



Adaptive two- and three-dimensional multiresolution computations of resistive magnetohydrodynamics

Anna Karina Fontes Gomes¹ · Margarete Oliveira Domingues² ·
Odim Mendes² · Kai Schneider³

Received: 29 February 2020 / Accepted: 14 January 2021 / Published online: 23 February 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

Fully adaptive computations of the resistive magnetohydrodynamic (MHD) equations are presented in two and three space dimensions using a finite volume discretization on locally refined dyadic grids. Divergence cleaning is used to control the incompressibility constraint of the magnetic field. For automatic grid adaptation a cell-averaged multiresolution analysis is applied which guarantees the precision of the adaptive computations, while reducing CPU time and memory requirements. Implementation issues of the open source code CARMEN-MHD are discussed. To illustrate its precision and efficiency different benchmark computations including shock-cloud interaction and magnetic reconnection are presented.

Keywords Magnetohydrodynamics · Numerical simulation · Adaptive grids · Cell-average multiresolution analysis · Divergence cleaning

Mathematics Subject Classification (2010) 65M08 · 65M50 · 65T60 · 76W05

Communicated by: Silas Alben

✉ Anna Karina Fontes Gomes
anna.gomes@ifsp.edu.br

Margarete Oliveira Domingues
margarete.domingues@inpe.br

Odim Mendes
odim.mendes@inpe.br

Kai Schneider
kai.schneider@univ-amu.fr

¹ Federal Institute of São Paulo, Cubatão, São Paulo, Brazil

² National Institute for Space Research, São José dos Campos, São Paulo, Brazil

³ Aix-Marseille Université, CNRS, Centrale Marseille, I2M, Marseille, France

1 Introduction

The constant need for understanding the nonlinear dynamics of different phenomena encountered in our daily life, which are typically governed by nonlinear partial differential equations (PDEs), calls for robust and efficient numerical methods to perform high fidelity numerical simulations. Many complex processes, necessitate high resolution computations to represent efficiently the dynamics of a given multiscale problem [25]. Increasing the resolution of the computational mesh directly impacts on the computational cost, which thus increases at best linearly, and thus can lead in many cases to computationally prohibitive simulations. In this context, dynamically adaptive multiscale methods play a prominent and important role, since their purpose is to adapt the computational mesh to the local structures present in the numerical solution, while preserving the accuracy of the adaptive computations. In particular, many phenomena in space physics can benefit from such adaptive approaches due to their intrinsic multiscale characteristics. The presence of multiple time and space scales appears to be optimal for adaptive methodologies and highly compressed data representation (see, e.g., [36]).

Different adaptive discretization schemes for magnetohydrodynamic (MHD) simulations have been proposed in the literature, an exhaustive review is beyond the scope of the present work. In the following we briefly describe related and competitive adaptive MHD approaches. In [16] a space-time adaptive method using high order discontinuous Galerkin discretizations (ADER-DG) is proposed and among others applied to viscous and resistive MHD in two and three space dimensions. This work is based on the ADER-DG schemes developed for hyperbolic conservation laws [56]. A parallel MHD code, NIRVANA, using adaptive mesh refinement with block-structure and domain decomposition is presented in [57]. A combination of adaptive mesh refinement and central weighted essentially non-oscillatory schemes has been put forward in [35]. The resulting third order accurate scheme has been applied to highly super-Alfvénic plasmas and to stiff Sedov-type explosion problems. A robust second order, shock-capturing numerical scheme for multidimensional special relativistic magnetohydrodynamics can be found in [53], again in the framework of adaptive mesh refinement and using a finite volume setting. Applications to relativistic MHD Riemann problems for which exact solutions are known, are shown to be successfully recovered.

A detailed discussion on space weather forecasting can be found in [51], focusing on a publicly available Space Weather Modeling Framework (SWMF). The foundations are a Block-Adaptive Tree Solarwind Roe-type Upwind Scheme (BATSR-US) code that can solve various forms of the MHD equations, including Hall, semi-relativistic, multi-species and multi-fluid MHD, anisotropic pressure, radiative transport and heat conduction. A block-adaptive mesh in Cartesian and generalized coordinates is used together with load balancing and message passing for one, two and three-dimensional problems. Time-stepping of SWMF can be either explicit, semi-implicit or fully implicit, depending on the application featuring likewise local time-stepping. The current status of MHD simulations for space weather is reviewed in the recent book of Feng [17], including AMR and data driven MHD modeling within the framework of cell-centered finite volume methods.

The review of Jardin [31] discusses the importance of implicit algorithms in the context of magnetically confined fusion plasma using the MHD description. A combination of implicit solvers with highly accurate spatial discretizations and anisotropic thermal conduction is shown to allow predicting accurately fusion experiments for realistic physical parameters in realistic toroidal geometries.

A Lagrangian parallel MHD code, GRADSPMHD, based on the Smooth Particle Hydrodynamics (SPH) formalism is introduced in [55]. A mixed hyperbolic-parabolic correction scheme is used for satisfying the divergence constraint on the magnetic field and a tree-based code for finding the neighbors. For validation classical benchmarks were computed, including the magneto-rotational instability and simulations of magnetized accretion disks. The performance of the code on a parallel supercomputer with distributed memory architecture is likewise assessed.

Meshless finite-volume Lagrangian methods for hydrodynamics have been extended for ideal MHD in [30] using a divergence cleaning scheme. Benchmark computations illustrate that the developed code GIZMO is competitive with adaptive mesh refinement (AMR) techniques. Compared to SPH these methods allow sharp shock-capturing, reduced noise, divergence errors, and diffusion. However the convergence of the method is found to be problem dependent.

Here we present an alternative to AMR, which is meanwhile a standard approach for solving PDEs on adaptive grids [1, 2]. We propose using multiresolution (MR) analysis for introducing adaptivity in MHD simulations. MR is based on the idea that a data set (i.e., the solution of the PDE) can be represented at different refinement levels, according to its local regularity. A detailed comparison of MR and AMR approaches has been carried out in [11, 12] for compressible Euler equations. There we found that the MR method yields a better memory compression than AMR together with improved convergence.

In particular, we focus on the adaptive multiresolution for cell averages, firstly introduced by Harten [26–29] in one dimension, which is directly related to biorthogonal wavelets. The wavelet coefficients provide the information about the regularity of the data, which are used to adapt the computational mesh to the problem of interest. After Harten's seminal work, numerous publications contributed to the development of this approach in a way that the local regularity of the solution is detected [8, 34, 44, 47]. Later, the adaptive MR for cell averages was extended for two and three dimensions [3, 47], making it possible to apply this methodology to different problems of practical interest. In the context of point-value MR an adaptive solver for the two-dimensional compressible Euler equations was proposed in [4]. For a detailed review on adaptive MR and wavelet methods for conservation laws and applications in computational fluid dynamics solving the Navier–Stokes equations, we refer to [15, 44, 50].

In this work, we combine the finite volume method with an adaptive MR approach to solve numerically the resistive magnetohydrodynamic equations, as discussed in [14, 24] for the ideal MHD. Magnetohydrodynamics describes the behavior of a macroscopic electrically conducting fluid, which can be used to model the dynamics of space plasma [19]. The MHD model is characterized by a set of nonlinear

evolutionary partial differential equations, presenting in some cases strong discontinuities of the solution. In order to solve these equations, we use robust numerical schemes that evaluate the numerical fluxes precisely, ensure the stability of the system and keep the physical constraints of the model [10, 42]. In particular, we study here the resistive MHD model, which is more realistic in the context of space physics. The resistivity adds diffusive effects to the system and allows the simulation of physical events such as magnetic reconnection, a phenomenon which happens, e.g., when the interplanetary magnetic field is merging in particular regions with the Earth's magnetosphere.

Our goal is to present the verification of the developed framework, that includes a combination of numerical schemes, and the influence, efficiency and stability of the adaptive MR approach for solving the ideal and resistive magneto-hydrodynamic equations for different problems, describing a variety of physical situations. Our motivation for developing adaptive MHD codes is triggered by space weather applications in science and technology for which there is a social demand. High fidelity real-time space weather predictions including the different involved physical phenomena and the computational cost are still challenging [37, 43].

MHD model simulations in the context of the adaptive MR approach were firstly presented for 1D and 2D Riemann problems [14, 24], in a two-dimensional ideal model, followed by the Kelvin–Helmholtz instability [22]. First results with the three-dimensional implementation were also presented for 1D and 2D Riemann problems in [23] and compared with results obtained with the FLASH code. The CARMEN–MHD code for ideal MHD was used in the AMROC framework [40] and a comparison of the results was performed later in [13], with a wavelet-based adaptive approach. In this work, we present a revised CARMEN–MHD code with new features, which is fully 3D, including resistive terms and which allows the simulation of different problems. Using this new implementation, we present here 2D and 3D resistive and ideal simulations, such as the magnetic reconnection and shock cloud problems. The results are compared with reference solutions.

The remainder of the manuscript is organized as follows. In Section 2, we briefly present the resistive MHD model we adopted in its quasi-conservative form and the MHD variables. The numerical approach, including the adaptive MR for cell averages, the divergence cleaning and the reference solution information, is presented in Section 3. In Section 5, we describe the implementation and the developed open source code in detail. In Section 6 the numerical results are presented and discussed, and comparisons with reference solutions are given. Conclusions are drawn in Section 7.

2 MHD model

We present the MHD model and consider the single fluid description of a plasma, i.e., neglecting the individual identity of each particle of the fluid. We are interested in the quasi-conservative form of the MHD model, which expresses local and global

conservation of mass and momentum, and quasi-conservation of energy density and magnetic flux. The resistive MHD model in its quasi-conservative form is given by

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

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

$$\frac{\partial \mathcal{E}}{\partial t} + \nabla \cdot \left[(\mathcal{E} + p) \mathbf{u} + \mathbf{u} \cdot \left(\frac{|\mathbf{B}|^2}{2} \mathbf{I} - \mathbf{B}^t \mathbf{B} \right) \right] = \nabla \cdot [\mathbf{B} \times \eta (\nabla \times \mathbf{B})], \quad (1c)$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{u}^t \mathbf{B} - \mathbf{B}^t \mathbf{u}) = -\nabla \times (\eta \nabla \times \mathbf{B}), \quad (1d)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (1e)$$

where the magnetic field $\mathbf{B} = (B_x, B_y, B_z)$ is the electromagnetic variable, and the fluid variables are the mass density ρ , pressure p and velocity $\mathbf{u} = (u_x, u_y, u_z)$. Without any loss of generality, we define the magnetic field as $\mathbf{B} = \mathbf{B}/\mu_0$, where μ_0 is the permeability of free space. It is important to note that the MHD variables are normalized. The scalar resistivity is denoted by $\eta = \eta(x, y, z)$, and γ is the adiabatic constant. The energy density \mathcal{E} is given by the constitutive law

$$\mathcal{E} = \frac{p}{\gamma - 1} + \frac{\rho u^2}{2} + \frac{B^2}{2}, \quad (2)$$

and depends thus on the other variables. Equations (1a) and (1b) describe the conservation of mass and momentum, respectively. On the other hand, Equations (1c) and (1d) describe the quasi conservation of energy density and magnetic flux. Equation (1e) is the magnetic field constraint, which ensures \mathbf{B} is divergence free in the continuous setting. In the absence of resistivity, i.e., $\eta = 0$, the model is called the ideal MHD model, which describes the dynamics of a perfectly conducting fluid. In this case, each equation of the system acts as a conservation law, and there is no source term on the right-hand side of the equations.

3 Numerical approach

In this section we present the numerical methods used in this work. We start with the space discretization of the model, which uses a finite volume formulation. We also recall the multiresolution approach used to adapt the computational mesh and the thresholding of the wavelet coefficients. Equation (1e) of the MHD model is not satisfied numerically, since we are considering a discretized version of the problem. In order to fix it, we use a divergence cleaning, which is also presented in this section.

