



HAL
open science

On the verification of adaptive three-dimensional multiresolution computations of the magneto hydrodynamic equations

Anna Karina Fontes Gomes, Margarete Oliveira Domingues, Odim Mendes,
Kai Schneider

► To cite this version:

Anna Karina Fontes Gomes, Margarete Oliveira Domingues, Odim Mendes, Kai Schneider. On the verification of adaptive three-dimensional multiresolution computations of the magneto hydrodynamic equations. 2024. hal-02088746

HAL Id: hal-02088746

<https://hal.science/hal-02088746v1>

Preprint submitted on 2 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the verification of adaptive three-dimensional multiresolution computations of the magnetohydrodynamic equations

Anna Karina Fontes Gomes^{*1}, Margarete Oliveira Domingues², Odin Mendes³, and Kai Schneider⁴

¹Post-graduation program in Applied Computing, National Institute for Space Research, São José dos Campos, Brazil

²Associated Laboratory for Computing and Applied Mathematics, National Institute for Space Research, São José dos Campos, Brazil

³Space Geophysics Division, National Institute for Space Research, São José dos Campos, Brazil

⁴Institut de Mathématiques de Marseille, Aix-Marseille Université, CNRS, Centrale Marseille, I2M UMR 7373

Abstract

Magnetohydrodynamics is an important tool to study the dynamics of plasma Space Physics. In this context, we introduce a three-dimensional magnetohydrodynamic solver with divergence-cleaning in the adaptive multiresolution CARMEN code. The numerical scheme is based on a finite volume discretization that ensures the conservation of physical quantities. The adaptive multiresolution approach allows for automatic identification of local structures in the numerical solution and thus provides an adaptive mesh refined only in regions where the solution needs more improved resolution. We assess the three-dimensional magnetohydrodynamic CARMEN code and compare its results with the ones from the well-known FLASH code.

1 Introduction

The Magnetohydrodynamic (MHD) theory helps in understanding the behavior of the macroscopic space plasma scenario. In particular, MHD modeling describes the flow dynamics of a conducting fluid under the influence of a magnetic field and its nonlinear interaction.

The MHD model can be formulated as an initial value problem composed by four nonlinear, vector valued partial differential equations. In most practical cases, it does not have an analytic solution and we need numerical methods to solve it. For instance, MHD simulations in three-dimensions can be found in [1, 2] and in more recent works [3, 4] using either finite difference or finite volume methods with and without adaptive mesh refinement techniques. Due to the high computational cost of three-dimensional ideal MHD simulations, it is adequate to use, when possible, an adaptive mesh procedure to create a mesh which has more cells at locations where local structures are present, and there are less cells in smooth regions.

Here, we introduce a verification of a new code for an adaptive ideal three-dimensional MHD model discretized with a finite volume method in which the domain is divided into small volumes or cells and a multiresolution technique is used to introduce an adapted mesh in order to reduce the number of operations and memory while preserving the precision of the discretization. Previous results were obtained for two-dimensional MHD equations in [5, 6, 7]. To perform the present simulations we extended the adaptive

*Corresponding author.

Email address: anna.gomes@inpe.br

multiresolution CARMEN code [8] and modified the MHD equations in three-dimensions [5, 7]. As a benchmark for our results we use the well-known astrophysical FLASH code¹ which uses a finite volume method on a regular Cartesian mesh. We present results for two Riemann problems in order to check our new adaptive MHD implementation.

The content of this paper is organized as follows: in Section 2, we introduce the equations of the ideal MHD model. The numerical approach to simulate the MHD equations is described in Section 3, where we also briefly describe the parabolic-hyperbolic divergence cleaning, the finite volume, and the MR methods. In Section 4, results are presented and discussed. Lastly, we make our conclusions and give some perspectives.

2 The MHD model

The ideal MHD model describes the physics of an ideal conducting fluid under the influence of a magnetic field. The study of the MHD model and its simulation is important to better understand how an ideal plasma behaves. It enables us to perform numerical investigations of more realistic physical phenomena, which are directly related to space sciences. Here we present the set of four partial differential equations that constitutes the ideal MHD model.

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

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

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

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{B} - \mathbf{B} \mathbf{u}) = 0, \quad (1d)$$

with the model variables: ρ density, $\mathbf{u} = (u_x, u_y, u_z)$ the fluid velocity vector, $\mathbf{B} = (B_x, B_y, B_z)$ the magnetic field vector and \mathcal{E} total mass density defined as

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

where p is the pressure, and u and B are the magnitude of \mathbf{u} and \mathbf{B} , respectively.

The model is written in its conservation form, *i.e.*, a differential equation of the form

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{0}, \quad (3)$$

with the vector of quantities \mathbf{U} and the flux tensor $\mathbf{F} = \mathbf{F}(\mathbf{U})$. In our case, we have

$$\mathbf{U} = \begin{pmatrix} \rho \\ \mathcal{E} \\ \rho \mathbf{u} \\ \mathbf{B} \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho \mathbf{u} \\ (\mathcal{E} + p + \frac{\mathbf{B} \cdot \mathbf{B}}{2}) \mathbf{u} - \mathbf{B} (\mathbf{u} \cdot \mathbf{B}) \\ \rho \mathbf{u} \mathbf{u} + \mathbf{I} (p + \frac{\mathbf{B} \cdot \mathbf{B}}{2}) - \mathbf{B} \mathbf{B} \\ \mathbf{u} \mathbf{B} - \mathbf{B} \mathbf{u} \end{pmatrix}. \quad (4)$$

The conservative formulation of the MHD equations describes the conservation of mass density, total energy density, momentum, and magnetic flux. It is important to note that the Gauss law of magnetism is also part of the model and can be understood as a physical constraint of the magnetic field. It implies that the divergence of the magnetic field is zero, *i.e.*, there are no magnetic monopoles in the solution of the MHD equations [9].

