Hierarchical based algorithm for the adaptive resolution of the Vlasov equation

Martin Campos-Pinto $^a$, Michel Mehrenberger $^b$, Eric Sonnendrücker $^b$, Eric Violard $^c$

LJLL Université Pierre et Marie Curie, Paris France $^a$, IRMA $^b$ & LSIIT $^c$ Université Louis Pasteur Strasbourg France
The Vlasov equation

Considering the evolution of a system of charged particles under the effect of external and/or self-consistant fields, the distribution function \( f(x, v, t) \) is given by

\[
\frac{\partial f}{\partial t} + v \cdot \nabla_x f + \frac{q}{m} (E + v \times B) \cdot \nabla_v f = 0,
\]

generally coupled with Poisson or Maxwell equations.
Motivations

In Vlasov model, we have appearance of very small scales and distribution function can be null most of the time.
Motivations

In Vlasov model, we have appearance of very small scales and distribution function can be null most of the time

⇒ Adaptive numerical methods
Motivations

In Vlasov model, we have appearance of very small scales and distribution function can be null most of the time

⇒ Adaptive numerical methods

We want to use coarse grids in smoothness regions and fine grids in poor smoothness (or high gradient) regions.
Motivations

In Vlasov model, we have appearance of very small scales and distribution function can be null most of the time

⇒ Adaptive numerical methods

We want to use coarse grids in smoothness regions and fine grids in poor smoothness (or high gradient) regions.
Principle of the Numerical Method

Solve Vlasov equation using a semi-Lagrangian algorithm
Principle of the Numerical Method

Solve Vlasov equation using a semi-Lagrangian algorithm

- Follow the characteristics backward,
Principle of the Numerical Method

Solve Vlasov equation using a semi-Lagrangian algorithm

● Follow the characteristics backward,

● Interpolate the distribution function on “a” grid at the origin of the characteristics,
Principle of the Numerical Method

Solve Vlasov equation using a semi-Lagrangian algorithm

- Follow the characteristics backward,

- Interpolate the distribution function on “a” grid at the origin of the characteristics,
Principle of the Numerical Method

Solve Vlasov equation using a semi-Lagrangian algorithm

- Follow the characteristics backward,

- Interpolate the distribution function on “a” grid at the origin of the characteristics,
Principle of the Numerical Method

Solve Vlasov equation using a semi-Lagrangian algorithm
- Follow the characteristics backward,
- Interpolate the distribution function on “a” grid at the origin of the characteristics,
What are the Grids and how to refine?

- dyadic refinement of the grid
What are the Grids and how to refine?

- dyadic refinement of the grid
- $j_0$ is the coarsest level
- $J$ is the finest level
What are the Grids and how to refine?
What are the Grids and how to refine?

- dyadic refinement of the grid
- $j_0$ is the coarsest level
- $J$ is the finest level
- logical cells of level $j$
MRA framework: projection/prediction operators
To map the distribution function from one level of grid $G_j$ to the next finer level $G_{j+1}$ we define
MRA framework: projection/prediction operators

To map the distribution function from one level of grid $G_j$ to the next finer level $G_{j+1}$ we define

- The projection operator (merely a restriction operator)

\[
P^j_{j+1} : G_{j+1} \rightarrow G_j,
\]

\[
C_{2k}^{j+1} \rightarrow C_k^j,
\]
MRA framework: projection/prediction operators
To map the distribution function from one level of grid $G_j$ to
the next finer level $G_{j+1}$ we define

- The projection operator (merely a restriction operator)

$$P_{j+1}^j : G_{j+1} \rightarrow G_j,$$

$$c_{2k}^{j+1} \mapsto c_k^j,$$

- The prediction operator

$$P_{j}^{j+1} : G_j \rightarrow G_{j+1},$$

such that

$$c_{2k}^{j+1} = c_k^j,$$

$$c_{2k+1}^{j+1} = Q(x_{2k+1}^{j+1}).$$
MRA framework: projection/prediction operators

To map the distribution function from one level of grid $G_j$ to the next finer level $G_{j+1}$ we define

- The projection operator (merely a restriction operator)

$$P_{j+1}^j : G_{j+1} \rightarrow G_j,$$

$$c_{2k}^{j+1} \mapsto c_k^j,$$

- The prediction operator

$$P_{j}^{j+1} : G_j \rightarrow G_{j+1},$$

such that

$$c_{2k}^{j+1} = c_k^j,$$

$$c_{2k+1}^{j+1} = Q(x_{2k+1}^{j+1}),$$

$Q$ stands for an interpolation polynomial.
Initialization:
Numerical algorithms: main steps

Initialization:

- computation of details from analytical initial function $f_0$
Numerical algorithms: main steps

Initialization:

- computation of details from analytical initial function $f_0$
  $\Rightarrow$ compression of $f_0$ by thresholding small details
Numerical algorithms: main steps

Initialization:

- computation of details from analytical initial function $f_0$
  - compression of $f_0$ by thresholding small details
  - corresponding adaptive grid
Numerical algorithms: main steps

Initialization:
- computation of details from analytical initial function $f_0$
  - compression of $f_0$ by thresholding small details
  - corresponding adaptive grid

Time marching step:
Numerical algorithms: main steps

Initialization:
- computation of details from analytical initial function $f_0$
  $\Rightarrow$ compression of $f_0$ by thresholding small details
  $\Rightarrow$ corresponding adaptive grid

Time marching step:
- forward advection of the adaptive grid
Numerical algorithms: main steps

Initialization:

- computation of details from analytical initial function $f_0$
  - compression of $f_0$ by thresholding small details
  - corresponding adaptive grid

Time marching step:

- forward advection of the adaptive grid
  - prediction of new adaptive grid
Numerical algorithms: main steps