To introduce the numerical approach of the MHD equations, we first rewrite the System (1) in its vector form

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathcal{F}(\mathbf{U}) = \mathcal{S}(\mathbf{U}), \quad (3)$$

where $\mathbf{U} = (\rho, \rho \mathbf{u}, \mathcal{E}, \mathbf{B})$ is the vector of conservative variables, $\mathcal{F} = \mathcal{F}(\mathbf{U})$ the flux tensor and $\mathcal{S} = \mathcal{S}(\mathbf{U})$ the vector of source terms, described in Equation (1).

3.1 Finite volume discretization

The finite volume (FV) method is based on the integral form of conservation laws, which define the rate of change of a quantity in a fixed volume [38]. To this end the 3D computational domain Ω is divided into grid cells of the form

$$C_{i,j,k} \equiv [x_{i-1/2,j,k}, x_{i+1/2,j,k}] \times [y_{i,j-1/2,k}, y_{i,j+1/2,k}] \times [z_{i,j,k-1/2}, z_{i,j,k+1/2}],$$

with $i, j, k \in \{0, \dots, N-1\}$, where N is the number of cells in each direction and $(x_{i,j,k}, y_{i,j,k}, z_{i,j,k})$ is the center of the cell $C_{i,j,k}$. In each cell center we define a corresponding cell average, given by

$$\bar{\mathbf{U}}_{i,j,k} = \frac{1}{|C_{i,j,k}|} \int_{C_{i,j,k}} \mathbf{U}(x, y, z, t) dV, \quad (4)$$

where $\mathbf{U} = \mathbf{U}(x, y, z, t)$ is the vector of the variables, V is the volume of the fluid, $|C_{i,j,k}| = \Delta x \Delta y \Delta z$ is the volume of the cell, with $\Delta x = x_{i+1/2,j,k} - x_{i-1/2,j,k}$, $\Delta y = y_{i,j+1/2,k} - y_{i,j-1/2,k}$ and $\Delta z = z_{i,j,k+1/2} - z_{i,j,k-1/2}$. By integrating Equation (3) over $C_{i,j,k}$, we obtain

$$\int_{C_{i,j,k}} \frac{\partial \mathbf{U}}{\partial t} dV + \int_{C_{i,j,k}} \nabla \cdot \mathbf{F}(\mathbf{U}) dV = \int_{C_{i,j,k}} \mathbf{S}(\mathbf{U}) dV. \quad (5)$$

By multiplying Equation (5) by $\frac{1}{|C_{i,j,k}|}$ and applying the divergence theorem on the divergence operator term, we get

$$\frac{\partial}{\partial t} \bar{\mathbf{U}}_{i,j,k} = -\frac{1}{|C_{i,j,k}|} \int_{\partial C_{i,j,k}} \mathbf{F}(\mathbf{U}) \cdot \mathbf{n}_{i,j,k} dS + \bar{\mathbf{S}}_{i,j,k}(\mathbf{U}), \quad (6)$$

where $\mathbf{n} = \mathbf{n}_{i,j,k}$ is the vector normal to the cell interfaces $C_{i,j,k}$, $\partial C_{i,j,k}$ denotes the boundary of the cell and dS is the surface element of the cell volume. We conclude that the flux tensor must be evaluated on the interfaces of the cell $C_{i,j,k}$, instead of its center. The 2D discrete formulation can be obtained analogously by removing the index k .

3.2 Multiresolution analysis for cell-averages

The multiresolution representation of cell average data is the essential building block to introduce adaptivity and sparse representation of the solution in the finite volume context [29].

To this end, we consider $\mathbf{U} = \mathbf{U}(x, y, z, t)$ an absolutely integrable function on Ω at a given time instant t , and we conceive a hierarchy of dyadic uniform meshes. The center of each cell $C_{i,j,k}^\ell$ is located by $i, j, k \in \{0, \dots, 2^\ell - 1\}$ and its size is defined as $|C_{i,j,k}^\ell| = h 2^{-D\ell}$, where D is the dimension of Ω , ℓ the refinement level and $h = |C_{i,j,k}|$. The total number of cells in each level ℓ is $2^{D\ell}$ cells.

Starting with the idea of nested meshes, i.e., a coarser mesh is contained in the finer one, it is possible to navigate between these meshes to obtain cell average values

of interest. For this procedure, it is necessary to define two operators: projection and prediction. The projection operator is exact and unique, denoted by $\mathcal{P}_{\ell+1}^\ell$, and consists in computing the values on the coarser level ℓ from values on the finer level $\ell + 1$. This evaluation is accomplished from the weighted average of the cell averages in $\ell + 1$, i.e., for three dimensions it is given by

$$\bar{\mathbf{U}}_{i,j,k}^\ell = \left(\mathcal{P}_{\ell+1}^\ell \bar{\mathbf{U}}^{\ell+1} \right)_{i,j,k} = \frac{1}{8} \sum_{m=0}^1 \sum_{p=0}^1 \sum_{q=0}^1 \bar{\mathbf{U}}_{2i+m, 2j+p, 2k+q}^{\ell+1}, \quad (7)$$

with $m, p, q \in \{0, 1\}$. Therefore, each cell average on level ℓ is obtained from eight or four values, according to the number of dimensions of the problem. On the other hand, the procedure that consists in obtaining the cell averages on level $\ell + 1$ from the cell averages on the coarser level ℓ is performed by the prediction operator, denoted by $\mathcal{P}_\ell^{\ell+1}$. Since this operator predicts the cell average values, it is not exact and can assume different definitions. In this work, we choose the approach proposed by Harten [26–28]. Harten's approach has been extended to two dimensions [3] and three dimensions [47, 48]. The set of approximated cell averages is denoted by $\hat{\mathbf{U}}$. Thus, in three dimensions, the approximation is given by

$$\begin{aligned} \hat{\mathbf{U}}_{2i+m, 2j+p, 2k+q}^{\ell+1} &= \left(\mathcal{P}_\ell^{\ell+1} \bar{\mathbf{U}}^\ell \right)_{2i+m, 2j+p, 2k+q} \\ &= \mathcal{I} \left(\bar{\mathbf{U}}^\ell, \ell + 1, 2i + m, 2j + p, 2k + q \right), \end{aligned} \quad (8)$$

where \mathcal{I} is the interpolation operator. The prediction operator approximates eight cell averages on each cell in three dimensions, and four in two dimensions. Moreover, it satisfies the localization property, on which the operator only needs the neighbor values to perform the approximation.

For each interpolation we have an associated error, computed by the difference between the cell average $\bar{\mathbf{U}}^{\ell+1}$ on level $\ell + 1$ and its approximation $\hat{\mathbf{U}}^{\ell+1}$, i.e.,

$$\mathbf{d}^\ell = \bar{\mathbf{U}}^{\ell+1} - \hat{\mathbf{U}}^{\ell+1} \rightarrow \bar{\mathbf{U}}^{\ell+1} = \mathcal{I} \left(\bar{\mathbf{U}}^\ell \right) + \mathbf{d}^\ell, \quad (9)$$

where $\mathbf{d}^\ell = \mathbf{d}_{i,j,k}^\ell$ is the local error, also called *detail* or *wavelet coefficient*. These coefficients, provide information about the local regularity of the numerical solution. The number of local wavelet coefficients for each approximation varies according to the dimension of the problem: three coefficients for two dimensions and seven for three dimensions. Thereby, it is possible to establish a one-to-one correspondence $\bar{\mathbf{U}}^{\ell+1} \longleftrightarrow \{ \bar{\mathbf{U}}^\ell, \mathbf{d}^\ell \}$. The wavelet coefficients along with the cell averages $\bar{\mathbf{U}}_{i,j,k}^\ell$, enable us to obtain the cell averages on level $\ell + 1$ whenever necessary, resulting in the relation

$$\bar{\mathbf{U}}_{2i+m, 2j+p, 2k+q}^{\ell+1} \longleftrightarrow \left\{ \bar{\mathbf{U}}_{i,j,k}^\ell, \mathbf{d}_1^\ell, \mathbf{d}_2^\ell, \dots, \mathbf{d}_7^\ell \right\}, \quad (10)$$

where $m, p, q \in \{0, 1\}$ varies in order to obtain every local cell average on level $\ell + 1$ in three dimensions. By defining the set of every wavelet coefficient obtained on a local approximation on level ℓ as

$$\mathbf{D}^\ell = \left\{ \mathbf{d}_m^\ell, 1 \leq m \leq 2^{D^\ell} - 1 \right\}, \quad (11)$$

the relation can be generalized to the entire mesh, reaching a one-to-one correspondence between $\bar{\mathbf{U}}^{\ell+1}$ and $\{\mathbf{D}^\ell, \bar{\mathbf{U}}^\ell\}$, given by

$$\bar{\mathbf{U}}^L \longleftrightarrow \{\mathbf{D}^{L-1}, \mathbf{D}^{L-2}, \dots, \mathbf{D}^0, \bar{\mathbf{U}}^0\}, \quad (12)$$

which characterizes the process of the multiresolution transform operator \mathbf{M} , defined as

$$\bar{\mathbf{U}}_{MR} = \mathbf{M} \bar{\mathbf{U}}^L, \quad \bar{\mathbf{U}}^L = \mathbf{M}^{-1} \bar{\mathbf{U}}_{MR}. \quad (13)$$

where $\bar{\mathbf{U}}_{MR} = (\mathbf{D}^{L-1}, \mathbf{D}^{L-2}, \dots, \mathbf{D}^0, \bar{\mathbf{U}}^0)$. The MR transform satisfies the properties of localization, polynomial cancellation and stability [15]. The stability guarantees that small perturbations introduced on transformed data on any scale are not uncontrollably amplified in the iterative applications of the multilevel transform. Therefore, we conclude the information of every cell average on the coarser level and the wavelet coefficients of all levels is equivalent to the information of the cell averages on the most refined level.

The conservation properties of the finite volume method are preserved in our adaptive multiresolution discretization taking special care in the flux evaluation. To ensure the balance of ingoing and outgoing fluxes at the cell interfaces on adjacent refinement levels, we use the conservative formulation proposed in [44, 48]. To guarantee the conservation, the ingoing fluxes at level ℓ are computed as the outgoing fluxes of the corresponding cells at level $\ell + 1$, as illustrated in Fig. 1. This is possible due to the graded-tree structure, which keeps the nearest cousins of a cell or creates a virtual leaf for the flux computations. Thus the flux computation is conservative between cells at different levels of refinement.

Dynamic mesh adaptation: thresholding To define the regions of the computational mesh that need more or less refinement, we apply the threshold operator $\mathcal{T}_{\epsilon^\ell}$ on the wavelet coefficients. For each approximation, this nonlinear operator is defined as

$$\mathcal{T}_{\epsilon^\ell}(\mathbf{d}_m^\ell) = \begin{cases} 0, & \text{if } |\mathbf{d}_m^\ell| \leq \epsilon^\ell, \\ \mathbf{d}_m^\ell, & \text{otherwise,} \end{cases} \quad (14)$$

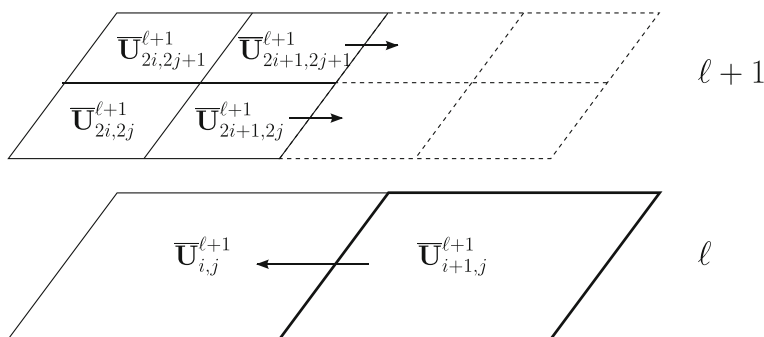


Fig. 1 Conservative flux computation in 2D for two different refinement levels, illustrating ingoing and outgoing fluxes. Adapted from [48]

where ϵ^ℓ is the threshold parameter and $m \in \{1, \dots, 2^{D\ell} - 1\}$. Given the level ℓ , the thresholding consists in removing the cells in which the magnitude of the details is smaller than ϵ^ℓ , replacing them by zero. Thereby, the number of cells required for the numerical simulation can be significantly decreased, impacting the computational cost which correspondingly decreases with this reduction. Hence the computational cost becomes smaller when more wavelet coefficients are removed. In regions where the solution presents smooth behavior, the wavelet coefficients have small magnitude ($|\mathbf{d}^\ell| \leq \epsilon^\ell$), allowing locally coarser meshes. On the other hand, the magnitude of the coefficients is significant ($|\mathbf{d}^\ell| > \epsilon^\ell$) in regions where local structures are present, requiring more refined meshes [26, 27].