¹<http://flash.uchicago.edu/>

3 Numerical Approach

When it comes to numerical simulations, where the domain of the physical problem has to be discretized, the divergence constraint is not satisfied anymore due to approximation errors. In order to minimize these errors, it is acceptable to introduce a correction to the model so that the divergence errors are well controlled. There are several types of corrections and some of them are discussed here [10, 11]. We have chosen a divergence cleaning approach, which does not impose zero divergence, but controls the errors such that non-physical behavior is avoided in the numerical solution of the model.

The parabolic-hyperbolic divergence cleaning was first proposed in [12] and it offers both propagation to the boundary and dissipation of the divergence errors. The correction consists in adding a differential operator $D = \frac{1}{c_h^2} \frac{\partial}{\partial t} + \frac{1}{c_p^2}$ to the divergence constraint of the \mathbf{B} field, resulting in the Generalized Lagrange Multiplier MHD model (GLM–MHD), composed by Eqs. (1b), (1c), (1d) and the additional equations

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{B} - \mathbf{B}\mathbf{u} + \psi \mathbf{I}) = 0, \quad (5a)$$

$$\frac{\partial \psi}{\partial t} + c_h^2 \nabla \cdot \mathbf{B} = -\frac{c_h^2}{c_p^2} \psi \quad (5b)$$

where ψ is a scalar-valued function, \mathbf{I} the identity tensor, c_p and c_h are the parabolic and hyperbolic constants, respectively. We can note that System 5 reduces to the ideal MHD model when $\psi = 0$. The GLM–MHD model is used for the numerical simulation and the divergence cleaning is implemented in the flux as a source term in Equation 5b. The constant c_h is defined as

$$c_h = \max \left[\nu \frac{\Delta h}{\Delta t}, \max(|u_d| \pm c_f) \right], \quad (6)$$

where $\Delta h = \min(\Delta x, \Delta y, \Delta z)$, with $\Delta x, \Delta y, \Delta z$ are the space steps in each direction, ν the Courant number, u_d is the velocity of the d -th component and c_f is the fast magnetoacoustic wave speed of the MHD model. The c_p value is defined in terms of the parameter $\alpha = \Delta h \frac{c_h}{c_p^2}$, as discussed in [13].

3.1 Finite volume discretization

The discretization of the model is performed with a finite volume method, which is based on the integral form of the conservation laws. The idea here is to divide the computational domain into mesh cells, assigning a cell average to the vector quantity $\bar{\mathbf{U}} = \bar{\mathbf{U}}_{i,j,k}$ at each cell $C = C_{i,j,k}$. A cell average is defined as the integral of the quantity over the cell, i.e., the cell average of the quantity $\mathbf{U} = \mathbf{U}_{i,j,k}$ is

$$\bar{\mathbf{U}} = \frac{1}{|C|} \int_C \mathbf{U} d\mathbf{x}, \quad (7)$$

where $d\mathbf{x} = dx dy dz$. By taking the integral form of Equation 3, along with the concept of the cell average and the divergence theorem, we have

$$\frac{\partial \bar{\mathbf{U}}}{\partial t} + \int_{\partial C} \mathbf{F} \cdot \mathbf{n}_d d\mathbf{x} = \mathbf{0}, \quad (8)$$

where $\mathbf{n} = \mathbf{n}_d$ is the normal vector to the interface of the cell in direction d and assuming that the boundary terms vanish. The integral on the left-hand side of the above equation means that the flux \mathbf{F} must be evaluated on the interface of cell C , denoted by ∂C , projected onto the normal vector. The flux function can be numerically computed for each cell center, but it is not possible to analytically estimate it on the interface of the cells. In this context, it is necessary to make approximations of the flux. Here we have chosen the Harten-Lax-van Leer-Discontinuities (HLLD) Riemann solver [14].

The HLLD scheme is useful to solve isolated discontinuities in the solution of the MHD system and it preserves the positivity of the quantities. The MHD eigensystem has seven eigenvalues, related to the

fast magnetoacoustic waves (C_f), Alfvén waves (C_a), slow magnetoacoustic waves (C_s), and entropy (C_e). These waves compose the MHD space-time Riemann fan, illustrated in Figure 1, in which they are organized according to their value regarding to their origin. The HLLD flux is efficient to solve isolated discontinuities and constructed by the means of an approximate Riemann problem in a four-state Riemann fan, divided by two Alfvén and one entropy wave.

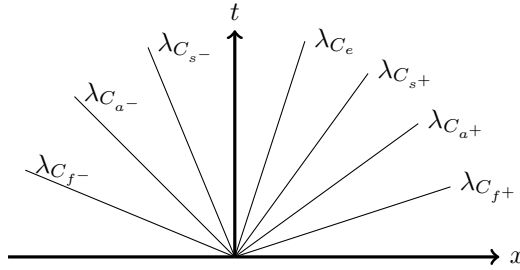


Figure 1: Illustration of the MHD Riemann fan.

