Guiding center simulations on curvilinear grids

Adnane Hamiaz, Michel Mehrenberger, Aurore Back

Guiding center simulations on curvilinear grids

A. Hamiaz, M. Mehrenberger, A. Back

January 8, 2015

[1]

Abstract

Semi-Lagrangian guiding center simulations are performed on sinusoidal perturbations of cartesian grids, thanks to the use of a B-spline finite element solver for the Poisson equation and the classical backward semi-Lagrangian method (BSL) for the advection. We are able to reproduce the standard Kelvin-Helmholtz instability test on such grids. When the perturbation leads to a strong distorted mesh, we observe that the solution differs if one takes standard numerical parameters that are used in the cartesian reference case. We can recover good results together with correct mass conservation, by diminishing the time step.

1 Introduction

Semi-Lagrangian schemes often deal with cartesian mesh; the extension to curvilinear grids is important in order to be able to deal with specific geometries and also for adapting the grid to save computational effort. This study is part of a general work on adding curvilinear capabilities for the simulation of drift kinetic and gyrokinetic equations in a semi-Lagrangian framework, and is in current development in the SeLaLib library [17].
In order to treat the case of a general geometry, semi-Lagrangian schemes on unstructured triangular meshes have been developed and applied to Vlasov-Poisson simulations [5]. Recently, a new approach has been developed which permits to stick on a cartesian mesh, with a suitable technique to treat boundary conditions [12]. Another option is to use curvilinear grids, with analytical or discrete transformation [1]. There, the choice has been made to keep the expressions of the advection equations in the physical space, rather than to rewrite these equations in the reference space. Here, we choose the other option (as in [3], for PIC simulations), which permits to avoid the cost overhead of the localization of the feet of the characteristics (which is also present on unstructured meshes),

[1] This work was carried out within the framework the European Fusion Development Agreement and the French Research Federation for Fusion Studies. It is supported by the European Communities under the contract of Association between Euratom and CEA. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

since the localisation is performed on the cartesian grid. The semi-Lagrangian method can be adapted to the new equation, which is of similar nature, and the cost of this step remains of the same order. We will use here the classical backward semi-Lagrangian method (BSL) with cubic splines. Note that other approaches can be developed [14]. There, conservative schemes are used and applied in analytical cases and $1D \times 1D$ Vlasov-Poisson simulations.
The work is devoted to the case of the $2D$ guiding center equation. The Poisson solver has to be adapted; it is here $2D$ (and not $1D$ like in [14]). The use of specific solvers on cartesian (or even polar) geometry with Fourier transform, are here no more available, as we consider general curvilinear coordinates. We use a B-spline finite element solver, which is a key point in order to solve the Poisson equation in a curvilinear grid.
To see the robustness of the numerical method, we test the method on some Colella grids found in [9], which are sinusoidal perturbations of cartesian grids. Note that preliminary work has been performed in [7]; but there the mesh was only made oblic and the Poisson solver was solved on cartesian grid. We refer also to [2], for recent work on curvilinear semi-Lagrangian schemes, in the context of Navier-Stokes equations.
In Section 2, we write the equations in curvilinear coordinates. We then detail the numerical method (in Section 3), which is here the classical semi-Lagrangian method for the advection and a B-spline finite element solver for the Poisson equation. In Section 4, we give the numerical results. Conclusion and perspectives are presented in Section 5.

2 Curvilinear framework

2.1 Mapping

We denote $\Omega \in \mathbf{R}^2$ the physical domain where the Physic equations are valid. To solve these equations, we consider a curvilinear coordinates system which is a mapping $\mathcal{F}$, defined on a logical domain Q:

$$\mathcal{F} : Q = [\eta_{1_{min}}, \eta_{1_{max}}] \times [\eta_{2_{min}}, \eta_{2_{max}}] \to \Omega \tag{1}$$

$$\begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \to \begin{bmatrix} x(\eta_1, \eta_2) \\ y(\eta_1, \eta_2) \end{bmatrix}.$$

We can define the Jacobian matrix of the transformation:

$$D\mathcal{F}(\eta_1, \eta_2) = \begin{pmatrix} \frac{\partial x}{\partial \eta_1} & \frac{\partial x}{\partial \eta_2} \\ \frac{\partial y}{\partial \eta_1} & \frac{\partial y}{\partial \eta_2} \end{pmatrix},$$

and we denote by $\sqrt{g}$ the jacobian of the matrix $D\mathcal{F}$:

$$\sqrt{g}\,(\eta_1, \eta_2) = \mathrm{Det}(D\mathcal{F}) = \frac{\partial x}{\partial \eta_1}\frac{\partial y}{\partial \eta_2} - \frac{\partial x}{\partial \eta_2}\frac{\partial y}{\partial \eta_1},$$

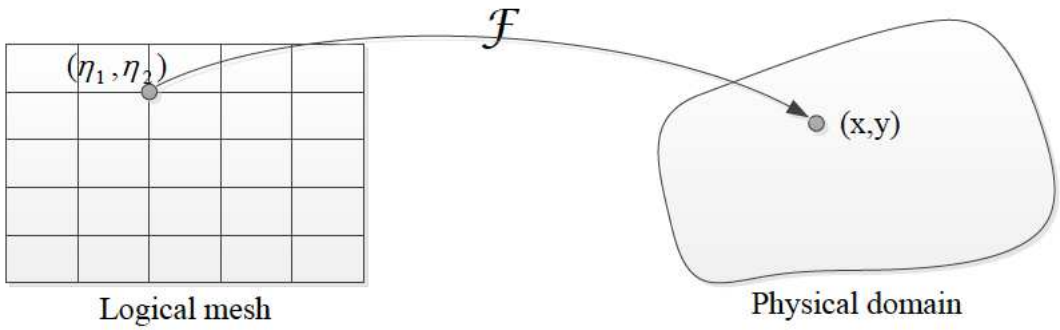

Figure 1: Representation of a mapped mesh in two dimensions.

also we can write the jacobian matrix:

$$D\mathcal{F} = (D\mathcal{F}^{-1})^{-1} = \begin{pmatrix} \frac{\partial \eta_1}{\partial x} & \frac{\partial \eta_1}{\partial y} \\ \frac{\partial \eta_2}{\partial x} & \frac{\partial \eta_2}{\partial y} \end{pmatrix}^{-1} = \sqrt{g} \begin{pmatrix} \frac{\partial \eta_2}{\partial y} & -\frac{\partial \eta_1}{\partial y} \\ -\frac{\partial \eta_2}{\partial x} & \frac{\partial \eta_1}{\partial x} \end{pmatrix},$$

And $D\mathcal{F}^{-1}$ can be also written by

$$D\mathcal{F}^{-1} = \frac{1}{\sqrt{g}} \begin{pmatrix} \frac{\partial y}{\partial \eta_2} & -\frac{\partial x}{\partial \eta_2} \\ -\frac{\partial y}{\partial \eta_1} & \frac{\partial x}{\partial \eta_1} \end{pmatrix}. \tag{2}$$

In the following section, we need to use the expression of the gradient operator ∇ in the new coordinates system (η_1, η_2):

$$\nabla = D\mathcal{F}^{-t}\tilde{\nabla}$$

where $\tilde{\nabla} = \nabla_{(\eta_1,\eta_2)}$.

