Multiscale Model. Simul.
Vol. 16, No. 2, pp. 583–614

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 that retains only the most significant coefficients. This allows considerable savings in terms of computational time and memory usage. The proposed scheme involves the mathematical formalism of multiresolution analysis and computer implementation of adaptive mesh refinement. We apply a finite difference method to approximate the Vlasov–Poisson equations, although other numerical methods could be considered. Numerical experiments are presented for the d-dimensional Vlasov–Poisson equations in the full $2d$-dimensional phase space for $d = 1, 2$, or 3. The six-dimensional case is compared to a Gadget N-body simulation.

Key words. adaptive mesh refinement, hierarchical basis, wavelet, multiresolution analysis, finite differences, phase-space simulations, Vlasov–Poisson equations, astrophysics

AMS subject classifications. 65M05, 65M50, 85-08

DOI. 10.1137/16M1108649

1. Introduction.

The general idea of adaptivity in numerical analysis lies in the representation of a complex system by a reduced number of elements. It aims to decrease the computational time and memory resources needed to simulate the system. Often, the situation boils down to a trade-off between a numerical volume (computer memory, number of elementary operations, etc.) and an increase in the implementation complexity (data structures or algorithms).

The most common adaptive technique used in mechanics [24, 40, 45, 52, 53], physics [7, 42], and astrophysics [1, 59, 60] is called adaptive mesh refinement (AMR). In AMR, a Cartesian grid refines locally into finer Cartesian grids. Such refinement is often associated with finite volume methods [40, 52, 60]. We can distinguish two different approaches among AMR methods: in patch-based AMR the data structure is a collection of grids of varying sizes and accuracies, which are embedded within each other [42, 56], whereas in the fully threaded tree AMR the points or cells, or groups thereof, are stored in the nodes or leaves of a large tree structure [11, 36, 52, 53, 60]. We use the latter method.

In the present study, we are particularly interested in using adaptive methods to tackle the simulation of large systems. Most of these rely on reduced functional basis discretization. This implicates finite elements where the underlying mesh does not have to be Cartesian [41, 55], wavelets associated with a vast nonlinear approximation theory [17], and spectral methods that show a good numerical accuracy [31, 51]. The hierarchical bases [7] are special kinds of wavelet bases for which the primal scaling function and the primal wavelet are identical, although their duals remain different. A sparse grid method itself [7, 50] is a subclass of hierarchical bases. It is similar

*Received by the editors December 20, 2016; accepted for publication (in revised form) January 5, 2018; published electronically April 3, 2018.
http://www.siam.org/journals/mms/16-2/M110864.html
Funding: This work was partially funded jointly by ANR grant ANR-13-MONU-0003 and the Eurofusion ER14 project EURATOM-CfP-WP14-ER-01/IPP-03.
†Faculté des Sciences et Technologies, Institut Jean Lamour, 54506 Vandoeuvre-lès-Nancy, France (erwan.deriaz@univ-lorraine.fr).
‡Institut d'Astrophysique de Paris, 75014 Paris, France (peirani@iap.fr).

to the hyperbolic—anisotropic—hierarchical base. It allows approximation of high-dimensional problems for which there is no medium or high frequency oscillation [10, 37]. Tensorization consists of expressing a multidimensional function as the product of matrices of one-dimensional functions, i.e., one matrix per dimension [49]. This allows simulation of high-dimensional models such as the three-dimensional Vlasov–Poisson equations for problems with low effective dimensionality [47].

Passing from the classical two- and three-dimensional AMR to a six-dimensional one forces us to adopt an efficient implementation and apply simplified numerical methods. More specifically, we propose a hierarchical basis associated with an Eulerian discretization onto which we apply finite difference schemes. As recommended [34], we avoid dissipation by using a third-order scheme for the convection.

The method presented largely relies on classical AMR [40, 60, 52, 45, 62, 7, 56, 11, 15]. It is not intended to compete with present and efficient Vlasov solvers using Eulerian [4, 32, 33, 34, 61], semi-Lagrangian [57], or Lagrangian [9, 19, 22, 59] schemes for the lower two- and four-dimensional phase-space simulations. For these, AMRs have already been widely tested [7, 42, 56, 62]. However, the method used compares favorably with the existing six-dimensional Eulerian Vlasov–Poisson simulations [48, 61] and appears complementary to the widely used N-body simulations [59].

The Vlasov solvers provide noiseless, accurate, numerical solutions and are useful, for example, in magneto-sheath simulations for which the Vlasiator code is developed in six dimensions [2]. In [54] the authors of the Vlasiator code point out that six-dimensional Vlasov simulations would benefit from adaptivity.

More specifically, the reader who is interested in comparing the merits of the various Vlasov solvers can refer to [14] for a short and well-argued discussion on this topic. The AMR adaptivity demonstrates valuable assets when the solution develops well-localized singularities [60, 45].

In the community research into rarefied gases, the Boltzmann equation is also six-dimensional and implies a phase-space distribution function. This differs from the Vlasov equations by the addition of a collision operator. To tackle the huge numerical figures in this domain, adaptive techniques were developed. For example, an efficient one consists of discretizing the distribution function in a fixed physical grid where each point contains an adaptive local velocity grid [12, 13]. Kolobov and Arslanbekov also proposed an adaptive scheme called the AMAR scheme [46]. Here the AMR of the velocity space is associated with an automatic selection of the kinetic or continuum solvers. In [5] a tree of trees discretization was proposed. This corresponds to an AMR tree in physical space where each leaf contains an AMR tree of the velocity space. Our scheme differs from such approaches by the fact that both physical and velocity spaces are refined simultaneously in an AMR which discretizes the phase space as a whole.

This paper is organized as follows. In section 2 we describe the wavelet basis derived from polynomial interpolation and the AMR finite differences discretization. In section 3 we introduce and detail the adaptive process. Finally, in the numerical experiments section (section 4), we apply this hierarchical basis adaptive scheme to Vlasov–Poisson equation simulations in the full two-dimensional, four-dimensional, and six-dimensional phase spaces. These equations model collision-free plasmas as well as gravitational systems.

2. The hierarchical basis adaptive scheme.

2.1. Deslauriers–Dubuc wavelet expansion. We consider a function $f : \Omega \to \mathbb{R}$, $\mathbf{x} \mapsto f(\mathbf{x})$ with $\Omega = [0,1]^d$ and periodic boundary conditions. We assume

that we know the values of 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

$$\mathbf{x}_\lambda = (2^{-j}k_1, \dots, 2^{-j}k_d). \tag{1}$$

The set Λ resembles the indices of the retained points $(\mathbf{x}_\lambda)_{\lambda\in\Lambda}$. It is constructed according to the local behavior of f through a wavelet expansion which is thresholded respecting a graduation condition (see section 3.3). We denote $\Omega_\Lambda = \{\mathbf{x}_\lambda, \lambda \in \Lambda\}$. We find 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 separate the set Λ into sets Λ_j containing the indices of level j such that $\lambda \in \Lambda_j \implies \lambda = (j, k_1, \dots, k_d)$. We denote $\Omega_j = \{\mathbf{x}_\lambda, \lambda \in \Lambda_j\}$. Hence

$$\Omega_\Lambda = \bigcup_{j\geq 0} \Omega_j. \tag{2}$$

The set Ω_{j+1} is constructed depending on the set $\Omega'_j \subset \Omega_j$ of points 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 belongs to Ω_{j+1}.

As a result, for most of the interpolation processes, a point has all its ancestors present in the coarser levels. Thus, 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 the definition of a fully threaded tree.

Remark 2.1. For the sake of simplicity of the algorithms, we make all the sets Ω_j self-sufficient regarding basic computations: the interpolation and 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 $\Omega'_{j-1} = \Omega_{j-1} \cap \Omega_j$.

Actually, this construction is closely related to the multiresolution analysis formalism (see Appendix A). The function f whose point values are known at $(\mathbf{x}_\lambda)_{\lambda\in\Lambda}$ is uniquely decomposed into the interpolet hierarchical basis $(\Phi_\lambda)_{\lambda\in\tilde{\Lambda}}$ (see Appendix B for the definition of *interpolet*) with the set $\tilde{\Lambda}$ 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. Thus, we exclude the case where 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

$$(f(\mathbf{x}))_{\mathbf{x}\in\Omega_\Lambda} \qquad \Longleftrightarrow \qquad f = \sum_{\lambda\in\tilde{\Lambda}} d_\lambda \Phi_\lambda. \tag{3}$$

As is usual in multiresolution analysis, the multidimensional interpolet Φ_λ is defined as the tensorial product of one-dimensional interpolets $\varphi_{j,k_i}(x_i)$. For $\lambda = (j, k_1, \dots, k_d)$,

$$\Phi_\lambda(\mathbf{x}) = \varphi_{j,k_1}(x_1) \times \cdots \times \varphi_{j,k_d}(x_d), \tag{4}$$

with the usual 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 [27] and Appendix B for more details). The algorithm that transforms $(f(\mathbf{x}_\lambda))_\lambda$ into $(d_\lambda)_\lambda$, as well as its inverse, has linear complexity. It simply boils down to a question of computing the residuals of the interpolation of level j by level $j-1$ for all levels where $j \geq 1$.

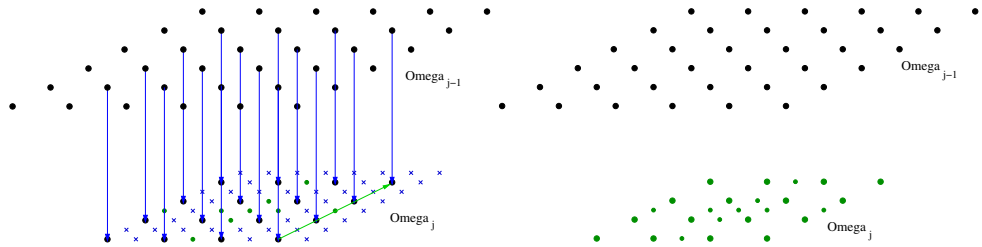

FIG. 1. *Refinement of a two-dimensional grid: the interpolation process is from the set Ω_{j-1} in black to the set Ω_j in green. The blue crosses represent all the possible points which are not activated because they cannot be interpolated from $\Omega_{j-1} \cap \Omega_j$.*

The expansion (3) allows a straightforward projection along any direction. For instance, let us assume that we want to integrate the two-dimensional function f,

$$f(x_1, x_2) = \sum_{j=0}^{j_{\max}} \sum_{k_1,k_2=0}^{2^j-1} d_{j,k_1,k_2} \varphi_{j,k_1}(x_1) \varphi_{j,k_2}(x_2), \tag{5}$$

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

$$\begin{aligned} F(x_2) &= \int_{x_1} f(x_1, x_2)\,dx_1 = \sum_{j=0}^{j_{\max}} \sum_{k_2=0}^{2^j-1} \left(\sum_{k_1=0}^{2^j-1} d_{j,k_1,k_2} \int_{x_1} \varphi_{j,k_1}(x_1)\,dx_1 \right) \varphi_{j,k_2}(x_2) \\ &= \sum_{j=0}^{j_{\max}} \sum_{k_2=0}^{2^j-1} 2^{-j} \left(\sum_{k_1=0}^{2^j-1} d_{j,k_1,k_2} \right) \varphi_{j,k_2}(x_2). \end{aligned} \tag{6}$$

This is particularly useful when solving the Vlasov–Poisson equation responsible for computing density; see (21) and (29) in the numerical experiments section (section 4).

2.2. Multidimensional interpolation. The sets (Ω_j) are linked by an interpolation relationship: all the points of Ω_j can be interpolated from the points of $\Omega_{j-1} \cap \Omega_j$. This relationship shapes the set (Ω_j), as can be seen in Figure 1.

The algorithm allocates a set of possible points: a node $(j, \mathbf{k})$ at level j with $\mathbf{k} = (k_1, \ldots, k_d) \in [0, 2^{j-1} - 1]^d$ 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. 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 using the interpolation algorithm. Then we carry on the computation only on active points.

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

$$I_4\, f(x) = \frac{-f(x - 3\,h) + 9\,f(x - h) + 9\,f(x + h) - f(x + 3\,h)}{16}. \tag{7}$$

In multidimensional 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 difference on the adaptive grid. We opted for finite difference because it is easy to implement. This choice also allows the scheme to reach the third-order of accuracy in any dimension for a sufficiently smooth solution at a linear price, irrespective of the number of dimensions. Meanwhile its accuracy and stability are comparable to classical two- and three-dimensional AMR finite volume methods [52, 45].

Nevertheless this method is not optimal for simulating the Vlasov–Poisson problem for which semi-Lagrangian methods perform much better [34]. In particular, it suffers from several drawbacks. First, the time step has to satisfy a Courant–Friedrichs–Lewy (CFL) condition (equation (10)). Second, one can note the appearance of spurious oscillations and nonpositive values in the distribution function. These are seen in the case of unsmooth solutions. These issues can be addressed thanks to WENO schemes and positivity preserving methods [16]. Third, in contrast to finite-volume methods, this method does not conserve the mass on nonuniform meshes.