This type of numerical flux is computed by sweeping the cells in each direction individually, a technique called splitting. Thus, the flux in x -direction, for instance, does not depend on the fluxes in the y and z directions. A brief summary of this Riemann solver can be found in [7]. To calculate the numerical flux, the conservative variables are needed and we reconstruct them with the Monotonized Central (MC) limiter, which improves the accuracy of the solution by achieving a second order approximation.

3.2 Adaptive multiresolution analysis

After performing the discretization and flux approximations, we use multiresolution analysis to adapt the computational mesh to the numerical solution of the model. The multiresolution analysis comes from the idea that data can be decomposed into several levels of refinement ℓ , from the coarser to the most refined level L , i. e., $0 \leq \ell \leq L$. For details on multiresolution analysis we refer to [15], or to textbooks [16, 17, 18]. In a more particular way, $\bar{\mathbf{U}}^\ell$ represents a set of cell averages of the discrete data \mathbf{U} at a refinement level ℓ , with a total of $2^{3\ell}$ cells in three space dimensions.

When the data are composed into many levels, for instance, $\bar{\mathbf{U}}^{\ell-1}, \bar{\mathbf{U}}^\ell, \dots, \bar{\mathbf{U}}^L$, it is possible to navigate between each level ℓ by defining prediction and projection operators such as

$$\mathcal{P}^{\ell \rightarrow \ell+1} : \bar{\mathbf{U}}^\ell \longrightarrow \bar{\mathbf{U}}^{\ell+1} \quad (9a)$$

$$\mathcal{P}^{\ell+1 \rightarrow \ell} : \bar{\mathbf{U}}^{\ell+1} \longrightarrow \bar{\mathbf{U}}^\ell. \quad (9b)$$

The projection operator $\mathcal{P}^{\ell \rightarrow \ell+1}$ is exact and unique and it allows us to project the cell averages onto a coarser level. In contrast, the prediction operator performs an approximation of the cell averages in a more refined level, and it is not unique – our choice here is linear interpolation.

From the prediction process, we can compute the interpolation errors $\mathbf{d}^\ell = \mathbf{d}_{i,j,k}^\ell$ at level ℓ . These values are called details or wavelet coefficients and they have the information about the local regularity of the data. The cell averages together with the details at level $\ell + 1$ enable us to obtain the cell averages at level $\ell + 1$, i. e., $\bar{\mathbf{U}}^{\ell+1} \leftrightarrow \{\mathbf{d}^{\ell+1}, \bar{\mathbf{U}}^\ell\}$. More generally, it is possible to obtain the cell averages in the most refined level by the meaning of the cell averages at the coarsest level along with the details of every level between them

$$L \leftrightarrow 0 : \bar{\mathbf{U}}^L \leftrightarrow \{\mathbf{d}^L, \mathbf{d}^{L-1}, \dots, \mathbf{d}^1, \bar{\mathbf{U}}^0\}. \quad (10)$$

The relation above is called multiresolution transform and it ensures the possibility of approximating from data at any level ℓ by the knowledge of the details and cell averages of interest. For the three-dimensional

case, 7 wavelet coefficient are needed to perform the prediction operation of a cell. This procedure is revisited and presented in detail in [19].

The idea of the adaptive multiresolution analysis for cell averages [15] is to decrease the complexity of the multiresolution transform by locally adapting the computational mesh to the structures present in the data. This is made possible when we introduce a threshold operator which determines where to refine the mesh according to the detail information and a threshold parameter ϵ^ℓ . When the details are larger than ϵ^ℓ , the computational mesh needs to be more refined locally; otherwise the mesh can remain coarser. This methodology enables the computational mesh to be efficiently adapted to the numerical solution. In the context of the MHD model, where we have 8 independent variables, the refinement criteria are applied to the maximum wavelet coefficient of these variables, so that the adapted mesh can be understood as a union of the meshes of each variable. In this work we are interested in the level-dependent threshold parameter $\epsilon^\ell = \epsilon^0$, which is modified according to the level of refinement and defined as following

$$\epsilon^\ell = \frac{\epsilon_0}{|\Omega|} 2^{3(\ell-L+1)}, \quad 0 \leq \ell \leq L-1, \quad (11)$$

where $|\Omega|$ is the volume of the domain and ϵ^0 is the initial threshold parameter. Starting from ϵ^0 , the threshold value ϵ^ℓ will increase for increasing ℓ . The number of cells on the finest mesh is defined as 3^{2L} , where L is the finest scale level.

3.3 Time evolution

To advance the solution in time from t^n to t^{n+1} , we use a second-order compact Runge–Kutta scheme. The time steps are computed in terms of the Courant number ν , the space steps, and the fast magnetoacoustic wave. Considering \mathbb{E} as a time evolution operator, the multiresolution transform operator \mathbf{M} and the threshold operator \mathcal{T} , we can summarize the entire process described in this section as

$$\bar{\mathbf{U}}^{n+1} = \mathbf{M}^{-1} \circ \left\{ \mathcal{T} \circ \left[\mathbf{M} \circ \left(\mathbb{E} \circ \bar{\mathbf{U}}^n \right) \right] \right\}, \quad (12)$$

where \circ denotes the composition of functions. The system is completed by choosing suitable initial and boundary conditions.

3.4 CARMEN and FLASH codes

The CARMEN code described in [8] is further developed in order to evaluate the numerical solution of the compressible Navier-Stokes including five more differential equations. An adaptive multiresolution algorithm for cell averages is used in combination with the finite volume discretization. The implementation of the two-dimensional version of the MHD equations, the Riemann solvers, parabolic-hyperbolic divergence cleaning and the variable reconstruction are already published in [5, 7]. In this work, we have implemented the three-dimensional part of the MHD model, including the physical and numerical fluxes.

