Magnetohydrodynamics adaptive solvers in the AMROC framework for space plasma applications

Müller Moreira Lopes, Margarete Oliveira Domingues, Ralf Deiterding, and Odim Mendes

Abstract Plasma disturbances affect satellites, spacecraft and can cause serious problems to telecommunications and sensitive sensor-systems on Earth. Considering the huge scale of the plasma phenomena, data collection at individual locations is not sufficient to cover this entire relevant environment. Therefore, computational plasma modelling has become a significant issue for space sciences, particularly for the near-Earth magnetosphere. However, the simulations of these disturbances present many physical as well as numerical and computational challenges. In this work, we discuss our recent magnetohydrodynamic solver, realised within the MPI-parallel AMROC (Adaptive Mesh Refinement in Object-oriented C++) framework, in which particular physical models and automatic mesh generation procedures have been implemented. A performance analysis using a selection of significant space applications validates the solvers capabilities and confirms the technical importance of our approach.

Müller Moreira Lopes
National Institute for Space Research (INPE), Av.dos Astronautas 1758, Jardim da Granja, São José dos Campos, São Paulo, Brazil, e-mail: muller.lopes@inpe.br,mullermslopes@gmail.com

Margarete Oliveira Domingues
INPE, Av.dos Astronautas 1758, Jardim da Granja, São José dos Campos, São Paulo, Brazil, e-mail: margarete.domingues@inpe.br

Ralf Deiterding
Aerodynamics and Flight Mechanics Research Group, University of Southampton, SO16 7QF, United Kingdom, e-mail: r.deiterding@soton.ac.uk

Odim Mendes
INPE, Av.dos Astronautas 1758, Jardim da Granja, São José dos Campos, São Paulo, Brazil, e-mail: odim.mendes@inpe.br

1 Introduction

The number of studies on space plasmas has increased during the last decades, driven by the significant effects that space plasma can have on sensitive electro-electronic technologies. The road map for 2015–2025 commissioned by the Committee on Space Research (COSPAR) and the International Living With a Star working group (ILWS) [43] describes damaging influences of space phenomena on current technologies and infrastructure, implying a high economic cost that – albeit not completely measurable – has a significant impact on the world economy [21].

There are significant international efforts to produce space weather forecasting systems in order to anticipate when very intense solar events can occur and how they can interact with the Earth or human-built space equipment. The global simulations of the Magnetohydrodynamics (MHD) model are a fundamental part of such forecasting systems, especially to clarify processes, quantify, and even in the next years predict a complete phenomenology of the interaction of plasmas with the Earth's magnetised and ionised atmosphere, as described, for instance, in [47].

In recent years, our research group has been contributing to such efforts by developing a high performance numerical MHD solver to simulate the near-Earth environment using the solar wind data as boundary conditions. This solver is being developed under the AMROC (Adaptive Mesh Refinement in Object-oriented C++) framework [17], which implements a patch-Structured Adaptive Mesh Refinement (SAMR) method [4] with the parallel strategy proposed in [14] using the Message Passing Interface (MPI) protocol.

Snapshots of these developments are given in [36], which introduces our solver by presenting the results of classical ideal MHD benchmarks using the adaptive and parallel strategies for both two-dimensional and three-dimensional formulations, and in [19], where we presented some improvements concerning the adaptive criteria of the SAMR method by using wavelet-based techniques. A core finding of these publications is that the Multiresolution (MR) approach, as it is mathematically more rigorous, leads to an adaptive mesh well fitted to the structures of solutions in the MHD context and to an improvement in the overall computational time. Furthermore, this approach is more suitable for the numerical approaches to solve the magnetic field divergence problem presented in MHD simulations.

In recent years the number of studies in computational plasma modeling has been increasing; many other MHD codes were developed for a variety of applications. For instance, [22] presents an extensive review of global magnetosphere models, codes and numerical methods. In particular the RAMSES code [23] was developed in Saclay to study large scale structure and galaxy formation, however it is now a rather flexible package to be used for general purpose simulations in self-gravitating fluid dynamics. It is written in Fortran 90 with extensive use of the MPI library and built on a grid-based hydro solver with adaptive mesh refinement. The underlying data structure is the so called fully threaded tree. As opposed to patch-based SAMR, cells are refined on a cell-by-cell basis: it is therefore called a tree-based AMR as described in [23]. A comparison of advantages and disadvantages of patch-based and tree-based AMR is presented in [24], stating that the patch-based SAMR codes provide a better memory layout and a simpler geometry at the cost of the refinement of unnecessary cells and the extra memory consumption due to the patch manager. Another widely used MHD solver is the ATHENA code [44]. It is a grid-based code for astrophysical magnetohydrodynamics (MHD). It was developed primarily for studies of the interstellar medium, star formation, and accretion flows. Athena has been made freely available to the community in the hope that others may find it useful. It uses Static Mesh Refinement (SMR), as decribed in [44]. This approach consists of using a fixed mesh that is more refined in some regions of the domain. Besides being faster, this approach is not as flexible as the full SAMR algorithm in the sense that, for many problems, the structures of interest may be formed dynamically or transported through the domain.

This paper aims to present the new milestones reached in the development of our MHD solver. In particular, this work develops the resistive formulation of the MHD equations using a set of physical benchmark problems that occur due to the resistive effects. Furthermore, we introduce an early two-dimensional MHD formulation for the Earth magnetosphere based on [39, 38, 40]. After the simulated magnetosphere converges to steady-state conditions, the model is implemented such that the boundary condition containing typical values for the solar wind proceeds with an evolution using actual interplanetary measurements, for instance, data collected from the OMNI web service, NASA.[1]

The outline of this work briefly presents the AMROC framework in Section 2, explaining the numerical, computational and design decisions of the software development process that are contributing to the success of our ideal MHD solver, as discussed in [36, 19]. Then, the numerical and implementation aspects of the ideal and resistive MHD solvers are presented in Section 3. In Section 4, the content refers to some fundamental experiments related to physical phenomena that occur in space weather. At last, in Section 5, we discuss the lessons learned and the next development steps.

2 AMROC

The compressible MHD equations are a system of nonlinear hyperbolic partial differential equations. Considering the vector of conservative physical quantities $\mathbf{q}$, these equations can be written as a conservation law, i.e.,

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbb{F}\left(\mathbf{q}\right) = \mathbf{S}(\mathbf{q}) , \tag{1}$$

where $\mathbb{F}$ and $\mathbf{S}$ are the flux function and the source term respectively. A suitable approach to simulate this type of model is the finite volume scheme [33]. This method consists of discretising the physical domain of the problem into cells or control volumes, so that each cell contains an average value for its coverage. Then, each cell

[1] OMNI web service, NASA: `https://omniweb.gsfc.nasa.gov/`

average is evolved in time according to the flux between its adjacent cells. Note that in here, simulations are always performed inside rectangular physical domains. This choice allows particular optimisations in comparison with the algorithms required for unstructured meshes, such as the implementation of a single numerical scheme routine, independent of the refinement.

During the discretisation process, the choices of proper refinements are challenging. An overly coarse mesh may cause the solution to be not adequately represented, especially in cases that contain localised structures or steep gradients, causing loss of information. On the other hand, an exaggerated refinement leads to a considerable amount of unnecessary computations, wasting a lot of computational time and memory. In this context, the use of adaptive techniques proposes to overcome these limitations. These techniques maximise the efficiency of the simulation by using an adaptive mesh, which is more refined in the regions where the localised structures are present and is coarser in the smooth solution regions.

2.1 Adaptive meshes

The MHD solver developed for this work uses an SAMR method to construct the adaptive meshes. The work of Berger and Oliger [5, 3] was the first to introduce this method. Subsequently, Berger and Colella [3, 4] proposed a simpler version, in which every mesh of the hierarchy must be aligned with cells of the next coarser level, allowing simpler interpolation operations. Bell, Berger and Saltzman [2] demonstrated that this version can be more efficient, especially with vector and super-scalar computers.

The strategy of the SAMR methods to construct adaptive meshes is based on refinement criteria that measure the local smoothness of the solution in every mesh element. If the result of the criteria exceeds a predetermined value ϵ, the mesh element is flagged for refinement. After that, the flagged elements are overlaid by sub-meshes, i.e. patches, of finer cells that are refined by a factor of r. Note that the presently implemented multiresolution criterion only uses a refinement factor of $r = 2$; however, AMROC principally permits arbitrary integer values for other refinement criteria.

The process of overlaying more refined sub-meshes over coarser meshes produces a hierarchy of embedded level domains, i.e. the domain covered by a finer refinement level is also covered by the next coarser level. This hierarchy can be expressed as the sequence of meshes $\mathcal{M}^\ell$, where ℓ is associated with the refinement levels $\ell = 0, 1, \ldots, \mathrm{L}$. Each mesh $\mathcal{M}^\ell$ has its spatial mesh widths denoted as Δx^ℓ, Δy^ℓ and Δz^ℓ so that they present a constant ratio r between adjacent levels. Considering that the meshes in this hierarchy are embedded, the physical domain covered by each mesh $\mathcal{M}^\ell$ follows the property $\mathcal{M}^{\mathrm{L}} \subset \mathcal{M}^{\mathrm{L}-1} \subset \cdots \subset \mathcal{M}^0 = \Omega$, where the base mesh $\mathcal{M}^0$ cover the entire physical domain Ω. Each one of these meshes is divided into a set of non overlapping rectangular submeshes $\mathcal{M}^\ell_m$ so that $\mathcal{M}^\ell := \cup_{m=1}^{M_\ell} \mathcal{M}^\ell_m$ with

$\mathcal{M}^{\ell}_{m_1} \cap \mathcal{M}^{\ell}_{m_2} = \emptyset$, $m_1 \neq m_2$, where M_ℓ is the number of submeshes, or patches, used to represent the mesh $\mathcal{M}^{\ell}$.

One of the advantages of this mesh hierarchy is allowing a single implementation of the numerical scheme to be executed on each patch, independent of the refinement level. Thereby, the time evolution process can be performed for every patch individually. However, to compute fluxes along the boundaries of every patch requires a solution of the cells in adjacent patches. This restriction compromises the patch independence to perform the time evolution. In order to overcome this limitation, the patch structure are complemented with extra auxiliary cells, called ghost cells, at their boundaries, allowing the boundary values to be stored in the same data structure as the submesh.

The values in the ghost cells must be set or updated before and during the time evolution. For that, the ghost cells are divided into three cases regarding their position. At the physical boundary, the ghost cells are updated according to the problem boundary conditions. If there exists an adjacent patch on the same level, the ghost cells are updated by copying the solution from the neighbour. If the adjacent patch is however coarser, the ghost cells are interpolated from the solution in the coarser level using a multi-linear interpolation. In order to ensure discrete conservation in the fluxes between patches of different levels, the SAMR approach eventually replaces the coarse-cell fluxes by averaged fine-mesh fluxes, hence modifying the numerical stencil on the coarse mesh, cf. [14, 15] for AMROC-specific implementation details.