2.2 Transport equation

We first consider as model, a $2D$ transport equation in cartesian geometry:

$$\frac{\partial f}{\partial t}(t,x,y) + a_1(t,x,y)\frac{\partial f}{\partial x}(t,x,y) + a_2(t,x,y)\frac{\partial f}{\partial y}(t,x,y) = 0, \tag{3}$$

with

$$\frac{\partial a_1}{\partial x}(t,x,y) + \frac{\partial a_2}{\partial x}(t,x,y) = 0.$$

Equation (3) can be written in the space (x, y) as standard advective:

$$\frac{\partial f}{\partial t} + \mathbf{A} \cdot \nabla f = 0, \tag{4}$$

$$\nabla \cdot \mathbf{A} = 0,$$

where $\mathbf{A} = (a_1, a_2)^t$ and since the divergence of $\mathbf{A}$ is zero then there exists a scalar potential Φ such that:

$$\mathbf{A} = \nabla \times \Phi.$$

In other words, we have

$$\mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \frac{\partial \Phi}{\partial y} \\ -\frac{\partial \Phi}{\partial x} \end{bmatrix}. \tag{5}$$

The advection term of the transport equation (4) can be written in the curvilinear coordinates (η_1, η_2) and using

$$\nabla f = D\mathcal{F}^{-t}\tilde{\nabla}\ \tilde{f},$$

we have that:

$$\begin{aligned} \mathbf{A} \cdot \nabla f &= \mathbf{A} \cdot (D\mathcal{F}^{-t}\tilde{\nabla}\ \tilde{f}) \\ &= \mathbf{A}^t (D\mathcal{F}^{-t}\tilde{\nabla}\ \tilde{f}) \\ &= (D\mathcal{F}^{-1}\mathbf{A})^t \tilde{\nabla}\ \tilde{f} \\ &= \tilde{\mathbf{A}} \cdot \tilde{\nabla}\ \tilde{f}. \end{aligned} \tag{6}$$

In the following we will denote $\tilde{f}$ by the functions composition $f \circ \mathcal{F}$ and so

$$\tilde{f}(t, \eta_1, \eta_2) = f(t, x(\eta_1, \eta_2), y(\eta_1, \eta_2)) = f(t, \mathcal{F}(\eta_1, \eta_2)).$$

Moreover with help of the equalities (2),(5) and (6), we obtain:

$$\tilde{\mathbf{A}}(t, \eta_1, \eta_2) = \frac{1}{\sqrt{g}} \begin{bmatrix} \frac{\partial \tilde{\Phi}}{\partial \eta_2} \\ -\frac{\partial \tilde{\Phi}}{\partial \eta_1} \end{bmatrix}.$$

Using (6) the transport equation can be written in the curvilinear coordinates (η_1, η_2) as :

$$\frac{\partial \tilde{f}}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial \tilde{\Phi}}{\partial \eta_2} \frac{\partial \tilde{f}}{\partial \eta_1} - \frac{1}{\sqrt{g}} \frac{\partial \tilde{\Phi}}{\partial \eta_1} \frac{\partial \tilde{f}}{\partial \eta_2} = 0. \tag{7}$$

2.3 Poisson equation

On the physical domain, the Poisson equation has the form:

$$-\nabla \cdot (b(x, y)\nabla \Phi(x, y)) + c(x, y)\Phi(x, y) = f(x, y),$$

where f is the function given by solving the transport equation, Φ is the electric potential, b and c are scalar functions.

We want to write the Poisson equation on curvilinear coordinates thanks to the transformation $\mathcal{F}$ defined in (1) and we have:

$$\nabla \cdot b(x,y)\nabla\Phi(x,y) = \frac{1}{\sqrt{g}} \sum_{i=1}^{2}\sum_{j=1}^{2} \frac{\partial}{\partial \eta_i}(\tilde{b}(\eta_1,\eta_2)\sqrt{g}\, \mathbf{g}^{ij} \frac{\partial \tilde{\Phi}}{\partial \eta_j}(\eta_1,\eta_2)), \qquad (8)$$

where $\tilde{\Phi} = \phi \circ \mathcal{F}$ and $\mathbf{g}^{ij}$ is the contravariant components of the metric tensor $\mathbf{g}$ which are the dot products of the contravariant basis vectors $(\nabla\eta_1, \nabla\eta_2)$:

$$\mathbf{g} = \begin{pmatrix} \mathbf{g}^{11} & \mathbf{g}^{12} \\ \mathbf{g}^{21} & \mathbf{g}^{22} \end{pmatrix} = D\mathcal{F}^{-1} D\mathcal{F}^{-t}.$$

Furthermore, the matrix $\mathbf{g}$ has the following properties:

- $\mathbf{g}$ is symmetric,
- $\mathbf{g}$ is a positive definite matrix (because $D\mathcal{F}$ is invertible),
- since $\det(D\mathcal{F}^{-1}) = \det(D\mathcal{F}^{-t}) = \frac{1}{\sqrt{g}}$ we have

$$\det(\mathbf{g}) = \frac{1}{g}.$$

By (8), the new Poisson equation on curvilinear coordinates has the form:

$$-\sum_{i=1}^{2}\sum_{j=1}^{2} \frac{\partial}{\partial \eta_i}(\tilde{b}\ \sqrt{g}\ \mathbf{g}^{ij} \frac{\partial \tilde{\Phi}}{\partial \eta_j}) + \sqrt{g}\ \tilde{c}\ \tilde{\Phi} = \sqrt{g}\ \tilde{f}. \qquad (9)$$

By developing the left hand side of (8),(9), takes the form:

$$-\tilde{\nabla}\ \cdot (\sqrt{g}\ \tilde{b}\ \mathbf{g} \quad \tilde{\nabla}\ \tilde{\Phi}) + \sqrt{g}\ \ \tilde{c}\ \tilde{\Phi} = \sqrt{g}\ \tilde{f}. \qquad (10)$$

where $\tilde{c} = c \circ \mathcal{F}$ and $\tilde{b} = b \circ \mathcal{F}$.

