Geosci. Model Dev., 15, 7903–7912, 2022
https://doi.org/10.5194/gmd-15-7903-2022

Geoscientific Model Development
Open Access

Development and technical paper

Spatial filtering in a 6D hybrid-Vlasov scheme to alleviate adaptive mesh refinement artifacts: a case study with Vlasiator (versions 5.0, 5.1, and 5.2.1)

Konstantinos Papadakis[1], Yann Pfau-Kempf[1], Urs Ganse[1], Markus Battarbee[1], Markku Alho[1], Maxime Grandin[1], Maxime Dubart[1], Lucile Turc[1], Hongyang Zhou[1], Konstantinos Horaites[1], Ivan Zaitsev[1], Giulia Cozzani[1], Maarja Bussov[1], Evgeny Gordeev[1], Fasil Tesema[1], Harriet George[1], Jonas Suni[1], Vertti Tarvus[1], and Minna Palmroth[1,2]

[1]Department of Physics, University of Helsinki, Helsinki, Finland
[2]Finnish Meteorological Institute, Space and Earth Observation Centre, Helsinki, Finland

Correspondence: Konstantinos Papadakis (konstantinos.papadakis@helsinki.fi)

Received: 31 May 2022 – Discussion started: 5 July 2022
Revised: 9 September 2022 – Accepted: 25 September 2022 – Published: 27 October 2022

Abstract. Numerical simulation models that are used to investigate the near-Earth space plasma environment require sophisticated methods and algorithms as well as high computational power. Vlasiator 5.0 is a hybrid-Vlasov plasma simulation code that is able to perform 6D (3D in ordinary space and 3D in velocity space) simulations using adaptive mesh refinement (AMR). In this work, we describe a side effect of using AMR in Vlasiator 5.0: the heterologous grid approach creates discontinuities due to the different grid resolution levels. These discontinuities cause spurious oscillations in the electromagnetic fields that alter the global results. We present and test a spatial filtering operator for alleviating this artifact without significantly increasing the computational overhead. We demonstrate the operator's use case in large 6D AMR simulations and evaluate its performance with different implementations.

1 Introduction

Investigation of the near-Earth space plasma environment benefits from numerical simulation efforts, which can model plasma effects on global scales compared with physical observations that are inherently local in space and time (Hesse et al., 2014). Vlasiator (Palmroth et al., 2018) is a hybrid-Vlasov plasma simulation code that models collisionless plasmas by solving the Vlasov–Maxwell system of equations for ion particle distribution functions on a 6D Cartesian mesh, representing three spatial and three velocity dimensions. The Vlasov equation (Eq. 1) is a form of the Boltzmann equation that neglects the collisional term to only account for electromagnetic interactions:

$$\frac{\partial f}{\partial t} + \boldsymbol{v} \cdot \frac{\partial f}{\partial \boldsymbol{r}} + \frac{q}{m} (\boldsymbol{E} + \boldsymbol{v} \times \boldsymbol{B}) \cdot \frac{\partial f}{\partial \boldsymbol{v}} = 0. \tag{1}$$

Here, $f(\boldsymbol{r}, \boldsymbol{v}, t)$ represents the phase space density of a species of mass m and charge q, where $\boldsymbol{r}$ is position, $\boldsymbol{v}$ is velocity, and t is time. $\boldsymbol{E}$ and $\boldsymbol{B}$ stand for the electric and magnetic fields, respectively. Vlasiator couples the Vlasov equation for ions with the electromagnetic fields through Maxwell's equations under the Darwin approximation which eliminates the displacement current in the Ampère equation to get rid of electromagnetic wave modes and enable longer time steps. This leads to the Ampère and Faraday laws taking the following form, while maintaining a divergence-free magnetic field:

$$\boldsymbol{j} = \frac{\nabla \times \boldsymbol{B}}{\mu_0}, \tag{2}$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial}{\partial t} \boldsymbol{B}, \tag{3}$$

$$\nabla \cdot \boldsymbol{B} = 0. \tag{4}$$

The system is closed using Ohm's law in the form

$$\boldsymbol{E} = -\boldsymbol{V}_{\mathrm{i}} \times \boldsymbol{B} + \frac{1}{\rho_q} \boldsymbol{j} \times \boldsymbol{B} - \frac{1}{\rho_q} \nabla \boldsymbol{P}_{\mathrm{e}}, \tag{5}$$

where $\boldsymbol{j}$ is the current density, μ_0 is the permeability of free space, $\boldsymbol{V}_\mathrm{i}$ is the ion bulk velocity, ρ_q is the charge density, and $\boldsymbol{P}_\mathrm{e}$ is the electron pressure tensor.

In its implementation, Vlasiator stores a 3D velocity grid in each spatial grid cell, which requires significant memory for large simulations. This leads to simulation results that are free from sampling noise, unlike simulations that employ stochastic particle representation methods such as particle-in-cell (PIC) codes (Nishikawa et al., 2021). While ion kinetics are resolved, Vlasiator models the electron population as a charge-neutralizing background fluid, as typical in hybrid-kinetic approaches, to keep computational cost down. Vlasiator employs a sparse velocity space representation (von Alfthan et al., 2014), where the parts of the velocity distribution function below a specific threshold are neither stored nor propagated. The electromagnetic fields are coupled to the Vlasov solver by taking velocity moments of the distribution function (density, flow velocity, pressure) and feeding them into Maxwell's equations (Eqs. 2–5) which are then solved through a constrained transport upwind method described in Londrillo and del Zanna (2004).

Vlasiator's core is made up of two separate solvers: the field solver and the Vlasov solver. The Vlasov solver solves the Vlasov equation in two steps using Strang splitting (Palmroth et al., 2018), namely spatial translation and acceleration in velocity space, using a semi-Lagrangian scheme based on the SLICE-3D method described in Zerroukat and Allen (2012). Vlasiator has been employed in a range of studies regarding Earth's foreshock formation (Turc et al., 2019; Kempf et al., 2015), ionospheric precipitation (Grandin et al., 2019), and magnetotail reconnection (Palmroth et al., 2017), for example.

Most scientific studies of Vlasiator have been limited to 5D (two spatial and three velocity dimensions) due to the large computational requirements. In Vlasiator 5.0, adaptive mesh refinement (AMR) has been applied to enable the simulation of 6D configurations. With the use of AMR, the Vlasov solver uses the highest spatial resolution available only in regions of high scientific interest. Regions of less interest are solved at a lower spatial resolution. As Vlasiator needs to store a velocity distribution function for every simulation cell, which is numerically described by a 3D grid, the memory requirements for 6D simulations are extreme. The AMR functionality previously added in Vlasiator 5.0 manages to alleviate the computational burden by reducing the effective cell count in a 6D simulation. Thus, the use of AMR is necessary for Vlasiator to venture into exploring the near-Earth space plasma in 6D.