Wavelet coefficients as refinement indicator

Wavelet theory shows that the decay of the wavelet coefficients estimates the local regularity of the solution [11]. Therefore we can use such coefficients to predict where we could not have an adequate local approximation and consequently, we need to improve the refinement. This can be used to determine dyadic refinement (i.e. $r = 2$) as these coefficients indicate regions of steep gradients or discontinuities [37, 20]. In this context, our proposed MHD solver implements ideas based on the adaptive multiresolution method introduced by [27] to use as the refinement criteria. For that, the solver uses the multiresolution operations to predict the expected solution at a finer scale based on the solution from a coarser scale. Then, this prediction is compared with the actual solution. Besides this criterion, the MHD solver also allows using the criteria discussed in [15, 19].

Our underlying idea of applying multiresolution techniques for mesh adaptation is based on representing the numerical solution in two or more different resolution levels. Compression of the number of mesh cells, corresponding to coarsening the mesh locally, can be obtained by checking what happens between subsequent mesh resolutions [18]. For instance, for a discrete solution of a FV discretisation we consider as initial cell average data $\overline{\mathbf{Q}}^{\ell+1}$ at level $\ell + 1$. The transformation of these data $\overline{\mathbf{Q}}^{\ell+1}$ into an equivalent multiscale representation in one decomposition level is

$$\overline{\mathbf{Q}}^{\ell+1} \underset{\text{prediction}}{\overset{\text{projection}}{\rightleftharpoons}} \overline{\mathbf{Q}}^{\ell+1}_{MR} = \{\overline{\mathbf{Q}}^{\ell}\} \cup \{\mathbf{d}^{\ell+1}\},$$

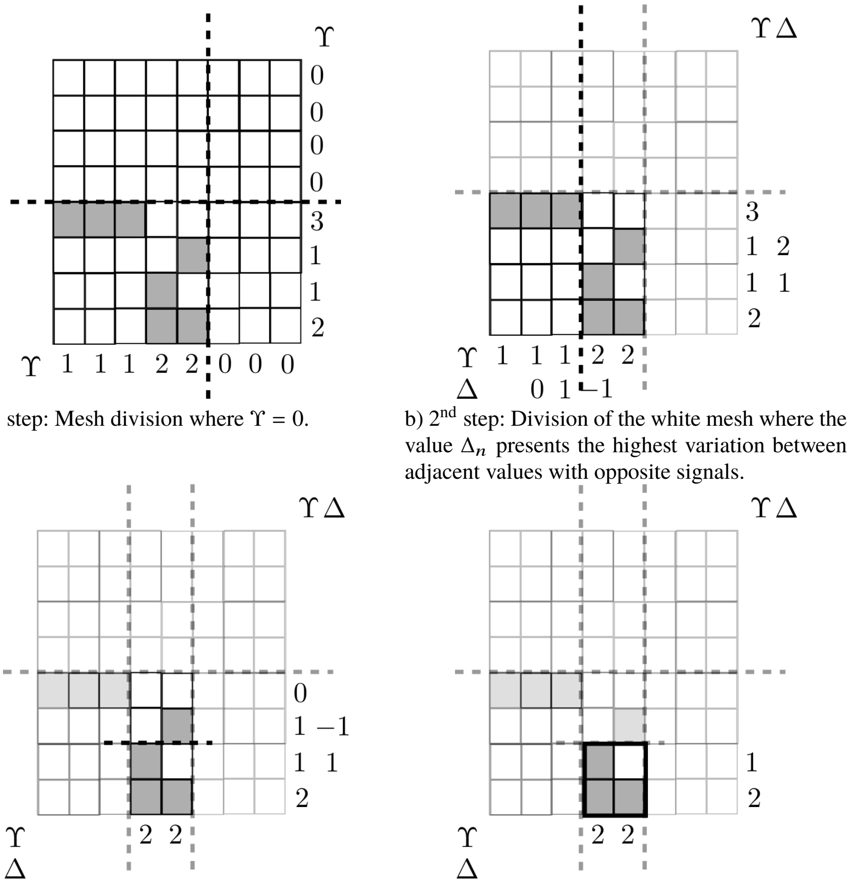
where the set $\mathbf{d}^{\ell+1}$ contains the information between the two consecutive levels ℓ and $\ell + 1$, and $\overline{\mathbf{Q}}^{\ell}$ stores a smoothed version of the original numerical solution $\overline{\mathbf{Q}}^{\ell+1}$. The data at the highest resolution level are transformed into a set of coarser scale approximations plus a series of prediction errors corresponding to wavelet coefficients $\mathbf{d}^{\ell}$ in a multiresolution analysis. In order to perform the MR method with finite volume data, operations for projection and prediction are required, where the cell values are local averages.

Patch creation

In SAMR methods, once the refinement criteria flag the coarser cells that require refinement, these cells are clustered into blocks by using a dedicated algorithm. Then, these blocks are used to construct patches of a finer refinement level. The AMROC framework uses the clustering algorithm proposed in [2], which was inspired by image detection techniques. This clustering algorithm is illustrated in Figure 1. Let Υ_i be the number of flagged cells, i.e. *signatures*, in the i-th row or column of cells on the current mesh $\mathcal{M}^{\ell}$. As the first step, this method splits the domain in every row and column where $\Upsilon_i = 0$. In the second step, the cuts are placed where the discrete second derivative $\Delta = \Upsilon_{i+1} - 2\,\Upsilon_i + \Upsilon_{i-1}$ crosses zero, starting from the steepest zero crossing and then using the lowest ones recursively. This step is repeated until the ratio among all cells and flagged ones in every new submesh is above the prescribed clustering ratio ϑ. In principle $\vartheta \in [0.00, 1.00]$, however, in practice, typical values for $\vartheta \in [0.80, 0.99]$.

Time evolution strategy

As the adaptive mesh is defined, the time evolution for simulations with adaptive meshes presents minor challenges regarding stability and conservation. Obviously, the time step parameter Δt needs to satisfy the Courant–Friedrichs–Lewy (CFL) condition in every patch. In this context, the time evolution process using adaptive meshes may be performed with two different approaches [15]. One is based on a global time-stepping where the patches of every level are updated with a Δt that satisfies the CFL condition on the finest meshes. The other strategy is a refinement-based recursive time-stepping where Δt_{ℓ} varies between levels in the same proportion as their spatial refinement. This latter strategy requires the solution in patches of different refinement levels to be available at different time instants, which is handled in the SAMR method by constructing time-interpolated coarse level data as a boundary condition for interior fine level patches.



a) 1st step: Mesh division where $\Upsilon = 0$.

b) 2nd step: Division of the white mesh where the value Δ_n presents the highest variation between adjacent values with opposite signals.

c) Recursive execution of the 2nd step over the new subdivisions.

d) Stop criteria: Percentage of flagged cells in the subdivision surpass the predefined clustering ratio ϑ

Fig. 1 Cell clustering algorithm. The value Υ_i associated with each row and column is defined as the number of flagged cells in that row or column, and $\Delta_i = \Upsilon_{i+1} - 2\Upsilon_i + \Upsilon_{i-1}$.

2.2 Implementation aspects

AMROC is a freely available framework[2] that uses object-oriented programming concepts in the C++ language to support the numerical simulation of partial differential equations using adaptive methods. In this framework, presented in [15], the SAMR core contains about 46, 000 lines of code C++ and around 6, 000 lines of code for visualisation and data conversion routines. Besides being written in the C++ language, the framework uses the FORTRAN language to perform mesh operations, such as prolongation and restriction, due to its better performance in mathematical

[2] AMROC webpage: `http://www.vtf.website/asc/wiki/bin/view/Amroc`.

computations. The AMROC framework uses a space-filling curve to implement a dynamic re-partitioning algorithm and to redistribute the workload among the processes in the adaptive cases. In here, this load balancing operation is carried out after each time step on level 0. AMROC is pursuing a rigorous domain decomposition strategy, in which the increased computational expense on higher refinement levels in the patch-based AMR algorithm is considered when evaluating parallel workload. However, only units of the smallest resolution corresponding to a cell on level 0 can be considered [14]. This approach simplifies the implementation and reduces the expense of the partitioning algorithm, but can lead to slight load imbalances on deep hierarchies. The algorithm used for partitioning is always a multi-dimensional space-filling curve [13, 15].

3 MHD modelling

The study of MHD phenomena presents a series of demands concerning their physics, such as the formation of instabilities, discontinuities and shocks in the physical quantities and the diversity of scales with which these behaviours may occur [32, 6]. MHD modelling describes the behaviour of a single, non-viscous, compressible and conducting fluid under a magnetic field. This model is applied in problems in which the plasma has macroscopic force balance, equilibrium and dynamics. In the scope of space sciences, the MHD formulation describes phenomena such as the Earth magnetosphere, the solar wind, the Heliosphere and many instabilities in plasma that occur in those regions.

The MHD model describes the plasma dynamo using the variables ρ, $\mathbf{u}$, $\mathbf{B}$, p, $\mathcal{E}$ and η, corresponding to density, velocity, magnetic flux, pressure, total energy and resistivity, respectively. These variables are modelled by combining the Euler equations and the Maxwell equations [32], obtaining a set of eight non-linear partial differential equations. In order to simplify the representation of these equations for the modelling purpose, they are rewritten in a non-dimensional form so that the magnetic permeability yields the identity $\mu = 1$ [25], obtaining:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{2a}$$

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

$$\frac{\partial \mathcal{E}}{\partial t} + \nabla \cdot \left[\left(\mathcal{E} + p + \frac{\mathbf{B}\cdot\mathbf{B}}{2}\right)\mathbf{u} - (\mathbf{u}\cdot\mathbf{B})\,\mathbf{B} + [\eta\,(\nabla\times\mathbf{B})] \times \mathbf{B}\right] = 0, \tag{2c}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot \left[\mathbf{u}\mathbf{B} - \mathbf{B}\mathbf{u} + \eta\left((\nabla\mathbf{B})^T - \nabla\mathbf{B}\right)\right] = 0. \tag{2d}$$

This system is completed by the definition of total energy, which is the combination of the hydrodynamic and magnetic energies, i.e.,

$$\mathcal{E} = \frac{p}{\gamma - 1} + \rho \frac{\mathbf{u} \cdot \mathbf{u}}{2} + \frac{\mathbf{B} \cdot \mathbf{B}}{2}, \tag{3}$$

where γ is the adiabatic index. Furthermore, the plasma must satisfy the Gauss law for the magnetism $\nabla \cdot \mathbf{B} = 0$, i.e., the divergence over the magnetic field is constant and zero over time and no monopole is allowed (since it is a nonphysical behaviour).

Numerical aspects

The numerical simulation of the MHD equations presents several inherent challenges, for instance, the operator $\nabla \cdot (\nabla \times \cdot)$ over a vector quantity is zero, what implies, in our study, that $\partial_t \, (\nabla \cdot \mathbf{B}) = 0$. So, $\nabla \cdot \mathbf{B}$ is a constant value, but as earlier mentioned, there is no monopole. However, in many numerical methods, due to numerical approximation errors, the term containing $\nabla \cdot (\nabla \times \cdot)$ does not indeed result in zero. Hence $\nabla \cdot \mathbf{B} \neq 0$, so that the MHD system does not satisfy Gauss' law as expected. The loss of this constraint causes spurious behaviours in the numerical solution, creating magnetic monopoles. This phenomenon leads to numerical instabilities as discussed, for instance, in [7, 46], and more recently in [29].

