Journal of Computational Physics 332 (2017) 376–417

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Adaptive multiresolution semi-Lagrangian discontinuous Galerkin methods for the Vlasov equations

N. Besse [a,b], E. Deriaz [b], É. Madaule [b,*]

[a] Laboratoire J.-L. Lagrange, UMR CNRS/OCA/UCA 7293, Université Côte d'Azur, Observatoire de la Côte d'Azur, Bd de l'Observatoire CS 34229, 06304 Nice Cedex 4, France
[b] Institut Jean Lamour, UMR 7198 CNRS – Université de Lorraine, Faculté des Sciences et Technologies – Bâtiment 2ème cycle, F-54506 Vandœuvre-lès-Nancy cedex, France

ARTICLE INFO

Article history:
Received 12 February 2016
Received in revised form 5 September 2016
Accepted 2 December 2016
Available online 6 December 2016

Keywords:
Vlasov–Poisson
Discontinuous Galerkin methods
Multi-wavelets
Multiresolution analysis
Semi-Lagrangian methods
Adaptive mesh refinement

ABSTRACT

We develop adaptive numerical schemes for the Vlasov equation by combining discontinuous Galerkin discretisation, multiresolution analysis and semi-Lagrangian time integration. We implement a tree based structure in order to achieve adaptivity. Both multi-wavelets and discontinuous Galerkin rely on a local polynomial basis. The schemes are tested and validated using Vlasov–Poisson equations for plasma physics and astrophysics.

1. Introduction

Many numerical experiments are performed on the Vlasov–Poisson problem since it is a well known (but not fully understood) system from plasma physics and a major issue for future simulation of large scale plasmas. Our goal is to develop adaptive numerical schemes using discontinuous Galerkin discretisation combined with semi-Lagrangian description whose mesh refinement is based on multi-wavelets. In [16], the authors start from the standard formulation of finite-volume schemes for conservation laws and transform them into adaptive finite-volume schemes by using multiscale representations of the data. In the same philosophy, starting from the standard formulation of the discontinuous Galerkin method for conservation laws the authors of Refs. [3,38,28,32,27] use a multi-wavelet representation of the unknowns to put these schemes into an adaptive setting. Following the spirit of these works, and in order to solve the transport and continuity equations, we use a multi-wavelet representation of the unknowns to transform the semi-Lagrangian discontinuous Galerkin schemes of references [29,36] into adaptive multiresolution versions. The discontinuous Galerkin formulation enables high-order accuracy with local data for computation. It has recently been widely studied by Heath [30], Ayuso de Dios et al. [4,5], Cheng et al. [12], Rossmanith et al. [37], Restelli et al. [33], etc. Those are done with Eulerian description of the time resolution but Guo et al. [29] or Qiu and Shu [36], Crouseilles et al. [19] or Bokanowski and Simarta [10] performed semi-Lagrangian time resolution. We use multi-wavelets framework for the adaptive part (more precisely, for the multiscale representation of the distribution function of particles). Those have been heavily studied by Alpert et al. [1,2]. Some works merging multiscale

[*] Corresponding author.
E-mail addresses: nicolas.besse@oca.eu (N. Besse), eric.madaule@univ-lorraine.fr (É. Madaule).

http://dx.doi.org/10.1016/j.jcp.2016.12.003

resolution and discontinuous Galerkin methods have been described by Archibald et al. [3] for convection based problems, by Müller et al. for non-linear hyperbolic conservation laws in the finite volume framework [32,28] or for compressible flows [27]. In the framework of relativistic Vlasov equation, Besse et al. [7] presented the advantage of using adaptive meshes. While they used wavelet representation, which requires large data stencil, multi-wavelet representation coupled to discontinuous Galerkin discretisation only requires local stencil. This favours the parallelisation but an efficient parallel version of our numerical schemes (presented hereafter) will be the matter of future work. Here we start with two semi-Lagrangian discontinuous Galerkin schemes which are presented in [36] and [29]. We call them the SLDG [36] and CDG [29] schemes. We then modify these schemes to obtain adaptive multiresolution schemes with an adaptive mesh coarsening and refinement procedure. To achieve this aim, among other things detailed in Secs. 4, 5 and 6, the data are represented in a different finite-dimensional space which is constructed by using a multi-wavelet basis [1,2]. The multiscale representation, given by the multi-wavelet basis, allows to give a well-defined and natural link between local small scales of the distribution function (or data) and the local size of the adaptive mesh. Then multi-wavelet representation of data is the relevant tool to develop natural and efficient criteria for designing an adaptive mesh (refinement versus coarsening) which follows the multiscale development of the distribution function. Therefore we obtain new numerical schemes that we call AMW-SLDG (Adaptive Multi-Wavelet SLDG) and AMW-CDG.

Let us note that the SLDG and CDG schemes are equivalent when we consider the one-dimensional advection or continuity equations on a one-dimensional grid with periodic boundary conditions. The proof is given in [36], where the authors assumed that all integrals involved in the schemes are exactly computed. Nevertheless it is important to start with two non-adaptive methods which can be proven equivalent to see if our adaptive procedure introduces a bias or some differences between them. For the moment there is no mathematical proof that the AMW-SLDG and AMW-CDG schemes are equivalent. It is not only a difficult task but it remains a true open question. Actually we do not know if they are rigorously equivalent. What is certain is that the proof performed in [36] can not be extended for non-conformal adaptive n-dimensional mesh (with $n \geq 2$), which is our case. In addition in our case, where the mesh is not uniform and non-conformal, integrals are not computed exactly. Nevertheless we may expect that the AMW-SLDG and AMW-CGD schemes are close to each other, and that our procedure does not break the “symmetry” between these two schemes. A way to confirm such intuition is to make systematic comparisons of the AMW-SLDG and AMW-CDG schemes.

The paper is organised as follows. Sec. 2 presents the Vlasov–Poisson equations and their conservation properties. Sec. 3 deals with several discontinuous Galerkin methods and their application in semi-Lagrangian framework. Sec. 4 describes the multi-wavelets used to get a multiscale reconstruction of the distribution function. In Sec. 5 we identify some numerical error sources which are responsible for a loss of some conservation laws. A brief overview of numerical implementation and a summary of the full algorithm are presented in Sec. 6. Some numerical results in plasma physics and astrophysics are presented in Sec. 7. Finally, the conclusion reminds the main points of our paper and gives future perspectives of work.

2. The Vlasov–Poisson equation

We considered the following Vlasov equation

$$\begin{cases} \partial_t f(x,v,t) + \partial_x (v f(x,v,t)) + \partial_v (E(x,t) f(x,v,t)) = 0, \quad x \in \Omega_x, \ v \in \mathbb{R}, \\ f(x,v,0) = f_0(x,v), \end{cases} \tag{1}$$

where $f(x,v,t)$ is the phase-space distribution function and $E(x,t)$ is the electric field obtained from the Poisson equation

$$-\Delta \Phi(x,t) = \rho(x,t) - 1, \quad x \in \Omega_x \tag{2a}$$

$$E(x,t) = -\partial_x \Phi(x,t), \tag{2b}$$

with $\Phi(x,t)$ the electric potential and $\rho(x,t)$ the charge density linked to the distribution function f according to

$$\rho(x,t) = \int_{\mathbb{R}} f(x,v,t)\, dv. \tag{3}$$

This equation is written in adimensional variables using a fixed background of ions such that the global plasma is neutral and the integral of the distribution function on the complete phase-space is equal to 1. This system is closed using periodic boundary conditions and zero average for $E(x,t)$. The set Ω_x is a compact domain of physical space with periodic boundary conditions. It is impossible from a practical point of view to use an infinite domain in v, except if the initial distribution function is compactly supported. For this reason most of the time we consider a domain $\Omega_v = [-L, L]$ with L large enough to ensure Dirichlet boundary conditions. In our simulations, given our initial conditions, we know that the solution remains bounded in velocity space by a decreasing exponential (uniformly in x). If we take a domain large enough in the velocity direction, then integrals are small enough to ensure that the error is bounded by a small exponential. Such estimation has been used in many semi-Lagrangian codes, such as the one from [6]. We call $\Omega = \Omega_x \times \Omega_v$.

Those equations represent the time evolution of a collisionless plasma distribution function and of its self-consistent electric field. This system satisfies many conservation laws that allow to qualify performances of the numerical methods.

- Conservation of maximum and minimum:

$$\frac{d}{dt}\left(\max_{(x,v)\in\Omega} f(x,v,t)\right)=0,$$
$$\frac{d}{dt}\left(\min_{(x,v)\in\Omega} f(x,v,t)\right)=0$$

- Conservation of L^p-norms, $1\le p\le\infty$:

$$\frac{d}{dt}\left(\int\limits_{\Omega_x\times\mathbb{R}} |f(x,v,t)|^p\,dxdv\right)^{1/p}=0,\quad 1\le p<\infty, \tag{4}$$

$$\frac{d}{dt}(\|f\|_\infty)=\frac{d}{dt}\left(\sup_{(x,v)\in\Omega_x\times\mathbb{R}} |f(x,v,t)|\right)=0,\quad p=\infty. \tag{5}$$

Note that for physical distribution function, $f_0(x,v)\ge 0$ everywhere, so $|f|=f$ and total mass is equivalent to L^1-norm

$$\frac{d}{dt}\int\limits_{\Omega_x\times\mathbb{R}} f(x,v,t)\,dxdv=0. \tag{6}$$

Because positivity is rarely conserved during simulation (unless enforced), there is often a difference between the numerical mass and the L^1-norm. Consequently we distinguish the two quantities in our diagnostics.

- Conservation of momentum:

$$\frac{d}{dt}\int\limits_{\Omega_x\times\mathbb{R}} vf(x,v,t)\,dxdv=0. \tag{7}$$

- Conservation of total energy:

$$\frac{d}{dt}\left(\int\limits_{\Omega_x\times\mathbb{R}} v^2 f(x,v,t)\,dxdv+\int\limits_{\Omega_x}|E(x,t)|^2\,dx\right)=0. \tag{8}$$

In order to preserve the maximum principle (in particular the positivity) for the distribution function, we could obviously use some limiters such as the positivity-preserving limiter of Qiu and Shu [36]. Nevertheless our aim is to evaluate the intrinsic properties of our schemes without adding an a posteriori procedure to enforce the positivity or the maximum principle. Moreover limiters might add a bit of diffusion sometimes. Therefore in our numerical simulations no limiter is used.

3. Discontinuous Galerkin formulations

3.1. General formulation

We explain here the principle of the discontinuous Galerkin formulation on a one-dimensional conservation law. Cockburn and Shu presented a large analysis of this formulation, mostly coupled with Runge–Kutta time resolution in, e.g. [15]. We perform the time resolution of the Vlasov–Poisson problem using Strang splitting. Indeed, here we use an (x,v)-area-preserving integrator in time obtained by decomposing the dynamics in integrable Hamiltonian steps. A natural and simple choice is the Strang time-splitting strategy also known as the centred leapfrog symplectic integrator. Roughly speaking, this intensively used scheme (see [13,23,8,6,36,18] and references therein) consists in splitting the full transport operator into two easily integrable Hamiltonian transport operators – one in the physical space, the second one in the velocity – and solving them successively in a right order to get high-order approximation in time of the complete transport operator. For the Strang splitting, the local (respectively global) error in time is of order three (respectively two) [8,22].

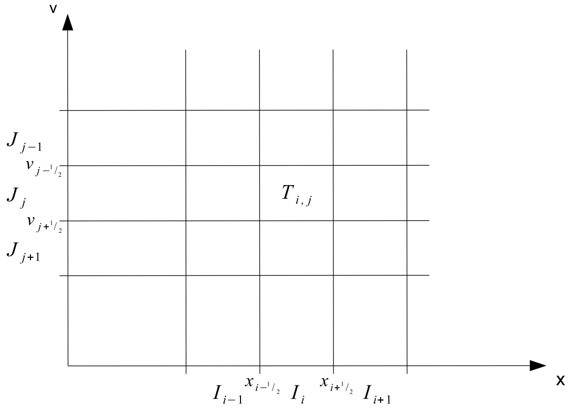


Fig. 1. Mesh representation and notation.

Let us consider the conservation equation

$$\partial_t f(x,t) + \partial_x(a(x,t)f(x,t)) = 0. \tag{9}$$

The discontinuous Galerkin formulation consists in solving a weak formulation of the problem on each cell of the mesh with a space of test functions. More precisely we first multiply (9) by a test function $g \in \mathcal{G}$ with $\mathcal{G}$ the space of test functions which we do not specify yet. We then integrate it on each cell $I_i = [x_{i-1/2}, x_{i+1/2}]$ of the domain partition to obtain the weak formulation

$$\int\limits_{I_i} \partial_t f(x,t)g(x)\,dx + \int\limits_{I_i} \partial_x(a(x,t)f(x,t))g(x)\,dx = 0. \tag{10}$$

The next stage is to perform an integration by part to transfer the space derivative onto the test function.

$$\int\limits_{I_i} \partial_t f(x,t)g(x)\,dx - \int\limits_{I_i} a(x,t)f(x,t)\partial_x g(x)\,dx + [a(x,t)f(x,t)g(x)]_{x_{i-1/2}}^{x_{i+1/2}} = 0. \tag{11}$$

Here comes the choice of the test function space $\mathcal{G}$ and numerical fluxes since the flux terms in (11) is meaningless when using discontinuous reconstruction at the cell interfaces. This choice will determine the properties (stability, consistency, accuracy, conservation) of the numerical scheme. For each cell we consider $\{\mathbb{P}_i^k\}_i$, the set of polynomials of degree at most k on I_i and null outside of I_i. The test functions g are then taken as basis functions of $\mathbb{P}_i^k$. Most of the time those are Legendre or Lagrange polynomials on I_i, but depending on the situation, other polynomial basis may also be adequate. According to the philosophy of these methods, we do a L^2-projection of f on the underlying polynomial space

$$f_h(x,t) = \sum_i \sum_{l=0}^{k} f_{l,i}(t)p_{l,i}(x) \tag{12}$$

where $\{p_{l,i}\}_{l=0..k}$ is a basis of $\mathbb{P}_i^k$. The major advantage of this formulation is that it is fully independent of the spatial order, which can be considered as a parameter. According to the Taylor series representation of the distribution function, the error committed by the polynomial projection is proportional to $\Delta x^{k+1} f^{(k+1)}$, where $f^{(k+1)}$ is the $k+1$-th derivative of f.

Let us now introduce the notation used in the next sections. We name $I_i = [x_{i-1/2}, x_{i+1/2}]$ and $J_j = [v_{j-1/2}, v_{j+1/2}]$ the intervals respectively along the x axis and along the v axis. The two-dimensional cell $T_{i,j} = I_i \times J_j$ is the tensorial product of I_i in direction x and J_j in direction v as depicted on Fig. 1. The Gauss–Lobatto nodes are noted x_{i_g} (respectively v_{j_g}) in direction x (respectively v) and the associated weight is w_{ig} (respectively w_{j_g}). Gauss weights are always expressed for a quadrature formula on the interval $[-1,1]$. $\mathbb{P}_i^k$ is the space of polynomials of degree at most k on cell I_i. In the two-dimensional case the polynomial basis also is 2D and is obtained as tensorial product of the 1D basis. We use the notation $P_{\mathbf{k},i,j}(x,v) = p_{k_1,i}(x)p_{k_2,j}(v)$ with $\mathbf{k} = (k_1, k_2)$. For all our work we used Legendre polynomials of degree up to k as basis of $\mathbb{P}^k([-1,1])$ and a L^2-normalisation. Superscript n is used to indicate time $t^n = n\Delta t$ and $f_h^n(x) = f_h(x,t^n)$. About characteristics, we call x the variable at time t^{n+1} and we call x^* the foot of the characteristic at time t^n.

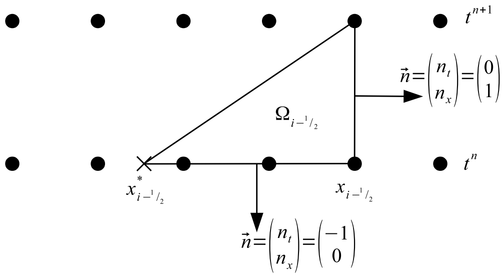

t^{n+1}
$\vec{n}=\begin{pmatrix} n_t \\ n_x \end{pmatrix}=\begin{pmatrix} 0 \\ 1 \end{pmatrix}$
$\Omega_{i-1/2}$
t^n
$x^*_{i-1/2}$
$x_{i-1/2}$
$\vec{n}=\begin{pmatrix} n_t \\ n_x \end{pmatrix}=\begin{pmatrix} -1 \\ 0 \end{pmatrix}$

Fig. 2. Characteristic scheme for SLDG method.

3.2. The semi-Lagrangian discontinuous Galerkin formulation

This formulation is taken from [36]. We only adapt it to obtain a formulation in a two-dimension phase-space and abbreviate it SLDG. The main idea is to perform a time-integration of (11) and to transform it into a space-integration using characteristics. We start from (11) and proceed to a time integration to obtain

$$\begin{aligned}\int_{I_i} f^{n+1}(x)g\,dx = \int_{I_i} f^n(x)g(x)\,dx + \int_{t^n}^{t^{n+1}}\int_{I_i} af(x,t)g'(x)\,dx\,dt \\ - \int_{t^n}^{t^{n+1}} \left((af(x,t)g(x))|_{x^-_{i+1/2}} - (af(x,t)g(x))|_{x^+_{i-1/2}} \right) dt, \quad \forall g \in \mathcal{G}.\end{aligned} \tag{13}$$

We respectively call the three integrals on the right hand side T_0, T_1 and T_2. The time integral in T_1 and T_2 are evaluated in a semi-Lagrangian fashion. We start from Fig. 2. On this figure, point $x^*_{i-1/2}$ is the foot of characteristic arriving at $x_{i-1/2}$ at time t^{n+1}. Domain $\Omega_{i-1/2}$ is the region bounded by the three points $(x_{i-1/2}, t^{n+1})$, $(x_{i-1/2}, t^n)$ and $(x^*_{i-1/2}, t^n)$. We integrate (9) over the domain $\Omega_{i-1/2}$ and apply the divergence theorem:

$$\begin{aligned} 0 &= \int_{\Omega_{i-1/2}} (\partial_t f(x,t) + \partial_x(a(x,t)f(x,t)))\,dx\,dt \\ &= \int_{\partial\Omega_{i-1/2}} (f(x,t)n_t + af(x,t)n_x)\,ds. \end{aligned} \tag{14}$$

Therefore,

$$\int_{x^*_{i-1/2}}^{x_{i-1/2}} f(x,t^n)\,dx = \int_{t^n}^{t^{n+1}} af(x_{i-1/2},t)\,dt. \tag{15}$$

Hence it is possible to write

$$T_2 = g(x^+_{i-1/2}) \int_{x^*_{i-1/2}}^{x_{i-1/2}} f^n_h(x)\,dx - g(x^-_{i+1/2}) \int_{x^*_{i+1/2}}^{x_{i+1/2}} f^n_h(x)\,dx. \tag{16}$$

By first performing spatial integration using Gaussian quadrature rule on T_1 and following the same process, the semi-Lagrangian formulation gives

$$\begin{aligned} T_1 &= \int_{t^n}^{t^{n+1}} \int_{I_i} af_h(x,t)g'(x)\,dx\,dt \\ &= \int_{t^n}^{t^{n+1}} \frac{\Delta x_i}{2} \sum_{i_g} af_h(x_{i_g},t)g'(x_{i_g})w_{i_g}\,dt \end{aligned} \tag{17}$$

$$= \frac{\Delta x_i}{2}\sum_{i_g} w_{i_g} g'(x_{i_g}) \int_{x^*_{i_g}}^{x_{i_g}} f_h^n(x)\,dx.$$

Foot of characteristics are obtained by solving backward the ODE

$$\begin{cases} \dfrac{dx}{dt} = a(x,t), \\ x(t^{n+1}) = x_i, \end{cases} \tag{18}$$

between t^n and t^{n+1}.

As a consequence, (11) becomes

$$\begin{aligned} \int_{I_i} f_h^{n+1}(x) g(x)\,dx = \int_{I_i} f_h^n(x) g(x)\,dx + \frac{\Delta x_i}{2}\sum_{i_g} w_{i_g} g'(x_{i_g}) \int_{x^*_{i_g}}^{x_{i_g}} f_h^n(x)\,dx + g(x^+_{i-1/2}) \int_{x^*_{i-1/2}}^{x_{i-1/2}} f_h^n(x)\,dx \\ - g(x^-_{i+1/2}) \int_{x^*_{i+1/2}}^{x_{i+1/2}} f_h^n(x)\,dx. \end{aligned} \tag{19}$$

One can clearly see that if g is chosen as a basis function of $\mathbb{P}_i^k$, then the left hand side is simply $f_{l,i}^{n+1}$ and the first line of the right hand side is $f_{l,i}^n$. The other terms are not simplified.

In the two-dimensional case, integration over cell $T_{i,j}$ requires an additional integration in the orthogonal direction. Polynomials are only defined on a reference element $[-1,1]$ and all variables and functions taken on this reference element are indicated with a tilde. The Gauss weights are not included in this statement since they are only used on this reference element. We define $\mathbf{k} = (k_1, k_2)$ such that $\mathbf{k}$ is the index for polynomials and coefficients in two-dimension and, k_1 is the corresponding index in direction x and k_2 the corresponding index in direction v. This leads to the integration in the x direction

$$\begin{aligned} \tilde{f}^{n+1}_{\mathbf{k},T_{i,j}} = & \tilde{f}^{n}_{\mathbf{k},T_{i,j}} + \frac{2}{\Delta x_i}\sum_{j_g} w_{j_g} \left(\sum_{i_g} w_{i_g} \int_{x^*_{i_g}}^{x_{i_g}} f_h^n(x, v_{j_g})\,dx\, \tilde{p}'_{k_1}(\tilde{x}_{i_g}) \right) \tilde{p}_{k_2}(\tilde{v}_{j_g}) \\ & + \frac{2}{\Delta x_i}\sum_{j_g} w_{j_g} \left(\int_{x^*_{i-1/2}}^{x_{i-1/2}} f_h^n(x, v_{j_g})\,dx\, \tilde{p}_{k_1}(\tilde{x}^+_{i-1/2}) - \int_{x^*_{i+1/2}}^{x_{i+1/2}} f_h^n(x, v_{j_g})\,dx\, \tilde{p}_{k_1}(\tilde{x}^-_{i+1/2}) \right) \tilde{p}_{k_2}(\tilde{v}_{j_g}), \end{aligned} \tag{20}$$

with

$$f^n_{\mathbf{k},T_{i,j}} = \frac{\Delta x_i \Delta v_j}{4} \tilde{f}^n_{\mathbf{k},T_{i,j}}, \tag{21}$$

and in the v direction

$$\begin{aligned} \tilde{f}^{n+1}_{\mathbf{k},T_{i,j}} = & \tilde{f}^{n}_{\mathbf{k},T_{i,j}} + \frac{2}{\Delta v_j}\sum_{i_g} w_{i_g} \left(\sum_{j_g} w_{j_g} \int_{v^*_{j_g}}^{v_{j_g}} f_h^n(x_{i_g}, v)\,dv\, \tilde{p}'_{k_2}(\tilde{v}_{j_g}) \right) \tilde{p}_{k_1}(\tilde{x}_{i_g}) \\ & + \frac{2}{\Delta v_j}\sum_{i_g} w_{i_g} \left(\int_{v^*_{j-1/2}}^{v_{j-1/2}} f_h^n(x_{i_g}, v)\,dv\, \tilde{p}_{k_2}(\tilde{v}^+_{j-1/2}) - \int_{v^*_{j+1/2}}^{v_{j+1/2}} f_h^n(x_{i_g}, v)\,dv\, \tilde{p}_{k_2}(\tilde{v}^-_{j+1/2}) \right) \tilde{p}_{k_1}(\tilde{x}_{i_g}). \end{aligned} \tag{22}$$

3.3. The characteristic-discontinuous-Galerkin formulation

We take the algorithm for this method from [29] and we call it characteristic-discontinuous Galerkin and abbreviate it CDG. Nevertheless it has previously been called Lagrange-characteristic in [14] or Lagrange projection even sooner in [21] and [35]. It also appeared in the framework of electron hole and drift diffusion for semi-conductors [20,26]. As for the

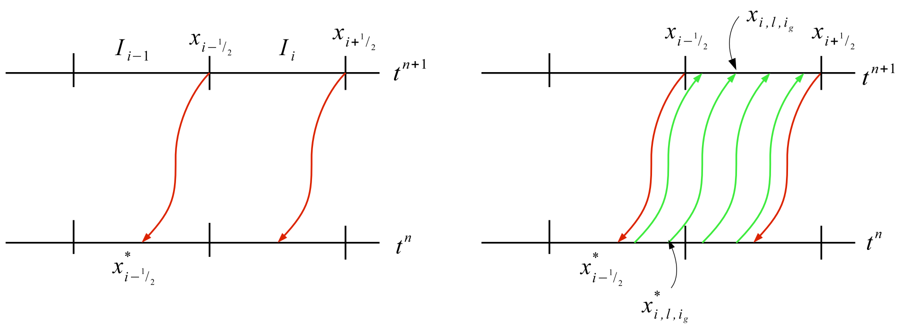


Fig. 3. Characteristic scheme for CDG.

