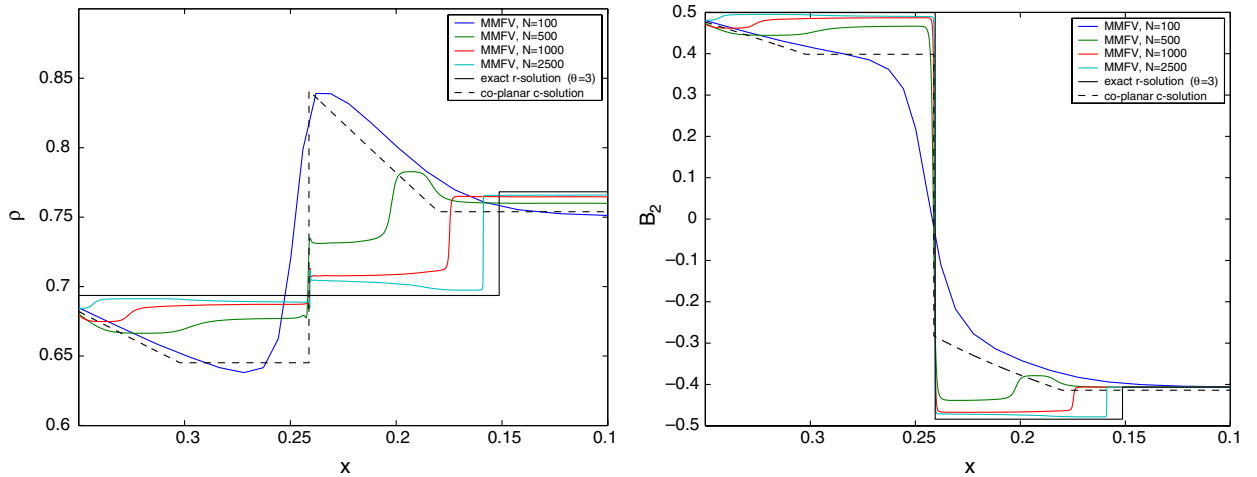


Fig. 8. Convergence to correct solution for the almost co-planar problem ((31), with $\theta = 3$).

where the difference in total energy e is only caused by the difference in v_2 . In primitive form, all quantities but v_2 are constant. Homogeneous Neumann boundary conditions are used for all components.

When considering linear effects, only v_2 and B_2 will be perturbed, and all other primitive quantities should remain constant. Quadratic terms in the flux for m_1 , however, cause nonlinear effects in the density and energy. Furthermore, thermal pressure should always be positive.

4.4.1. Numerical experiments

Fig. 9 shows the B_2 component of the magnetic induction at $t = 0.8$ from both the adaptive mesh solution and the reference solution. In the right diagram, the mesh history is shown. Again, the $N = 250$ adaptive solution compares favorably with the 1000 point VAC solution. Analytic computation of the exact solution is more complicated than with the shock tube problems, because of the interacting right- and left-going waves. An $N = 2500$ adaptive solution shows the sharp profile here.

The mesh history reveals that some intermediate structures were captured too, although those are not in the B_2 (nor v_2) component. A closer look at the almost zero momentum shows levels slightly off from 0. These are caused by the nonlinear terms in the m_1 flux. The left diagram in Fig. 10 shows multiple levels, instead of a constant value of 0. Not only do the physical equations justify these levels; changing the number of mesh points to 100 or 1000 results in the same levels. Furthermore, when changing the initial velocity perturbation from 10^{-3} to 10^{-6} changes the momentum offset from $O(10^{-7})$ to $O(10^{-13})$; clearly a quadratic effect.

The right diagram in Fig. 10 shows the absolute, local errors in the density for the $N = 250$ solution, obtained by subtracting the 2500 points reference solution from it. At $x = 1$ and $x = 2$, local errors are the largest, at 10^{-4} . Elsewhere, errors are very small, $O(10^{-8})$, compared to VAC ($O(10^{-3})$) and the adaptive method of lines ($O(10^{-6})$), cf. Zegeling and Keppens [27, Fig. 4].

4.5. Oscillating plasma sheet in 1.5D: fast wave effects

To investigate the necessity of an implicit solver, Tóth et al. [20] set up a problem that leads to a very strict CFL limit, i.e. very small time steps. A plasma sheet is surrounded by a vacuum which is modeled by a low density, low pressure plasma. At the left and right boundaries are perfectly conducting walls with reflective boundary conditions.