To deal with this problem, the AMROC-MHD module implements the Generalised Lagrangian Multiplier (GLM) formulation for the MHD equations [12]. In general, GLM methods are used to maximise or minimise a function under some constraints. In this context, the GLM method is used to maximise the induction equation while imposing $\nabla \cdot \mathbf{B} = 0$. This is done by coupling a differential operator $\mathcal{D}$ to the Gauss's Law so that $\mathcal{D}(\psi) + \nabla \cdot \mathbf{B} = 0$. Then, the solution ψ is coupled to the induction equation, obtaining

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot \left[\mathbf{u}\mathbf{B}^{\mathbf{T}} - \mathbf{B}\mathbf{u}^{\mathbf{T}} + \eta \left((\nabla \mathbf{B})^T - \nabla \mathbf{B} \right) + \psi \mathbf{I} \right] = \mathbf{0}, \tag{4}$$

In particular, the MHD simulations in this work use the parabolic-hyperbolic divergence cleaning approach as proposed in [12] and updated in [34]. This correction is characterised by the operator $\mathcal{D}(\psi) = \frac{1}{c_p^2}\psi + \frac{1}{c_h^2}\frac{\partial \psi}{\partial t}$, with c_p and $c_h \in (0, \infty)$, where the parameter $c_h = \max\left[\sigma \frac{\Delta h}{\Delta t}, \max\left(|u_i| \pm c_f\right)\right]$, with Δh being the minimal value of the mesh sizes in each direction, σ the Courant number, u_i the velocity of the i-th component, c_f the fast magneto-acoustic wave of the MHD model, and the c_p value is defined in terms of the parameter $\alpha_p = \frac{\Delta h \, c_h}{c_p^2}$ for $\alpha_p \in [0, 1]$. Applying this operator into the modified Gauss' Law, the equation that describes the evolution of ψ is obtained as

$$\frac{\partial \psi}{\partial t} + c_h^2 \nabla \cdot \mathbf{B} = -\frac{c_h^2}{c_p^2}\psi. \tag{5}$$

The choice of this operator results in a method that transports and diffuses the components of $\nabla \cdot \mathbf{B}$ to the boundaries [12]. Besides not eliminating these components completely, this method is capable of maintaining the accuracy and the stability of the solution at a very low computational cost.

Moreover, the AMROC-MHD module also implements the Extended GLM-MHD formulation (EGLM-MHD), as described in [12], that includes Powell's source terms into the GLM-MHD model. However we will not discuss EGLM experiments herein as the GLM approach presents solutions that are mathematically more rigorous in the sense of conservation laws.

The discussed GLM-MHD system is completed by suitable initial and boundary conditions as presented in the numerical experiments section. Note that the variable ψ is always initialised as zero for the entire domain.

The AMROC-MHD solver implements the HLL [28] and the HLLD [35] fluxes using a two-stage second-order-accurate Runge–Kutta method to perform the flux computations of the MHD equations. Furthermore, the AMROC-MHD module uses MUSCL (Monotone Upstream-Centered Scheme for Conservation Laws) [48] as a high-resolution scheme. This approach is based on the usage of a slope limiter, which consists of piece-wise linear reconstructions to extrapolate each variable $\mathbf{q}$ of the solution to the left and right boundaries of the cell. These extrapolations are performed as presented in [45].

Besides the GLM formulation, in order to ensure that the solution does not develop spurious oscillations around discontinuities or shocks, the AMROC-MHD module implements the following slope limiters functions to be used in a MUSCL scheme: Minmod [41], Monotonized Central (MC) [51], Superbee [41], van Albada [49], van Leer [50], and Koren [31].

4 Experiments and discussions

This section presents the results of the adaptive simulations of two benchmark cases for a three-dimensional ideal MHD and a two-dimensional resistive MHD model, respectively. These benchmarks represent significant phenomena that appear in the physics of the space environment involving the Earth magnetosphere model. The experiments aim to quantify the scheme accuracy in the $\mathbb{L}_1$-norm, the memory compression and the CPU-time gain concerning a non-adaptive mesh with the same refinement as the finest level, as discussed in [18, 16]. Finally, a more complex configuration setup involving the Earth magnetosphere is discussed.

All experiments are conducted using a Cartesian mesh, the HLLD numerical flux introduced in [35], and the MinMod limiter as discussed, for instance, in [45]. The computations were run in parallel using nodes of a recent GNU/LINUX computer cluster that provides 20 cores with shared memory per node.

4.1 Magnetic shock-cloud (MSC)

The magnetic shock-cloud problem is a benchmark test that verifies the performance of the numerical scheme when dealing with super-fast flows [46]. It describes the disruption of a high-density magnetic cloud by a strong shock wave. For that, an advancing plasma is considered, which causes a shock with a stationary state that contains a high-density cloud. This is a conceptual problem, not yet computationally exploited in the space science literature. These simulations are performed inside the computational domain $[0, 1]^3$ with outlet boundaries, using the GLM factor $\alpha_p = 0.4$ and $\eta = 0$ (ideal MHD case). The time steps are performed under the Courant number $\sigma = 0.4$ until the final time $t = 0.06$. The initial state of both the advancing (delimited by $x < 0.05$) and steady ($x > 0.05$) plasma regions are described in Table 1 for the adiabatic constant $\gamma = \frac{5}{3}$, with non-dimensional quantities compatible with the defined MHD model. The density solution in the steady plasma ρ^0 is given by value $\rho^0 = 10$ if the coordinates are inside the cloud with centre in $(0.25, 0.5, 0.5)$ and radius $r = 0.15$. Otherwise, this density is set as $\rho^0 = 1$. Figure 2 shows a slice representation of the density initial condition configuration and the solutions at the instants $t = 0.03$ and $t = 0.06$ alongside with the respective adaptive meshes and the mesh distribution among the 48 processors used for these simulations.

The figure containing the refinement of the adaptive mesh is interpreted such that the blue regions of the domain represents the coarsest scale, while the red regions represent the most refined scale. The adaptive mesh localises structures in the solution, such as the bow shock, the centre of the explosion area and the tail. The visualisation of the mesh distribution per processor is a complicated task in this three-dimensional case. However, we can roughly estimate the form of this distribution using planar cuts, cf. right graphic of Figure 2, so that each colour represents the sub-domain evolved by each processor. The workload balance algorithm is expected to create small sub-domains in the most refined regions, while the coarser regions are evolved using less processors. This can be observed in the initial condition, where the refinement concentrates in the shock wave and in the sphere borders and hence more processors are being used in these regions, in contrast with the region with $x > 0$, which is predominantly being evolved by the same processor. In the instant $t = 0.03$, the processors maintained their concentration around the cloud region and the yz-plane, where the shock wave is located, while the region $x > 0$ is still evolved by few processors. At the final instant $t = 0.06$, the processors are more spread into the domain, but a concentration is still visible in the cloud and tail regions. These graphics indicate an excellent distribution in the regions where more refinement is desired.

Table 1 MSC: Initial conditions.

	ρ	p	u_x	u_y	u_z	B_x	B_y	B_z
$x < 0.05$	3.86859	167.34500	11.25360	0.00000	0.00000	0.00000	2.1826182	−2.1826182
$x > 0.05$	ρ^0	1	0	0	0	0	0.56418958	0.56418958

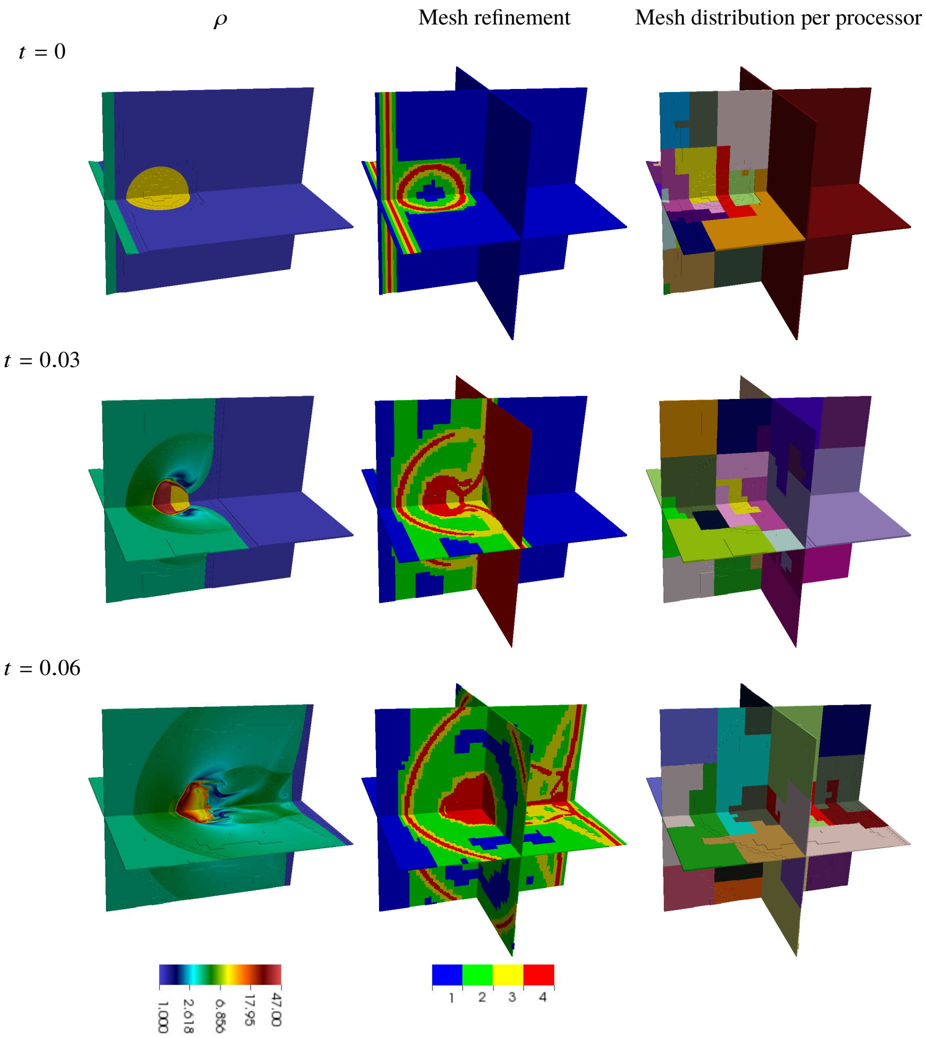


Fig. 2 MSC: Results for density ρ, the adaptive mesh refinement and distribution per processors. This simulation is performed using L = 4 refinement levels with a base mesh size of 128^3 cells, MinMod limiter, threshold value $\epsilon = 0.025$, and 48 processors until $t_{\text{end}} = 0.06$.

