



HAL
open science

SIX-DIMENSIONAL ADAPTIVE SIMULATION OF THE VLASOV EQUATIONS USING A HIERARCHICAL BASIS

Erwan Deriaz, Sébastien Peirani

► **To cite this version:**

Erwan Deriaz, Sébastien Peirani. SIX-DIMENSIONAL ADAPTIVE SIMULATION OF THE VLASOV EQUATIONS USING A HIERARCHICAL BASIS. 2016. hal-01419750v1

HAL Id: hal-01419750

<https://hal.science/hal-01419750v1>

Preprint submitted on 19 Dec 2016 (v1), last revised 12 Jan 2018 (v2)

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

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

SIX-DIMENSIONAL ADAPTIVE SIMULATION OF THE VLASOV EQUATIONS USING A HIERARCHICAL BASIS

ERWAN DERIAZ* AND SÉBASTIEN PEIRANI†

Abstract. We present an original adaptive scheme using a dynamically refined grid for the simulation of the six-dimensional Vlasov-Poisson equations. The distribution function is represented in a hierarchical basis retaining only the most significant coefficients. This allows considerable savings in terms of computational time and memory usage. The proposed scheme massively relies on the computational Adaptive Mesh Refinement framework. For the moment we embed it into a simple finite difference method. Numerical experiments are presented for the d -dimensional Vlasov-Poisson equations in the full $2d$ -dimensional phase space for $d = 1, 2$ and 3 . The six-dimensional case is compared to a GADGET N-body simulation.

keywords: adaptive mesh refinement, hierarchical basis, wavelet, multiresolution analysis, finite differences, phase-space simulations, Vlasov-Poisson equations

1. Introduction. The essential idea of Adaptivity in Numerical Analysis lies in the representation of a complex system with a reduced number of elements. It aims at decreasing the memory resources and the computational time necessary to simulate this system. Often it boils down to trading off numerical volume (computer memory, number of elementary operations) against an increase in the implementation complexity (data structures, algorithms).

The most common adaptive technique used in Mechanics [18, 33, 38, 44, 45], Physics [5, 35] and Astrophysics [1, 49, 50] is called the Adaptive Mesh Refinement where a Cartesian grid refines locally into finer Cartesian grids. It is often associated with finite volume methods [33, 44, 50]. Among AMR methods we can distinguish two different approaches: the patch-based AMR where the data structure is a collection of grids of varying sizes and accuracies embedded the ones in the others [35, 47] and the fully-threaded tree AMR where the points or cells (or groups of points or cells) are stored in the nodes or in the leaves of a large tree structure [9, 29, 44, 45, 50]. We use this latter method.

We are interested in the adaptive methods tackling the simulation of large systems. Most of them rely on the discretisation of the solution in a reduced functional basis: the finite elements where the underlying mesh does not have to be a Cartesian one [34, 46], the wavelets which are associated to a vast non-linear approximation theory [13], the spectral methods which show a good numerical accuracy [24, 43]. The hierarchical bases [5] are special kinds of wavelet bases where the primal scaling function and the primal wavelet are identical –thought the duals being still different. Sparse grids method itself [5, 42] is a sub-class of hierarchical bases. It is similar to the hyperbolic –anisotropic– hierarchical bases. Dealing with high dimensional problems it allows one to approximate multi-dimensional functions which have no medium or high frequency oscillations [8, 30]. The tensorisation technique which consists in expressing a multi-dimensional function as the product of matrices of one-dimensional functions –one matrix per dimension– also allows to simulate high dimensional problems such as the three-dimensional Vlasov-Poisson equations when the effective dimensionality of the problem is low [39, 41].

Passing from the classical two and three dimensional AMR to a six-dimensional

*Institut Jean Lamour - Faculté des Sciences et Technologies - 54506 Vandoeuvre-lès-Nancy (erwan.deriaz@univ-lorraine.fr),

†Institut d’Astrophysique de Paris - 98 bis boulevard Arago - 75014 Paris (peirani@iap.fr)

one forces us to adopt an efficient implementation and to apply simplified numerical methods. Hence we propose a rather simple method: the interpolatory hierarchical basis is associated with an Eulerian discretisation on which we apply finite difference schemes. We avoid dissipation by using a third-order scheme for the convection.

The presented method –which largely relies on classical AMR technics [33, 50, 44, 38, 52, 5, 47, 9, 11]– is not intended to compete with present and efficient Vlasov solvers using Eulerian [3, 25, 26, 27, 51], semi-Lagrangian [48] or Lagrangian [7, 15, 17, 49] schemes, for the lower two and four dimensional phase space simulations where AMR technics have already been widely tested [5, 35, 47, 52]. But it compares advantageously with the existing six-dimensional Eulerian Vlasov-Poisson simulations [40, 51] and stands the comparison with N-body simulations [49] which are widely used for such kinds of simulations.

The reader interested in comparing the merits of the various Vlasov solvers can refer to [10] for a short and well-argued discussion on this topic. The AMR adaptivity demonstrates valuable assets when the solution develops well-localized singularities [50, 38].

In the first part Sec. 2 we describe the interpolet expansion and the AMR finite differences discretisation. In the second part Sec. 3 we introduce and detail the adaptive process. Finally, in the numerical experiments Sec. 4, we apply this hierarchical basis adaptive scheme to Vlasov-Poisson equations simulations in the full two-dimensional, four-dimensional and six-dimensional phase space. These equations modelise collisionless plasmas as well as gravitational systems.

2. The hierarchical basis adaptive scheme.

2.1. Interpolet expansion. We consider a function $f : \Omega \mapsto \mathbb{R}$, $\mathbf{x} \rightarrow f(\mathbf{x})$ with $\Omega = [0, 1]^d$ and periodic boundary conditions. We assume we know f at points $(\mathbf{x}_\lambda)_{\lambda \in \Lambda}$ located at dyadic positions indexed by $\lambda = (j, k_1, \dots, k_d)$, $j \geq 0$ and $0 \leq k_i < 2^j \forall i \in [1, d]$, and defined by:

$$\mathbf{x}_\lambda = (2^{-j}k_1, \dots, 2^{-j}k_d).$$

The set Λ recollecting the indices of the retained points $(\mathbf{x}_\lambda)_{\lambda \in \Lambda}$ is constructed according to the local behavior of f through a wavelet thresholding and respecting a gradation condition (see Sec. 3.3). We set $\Omega_\Lambda = \{\mathbf{x}_\lambda, \lambda \in \Lambda\}$. We notice that $\mathbf{x}_{(j, k_1, \dots, k_d)} = \mathbf{x}_{(j+1, 2k_1, \dots, 2k_d)}$, so the set Λ may contain different indices corresponding to a unique point.

We part the set Λ into sets Λ_j containing the indices of the level j i.e. such that $\lambda \in \Lambda_j \implies \lambda = (j, k_1, \dots, k_d)$. We set $\Omega_j = \{\mathbf{x}_\lambda, \lambda \in \Lambda_j\}$. Hence

$$\Omega_\Lambda = \bigcup_{j \geq 0} \Omega_j.$$

The set Ω_{j+1} is constructed depending on the set of points $\Omega'_j \subset \Omega_j$ where we want to refine the approximation of the function f . Hence $\Omega'_j = \Omega_j \cap \Omega_{j+1}$. Then if a point of the level $j+1$ can be interpolated from Ω'_j –with a given interpolation process–, it is activated and remains to Ω_{j+1} .

As a result, in particular, for most of the interpolation processes, a point has all its ancestors present in the coarser levels. That is if $\lambda = (j, k_1, \dots, k_d)$ belongs to Λ_j , then $(j-1, k_1/2, \dots, k_d/2)$ –where $k/2$ denotes the integer division of k by two– belongs to Λ_{j-1} . This corresponds to having a fully-threaded tree.

REMARK 2.1. *For sake of simplicity of the algorithms, we manage to make all the sets Ω_j self-sufficient regarding the basic computations: the interpolation and the differentiation at points in Ω_j call point values from Ω_j exclusively. Hence the transfer of information between the levels $j - 1$ and j is carried out only through common elements of the set $\Omega'_{j-1} = \Omega_{j-1} \cap \Omega_j$.*

The function f whose point values are known at $(\mathbf{x}_\lambda)_{\lambda \in \Lambda}$ is uniquely decomposed into the interpolet hierarchical basis $(\Phi_\lambda)_{\lambda \in \tilde{\Lambda}}$ where the set $\tilde{\Lambda}$ is such that $\Omega_{\tilde{\Lambda}} = \Omega_\Lambda$ and for all $\lambda = (j, k_1, \dots, k_d) \in \tilde{\Lambda}$ at least one k_i is odd. Doing so we exclude the case when one point belongs to two different levels: $\mathbf{x}_{(j, k_1, \dots, k_d)} = \mathbf{x}_{(j+1, 2k_1, \dots, 2k_d)}$. This induces a bijection between the sets $\tilde{\Lambda}$ and Ω_Λ . Hence the decomposition:

$$(1) \quad (f(\mathbf{x}))_{\mathbf{x} \in \Omega_\Lambda} \quad \iff \quad f = \sum_{\lambda \in \tilde{\Lambda}} d_\lambda \Phi_\lambda.$$

The multi-dimensional interpolet Φ_λ is defined as the tensorial product of one the dimensional interpolets $\varphi_{j k_i}(x_i)$. For $\lambda = (j, k_1, \dots, k_d)$,

$$\Phi_\lambda(\mathbf{x}) = \varphi_{j k_1}(x_1) \times \dots \times \varphi_{j k_d}(x_d),$$

where we use the familiar wavelet notation $\varphi_{j k}(x) = \varphi(2^j x - k)$. The interpolets $(\varphi_{j k})$ are interpolant scaling functions: $\varphi(0) = 1$ and $\varphi(k) = 0 \ \forall k \neq 0$ (see [21] and the Appendix A for more details). The algorithm transforming $(f(\mathbf{x}_\lambda))_\lambda$ into $(d_\lambda)_\lambda$ as well as its inverse have a linear complexity. This transform simply boils down to computing the residue of the interpolation of level j by level $j - 1$ for all levels $j \geq 1$.

The expansion (1) allows a straight-forward projection along any directions. For instance, let us assume we want to integrate the two-dimensional function f :

$$f(x_1, x_2) = \sum_{j=0}^{j_{\max}} \sum_{k_1, k_2=0}^{2^j-1} d_{j k_1 k_2} \varphi_{j k_1}(x_1) \varphi_{j k_2}(x_2)$$

along x_1 . As $\int_x \varphi_{j k}(x) dx = 2^{-j} \int_x \varphi(x) dx = 2^{-j}$, we have the exact integration formula

$$\begin{aligned} F(x_2) &= \int_{x_1} f(x_1, x_2) dx_1 = \sum_{j=0}^{j_{\max}} \sum_{k_2=0}^{2^j-1} \left(\sum_{k_1=0}^{2^j-1} d_{j k_1 k_2} \int_{x_1} \varphi_{j k_1}(x_1) dx_1 \right) \varphi_{j k_2}(x_2) \\ &= \sum_{j=0}^{j_{\max}} \sum_{k_2=0}^{2^j-1} 2^{-j} \left(\sum_{k_1=0}^{2^j-1} d_{j k_1 k_2} \right) \varphi_{j k_2}(x_2). \end{aligned}$$

This is particularly useful when solving the Vlasov-Poisson equations for computing the density, see Eqs. (12) and (18) in the numerical experiments Sec. 4.

2.2. Multi-dimensional interpolation. The sets (Ω_j) are dependent the ones on the others and respect an interpolation relation: all the points of Ω_j can be interpolated from the points $\Omega_{j-1} \cap \Omega_j$. This relationship shapes the set (Ω_j) as one can see in Fig. 1.

The algorithm allocates a set of possible points: a node (j, \mathbf{k}) at level j with $\mathbf{k} = (k_1, \dots, k_d) \in [0, 2^{j-1} - 1]^d$ may contain the 2^d points $\{2^{-j}(2\mathbf{k} + \varepsilon), \varepsilon \in \{0, 1\}^d\}$. From this set, we determine which points are activated and which are not under the

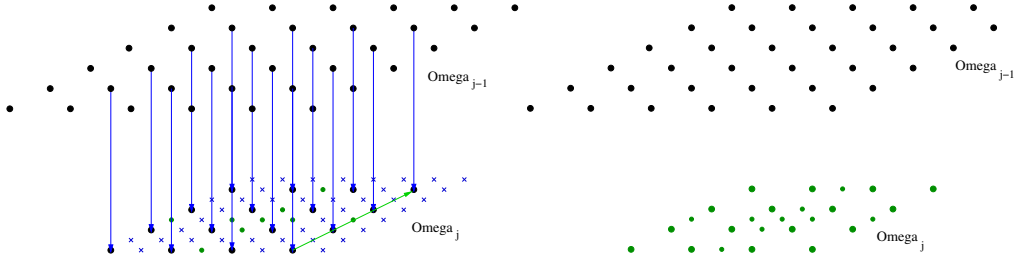


FIG. 1. Refinement of a two dimensional grid: the interpolation process from the set Ω_{j-1} in black to the set Ω_j in green. The blue crosses represent all the possible points which are not activated because they cannot be interpolated from $\Omega_{j-1} \cap \Omega_j$.

condition: a point is activated if and only if it has a place in a node and it can be interpolated from the other active points with the interpolation algorithm. Then we carry on the computation only on the active points.

