Velocity-space adaptive mesh refinement for Vlasiator: Strategy and performance metrics

M. Antoine[1], U. Ganse[1], M. Alho[1], L. Kotipalo[1], K. Papadakis[1], and M. Palmroth[1,2].

[1]Department of Physics, University of Helsinki, Helsinki, Finland. [2]Finnish Meteorological Institute, Helsinki, Finland.

(maxence.antoine@helsinki.fi)

Abstract

To study near-Earth plasma environments without assuming a given velocity distribution function, kinetic models are required. Vlasiator [1, 2] is a 6D hybrid-Vlasov solver which treats ions kinetically using a semi-Lagrangian method [3], while electrons are modeled as a charge-neutralizing fluid, enabling global simulations of the interaction between the solar wind and the Earth's magnetosphere. Solving the evolution of a six-dimensional distribution function over such a large domain with the required accuracy would require excessive computational resources. Vlasiator addresses this challenge by using a sparse velocity space grid to reduce the computational cost by up to 98%, together with spatial mesh refinement [1]. Initially, the mesh refinement was static and based on predefined regions of interest (e.g. the magnetopause and magnetotail); however, it has recently evolved towards an adaptive mesh refinement (AMR) strategy that targets regions with strong dimensionless spatial gradients and magnetic reconnection sites [4]. One of the next major developments for Vlasiator is the implementation of a velocity-space AMR. This enhancement aims to further reduce computational costs while accurately capturing filamentation of the velocity distribution function, which plays a key role in plasma instabilities [5]. We present the possible refinement strategy and discuss its associated technical challenges.

The Vlasiator model

Vlasiator is a **global hybrid-Vlasov** model of the coupled solar wind–magnetosphere system in six-dimensional phase space. Ions are treated kinetically, while electrons are approximated as a massless charge-neutralizing fluid. Vlasiator solves the ion Vlasov equation coupled to Maxwell's equations and the generalized Ohm's law:

$$\frac{\partial f_i}{\partial t} + \mathbf{v}\cdot\frac{\partial f_i}{\partial \mathbf{x}} + \frac{e}{m_i}(\mathbf{E}+\mathbf{v}\times\mathbf{B})\cdot\frac{\partial f_i}{\partial \mathbf{v}} = 0 \qquad \frac{\partial \mathbf{B}}{\partial t} = -\boldsymbol{\nabla}\times\mathbf{E}$$

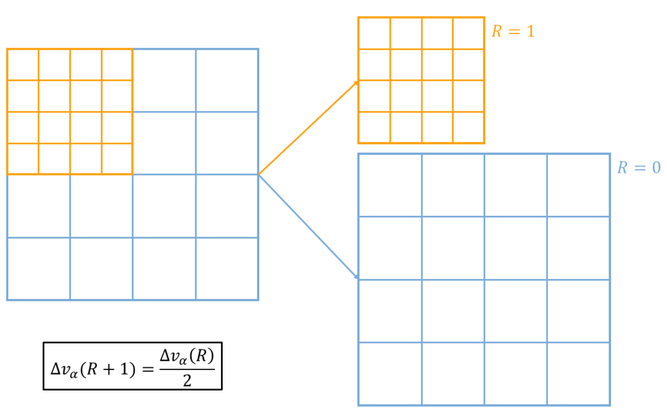
$$\mathbf{E} = -\mathbf{V}_i\times\mathbf{B} + \frac{\mathbf{J}}{\sigma} + \frac{\mathbf{J}\times\mathbf{B}}{en} - \frac{\nabla\cdot\mathbf{P}_e}{en}$$

where $\mathbf{P}_e = p_e\mathbf{I}$ and $p_e = (k_B T_e n^{1-\gamma})\ n^\gamma$, $\mathbf{J} = (\boldsymbol{\nabla}\times\mathbf{B})/\mu_0$, $n = \int f\, d^3v$ and $\mathbf{V_i} = \int f\,\mathbf{v_i}\, d^3v / \int f\, d^3v$.