SLDG formulation, we only adapt it to our two-dimensional problem. The starting point of the method is to determine $I_i^* = [x^*_{i-1/2}, x^*_{i+1/2}]$ using (18), as presented on the left part of Fig. 3. The conservation equation (9) implies that

$$f_h^{n+1}(x) = f_h^n(x^*)\left|\frac{\partial x^*}{\partial x}\right|. \tag{23}$$

Multiplying (23) by $g(x)$, integrating the result in the variable x on cell I_i, and using the change of variable from x to x^* in the right hand side of the resulting equality, we then obtain

$$\int_{I_i} f_h^{n+1}(x) g(x)\, dx = \int_{I_i^*} f_h^n(x^*) g(x(x^*))\, dx^*. \tag{24}$$

Note that in the right hand side, the integration variable is x^* but test function g depends on variable x. These two variables are linked by (18). Variable x^* is at time t^n while x is at time t^{n+1}. This second integral is sliced on each sub-interval $I_i^* \cap I_l$, and on every sub-interval the integral is computed using Gaussian quadrature rule so that (24) becomes

$$\begin{aligned}\int_{I_i^*} f_h^n(x^*) g(x)\, dx^* &= \sum_l \int_{I_i^* \cap I_l} f_h^{n+1}(x^*) g(x(x^*))\, dx^* \\ &= \sum_l \left(\sum_{i_g} w_{i_g} f_h^n(x^*_{i,l,i_g}) g(x(x^*_{i,l,i_g})) \frac{\Gamma(I^*_{i,l})}{2} \right),\end{aligned} \tag{25}$$

with $\Gamma(I^*_{i,l})$ the length of $I_i^* \cap I_l$. Here we have $I_i^* = [x^*_{i-1/2}, x^*_{i+1/2}]$, where x^*_{i,l,i_g} are foot of characteristics of Gauss points on $I_i^* \cap I_l$ and x_{i,l,i_g} are the associated end of characteristics on I_i. This process is illustrated on the right part of Fig. 3. Here again, taking $g = p_{l,i}$, the left hand side equals to $f_{l,i}^{n+1}$.

Performing just the same as previously we integrate on the orthogonal direction over J_j to obtain the formulation in direction x:

$$\tilde{f}^{n+1}_{\mathbf{k},T_{i,j}} = \sum_{j_g} w_{j_g} \sum_l \sum_{i_g} w_{i_g} f_h^n(x^*_{i_g,l,j_g}, v_{j_g}) p_{\mathbf{k},l}(x_{i_g,l,j_g}, v_{j_g}) \frac{\Gamma(I^*_{i,l,j_g})}{\Delta x_i}. \tag{26}$$

Note that here, because the foot of characteristics depends on the orthogonal direction, Γ also depends on the orthogonal direction. We do the same in direction v:

$$\tilde{f}^{n+1}_{\mathbf{k},T_{i,j}} = \sum_{i_g} w_{i_g} \sum_l \sum_{j_g} w_{j_g} f_h^n(x_{i_g}, v^*_{j_g,l,i_g}) p_{\mathbf{k},l}(x_{i_g}, v_{j_g,l,i_g}) \frac{\Gamma(J^*_{j,l,i_g})}{\Delta v_j}. \tag{27}$$

The following three sections describe how we modify the SLDG and CDG schemes to transform them into the adaptive AMW-SLDG and AMW-CDG schemes. These adaptive schemes rely on a multi-wavelet representation of the data (Sec. 4) and some special considerations concerning numerical evaluation of the integrals involved in the schemes (Sec. 5). These schemes also rely on a prediction procedure to achieve adaptive mesh refinement (Sec. 6).

4. Multi-wavelets representation

The multi-wavelets are a tool to build a multiscale representation of any function belonging to a suitable functional space, typically L^2. Their properties in one dimension are detailed in [1]. It then has been widely used, such as [32] and

[28]. An algorithm for the construction of one-dimensional multi-wavelets can be found in [2]. Let us remind the most important points from [1] about multi-wavelets.

Polynomial multi-wavelets form a multiscale basis of $L^2([0,1])$. Let us consider a nested sequence of spaces

$$\mathcal{V}_n^k = \left\{ f|_{[2^{-n}l\,;\,2^{-n}(l+1)]} \in \mathbb{P}^k, \quad l \in [\![0\,;\,2^n-1]\!] \right\},$$

where $\mathbb{P}^k$ is the set of polynomials of degree at most k. The dimension of $\mathcal{V}_n^k$ is $2^n(k+1)$ and

$$\mathcal{V}_0^k \subset \mathcal{V}_1^k \subset \cdots \subset \mathcal{V}_n^k \subset \cdots \subset L^2([0,1]), \tag{28}$$

such that

$$\operatorname{clos}_{L^2([0,1])}\left(\bigcup_{n=0}^{\infty} \mathcal{V}_n^k\right) = L^2([0,1]). \tag{29}$$

Given an orthogonal basis $\{\phi_j\}$ of $\mathcal{V}_0^k$, it is possible to obtain an orthogonal basis $\{\phi_{j,l}^n(x)\}$ of $\mathcal{V}_n^k$ by dilatation and translation such that

$$\phi_{j,l}^n(x) = 2^{^n/2}\phi_j(2^n x - l), \quad j \in [\![0\,;\,k]\!], \quad l \in [\![0\,;\,2^n-1]\!], \quad n \in \mathbb{N}. \tag{30}$$

It is always possible to find a space $\mathcal{W}_n^k$ such that

$$\mathcal{V}_n^k \oplus \mathcal{W}_n^k = \mathcal{V}_{n+1}^k, \quad \mathcal{V}_n^k \perp \mathcal{W}_n^k, \tag{31}$$

with $\mathcal{W}_n^k$ of dimension $2^n(k+1)$. As a consequence of (31), we have

$$\mathcal{V}_n^k = \mathcal{V}_0^k \bigoplus_{\eta=0}^{n-1} \mathcal{W}_\eta^k. \tag{32}$$

Given an orthogonal basis $\{\psi_j\}$ of $\mathcal{W}_0^k$, following (30), the piecewise polynomial basis functions of $\mathcal{W}_n^k$ are $\{\psi_{j,l}^n(x)\}$ such that

$$\psi_{j,l}^n(x) = 2^{^n/2}\psi_j(2^n x - l), \quad j \in [\![0\,;\,k]\!], \quad l \in [\![0\,;\,2^n-1]\!]. \tag{33}$$

Functions $\phi_{j,l}^n$ are called scaling functions while functions $\psi_{j,l}^n$ are called multi-wavelets functions. In our case, scaling functions are Legendre polynomials rescaled to the corresponding cell.

Once $\{\phi_j\}$ and $\{\psi_j\}$ are known, it is then possible to build filters $h_{i,j}^0$, $h_{i,j}^1$, $g_{i,j}^0$ and $g_{i,j}^1$ according to [1] such that we have a restriction operator

$$\phi_i(x) = \sqrt{2}\sum_{j=0}^{k}\left(h_{i,j}^{(0)}\phi_j(2x) + h_{i,j}^{(1)}\phi_j(2x-1)\right), \qquad i \in [\![0,k]\!], \tag{34a}$$

$$\psi_i(x) = \sqrt{2}\sum_{j=0}^{k}\left(g_{i,j}^{(0)}\phi_j(2x) + g_{i,j}^{(1)}\phi_j(2x-1)\right), \qquad i \in [\![0,k]\!], \tag{34b}$$

and an interpolation operator

$$\phi_i(2x) = \frac{1}{\sqrt{2}}\sum_{j=0}^{k}\left(h_{j,i}^{(0)}\phi_j(x) + g_{j,i}^{(0)}\psi_j(x)\right), \qquad i \in [\![0,k]\!], \tag{35a}$$

$$\phi_i(2x-1) = \frac{1}{\sqrt{2}}\sum_{j=0}^{k}\left(h_{j,i}^{(1)}\phi_j(x) + g_{j,i}^{(1)}\psi_j(x)\right), \qquad i \in [\![0,k]\!]. \tag{35b}$$

Note that if you consider only polynomials of degree 0, you obtain Haar wavelets.

Starting from (31) it is possible to get the multiscale decomposition $P_n f$ of f at level m on multi-wavelets functions basis

$$P_n f(x) = \sum_{j=0}^{k} s_{j,0}^0 \phi_j(x) + \sum_{m=0}^{n-1}\sum_{l=0}^{2^m-1}\sum_{j=0}^{k} d_{j,l}^m \psi_{j,l}^m(x), \tag{36}$$

where coefficients $\{s_{j,l}^m\}$ and $\{d_{j,l}^m\}$ are obtained from a L^2-projection on the corresponding space,

$$s^n_{j,l} = \int\limits_{2^{-n}l}^{2^{-n}(l+1)} f(x)\phi^n_{j,l}(x)\,dx, \tag{37a}$$

$$d^n_{j,l} = \int\limits_{2^{-n}l}^{2^{-n}(l+1)} f(x)\psi^n_{j,l}(x)\,dx. \tag{37b}$$

Filters can be built for any polynomial basis. Alpert et al. [1] present their construction for Legendre polynomial basis and Lagrange polynomial basis. In our code we use only Legendre polynomial basis.

According to (34)–(36) it is possible to compute level m from level $m+1$, using

$$s^m_{i,l} = \sum_{j=0}^{k}\left(h^{(0)}_{i,j}s^{m+1}_{j,2l} + h^{(1)}_{i,j}s^{m+1}_{j,2l+1}\right), \tag{38a}$$

$$d^m_{i,l} = \sum_{j=0}^{k}\left(g^{(0)}_{i,j}s^{m+1}_{j,2l} + g^{(1)}_{i,j}s^{m+1}_{j,2l+1}\right), \tag{38b}$$

and level $m+1$ from level m, using

$$s^{m+1}_{i,2l} = \sum_{j=0}^{k}\left(h^{(0)}_{j,i}s^m_{j,l} + g^{(0)}_{j,i}d^m_{j,l}\right), \tag{39a}$$

$$s^{m+1}_{i,2l+1} = \sum_{j=0}^{k}\left(h^{(1)}_{j,i}s^m_{j,l} + g^{(1)}_{j,i}d^m_{j,l}\right). \tag{39b}$$

Formulae (38)–(39) can be extended to two-dimensional case using tensorial products. Therefore space $\mathcal{W}^k_n$ becomes

$$\mathcal{W}^k_n = \left\{\{\phi^n_{i,l}(x)\psi^n_{j,l'}(y)\},\ \{\psi^n_{i,l}(x)\phi^n_{j,l'}(y)\},\ \{\psi^n_{i,l}(x)\psi^n_{j,l'}(y)\}, i,j \in [\![0\,;\,k]\!],\ l,l' \in [\![0\,;\,2^n-1]\!]\right\}. \tag{40}$$

In order to perform restriction or interpolation in the two dimensional case the restriction or interpolation operators are applied in the first direction then in the second direction.

This means that there are two equivalent ways to consider $f_h(x,v)$, the distribution function on a given cell, by using only coefficients of levels m and $m+1$. The first way is to consider only the level $m+1$

$$f_h(x,v) = \sum_{c=0,1,2,3}\sum_{\mathbf{q}}\phi^{m+1}_{\mathbf{q},[c]}(x,v)s^{m+1}_{\mathbf{q},[c]}. \tag{41}$$

In this case, q is the index over the two dimensional basis functions and c is the index over the children cells. The second way is to consider the distribution function at level m plus the details

$$f_h(x,v) = \sum_{\mathbf{q}}\phi^m_{\mathbf{q}}(x,v)s^m_{\mathbf{q}} + \sum_{\Theta=a,b,c}\sum_{\mathbf{k}}\psi^m_{\mathbf{k},[c]}(x,v)d^m_{\mathbf{k},[\Theta]} \tag{42}$$

where a, b and c respectively are the subspaces of $\mathcal{W}^k_n$ generated by $\{\phi^n_{i,l}(x)\psi^n_{j,l'}(y)\}$, $\{\psi^n_{i,l}(x)\phi^n_{j,l'}(y)\}$ and $\{\psi^n_{i,l}(x)\psi^n_{j,l'}(y)\}$. The thresholding operation consists in comparing the l^2-norm of the details $\{d^m_{.,[.]}\}$ to a given threshold and ignore it where it is smaller than this threshold, that is,

$$\text{if}\ \left(\sum_{\Theta=a,b,c}\sum_{\mathbf{k}}\left(d^m_{\mathbf{k},[\Theta]}\right)^2\right)^{1/2} \le \epsilon_0,\ \text{then ignore details of level}\ m, \tag{43}$$

and repeat this operation until all details kept are assessed significant, proceeding from highest level to lowest level. This thresholding criterion is presented for l^p-norm of details in [7]; here we use only the l^2-norm. Parameter ϵ_0 and maximum level of refinement are chosen empirically to have a good description of initial distribution function, with the possibility to follow details during simulation, but still big enough to have reasonable number of points and simulation time. We refer to [7] for a discussion concerning thresholding, adaptivity and optimality.

Multi-wavelets are built and used with respect to the L^2-norm, which means that part of the L^2-norm depends on the multi-wavelet coefficients and then the L^2-norm is not conserved when coarsening. Nevertheless, contrary to some usual interpolation schemes, the interest and the power of the multi-wavelets representation is not only to control but also to minimise the loss of L^2-norm by choosing locally in an optimal way the details which deserve to be conserved or discarded.

Table 1
Difference of momentum between levels.

Levels	Difference
7 with 6	9.05942E−14
6 with 5	7.10543E−15
7 with 5	9.76996E−14

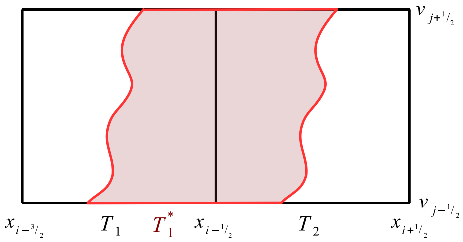


Fig. 4. Transport of a 2D cell in one direction. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

Such property is confirmed by the numerical experiment of Sec. 7. Equations (41) and (42) also ensure that moments of the distribution function f are conserved during coarsening up to the degree of scaling functions, thanks to the orthogonality properties of the multi-wavelet functions $\psi^n_{j,l}$. This is confirmed with numerical experiment. We consider the distribution function

$$f(x,v) = \begin{cases} 0 & \text{if } v < 0, \\ \frac{v^2}{\sqrt{2\pi}}\,(1+0.05\cos(0.5x))\exp\left(\frac{-v^2}{2}\right) & \text{else,} \end{cases} \qquad (x,v) \in [0,4\pi] \times [-6,6]. \tag{44}$$

For this distribution function, we consider uniform meshes with respectively 7, 6 and 5 level of refinement and use polynomials of degree up to two for each direction. We first express the distribution function on uniform mesh of level 7 and compute its first momentum in velocity $vf(x,v)$, that is

$$\int_0^{4\pi}\int_{-6}^{6} vf(x,v)\,dv\,dx. \tag{45}$$

We then compute the distribution function on uniform mesh of level 6 following (38a) and compute the momentum. This means we only compute the scaling coefficients $s^m_{i,l}$ and eliminate the multi-wavelet coefficients $d^m_{i,l}$. We do the same to get the distribution function on uniform mesh of level 5 and compute the momentum. The difference of momentum computed is displayed in Table 1. This difference is of the magnitude of machine error which validate our code.

5. Error analysis

In this section we identify some sources of numerical error which are responsible for a loss of some conservation laws. Therefore, by explaining their origins and mechanisms we could find a way to fix them. In addition, even if the conservation laws are not exactly preserved a priori, such conservation laws are preserved asymptotically thanks to the convergence property of the AMW-SLDG and AMW-CDG schemes, which is numerically shown in Secs. 7.1 and 7.2.

5.1. Error linked to integration on transported cells

In this section we explain one of the limit and error source of our schemes. We present it on a uniform mesh since it is sufficient to understand the source of error and it is much easier than on non-conformal mesh.

Let us start with the AMW-CDG method on uniform mesh and the equation

$$\frac{\partial f}{\partial t}(x,v) + g(v)\frac{\partial f}{\partial x}(x,v) = 0, \tag{46}$$

where $g(v)$ is a polynomial of degree m and m smaller than or equal to K the degree of our Legendre polynomials. In our algorithm we then use $K+1$ Gauss points. We consider time t^n.

On Fig. 4, domains T_1 and T_2 are bounded with black lines and domain T_1^* in red is obtained by transporting T_1 according to Sec. 3.3. Domains $T_1 \cap T_1^*$ and $T_2 \cap T_1^*$ are equivalent for the following analysis. Let us use only $T_2 \cap T_1^*$.

We start from

$$\begin{aligned} T_1^* \cap T_2 =& [x_{i-1/2}, x^*_{i-1/2}] \times [v_{j-1/2}, v_{j+1/2}] \\ =& \bigcup_{v \in [v_{j-1/2}, v_{j+1/2}]} [x_{i-1/2}, \mathcal{P}(v)]. \end{aligned} \tag{47}$$

with

$$x^*_{i-1/2} = \mathcal{P}(v) = x_{i-1/2} + g(v)dt. \tag{48}$$

Consequently, $\mathcal{P} \in \mathbb{P}^m$. According to the method we then must solve

$$\int_{T_1^* \cap T_2} f_h^n(x^*, v) P_{k_1,k_2}(x(x^*), v)\, dx^*\, dv = \int_{v_{j-1/2}}^{v_{j+1/2}} \left(\int_{x_{i-1/2}}^{\mathcal{P}(v)} f_h^n(x^*, v) P_{k_1,k_2}(x(x^*, v), v)\, dx^* \right) dv. \tag{49}$$

According to the resolution of characteristics equations (18), one has

$$x(x^*, v) = x^* - g(v)dt. \tag{50}$$

Here, x is a polynomial of degree one in the variable x^* and a polynomial of degree m in the variable v.

We proceed to the following change of variable

$$x^* = \frac{a - \mathcal{P}(v)}{a-b}\bar{x} + a\frac{\mathcal{P}(v) - b}{a-b}, \tag{51}$$

in order to integrate over rectangle $(\bar{x}, v) \in [a, b] \times [v_{j-1/2}, v_{j+1/2}]$. Formula (47) then becomes

$$\begin{aligned} \int_{T_1^* \cap T_2} f_h^n(x^*, v) P_{k_1,k_2}(x(x^*), v)\, dx^*\, dv = & \int_{v_1}^{v_2}\int_a^b f_h^n\left(\frac{a - \mathcal{P}(v)}{a-b}\bar{x} + a\frac{\mathcal{P}(v) - b}{a-b}, v\right) \\ & P_{k_1,k_2}\left(x\left(\frac{a - \mathcal{P}(v)}{a-b}\bar{x} + a\frac{\mathcal{P}(v) - b}{a-b}\right), v\right) \frac{a - \mathcal{P}(v)}{a-b}\, d\bar{x}\, dv. \end{aligned} \tag{52}$$

We must then search the maximum degree of

$$f_h^n\left(\frac{a - \mathcal{P}(v)}{a-b}\bar{x} + a\frac{\mathcal{P}(v) - b}{a-b}, v\right) P_{k_1,k_2}\left(x\left(\frac{a - \mathcal{P}(v)}{a-b}\bar{x} + a\frac{\mathcal{P}(v) - b}{a-b}\right), v\right) \frac{a - \mathcal{P}(v)}{a-b}.$$

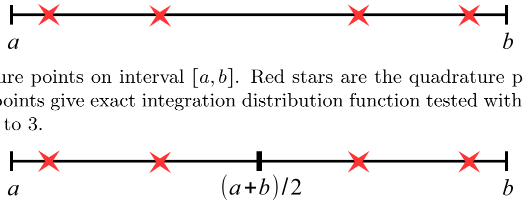
Let us write

$$\begin{aligned} f_h^n(x^*, v) P_{k_1,k_2}(x(x^*, v), v) =& f_h^n(x^*, v) P_{k_1,k_2}(x^* - g(v)dt, v) \\ =& f_h^n\left(\frac{a - \mathcal{P}(v)}{a-b}\bar{x} + a\frac{\mathcal{P}(v) - b}{a-b}, v\right) \\ & P_{k_1,k_2}\left(x\left(\frac{a - \mathcal{P}(v)}{a-b}\bar{x} + a\frac{\mathcal{P}(v) - b}{a-b} - g(v)dt\right), v\right). \end{aligned} \tag{53}$$

The distribution function f_h^n is a polynomial of degree K in x and in v. Then, considering only monomials of highest degree, we obtain

$$\underbrace{\left((v^m\bar{x})^K v^K\right)}_{f_h^n} \underbrace{\left((v^m\bar{x})^K v^K\right)}_{P_{K,K}} \underbrace{v^m}_{\text{Jacobian of (51)}} = \bar{x}^{2K} v^{2K+2Km+m}. \tag{54}$$

Using $K+1$ Gauss points we can perform exact integration only for polynomials of degree up to $2K+1$. Consequently, according to (54), integral (52) is computed exactly in direction x but not in direction v if m is larger than or equal to 1. Then, as we implemented and use our AMW-CDG scheme, it does not conserve mass.



(a) Quadrature points on interval $[a, b]$. Red stars are the quadrature points. These quadrature points give exact integration distribution function tested with polynomials of degree up to 3.

(b) Quadrature points on interval $[a, b]$. Red stars are the quadrature points. Here polynomials are independently defined on $[a, (a+b)/2]$ and on $[(a+b)/2, b]$. The quadrature points do not give exact integration.

Fig. 5. Quadrature points on interval $[a, b]$ when $[a, b]$ is single segment and when it is two segments. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For the AMW-SLDG method, starting from (46) and term T_1 from (13), we get

$$\begin{aligned} T_1 &= \int_{J_j} \left[p_{k_2}(v) \int_{I_i} \left(\tilde{p}_{k_1}(\tilde{x}(x)) \int_{\underbrace{x-g(v)dt}_{x^*}}^{x} f_h^n(\xi, v)\, d\xi \right) dx \right] dv \\ &= \int_{I_i} \left[\tilde{p}_{k_1}(\tilde{x}(x)) \int_{J_j} p_{k_2}(v) \left(\int_{x-g(v)dt}^{x} f_h^n(\xi, v)\, d\xi \right) dv \right] dx. \end{aligned} \tag{55}$$

In the second line, for a given x, the problem is equivalent to the one presented above for the AMW-CDG scheme. This is also true for term T_2 in (13). Consequently, as for our AMW-CDG scheme, our implementation of the method AMW-SLDG does not conserve mass a priori. Nevertheless, in the numerical experiments of Sec. 7, we observe a rather good conservation of mass a posteriori. Such result is consistent with the fact that the schemes are convergent (see Secs. 7.1 and 7.2).

In both case (AMW-CDG and AMW-SLDG) we see that an exact integration would be a first step to a priori mass conservation. An exact integration (considering a possibly piecewise polynomial $\mathcal{P}(v)$) would require to compute the equation of the transported border. Since in our case, function $g(v)$ is always known (it is either v or $E_h^n(x)$ depending on the direction) and a polynomial, the transported border can always be computed. It is possible to integrate exactly polynomials on domains with polynomial boundaries but the computation cost increases. In addition, the fact that the boundary may be piecewise polynomial if $g = E_h$ also increases the complexity. However, this last point can be avoided if one consider level of refinement for the electric field E_h coarse enough.

5.2. Error linked to the a priori lack of flux compensation

We perform most of our integration using quadrature formula with Gauss points. For a one-dimensional mesh (uniform or not) and for conformal n-dimensional mesh, $n \geq 2$, this formula is exact because the quadrature points are always adapted to each segment, as shown on Fig. 5a. In the case of non-conformal mesh, we use a quadrature formula which is exact for given polynomials. However, our distribution function is not a single polynomial on the domain, as depicted on Fig. 5. It is piecewise polynomial. Therefore, integration performed with Gauss quadrature does not give the exact value. This particular case happen for the integration in direction v in equations (20) and (26) and for the integration in direction x in equations (22) and (27).

