3D magnetospheric parallel hybrid multi-grid method applied to planet-plasma interactions

Ludivine Leclercq, Ronan Modolo, François Leblanc, Sebastien L. G. Hess, Marco Mancini

3D Magnetospheric parallel hybrid multi-grid method applied to planet-plasma interactions

L.Leclercq[a,b,*], R.Modolo[a,b,*], F.Leblanc[a], S.Hess[c], M.Mancini[d]

[a]*LATMOS-IPSL-CNRS, Paris, France*
[b]*Université Versailles Saint-Quentin en Yvelines, France*
[c]*ONERA, Toulouse, France*
[d]*LUTH, Observatoire Paris-Meudon, France*

Abstract

We present a new method to exploit multiple refinement levels within a 3D parallel hybrid model, developed to study planet-plasma interactions. This model is based on the hybrid formalism: ions are kinetically treated whereas electrons are considered as a inertia-less fluid. Generally, ions are represented by numerical particles whose size equals the volume of the cells. Particles that leave a coarse grid subsequently entering a refined region are split into particles whose volume corresponds to the volume of the refined cells. The number of refined particles created from a coarse particle depends on the grid refinement rate. In order to conserve velocity distribution functions and to avoid calculations of average velocities, particles are not coalesced. Moreover, to ensure the constancy of particles' shape function sizes, the hybrid method is adapted to allow refined particles to move within a coarse region. Another innovation of this approach is the method developed to compute grid moments at interfaces between two refinement levels. Indeed, the hybrid method is adapted to accurately account for the special grid structure at the interfaces, avoiding any overlapping grid considerations. Some fundamental test runs were performed to validate our approach (e.g quiet plasma flow, Alfven wave propagation). Lastly, we also show a planetary application of the model, simulating the interaction between Jupiter's moon Ganymede and the Jovian plasma.

Keywords: Hybrid Model, Mesh Refinement, Particle Splitting, Planet-plasma interaction

1. Introduction

Numerical models describing solar system plasmas objects such as planets, moons, asteroids and comets, have been widely developed and used since the 80's. A complete description of phenomena occuring at electron and ion scales requires a full kinetic approach and is achieved by solving the Vlasov-Maxwell equations. These relations determine the evolution, in space and time, of the electromagnetic field and distribution function for each charged particle species in the region of interest. Due to computational limitations, most of these models are generally not capable to describe the entire physical system of an ionized planetary environment and are often restricted to investigate local processes. However, simulation efforts are undertaken to fully kinetically simulate 3D magnetospheric environments [43]. To study large-scale phenomena it is therefore required to simplify the fully kinetic approach by abandoning particle approach. The usual procedure to decrease the complexity of the model consists in a substitution of the Vlasov equation for a given species by a hierarchy of time-dependent momentum equations (up to a given order) of the distribution function integrated over velocity space. When the reduction is applied to all plasma species we obtain a magnetohydrodynamic (MHD) model.

The three main formalisms used in modeling space plasma environments are: the MHD approach which describes the plasma as a conducting fluid, the 'particle' approach such as the Particle-In-Cell (PIC) models which describes the kinetic behavior of both ions and electrons, and the Hybrid approach that uses a fluid description for electrons and a kinetic description for ions. A detailed review of model assumptions, numerical methods and limitations are described in [33, 23, 46]. Due to the necessity to describe phenomena on scales several times the planetary radius, MHD and Hybrid models are often preferred for modeling global planetary environments. Interestingly, the 2D MHD-EPIC (MHD with Embedded Particle-In-Cell regions) developed by [8], has the ability to describe kinetic

---

effects in a relatively localized region of a global MHD model. The model scheme is a two-way coupling of a global Hall MHD model and an implicit PIC model. Such approaches allow to study a magnetosphere in 2D with a small PIC region around the dayside reconnection site.

All solar system bodies are embedded in flowing magnetized and collisionless plasmas with which they interact via energy and momentum-exchange. The properties of the atmosphere/ionosphere of a body play a key role in the interaction of solar system bodies with plasma flow [35]. The features of the interaction can be very different depending upon the nature of the plasma flow (fast or slow), the characteristics of the atmosphere/ionosphere (density, composition) and the magnetic nature of the obstacle (magnetized object or not). The electromagnetic coupling of neutral environments takes place through ionization processes: ionization by solar photons, electron impact ionization (incident plasma electrons ionize the upper atmosphere), and charge exchange between ionized and neutral particles producing a cold ion and a fast neutral. When the ion gyroradius represents a significant fraction of the obstacle radius, a kinetic description is the most consistent approach to describe the interaction of the incoming plasma and the planetary environments. In such cases (e.g. Mars, Venus, comets, asteroids, several kronian and jovian moons), hybrid models are, so far, best suited to globally describe such interactions at some distances from the ionosphere of the objects [33, 34]. Besides the kinetic aspect, they stand out by their ability to have a multi-species description of the plasma allowing the description of several ion species (although multi-fluid model posseses similar ability), the possibility to represent a non-Maxwellian distribution, the aptitude to take into account self-consistent charge exchange reactions, and diagnose ion distribution function in various regions and regimes.

The spacing of the grid points determines the local error and thus the accuracy of the solution, and the length and temporal scales of the simulated processes if the algorithm is able to reach them. During the last few decades, considerable efforts have been carried out to improve the spatial resolution in all of the different model aproaches (MHD, Hybrid and PIC). Improvements were usually handled through mesh refinement methods.

For kinetic models (PIC and Hybrid), spatial refinement of a specific region of the computational domain is usually achieved either 1- by moving mesh point algorithms which provide a direct control of the mesh points or 2- by adding mesh points in required zones. The first method, called the Moving Mesh Adaptation (MMA) method, consists of mapping a non-uniform adaptive grid to a uniform mesh. This approach was originally proposed by [47], generalized by [5] and extended to include direction control in mesh adaptation to numerically solve partial differential equations [4]. A variant of this concept has been applied to PIC code to adapt the grid specifically designed to capture the physics of objects-plasma interaction [31]. Also, recently, a new approach to optimally create non-uniform meshes based on Monge-Kantorovich optimization was developed in a time-independant case [9] and including a time stepping [6]. The second approach consists of superimposing finer grids to a coarse grid in localized region requiring more resolution. Finer grids can be added recursively leading to a hiearchy of grid refinements. As the solution converges, identification of regions where a better resolution is required are determined by some parameter characterizing the solution. Such methods are commonly known as Adaptive Mesh Refinement (AMR) method. Compared to MHD models using AMR techniques, an additional problem encountered by the kinetic approach is the treatment of numerical particles when they move from a given refinement level to another. Most of the time, the number of numerical particles in each cell is controled by the splitting and coalescence of particles [30]. This technique has been widely used in PIC models [14, 15, 13, 38]. This problem becomes even more challenging when the AMR technique is applied on a MPI (Message Passing Interface) parallel code. It introduces an unbalanced workload among the processors and decreases the efficiency of the code. Similarly, hybrid models investigating global interaction between the solar wind and the Martian or Hermean environment have used nested grid or AMR techniques [22, 24, 39, 40]. A detailed review of AMR and MMA techniques applied to particle code is presented in [32]. As mentioned before the main problem with classical Hybrid and PIC codes remains the treatment of the numerical particles. Numerical particles usually have a constant size equivalent to the grid size. When a numerical particle moves to a finer region, either the size of the particle is kept constant and the size does not match the grid size any longer, or the particle size adapts itself to the local grid size and loses the constancy of the particle size. This second option is mainly used in most AMR based models. Notice that some PIC models apply methods that allow to reduce the self-forces errors and to take into account the special configuration of boundaries between two refinement levels [7].

A new innovative approach concerns a Multi Level Multi Domain (MLMD) technique [19]. In this method, the

physical domain is divided into several subdomains, each corresponding to a different refinement level. Numerical particles are initialized in each domain, with sizes corresponding to the grid spacing. Both electromagnetic fields and particles are fully simulated on each subdomain of each level, allowing particles to remain in their native subdomain. Indeed, particles that leave refined subdomains are deleted from the simulation. At boundaries of refined subdomains, information such as position, velocity and statistical weight of particles of coarser grids are used to repopulate the refined grids through a splitting algorithm adapted from [30]. The MLMD technique has been tested in 1D [19] and in 2D [1].

Whereas most models using AMR techniques break the constancy of the particles' size, the MLMD technique allows preservation of the spatial shape function of particles which is consistent with the grid spacing in every level of refinement. Nevertheless, in this approach, the continuity of particle motion is not ensured since particles are deleted when leaving a refined level to a coarser grid. Therefore, the self-consistency of the algorithm is possibly altered. In addition, in regions where several refinement levels overlap, both particles and fields are simulated in each refinement level, increasing the computational cost.

In this study, a new algorithm for Multi-Grid approach is proposed in the case where one coarse grid and one refined grid are considered, the grid spacing of the coarse mesh being divided by two in the refined region. First, similarly to MLMD method, particles are injected in the refined regions following a splitting algorithm that creates refined macroparticles from one coarse macroparticule. Secondly, particles leaving refined region continue their trajectory in the coarse mesh, keeping their size shape function constant. Those two criteria allow to conserve the consistency of both particles' size and the continuity between the different levels of refinement. In the coarse mesh, the computation of grid moments and forces acting on particles has been adapted to take into account the presence of refined particles in coarse cells. In most of PIC and hybrid models, particles are coalesced in order to control the number of particles per cell. Nevertheless, the coalescence of particles alters the distribution function, and does not ensure the continuity of particles' trajectories. For these reasons, particles are not merged in this model. Another novelty of this approach is that, at the interfaces, moments and fields are obtained taking into consideration the specific volume of the influence region of grid points.

This paper is organized as following. Section 2 describes the Hybrid approach, the algorithm used and the boundary conditions. Section 3 presents the detailed method to achieve the mesh refinement and the treatment of numerical particles and fields at the refinement interfaces. Simulations performed with this model mentioned above are presented to validate the approach in section 4. Next, we show an application to Ganymede with a with a 120 km resolution of Ganymede's environment. Lastly, results and conclusions are discussed in part 5.

2. The hybrid model

2.1. Model description

Hybrid models are able to describe physical phenomena at ion scales, which is not viable with the MHD formalism. In the hybrid approach, electrons are treated as zero-inertia fluid ($m_e = 0$), enforcing instantaneous charge neutrality of the plasma: it exerts a pressure on the ions via the electric fields and contributes to the total current density. Thus it restricts the validity of the model to spatial scales which are much larger than the electron inertial length. Ions are represented by individual particles, called macroparticles. A macroparticle does not represent one ion but a cloud of ions with similar properties (same charge, mass, velocity). Macroparticles behave like physical particles and follow the Newtonian-equations of motion (1) :

$$\begin{cases} \dfrac{d\mathbf{v}_p}{dt} = \dfrac{q_p}{m_p}(\mathbf{E} + \mathbf{v}_p \times \mathbf{B}) \\ \dfrac{d\mathbf{x}_p}{dt} = \mathbf{v}_p \end{cases} \tag{1}$$

where $\mathbf{x}_p$, $\mathbf{v}_p$, q_p and m_p respectively are the position, the velocity, the charge and the mass of the macroparticle p. $\mathbf{E}$ and $\mathbf{B}$ are the electric and magnetic fields. Due to the charge neutrality relation, the electron density corresponds to the sum of ion species densities: $n_e = \sum_s n_s$ where n_e is the electron number density and n_s is the ion number density of the species "s". The density n_s of species "s" is determined from the distribution function of the species "s" f_s:

$$n_s(\mathbf{x}, t) = \int f_s(\mathbf{x}, \mathbf{v}, t) d^3 v \tag{2}$$

while the total ion charge density is derived from

$$\rho(\mathbf{x}, t) = \sum_s q_s \int f_s(\mathbf{x}, \mathbf{v}, t) d^3 v \tag{3}$$

The bulk velocity and the ionic current of species "s" is therefore

$$\mathbf{U_s}(\mathbf{x}, t) = \frac{1}{n_s(\mathbf{x}, t)} \int \mathbf{v} f_s(\mathbf{x}, \mathbf{v}, t) d^3 v \tag{4}$$

$$\mathbf{J_s}(\mathbf{x}, t) = q_s n_s(\mathbf{x}, t) \mathbf{U_s}(\mathbf{x}, t) \tag{5}$$

The total ionic current corresponds to the sum of ionic current of all of ionic species ($\mathbf{J_i} = \sum_s \mathbf{J_s}(\mathbf{x}, t)$). The electric field is determined by the balance of forces exerted on the unit volume of the fluid:

$$0 = qn_e(\mathbf{E} + v_e \times \mathbf{B}) - \nabla p_e \tag{6}$$

with p_e the electron pressure. Since low frequency phenomena are of interest the displacement current term can be neglected in Ampère's equation and the electric field can thus be computed as a state function:

$$\mathbf{E} = \frac{1}{\rho_c} \left(-\mathbf{J}_i \times \mathbf{B} + \frac{1}{\mu_0} (\nabla \times \mathbf{B}) \times \mathbf{B} - \nabla p_e \right) \tag{7}$$

The temporal evolution of the magnetic field is obtained from Faraday's equation (8) while satisfying the conservation of magnetic flux (9).