This system of equations is solved using a semi-Lagrangian method with Strang splitting $(\mathcal{X},\mathcal{Y},\mathcal{Z},\mathcal{R})$. Here, $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ represent the spatial advections, while $\mathcal{R}$ corresponds to the velocity-space rotation, which is computed using the 3D slice scheme of [3].

★ Velocity mesh refinement method ★

A further step toward reducing the numerical cost of Vlasiator is the introduction of **velocity-space refinement**. One approach is to treat velocity space as a set of coupled grid that exchange information at their interfaces. This representation can be interpreted as a **patch-based** adaptive mesh refinement (AMR) approach [5], for which an **octree-based** structure is considered as a natural implementation. This type of velocity-space refinement has the additional advantage of not impacting the parallelisation of the code, since it is typically handled in velocity space independently of the spatial domain decomposition.

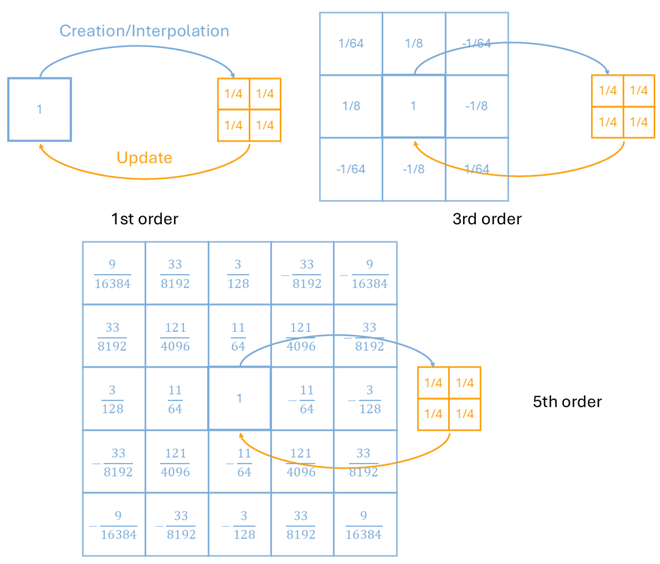


A key aspect of velocity-space AMR is the **conservation of physical moments**, in particular the density. This requires defining consistent communication operators between refinement levels that preserve the desired interpolation accuracy. The simplest restriction operator from fine to coarse grids is a cell-average projection. In this context, only **odd-order interpolation** schemes can be used [7], with the order chosen according to the desired accuracy and computational cost.

Higher-order schemes (e.g. third or fifth order) require the introduction of intermediate **ghost cells** between refinement levels (R) and (R+2) to ensure consistency of the reconstruction.

Update exchanges are performed after each advection step, whereas **creation** exchanges occur only when and where a new grid is generated. Performing the refinement procedure **every five time steps** is expected to keep the additional computational cost low.

Since the representation is based on blocks, velocity-space integrals (e.g. n and V_i) can be evaluated as simple sums over blocks.



The next step is to define **refinement criteria**. Such a representation can be described using wavelets, and a commonly used criterion to minimize the error L^q of the s derivative for a d dimension system is [8] :

$$d_R = f_R - f_{\text{interpolated by } R-1} > \epsilon\, 2^{(-s+\frac{d}{q})R}$$

Two criteria will be investigated for the creation of level $R+1$:

$$d_R > \epsilon = 10^{-15}$$

This first criterion enforces the sparse velocity threshold by minimising the error L^∞ of the distribution function.

$$d_R > \epsilon\, 2^{-R}$$

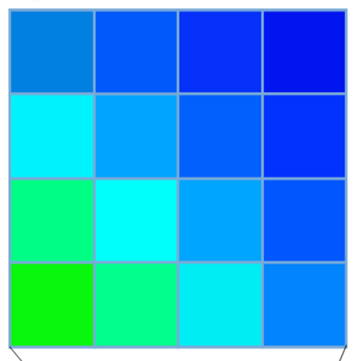
The second criterion ensures accurate tracking of the filamentation [5] of the distribution function, which is related to energy transfer.