The problem is set up on the domain $x \in [0, 1]$, with the plasma sheet on $x \in [0.45, 0.55]$. We simulate for times $t \in [0, 2]$. The plasma has $\gamma = 1.4$, and $\bar{B}_1 = 0$. In primitive form, the initial conditions are:

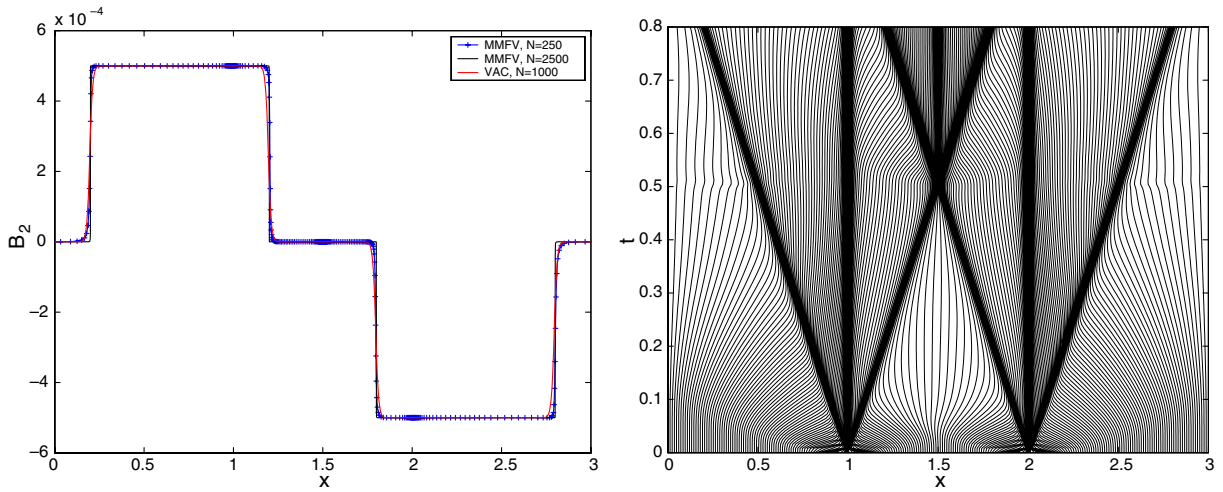


Fig. 9. Moving mesh solution to Shear Alfvén problem at $t = 0.8$, with $N = 250$ mesh points. Left: Transverse component B_2 of magnetic field. Right: mesh history.

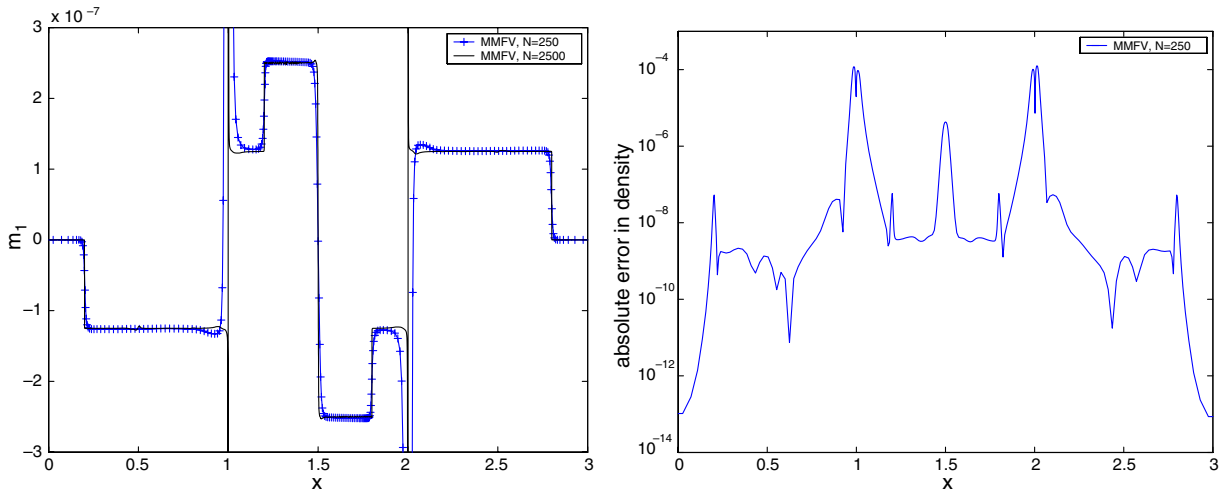


Fig. 10. Nonlinear effects in the Shear Alfvén problem. Left: Nonlinear effects in m_1 . Right: Local errors in density.