Using the fourth order interpolet hierarchical basis presented in Appendix A is equivalent to applying the fourth order interpolation:

$$(2) \quad I_4 f(x) = \frac{-f(x-3h) + 9f(x-h) + 9f(x+h) - f(x+3h)}{16}.$$

In multi-dimensional simulations, the interpolation operates in a tensorial way. Hence, each point of the set $\Omega_j \setminus \Omega_{j-1}$ calls one interpolation involving four points. As a result the complexity of the interpolation algorithm remains linear with respect to the number of points independently of the dimensionality.

2.3. Finite differences on the adaptive grid. We opted for finite differences because they are easy to implement. This choice also allows the scheme to reach the third order of accuracy in any dimension for sufficiently smooth solution at a linear price whatever the number of dimensions. Meanwhile its accuracy and its stability are comparable to classical two- and three-dimensional AMR finite-volume methods [44, 38].

Nevertheless it is not optimal for simulating Vlasov-Poisson problem where semi-Lagrangian methods perform much better [27]. In particular it suffers several drawbacks. First the time step has to satisfy a CFL condition Eq. (5). Second one may observe the appearance of spurious oscillations and non-positive values of the distribution function in the case of non-smooth solutions. These issues can be addressed thanks to WENO schemes and positivity-preserving methods [12]. Third contrarily to finite-volume methods, this method does not conserve the mass on non-uniform meshes.

To apply the finite difference scheme in the multi-level adaptive grid Fig. (1), we navigate through the sets Ω_j for j going from 1 to j_{\max} . The set Ω_1 is a uniform grid with 2^d points and a step equal to $1/2$. We apply the fifth order differentiation with periodic boundary conditions. Then we apply recursively the following operations for j from 2 to j_{\max} :

- we transfer the differentiation values obtained at the points belonging to the intersection $\Omega_{j-1} \cap \Omega_j$ from Ω_{j-1} to Ω_j ,
- we interpolate them on all the points of Ω_j ,
- we compute the differentiation using a high-order scheme and replace the interpolated values by these more local ones when they exist.

For the numerical experiments, we apply the differentiations formula of Appendix B. Choosing the fifth-order scheme when possible comes with little cost (an increase of less than 20%) and improves the accuracy and the conservations. The numerical errors resulting from the differentiation scheme Appendix B are at least of third-order as shown in the numerical experiments Sec. 4.1.

2.4. CFL stability condition on the adaptive grid. Let us consider the following equation for the convection of a distribution function f by a velocity function $\mathbf{v} : \mathbb{T}^d \mapsto \mathbb{R}^d$:

$$(3) \quad \partial_t f(t, \mathbf{x}) + \mathbf{v}(\mathbf{x}) \cdot \nabla f(t, \mathbf{x}) = 0$$

where $\nabla f = (\partial_1 f, \partial_2 f, \dots, \partial_d f)$ and $\mathbf{v} \cdot \nabla f = \sum_i v_i \partial_i f$. In order to deal with any convection equation of the form Eq. (3) using the hierarchical basis adaptive scheme from Part 2.1, we apply the third-order upwind finite difference scheme described in Appendix B. A particular care was given to the CFL condition which provides an upper limit to the time step.

We optimize the CFL condition in order to benefit from the adaptivity of the grid. In time coordinate we apply a fourth-order Runge-Kutta scheme [31] to the equation (3). Given the fact that this adaptive scheme does not conserve the mass yet, we overlook the symplectic nature of the Vlasov equations. For finite difference schemes, on a uniform grid with space step h , the CFL condition on the time step δt is given by:

$$(4) \quad \delta t \leq C \frac{h}{\max_{\mathbf{x} \in \Omega} \|\mathbf{v}(\mathbf{x})\|_{\ell^1}}$$

with the constant C approximately equal to 1.73 in our case (from [19] and experiments by the author) and $\|\mathbf{v}(\mathbf{x})\|_{\ell^1} = \sum_{i=1}^d |v_i|$. As already noticed by [35], in the case of an adaptive grid, the space step h of the grid depends on the position \mathbf{x} in the computational domain Ω . Then this allows to mitigate the CFL condition. Hence the CFL condition becomes:

$$(5) \quad \delta t \leq C \min_{\mathbf{x} \in \Omega} \frac{h(\mathbf{x})}{\|\mathbf{v}(\mathbf{x})\|_{\ell^1}},$$

In phase-space coordinates, this condition on the time stepping is particularly advantageous since the high velocity parts of the grid generally present a low refinement: usually $h(\mathbf{x}) \gg h(\mathbf{0})$ for \mathbf{x} such that $\|\mathbf{v}(\mathbf{x})\| \gg \|\mathbf{v}(\mathbf{0})\|$, so the time step δt remains large independently of $\max_{\mathbf{x} \in \Omega} \|\mathbf{v}(\mathbf{x})\|_{\ell^1}$.

3. Adaptivity.

3.1. Brief error analysis. Numerical errors appear in three circumstances: when a new point is interpolated, when the differentiation scheme is applied and when the solution is updated with the time scheme.

We assume that the solution is C^∞ , which is true for Vlasov-Poisson equations starting from a C^∞ initial condition. Then with the fourth order hierarchical basis we use in the numerical experiments, the interpolation error is of order $O(h^4)$, the finite differences Sec. 2.3 provide a differentiation of order $O(h^3)$, and the fourth-order Runge-Kutta scheme is of order $O(\delta t^4)$ which means $O(h^4)$ under the CFL condition.

As a result the dominating error theoretically comes from the third-order approximation of the derivative of the distribution function f . This error is detailed in Appendix B. For the various third-order finite differences, it is of the type : $Ch^3 f^{(4)}$.

3.2. Minimization of the nonlinear approximation error. It is known from the nonlinear approximation theory [13] that for $\Phi \in B_q^s(L^p)$ the hierarchical basis expansion Eq. (1) allows to compute the superior bound:

(6)

$$\|f\|_{B_q^s(L^p)} = O\left(\left\|\left(2^{sj}2^{-\frac{dj}{p}}\|(d_\lambda)_{\lambda \in \Omega_j}\|_{\ell_p}\right)_{j \geq 0}\right\|_{\ell_q}\right)$$

Minimizing the L^p -norm of the error on the s -th derivative of the distribution function f is equivalent to minimizing the $B_p^s(L^p)$ -norm of the residue. For a given $\varepsilon > 0$ this corresponds to discarding the coefficients d_λ such that $2^{(s-\frac{d}{p})j}|d_\lambda| < \varepsilon$. This provides the criterion to save the most significant coefficients d_λ :

(7)

$$|d_\lambda| \geq 2^{(-s+\frac{d}{p})j}\varepsilon.$$

The exponent $\alpha = -s + \frac{d}{p}$ advocates a flatter refinement in higher number of dimensions. In our case, we aim at minimizing the L^p -norm of the error on the gradient of the distribution function ($s = 1$) in order to minimize the L^p -norm of the error on the distribution function in future times. But because of the intrication of the levels Ω_j the ones in the other through the interpolation scheme, and because of the scarcity of points in six dimensions, the theoretical exponent $\alpha = -1$ to minimize the L^∞ -norm of the error is not optimal as seen in the numerical experiments. There is a kind of shift to a larger optimal α .

In a similar fashion, the AMR users apply criteria on derivatives of lower orders such as: the absolute value of the function, *e.g.* in [50] the AMR maps a mass repartition function and each cell contains the same amount of mass, the gradient in Harten's Discrete Framework [33, 45] or, more recently, the second derivatives in [52]. Such criteria allow one to automatically and efficiently refine or de-refine the computational grid according to the local regularity of the function without too many theoretical considerations about the nonlinear approximation.

3.3. Refining and coarsening in practice. Assume we have defined an $\alpha \in \mathbb{R}$ relevant for the numerical experiment and an $\varepsilon > 0$ adapted to the available computer memory.

First, in a bottom-up algorithm –for j going from j_{\max} to 2– we compute the residue of the fourth-order interpolation of the level Ω_j by the level Ω_{j-1} . It is equivalent to a transform in the isotropic fourth-order hierarchical basis Eq. (1).

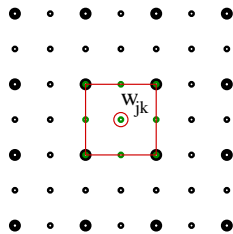


FIG. 2. Elementary cube for the refinement.

Then we gather the residues $d_{j,\mathbf{k}'}$ from Eq. (1) of the points $2^{-j}\mathbf{k}'$ inside cubes centered on points of the type $2^{-j}(2\ell_1 + 1, \dots, 2\ell_d + 1)$ as indicated in Fig. 2. For $\mathbf{k} = (2\ell_1 + 1, \dots, 2\ell_d + 1)$, this yields a refinement weight associated to this cube:

$$w_{j,\mathbf{k}}^2 = \sum_{\mathbf{e} \in \{-1,0,+1\}^d} d_{j,\mathbf{k}+\mathbf{e}}^2.$$

Then we apply the following refinement instructions in this order:

- if $w_{j,\mathbf{k}} > 2^{\alpha j}\varepsilon$, the 3^d points inside the area are refined creating 3^d nodes at level $j + 1$ (if it is not the case already);
- if $w_{j,\mathbf{k}} > 2^{\alpha j-1}\varepsilon$, then the nodes inside the area are preserved; we also activate (create or preserve) the neighboring nodes within two ranks or one rank depending if it is respectively located under the wind or not;
- we simulate an interpolation in order to see which points inside the nodes are reachable by the interpolation algorithm (see Sec. 2.2); these are activated;
- if a point $2^{-j}\mathbf{k}$ is not active then it is removed from the memory, and if a node contains no active point, it is also removed (see Appendix C for the relation between a node and the points it contains).

Sometimes we also fix a maximum level of refinement j_{Max} and we impose: $j \leq j_{\text{Max}}$. The larger α , the closer the grid will be to a uniform grid. Varying ε allows an increase or a decrease in the number of grid points in order to cope with the memory storage limit of the computer (see the numerical experiments in Sec. 4).

4. Numerical experiments. In this Section we test the adaptive numerical scheme developed in this paper in various situations. First, as this method specifically targets six-dimensional simulations, we test the transport of a Gaussian in a six-dimensional box. Then we check its properties on classical two-dimensional plasma test cases: the Landau Damping and the Bump-on-Tail instability. The simulation of the two stream instability in 2D (four-dimensional phase space) from [25] shows the limits of this method with finite differences when applied to the electric case. Finally, we present an academic six-dimensional astrophysics simulation: the merging of two halos in the full six-dimensional phase space.

4.1. Convection and stretching of a Gaussian in six dimensions. The numerical experiment consists in convecting a Gaussian function

$$f_0(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_0\|^2}{2\sigma^2}\right)$$

with $\mathbf{x}_0 = (\frac{1}{2}, \dots, \frac{1}{2})$ in the six dimensional box $\mathbb{T}^6 = [0, 1]^6$ with periodic boundary conditions, for the velocity field:

(8)

$$\mathbf{v}(x_1, x_2, \dots, x_6) = \begin{bmatrix} 1 \\ \cos(2\pi x_1 + 1) \\ \cos(2\pi x_1 + 2) \\ \vdots \\ \cos(2\pi x_1 + 5) \end{bmatrix}.$$

For any particle at position $X(t) = (X_i(t))_{1 \leq i \leq 6}$ at time $t = 0$, the constant velocity $v_1 = 1$ gives $X_1(t) = X_1(0) + t$ and, for $2 \leq i \leq 6$,

$$\int_{t=0}^1 v_i(X(t)) dt = \int_{t=0}^1 \cos(2\pi t + 2\pi X_1(0) + i - 1) dt = 0.$$

Hence the Gaussian comes back with its initial shape to its initial position at integer t . The exact solution is given by:

$$f(t, \mathbf{x}) = f_0(x_1 - t, x_2 - \frac{1}{2\pi} (\sin(2\pi x_1 + 1) - \sin(2\pi(x_1 - t) + 1)), \dots$$

(9)

$$\dots, x_6 - \frac{1}{2\pi} (\sin(2\pi x_1 + 5) - \sin(2\pi(x_1 - t) + 5)))$$

so we can compute the error exactly.

The code is written in C and parallelized with Open-MP. This experiment requires approximately $2e+8$ points over 400 time steps. It takes 33 hours on the 40 threads of a 2.50 GHz Intel Xeon E5-2670 processor with 32 Go RAM memory.