In addition, in the case of non-conformal mesh, fluxes are not computed at the same points for adjacent cells. This is illustrated on Fig. 6. The blue points are quadrature points for the big cell on the left. The red points are quadrature points for the small cell on the right. We clearly see that blue points do not match red points. Consequently, the distribution function, and thus fluxes, computed at the blue points do not have the same values as the distribution function and fluxes computed at the red points. Therefore there is no guaranty of a priori flux conservation. In a sense, to ensure a priori flux conservation, the computation of fluxes must be independent of the scale of the interface. Therefore we can imagine a process which assigns to each interface a unique flux value from the flux values computed at the different scales considered at this interface.

6. Algorithm and numerical implementation

This section is devoted to the description of our algorithm and to a few implementation points. Our algorithm is as follows.

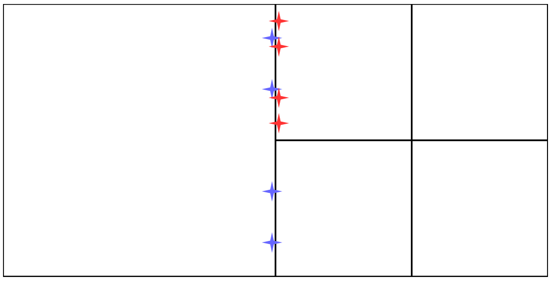
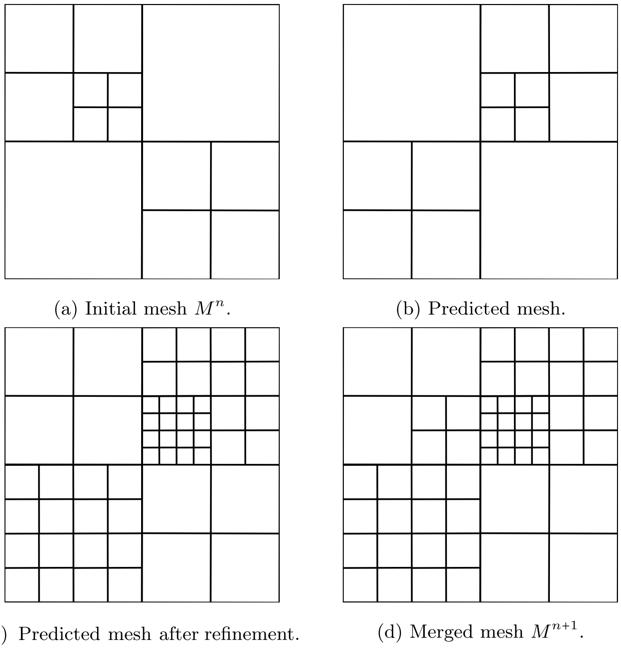


Fig. 6. Points for flux computation. Blue stars correspond to the points where flux are computed on the big cell on the left. Red stars correspond to the points where flux are computed on the small cell on the right. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Initial mesh M^n.

(b) Predicted mesh.

(c) Predicted mesh after refinement.

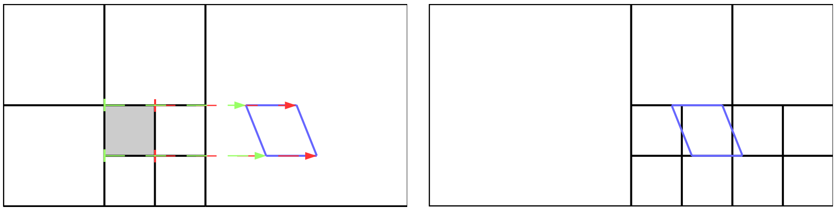
(d) Merged mesh M^{n+1}.

Fig. 7. Drawing of mesh prediction.

Step 1: **initialisation.** We project the distribution function f on the coarsest level of mesh (usually level 0) and compute the details at same level. If details are large enough, the cell is refined. We repeat this operation until description of f is precise enough according to (43). The thresholding used for refinement is the same as the one that will be used for coarsening on future steps.

Step 2: **first step of Strang splitting.** The first step of the Strang splitting is a half time step in direction x only (from t^n to $t^n + dt/2$). Each step of our Strang splitting can be divided into three stages.

- *Prediction.* We first predict the mesh by using a time explicit Eulerian scheme to compute characteristics forward and to ensure that the cell containing end point of characteristics are at least one level finer than their departure point. This prediction will enable the mesh to follow the small structures and anticipate the creation of smaller ones. Fig. 7 is an illustration of this process. Fig. 7a represents the initial mesh M^n. Fig. 7b represents the mesh obtained by transporting the initial mesh M^n. To do so we use an explicit Euler scheme to transport the boundaries of each cell of Fig. 7a. We then make sure that all the area described by the transported cell has the expected level of refinement. This process is illustrated in Fig. 8. Each corner (x, v) is transported to obtain a new corner $(x + v\,dt, v)$. Then we refine the mesh such that the area described by the transported corners is at least of the same level of refinement. Mesh of Fig. 7c is obtained by adding one level of refinement to the mesh of Fig. 7b to anticipate creation of smaller structures during the computation of the time step. This choice of prediction is taken from [7]. While they add a criterion to choose whether to add or not one (or more than one) level of refinement, we do not have this criterion and always make the mesh one level thinner. We refer to [7] for a discussion of accuracy and optimality during the prediction step. Mesh M^{n+1} of Fig. 7d is the merging of initial mesh M^n (Fig. 7a) and the predicted mesh of Fig. 7c.
The prediction of the mesh is done using an independent structure that contains the geometry but not the data.



(a) Prediction of the transport of the grey cell. The initial mesh is black. The considered cell is grey. Red and green arrows show the displacement of the corner of the cell and blue is the transported cell.

(b) Creation of the predicted mesh. In blue is the transported cell and in black are all the newly created cells.

Fig. 8. Drawing of the transport prediction of the grey cell. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We manipulate two trees. The one on Figs. 7a and 7d contains the distribution function while the one presented on Figs. 7b and 7c contains only the geometry and its tree data. The mesh M^{n+1} (for computation) must be at least as locally refined as the initial mesh M^n, otherwise significant details of the distribution function f_h^n would be lost during the next computation step. The distribution function f_h^n known on the mesh M^n (Fig. 7a) is projected on the mesh M^{n+1} (Fig. 7d) with all unknown thinner details set to zero. From (28) and thanks to the merging of the predicted mesh with the initial mesh, there is neither loss of information nor of moments conservation during the prediction step (for moments of order lower than the degree of scaling functions).

- *Computation.* The chosen numerical scheme (SLDG or CDG) is applied on the final mesh to update the distribution function. We solve

$$\partial_t f + \partial_x(vf) = 0 \tag{56}$$

between t^n and $t^{n+1/2}$, as presented in Secs. 3.2 and 3.3.
- *Coarsening.* We perform recursive coarsening operation according to (43) in order to keep only the finest significant details.

Step 3: **resolution of the Poisson equation.** We solve the Poisson equation using the distribution function obtained after Step 2. The field is always computed on the finest level of refinement as its numerical cost is very small compared to the phase-space numerical cost.

Step 4: **second step of Strang splitting.** The second step of Strang splitting is a full time step in direction v only (from t^n to t^{n+1}). We solve

$$\partial_t f + \partial_v(Ef) = 0. \tag{57}$$

This step follows the scheme of Step 2 and uses data from Steps 2 and 3.

Step 5: **third step of Strang splitting.** The third and last step of Strang splitting is identical to Step 2 (first step of Strang splitting) except that it uses data from Step 4 and its result is the distribution function at time t^{n+1}.

The mesh is altered only at very specific steps. It is refined only during the refinement stage of steps 2, 4 and 5, and coarsened only during the coarsening stage of these steps 2, 4 and 5. The mesh is fixed for the computation and the evolution of the distribution function, and for solving the Poisson problem.

The mesh is organised as a tree structure, but unlike many tree structures we do not concentrate data on leaf cells. In our case, we remark that for computation of data of level m, considering or not cells of level higher than m where they were available, had absolutely no impact on the solution. This means that coarse cells are not affected by fine details during the time integration process. Therefore, storing the distribution function at every level according to (41) and using only the coarsest significant level (m where available or the coarser level if level m is not available) reduces the number of computation needed. This operation requires to compute the distribution function on each cell of the tree at the end of steps 2, 4 and 5 (at the end of each step the distribution function is known only on leaves). The computational impact is significant only when the mesh becomes highly heterogeneous in terms of local density of cell.

Here is another optimisation we use. Let p_l be the l-th normalised Legendre polynomial. Then we have

$$\int_{-1}^{1} p_l(x)\,dx = \begin{cases} \sqrt{2}/2 & \text{if } l = 0, \\ 0 & \text{else.} \end{cases} \tag{58}$$

This is used in the computation of (20)–(22) to reduce the computational cost where the integration domain includes full cells with

Table 2
Error for the rotation with AMW-SLDG and AMW-CDG schemes for polynomials of degree 2 and 3.

Threshold	$\langle h\rangle$	L^1		L^2		L^∞	
		Error	Order	Error	Order	Error	Order
1E−1	4.47	2.57	–	0.401	–	0.210	–
1E−2	2.09	0.0523	5.10	0.0149	4.32	0.0173	3.27
1E−3	1.14	5.98E−3	4.43	1.39E−3	4.14	1.52E−3	3.60
1E−4	0.529	6.31E−4	3.89	1.46E−4	3.71	2.94E−4	3.08
1E−5	0.236	6.53E−5	3.60	1.42E−5	3.48	2.84E−5	3.03
(a) Error obtained with the AMW-SLDG scheme for polynomials of degree 2.							
1E−1	4.47	0.786	–	0.179	–	0.357	–
1E−2	2.43	0.0118	6.87	3.72E−3	6.33	6.81E−3	6.47
1E−3	2.09	5.85E−3	6.42	1.34E−3	6.42	1.72E−3	6.99
1E−4	1.23	1.38E−3	4.92	3.03E−4	4.95	4.63E−4	5.16
1E−5	0.625	7.11E−5	4.73	1.76E−5	4.69	4.22E−5	4.59
(b) Error obtained with the AMW-SLDG scheme for polynomials of degree 3.							
1E−1	4.47	2.57	–	0.401	–	0.210	–
1E−2	2.09	0.0538	5.07	0.0149	4.31	0.0184	3.19
1E−3	1.14	7.87E−3	4.23	1.77E−3	3.97	1.67E−3	3.54
1E−4	0.527	6.90E−4	3.85	1.45E−4	3.71	2.81E−4	3.09
1E−5	0.236	1.31E−4	3.36	2.36E−5	3.31	2.84E−5	3.03
(c) Error obtained with the AMW-CDG scheme for polynomials of degree 2.							
1E−1	4.47	0.787	–	0.179	–	0.357	–
1E−2	2.43	0.0118	6.87	3.72E−3	6.33	6.81E−3	6.47
1E−3	2.09	5.88E−3	6.42	1.34E−3	6.42	1.72E−3	6.99
1E−4	1.24	1.45E−3	4.90	3.08E−4	4.95	4.63E−4	5.17
1E−5	0.626	1.18E−4	4.47	2.71E−5	4.49	5.57E−5	4.46
(d) Error obtained with the AMW-CDG scheme for polynomials of degree 3.							

$$
\begin{aligned}
\int\limits_{I_i} f(x)\,dx = & \int\limits_{I_i} \sum_k f_k p_k(x)\,dx \\
= & f_0 \int\limits_{I_i} p_0(x)\,dx \\
= & f_0\sqrt{2}/2.
\end{aligned}
\tag{59}
$$

This simplification is particularly efficient when the time step is large compared to the size of the cell.

7. Numerical results

7.1. Linear transport: rotation

We consider the rotation problem

$$\partial_t f + v\partial_x f - x\partial_v f = 0, \quad (x, v) \in [-10, 10]^2, \tag{60a}$$

$$f(x, v, 0) = \exp\left(-(x-5)^2 - v^2\right). \tag{60b}$$

The exact solution in time is

$$f(x, v, t) = \exp\left(-(x\,cos(t) + v\,sin(t) - 5)^2 - (v\,cos(t) - x\,sin(t))^2\right), \tag{61}$$

we use $\Delta t = 10^{-5}$ and $1,000$ time steps, which makes final time $T_f = 10^{-2}$. Our goal here is to check the order of convergence to zero of the L^2-error in phase-space. We perform simulations with polynomials of degree 2 and 3 as indicated in Table 2. The maximum level of refinement is 8 but is not reached in these simulations. The threshold criteria are given in Table 2. For each threshold value, the average phase-space discretisation step $\langle h\rangle$ is the average length of cell side over phase-space and time, that is

$$\langle h\rangle = \left(\frac{L_x L_v (N_t + 1)}{\sum_{n=0}^{N_t} C^n}\right)^{1/2}, \tag{62}$$

Table 3
Error for the Burgers' equation with AMW-SLDG and AMW-CDG schemes for polynomials of degree 2 and 3.

Threshold	$\langle h\rangle$	L^1		L^2		L^∞	
		Error	Order	Error	Order	Error	Order
1E−1	1.25	0.204	–	0.0860	–	0.0839	–
1E−2	0.313	0.0105	2.14	3.80E−3	2.25	3.54E−3	2.28
1E−3	0.157	1.30E−3	2.43	4.78E−4	2.50	4.65E−4	2.50
1E−4	0.0869	2.49E−4	2.51	9.00E−5	2.57	1.34E−4	2.41
1E−5	0.0412	2.40E−5	2.65	8.28E−6	2.71	1.00E−5	2.64
(a) Error obtained with the AMW-SLDG scheme for polynomials of degree 2.							
1E−1	1.25	0.187	–	0.0612	–	0.0454	–
1E−2	0.627	8.94E−3	4.38	2.99E−3	4.36	2.88E−3	3.98
1E−3	0.313	5.26E−4	4.24	1.89E−4	4.17	2.09E−4	3.88
1E−4	0.198	1.78E−4	3.77	7.46E−5	3.64	1.09E−4	3.27
1E−5	0.157	3.28E−5	4.16	1.19E−5	4.11	1.38E−5	3.89
(b) Error obtained with the AMW-SLDG scheme for polynomials of degree 3.							
1E−1	1.25	0.204	–	0.0860	–	0.0839	–
1E−2	0.313	0.0105	2.14	3.80E−3	2.25	3.54E−3	2.28
1E−3	0.157	1.30E−3	2.43	4.78E−4	2.50	4.65E−4	2.50
1E−4	0.0869	2.48E−4	2.52	9.00E−5	2.57	1.34E−4	2.41
1E−5	0.0412	2.30E−5	2.66	8.16E−6	2.71	1.03E−5	2.64
(c) Error obtained with the AMW-CDG scheme for polynomials of degree 2.							
1E−1	1.25	0.187	–	0.0612	–	0.0454	–
1E−2	0.627	8.94E−3	4.38	2.99E−3	4.36	2.88E−3	3.98
1E−3	0.313	5.26E−4	4.24	1.89E−4	4.17	2.08E−4	3.88
1E−4	0.198	1.78E−4	3.77	7.46E−5	3.64	1.08E−4	3.27
1E−5	0.157	3.24E−5	4.16	1.19E−5	4.11	1.34E−5	3.91
(d) Error obtained with the AMW-CDG scheme for polynomials of degree 3.							

with L_x and L_v respectively the length of the domain in direction x and v, N_t the number of time steps, and C^n the number of cells at time t^n.

For polynomials of degree k we expect the error to be of order $k+1$. We observe in Table 2 that the rate of convergence is between $k+1$ and $k+2$ for $k=2,\ 3$ and for both schemes.

7.2. Burgers equation

We consider the Burgers' equation

$$\partial_t f(x,v,t)+\partial_x(f(x,v,t)^2)/2=0, \quad (x,v)\in[0,2\pi]\times[-1,1], \tag{63a}$$

$$f(x,v,0)=1+sin(x). \tag{63b}$$

We perform simulations with $\Delta t=10^{-5}$ and $1,000$ time steps for a final time $T_f=10^{-2}$. The threshold criteria are given in Table 3. For each threshold value, average size $\langle h\rangle$ is the average length of cell side over space and time (62). For this test case, we do not enforce the maximum level of refinement. These simulations are done either with polynomials of degree two or three, as indicated in captions of Table 3. For polynomials of degree k, the order of the error is expected to be $k+1$. In the Table 3 we observe a convergence rate between $k+1/2$ and $k+1$. Even if we do not always obtain the optimal convergence rate $k+1$, we must point out that, in nonlinear cases, convergence rate may crucially depend on the properties of the reconstruction operator (here multi-wavelets) and on the geometric features of the mesh (here a two-dimensional non-conformal adaptive mesh). Here, in order to understand better this behaviour, a careful and rigorous mathematical analysis of the schemes for the Burgers equations should be done. Nevertheless it is beyond the scope of this paper. Moreover the nature and behaviour of the Burgers nonlinearity is quite different from the quadratic nonlinearity appearing in Vlasov–Poisson equations. Therefore, in this paper, we do not pursue further with a finer analysis.

7.3. Swirling deformation

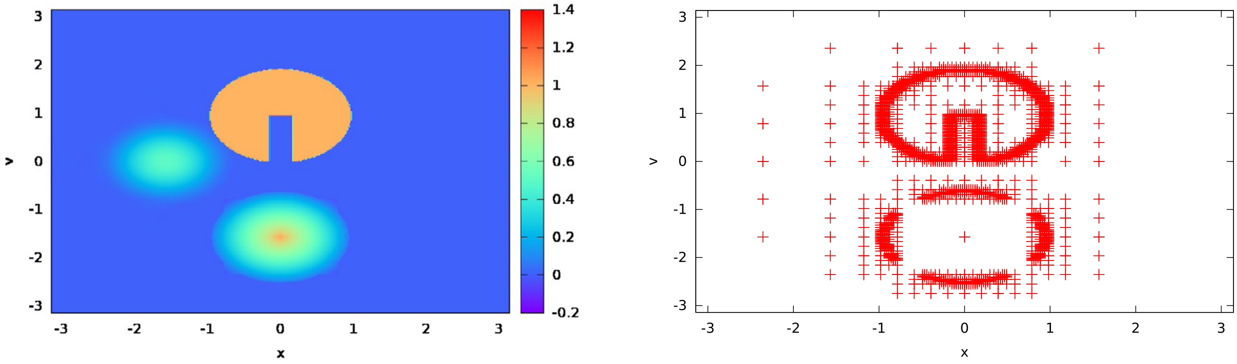
We consider solving

$$\partial_t f-\partial_x\left(g(x,v,t)f\right)+\partial_v\left(g(v,x,t)f\right)=0, \quad x\in[-\pi;\pi], \quad v\in[-\pi;\pi], \tag{64}$$

with

$$g(x,v,t)=\pi\cos^2\left(\frac{x}{2}\right)\sin(v)\cos\left(\frac{\pi t}{T}\right), \tag{65}$$

and $T=1.5$. This test case is the example 5.5 of [36].



(a) Initial distribution function.

(b) Initial mesh

Fig. 9. Initial distribution function and mesh for swirling deformation.

The initial conditions are

$$f_0(x,v) = \begin{cases} 1 \quad \text{if} \quad \sqrt{x^2+(v-\pi/2)^2} \leq 0.30\pi \text{ and } (|x| \geq \pi/20 \text{ or } v \geq 0.30\pi), \\ 1-\sqrt{x^2+(v+\pi/2)^2} \quad \text{if} \quad \sqrt{x^2+(v+\pi/2)^2} \leq 0.30\pi, \\ 1/4(1+\cos(\pi\sqrt{(x+\pi/2)^2+v^2})) \quad \text{if} \quad \sqrt{(x+\pi/2)^2+v^2} \leq 0.30\pi, \\ 0 \quad \text{else}. \end{cases}$$

The domain for this case is $[-\pi,\pi]^2$. Final time is 1.5 and this simulation is done using 100 time steps for a time step $\Delta t = 0.015$. The maximum level of refinement is 8, the level 0 being the full domain, and the thresholding parameter is $\epsilon_0 = 0.01$. Our polynomials are of degree up to 3 per direction.

The initial distribution function and the initial mesh are presented on Fig. 9. For convenience, oscillations at boundaries of the truncated cylinder have been truncated on this plot. On Fig. 9b, each cross indicates a cell corner. It is obvious that our mesh is non-conformal, and according to equations (20)–(22) and (26)–(27), conform mesh is not required. Although the mesh may seem sparse, the distribution function on each cell is described using several polynomials. For polynomials of degree up to k, there are $(k+1)^2$ polynomials per cell. Consequently, in this case, the distribution function is described using 16 polynomials per cell.

Fig. 10 presents the distribution function and mesh of the swirling deformation during the deformation and at the final time for the AMW-SLDG scheme. In both case the mesh follows very well details of the distribution function, especially the discontinuity of the cylinder. One can see that its sharp border matches with high mesh refinement along time, which is the expected behaviour. The results of the two methods (AMW-SLDG and AMW-CDG) also look extremely similar during simulation and at final time. Because for all our test cases distribution functions and meshes obtained with both methods always give results that can hardly be distinguished, we only present them for one method.

One can clearly see that the final mesh is very different from the initial mesh, especially at the boundary of the truncated cone. Initially this part requires many points because the distribution function is not $\mathscr{C}^1$. However, the use of a polynomial basis smooths the distribution function on each cell. After just a few time steps, the distribution function acquires higher local regularity than it initially had.

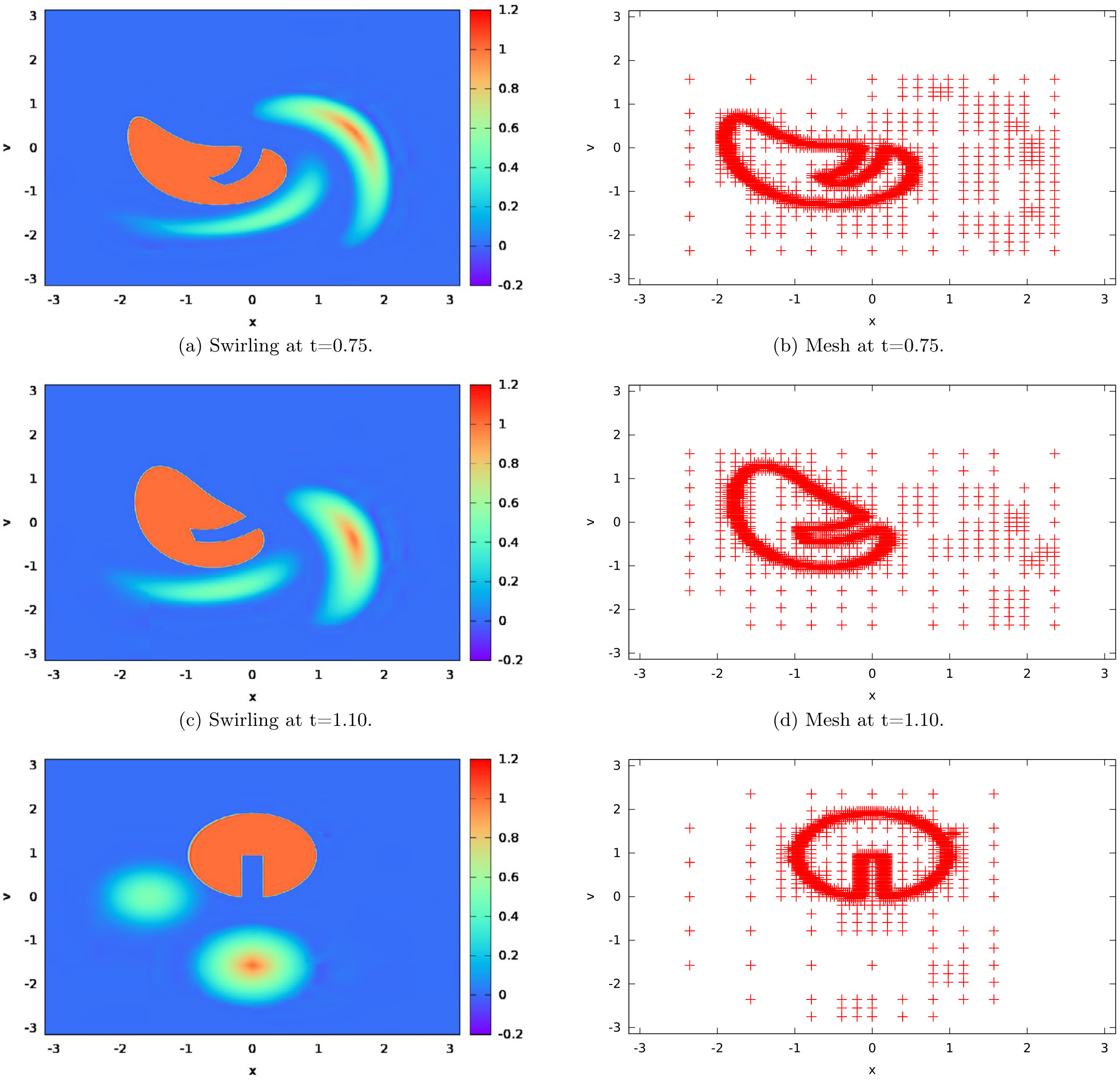
Since this case projects a strongly discontinuous function on a polynomial space, the initial distribution function presents strong artificial oscillations and it can hardly be considered good test case for conservation properties. Nevertheless, time evolution of numerical mass, L^1-norm and L^2-norm for this test case are presented on Fig. 11 and show good conservations.