To apply the finite difference scheme in the multilevel adaptive grid in Figure 1, we need to 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 the following operations recursively for j from 2 to $j_{\max}$:

- we transfer the differentiation values obtained at the points belonging to the $\Omega_{j-1} \cap \Omega_j$ intersection from Ω_{j-1} to Ω_j,
- we interpolate from these to 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 numerical experiments, we apply the differentiation formulae of Appendix C. When possible, we choose the fifth-order scheme as this comes with little cost (an increase of less than 20%) and improves both accuracy and conservation. The numerical errors resulting from the differentiation scheme of Appendix C are at least of the third order, as can be verified in the numerical experiments section, section 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 $\mathbf{v} : \mathbb{T}^d \to \mathbb{R}^d$:

$$\partial_t f(t, \mathbf{x}) + \mathbf{v}(\mathbf{x}) \cdot \nabla f(t, \mathbf{x}) = 0, \tag{8}$$

where $\nabla f = (\partial_1 f, \partial_2 f, \ldots, \partial_d f)$ and $\mathbf{v} \cdot \nabla f = \sum_i v_i \partial_i f$. To deal with any convection equation of the form (8) using the hierarchical basis adaptive scheme developed in section 2.1, we apply the third-order upwind finite difference scheme described in Appendix C. Particular care was given to the CFL condition that provides an upper limit to the time step.

We optimize the CFL condition, to benefit from the adaptability of the grid. We apply a fourth-order Runge–Kutta scheme in time coordinate [38] to (8). For finite difference schemes, on a uniform grid with a space step h, the CFL condition on the time step δt is given by

$$\delta t \leq C \frac{h}{\max_{\mathbf{x} \in \Omega} \|\mathbf{v}(\mathbf{x})\|_{\ell^1}} \tag{9}$$

with the constant C equal to approximately 1.73 in our case (from [25] and experiments carried out by ourselves) and $\|\mathbf{v}(\mathbf{x})\|_{\ell^1} = \sum_{i=1}^d |v_i(\mathbf{x})|$. 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 as already noticed [42] this allows mitigation of the CFL condition. The latter becomes

$$\delta t \leq C \min_{\mathbf{x}\in\Omega} \frac{h(\mathbf{x})}{\|\mathbf{v}(\mathbf{x})\|_{\ell^1}}. \tag{10}$$

This condition on the time step, written in phase-space coordinates, 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 $\|\mathbf{v}(\mathbf{x})\| \gg \|\mathbf{v}(\mathbf{0})\|$. Thus the time step δt remains large even when large values of $\max_{\mathbf{x}\in\Omega} \|\mathbf{v}(\mathbf{x})\|_{\ell^1}$ appear in the phase-space grid coordinates.

3. Adaptation.

3.1. Brief error analysis. Numerical errors appear under 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 precisely the case for Vlasov–Poisson equations if we begin at a C^∞ initial condition. Then with the fourth-order hierarchical basis, the interpolation error is of the order of $O(h^4)$. The finite differences described in section 2.3 provide a differentiation of the order of $O(h^3)$ and the fourth-order Runge–Kutta scheme is of the order of $O(\delta t^4)$, i.e., $O(h^4)$ in 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 C. For the various third-order finite differences, it is of the type $C\,h^3 f^{(4)}$.

3.2. Minimization of the nonlinear approximation error. The function f is decomposed into a basis (Φ_λ) in (3). We are looking for a criterion to reduce the number of coefficients d_λ. The most significant ones are kept and the others are taken to be equal to zero.

The AMR users apply criteria on low-order derivatives of f such as the absolute value of the function f, e.g., in [60] the AMR maps a mass distribution function and each cell contains the same quantity of mass; the gradient of f in Harten's discrete framework [40, 53]; or, more recently, the second derivatives of f described in [62]. Such criteria allow us to automatically and efficiently refine or unrefine the computational grid according to the local regularity of the function f, without too many theoretical considerations concerning the nonlinear approximation.

In our case, the overhead cost of the six-dimensional approximation incites us to follow more optimal criteria. It is known from the nonlinear approximation theory [17] that for $\Phi \in B^s_q(L^p)$ the hierarchical basis expansion (3) allows the superior bound to be computed:

$$\|f\|_{B^s_q(L^p)} = O\left(\left\|\left(2^{s\,j} 2^{-\frac{d\,j}{p}} \|(d_\lambda)_{\lambda\in\Omega_j}\|_{\ell_p}\right)_{j\geq 0}\right\|_{\ell_q}\right). \tag{11}$$

Minimizing the L^p-norm of the error on the sth derivative of f, the distribution function, is equivalent to minimizing the $B^s_p(L^p)$-norm of the residuals. 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:

$$|d_\lambda| \geq 2^{(-s+\frac{d}{p})j}\varepsilon. \tag{12}$$

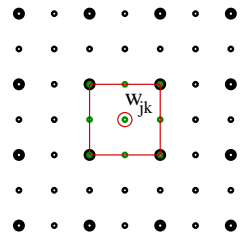

FIG. 2. *Elementary cube for the refinement.*

Contrary to a tensorial intuition, the exponent $\alpha = -s + \frac{d}{p}$ has the advantage of a flatter adaptive grid for a higher number of dimensions. In the future, we aim to minimize the L^p-norm of the error on the gradient of the distribution function ($s = 1$). Nevertheless, because of the interlinking of the levels Ω_j between themselves through the interpolation scheme, and because of the scarcity of the points in six dimensions, the $\alpha = -1$ exponent which theoretically minimizes the L^∞-norm of the error is not optimal in practice, as we will see in the numerical experiments shown later in Figure 7. We observe a kind of shift to a larger optimal α.

3.3. Refining and coarsening in practice. Let us assume we defined a value $\alpha \in \mathbb{R}$ that is relevant for the numerical experiment and an $\varepsilon > 0$ adapted to the available computer memory.

First, in a bottom-up algorithm—where j ranges from $j_{\max}$ to 2—we compute the residuals of the fourth-order interpolation of level Ω_j by level Ω_{j-1}. This is equivalent to a transformation in the isotropic fourth-order hierarchical basis (3).

Then we gather the residual $d_{j,\mathbf{k}'}$ values from (3), where the points $2^{-j}\mathbf{k}'$ inside the cubes are centered on points of the type $2^{-j}(2\ell_1 + 1, \ldots, 2\ell_d + 1)$ as indicated in Figure 2. For $\mathbf{k} = (2\ell_1 + 1, \ldots, 2\ell_d + 1)$, this yields a refinement weight associated to this cube:

$$w_{j,\mathbf{k}}^2 = \sum_{\mathbf{e}\in\{-1,0,+1\}^d} d_{j,\mathbf{k}+\mathbf{e}}^2. \tag{13}$$

Finally we apply the following refinement events in the following order:

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

Sometimes, we also fix a maximum level of refinement j_{Max} such that $j \leq j_{\text{Max}}$.

The larger α is, the closer the grid will be to a uniform one. Varying ε allows an increase or a decrease in the number of grid points to cope with the memory storage limitation of the computer (see the numerical experiments in section 4).

3.4. Adaptive numerical scheme for the Vlasov–Poisson equation. To simulate the Vlasov–Poisson equations (19), we apply a fourth-order Runge–Kutta

scheme in time and a third-order upwind finite difference in space. The refinement of the grid is applied once at every time step. Hence we iterate the following algorithm:

0) at time t_n the numerical approximation f_n of the distribution function $f(t_n, \mathbf{x}, \mathbf{v})$ is given on the adaptive grid Ω^n,
1) we project f_n onto the physical space $\mathbf{x}$ and we compute the electric field E_n,
2) we compute $k_1 = -\mathbf{v} \cdot \nabla_{\mathbf{x}} f_n - E_n \cdot \nabla_{\mathbf{v}} f_n$ with the finite difference scheme detailed in Appendix C,
3) we compute δt due to formula (10),
4) we carry out the first intermediate step of the Runge–Kutta scheme, $f_{(1)} = f_n + \frac{k_1}{2}\delta t$,
5) similarly to steps 2) and 4) we compute the next steps k_i and $f_{(i)}$ of the Runge–Kutta scheme,
6) we obtain the distribution function at time $t_{n+1} = t_n + \delta t$, that is, $f_{n+1} = f_n + \frac{1}{6}(k_1 + 2\,k_2 + 2\,k_3 + k_4)\delta t$,
7) finally, given f_{n+1} on Ω^n and the process detailed in section 3.3 we update the grid Ω^n to obtain Ω^{n+1}.

4. Numerical experiments. In this section we describe tests in various situations of the adaptive numerical scheme developed in this paper. First, as this method specifically targets six-dimensional simulations, we test the transport of a Gaussian function in a six-dimensional box. We compare the AMR results with those obtained with the help of a semi-Lagrangian scheme on a uniform grid. Then we check its properties on classical two-dimensional plasma test cases: the Landau damping and the bump-on-tail instability cases. The simulation of the two-stream instability in two dimensions (four-dimensional phase space) from [32] 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 six-dimensional Gaussian function. The numerical experiment consists of convecting a Gaussian function

$$f_0(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x_0}\|^2}{2\sigma^2}\right) \tag{14}$$

with $\mathbf{x_0} = (\frac{1}{2}, \ldots, \frac{1}{2})$ and the standard deviation of the Gaussian $\sigma = 0.1$ inside the six-dimensional box $\mathbb{T}^6 = [0,1]^6$ with periodic boundary conditions, for the velocity field:

$$\mathbf{v}(x_1, x_2, \ldots, x_6) = \begin{bmatrix} 1 \\ \cos(2\pi x_1 + 1) \\ \cos(2\pi x_1 + 2) \\ \vdots \\ \cos(2\pi x_1 + 5) \end{bmatrix}. \tag{15}$$

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

$$\begin{aligned} \int_{s=0}^{t} v_i(X(s))\, ds &= \int_{s=0}^{t} \cos(2\pi s + 2\pi X_1(0) + i - 1)\, ds \\ &= \left[\frac{1}{2\pi}\sin(2\pi s + 2\pi X_1(0) + i - 1)\right]_0^t \\ &= \frac{1}{2\pi}\left(\sin(2\pi(X_1(0) + t) + i - 1) - \sin(2\pi(X_1(0)) + i - 1)\right). \end{aligned} \tag{16}$$

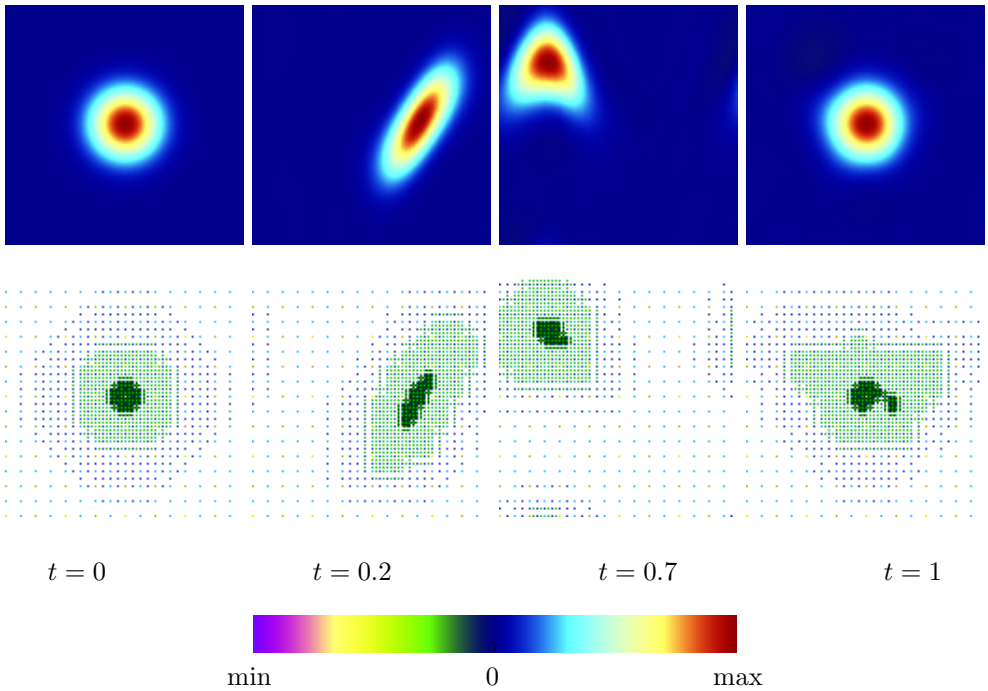

FIG. 3. *Projective view in the (x_1, x_2) plan of the six-dimensional convected Gaussian function at times $t = 0$, $t = 0.2$, $t = 0.7$, and $t = 1$. First row: the distribution function. Second row: the adaptive grid.*

Hence, the exact solution is given by

$$f(t, \mathbf{x}) = f_0(x_1 - t, x_2 - \frac{1}{2\pi}\left(\sin(2\pi x_1 + 1) - \sin(2\pi(x_1 - t) + 1)\right), \ldots \\ \ldots, x_6 - \frac{1}{2\pi}\left(\sin(2\pi x_1 + 5) - \sin(2\pi(x_1 - t) + 5)\right)) \tag{17}$$

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 GB RAM memory.

