

Décomposition de Domaine des méthodes PIC-parcimonieuses par composyx

Paul Pace

paul.pace@math.univ-toulouse.fr



INSTITUT DE MATHÉMATIQUES
de TOULOUSE

13 Juin 2025

1. Remise en contexte

1.1 La méthode Particle-In-Cell

1.2 La méthode de recombinaison parcimonieuse (Sparse)

1.3 Technique offset

2. Décomposition de domaine

2.1 Motivations

2.2 Etat de l'art

2.3 Algorithme avec DD

2.4 Besoins Spécifique

3. Composyx

3.1 Présentation

3.2 Application

3.3 Mise à jour

4. Bilan

Remise en contexte/ -Équation de Vlasov Poisson

Quel est le problème considéré?

On s'intéresse au problème suivant (*Vlasov-Poisson*) :

$$\partial_t f + \mathbf{v} \cdot \nabla_x f + \frac{q}{m} (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \nabla_v f = 0 \quad (\text{Vlasov})$$

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \quad \text{and} \quad \mathbf{E} = -\nabla \Phi \quad (\text{Poisson})$$

Où $f(x, v, t)$ est la fonction de distribution définie dans l'espace des phases : position, vitesse, temps (donc **7 dimensions**).

Et la densité de charge $\rho = n_i - q \int_{\Omega_v} f(x, v, t) dv$

Avec q, m, ϵ_0, n_i , respectivement la charge électrique, la masse, la permittivité diélectrique du vide, la densité de particules.

Et \mathbf{E} le champ électrique, \mathbf{B} le champ magnétique (constant).

Remise en contexte/ -Méthode PIC Standard

Pourquoi utiliser PIC?

La méthode **Particle In Cell** (PIC) est une approche mixte, Eulerienne-Lagrangienne :

- Résolution de *Poisson* sur grille par **Différences Finies**.
- Résolution de *Vlasov* par l'intégration des trajectoires de **particules** numériques.

Avantages :

Meilleure mise à l'échelle

Rapidité d'exécution

Mise en oeuvre simple

Interprétabilité physique du modèle

Conservation de quantités physiques

Inconvénients :

Méthode Aléatoire

Grilles pouvant être lourdes

Algorithm 1: PIC method

Data: The particles' initial position and speed

```
1 for  $t \leftarrow 0$  to  $T_{end}$  do
2   | Charge deposition from particle to grid
3   | Poisson Solving and Field computation using FDM
4   | Interpolate Field to the particles' position
5   | Move particles using the computed forces (Newton's equation)
6 end
```

Remise en contexte/ -Le Problème des méthodes PIC

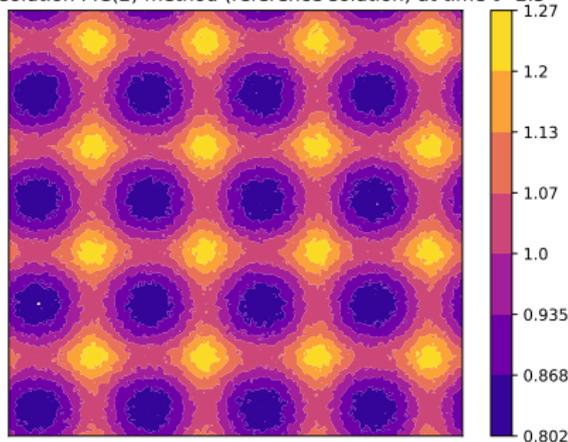
Bruit numérique > Erreur de grille

$$\hat{\rho}_{h_n, N} - \rho = \underbrace{(\hat{\rho}_{h_n, N} - \mathbb{E}(\hat{\rho}_{h_n, N}))}_{\text{Bruit numérique}} + \underbrace{(\mathbb{E}(\hat{\rho}_{h_n, N}) - \rho)}_{\text{Erreur de grille}}$$

Bruit numérique

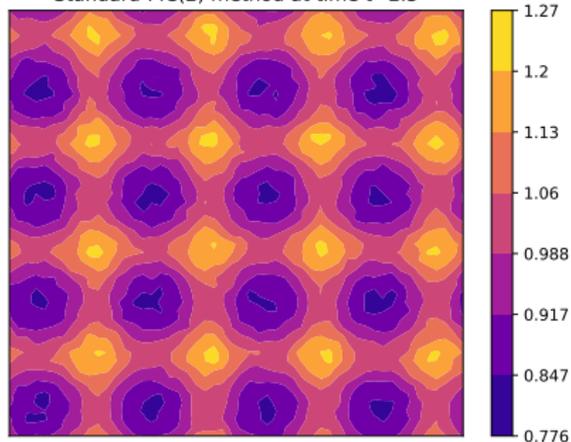
Erreur de grille

High Resolution PIC(2) method (reference solution) at time t=2.5



Solution de référence

Standard-PIC(2) method at time t=2.5



Méthode PIC Standard

Remise en contexte/ -La solution Sparse-grid

Comment réduire le bruit et gagner en efficacité

Algorithm 2: Sparse-PIC

Data: The particles' initial position and speed

```
1 for  $t \leftarrow 0$  to  $T_{end}$  do
2   for each  $g \in Grids$  do
3     Charge deposition from particle to grid
4     Poisson Solving and Field computation using FDM
5     Interpolate Field to the particles' position
6   end
7   Move particles using the computed forces (Newton's equation)
8 end
```