$$[\rho, m_1, m_2, B_2, p] = \begin{cases} [10^{-3}, 0, 0, 1.1, 10^{-4}] & \text{for } x \in [0, 0.45], \\ [1, 0, 0, 0.6, 0.3201] & \text{for } x \in [0.45, 0.55], \\ [10^{-3}, 0, 0, 1.0, 10^{-4}] & \text{for } x \in [0.55, 1]. \end{cases} \quad (33)$$

In the plasma sheet, the total pressure $p_{\text{tot}} = p + B^2/2 = 0.5001$ is in balance with the pressure in the ‘vacuum’ at the right, and is about 10% less than in the ‘vacuum’ at the left. Therefore the sheet will start to move rightward until the changing pressure imbalance reverses the movement leftward. Because of conservation of magnetic flux in the left and right ‘vacuum’, this will result in an ongoing oscillation of the sheet.

A reflective wall means zero flux for all components except for the ones orthogonal to the boundary, hence only m_1 is nonzero here. The zero fluxes cannot be obtained by setting the values in the ghost cells outside the domain to zero. As fluxes are computed at cell edges, and solution values are set on cell centers, interpolation will yield slightly nonzero flux values on the boundaries. Instead, we make the m_1 values asymmetric around the two boundaries (i.e. $Q_{-j-1/2} = -Q_{j+1/2}$ at the left, see also Fig. 2), and impose an exactly zero flux for all but the first momentum equation on the two boundaries (i.e. $F_0 = F_N = 0$, except for the second component of

the flux vector F). Now, total mass, magnetic field and energy are conserved numerically up to machine precision.

4.5.1. Numerical experiments

We first look at the slow oscillation that should set in. The oscillating sheet can be approximated by a point mass with total mass $M = 0.1$ at distance $L_0 = 0.5$ from the walls with some equilibrium value B_0 for the magnetic field. The point mass oscillates around this equilibrium, driven by the difference in magnetic pressure between the left and right half. By conservation of magnetic flux (3), the total magnetic flux in the equilibrium and at an extreme position are equal:

$$\begin{aligned} B_L(L_0 + \Delta L) &= (B_0 - \Delta B)(L_0 + \Delta L) = B_0 L_0, \\ B_R(L_0 - \Delta L) &= (B_0 + \Delta B)(L_0 - \Delta L) = B_0 L_0. \end{aligned}$$

A linear approximation gives: $\Delta B/B_0 \approx \Delta L/L_0$. Describing the oscillation as $x(t) = L_0 + \Delta L \sin(\omega t)$, and differentiating twice gives $x''(t) = -\Delta L \omega^2 \sin(\omega t)$. Inserting this into $F = Mx''$ for the rightmost extremum gives: $-M\Delta L = B_L^2/2 - B_R^2/2 = -2B_0^2/L_0 \Delta L$. The oscillation is now characterized by its frequency and amplitude:

$$\omega \approx \sqrt{\frac{2B_0^2}{ML_0}}, \quad \text{and} \quad \Delta L \approx (\Delta B/B_0)L_0.$$

We estimate $B_0 \approx 0.5 \cdot 1.1 + 0.5 \cdot 1 = 1.05$ and $\Delta B \approx 0.1$. This yields $\omega \approx 6.64$, i.e. the period $T \approx 0.946$. The maximum of the total momentum Mv_1 is $M\omega\Delta L \approx 0.158$. Our numerical experiments yield a period of $T = 0.942$ and a momentum amplitude of 0.15. This is quite accurate, considering the simplistic approximation sketched above.