3 Numerical method

3.1 The Semi-Lagrangian method for the guiding center model

According to the study in the previous section, the equation of the guiding-center on a general mesh can be written in the form (7). Characteristics associated with this equation are given by the following system of differential equations:

$$\frac{\partial \gamma_1(t)}{\partial t} = \frac{\partial_{\eta_2}\tilde{\Phi}(t,(\gamma_1(t),\gamma_2(t)))}{\sqrt{g(\gamma_1(t),\gamma_2(t))}},$$
$$\frac{\partial \gamma_2(t)}{\partial t} = -\frac{\partial_{\eta_1}\tilde{\Phi}(t,(\gamma_1(t),\gamma_2(t)))}{\sqrt{g(\gamma_1(t),\gamma_2(t))}},$$

and at time s we assume that $\gamma_1(s) = \eta_1$ and $\gamma_2(s) = \eta_2$. So let us denote the solution of this system by:

$$\Gamma(t;\eta,s) = \begin{bmatrix}\gamma_1(t;\eta_1,s)\\ \gamma_2(t;\eta_2,s)\end{bmatrix}.$$

The distribution function $\tilde{f}$ is constant along the characteristic curves $(\gamma_1(t), \gamma_2(t))$

$$\tilde{f}(t,\Gamma(t;\eta,s)) = \tilde{f}(s,\Gamma(s;\eta,s)) = \tilde{f}(s,\eta), \qquad \forall\, t,s,\eta. \tag{11}$$

This property will be used to solve a discrete problem, which is defined by introducing mesh points $\eta_{ij} = (\eta_{1,i}, \eta_{2,j})$ for $i = 1,\ldots,N_1$ and $j = 1,\ldots,N_2$ where N_1 and N_2 the numbers of points in each direction η_1 and η_2 respectively. Using the property (11), the classical semi-Lagrangian method, or backward (BSL: Backward Semi Lagrangian see the article [18]) is divided into two steps to compute the distribution function $\tilde{f}_{ij}^{n+1} = \tilde{f}(t^{n+1}, \eta_{ij})$ at time t^{n+1} from the distribution function $\tilde{f}_{ij}^{n} = \tilde{f}(t^{n}, \eta_{ij})$ at time t^n:
For each mesh point η_{ij}

1. Calculating $\Gamma(t^n;\eta_{ij},t^{n+1})$ the value of the characteristic at t^n which is equal to η_{ij} at time t^{n+1}.

2. As the distribution function solution of the guiding center equation reads:

$$\tilde{f}_{ij}^{n+1} = \tilde{f}^n(\Gamma(t^n;\eta_{ij},t^{n+1})),$$

   and since usually the point $\Gamma(t^n;\eta_{ij},t^{n+1})$ is not a point of the logic grid, the value of $\tilde{f}_{ij}^{n+1}$ is obtained by interpolation of the function $\tilde{f}(t^n,\cdot)$ at mesh points $\Gamma(t^n;\eta_{ij},t^{n+1})$ for $i = 1,\ldots,N_1$ and $j = 1,\ldots,N_2$ at time t^n.

The interpolation that is used here is cubic splines. For the computation of the origin of the characteristics, we use Verlet algorithm, and cubic splines for the field A^n.

Computational algorithm

We define a uniform mesh for logical mesh $[\eta_{1_{min}}, \eta_{1_{max}}] \times [\eta_{2_{min}}, \eta_{2_{max}}]$ whose coordinates of points $\eta_{i,j}$ are defined by:

$$\eta_{1,i} = \eta_{1_{min}} + i\Delta\eta_1, \qquad \eta_{2,j} = \eta_{2_{min}} + j\Delta\eta_2$$

where $\Delta\eta_1 = \frac{\eta_{1_{max}} - \eta_{1_{min}}}{N_1}$, $\Delta\eta_2 = \frac{\eta_{2_{max}} - \eta_{2_{min}}}{N_2}$ and N_1, N_2 are respectively the number of cells in each direction η_1 et η_2. We define also a coordinates transformation $\mathcal{F} : (\eta_1,\eta_2) \to (x,y)$ and its Jacobian matrix $D\mathcal{F}$. The algorithm for solving (7) with a predictor corrector-time scheme can be written as follows.

1. **Initialization:**

- compute the initial distribution: $\tilde{f}(0,\eta_1,\eta_2) = \tilde{f}^0(\eta_1,\eta_2)$ at grid points $\eta_{i,j}$ of the logical mesh,
- compute the spline coefficients of $\tilde{f}^0$,
- solve the Poisson equation with $\tilde{f}^0$ as a source term to get the potential electric on logical mesh $\tilde{\Phi}(0,\eta_1,\eta_2) = \tilde{\Phi}^0(\eta_1,\eta_2)$.

2. **Time loop:**

- $t^n \to t^{n+1/2}$
  - compute $\tilde{\Phi}^n$ (solving the Poisson equation),
  - compute $\frac{\partial\tilde{\Phi}^n}{\partial\eta_1}, \frac{\partial\tilde{\Phi}^n}{\partial\eta_2}$ at grid points (here derivatives of the cubic splines)
  - find the feet $(\eta_1(t^n),\ \eta_2(t^n))$ of the characteristics

$$\frac{\partial\eta_1(t)}{\partial t} = \frac{1}{\sqrt{g}}\frac{\partial\tilde{\Phi}^n}{\partial\eta_2}, \quad \frac{\partial\eta_2(t)}{\partial t} = -\frac{1}{\sqrt{g}}\frac{\partial\tilde{\Phi}^n}{\partial\eta_1},$$

    ending at grid points $(\eta_1(t^{n+1/2}),\ \eta_2(t^{n+1/2}))$.
  - interpolate $\tilde{f}^n$ at feet $(\eta_1(t^n),\ \eta_2(t^n))$ to have $\tilde{f}^{n+1/2}$ at grid points,
  - compute the spline coefficients of $\tilde{f}^{n+1/2}$,
  - compute $\tilde{\Phi}^{n+1/2}$ (solving the Poisson equation).

- $t^{n+1/2} \to t^{n+1}$
  - compute $\frac{\partial\tilde{\Phi}^{n+1/2}}{\partial\eta_1}, \frac{\partial\tilde{\Phi}^{n+1/2}}{\partial\eta_2}$ at grid points.
  - find the feet $(\eta_1(t^n),\ \eta_2(t^n))$ of the characteristics

$$\frac{\partial\eta_1(t)}{\partial t} = \frac{1}{\sqrt{g}}\frac{\partial\tilde{\Phi}^{n+1/2}}{\partial\eta_2}, \quad \frac{\partial\eta_2(t)}{\partial t} = -\frac{1}{\sqrt{g}}\frac{\partial\tilde{\Phi}^{n+1/2}}{\partial\eta_1},$$

    ending at grid points $(\eta_1(t^{n+1}),\ \eta_2(t^{n+1}))$.
  - interpolate $\tilde{f}^n$ at feet $(\eta_1(t^n),\ \eta_2(t^n))$ to have $\tilde{f}^{n+1}$ at grid points,
  - compute the spline coefficients of $\tilde{f}^{n+1}$.