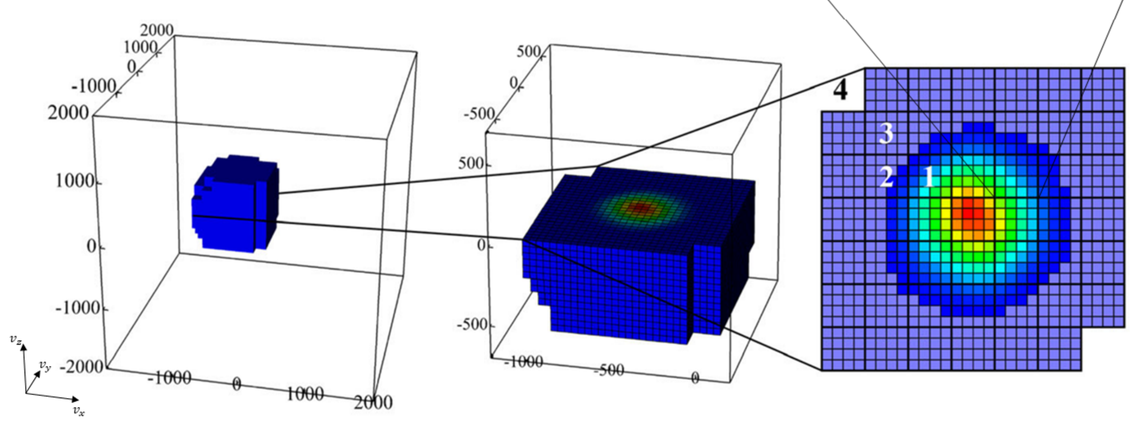
Open to other suggestions!

Numerical techniques in Vlasiator

Solving the six-dimensional Vlasov equation using a semi-Lagrangian method leads to a very high computational cost, which motivates the use of additional numerical techniques. The distribution function is represented on a phase-space grid, where cell-averaged values are stored. In velocity space, the domain is decomposed into blocks of $4\times4\times4$ cells to ensure efficient parallelisation. However, near-Earth plasma systems exhibit strong variations in the distribution function, requiring a large velocity domain while maintaining high resolution only where the distribution is non-negligible. A uniform discretisation would therefore result in a large number of unused cells.

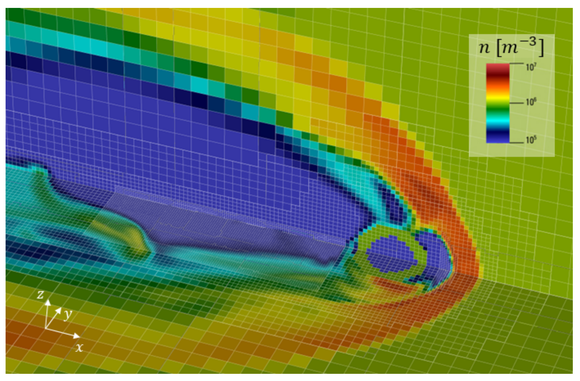
To address this issue, Vlasiator employs a **sparse velocity-space** representation based on a threshold $f > 10^{-15}\,\mathrm{s^3/m^6}$, ensuring density conservation within 0.1% (with simulation termination if exceeded). This criterion allows velocity-space blocks to be extended with an additional buffer layer only where needed. This approach has been shown to reduce the computational cost by up to 98% [1].





[6]

Another approach to reduce the computational cost is the use of a **spatially refined mesh**. Since the physical processes of interest are governed by ion kinetic scales that vary significantly across regions (e.g. $r_L = mv/(qB) \simeq 10{-}1500\,\mathrm{km}$), the mesh can be refined a priori using empirical relationships for the bow shock and magnetopause locations. Refinement is applied only to the distribution function, while the electric and magnetic fields remain defined on a uniform grid to enable the Yee scheme. This mesh hierarchy reduces the computational cost by more than 90%.