The FLASH code is an open-access parallel multiphysics multiscale simulation code provided by the Flash Center for Computational Science of the University of Chicago [20]. As an astrophysical code, it can simulate a great variety of problems. Here we are interested in the ideal MHD simulations in the context of the non-adaptive approach of the FLASH code, i. e., the full mesh finite volume simulation, to create a benchmark for our adaptive multiresolution results. We use version 4.3 of the FLASH code.

The time evolution in the FLASH code uses a one-step Hancock scheme and the 8-wave algorithm [21] for divergence cleaning with a finite-volume cell-centered method. The variables are reconstructed with the MC limiter. As the numerical schemes are not the same, we chose the previous settings of the FLASH code to ensure that both codes are second-order accurate.

4 Results

Herein we present and discuss the numerical results of the ideal MHD model for three-dimensional simulations using as initial condition one and two-dimensional Riemann problems. In the first study we compare the results with an exact solution and in the latter we compare the results with a benchmark solution obtained using the FLASH code at the same refinement level. The visualization of the solution is shown along with its mesh and the errors from the comparison between the results.

4.1 One-dimensional Riemann initial condition

In this case, the initial condition comes from the idea of a Riemann problem, which is basically characterized by a function that assumes a determined constant value in each half of the domain. At the interface between the two parts of the domain there is a discontinuity that must be represented accurately. The following setting, which has an exact solution², is used as a benchmark for our results. The initial condition is based on [22] and given in Table 1.

Table 1: One-dimensional Riemann initial condition

	ρ	p	v_x	v_y	v_z	B_x	B_y	B_z
$x \leq 0$	1.08	0.95	1.2	0.01	0.5	$2.0/\sqrt{4\pi}$	$3.6/\sqrt{4\pi}$	$2.0\sqrt{2\pi}$
$x > 0$	1.0	1.0	0.0	0.0	0.0	$2.0/\sqrt{4\pi}$	$4.0/\sqrt{4\pi}$	$2.0\sqrt{2\pi}$

We use the following parameters in the simulations: physical time $t = 0.1$, the Courant number $\nu = 0.3$, $\gamma = 5/3$, $\alpha = 0.4$, refinement level $L = 8$, computational domain $[-0.5, 0.5]^3$, and the threshold parameters $\epsilon^0 = 0.1$.

This initial condition can be extended to two and three-dimensions by tensor products, so that we have constant values of the variables in quadrants and sub-cubes, respectively. This test is useful to verify if the fluxes are being computed properly in every direction. Thus, we simulate this Riemann problem in each of the three directions and compare the results with the exact solution.

In Figure 2 we can see the variables ρ , p , u_x , u_y , B_y , and B_z of the solution of the Riemann problem obtained with the MR code for the threshold $\epsilon^0 = 0.1$ together with the exact solution at $t = 0.1$ and for level $L = 8$, which is computed in the planes xy , yz and zx . The results show that the numerical solution converges to the exact solution for every variable of the model. The fluxes are being estimated in each direction of the domain and, from the figures, we can notice that their computations are reasonable. The one-dimensional projection of the mesh for this problem, composed by 36.6% of the entire mesh, is shown in Figure 3 for each case. The percentage of cells needed for the simulation over time is about 32%.

We can observe that the cells in the most refined level are limited to where the local structures are located. Levels 5 and 6, which are coarser, are representing the constant regions in the solution and more refined levels are needed only close to the localized structures. Table 2 shows the \mathcal{L}^1 , \mathcal{L}^2 and \mathcal{L}^∞ errors computed between the numerical and the exact solution for the one-dimensional Riemann problem for each variable of the MHD model.