The threshold parameter ϵ^ℓ can have either a fixed value or be level dependent. In the former case, we define the value $\epsilon = \epsilon^\ell$ which remains the same during the simulation. In the level dependent case, an initial parameter ϵ^0 is defined and it changes according to the local refinement of the region of interest, given by the equation

$$\epsilon^\ell = \frac{\epsilon^0}{|\Omega|} 2^{D(\ell-L+1)}, \quad 0 \leq \ell \leq L-1, \quad (15)$$

where $|\Omega|$ is the global volume of the computational region and L is the maximum refinement level. Equation (15) is called Harten's strategy to determine the choice of the threshold parameter $\epsilon^\ell = \epsilon(\epsilon^0, \ell)$.

Each detail \mathbf{d}^ℓ is understood as a vector with the details of the MHD conservative variables as its components, i.e., $\mathbf{d}^\ell = (\mathbf{d}_\rho^\ell, \mathbf{d}_{\rho\mathbf{u}}^\ell, \mathbf{d}_\mathcal{E}^\ell, \mathbf{d}_\mathbf{B}^\ell)$. As the MHD variables are stored in the vector $\bar{\mathbf{U}}_{i,j,k}^\ell$, the detail components are computed from the approximation error in each variable and its maximum value. In the scalar-valued approach, the details are computed as $\mathbf{d}_\varpi^\ell = \frac{\mathbf{d}_\varpi^\ell}{\max(|\varpi|)}$, where ϖ denotes a MHD quantity. In this case, each component of the vector variables $\rho\mathbf{u}$ and \mathbf{B} is considered separately to compute \mathbf{d}_ϖ^ℓ , as scalar variables. Thus the vector of details \mathbf{d}^ℓ has in total 8 components.

After performing some numerical experiments, we found the optimal computation of the details of the MHD variables in three dimensions, given by the following,

$$\mathbf{d}_\rho^\ell = \frac{\mathbf{d}_\rho^\ell}{\max |\rho|}, \quad (16)$$

$$\mathbf{d}_{\rho\mathbf{u}}^\ell = \frac{\sqrt{(\mathbf{d}_{\rho u_x}^\ell)^2 + (\mathbf{d}_{\rho u_y}^\ell)^2 + (\mathbf{d}_{\rho u_z}^\ell)^2}}{\max(|\rho u_x|, |\rho u_y|, |\rho u_z|)}, \quad (17)$$

$$\mathbf{d}_\mathcal{E}^\ell = \frac{\mathbf{d}_\mathcal{E}^\ell}{\max |\mathcal{E}|}, \quad (18)$$

$$\mathbf{d}_\mathbf{B}^\ell = \frac{\sqrt{(\mathbf{d}_{B_x}^\ell)^2 + (\mathbf{d}_{B_y}^\ell)^2 + (\mathbf{d}_{B_z}^\ell)^2}}{\max(|B_x|, |B_y|, |B_z|)}. \quad (19)$$

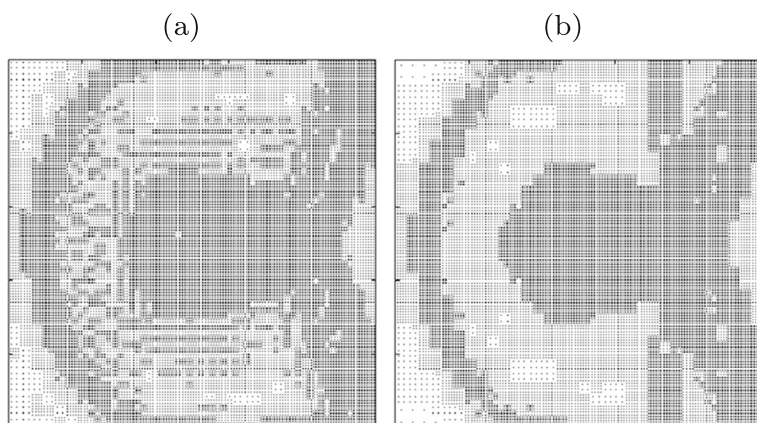


Fig. 2 2D MHD shock-cloud problem: adaptive meshes obtained at the final time of simulation with **a** scalar-value and **b** vector-based threshold approaches

In two dimensions we have a special treatment for the z -components, namely

$$\mathbf{d}_{\rho\mathbf{u}}^\ell = \frac{\sqrt{(\mathbf{d}_{\rho u_x}^\ell)^2 + (\mathbf{d}_{\rho u_y}^\ell)^2}}{\max(|\rho u_x|, |\rho u_y|)}, \quad (20)$$

$$\mathbf{d}_{\rho u_z}^\ell = \frac{\mathbf{d}_{\rho u_z}^\ell}{\max |\rho u_z|}, \quad (21)$$

$$\mathbf{d}_{\mathbf{B}}^\ell = \frac{\sqrt{(\mathbf{d}_{B_x}^\ell)^2 + (\mathbf{d}_{B_y}^\ell)^2}}{\max(|B_x|, |B_y|)}, \quad (22)$$

$$\mathbf{d}_{B_z}^\ell = \frac{\mathbf{d}_{B_z}^\ell}{\max |B_z|}, \quad (23)$$

which is called vector-based approach. Therefore, for vector-valued variables $\rho\mathbf{u}$ and \mathbf{B} , the associated wavelet coefficients take into account each component on the computation. In two dimensions, only two components are used and the third one is computed in a independent way. We found that this vector-based approach optimizes the local mesh refinement, compared to the scalar-valued approach, where we compute the details for each vector component individually [21].

The adaptive mesh is organized into a tree data structure where the different levels define the resolution, which are represented in the tree hierarchically. Following the tree nomenclature, the children of a given cell $C = C^\ell$ are the cells descendent of C in level $\ell - 1$. The neighborhood of a cell at the same level corresponds to their brothers and in the upper level to their uncles. The adaptive mesh must follow a graded structure as discussed in [5]. This contributes to maintain the accuracy and stability in the time evolution with the inclusion of extra neighbors and uncles cells in the neighborhood of the selected cells. We also add virtual leaves to avoid unnecessary computation. Virtual leaves are not evolved in time. More details can be found

in [48]. The following Algorithm 1 describes the procedure to construct the adaptive mesh at $t = t_0$. The adaptive mesh is then updated at each time step.

Algorithm 1 Adaptive mesh construction.

Require: Consider the adaptive mesh, containing all cells at level L

Require: Threshold parameter ϵ^ℓ

Obtain the set of details \mathbf{D}^ℓ of every level ℓ

for $\ell = L - 1$; $\ell \geq \ell_{\min}$; $\ell = \ell - 1$ **do**

for Every cell C at level ℓ **do**

if Every child of C is in the adaptive mesh **then**

if The significant details are kept, *i.e.* the detail corresponds to cell C is smaller than ϵ **then**

if The refinement level of every cell adjacent to the set of cells to be merged is greater than ℓ **then**

 Remove the child of C from the adaptive mesh

 Insert C in the adaptive mesh

 Check if the tree is graded, if not impose it.

 Add virtual leaves.

end if

end if

end if

end for

end for

Refine every cell at a level $\ell < L$;

In Fig. 2, we present two adaptive meshes obtained at the final time for 2D shock-cloud simulations for the same ϵ using either the (a) scalar-valued or (b) a vector-based threshold approach. By comparing the adaptive meshes with the solutions of the variables presented in Fig. 11, computed with the vector-based approach on a adaptive mesh, we can observe that the vector-based approach is more efficient to capture the structures of the numerical solution, avoiding unnecessary refinement in smoother regions of the solution.

To apply the threshold operator to previously defined values, we define the maximum value among the details, *i.e.*, $\max |\mathbf{d}^\ell| = \max \{|\mathbf{d}_\rho^\ell|, |\mathbf{d}_{\rho\mathbf{u}}^\ell|, |\mathbf{d}_E^\ell|, |\mathbf{d}_B^\ell|\}$. Thus, after this procedure, the threshold operator is applied to the details of the MHD variables. In this way, the adaptive mesh becomes the union of the mesh of each variable, since the wavelet coefficients of each variable are used to decide in which local region a more refined mesh is necessary.

3.3 Divergence of the magnetic field correction

Gauss' law of magnetism, given by Equation (1e), imposes a physical constraint on the magnetic field, which is satisfied in the continuous medium. To ensure the absence of nonphysical behavior in the numerical MHD solution, we add to the MHD system the so-called parabolic-hyperbolic divergence cleaning [10, 41], which does

not impose a vanishing divergence of the magnetic field, but damps and propagates the associated numerical divergence errors. In this case, we add a new scalar variable ψ to the MHD System (1), thus modifying Equation (1d) and adding a transport equation for ψ ,

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{u}^t \mathbf{B} - \mathbf{B}^t \mathbf{u}) + \nabla \psi = -\nabla \times (\eta \nabla \times \mathbf{B}), \quad (24)$$

$$\frac{\partial \psi}{\partial t} + c_h^2 \nabla \cdot \mathbf{B} = -\frac{c_h^2}{c_p^2} \psi, \quad (25)$$

where c_p and c_h are the parabolic and hyperbolic constants, with $c_h > 0$. The complete model is called MHD model with Generalized Lagrange Multipliers (GLM–MHD). It is important to note that the GLM–MHD model is originally proposed for ideal MHD and we use this formulation here too. In the resistive case, we are only considering the additional source terms present Equations (1c) and (1d).

By using appropriated initial and boundary conditions, the MHD system presented is completed and ready for the numerical simulation in two or three space dimensions.

4 Time evolution

The adaptive mesh has to be updated at each time step, because the local structures present in the numerical solution can change at each iteration. By defining the operators $\mathbb{E} : \bar{\mathbf{U}}^n \rightarrow \bar{\mathbf{U}}^{n+1}$ of the time evolution, $\mathcal{T} : \mathbf{d}^{L-1}, \dots, \mathbf{d}^\ell, \bar{\mathbf{U}}^\ell$ to $\bar{\mathbf{U}}^{n*}$ the thresholding operator, $\mathbf{M} : \bar{\mathbf{U}}^L \rightarrow \{\mathbf{d}^{L-1}, \dots, \mathbf{d}^\ell, \bar{\mathbf{U}}^\ell\}$ of the multiresolution transform and $\mathbf{M}^{-1} : \{\mathbf{d}^{L-1}, \dots, \mathbf{d}^\ell, \bar{\mathbf{U}}^\ell\} \rightarrow \bar{\mathbf{U}}^L$ of the inverse multiresolution transform, the adaptive MR process can be described as follows

$$\bar{\mathbf{U}}_{MR}^n = \mathbf{M}(\bar{\mathbf{U}}^n) \quad (26)$$

$$\bar{\mathbf{U}}_{MR}^{n*} = \mathcal{T}(\bar{\mathbf{U}}_{MR}^n), \quad (27)$$

$$\bar{\mathbf{U}}_{MR}^{n+1} = \mathbb{E}(\bar{\mathbf{U}}_{MR}^{n*}), \quad (28)$$

$$\bar{\mathbf{U}}^{n+1} = \mathbf{M}^{-1}(\bar{\mathbf{U}}_{MR}^{n+1}). \quad (29)$$

This process indicates that, after the MR representation of the solution $\bar{\mathbf{U}}^n$, we apply the thresholding operator \mathcal{T} and obtain the cell averages at the intermediate step n^* . Then, these values are evolved to the time t^{n+1} . To finish the process, the inverse MR operator is performed [48].