3. **End of time loop.**

3.2 Finite element method with B-spline for the Poisson equation on curvilinear coordinates

The equation (10) leads to the elliptic equation:

$$-\tilde{\nabla}\ \cdot\sqrt{g}\ A\tilde{\nabla}\ \tilde{\Phi} + \tilde{c}\sqrt{g}\ \tilde{\Phi} = \tilde{f}\sqrt{g}\,, \tag{12}$$

where the matrix A corresponds to

$$A = \tilde{b}\,\mathbf{g}.$$

We discretize this equation using B-splines basis. For N points in an one dimension mesh, we construct a B-splines family $(B_i)_{1\leqslant i\leqslant N}$ of degree k. It can be generated using a non-decreasing sequence of knots $T = (t_i)_{1\leqslant i\leqslant N+k+1}$, also called *knots vector*.

Definition 3.1 (B-Splines series)**.** *The i-th B-Spline of order k (or of degree $k-1$) is defined by the recurrence relation:*

$$B_i^k(x) = w_i^k(x)B_i^{k-1}(x) + (1 - w_{i+1}^k(x))B_{i+1}^{k-1}(x),$$

where

$$w_i^k(x) = \frac{x - t_i}{t_{i+k-1} - t_i}, \qquad B_i^0(x) = \chi_{[t_i,t_{i+1}[}(x),$$

for $k \geq 1$, $1 \leq i \leq n$ and χ is the characteristic function of $[t_i, t_{i+1}[$. The B-spline B_i^k has a compact support $[t_i, t_{i+k+1}]$.

We construct the knots vector as the article [4] and [6] and we introduce a B-spline discretization of $\tilde{\Phi}$

$$\tilde{\Phi}_h(\eta_1, \eta_2) = \sum_{1\leqslant i\leqslant N_1} \sum_{1\leqslant j\leqslant N_2} \tilde{\Phi}_{i,j} B_i^{k_1}(\eta_1) B_j^{k_2}(\eta_2),$$

where N_1 and N_2 correspond to the number of points in the grid in each direction, k_1 and k_2 the degree of spline in each direction respectively and $\tilde{\Phi}_{i,j}$ are spline coefficients of $\tilde{\Phi}$. We take (12) with $\tilde{\Phi} = \tilde{\Phi}_h$, $\tilde{f} = \tilde{f}_h$, we multiply it by $B_k^{k_1} B_l^{k_2}$ and integrate it over our logical grid Q, which leads, by integrating by part, the following variational equation:

$$\begin{aligned}
&-\sum_{i,j} \tilde{\Phi}_{i,j} \int_Q \left[\left(A_{1,1}(B_i^{\alpha_1})' B_j^{\alpha_2} + A_{1,2} B_i^{\alpha_1} (B_j^{\alpha_2})'\right) (B_k^{\alpha_1})' B_l^{\alpha_2}\right] |\sqrt{g}\,|d\eta \\
&-\sum_{i,j} \tilde{\Phi}_{i,j} \int_Q \left[\left(A_{2,1}(B_i^{\alpha_1})' B_j^{\alpha_2} + A_{2,2} B_i^{\alpha_1} (B_j^{\alpha_2})'\right) B_k^{\alpha_1} (B_l^{\alpha_2})'\right] |\sqrt{g}\,|d\eta \\
&+\sum_{i,j} \tilde{\Phi}_{i,j} \int_Q B_i^{\alpha_1} B_j^{\alpha_2} B_k^{\alpha_1} B_l^{\alpha_2} \tilde{c} |\sqrt{g}\,|d\eta \\
&= \sum_{i,j} \tilde{f}_{i,j} \int_Q B_i^{\alpha_1} B_j^{\alpha_2} B_k^{\alpha_1} B_l^{\alpha_2} |\sqrt{g}\,|d\eta,
\end{aligned}$$

where $A_{i,j}$ is the coefficient (i, j) of the matrix A. We obtain a linear system on spline coefficients of $\tilde{\Phi}_h$ such that

$$\mathbf{M}\tilde{\mathbf{\Phi}}_h = \tilde{\mathbf{f}}_h, \tag{13}$$

where the matrix $\mathbf{M}$ is equal to

$$
\begin{aligned}
\mathbf{M}_{i+(j-1)N_1,k+(l-1)N_1} &= -\int_Q \left[\left(A_{1,1}(B_i^{\alpha_1})'B_j^{\alpha_2} + A_{1,2}B_i^{\alpha_1}(B_j^{\alpha_2})'\right)(B_k^{\alpha_1})'B_l^{\alpha_2}\right]|\sqrt{g}\,|d\eta \\
&- \int_Q \left[\left(A_{2,1}(B_i^{\alpha_1})'B_j^{\alpha_2} + A_{2,2}B_i^{\alpha_1}(B_j^{\alpha_2})'\right)B_k^{\alpha_1}(B_l^{\alpha_2})'\right]|\sqrt{g}\,|d\eta \\
&+ \int_Q B_i^{\alpha_1}B_j^{\alpha_2}B_k^{\alpha_1}B_l^{\alpha_2}\tilde{c}|\sqrt{g}\,|d\eta,
\end{aligned}
$$

and $\tilde{\mathbf{\Phi}}_{h,i+(j-1)N_1} = \tilde{\Phi}_{i,j}$ and

$$
\tilde{\mathbf{f}}_{h,k+(l-1)N_1} = \sum_{i,j}\tilde{f}_{i,j}\int_Q B_i^{\alpha_1}B_j^{\alpha_2}B_k^{\alpha_1}B_l^{\alpha_2}|\sqrt{g}\,|d\eta.
$$

In the next section, we use the Poisson equation with $c = 0$ in (12). In that case, dealing with periodic boundary conditions leads to non invertible matrix. Penalization method, conjugate gradient method or Lagrange multipliers can be used to overcome this problem.

Penalization method

It consists in introducing a small parameter ϵ in the Poisson equation which leads to:

$$
-\tilde{\nabla}\ \cdot\sqrt{g}\,A\tilde{\nabla}\ \tilde{\Phi}_\epsilon + \epsilon\tilde{\Phi}_\epsilon = \tilde{f}\sqrt{g}\,. \tag{14}
$$

This method is attractive but poses problems numerically. In fact the choice of the penalization parameter ϵ is not easy: too big, $\tilde{\Phi}_\epsilon$ is a bad approximation to the exact solution $\tilde{\Phi}$, too small, the problem (14) presents numerical instabilities.

Conjugate gradient method

The system (13) is well-understood non-reversible, and checks ($\mathbb{I}$ is the vector whose components are 1):

- $Im(\mathbf{M}) = \mathbb{I}^\perp$ (The kernel of $\mathbf{M}$ is composed of constants).
- $\tilde{\mathbf{f}} \in Im(\mathbf{M})$ (The compatibility condition $\int_Q \tilde{f}_h|\sqrt{g}\,|d\eta = 0$ of the second member of (12)).