²<https://web.mathcces.rwth-aachen.de/mhdsolver/>

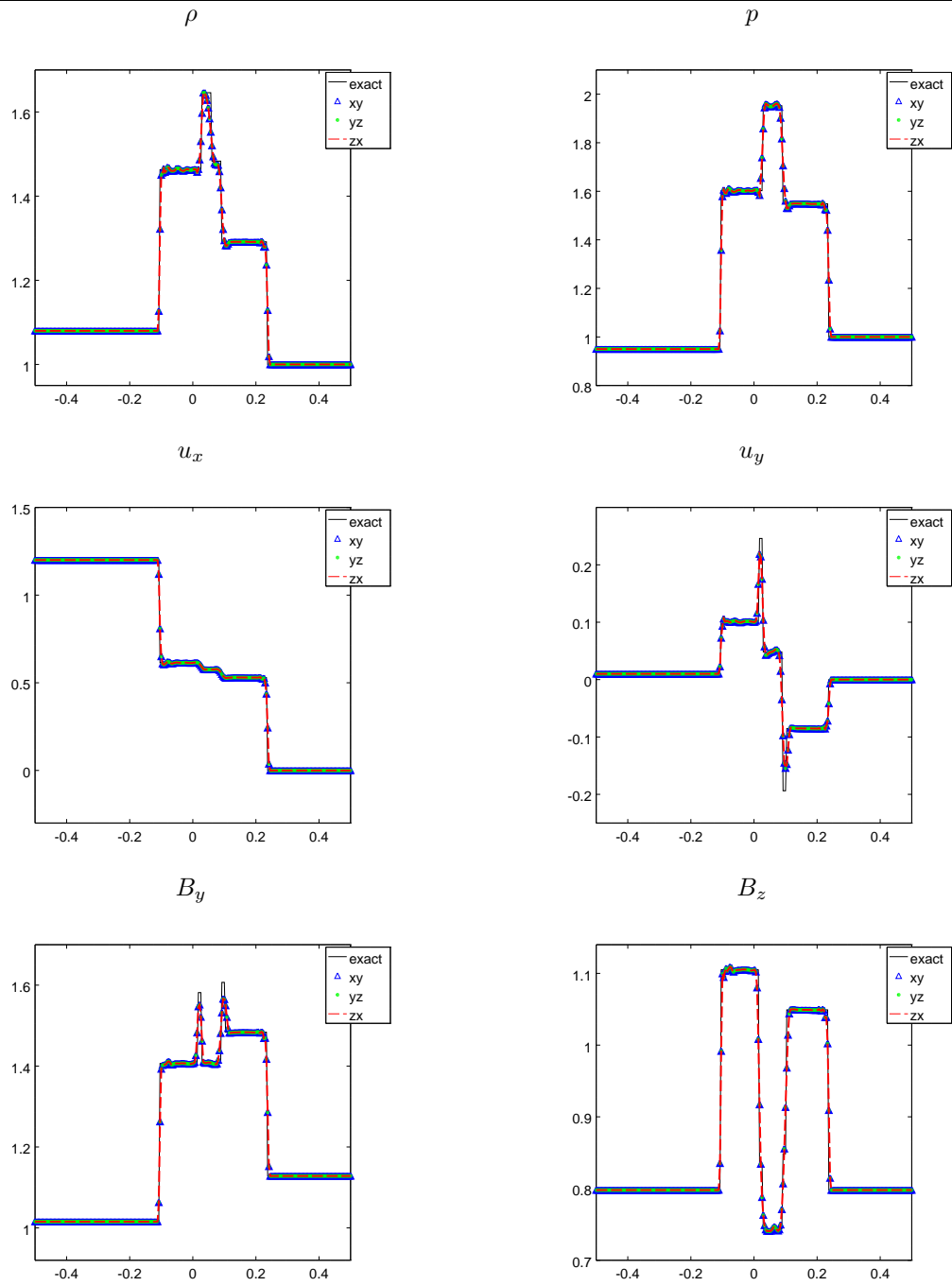


Figure 2: Adaptive multiresolution computations. Shown are ρ , p , u_x , u_y , B_y , and B_z obtained at $t = 0.1$ and $L = 8$, in the xy , yz , and zx planes with $\epsilon^0 = 0.1$, superimposed is the exact solution (in black lines).

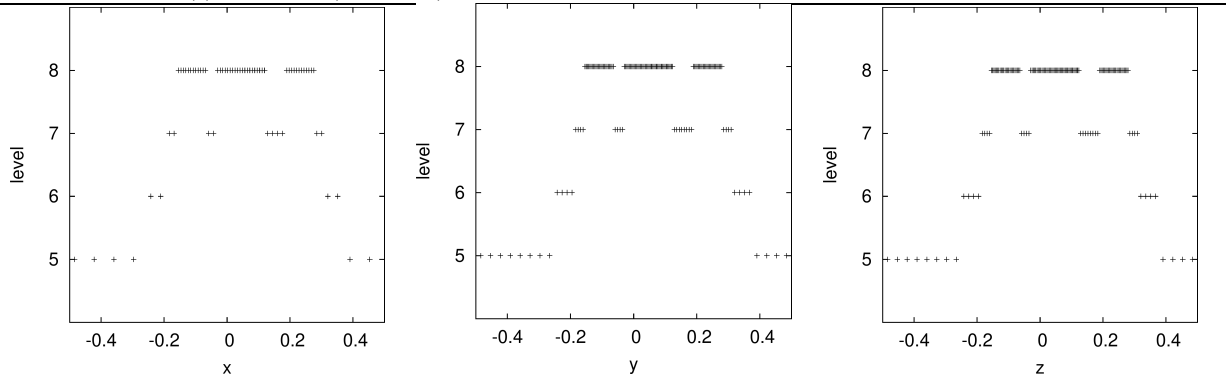


Figure 3: Projection of the mesh in one-dimension in the x , y and z directions.

Table 2: Errors for the one-dimensional Riemann problem in three-dimensions for $L = 8$ and $\epsilon^0 = 0.1$, obtained by comparing the results of the CARMEN code with the exact solution.

Variables	\mathcal{L}^1 error	\mathcal{L}^2 error	\mathcal{L}^∞ error
ρ	$4.5597 \cdot 10^{-3}$	$1.0886 \cdot 10^{-3}$	$1.4087 \cdot 10^{-1}$
p	$6.7027 \cdot 10^{-3}$	$1.8227 \cdot 10^{-3}$	$2.4520 \cdot 10^{-1}$
u_x	$3.5778 \cdot 10^{-3}$	$1.3390 \cdot 10^{-3}$	$2.4291 \cdot 10^{-1}$
u_y	$3.4091 \cdot 10^{-3}$	$8.2139 \cdot 10^{-4}$	$9.6215 \cdot 10^{-2}$
u_z	$2.5392 \cdot 10^{-3}$	$6.6540 \cdot 10^{-4}$	$7.8862 \cdot 10^{-2}$
B_x	$2.8865 \cdot 10^{-8}$	$1.8041 \cdot 10^{-9}$	$2.8865 \cdot 10^{-8}$
B_y	$4.7422 \cdot 10^{-3}$	$1.1685 \cdot 10^{-3}$	$1.5728 \cdot 10^{-1}$
B_z	$3.7892 \cdot 10^{-3}$	$9.8471 \cdot 10^{-4}$	$1.1345 \cdot 10^{-1}$

4.2 Two-dimensional Riemann initial condition

The two-dimensional Riemann initial condition follows the same idea as the one-dimensional Riemann problem. This time the domain is divided into four parts with constant values, analogous to a Cartesian plane. The original problem is in two-dimensions and based on the example presented in [12]. However, in this work we extend the problem to three-dimensions presented in Table 3.