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \tag{8}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{9}$$

Several algorithms have been developed to solve the time integration of the coupled system of equations 1-9. This model uses the Current Advanced Method and Cyclic Leapfrog (CAM-CL) algorithm [36]. [36] presents a detailed algorithm with validation tests.

2.2. Boundary conditions

Simulations are performed on a 3D cartesian grid. The **X** axis corresponds to the main direction of the flowing plasma. The incident plasma is injected at the entry plane **YZ** following a Maxwellian velocity distribution. Boundary conditions are open at the **YZ** exit plane: particles that leave the simulation domain in this direction are removed from the simulation. In the **Y** and **Z** directions, two types of boundary conditions are applied depending on the origin of the particle. Periodic boundary conditions are used to deal with particles coming from the incident plasma flow. On the contrary, planetary particles are lost when leaving the simulation domain. The electric field is set to $\mathbf{E} = -\mathbf{v} \times \mathbf{B}$ at the entry plane **YZ** (Dirichlet conditions), whereas the electric field gradient is equal to zero at the outer plane **YZ** (Neumann conditions). In **Y** and **Z** directions, periodic boundary conditions are applied for electromagnetic fields.

When a particle hits the planetary body, its velocity is fixed to 0, and the particle is kept in the simulation. This particle therefore still contributes to grid points density. Everywhere in the simulation domain, including inside the planet, the magnetic field is smoothed (see section 2.3). If the planetary object has its own intrinsic magnetic field, the principle is to subtract the dipolar components of the simulated magnetic field before smoothing it. Next, dipolar components are added to the smoothed magnetic field. In other words, at planetary boundaries, the magnetic field is equal to a magnetic dipole plus a draped magnetic field contribution.

2.3. Smoothing

At each time step, all grid moments are smoothed. In 1D, moments of each grid point x_j are smoothed using the three grid points x_{j-1}, x_j and x_{j+1}. The smoothing method is described by the equation: $m_j = 0.25m_{j-1} + 0.5m_j + 0.25m_{j+1}$, where m_j designates the moment on the grid point x_j. This approach is applied in each direction to smooth both the moments and the magnetic field. .

3. The multi-grid method

3.1. Grid refinement

In this paper, only one refinement level is considered in a specific (cubic) region of the simulation domain. This region is determined at initialization and does not change in time. The spatial resolution is divided by two in each direction with respect to the coarse grid defining refined cells with a volume eight times smaller than the coarse cells' volume. The method presented here considers only one refinement level, where the grid spacing is divided by 2 with respect to the grid spacing of the coarse mesh, but could be adapted in order to introduce several refinement levels or to consider higher resolution jumps between the two grids.

In several fluid models, each refinement level solves equations with its own time step adapted to the spatial resolution of the grid, as in [12]. It allows to reduce the number of numerical operations, especially in the coarser grids where time steps can be higher. However each level must wait for other levels before performing the next iteration, as explained by [38] which compute fields with a different time step depending on the refinement level. In this model, to avoid the introduction of a waiting time between the different levels, the same time step is kept constant in each grid. Since the refinement ratio between the grids is small, the use of a different time steps adapted to each refinement level in our model is not planned. However, note that if the resolution jump between refined and coarse grids was larger, different time steps in each refinement level should be considered for stability reasons, similarly to the MLMD method [18].

3.2. Parallelization

Originally, this hybrid model was executed sequentially. Therefore, the simulation time was too long and the size of the simulation box was strongly constrained due to the limited memory associated to a processor. To overcome this computational limitation, the code has been parallelized with the Message Passing Interface protocole (MPI, [11]). The simulation domain is a 3D cartesian box with a main axis $\mathbf{X}$ which corresponds to the bulk flow direction. The parallelization is based on a domain decomposition method. The grid topology of the decomposition is 2D and is done on the $\mathbf{YZ}$ plane, no decomposition being taken into account along the $\mathbf{X}$ direction. Such decomposition minimizes the communication between processes for the particle administration, i.e it limits the number of particles exchanged from one sub-domain to another.

The domain decomposition can be uniform, where each sub-domain has the same volume, or non-uniform. Core processes administrate sub-domains in the center simulation box, and border processes administrate sub-domains that constitute the boundaries of the simulation box, relatively far from the main region of interaction. When a simulation is dedicated to model a dense ionosphere (like Mars or Titan), or with a refined grid level, particles are expected to be spatially located close to the obstacle. Thus it introduces an non-ideal work balance between core processes and border processes. The use of a non-uniform domain decomposition allows for a balance of CPU workload more efficiently depending upon the property of the simulated environment. The domain decomposition scheme currently applied in this model consists in creating twice more subdomains in a certain region whose boundaries are choosen by the user. Generally, in this model, this region corresponds to the close environment of the planetary object, where a dense atmosphere and/or ionosphere are located. The domain decomposition is fixed in time. Note that a minimimum/maximum number of particles per cell is not imposed in this model.

Due to the domain decomposition procedure, sub-domain boundaries do not coincide with refinement outer area, in order to simplify the communications and their treatments in a generic way.

A particle is not allowed to cross more than one grid cell during one time step due to the Courant-Friedrichs-Levy (CFL) condition. Each process can communicate with the neighbor processes only. Communications are mostly "point-to-point" communications and very few global communications are used, except at initialization and for diagnostics. These communications are performed with send/receive directives to send/get data between neighbor processes. Communications are used to exchange particle information between one sub-domain to another, and to compute moments at the interfaces between sub-domains.

In kinetic (PIC and Hybrid) models, the distribution function of the species s of the plasma $f_s(\mathbf{x}, \mathbf{v}, t)$ is described by a set of numerical particles which represent physical particles close to each other in phase space [32].

$$f_s(\mathbf{x}, \mathbf{v}, t) = \sum_p f_p(\mathbf{x}, \mathbf{v}, t) \tag{10}$$

Equation (10) mathematically translates this concept with $\mathbf{x}$ and $\mathbf{v}$ being the position and velocity vectors of physical particles respectively and $f_p(\mathbf{x}, \mathbf{v}, t)$ defined as the distribution function of the numerical particle p. $f_p(\mathbf{x}, \mathbf{v}, t)$ is generally expressed in the following way :

$$f_p(\mathbf{x}, \mathbf{v}, t) = \mathrm{N}_p S_x(\mathbf{x} - \mathbf{x}_p) S_v(\mathbf{v} - \mathbf{v}_p) \tag{11}$$

In this equation, N_p is the number of physical particles present inside the numerical particle p while S_x and S_v are its shape functions (in spatial and velocity spaces), $\mathbf{x}_p$ the mass center and $\mathbf{v}_p$ the velocity of the numerical particle. The number of physical particles represented by a given macroparticle is referred to as statistical weight. Each numerical particle can represent a different number of physical particles which can describe major and minor ion species with a good statistical sampling at lower computational cost and therefore cover a wide dynamic range of density. S_v is usually a Dirac delta function (equation 12) meaning that all physical particles have the same velocity $\mathbf{v}_p$ as the numerical particle.

$$S_v(\mathbf{v} - \mathbf{v}_p) = \delta(\mathbf{v} - \mathbf{v}_p) \tag{12}$$

[3] summarized the different possibilities to determine the spatial morphology of numerical particle. Particle methods makes use of the b-spline functions to represent the spatial shape factor of a numerical particle [30].

$$S_x(\mathbf{x} - \mathbf{x}_p) = S_x(x - x_p)S_y(y - y_p)S_x(z - z_p) = \frac{1}{\Delta x_p \Delta y_p \Delta z_p} b_l\left(\frac{x - x_p}{\Delta x_p}\right) b_l\left(\frac{y - y_p}{\Delta y_p}\right) b_l\left(\frac{z - z_p}{\Delta z_p}\right) \tag{13}$$

with x_p, y_p, z_p define the position of the center of mass of the numerical particle, $\Delta x_p, \Delta y_p, \Delta z_p$ the size of the numerical particle in each spatial dimension and b_l the b-spline function of order l. The most commonly used b-spline function, which is also the case in this model, is the zero-order b-spline, i.e. a top hat function (in 1D, a cube in 3D):

$$b_0(\xi) = \begin{cases} 1 & \text{if } |\xi| < \dfrac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

This spatial shape factor can be represented in 1D as shown by figure 1 adapted from [34]. This figure shows a one-dimensional mesh indicated by x_j's and the size of the numerical particle with its mass center position x_p. Dashed green lines delimit the area of influence of the grid point j. The fraction of the numerical particle's weight assigned to the point x_j depends on the relative length (volume) of the numerical particle included in the grid point's influence region (IR). The IR of a grid point is the volume within $\pm \Delta x/2$ from this point, where Δx is the grid spacing. Such deposition method, known as Cloud-In-Cell [2, 17], is commonly used in particle simulation. Therefore, the charge density affecting the grid points, according to the example on figure 1, are indicated by equations 15.

$$\rho_{j-1} = q_p w_p \left(\frac{x_j - x_p}{\Delta x_p}\right) \quad ; \quad \rho_j = q_p w_p \left(\frac{x_p - x_{j-1}}{\Delta x_p}\right) \tag{15}$$

with Δx_p the size of the numerical particle, which equals the grid spacing. Extension to the three-dimensional model is straigthforward. CIC approach smoothly treats moments such as charge density on the grid points as illustrated on figure 1 (bottom panel). Notice that according to the CIC approach, particles' size corresopnds to the size of grid cells. However, in the multi-grid method presented in this paper, particles whose volume differs from grid cells' volume will be considered (see section 3.3.2).

Numerical particles are therefore a discretization of the distribution function. This introduces numerical noise. It is possible to improve the accuracy in some regions, even without the introduction of finer levels, by increasing the number of particles in regions of interest. The most common method, described by [30], consists of a splitting algorithm that creates several daughter particles from a parent particle, each daughter particle conserves the
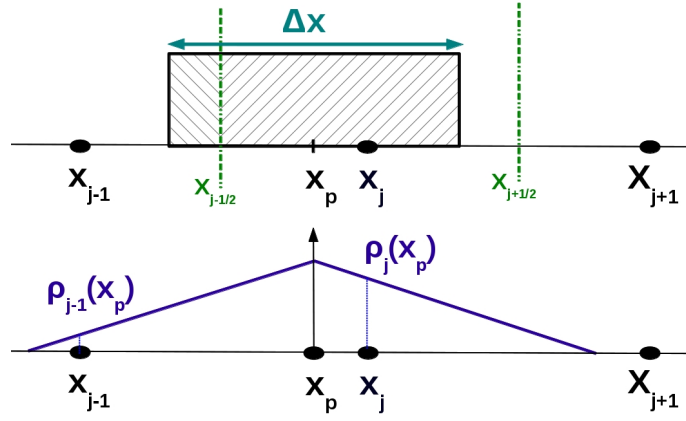
Figure 1: On the top panel is a representation of the first-order particle weighting (CIC). The macroparticle whose mass center is $\mathbf{x}_p$ has a finite size corresponding to that of the grid cell. The moments are interpolated on grid points allowing to obtain a fairly smooth repartition of moments in mesh nodes (bottom panel) [34]

velocity and the shape size function of the initial particle. To compute the position of the new particles, a small displacement with respect to the mass center of the parent particle is introduced. This technique has been widely employed in PIC codes [15, 14, 13] and in hybrid models, in particular by [22, 24] and [39], in order to keep the number of macroparticles per cell in the grid simulation constant.

3.3.1. From coarse to fine mesh: particle splitting

Most of hybrid and PIC models apply the AMR method [38, 39, 22, 24, 13]. In these models, when a particle enters in a grid whose spatial resolution is different, its size/volume is adapted to the cell's size/volume. In other words, this particle is replaced by a particle which has a different volume. First, the constancy of the particle's size shape function is not ensured. In consequence, since a macroparticle respresents a number of physical particles, the density which represents a macroparticle is not equivalent in the different refinement levels. Therefore, when the size of a particle is changed, its contribution to the grid moments are changed too. According to [30], when a set of particles is replaced by another group of particles, the two sets must be equivalent in term of contributions to the grid moments. In order to respect this criterion, a different approach is used here to treat macroparticles that enter in a refined grid.

