

RAPPORT DU PROJET

UNIVERSITÉ DE NANTES

Méthodes semi-Lagrangiennes pour la simulation numérique des plasmas

Auteurs :

Manon CARREAU

manon.carreau@etu.univ-nantes.fr

Alexandre MECHINEAU

alexandre.mechineau@etu.univ-nantes.fr

À rendre le : 27 mars 2020

1 Introduction

Dans ce projet, le système qui est étudié est celui Vlasov-Poisson.

$$\left\{ \begin{array}{ll} \partial_t f(t, x, v) + v \partial_x f(t, x, v) - E(t, x) \partial_v f & = 0, & \text{dans } (0, T) \times [0, L]_{per} \times \mathbb{R} \\ \partial_x E(t, x) & = 1 - \int_{\mathbb{R}} f dv, & \text{dans } (0, T) \times [0, L]_{per} \\ \int_{[0, L]_{per}} E dx & = 0, & \text{dans } (0, T) \\ f(t = 0, x, v) & = f_0(x, v), \end{array} \right. \quad (1)$$

où $f_0(x, v) : [0, L]_{per} \times \mathbb{R} \rightarrow \mathbb{R}^+$ une fonction initiale donnée. $T > 0$ est le temps final. $L > 0$ est la longueur de l'espace des x et $[0, L]_{per}$ est le segment entre 0 et L périodique.

Pour résoudre numériquement ce système, deux schémas numériques ont été implémenté : un schéma de différence finie et un schéma semi-lagrangien. Ces deux schémas sont testés et comparé sur le cas test de l'amortissement de Landau. La condition initiale sera une maxwellienne perturbée légèrement. On s'attend à un retour à l'équilibre que l'on étudiera principalement à travers la grandeur nommée énergie électrique définie par :

$$\varepsilon(t) = \sqrt{\int_{[0, L]} \frac{E(t, x)^2}{2} dx}. \quad (2)$$

Pour la résolution numérique, l'espace des vitesses sera discrétisé sur $[-V_{max}, V_{max}]_{per}$ où $V_{max} = 10$ dans les cas étudiés. De plus la solution sera périodique en vitesse.

2 Schémas numériques

Pour toute la suite, la discrétisation en (x, v) est une discrétisation à pas de temps de constant. Pour N_x le nombre de points dans la direction x et N_v le nombre de points en vitesse, les $(x_i)_{i=0}^{N_x}$ et les $(v_j)_{j=0}^{N_v}$ seront définis tels que :

$$x_i = i\Delta x, \Delta x = \frac{1}{N_x} \text{ et } v_j = j\Delta v, \Delta v = \frac{1}{N_v}.$$

Le pas de temps en t , Δt est géré de façon différente selon la méthode employé.

L'approximation de la fonction $f(t^n, x_i, v_j)$ est noté : $f_{i,j}^n$ et l'approximation du champ électromagnétique $E(t^n, x_i)$ est noté : E_i^n .

Les deux schémas numériques sont résolus à l'aide d'un splitting du système de Vlasov-Poisson :

$$\left\{ \begin{array}{ll} \partial_t f(t, x, v) + v \partial_x f(t, x, v) & = 0 \\ \partial_t f(t, x, v) - E(t, x) \partial_v f(t, x, v) & = 0 \end{array} \right. \quad (3)$$

2.1 Méthode des différences finies

L'**initialisation** du schéma numérique est faite au temps $t = 0$ avec $f_{i,j}^0 = f_0(x_i, v_j)$. Les trois prochaines étapes sont ensuite valables quelque soit le temps t^n .

1) **Calcul du champ électriques** à l'aide de l'équation de l'équation de Poisson.

$$E_{i+1}^n = E_i^n + \Delta x \left(1 - \sum_{j=0}^{N_v} \Delta v f_{i,j}^n \right)$$

2) **Calcul de l'équation en vitesse au demi-temps $n + \frac{1}{2}$** en prenant en compte la CFL : $\max_i |E_i^n| \Delta t \leq \Delta v$ et $V_{max} \Delta t \leq \Delta x$. Δt doit répondre au deux inégalités.

On calcule tout d'abord le pas de temps, $\Delta t = \min(\frac{\Delta v}{\max_i |E_i^n|}, \frac{\Delta x}{V_{max}})$. Puis $f_{i,j}^{n+\frac{1}{2}}$ la fonction approchée au demi-temps $n + \frac{1}{2}$:

$$f_{i,j}^{n+\frac{1}{2}} = f_{i,j}^n + \frac{\Delta t}{\Delta x} \left[\max(0, E_i^n)(f_{i,j}^n - f_{i,j-1}^n) + \min(0, E_i^n)(f_{i,j+1}^n - f_{i,j}^n) \right] \quad (4)$$

3) Calcul de l'équation en vitesse au temps $n + 1$.