In Figure 3, we represent a two-dimensional projection onto (x_1, x_2) of both the distribution function and the adaptive grid. The maximum refinement level (in dark green) corresponds to a 128^6 uniform grid equivalent accuracy. The time step is taken as being constant and 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 $\|\mathbf{v}\|_{\ell^1} \approx 4.642$.

Let us assume that we want to minimize the L^∞ error on $f(t, \mathbf{x})$. The parameter α forces the threshold $2^{\alpha j}\varepsilon$ to depend on j. From section 3.2 its theoretical optimal value is given by $\alpha = -s + \frac{d}{p}$ to minimze the L^p-norm of the sth derivative in dimension d. Hence, for $s = 0$, $p = +\infty$, and $d = 6$, we obtain $\alpha = 0$. Since we need to compute the gradient of f to solve the convection equation, a part of the error comes from the first derivative. For $s = 1$, $d = 6$, and $\alpha = 0$ we find $p = 6$. So the choice $\alpha = 0$

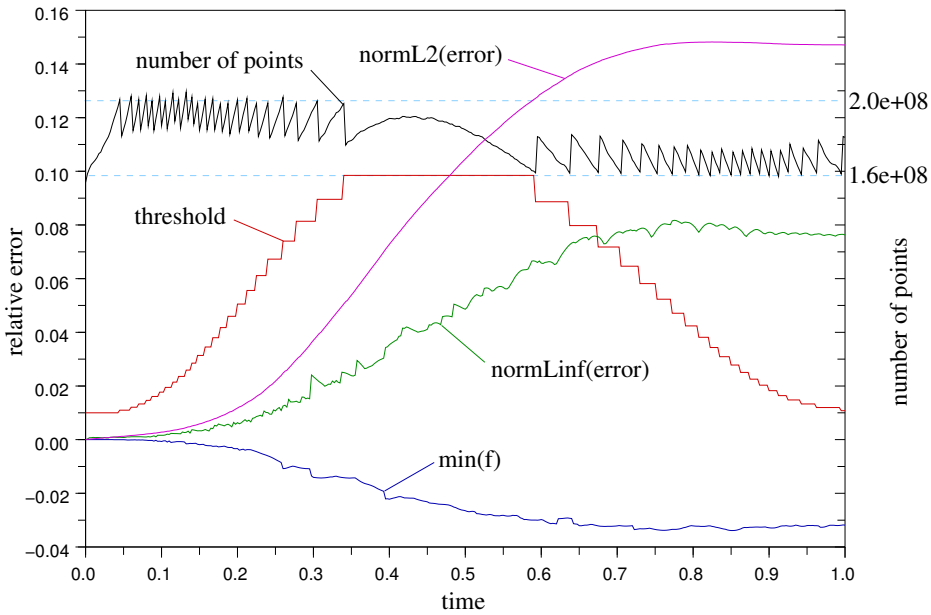

FIG. 4. *Error analysis and the number of points. The curve for the threshold ε was rescaled to fit the scale of the other curves. Relative errors ($\|f_{\text{num}} - f_{\text{exact}}\|/\|f_{\text{exact}}\|$) are given on the distribution function in L^2- and L^∞-norms.*

minimizes the L^6-norm of the gradient of f and, through the solution of the transport equation, the L^6-norm of f. The initial value of ε is fixed at $3e$–4.

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

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

The L^∞-norm of the error remains inferior to 8%, as shown in Figure 4. Relative error means the ratio between the norm of the error and the norm of the Gaussian function. 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 positivity), the threshold ε used to form the adaptive grid, and the number of points of the grid.

These results are compared with a semi-Lagrangian simulation of the same six-dimensional experiment (Figure 5). This simulation uses a Strang splitting and an eighth-order Lagrange polynomial interpolation scheme. With an OpenMP parallelization, the experiment took 6 hours for 100 time steps on the same 2.50 GHz Intel Xeon E5-2670 computer as for the AMR code. The 32 GB RAM memory only allowed a maximum of 39 points accuracy in each direction. This makes a total of $3.5e{+}9$ points. To compare with the $2e{+}8$ points of the AMR scheme, the L^∞ error is three times greater than for the AMR scheme and reaches 26%. The semi-Lagrangian schemes of lower order (fourth and sixth) have even larger errors.

In Figure 4 the L^2-norm of the error increases to twice the size of the L^∞-norm. This is due to the refinement criterion which minimizes the L^∞-norm of the error and not its L^2-norm. When $\alpha = 0$, the AMR scheme makes smaller errors on small scales with large L^∞-norms than on large scales with small L^∞-norms. This is in contrast

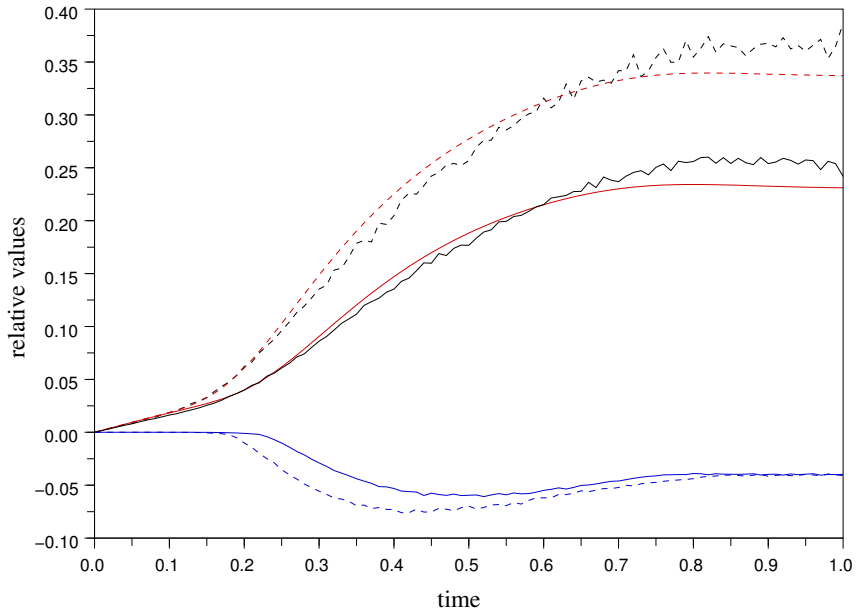

FIG. 5. *Error analysis of a semi-Lagrangian simulation with* 32 *points (dashed lines) and* 39 *points (plain lines) in each direction. The minimum of the negative values is in blue, the* L^∞ *error in black, and the* L^2 *error in red.*

to the semi-Lagrangian uniform grid situation shown in Figure 5 where both L^∞ and L^2-norms of the error evolve similarly.

The memory needed coincides with the number of points allocated (see Appendix D). So as not to exceed the memory limit and to optimize memory use, we bound the number of points with a maximum and a minimum materialized in Figure 4 by two horizontal light blue dashed lines. The maximum is fixed to $2e{+}8$ and the minimum to $1.6e{+}8$. Each time the number of points exceeds the maximum, the threshold ε is increased by 10%. This triggers a drop in the number of allocated points and memory required. It explains the sawtoothed aspect of the curve on which the number of points is plotted in Figure 4 for $t \in [0, 0.5]$. In contrast, when the number of points decreases, we can increase the accuracy by decreasing the threshold by 10%. This occurs in the second part of the graph for $t \in [0.5, 1]$ in Figure 4.

The error increases dramatically when the threshold ε is large. This coincides with a major stretching of the distribution function at $t = 0.5$ in Figure 4.

In Figure 6, we plot the mass, the L^2-norm, and the maximum of the distribution function against time. These curves show a precision of 6% for the maximum and 2% for the mass. The curves represent relative values compared to the initial one, so they are equal to 1 at time $t = 0$.

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

We apply the same experimental conditions to assess the convergence of the hierarchical basis adaptive scheme and estimate the optimal choice for the value of the parameter α involved in the refinement process. This is discussed in sections 3.2 and 3.3. We perform the experiment until $t = 0.1$ and compare the numerical result obtained with the exact solution. In Figure 7 we plot the error in L^∞-norm and L^2-norm for the numbers of points varying from approximatively $2e{+}6$ to $4e{+}8$.

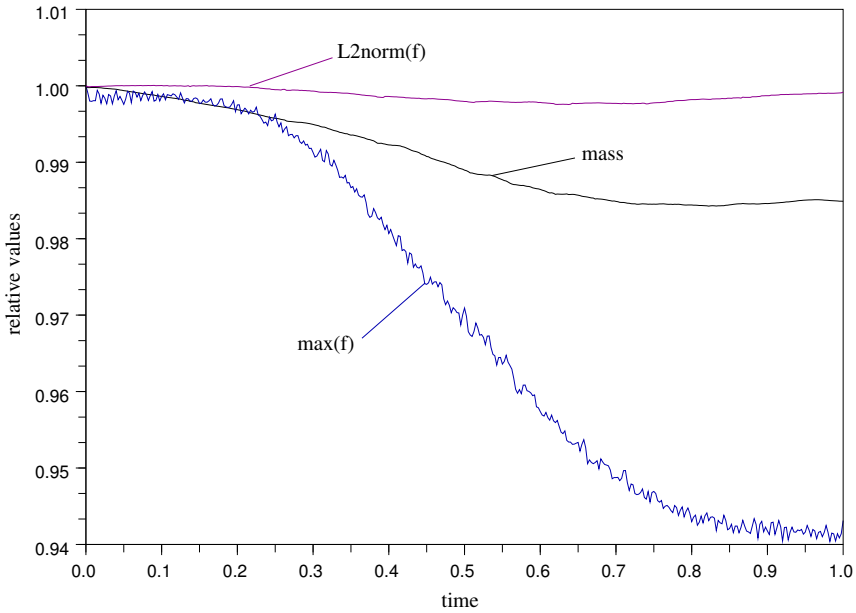

FIG. 6. *Conservation analysis.*

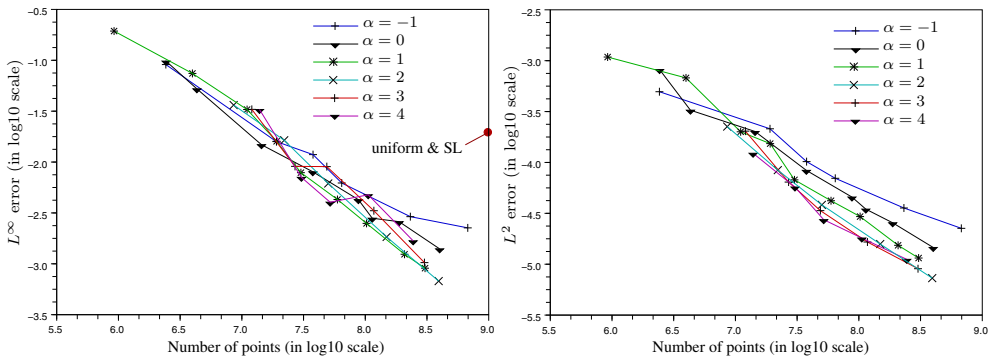

FIG. 7. *Error plots in L^∞- (left panel) and L^2- (right panel) norms for different values of the refinement criterion α. Both x and y axes have a* Log_{10} *scale. We also plot an instance of a semi-Lagrangian result on a uniform grid.*

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

$$e = O\left(\overline{h}^p\right) = O\left((N^{1/6})^{-p}\right) = O\left(\frac{1}{N^{p/6}}\right), \tag{18}$$

which leads to a convergence in $O(N^{-0.5})$.

Nevertheless, in Figure 7 we observe experimentally 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. This corresponds to $p = 5$ and a convergence rate of $e = O(N^{-0.83})$.

Although we would expect $\alpha = -1$ to be optimal for L^∞ approximations, it rather looks as though $\alpha = 2$ provides the best results. For 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 $\alpha = 2$. The experiments shown in Figures 3 and 6 would have been even more successful using $\alpha \in [1, 2]$ instead of $\alpha = 0$.

Since the error is concentrated at the boundary of the refinement levels, the activation of a new level triggers an increase in the L^∞ error. This explains the jumps in the L^∞-norm of the error in the convergence plot shown in Figure 7.

4.2. Application to plasma physics. In plasma physics and astrophysics, most of the Vlasov–Poisson simulations rely on particle-in-cell methods [19, 22], as these allow six-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 [34], but the curse of dimensionality [50] limits their application to at most four dimensions of the phase space [39].

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

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + E(t, \mathbf{x}) \cdot \nabla_{\mathbf{v}} f = 0 \tag{19}$$

with the electric force

$$E(t, \mathbf{x}) = \nabla_{\mathbf{x}} \phi(t, \mathbf{x}), \tag{20}$$

$$\Delta_{\mathbf{x}} \phi(t, \mathbf{x}) = \int_{\mathbf{v} \in \mathbb{R}^d} f(t, \mathbf{x}, \mathbf{v})\, d\mathbf{v} - \int_{\mathbf{x}, \mathbf{v} \in \mathbb{R}^d} f(t, \mathbf{x}, \mathbf{v})\, d\mathbf{v}\, d\mathbf{x}. \tag{21}$$

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

$$f_0(x, v) = \frac{1 - 0.05 \cos(0.5x)}{\sqrt{2\pi}} \exp(-v^2/2), \tag{22}$$

and we apply the hierarchical basis adaptive scheme for $t \in [0, 100]$. The refinement parameter α is taken as being equal to 1.7, while the grid (Figure 8) begins with 3,332 points and reaches a maximum of 45,522 at $t = 57.08$.