The main idea of our approach is based on the concept of uniformly distributed physical particles inside the volume of a numerical particle. When a numerical particle moves from the coarse to the refined grid, it is split into eight daughter particles, i.e the cloud of physical particles is divided into eight cubes with a volume equal to the refined cell volume. Here parent (respectively daughter) particle refers to a particle having a volume of a coarse (respectively refined) cell. Each daughter particle has the same velocity as its parent particle and the same chemical identity. This approach is summarized on Figure 2 in a 2D case. Particular attention must be drawn on
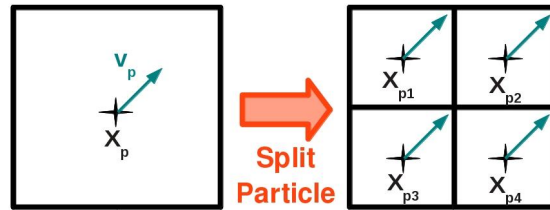

Figure 2: Schematic representation of the method used to split numerical particles. The parent particle, with a velocity v_p and a position x_p, is split into daughter particles. These particles have the same velocity as the parent particle, and their position is equally shifted from the parent particle's mass center.

7

the interface between the different grids (section 3.4). The finer mesh is only populated by daughter particles, and particles are divided one cell before (i.e. outside) the refined grid boundaries as explained in section 3.4. Such splitting approach is similar to the 1D technique used by [19] to populate refined mesh from parent particles, as applied by [1] in 2D. The only notable difference is that we do not simulate the whole physical domain with the coarse mesh. In other words, we do not have any information about particles or fields calculated on the coarse grid in the refined region.

Mass centers of the eight daughter particles are shifted away from the parent particle mass center by one quarter of a coarse cell in each direction (that is, at the barycenter of each eight parent sub-volumes). The volume of the parent particle is, after the splitting process, composed of eight homogenous cubes ensuring conservation of grid moments. Unlike [39], a random shift of the position of newly created particles is not introduced here, due to a numerical noise limit, which is reasonable in this setting.

According to equations (10,12,13), the distribution of the numerical particle f_p can be written as

$$f_p(\mathbf{x}, \mathbf{v}, t) = w_p b_0 \left(\frac{\mathbf{x} - \mathbf{x_p}}{\Delta \mathbf{x_p}} \right) \delta(\mathbf{v} - \mathbf{v_p}) \tag{16}$$

with $w_p = \frac{N_p}{\Delta x_p \Delta y_p \Delta z_p}$ the number of physical particle contained in the volume of the numerical particle. The volume of a numerical particle is $\Delta V_p = \Delta x_p \Delta y_p \Delta z_p$. w_p is the number density associated to the numerical particle p. This statistical weight is used to compute charge or mass density. When a particle is split, the number of physical particles and the volume of the numerical particle both are divided into eight: $N_p^d = N_p^p/8$ and $\Delta V_p^d = \Delta V_p^p/8$ where N_p^d and N_p^p are the number of physical particles in the daughter and parent particle populations respectively, and ΔV_p^d (ΔV_p^p) are the volume of the daughter (parent) numerical particle. The statistical weight $w_p = \frac{N_p^p}{\Delta V_p^p} = \frac{N_p^d}{\Delta V_p^d}$ remains constant. Similarly to [30], each daughter particle acquires the velocity of the parent particle in order to leave the velocity distribution unaltered and to conserve moments such as current density. As mentionned previously, the set of created particles replacing the parent particle is equivalent to the initial particle obeying the splitting particle conditions outlined by [30] where 1- the contributions of the two sets of particles to the grid moments are identical, and 2- the two sets of particles describe the same velocity distribution function. In AMR models, only the second criteria is respected when the volume of a particle is adapted to the spatial resolution of the different refinement levels.

However, applying the same velocity to each daughter particle still introduces a numerical error and slightly breaks down the continuity of the motion of the particles. Indeed, daughter particles positions are shifted with respect to the parent particle's mass center. Therefore, daughter particles are shifted during the splitting without solving the equation of motion. In other words, an artificial acceleration or deceleration is applied to the daughter particles. To avoid such an error, a velocity correction should be applied to each daughter particles during the splitting step. Tests performed with a 1-D electrostatic code (not shown in this paper) with the implementation of such velocity corrections show that the solution does not significantly improve while additional computations are needed to calculate the new velocity of the daughter particles. For these reasons, no velocity correction is applied to the daughter particles.

3.3.2. From fine to coarse mesh

The strength and the novelty of the presented method is that particles from the refined grid are able to move into the coarse grid, allowing a continuous description of the particles' trajectory. The numerical particle keeps the information about its size (ΔV_p^p or ΔV_p^d) to respect the criteria described in [30], as explained in section 3.3.1, unlike in the AMR methods [22, 24, 39]. Daughter particles are neither split nor merged. Therefore, in the coarse grid case, both daugther and parent particles can be present. A special attention is then required to achieve the grid moments calculations from daughter particles which have one eigth of the volume of the local cell. A daughter particle evolving in a coarse cell can either affect one or several grid points. When all the weight of the particle is introduced to a single grid point, the deposition technique is similar to a Near Grid Point (NGP) method [3]. In the coarse region, to compute the charge density taking into account the size of particles, in the 1D case, equations 15 become :

$$\rho_{j-1} = q_p w_p \frac{\min \left(\Delta x_p, \max \left(x_{j-1/2} - b_p^{inf}, 0 \right) \right)}{\Delta x} \quad ; \quad \rho_j = q_p w_p \frac{\min \left(\Delta x_p, \max \left(b_p^{sup} - x_{j-1/2}, 0 \right) \right)}{\Delta x} \tag{17}$$

295 where Δx is the grid spacing, Δx_p designates the size of the macroparticle p, and where $b_p^{inf} = x_p - \Delta x_p/2$ (respectively $b_p^{sup} = x_p + \Delta x_p/2$) corresponds to the lower (respectively upper) boundary of the macroparticle. This formulation allows to treat both coarse and refined particles, whose size respectively corresponds to the grid spacing, and to the half of the grid spacing. The extension to the 3D case is straightforward. Such a technique is known to produce numerical noise, which is partly reduced since statistically eight times as many daughter

300 particles will be present in a coarse cell of the computational grid. In order to avoid 1-to change the volume of refined particles that enter in a coarse mesh, and 2- to consider particles whose size is different to the grid spacing how it is done in the presented method, in the MLMD method [19], particles and fields are fully simulated in all refinement levels. Indeed, in the coarse levels, fields are obtained from grid moments of refined levels. The motion of coarse particles is then influenced by these fields. Therefore, it allows to delete refined particles that leave their

305 refined grid. This method does not ensure the continuity of particles' trajectories, and alters the self-consistency in the coarse grids. Moreover, as particles and fields are simulated in all entire grids, this technique tends to increase the computational cost with respect to the method described in this paper. However, using our method at high jumps in spatial resolution, the number of macroparticles would be too large and the MLMD technique would probably be more efficient.

310 Obviously, merging refined particles to create coarse particles would allow one to deal with less macroparticles effectively reducing the computational cost. The coalescing particles method described by [30] has successfully been applied to lots of models using mesh refinement methods [22, 24, 39, 19] in order to respect a criterion about maximum number of particles per cell. This method consists of selecting three computational particles close in space, representing the same species with very similar velocities in order to generate two macroparticles combining

315 the velocities of parent particles. This method leads to the creation of particles having the same spatial shape function than the parent particles. Here, to create a coarse particle, eight refined particles should be chosen under the same criteria: particles have to be close in space, in velocity, and to have the same chemical identity. Because of the low number of particles per cell (typically about 10), these criteria are quite difficult to respect. Moreover, a coalescence algorithm introduces a discontinuity in particles trajectories and alters the velocity distribution

320 function.

3.4. Moment computation at the interfaces

Introducing a refined mesh, leads to a peculiar configuration of the grid points at the interfaces. To deal with moment computation in this special case, a prevalent solution is to create some overlaping regions on which numerical particles are copied before entering into the refined grid [38]. Usually, extrapolation and/or interpolation

325 of moments from both coarse and refined grids are applied to determine moments on refinement levels interfaces. To avoid additional computations in these overlap regions, we have implemented a new original method to evaluate moments and fields contributions to interfaces grid points, taking into account IR volume. Figure 3.4 displays the structure of the mesh at the refinement interface between the two grids (in 2D). In the following, Δx^c and Δx^r designate the spatial resolution of the coarse and refined grids respectively. Blue and red cells represent the coarse

330 and refined cells while the dashed green and blue cells describe the IRs. The volume of the IR of a grid point corresponds to the grid spacing size, except at the interfaces where the IR of grid point size differs. For simplicity the evaluation of moments and fields at the interface are described for a 2D case (Figure 3.4). The weight deposition of a numerical particle on a specific grid point depends on the area (volume) of the numerical particle included in the area (volume) of the IR's grid point, as stipulated by the CIC concept. The area of the IR's grid point close

335 to the refinement level interface is different if we consider a grid point which belongs to a vertex or corner of the refined mesh. In case of an edge's grid point, the area of the corresponding IR is $\Delta x^r \times \frac{3}{2}\Delta x^r$. In case of a vertex grid point, the IR's area is changed to $\frac{3}{2}\Delta x^r \times \frac{3}{2}\Delta x^r$. The 3/2 coefficient comes from the refinement factor between the spatial resolution of the coarse and refined grid ($\Delta x^r = \frac{\Delta x^c}{2}$). Obviously, this coefficient will be different in case of a different refinement factor, but the method described below remains valid. Moment contributions calculated by

340 equation 15 need to be multiplied by the 2/3 (or 4/9) factor for an edge (vertex) grid point. Although, extension to the 3D case is straightforward, new IR's have to be considered for grid points belonging to the faces of the interface. Such methods avoid using overlap regions where interpolation and/or extrapolation rules are usually applied to exchange moments between two refinement levels. The concept can easily be applied to multi-refinement levels and/or different spatial resolution factors.

345 Coarse particles whose center of mass is close to the refined grid can affect moments of internal refined grid
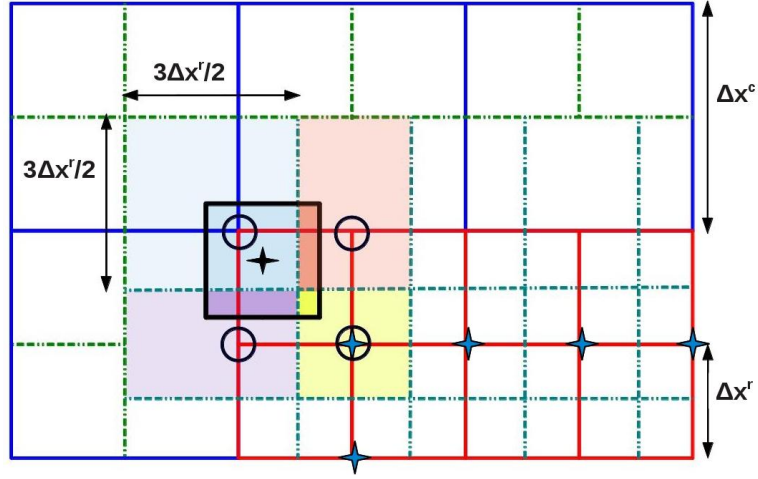
Figure 3: Computation of the grid point moments at the interfaces between the two grids: size of the IR (influence region) of the interface nodes, and weighting computation to calculate moments at the interface points in a 2D case.

points (c.f blue crosses on figure 3.4) due to their volume. However, grid moments on these internal points are well calculated from refined particles within the refined grid. The contributions of the coarse particles to these points then could lead to an overestimation of moments on these nodes. In order to avoid this, and since refined particles are authorized to retain in the coarse grid, coarse particles are split one cell before interfaces between the two refinement levels. In other words, coarse particles are divided in the coarse grid, when they are within Δx^c distance from an interface in all directions. Like this, particles whose center of mass is in a coarse cell can affect only interface nodes of the refined grid, and do not contribute to the moments of internal refined grid points.