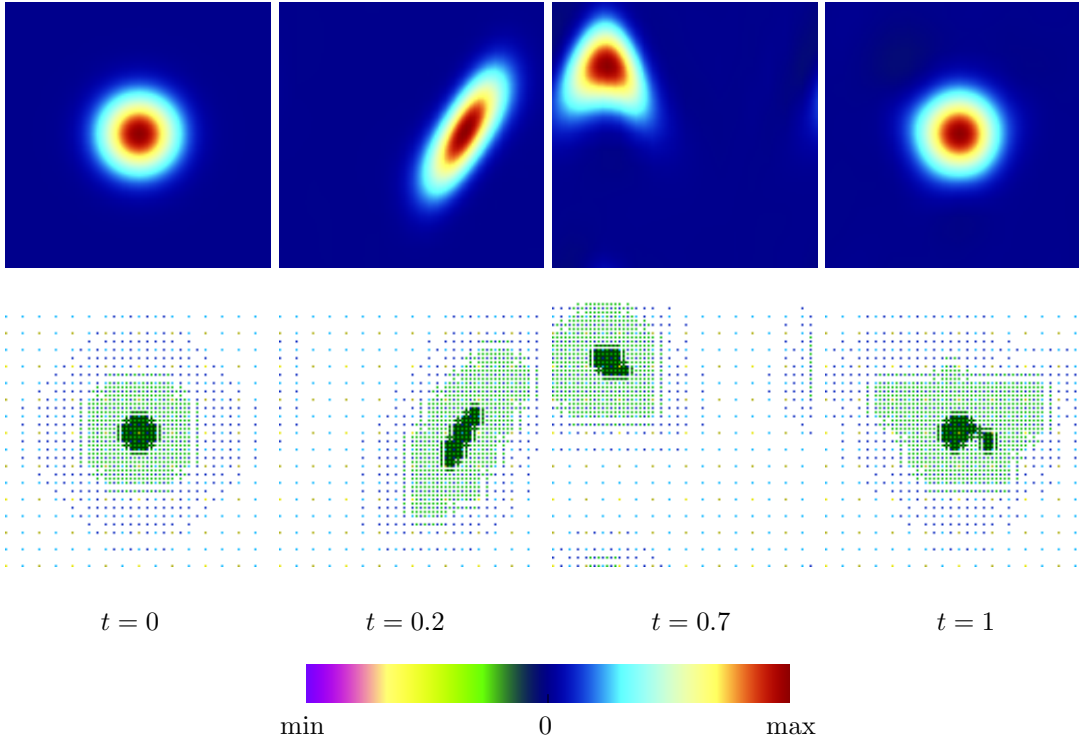


FIG. 3. Projective view in the (x_1, x_2) plan of the six-dimensional convected Gaussian at times $t = 0$, $t = 0.2$, $t = 0.7$ and $t = 1$. First row: the distribution function. Second row: the adaptive grid.

In Fig. 3, we represent a two-dimensional projection onto (x_1, x_2) of the distribution function and of the adaptive grid. The maximum refinement level (in dark green) corresponds to a 128^6 uniform grid equivalent accuracy. The time step is taken constant equal to

$$\delta t = \frac{1}{400} \quad \text{close to the CFL maximum} \quad C \frac{h}{\|\mathbf{v}\|_{\ell^1}} \approx \frac{1}{344}$$

with $C = 1.73$, $h = 2^{-7}$ and $\|\mathbf{v}\|_{\ell^1} \approx 4.642$.

Since we want to minimize the L^∞ error, we take the thresholding parameter α equal to 0 (see Sec. 3.2). The initial value of ε is equal to $3e-4$ and the standard deviation of the Gaussian $\sigma = 0.1$.

In Fig. 3, we observe that the refinement of the grid follows the stretched Gaussian. The L^∞ error primarily appears at the boundaries between the different levels of refinement ahead of the Gaussian.

Regarding the color scale of Fig. 3, the dark blue color stands for zero and the dark red for the maximum value. This color scale is used in all the other figures representing a distribution function.

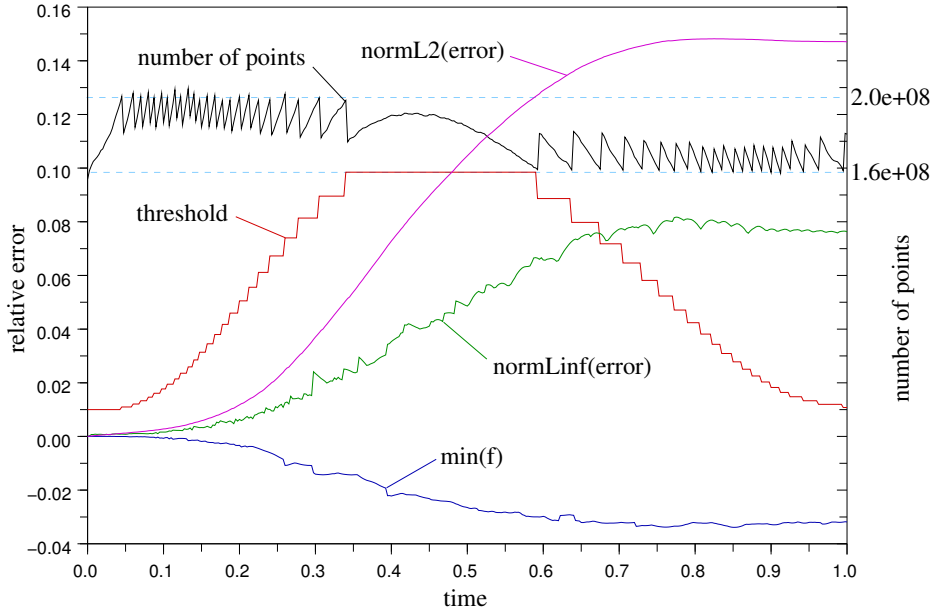


FIG. 4. Error analysis, the curve for the threshold ε was rescaled to fit the scale of the other curves. Relative errors ($\|f_{\text{num}} - f_{\text{exact}}\| / \|f_{\text{exact}}\|$) on the distribution function in L^2 and L^∞ norms.

The L^∞ -norm of the error remains inferior to 8% as shown in Fig. 4. Relative error means the ratio between the norm of the error and the norm of the Gaussian. In this figure, we also represent the L^2 -norm of the error, the minimum value of the solution (which shows that the scheme does not conserve the positivity), the threshold ε used to form the adaptive grid, the number of points contained in the grid.

The memory use coincides with the number of allocated points (see Appendix C). In order not to exceed the memory limit and optimize the memory use, we bound the number of points with a maximum and a minimum. In the present test case the maximum of points is taken equal to $2e+8$ and the minimum represents 80% of the maximum. Each time the number of points exceeds the maximum we have fixed, the threshold ε is increased by 10%, leading to a drop in the number of allocated points and memory needs. This explains the saw-teeth aspect of the curve plotting the number of points, for $t \in [0, 0.5]$. On the contrary when the memory need decreases, we take advantage of this to increase the accuracy and we decrease the threshold by 10%. This is what happens in the second part of the graph, for $t \in [0.5, 1]$.

The error increases dramatically when the threshold ε is maximum when a major stretching of the distribution function occurs, around $t = 0.5$.

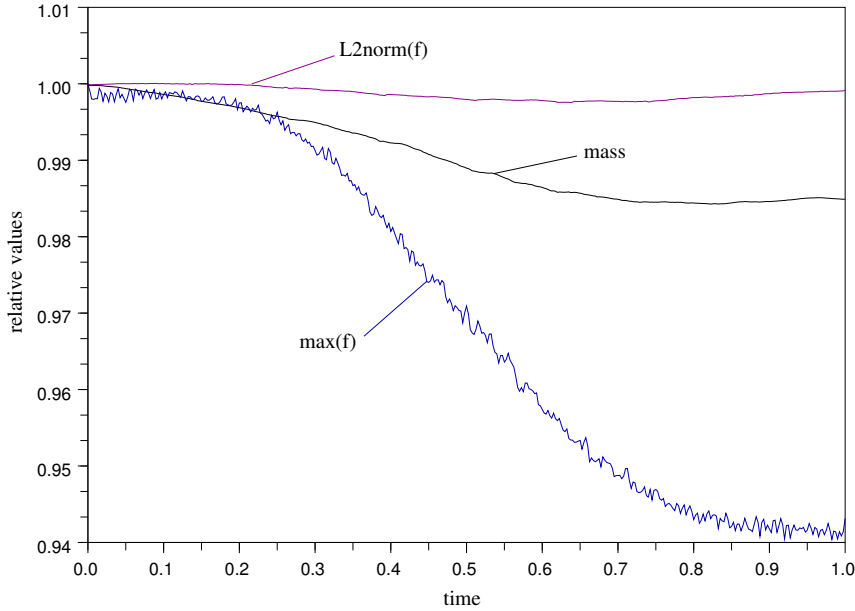


FIG. 5. Conservation analysis.

In Fig. 5, we plot the mass, the L^2 -norm and the maximum of the distribution during time. They indicate a precision of a few percents. The curves represent relative values compared to the initial one, so they equal 1 at time $t = 0$.

The fast oscillations of the maximum correspond to the shifting of the actual maximum of the solution from a point of the mesh to another one.

We apply the same experiment to assess the convergence of the hierarchical basis adaptive scheme and estimate an optimal choice for the value of the parameter α involved in the refinement process Sec. 3.2 and Sec. 3.3. We run the experiment until $t = 0.1$ and compare the numerical result with the exact solution. We plot the error in L^∞ -norm and L^2 -norm for numbers of points varying approximatively from $2e+6$ to $4e+8$ in Fig. 6.

We would expect a third-order convergence and as the number of points is linked to an average \bar{h} : $N = O(\bar{h}^{-6})$, for a p -th-order method, we would have

$$e = O(\bar{h}^p) = O((N^{1/6})^{-p}) = O\left(\frac{1}{N^{p/6}}\right),$$

leading to a convergence in $O(N^{-0.5})$. Nevertheless, experimentally on Fig. 6 we rather observe an order 0.8–0.9 convergence in L^∞ and L^2 norms. This probably means that the fifth-order finite difference scheme used in the interior of the refinement levels dominates the convergence rate. It corresponds to $p = 5$ and a convergence rate $e = O(N^{-0.83})$.

Although we would expect $\alpha = -1$ to be optimal in L^∞ approximation, it rather looks like that $\alpha = 2$ provides the best results. In the L^2 norm, the experimental results are in better agreement with the theory since $\alpha = 3$ seems to be optimal, close to the theoretically optimal $\alpha = 2$.

Since the error concentrates at the boundary of the refinement levels, the activation of a new level triggers an increase of the L^∞ error. This explains the jumps in

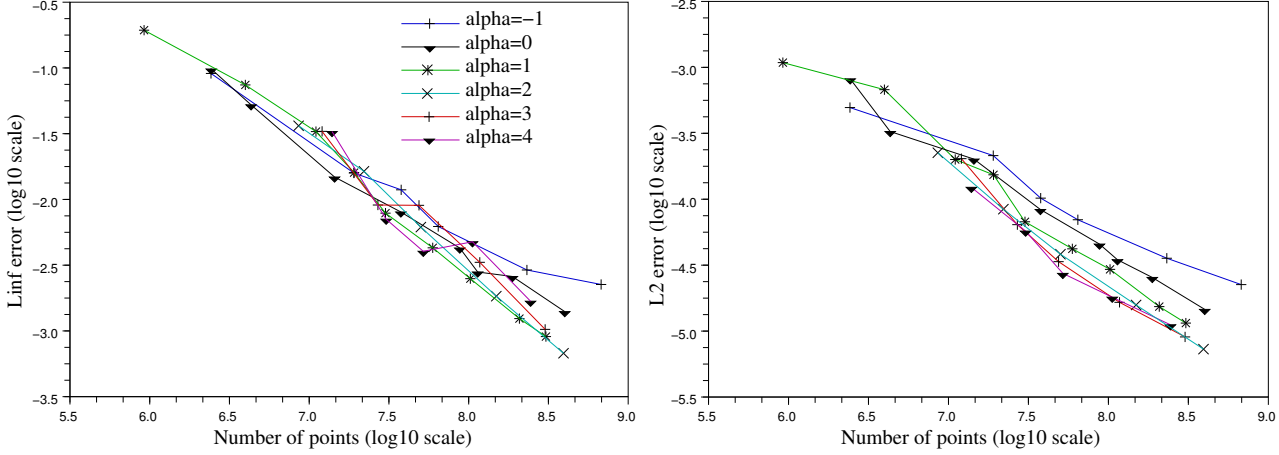


FIG. 6. Error plots in L^∞ (left) and L^2 (right) norms. Both x and y axes have a Log_{10} scale.

the L^∞ norm of the error in the convergence plot Fig. 6.

4.2. Application to plasma physics. In Plasma physics and astrophysics, most of the Vlasov-Poisson simulations rely on Particle-In-Cell methods [15, 17] as they allow 6-dimensional phase-space simulations at a reasonable computational cost. In these methods, the distribution function is coarsely discretized with Dirac functions (particles) whose positions evolve through Lagrangian schemes. Recently, grid-based Eulerian schemes have been developed [27] but they rarely pass the curse of dimensionality [42] and most of the time do not exceed four dimensions for the phase-space [32].