Initialization:
- computation of details from analytical initial function $f_0$
  - compression of $f_0$ by thresholding small details
  - corresponding adaptive grid

Time marching step:
- forward advection of the adaptive grid
  - prediction of new adaptive grid
  - with one level refinement
Numerical algorithms: main steps

**Initialization:**
- computation of details from analytical initial function $f_0$
  - compression of $f_0$ by thresholding small details
  - corresponding adaptive grid

**Time marching step:**
- forward advection of the adaptive grid
  - prediction of new adaptive grid
    - with one level refinement
- semi-Lagrangian method
**Numerical algorithms: main steps**

**Initialization:**
- computation of details from analytical initial function $f_0$
  - compression of $f_0$ by thresholding small details
  - corresponding adaptive grid

**Time marching step:**
- forward advection of the adaptive grid
  - prediction of new adaptive grid
    - with one level refinement
- semi-Lagrangian method
  - interpolation on old adaptive grid
Numerical algorithms: main steps

Initialization:
- computation of details from analytical initial function $f_0$
  $\Rightarrow$ compression of $f_0$ by thresholding small details
  $\Rightarrow$ corresponding adaptive grid

Time marching step:
- forward advection of the adaptive grid
  $\Rightarrow$ prediction of new adaptive grid
  ... with one level refinement
- semi-Lagrangian method
  $\Rightarrow$ interpolation on old adaptive grid
- compression of new $f$ and coarsening of the grid
Numerical algorithm: a specific approach...

- hierarchical finite elements framework
Numerical algorithm: a specific approach...

- hierarchical finite elements framework
  ⇒ cell compression
Representation of the solution

The representation of the solution is based on
Representation of the solution

The representation of the solution is based on

- a dyadic partition of square $[0, 1]^2$ into “cells” $\alpha$
Representation of the solution

The representation of the solution is based on

- a dyadic partition of square $[0, 1]^2$ into “cells” $\alpha$
  $\Rightarrow$ Adaptive mesh $\mathcal{M}_n$
Representation of the solution

The representation of the solution is based on

- a dyadic partition of square $[0, 1]^2$ into “cells” $\alpha$
  $\Rightarrow$ Adaptive mesh $\mathcal{M}_n$

- a biquadratic interpolation (basis=$\{1, x, v, x^2, v^2, xv, x^2v, xv^2, x^2v^2\}$
The representation of the solution is based on

- a dyadic partition of square $[0, 1]^2$ into “cells” $\alpha$
  $\Rightarrow$ Adaptive mesh $\mathcal{M}_n$

- a biquadratic interpolation (basis=$\{1, x, v, x^2, v^2, xv, x^2v, xv^2, x^2v^2\}$
  $\Rightarrow$ need of the 9 values of the distribution function
  on each cell at nodes $\alpha$: $f_n(\alpha)$
Representation of the solution

The representation of the solution is based on

- a dyadic partition of square $[0, 1]^2$ into “cells” $\alpha$
  $\Rightarrow$ Adaptive mesh $\mathcal{M}_n$

- a biquadratic interpolation (basis=$\{1, x, v, x^2, v^2, xv, x^2v, xv^2, x^2v^2\}$
  $\Rightarrow$ need of the 9 values of the distribution function on each cell at nodes $\alpha$: $f_n(\alpha)$

We are in the framework of finite elements in a non uniform mesh.
Initialization (detail)

- start with an empty mesh $\mathcal{M}_0$
Initialization (detail)

- start with an empty mesh $\mathcal{M}_0$

- from level $j = J - 1$ to $j_0$, consider a logical cell of level $j$ and the four “daughters” of level $j + 1$
Initialization (detail)

- start with an empty mesh $\mathcal{M}_0$
- from level $j = J - 1$ to $j_0$, consider a logical cell of level $j$ and the four “daughters” of level $j + 1$
  are the daughters really necessary?
Initialization (detail)

- start with an empty mesh $\mathcal{M}_0$

- from level $j = J - 1$ to $j_0$, consider a logical cell of level $j$ and the four “daughters” of level $j + 1$

  are the daughters really necessary?

  $\Rightarrow$ computation of the details on the subnodes
Initialization (detail)

- start with an empty mesh $\mathcal{M}_0$

- from level $j = J - 1$ to $j_0$, consider a logical cell of level $j$
  and the four “daughters” of level $j + 1$
  are the daughters really necessary?
  $\Rightarrow$ computation of the details on the subnodes

- if necessary, add these 4 cells in $\mathcal{M}_0$ and corresponding $f_0(a)$
Initialization (detail)

- start with an empty mesh $\mathcal{M}_0$

- from level $j = J - 1$ to $j_0$, consider a logical cell of level $j$
- and the four “daughters” of level $j + 1$

  - are the daughters really necessary?

  - $\Rightarrow$ computation of the details on the subnodes

- if necessary, add these 4 cells in $\mathcal{M}_0$ and corresponding $f_0(a)$
- respect of “tree” structure
Time marching step (detail)

- Prediction of the set of cells $\mathcal{M}_{n+1}$ following the characteristics forward from each center cell
Time marching step (detail)

- Prediction of the set of cells $M_{n+1}$ following the characteristics forward from each center cell
- One level refinement (except at finest level)
Time marching step (detail)

- Prediction of the set of cells $\mathcal{M}_{n+1}$ following the characteristics forward from each center cell
- One level refinement (except at finest level)
- semi-Lagrangian step: backward advection $\rightarrow f^{n+1}(a)$
Time marching step (detail)

- Prediction of the set of cells $\mathcal{M}_{n+1}$ following the characteristics forward from each center cell
- One level refinement (except at finest level)
- semi-Lagrangian step: backward advection $\rightarrow f^{n+1}(a)$
- compression
parallelization
Numerical results
Perspectives