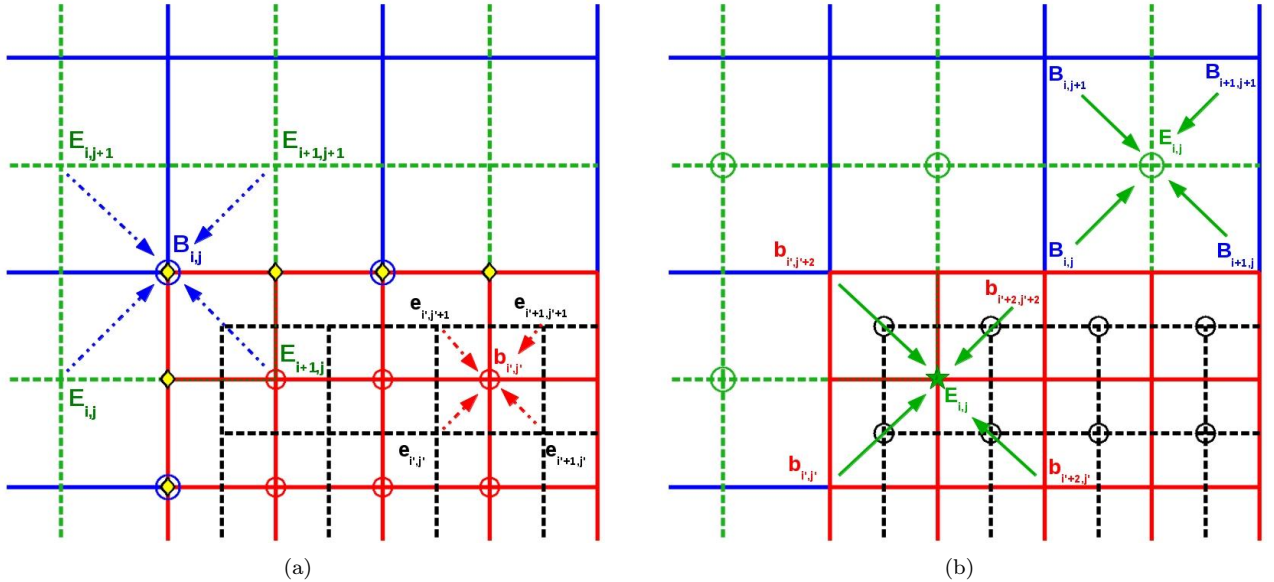### 3.5. Electromagnetic field at the interfaces

3.5.1. Computation of fields

Figure 4: Computation of electromagnetic field at interfaces between the two grids. In both right and left panels, B, b, E and e (defined in the text) grids respectively are displayed in full blue lines, full red lines, dashed green lines and dashed black lines. - (a) : Blue and red circles represent the nodes where magnetic field is computed and yellow diamonds are interface points where special calculations are made to obtain the magnetic field (details are given in the text). - (b) : Green and black circles represent the nodes where electric field is computed. Green stars display nodes needed some peculiar calculations (details are given in the text).

Electric and magnetic fields are solved on two different grids separated by half cells in order to approximate

second-order spatial derivatives and gradients by finite-differences [3]. The magnetic (respectively electric) coarse mesh is called B (respectively E) and the magnetic (respectively electric) refined mesh is called b (respectively e).

The magnetic field is obtained from integrating Faraday's equation (8). Figure 4(a) displays the procedure used to compute the magnetic field, in both B and b grids, including interfaces. In the following method described in 2D, the adaptation to the 3D case is straigthforward. $\mathbf{B}_{i,j}^n$ (respectively $\mathbf{b}_{i',j'}^n$) designates the magnetic field computed on the (i,j) grid point of the coarse B grid (respectively the (i',j') of the refined b grid) at iteration n. The same notation is used for the electric field. Equations below allow one to calculate the magnetic field at iteration $n+1$ from the magnetic field at iteration n and from the electric field at iteration $n+1/2$. dt is the time step.

- In the coarse B grid (in blue), including grid points at interfaces between grids B and b, electric field values from the coarse E grid (in green) are used: $\mathbf{B}_{i,j}^{n+1} = \mathbf{B}_{i,j}^n - dt(\nabla \times \mathbf{E})_{i,j}^{n+1/2}$.

- Similarly, in the refined b grid (in red), electric field values carrying the refined e grid (in black) are applied: $\mathbf{b}_{i',j'}^{n+1} = \mathbf{b}_{i',j'}^n - dt(\nabla \times \mathbf{e})_{i',j'}^{n+1/2}$. This procedure is applied everywhere in the refined b grid, except at the interface grid points.

- b interface grid points are represented by yellow crosses on Figure 4(a). The magnetic field on these grid points is obtained from the magnetic field of B grid points at corresponding interfaces. On grid points that belong to both B and b grids, the equation $\mathbf{b}_{i',j'}^{n+1} = \mathbf{B}_{i,j}^{n+1}$ is applied. On other interfaces grid points of B_r, interpolation calculations are used: $\mathbf{b}_{i',j'+1}^{n+1} = 1/2 \times (\mathbf{b}_{i',j'}^{n+1} + \mathbf{b}_{i',j'+2}^{n+1})$.

The electric field depends on the charge density ρ_c, the ionic current $\mathbf{J}_i$, the magnetic field $\mathbf{B}$ and the electronic pressure p_e (equation 7). Figure 4(b) displays the method used to compute the electric field in both E and e grids, including the interfaces. In the following, $\rho_{i,j}^n$, $\mathbf{J}_{i,j}^n$, and $\mathbf{P}_{i,j}^n$ (respectively $\varrho_{i',j'}^n$, $\mathbf{j}_{i',j'}^n$, and $\mathbf{p}_{i',j'}^n$) designate the charge density, the ionic current, and the pressure on the coarse grid point (i,j) (respectively on the refined grid point (i',j') at iteration n. The pressure gradient is calculated with a centered finite difference scheme. Calculations presented below allow one to compute the electric field at time n from grid moments and magnetic fields at time n.

- On the coarse E grid (in green) points which are outside the refined b grid (in red), the electric field is obtained from moments and magnetic field of the B grid (in blue):

$$\mathbf{E}_{i,j}^n = \frac{1}{\rho_{i,j}^n} f(\mathbf{J}_{i,j}^n, \mathbf{B}_{i,j}^n, \nabla \mathbf{P}_{i,j}^n)$$

- Everywhere in the refined e grid (in black), the electric field is calculated with moments and magnetic field of the fine b grid (in red):

$$\mathbf{e}_{i',j'}^n = \frac{1}{\varrho_{i',j'}^n} f(\mathbf{j}_{i',j'}^n, \mathbf{b}_{i',j'}^n, \nabla \mathbf{p}_{i',j'}^n)$$

- Some coarse E grid points (green star on figure 4(b) for instance) are within the refined b mesh. To compute the electric field on these nodes, moments and magnetic field of the refined b grid (in red) points surrounding these nodes are used. applying the following equation:

$$\mathbf{E}_{i,j}^n = \frac{1}{\varrho_{i'+1,j'+1}^n} f(\mathbf{j}_{i'+1,j'+1}^n, \mathbf{b}_{i'+1,j'+1}^n, \nabla \mathbf{p}_{i'+1,j'+1}^n)$$

where $\varrho_{i'+1,j'+1}^n$, $\mathbf{j}_{i'+1,j'+1}^n$, $\mathbf{b}_{i'+1,j'+1}^n$ and $\nabla \mathbf{p}_{i'+1,j'+1}^n$ are obtained from magnetic field and moments on refined b grid points (i',j'), $(i'+2,j')$, $(i',j'+2)$ and $(i'+2,j'+2)$.

3.5.2. Electromagnetic fields acting on particles

In this model, to treat particles whose size equals the grid spacing, the size shape function used to compute forces acting on particles is equivalent to that applied to compute grid moments. In 1D, the force F_p applied to a macroparticle p whose center of mass is within grid points x_{j-1} and x_j (see figure 1) is obtained with the following equation:

$$F_p = \left(\frac{x_j - x_p}{\Delta x} \right) F_{j-1} + \left(\frac{x_p - x_{j-1}}{\Delta x} \right) F_j \tag{18}$$

11

where F_j designates the force at grid point x_j, x_p is the center of mass of the macroparticle p, and Δx is the grid spacing (which equals the macroparticle's size).

To compute the electromagnetic field applied to a particle, two cases have to be considered. First, if the mass center of a particle is within the refined region, usual calculations are performed : the magnetic field acting on a particle is interpolated from the magnetic values of the eight grid points surrounding the particle, similar to the CIC approach. Weights are obtained following the method used to calculate grid point moments [3], as described in section 3.4. Only E, B (respectively e, b) grid points are used when the mass center of a macroparticle is within the B (respectively b) grid. Similarly the approach used to compute moments (section 3.4), a combination of B and b magnetic field values could be used to compute the magnetic force applied to particles. We do not use a combination of B, b and E, e for particles close to the interface region since some interface points of the b grid are obtained via interpolation. Therefore, IR of interface grid points are not considered as in section 3.4.

As detailed previously, refined particles are capable to continue their trajectory within the coarse grid. Therefore, the motion of some refined particles is influenced by the electromagnetic field computed on the coarse mesh. Whereas particles' size is considered to compute grid moments in the coarse grid, the algorithm used to compute forces acting on particles is the same regardless of their volume. In other words, equation 18 is applied to compute electromagnetic fields acting on all particles, without taking into account particles' size. In consequence, to treat refined particles in the coarse grid, the scheme used to compute forces acting on particles and the scheme used to compute grid moments are not exactly equivalent. Therefore, the momentum is not exactly conserved [3, 34]. For the same reasons, self-forces are generated. However, some numerical tests show that the impact on the momentum conservation and that that generation of self-forces acting on particles can be neglicted.

4. Results

The hybrid model described here aims at investigating the interaction between a planetary environment (which can include an atmosphere and/or a magnetosphere) with a magnetized plasma such as the solar wind or planetary magnetospheric plasma. However, in order to check the strength and the efficiency of our multi-grid approach, some basic tests were performed. In the following, three simple cases are presented : 1- the flow of a quiet plasma representing the solar wind (supersonic and super-Alfvenic), 2- the jovian magnetospheric plasma (sub-alfvenic flow) and 3- the propagation of an Aflvén wave. Validating the code consists of verifying that no artifact or numerical noise appears at the interfaces between the two grids, verifying that physical parameters such as the energy or momentum and numerical parameters or the number of numerical particles are conserved during the simulation.

4.1. Quiet plasma flow without obstacle

Species	T_i [eV]	β	n_i [cm^{-3}]	Ω_i [rad/s]	
H$^+$	5	0.5	2.3	0.29	
He^{++}	20	0.1	0.12	0.14	
e$^-$	10	1.08	2.42	-	
Parameter	B_0 [nT]	V_{SW} [km/s]	V_A [km/s]	M_A	M_S
Value	3	400	40	10	5

Table 1: Physical parameters used to simulate the solar wind at Mars

A first simple test validating the multi-grid approach is to simulate a flow of a quiet plasma such as the solar wind. The physical parameters of such a flow are summarized in table 1, and correspond typically to standard solar wind properties at Mars orbit. The plasma is composed of protons H$^+$ and alpha particles He^{++} (5%). For each ion species, the first part of Table 1 provides the temperature T_i, the plasma beta parameter β , the density n_i, the cyclotron frequency Ω and the necessary time for an ion to rotate around a magnetic field line Ω^{-1}. The

magnetic field has a magnitude of 3 nT and is oriented along the **Z** axis, the velocity of the plasma is 400 km/s. Considering all these values, the Aflvén speed is about ~ 40 km/s and the speed of sound is set to ~ 80 km/s, the plasma is therefore supersonic and super-Alfvénic.

To run this simulation, we use a coordinate system such that the **X** axis is in the flowing direction the plasma, the **Y** axis being toward the motional electric field $\mathbf{E}_{conv} = -\mathbf{v}_{SW} \times \mathbf{B}$ (where $\mathbf{v}_{SW}$ is the velocity of the solar wind) and the **Z** axis completes the right-handed system. The run test is performed in a 3D simulation box containing $50 \times 56 \times 56$ cells with a spatial resolution of about 150 km in each direction, which corresponds to the proton inertial length of c/ω_{pi} (ω_{pi} is the plasma frequency). $20 \times 24 \times 24$ cells with a spatial resolution of 75 km compose the refined region which is centered in the simulation box. The initial conditions are such that 40 macroparticles are injected in each cell, except in coarse cells sharing a boundary with the refined grid, where 320 particles are injected. To properly describe the ion rotation around magnetic field lines, the time step is choosen such that a proton gyration is described with 100 points : $dt = 0.01\Omega_{\mathrm{H}+}^{-1}$. 20 transit times are simulated, a transit time being defined as the necessary time for a particle to cross the whole simulation domain along the **X** axis.