5 Implementation issues and the CARMEN–MHD code

The CARMEN–MHD code is based on the CARMEN code, originally developed by O. Roussel during his PhD thesis [47, 48] using finite volumes together with adaptive MR for cell-averages. This C^{++} code with tree-data structures was implemented to

simulate reaction-diffusion equations modeling combustion problems and later also extended for the compressible Euler and Navier–Stokes equations.

The implementation of the CARMEN–MHD code started with the ideal 2D MHD equations, with HLL and HLLD numerical fluxes and GLM divergence cleaning [14, 20]. After some adjustments, including an eigenvalue fix, TVD limiters for the conservative variables and improvement of the boundary conditions [22, 24], we started the implementation of the 3D MHD equations, first for 2.5D simulations [23]. The uniform mesh MHD implementation in 2D and 3D, which allowed the CPU time comparison, resistive terms, artificial diffusion terms, fixed time steps and more, were implemented later [21]. The code became more robust and different types of MHD simulations could be done properly.

The adaptive MR algorithm creates a computational mesh which becomes more refined in regions where local structures are present. The mesh refinement reduces the cost of the numerical flux computation significantly, which normally requires the major memory percentage. The numerical simulation of the ideal and resistive MHD equations is performed with the CARMEN–MHD code.

To compute the flux \mathcal{F} on the cell interface, we firstly reconstruct the conservative variables using a MUSCL-type monotonized central scheme [54] to achieve second order accuracy in space. These reconstructed values are used to compute the intermediate states of the Harten-Lax-Van Leer-Discontinuities Riemann solver [42] and then the numerical flux is evaluated on the cell interfaces. We should note that the FV method is strictly conservative, since the outflux of a volume is imposed to be equal to its influx. The physics of the problem can be reproduced in such way that the conservative principles of the model are sustained.

The current CARMEN–MHD code is able to solve ideal and resistive MHD equations, by using the adaptive MR for cell averages or a uniform finite volume computational mesh. The simulations performed are stable and the numerical solution remains coherent and does not show oscillations or non physical behavior. The code and its documentation are available on a repository for download and all the presented problems can be reproduced and visualized properly.¹

In Fig. 3, we present a flowchart illustrating the algorithm of the CARMEN–MHD code. First, the code is initialized with the initial condition (IC) and other parameters. The cell averages are computed and the initial mesh is created. In the second step, the time evolution is performed. In this part, the code evolves the quantities of the MHD model, by evaluating the numerical fluxes, GLM divergence cleaning and evolving in time with a second order Runge–Kutta scheme. The stability of the new solution is checked and the time step of the next iteration is computed. After that, we proceed to the third step, where the adaptive mesh is updated based on the new cell averages, and the mesh is adapted again. To finish the algorithm, the mesh and solution are written into a file. The procedure is repeated until the final time is reached.

As a reference solution for our results, we use the FLASH code,² developed in the *Flash Center* at the University of Chicago. This code includes the implementation

¹waveletapplications.github.io/carmenMHD/

²flash.uchicago.edu/site/flashcode/

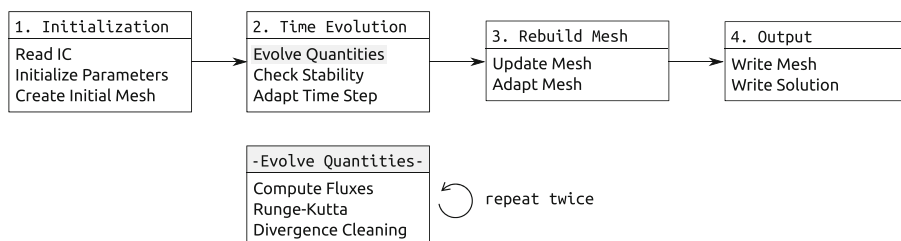


Fig. 3 Flowchart of the CARMEN-MHD code

of the ideal and resistive (constant resistivity) MHD models, as well as the FV discretization. It is possible to perform adaptive simulations by using an adaptive mesh refinement algorithm. However, here we are interested only in the results obtained on a uniform mesh, as the comparison between two adaptive methods is not part of this work. The FLASH code results are used only for the comparison of the solutions and error computations. The version of the code used here is 4.3. For the FLASH code simulations, the following settings are used: one-step Hancock for time evolution, *8-wave* divergence cleaning, MC limiter and HLLD Riemann solver. These settings yield second order of the numerical scheme in time and space.

6 Numerical simulation

In the following we present several test cases to verify the CARMEN-MHD code, and, in particular, the adaptive multiresolution algorithm for the MHD equations. These test cases can assess how our solvers deal with different physical situations, such as, magnetohydrodynamic shocks, local structures and magnetic field lines topology changes, and also numerical challenges, such as stability, strong discontinuities and divergence free correction.

In previous works, we found that the MR approach is efficient to represent the numerical solution 2D and 2.5D for ideal MHD problems [14, 23, 24]. The MR algorithm decreases significantly the number of the cells in the computational mesh and, consequently, the required CPU time. It also provides an accurate solution, compared to the regular mesh solution, demanding much less cells and memory. To verify the solutions of the CARMEN-MHD code, we use the FLASH code [18] as reference solution for our results. The reference solutions obtained with FLASH code are simulated using finite volumes on a regular Cartesian mesh.

6.1 Orszag-Tang vortex

The Orszag-Tang vortex (O-T) in two dimensions [45] is a well-known benchmark for MHD simulations, which allows us to test the transition to two-dimensional supersonic MHD turbulence. Thus, the Orszag-Tang vortex is adequate to verify the robustness of the code when it comes to deal with the formation of magneto-hydrodynamic shocks and shock-shock interactions. It is also interesting to

quantitatively estimate how significant the magnetic monopoles affect the numerical solution, by testing the divergence constraint of the magnetic field. In summary, this problem is a common and classical numerical test for MHD codes and consistent to perform comparisons between codes. This problem presents physical structures over the entire domain, characterizing a challenge to our proposed adaptive multiresolution algorithm. We want to measure the quality of the CARMEN–MHD solution and compare it to the reference. The initial condition for the O-T problem is given in Table 1. The domain interval is defined by $\Omega = [0, 2\pi] \times [0, 2\pi]$, $L = 9$ (512×512) is the most refined level, the final time is $t = \pi$, the Courant number $\nu = 0.4$, and $\alpha_p = 0.4$ and $\gamma = 5/3$. The boundary conditions are periodic in all directions.

Firstly, we present a comparative study of the numerical solution obtained on a regular full mesh, to ensure the reference solution is adequate and to show that CARMEN–MHD results are coherent. To evaluate the local convergence of the solution, we present cuts of the variable p at $y = 0.64\pi$, which is largely used on the literature (see, e.g., [32, 39, 42, 49]). By collecting a set of 200 points of the solution presented in Londrillo and Del Zanna [39], Miyoshi and Kusano [42], along with the FLASH code solution, we can observe in Fig. 4a that these cuts present similar behavior. It also suggests that the reference solution obtained with the FLASH code is adequate to be used as a benchmark for the CARMEN–MHD code solution. The solutions at levels $L = 7, 8$ and 9 obtained with CARMEN–MHD are shown in Fig. 4b. For each resolution level, the solution behaves as expected.

In the adaptive context, the simulations are performed by choosing fixed truncation parameters $\epsilon = 0.01$, $\epsilon = 0.03$ and the level-based one, $\epsilon_0 = 0.02$. In Fig. 5, we show the visualization of the MHD variables p and u_y obtained with $\epsilon = 0.03$ at $t = \pi$. The symmetry of the solution is kept and the physical structures are well represented over the computational domain. The one dimensional cuts for p at $x = \pi$ and $y = \pi$, including the reference solution, are presented in Fig. 6. We can notice that the similarity between the solutions, and observe the convergence towards the reference thus reproducing the expected physical behavior. In particular, the total pressure is a macroscopic entity that is the result of the environment variables \mathbf{B} , \mathbf{u} and ρ , which makes it a suitable numerical sensor in studies such as instabilities, environment morphology, and reconnection phenomena.

The optimal threshold parameter found for this problem is $\epsilon = 0.03$, because it provides a significant economy of cells and CPU time. Moreover, it also maintains the error in the same order of accuracy when compared to the other adaptive cases, as presented in Fig. 8 for the density variable. For $\epsilon = 0.03$, only 26% of the cells over time are needed for the simulation, decreasing the CPU time by 77%. The percentage of cells required for $\epsilon = 0.01$ and $\epsilon_0 = 0.2$ simulations are, respectively, 45% and 85%, leading to a 56% and 11% CPU time reduction. The adaptive meshes for $\epsilon = 0.01$ and $\epsilon = 0.03$ at $t = \pi$ are shown in Fig. 7. In both cases, the cells are

Table 1 Initial condition of the Orszag–Tang vortex

ρ	p	u_x	u_y	u_z	B_x	B_y	B_z
γ^2	γ	$-\sin y$	$\sin x$	0.0	$-\sin y$	$\sin 2x$	0.0

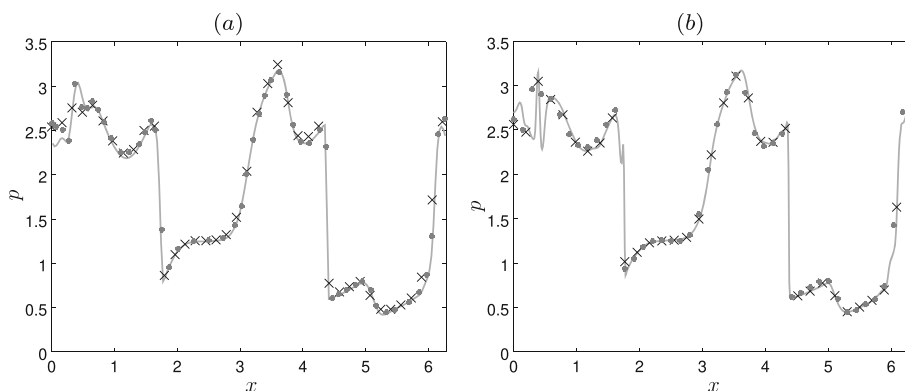


Fig. 4 Cuts in variable p on a uniform mesh: **a** comparison between reference solution (solid line), Londrillo and Del Zanna 2000 (cross) and Miyoshi and Kusano 2005 (dot); **b** CARMEN-MHD code solution obtained for $L = 7$ (cross), $L = 8$ (dot) and $L = 9$ (solid line)

located in regions that present stronger discontinuities, according to the visualization of the variables provided, allowing the structures to be well represented even in the case with less cells.