Landau damping. We consider a distribution function $f : \mathbb{R}^{2d} \rightarrow \mathbb{R}_+$, $(\mathbf{x}, \mathbf{v}) \mapsto f(\mathbf{x}, \mathbf{v})$ subject to the Vlasov-Poisson equation:

$$(10) \quad \partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + E(t, \mathbf{x}) \cdot \nabla_{\mathbf{v}} f = 0$$

with the electric force

$$(11) \quad E(t, \mathbf{x}) = \nabla_{\mathbf{x}} \phi(t, \mathbf{x}),$$

$$(12) \quad \Delta_{\mathbf{x}} \phi(t, \mathbf{x}) = \int_{\mathbf{v} \in \mathbb{R}^d} f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v} - \int_{\mathbf{x}, \mathbf{v} \in \mathbb{R}^d} f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v} d\mathbf{x}.$$

To make the following simulation in dimension $d = 1$, f is taken periodic in the x variable. The initial condition is given by: for $(x, v) \in [-2\pi, 2\pi] \times [-6, 6]$,

$$f_0(x, v) = \frac{1 - 0.05 \cos(0.5x)}{\sqrt{2\pi}} \exp(-v^2/2),$$

and we apply the hierarchical basis adaptive scheme for $t \in [0, 100]$. The refinement parameter α is taken equal to 1.7, the grid (Fig. 7) starts with 3,332 points and reaches a maximum of 45,522 at $t = 57.08$. The electric field (Fig. 8) reproduces its classical curve in the Landau damping with the correct damping rate.

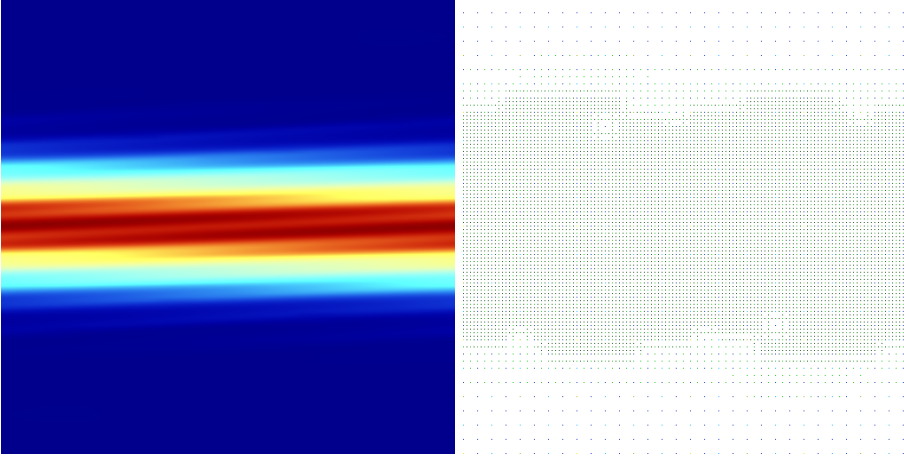


FIG. 7. The distribution function and the adaptive grid at $t = 22$ during the Landau damping experiment.

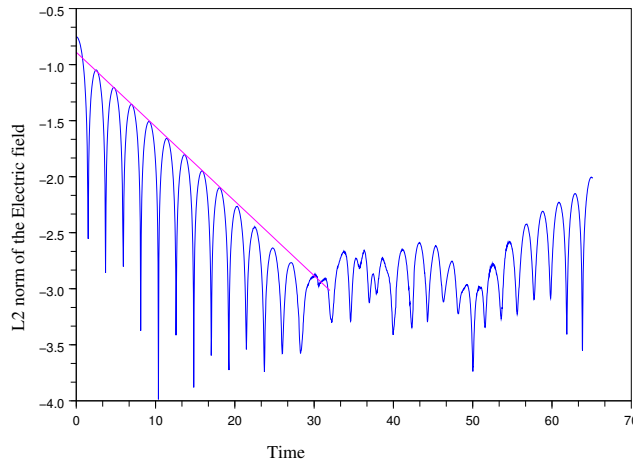


FIG. 8. Landau damping of the electric field.

Bump-on-Tail instability. In this paragraph we simulate the Bump-on-Tail instability with the hierarchical basis adaptive scheme. In [35], the authors tested a block-structured AMR using a high-order finite-volume scheme on this case. We obtain similar results.

The simulation box is $(x, v) \in [-\frac{10}{3}\pi, \frac{10}{3}\pi] \times [-10, 10]$. The initial condition is given by:

$$f_0(x, v) = \left(\frac{0.9}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} + \frac{0.2}{\sqrt{2\pi}} e^{-4(v-4.5)^2} \right) (1 + 0.04 \cos(0.3x)).$$

We integrate f in Eq. (12) thanks to its hierarchical basis expansion Eq. (1) and summing the coefficients in the velocity direction as indicated in Sec. 2.1. Then Eq. (12) and (11) are solved thanks to one-dimensional Fourier transforms.

The simulation is plotted in Fig. 9 which shows a very good agreement with the results presented in [35]. We observe that the fine mesh, in purple, equivalent to a

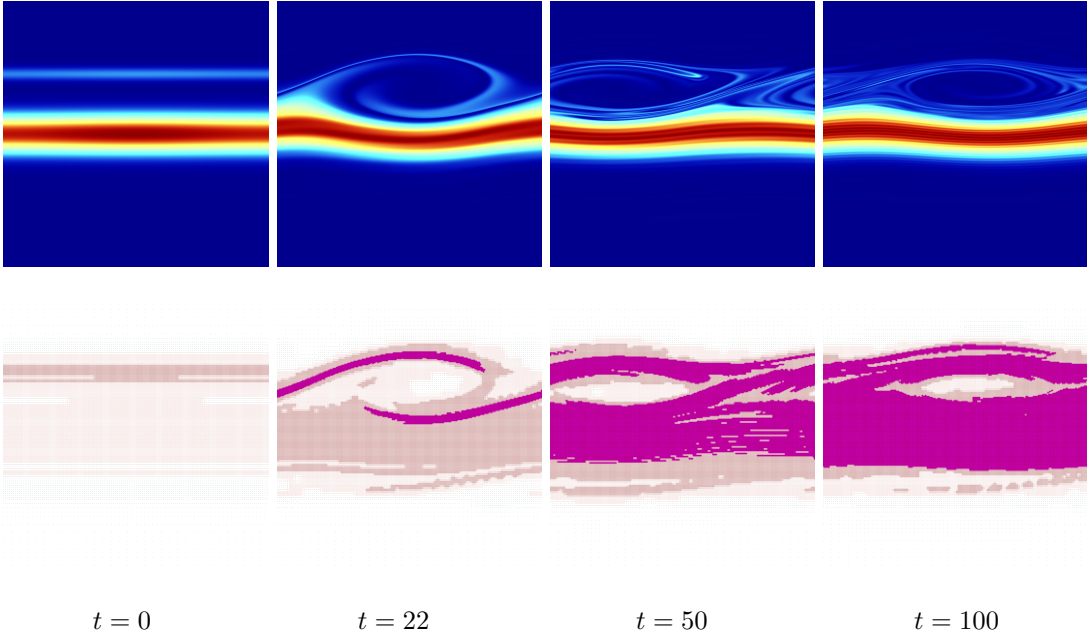


FIG. 9. *Bump-on-Tail instability (1D). First row: the distribution function. Second row: the adaptive grid .*

1024^2 uniform mesh, follows the filaments accurately. If one is interested in following precisely the behavior and the shape of the filaments, the adaptive mesh refinement method may compete with uniform grid methods even in one dimension.

In Fig. 10 we observe that the number of points varies from 17,000 at the beginning to 150,000 at the maximum complexity. The refinement parameters were fixed to $\alpha = 1.7$ and $\varepsilon = 7.18e-9$.

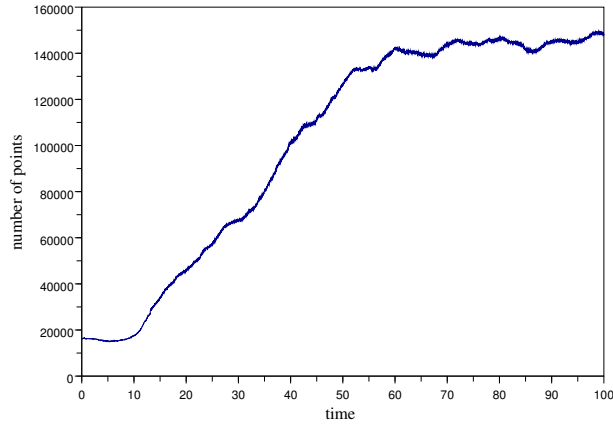


FIG. 10. *Evolution of the number of grid points in the AMR.*

Fig. 11 illustrates important defects of the proposed AMR method for solving the Vlasov-Poisson equations: it does not conserve the mass and it makes negative values appear.

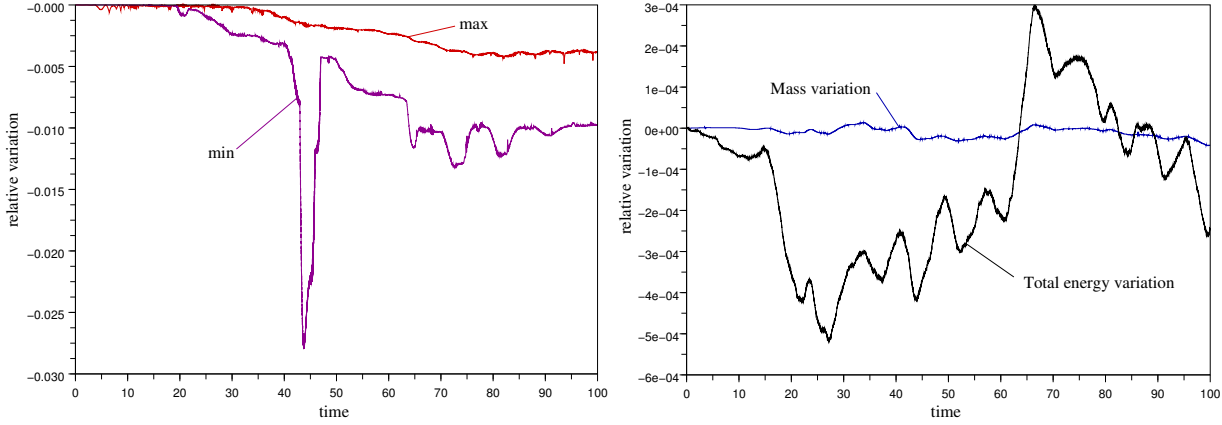


FIG. 11. Plots of some theoretically conserved quantities. Left: minimum and maximum values of the distribution function. Right: mass and total energy. Representation in relative variation ($\Delta f/f$) except for the min ($\min(f)/\max(f_0)$).

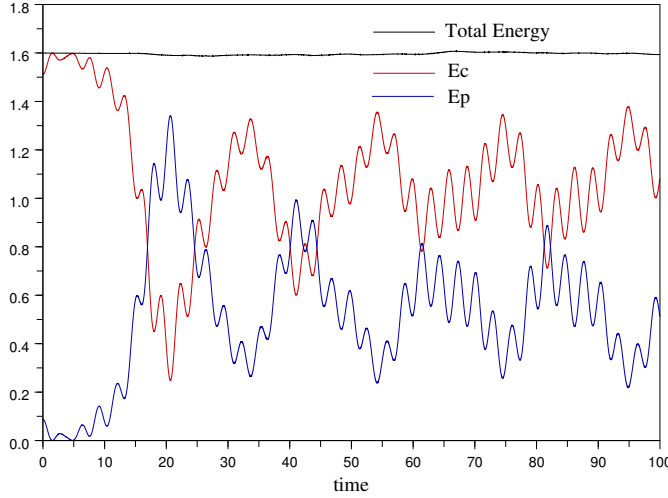


FIG. 12. Variation of the kinetic (E_c) and potential (E_p) energies. The kinetic energy and the total energy were vertically shifted by -23 .

In Fig. 12 we can see that although the kinetic energy and the potential energy tend to compensate each other, the total energy drifts away from its initial value.

Four-dimensional phase space case: the two beams instability. We apply the hierarchical basis adaptive scheme to a two-dimensional Vlasov Poisson test case: the Fijalkow Two Beams instability first presented in [25]. This is a rather new result regarding AMR simulation of Vlasov Poisson equations since except in computer science [36], authors of AMR Vlasov Poisson solvers usually restrict to one-dimensional cases [35, 47].