As shown in figure 5(a), since computational particles are split into eight daughter particles before they enter the refined mesh, the total number of particles in the simulation increases rapidly but then quickly stabilizes. Since no merging particle algorithm is considered, this result was expected. The stabilization of the total number of particles reflects the good behavior of our splitting algorithm. Indeed, this approach does not lead to the creation of too many particles. Figure 5(b) presents the temporal evolution of the thermal and kinetic energies per particle, the magnetic energy per cell and total energy. These quantities vary by less than 0.1% during the whole simulation time (20 crossing times), which demonstrates that energy is succesfully conserved. Figure 5(c) shows the evolution of the x, y and z momentum. Whereas the x and z momentum components present an excellent behavior being constant throughout the entire simulation with fluctuations that do not exceed 0.1%, y-component shows a different evolution with a significant decrease from the beginning of the run up to less than one transit time, and then stabilizes for the rest of the simulation. This peculiar decrease is due to the way particles are split. Indeed, as discussed in section 3.3, a velocity correction of newly split particles should be taken into account to compensate the artificial acceleration (or slow down) imposed to particles. Since the magnetic field is oriented along the **Z** direction, and flows mainly along **X**, the motional electric field has only a E_y component. Thus, particles undergo a global acceleration in the **Y** direction only, explaining this behavior. It should be noted that the y-component of the momentum is nevertheless kept at 1% after the decrease. Figure 5(d) displays the temporal evolution of the standard deviation of the simulated magnetic field in the entire simulation domain including both coarse and refined meshes, for x, y and z components. This quantity, for each component, increases rapidly and then asymptotes. This is due to the initialization step with the magnetic field strength defined uniformly in the entire simulation domain. The natural numerical noise due to the collection of moments on mesh nodes (and then on the electric field) induces the same effects on the magnetic field. As a first approximation, x and y components of the simulated magnetic field are distributed around the mean magnetic field value by 0.2%, whereas it is slightly more enhanced for the z-component with 0.3% due to the principal orientation of the magnetic field toward the **Z** direction. A test performed without the multi-grid approach, shows fluctuations of 0.1% and 0.2% (not shown here) meaning that the numerical noise slightly increases when introducing a refinement level. As discussed before, this numerical noise is certainly due to the way particles are split as suggested by [39]. Similarly, the evolution of the standard deviation of the mean velocity calculated on grid points is shown in figure 5(e) and is almost constant. The standard deviation is slightly smaller in the case where the simulation is performed on a unique uniform grid, but very close.

To further demonstrate the efficiency of the proposed method, we check that the velocity distribution functions are consistent with the expected distributions, for both parent and daughter particles. Figure 6 shows the theoretical and simulated proton velocity distributions. Besides the y-component of daughter particles (see blue curve, figure 6(b)), the simulation matches very well. Indeed, daughter particles are slightly slowed down along the **Y** axis (results are similar to He^{++}). This confirms the results previously discussed concerning the momentum conservation along the **Y** axis. Heating and cooling is however not applied to the daughter particles, implying that our particle splitting algorithm conserves the ion temperature.

Figure 7 presents the electron density in a **XZ** plane that crosses the refined region in the center. Although the numerical fluctuations are about twice larger in the refined grid (8% against 4% in the coarse grid), and since
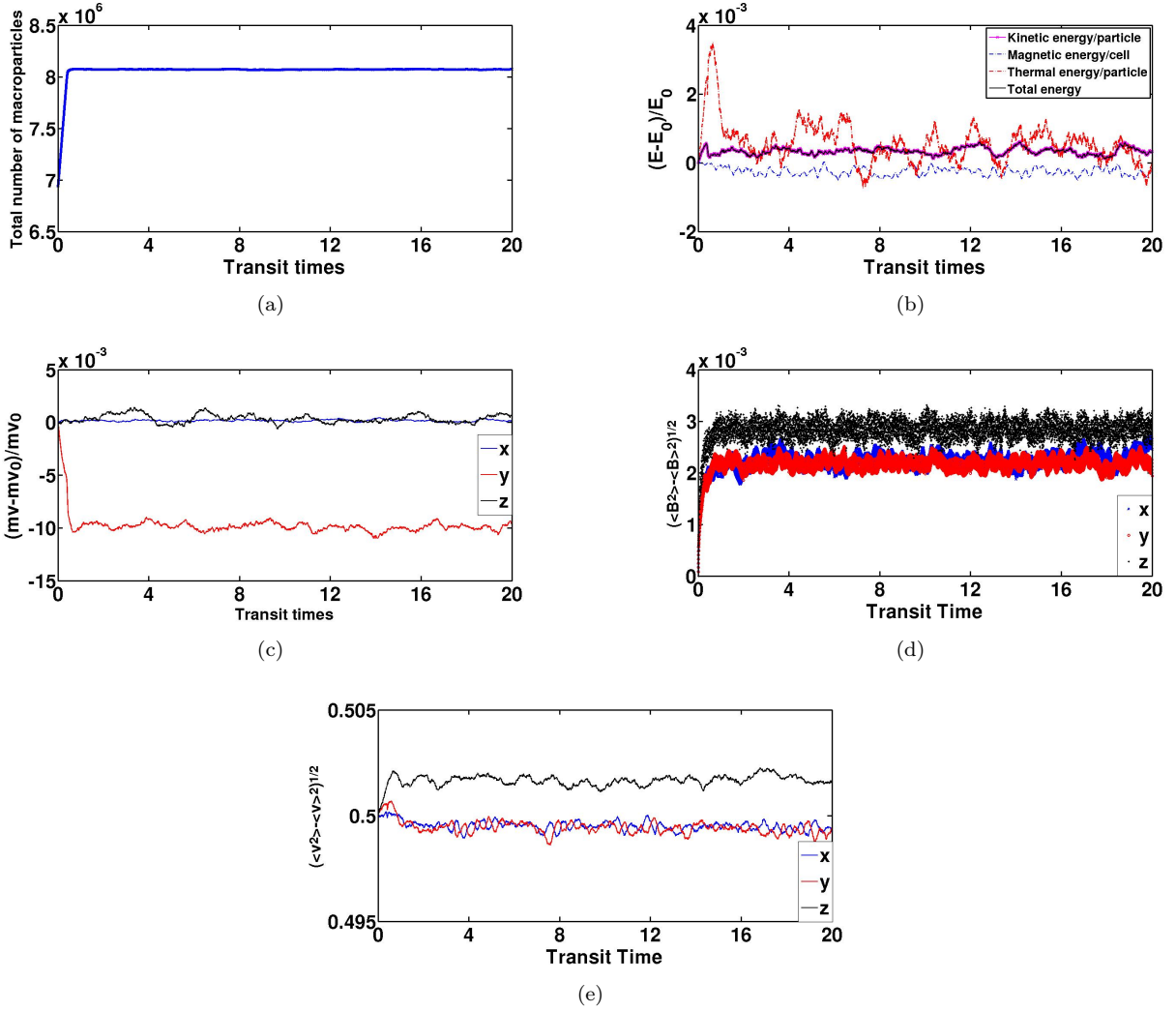
Figure 5: Conservation tests in the case of the simulation of the solar wind. - (a) : Evolution of the total number of particles in the simulation box, including both all macroparticles, in function of time. The reasons of the increase of the number of particles at the beginning is discussed in the text. - (b) : Temporal evolution of the mean kinetic energy per particle (in pink), mean thermal energy per particle (in red), mean magnetic energy per cell (in blue), and total energy (in black). - (c) : Temporal evolution of the total momentum which is normalized with respect to the initial value for x (in blue), y (in red) and z (in black) components. - (d) : Temporal evolution of the standard deviation of the magnetic field in the whole domain for x (in blue), y (in red) and z (in black) components. - (e) : Temporal evolution of the standard deviation of velocity in the whole domain for x (in blue), y (in red) and z (in black) components.

the number of particles per cell is small, the numerical noise remains suitable. The same fluctuations are observed for the electronic or ion density in other planes and for physical quantities such as bulk velocity, temperature or electromagnetic field. Since the daughter particles are equally displaced from the mass center of the parent particle, an additional numerical noise is introduced [39]. In addition, no artificial numerical fluctuations occur at the interfaces between the two levels of refinement, which corroborates the efficiency of our splitting particles algorithm and of the moment computation method used at the interfaces. We also notice that along $\mathbf{X}$ axis, where we introduce a population of computational particles smaller than the grid cell size (that is for $\mathbf{X} > 30$), no artifact or numerical variations occur.

Performances of the parallelization have been evaluated by establishing a scalibility test. Figure 8 displays the linear case in red, the acceleration obtained with the hybrid model without multi-grid approach in blue and with the multi-grid approach in green. Since some computation time is required to perform communications between the processors, the linear behavior can not be reproduced. In the multi-grid case, the refined region manages about eight more particles with respect the coarse region. Thus, each process deal with a different number of particles, impacting the performances of the parallelizetion. Nevertheless, Figure 8 shows a reasonable behavior of the code.

Similar tests have been performed simulating a jovian plasma flow, to check the strength of the multi-grid
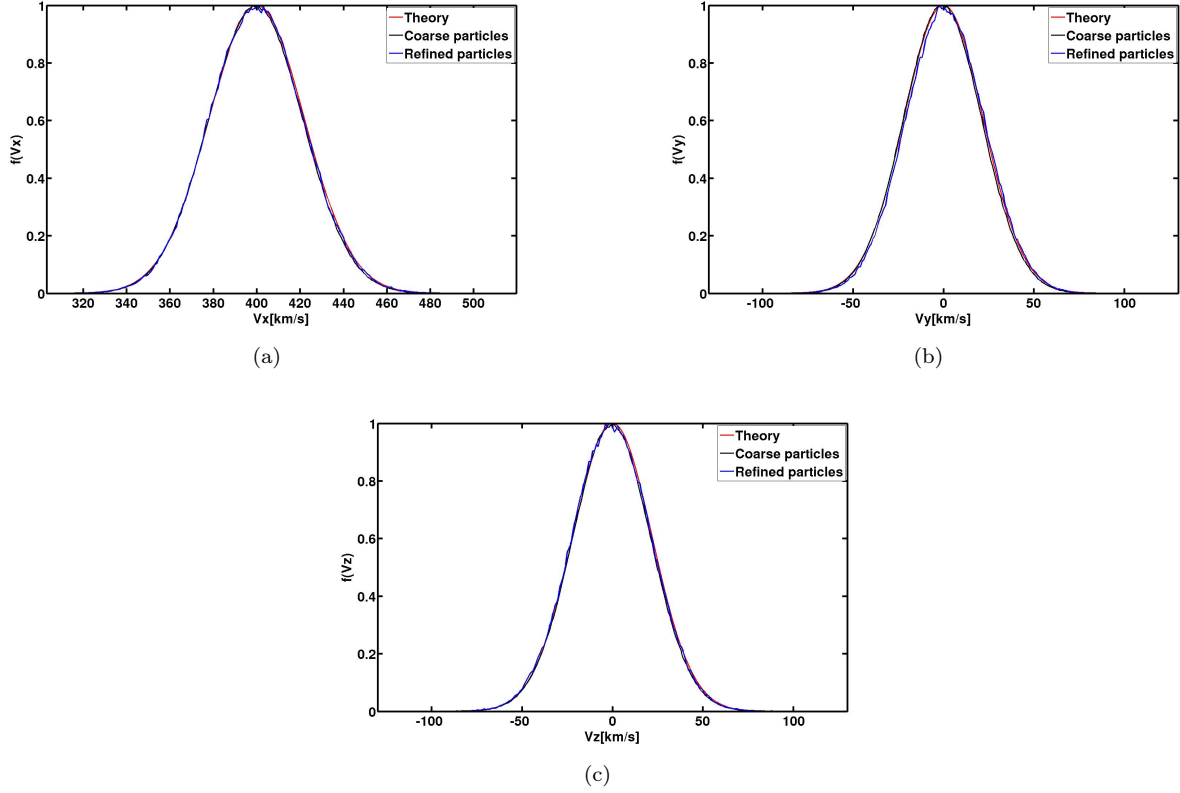
14

(a)

(b)

(c)

Figure 6: Velocity distribution of protons for the test simulation of solar wind. For each component (x in figure (a), y in figure (b) and z in figure (c)), the simulated velocity distribution of big particles is represented in black whereas the blue lines represent the distribution of small particles. In each case, the red curve corresponds to the Maxwell distribution, and the plotted curves are normalized.
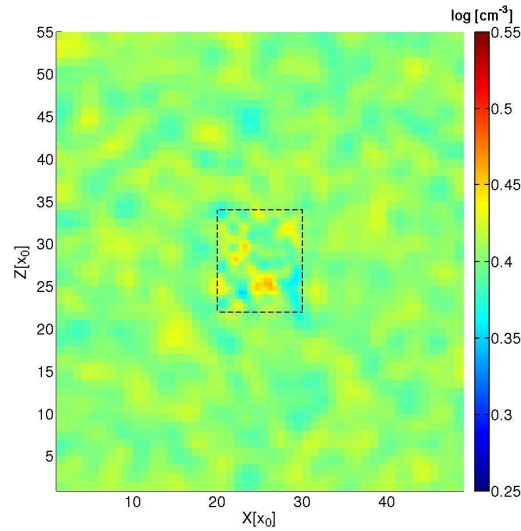

Figure 7: Electron density after 20 transit times in the **XZ** plane in log scale. The black box corresponds to the limits of the refined region.

approach in the case of a subsonic, sub-Alfvénic plasma. Results, which are not presented in this study, lead to similar conclusions.