Under these conditions, conjugate gradient method satisfies the following property: if we initialize with an orthogonal vector $\tilde{\mathbf{\Phi}}_h^0$ to the vector $\mathbb{I}$, then it is the same for all the iterates. In the case where the mesh is a regular grid, this leads to a numerical solution with zero mean, which corresponds to the desired solution. However, if the mesh is not regular, it is sufficient to subtract from $\tilde{\Phi}_h$ its mean to find an approximation of the solution.

Lagrange multipliers method

This method consists in introducing the zero mean constraint using a Lagrange multiplier. We multiply (12) by a test function $\psi(\eta)$ vanishing on the boundary and we integrate it by part over the logical domain Q, thats give us:

$$-\int_Q A\tilde{\nabla}\ \tilde{\Phi}\cdot\tilde{\nabla}\ \psi|\sqrt{g}\ |d\eta=\int_Q \tilde{f}\psi|\sqrt{g}\ |d\eta. \tag{15}$$

Specifically, the variational formulation (15) (in which $\psi = B_k^{\alpha_1}B_l^{\alpha_2}$) with the constraint

$$\int_Q \tilde{\Phi}|\sqrt{g}\ |d\eta=0,$$

is equivalent to the problem

$$\left\{\begin{array}{lll} \mathbf{M}\tilde{\mathbf{\Phi}}_h & +\lambda B_h & = \tilde{\mathbf{f}}_h \\ B_h^T\tilde{\mathbf{\Phi}}_h & & = 0 \end{array}\right.$$

with $\lambda\in\mathbb{R}$ and where the vector B_h is defined by

$$(B_h)_i=\int_Q B_k^{\alpha_1}B_l^{\alpha_2}|\sqrt{g}\ |d\eta.$$

Thus the new linear system

$$\left(\begin{array}{c|c} \mathbf{M} & B_h \\ \hline B_h^T & 0 \end{array}\right)\left(\begin{array}{c} \tilde{\mathbf{\Phi}}_h \\ \lambda \end{array}\right)=\left(\begin{array}{c} \tilde{\mathbf{f}}_h \\ 0 \end{array}\right)$$

is invertible and provides both the zero mean solution and the associated multiplier.

4 Numerical results

We first give the expression of the mass and energy in this curvilinear context.

Proposition 4.1. *We define the electric energy in the cartesian case by:*

$$\mathcal{E}(t)=\int_\Omega(\nabla\Phi)^T\nabla\Phi\ dxdy$$

and the mass

$$\mathcal{M}(t)=\int_\Omega f dxdy.$$

The expressions on the curvilinear geometry are

$$\mathcal{E}(t)=\int_Q(\tilde{\nabla}\ \tilde{\Phi})^T\mathbf{g}\tilde{\nabla}\ \tilde{\Phi}\ |\sqrt{g}\ |d\eta_1 d\eta_2,$$

and

$$\mathcal{M}(t) = \int_Q \tilde{f}|\sqrt{g}\ |d\eta_1 d\eta_2,$$

with

$$\sqrt{g}\ = \det(D\mathcal{F}).$$

Proof 4.1. *Let remind that we have*

$$\nabla\phi = (D\mathcal{F})^{-t}\tilde{\nabla}\ \phi,$$

so we deduce that

$$\begin{aligned}
\mathcal{E}(t) &= \int_\Omega (\nabla\Phi)^T \nabla\Phi\ dxdy \\
&= \int_Q (\tilde{\nabla}\ \Phi)^T (D\mathcal{F})^{-1}(D\mathcal{F})^{-t}\tilde{\nabla}\ \Phi|\sqrt{g}\ |d\eta \\
&= \int_Q (\tilde{\nabla}\ \Phi)^T \mathbf{g}\tilde{\nabla}\ \Phi|\sqrt{g}\ |d\eta.
\end{aligned}$$

Kelvin-Helmholtz instability in a periodic box with Colella mesh

We refer for example to [11] for this test case. The initial distribution f_0 is given by the formula :

$$f_0(x,y) = \sin(y) + \beta\cos(\sigma x)$$

where $\beta = 0.015$ and $\sigma = 0.5$. Periodic conditions are considered both in x and y direction. The domain is $[0, L_x] \times [0, L_y]$, with $L_x = \frac{2\pi}{\sigma}$, $L_y = 2\pi$. B-splines of order $k = 4$ are taken for the Poisson solver.
Such test case has been proposed in [11] for example. We test here the robustness of the numerical method on a Colella mesh [9] in order to see the influence of the mesh. The mapping is given by

$$x(\eta_1,\eta_2) = \eta_1 + \alpha\sin(\frac{2\pi}{L_x}\eta_1)\sin(\frac{2\pi}{L_y}\eta_2),\ \ y(\eta_1,\eta_2) = \eta_2 + \alpha\sin(\frac{2\pi}{L_x}\eta_1)\sin(\frac{2\pi}{L_y}\eta_2)$$

for $(\eta_1, \eta_2) \in [0, L_x] \times [0, L_y]$ and for $0 \leq \alpha < 1$.
Distribution function $f(t, x, y)$ is plotted for different meshes at time $t = 45$, on Figure 2 and at time $t = 60$ on Figure 3. The curvilinear mesh is depicted on a 64×64 grid for $\alpha = 0.9$ (Figure 2 top left) and $\alpha = 0.7$ (Figure 3 top left). Note that the mesh corresponding to $\alpha = 0.9$ is quite distorted.
Reference run is given bottom right, with an almost uniform mesh ($\alpha = 10^{-6}$) and 256×256 grid. Note that the results are non distinguishable with those obtained with a uniform mesh ($\alpha = 0$), which are not shown here to gain place. Using α up to 0.7 leads to similar result until time $t = 60$, with the same numerical parameters (see Figure 3 top right). When the mesh is more distorted ($\alpha = 0.9$), we observe that the results differ (see Figure 2 top right). The effect

of refining the grid does not help (see Figures 2 and 3 middle right) but by refining the time step, we observe a better behavior (see Figures 2 and 3 middle left). Nevertheless the numerical method gives raise to numerical dispersion which is diminished by taking a finer grid (see Figures 2 and 3 middle/bottom left). It is known that cubic splines lead to numerical dispersion, when the time step is small (see e.g. [8]). One remedy could be to use Lagrange/Hermite interpolation with odd degree reconstruction of the derivatives (see [19]).
On Figure 4 and 5, $1D$ diagnostics are shown. We observe the same features as on the $2D$ plots, for the mass conservation: mass is increasing for $\alpha = 0.9$ (Figure 4 top left). We see on logarithmic plot (Figure 4 top right) that it is also the case for the other values of α, but the amplitude is smaller. It seems that if the amplitude is big enough, we observe a change in the solution. On the other hand, we also see that by taking a smaller Δt, the mass conservation is enhanced; the evolution is similar, but with a much lower amplitude, which seems to permit not to affect the $2D$ plots (see Figure 4 top left and right).
The curves of energy conservation are similar, when the mesh is not too distorted (Figure 4 middle left), and again for $\alpha = 0.9$ we observe a different behavior. By refining the grid and above all by taking a smaller time step, the energy is also better conserved. L^1 and L^2 norms have a standard behavior (Figure 4 and 4 bottom). We can also note, that L^∞ norm is not good preserved (Figure 4 and 4 middle right), especially with fine grids, but this does not much affect the $2D$ plots; we can observe some pics that appear in the simulation.