A simulation up to $t = 2$ with $N = 250$ mesh points takes about 25,000 time steps, and only 220 s to run, with the CFL number limited to 0.5. The right diagram in Fig. 11 clearly shows how the adaptive mesh captures the oscillation. The left diagram shows the solution profiles of the density ρ and total pressure p_{tot} at $t = 1$. The oscillation is driven by the imbalance in magnetic (and hence total) pressure. In the diagram the sheet is moving rightward, because of high pressure at the left. The solution profile is much less diffused than in the results by Tóth et al. [20, Fig. 3] and Zegeeling and Keppens [27, Fig. 5].

We now focus on fast waves in the solution and simulate for early times $t \in [0, 0.15]$. The right diagram in Fig. 12 shows the mesh history in more detail for early times. Within the sheet, additional waves are tracked repeatedly. They are initiated by a wave that continuously moves through the ‘vacuum’ between the left wall and the left edge of the sheet; it touches the sheet for the first time at $t \approx 0.026$. The left diagram again shows the density and total pressure, for $t = 0.1$. Both show ‘physical staircasing’ on top of their profile, initiated by three touches of the fast wave. Notice that a similar fast wave moves through the ‘vacuum’ at the right. The wave is less strong and hence causes hardly any ‘staircasing’ at first.

To study the formation of the ‘physical staircase’, Fig. 13 shows four snapshots in time of the density profile. The top row shows the left ‘vacuum’ part and the bottom row shows the high density plasma sheet. Not the entire plasma sheet starts to oscillate at once: first only the left edge of the sheet slowly moves rightward. This leads to an increased density shock on top of the sheet that moves towards the right edge. The bottom diagrams show this expanding shock wave. Only when it touches the right edge, the entire sheet is in oscillation (not shown). In the meantime, other movement takes place as well. As the diagrams in the top row of Fig. 13 show, a fast magnetosonic wave moves between the left wall and the left edge of the plasma sheet. The wave is reflected on both sides because of the reflective wall and the high density in the plasma sheet. The fast wave speed should be equal to $v_1 + c_f$, with c_f as defined in (9). As $v_1 = 0$ here, this yields $c_f \approx 34.8$ in the ‘vacuum’ at the left, and $c_f \approx 31.6$ in the ‘vacuum’ at the right. The wave speed at the left in our numerical simulation is equal to 34.85, which is very accurate. The first staircase formation is about to occur in the third column of Fig. 13: the fast wave will soon touch the sheet edge. In the fourth column, the fast wave has almost completed its second period, and in the meantime the first step in the staircase has properly formed. This process will continue forever, although left-moving waves will start to interact after $t \approx 0.11$. We stress that the observed staircasing is

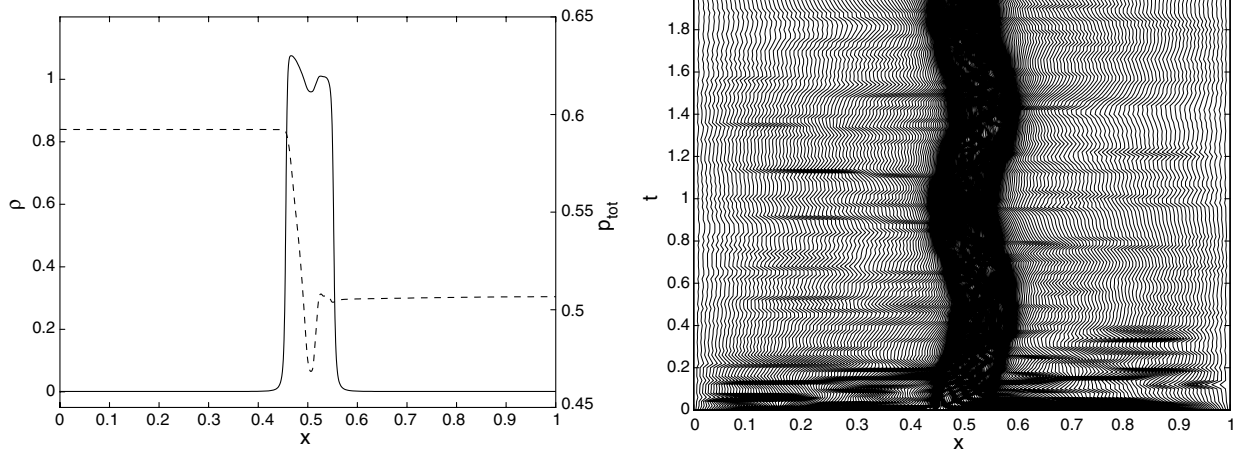


Fig. 11. Oscillating plasma sheet. Left: density (solid line) and total pressure (dashed line) at $t = 1$. Right: mesh history over $t \in [0, 2]$.