4.2. Alfvén wave propagation

490   The test presented here aims to follow the propagation of a circularly polarized Alfvén wave through the refined level. It validates that the waves travel properly across refinement boundaries. The simulated plasma has the
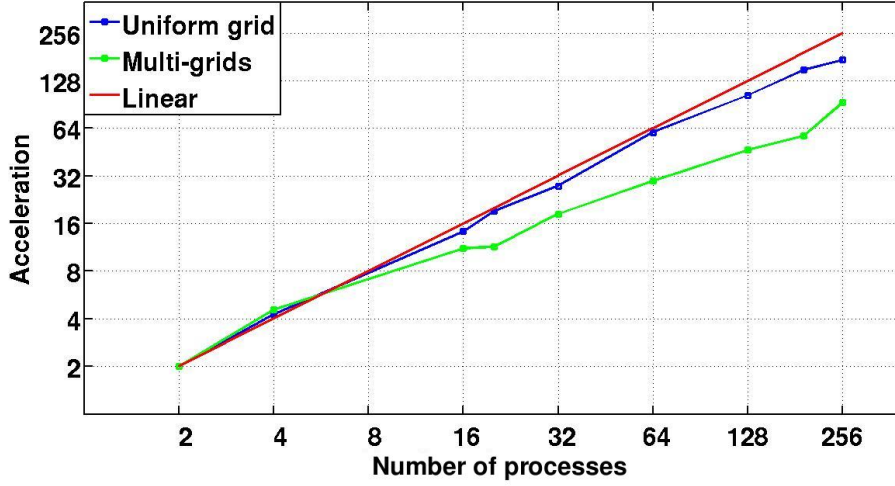
15

Figure 8: Performances of the parallelization : the linear scaling is displayed in red, the uniform grid simulation is shown in blue, the multi-grid simulation in green.

same properties as the previous section (Table 1): the plasma density is about 2.42 cm^{-3} and the flow velocity is 400 km/s. The electron beta parameter β is 10, and the plasma beta parameters for H$^+$ and He^{++} respectively are 5 and 20. The simulation domain contains $75 \times 40 \times 40$ cells, with $\Delta x = 2x_0$. The refined region contains $74 \times 32 \times 32$ cells, whose size is $dx = x_0$ in each direction, x_0 being the proton inertial length. Similar to the previous test, the chosen time step is $dt = 0.01\Omega_{\mathrm{H}^+}^{-1}$, satisfying the CFL condition and allowing a detailed description of ion gyrations. The x-component of the magnetic field is equal to B$_0$ = 3 nT. The magnetic field perpendicular to the plasma flow is defined as:

$$\mathbf{B}_\perp = \left(\begin{array}{c} \mathrm{B_y} \\ \mathrm{B_z} \end{array} \right) = \left(\begin{array}{c} 0.5\mathrm{B_0}\cos(kx - \omega t) \\ -0.5\mathrm{B_0}\sin(kx - \omega t) \end{array} \right) \tag{19}$$

$k = 2\pi n/L_x$, n being the wave number, equal to 2 in the present case, and L_x corresponding to the length of the simulation domain in the **X** direction. These magnetic field strengths are applied the whole simulation domain at initialization, and at the entry plane **YZ** during each time step of the simulation : boundary conditions are open in the **X** direction. Exactly the same perturbation is applied to particles' velocities at initialization. 60 macroparticles per cell are injected at initialization to represent the two ion species, along with about 11 simulated transits. We have compared and contrasted two simulation results, one simulation performed with a uniform grid, and one simulation utilizing the multi-grid approach, with the theoretical case. The goal is to check that the propagation of the waves are not affected by the introduction of a refinement level, and that the model efficienctly converges toward the analytical solution.

Figure 9 displays the average normalized y (at the top) and z (at the bottom) components of the magnetic field along the **X** axis without multi-grid (in black) and using our multi-grid method (in red). These averages have been obtained from all magnetic field values on all grid points of the simulation box. Boundaries of the refined grid are shown by dashed green lines and blue curve represents the analytical solution. First, the comparison of the magnetic fields profiles resulting from the two runs are very similar. No artifact, or numerical noise is created at the interfaces between the two grids. The Alfvén wave is not altered, neither by its entry into the refined region, nor by its passage from refined to coarse meshes. Both simulations provide a very small phase error with respect to the theoretical case, particularly at the end of the simulation box, due to the use of a mesh. However, simulations results are clearly very close to the analytical solution. Similar to the previous run test, energies, momentum and the number of macroparticles are properly conserved.
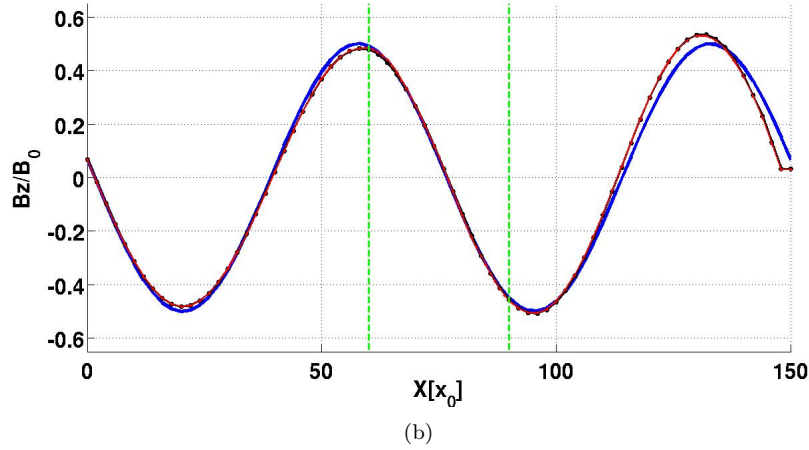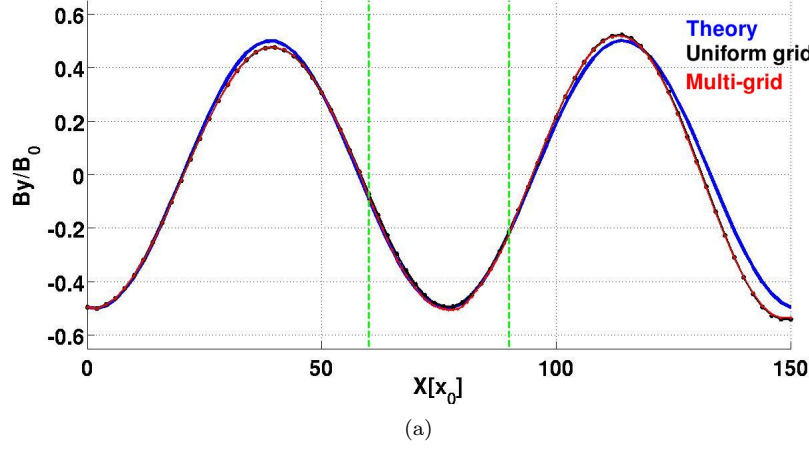
16

(a)

(b)

Figure 9: Top panel: Normalized simulated y-component of the magnetic field along **X** axis obtained in the uniform grid (respectively multi-grid) case in black (respectively in red). The blue curve represents the analytical solution. Bottom panel: Normalized simulated y-component of the magnetic field along **X** axis obtained in the uniform grid (respectively multi-grid) case in black (respectively in red). The blue curve represents the analytical solution.

4.3. Plasma-planet interaction : Ganymede case

4.3.1. Ingredients

The *Galileo* mission revealed that Ganymede is the only satellite in our solar system possessing an intrinsic magnetic field. Its magnetic field can be approximated by a dipole [26, 27]. Some induced magnetic fields have also been detected by the *Galileo* mission, likely due to the presence of a conducting liquid layer under the surface of the Galilean moon [27, 45]. The interaction between Ganymede's magnetic field and the jovian plasma leads therefore to the formation of a complex mini-magnetosphere surrounding the satellite. This magnetosphere is characterized by the creation of Alfvén wings, with no bow shock creation due to the jovian plasma flow being subsonic, and sub-Alfvénic.

The simulation is performed on a 3D simulation domain, using the Gphio coordinate system, centered on Ganymede, which is defined as follows : the corotating jovian plasma flows radially along the **X** direction, the **Y** axis points along the Ganymede-Jupiter direction and the **Z** axis completes the right-handed system, and corresponds to the spin axis of Ganymede.

The jovian plasma is composed of 87% of O^+, or S^{n+}, the rest being constituted of H^+, with an electron density equal to 4 cm^{-3} [16]. According to *Galileo* observations, the jovian plasma velocity is equal to 140 km/s [29]. The total ion temperature is 360 eV whereas the electron temperature is 100 eV [41]. The jovian magnetic field at Ganymede is equal to $B_0 \simeq 111$ nT (KK97 model of [25]), and is oriented along the $-\mathbf{Z}$ direction.

Ganymede's environment is composed of a magnetic field and an ionosphere. The magnetic field at the initialization includes both dipolar components and induced fields [27] leading to $\mathbf{B} = (-716.8, 82.5, -24.7)$ nT, at the surface at the equator. These values correspond to *Galileo*'s G1 flyby of Ganymede, which occured in 1996, and

17

have also been used by [21]. The ionosphere is composed of O$^+$ ions, with a surface density of 5000 cm^{-3} [28] and a scale height of 125 km [10] similar to the multi-fluid model of [42]. In order to have a correct characterization of this key region, it is important that the spatial resolution is comparable or smaller than the ionospheric plasma scale height. A spherical symmetry of the ionospheric density profile is assumed although may not be realistic. During the simulation, ionospheric particles are permanently injected in order to maintain the theoretical density profile.

4.3.2. Simulation parameters

Two simulations are performed : one with a uniform grid and another with a refined grid encompassing a coarse grid. The inertial length of jovian O$^+$ ions is c/ω_{pi}, which is about 487 km. The spatial resolution is uniform and equal to $\Delta x = 0.5 c/\omega_{pi}$ in the coarse grid of the multi-grid simulation and of the uniform grid simulation. In the refined region of the multi-grid simulation, spatial resolution is twice as small (about 120 km). The simulation domain is limited to -12 R$_G <$ **X** < 6 R$_G$, -18 R$_G <$ **Y** < 18 R$_G$ and -27 R$_G <$ **Z** < 27 R$_G$, R$_G$ being Ganymede's radius. The refined region is within -3 R$_G <$ **X** < 3.5 R$_G$, -3.5 R$_G <$ **Y** < 5.5 R$_G$ and -5.5 R$_G <$ **Z** < 5.5 R$_G$. 10 macroparticles per cell are used to describe the jovian plasma, whereas about 50 macroparticules per cell are injected near the surface to simulate the ionosphere. The simulation time is 160 $\Omega_{O^+}^{-1}$, $\Omega_{O^+}^{-1}$ designating the gyration time of the jovian oxygen ions, with a time step of $dt = 0.005\ \Omega_{O^+}^{-1}$. This time step choice respects the CFL condition in both grids and allows to describe H$^+$ (O$^+$) gyrations in 13 (200) points in the jovian plasma. 128 CPUs were used to perform the uniform grid simulation, whereas 256 CPUs were required to perform the simulation including the multi-grid approach.