The use of AMR can lead to a big performance gain for a simulation; however, it can also introduce spurious artifacts that can alter the simulation results. As an example, WarpX (Vay et al., 2018), a particle-in-cell code, uses mesh refinement to accelerate simulations, but it has to deal with spurious self-forces experienced by particles and short-wavelength electromagnetic waves reflecting at mesh refinement boundaries. However, Vlasiator, being a Vlasov code, does not suffer from the same artifacts that WarpX suffers from, as it does not use a particle representation to describe plasma, and its field solver operates on a uniform mesh.

In this work, we demonstrate the use of low-pass filtering in Vlasiator to help eliminate artifacts caused in numerical simulations using AMR. The structure of this publication is such that we first provide an insight to Vlasiator's grid topology as well as the heterologous grid-coupling mechanism to exchange the required variables between its two core solvers (Sect. 2). Moreover, we describe the staircase effect, which is an artifact caused by the grid-coupling process in AMR runs. In Sect. 3, we give a brief summary of low-pass filtering and propose our method for alleviating the staircase effect in Vlasiator. In Sect. 4, we demonstrate simulation results and evaluate the performance of two different implementations of the filtering operator. Further discussion and conclusions are presented in the final sections of this work.

2 Heterologous grid structure in Vlasiator

Vlasiator's field solver uses the FsGrid library (Palmroth and the Vlasiator team, 2020) to store field quantities and plasma moments as well as to propagate the electromagnetic fields. FsGrid is a message passing interface (MPI) aware library that uses ghost cell communication to make data available to neighboring processes in the simulation domain. The electromagnetic field update is not an expensive operation in Vlasiator and thus requires no AMR, even in a spatially 3D high-resolution setup. Hence, the spatial resolution for FsGrid is kept uniform in the simulation domain. FsGrid uses a block decomposition scheme to evenly distribute the simulation cells over the MPI tasks, and this is kept constant for the rest of the simulation when the number of MPI tasks remains the same. On the other hand, velocity distribution function translation and acceleration are the most expensive operations in Vlasiator. Quantities of the Vlasov solver are discretized using DCCRG (Honkonen et al., 2013; Honkonen, 2022), a parallel grid library that supports cell-based AMR. Hence, Vlasiator employs a heterologous AMR scheme, where the field solver and Vlasov solver operate on separate grids and with different spatial resolutions depending on the local DCCRG AMR level, with the field solver operating at the highest resolution allowed by DCCRG.

Currently, for Vlasiator's 6D simulations, the refinement levels are manually chosen. Regions of interest closer to the Earth (like the bow shock and the magnetotail reconnection site) operate at the maximum spatial resolution, whereas lower spatial resolution is used for regions of less interest (such as the inflow and outflow boundaries). In Fig. 1, a typical configuration of the AMR Vlasov grid is illustrated. Figure 1a shows three slices of the simulation domain at $X = 0$, $Y = 0$, and $Z = 0$, and Fig. 1b shows only the $Z = 0$ plane in which four levels of refinements are depicted from the coarsest (white) to the finest (black). Dynamically adjusting the AMR levels based on physical criteria during runtime is under development and will be the subject of a future study.

2.1 Grid coupling

DCCRG operates on a base refinement level, and each successive refinement level has twice the resolution of the previous one. At the highest refinement region there is a one-to-one match between the field solver and DCCRG's cells. However, the electromagnetic fields and plasma propagation are inherently dependent upon each other; thus, a coupling process takes place during every simulation time step. The coupling scheme is illustrated in Fig. 2. The Vlasov solver at the end of every time step feeds moments into the field solver grid. In regions where the one-to-one match is not fulfilled, one set of moments is communicated to all FsGrid cells which occupy the same volume as the underlying DCCRG cell. The field solver then propagates the fields and communicates those back to the Vlasov solver before the next time step begins. In mismatching regions, the field solver grid is fed uniform input in all the cells that are children of a lower-resolution DCCRG cell, and the parent DCCRG cell is later fed an averaged value of all the higher-resolution corresponding FsGrid cells. The association between the two grids is calculated during the initialization and after every load-balancing operation where the Cartesian spatial decomposition scheme over different MPI tasks changes for the DCCRG grid. When no AMR is used, the two solvers operate on the same spatial resolution; thus, there is a one-to-one grid match, making the coupling scheme trivial.

2.2 Staircase effect

In 6D AMR simulations, the one-to-one grid matching is restricted to only the highest refinement regions where both solvers operate at the highest spatial resolution. In less-refined regions, the Vlasov solver cells span multiple field solver cells and the grids mismatch. If the trivial coupling scheme described in the previous subsection is maintained, the field solver is subject to discontinuous plasma moment input at the Vlasov grid cell interfaces, which can be seen in Fig. 3a and b, as an effect that we dub the "staircase effect". The discontinuities caused by the staircase effect lead to the development of unphysical oscillations in the field quantities on the field solver grid. The oscillations can be observed in the profiles demonstrated in Fig. 3d and f. Those oscillations can act as a source of spurious wave excitation and propagate artifacts in the whole simulation domain, as visible in Fig. 3c and e, where artifacts have propagated downstream from the bow shock of the global magnetospheric simulation, causing significant distortion of the physics in the nightside magnetosheath and lobes.

3 Method

3.1 Low-pass filtering

Low-pass filtering is a well-known tool from digital signal processing theory that effectively attenuates unwanted high-frequency parts of the spectrum. The boxcar filter is the simplest finite impulse response low-pass filter, and it smooths out a signal by substituting a value with the average of itself and its two closest neighbors. Boxcar filters are usually cascaded with other boxcars in an attempt to reduce the high side lobes in their frequency response (Roscoe and Blair, 2016). Techniques like low-pass filtering are not limited to the time domain; they can be applied to other dimensions like ordinary space and find wide application in fields like image processing (Cook, 1986) and numerical modeling (Vay and Godfrey, 2014).