We integrate f in (21) due to its hierarchical basis expansion (equation (3)). Hence we sum the coefficients in the velocity direction as indicated in section 2.1 (equation (6)). Then (21) and (20) are solved using one-dimensional Fourier transforms.

The electric field (Figure 9) reproduces its classical curve in the Landau damping with the correct damping rate.

Bump-on-tail instability. Here we simulate the bump-on-tail instability with the hierarchical basis adaptive scheme. In [42], the authors tested a block-structured AMR using a high-order finite volume scheme. We obtain results similar to those described in that paper.

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

$$f_0(x, v) = \left(\frac{0.9}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} + \frac{0.2}{\sqrt{2\pi}} e^{-4(v-4.5)^2} \right) (1 + 0.04 \cos(0.3\, x)) \,. \tag{23}$$

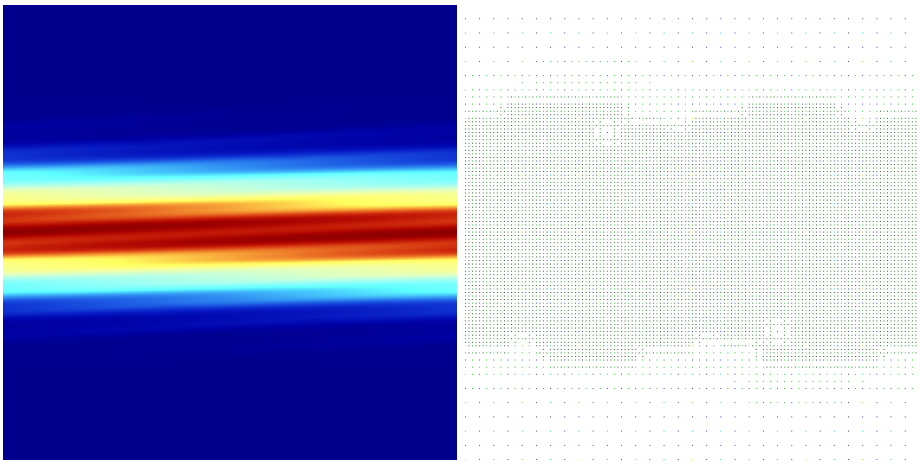

FIG. 8. *The distribution function (left panel) and the adaptive grid (right panel) at $t = 22$ during the Landau damping experiment.*

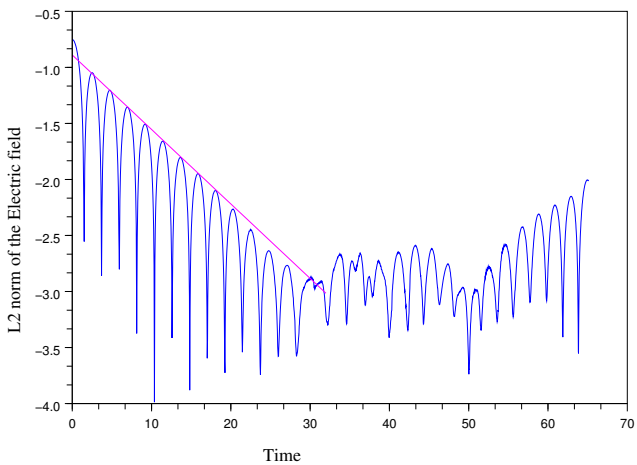

FIG. 9. *Landau damping of the electric field.*

The simulation is plotted in Figure 10 and shows a very good agreement with the results presented in [42]. We observe that the fine mesh shown in purple, equivalent to a 1024^2 uniform mesh, follows the filaments accurately. If the reader is interested in following precisely the behavior and shape of the filaments, the AMR method may compete with uniform grid methods even in one dimension.

In Figure 11 we demonstrate that the number of points varies from 17,000 at the beginning to 150,000 at the maximum complexity. The refinement parameters were fixed at $\alpha = 1.7$ and $\varepsilon = 7.18e\text{–}9$.

Figure 12 illustrates major defects of the proposed AMR method for the simulation of the Vlasov–Poisson equations. It does not conserve the mass and it makes negative values appear.

Figure 13 shows that although the kinetic energy and the potential energy tend to compensate each other, the total energy drifts away from its initial value.

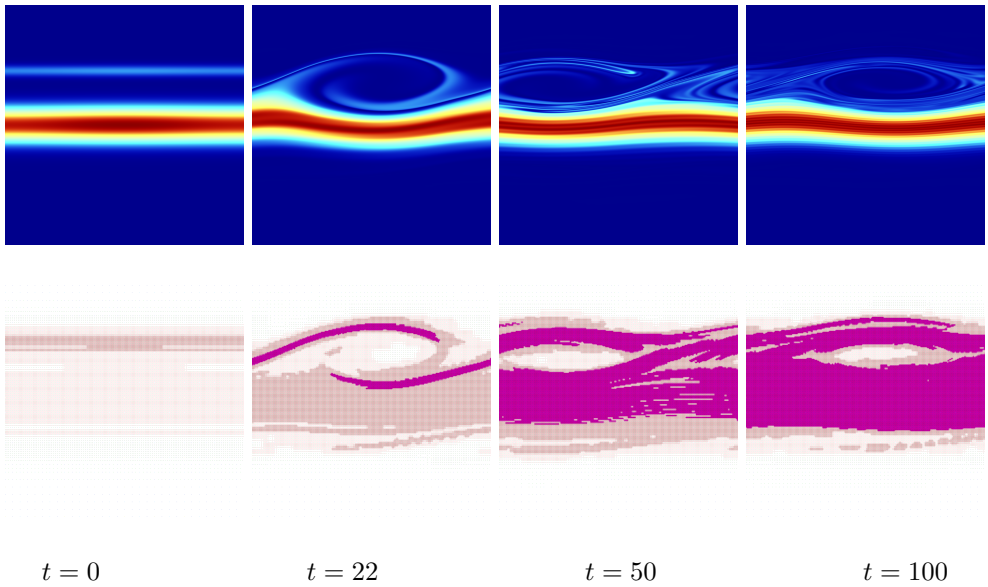

FIG. 10. *Bump-on-tail instability (1D). First row: the distribution function. Second row: the adaptive grid.*

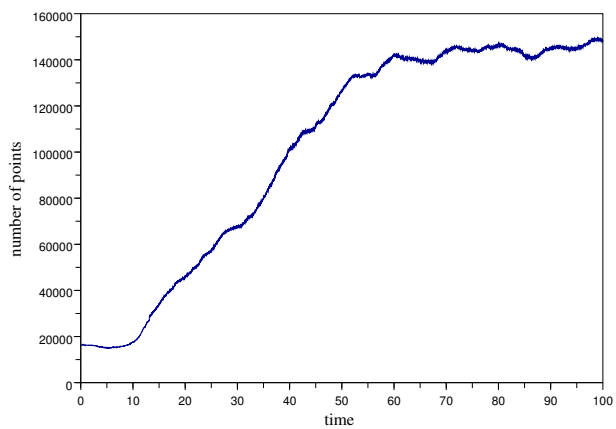

FIG. 11. *Evolution of the number of grid points in the AMR.*

The case of four-dimensional phase space: The two-beam instability. We apply the hierarchical basis adaptive scheme to a two-dimensional Vlasov–Poisson test case: the Fijalkow two-beam instability. Although this experiment does not fully correspond to the general initial conditions presented in [32], we refer to this author because he has extensively simulated the two-dimensional behavior of phase-space holes. This is a rather novel result concerning AMR simulation of Vlasov–Poisson equations, since authors of AMR Vlasov–Poisson numerical simulations usually restrict their experiments to one-dimensional cases [42, 56] with an exception in [43].

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:

$$f_0(x, y, u, v) = \frac{7}{4\,\pi}\,\exp\left(-\frac{u^2 + 4\,v^2}{8}\right)\sin^2\left(\frac{u}{3}\right)(1 + 0.05\,\cos(0.3\,x))\,. \tag{24}$$

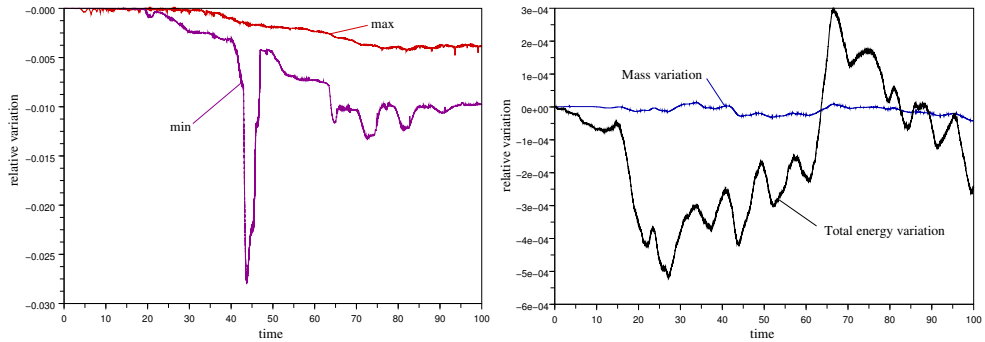

FIG. 12. *Plots of some theoretically conserved quantities. Left: minimum and maximum values of the distribution function. Right: mass and total energy. Representation in relative variation* $(\Delta f/f)$ *except for the min* $(\min(f)/\max(f_0))$.

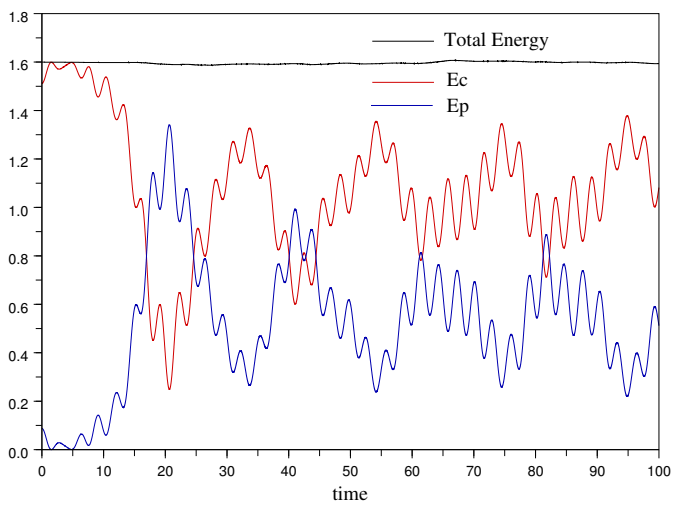

FIG. 13. *Variation of the kinetic (Ec) and potential (Ep) energies. The kinetic energy and the total energy were vertically shifted by* -23.

This initial distribution is represented in Figure 14. It was chosen to maximize the two-stream instability. 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.

In Figure 15 we represent the numerical solution at $t = 12$ when instability begins to develop. The maximum refinement goes up to 256^4 (red points in Figure 15, right). At this moment, the number of active points increases to $53e{+}6$, while the threshold ε stays constant at $1.59e{-}6$ (see Figure 16).

We take $60e{+}6$ as the maximum number of points. When the nonlinear regime develops after $t = 12$, steep fronts appear throughout the domain in the (x, y) direction. This results in a large increase in the number of points and then of the threshold ε (Figure 16). When the threshold becomes too large, the conserved quantities begin to vary by a few percent (Figure 17) and the energy drops (Figure 18).

Here, we measure the inability of the present finite difference AMR scheme to capture steep filaments distributed homogeneously in the spatial domain (x, y). It would

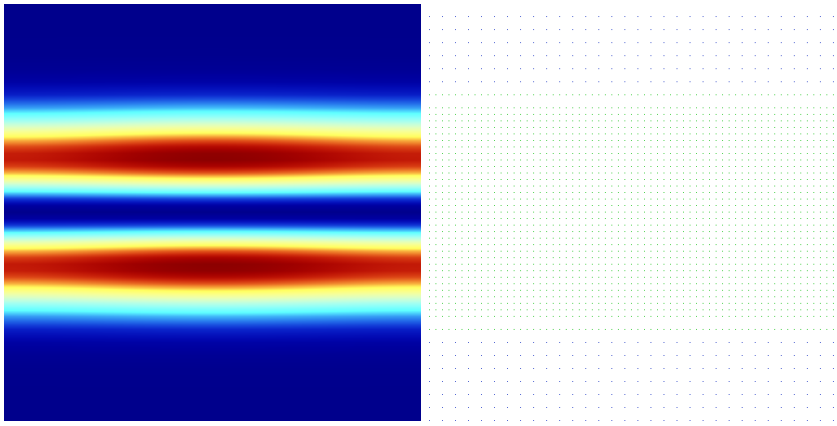

FIG. 14. *Phase space view of the initial condition, equation* (24). *Cuts of the distribution function and of the adaptive grid along* (x, u) *at* $y = v = 0$ *are shown.*

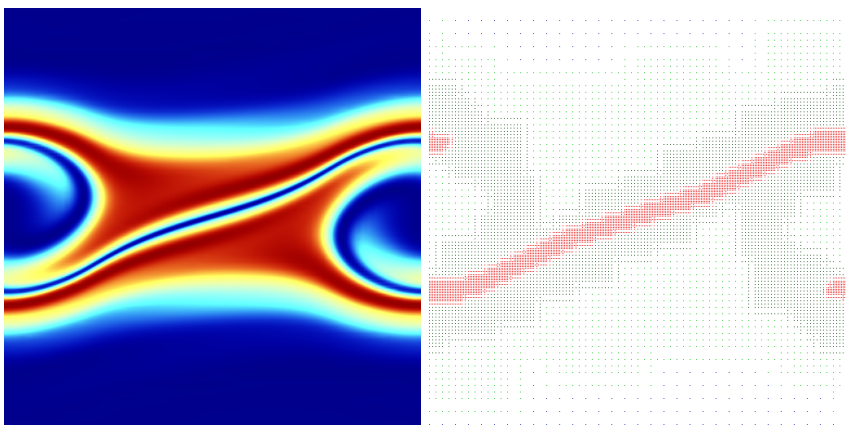

FIG. 15. *Cut along* (x, u) *of the distribution function at* $y = v = 0$ *in the phase space at time* $t = 12$.

probably work better if we refine only in the velocity (u, v) direction as suggested in [46, 5, 12, 13].