7.4. *Landau damping*

Landau damping is a standard test case to validate codes for Vlasov–Poisson in two-dimensional phase-space for plasma physics. The initial distribution function is a Gaussian function in velocity with a perturbation in space:

$$f_0(x,v) = \frac{(1-\alpha\cos(x/2))}{\sqrt{2\pi}} \exp\left(\frac{-v^2}{2}\right), \quad (x,v) \in [0,4\pi] \times [-L_v, L_v], \tag{66}$$

where L_v is large enough to ensure Dirichlet boundary condition. Usually $L_v = 5$ is sufficient. Two main values are chosen for α. First, $\alpha = 0.01$ corresponds to the linear Landau damping, also called weak Landau damping. With this small perturbation it is possible to get a linearised approximation on the electric energy along time, and especially the damping rate. Secondly, $\alpha = 0.5$ corresponds to the non-linear Landau damping, also called strong Landau damping. Although there is no analytical solution for this case, it has been heavily simulated and there are many references to compare the damping rate in first stage and the growth rate of second stage, such as [4,37,31] and many others. For both linear and non-linear



(a) Swirling at t=0.75.

(b) Mesh at t=0.75.

(c) Swirling at t=1.10.

(d) Mesh at t=1.10.

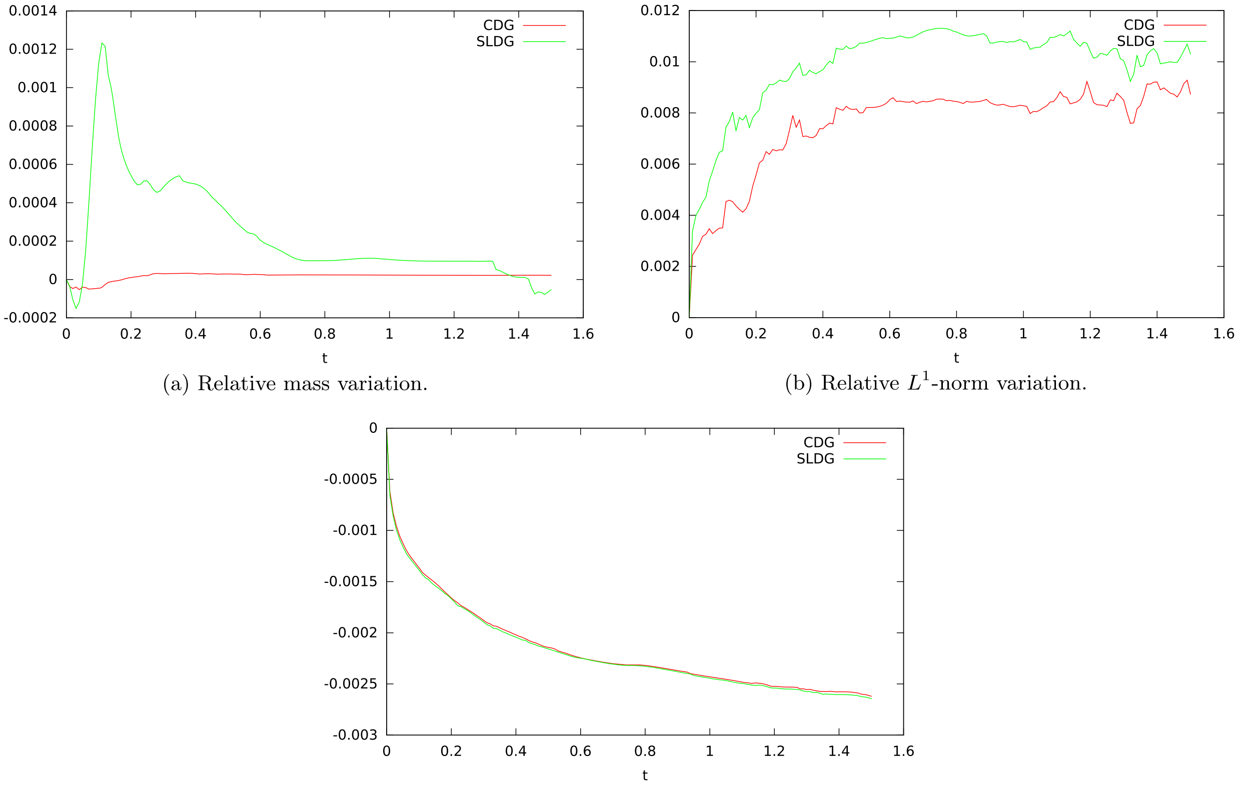
(e) Swirling at t=1.50.

(f) Mesh at t=1.50.

Fig. 10. Swirling deformation at various time for the AMW-SLDG scheme.

cases, final time is $T = 100$ with 10^3 time steps for a time step $\Delta t = 0.1$. The maximum level of refinement is 5 for the weak Landau damping test case and 8 for the strong Landau damping one. We use polynomials of degree up to 2 for each direction of space. Threshold is $\epsilon_0 = 10^{-4}$ for weak Landau damping, while it is $\epsilon_0 = 10^{-2}$ for strong Landau damping.

For the weak Landau damping, at final time, the relative variation of mass and the relative variation of L^1-norm are around 2.8×10^{-7}, the relative loss of L^2-norm is around 1.7×10^{-5}, the relative variation of L^∞-norm is almost -0.7%, the relative loss of total energy is slightly less than 0.4% and momentum remains between -1.5×10^{-10} and 0. Fig. 12 depicts the evolution of electric field for the AMW-CDG and AMW-SLDG methods (red and green curves) and the analytical damping (blue curve) for weak Landau damping. Here again the two methods give very similar results. (It is possible to see a difference between the AMW-CDG and AMW-SLDG methods in the long time behaviour.) The expecting damping rate is also recovered until the details become finer than our thresholding criterion. On long time the electric field increases again. This phenomenon is clearly due to the recurrence phenomena described in [13] and [34]. For uniform mesh, the recurrence time is $T_r = 2\pi/(k\Delta v)$. If we consider the recurrence time computed using a uniform mesh with the maximum number of cells, we have $\Delta v = 0.3125$, which gives us a recurrence time $T_r = 40.21$. On Fig. 12, we observe two recurrence phenomena. The first one is at time $t = 38$ and the second one is around time $t = 76$. The numerical recurrence time $T_{r,\text{num}} = 38$ is in good agreement with the predicted one $T_r = 40.21$, since the relative error is about 5.5%



(a) Relative mass variation.

(b) Relative L^1-norm variation.

(c) Relative L^2-norm variation

Fig. 11. Relative variations of numerical mass, L^1-norm and L^2-norm for swirling deformation.

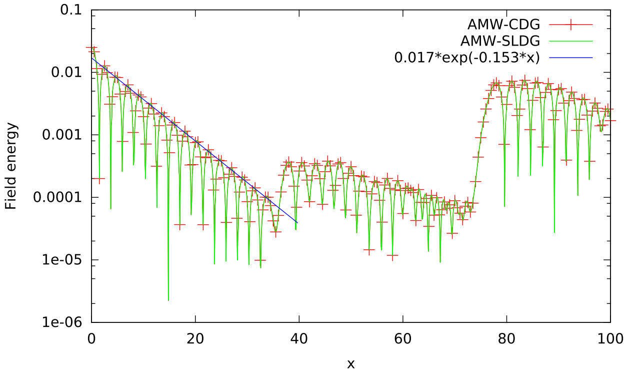
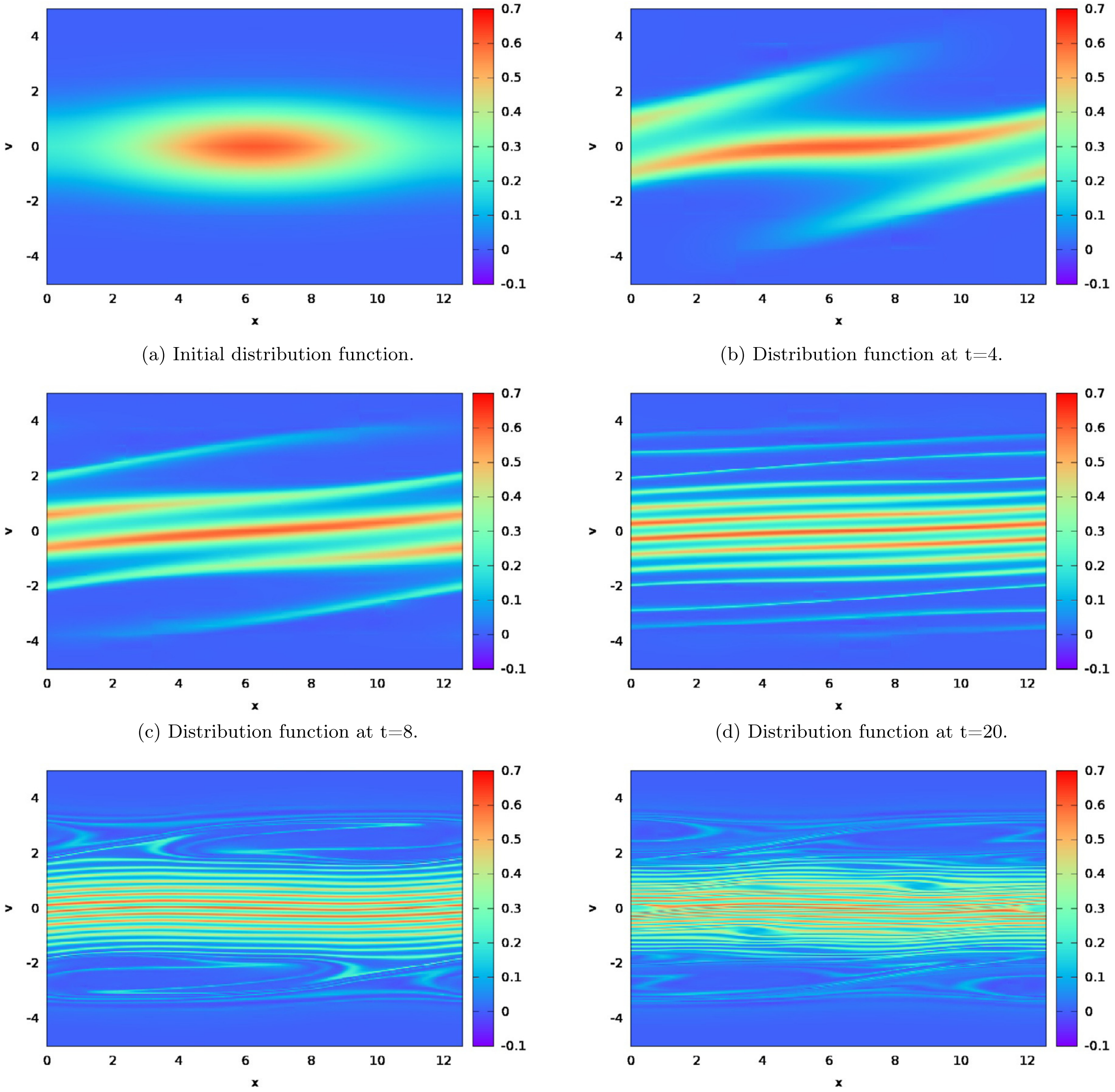


Fig. 12. Electric energy for weak Landau damping. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

Fig. 13 shows the distribution function at various time for strong Landau damping while Fig. 14 presents the mesh. These are obtained with the AMW-CDG method. The distribution function can not be distinguished from the one obtained with the AMW-SLDG method and meshes are extremely similar. At time $t = 4$ and $t = 8$ it is clearly visible that the mesh is finer where details of the distribution function are more important. At large time, because fine details are present almost everywhere on the domain, the mesh is at its finest level almost everywhere for $v \in [-4, 4]$ and looks like a static non-uniform mesh which would be thin for $v \in [-4, 4]$ and coarse outside.

Fig. 15 indicates the variation of the main diagnostics with both methods. For this test case, the numerical mass is very well conserved but L^1-norm varies more, which is conform to the loss of positivity. None of the methods preserve total energy but the damping rate and growing rate of electric energy match the literature values. Even if not perfect, momentum appears to be quite well conserved too. It will be easier to observe momentum time variation on bump-on-tail since initial momentum is then not zero. The large increase of total energy occurs during the first plasma period (about 12 time steps).



(a) Initial distribution function.

(b) Distribution function at t=4.

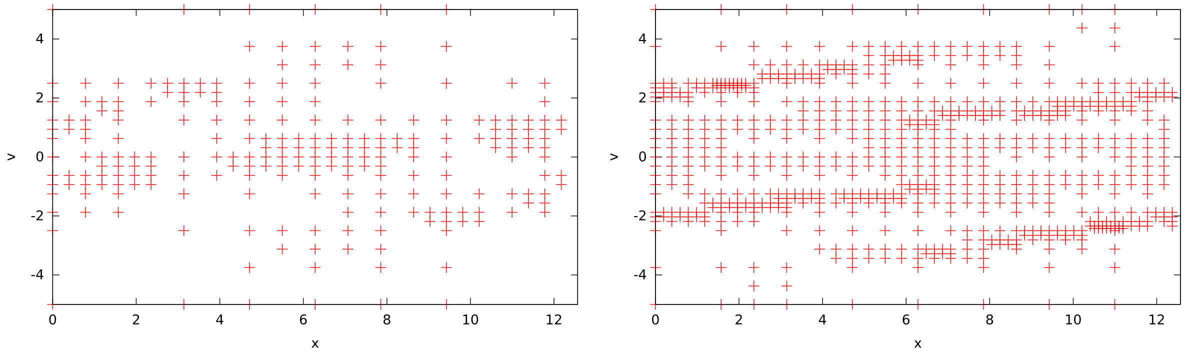
(c) Distribution function at t=8.

(d) Distribution function at t=20.

(e) Distribution function at t=50.

(f) Distribution function at t=80.

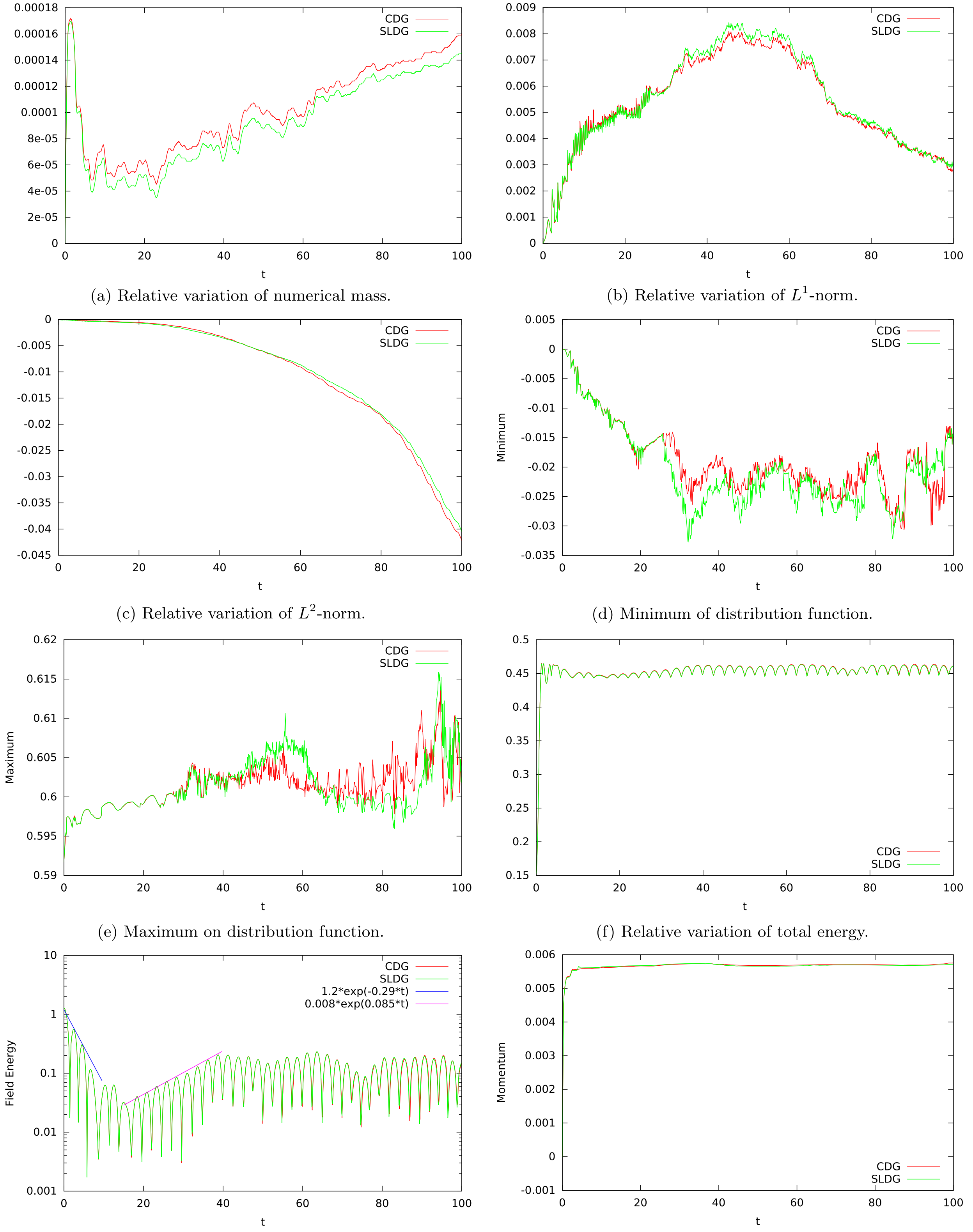
Fig. 13. Distribution function for strong Landau damping for the AMW-CDG scheme.



(a) Mesh at t=4.

(b) Mesh at t=8.

Fig. 14. Mesh for strong Landau damping for the AMW-CDG scheme.



(a) Relative variation of numerical mass.

(b) Relative variation of L^1-norm.

(c) Relative variation of L^2-norm.

(d) Minimum of distribution function.

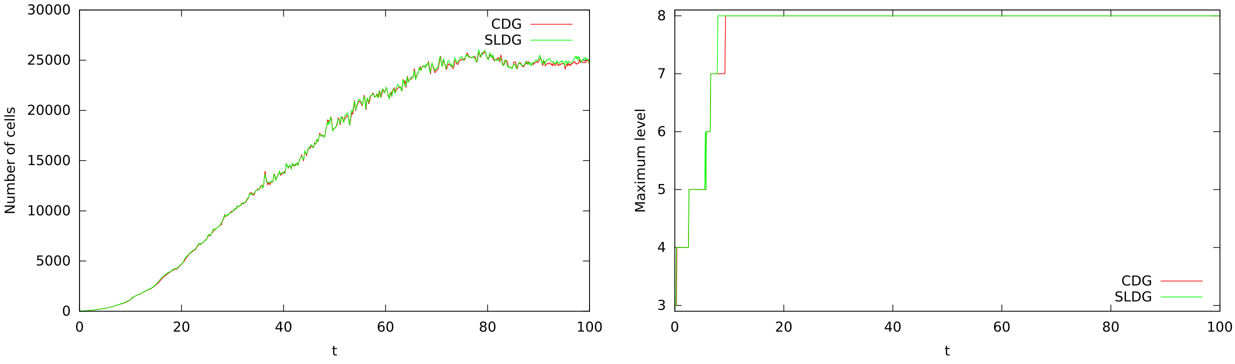
(e) Maximum on distribution function.

(f) Relative variation of total energy.

(g) Electric energy.

(h) Momentum.

Fig. 15. Relative variation of numerical mass, L^1-norm, L^2-norm and of total energy for strong Landau damping. Variation of minimum, maximum, momentum and electric field for strong Landau damping.



(a) Number of cells.

(b) Maximum level of refinement.

Fig. 16. Number of cells and level of refinement for strong Landau damping.

Fig. 16 displays the evolution of the number of cells and of the maximum level of refinement. If the maximum level is reached very fast, the number of cells increases for a much longer time as the refinement is local and small details spread among the domain.

7.5. Two stream instability

We consider the initial condition

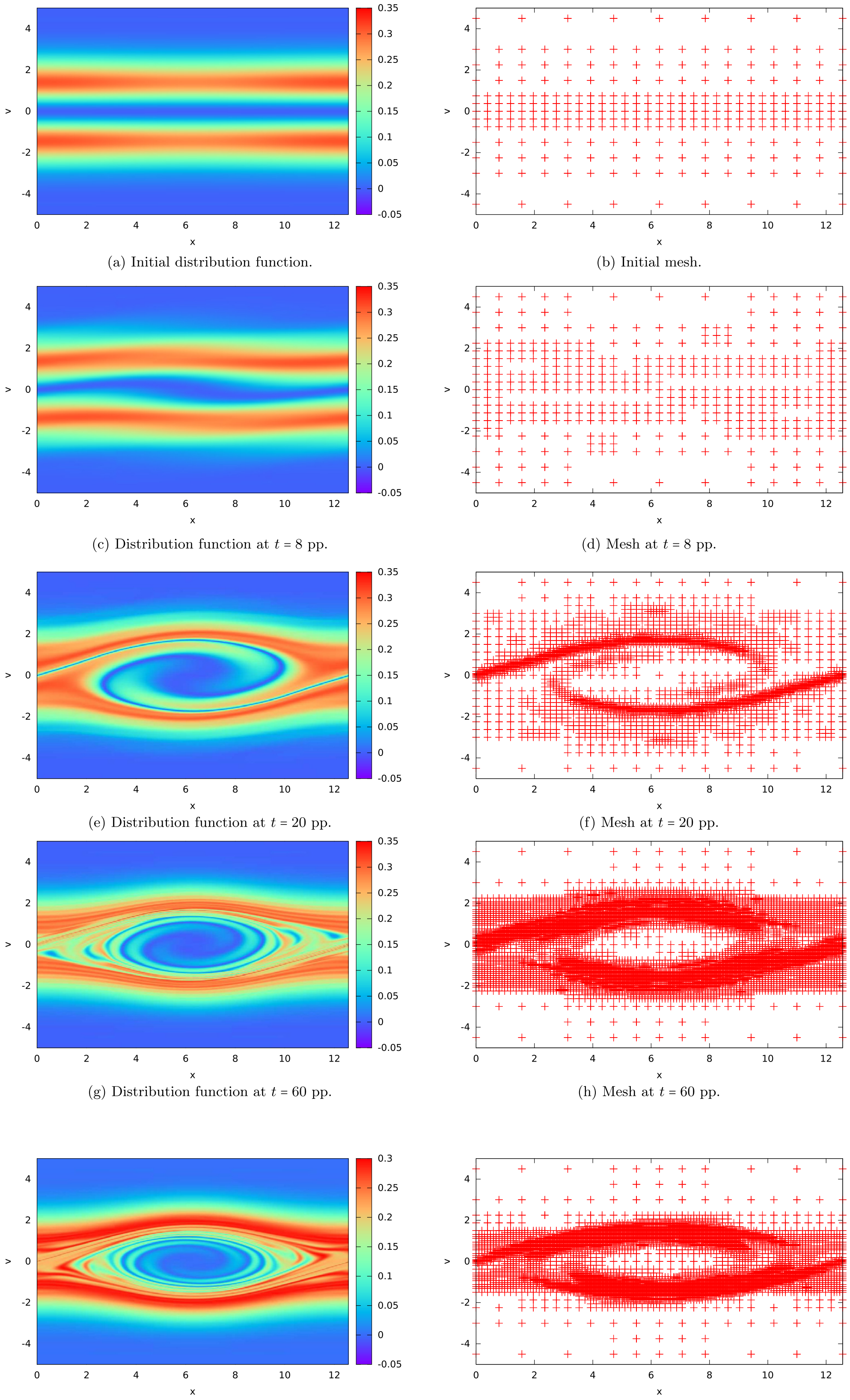
$$f_0(x,v) = \frac{v^2}{\sqrt{2\pi}}(1+0.05\cos(0.5x))\exp\left(\frac{-v^2}{2}\right), \quad (x,v) \in [0,4\pi] \times [-6,6]. \tag{67}$$

The final time is still 100 with 10^3 time steps for a time step $\Delta t = 0.1$. Maximum level of refinement still is 8 and our polynomials still are of degree up to 2. The threshold is $\epsilon_0 = 0.003$. This initial condition and the domain for this test case are the same as in [31].