3.2 Spatial filtering

In this work, we present two implementations of the spatial filtering operator used in Vlasiator to smooth out the discontinuities in AMR simulations that are illustrated in Fig. 3a and b. First, in Vlasiator 5.1, the filtering operator is realized by a 3D 27-point (3 points per spatial dimension) boxcar kernel with equally assigned weights. The kernel operates in position space ($\boldsymbol{r}$) and is passed over the field solver grid cells immediately after the coupling process finishes transferring Vlasov moments to the field solver grid. The filtering operator is only applied when there is a grid mismatch between the two solvers, so it is not used in the highest refinement level where the two solvers operate at the same spatial resolution. The number of times that the operator is applied is not constant but linearly depends on the refinement ratio between the two grids. Each filtering operator pass attenuates the high-frequency signals on the field solver grid and smooths out the discontinuities shown in Fig. 3a and b. Larger refinement ratios between the two grids give rise to spatially larger discontinuities, requiring more filtering passes in order to smooth the discontinuity. In practice, treating the finest to coarsest levels with 0, 2, 4, and 8 passes of the boxcar kernel, for example, has proved to alleviate the discontinuities illustrated in Fig. 3a and b.

At the end of each pass of the filtering scheme, a ghost cell communication update is required for the respective FsGrid structure prior to continuing on to the next pass. This manifests as a performance penalty, as the ghost communication is a global process involving all MPI tasks in the simulation. From the associative properties of the convolution operation, where B is a boxcar kernel, g is a function, and $\circledast$ is the convolution operator, we define the triangle kernel T, where

$$g \circledast T = g \circledast (B \circledast B) = (g \circledast B) \circledast B. \tag{6}$$

Two boxcar passes are the equivalent of a single triangular kernel pass; therefore, in Vlasiator 5.2.1, we update our fil-

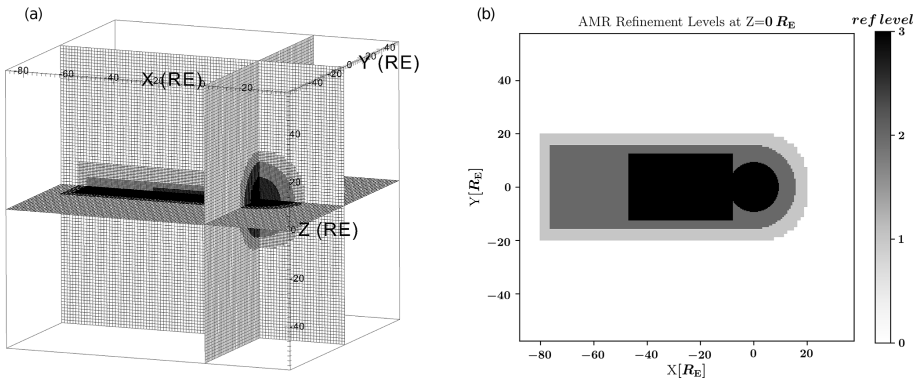

Figure 1. (a) Perspective 3D view of the AMR Vlasov grid. Here, the Earth is located at the axes' origin. **(b)** Equatorial distribution of the AMR refinement levels on the Vlasov grid in a 3D AMR simulation. Refinement level n corresponds to 2^n times the base resolution. Darker regions closer to the Earth ($n = 3$) are solved with the highest spatial resolution used in the simulation.

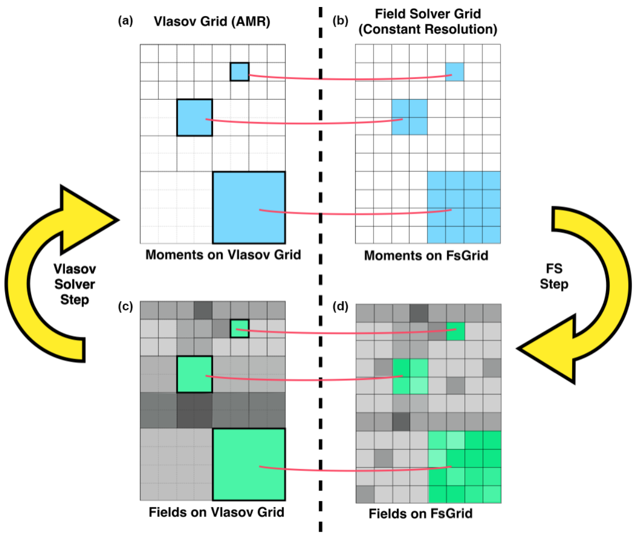

Figure 2. Schematic of the grid-coupling procedure. Moments evaluated on the Vlasov grid are copied over to the FsGrid. **(a, b)** When there is a grid mismatch, one DCCRG cell copies over its values to many FsGrid cells. Then the fields are propagated and finally fed back to the Vlasov grid. **(c, d)** In areas where the grids mismatch, multiple FsGrid cells are averaged to get the value for the corresponding DCCRG cell.

Algorithm 1 Filtering operator pseudo-code

```
if method=="boxcar" then
    kernelWidth ← 1
    kernelWeights ← [1, 1, 1]
    sum ← sum(kernelWeights)
else if method=="triangle" then
    kernelWidth ← 2
    kernelWeights ← [1, 2, 3, 2, 1]
    sum ← sum(kernelWeights)
end if
swapGrid ← momentsGrid
for pass = 0, 1, ...maxPasses do
    for i = 0, 1, ...momentsGrid.GridSize do
        refLevel ← momentsGrid.getRefLevel(i)
        if pass>=momentsGrid.numPasses(refLevel) then
            continue
        end if
        swapGrid[i] ← 0.0
        for j = −kernelWidth...kernelWidth do
            ii ← i + j
            swapGrid[i]+ = momentsGrid[ii] ∗
            kernelWeights[kernelWidth + j]/sum
        end for
    end for
    momentsGrid ← swapGrid
    momentsGrid.updateGhostCells()
end for
```

tering operator to a 3D 125-point triangle kernel. We choose the 3D triangle kernel as the convolution of a boxcar kernel with itself to match the frequency characteristics of the boxcar operator in half the number of passes, as shown in Fig. 4. The triangle kernel is used in the same fashion as the boxcar kernel; however, as half the number of passes per refinement level are required to achieve the same amount of smoothing this time, the triangular kernel operator performs half the amount of ghost cell communications. A graphical representation of the filtering kernels is given in Fig. 5. In Algorithm 1, we demonstrate the two implementations of the spatial filtering algorithm with a 1D pseudo-code example. The modified coupling scheme is demonstrated in Fig. 6, where it is now supplemented with the filtering mechanism, in contrast to Fig. 2.