4.3. Six-dimensional astrophysics case: The merging of two dark matter halos. 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. This latter case allows the adaptive scheme to refine both velocity variables and space variables (see Figures 19 and 22). As a result 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 $\mathbf{v} = (u, v, w)$,

$$f : \begin{array}{ccc} \mathbb{R} \times \mathbb{R}^3 \times \mathbb{R}^3 & \to & \mathbb{R}^+, \\ t, \mathbf{x}, \mathbf{v} & \mapsto & f(t, \mathbf{x}, \mathbf{v}), \end{array} \tag{25}$$

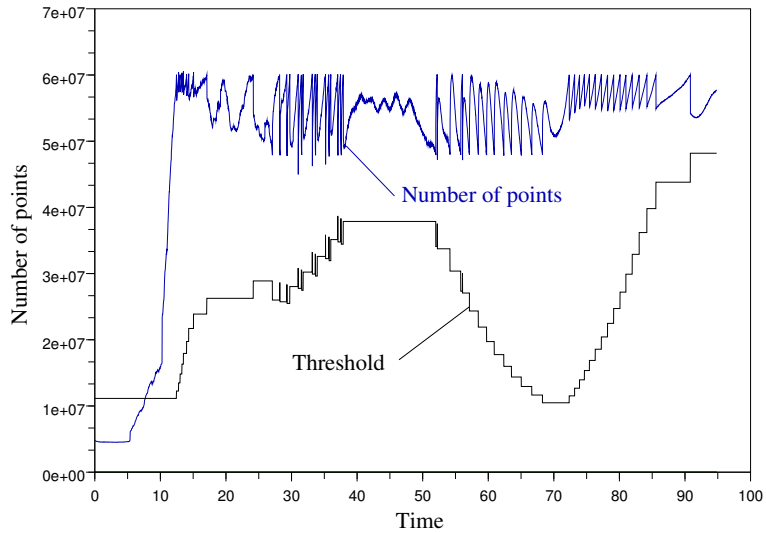

FIG. 16. *Number of points and threshold plotted against time. The threshold curve is linearly scaled to fit the window.*

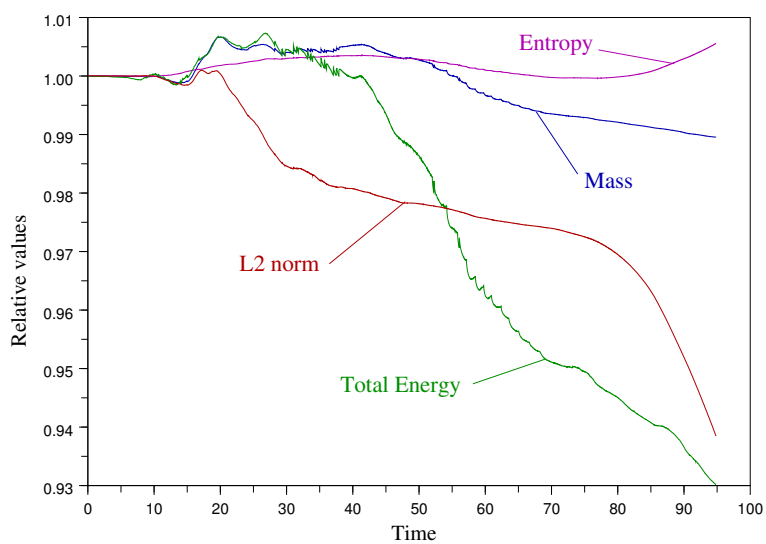

FIG. 17. *Plot of theoretically conserved quantities mass, L^2-norm, entropy, and total energy. Relative values with respect to the initial quantities are given.*

and its motion is modeled by the gravitational Vlasov–Poisson equations:

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + E(t, \mathbf{x}) \cdot \nabla_{\mathbf{v}} f = 0, \tag{26}$$

$$E(t, \mathbf{x}) = -\nabla_{\mathbf{x}} \phi(t, \mathbf{x}), \tag{27}$$

$$\Delta_{\mathbf{x}} \phi(t, \mathbf{x}) = 4\pi G \rho(t, \mathbf{x}) \tag{28}$$

in which the density function ρ is given by

$$\rho(t, \mathbf{x}) = \int_{\mathbb{R}^3} f(t, \mathbf{x}, \mathbf{v})\, d\mathbf{v}. \tag{29}$$

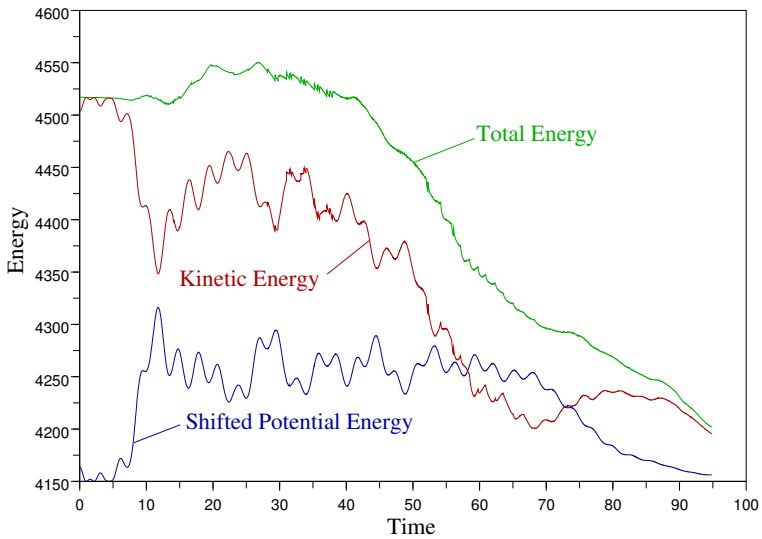

FIG. 18. *Plot of the energies. The potential energy is shifted by* +4150.

In comparison with plasma equations (20) and (21), we note, in particular, the sign "–" in the force equation (27) which switches from repulsive to attractive, and the factor "4π" (not eliminated when the equations are nondimensionalized in astrophysics) in (28).

These equations admit stationary solutions [3, 8]. One of them, the Plummer model, was successfully tested by Fujiwara in [35]. He used the symmetries of the problem in a two-dimensional plus one invariant simulation. It is given by the following distribution function:

$$f(r, \|\mathbf{v}\|) = \frac{3M}{7\pi^3 a^3}\left(2\left(1+\left(\frac{r}{a}\right)^2\right)^{-1/2} - \|\mathbf{v}\|^2\right)^{7/2} \quad \text{if} \quad 2\left(1+\left(\frac{r}{a}\right)^2\right)^{-1/2} - \|\mathbf{v}\|^2 \geq 0 \tag{30}$$

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

$$\phi(r) = -\frac{GM}{a}\left(1+\left(\frac{r}{a}\right)^2\right)^{-1/2} \tag{31}$$

and the density function by

$$\rho(r) = \frac{3M}{4\pi a^3}\left(1+\left(\frac{r}{a}\right)^2\right)^{-5/2}. \tag{32}$$

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

This distribution function (30) is compactly supported in velocity but not in space.

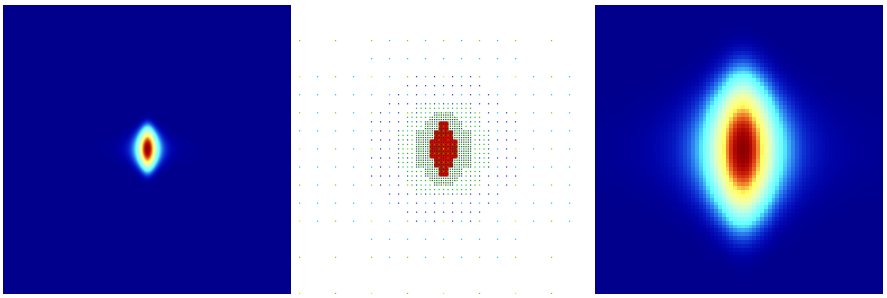

FIG. 19. *The Plummer model. Left: two-dimensional cut at zero* $(0,0,0,0,0,0)$ *in* (x,u)*-view of the Plummer model. Center: cut of the corresponding adaptive grid. Right: zoom at the center.*

In Figure 19, 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).

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 those obtained with GADGET, an N-body code extensively used in the literature [59]. Only the tree code part of this "tree PM" algorithm is employed. A total of 500 million particles and 10,932 time steps 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 used the same values as those in [18]. Running this GADGET simulation took 205.83 hours on 512 threads at the Center for Computational Astrophysics of the National Astronomical Observatory of Japan.

Concerning the AMR simulation, we solve the convection due to the scheme described in section 2.4 with a fourth-order finite element hierarchical basis built from polynomial interpolation (see Appendix B for more details) in expansion equation (3). The gravitational forces are computed in the three-dimensional Fourier space on a uniform grid of the smallest scale (i.e., with an FFT in a 512^3 grid).

This FFT is of low cost compared to the six-dimensional grid computations. It involves $2^{9\times 3} = 2^{27} \sim 128e{+}6$ points, which is smaller by a factor 40 than the $5e{+}9$ points of the six-dimensional AMR grid. Nevertheless we plan to use a high-order multigrid Poisson solver on the three-dimensional projected AMR grid in the near future. A two-dimensional version of this Poisson solver has already been presented in [23]. The three-dimensional version will be based on the HOC stencil [58].

The time step δt is computed at the beginning of each Runge–Kutta cycle by the CFL formula (10) 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 section 3.3 with $\alpha = 1.5$ and $\varepsilon = 3.128e$–7 at $t = 0$ and we prescribe the maximum number of points to be $5e{+}9$. 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 arbitrarily. The choice of $\varepsilon = 3.128e$–7 implies the activation of 451,686,490 points at $t = 0$. This is a compromise between the speed of the computations and the accuracy at the beginning of the simulation.

The initial conditions are sketched in Figure 20. Each sphere represents a stationary solution to equations (26–29) explicited in formula (30). The size of the box

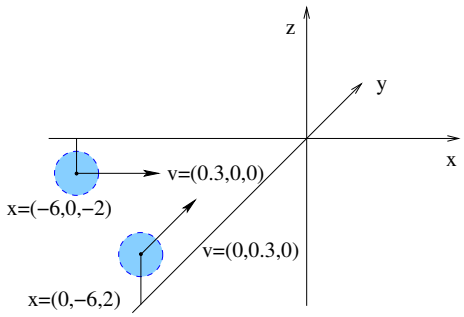

FIG. 20. *Initial conditions for the interaction of two Plummer models.*

is $[-12,12]^3 \times [-8,8]^3$. We apply periodic boundary conditions. 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 Figure 3.

In Figure 21, we see the collision and merging of the two spheres in a three-dimensional (x, y, z) projection.

Figure 22 represents cuts along the (z, w) plan of the six-dimensional phase space at $x = y = u = v = 0$. It only gives insight into the complexity of the six-dimensional solution. Figure 22, 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 probabilistic distribution of the particles inside the pixels which follows a Poisson distribution [59]. For $\lambda \in (0, +\infty)$ the average value of f in a given pixel, the probability of having $k \in \mathbb{N}$ particles in this pixel is given by

$$\mathbb{P}(k) = \frac{e^{-\lambda}\lambda^k}{k!}. \tag{33}$$

The comparison with the GADGET simulation validates the results obtained with the hierarchical basis adaptive scheme simulation in Figure 22.

In Figure 23, 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 $\|f\|_2(t) = (\int_{\mathbf{x},\mathbf{v}} f(\mathbf{x}, \mathbf{v}, t)^2\, d\mathbf{x}\, d\mathbf{v})^{\frac{1}{2}}$,
- 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 Vlasov–Poisson equations.