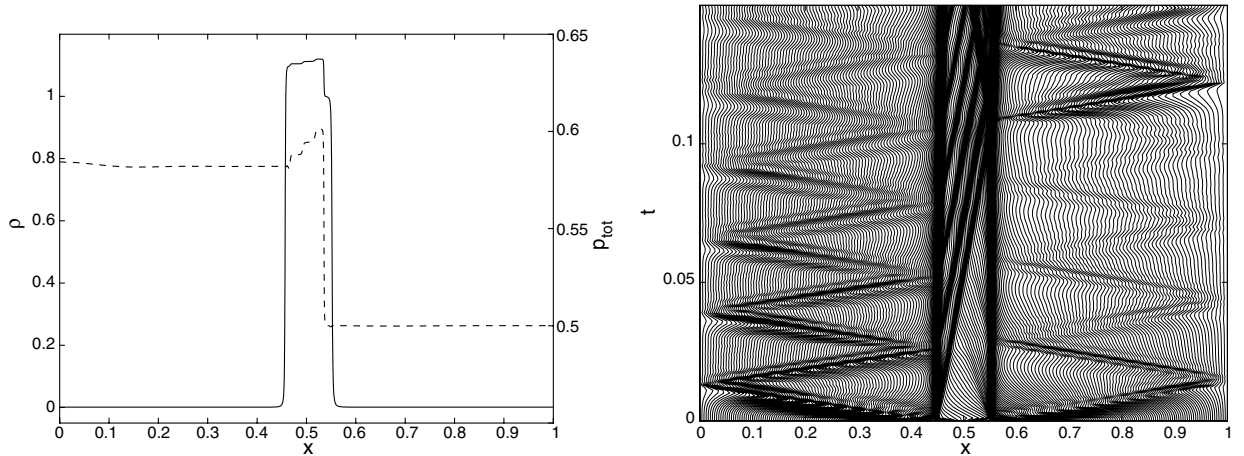


Fig. 12. Oscillating plasma sheet, initial details. Left: density (solid line) and total pressure (dashed line) at $t = 0.1$. Right: mesh history over $t \in [0, 0.15]$.

definitely physical and should not be confused with numerical staircasing sometimes seen in finite volume methods. The recurring interactions between the fast wave and the plasma sheet are in fact repeated, distinct shock tube problems which change density and momentum levels in steps. Local shock tube experiments near the plasma’s edge have also confirmed this.

Both Tóth et al. [20] and Zegeling and Keppens [27] have not shown the above fast wave effects. A probable explanation is their use of implicit time solvers, which take too big time steps to properly capture the fast waves. We also tested an explicit AMRVAC solution. Having seen that the staircasing is mainly visible in the m_1 component of the momentum, we base the refinement on m_1 by 80% and on the density by 20%. Again we use the TVDLF solver with a Van Leer limiter. The initial mesh has $N = 100$. The refinement tolerance ϵ_{tol} had to be lowered to 0.0005. Fig. 14 shows the results. The left diagram shows the AMRVAC solution for the density. Notice how the refinement has properly detected the fast magnetosonic wave in the left vacuum. The right diagram focuses on the staircase formation in m_1 within the plasma sheet. It compares AMRVAC and our MMFV result. AMRVAC seems more diffused, and the refinement could be better at the stair steps. Running time was 37 s (FORTRAN), our MMFV run took 36 s (MATLAB).