In the simulation box $(x, y, u, v) \in [-\frac{10\pi}{3}, \frac{10\pi}{3}]^2 \times [-3\pi, 3\pi]^2$ with periodic boundary conditions, we consider the initial distribution function:

(13)

$$f_0(x, y, u, v) = \frac{7}{4\pi} \exp\left(-\frac{u^2 + 4v^2}{8}\right) \sin^2\left(\frac{u}{3}\right) (1 + 0.05 \cos(0.3x))$$

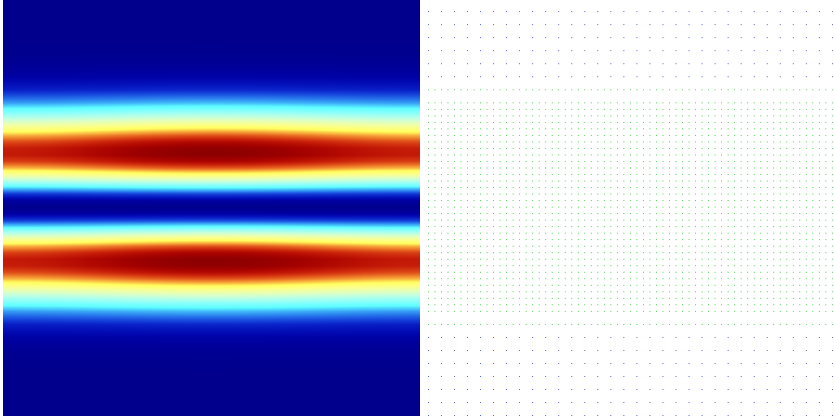


FIG. 13. *Phase space view of the initial condition Eq. (13). Cuts of the distribution function and of the adaptive grid along (x, u) at $y = v = 0$.*

This initial distribution, represented in Fig. 13, was chosen in order to maximize the two stream instability [25]. Initially, it is discretized with $4.7e+6$ points and has a maximum level of refinement equivalent to a 64^4 uniform grid. The thresholding parameters are taken as follows: $\alpha = 1.5$ and $\varepsilon = 1.59e-6$ initially, but ε varies thereafter.

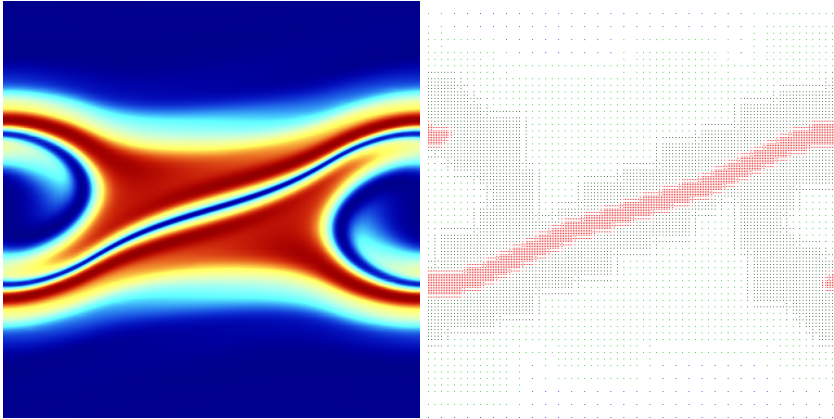


FIG. 14. *Cut along (x, u) of the distribution function at $y = v = 0$ in the phase space at time $t = 12$.*

in Fig. 14 we represent the numerical solution at $t = 12$ when the instability begins to develop. The maximum refinement goes up to 256^4 (the red points in Fig. 14 right). The number of active points increases to $53e+6$ but the threshold ε stays at $1.59e-6$ (see Fig. 15).

We limitate the number of points to a maximum of $60e+6$.

When the non-linear regime develops after $t = 12$, steep fronts appear all over the domain in the (x, y) direction. This results in a large increase of the number of points, and then of the threshold ε (Fig. 15). When the threshold becomes to large, the conserved quantities begin to vary by few percents (Fig. 16) and the energy drops (Fig. 17).

Here we measure the inability of the present finite difference AMR scheme to

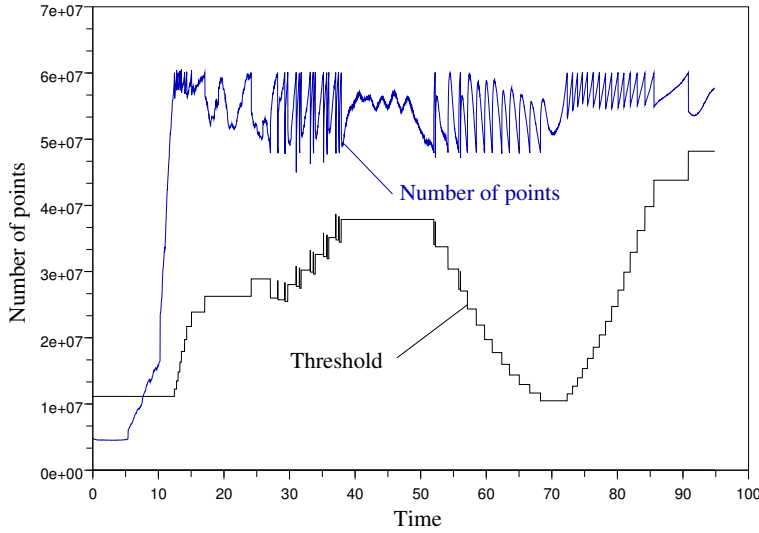


FIG. 15. *Number of points and threshold along time. The threshold curve is linearly scaled in order to fit the window.*

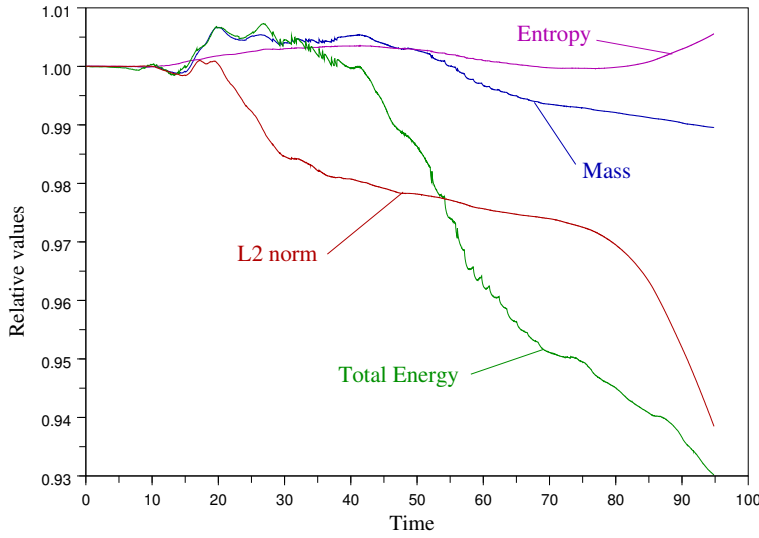


FIG. 16. *Plot of theoretically conserved quantities: the mass, the L^2 -norm, the entropy and the total energy. Relative values with respect to the initial quantities.*

capture steep filamentations distributed homogeneously in the spatial domain (x, y) . It would probably work better if we refine only in the velocity (u, v) direction.

4.3. Six-dimensional astrophysics case: the merging of two halos of dark matter. In order to apply the hierarchical basis adaptive scheme to a six-dimensional phase space problem, we switch from plasma physics to astrophysics. In plasma physics the distribution function tends to occupy the whole three-dimensional physical space while in the gravitational case it collapses, allowing the adaptive scheme to refine both in velocity variables and in space variables (see Fig. 18 and 21). As a re-

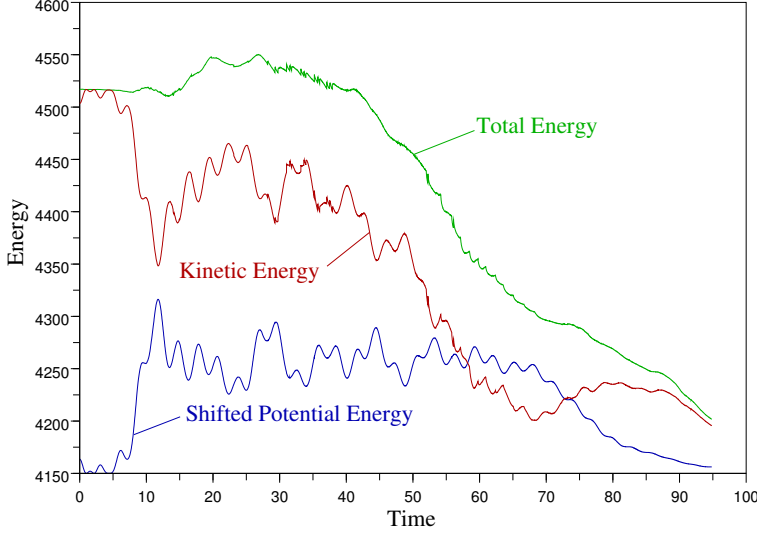


FIG. 17. Plot of the energies. The potential energy is shifted by +4150.

sult we are able to simulate the gravitational Vlasov-Poisson equations at a reasonable cost.

For a collisionless self-gravitating system such as a galaxy or a dark matter halo, the distribution function evolves in the phase space $(\mathbf{x}, \mathbf{v}) \in \mathbb{R}^6$, with $\mathbf{x} = (x, y, z)$ and $\mathbf{v} = (u, v, w)$

(14)

$$f : \begin{array}{ccc} \mathbb{R} \times \mathbb{R}^3 \times \mathbb{R}^3 & \rightarrow & \mathbb{R}^+ \\ t, \mathbf{x}, \mathbf{v} & \mapsto & f(t, \mathbf{x}, \mathbf{v}) \end{array}$$

and its motion is modelised by the gravitational Vlasov-Poisson equations:

(15)

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + E(t, \mathbf{x}) \cdot \nabla_{\mathbf{v}} f = 0$$

(16)

$$E(t, \mathbf{x}) = -\nabla_{\mathbf{x}} \phi(t, \mathbf{x})$$

(17)

$$\Delta_{\mathbf{x}} \phi(t, \mathbf{x}) = 4\pi G \rho(t, \mathbf{x})$$

where the density function ρ is given by:

(18)

$$\rho(t, \mathbf{x}) = \int_{\mathbb{R}^3} f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v}.$$

Comparing with plasma equations (11) and (12), note, in particular, the sign “−” in the force Eq. (16) switching from repulsive to attractive, and the factor “ 4π ” (not gone when the equations are non-dimensionalised in astrophysics) in Eq. (17).

These equations admit stationary solutions [2, 6]. One of them, the Plummer model, was successfully tested by Takao Fujiwara in [28] using the symmetries of

the problem in a two-dimensional plus one invariant simulation. It is given by the following distribution function:

$$(19) \quad f(r, \|\mathbf{v}\|) = \frac{3M}{7\pi^3 a^3} \left(2 \left(1 + \left(\frac{r}{a} \right)^2 \right)^{-1/2} - \|\mathbf{v}\|^2 \right)^{7/2}$$

$$\text{if } 2 \left(1 + \left(\frac{r}{a} \right)^2 \right)^{-1/2} - \|\mathbf{v}\|^2 \geq 0$$

and $f(r, \|\mathbf{v}\|) = 0$ otherwise.

We note $r^2 = x^2 + y^2 + z^2$ and $\|\mathbf{v}\|^2 = u^2 + v^2 + w^2$.

Then the potential function is given by:

$$(20) \quad \phi(r) = -\frac{GM}{a} \left(1 + \left(\frac{r}{a} \right)^2 \right)^{-1/2},$$

and the density function by:

$$(21) \quad \rho(r) = \frac{3M}{4\pi a^3} \left(1 + \left(\frac{r}{a} \right)^2 \right)^{-5/2}.$$

In these formulas, G denotes the constant of gravity, M the total mass and a a length parameter. In the following simulations, these constants are taken equal to 1.

This distribution function Eq. (19) is compactly supported in velocity but not in space.

In Fig. 18, we represent this stationary solution in an adaptive grid with a criterion based on a second-order hierarchical basis decomposition. The maximum level of refinement corresponds to a 256^6 uniform grid (in red).

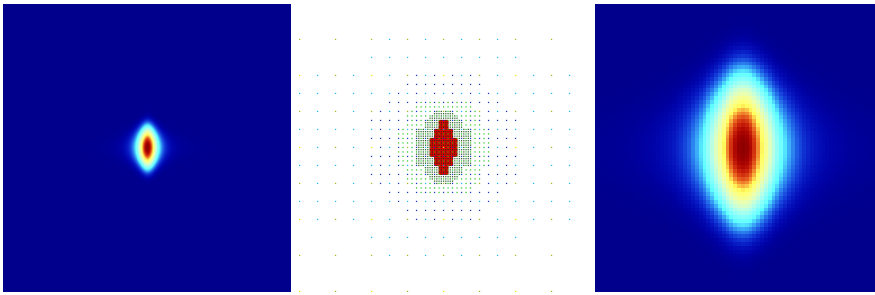


FIG. 18. *The Plummer Model. Left: two-dimensional cut at zero $(0, 0, 0, 0, 0, 0)$ in (x, u) -view of the Plummer Model. Center: cut of the corresponding adaptive grid. Right: zoom at the center.*

We apply the hierarchical basis adaptive scheme to the six-dimensional merging of two spherical Plummer models. Hence we simulate the ‘collision’ of two halos which, when separated, form stable systems. We compare the results to the ones obtained with GADGET, a N-body code extensively used in the literature [49]. Only the treecode part of this “treePM” algorithm is employed and 50 millions particles have been used to describe the merger process. We have also considered a softening length of $6.84e-4$. For other specific parameters (time step, accuracy of the relative cell-opening criterion, etc. . .), we have used the same values as those in [14]. Running

this GADGET simulation takes around 87 hours on a 64 threads node of the Horizon cluster at Institut d’Astrophysique de Paris.