Table 2 presents the results of the AMR simulations using a refinement threshold value $\epsilon = 0.025$ with a number L of extra refinement levels, so that the finest level allowed corresponds to a uniform mesh size of 512^3 cells. This table also presents the $\mathbb{L}^1$-norm error for pressure, the CPU time, the number of cells used in the discretisation at the final instant and the number of patches used in the grid hierarchy for each case. For these parameters, the simulation with L = 3 refinement levels presented a better structure localisation than the one with L = 2, due to the number of cells and patches required. This resulted in a significative reduction in the CPU time (around 13%) with a small increase in the overall $\mathbb{L}_1$-error in the order of 10^{-4}. Moreover, the simulations with L = 3 and 4 refinement levels presented similar number of cells and patches in their adaptive meshes. Consequently, these cases presented similar CPU time. This indicates that the fourth refinement level has barely been used, causing the number of cells and patches to be similar. Nevertheless, the small reduction in the CPU time by 1% obtained by the L = 4 simulation, in relation to the L = 3 simulation, increased the overall $\mathbb{L}_1$-error by the order of 10^{-5}. Therefore, the result with L = 4 is considered to present the best gain considering precision and CPU time.

Table 2 MSC: Errors in pressure p, memory consumption (number of cells and patches used) and CPU time obtained by using several refinement levels L using MinMod limiter at $t_{\text{end}} = 0.06$ with a threshold value $\epsilon = 0.025$.

Mesh		L	Accuracy (p)	Cells		Patches	CPU Time	
size	base		$\mathbb{L}_1$ error ($\times 10^{-3}$)	#	%	#	(min)	%
512^3	64^3	4	2.481	34,032,848	25	10,812	442.8	32
512^3	128^3	3	2.4176	33,967,120	25	11,548	458.9	33
512^3	256^3	2	2.1898	40,190,312	30	22,420	636.4	46

In Table 3, we present a comparison between the uniform and an adaptive computation with refinement threshold value $\epsilon = 0.025$ and clustering ratio $\vartheta = 0.80$ for meshes with different finest refinement and using the same base mesh 32^3. The gain of the adaptive computations sharply increases with the enlargement of the mesh. In particular, for the adaptive computation, we use less than 3% of the number of cells for computation related to a uniform mesh 1024^3, and spend a CPU time close to the computation of the uniform mesh 512^3. Moreover, in the highest resolved computation, the uniform mesh exceeds our cluster memory and we therefore cannot provide the CPU time for this computation. The numbers of patches are almost similar among the two refined meshes; however, the most refined mesh presents a slightly smaller number of patches, which seems to indicate a better localisation of the structures.

Table 3 MSC on uniform and MR adaptive mesh computations using MinMod limiter.

Mesh	Uniform mesh		Adaptive mesh, $\epsilon = 0.025$, $\vartheta = 0.80$			
(size)	# Cells	CPU Time (s)	Base mesh (size)	# Patches	# Cells	CPU Time (s)
256^3	16, 777, 216	4, 563.6	32^3	148	5, 917, 288	1, 468.5
512^3	134, 217, 728	74, 185.5	32^3	894	27, 444, 592	11, 350.3
$1, 024^3$	1, 073, 741, 824	–	32^3	847	28, 714, 656	117, 731.4

Considering the finest adaptive mesh 1024^3, Table 4 shows the comparison with different cluster parameters and base meshes taking into account the number of cells and the CPU time. In both cases, the clustering ratio $\vartheta = 0.99$ presented the largest number of patches, which is expected. In contrast to what we observed in the two-dimensional case [19] that these results produced less computational effort and number of cells. This could be related to the localisation of the structures in the solution in this three-dimensional case. Therefore, in this case a clustering ratio $\vartheta = 0.99$ presented better results.

Table 4 MSC: Cell-cluster comparison for MR adaptive computations considering different clustering ratios ϑ and base meshes, with $\epsilon = 0.025$ and MinMod limiter.

Mesh		ϑ	# Patches	# Cells	CPU
size	base				Time (s)
1024^3	16^3	0.99	392	28, 730, 312	158, 152.4
1024^3	16^3	0.80	351	28, 991, 200	180, 701.6
1024^3	32^3	0.99	868	28, 476, 728	116, 871.8
1024^3	32^3	0.80	847	28, 714, 656	117, 731.4

4.2 Magnetic reconnection (REC)

As a relevant question of space science, the magnetic reconnection problem is described in [30] as the merging of the magnetic field lines from two predominantly opposing magnetic fields, liberating a considerable amount of energy, and redirecting the direction of particle flows. This type of phenomenon is common in solar physics and is highly studied due to the effects of the interaction between the Earth‘s magnetic field and the interplanetary magnetic field that creates complicated space environment processes. This test aims to verify the implementation of the resistive terms in MHD, once the resulting effects can be connected with the process responsible for the morphological transition of the magnetic field lines and changing in the plasma's flux, as presented in [42].

The problem is initialised considering two different states, divided at $x = 0$, with a small transition gap between them. The two states have magnetic fields that present opposing orientation over direction, and the reconnection occurs inside a small region inside the transition gap where there is a small resistivity. The computational domain for this problem is $\left[-\frac{1}{2}, \frac{1}{2}\right] \times [-2, 2]$ with Dirichlet boundary conditions. Inside this domain, the resistivity region

$$\eta\,(x, y) = \frac{\eta_0}{4}\,[1 + \cos(10\pi x)]\,[1 + \cos(2.5\pi y)]$$

is defined in the subdomain $[-\mathrm{L}_r, \mathrm{L}_r] \times \left[-\frac{1}{5}, \frac{1}{5}\right]$, and zero elsewhere with the parameters $\eta_0 = 6 \cdot 10^{-4}$ and $\mathrm{L}_r = 0.05$. The initial conditions of the physical quantities are $\rho = 1$, $p = 0.1$, $\mathbf{u} = \mathbf{0}$. The components of the initial magnetic field are given according to its corresponding state, as presented in Table 5, with non-dimensional quantities compatible with the defined MHD model. The simulations of this problem are performed using the adiabatic constant $\gamma = \frac{5}{3}$ and the HLLD Riemann solver combined with the MC limiter. The parabolic-hyperbolic correction uses the factor $\alpha_p = 0.4$. All simulations are performed under Courant number $\sigma = 0.4$ until the final time $t = 2.5$.

Table 5 REC: Magnetic field initial condition.

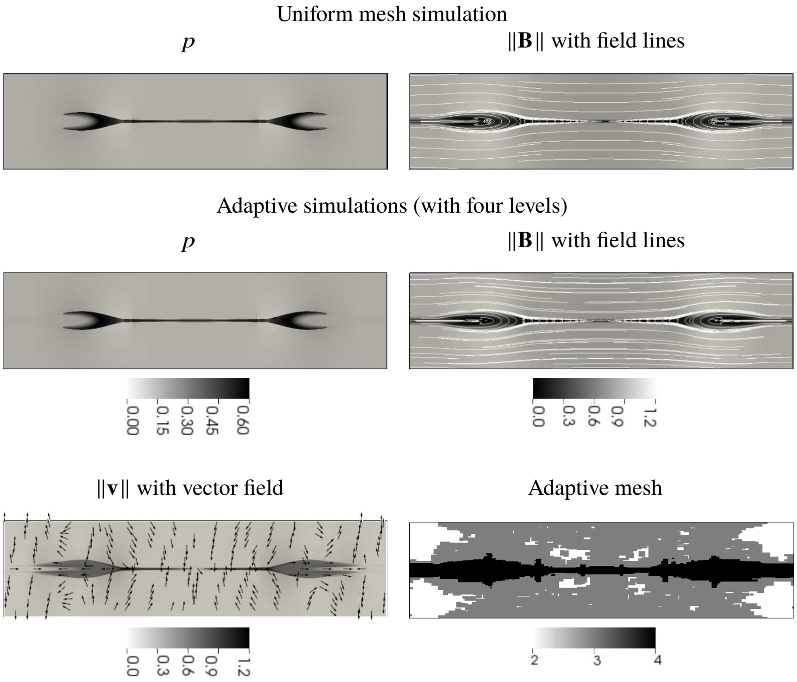
Region	B_x	B_y	B_z
$x < -\mathrm{L}_r$	0	-1	0
$x > \mathrm{L}_r$	0	1	0
Transition zone	0	$\sin\left(\frac{\pi}{2\mathrm{L}_r}x\right)$	$\cos\left(\frac{\pi}{2\mathrm{L}_r}x\right)$

Table 6 REC: Errors in pressure p, memory consumption and CPU time with refinement levels L.

Mesh		L	Accuracy (p)	Cells		Patches	CPU Time	
size	base		$\mathbb{L}_1$ error	#	%	#	(min)	%
1024 × 2048	128 × 256	4	0.0231	928,932	44	852	27.7	29
1024 × 2048	256 × 512	3	0.0226	895,292	42	809	48.1	51
1024 × 2048	512 × 1024	2	0.0050	856,968	40	466	48.3	51

Table 6 presents the error in pressure p using the $\mathbb{L}_1$-norm, the CPU time and the number of cells and patches used in the adaptive mesh for simulations with several refinement levels. These simulations are performed using the threshold value of $\epsilon = 0.001$ for the MR refinement criteria. The number of levels used in each simulation is configured such that the most refined scale corresponds to a 1024×2048 mesh. The simulation that presented the best results is obtained with L = 2, which corresponds to a reduction of 49% of the CPU time, while maintaining an error in the order of 5×10^{-3}. In that case the gain is roughly four times concerning the simulation with L = 4. Besides, this case presented the lowest number of cells concerning the uniform mesh simulation and also the lowest number of patches. Furthermore, these adaptive cases required only around 40% of the cells of the uniform mesh simulation.

Figure 3 presents the solution for p, and the adaptive mesh for the simulation with 4 levels. Showing the adaptive mesh refinement, the figure can be interpreted such that the brightest regions of the domain represent the coarsest scales; while the darkest regions correspond to the most refined scales. Physically, this figure represents a snapshot of the magnetic reconnection process. One can identify the interface between the domains of opposing orientation connected to the field line merging processes. Furthermore, there is a convergence of plasma in one direction and divergence in a perpendicular direction, as can be seen in the velocity plot. These adaptive results are in agreement with the MHD solutions presented in [30], and the modelling results in agreement with the underlying physics, as considered in [42].



Uniform mesh simulation

p $||\mathbf{B}||$ with field lines

Adaptive simulations (with four levels)

p $||\mathbf{B}||$ with field lines

$||\mathbf{v}||$ with vector field

Adaptive mesh

Fig. 3 REC solution for uniform mesh (p, and $||\mathbf{B}||$), and the respective adaptive simulations for (p, $||\mathbf{B}||$, and $||\mathbf{v}||$) with their adaptive meshes. This simulation is performed using four refinement levels with threshold value $\epsilon = 0.001$ and 24 processors. These Figures are presented using the y-axis in the horizontal direction.

Table 7 presents a breakdown of the most computationally costly tasks of the adaptive REC simulations with different number of refinement levels. In comparison with the uniform mesh simulation, the *Integration* costs of the adaptive simulations exhibit a significant reduction, as expected from the lower number of integrated cells. Moreover, the *Boundary* costs also show a reduction for three and four refinement levels, while costs for not explicitly timed operations (*Misc*) reduce as more levels are included. The *Output* production cost is generally insignificant, however also decreases for larger level number. Besides these cost reductions, the *Recomposition* and *Remeshing* tasks associated with the adaptive simulations present a significant cost,

Table 7 REC: Breakdown of the CPU time, in seconds, spent in main computation tasks for different numbers of refinement levels L using $\epsilon = 0.001$.