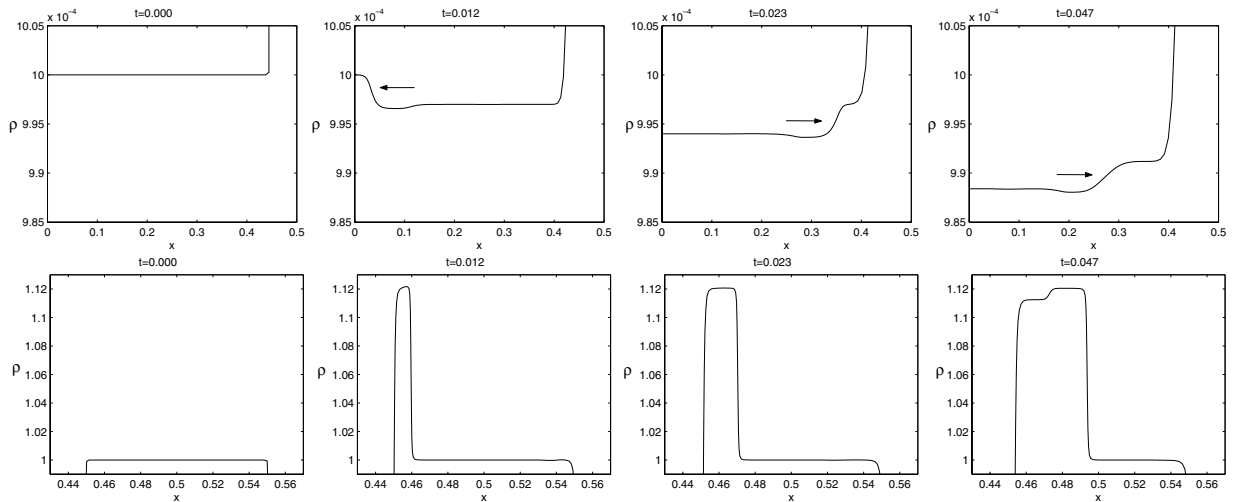


Fig. 13. Oscillating plasma sheet, physical staircasing. Density profiles at $t = 0$, $t = 0.012$, $t = 0.023$, and $t = 0.047$. Top row: movement of a fast magnetosonic wave through the left ‘vacuum’. Bottom row: staircase formation in the high density sheet.

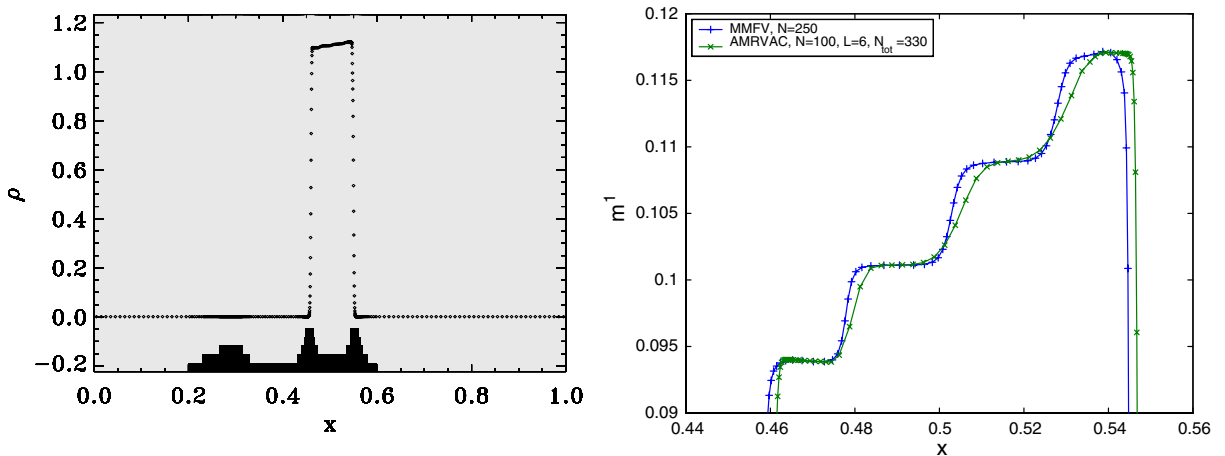


Fig. 14. AMRVAC results for the staircasing effect in the oscillating plasma sheet problem. Left: AMRVAC solution for the density. Black bars represent mesh refinement levels. Right: Comparison of r-refinement with h-refinement, detail of m_1 component of the moment in the plasma sheet.

5. Conclusions

Adaptive methods for solving PDE systems are a commonly used technique to increase numerical accuracy and save computing costs. Often, the adaptive methods are manually finetuned for the specific problem under consideration. A truly robust adaptive method should adapt itself to each new problem considered, without additional fine-tuning. In this paper, we considered such a method. Using a sophisticated monitor function, conservative solution interpolation and a robust finite volume solver, the method is suitable for any nonlinear system of hyperbolic PDEs based on conservation laws, where numerical conservation is guaranteed. After earlier successful application to hyperbolic traffic flow PDEs and problems from gas dynamics, we now used the method on a selection of problems from MHD.