In Figure 25 we also plot

- the kinetic energy

$$E_c(t) = \int_{\mathbf{x},\mathbf{v}} \frac{\|\mathbf{v}\|^2}{2} f(\mathbf{x}, \mathbf{v}, t)\, d\mathbf{x}\, d\mathbf{v}. \tag{34}$$

The mass is not conserved and undergoes a decrease of 1.8% at $t = 40$. The entropy increases reasonably, by 3.0%, and this increase can be explained by the usual numerical averaging of the filaments [59].

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

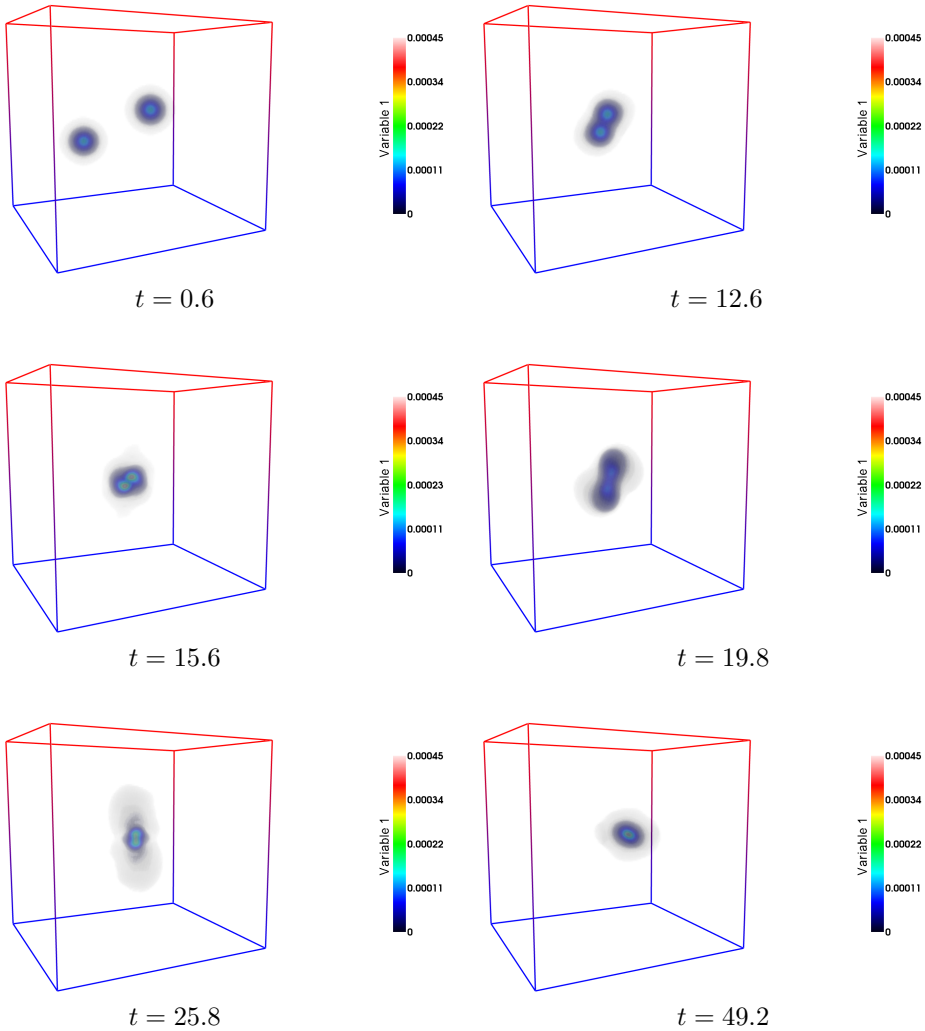

FIG. 21. *Three-dimensional view in the physical space* (x, y, z) *of the collision of two Plummer spheres at times* $t = 0.6, 12.6, 15.6, 19.8, 25.8, 49.2$. *This visualization was obtained for* $t \in [0, 60]$ *with the hierarchical basis adaptive scheme of second order.*

(the threshold ε) for a given number of points. With respect to the evolution of the threshold Figure 24, we can distinguish five phases:

1. for $t \in [0, 13]$, the two spheres grow close, and the complexity stays steady;
2. for $t \in [13, 16]$, at the 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, and the complexity only increases slowly;
4. for $t \in [18, 30]$, the spheres interact again, which engenders filamentation phenomena;
5. for $t \in [30, 40]$, damping simplifies the solution; the number of points decreases; small scales are dissipated by the numerical scheme at scales close to the local resolution of the grid.

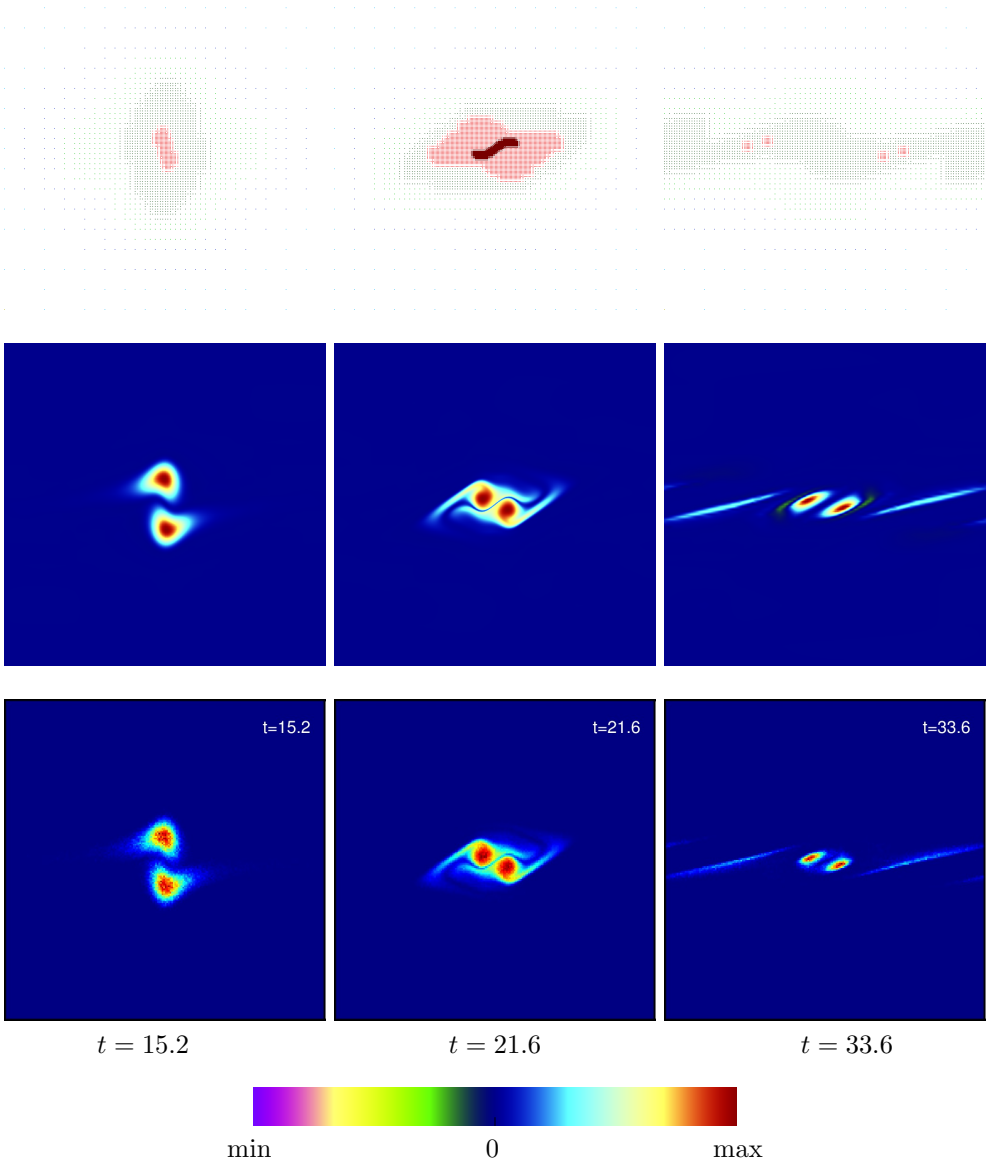

FIG. 22. *Cut in the (z, w)-direction of the six-dimensional phase space at times $t = 15.2$, $t = 21.6$ and $t = 33.6$. First row: cut of the adaptive grid. Second row: cut of the density function obtained with the hierarchical basis adaptive scheme. Third row: cut of a* GADGET *simulation.*

The number of points peaks at a maximum of 5,000,000,000 points notwithstanding a small incident at $t = 20$ without consequences. This triggers an increase of the threshold ε due to memory limitations. Then, for a value of t greater than 30, it starts to decrease. Running this experiment took 2,103 time steps and about 3,000 wall-clock hours (i.e., about 120 days) of computation on a node with 32 threads and 512 GB RAM on the Horizon cluster at the Institut d'Astrophysique de Paris. This corresponds to almost 100,000 total CPU hours, i.e., very close to the 105,385 total CPU hours of the GADGET simulation.

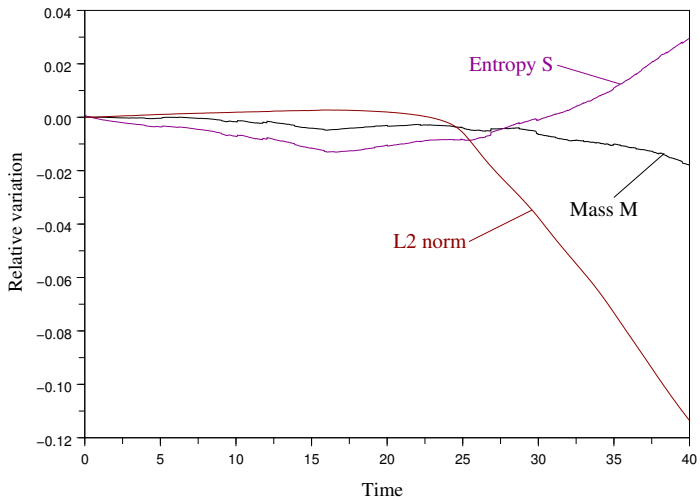

FIG. 23. *Conservation analysis: plots of the mass, entropy and L^2-norm.*

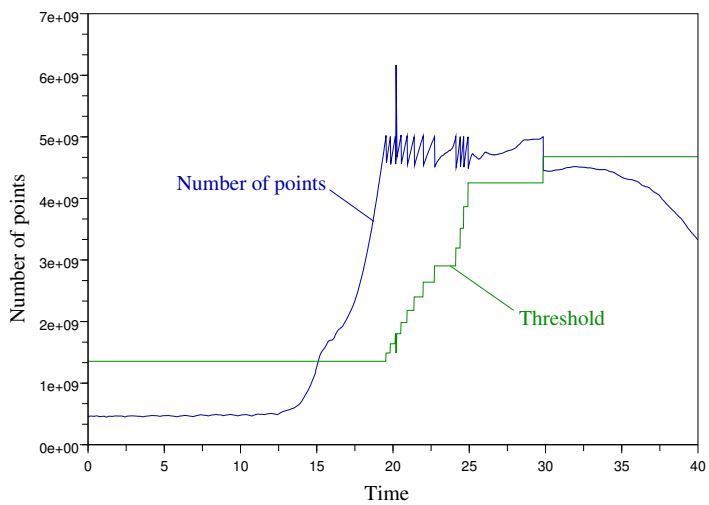

FIG. 24. *Plots of the threshold ε (normalized scale) and the number of points.*

It is complicated to compare these results with other Eulerian simulations since the maximum accuracy of a six-dimensional uniform grid with double accuracy on a node with 512 GB RAM is $(512 \times 2^{10\times 3}/8)^{1/6} = 64$ points in each direction instead of the 512 of the AMR scheme. The comparison on the Gaussian transport case shows that the AMR approach is particularly interesting regarding the L^∞ error, rather than for computing macroscopic quantities such as the kinetic energy (Figure 25).

Yoshikawa, Yoshida, and Umemura made an Eulerian simulation of a six-dimensional phase space astrophysics case [61]. In this latter reference, two King spheres interact in a 64^6 uniform mesh ($\sim 6.87e{+}10$ points, applying a positive flux conservation scheme [33] which is a kind of finite volume method, associated to a massive MPI/OpenMP hybrid parallelization on 1,024 CPU cores on 64 nodes. Contrary to the AMR scheme in Figure 22, the coarseness of the grid in [61] prevents any observation of fine details.