Main task	Refinement Levels			
	one	two	three	four
Integration	3756.8	931.5	747.2	737.9
Boundary	796.1	852.9	591.0	503.5
Memory restart	142.0	29.5	11.0	5.7
Recomposition	-	500.7	1141.0	228.2
Remeshing	-	24.1	21.4	21.4
Misc	518.6	359.8	203.0	116.6
Output	3.8	6.4	5.4	1.2

especially for the three-level simulation. Nevertheless, even considering these costs, the adaptive simulations require less CPU time than the uniform mesh simulations.

4.3 Magnetosphere (MAG)

Initially as conceived from the incidence of electrically charged solar particles, the Chapman-Ferraro model proposed that the close Earth's space environment was an empty region that avoids the presence of those particles [8, 9]. Nowadays, governed primarily by the geomagnetic field, a much more complicated electrodynamics region develops surrounding the planet, designated as magnetosphere [42], populated by plasmas.

As application cases for the solver, the first analysis deals with a predominantly northward oriented interplanetary magnetic field (IMF) causing – in principle – a geomagnetically-closed frontal magnetosphere; while the second case takes into account the southward-oriented IMF, which causes a geomagnetically-open frontal magnetosphere. The main modelling ideas are described in details in [39, 40], and references therein. For these studies, the close Earth's environment is considered basically as a sphere with density and pressure constant in time and containing a magnetic dipole. This region is connected to an outer region where the geomagnetic field is compressed or stretched by the solar wind, defining a region where the well-known phenomenon of magnetic reconnection can occur.

The physical model of this problem requires changes in the resistive MHD formulation presented in Eqs. 2. These changes consist of the inclusion of an external gravity field, an artificial viscosity over the density and pressure field to reduce the MHD fluctuation that arises from the unbalanced forces in the initial condition, and a modification in the Ampère law so that $\mathbf{J} = \nabla \times (\mathbf{B} - \mathbf{B_d})$, where $\mathbf{B_d}$ is the intrinsic dipole magnetic field of the Earth [42, p. 223]. The reason to subtract the dipole field from the Ampère law is supported to the expectation of a significant electric current to be generated in the frontal interface layer between the two media, i.e., the interplanetary space and the outer terrestrial region [39, 38, 40]. These modifications imply the inclusion of the source terms $D\nabla^2\rho$ in the continuity equation, Eq. 2a, $\rho\mathbf{g} + \Phi + \mathbf{u}D\nabla^2\rho + \mathbf{B} \times (\nabla \times \mathbf{B_d})$ in the momentum equation, Eq. 2b, and $\rho\mathbf{u} \cdot \mathbf{g} + \frac{D_p\nabla^2 p}{\gamma-1} + \frac{\|\mathbf{u}\|^2}{2}D\nabla^2\rho + \mathbf{u} \cdot \Phi + \eta\|\nabla \times \mathbf{B_d}\|^2 + (\nabla \times \mathbf{B_d}) \cdot (\mathbf{u} \times \mathbf{B} - \eta\nabla \times \mathbf{B})$ in the total energy equation, Eq. 2c. The diffusive terms in this model are given in [39] as $D = D_p = \frac{\mu}{\rho_{SW}} = 0.02$ and $\Phi = \mu\nabla^2\mathbf{u}$, where $\rho_{SW} = 5.00 \cdot 10^{-4}$ corresponds to the typical value (5 n/cc) for the solar wind density. Moreover, we use the following induction equation for the magnetic field:

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot \left[\mathbf{uB} - \mathbf{Bu} + \eta\left((\nabla\mathbf{B})^T - \nabla\mathbf{B} - (\nabla\mathbf{B}_d)^T + \nabla\mathbf{B_d}\right) + \psi\mathbf{I}\right] = \mathbf{0} \tag{6}$$

Physical quantities

The physical quantities included into the model are described by a function related to the distance to the centre of Earth, represented by ξ $(= \sqrt{x^2 + y^2})$ defined with the Earth located in the xy-plane origin. This two-dimensional formulation considers the x-axis in the Sun-Earth direction crossing the Earth Equator, while the y-axis is orthogonal to the x-axis so that it contains the Earth north and south poles. This reference axis orientation is used for computational convenience in the code. Nevertheless, to be clear, in the domains of the space scientists that use an own reference coordinate system representation, this axis is considered to be the z-axis. Besides the initial conditions, the distance ξ is used to determine the external fields used in these simulations, such as the external gravity $\mathbf{g}$, defined by the vector field $\mathbf{g}(x, y) = -\frac{g_0}{\xi^2}\,[x,\ y,\ 0]$, where $g_0 = 1.35 \times 10^{-6}$; the line dipole magnetic field $\mathbf{B_d}$, given by $\mathbf{B_d}(x, y) = \left[-2xy\xi^{-4},\ \xi^{-4}\left(x^2 - y^2\right),\ 0\right]$, and the resistivity function $\eta(x, y) = \eta_0\frac{w}{w+1}$, where the weight value reads $w = 30(\max\left[(\xi/16)^2 - 1,\ 0\right])^2$, as defined in Ogino's implementation.

Those non-dimensional quantities are obtained from a physical system of units in which the distance, magnetic induction and time are established based in the Earth's radius (RE=$6.37 \cdot 10^6$ m). We provide some fundamental physical information used in this context. The Earth magnetic field at the equator is $3.12 \cdot 10^{-5}$ T and the Alfvén transit time, a time required by the Alfvén wave to go through the equivalent of the Earth radius, is taken as 0.937 s. Based on those quantities, the pressure unit corresponds to 7.75×10^{-4} N/m^2, the velocity unit to $6.80 \cdot 10^6$ m/s, the acceleration unit to $7.26 \cdot 10^6$ m/s^2, and the current density unit to $3.90 \cdot 10^{-6}$ A/m^2.

Initial and boundary conditions

The initial configuration for this problem consists in an approximation of the ionosphere, which describes the plasma in the Earth's neighbourhood, based on the initial condition proposed in [39, 38, 40]. This ionosphere is constructed so that its pressure and density are proportional to ξ. The initial magnetic field in this work agrees with the one proposed in the works [39, 38] by including an imaginary dipole at the coordinate $x = 2x_m$ so that $\mathbf{B}^0(x, y) = \mathbf{B_d}(x, y) + \mathbf{B_d}(2x_m, y)$, where x_m is the equilibrium point on the x-axis where the solar wind kinetic energy counterbalances the dipole's magnetic energy, i.e., the coordinate at which $\rho_{SW} u_{SW}^2 = \mathbf{B_d}^2$ for $y = 0$. This imaginary dipole, known as the image method in electrodynamics [42, p. 225-227], aids to produce an initial magnetic field that has its field lines compressed, and not crossing the line $x = x_m$. Its initial structures are constrained to the domain of the Earth (where $x \geq x_m$). This magnetic configuration is done to accelerate the convergence into a steady-state magnetosphere so that the Earth magnetic field does not cross the position immediately beyond the magnetopause, formed around $x = x_m$. In the domain of the interplanetary medium (where $x < x_m$), the initial magnetic field is set to be equal to the typical interplanetary magnetic field. This problem is simulated inside the physical domain $[-150, 450] \times [-150, 150]$ RE, which is complemented with Neumann boundary conditions so that the derivatives of the physical quantities are zero at the boundaries $x_e = 450$, $y_s = -150$ and $y_e = 150$. The boundary at $x_s = -150$ is used to prescribe the solar wind parameters.

The entire ionosphere initial condition and the typical solar wind parameters, used as boundary conditions, are given in Table 8, where $p_{00} = g_0(\gamma - 1)/\gamma = 5.40 \cdot 10^{-7}$ for $\gamma = 2$, $p_0 = 3.56 \cdot 10^{-8}$ (corresponding to a temperature of $T_{SW} = 2 \times 10^5$ K), $u_{SW} = 4.41 \cdot 10^{-2}$ (300 Km/s) and $B_{SW} = \pm 1.5 \cdot 10^{-4}$ (5 nT), in which the positive (negative) signal of B_{SW} is associated with the northward (southward) orientation of the magnetic field of the solar wind.

Furthermore, the physical domain also presents an internal boundary corresponding to a near-Earth region, i.e. a region defined indeed by the ionosphere and the plasmasphere (details can be seen in [26, p. 208-222, and p. 164-173, respectively]). Considering the Earth positioned at the origin, this internal boundary removes the points with $\xi < 16$ from the computational domain. In order to dampen out all perturbations near the ionosphere, the near-Earth neighbourhood is smoothed in relation with the initial condition after every time step by the operation $\mathbf{q}^{n+1} = f\,\mathbf{q}_*^{n+1} + (1 - f)\,\mathbf{q}^0$, where $\mathbf{q}_*^{n+1}$ is the solution obtained after the time evolution and $\mathbf{q}^0$ is the ionospheric initial condition. The weight value f is computed as $f = \dfrac{\bar{f}^2}{\bar{f}^2 + 1}$, where $\bar{f} = 100\left(\max\left[\left(\frac{\xi}{16}\right)^2 - 1,\ 0\right]\right)^2$. As proposed, this function guarantees a smooth transition of quantities in a thin layer immediately surrounding the ionosphere boundary, which presents constant values. This approach avoids the MHD solver run stopping due to numerical instabilities.

Configuration of the magnetosphere

The initial condition presented earlier describes the initial state of the ionosphere, without contemplating an initial state for the magnetosphere. Thus, before introducing inputs composed of realistic data from the interplanetary environment, an approximate configuration for the magnetosphere should be realised. The initial ionospheric configuration is simulated using the typical solar wind parameters $\mathbf{q}_{SW}$, presented in Table 8, in order to obtain an initial state for a magnetosphere in equilibrium. Figure 4 presents the density and, in white, the magnetic field lines. The panels in the figure contain the initial condition and intermediary states of the magnetosphere configuration before a stationary state is obtained using southward IMF. These configurations are presented alongside with its corresponding adaptive grids. An IMF orientation choice is made to match the orientation of the magnetic field at the beginning of the dataset. This choice is due to the Earth magnetic field, under typical northward and southward IMF, converging to different states, as presented in Figure 5. In the lower panel in the figure, there is a frontal reconnection, and in the upper panel, reconnection does not occur in this position. The errors, CPU time and number of cells and patches of the adaptive simulations using two and three refinement levels are presented in Table 9 for both northern and southern oriented solar wind. The simulations with southern oriented solar wind presented better performance due to its simpler mesh configuration, as illustrated in Figure 5. Besides the dipole, which is being compressed on the solar wind side while being elongated downstream for both cases, these steady-state magnetospheres differ by potential locations of the magnetic reconnection process. These simulations with typical solar wind parameters are assumed to have converged to steady-state at the instant $t = 172,799.666$ s, that corresponds to around two days of simulated time. Both steady stationary states reached are in agreement with the space physics conditions. After that, the solar wind dataset input starts and the evolution of the solar wind-magnetosphere coupling process is simulated.

Table 8 Initial conditions and typical solar wind parameters.