Table 3: Two-dimensional Riemann initial condition

Interval	ρ	p	v_x	v_y	v_z	B_x	B_y	B_z
$x \geq 0$ and $y \geq 0$	1.0304	2.2874	1.4127	-1.0146	-1.0691	0.3501	0.5078	0.1576
$x < 0$ and $y > 0$	1.0000	2.4323	1.7500	-1.0000	0.0000	0.5642	0.5078	0.2539
$x \leq 0$ and $y \leq 0$	1.8887	7.6110	0.1236	-0.9224	0.0388	0.5642	0.9830	0.4915
$x > 0$ and $y < 0$	0.9308	2.1583	1.5639	-0.4977	0.0618	0.3501	0.9830	0.3050

The simulation parameters are: the physical time $t = 0.1$, the Courant number $\nu = 0.3$, $\gamma = 5/3$, $\alpha = 0.4$, refinement level $L = 8$, computational domain $[-1, 1]^3$ and the threshold parameters $\epsilon^0 = 0.08$.

In Figure 4 we show the variable p and the corresponding adaptive mesh, composed by the cell-centers. The presented mesh is a projection of the 3D mesh onto the xy plane and the darker regions mean the refinement is larger. The pressure is shown on Figure 4 (left) and illustrates that the mesh is refined in regions of strong gradients. For the yz and zx plane simulation, the meshes are similar and at $t = 0.1$ there are 40.8% of the cells.

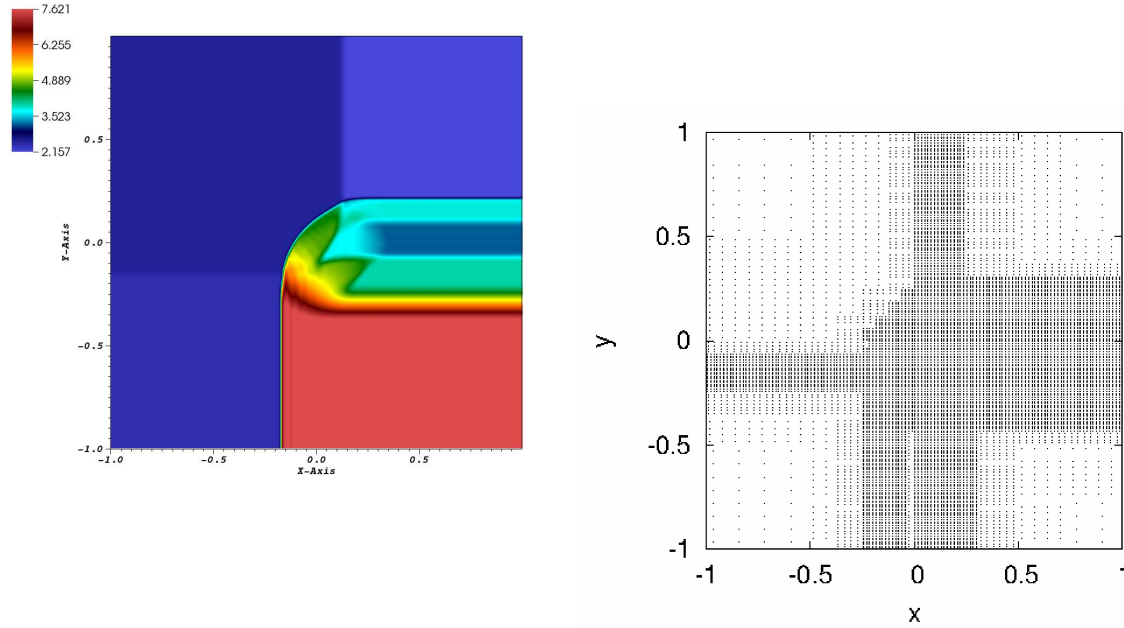


Figure 4: Variable p and the projection of the mesh in two-dimensions ($x - y$ plane).