On calcule $f_{i,j}^{n+1}$ la fonction approchée au temps $n + 1$ en utilisant la fonction approchée au demi-temps $n + \frac{1}{2}$:

$$f_{i,j}^{n+1} = f_{i,j}^{n+\frac{1}{2}} + \frac{\Delta t}{\Delta x} \left[\max(0, v_j)(f_{i,j}^{n+\frac{1}{2}} - f_{i-1,j}^{n+\frac{1}{2}}) + \min(0, v_j)(f_{i+1,j}^{n+\frac{1}{2}} - f_{i,j}^{n+\frac{1}{2}}) \right] \quad (5)$$

Ensuite il suffit de recommencer à l'étape 1) jusqu'au temps final T .

2.2 Méthode semi-lagrangienne

L'**initialisation** et l'**étape 1)** sont les même que pour la méthode des différences finies.

2) le calcul de $f_{i,j}^*$ l'étape intermédiaire de remontée des caractéristiques pour le deuxième équation de (3).

On commence par calculer $p_i = \lfloor \frac{-E_i^n \Delta t}{\Delta v} \rfloor$ l'indice pour remonté le long des caractéristiques où $\lfloor . \rfloor$ est la fonction partie entière et $\alpha_i = \frac{-E_i^n \Delta t}{\Delta v} - \lfloor \frac{-E_i^n \Delta t}{\Delta v} \rfloor$ le coefficient d'interpolation de $f_{i,j}^*$.

$$f_{i,j}^* = (1 - \alpha_i)f_{i,j-p_i}^n + \alpha_i f_{i,j-p_i-1}^n \quad (6)$$

3) le calcul de $f_{i,j}^{n+1}$ se fait à l'aide de la première équation de (3).

On commence encore une fois pas calculer $p_j = \lfloor \frac{v_j \Delta t}{\Delta x} \rfloor$ et $\alpha_j = \frac{v_j \Delta t}{\Delta x} - \lfloor \frac{v_j \Delta t}{\Delta x} \rfloor$ puis on calcule :

$$f_{i,j}^{n+1} = (1 - \alpha_j)f_{i-p,j}^* + \alpha_j f_{i-p-1,j}^* \quad (7)$$

3 Résultats numériques

3.1 Validation du code

Dans le but de vérifier les résultat obtenus pour le calcul du champ électrique et du déplacement dans l'axe de la vitesse, nous utilisons le résultat qui nous dit que le champ électrique n'est pas modifié par le déplacement en vitesse des particules. De ce fait, si nous n'appliquons pas le déplacement spatial, nous devons trouver que l'énergie électrique est donc constante. Ce résultat nous permet de vérifier le schéma utilisant la méthode des différences finies et la méthode semi-lagrangienne. De plus, lors de l'implémentation de la résolution de l'équation de Poisson utilisant la FFT, cela nous permet de vérifier la validité pratique du code.

La seconde méthode pour vérifier la validité du code est de ce rappeler que l'on étudie des équations de transport. En imposant la vitesse dans les routines de déplacement, on s'assure du bon fonctionnement. Pour ce faire, on définit une condition initiale valant h sur un carré et 0 ailleurs. La valeur de h est défini de manière à avoir une distribution. On peut alors tester chaque méthode permettant le déplacement dans une direction.

On vérifie de plus que 0 est bien solution aussi de l'équation.

3.2 Cas test : Amortissement Landau

On cherche à étudier le retour à l'équilibre d'une solution perturbée légèrement.

$$f_0(x, v) = (1 + \epsilon \cos(kx)) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{v^2}{2}\right), k = \frac{2\pi}{L}$$

On s'intéresse non pas a la solution en elle même, mais en une quantité dérivée du champs électrique, nommée énergie électrique :

$$\mathcal{E}(t) = \sqrt{\int_{[0,L]} \frac{E(t, x)^2}{2} dx}$$

On cherche alors à vérifier le retour à un niveau nul de l'énergie électrique. Pour ce faire l'on va comparer les 2 méthodes implémentées. On sait que l'énergie décroît exponentiellement à une vitesse $\gamma = 0.1533$. Nous obtenons alors les résultats suivants :