5 Conclusion and perspectives

First semi-Lagrangian simulations have been performed on curvilinear grids for the guiding center model, with both Poisson and advection solved on the curvilinear grid. For not too large deformations of the mesh, we are able to reproduce the right results and we notice that time step has to be diminished in order to handle more distorted meshes. Further work will be continued in order to deal with other geometries (like D-shape or polygonal shape) and different boundary conditions (like Neumann for the diocotron instability test [10, 13]). Note that one strength of the method is that we can use the same code and just implement the mapping for dealing with another geometry (if the latter can be described by a mapping). We plan also to work on the conditions which lead to conservation properties. We are also interested in better computing the characteristics when the mesh is strongly distorted; this could have an impact on the numerical results. Finally, we can also compare with eulerian approaches which are subject to a CFL condition.

References

[1] J. Abiteboul, G. Latu, V. Grandgirard, A. Ratnani, E. Sonnendrücker, A. Strugarek, *Solving the Vlasov equation in complex*

geometries, Esaim. Proc., Oct 2011 (CEMRACS'2010), Vol. 32, p103–117.

[2] V.C. Azevedo, M. M. Oliveira, *Efficient Smoke Simulation on Curvilinear Grids*, Pacific Graphics 2013, Vol 32(7) (2013),

[3] A. Back, A. Crestetto, A. Ratnani, E. Sonnendrücker, *An axisymmetric PIC code based on Isogeometric Analysis,* Esaim. Proc., Oct 2011 (CEMRACS'2010), Vol. 32, p118–133.

[4] A. Back, E. Sonnendrücker, *Spline Discrete differential forms and a new finite difference discrete Hodge operator*,hal-00822164

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

[6] C. DeBoor, *A practical guide to splines*, Springer-Verlag, New York, Applied Mathematical Sciences 27, 2001.

[7] J. P. Braeunig, N. Crouseilles, M. Mehrenberger, E. Sonnendrücker, *Guiding-center simulations on curvilinear meshes,* Discrete and Continuous Dynamical Systems Series S, Volume 5, Number 3, June 2012

[8] F. Charles, B. Després, M. Mehrenberger, *Enhanced convergence estimates for semi-lagrangian schemes Application to the Vlasov-Poisson equation,* SIAM J. Numer. Anal. 2013, 51(2), 840-863.

[9] P. Colella, M. R. Dorrand, J.A. Hittinger and D.F. Martin, *High-order, finite-volume methods in mapped coordinates*, J. Comput. Phys. **230**(8), 2952–2976 (2011).

[10] N. Crouseilles, P. Glanc, S. A. Hirstoaga, E. Madaule, M. Mehrenberger, J. Petri, *A new fully two-dimensional conservative semi-Lagrangian method: applications on polar grids, from diocotron instability to ITG turbulence,* hal-00977342, accepted in EPJD, topical issue of Vlasovia 2013.

[11] N. Crouseilles, M. Mehrenberger, E. Sonnendrücker, *Conservative semi-Lagrangian schemes for Vlasov equations,* Journal of Computational Physics 229 (2010), 1927–1953.

[12] F. Filbet, C. Yang, *Mixed semi-Lagrangian/finite difference methods for plasma simulations*, arXiv:1409.8519

[13] S. Hirstoaga, E. Madaule, M. Mehrenberger, J. Pétri, *Semi-Lagrangian simulations of the diocotron instability*, hal-00841504.

[14] M. Mehrenberger, M. Bergot, V. Grandgirard, G. Latu, H. Sellama, E. Sonnendrücker *Conservative Semi-Lagrangian solvers on mapped meshes*, hal-00759823.

[15] J. Pétri, *The diocotron instability in a pulsar "cylindrical" electrosphere,* Astronomy & Astrophysics, February 5, 2008.

[16] J. Pétri, *Non-linear evolution of the diocotron instability in a pulsar electrosphere: 2D PIC simulations,* Astronomy & Astrophysics, May 7, 2009.

[17] SELALIB, `http://selalib.gforge.inria.fr/`

[18] E. Sonnendrücker, J. Roche, P. Bertrand and A. Ghizzo, *The Semi-Lagrangian Method for the Numerical Resolution of the Vlasov Equation*, J. Comput. Phys. **149**, 201-220 (1999).

[19] C. Steiner, *Résolution numérique de l'opérateur de gyromoyenne, schémas d'advection et couplage. Applications l'équation de Vlasov*, PhD, Université de Strasbourg, 2014.