Each of the example problems has one or more interesting physical features that were accurately tracked by the adaptive method. The 1.75D shock wave problem showed automatic and balanced refinement for all individual solution components, thanks to the monitor function used. The study of regular and critical solutions

showed how nearby critical solutions are a strong attractor for numerical solutions. The use of our adaptive method shows convergence to the correct solution with 20 times fewer mesh points than for a uniform method. The shear Alfvén problem showed correct tracking and propagation of Alfvén waves. Moreover, nonlinear effects in the flux terms were accurately computed, with average errors of $O(10^{-8})$. Finally, the oscillating plasma sheet problem challenged the method because of the severe limit on the time step. Even after a large number of time steps, the important parts in the solution are tracked by our adaptive method. The oscillation that should set in is correctly represented. Moreover, the high speed magnetosonic waves in the two ‘vacuum’ parts turn out to cause a ‘physical staircasing’ in the plasma sheet. Although this effect can be explained from the physical formulas, it had not been studied before. The use of an adaptive method increased the accuracy sufficiently to let these effects show up noticeably in the numerical results.

The Brio and Wu shock tube problem was used to benchmark our adaptive method. The gain with respect to a uniform method is at least a factor three. For two- or higher-dimensional models this gain factor counts exponentially. The overall accuracy of the finite volume method is first order, due to first order accuracy of the method at discontinuities. Focusing on smooth parts, however, correctly shows the second order nature of the method. Also, a short comparison with h-refinement shows that our r-refinement method can reach smaller errors more efficiently.

Although Lax–Friedrichs-type methods are known for their numerical viscosity, the combination of a *local* Lax–Friedrichs flux in combination with a moving mesh yields very accurate results, with still good computational performance. We will extend the use of this robust adaptive technique to higher-dimensional models. The use of higher-order solvers, and a more accurate solution interpolation step during mesh moving, are possible future improvements during that process.

Acknowledgements

The first author performs his research in the project on ‘Adaptive moving mesh methods for higher-dimensional nonlinear hyperbolic conservation laws’, funded by the Netherlands Organisation for Scientific Research (NWO) under Project No. 613.002.055.

The authors thank Rony Keppens at the FOM Institute for Plasma Physics Rijnhuizen, for his valuable help with the physical aspects of ideal MHD and assistance in using the VAC and AMRVAC packages. They are also grateful to Manuel Torrilhon at the Hong Kong University of Science and Technology, for kindly providing exact solutions to the shock tube problems in Section 4. Huazhong Tang at Peking University gave some additional details on the numerical scheme used in [15].

References

- [1] G. Beckett, J.A. Mackenzie, Convergence analysis of finite difference approximations on equidistributed grids to a singularly perturbed boundary value problem, *Appl. Numer. Math.* 35 (October) (2000) 87–109.
- [2] G. Beckett, J.A. Mackenzie, On a uniformly accurate finite difference approximation of a singularly perturbed reaction-diffusion problem using grid equidistribution, *J. Comput. Appl. Math.* 131 (June) (2001) 381–405.
- [3] G. Beckett, J.A. Mackenzie, A. Ramage, D.M. Sloan, On the numerical solution of one-dimensional PDEs using adaptive methods based on equidistribution, *J. Comput. Phys.* 167 (March) (2001) 372–392.
- [4] M. Brio, C.C. Wu, An upwind differencing scheme for the equations of ideal magnetohydrodynamics, *J. Comput. Phys.* 75 (April) (1988) 400–422.
- [5] W. Cao, W. Huang, R.D. Russell, A study of monitor functions for two-dimensional adaptive mesh generation, *SIAM J. Sci. Comput.* 20 (6) (1999) 1978–1994 (electronic).
- [6] W. Huang, Practical aspects of formulation and solution of moving mesh partial differential equations, *J. Comput. Phys.* 171 (2) (2001) 753–775.
- [7] W. Huang, Measuring mesh qualities and application to variational mesh adaptation, *SIAM J. Sci. Comput.* 26 (5) (2005) 1643–1666.
- [8] W. Huang, Y. Ren, R.D. Russell, Moving mesh partial differential equations (MMPDES) based on the equidistribution principle, *SIAM J. Numer. Anal.* 31 (3) (1994) 709–730.
- [9] W. Huang, R.D. Russell, Moving mesh strategy based on a gradient flow equation for two-dimensional problems, *SIAM J. Sci. Comput.* 20 (3) (1999) 998–1015.
- [10] R. Keppens, M. Nool, G. Tóth, J.P. Goedbloed, Adaptive mesh refinement for conservative systems: multi-dimensional efficiency evaluation, *Comput. Phys. Commun.* 153 (July) (2003) 317–339.
- [11] R. Keppens, Nonlinear magnetohydrodynamics: numerical concepts, *Fusion Sci. Technol.* 45(2T) (March) (2004) 107–114.