Fig. 17 illustrates the distribution function and the mesh at different times. Those figures are obtained using the AMW-SLDG scheme but distribution function obtained with the AMW-CDG scheme can not be distinguished and meshes are very similar. Here we can clearly see that the mesh follows details of the distribution function during the vortex creation and on longer time.

Fig. 18 presents evolution of numerical mass, L^1-norm, L^2-norm, minimum of the distribution function, total energy and momentum. Positivity is not preserved, but numerical mass is well preserved since its relative variation is of order 10^{-5}. Relative variation of energy is at most of 0.7%, and absolute time variation of momentum is less than 10^{-5} (starting from null initial momentum). The minimum of the distribution function knows a peak to -0.0045 near $t = 25$ plasma period. Compared to the initial maximum of the distribution function (about 0.31), this is a bit more than 1.5%. Although it is not presented here, the maximum has a peak at the same time. This may be caused by Runge phenomena where we try to describe some variations in the phase-space that have high amplitude over short time. On Fig. 18, the relative variation of minimum is the variation of the minimum compared with the initial maximum of the distribution function.

Fig. 19 depicts the distribution function as a function of particular energy $E(t) = v^2/2 - \phi(t)$ at final time with both schemes. We see that a stationary state of BGK type has not been reached at the end of our simulation ($t = 100$) since the distribution function is not a single-valued function of the energy E (especially for small values of E). We also observe that until time $t = 100$ numerical dissipation is very small (see the decrease of L^2-norm on Fig. 18, less than 2%). This weak diffusion in terms of L^2-norm reflects well the power of multi-wavelets representation in the Lebesgue space L^2, because, contrary to some usual interpolations schemes, such representation controls and minimises the loss of L^2-norm by choosing locally in an optimal way the details which deserve to be conserved on discarded. To see the effect of numerical dissipation we look at the behaviour of the solution on longer time. Using the same time step $\Delta t = 0.1$ and a coarser description in space (polynomials of degree two, 8 levels of refinement, $\epsilon_0 = 0.03$) we perform some simulation up to $t = 210$. Fig. 20 shows the distribution function and its contour plot at final time obtained with the AMW-SLDG scheme. The results obtained with the AMW-CDG scheme are extremely similar. On this figure we can see the impact of the coarsened mesh on the smoothed distribution function. The corresponding plots of the distribution function as a function of particular energy are on Fig. 21. At this point we clearly have a single-valued function of the energy E. Therefore it seems that at time $t = 210$, the solution has reached a stationary state of BGK type. Moreover, beyond time $t = 100$ and before time $t = 210$ we observe greater numerical diffusion, almost 4%, as presented on Fig. 22. Our conjecture is that the numerical diffusion might play the part of the physical mechanism which promotes the relaxation of the solution towards a stationary state of BGK type. Actually we can not say if the convergence towards such stationary state of BKG type comes from a true physical relaxation process or if it is a numerical diffusion effect, because at this stage of the simulation, numerical diffusion — in term of L^2-norm — is not negligible.



(a) Initial distribution function.

(b) Initial mesh.

(c) Distribution function at $t = 8$ pp.

(d) Mesh at $t = 8$ pp.

(e) Distribution function at $t = 20$ pp.

(f) Mesh at $t = 20$ pp.

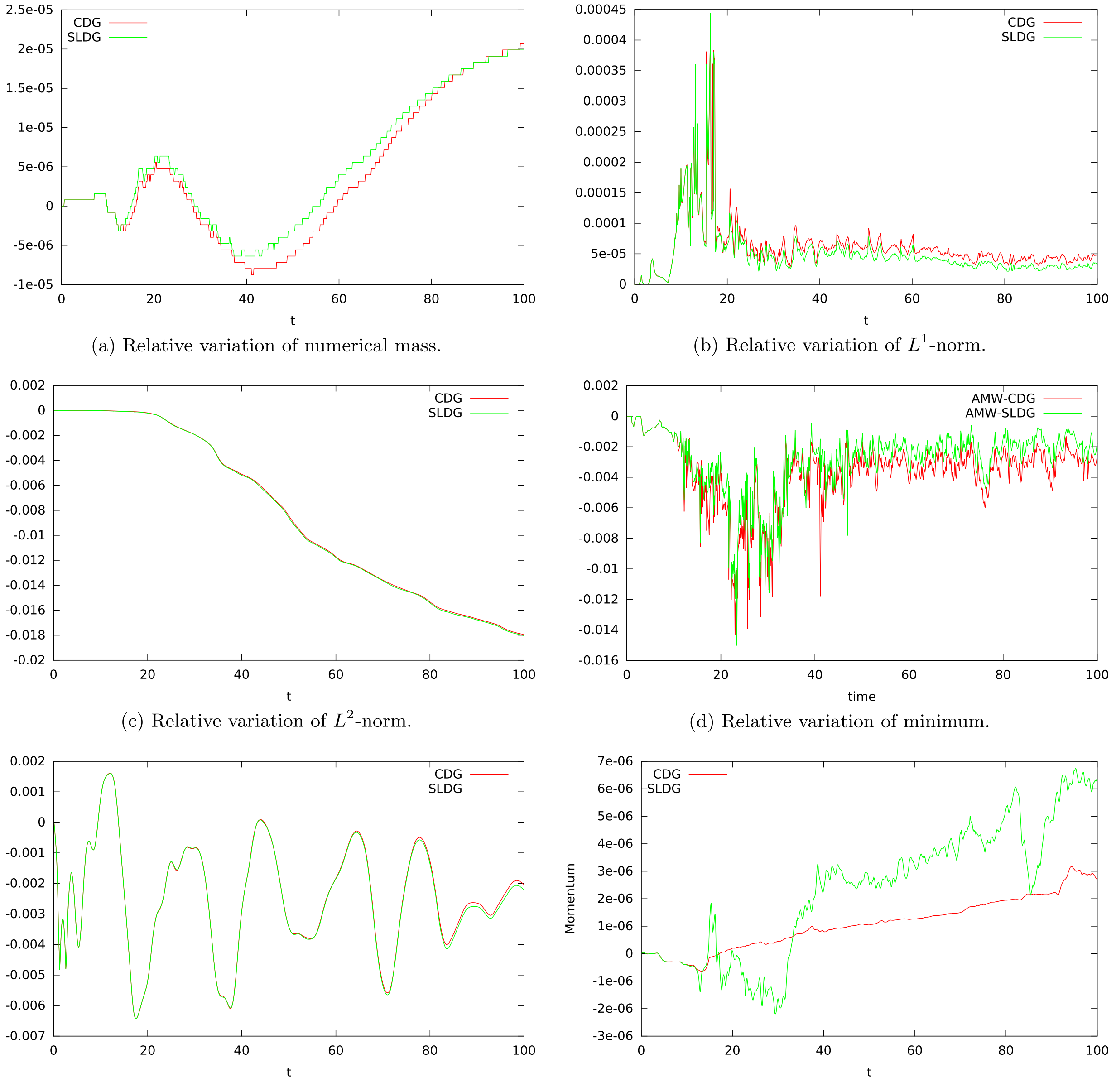
(g) Distribution function at $t = 60$ pp.

(h) Mesh at $t = 60$ pp.

(i) Distribution function at $t = 100$ pp.

(j) Mesh at $t = 100$ pp.

Fig. 17. Distribution function and Mesh for two stream instability for the AMW-SLDG scheme (pp = plasma period).



(a) Relative variation of numerical mass.

(b) Relative variation of L^1-norm.

(c) Relative variation of L^2-norm.

(d) Relative variation of minimum.

(e) Relative variation of total energy.

(f) Absolute variation of momentum.

Fig. 18. Evolution of numerical mass, L^1-norm, L^2-norm, minimum of distribution function, total energy and momentum for two stream instability.

7.6. Bump on tail

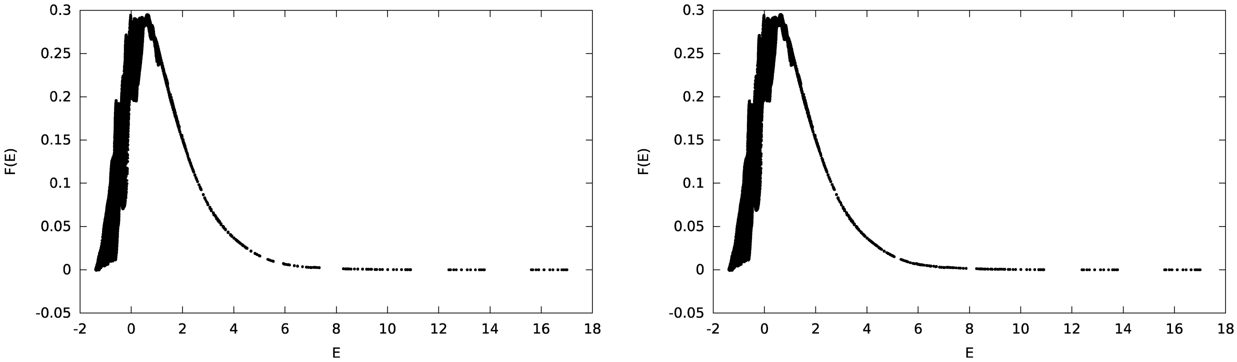
We consider the initial condition

$$f_0(x,v) = \frac{(1+0.04\cos(0.5x))}{10\sqrt{2\pi}}\left(9\exp\left(\frac{-v^2}{2}\right)+2\exp\left(-(v-4.5)^2\right)\right), \quad (x,v)\in[0,20\pi]\times[-9,9]. \tag{68}$$

Final time is 300 with 3×10^3 time steps for a time step $\Delta t = 0.1$. Maximum level of refinement is 8 and polynomial degree is 2. Threshold is $\epsilon_0 = 3\times10^{-3}$.

Fig. 23 presents the distribution function and mesh obtained with the method AMW-SLDG. The method AMW-CDG gives extremely similar results. Here we can see initial mesh slightly refined on the two beams. Along time it follows and matches vortices during their formation, existence, and during their merging.

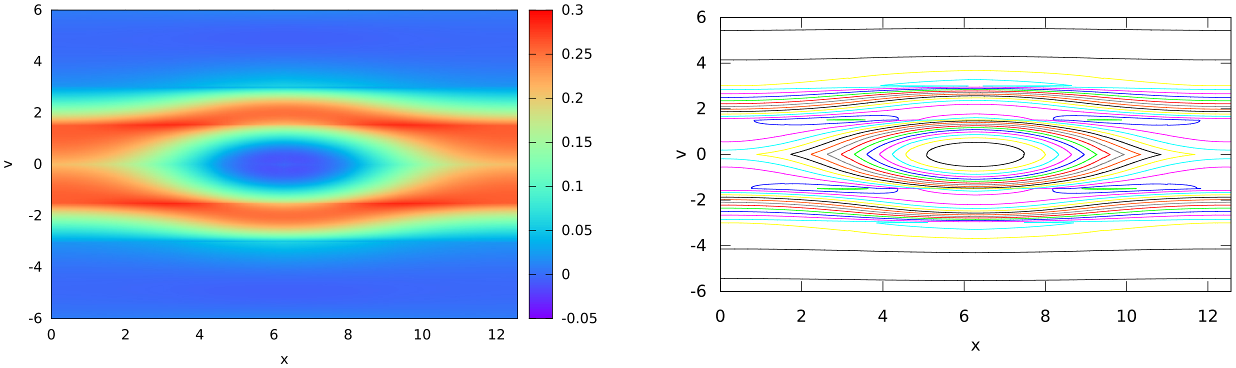
Fig. 24 shows the number of cells and finest level of computation during simulation. During creation and existence of vortices the mesh is locally refined to its maximum and the number of cells increases.



(a) With the AMW-SLDG scheme.

(b) With the AMW-CG scheme.

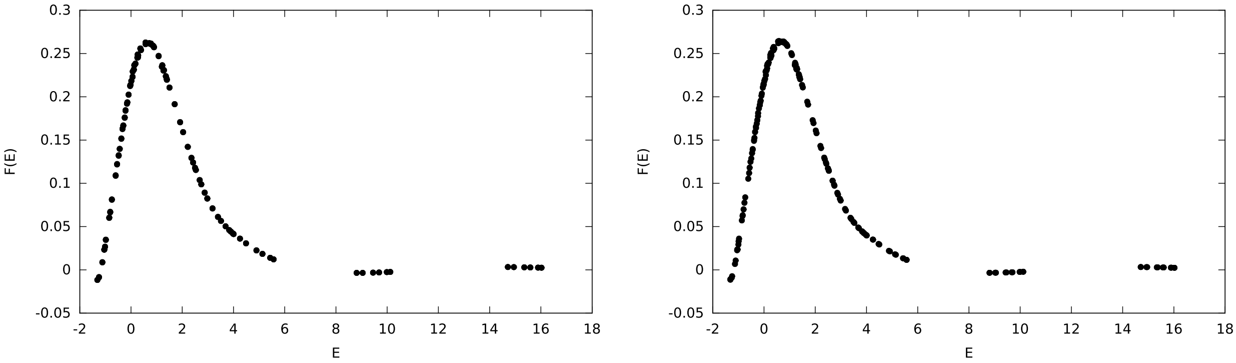
Fig. 19. Distribution function as a function of particular energy at time $T_f = 100$.



(a) Distribution function at time $t = 210$.

(b) Contour plot at time $t = 210$.

Fig. 20. Distribution function for two stream instability with the AMW-SLDG scheme at time $t = 210$.



(a) With the AMW-SLDG scheme.

(b) With the AMW-CG scheme.

Fig. 21. Distribution function as a function of particular energy at time $T = 210$.

The relative variations of mass, of L^1-norm, L^2-norm and of total energy are presented on Fig. 25. The four quantities are very well conserved compared to some results presented in [31,9]. Relative variations of mass and of L^1-norm show a difference between the two schemes, especially during the merging of vortices.

7.7. Polar-like test cases

This part is devoted to experiment considering the following formulation of Vlasov equation from [25] and [24]

$$\partial_t f(r,v,t) + \partial_r \left(\frac{v}{\varepsilon} f(r,v,t)\right) + \partial_v \left((E_\varepsilon(r,t) + \Xi_\varepsilon(r,t))\, f(r,v,t)\right) = 0, \tag{69}$$

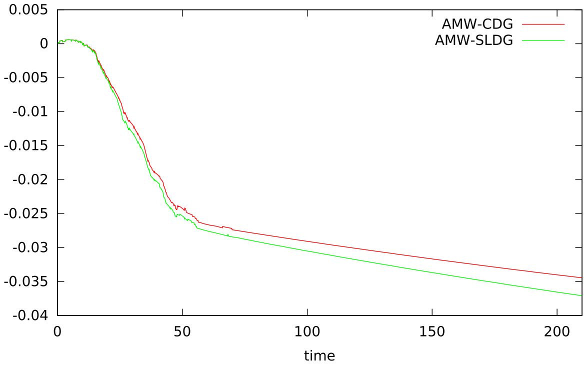


Fig. 22. Relative variation of the L^2-norm for the two stream instability with $\epsilon_0 = 0.03$.

where the constant ε and fields E and Ξ will be defined for each test case. Under this formulation, the corresponding Poisson equation is

$$\frac{1}{r}\partial_r(rE_\varepsilon(r,t)) = \rho(r,t). \tag{70}$$

The boundary condition for equation (70) is $E_\varepsilon(0,t) = 0$. This is necessary for a solution to exist due to factor $1/r$ in the equation.

7.7.1. Focusing beam

For this case $\varepsilon = 0.01$, E_ε is obtained by solving (70), and $\Xi_\varepsilon(r,t) = -r/\varepsilon$. The initial distribution is

$$f_0(r,v) = \frac{1}{\sqrt{2\pi}\,v_{th}}\exp\left(-\frac{v^2}{2v_{th}^2}\right)\mathbb{1}_{[-0.75,0.75]}(r), \quad (r,v)\in[-1,1]\times[-1.2,1.2], \tag{71}$$

with $v_{th} = 0.0727518214392$ and

$$\mathbb{1}_{[a,b]}(r) = \begin{cases} 0 & \text{if } r < a \text{ or } r > b, \\ 1 & \text{else.} \end{cases}$$

The final time is $T = 100$ with 10^3 time steps for a time step $\Delta t = 0.1$. We use polynomials of degree up to 2. Maximum level of refinement is 8 and threshold $\epsilon_0 = 3\times 10^{-3}$.

Fig. 26 illustrates the distribution function and the mesh obtained with the method AMW-CDG. The method AMW-SLDG gives results that can not be distinguished. In this test case, we can see the formation of a central vortex with two coiling filaments. Fig. 27 presents the relative variations of numerical mass, L^1-norm and L^2-norm. Here again the variation of mass and of L^1-norm indicate that the distribution function has some negative values.

7.7.2. Focusing channel

For this test case $\varepsilon = 0.1$, E_ε is computed using (70) and

$$\Xi_\varepsilon(r,t) = r\left(\frac{-1}{\varepsilon} + \cos^2\left(\frac{t}{\varepsilon}\right)\right).$$

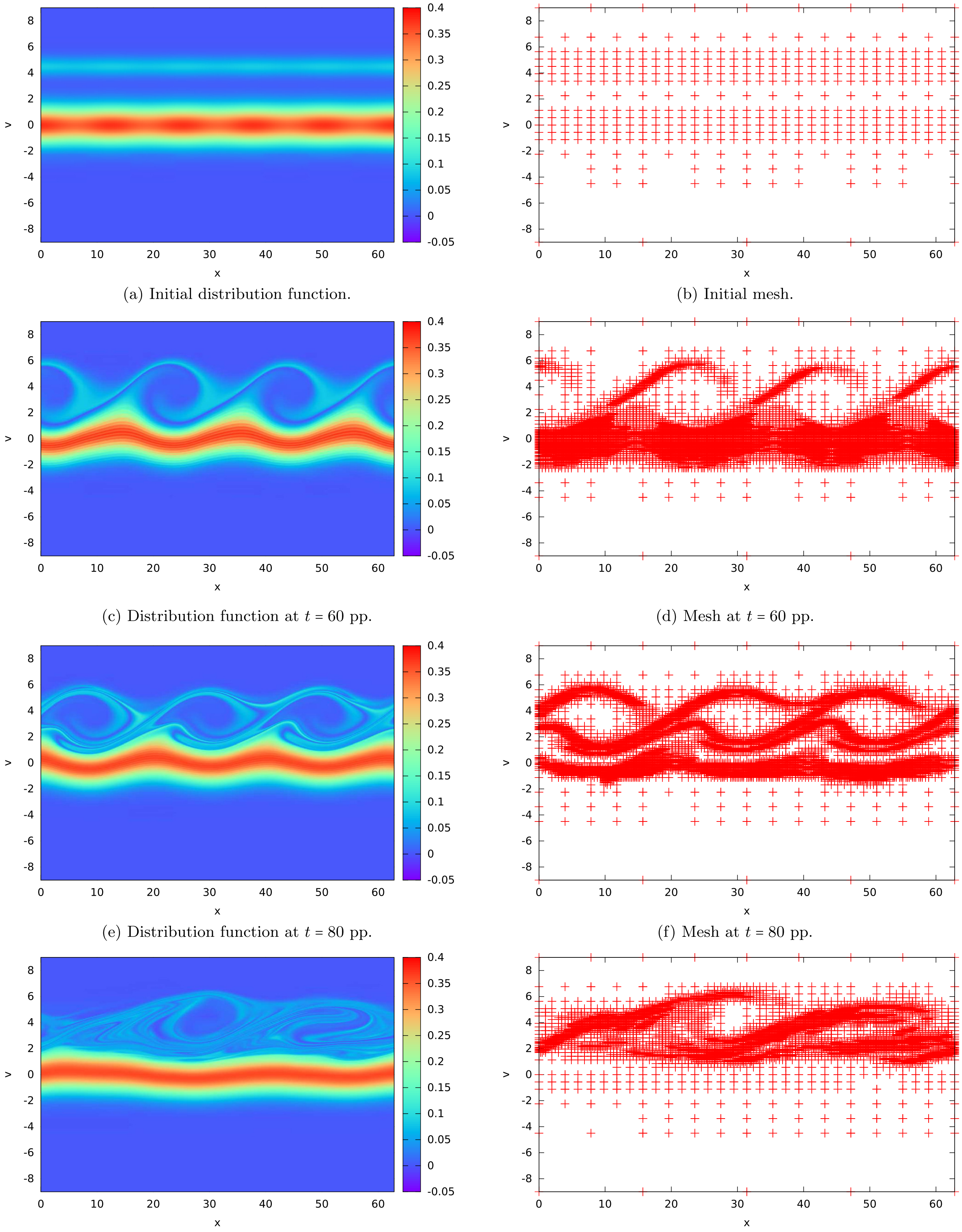
The initial distribution function is

$$f_0(r,v) = \frac{3}{4v_{th}}\exp\left(\frac{-v^2}{2v_{th}^2}\right)\mathbb{1}_{[-1.83271471003,1.83271471003]}(r), \quad (r,v)\in[-3,3]^2, \tag{72}$$

with $v_{th} = 0.0727518214392$. This test case is detailed in [25].

The final time is $T = 100$ with 5×10^3 time steps for a time step $\Delta t = 0.02$. We use polynomials of degree up to 2. The maximum level of refinement is 8 and the threshold is $\epsilon_0 = 10^{-2}$.

Fig. 28 shows snapshots of the distribution function and the mesh obtained with the AMW-SLDG scheme. We obtain very similar result using the AMW-CDG scheme. The mesh accurately follows the filamentation of the distribution function.



(a) Initial distribution function.

(b) Initial mesh.

(c) Distribution function at $t = 60$ pp.

(d) Mesh at $t = 60$ pp.

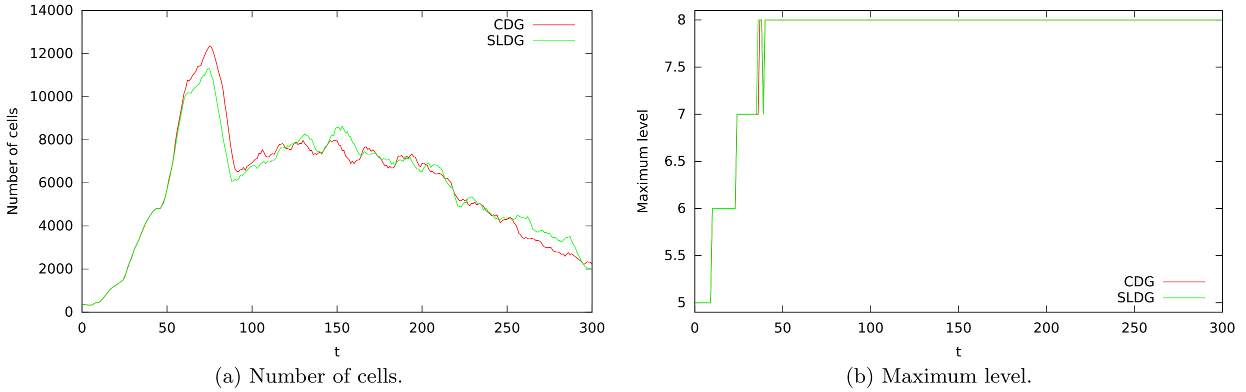
(e) Distribution function at $t = 80$ pp.

(f) Mesh at $t = 80$ pp.

(g) Distribution function at $t = 200$ pp.

(h) Mesh at $t = 200$ pp.