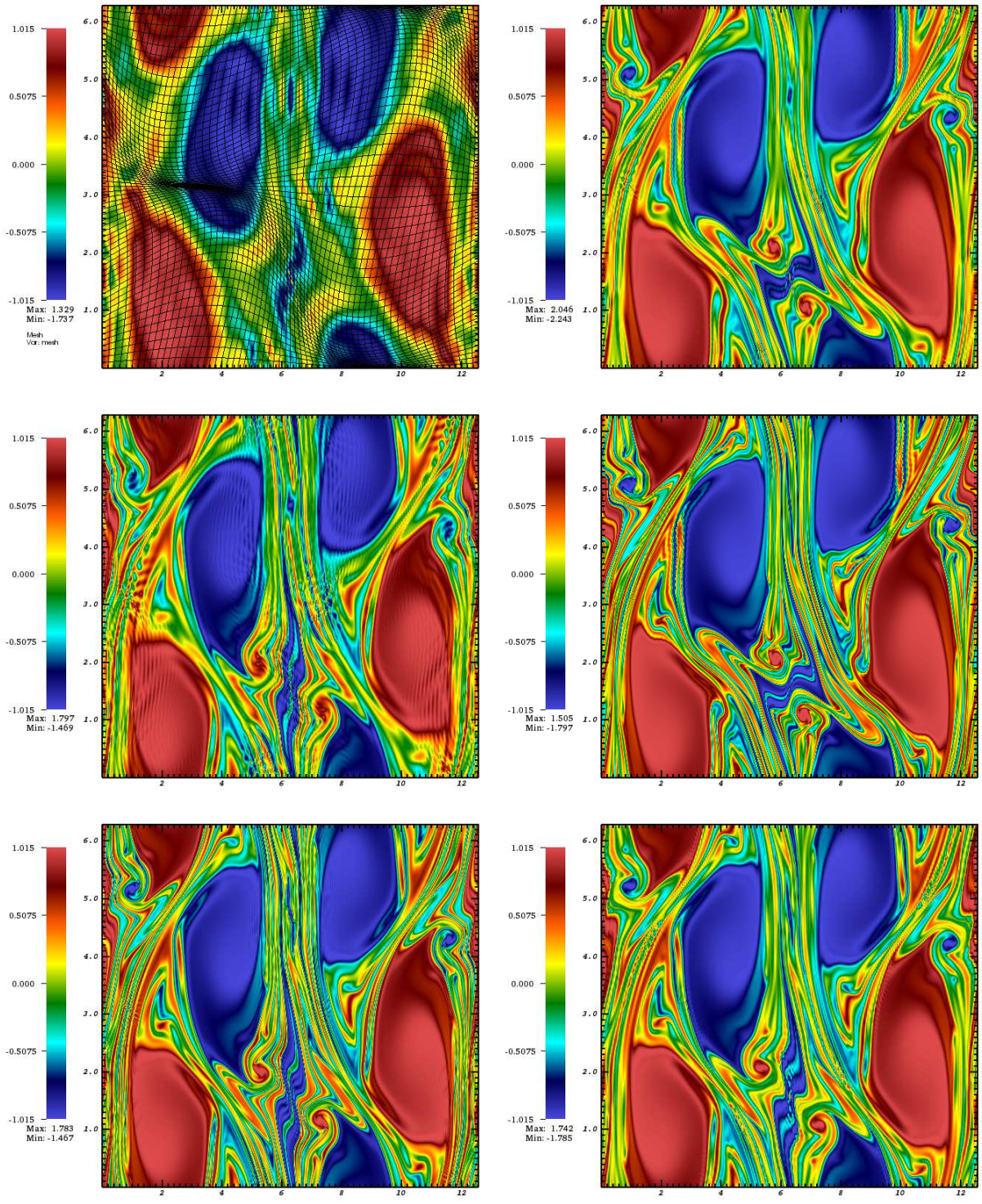

Figure 2: Distribution function $f(t = 45, x, y)$ for different grid sizes, Δt and α parameter of the Colella mesh. From left to right, top to bottom:
64×64 grid, $\Delta t = 0.1$, $\alpha = 0.9$, 256×256 grid, $\Delta t = 0.1$, $\alpha = 0.9$,
256×256 grid, $\Delta t = 0.01$, $\alpha = 0.9$, 512×512 grid, $\Delta t = 0.1$, $\alpha = 0.9$,
512×512 grid, $\Delta t = 0.01$, $\alpha = 0.9$, 256×256 grid, $\Delta t = 0.1$, $\alpha = 10^{-6}$.

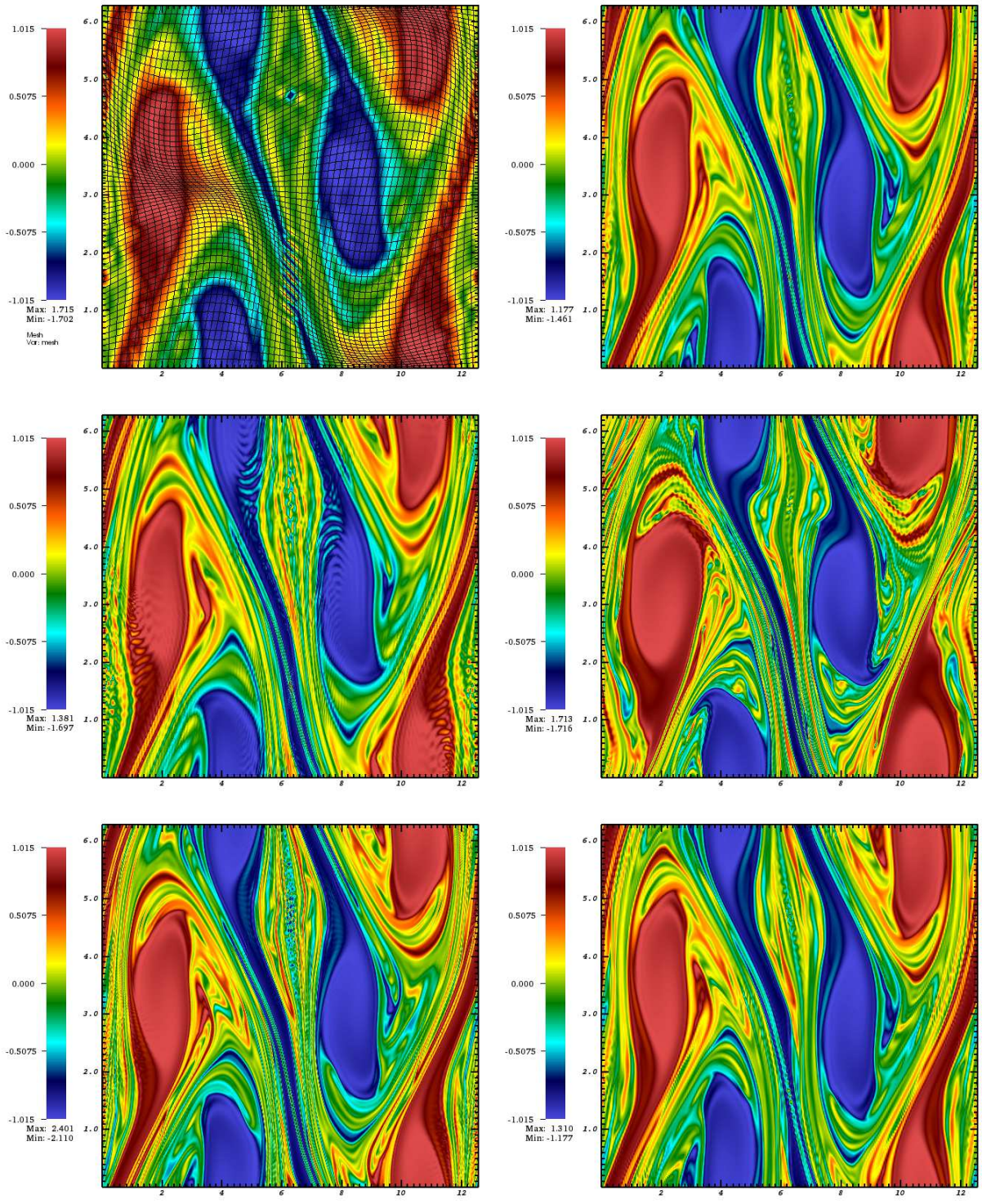

Figure 3: Distribution function $f(t = 60, x, y)$ for different grid sizes, Δt and α parameter of the Colella mesh. From left to right, top to bottom:
64×64, $\Delta t = 0.1$, $\alpha = 0.7$, 256×256, $\Delta t = 0.1$, $\alpha = 0.7$,
256×256, $\Delta t = 0.01$ $\alpha = 0.9$, 512×512, $\Delta t = 0.1$, $\alpha = 0.9$,
512×512, $\Delta t = 0.01$, $\alpha = 0.9$, 256×256, $\Delta t = 0.1$, $\alpha = 10^{-6}$.