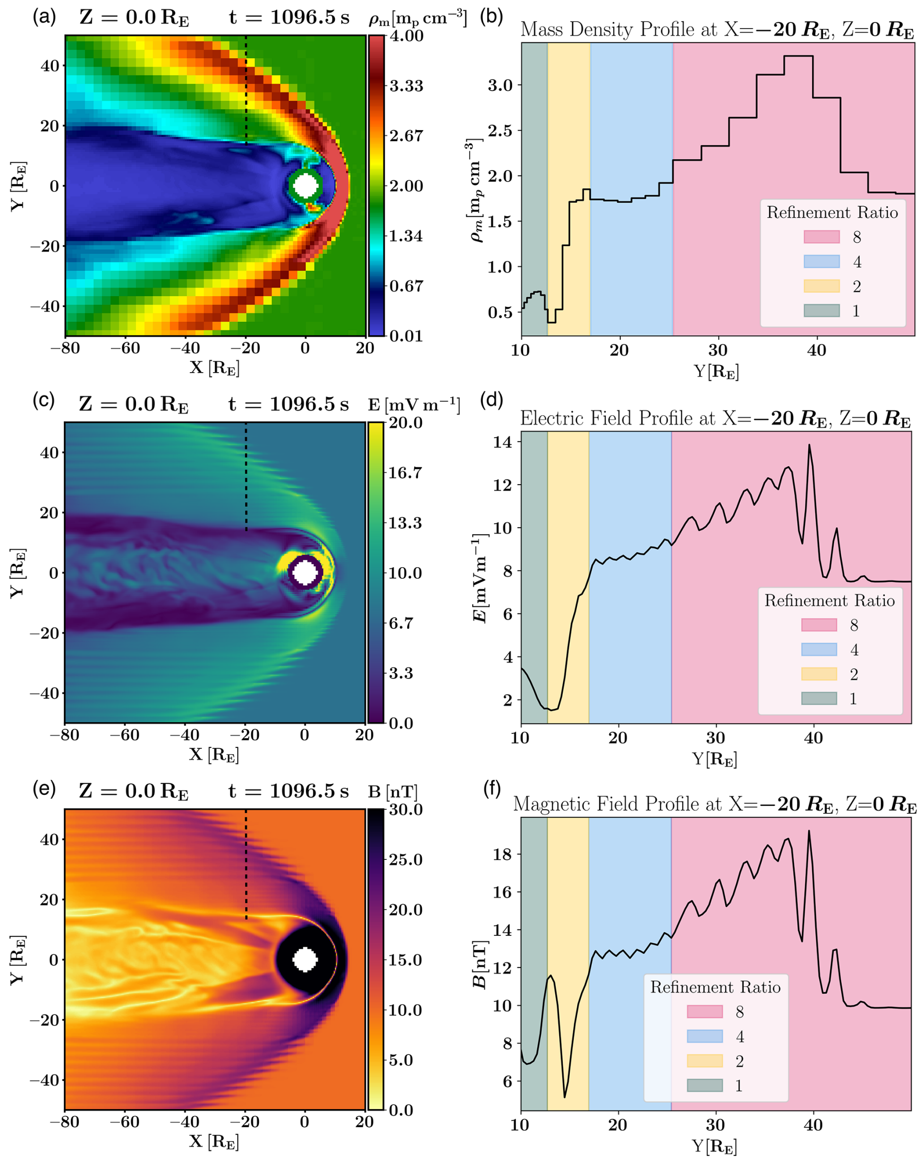

Figure 3. Visualization of a 6D Vlasiator simulation with heterologous grids. **(a)** Equatorial color map of mass density (on the AMR Vlasov grid). **(b)** Mass density along the profile shown as a dashed line in panel **(a)**. In panels **(a)** and **(b)**, m_p stands for proton mass. **(c)** Electric field magnitude color map (on the uniform FsGrid). **(d)** Electric field magnitude along the profile shown as a dashed line in panel **(c)**. **(e)** Magnetic field magnitude color map (on the uniform FsGrid). **(f)** Magnetic field magnitude along the profile shown as a dashed line in panel **(e)**. The colored regions in panels **(b)**, **(d)**, and **(f)** correspond to the resolution ratio between the field solver grid and the Vlasov grid. Unphysical oscillations in the field quantities, triggered by discontinuities in the moments, can be observed in the green and red regions.

4 Results

To demonstrate the effect of the spatial filtering employed in Vlasiator 5.1 to alleviate the AMR discontinuities, we create a simple configuration with a density step in the middle of a 3D simulation box. The step, shown in Fig. 7, poses a discontinuity in the otherwise smooth mass density. Similar steps are created during AMR simulations in regions where there is no one-to-one match between the Vlasov and field solver grid cells. We apply the boxcar filtering operator an increasing number of times to evaluate its performance, and the results are illustrated in Fig. 7. When the filtering operator is cascaded with itself, its effect becomes more significant in damping high-frequency signals, as shown in Fig. 4. In Fig. 7, we only show the effect of the boxcar operator because the results are identical to using the triangular kernel operator, as shown by the frequency response of the two kernels depicted in Fig. 4.

Furthermore, we demonstrate the results of the boxcar filtering operator in a large magnetospheric production-scale

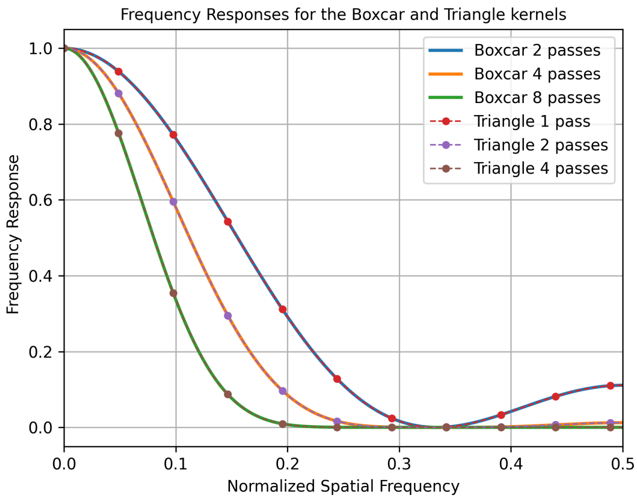

Figure 4. Solid lines show the frequency response of the boxcar filter operator for different numbers of passes as used in Vlasiator. Dashed lines show the frequency response of the triangular filter operator for different numbers of passes as used in Vlasiator.

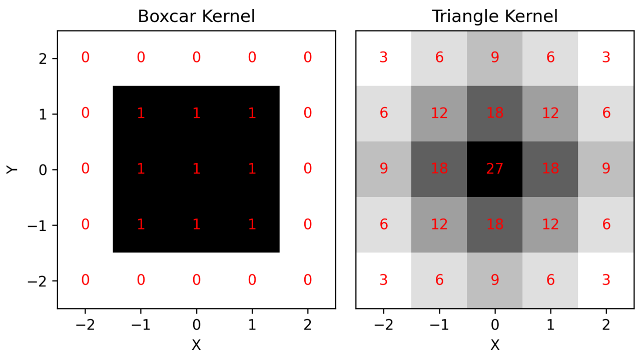

Figure 5. Graphical representation of the boxcar and the triangle kernels. The 2D slices are taken from the middle of the kernel's third dimension. For better clarity, the boxcar kernel is padded with zeros and the kernel weights, shown in red font, are not normalized.