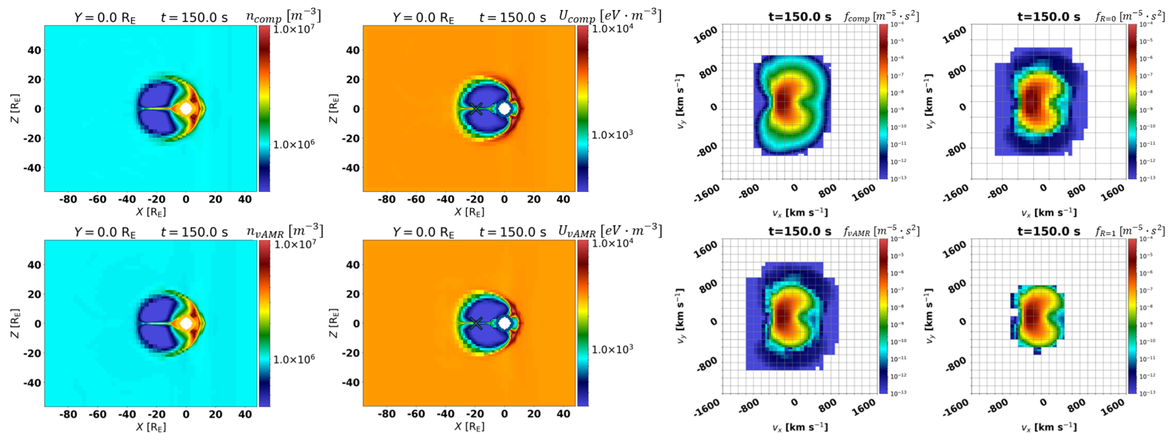
[1]

$$\alpha_1 = \max\begin{cases} \frac{|\Delta\rho|}{\hat{\rho}} \\ \frac{|\Delta U|}{\hat{U}} \\ \frac{(\Delta\mathbf{p})^2}{2\rho\hat{U}} \\ \frac{(\Delta\mathbf{B})^2}{2\mu_0\hat{U}} \\ \frac{|\Delta\mathbf{B}|}{\hat{B}} \end{cases} \qquad \alpha_2 = \frac{\mu_0|\mathbf{J}|}{B_\perp+\epsilon}\Delta x$$

Although this mesh is well suited to the problem, some regions may remain under-resolved during the simulation. To address this issue, a **dynamic mesh refinement method** has recently been introduced. The first refinement criterion (α_1) is based on temporal variations of particle density, total energy, plasma energy, magnetic-field energy, and magnetic-flux energy. A second criterion, (α_2), is used to identify magnetic reconnection regions. Cells with ($\alpha < 0.3$) are coarsened, whereas cells with ($\alpha > 0.6$) are refined [4].

Preliminary results

Preliminary test cases have already demonstrated a **24% reduction in computational time**, while maintaining good agreement in density, energy density, and velocity distribution functions between the original and velocity-space AMR implementations.



References

[1] U. Ganse *et al.*, Phys. Plasmas. **30**, 042902 (2023).
[2] M. Palmroth *et al.*, Living Rev. Comput. Astrophys. **11**, 3 (2025).
[3] M. Zerroukat and T. Allen, Q.J.R. Meteorol. Soc. **138**, 1640 (2012).
[4] L. Kotipalo *et al.*, Geosci. Model Dev. **17**, 6401 (2024).
[5] M. Antoine *et al.*, Phys. Plasmas. **32**, 043905 (2025).
[6] Y. Pfau-Kempf *et al.*, ANGEO **34**, 943 (2016).
[7] E. Deriaz *et al.*, ESAIM **70**, 107 (2021).
[8] E. Deriaz *et al.*, Multiscale Model. Simul. **16**, 583 (2018).