The \mathcal{L}^1 and \mathcal{L}^2 errors for the uniform and adaptive simulations are shown in Table 2 for p and u_y . The uniform mesh errors are slightly smaller, which is expected since the number of cells is significantly larger. In Fig. 8, we show that as we increase the ϵ value, the error values also increase. We are comparing the uniform mesh, denoted by $\epsilon = 0$, and two adaptive cases, $\epsilon = 0.01$ and $\epsilon = 0.03$. It is important to notice here that the ϵ value is also related to the number of cells, i.e., the number of cells tends to decrease as we increase ϵ . Thus having more cells implies smaller error values. This shows that in the adaptive simulations we should find the optimal relation between computational gain and accuracy. We can also observe that the errors

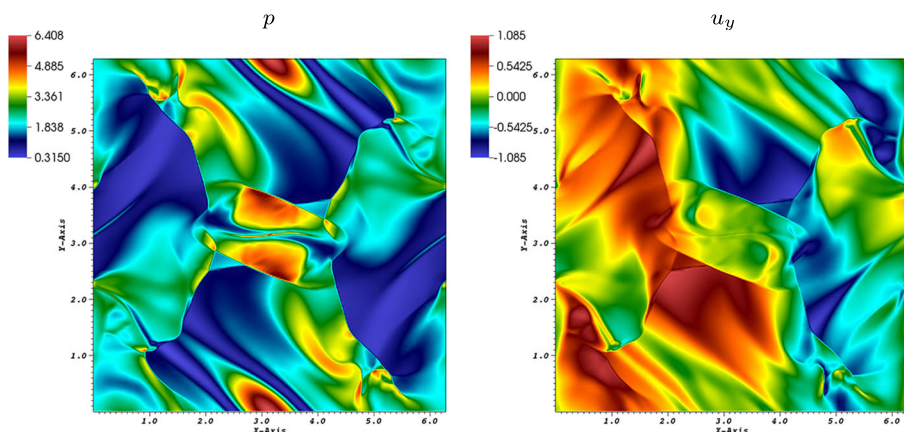


Fig. 5 Variables p and u_y at $t = \pi$ and $L = 9$, obtained with resistive CARMEN-MHD code for the 2D Orszag-Tang vortex and $\epsilon = 0.03$

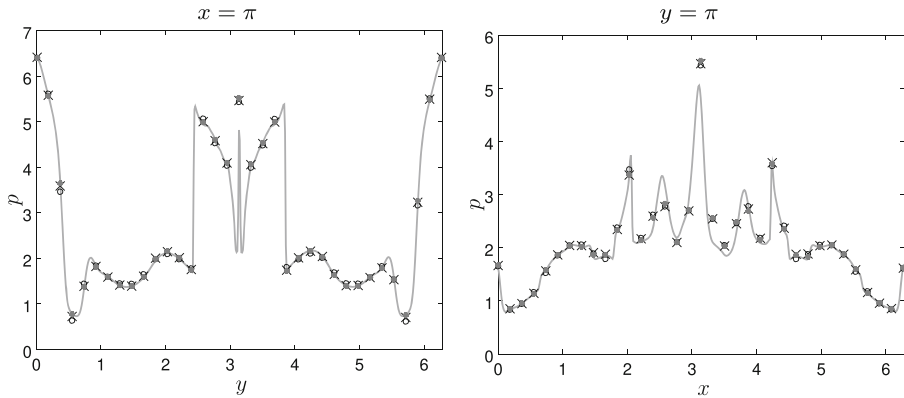


Fig. 6 Cuts of the variable p (from top to bottom) at $t = \pi$ and $L = 9$, for reference solution (line) and CARMEN-MHD adaptive solutions with $\epsilon = 0.01$ (cross), $\epsilon = 0.03$ (circle), $\epsilon^0 = 0.2$ (dot), at $x = \pi$ and $y = \pi$

of the pressure variable are larger, which happens because this variable is obtained from the other 7 MHD variables and thus their errors accumulate.

To evaluate the conservation of the total energy density \mathcal{E} , we present the values referring to $\log_{10}(\mathcal{E}_{MR}/\mathcal{E}_{UM})$ in Fig. 9, where \mathcal{E}_{MR} and \mathcal{E}_{UM} are the integral values of \mathcal{E} over time on adaptive and uniform meshes, respectively. By using this measure, it is possible to study the energy conservation for $\epsilon_0 = 0.2$ (circle), $\epsilon = 0.01$ (cross) and $\epsilon = 0.03$ (dotted) and also to verify how close these values are to the uniform case (solid line). The conservation of energy over time holds for every case presented, maintaining the physical properties. As much as we increase the number of cells of the simulations, the integral values converge to the uniform case, e.g., the $\epsilon_0 = 0.2$ case.

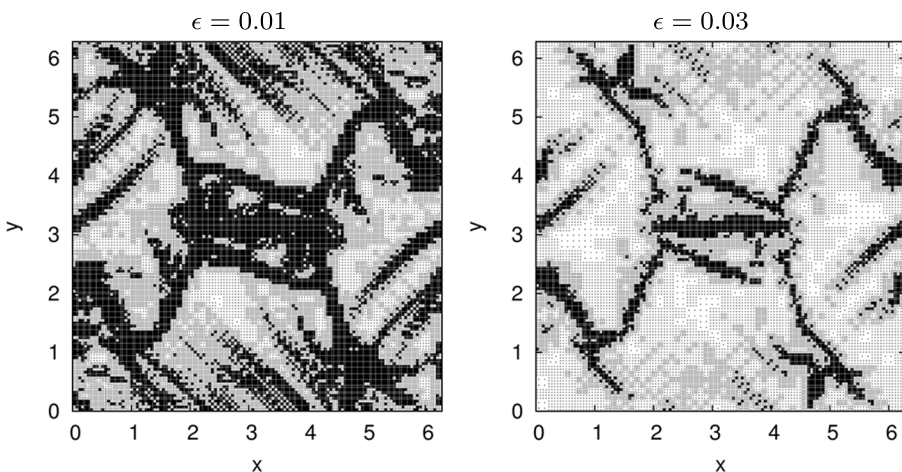


Fig. 7 Adaptive meshes at $t = \pi$ and $L = 9$, for the 2D Orszag-Tang vortex with $\epsilon = 0.01$ and $\epsilon = 0.03$

Table 2 Errors for the ideal 2D O–T vortex for the uniform and adaptive, compared to the reference solution at $L = 9$

CARMEN–MHD solver	Variables	Errors ($\times 10^{-2}$)	
		\mathcal{L}^1	\mathcal{L}^2
Uniform	p	2.256	7.052
	u_y	0.628	1.653
Adaptive ($\epsilon = 0.03$)	p	5.337	11.79
	u_y	1.954	3.582

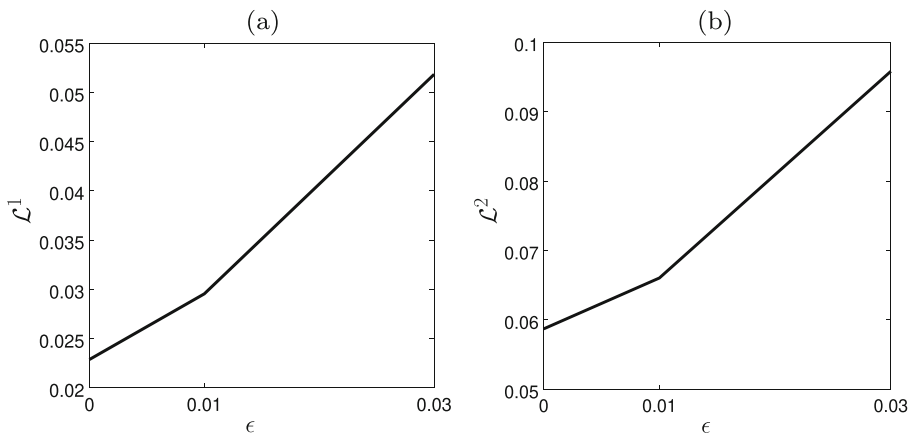


Fig. 8 Mass density ρ errors at $t = \pi$ and $L = 9$, for uniform and adaptive cases. **a** $\epsilon \times \mathcal{L}^1$ error. **b** $\epsilon \times \mathcal{L}^2$ error

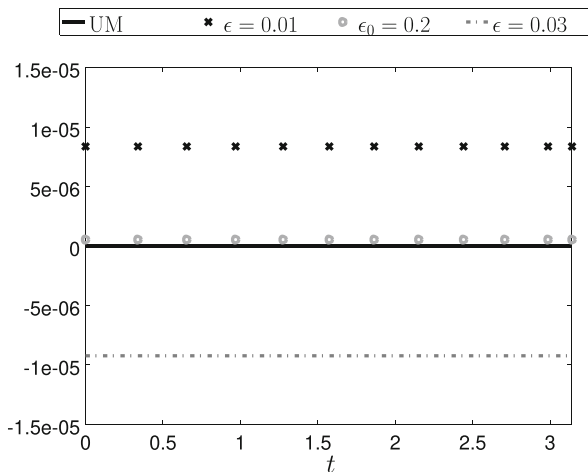


Fig. 9 Global values of $\log_{10}(\mathcal{E}_{MR}/\mathcal{E}_{UM})$ over t , obtained for the uniform mesh (black) and adaptive cases with $\epsilon_0 = 0.2$ (circle), $\epsilon = 0.01$ (cross) e $\epsilon = 0.03$ (dotted)

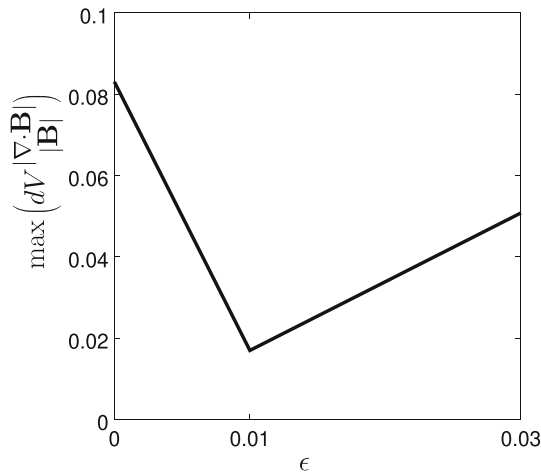


Fig. 10 Quantity $\epsilon \times \varepsilon_{div}$ at $t = \pi$ and $L = 9$, for uniform ($\epsilon = 0$) and adaptive cases with $\epsilon = 0.01$, $\epsilon = 0.03$

In Fig. 10, values of the divergence error ε_{div} are presented for two different values of ϵ and the uniform mesh computation with $L = 9$. For all cases, the values are below 10^{-1} and, consequently, the computations satisfy the restriction $\varepsilon_{div} < 1$. This shows that the GLM-MHD parabolic-hyperbolic strategy combined with the adaptive MR technique still presents small numerical values of $\nabla \cdot \mathbf{B}$, as expected. Hence this combination preserves the desired precision of the numerical solution at the final time of the simulation.

6.2 Shock-cloud interaction

The shock-cloud interaction models the disruption of a dense cloud with a shock-wave. The problem presents strong discontinuities and it is a challenging test case to evaluate the stability of numerical schemes. This type of simulation was firstly proposed in [9] and the initial condition we use is based on [52], given in Table 3 and with $u_y = u_z = B_x = 0.0$. We consider a circular cloud, 10 times denser than its background, with radius $r = 0.15$ and center $(0.25, 0.5, 0.5)$ in three dimensions and $(0.25, 0.5)$ in two dimensions. The shock is located at $x = 0.05$.

As simulation parameters we choose the physical time $t = 0.06$, the divergence cleaning parameter $\alpha = 0.4$, the threshold parameter $\epsilon^0 = 0.01$ and $\gamma = 5/3$.