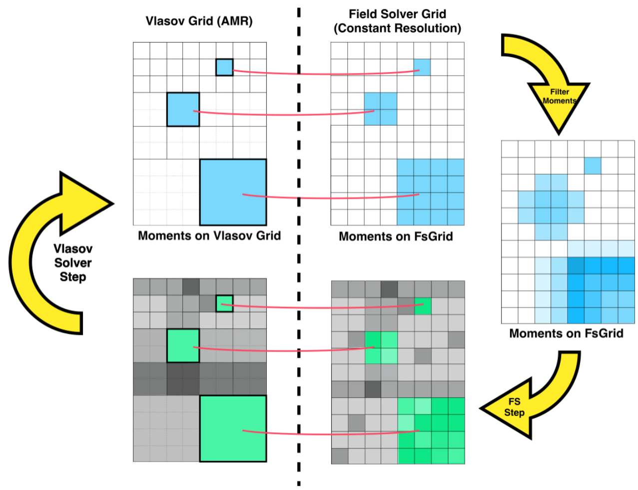

Figure 6. Schematic of Vlasiator's grid-coupling scheme supplemented with the extra filtering step to alleviate the staircase effect in 6D AMR runs.

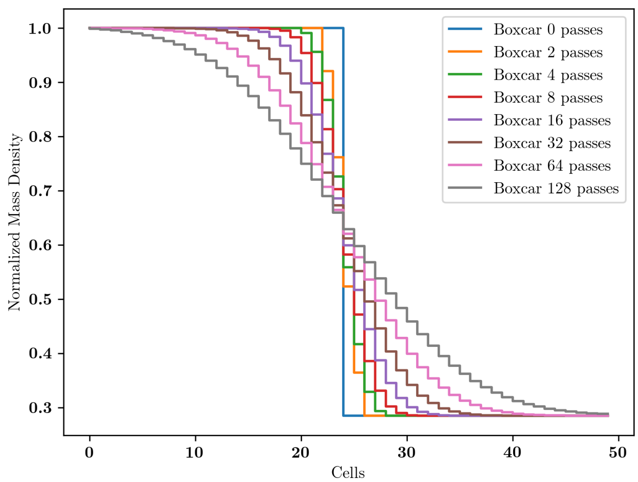

Figure 7. Smoothing of a step in mass density along a line in the simulation domain intersecting a discontinuity. The unfiltered normalized mass density line profile is shown in blue, and the other colors correspond to an increasing number of boxcar passes.

run using four AMR levels. Simulation quantities on the heterogeneous grid structure are illustrated in Fig. 8. In Fig. 8a, a color map of the mass density on the AMR Vlasov grid is depicted. The discontinuities at the AMR level interfaces are visible. In Fig. 8b, a color map of the mass density on the uniform field solver grid is illustrated after the coupling process has taken place. During the coupling process, the filtering operator is used, and the AMR levels are treated, from finest to coarsest, with 0, 2, 4, and 8 passes, respectively. Figure 8c and e show the respective electric field and magnetic field magnitudes simulated with filtered moments on FsGrid. The profiles demonstrated in Fig. 8d and f are sampled along the dashed paths in Fig. 8c and e, respectively.

The simulation used to evaluate the filtering operator models a 3D space around Earth in the Geocentric Solar Magnetic (GSM) coordinate system with no dipole tilt. The modeled space extends from $-560\,000$ to $240\,000$ km in the X dimension, from $-368\,000$ to $368\,000$ km in both the Y and Z dimensions, and is represented by a $100 \times 92 \times 92$ Cartesian mesh with $\Delta r = 8000$ km at the lowest refinement level and with $\Delta r = 1000$ km at the highest refinement level. Each spatial cell contains a velocity space with a 3D grid with $\Delta v = 40\,\mathrm{km\,s^{-1}}$. The solar wind is modeled with proton density $n = 7\,\mathrm{cm^{-3}}$, temperature $T = 0.5 \times 10^6$ K and solar wind speed $V_x = -1000\,\mathrm{km\,s^{-1}}$. The interplanetary magnetic field points mostly southward with $\boldsymbol{B} = [-0.5, 0, -20]$ nT.