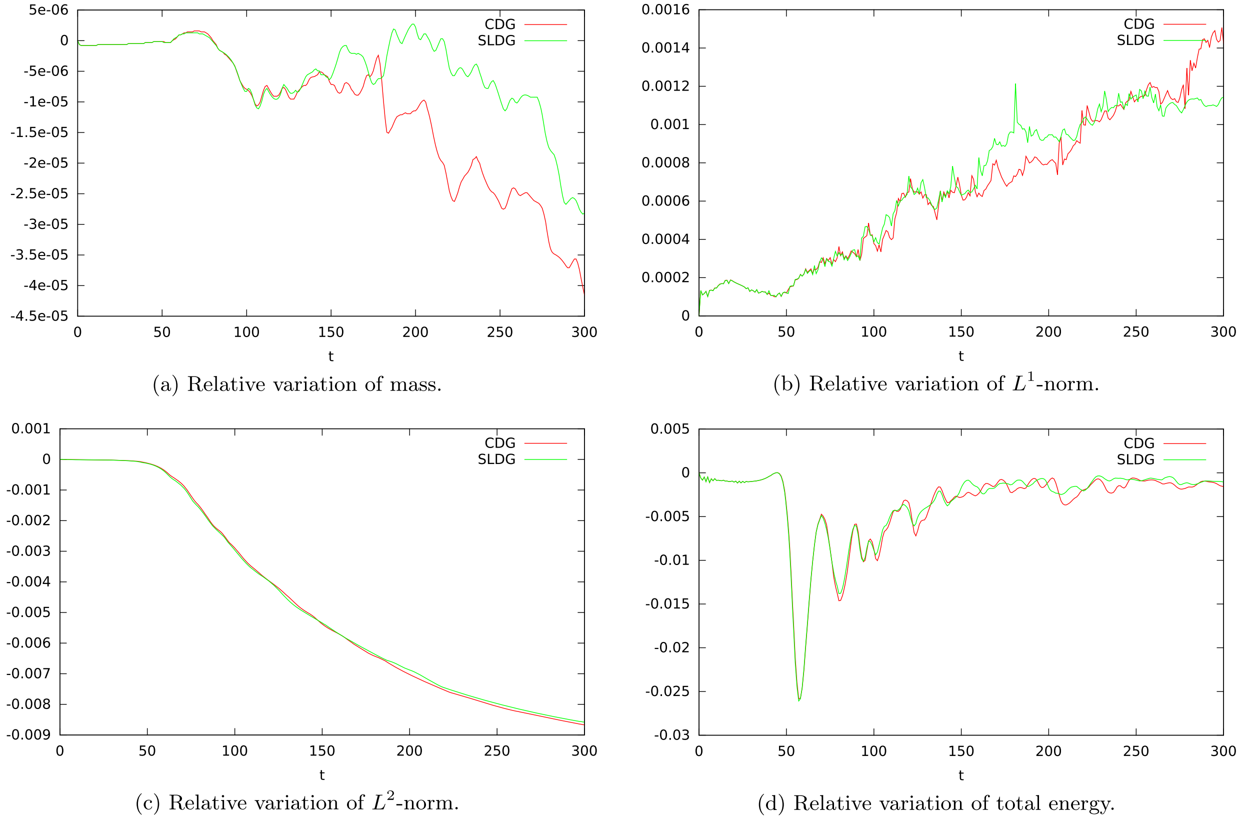
Fig. 23. Distribution function and Mesh for bump on tail with the AMW-SLDG scheme.



(a) Number of cells.

(b) Maximum level.

Fig. 24. Number of cells and maximum level for computation for bump on tail.



(a) Relative variation of mass.

(b) Relative variation of L^1-norm.

(c) Relative variation of L^2-norm.

(d) Relative variation of total energy.

Fig. 25. Variation of mass, L^1-norm, L^2-norm and total energy for Bump on tail.

7.8. Astrophysics test cases

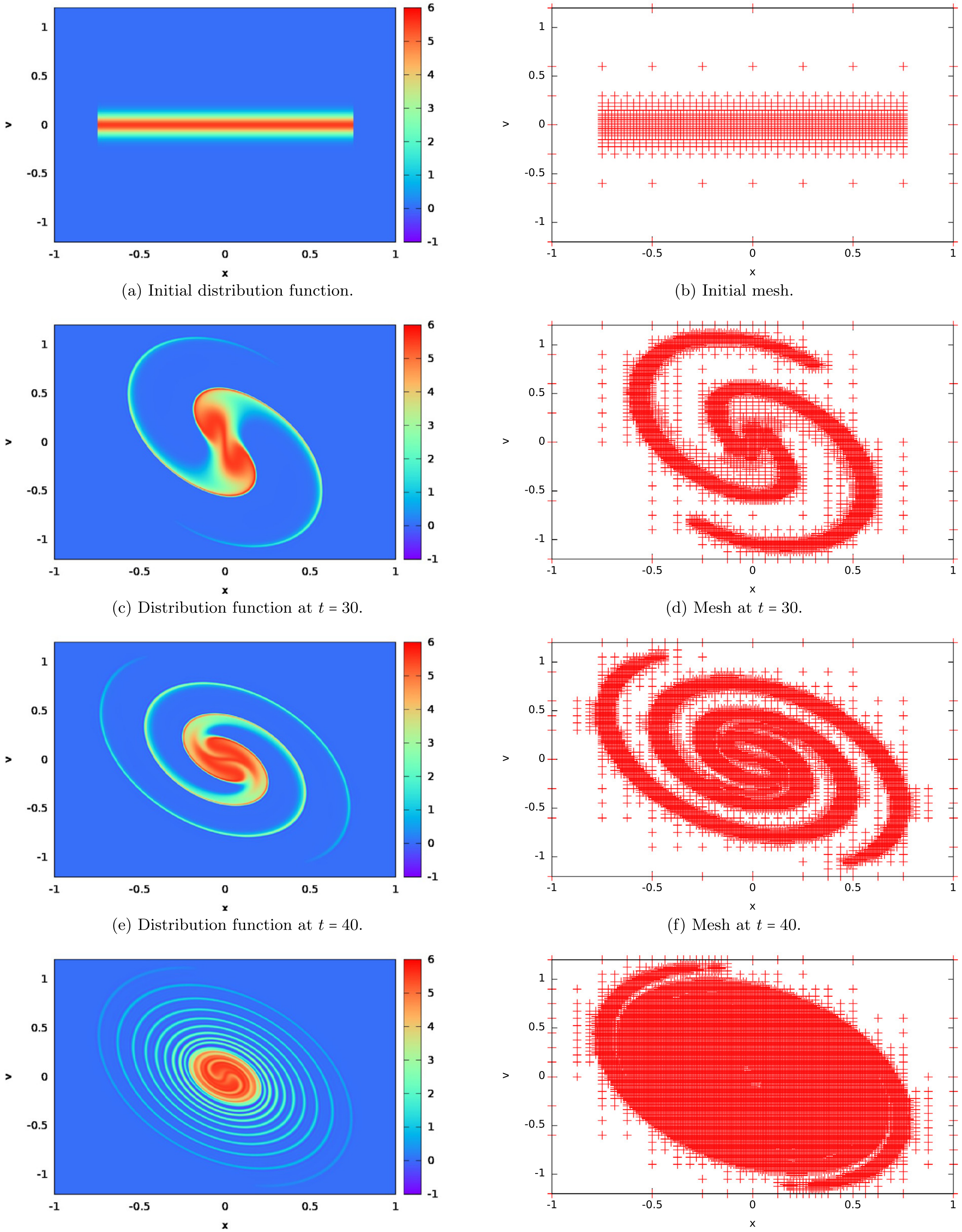
In astrophysics, (1) becomes

$$\begin{cases} \partial_t f(x,v,t) + \partial_x(vf(x,v,t)) + \partial_v(E(x,t)f(x,v,t)) = 0, & x \in \Omega_x,\ v \in \mathbb{R}, \\ f(x,v,0) = f_0(x,v), \end{cases} \tag{73}$$

while (2) becomes

$$\Delta\Phi(x,t) = 4\pi(\rho(x,t) - 1), \quad x \in \Omega_x \tag{74a}$$

$$E(x,t) = -\partial_x\Phi(x,t). \tag{74b}$$



(a) Initial distribution function.

(b) Initial mesh.

(c) Distribution function at $t = 30$.

(d) Mesh at $t = 30$.

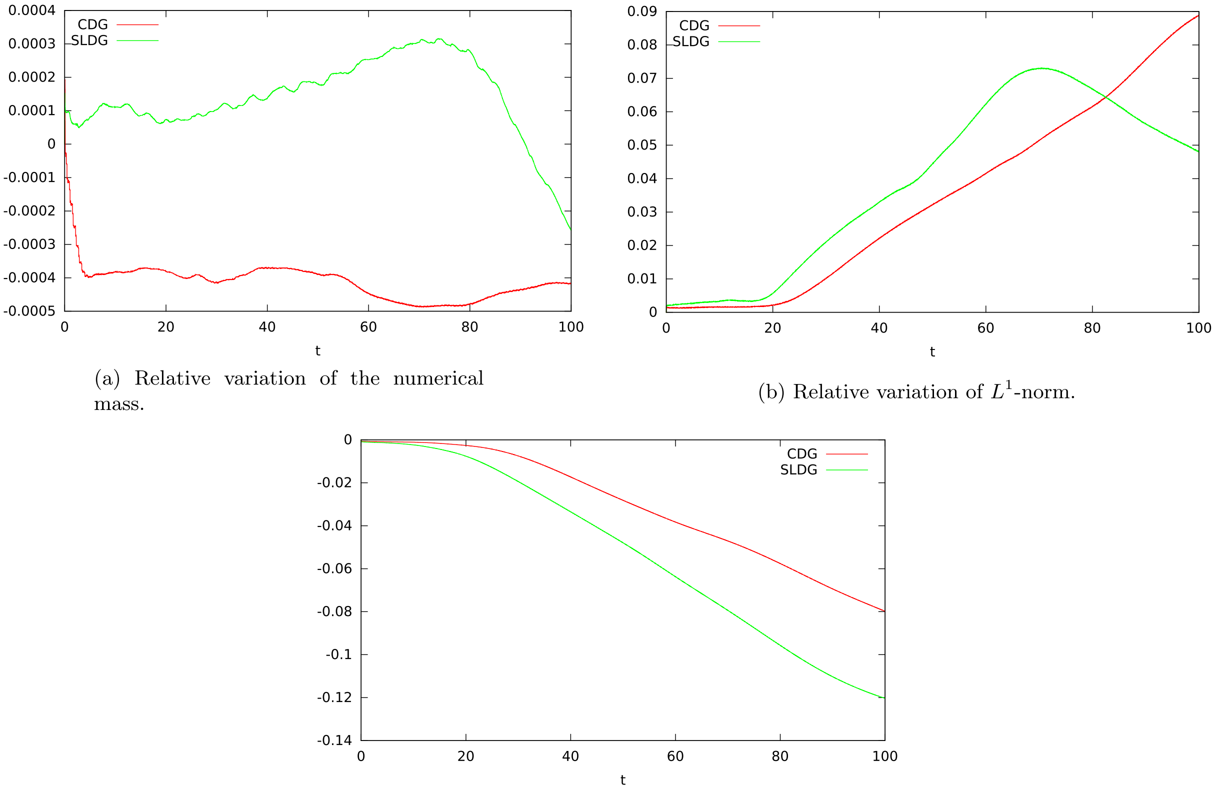
(e) Distribution function at $t = 40$.

(f) Mesh at $t = 40$.

(g) Distribution function at $t = 100$.

(h) Mesh at $t = 100$.

Fig. 26. Distribution function and mesh for focusing beam with the AMW-CDG scheme.



(a) Relative variation of the numerical mass.

(b) Relative variation of L^1-norm.

(c) Relative variation of L^2-norm.

Fig. 27. Relative variations of numerical mass, L^1-norm, L^2-norm for focusing beam.

Here, the field E is gravity field. Note that we still add the constant -1 in Poisson equation for periodicity. The first simulation of Vlasov–Poisson for gravitational case using discontinuous Galerkin method was done in [11].

7.8.1. Cold layer

We consider initial condition

$$f_0(x,v) = \frac{1}{0.15\sqrt{2\pi}} \exp\left(-\frac{(v-u(x))^2}{2\times 0.15^2}\right), \quad (x,v)\in[0,2\pi]\times[-10,10], \tag{75}$$

with

$$u(x) = 0.01\sin(x).$$

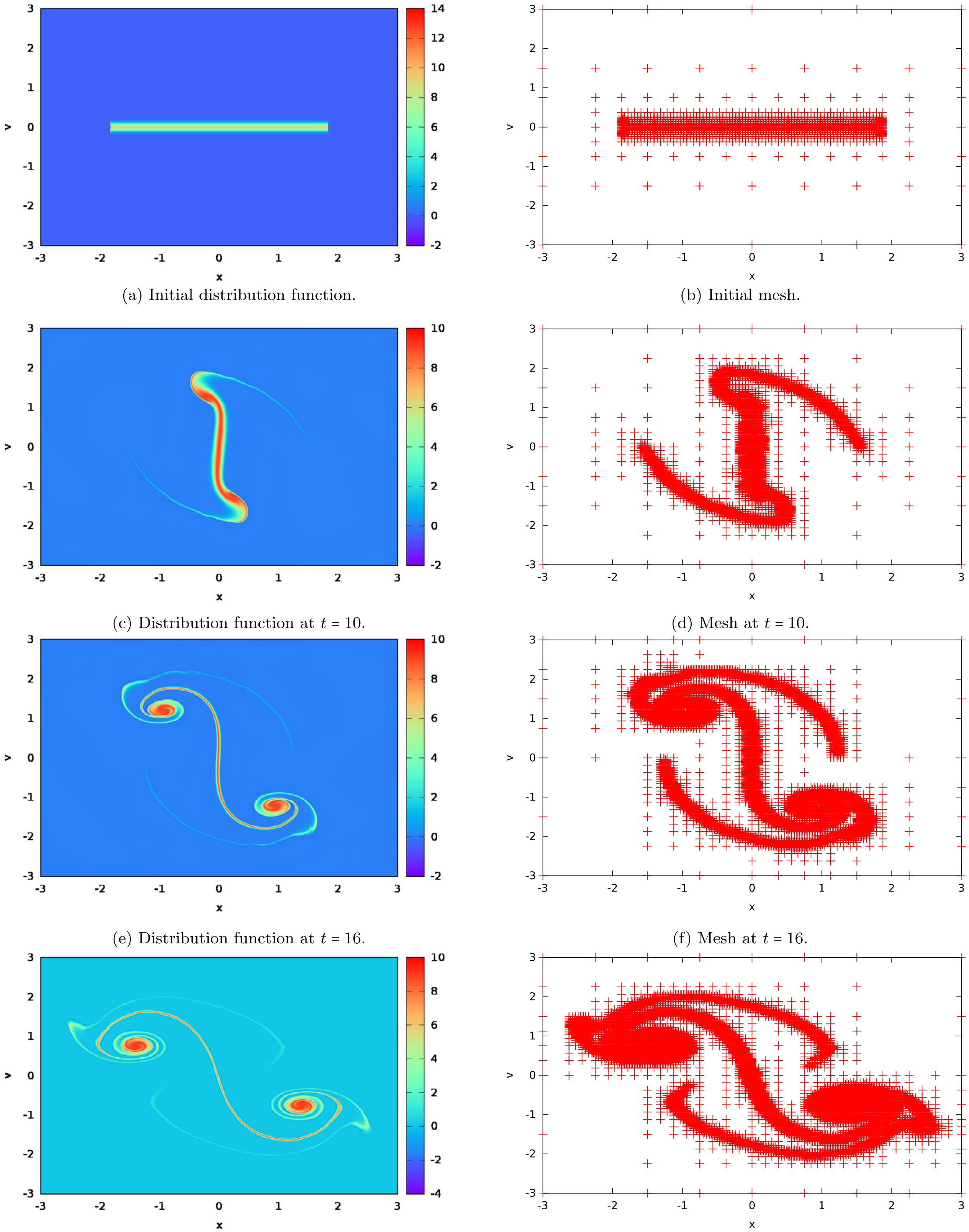
The final time is 40 with 2×10^3 time steps for a time step $\Delta t = 0.02$. We use polynomials of degree 2. The maximum level of refinement is 8 and the threshold is $\epsilon_0 = 3\times 10^{-3}$. This test case is designed to always have cells on the finest level of refinement.

Fig. 29 depicts the evolution of the distribution function and the mesh with the initial condition and the coiling of filaments at $t=2$, $t=4$ and $t=6$ with the AMW-CDG scheme. Finally, our mesh becomes coarse and dissipation of filaments at time $t=14$ and $t=20$ appears. Here again the distribution functions from the two schemes (AMW-SLDG and AMW-CDG) can not be distinguished. Periodicity is an artefact that we have introduced in our code and we can see its side effect at $t=6$ on boundaries $x=0$ and $x=2\pi$, and later on the simulation. The next test case 7.8.2 will be adapted to avoid this numerical artefact. We see filaments that go under $x=0$ or above $x=2\pi$ and appear on the other side. This is a non-physical effect but a spurious result of periodicity. The filamentation and dissipation processes can be seen on Fig. 30 too. Filamentation occurs between times $t=2$ and $t=5$ and the number of cells increases a lot during this period. After time $t=12$ dissipation becomes significant, details are lost and number of cells starts to decrease. The solution is not physical any-more.

7.8.2. Gaussian initial condition

We consider the Gaussian initial condition from Sec. 2.1.1 from [17]:

$$f_0(x,v) = 4\exp\left(-\frac{(x^2+v^2)}{0.08}\right), \quad (x,v)\in[-2,2]^2 \tag{76}$$



(a) Initial distribution function.

(b) Initial mesh.

(c) Distribution function at $t = 10$.

(d) Mesh at $t = 10$.

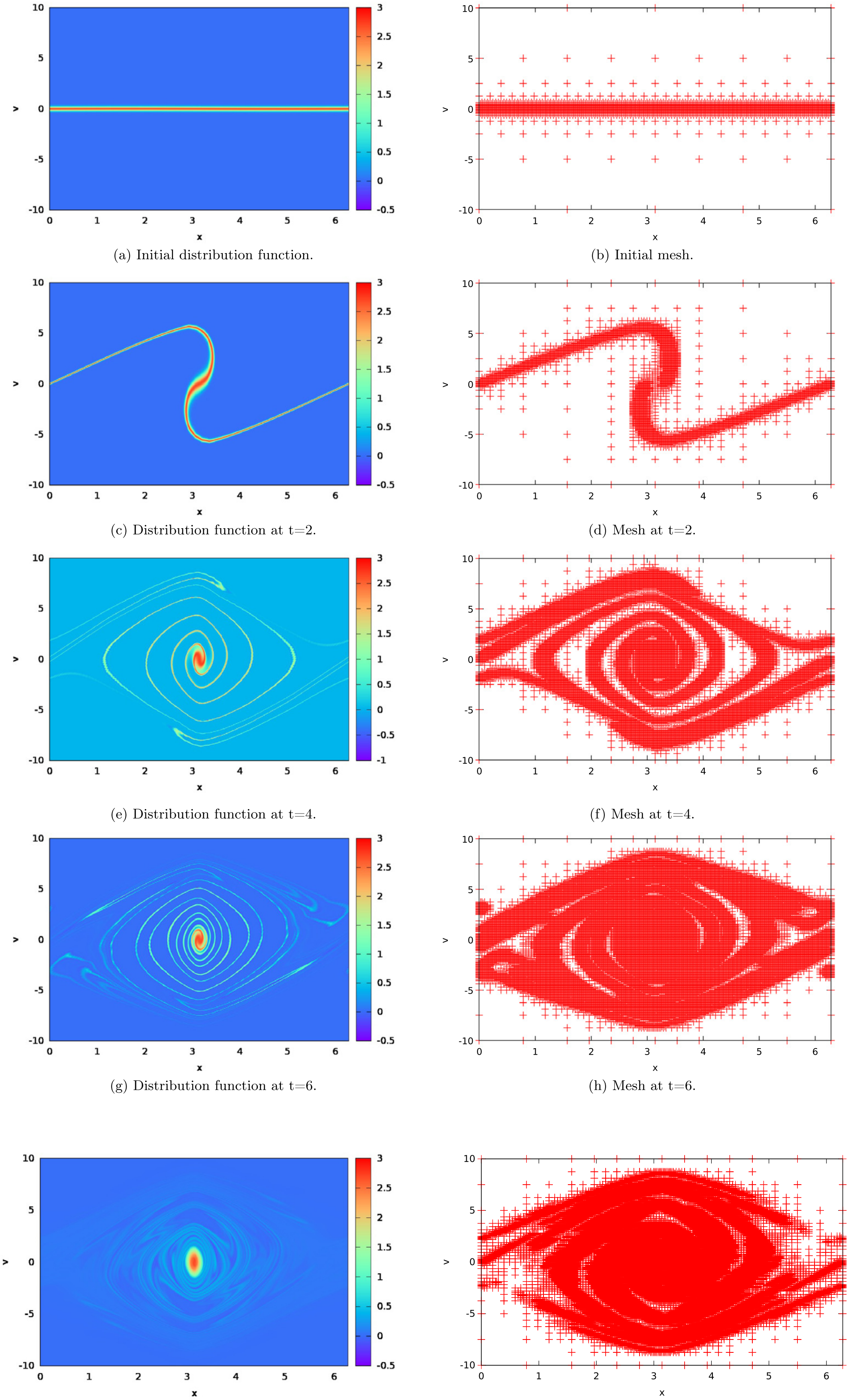
(e) Distribution function at $t = 16$.

(f) Mesh at $t = 16$.

(g) Distribution function at $t = 20$.

(h) Mesh at $t = 20$.

Fig. 28. Distribution function and mesh for Focusing channel with the AMW-SLDG scheme.



(a) Initial distribution function.

(b) Initial mesh.

(c) Distribution function at t=2.

(d) Mesh at t=2.

(e) Distribution function at t=4.

(f) Mesh at t=4.

(g) Distribution function at t=6.

(h) Mesh at t=6.

(i) Distribution function at t=20.

(j) Mesh at t=20.

Fig. 29. Distribution function and mesh for initial conditions (75) and during simulation with the AMW-CDG scheme.

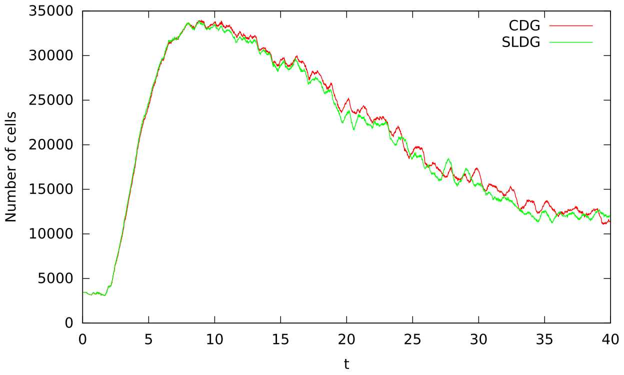


Fig. 30. Number of cells for initial condition (75).

with the proper normalisation to respect periodicity in Poisson equation. This test case is designed with open boundary conditions so we adapted our domain with their results to ensure an almost null distribution function everywhere on boundaries.

The final time is $T = 100$ with 10^3 time steps for a time step $\Delta t = 0.1$. We use polynomials of degree up to 3. The maximum level of refinement is 9 and the threshold is $\epsilon_0 = 3 \times 10^{-3}$.

Fig. 31 displays the evolution of the distribution function and the mesh obtained with the method AMW-SLDG. Results from the method AMW-CDG are extremely similar. The simulation is performed on $[-2, 2]^2$ to enforce Dirichlet boundary conditions, but because nothing happen outside of $[-1, 1]$, we only present a zoom on $[-1, 1]$. Because it is nearly impossible to distinguish filaments on long time, Fig. 32 is a contour plot to highlight the very thin filaments at final time. We clearly see the very thin filaments coiling without diffusion, even thought the filaments are extremely close from each others. In this test case the domain is large enough so that no numerical artefacts due to periodicity occur. Fig. 33 shows the increase of number of cell as thin filaments appear and coil.

The main diagnostics are presented on Fig. 34. Figs. 34b and 34e show the correlation between the increase of the L^1-norm and the fall of the minimum. The decrease of the L^2-norm occurs at the same time. Step variation of mass is typical of amplified machine error variation.