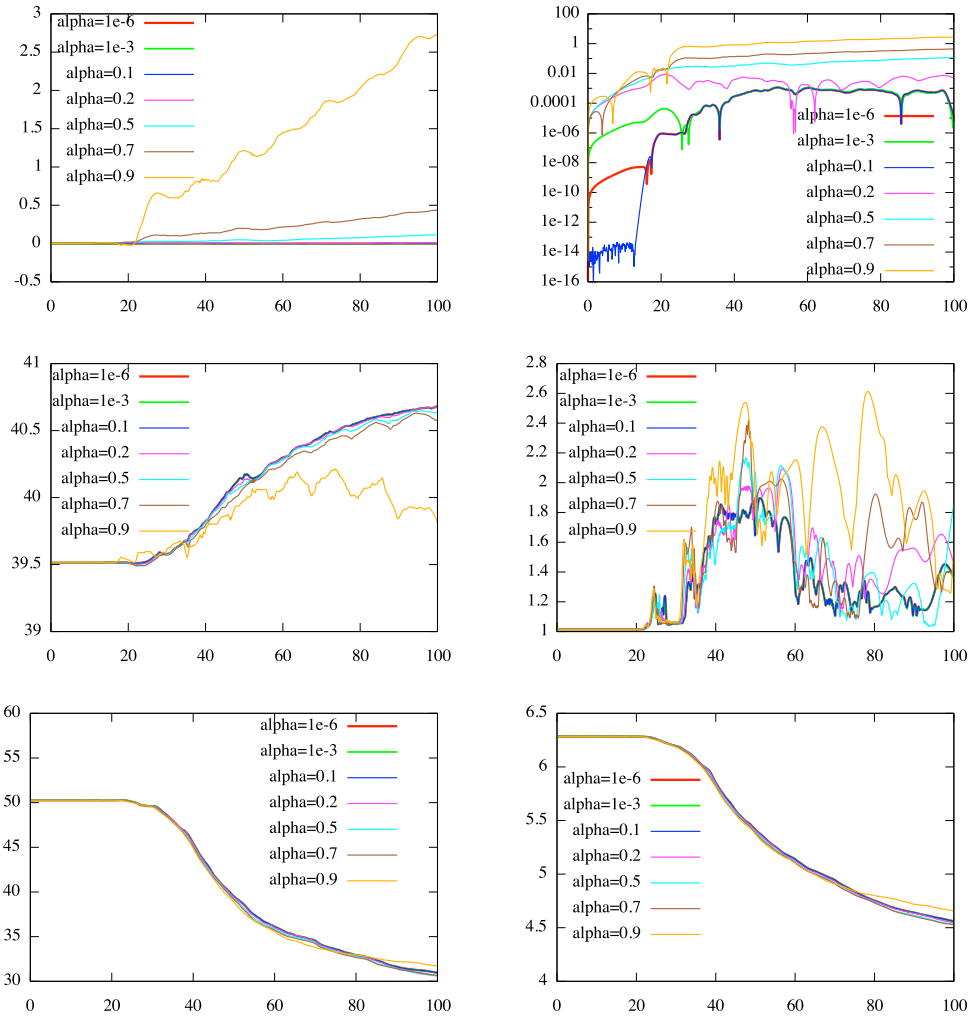

Figure 4: Time evolution of theoretically conserved quantities on 256×256 grid with $\Delta t = 0.1$. From left to right, top to bottom: mass, absolute value of mass, energy, L^∞ norm, L^1 norm, L^2 norm.

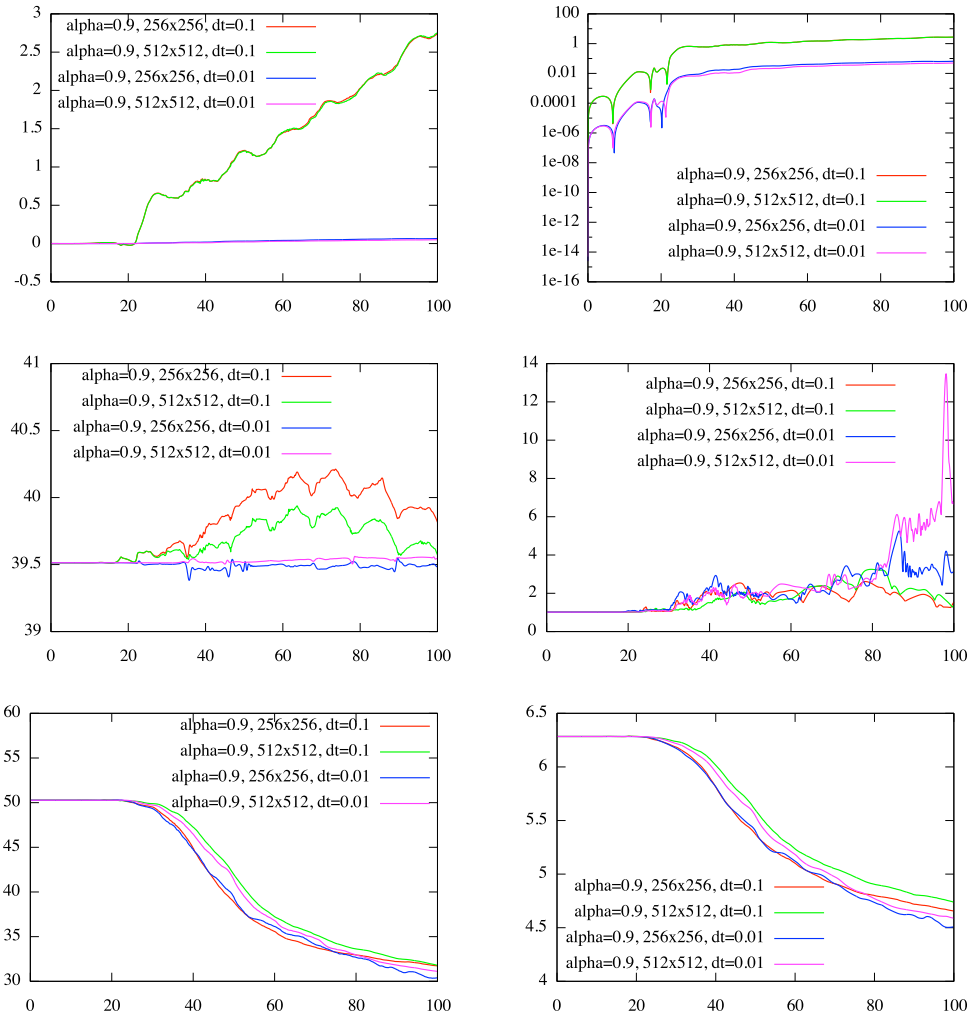

Figure 5: Time evolution of theoretically conserved quantities on 256×256 or 512×512 grid with $\Delta t \in \{0.01, 0.1\}$. From left to right, top to bottom: mass, absolute value of mass, energy, L^∞ norm, L^1 norm, L^2 norm.