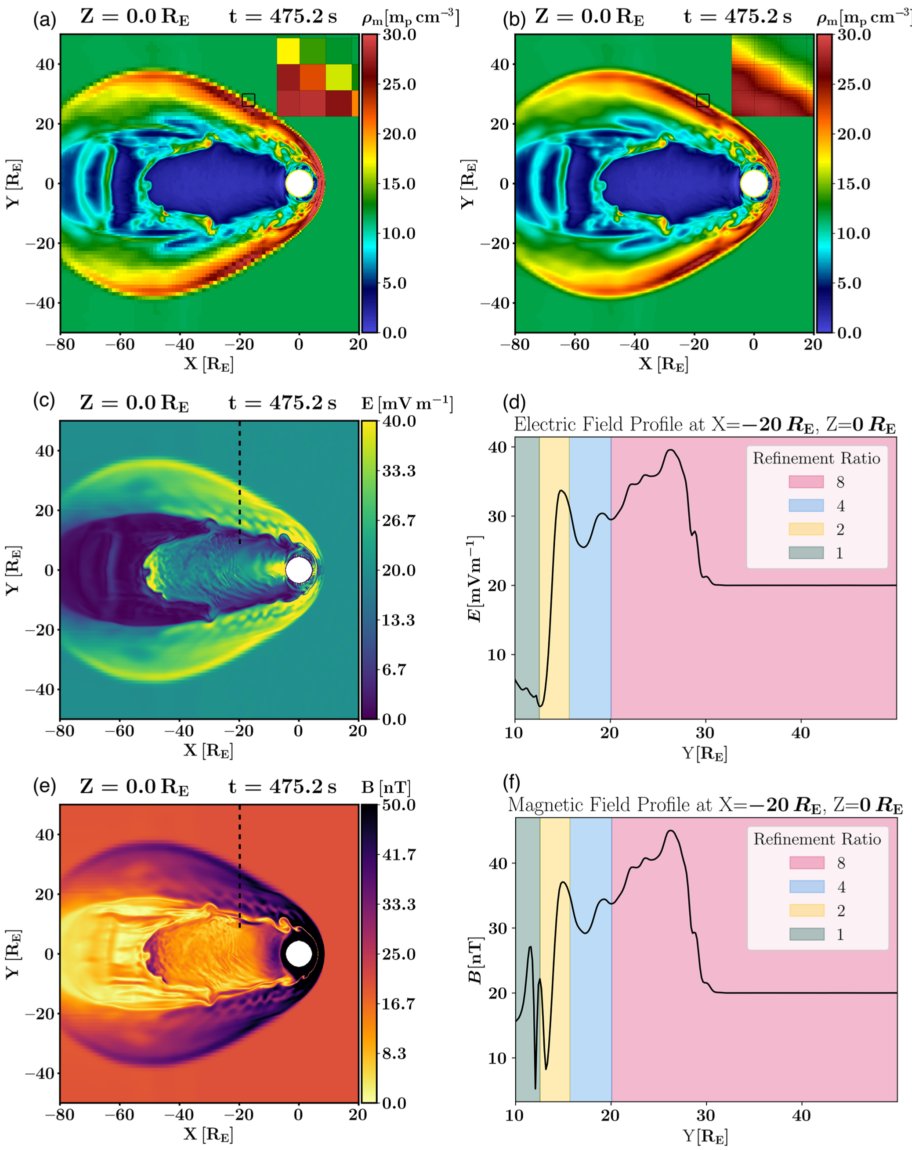

Figure 8. Vlasiator magnetospheric simulation results. **(a)** Mass density color map (on the AMR Vlasov grid) of a 6D AMR simulation. **(b)** Mass density color map (on the uniform FsGrid) of a 6D AMR simulation with the filtering operator in use. In panels **(a)** and **(b)**, m_p stands for proton mass. There are four refinement levels, and they are treated with 0, 2, 4, and 8 filtering passes from finest to coarsest, respectively. The insets in panels **(a)** and **(b)** illustrate the mass density on the uniform field solver grid. The AMR Vlasov mesh is denoted by the black lines. The insets are taken from the regions highlighted by black squares in panels **(a)** and **(b)**. The step discontinuities visible in the inset in panel **(a)** are spatially co-located with the DCCRG grid cells. **(c)** Electric field magnitude color map (on the uniform FsGrid) with the filtering operator in use. **(d)** Electric field profile sampled along the dashed line in panel **(c)**. **(e)** Magnetic field magnitude color map (on the uniform FsGrid) with the filtering operator in use. **(f)** Magnetic field profile sampled along the dashed line in panel **(e)**.

4.1 Performance overhead

Care has to be taken to keep the performance overhead of the filtering operator small. The boxcar operator is applied at every simulation time step and makes use of a duplicate FsGrid structure of the Vlasov moments because the filtering cannot happen in place. The boxcar operator uses OpenMP threading to parallelize the filtering over the local domain of each MPI task. In Table 1, we report the extra memory needed for the filtering operator and the time spent filtering the Vlasov moments during the grid-coupling process for the production 6D AMR run shown in Fig. 8.

From Table 1, we see that the boxcar filtering operator used in Vlasiator 5.1 amounts to 6 % of the total simulation time for a production run like the one in Fig. 8. To evaluate the performance improvement of the five-stencil triangle kernel implementation, used in Vlasiator 5.2.1, we set up smaller tests and compare the two methods. The triangle

Table 1. Profiling statistics for the boxcar filtering operator.

Memory used for the simulation [GB]	11 000
Memory used for the boxcar filtering [GB]	15.68 (0.14 %)
Time spent on simulation [s]	50 410
Time spent on boxcar filtering [s]	2818 (6 %)

Table 2. Comparison of the two filtering kernels' performance.

Memory used for the simulation [GB]	250
Memory used for the boxcar operator [GB]	0.035 (0.014 %)
Memory used for the triangle operator [GB]	0.035 (0.014 %)
Time spent on simulation [s]	1260
Time spent by the boxcar operator [s]	147 (11.8 %)
Time spent by the triangle operator [s]	87 (7.0 %)

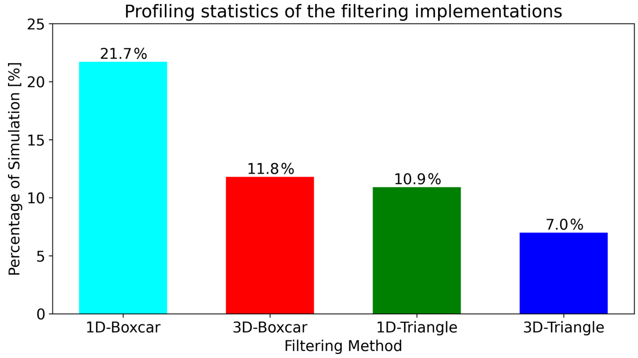

Figure 9. Profiling statistics for the different implementations of the filtering operators in Vlasiator. The results are derived from a custom simulation setup designed to test filtering performance.

kernel operates in the same way but only needs half the numbers of passes to achieve proper smoothing, so we treat the finest to coarsest levels with 0, 1, 2, and 4 passes. We report the results in Table 2.

While both approaches require the same amount of memory, the time spent by the triangular operator amounts to 59 % of that spent by the boxcar operator.

Both the boxcar and the triangular filtering operators are 3D spatial convolutions that can be expressed as three 1D convolutions (Birchfield, 2017). This is known as kernel separability and can improve the performance of the two operators significantly. Formally, the use of a separable kernel instead of a 3D one, would reduce the complexity from $\mathcal{O}(N_x \times N_y \times N_z \times d^3)$ (where N_x, N_y, and N_z are the dimensions of the simulation mesh, and d is the dimension size of the 3D kernel) to $\mathcal{O}(3 \times N_x \times N_y \times N_z \times d)$. We modify our implementations of the 3D boxcar and triangle operators to take advantage of the kernel separability property and test their performance using the same configuration used to produce the results in Table 2. We demonstrate the performance statistics of all four methods in Fig. 9.

While the separable operators should theoretically lead to a significant performance gain, the 1D operators are slower than their 3D counterparts in practice. This is due to the fact that an interim ghost-update communication needs to take place after every pass done by the operators, and, as the 1D implementations require more mesh traversals per pass, they end up spending more time on updating their ghost cell values. Additionally, we note that kernel separability does not hold if the stencil is altered – for example, when part of the kernel covers the highest refinement level.