	ρ	p	u_x	u_y	u_z	B_x	B_y	B_z
$\mathbf{q}^0(x > x_m, y)$	$\max\left(\frac{1}{\xi^2}, 10^{-4}\right)$	$\max\left(\frac{p_{00}}{\xi}, p_0\right)$	0	0	0	B_x^0	B_y^0	0
$\mathbf{q}^0(x \leq x_m, y)$	$\max\left(\frac{1}{\xi^2}, 10^{-4}\right)$	$\max\left(\frac{p_{00}}{\xi}, p_0\right)$	0	0	0	0	B_{SW}	0
$\mathbf{q}_{SW}(x_s, y, t)$	ρ_{SW}	p_0	u_{SW}	0	0	0	B_{SW}	0

This problem is simulated using the HLLD Riemann solver [35] combined with the MinMod limiter under the Courant number $\sigma = 0.4$. The GLM formulation uses the factor $\alpha_p = 0.4$.

Solar wind experiments in the inflow

After the establishment of the bow shock and the magnetosphere at the stationary state, we introduce the interplanetary parameters obtained from the OMNI web service, NASA, corresponding to the period from January 16th to 18th of 2018. By the examination of the geomagnetic disturbance indices (using for instance planetary index Kp, auroral electrojet index AE, and low latitude geomagnetic index Dst) from

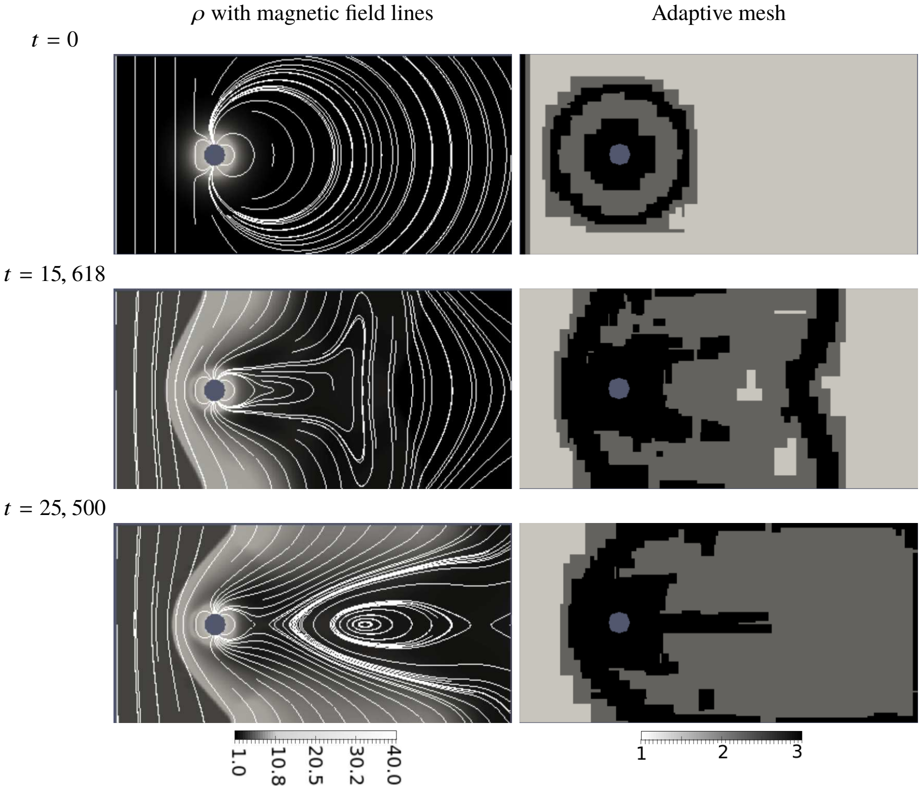


Fig. 4 Density solution (left) at various time instants during the southern solar wind steady-state magnetosphere construction with its correspondent adaptive mesh (right) with finest level corresponding to a 512×256 mesh and threshold value $\epsilon = 0.05$. Magnetic field lines represented in white evolving from a southward orientation.

the information of the World Data Center for Geomagnetism, Kyoto[3], this interval corresponds to a typical period of geomagnetically quiet conditions (details in the Appendix). Nevertheless, it still represents a situation of interesting variation in the IMF orientation, which challenges the actual MHD modelling. We use a smoothed version of the data that preserves the primary realistic features, plotted in Figure 6, to avoid unnecessary oscillations in the simulation. In detail, the filtering process is built with an orthogonal wavelet transform reconstruction with Daubechies family 8 removing the first 6 levels of the wavelet coefficients.

These datasets, displayed in Figure 6, consist of interplanetary magnetic field components (in Geocentric Solar Magnetospheric refence system, GSM [42, p. 536]) Bx and Bz in nanotesla, the plasma flow speed in kilometre per second, the numerical density in particle per cubic centimetre, and solar wind temperature in Kelvin. All these quantities are given in a function of the time in Day Of Year

[3] World Data Center for Geomagnetism: `http://wdc.kugi.kyoto-u.ac.jp/wdc/Sec3.html`

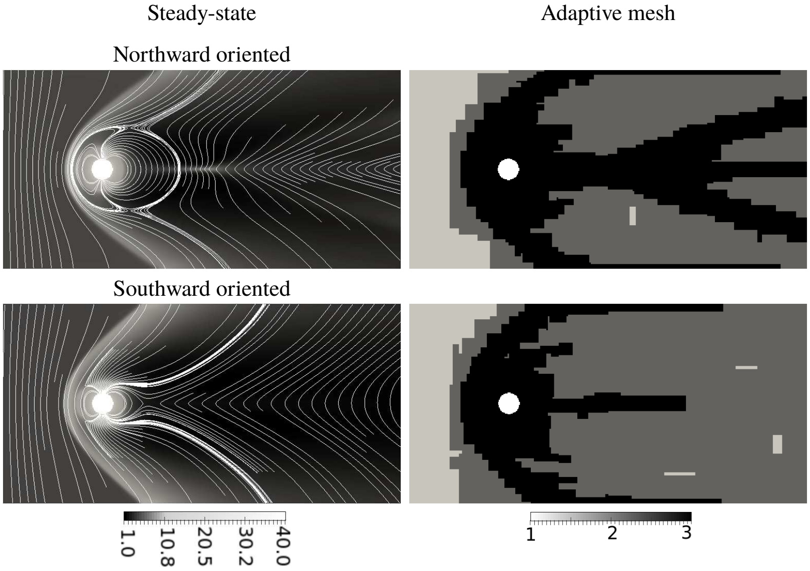


Fig. 5 Steady-state magnetosphere predictions obtained respectively from northward (top) and southward (bottom) oriented interplanetary magnetic field using an adaptive mesh with finest level corresponding to a 512×256 mesh. Left: density solution (in n/cc) with magnetic field lines. Right: adaptive mesh using the threshold value $\epsilon = 0.05$.

Table 9 MAG: Errors in pressure p, memory consumption (number of cells and patches used) and CPU time obtained by using several refinement levels L using MinMod limiter at $t_{\text{end}} = 184396$ with a threshold value $\epsilon = 0.05$. The experiments were performed using magnetic field orientation in both north and south directions until the stationary state is obtained.

Solar Wind	Mesh		L	Accuracy (p)	Cells		Patches	CPU Time	
Orientation	size	base		$\mathbb{L}_1$ error ($\times 10^{-3}$)	#	%	#	(min)	%
North	512×256	128×64	3	3.4158	84,468	65	313	161	74
	512×256	256×128	2	2.6703	81,784	62	323	159	67
	1024×512	256×128	3	2.0036	191,176	36	505	849	74
South	512×256	128×64	3	1.4590	78,556	60	266	93	66
	512×256	256×128	2	1.1366	74,624	57	257	84	60
	1024×512	256×128	3	1.1953	167,228	32	416	621	36
	1024×512	512×256	2	0.9073	189,490	36	293	740	43

(DOY). Also, the pressure value is obtained using the particle density data n and the temperature T as $p = nkT$, where $k = 1.38064852 \cdot 10^{-23}$ m^{2}kgs^{-2}K^{-1} is the Boltzmann constant.

Furthermore, due to our computational formulation of considering the y-axis as the axis which the dataset considers as being the z-axis, the component B_z of the dataset is inputted as the B_y component of the solar wind.

These smoothed time series are introduced as an inflow boundary condition of the northward orientated steady-state magnetosphere, that is used as an approximation of the Earth magnetosphere on Jan 16th 00 : 00h-UT (shown at the top-left panel in Figure 5). Selected from the numerical evolution as an example, Figure 7 presents the configuration of the Earth magnetosphere on the instants Jan 17th at 07 : 04h-UT, Jan 17th at 08 : 00h-UT and Jan 17th at 10 : 04h-UT, which respectively refer to the effect of the IMF orientation. The upper panel in the figure corresponds to the northward-oriented B_z interval (letter A in Figure 6), the intermediate one corresponds to the start of negative value (letter B), and the lower panel corresponds to the peak of the southward-oriented interval (letter C). As expected, reconnection processes are noticed in a remarkable way, at the lower panel, in the frontal face and inside the tail of the magnetosphere. The analysis of the dynamic evolution represented in the plots can justify the importance of these kind of simulations even for geomagnetically quiet conditions on the ground. Interestingly, related to letter C, a time-coincident small intensity effect was noted in the auroral index, and no effect stands out in the other equatorial index (condition reported in the Figure 8 in Appendix). A more complete physical investigation is out of the scope of this work. In the simulation figure, the grey pattern used for the density allows clear identification of the interplanetary region, the bow shock, the interplanetary magnetosheath, and the own magnetosphere.

5 Conclusion

The framework AMROC with its implemented SAMR algorithm, MPI parallelisation, multiresolution criterion, and clustering structure was the base of the presented ideal and resistive GLM-MHD solvers with parabolic-hyperbolic magnetic divergence-free correction. These solvers achieved a milestone in the development of an Earth's magnetosphere model. In this work, three physical and numerical challenging tests demonstrate the computational efficiency and memory utilisation of the framework for these new solvers with parallel mesh adaptation, compared to the correspondent solvers using a uniform mesh.

As expected, the performance gain of the MR adaptive MHD simulations depends on the local regularity of the solution in all physical quantities. For instance, excellent adaptive performance can be observed when the solution requires a high-resolution discretisation and has few localised disturbances relative to its domain size. Furthermore, we successfully have validated the proposed solver for a real-world magnetosphere scenario of space plasma, performing a challenging two-dimensional test for the Earth magnetosphere, and additionally, a three-dimensional implementation is also already a work in progress. Nevertheless, from our analyses, further performance enhancements also seem feasible through better parameter choices. In summary, our MHD solvers can deal with complex problems in space research; for instance, as it is, we already have a potentially straightforward numerical space weather forecast model that forthcoming projects will complete.