Table 3 Shock-cloud initial condition

	ρ	p	u_x	B_y	B_z
$x \leq 0.05$	3.86859000	167.34500000	0.00000000	2.18261820	-2.18261820
$x > 0.05$	1.00000000	1.00000000	11.25360000	0.56418958	0.56418958

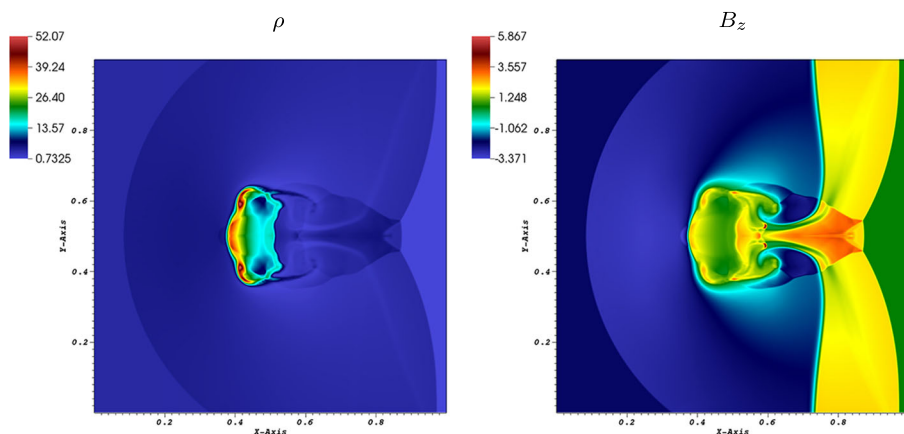


Fig. 11 Variables ρ and B_z at $t = 0.06$ and $L = 9$, obtained with the CARMEN-MHD code for ideal MHD, for the 2D shock-cloud problem and $\epsilon_0 = 0.01$

In two dimensions, we consider a high density cloud such as a circle centered in $(0.25, 0.5)$, with radius $r = 0.15$ and $\rho = 10$. The variables ρ and B_z are presented in Fig. 11, obtained with the CARMEN-MHD simulations. We can notice there is a strong discontinuity present in the interval $x \in [0.4, 0.5]$, which appears after the explosion coming from the interaction between the shock and the cloud. There are several local structures that are well represented in the visualization.

In Fig. 12, we present cuts of the variable B_z at $x = 0.5$, $y = 0.5$ and $t = 0.06$, for $\epsilon_0 = 0.01$ and $\epsilon = 0.003$. The variable B_z presents many discontinuities over the domain and the adaptive MR approach captures them. Moreover, these results reinforce the convergence of our numerical solutions. A slightly different topology may be observed in the solution, mainly due to the different numerical schemes. Nevertheless the solutions present the same behavior and have the same accuracy order.

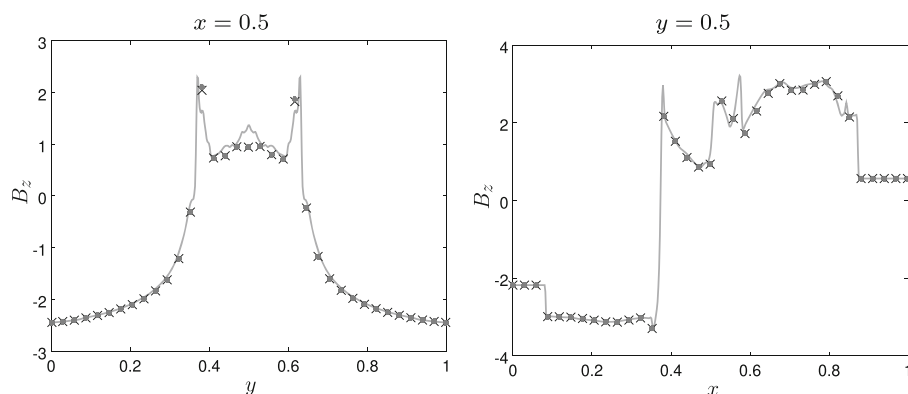


Fig. 12 Cuts of the variable B_z at $t = 0.06$, and $L = 9$, for reference solution (solid line) and CARMEN-MHD adaptive solutions with $\epsilon_0 = 0.01$ (cross)

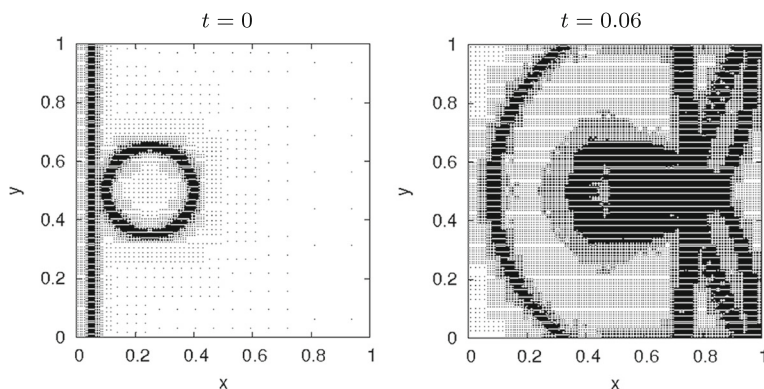


Fig. 13 Adaptive meshes at $t = 0$, $t = 0.06$ and $L = 9$, for the ideal 2D shock-cloud problem with $\epsilon_0 = 0.01$

The adaptive meshes are presented in Fig. 13 for simulations with threshold parameter $\epsilon_0 = 0.01$, that uses approximately 40% of the cells over time at $t = 0.06$. It implies a 55% reduction of CPU time. The meshes outline the local structures of the problem at times $t = 0$ and $t = 0.06$, with most of the cells being concentrated in the shock front and the border of the cloud for the initial time, and on the sharper local structures for the final time.

In Table 4, the errors evaluate quantitatively the proximity of the results obtained with uniform and adaptive meshes. The \mathcal{L}^1 and \mathcal{L}^2 errors remain of order 10^{-1} for both cases, showing that it is possible to obtain the same order of accuracy by using only 40% of the cells. We should recall at this point that the complexity of the shock-cloud problem, as it models an explosion with strong discontinuities, can influence on the elevated error values presented.

In three dimensions, the magnetic cloud is centered on $(0.25, 0.5, 0.5)$. We present the 3D shock-cloud simulation at level $L = 7$ (128^3 cells) and $\epsilon_0 = 0.01$. Figure 14 shows variables ρ and B_z obtained with the ideal MHD model at $t = 0.06$. The adaptive structures are coherent with the expected ones from the reference. The approximation errors are presented in Table 5. The errors \mathcal{L}^1 and \mathcal{L}^2 stay at the order 10^{-3} and 10^{-4} , respectively, for both adaptive and uniform cases. If we compare these values to the two dimensional case, we can notice that they decrease for the three-dimensional case. This probably occurs because of the z component,

Table 4 Errors obtained for the ideal 2D shock-cloud problem compared to the reference solution at level $L = 9$

CARMEN-MHD solver	Variables	Errors ($\times 10^{-1}$)	
		\mathcal{L}^1	\mathcal{L}^2
Uniform	ρ	1.543	11.55
	B_z	0.391	1.509
Adaptive ($\epsilon^0 = 0.01$)	ρ	1.539	11.60
	B_z	0.390	1.511

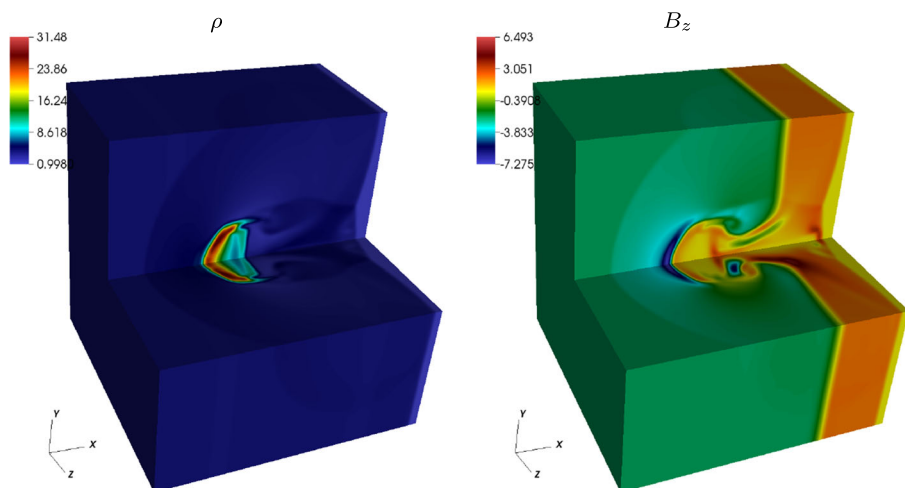


Fig. 14 Variables ρ and B_z at $t = 0.06$ and $L = 7$, obtained with ideal CARMEN–MHD code for the 3D shock-cloud problem and $\epsilon_0 = 0.01$

which can present a solution with a globally smoother behavior compared to the two dimensional case, where complex structures are located all over the domain.

The adaptive mesh is presented in Fig. 15 for $\epsilon_0 = 0.01$, decomposed according to the xy , yz and zx , as we take sections at the interval $[0.45, 0.55]$ on z , x and y axes, respectively. This type of visualization makes the mesh adaptivity clearer for the 3D case. It is possible to observe that the cells of this simulation are located exactly in regions of the stronger discontinuities. The xy mesh is similar to the 2D case, in a coarser level. The 3D simulation demands 58% of the cells over time, when compared to a uniform mesh, and causes a reduction of 34% in CPU time. When we increase the refinement level to $L = 8$, the percentage of cells required decreases to 42% and reduces the CPU time by 54%. This suggests that the adaptivity tends to improve as we increase the maximum level of refinement of the 3D case.

For the simulation of the resistive 3D shock-cloud, we add a constant resistivity $\eta = 0.02$ all over the computational domain. Numerical tests showed that this value is reasonable for this problem, as the diffusive effect is sufficient to smooth the local structures of the problem without losing its intrinsic topology.

Table 5 Errors obtained for the ideal 3D shock-cloud problem for the uniform and adaptive ($\epsilon^0 = 0.01$), compared to the reference solution at level $L = 7$

CARMEN–MHD solver	Variables	Errors ($\times 10^{-3}$)	
		\mathcal{L}^1	\mathcal{L}^2
Uniform	ρ	0.143	4.798
	B_z	0.132	3.966
Adaptive	ρ	0.146	4.818
	B_z	0.138	4.089

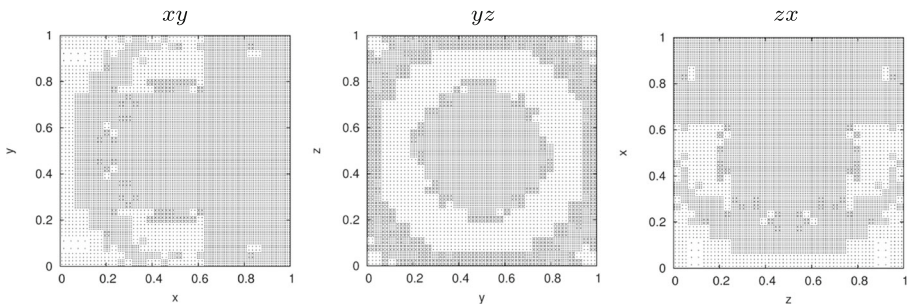


Fig. 15 Ideal 3D shock-cloud problem: adaptive mesh ($\epsilon_0 = 0.01$) sections on planes xy , yz , zx at $t = 0.06$ and $L = 7$

To allow a quantitative comparison with the reference, we choose a coarser level, $L = 6$, corresponding to 64^3 cells. In Fig. 16, we present visualizations of the variables ρ and B_z . Due to the dissipative effects and the refinement level, the structures of the solution are much smoother, compared to the ideal case. However, we can notice that the topology of the solution is maintained. This adaptive simulation requires 76% of the cells over time for $L = 6$. When we refine the mesh, we obtain a reduction of the cells with an improvement of the CPU time. In particular, for a simulation with $L = 7$, 55% of the cells are needed over time and there is a 35% reduction in CPU time, which is significant in the computational context and reinforces the efficiency of the adaptive MR approach. The corresponding errors are presented in Table 6.