The convection is solved using the scheme described in Sec. 2.4 with the interpolatory hierarchical basis of fourth-order in the expansion Eq. (1). The gravitational forces are computed in the three-dimensional Fourier space on a uniform grid of the smallest scale (i.e. with a FFT in a 256^3 grid). The time step δt is computed at the beginning of each Runge-Kutta cycle by the CFL formula Eq. (5) with $C = 1$. This value is below the CFL constant for uniform domains $C = 1.73$. At the end of each Runge-Kutta cycle, we let the mesh evolve as described in Sec. 3.3 with $\alpha = 1.5$ and $\varepsilon = 7.819e-7$ at $t = 0$ and we prescribe the maximum number of points to be $4e+8$. In order to exploit memory optimally, we fix the minimum number of nodes at 80% of this maximum. The value $\alpha = 1.5$ corresponds to the theoretical minimization of the approximation error of f in L^4 -norm and of ∇f in $L^{12/5}$ -norm. It is fixed arbitrary. The choice of $\varepsilon = 7.819e-7$ implies the activation of 151,416,766 points at $t = 0$. This is a compromise between the speed of the computations and the accuracy at the beginning of the simulation.

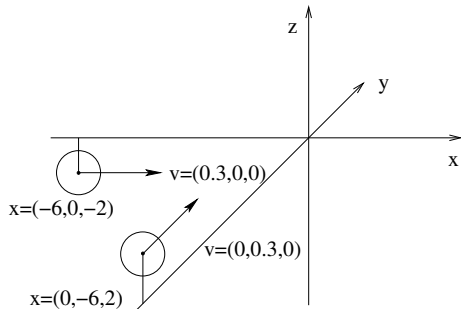


FIG. 19. *Initial conditions for the interaction of two Plummer models.*

The initial condition is sketched in Fig. 19. Each sphere represents a stationary solution Eq. (19). The size of the box is $[-12, 12]^3 \times [-8, 8]^3$. We apply periodic boundary conditions. In order to visualize the evolution of the six-dimensional numerical solution in the phase space, we make two-dimensional cuts at zero, and both two-dimensional and three-dimensional projections.

The color scale is the same as in Fig. 3.

In Fig. 20, we see the collision and merging of the two spheres in a three-dimensional (x, y, z) projection.

Fig. 21 represents cuts along the (z, w) plan of the six-dimensional phase space at $x = y = u = v = 0$. It only gives an insight of the complexity of the six-dimensional solution. Fig. 21 last row represents a cut of thickness one fiftieth of the domain length (i.e. $0 \pm R_{\max}/50$ in the space directions and $0 \pm V_{\max}/50$ for the velocity directions). The GADGET N-body simulation provides noisy images due to the Poissonian nature of the particles distribution inside the pixels [49]. Comparing with the GADGET simulation validates the results obtained with the hierarchical basis adaptive scheme simulation.

In Fig. 22, we plot:

- the mass $M(t) = \int_{\mathbf{x}, \mathbf{v}} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{x} d\mathbf{v}$
- the L^2 -norm $\|f\|_2(t) = \left(\int_{\mathbf{x}, \mathbf{v}} f(\mathbf{x}, \mathbf{v}, t)^2 d\mathbf{x} d\mathbf{v} \right)^{\frac{1}{2}}$

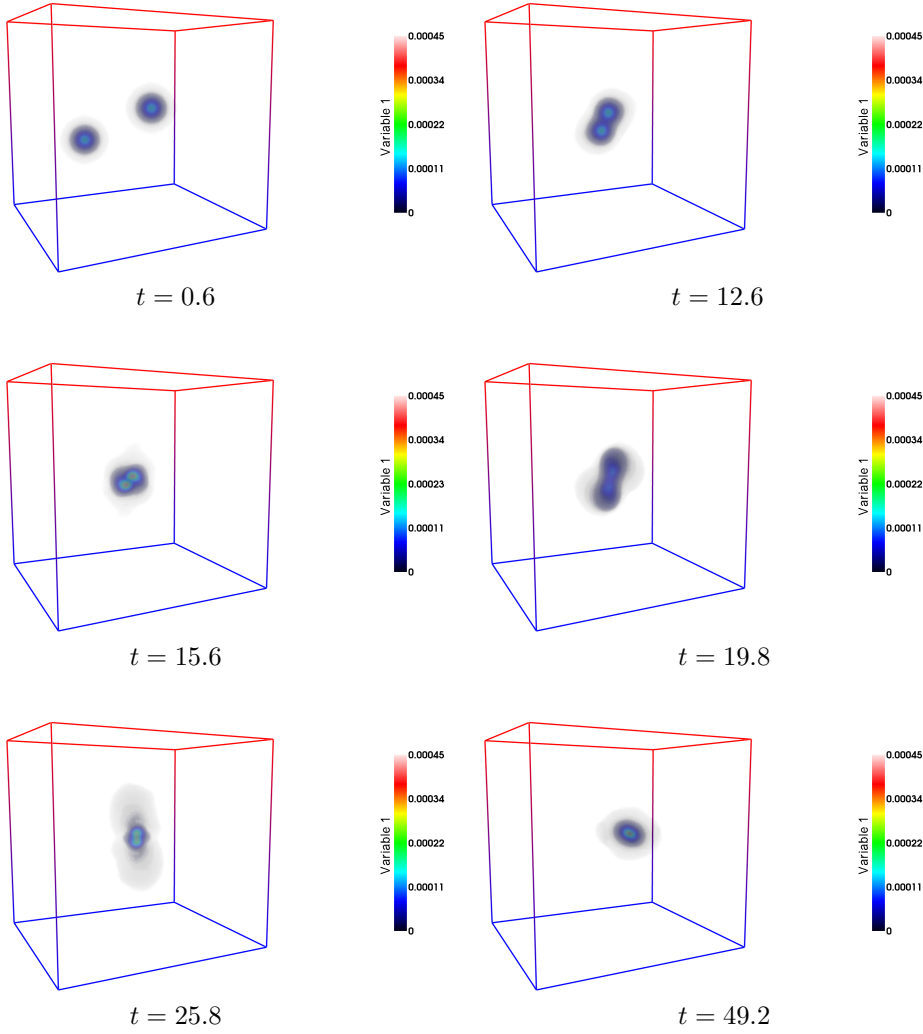


FIG. 20. Three-dimensional view in the physical space (x, y, z) of the collision of two Plummer spheres at times $t = 0.6, 12.6, 15.6, 19.8, 25.8, 49.2$. This visualization was obtained for $t \in [0, 60]$ with the hierarchical basis adaptive scheme of second order.

- the entropy $S(t) = \int_{\mathbf{x}, \mathbf{v}} -f(\mathbf{x}, \mathbf{v}, t) \ln(f(\mathbf{x}, \mathbf{v}, t)) d\mathbf{x} d\mathbf{v}$ which are conserved quantities for the Vlasov-Poisson equations.

In Fig. 24 we also plot

- the kinetic energy
- (22)

$$E_c(t) = \int_{\mathbf{x}, \mathbf{v}} \frac{\|\mathbf{v}\|^2}{2} f(\mathbf{x}, \mathbf{v}, t) d\mathbf{x} d\mathbf{v}$$

The mass is not conserved and undergoes variations in $\pm 3\%$ range. The entropy increases reasonably (+4%) and this increase can be explained by the usual numerical averaging of the filamentation [49].

We define the complexity of the solution at a given time t as the number of points

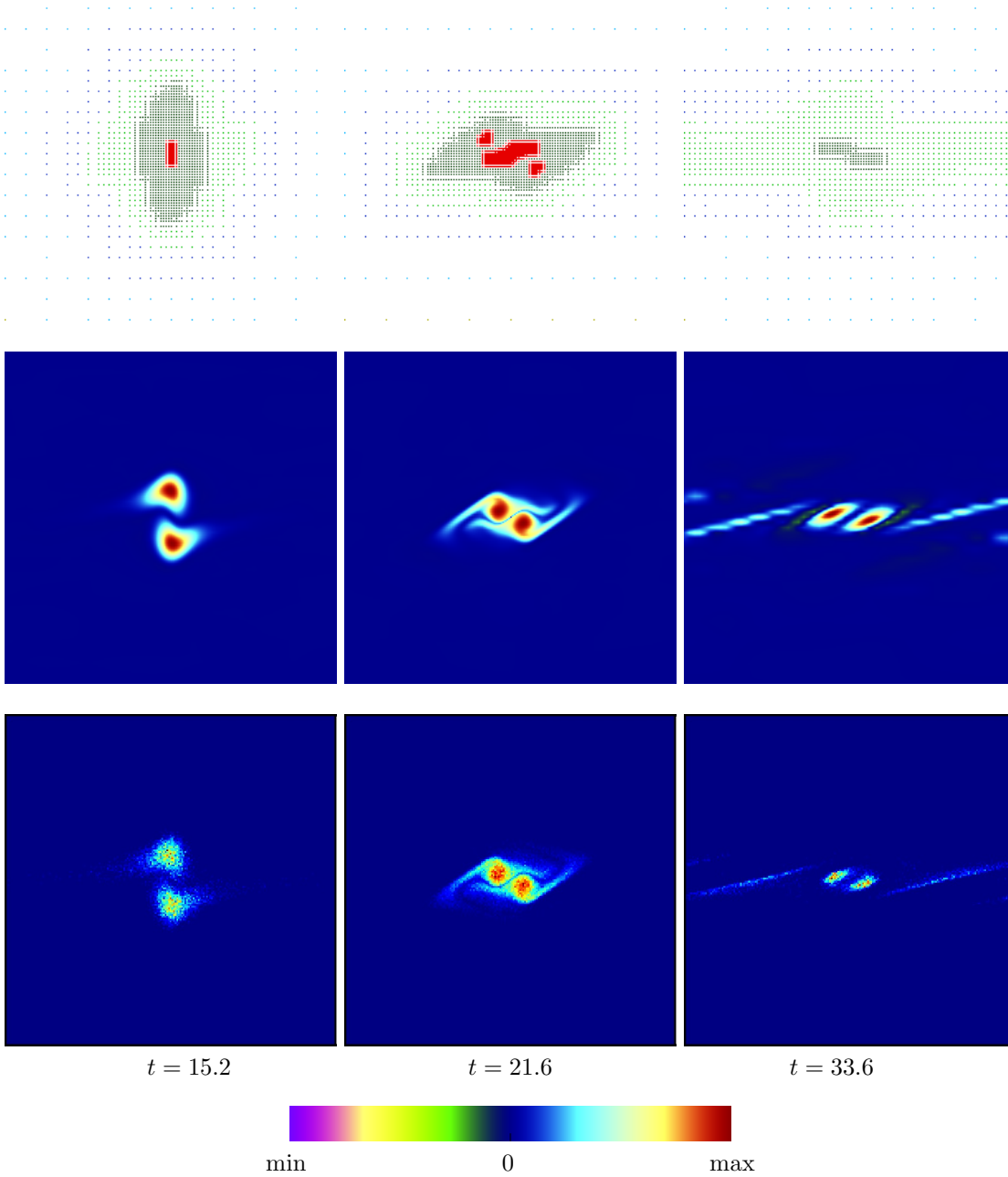


FIG. 21. *Cut in the (z,w) -direction of the six-dimensional phase space at times $t = 15.2$, $t = 21.6$ and $t = 33.6$. First row: cut of the adaptive grid. Second row: cut of the density function obtained with the hierarchical basis adaptive scheme. Third row: cut of a GADGET simulation.*

necessary to represent it in the hierarchical basis with a given accuracy (fixed by the value of the threshold ε). Similarly it is defined as the inverse of the accuracy (threshold ε) for a given number of points. Watching at the threshold evolution Fig. 24, we can distinguish five phases:

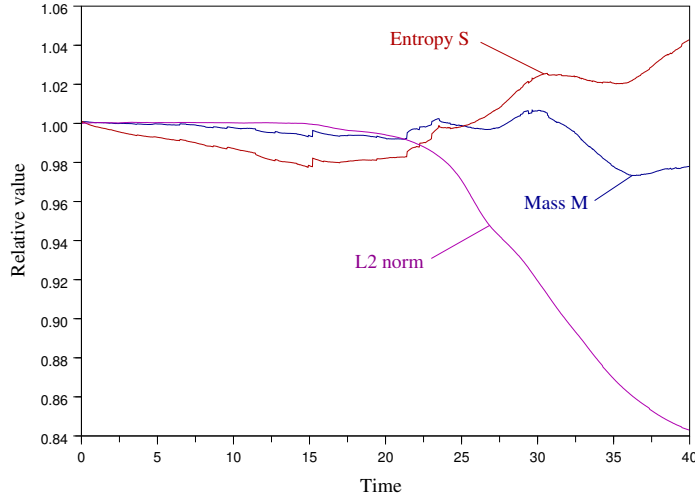


FIG. 22. Conservation analysis: plots of the mass, entropy and L^2 -norm.