4.2 Moment conservation

The filtering operator, as described above, is not conservative at the interface of adjacent refinements levels. However, this is not a cumulative effect because moments on FsGrid, used to propagate the electromagnetic fields, are provided to the field solver by the Vlasov solver at each time step and are not copied back from the FsGrid. Furthermore, the moment conservation is also violated due to numerical precision round-off errors during the filtering passes. The amount by which the moments are not conserved depends on the number of filtering passes and on the number of cells in a given simulation. We measure the relative difference in mass density caused by the filtering operator in the simulations presented in this work and find it to always remain below 10^{-5}, which we deem acceptable given the non-cumulative nature of the filtering operation.

5 Discussion

The first 6D simulations with Vlasiator 5.0 would not have been possible without the use of AMR. However, the heterologous grid structure and the grid-coupling mechanism in Vlasiator create artifacts in simulations using AMR that alter the global physics. In this work, we report on a new development employing spatial filtering in the hybrid-Vlasov code Vlasiator (versions 5.1 and 5.2.1) in order to alleviate the staircase effect created due to the heterologous AMR scheme used in 6D simulations. Based on the results of this study, the use of a linearly increasing number of passes per refinement level minimizes the aliasing effect at the Vlasov–field solver grid interfaces. Treating the finest to coarsest levels with 0, 2, 4, and 8 passes of the boxcar filter, for example, has proved to alleviate the staircase effect satisfactorily, as can be seen in Fig. 8a and b. As a result, the electric and magnetic field magnitude profiles in Fig. 8d and f show none of the oscillatory behavior caused by the staircase effect, in contrast to those demonstrated in Fig. 3d and f. As the filtering operator is applied at every simulation time step, it has to be well optimized so that it does not increase the computational overhead significantly. From Table 1, we see that the filtering in a production simulation amounts to 6 % of the computational time, which we deem significant. To improve the filtering performance, we develop a 3D five-point stencil triangle kernel in Vlasiator 5.2.1, which is equivalent with respect to alleviating the staircase effect but only needs half the number of passes per refinement level. We test the triangle kernel on a smaller simulation and report on its performance in Table 2. The triangle kernel provides a 41 % performance improvement over the boxcar approach; thus, we estimate that, in a similar simulation to the one shown in Fig. 8 it would amount to 3.5 % of the total simulation time, which we deem acceptable. The improved performance is a combination of both halving the ghost cell updates needed for the triangular kernel operator and reducing the operations needed because the wider kernel operator requires half the number of passes compared with the boxcar operator. Furthermore, we evaluate the performance gain acquired by exploiting the filter separability property of the two filtering operators and conclude that the separable kernels in fact perform worse in the context of Vlasiator than their 3D counterparts, as they are hindered by the higher number of ghost cell updates that they require. Another approach to improve the performance of the filtering methods would be to use an even wider kernel to completely eliminate the ghost cell updates; however, that would require increasing the number of ghost cells used by FsGrid. We choose to limit the number of ghost cells to four per dimension (two ghost cells per side) to avoid the extra memory penalty; thus, we limit ourselves to using five-point stencils. A larger ghost domain would also make existing ghost communication more expensive. Furthermore, the memory footprint is the same for both methods and insignificant compared with the memory needed to store the velocity distribution function for each spatial cell, as shown in Table 1. The filtering operator presented in this work has been used to aid in 6D simulations performed with Vlasiator with respect to efficiently alleviating the artifacts introduced by the staircase effect.

Code and data availability. The Vlasiator simulation code is distributed under the GPL-2 open-source license at https://github.com/fmihpc/vlasiator (last access: June 2022). In Vlasiator 5.0 (https://doi.org/10.5281/ZENODO.3640594, von Alfthan et al., 2020), spatial AMR was introduced to enable the 6D simulations. The spatial filtering method as discussed in this work was introduced in Vlasiator 5.1 (https://doi.org/10.5281/ZENODO.4719554, Pfau-Kempf et al., 2021). The more efficient triangle filtering operator was introduced in Vlasiator 5.2.1 (https://doi.org/10.5281/ZENODO.6782211, Pfau-Kempf et al., 2022). The Analysator software (https://doi.org/10.5281/zenodo.4462515, Battarbee et al., 2021) was used to produce the presented figures. Data presented in this paper can be accessed by following the data policy on the Vlasiator website.

Author contributions. KP carried out most of the study, wrote most of the manuscript, and led the development of the numerical method presented in this work. MB, YPK, and UG actively participated in the development, testing, and optimization of the filtering method. MP is the PI of the Vlasiator model. All listed co-authors have reviewed this work and actively contributed to the discussion of the results and the writing of this manuscript. All co-authors have seen the final version of the paper and have agreed to its submission for publication.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Acknowledgements. The simulations presented in this work were run on the LUMI supercomputer (Palmroth and the Vlasiator team, 2022) during its first pilot phase, and the analysis of the results was done on the "Vorna" cluster of the University of Helsinki. We acknowledge the European Research Council for the QuESpace Starting Grant (grant no. 200141), with which Vlasiator was developed, and the PRESTISSIMO Consolidator Grant (grant no. 682068), awarded to further develop Vlasiator and use it for scientific investigations. The CSC – IT Center for Science in Finland and the PRACE Tier-0 supercomputer infrastructure at HLRS Stuttgart (grant no. 2019204998) are acknowledged, as they made these results possible. The authors wish to thank the Finnish Grid and Cloud Infrastructure (FGCI) for supporting this project with computational and data storage resources. Financial support from the Academy of Finland (grant nos. 312351, 328893, 322544, 339756, and 338629) is also acknowledged.

Financial support. This research has been supported by the Academy of Finland (grant nos. 312351, 328893, 322544, 339756, and 338629) and the European Research Council, H2020 European Research Council.

Open-access funding was provided by the Helsinki University Library.

Review statement. This paper was edited by Ludovic Räss and reviewed by Yuri Omelchenko and one anonymous referee.

References

Battarbee, M., Hannuksela, O. A., Pfau-Kempf, Y., von Alfthan, S., Ganse, U., Jarvinen, R., Suni, L. J., Alho, M., Turc, L., Honkonen, I., Brito, T., and Grandin, M.: fmihpc/analysator: v0.9 (v0.9), Zenodo [code], https://doi.org/10.5281/zenodo.4462515, 2021.

Birchfield, S.: Image processing and analysis, Cengage Learning, https://www.vlebooks.com/vleweb/product/openreader?id=none&isbn=9781337515627 (last access: March 2022), oCLC: 1256700118, 2017.

Cook, R. L.: Stochastic sampling in computer graphics, ACM Transactions on Graphics, 5, 51–72, https://doi.org/10.1145/7529.8927, 1986.

Grandin, M., Battarbee, M., Osmane, A., Ganse, U., Pfau-Kempf, Y., Turc, L., Brito, T., Koskela, T., Dubart, M., and Palmroth, M.: Hybrid-Vlasov modelling of nightside auroral proton precipitation during southward interplanetary magnetic field conditions, Ann. Geophys., 37, 791–806, https://doi.org/10.5194/angeo-37-791-2019, 2019.