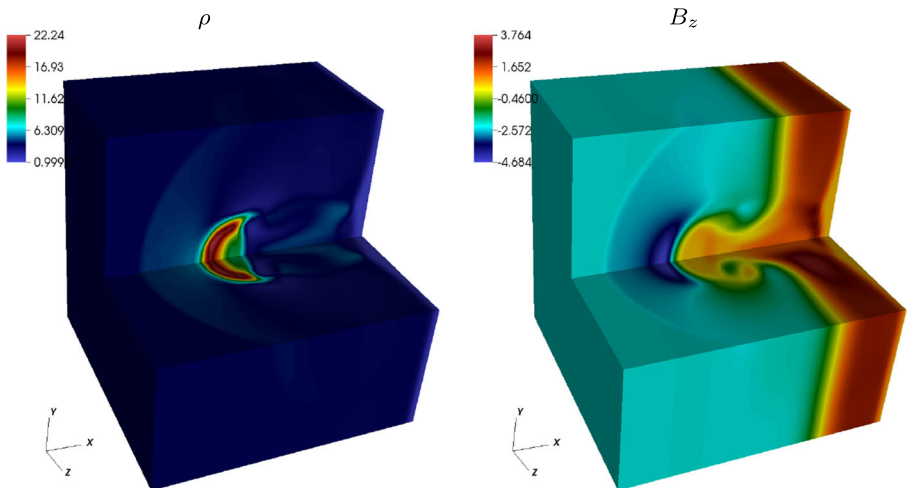


Fig. 16 Constant resistive 3D shock-cloud problem: adaptive simulation for variables ρ and B_z with $\epsilon_0 = 0.01$ at $t = 0.06$ with $L = 6$, and $\eta = 0.02$

Table 6 Errors obtained for the constant resistive 3D shock-cloud problem for the uniform and adaptive ($\epsilon^0 = 0.01$) cases, compared to the reference solution at level $L = 6$

CARMEN-MHD solver	Variables	Errors ($\times 10^{-3}$)	
		\mathcal{L}^1	\mathcal{L}^2
Uniform	ρ	0.401	5.549
	B_z	0.214	3.234
Adaptive	ρ	0.403	5.558
	B_z	0.214	3.230

6.3 Magnetic reconnection

When we include Ohmic resistivity effects in the MHD equations, there is no conservation of magnetic flux anymore. This type of physical situation can change the topology of the magnetic field lines, allowing us to study different plasma problems, in particular, magnetic reconnection phenomena. The magnetic reconnection is a fundamental process in highly electrically conductive plasmas, which allows the conversion of magnetic energy to kinetic energy. It occurs when the magnetic field lines disconnect and reconnect again, changing its directions and restructuring the macroscopic plasma quantities.

We consider the Petschek reconnection model [46], in which the reconnection rate is faster compared to the Sweet–Parker setting. The simulation proposed here was first presented in [33] with the following configuration: The initial condition for the magnetic reconnection is given by $\rho = 1$, $p = 0.1$, $\mathbf{u} = \mathbf{0}$, $B_x = 0$ and

$$B_y = \begin{cases} -1, & \text{if } x < -0.05 \\ \sin(\pi x/0.01), & \text{if } |x| \leq 0.05 \\ 1, & \text{if } x > 0.05 \end{cases} \quad (30)$$

The computational domain is $\Omega = [-0.5, 0.5] \times [-2, 2]$, where the diffusion region is defined as $[-0.05, 0.05] \times [0.2, 0.2]$, and the resistivity inside this region is given by $\eta(x, y) = 0.25 \eta_0 (\cos(\pi x/0.1) + 1) (\cos(\pi y/0.4) + 1)$, where $\eta_0 = 0.00075$ is the initial resistivity.

In absence of a reference solution, we compare our results to the ones presented in [33]. The benchmark results are obtained on a 2048×4096 high resolution mesh, while we use a 512×512 mesh (corresponding to a refinement level $L = 9$), employing a WENO scheme with Lax-Friedrich flux, a second order TVD Runge-Kutta time scheme and a damping approach at the boundaries.

In the presented simulations we use the final time $t = 2.5$, the adiabatic constant $\gamma = 5/3$, the parabolic-hyperbolic correction parameter $\alpha_p = 0.4$, the Courant number $\nu = 0.4$, the threshold parameter $\epsilon = 0.0005$, and Neumann boundary conditions in all directions.

The cuts at $y = 0.5$ are presented for the variables ρ , u_y , in Fig. 17, for the interval $[-0.2, 0, 2]$. These cuts are similar between each other, however we can find some differences in the solutions, which can be attributed to the resolution or the

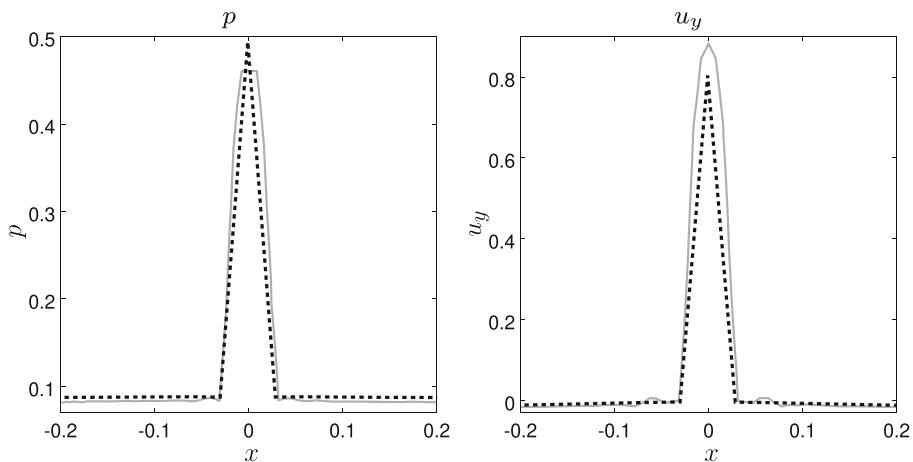


Fig. 17 Cuts in variables p , u_y , at $t = 2.5$ and $L = 9$, obtained with non-constant resistive CARMEN–MHD with an adaptive simulation using $\epsilon_0 = 0.005$ (dotted); and the reference solution presented in [33, p. 1630, Fig. 11](solid line)

chosen numerical scheme. High resolution can increase the accuracy of the numerical solution, however, the solution obtained at $L = 9$ with the CARMEN–MHD code already presents the expected structures.

The variables B_x and B_z obtained with the CARMEN–MHD code are shown in Fig. 18. The reconnection structure occurs well defined and is present in the computational domain in each variable. Neither numerical instabilities nor oscillations are observed during the simulation.

The adaptive approach needs 55% of the cells over time for this simulation, leading to a 40% reduction in CPU time. We found that the threshold parameter $\epsilon = 0.005$ is

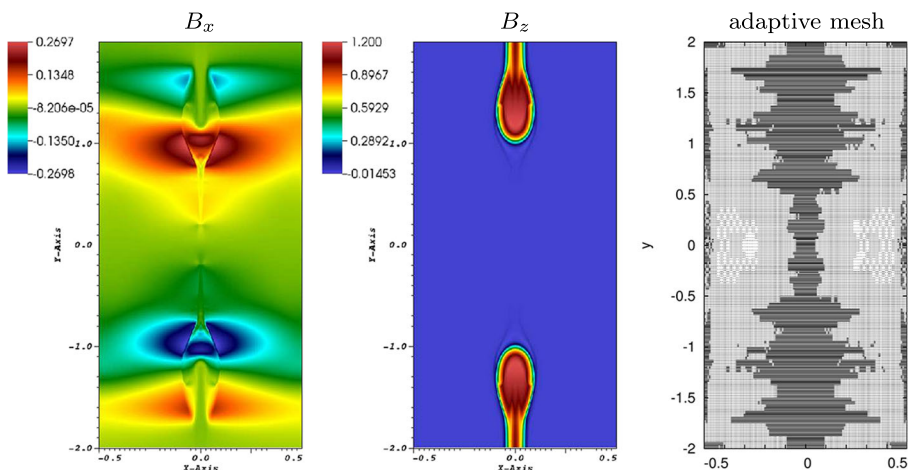


Fig. 18 Resistive magnetic reconnection problem: adaptive simulations with $\epsilon = 0.005$ for the variables B_x , B_z , and the adaptive mesh at $t = 2.5$, and $L = 9$

optimal for this case, since we can obtain a good compromise between compression and coherent representation of the physical structures. When we choose a slightly larger value, e.g., $\epsilon = 0.008$, the central structures of the problem are not well represented anymore. It is also possible to use smaller values for ϵ , nevertheless this would lead to lower compression. The majority of the refined cells in the adaptive mesh is located in the central region of the domain (corresponding to darker symbols). The other regions present a coarser refinement (corresponding to lighter symbols).

If we compare the structures present in the variables B_x and B_y with the adaptive mesh, we can conclude that the mesh is efficiently adapted where large discontinuities and diffusion regions are located. This shows that the adaptive algorithm is indeed efficient to represent and identify automatically the structures of the problem. Moreover, we expect that higher resolution simulations will further improve the adaptive representation and thus the gains in memory and CPU time reduction.

In this adaptive simulation the physical behavior of the magnetic reconnection phenomena is sustained. In particular, we verified that the velocity field is compatible with the magnetic reconnection settings, in which the velocity direction initially points to the diffusion region and, after reconnection, its orientation does change.

7 Conclusion

Fully adaptive numerical simulations using the CARMEN-MHD code were performed in two and three space dimensions in order to verify the implementation and its computational efficiency. The numerical method is based on a finite volume discretization on Cartesian grids and uses an adaptive multiresolution approach for introducing dynamically refined dyadic meshes. Our choice using Cartesian geometries instead of general geometries is motivated by the fact that multiresolution analysis is particularly attractive in this context. However generalization are possible in future work considering, e.g., multiresolution on triangles proposed by Cohen et al. [7] or more general tessellations borrowing techniques introduced in the field of computer graphics, e.g., using hierarchy refinement procedures [6]. This would allow designing adaptive multiresolution solvers on unstructured grids considering also complex geometries. Selected benchmarks were chosen in order to comprehend different physical and numerical phenomena, and to ensure the correct behavior of the code in capturing the intrinsic topology of each situation. The obtained results were then compared with the FLASH code which served as reference.

Quantitative and qualitative comparisons of the numerical solutions were carried out and their convergence towards reference solutions was shown. The obtained results are coherent for both, adaptive and uniform grid approaches. The physical restrictions of the MHD model are maintained in the context of the numerical solution, contributing to the reliability of the results and its adequate reproduction.

Depending on the benchmark the structures present in the solution of the MHD model are located in different regions of the domain, varying according to the chosen variable. This type of situation is challenging in the context of the adaptivity, since the mesh must be adapted adequately. The adaptivity criteria used here were shown to be efficient for identifying the structures of the solution, even in cases where the

structures do not present local features at all. We also showed that the numerical simulations are stable and do not require additional stabilization, e.g., adding numerical diffusion.

Moreover, we observed that it is possible to design an optimal mesh adaptivity, by evaluating the relation between the threshold parameter and the approximation errors. The optimal choice can decrease the CPU time, while ensuring the accuracy of the numerical solution. The adaptive multiresolution approach can increase significantly the computational gains of the simulations, even in non-parallel simulations. Thereby, this approach is shown to be computationally efficient to deal with the proposed MHD models.

Finally, let us mention that in the context of MHD, adaptive multiresolution computations, especially in 3D, are recent and in this work we presented their potential, by showing their efficiency using adaptive meshes, while preserving the accuracy of the underlying discretization. We thus conclude that the verification of the CARMEN–MHD code was successful and we provide its open source-code and documentation for the community, in order to continue the research on MHD simulations of other interesting and challenging physical problems.

Acknowledgments We are indebted to Eng. V. E. Menconi for his invaluable computational assistance.