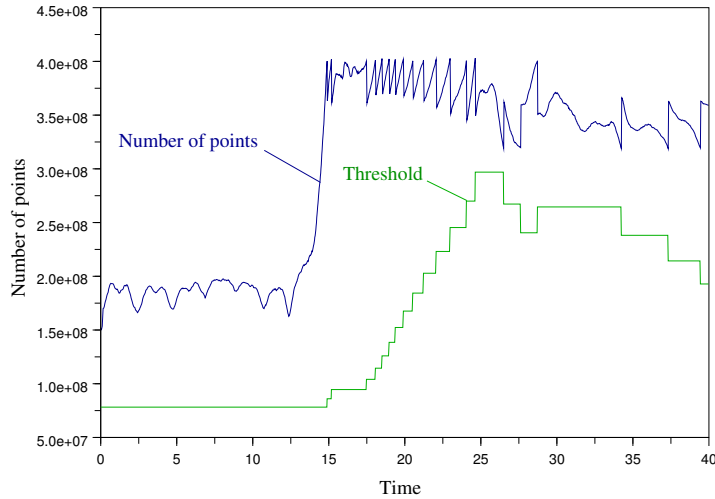
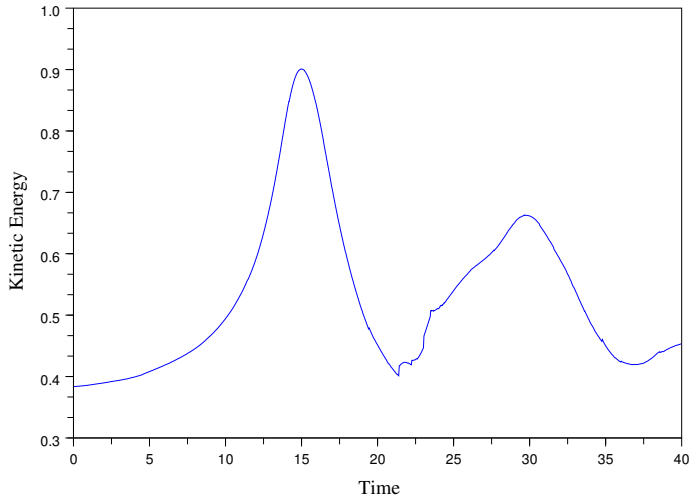


FIG. 23. Plots of the threshold ε (normalized scale) and of the number of points.

1. for $t \in [0, 13]$, the two spheres get close, the complexity stays steady,
2. for $t \in [13, 16]$, first collision, the complexity surges and the kinetic energy makes a jump,
3. for $t \in [16, 18]$, the spheres recede one from the other, the complexity stays steady,
4. for $t \in [18, 25]$, the spheres interact anew, creating a complex situation,
5. for $t \in [25, 40]$, damping, the solution simplifies itself, the threshold decreases to retain more and more small scales while these are dissipated by the numerical scheme at scales close to the local resolution of the grid.

The number of points peaks at a maximum of 400,000,000 triggering an increase of the threshold ε due to available memory. Then, for t greater than 30, it decreases to a minimum of 320,000,000 which makes the threshold ε decrease. Running this ex-

FIG. 24. Plot of the kinetic energy E_c .

periment took 1,390 time steps, and 249 wallclock hours of computation on a 2.66GHz Intel Xeon X5650 (12 threads and 48 Go RAM memory).

This experiment can be compared to the one presented in [51] where one can barely see what is going on given the coarseness of the grid and the diffusivity of the scheme. In this paper [51], two King spheres interact in a 64^6 uniform mesh ($\sim 6.87e+10$ points, i.e. 150 times what we are using) applying a Positive Flux Conservation (PFC) scheme [26] which is a kind of finite volume method, associated to a massive MPI/OpenMP hybrid parallelization on 1024 CPU cores on 64 nodes (i.e. 85 times the resources we are consuming).

5. Conclusion. This work shows the possibilities offered by the hierarchical basis adaptive scheme to solve the Vlasov equations. The HBA scheme provides a good accuracy that is out of reach of the traditional Eulerian schemes in six dimensions. Its basic simplicity –finite differences and polynomial interpolation– caused some defects that must be corrected: the mass and the energy are not conserved and under-resolved shock-like structures may deteriorate the solution.

This opens the way to further developments: to cope with the conservation trouble we can switch to Finite Volume schemes adapting the existing literature *e.g.* [47] to the six-dimensional constraint, or we will continue with Interpolet/Finite Differences enforcing the mass conservation by correcting the finite difference scheme. The shock and negative value problems may be resolved using limiters and weighted essentially non-oscillatory (WENO) schemes.

Other perspectives are to apply a semi-Lagrangian scheme [4] instead of the finite differences and to further reduce the number of degrees of freedom by using six-dimensional Hyperbolic Wavelets [22] whose use for dimensional reduction has been put into practice in the Sparse Grids [8]. It could be adapted to turbulent systems using the wavelets described in [20] that are intermediate between multiresolution analysis isotropic wavelets and hyperbolic anisotropic wavelets. It may be the best way to make the six-dimensional Vlasov solver a routine numerical tool.

Appendix A. Interpolets.

The first interpolet is $\varphi_2(x) = \max(1 - |x|, 0)$ which corresponds to the second-

order interpolation $f\left(\frac{1}{2}\right) = \frac{f(0)+f(1)}{2}$. This function satisfies the dyadic self-similarity relation:

$$(23) \quad \varphi_2\left(\frac{x}{2}\right) = \frac{1}{2}\varphi_2(x+1) + \varphi_2(x) + \frac{1}{2}\varphi_2(x-1).$$

The fourth-order interpolation

$$(24) \quad f\left(\frac{1}{2}\right) = \frac{-f(-1) + 9f(0) + 9f(1) - f(2)}{16},$$

provides a scaling function whose self-similarity relation is written:

$$(25) \quad \varphi_4\left(\frac{x}{2}\right) = -\frac{1}{16}\varphi_4(x+3) + \frac{9}{16}\varphi_4(x+1) + \varphi_4(x) + \frac{9}{16}\varphi_4(x-1) - \frac{1}{16}\varphi_4(x-3).$$

It is represented in Fig. 25. This function is also called the Deslauriers-Dubuc function [21].

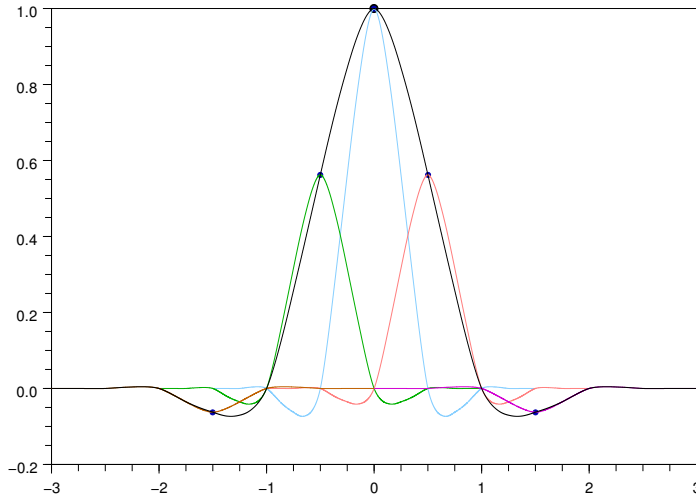


FIG. 25. The Deslauriers-Dubuc function and its self-similarity relation.

More generally the interpolant of order $2n$ is given by the self-similarity relation:

$$(26) \quad \varphi_{2n}\left(\frac{x}{2}\right) = \sum_{\ell} c_{\ell}\varphi_{2n}(x-l),$$

with the symmetry $c_{-\ell} = c_{\ell} \forall \ell$, the conditions $c_0 = 1$, $c_{2\ell} = 0$ for $\ell > 0$. The sequence $(c_{2\ell+1})_{0 \leq \ell \leq n-1}$ verifies a Vandermonde system of equations

$$(27) \quad \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 3^2 & \dots & (2n-1)^2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 3^{2n-2} & \dots & (2n-1)^{2n-2} \end{bmatrix} \begin{pmatrix} c_1 \\ c_3 \\ \vdots \\ c_{2n-1} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

whose solution is given by:

$$(28) \quad c_{2\ell+1} = \frac{1}{2} \prod_{j=0, j \neq \ell}^{n-1} \frac{(2j+1)^2}{(2j+1)^2 - (2\ell+1)^2} = \frac{n}{2^{4n-3}} C_{2n-1}^{n-1} \frac{(-1)^\ell}{2\ell+1} C_{2n-1}^{m+\ell}.$$

Appendix B. Finite differences and interpolation. The fourth-order interpolet refines with the classical fourth-order interpolation scheme:

$$(29) \quad \begin{aligned} I_4 f(x) &= \frac{-f(x-3h) + 9f(x-h) + 9f(x+h) - f(x+h)}{16} \\ &= f(x) + \frac{3}{8}h^4 f^{(4)}(x) + O(h^6). \end{aligned}$$

The choice of the finite difference schemes is more delicate since various instabilities may appear. The downwind instability manifests itself by the local appearance of bumps or gaps in the numerical solution. Here are the finite differences used in the scheme. Although probably far from optimal, their combination with the fourth order interpolation proved rather stable. To compute a differentiation, for a right to left wind, we apply the following finite differences:

- the fifth-order upwind finite difference:

$$(30) \quad \begin{aligned} Df(x) &= \frac{2f(x+3h) - 15f(x+2h) + 60f(x+h) - 20f(x) - 30f(x-h) + 3f(x-2h)}{60h} \\ &= f'(x) + \frac{h^5}{60}f^{(6)}(x) + O(h^6), \end{aligned}$$

- the third-order upwind finite difference:

$$(31) \quad \begin{aligned} Df(x) &= \frac{-f(x+2h) + 6f(x+h) - 3f(x) - 2f(x-h)}{6h} \\ &= f'(x) - \frac{h^3}{12}f^{(4)}(x) + O(h^4), \end{aligned}$$

- a third-order finite difference at the rear of the level (with respect to the wind)

$$(32) \quad \begin{aligned} Df(x) &= \frac{-3f(x+2h) + 16f(x+h) - 12f(x) - f(x-2h)}{12h} \\ &= f'(x) - \frac{h^3}{8}f^{(4)}(x) + O(h^4), \end{aligned}$$

- a third-order finite difference at the head of the refinement level

$$(33) \quad \begin{aligned} Df(x) &= \frac{-f(x+3h) + 18f(x+h) - 8f(x) - 9f(x-h)}{24h} \\ &= f'(x) - \frac{h^3}{8}f^{(4)}(x) + O(h^4), \end{aligned}$$

- a third-order finite difference at the head of the refinement level, using the differentiation computed at the coarser level:

$$(34) \quad Df(x) = \frac{5f(x+2h) - 4f(x) - f(x-2h)}{8h} - \frac{1}{2}D_{2h}f(x+2h)$$

$$= f'(x) - \frac{h^3}{3} f^{(4)}(x) + O(h^4).$$

This error estimate stands when the differentiation $D_{2h} f(x + 2h)$ is computed with a fifth-order finite difference.

If one of these schemes is not available because the point is near the boundary of the refinement level, we apply the next one. If none is available, we keep the fourth-order interpolation of the differentiation computed at the coarser level. This interpolate is computed by default at every point.

These error estimates stand for sufficiently smooth functions.

Appendix C. Data structure and C encoding. The encoding of tree structures usually relies either on hash-tables of keys referring directly to the positions of the nodes [9] either on real tree structures built with pointers. In Fortran, these pointers can be emulated thanks to tables of indices [50].

Using C programming, we chose to implement the whole fully-threaded tree structure. The basic element allocated in memory is the node. In dimension d , its building block is given by:

```

struct Node {
    double *tab;           /* table of maximally 2d points */
    struct Node *prt;     /* pointer to the parent node */
    struct Node **chd;    /* pointers to the children */
    struct noeud *vsn[2d]; /* 2d pointers to the neighbors */
    signed char frk;      /* filiation rank */
    unsigned long long flg; /* flag of point current activation */
    unsigned long long nwf; /* flag of point next activation */
    unsigned long long ldf; /* flag of point previous activation */
    unsigned long long chf; /* flag for the children presence */
    int ell;              /* parameter with different uses */
    short int dir;       /* direction of the wind */
    signed char num[2d]; /* renumbering of the points */
    signed char szp;     /* number of active points in the node */
};

```

An `unsigned long long` contains 64 bits, each bit corresponds to one of the 2^6 six-dimensional cube points. Thanks to this, it is possible to mark the point activation inside a node with a single `unsigned long long` variable.