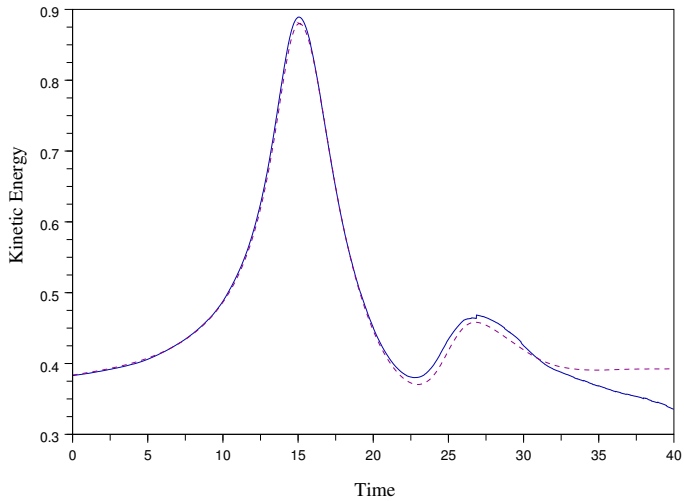

FIG. 25. *Plot of the kinetic energy E_c obtained with the AMR code (solid line) compared to the one obtained in the* GADGET *simulation (dashed line).*

5. Conclusion. This work shows the possibilities offered by the hierarchical basis adaptive scheme to solve the Vlasov equations. This scheme provides a good L^∞ accuracy that is out of reach of 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 underresolved shock-like structures may deteriorate the solution.

This technique opens the way to further developments: to cope with the conservation difficulty we can switch to finite volume schemes and adapt the AMR method presented in [56] to the six-dimensional constraint, or we can continue with interpolet/finite differences and enforce the mass conservation by correcting the finite difference scheme. The shock and negative value problems may be resolved by the use of limiters and WENO schemes.

Other perspectives are to apply a semi-Lagrangian scheme [6] instead of the finite differences and to further reduce the number of degrees of freedom with six-dimensional hyperbolic wavelets [28] whose use for dimensional reduction has been put into practice in sparse grids [10]. It could be adapted to turbulent systems due to the wavelets described in [26] 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. Multiresolution analysis. In this appendix, first we reintroduce briefly the definition of the wavelet bases as it is given in reference books [21]. Then we introduce multidimensional multiresolutions built on hierarchical bases.

DEFINITION A.1 (multiresolution analysis). *A multiresolution analysis of $L^2(\mathbb{R})$ corresponds to the existence of a sequence of closed subspaces $(V_j)_{j\in\mathbb{Z}}$ such that*

1. *$\forall j$, $V_j \subset V_{j+1}$, $\bigcap_{j\in\mathbb{Z}} V_j = 0$, $\bigcup_{j\in\mathbb{Z}} V_j$ is dense in $L^2(\mathbb{R})$,*
2. *(self-similarity in scale) $f \in V_j \iff f(2\cdot) \in V_{j+1}$,*
3. *(self-similarity in time) there exists a function $\varphi \in V_0$ such that the set $\{\varphi(\cdot - k); k \in \mathbb{Z}\}$ forms a Riesz basis of V_0.*

Such a function $\varphi : \mathbb{R} \to \mathbb{R}$ is called the *scaling function* of the multiresolution analysis. The parameter j defines the refinement level. The self-similarity in scale implies that each space V_j is generated by the set $\{\varphi(2^j \cdot -k); k \in \mathbb{Z}\}$. We denote $\varphi_{j,k}(x) = \varphi(2^j x - k)$.

The wavelets are defined as generators of a supplementary space W_j:

$$V_{j+1} = V_j \oplus W_j, \tag{35}$$

where the sum is direct but not necessarily orthogonal. If the spaces (W_j) are self-similar in scale, it is possible to define a function $\psi : \mathbb{R} \to \mathbb{R}$ called *wavelet* such that the sets $\{\psi(2^j \cdot -k); k \in \mathbb{Z}\}$ generate the spaces W_j.

Notice that with no further condition on the function ψ we can take $\psi(x) = \varphi(2x - 1)$. This choice corresponds to a *hierarchical basis*.

If we iterate the decomposition of V_j we obtain

$$V_j = V_0 \bigoplus_{\ell=0}^{j-1} W_\ell. \tag{36}$$

This entails the wavelet decomposition of $L^2(\mathbb{R})$:

$$L^2(\mathbb{R}) = V_0 \bigoplus_{\ell=0}^{+\infty} W_\ell. \tag{37}$$

Hence any function $f \in L^2(\mathbb{R})$ admits a unique decomposition onto the basis functions $\{\varphi_{0,k}, \psi_{j,k}; j \geq 0, k \in \mathbb{Z}\}$:

$$f = \sum_{k\in\mathbb{Z}} c_k \varphi_{0,k} + \sum_{j\geq 0} \sum_{k\in\mathbb{Z}} d_{j,k} \psi_{j,k}. \tag{38}$$

The multiresolution analyses of $L^2(\mathbb{R}^d)$ defined by the relation $\mathbb{V}_{j+1} = \mathbb{V}_j \oplus \mathbb{W}_j$ derive from the relation

$$\underbrace{(V_{j+1} \otimes \cdots \otimes V_{j+1})}_{\mathbb{V}_{j+1}} = \underbrace{(V_j \otimes \cdots \otimes V_j)}_{\mathbb{V}_j} \oplus \underbrace{\bigoplus_{\epsilon\in\{0,1\}^d\setminus\{(0,\ldots,0)\}} \left(Z_j^{\epsilon_1} \otimes \cdots \otimes Z_j^{\epsilon_d}\right)}_{\mathbb{W}_j} \tag{39}$$

with $Z^0 = V$ and $Z^1 = W$.

We can use the fact that $V_{j+1} = W_j^{(0)} \oplus W_j^{(1)}$ with $W_j^{(1)} = \langle \{\varphi(2^{j+1} \cdot -2k-1)\}_{k\in\mathbb{Z}} \rangle$ and $W_j^{(0)} = \langle \{\varphi(2^{j+1} \cdot -2k)\}_{k\in\mathbb{Z}} \rangle$ to write

$$\mathbb{W}_j \subset \bigoplus_{\epsilon\in\{0,1\}^d} \left(W_j^{(\epsilon_1)} \otimes \cdots \otimes W_j^{(\epsilon_d)}\right). \tag{40}$$

If we consider a hierarchical basis, so $\psi = \varphi(2 \cdot -1)$, then $W_j^{(1)} = W_j$, hence we are able to redefine $\mathbb{W}_j$ as

$$\mathbb{W}_j = \bigoplus_{\epsilon\in\{0,1\}^d\setminus\{(0,\ldots,0)\}} \left(W_j^{(\epsilon_1)} \otimes \cdots \otimes W_j^{(\epsilon_d)}\right). \tag{41}$$

Appendix B. Interpolets. The name *interpolet*—short for interpolating wavelet—refers both to the scaling function and the wavelet of a multiresolution analysis linked to the Lagrange interpolation. The scaling function satisfies

$$\varphi(0) = 1 \quad \text{and} \quad \forall k \in \mathbb{Z}^*,\ \varphi(k) = 0. \tag{42}$$

The interpolating wavelets were introduced in [29].

The first interpolet is $\varphi_2(x) = \max(1 - |x|, 0)$, which corresponds to the second-order interpolation $f(\frac{1}{2}) = \frac{f(0)+f(1)}{2}$. This function satisfies the dyadic self-similarity relationship:

$$\varphi_2\left(\frac{x}{2}\right) = \frac{1}{2}\varphi_2(x+1) + \varphi_2(x) + \frac{1}{2}\varphi_2(x-1). \tag{43}$$

The fourth-order interpolation

$$f\left(\frac{1}{2}\right) = \frac{-f(-1) + 9\,f(0) + 9\,f(1) - f(2)}{16} \tag{44}$$

provides a scale function whose self-similarity relation is written

$$\varphi_4\left(\frac{x}{2}\right) = -\frac{1}{16}\varphi_4(x+3) + \frac{9}{16}\varphi_4(x+1) + \varphi_4(x) + \frac{9}{16}\varphi_4(x-1) - \frac{1}{16}\varphi_4(x-3). \tag{45}$$

This is represented in Figure 26. This function is also called the Deslauriers–Dubuc function [27].

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

$$\varphi_{2n}\left(\frac{x}{2}\right) = \sum_{\ell} c_\ell \varphi_{2n}(x - l), \tag{46}$$

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

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 3^2 & \dots & (2n-1)^2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 3^{2n-2} & \dots & (2n-1)^{2n-2} \end{bmatrix} \begin{pmatrix} c_1 \\ c_3 \\ \vdots \\ c_{2n-1} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{47}$$

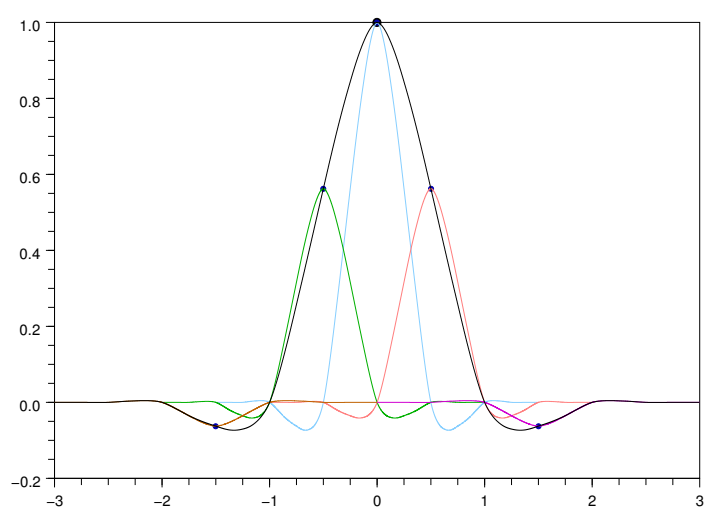

FIG. 26. *The Deslauriers–Dubuc function and its self-similarity relation.*

where the solution is given by

$$c_{2\ell+1} = \frac{1}{2}\prod_{j=0,j\neq\ell}^{n-1}\frac{(2j+1)^2}{(2j+1)^2-(2\ell+1)^2} = \frac{n}{2^{4n-3}}\binom{2n-1}{n-1}\frac{(-1)^\ell}{2\ell+1}\binom{2n-1}{n+\ell}. \tag{48}$$

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

$$\begin{aligned} I_4\, f(x) &= \frac{-f(x-3\,h)+9\,f(x-h)+9\,f(x+h)-f(x+3\,h)}{16} \\ &= f(x)-\frac{3}{8}h^4 f^{(4)}(x)+O(h^6). \end{aligned} \tag{49}$$

The choice of the finite difference schemes is 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 this paper. 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,

$$\begin{aligned} &Df(x) \\ &= \frac{2\,f(x+3\,h)-15\,f(x+\ h)+60\,f(x+h)-20\,f(x)-30\,f(x-h)+3\,f(x-2\,h)}{60\,h} \\ &= f'(x)+\frac{h^5}{60}f^{(6)}(x)+O(h^6), \end{aligned} \tag{50}$$

- the third-order upwind finite difference,

$$\begin{aligned} Df(x) &= \frac{-f(x+2\,h)+6\,f(x+h)-3\,f(x)-2\,f(x-h)}{6\,h} \\ &= f'(x)-\frac{h^3}{12}f^{(4)}(x)+O(h^4), \end{aligned} \tag{51}$$

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

$$\begin{aligned} Df(x) &= \frac{-3\,f(x+2\,h)+16\,f(x+h)-12\,f(x)-f(x-2h)}{12\,h} \\ &= f'(x)-\frac{h^3}{8}f^{(4)}(x)+O(h^4), \end{aligned} \tag{52}$$

- a third-order finite difference at the inward boundary of the refinement level,

$$\begin{aligned} Df(x) &= \frac{-f(x+3\,h)+18\,f(x+h)-8\,f(x)-9\,f(x-h)}{24\,h} \\ &= f'(x)-\frac{h^3}{8}f^{(4)}(x)+O(h^4) \end{aligned} \tag{53}$$

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

$$
\begin{aligned}
Df(x) &= \frac{5\,f(x+2\,h) - 4\,f(x) - f(x-2\,h)}{8\,h} - \frac{1}{2} D_{2h}\, f(x+2\,h) \\
&= f'(x) - \frac{h^3}{3} f^{(4)}(x) + O(h^4).
\end{aligned} \tag{54}
$$

This error estimate stands when the differentiation $D_{2h}\, f(x+2\,h)$ 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 interpolation is computed by default at every point.

These error estimates stand for sufficiently smooth functions.

Appendix D. Data structure and C encoding. The encoding of tree structures usually relies either on hash-tables of keys which refer directly to the positions of the nodes [11] or on real tree structures built with pointers. In Fortran, these pointers can be emulated due to tables of indices [60].