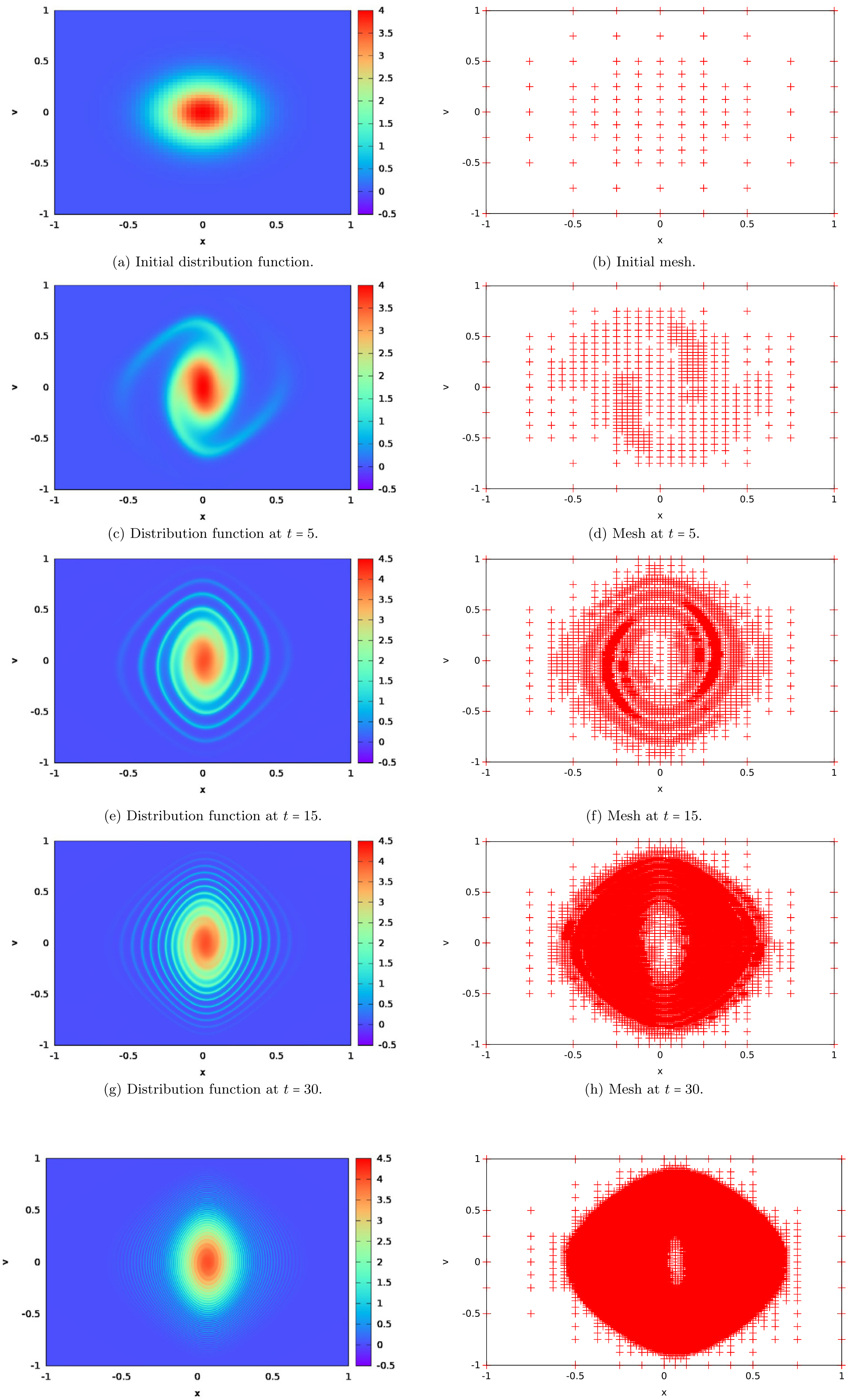
7.8.3. *Jeans instability*

This test case is based on astrophysics formulation of the Vlasov equation (73) but uses the same normalisation of the Poisson equation as plasma physics (2). The initial condition is

$$f(x, v, 0) = \frac{\exp\left(\frac{-v^2}{2}\right)}{\sqrt{2\pi}} (1 - A\cos(kx)), \quad (x, v) \in [0, 2\pi/k] \times [-V_c, V_c]. \tag{77}$$

We use the settings from [11]. We consider $A = 0.01$, $k = 0.8$ and $V_c = 6$. The final time is $T_f = 100$ with $1,000$ time steps, which gives $\Delta t = 0.1$. We use polynomials of degree 2 in each direction. The maximum level of refinement is 8 and the threshold parameter is $\epsilon_0 = 0.003$.

The distribution function and the mesh obtained with the AMW-CDG scheme are presented on Fig. 35. The results from the AMW-SLDG scheme are extremely similar. Variations of mass, L^1-norm, L^2-norm and momentum are displayed on Fig. 36. The two schemes (AMW-CDG and AMW-SLDG) give very similar results. Fig. 37 depicts the distribution function as a function of the particular energy $E = v^2/2 - \phi(x)$. This figure is very similar to Fig. 7a from [11]. As in the two stream instability, Sec. 7.5, for small and moderate value of the energy E, the distribution function f is not a single-valued function of E. Then a stationary state of BGK type is not reached yet. We still observe that numerical diffusion is very small until $t = 100$ (see Fig. 36c of L^2-norm evolution, the loss of relative L^2-norm is less than 0.5%). In order to see the effect of numerical diffusion at $T_f = 100$, we perform some simulations with the threshold parameter $\epsilon_0 = 0.01$ and 7 levels of refinement. The corresponding plots of the distribution function as a function of the particular energy are presented on Fig. 38. The corresponding distribution function at time $T_f = 100$ is displayed on Fig. 39. We observe on Fig. 38 that the distribution function f is a single-valued function of the energy E. Then a stationary state of BGK type seems to be reached. Moreover for such test-case we observe a greater numerical diffusion than in the first parameter setting (i.e. $\epsilon_0 = 0.003$, 8 levels of refinement); let us see Fig. 40 where we observe a loss of relative L^2-norm of 0.7%. Therefore, as for the two stream instability (see Sec. 7.5), we may conclude the same hypothesis about the role of numerical diffusion on the behaviour of the solution.



(a) Initial distribution function.

(b) Initial mesh.

(c) Distribution function at $t = 5$.

(d) Mesh at $t = 5$.

(e) Distribution function at $t = 15$.

(f) Mesh at $t = 15$.

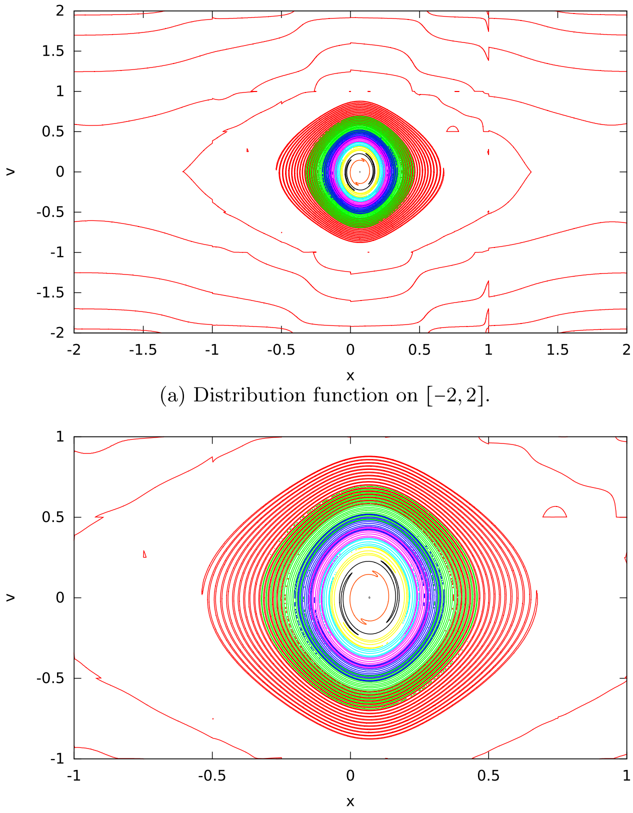
(g) Distribution function at $t = 30$.

(h) Mesh at $t = 30$.

(i) Distribution function at $t = 100$.

(j) Mesh at $t = 100$.

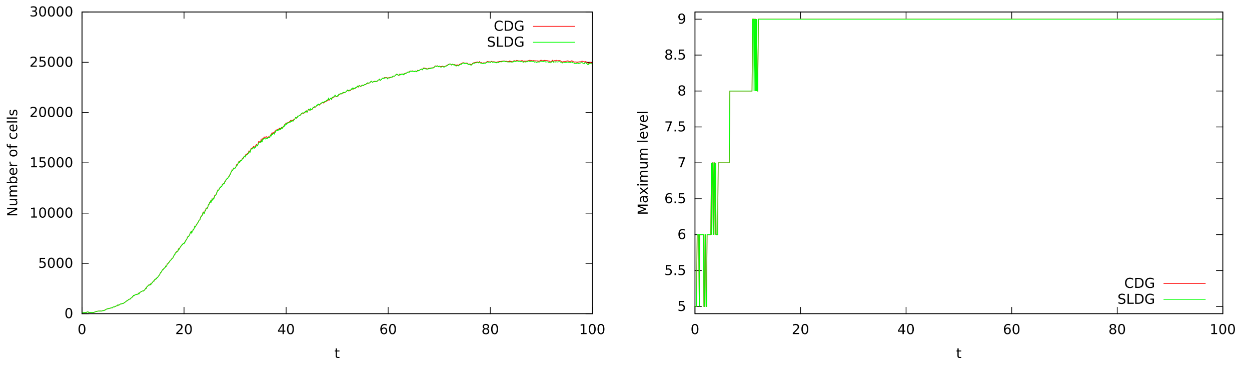
Fig. 31. Distribution function and mesh for Gaussian initial distribution function and evolution in astrophysics framework with the AMW-SLDG scheme.



(a) Distribution function on $[-2, 2]$.

(b) Distribution function on $[-1, 1]$.

Fig. 32. Distribution function contours at $t = 100$ for Gaussian initial distribution function with the AMW-SLDG scheme.



(a) Number of cells.

(b) Maximum level.

Fig. 33. Number of cells and maximum level for Gaussian initial condition.

7.9. *Extension to the four-dimensional problem*

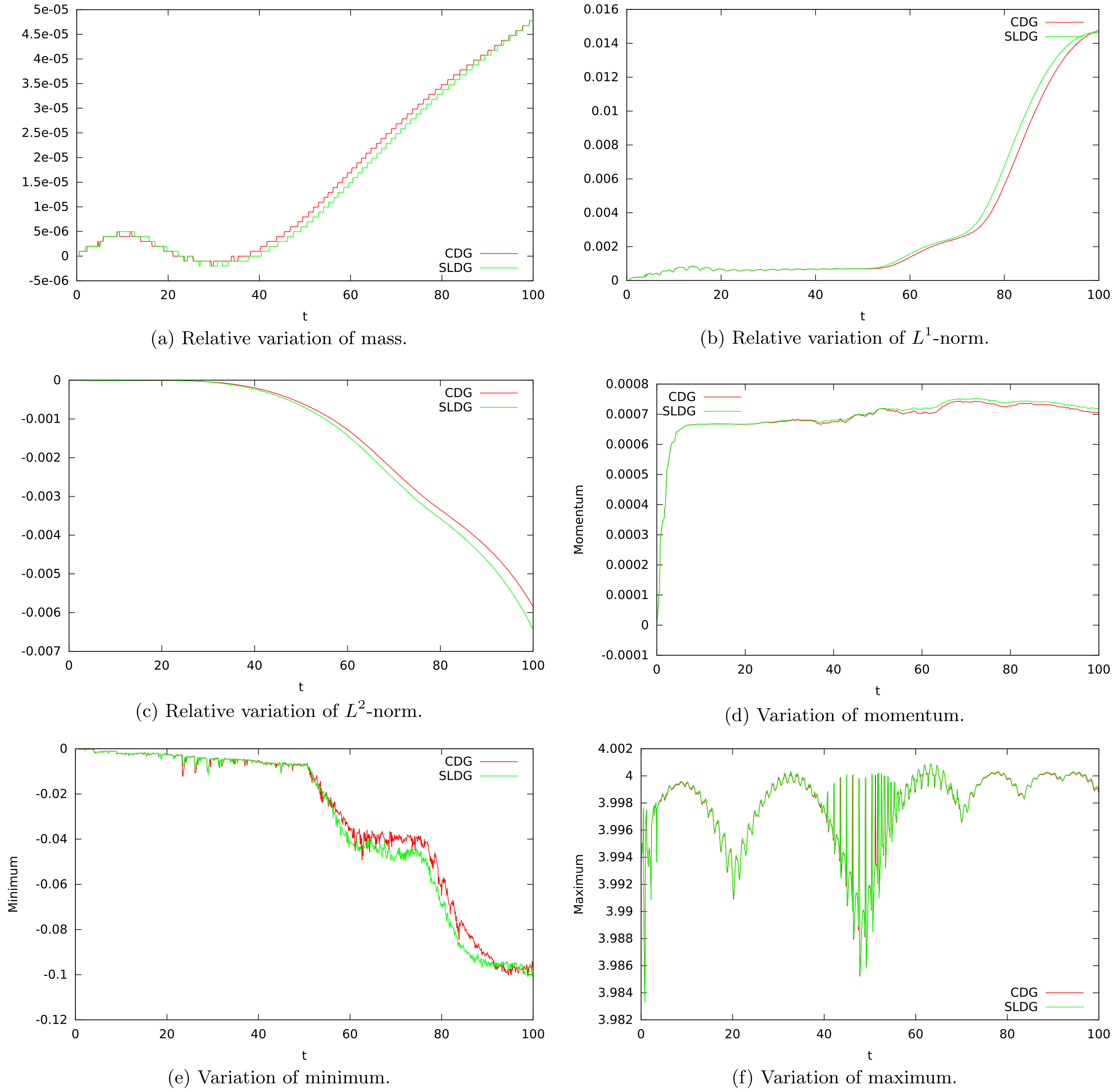
We extend the schemes to the four-dimensional problem. We still use directional splitting. Then the extension to four dimensions is just an addition of one dimension steps, as presented Secs. 3.2 and 3.3. Adaptivity is achieved by extended equations (40) to the four-dimensional problem.

We consider the equation

$$\partial_t f + v_x \partial_x f - v_y \partial_y f - x \partial_{v_x} f + y \partial_{v_y} f = 0, \tag{78}$$

with initial condition

$$f_0(x, y, v_x, v_y) = \exp\left(-(x+1)^2 - (y-3)^2 - v_x^2 - v_y^2\right) - \exp\left(-(x-1)^2 - (y+3)^2 - v_x^2 - v_y^2\right),$$



(a) Relative variation of mass.

(b) Relative variation of L^1-norm.

(c) Relative variation of L^2-norm.

(d) Variation of momentum.

(e) Variation of minimum.

(f) Variation of maximum.

Fig. 34. Variation of the main diagnostics for Gaussian distribution function.

$$(x, y, v_x, v_y) \in [-10, 10]^4. \tag{79}$$

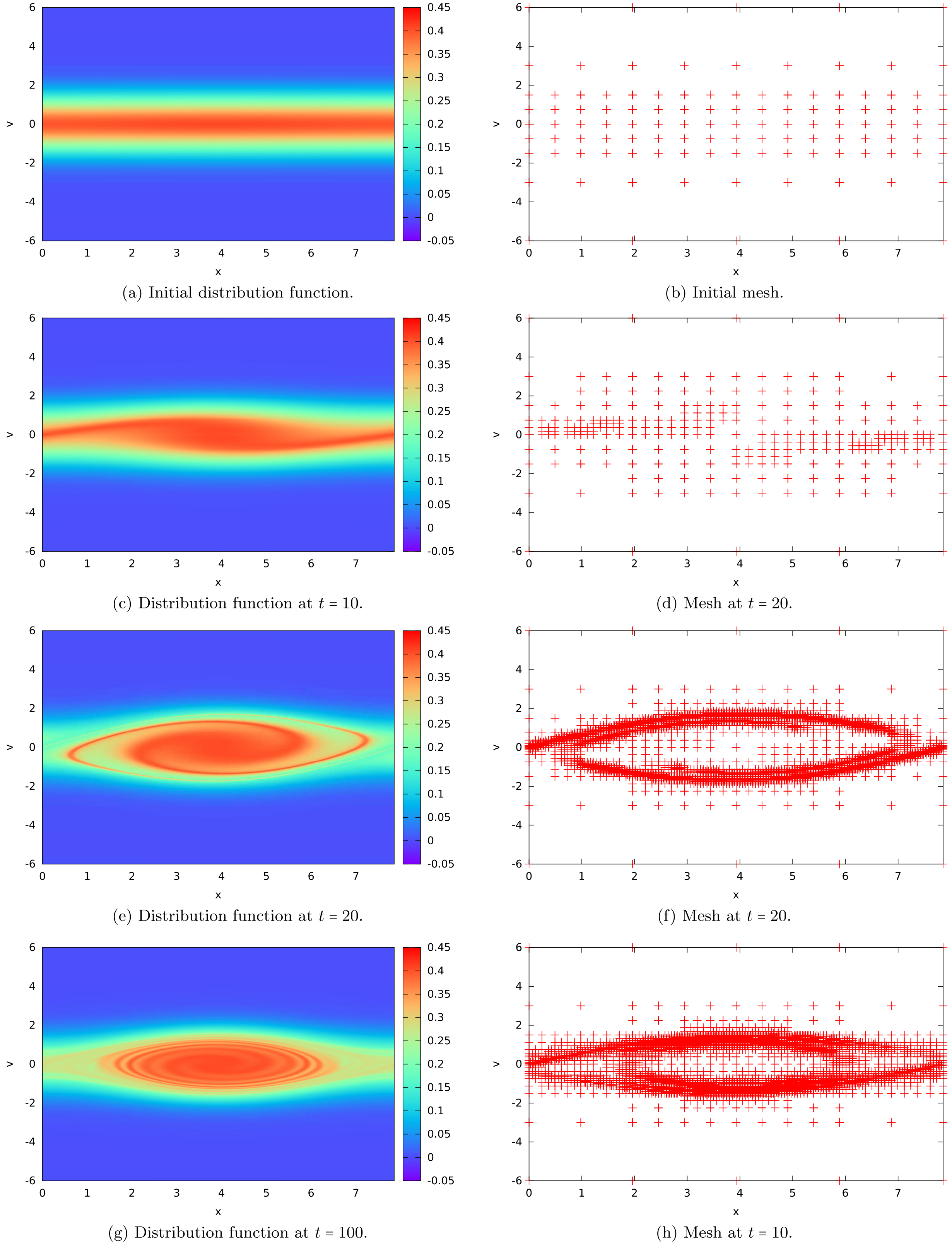
Equation (78) depicts the combination of a rotation in plan (x, v_x) of angular momentum $+1$ with a rotation in plan (y, v_y) of angular momentum -1.

The final time of simulation is $T = 2\pi$ with 200 time steps for a time step $\Delta t = \pi/100$. We use polynomials of degree at most 3. The maximum level of refinement is 6 and the threshold is $\epsilon_0 = 10^{-10}$. The maximum level of refinement is always reached in this test case.

The projections presented on Figs. 41 and 42 are obtained using the formulae

$$Pf(x, y) = \int_{-10}^{10} \int_{-10}^{10} f(x, y, v_x, v_y)\, dv_x dv_y \tag{80}$$

with $Pf(x, y)$ the projection on plane (x, y). The initial projection on plane (v_x, v_y) is null because the two Gaussian compensate each other. The final time $T = 2\pi$ is supposed to give the exact same distribution function than at initial time. However, it is clear with the projection on plane (v_x, v_y) that we do not have an exact numerical rotation, but still very



(a) Initial distribution function.

(b) Initial mesh.

(c) Distribution function at $t = 10$.

(d) Mesh at $t = 20$.

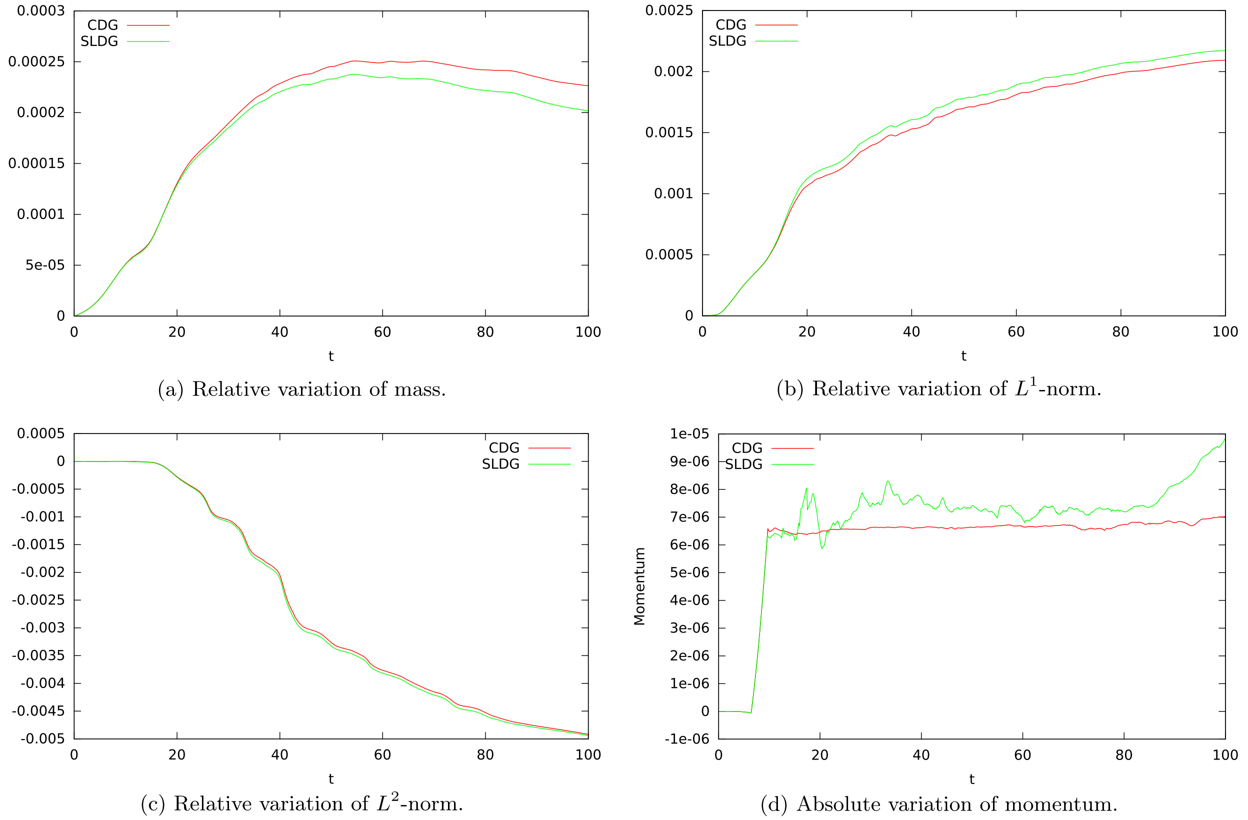
(e) Distribution function at $t = 20$.

(f) Mesh at $t = 20$.

(g) Distribution function at $t = 100$.

(h) Mesh at $t = 10$.

Fig. 35. Distribution function and mesh for the Jeans instability with the AMW-CDG scheme.



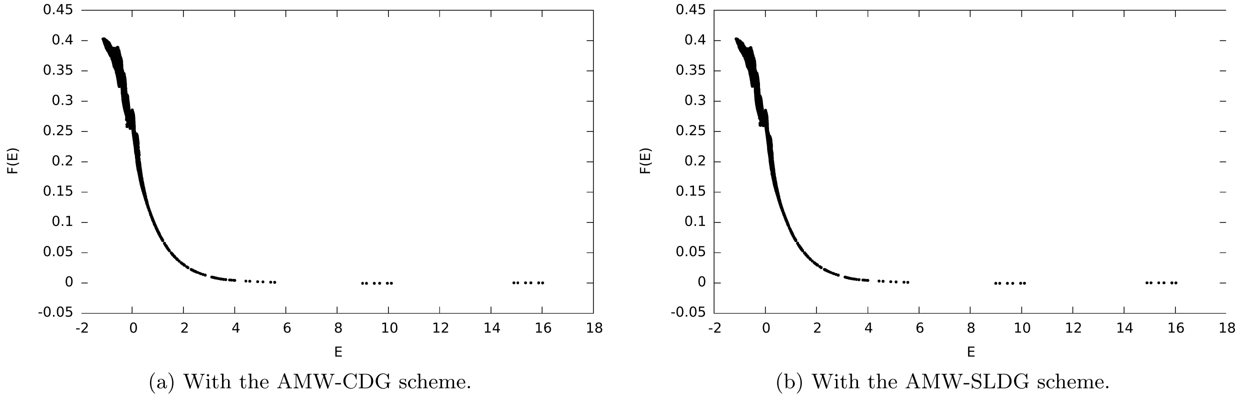
(a) Relative variation of mass.

(b) Relative variation of L^1-norm.

(c) Relative variation of L^2-norm.

(d) Absolute variation of momentum.

Fig. 36. Variation of the main diagnostics for the Jeans instability.



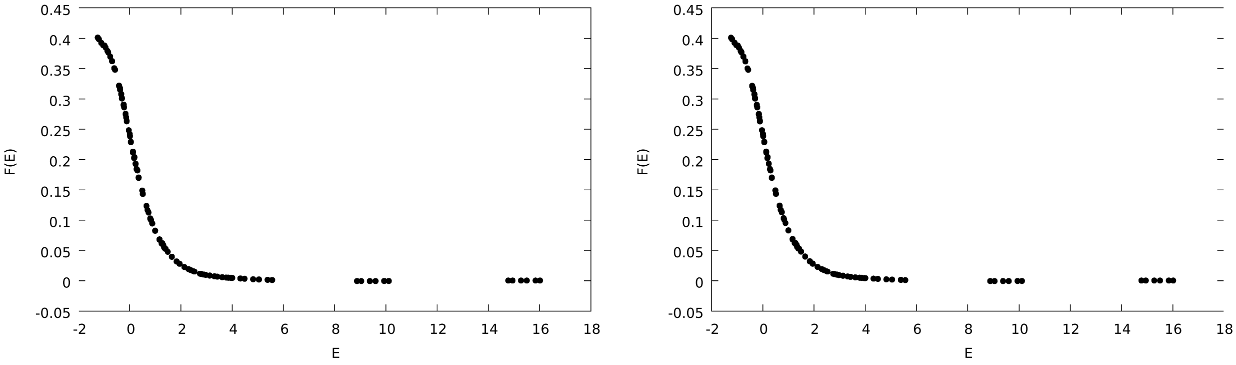
(a) With the AMW-CDG scheme.