4.3.3. Results

The restitution time of the uniform simulation is about 108 hours, compared to 152 hours for the multi-grid simulation. According to the simulation performance, a uniformly refined resolution performed on 256 CPUs would require at least 432 hours of calculations. Since the refined region describes 2% of the total domain simulation, the efficiency of the parallel multi-grid approach is not fully satisfying. This could be clearly improved by a better method for sharing the number of numerical particles between processes (see section 3.2).

Figure 10 displays the density of ionospheric O$^+$ ions, in the **XZ** plane centered on Ganymede, obtained with a uniform grid (left) and with the multi-grid approach (right). most of the planetary plasma remains within the
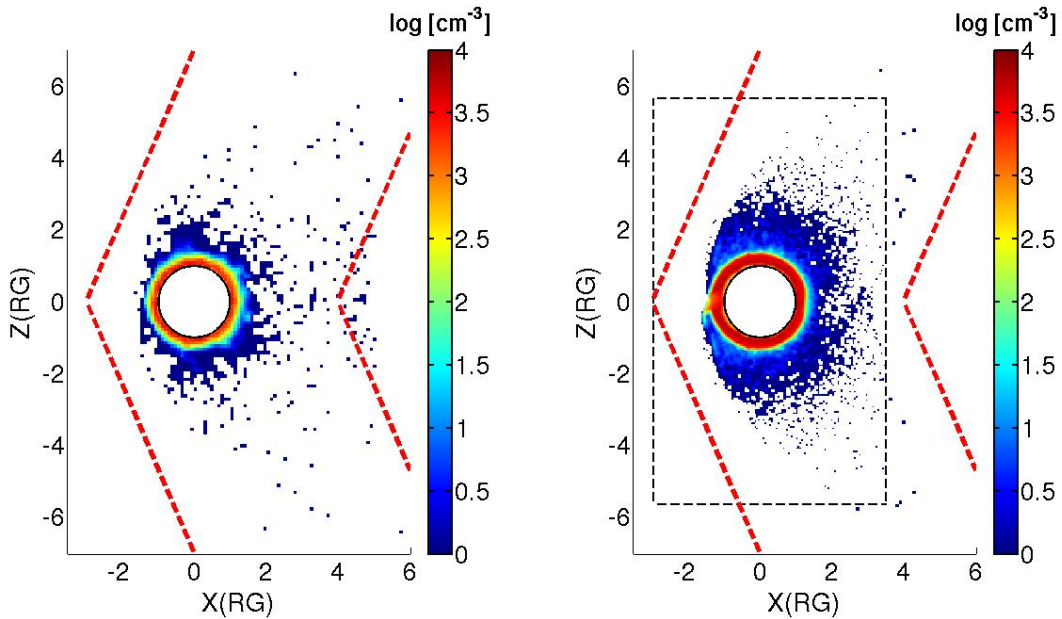

Figure 10: Color countours indicate density of planetary O$^+$, in log[cm^{-3}], in the **XZ** centered on Ganymede. Left panel : Uniform grid simulation. Right panel : multi-grid simulation. Dashed red lines correspond to the theoretical location of Alfvén wings and dashed black lines indicate the location of refined grid boundaries.

refined region. A similar behavior of the ionospheric plasma is observed in both simulations, with an escape through

the Alfvén wings, where magnetic field lines are open. The simulation including the multi-grid approach provides a much better description of the ionosphere. Indeed, since the number of injected macroparticles per cell is equal regardless of the grid spacing, the ionospheric plasma is represented by about eight times more macroparticles. In addition, since the scale height of the ionosphere is 125 km, a refined spatial resolution of 120 km is more suitable than a coarse grid with a resolution of 240 km. Note that the surface density is higher in the multi-grid case. Indeed, the modeled density is underestimated using a coarse resolution. When the spatial resolution is smaller, the modeled density profile is closer to the analytical one.

Figure 11 shows the x component of the magnetic field in the **XZ** plane centered on Ganymede, obtained with a uniform grid (left panel) and with the multi-grid approach (right panel), where the refined region is indicated by dashed black lines. Solid black lines represent simulated magnetic field lines, which are open in polar regions, and
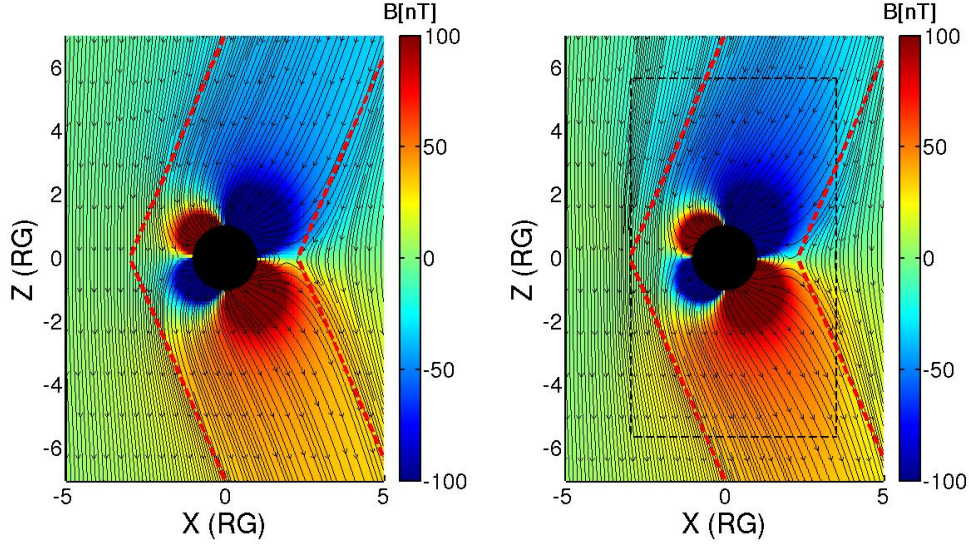

Figure 11: Color countours indicate the x component of the magnetic field, in $\log[\text{cm}^{-3}]$, in the **XZ** centered on Ganymede. Black lines represent the simulated magnetic field lines. Dashed black lines indicate the boundaries of the refined region whereas dashed red lines are the theoretical location of Alfvén wings (see the text for more details). Left panel : Uniform grid simulation. Right panel : multi-grid simulation.

closed elsewhere. Both simulations successfully creates the Alfvén wings, whose theoretical boundaries are indicated as dashed red lines. These limits are tilted from the ambient magnetic field with an angle $\theta = \text{atan}(V_f/V_A)$, V_f and V_A being the incident flow and Aflvén velocities respectively. Near Ganymede, the dipole components of the magnetic field are explicitly shown. The multi-grid simulation shows that the introduction of a refined grid globally does not affect the structure of Alfvén wings at the interfaces between the two grids. However, upstream the Alfvén wings, the two simulations show magnetic configurations that are slightly different. Indeed, since the ionosphere is better described with a refined spatial resolution, ionospheric currents, and therefore the magnetic structure are impacted by the spatial resolution.

Parallel currents **J** are represented in figure 12. Left (right) panel shows results from the uniform (multi-grid) simulation. Black lines are the modeled magnetic field lines. A quite important difference in current-magnitude is noticable: currents are stronger in the case of a refined grid spacing. This difference is generated by the way we compute the currents which are obtained from simulated magnetic field **B** via the Ampere's equation $\nabla \times \mathbf{B} = \mu_0 \mathbf{J}$. The derivatives used to compute the curl of the magnetic field are calculated using a finite difference scheme, including a division by a spatial step twice as small in the case of the multi-grid simulation. In regions where the variations of the magnetic field are not significant from one grid point to another, calculated currents are then twice as important with a refined resolution, as reported by [20]. Despite this difference, the global structure of parallel currents is very similar in both cases, with characteristic intense currents at the boundaries of Alfvén wings, and an asymmetry between the poles.
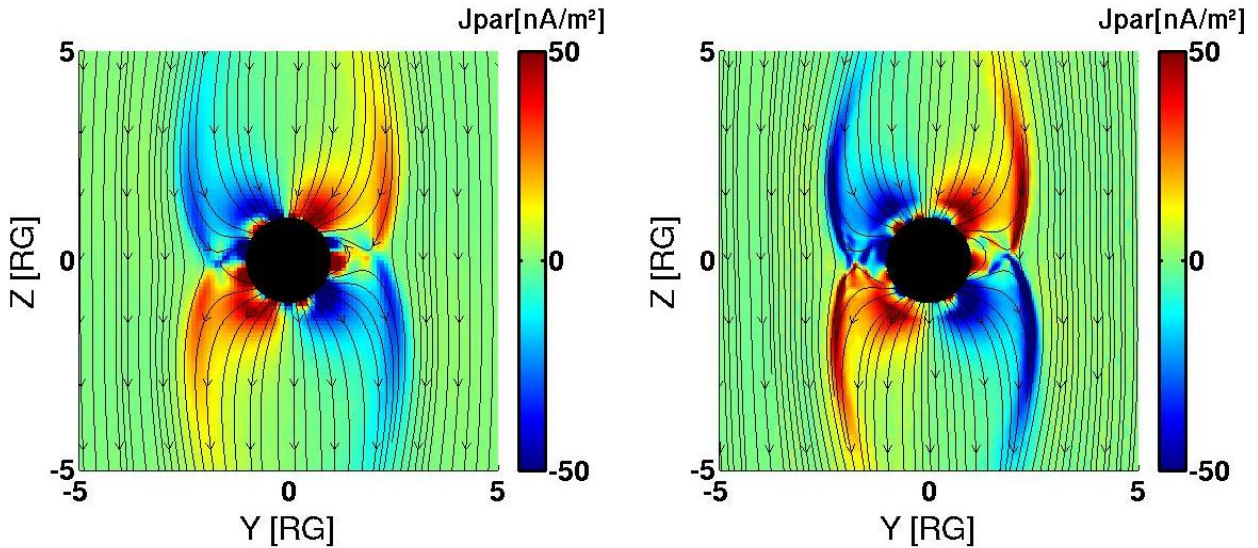
Figure 12: Color contours indicate parallel currents, in nA/m^2, in the **YZ** centered on Ganymede. Black lines represent the simulated magnetic field lines. Left panel : Uniform grid simulation. Right panel : multi-grid simulation.

5. Conclusion

This study presents a new multi-grid approach to refine the spatial resolution for a given region of interest in a magnetospheric hybrid model dedicated to the study of planetary object-plasma interactions. Simulations are performed on a 3D uniform cartesian grid that is refined for a defined region of interest. In this region, the grid spacing is divided by two in each direction with respect to the coarse grid, creating cells eight times smaller in volume. Structure and size of this refined mesh are not time dependent, and, only one level of refinement is created. The strength of our method is that the particle volume is constant and the trajectory of particles is not perturbed passing from one refinement level to another. Coarse particles are replaced by a set of eight refined particles before their entry into the refined region, similar to the 1D and 2D PIC codes by [19] and [1]. Daughter particles acquire the velocity of the parent particles, and their mass center is shifted from the parent particle position by a quarter of a cell width in each direction. This splitting algorithm allows the conservation of grid moments, such as current or charge density, and velocity distribution functions. Numerical particles are then authorized to exit from the refined region into the coarse grid, keeping the spatial shape function constant of particles. To compute grid moments in coarse regions, weights affecting the grid points are adapted depending on the particle's volume. This approach also avoids particle merging, which would induce the loss of informations. At the interfaces, overlap regions are not considered. The novelty of our approach is that the size of the interaction region of each grid point of the interface is taken into account to compute moments, increasing accuracy and avoiding additional computations in overlap cells.

This development has been validated in its 3D parallel hybrid case, having been applied to several space plasma applications. In particular, the flow of the solar wind and the propagation of an Aflvén wave were simulated successfully. A planetary environment interacting with an external source of plasma has also been used to illustrate the performances of this approach (Ganymede interacting with the corotating jovian magnetospheric plasma). The refinement model has allowed a better description of the ionosphere of the satellite and of the Alfvén wings formed by this interaction. This Multi-Grid approach will be used to model other planetary environments such as Mars or Mercury which have been modeled with the uniform grid version (or the sequential version) of this hybrid model [37, 44].

References

₆₂₅ [1] Beck, A., Innocenti, M., Lapenta, G., Markidis, S., Aug. 2014. Multi-level multi-domain algorithm implementation for two-dimensional multiscale particle in cell simulations. Journal of Computational Physics 271, 430–443.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999113008152

[2] Birdsall, C. K., Fuss, D., Apr. 1969. Clouds-in-clouds, clouds-in-cells physics for many-body plasma simulation.
₆₃₀ Journal of Computational Physics 3 (4), 494–511.
URL http://linkinghub.elsevier.com/retrieve/pii/0021999169900588

[3] Birdsall, C. K., Langdon, A. B., 1985. Plasma physics via computer simulation. McGraw-Hill, New York.