We chose to implement the whole fully threaded tree structure in a program written in C. 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 2^d 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[2^d];     /* renumbering of the points              */
 signed char szp;          /* number of active points in the node    */
};
```

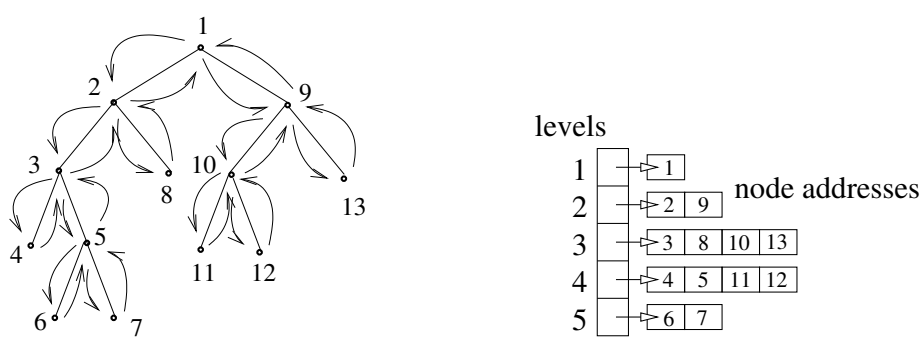

FIG. 27. *Accessing the nodes in the tree structure. Left: graph transversal. Right: table of node addresses ordered level by level.*

An `unsigned long long` contains 64 bits, and each bit corresponds to one of the $2^6 = 64$ six-dimensional cube points. Hence, 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 Figure 27, left). Then a table ordered level by level (Figure 27, right) allows us to make an Open-MP parallelization.

Acknowledgments. The first author is grateful for the assistance from Saclay Maison de la Simulation. He is also thankful to the Institute of Mathematics Polska Akademia Nauk Warsaw for hosting him for two months in the spring of 2014. He would like to extend his thanks and appreciation to Nicolas Besse for the introduction to the Vlasov simulation; to Thierry Sousbie and Stéphane Colombi for the astrophysical test case; to Alain Ghizzo and Maxime Lesur for the four-dimensional test case; and to Teresa Regińska, Siegfried Müller, and Romain Teyssier for informative/educational discussions. Finally, the authors are grateful to Stéphane Colombi and Gaëlle Deriaz for their help in improving the English of the paper.

REFERENCES

[1] T. Abel, O. Hahn, and R. Kaehler, *Tracing the dark matter sheet in phase space*, MNRAS, 427 (2012), pp. 61–76.

[2] S. von Alfthan, D. Pokhotelov, Y. Kempf, S. Hoilijoki, I. Honkonen, A. Sandroos, and M. Palmroth, *Vlasiator: First global hybrid-Vlasov simulations of Earth's foreshock and magnetosheath*, J. Atmospheric Solar-Terrestrial Phys., 120 (2014), pp. 24–35.

[3] V. A. Antonov, *Solution to the problem concerning the stability of a stellar system with Emden's density law and with a spherical velocity distribution*, Leningrad Univ. Bull., 19 (1962), pp. 96–111.

[4] T. D. Arber and R. G. L. Vann, *A critical comparison of Eulerian-grid-based Vlasov solvers*, J. Comput. Phys., 180 (2002), pp. 339–357.

[5] R. R. Arslanbekov, V. Kolobov, and A. Frolova, *Kinetic solvers with adaptive mesh in phase space*, Phys. Rev. E, 88 (2013), 063301.

[6] N. Besse, E. Deriaz, and E. Madaule, *Adaptive multiresolution semi-Lagrangian discontinuous Galerkin methods for the Vlasov equations*, J. Comput. Phys., 332 (2017), pp. 376–417. https://doi.org/10.1016/j.jcp.2016.12.003.

[7] N. Besse, G. Latu, A. Ghizzo, E. Sonnendrücker, and P. Bertrand, *A wavelet-MRA-based adaptive semi-Lagrangian method for the relativistic Vlasov–Maxwell system*, J. Comput. Phys., 227 (2008), pp. 7889–7916.

[8] J. Binney and S. Tremaine, *Galactic Dynamics*, 2nd ed., Princeton Ser. Astrophys. 1987, Princeton University Press, Princeton, NJ, 2011.

[9] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation*, Taylor & Francis, London, 1991.

[10] O. Bokanowski, J. Garcke, M. Griebel, and I. Klompmaker, *An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations*, J. Sci. Comput., 55 (2013), pp. 575–605.

[11] K. Brix, S. Melian, S. Müller, and M. Bachmann, *Adaptive multi-resolution methods: Practical issues on data structures, implementation and parallelization*, V. Louvet and M. Massot, eds., ESAIM Proc., 34 (2011), pp. 151–183.

[12] S. Brull and L. Mieussens, *Local discrete velocity grids for deterministic rarefied flow simulations* , J. Comput. Phys., 266 (2014), pp. 22–46.

[13] S. Brull, L. Forestier-Coste, and L. Mieussens, *Two dimensional local adaptive discrete velocity grids for rarefied flow simulations*, AIP Conf. Proc. 1786, American Institute of Physics, Melville, NY, 2016, 180002.

[14] J. Büchner, *Vlasov-code simulation*, in Proceedings of ISSS-7, 2005.

[15] C. Burstedde, L. C. Wilcox, and O. Ghattas, *p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees*, SIAM J. Sci. Comput., 33 (2011), pp. 1103–1133.

[16] 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 (2015), pp. A1825–A1845.

[17] A. Cohen, R. DeVore, G. Kerkyacharian, and D. Picard, *Maximal spaces with given rate of convergence for thresholding algorithms*, Appl. Comput. Harmon. Anal., 11 (2001), pp. 167–191.

[18] S. Colombi, T. Sousbie, S. Peirani, G. Plum, and Y. Suto, *Vlasov versus N-body: The Hénon sphere*, Monthly Not. Roy. Astronomical Soc., 450 (2015), pp. 3724–3741.

[19] G.-H. Cottet and P.-A. Raviart, *Particle methods for the one-dimensional Vlasov–Poisson equations*, SIAM J. Numer. Anal., 21 (1984), pp. 52–76.

[20] R. Courant, K. Friedrichs, and H. Lewy, *On the partial difference equations of mathematical physics*, IBM J. Res. Development, (1967), pp. 215–234.

[21] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conf. Ser. in Appl. Math. 61, SIAM, Philadelphia, 1992.

[22] P. Degond, F. Deluzet, L. Navoret, A.-B. Sun, and M.-H. Vignal, *Asymptotic-preserving particle-in-cell method for the Vlasov–Poisson system near quasineutrality*, J. Comput. Phys., 229 (2010), pp. 5630–5652.

[23] D. Del Sarto and E. Deriaz, *A multigrid AMR algorithm for the study of magnetic reconnection*, J. Comput. Phys., 351 (2017), doi10.1016/j.jcp.2017.08.046.

[24] S. Delage Santacreu, *Méthode de raffinement adaptatif hybride pour le suivi de fronts dans des écoulements incompressibles*, Thèse de Mécanique de l'Univérsité Bordeaux I, 2006.

[25] E. Deriaz, *Stability conditions for the numerical solution of convection-dominated problems with skew-symmetric discretizations*, SIAM J. Numer. Anal., 50 (2012), pp. 1058–1085.

[26] E. Deriaz and V. Perrier, *Direct numerical simulation of turbulence using divergence-free wavelets*, SIAM Multiscale Model. Simul., 7 (2008), pp. 1101–1129.

[27] G. Deslauriers and S. Dubuc, *Symmetric iterative interpolation processes*, Constr. Approx., 5 (1989), pp. 49–68.

[28] R. A. DeVore, S. V. Konyagin, and V. N. Temlyakov, *Hyperbolic wavelet approximation*, Constr. Approx., 14 (1998), pp. 1–26.

[29] D. L. Donoho, *Interpolating Wavelet Transforms*, Technical report, Stanford University, 1992.

[30] M. Dumbser, V. A. Titarev, and S. V. Utyuzhnikov, *Implicit multiblock method for solving a kinetic equation on unstructured meshes*, Comput. Math. Math. Phys., 53 (2013), pp. 601–615 .

[31] B. Eliasson, *Numerical simulations of the Fourier-transformed Vlasov-Maxwell system in higher dimensions—Theory and applications*, J. Transport Theory Statist. Phys., 39 (2010), pp. 387–465.

[32] E. Fijalkow, *Behaviour of phase-space holes in 2D simulations*, J. Plasma Phys., 61 (1999), pp. 65–76.

[33] F. Filbet, E. Sonnendrücker, and P. Bertrand, *Conservative numerical schemes for the Vlasov equation*, J. Comput. Phys., 172 (2001), pp. 166–187.

[34] F. Filbet and E. Sonnendrücker, *Comparison of Eulerian Vlasov solvers*, Comput. Phys. Commun., 150 (2003), pp. 247–266.

[35] T. Fujiwara, *Integration of the collisionless Boltzmann equation for spherical stellar systems*, Publ. Astron. Soc. Japan, 35 (1983), pp. 547–558.

[36] M. Grandin, *Data structures and algorithms for high-dimensional structured adaptive mesh refinement*, Adv. Engineering Software, 82 (2015), pp. 75–86.

[37] W. Guo and Y. Cheng, *A sparse grid discontinuous Galerkin method for high-dimensional transport equations and its application to kinetic simulations*, SIAM J. Sci. Comput., 38 (2016), pp. A3381–A3409.

[38] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations* I. *Nonstiff Problems*, Springer Ser. Comput. Math. 8, Springer-Verlag, Berlin, 1987.

[39] V. Grandgirard, M. Brunetti, P. Bertrand, N. Besse, X. Garbet, P. Ghendrih, G. Manfredi, Y. Sarazin, O. Sauter, E. Sonnendrücker, and J. Vaclavik, L. Villard, *A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation*, J. Comput. Phys., 217 (2006), pp. 395–423.

[40] A. Harten, *Multi-resolution algorithms for the numerical solution of hyperbolic conservation laws*, Comm. Pure Appl. Math., 48 (1995), pp. 1305–1342.

[41] F. Hecht, *New development in FreeFem++*, J. Numer. Math., 20 (2012), pp. 251–265.

[42] J. A. F. Hittinger and J. W. Banks, *Block-structured adaptive mesh refinement algorithms for Vlasov simulation*, J. Comput. Phys., 241 (2013), pp. 118–140.

[43] O. Hoenen and E. Violard, *Load-balancing for a block-based parallel adaptive 4D Vlasov solver*, in Proceedings of the 14th International Euro-Par Conference, 2008.

[44] M. Holmström, *Solving hyperbolic PDEs using interpolating wavelets*, SIAM J. Sci. Comput., 21 (1999), pp. 405–420.

[45] N. K.-R. Kevlahan and O. V. Vasilyev, *An adaptive wavelet collocation method for fluid-structure interaction*, SIAM J. Sci. Comput., 26 (2005), pp. 1894–1915.

[46] V. I. Kolobov and R. R. Arslanbekov, *Towards adaptive kinetic-fluid simulations of weakly ionized plasmas*, J. Comput. Phys., 231 (2012), pp. 839–869.

[47] K. Kormann, *A semi-Lagrangian Vlasov solver in tensor train format*, SIAM J. Sci. Comput., 37 (2015), pp. B613–B632.

[48] T. Nakamura and T. Yabe, *Cubic interpolated propagation scheme for solving hyper-dimensional Vlasov-Poisson equation in phase space*, Comput. Phys. Commun., 120 (1999), pp. 122–154.

[49] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.

[50] D. Pflüger, *Spatially Adaptive Sparse Grids for High-Dimensional Problems*, Ph.D. thesis, Technischen Universität München, 2010.

[51] N. Pham, P. Helluy, and L. Navoret, *Hyperbolic approximation of the Fourier transformed Vlasov equation*, ESAIM Proc., 45 (2014), pp. 379–389.

[52] S. Popinet, *Gerris: A tree-based adaptive solver for the incompressible Euler equations in complex geometries*, J. Comput. Phys., 190 (2003), pp. 572–600.

[53] O. Roussel, K. Schneider, A. Tsigulin, and H. Bockhorn, *A conservative fully adaptive multi-resolution algorithm for parabolic PDEs*, J. Comput. Phys., 188 (2003), pp. 493–523.

[54] A. Sandroos, I. Honkonena, S. von Alfthan, and M. Palmroth, *Multi-GPU simulations of Vlasov's equation using Vlasiator*, Parallel Comput., 39 (2013), pp. 306–318.

[55] K. Selim, A. Logg, and M. G. Larson, *An adaptive finite element splitting method for the incompressible Navier-Stokes equations*, Comput. Methods Appl. Mech. Engrg., 209–212 (2011), pp. 54–65.

[56] C. Shen, J.-M. Qiu, and A. Christlieb, *Adaptive mesh refinement based on high order finite difference WENO scheme for multi-scale simulations*, J. Comput. Phys., 230 (2011), pp. 3780–3802.

[57] E. Sonnendrücker, J. Roche, P. Bertrand, and A. Ghizzo, *The semi-Lagrangian method for the numerical resolution of the Vlasov equation*, J. Comput. Phys., 149 (1999), pp. 201–220.

[58] W. F. Spotz and G. F. Carey, *High-order compact finite difference methods*, in Proceedings of ICOSAM'95, 1995, pp. 397–408.

[59] V. Springel, *The Cosmological Simulation Code, GADGET*-2, MNRAS, 364 (2005), 1105.

[60] R. Teyssier, *Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES*, Astron. Astrophys., 385 (2002), pp. 337–364.

[61] K. Yoshikawa, N. Yoshida, and M. Umemura, *Direct integration of the collisionless Boltzmann equation in six-dimensional phase space: Self-gravitating systems*, Astrophys. J., 762 (2013), 116.

[62] O. Zanotti and M. Dumbser, *A high order special relativistic hydrodynamic and magnetohydrodynamic code with space-time adaptive mesh refinement*, Comput. Phys. Commun., 188 (2015), pp. 110–127.