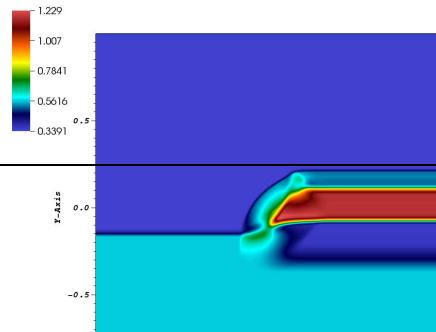
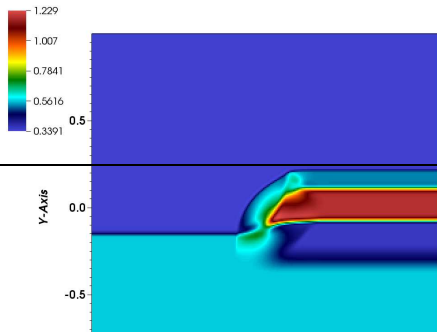
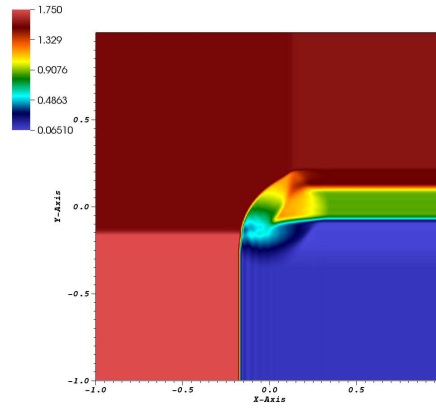
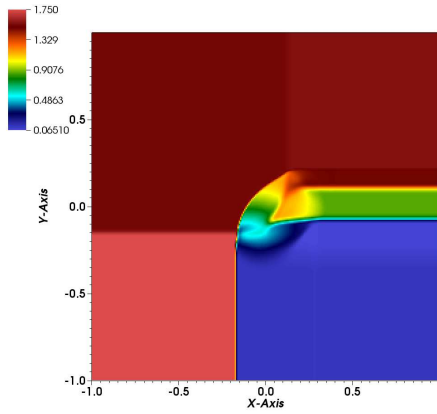
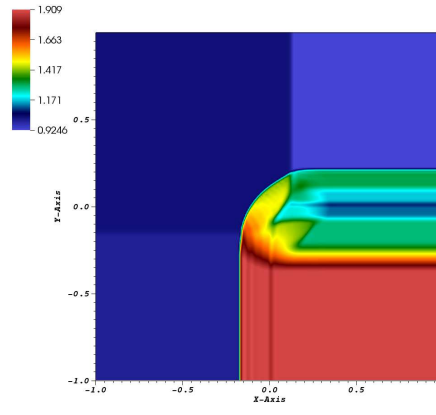
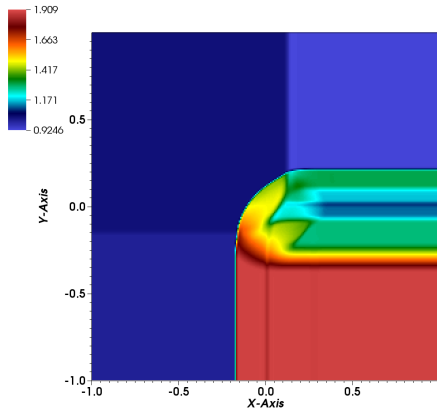
In Table 3 the errors between the CARMEN and FLASH code results quantify the accuracy of each variable of the adaptive multiresolution simulation. The \mathcal{L}^1 , \mathcal{L}^2 , and \mathcal{L}^∞ errors are approximately of the order of 10^{-3} , 10^{-5} , and 10^{-1} , respectively. As expected, we have similar values for the yz and zx simulations.

Table 4: Errors for the two-dimensional Riemann problem in three-dimensions for $L = 8$ and $\epsilon^0 = 0.08$, obtained by comparing the results of CARMEN and FLASH codes.

Variables	\mathcal{L}^1 error	\mathcal{L}^2 error	\mathcal{L}^∞ error
ρ	$1.8604 \cdot 10^{-3}$	$2.9467 \cdot 10^{-5}$	$1.1195 \cdot 10^{-1}$
p	$9.4471 \cdot 10^{-3}$	$1.6666 \cdot 10^{-4}$	$6.6384 \cdot 10^{-1}$
u_x	$2.5411 \cdot 10^{-3}$	$6.3577 \cdot 10^{-5}$	$2.2006 \cdot 10^{-1}$
u_y	$1.7706 \cdot 10^{-3}$	$2.5164 \cdot 10^{-5}$	$7.4709 \cdot 10^{-2}$
u_z	$7.3367 \cdot 10^{-4}$	$2.2670 \cdot 10^{-5}$	$9.9015 \cdot 10^{-2}$
B_x	$1.5709 \cdot 10^{-3}$	$4.9229 \cdot 10^{-5}$	$2.1496 \cdot 10^{-1}$
B_y	$6.9340 \cdot 10^{-4}$	$1.4988 \cdot 10^{-5}$	$6.8804 \cdot 10^{-2}$
B_z	$1.0497 \cdot 10^{-3}$	$2.3591 \cdot 10^{-5}$	$9.6685 \cdot 10^{-2}$

The variables ρ , v_x , and B_x are displayed in Figure 5, obtained with the FLASH (left) and CARMEN (right) codes. The solutions for both cases are similar using the same color scale. The structures located

in the center of the domain are well represented and detached. The discontinuities between the constant regions are sharp in both directions in which they appear. For the problems computed on the yz and zx planes, the behavior of the solution is retained.



5 Conclusions

We have presented three-dimensional simulations of the ideal MHD model with parabolic-hyperbolic divergence cleaning in the context of an adaptive multiresolution approach. It is shown that this type of approach indeed converges towards the expected results. With the MR method, the percentage of cells needed for the achievement of the solution is reduced to 32% for the one-dimensional Riemann problem and about 39% for the two-dimensional Riemann problem. Even with the reductions of cells in the simulation, the accuracy of the solution is well preserved. The presented results are part of a verification of the 3D MHD CARMEN code and we can conclude that the numerical fluxes are computed properly in each direction and the MR approach is efficient to create an adaptive mesh for the studied problems.