The multiple nodes of a tree can be accessed through the parent–children connection (see Fig. 26 left). Then a table ordered level by level (Fig. 26 right) allows to make an Open-MP parallelization.

Acknowledgements. This work has been funded in part by ANR grant ANR-13-MONU-0003 and by the Eurofusion ER14 project EURATOM-CfP-WP14-ER-01/IPP-03. Erwan Deriaz acknowledges the help of the Maison de la Simulation Saclay and thanks the Institute of Mathematics Polska Akademia Nauk Warsaw for its two month hosting in Spring 2014.

He is grateful to Nicolas Besse who introduced him to Vlasov simulation, to Thierry Sousbie and Stéphane Colombi who provided the astrophysical test case and kindly participated to the correction of the manuscript, to Alain Ghizzo and Maxime Lesur who kindly provided the four-dimensional test case, and to Teresa Regińska, Siegfried Müller and Romain Teyssier for instructive discussions.

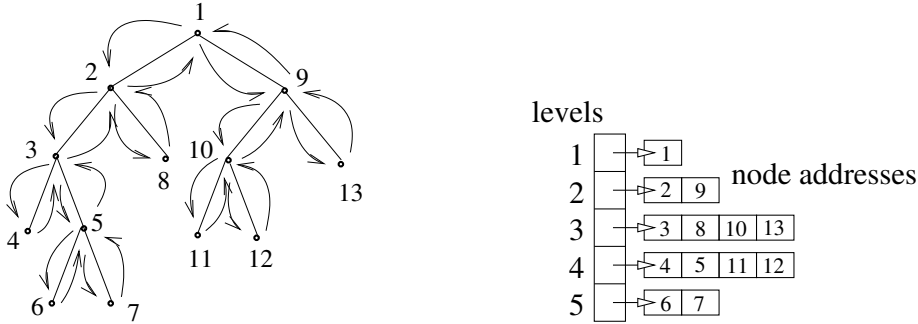


FIG. 26. Accessing the nodes in the tree structure. Left: graph transversal. Right: table of node addresses ordered level by level.

REFERENCES

- [1] T. ABEL, O. HAHN, AND R. KAEHLER, *Tracing the dark matter sheet in phase space*, MNRAS **427** 61–76, 2012.
- [2] ANTONOV, V. A., Vest. Leningrad Gos. Univ. **19**(96), 1962.
- [3] T.D. ARBER, R.G.L. VANN, *A Critical Comparison of Eulerian-Grid-Based Vlasov Solvers*, J. Comput. Phys. **180**(1) 339–357, 2002.
- [4] N. BESSE, E. DERIAZ, E. MADAULE, *Adaptive multiresolution semi-Lagrangian discontinuous Galerkin methods for the Vlasov equations*, to appear in J. Comput. Phys., <http://dx.doi.org/10.1016/j.jcp.2016.12.003>.
- [5] N. BESSE, G. LATU, A. GHIZZO, E. SONNENDRÜCKER, P. BERTRAND, *A wavelet-MRA-based adaptive semi-Lagrangian method for the relativistic Vlasov–Maxwell system*, J. Comput. Phys. **227**(16) 7889–7916, 2008.
- [6] JAMES BINNEY & SCOTT TREMAINE, *Galactic Dynamics*, book, Princeton Series in Astrophysics, 1987, Second Edition, 2011.
- [7] C.K. BIRDSALL, A.B. LANGDON, *Plasma Physics via Computer Simulation*, book, Taylor & Francis ed., reprint, 1991.
- [8] O. BOKANOWSKI, J. GARCKE, M. GRIEBEL, I. KLOMPMAKER, *An Adaptive Sparse Grid Semi-Lagrangian Scheme for First Order Hamilton-Jacobi Bellman Equations*, Journal of Scientific Computing **55**(3) 575–605, 2013.
- [9] K. BRIX, S. MELIAN, S. MÜLLER, M. BACHMANN, *Adaptive multi-resolution Methods: Practical issues on Data Structures, Implementation and Parallelization*, ESAIM Proc. **34** 151–183, V. Louvet, M. Massot (eds.), 2011.
- [10] J. BÜCHNER, *Vlasov-code simulation*, Proceedings of ISSS-7, 26-31 March, 2005.
- [11] C. BURSTEDDE, L. C. WILCOX, O. GHATTAS, *p4est: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees*, SIAM J. Sci. Comput. **33**(3) 1103–1133, 2011.
- [12] A. J. CHRISTLIEB, Y. LIU, Q. TANG, AND Z. XU, *Positivity-preserving finite difference weighted ENO schemes with constrained transport for ideal magnetohydrodynamic equations*, SIAM J. Sci. Comput. **37**(4) A1825–A1845, 2015.
- [13] A. COHEN, R. DEVORE, G. KERKYACHARIAN AND D. PICARD, *Maximal spaces with given rate of convergence for thresholding algorithms*, Appl. Comput. Harmon. Anal. **11** 167–191, 2001.
- [14] S. COLOMBI, T. SOUSBIE, S. PEIRANI, G. PLUM, Y. SUTO, *Vlasov versus N-body: the Hénon sphere*, Monthly Notices of the Royal Astronomical Society **450**(4) 3724–3741, 2015.
- [15] G.-H. COTTET, P.-A. RAVIART, *Particle methods for the one-dimensional Vlasov–Poisson equations*, SIAM J. Numer. Anal. **21** p. 52, 1984.
- [16] R. COURANT, K. FRIEDRICHS, H. LEWY, *On the Partial Difference Equations of Mathematical Physics*, IBM Journal, march 1967, translation from a paper originally appeared in *Mathematische Annalen* **100** 32–74, 1928.
- [17] P. DEGOND, F. DELUZET, L. NAVORET, A.-B. SUN, M.-H. VIGNAL, *Asymptotic-Preserving Particle-In-Cell method for the Vlasov–Poisson system near quasineutrality*, J. Comput. Phys. **229**(16) 5630–5652, 2010.
- [18] S. DELAGE SANTACREU, *Méthode de raffinement adaptatif hybride pour le suivi de fronts dans des écoulements incompressibles*, Thèse de Mécanique de l’Université Bordeaux I, 2006.

- [19] E. DERIAZ, *Stability conditions for the numerical solution of convection-dominated problems with skew-symmetric discretizations*, SIAM J. Numer. Anal. **50**(3) 1058–1085, 2012.
- [20] E. DERIAZ AND V. PERRIER, *Direct Numerical Simulation of turbulence using divergence-free wavelets*, SIAM Multiscale Model. Simul. **7**(3) 1101–1129, 2008.
- [21] G. DESLAURIERS, S. DUBUC, *Symmetric iterative interpolation processes*, Constructive Approximation **5**(1) 49–68, 1989.
- [22] R. A. DEVORE, S. V. KONYAGIN, AND V. N. TEMLYAKOV, *Hyperbolic Wavelet Approximation*, Constructive Approximation **14** 1–26, 1998.
- [23] M. DUMBSER, V. A. TITAREV, AND S. V. UTUZHNIKOV, *Implicit Multiblock Method for Solving a Kinetic Equation on Unstructured Meshes*, Comput. Math. Math. Phys. **53**(5) 601–615, Pleiades Publishing, Ltd., 2013.
- [24] B. ELIASSON, *Numerical Simulations of the Fourier-Transformed Vlasov-Maxwell System in Higher Dimensions—Theory and Applications*, Journal of Transport Theory and Statistical Physics **39**(5–7) 387–465, 2010.
- [25] E. FIJALKOW, *Behaviour of phase-space holes in 2D simulations*, Journal of Plasma Physics **61**(01) 65–76, 1999.
- [26] F. FILBET, E. SONNENDRÜCKER, P. BERTRAND, *Conservative numerical schemes for the Vlasov equation*, J. Comput. Phys. **172** 166–187, 2001.
- [27] F. FILBET, E. SONNENDRÜCKER, *Comparison of Eulerian Vlasov solvers*, Comput. Phys. Commun. **150** 247–266, 2003.
- [28] T. FUJIWARA, *Integration of the collisionless Boltzmann Equation for Spherical Stellar Systems*, Publ. Astron. Soc. Japan **35**, 547–558, 1983.
- [29] M. GRANDIN, *Data structures and algorithms for high-dimensional structured adaptive mesh refinement*, Advances in Engineering Software **82** 75–86, 2015.
- [30] W. GUO, Y. CHENG, *A Sparse Grid Discontinuous Galerkin Method for High-Dimensional Transport Equations and Its Application to Kinetic Simulations*, SIAM J. Sci. Comput. **38** (6) A3381–A3409, 2016.
- [31] ERNST HAIRER, SYVERT PAUL NØRSETT, GERHARD WANNER, *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer Series in Comput. Mathematics, Vol. 8, Springer-Verlag 1987, Second revised edition 1993.
- [32] V. GRANDGIRARD, M. BRUNETTI, P. BERTRAND, N. BESSE, X. GARBET, P. GHENDRIH, G. MANFREDI, Y. SARAZIN, O. SAUTER, E. SONNENDRÜCKER, J. VACLAVIK, L. VILLARD, *A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation*, J. Comput. Phys. **217**(2) 395–423, 2006.
- [33] A. HARTEN, *Multi-resolution algorithms for the numerical solution of hyperbolic conservation laws*, Comm. Pure and Applied Math. **48** 1305–1342, 1995.
- [34] F. HECHT, *New development in FreeFem++*, J. Numer. Math. **20**(3-4) 251–265, 2012.
- [35] J.A.F. HITTINGER, J.W. BANKS, *Block-structured adaptive mesh refinement algorithms for Vlasov simulation*, J. Comput. Phys. **241** 118–140, 2013.
- [36] O. HOENEN, E. VIOLARD, *Load-Balancing for a Block-Based Parallel Adaptive 4D Vlasov Solver*, Conference Paper, Euro-Par 2008 - Parallel Processing, 14th International Euro-Par Conference Proceedings, 2008.
- [37] M. HOLMSTRÖM, *Solving Hyperbolic PDEs Using Interpolating Wavelets*, SIAM J. Sci. Comput. **21**(2) 405–420, 1999.
- [38] KEVLAHAN, N.K.-R. AND VASILYEV, O.V., *An adaptive Wavelet Collocation Method for Fluid-Structure Interaction*, SIAM J. Sci. Comput. **26**(6) 1894–1915, 2005.
- [39] K. KORMANN, *A Semi-Lagrangian Vlasov Solver in Tensor Train Format*, SIAM J. Sci. Comput. **37**(4), B613–B632, 2015.
- [40] T. NAKAMURA, T. YABE, *Cubic interpolated propagation scheme for solving hyper-dimensional Vlasov-Poisson equation in phase space*, Comput. Phys. Commun. **120** 122–154, 1999.
- [41] I. V. OSELEDETS, *Tensor-Train Decomposition*, SIAM J. Sci. Comput. **33**(5) 2295–2317, 2011.
- [42] D. PFLÜGER, *Spatially Adaptive Sparse Grids for High-Dimensional Problems*, Ph.D. thesis of Technischen Universität München, 2010.
- [43] N. PHAM, P. HELLUY AND L. NAVORET, *Hyperbolic approximation of the Fourier transformed Vlasov equation*, ESAIM Proc. **45** 379–389, 2014.
- [44] S. POPINET, *Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries*, J. Comput. Phys. **190**(2) 572–600, 2003.
- [45] O. ROUSSEL, K. SCHNEIDER, A. TSIGULIN, H. BOCKHORN, *A conservative fully adaptive multi-resolution algorithm for parabolic PDEs*, J. Comput. Phys. **188**(2) 493–523, 2003.
- [46] K. SELIM, A. LOGG AND M. G. LARSON, *An Adaptive Finite Element Splitting Method for the Incompressible NavierStokes Equations*, Computer Methods in Applied Mechanics and Engineering **209–212** 54–65, 2011.

- [47] C. SHEN, J.-M. QIU, A. CHRISTLIEB, *Adaptive mesh refinement based on high order finite difference WENO scheme for multi-scale simulations*, J. Comput. Phys. **230**(10) 3780–3802, 2011.
- [48] E. SONNENDRÜCKER, J. ROCHE, P. BERTRAND, A. GHIZZO, *The Semi-Lagrangian Method for the Numerical Resolution of the Vlasov Equation*, J. Comput. Phys. **149**(2) 201–220, 1999.
- [49] V. SPRINGEL, *The cosmological simulation code GADGET-2*, MNRAS **364** 1105, 2005.
- [50] R. TEYSSIER, *Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES*, Astron. Astrophys. **385** 337–364, 2002.
- [51] YOSHIKAWA K., YOSHIDA, N., UMEMURA, M., *Direct Integration of the collisionless Boltzmann Equation in Six-dimensional Phase Space: Self-gravitating Systems*, The Astrophysical Journal **762** 116, 2013.
- [52] O. ZANOTTI, M. DUMBSER, *A high order special relativistic hydrodynamic and magnetohydrodynamic code with spacetime adaptive mesh refinement*, Comput. Phys. Commun. **188** 110–127, 2015.