Funding This work received financial support from the FAPESP (Grant: 2015/ 25624-2), CNPq (Grants: 302226/2018-4, 307083/2017-9, 306038/2015-3, 302226/2018-4), and FINEP (Grant: 0112052700) for financial support of this research. K.S. received partial support from the French Federation for Magnetic Fusion Studies (FR-FCM) and the Eurofusion consortium, funded by the Euratom research and training programme 2014–2018 and 2019–2020 under grant agreement no. 633053.

Declarations

Disclaimer The views and opinions expressed herein do not necessarily reflect those of the European Commission.

References

1. Berger, M.J., Colella, P.: Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* **82**, 64–84 (1989)
2. Berger, M.J., Oliger, J.: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* **53**(3), 484–512 (1984)
3. Bihari, B.L., Harten, A.: Multiresolution schemes for the numerical solution of 2-D conservation laws i. *SIAM J. Sci. Comput.* **18**(2), 315–354 (1997)
4. Chiavassa, G., Donat, R.: Point value multiscale algorithms for 2D compressible flows. *SIAM J. Sci. Comput.* **23**(3), 805–823 (2001)
5. Cohen, A.: *Numerical Analysis of Wavelet Methods*. Elsevier, Paris (2003)
6. Cohen, A., Dyn, N., Hecht, F., Mirebeau, J.: Adaptive multiresolution analysis based on anisotropic triangulations. *Math. Comput.* **81**(278), 789–810 (2012)
7. Cohen, A., Dyn, N., Kaber, S., Postel, M.: Multiresolution schemes on triangles for scalar conservation laws. *J. Comput. Phys.* **161**, 264–286 (2000)
8. Cohen, A., Kaber, S.M., Müller, S., Postel, M.: Fully adaptive multiresolution finite volume schemes for conservation laws. *Math. Comput.* **72**(241), 183–225 (2003)

9. Dai, W., Woodward, P.R.: A simple finite difference scheme for multidimensional magnetohydrodynamical equations. *J. Comput. Phys.* **142**(2), 331–369 (1998)
10. Dedner, A., Kemm, F., Kröner, D., Munz, C.D., Schnitzer, T., Wesenberg, M.: Hyperbolic divergence cleaning for the MHD equations. *J. Comput. Phys.* **175**, 645–673 (2002)
11. Deiterding, R., Domingues, M.O., Gomes, S.M., Roussel, O., Schneider, K.: Adaptive multiresolution or adaptive mesh refinement: A case study for 2D Euler equations. *ESAIM Proceedings* **29**, 28–42 (2009)
12. Deiterding, R., Domingues, M.O., Gomes, S.M., Schneider, K.: Comparison of adaptive multiresolution and adaptive mesh refinement applied to simulations of the compressible Euler equations. *SIAM J. Sci. Comput.* **38**(5), S173–S193 (2016)
13. Domingues, M.O., Deiterding, R., Lopes, M.M., Gomes, A.K.F., Mendes, O., Schneider, K.: Wavelet-based parallel dynamic mesh adaptation for magnetohydrodynamics in the AMROC framework. *Comput Fluids* **190**, 374–381 (2019)
14. Domingues, M.O., Gomes, A.K.F., Gomes, S., Mendes, O., Di Pierro, B., Schneider, K.: Extended generalized lagrangian multipliers for magnetohydrodynamics using adaptive multiresolution methods. *ESAIM Proc.* **43**, 95–107 (2013)
15. Domingues, M.O., Gomes, S.M., Roussel, O., Schneider, K.: Adaptive multiresolution methods. *ESAIM Proc.* **34**, 1–96 (2011)
16. Fambri, F., Dumbser, M., Zanotti, O.: Space-time adaptive ADER-DG schemes for dissipative flows: Compressible navier–Stokes and resistive MHDx equations. *Comput. Phys. Commun.* **220**, 297–318 (2017)
17. Feng, X.: *Magnetohydrodynamic Modeling of the Solar Corona and Heliosphere*. Springer, Berlin (2020)
18. Fryxell, B., Olson, K., Ricker, P., Timmes, F.X., Zingale, M., Lamb, D.Q., MacNeice, P., Rosner, R., Truran, J.W., Tufo, H.: FLASH: an adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. *Astrophys. J. Suppl. Ser.* **131**, 273–334 (2000)
19. Goedbloed, J.P., Poedts, S.: *Principles of Magnetohydrodynamics*. Cambridge University Press, Cambridge (2004)
20. Gomes, A.K.F.: Análise multirresolução adaptativa no contexto da resolução numérica de um modelo de magnetohidrodinâmica ideal. Master's thesis, Instituto Nacional de Pesquisas Espaciais (INPE) São José dos Campos (2012)
21. Gomes, A.K.F.: Simulação Numérica De Um Modelo Magneto-Hidrodinâmico Multidimensional No Contexto Da Multirresolução Adaptativa Por MáDias Celulares. Ph.D. Thesis. Instituto Nacional de Pesquisas Espaciais, São José dos Campos (2018)
22. Gomes, A.K.F., Domingues, M.O., Mendes, O.: Ideal and resistive magnetohydrodynamic two-dimensional simulation of the Kelvin–Helmholtz instability in the context of adaptive multiresolution analysis. *TEMA (São Carlos)* **18**(2), 317–333 (2017)
23. Gomes, A.K.F., Domingues, M.O., Mendes, O., Schneider, K.: On the verification of adaptive three-dimensional multiresolution computations of the magnetohydrodynamic equations. *J Appl Nonlinear Dyn* **7**, 231–242 (2018)
24. Gomes, A.K.F., Domingues, M.O., Schneider, K., Mendes, O., Deiterding, R.: An adaptive multiresolution method for ideal magnetohydrodynamics using divergence cleaning with parabolic–hyperbolic correction. *Appl. Numer. Math.* **95**, 199–213 (2015)
25. Groen, D., Zasada, S.J., Coveney, P.V.: Survey of multiscale and multiphysics applications and communities. *Comput Sci Eng* **16**(2), 34–43 (2014)
26. Harten, A.: Discrete multi-resolution analysis and generalized wavelets. *Appl. Numer. Math.* **12**(1), 153–192 (1993)
27. Harten, A.: Adaptive multiresolution schemes for shock computations. *J. Comput. Phys.* **115**(2), 319–338 (1994)
28. Harten, A.: Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Commun. Pure Appl. Math.* **48**(12), 1305–1342 (1995)
29. Harten, A.: Multiresolution representation of data: a general framework. *SIAM J Numer Anal* **33**(3), 385–394 (1996)
30. Hopkins, P.F., Raives, M.J.: Accurate, meshless methods for magnetohydrodynamics. *Mon. Not. R. Astron. Soc.* **455**(1), 51–88 (2016)
31. Jardin, S.C.: Review of implicit methods for the magnetohydrodynamic description of magnetically confined plasmas. *J. Comput. Phys.* **231**(3), 822–838 (2012)

32. Jiang, G.S., Wu, C.C.: A high-order WENO finite difference scheme for the equations of ideal magnetohydrodynamics. *J. Comput. Phys.* **150**(2), 561–594 (1999)
33. Jiang, R.L., Fang, C., Chen, P.F.: A new MHD code with adaptive mesh refinement and parallelization for astrophysics. *Comput. Phys. Commun.* **183**(8), 1617–1633 (2012)
34. Kaibara, M.K., Gomes, S.M.: A fully adaptive multiresolution scheme for shock computations. In: *Godunov methods: theory and applications*, pp. 497–503. Springer US, Boston (2001)
35. Kleimann, J., Kopp, A., Fichtner, H., Grauer, R., Germaschewski, K.: Three-dimensional mhd high-resolution computations with CWENO employing adaptive mesh refinement. *Comput. Phys. Commun.* **158**(1), 47–56 (2004)
36. Kolomenskiy, D., Onishi, R., Uehara, H.: Data compression for environmental flow simulations. [arXiv:1810.04822](https://arxiv.org/abs/1810.04822) (2018)
37. Koskinen, H.E.J., Baker, D.N., Balogh, A., Gombosi, T., Veronig, A., von Steiger, R.: Achievements and challenges in the science of space weather. *Space Sci. Rev.* **212**, 1137–1157 (2017)
38. LeVeque, R.J.: *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge (2002)
39. Londrillo, P., Del Zanna, L.: High-order upwind schemes for multidimensional magnetohydrodynamics. *Astrophys. J.* **530**(1), 508 (2000)
40. Lopes, M.M., Deiterding, R., Gomes, A.K.F., Mendes, O., Domingues, M.O.: An ideal compressible magnetohydrodynamic solver with parallel block-structured adaptive mesh refinement. *Comput Fluids* **173**, 293–298 (2018)
41. Mignone, A., Tzeferacos, P.: A second-order unsplit Godunov scheme for cell-centered MHD: The CTU-GLM scheme. *J. Comput. Phys.* **229**(6), 2117–2138 (2010)
42. Miyoshi, T., Kusano, K.: A multi-state HLL approximate Riemann solver for ideal magnetohydrodynamics. *J. Comput. Phys.* **208**, 315–344 (2005)
43. Morley, S.K.: Challenges and opportunities in magnetospheric space weather prediction. *Space Weather* **18** e2018SW002108 (2019)
44. Müller, S.: *Adaptive Multiscale Schemes for Conservation Laws*. In: *Lectures Notes in Computational Science and Engineering*, vol. 27. Springer, Heidelberg (2003)
45. Orszag, S.A., Tang, C.M.: Small-scale structure of two-dimensional magnetohydrodynamic turbulence. *J. Fluid Mech.* **90**(01), 129–143 (1979)
46. Petschek, H.E.: Magnetic field annihilation. *NASA Special Publication* **50**, 425 (1964)
47. Roussel, O.: Développement d'un algorithme multiresolution adaptatif tridimensionnel pour la résolution des équations aux dérivées partielles paraboliques. Ph.D. thesis, Université de la Méditerranée (2003)
48. Roussel, O., Schneider, K., Tsigulin, A., Bockhorn, H.: A conservative fully adaptative multiresolution algorithm for parabolic PDEs. *J. Comput. Phys.* **188**, 493–523 (2003)
49. Ryu, D., Miniati, F., Jones, T., Frank, A.: A divergence-free upwind code for multidimensional magnetohydrodynamic flows. *Astrophys. J.* **509**(1), 244 (1998)
50. Schneider, K., Vasilyev, O.V.: Wavelet methods in computational fluid dynamics. *Annu. Rev. Fluid Mech.* **42**, 473–503 (2010)
51. Tóth, G., Van der Holst, B., Sokolov, I.V., De Zeeuw, D.L., Gombosi, T.I., Fang, F., Manchester, W.B., Meng, X., Najib, D., Powell, K.G., et al.: Adaptive numerical algorithms in space weather modeling. *J. Comput. Phys.* **231**(3), 870–903 (2012)
52. Tóth, G., Arminjon, P.: Central finite volume schemes with constrained transport divergence treatment for three-dimensional ideal MHD. *J. Comput. Phys.* **212**(2), 617–636 (2006)
53. Van der Holst, B., Keppens, R., Meliani, Z.: A multidimensional grid-adaptive relativistic magnetofluid code. *Comput. Phys. Commun.* **179**(9), 617–627 (2008)
54. Van Leer, B.: Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme. *J. Computat Phys* **14**(4), 361–370 (1974)
55. Vanaverbeke, S., Keppens, R., Poedts, S.: GRADSPMHD: A parallel MHD code based on the sph formalism. *Comput. Phys. Commun.* **185**(3), 1053–1073 (2014)
56. Zanotti, O., Fambri, F., Dumbser, M., Hidalgo, A.: Space-time adaptive ADER discontinuous Galerkin finite element schemes with a posteriori sub-cell finite volume limiting. *Comput Fluids* **118**, 204–224 (2015)
57. Ziegler, U.: The NIRVANA code: Parallel computational MHD with adaptive mesh refinement. *Comput. Phys. Commun.* **179**(4), 227–244 (2008)