[4] Brackbill, J. U., Sep. 1993. An Adaptive Grid with Directional Control. Journal of Computational Physics 108 (1), 38–50.
₆₃₅ URL http://linkinghub.elsevier.com/retrieve/pii/S0021999183711617

[5] Brackbill, J. U., Saltzman, J. S., Jun. 1982. Adaptive zoning for singular problems in two dimensions. Journal of Computational Physics 46 (3), 342–368.
URL http://linkinghub.elsevier.com/retrieve/pii/0021999182900201

[6] Chacón, L., Delzanno, G., Finn, J., Jan. 2011. Robust, multidimensional mesh-motion based on
₆₄₀ Monge–Kantorovich equidistribution. Journal of Computational Physics 230 (1), 87–103.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999110005073

[7] Colella, P., Norgaard, P. C., Feb. 2010. Controlling self-force errors at refinement boundaries for AMR-PIC. Journal of Computational Physics 229 (4), 947–957.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999109003696

₆₄₅ [8] Daldorff, L. K., Tóth, G., Gombosi, T. I., Lapenta, G., Amaya, J., Markidis, S., Brackbill, J. U., Jul. 2014. Two-way coupling of a global Hall magnetohydrodynamics model with a local implicit particle-in-cell model. Journal of Computational Physics 268, 236–254.
URL http://linkinghub.elsevier.com/retrieve/pii/S002199911400182X

[9] Delzanno, G., Chacón, L., Finn, J., Chung, Y., Lapenta, G., Dec. 2008. An optimal robust equidistribution
₆₅₀ method for two-dimensional grid adaptation based on Monge–Kantorovich optimization. Journal of Computational Physics 227 (23), 9841–9864.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999108004105

[10] Eviatar, A., Vasyliunas, V. M., Gurnett, D. A., Mar. 2001. The ionosphere of Ganymede. Planetary and Space Science 49 (3-4), 327–336.
₆₅₅ URL http://linkinghub.elsevier.com/retrieve/pii/S0032063300001549

[11] Forum, M. P. I., 2009. MPI : A Message Passing Interface Standard ; Version 2.2. High-Performance Computing Center.
URL http://books.google.nl/books?id=pnVicgAACAAJ

[12] Friedel, H., Grauer, R., Marliani, C., Jun. 1997. Adaptive Mesh Refinement for Singular Current Sheets in
₆₆₀ Incompressible Magnetohydrodynamic Flows. Journal of Computational Physics 134 (1), 190–198.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999197956837

[13] Fujimoto, K., Sep. 2011. A new electromagnetic particle-in-cell model with adaptive mesh refinement for high-performance parallel computation. Journal of Computational Physics 230 (23), 8508–8526.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999111004608

₆₆₅ [14] Fujimoto, K., Machida, S., May 2006. Electromagnetic full particle code with adaptive mesh refinement technique: Application to the current sheet evolution. Journal of Computational Physics 214 (2), 550–566.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999105004614

21

[15] Fujimoto, K., Machida, S., Jan. 2006. Full particle simulation of the plasma sheet using adaptive mesh refinement technique. Advances in Space Research 37 (7), 1348–1353.
URL http://linkinghub.elsevier.com/retrieve/pii/S0273117705003753

[16] Gurnett, D. A., Kurth, W. S., Roux, A., Bolton, S. J., Kennel, C. F., Dec. 1996. Evidence for a magnetosphere at Ganymede from plasma-wave observations by the Galileo spacecraft. Nature 384 (6609), 535–537.
URL http://www.nature.com/doifinder/10.1038/384535a0

[17] Hockney, R. W., Eastwood, J. W., 1988. Computer simulation using particles. Taylor & Francis, New York.

[18] Innocenti, M., Beck, A., Ponweiser, T., Markidis, S., Lapenta, G., Apr. 2015. Introduction of temporal sub-stepping in the Multi-Level Multi-Domain semi-implicit Particle-In-Cell code Parsek2d-MLMD. Computer Physics Communications 189, 47–59.
URL http://linkinghub.elsevier.com/retrieve/pii/S001046551400410X

[19] Innocenti, M., Lapenta, G., Markidis, S., Beck, A., Vapirev, A., Apr. 2013. A Multi Level Multi Domain Method for Particle In Cell plasma simulations. Journal of Computational Physics 238, 115–140.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999112007590

[20] Ip, W.-H., Kopp, A., 2002. Resistive MHD simulations of Ganymede's magnetosphere 2. Birkeland currents and particle energetics. Journal of Geophysical Research 107 (A12).
URL http://doi.wiley.com/10.1029/2001JA005072

[21] Jia, X., Walker, R. J., Kivelson, M. G., Khurana, K. K., Linker, J. A., Jun. 2008. Three-dimensional MHD simulations of Ganymede's magnetosphere. Journal of Geophysical Research 113 (A6).
URL http://doi.wiley.com/10.1029/2007JA012748

[22] Kallio, A., Janhunen, P., 2001. Atmospheric effects of proton precipitation in the Martian atmosphere and its connection to the Mars-solar wind interaction. Journal of Geophysical Research 106 (A4), 5617.
URL http://doi.wiley.com/10.1029/2000JA000239

[23] Kallio, E., Chaufray, J.-Y., Modolo, R., Snowden, D., Winglee, R., Dec. 2011. Modeling of Venus, Mars, and Titan. Space Science Reviews 162 (1-4), 267–307.
URL http://link.springer.com/10.1007/s11214-011-9814-8

[24] Kallio, E., Janhunen, P., 2003. Modelling the solar wind interaction with Mercury by a quasi-neutral hybrid model. Annales Geophysicae 21 (11), 2133–2145.
URL http://www.ann-geophys.net/21/2133/2003/

[25] Khurana, K. K., 1997. Euler potential models of Jupiter's magnetospheric field. Journal of Geophysical Research 102 (A6), 11295.
URL http://doi.wiley.com/10.1029/97JA00563

[26] Kivelson, M. G., Khurana, K. K., Russell, C. T., Walker, R. J., Warnecke, J., Coroniti, F. V., Polanskey, C., Southwood, D. J., Schubert, G., Dec. 1996. Discovery of Ganymede's magnetic field by the Galileo spacecraft. Nature 384 (6609), 537–541.
URL http://www.nature.com/doifinder/10.1038/384537a0

[27] Kivelson, M. G., Khurana, K. K., Volwerk, M., Jun. 2002. The Permanent and Inductive Magnetic Moments of Ganymede. Icarus 157 (2), 507–522.
URL http://linkinghub.elsevier.com/retrieve/pii/S001910350296834X

[28] Kliore, A. J., Anabtawi, A., Nagy, A. F., 2001. The ionospheres of Ganymede and Callisto from Galileo radio occultations. In: BAAS. Vol. 33. p. 1084.

[29] Krupp, N., Lagg, A., Livi, S., Wilken, B., Woch, J., Roelof, E. C., Williams, D. J., 2001. Global flows of energetic ions in Jupiter's equatorial plane: First-order approximation. Journal of Geophysical Research 106 (A11), 26017.
URL http://doi.wiley.com/10.1029/2000JA900138

[30] Lapenta, G., Sep. 2002. Particle Rezoning for Multidimensional Kinetic Particle-In-Cell Simulations. Journal of Computational Physics 181 (1), 317–337.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999102971263

[31] Lapenta, G., Jun. 2011. DEMOCRITUS : An adaptive particle in cell (PIC) code for object-plasma interactions. Journal of Computational Physics 230 (12), 4679–4695.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999111001306

[32] Lapenta, G., Feb. 2012. Particle simulations of space weather. Journal of Computational Physics 231 (3), 795–821.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999111001860

[33] Ledvina, S. A., Ma, Y.-J., Kallio, E., Aug. 2008. Modeling and Simulating Flowing Plasmas and Related Phenomena. Space Science Reviews 139 (1-4), 143–189.
URL http://link.springer.com/10.1007/s11214-008-9384-6

[34] Lipatov, A. S., 2002. The hybrid multiscale simulation technology : an introduction with application to astrophysical and laboratory plasmas. Springer, Berlin; New York.
URL http://dx.doi.org/10.1007/978-3-662-05012-5

[35] Ma, Y.-J., Altwegg, K., Breus, T., Combi, M. R., Cravens, T. E., Kallio, E., Ledvina, S. A., Luhmann, J. G., Miller, S., Nagy, A. F., Ridley, A. J., Strobel, D. F., Aug. 2008. Plasma Flow and Related Phenomena in Planetary Aeronomy. Space Science Reviews 139 (1-4), 311–353.
URL http://link.springer.com/10.1007/s11214-008-9389-1

[36] Matthews, A. P., May 1994. Current Advance Method and Cyclic Leapfrog for 2d Multispecies Hybrid Plasma Simulations. Journal of Computational Physics 112 (1), 102–116.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999184710849

[37] Modolo, R., Chanteur, G. M., Dubinin, E., Jan. 2012. Dynamic Martian magnetosphere: Transient twist induced by a rotation of the IMF: DYNAMIC OF THE MARTIAN MAGNETOSPHERE. Geophysical Research Letters 39 (1), n/a–n/a.
URL http://doi.wiley.com/10.1029/2011GL049895

[38] Moritaka, T., Nunami, M., Usui, H., 2010. Development of full particle-in-cell simulation code with adaptive mesh refinement technique. J. Plasma Fusion 9, 586–591.
URL http://www.jspf.or.jp/JPFRS/index_vol9.html

[39] Müller, J., Simon, S., Motschmann, U., Schüle, J., Glassmeier, K.-H., Pringle, G. J., Apr. 2011. A.I.K.E.F. : Adaptive hybrid model for space plasma simulations. Computer Physics Communications 182 (4), 946–966.
URL http://linkinghub.elsevier.com/retrieve/pii/S0010465510005266

[40] Müller, J., Simon, S., Wang, Y.-C., Motschmann, U., Heyner, D., Schüle, J., Ip, W.-H., Kleindienst, G., Pringle, G. J., Mar. 2012. Origin of Mercury's double magnetopause: 3d hybrid simulation study with A.I.K.E.F. Icarus 218 (1), 666–687.
URL http://linkinghub.elsevier.com/retrieve/pii/S0019103512000164

[41] Neubauer, F. M., 1998. The sub-Alfvénic interaction of the Galilean satellites with the Jovian magnetosphere. Journal of Geophysical Research 104 (E2), 3863.
URL http://doi.wiley.com/10.1029/1998JE900031

[42] Paty, C. S., Winglee, R., May 2006. The role of ion cyclotron motion at Ganymede: Magnetic field morphology and magnetospheric dynamics. Geophysical Research Letters 33 (10).
URL http://doi.wiley.com/10.1029/2005GL025273

[43] Peng, I. B., Markidis, S., Vaivads, A., Vencels, J., Amaya, J., Divin, A., Laure, E., Lapenta, G., 2015. The Formation of a Magnetosphere with Implicit Particle-in-Cell Simulations. Procedia Computer Science 51, 1178–1187.
URL http://linkinghub.elsevier.com/retrieve/pii/S1877050915010960

[44] Richer, E., Modolo, R., Chanteur, G. M., Hess, S., Leblanc, F., Oct. 2012. A global hybrid model for Mercury's interaction with the solar wind: Case study of the dipole representation. Journal of Geophysical Research 117 (A10).
URL http://doi.wiley.com/10.1029/2012JA017898

[45] Saur, J., Duling, S., Roth, L., Jia, X., Strobel, D. F., Feldman, P. D., Christensen, U. R., Retherford, K. D., McGrath, M. A., Musacchio, F., Wennmacher, A., Neubauer, F. M., Simon, S., Hartkorn, O., Mar. 2015. The search for a subsurface ocean in Ganymede with Hubble Space Telescope observations of its auroral ovals. Journal of Geophysical Research: Space Physics 120 (3), 1715–1737.
URL http://doi.wiley.com/10.1002/2014JA020778

[46] Wahlund, J. E., Modolo, R., Bertucci, C., Coates, A., 2013. Titan's Magnetospheric and Plasma Environment. In: Titan: Surface, Atmosphere and Magnetosphere. Cambridge Planetary Science (14). Cambridge University Press.
URL https://hal.archives-ouvertes.fr/hal-00731543

[47] Winslow, A., Apr. 1981. Adaptive-mesh zoning by the equipotential method. Tech. Rep. UCID-19062, 6227449.
URL http://www.osti.gov/servlets/purl/6227449-mUQTXS/