(b) With the AMW-SLDG scheme.

Fig. 37. Distribution function as a function of particular energy for the Jeans instability with few diffusion at time $T_f = 100$, with $\epsilon_0 = 0.003$ and 8 levels of refinement.

close to the exact solution. Fig. 42 is obtained with the AMW-CDG scheme. The AMW-SLDG scheme gives extremely similar results. The projection on plane (v_x, v_y) enables to see the integral of the difference between the numerical solution and the exact solution as the integral of the exact solution on plane (v_x, v_y) is null.

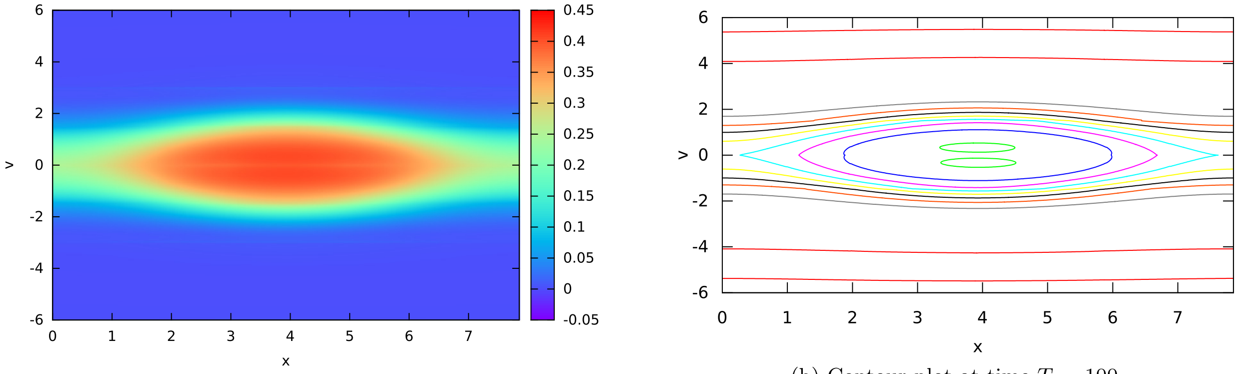
Fig. 43 shows the evolution of mass, L^1-norm, L^2-norm and L^∞-norm of the two methods. The variation of total mass remains on range of slight amplification of machine error. Variation of L^1-norm and L^2-norm are small, less than 2% at most for the L^1-norm, and present some periodic peak. These are linked to the mesh and come from numerical artefacts. The total number of cells, displayed on Fig. 44, is very stable with a small diminution at the beginning. The total number of cells remains around 500,000 cells. If one used a uniform mesh scaled on the finest mesh, this would requires 16, 777, 216 cells. This means that we use only 3% of the computation that would be required for a uniform-mesh simulation. Although the mass and norm variations are slightly better with the AMW-SLDG scheme, we see that it also uses a very few more cells.



(a) With the AMW-CDG scheme.

(b) With the AMW-SLDG scheme.

Fig. 38. Distribution function as a function of particular energy for the Jeans instability with higher diffusion at time $T_f = 100$ with $\epsilon_0 = 0.01$ and 7 levels of refinement.



(a) Distribution function at time T_f = 100.

(b) Contour plot at time T_f = 100.

Fig. 39. Final distribution function for Jeans instability with the AMW-CDG scheme, $\epsilon_0 = 0.01$ and 7 levels of refinement.

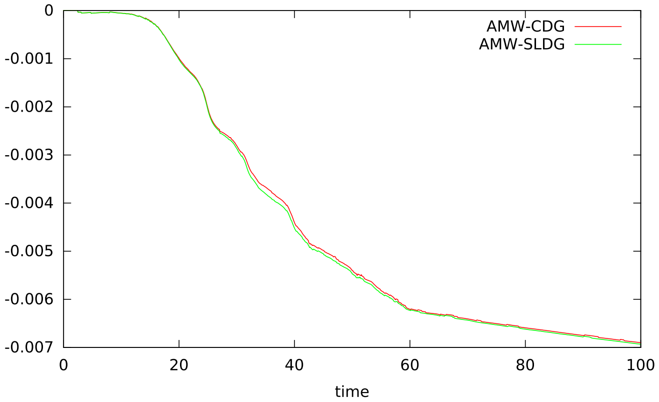
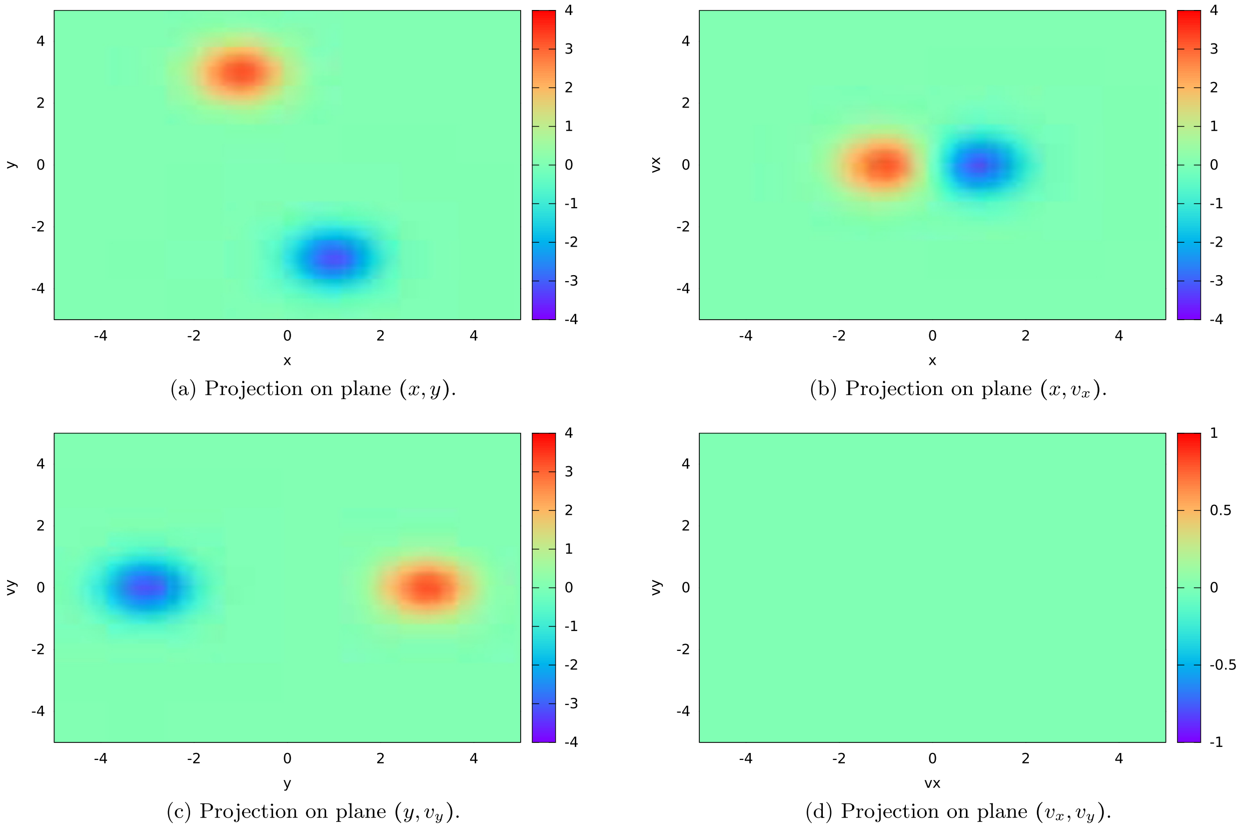


Fig. 40. Relative variation of the L^2-norm for the Weak Jeans instability with $\epsilon_0 = 0.01$ and 7 levels of refinement.

Such four-dimensional simulations show the potential of the AMW-SLDG and AMW-CDG schemes to be used and work well in four dimensions.

8. Concluding remarks

We have implemented and tested an adaptive version of two numerical schemes, namely the semi-Lagrangian discontinuous Galerkin scheme and the characteristics-discontinuous Galerkin scheme. Both schemes include a multiresolution discretisation using a multi-wavelet representation. Most of the time they give identical or extremely similar results. This observation is consistent with the results of Ref. [36] and confirm that our adaptive procedure



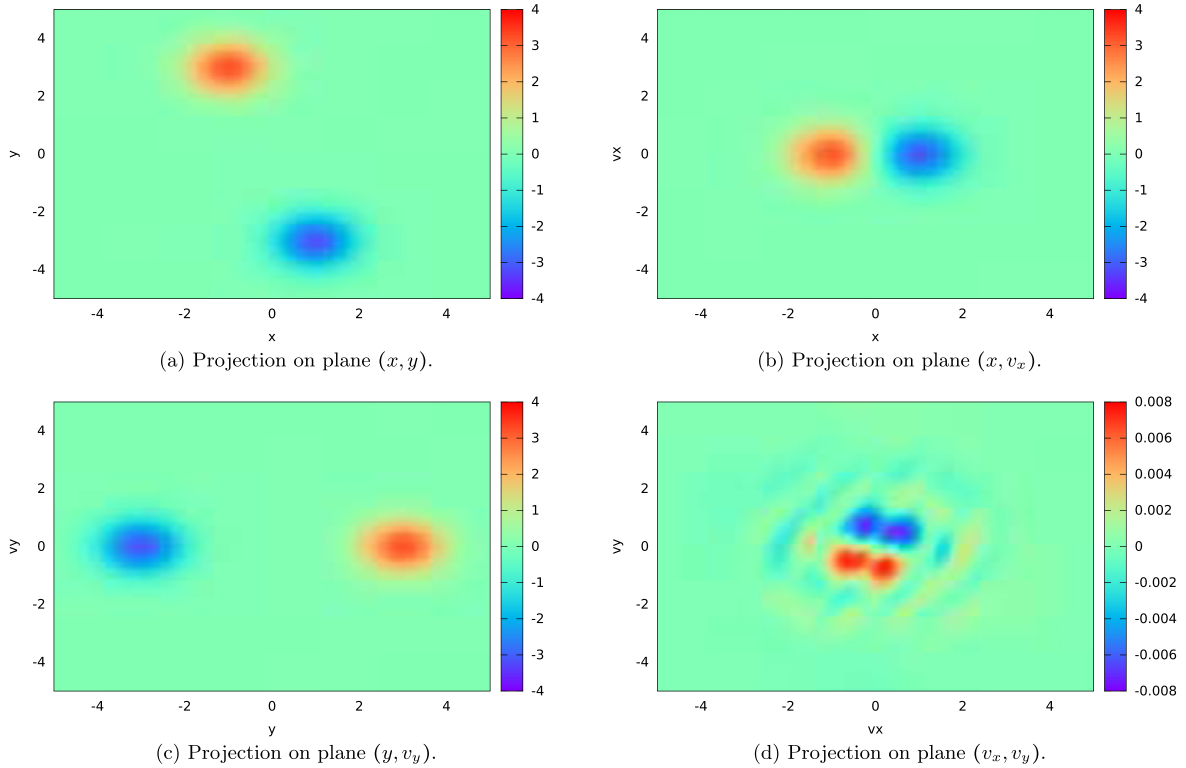
(a) Projection on plane (x, y).

(b) Projection on plane (x, v_x).

(c) Projection on plane (y, v_y).

(d) Projection on plane (v_x, v_y).

Fig. 41. Initial projections.



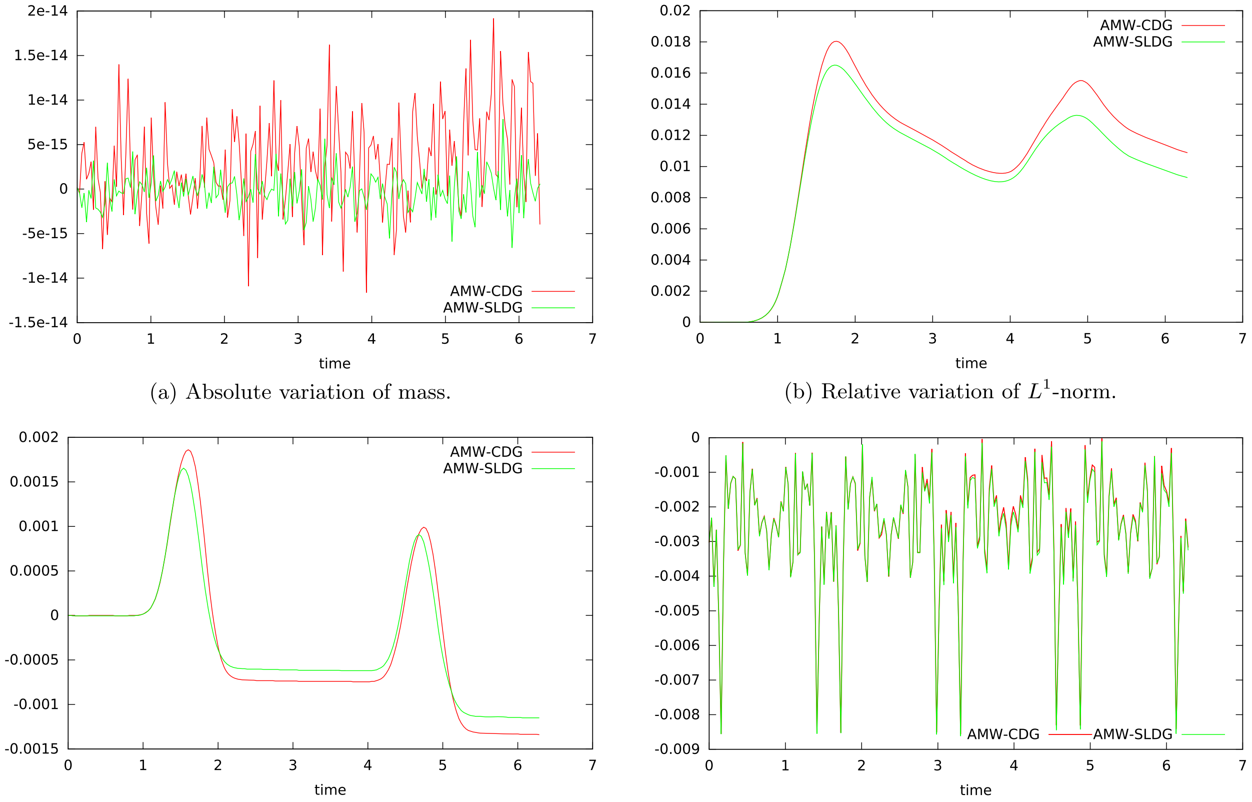
(a) Projection on plane (x, y).

(b) Projection on plane (x, v_x).

(c) Projection on plane (y, v_y).

(d) Projection on plane (v_x, v_y).

Fig. 42. Projections at final time $t = 2\pi$ with the AMW-CDG scheme.



(a) Absolute variation of mass.

(b) Relative variation of L^1-norm.

(c) Relative variation of L^2-norm.

(d) Relative variation of L^∞-norm.

Fig. 43. Variation of the main diagnostics for the four-dimensional test case.

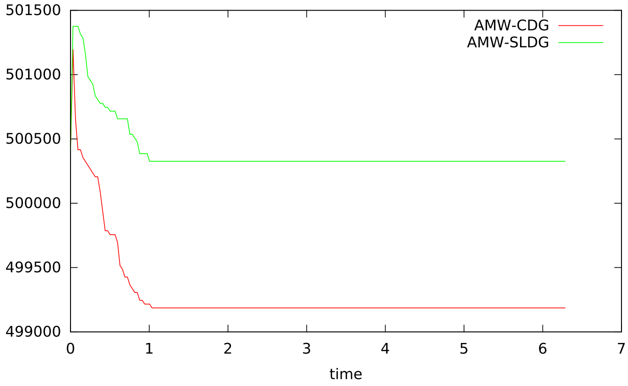


Fig. 44. Number of cells in the four-dimensional test case.

does not introduce a bias or break the symmetry between these two schemes. It is an interesting and open problem to understand whether the AMW-SLDG and AMW-CDG schemes are equivalent or not. In the affirmative case, it is even ambitious but crucial to show it. Both methods have proven to react very well to multiscale decomposition and adaptive mesh. Our future works include a two-dimensional relativistic Vlasov–Maxwell solver to compare our results with [7] or a possible extension to a four-dimensional Vlasov–Poisson solver. A challenging problem is to perform a rigorous mathematical analysis of the AMW-SLDG and AMW-CDG schemes to show their convergence and to obtain a priori error estimates. Finally an efficient parallel version of our algorithm will be a matter of future work.

Acknowledgements

We are grateful to Stéphane Colombi and Thierry Sousbie for many fruitful discussions concerning numerical methods, astrophysics test cases and OpenMP handling. This work was supported by the VLASIX and EUROFUSION projects, respectively under grants No. ANR-13-MONU-000-01 and EURATOM-WP15-ENR-01/IPP-01. We are also grateful to referees whose constructive comments lead to some improvements of the paper.

References

[1] B. Alpert, G. Beylkin, D. Gines, L. Vozovoi, Adaptive solution of partial differential equations in multiwavelet bases, J. Comput. Phys. 182 (1) (2002) 149–190.

[2] B.K. Alpert, A class of bases in L^2 for the sparse representation of integral operator, SIAM J. Math. Anal. 24 (1) (1993) 246–262.

[3] R. Archibald, G. Fann, W. Shelton, Adaptive discontinuous Galerkin methods in multiwavelets bases, J. Comput. Phys. 61 (7) (2011) 879–890.

[4] B. Ayuso de Dios, J.A. Carrillo de la Plata, C.-W. Shu, Discontinuous Galerkin methods for the one-dimensional Vlasov–Poisson system, Kinet. Relat. Models 4 (4) (2011) 955–989.

[5] B. Ayuso de Dios, J.A. Carrillo de la Plata, C.-W. Shu, Discontinuous Galerkin methods for the multi-dimensional Vlasov–Poisson problem, Math. Models Methods Appl. Sci. 22 (12) (2012) 1–45.

[6] N. Besse, Y. Elskens, D.F. Escande, P. Bertrand, Validity of quasilinear theory: refutations and new numerical confirmation, Plasma Phys. Control. Fusion 53 (2) (2011).

[7] N. Besse, G. Latu, A. Ghizzo, E. Sonnendrüker, P. Bertrand, A wavelet-MRA-based adaptive semi-Lagrangian method for the relativistic Vlasov–Maxwell system, J. Comput. Phys. 227 (16) (2008) 7889–7916.

[8] N. Besse, M. Mehrenberger, Convergence of classes of high-order semi-Lagrangian schemes for the Vlasov–Poisson system, Math. Comput. 77 (2008) 96–123.

[9] N. Besse, E. Sonnendrücker, Semi-Lagrangian schemes for the Vlasov equation of an unstructured mesh of phase space, J. Comput. Phys. 191 (2003) 341–376.

[10] O. Bokanowski, G. Simamarta, Semi-Lagrangian discontinuous Galerkin schemes for some first and second-order partial differential equations, ArXiv e-prints, arXiv:1503.08554, 2015.

[11] Y. Cheng, I.M. Gamba, Numerical study of Vlasov–Poisson equations for infinite homogeneous stellar systems, Commun. Nonlinear Sci. Numer. Simul. 17 (5) (2012) 2052–2061.

[12] Y. Cheng, I.M. Gamba, P.J. Morrison, Study of conservation and recurrence of Runge–Kutta discontinuous Galerkin scheme for Vlasov–Poisson system, J. Sci. Comput. 56 (2) (2013) 319–349.

[13] C.Z. Cheng, G. Knorr, The integration of the Vlasov equation in configuration space, J. Comput. Phys. 22 (3) (1976) 330–351.

[14] P.N. Childs, K.W. Morton, Characteristic Galerkin methods for scalar conservation laws in one dimension, SIAM J. Numer. Anal. 27 (3) (1990) 553–594.

[15] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (3) (2001) 173–261.

[16] A. Cohen, S.M. Kaber, S. Müller, M. Postel, Full adaptive multiresolution finite volume scheme for conservation laws, Math. Comput. 72 (2003) 183–225.

[17] S. Colombi, J. Touma, Vlasov–Poisson in 1D: waterbags, Mon. Not. R. Astron. Soc. 441 (3) (2014) 2414–2432.

[18] S. Colombi, T. Sousbie, S. Peirani, G. Plum, Y. Suto, Vlasov versus N-body: the Hénon sphere, Mon. Not. R. Astron. Soc. 450 (4) (2015) 3724–3741.

[19] N. Crouseille, M. Mehrenberger, F. Vecil, Discontinuous Galerkin semi-Lagrangian method for Vlasov–Poisson, ESAIM Proc. 32 (2011) 211–230.

[20] J. Douglas, I.M. Gamba, M.C.J. Squeff, Simulation of the transient behavior of a one dimensional semi-conductor device, Mat. Apl. Comput. 5 (1986) 103–122.

[21] J. Douglas, T.F. Russel, Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures, SIAM J. Numer. Anal. 5 (19) (1982) 871–885.

[22] L. Einkemmer, A. Ostermann, Convergence analysis of Strang splitting for Vlasov-type equations, SIAM J. Numer. Anal. 1 (52) (2014) 140–155.

[23] M.R. Feix, P. Bertrand, A universal model: the Vlasov equation, Transp. Theory Stat. Phys. 34 (1–2) (2005) 7–62.

[24] E. Frénod, S.A. Hirstoaga, M. Lutz, Long-time simulation of a highly oscillatory Vlasov equation with an exponential integrator, C. R., Méc. 342 (10–11) (2014) 595–609.

[25] E. Frénod, F. Salvarani, E. Sonnendrücker, Long time simulation of a beam in a periodic focusing channel via a two-scale PIC-method, Math. Models Methods Appl. Sci. 19 (02) (2009) 175–197.

[26] I.M. Gamba, M.C.J. Squeff, Simulation of the transient behavior of a one dimensional semiconductor device II, SIAM J. Numer. Anal. 2 (1989) 539–552.

[27] N. Gerhard, F. Iacono, G. May, S. Müller, R. Schäffer, A high-order discontinuous Galerkin discretization with multiwavelet-based grid adaptation for compressible flows, J. Sci. Comput. 62 (1) (2015) 25–52.

[28] N. Gerhard, S. Müller, Adaptive multiresolution discontinuous Galerkin schemes for conservation laws: multi-dimensional case, Comput. Appl. Math. (April 2014) 1–29.

[29] W. Guo, R.D. Nair, J.-M. Qiu, A conservative semi-Lagrangian discontinuous Galerkin scheme on the cubed-sphere, Mon. Weather Rev. 142 (1) (2014) 457–475.

[30] R.E. Heath, Analysis of the Discontinuous Galerkin Method Applied to Collisionless Plasma Physics, PhD thesis, The University of Texas at Austin, 2007.

[31] E. Heath, I.M. Gamba, P.J. Morrison, C. Michler, A discontinuous Galerkin method for the Vlasov–Poisson system, J. Comput. Phys. 231 (4) (2012) 1140–1174.

[32] N. Hovhannisyan, S. Müller, R. Schäfer, Adaptive multiresolution discontinuous schemes for conservation laws, Math. Comput. 83 (2014) 113–151.

[33] É. Madaule, M. Restelli, E. Sonnendrüker, Energy conserving discontinuous Galerkin spectral element method for the Vlasov–Poisson system, J. Comput. Phys. 279 (2014) 261–288.

[34] T. Nakamura, T. Yabe, Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov—Poisson equation in phase space, Comput. Phys. Commun. 120 (2) (1999) 122–154.

[35] O. Pironneau, On the transport-diffusion algorithm and its application to the Navier–Stockes equations, Numer. Math. 38 (3) (1982) 309–332.

[36] J.-M. Qiu, C.-W. Shu, Positivity preserving semi-Lagrangian discontinuous Galerkin formulation: theoretical analysis and application to the Vlasov–Poisson system, J. Comput. Phys. 230 (23) (2011) 8386–8409.

[37] J.A. Rossmanith, D.C. Seal, A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov–Poisson equations, J. Comput. Phys. 230 (16) (March 2011) 6203–6232.

[38] A.B. Shelton, A Multi-Resolution Discontinuous Galerkin Method for Unsteady Compressible Flows, PhD thesis, Georgia Institute of Technology, 2008.