Acknowledgments

The authors thank the Brazilian agencies CNPq (306038/2015 – 3, 312246/2013 – 7), FINEP/CT-INFRA (01120527 – 00), and FAPESP (2015/50403 – 0, 2015/25624 – 2) projects for financial support. We thank Dr. Olivier Roussel for developing the original Carmen Code and for the fruitful scientific discussions. We also thank Eng. V. E. Menconi for his helpful computational assistance. The Flash software used in this work was in part developed by the DOE NNSA-ASC OASCR Flash Center at the University of Chicago. AKFG thanks CNPq for the PhD scholarship (project 141741/2013 – 9). MD thankfully acknowledges financial support from Ecole Centrale Marseille. KS acknowledges financial support from the ANR-DFG, grant AIFIT, and support by the French Research Federation for Fusion Studies within the framework of the European Fusion Development Agreement (EFDA).

References

- [1] A. Otto, “3d resistive mhd computations of magnetospheric physics,” *Computer Physics Communications*, vol. 59, no. 1, pp. 185–195, 1990.
- [2] R. Touma and P. Arminjon, “Central finite volume schemes with constrained transport divergence treatment for three-dimensional ideal MHD,” *Journal of Computational Physics*, vol. 212, no. 2, pp. 617–636, 2006.
- [3] D. Lee, “A solution accurate, efficient and stable unsplit staggered mesh scheme for three dimensional magnetohydrodynamics,” *Journal of Computational Physics*, vol. 243, pp. 269–292, 2013.
- [4] J. Skála, F. Baruffa, J. Büchner, and M. Rampp, “The 3d mhd code goemhd3 for astrophysical plasmas with large Reynolds numbers-code description, verification, and computational performance,” *Astronomy & Astrophysics*, vol. 580, p. A48, 2015.
- [5] A. K. F. Gomes, “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.
- [6] M. O. Domingues, A. K. F. Gomes, S. M. Gomes, O. Mendes, B. Di Pierro, and K. Schneider, “Extended generalized Lagrangian multipliers for magnetohydrodynamics using adaptive multiresolution methods,” *ESAIM Proceedings*, vol. 43, pp. 95–107, 2013.
- [7] A. K. F. Gomes, M. O. Domingues, K. Schneider, O. Mendes, and R. Deiterding, “An adaptive multiresolution method for ideal magnetohydrodynamics using divergence cleaning with parabolic-hyperbolic correction,” *Applied Numerical Mathematics*, vol. 95, pp. 199–213, 2015. Fourth Chilean Workshop on Numerical Analysis of Partial Differential Equations (WONAPDE 2013).

- [8] O. Rousell, K. Schneider, A. Tsigulin, and H. Bockhorn, “A conservative fully adaptative multiresolution algorithm for parabolic PDEs,” *Journal of Computational Physics*, vol. 188, pp. 493–523, 2003.
- [9] J. U. Brackbill and D. C. Barnes, “Note: The effect of nonzero $\nabla \cdot \mathbf{B}$ on the numerical solution of the magnetohydrodynamic equations,” *Journal of Computational Physics*, vol. 35, no. 3, pp. 426–430, 1980.
- [10] G. Tóth, “The $\nabla \cdot \mathbf{B}$ constraint in shock-capturing magnetohydrodynamics codes,” *Journal of Computational Physics*, vol. 161, pp. 605–652, 2000.
- [11] G. Tóth, B. Van der Holst, I. V. Sokolov, D. L. De Zeeuw, T. I. Gombosi, F. Fang, W. B. Manchester, X. Meng, D. Najib, K. G. Powell, *et al.*, “Adaptive numerical algorithms in space weather modeling,” *Journal of Computational Physics*, vol. 231, no. 3, pp. 870–903, 2012.
- [12] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg, “Hyperbolic divergence cleaning for the MHD equations,” *Journal of Computational Physics*, vol. 175, pp. 645–673, 2002.
- [13] A. Mignone and P. Tzeferacos, “A second-order unsplit godunov scheme for cell-centered mhd: The ctu-glm scheme,” *Journal of Computational Physics*, vol. 229, no. 6, pp. 2117–2138, 2010.
- [14] T. Miyoshi and K. Kusano, “A multi-state HLL approximate Riemann solver for ideal magnetohydrodynamics,” *Journal of Computational Physics*, vol. 208, pp. 315–344, 2005.
- [15] A. Harten, “Multiresolution representation of data: a general framework,” *SIAM Journal of Numerical Analysis*, vol. 33, no. 3, pp. 385–394, 1996.
- [16] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.
- [17] Y. Meyer, *Wavelets and applications: proceedings of the international conference*. Marseille, France: Paris: Masson; Berlin: Springer-Verlag., May 1999.
- [18] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.
- [19] M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider, “Adaptive multiresolution methods,” *ESAIM Proceedings*, vol. 34, pp. 1–96, 2011.
- [20] A. Dubey, K. Antypas, A. C. Calder, C. Daley, B. Fryxell, J. B. Gallagher, D. Q. Lamb, D. Lee, K. Olson, L. B. Reid, *et al.*, “Evolution of flash, a multi-physics scientific simulation code for high-performance computing,” *The International Journal of High Performance Computing Applications*, vol. 28, no. 2, pp. 225–237, 2014.
- [21] K. G. Powell, P. L. Roe, T. J. Linde, T. I. Gombosi, and D. L. D. Zeeuw, “A Solution-Adaptative Upwind Scheme for Ideal Magnetohydrodynamics,” *Journal of Computational Physics*, vol. 154, pp. 284–309, 1999.
- [22] S. Li, “An HLLC Riemann solver for Magnetohydrodynamics,” *Journal of Computational Physics*, vol. 203, pp. 344–357, 2005.