- [12] J.M. Stockie, J.A. Mackenzie, R.D. Russell, A moving mesh method for one-dimensional hyperbolic conservation laws, *SIAM J. Sci. Comput.* 22 (5) (2000) 1791–1813 (electronic).
- [13] J.M. Stone, M.L. Norman, ZEUS-2D: a radiation magnetohydrodynamics code for astrophysical flows in two space dimensions. II. The magnetohydrodynamic algorithms and tests, *Astrophys. J. Suppl.* 80 (June) (1992) 791–818.
- [14] G. Tóth, A general code for modeling MHD flows on parallel computers: versatile advection code, *Astrophys. Lett. Commun.* 34 (1996) 245+.
- [15] H. Tang, T. Tang, Adaptive mesh methods for one- and two dimensional hyperbolic conservation laws, *SIAM J. Numer. Anal.* 41 (2) (2003) 487–515 (electronic).
- [16] T. Tang, Moving mesh methods for computational fluid dynamics, in: Z.-C. Shi, Z. Chen, T. Tang, D. Yu (Eds.), *Recent Advances in Adaptive Computation*, Contemporary Mathematics, vol. 383, American Mathematical Society, 2005, pp. 185–218.
- [17] J.F. Thompson, Z.U.A. Warsi, C. Wayne Mastin, *Numerical Grid Generation: Foundations and Applications*, Elsevier North-Holland, Inc., New York, NY, USA, 1985.
- [18] M. Torrilhon, Exact solver and uniqueness conditions for Riemann problems of ideal magnetohydrodynamics. Research Report 2002-06, Eidgenössische Technische Hochschule, Seminar für Angewandte Mathematik, Zürich, April, 2002.
- [19] M. Torrilhon, Non-uniform convergence of finite volume schemes for riemann problems of ideal magnetohydrodynamics, *J. Comput. Phys.* 192 (November) (2003) 73–94.
- [20] G. Tóth, R. Keppens, M.A. Botchev, Implicit and semi-implicit schemes in the versatile advection code: numerical tests, *Astron. Astroph.* 332 (April) (1998) 1159–1170.
- [21] G. Tóth, D. Odstrčil, Comparison of some flux corrected transport and total variation diminishing numerical schemes for hydrodynamic and magnetohydrodynamic problems, *J. Comput. Phys.* 128 (October) (1996) 82–100.
- [22] Arthur van Dam, A moving mesh finite volume solver for macroscopic traffic flow models, Master's thesis, Utrecht University, May, 2002.
- [23] B. Van Leer, Towards the ultimate conservative difference scheme III. Upstream-centered finite-difference schemes for ideal compressible flow, *J. Comput. Phys.* 23 (March) (1977) 263–275.
- [24] B. Van Leer, Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection, *J. Comput. Phys.* 23 (March) (1977) 276–299.
- [25] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, *J. Comput. Phys.* 32 (July) (1979) 101–136.
- [26] P.A. Zegeling, W.D. de Boer, H.Z. Tang, Robust and efficient adaptive moving mesh solution of the 2-D euler equations, in: Z.-C. Shi, Z. Chen, T. Tang, D. Yu (Eds.), *Recent Advances in Adaptive Computation*, Contemporary Mathematics, vol. 383, American Mathematical Society, 2005, pp. 419–430.
- [27] P.A. Zegeling, R. Keppens, Adaptive method of lines for magnetohydrodynamic PDE models, in: A. Vande Wouwer, Ph. Saucez, W.E. Schiesser (Eds.), *Adaptive Method of Lines*, Chapman & Hall/CRC Press, 2001, pp. 117–137.