Réduction de bruit :

Sparse :

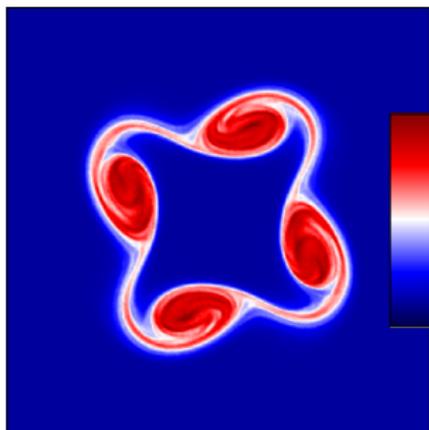
$$(Nh_n)^{-\frac{1}{2}} |\log h_n|^{d-1} \|\rho\|_\infty$$

Standard :

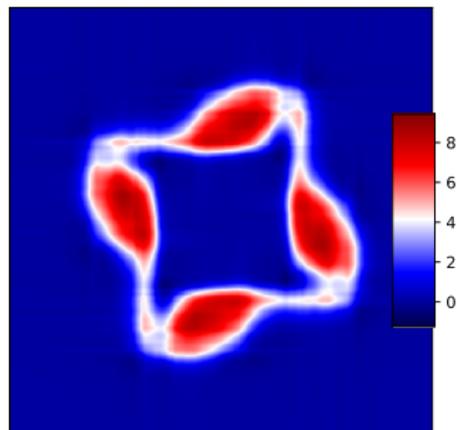
$$(Nh_n^d)^{-\frac{1}{2}} \|\rho\|_\infty$$

Remise en contexte/ -Le revers des Sparse-grid

Illustration de cas inhomogène : instabilité Diocotron



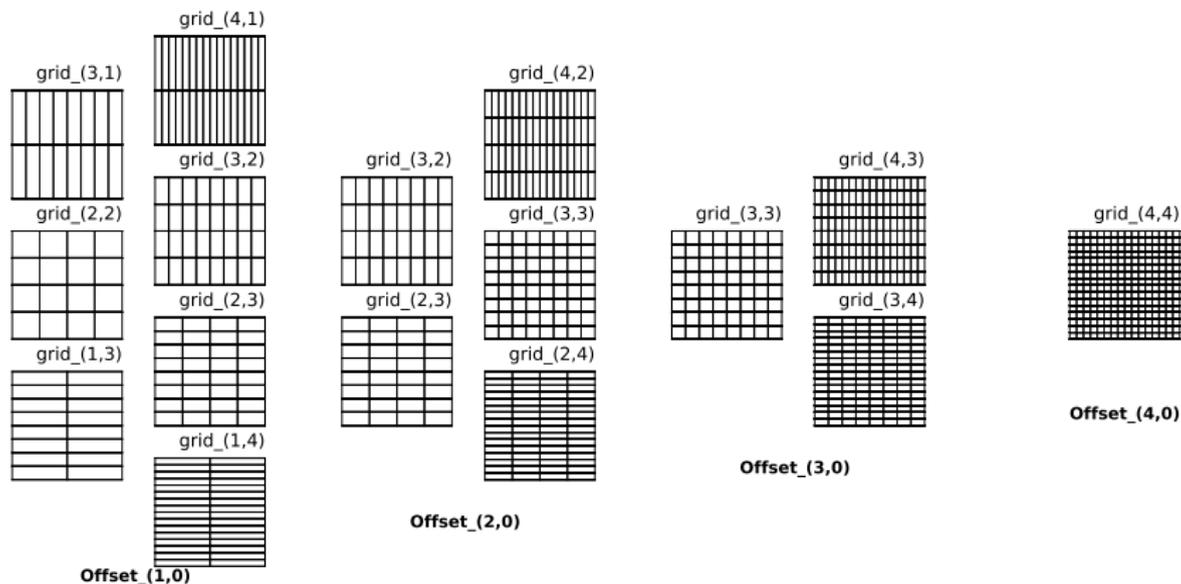
PIC Standard



Sparse-PIC

Remise en contexte/ -Sélection de grille "offset"

Entre PIC et Sparse-PIC



Plus le niveau minimal de résolution des grilles augmente, plus on réduit l'erreur de grille.

Décomposition de domaine/ -Higher Dimension

Comment résoudre des gros problèmes 3D?