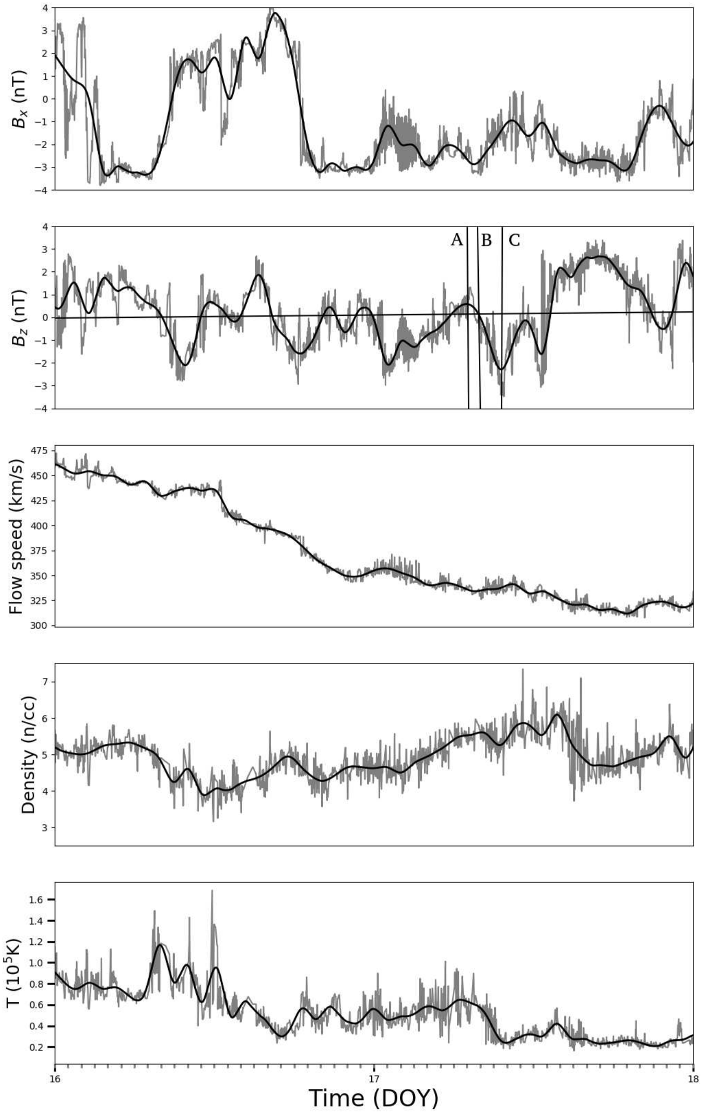
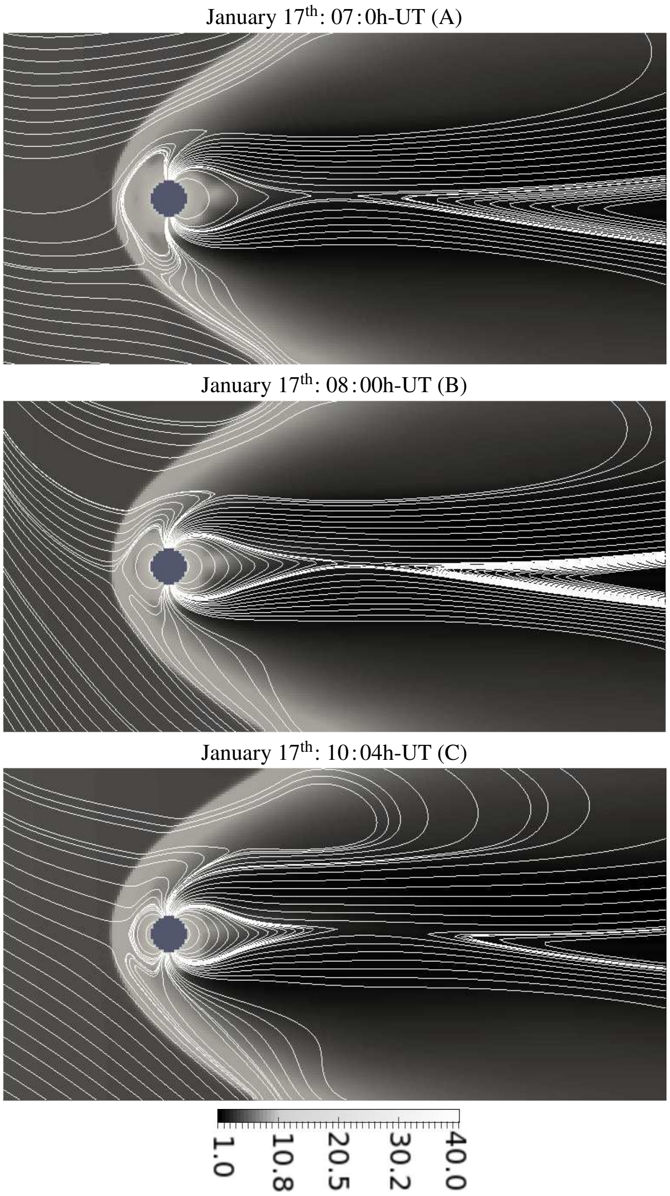


Fig. 6 MAG solar wind inflow conditions corresponding to the period between January 16th to 18th of 2018. The dataset is presented in grey while the smoother version effectively used on the simulation is presented in black.

This work contributes to the rapidly developing area of space weather forecast and is concerned with the inherent computational challenges of reducing memory and CPU time which – in the three-dimensional case – is still urgent work in process, even for supercomputers.


January 17th: 07:0h-UT (A)

January 17th: 08:00h-UT (B)

January 17th: 10:04h-UT (C)

Fig. 7 MAG plot of density (in n/cc) with representative magnetic field lines for the scenarios after the inclusion of the smoothed solar wind inflow satellite data.

Acknowledgement The authors thank the FAPESP (Grants: 2018/03039 – 9, 2015/25624–2), CNPq (Grants: 424352/2018–4, 302226/2018–4, 307083/2017–9, 306038/2015–3), and FINEP (Grant: 0112052700) for financial support of this research. MML thanks CNPq (Grant: 140626/2014 – 0) for his doctorate and CAPES (Grant: 88882.463276/2019 – 01) for his post-doctorate scholarship. We are indebted to Eng. V. E. Menconi for his invaluable computational assistance, to M. Sierra-Lorenzo and A. K. F. Gomes for the fruitful discussions, and to Prof. Ogino for the MHD code and scientific discussions that inspired our magnetosphere implementation. We thank also to the teams of World Data Center for Geomagnetism, Kyoto, for the geomagnetic indices dataset, and the OMNI web service, NASA, for access to the interplanetary dataset.

We also thank the anonymous reviewers for their comments and suggestions that have improved the final version of this manuscript.

References

1. J. Ahrens, B. Geveci, and C. Law. *ParaView*. Los Alamos National Laboratory, Los Alamos, 2005.
2. J. Bell, M. J. Berger, J. Saltzmann, and M. Welcome. Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 15:127–138, 1994.
3. M. J. Berger. *Adaptive mesh refinement for hyperbolic partial differential equations*. PhD thesis, Stanford University, Stanford, 1982.
4. M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82(1):64–84, 1989.
5. M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.
6. J. A. Bittencourt. *Fundamentals of Plasma Physics*. Springer–Verlag, New York, 3 edition, 2004.
7. J. U. Brackbill and D. C. Barnes. The effect of nonzero $\nabla \cdot B$ on the numerical solution of the magnetohydrodynamic equations. *J. Comput. Phys.*, 35(3):426–430, 1980.
8. S. Chapman and V. C. A. Ferraro. A new theory of magnetic storms. *Nature*, 126(3169):129–130, 1930.
9. S. Chapman and V. C. A. Ferraro. A new theory of magnetic storms. *Terrest. Magnetistm and Atmospheric Elec.*, 36:171–186, 1931.
10. H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rübel, M. Durant, J. M. Favre, and P. Navrátil. VisIt: an end-user tool for visualizing and analyzing very large data. In E. W. Bethel, H. Childs, and C. Hansen, editors, *High performance visualization–enabling extreme-scale scientific insight*, chapter 16, pages 357–372. Chapman and Hall/CRC, New York, 1 edition, 10 2012.
11. A. Cohen. Wavelet methods in numerical analysis. In P. G. Ciarlet and J. L. Lions, editors, *Handbook of Numerical Analysis*, volume VII. Elsevier, Amsterdam, 2000.
12. A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg. Hyperbolic divergence cleaning for the MHD equations. *J. Comput. Phys.*, 175(2):645–673, 2002.
13. R. Deiterding. *Parallel adaptive simulation of multi-dimensional detonation structures*. PhD thesis, Brandenburgische Technische Universität Cottbus, Sep 2003.
14. R. Deiterding. Construction and application of an AMR algorithm for distributed memory computers. In T. Plewa, T. Linde, and V. G. Weirs, editors, *Adaptive Mesh Refinement - Theory and Applications*, pages 361–372. Springer, 2005.
15. R. Deiterding. Block-structured adaptive mesh refinement - theory, implementation and application. *ESAIM: Proc.*, 34:97–150, 2011.
16. R. Deiterding and M. O. Domingues. Evaluation of multiresolution mesh adaptation criteria in the AMROC framework. *Civil-Comp Press*, 111, 2017.

17. R. Deiterding, M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider. Adaptive Multiresolution or Adaptive Mesh Refinement: A Case Study for 2D Euler Equations. *ESAIM: Proc.*, 29:28–42, 2009.
18. R. Deiterding, M. O. Domingues, and K. Schneider. Multiresolution analysis as a criterion for effective dynamic mesh adaptation - a case study for euler equations in the SAMR framework AMROC. *Comput. Fluids*, 2019. (Under revision).
19. M. O. Domingues, R. Deiterding, M. Moreira Lopes, A. K. F. Gomes, O. Mendes, and K. Schneider. Wavelet-based parallel dynamic mesh adaptation for magnetohydrodynamics in the AMROC framework. *Comput. Fluids*, 190:374 – 381, 2019.
20. M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider. Adaptive multiresolution methods. *ESAIM: Proc.*, 34:1–96, 2011.
21. J. P. Eastwood, E. Biffis, M. A. Hapgood, L. Green, M. M. Bisi, R. D. Bentley, R. Wicks, L.A. McKinnell, M. Gibbs, and C. Burnett. The economic impact of space weather: Where do we stand? *Risk Anal.*, 37(2), 2017.
22. X. Feng. *Magnetohydrodynamic Modeling of the Solar Corona and Heliosphere*. Springer, Singapore, 2020.
23. S. Fromang, P. Hennebelle, and R. Teyssier. A high order Godunov scheme with constrained transport and adaptive mesh refinement for astrophysical MHD. *Astronomy & Astrophysics*, 457(2):371–384, 2006.
24. C. Gheller. ENZO and RAMSES codes for computational astrophysics. Technical report, Swiss National Supercomputing Centre, Nov 2017. Available online in `https://hpc-forge.cineca.it/files/CoursesDev/public/2017/HPC_methods_for_Computational_Fluid_Dynamics_and_Astrophysics/Bologna/ASTRO2-ENZO_and_RAMSES_Codes-Gheller.pdf`.
25. P. Goedbloed and R. Keppens. Lecture notes in magnetohydrodynamics of astrophysical plasmas. Chapter 4: The MHD model. Technical report, Utrecht University, Sep-2004 / Jan-2005. Available online in `https://perswww.kuleuven.be/~u0016541/MHD_sheets_pdf/nsap430m.06.4.pdf`.
26. J. K. Hargreaves. *The solar-terrestrial environment*. Cambridge University Press, 1992.
27. A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Commun. Pure Appl. Math.*, 48(12):1305–1342, 1995.
28. A. Harten, P. D. Lax, and B. van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35–61, 1983.
29. P. F. Hopkins. A constrained-gradient method to control divergence errors in numerical MHD. *Mon. Not. R. Astron. Soc.*, 462:576–587, 2016.
30. R.-L. Jiang, C. Fang, and P.-F. Chen. A new MHD code with adaptive mesh refinement and parallelization for astrophysics. *Comput. Phys. Commun.*, 183(8):1617–1633, 2012.
31. B. Koren. A robust upwind discretization method for advection, diffusion and source terms. In C.B. Vreugdenhil and B. Koren, editors, *Numerical methods for advection-diffusion problems*, Notes on Numerical Fluid Mechanics, pages 117–138. Vieweg, Germany, 1993.
32. L. D. Landau, E. M. Lifshitz, and L. P. Pitaevskii. *Electrodynamics of Continuous Media*, volume 8 of *Course of Theoretical Physics S*. Pergamon, 2nd edition, 2004.
33. R. J. LeVeque. *Numerical methods for conservation laws*. Birkhäuser Verlag, Basel, 1990.
34. A. Mignone, P. Tzeferacos, and G. Bodo. High-order conservative finite difference GLM–MHD schemes for cell-centered MHD. *J. Comput. Phys.*, 229(17):5896–5920, 2010.
35. T. Miyoshi and K. A. Kusano. A multi-state HLL approximate Riemann solver for ideal magnetohydrodynamics. *J. Comput. Phys.*, 208:315–344, 2005.
36. M. Moreira Lopes, R. Deiterding, A.K.F. Gomes, O. Mendes, and M. Domingues. An ideal compressible magnetohydrodynamic solver with parallel block-structured adaptive mesh refinement. *Comput. Fluids*, 173:293–298, 2018.
37. S. Müller. *Adaptive Multiscale Schemes for Conservation Laws*, volume 27 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Heidelberg, 2003.
38. T. Ogino. A Three-Dimensional MHD simulation of the interaction of the Solar Wind with the Earth's Magnetosphere: the generation of field-aligned currents. *J. Geophys. Res.*, 91(A6):6791–6806, 1986.