Hesse, M., Aunai, N., Birn, J., Cassak, P., Denton, R. E., Drake, J. F., Gombosi, T., Hoshino, M., Matthaeus, W., Sibeck, D., and Zenitani, S.: Theory and Modeling for the Magnetospheric Multiscale Mission, Space Sci. Rev., 199, 577–630, https://doi.org/10.1007/s11214-014-0078-y, 2014.

Honkonen, I.: DCCRG: Distributed cartesian cell-refinable grid, GitHub, https://github.com/fmihpc/dccrg/ (last access: June 2022), 2022.

Honkonen, I., von Alfthan, S., Sandroos, A., Janhunen, P., and Palmroth, M.: Parallel grid library for rapid and flexible simulation development, Comput. Phys. Commun., 184, 1297–1309, https://doi.org/10.1016/j.cpc.2012.12.017, 2013.

Kempf, Y., Pokhotelov, D., Gutynska, O., III, L. B. W., Walsh, B. M., von Alfthan, S., Hannuksela, O., Sibeck, D. G., and Palmroth, M.: Ion distributions in the Earth's foreshock: Hybrid-Vlasov simulation and THEMIS observations, J. Geophys. Res.-Space Physics, 120, 3684–3701, https://doi.org/10.1002/2014ja020519, 2015.

Londrillo, P. and del Zanna, L.: On the divergence-free condition in Godunov-type schemes for ideal magnetohydrodynamics: the upwind constrained transport method, J. Comput. Phys., 195, 17–48, https://doi.org/10.1016/j.jcp.2003.09.016, 2004.

Nishikawa, K., Duţan, I., Köhn, C., and Mizuno, Y.: PIC methods in astrophysics: simulations of relativistic jets and kinetic physics in astrophysical systems, Living Rev. Comput. Astrophys, 7, 1, https://doi.org/10.1007/s41115-021-00012-0, 2021.

Palmroth, M. and the Vlasiator team: FsGrid – a lightweight, static, cartesian grid for field solvers, GitHub [code], https://github.com/fmihpc/fsgrid/ (last access: June 2022), 2020.

Palmroth, M. and the Vlasiator team: Vlasiator: Hybrid-Vlasov Simulation Code, GitHub [code], https://github.com/fmihpc/vlasiator/ (last access: June 2022), 2022.

Palmroth, M., Hoilijoki, S., Juusola, L., Pulkkinen, T. I., Hietala, H., Pfau-Kempf, Y., Ganse, U., von Alfthan, S., Vainio, R., and Hesse, M.: Tail reconnection in the global magnetospheric context: Vlasiator first results, Ann. Geophys., 35, 1269–1274, https://doi.org/10.5194/angeo-35-1269-2017, 2017.

Palmroth, M., Ganse, U., Pfau-Kempf, Y., Battarbee, M., Turc, L., Brito, T., Grandin, M., Hoilijoki, S., Sandroos, A., and von Alfthan, S.: Vlasov methods in space physics and astrophysics, Living Reviews in Computational Astrophysics, 4, 1 https://doi.org/10.1007/s41115-018-0003-2, 2018.

Pfau-Kempf, Y., Von Alfthan, S., Sandroos, A., Ganse, U., Koskela, T., Battarbee, M., Hannuksela, O. A., Honkonen, I., Papadakis, K., Grandin, M., Pokhotelov, D., Zhou, H., Kotipalo, L., and Alho, M.: fmihpc/vlasiator: Vlasiator 5.1, Zenodo [code], https://doi.org/10.5281/ZENODO.4719554, 2021.

Pfau-Kempf, Y., Von Alfthan, S., Ganse, U., Sandroos, A., Battarbee, M., Koskela, T., Hannuksela, O., Honkonen, I., Papadakis, K., Kotipalo, L., Zhou, H., Grandin, M., Pokhotelov, D., and Alho, M.: fmihpc/vlasiator: Vlasiator 5.2.1, Zenodo [code], https://doi.org/10.5281/ZENODO.6782211, 2022.

Roscoe, A. J. and Blair, S. M.: Choice and properties of adaptive and tunable digital boxcar (moving average) filters for power systems and other signal processing applications, in: 2016 IEEE International Workshop on Applied Measurements for Power Systems (AMPS), IEEE, https://doi.org/10.1109/amps.2016.7602853, 2016.

Turc, L., Roberts, O. W., Archer, M. O., Palmroth, M., Battarbee, M., Brito, T., Ganse, U., Grandin, M., Pfau-Kempf, Y., Escoubet, C. P., and Dandouras, I.: First Observations of the Disruption of the Earth's Foreshock Wave Field During Magnetic Clouds, Geophys. Res. Lett., 46, 12644–12653, https://doi.org/10.1029/2019gl084437, 2019.

Vay, J.-L. and Godfrey, B. B.: Modeling of relativistic plasmas with the Particle-In-Cell method, Comptes Rendus Mécanique, 342, 610–618, https://doi.org/10.1016/j.crme.2014.07.006, 2014.

Vay, J.-L., Almgren, A., Bell, J., Ge, L., Grote, D., Hogan, M., Kononenko, O., Lehe, R., Myers, A., Ng, C., Park, J., Ryne, R., Shapoval, O., Thévenet, M., and Zhang, W.: Warp-X: A new exascale computing platform for beam–plasma simulations, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 909, 476–479, https://doi.org/10.1016/j.nima.2018.01.035, 2018.

von Alfthan, S., Pokhotelov, D., Kempf, Y., Hoilijoki, S., Honkonen, I., Sandroos, A., and Palmroth, M.: Vlasiator: First global hybrid-Vlasov simulations of Earth's foreshock and magnetosheath, J. Atmos. Sol.-Terr. Phys., 120, 24–35, https://doi.org/10.1016/j.jastp.2014.08.012, 2014.

von Alfthan, S., Pfau-Kempf, Y., Sandroos, A., Ganse, U., Hannuksela, O., Honkonen, I., Battarbee, M., Koskela, T., and Pokhotelov, D.: fmihpc/vlasiator: Vlasiator 5.0, Zenodo [code], https://doi.org/10.5281/ZENODO.3640594, 2020.

Zerroukat, M. and Allen, T.: A three-dimensional monotone and conservative semi-Lagrangian scheme (SLICE-3D) for transport problems, Q. J. Roy. Meteorol. Soc., 138, 1640–1651, https://doi.org/10.1002/qj.1902, 2012.