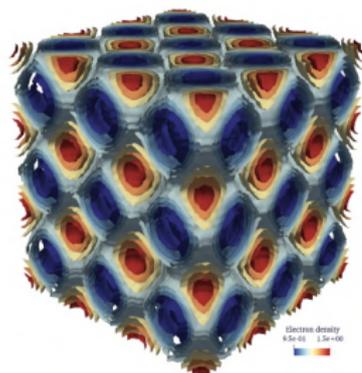


Figure – Méthode Sparse-PIC pour le Landau Damping en 3 dimensions, image tirée de la thèse de Clément Guillet

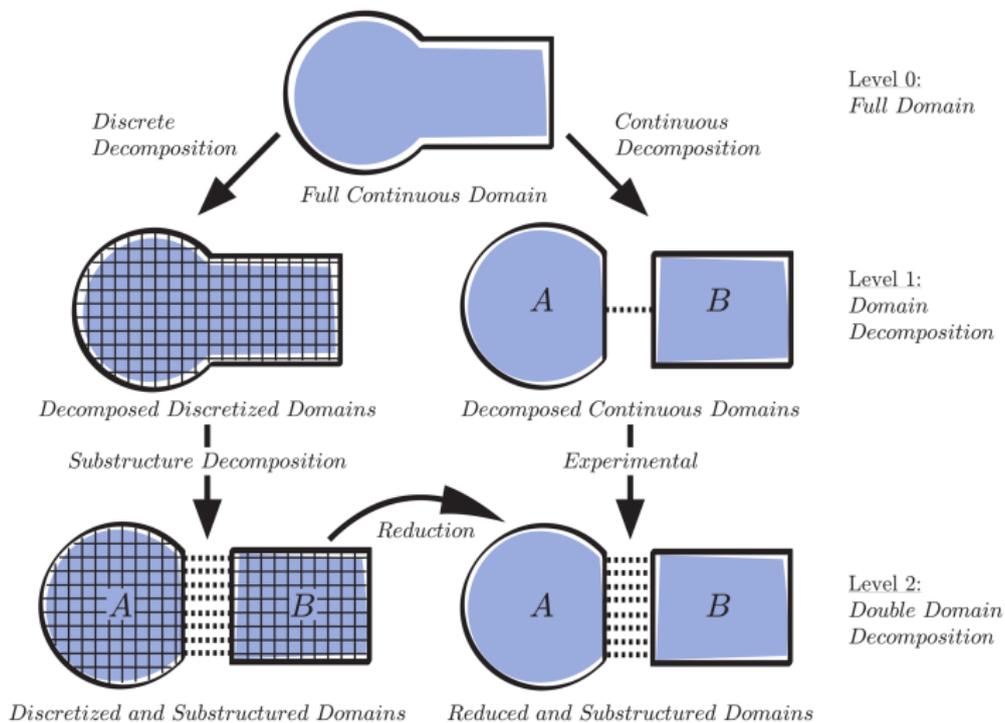


Figure – High Performance Computer (HPC) Jean Zay

Les stratégies de calculs doivent être alors **adaptées** au calcul sur HPC, ou sur GPU.

Décomposition de domaine/ -Différentes méthodes

Recettes pour découper un volume



Algorithm 3: Sparse-PIC with Domain Decomposition

Data: The particles' initial position and speed **for each subdomain**

```
1 for  $t \leftarrow 0$  to  $T_{end}$  do
2   for  $s \in Subdomains$  do
3     for each  $g \in Grids$  do
4       Charge deposition from particles to grid
5       Poisson Solving with Interface interaction
6       Field computation using FDM
7       Interpolate Field to the particles' position
8     end
9   end
10  Move particles using the computed forces (Newton's equation)
11  Exchange particles that cross the border
12 end
```

Ce qu'on voit, c'est donc qu'on a besoin de plusieurs choses :

- Une définition cartésienne du voisinage des sous-domaines.
- Des communications entre les sous-domaines pour échanger les particules.
- Des méthodes de décomposition de domaine **efficace** pour le calcul de Poisson. La propriété la plus importante étant la **scalabilité** de cette méthode.

Composyx est développé par INRIA Bordeaux (Emmanuel Agullo, Gilles Marait, Florent Pruvost, Matthieu Simonin), est une bibliothèque de méthodes de Décomposition de Domaine écrite en C++.

Composyx propose exactement ce qu'on cherche : depuis une interface, on gère de loin, une gamme de solveurs. Et il s'occupe **efficacement** d'effectuer communications entre sous-domaines et les calculs.

On a donc deux outils à mettre en place :

MPI pour les particules,
au niveau de la gestion mémoire,
au niveau des échanges de particules.

Composyx pour la grille,
pour la résolution de Poisson,
pour le calcul du champs.

Composyx/ -Avancement actuel

Où en sommes-nous ?

Algorithm 4: Sparse-PIC with Domain Decomposition

Data: The particles' initial position and speed for each subdomain

```
1 for  $t \leftarrow 0$  to  $T_{end}$  do
2   for  $s \in \text{Subdomains}$  do
3     for each  $g \in \text{Grids}$  do
4       Charge deposition from particles to grid
5       Poisson Solving with Interface interaction
6       Field computation using FDM
7       Interpolate Field to the particles' position
8     end
9   end
0   Move particles using the computed forces (Newton's equation)
1   Exchange particles that cross the border
2 end
```

En premier lieu :

- Finir la méthode décomposée
- L'exporter sur des HPC

Plus largement :

- Développer les méthodes Sparse-PIC Hierarchiques
- Obtenir des estimations dans ce cadre
- Ajouter un niveau de parallélisme au niveau de la Décomposition de Domaine