39. T. Ogino. Two-dimensional MHD code. In Y. Omura H. Matsumoto, editor, *Computer Space Plasma Physics: Simulations and Software*, pages 161–191. Terra Scientific Publishing, Tokyo, 1993.
40. T. Ogino, R. J. Walker, and M. Ashour-Abdalla. A global magnetohydrodynamic simulation of the magnetosheath and magnetosphere when the interplanetary magnetic field is northward. *IEEE Trans. Plasma Sci.*, 20(6):817–828, 1992.
41. P L Roe. Characteristic-based schemes for the Euler equations. *Annu. Rev. Fluid Mech.*, 18(1):337–365, 1986.
42. C. T. Russell, J. G. Luhmann, and R. J. Strangeway. *Space Physics: an Introduction*. Cambridge University Press, 2016.
43. C. J. Schrijver, K. Kauristie, A. D. Aylward, C. M. Denardini, S. E. Gibson, A. Glover, N. Gopalswamy, M. Grande, M. Hapgood, D. Heynderickx, N. Jakowski, V. V. Kalegaev, G. Lapenta, J. A. Linker, S. Liu, C. H. Mandrini, I. R. Mann, T. Nagatsuma, D. Nandy, T. Obara, T. P. O'Brien, T. Onsager, H. J. Opgenoorth, M. Terkildsen, C. E. Valladares, and N. Vilmer. Understanding space weather to shield society: A global road map for 2015–2025 commissioned by COSPAR and ILWS. *Adv. Space Res.*, 55(12):2745–2807, 2015.
44. James M. Stone, Thomas A. Gardiner, Peter Teuben, John F. Hawley, and Jacob B. Simon. Athena: A new code for astrophysical mhd. *The Astrophysical Journal Supplement Series*, 178(1):137–177, 2008.
45. E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer, Berlin, 1999.
46. G. Tóth. The $\nabla \cdot B$ constraint in shock-capturing magnetohydrodynamics codes. *J. Comput. Phys.*, 161:605–652, 2000.
47. 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, Q. F. Stout, A. Glocer, Yi-J. Ma, and M. Opher. Adaptive numerical algorithms in space weather modeling. *J. Comput. Phys.*, 231:870–903, 2012.
48. B. van Leer. Towards the ultimate conservative difference scheme. V. a second-order sequel to Godunov's method. *J. Comput. Phys.*, 32(1):101–136, 1979.
49. G.D. van Albada, B. van Leer, and W.W. Roberts. A comparative study of computational methods in cosmic gas dynamics. In M.Y. Hussaini, B. van Leer, and J. van Rosendale, editors, *Upwind and high-resolution schemes*, pages 95–103. Springer, Berlin, Heidelberg, 1997.
50. B. van Leer. Towards the ultimate conservative difference scheme. II. monotonicity and conservation combined in a second-order scheme. *J. Comput. Phys.*, 14(4):361–370, 1974.
51. B. van Leer. Towards the ultimate conservative difference scheme III. upstream-centered finite-difference schemes for ideal compressible flow. *J. Comput. Phys.*, 23(3):263–275, 1977.

Appendix

Code organisation

In the context of this work, the AMROC framework, as described in [15] and published online[4] is divided into two main folders, the implementation and compilation folders. The folder *vtf/amroc/amr* contains the base algorithm for a numerical simulation using SAMR methods for a generic system of hyperbolic equations. The files contained in this folder specify the data structures and routines outside the scope

[4] AMROC webpage: `http://www.vtf.website/asc/wiki/bin/view/Amroc`

of the simulated equations, such as mesh adaptation, mesh distribution per processor, boundary conditions, restriction and prolongation operators, etc. In particular, the function *IntegrateLevel()* in the file *AMRSolver.h* calls the numerical scheme associated with the simulated equation, implemented in the base module, using the *mpass* counter. For each iteration of this counter, the scheme defined in the base is computed, then the ghost cells are updated. Considering the implemented MHD solver, this counter performs three iterations, corresponding to the first Runge-Kutta (RK) step, the second RK step and the divergence cleaning step, respectively. The GLM implementation files are located in the *mhd* directory of the implementation folder. They contain the base virtual functions to perform a generic simulation of the MHD equations for two and three dimensions.

In special, these files contain the time evolution function *Step()*, called from the Generic SAMR solver, and the virtual functions called from this function. The use of virtual functions allows the definition of base functions that may be used for most of the experiments, while allowing the redefinition of these functions in the specific MHD module, if required by the studied problem. In general, the functions from the base module implement numerical operations that are independent from the problem simulated, such as flux computations, limiters and divergence cleaning routines. The problem-specific file located in the respective source directory *src*, implements functions that are particular to each experiment. In general, this file contains initial conditions, resistivity and gravity fields. However, if necessary for the experiment, this file may contain redefinitions of virtual functions implemented in the base module. We also have for each simulation an input parameter namelist called solver.in.

Finally, the MHD module in AMROC runs scripts that already contain the commands to convert the output HDF (Hierarchical Data Format) files into binary VTK (Visualization ToolKit) files used for data visualisation in tools such as VisIt [5][10] and ParaView [6][1].

Geomagnetic disturbances

To attend the interests of the geophysical community, devices to measure the geomagnetic field, designated in a general sense as magnetometers, have been installed on the ground, nowadays composing a large net spread around the world. One can find more information and specific documentation in the World Data Center for Geomagnetism, Kyoto[7]. Also, related fundamentals on space physics are available in [42]. To quantify the level of geomagnetic disturbance occurring upon the Earth, the interested reader can survey and examine some geomagnetic disturbance indices, for instance, the index Kp for an estimated planetary disturbance behaviour, the index AE for auroral electrojet disturbance effects, and the index Dst for a low latitude

[5] Visit webpage: `https://wci.llnl.gov/simulation/computer-codes/visit/downloads`

[6] Paraview webpage: `https://www.paraview.org/download/`

[7] World Data Center for Geomagnetism: `http://wdc.kugi.kyoto-u.ac.jp/wdc/Sec3.html`

magnetic disturbance. In our case, we choose and present in Figure 8 the interplanetary magnetic field Bz, the index AE, and the index Sym-H. This information can be collected effortlessly from the OMNI web service, NASA.[8] Bz is the primary variable responsible for triggering of the magnetic reconnection process (merging of the interplanetary magnetic field lines with the geomagnetic field lines), when this IMF component is a predominantly southward-oriented field (i.e., in opposition to the geomagnetic field orientation), in the frontal side, i.e. towards to the Sun, of the Earth's magnetosphere. AE is the geomagnetic index concerning the modification of the auroral electrojet currents, that produce magnetic disturbances in the higher latitudes. $Sym - H$ is the index concerning the intensification of an equatorial, symmetrical ring electrical current (at a distance about $6 - 7$ Earth radii), that produce magnetic disturbances in the lower latitudes. Recorded by geomagnetic indices, any geomagnetic variations link intrinsically to the electrodynamical coupling between the solar plasma and the Earth's magnetosphere.

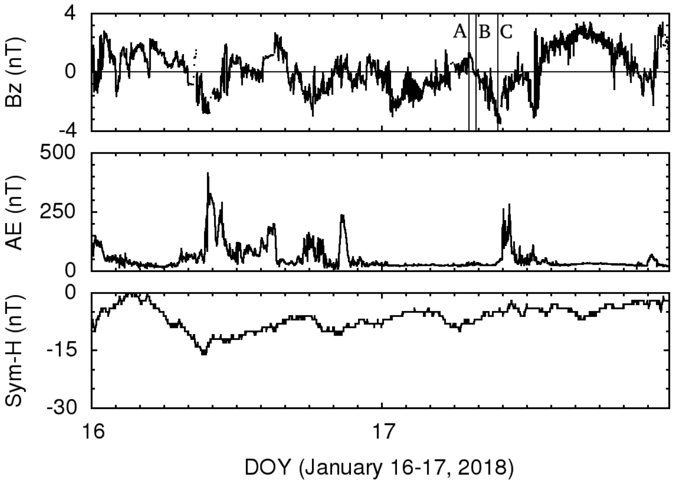


Fig. 8 MAG: Geomagnetic disturbances shown by the geomagnetic indices: Auroral Electroject (AE) and Symmetric Equatorial ring current effect (Sym-H) corresponding to the period between January 16th to 18th of 2018. Indicated in the interplanetary magnetic field Bz. Letter A indicates northward-oriented field interval, B a transition value, and C southward-oriented interval.

From the figure, indicated in the interplanetary magnetic field Bz, the letter A identifies a corresponding maximum-value time in the northward-oriented field interval, B a time under a transition value (close to zero), and C a corresponding minimum-value time in the southward-oriented interval. There are two reasons to select this dataset region: to pick up distinct interplanetary behaviours, and to be far from the simulation beginning. This procedure allows for exemplifying evolution consistency related to record inputs and tangible results. As shown in the plot, a

[8] OMNI web service, NASA: `https://omniweb.gsfc.nasa.gov/form/omni\_min.html`

time-coincident small intensity effect was noted in the auroral index, AE, and no effect stands out in the equatorial index, Sym-H. Shown in Figure 7, the simulation results for the Earth's magnetic field configuration are in physical agreement with the magnetic effects on the ground, as the physics presented and discussed, for instance, by [42]. The current code features provide the means for evolution analysis of the Earth's magnetosphere in complicated scenarios, such as investigations for geomagnetically quiet conditions.