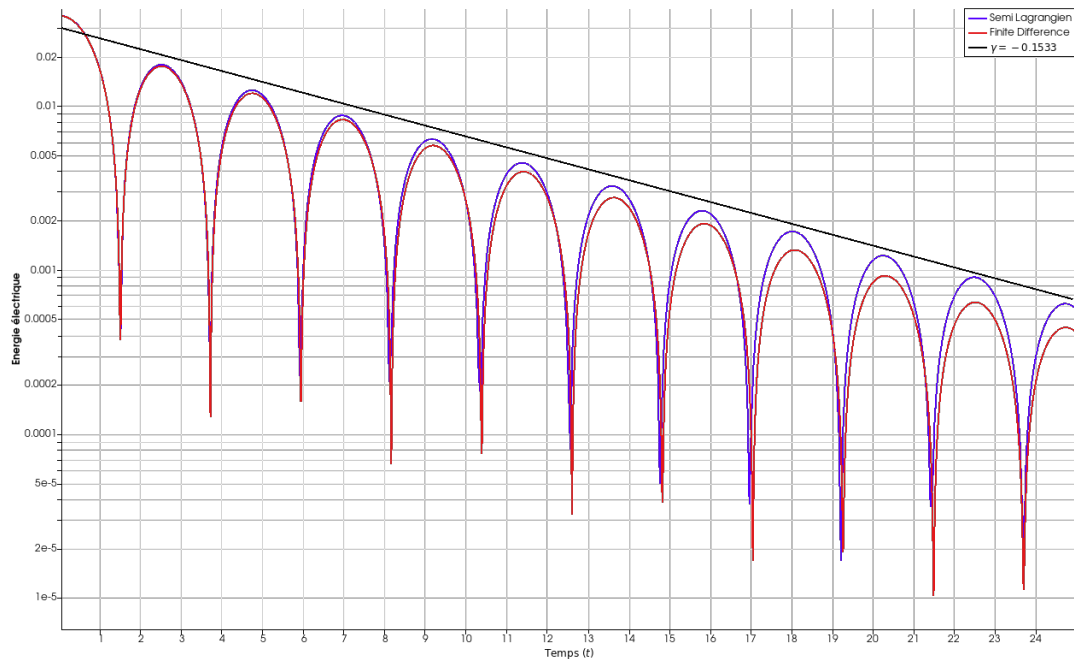


FIGURE 1 – $N = 200, dt = .05$

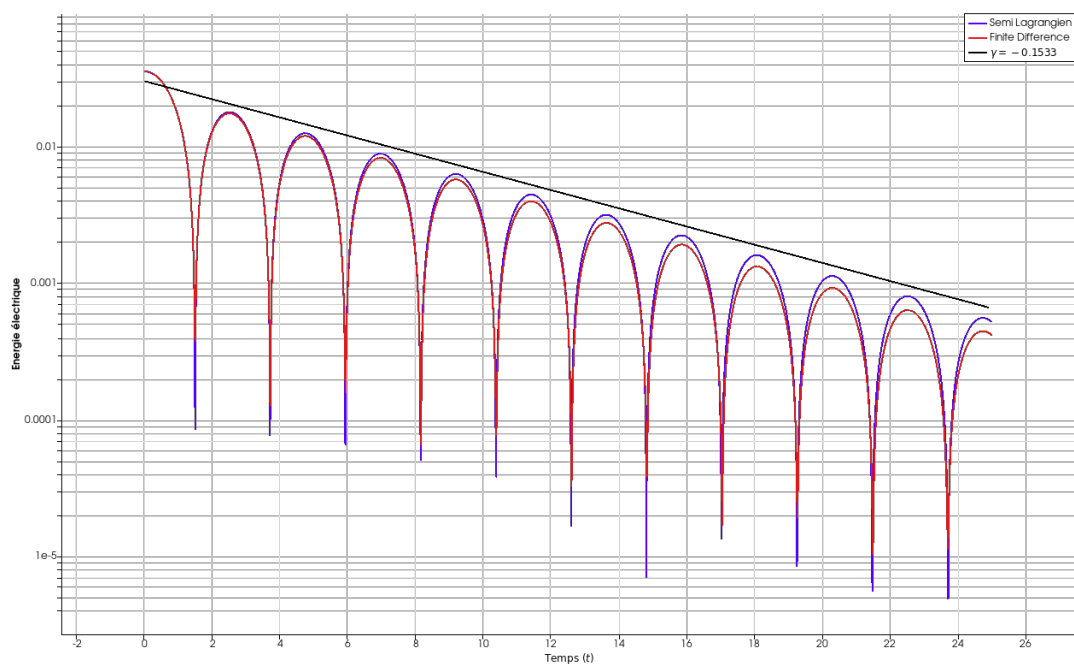


FIGURE 2 – $N = 700, dt = .01$

On peut remarquer que nous obtenons bien le bon phénomène. D'un point de vue calculatoire la différence finie est bien plus longue dû à une restriction de la CFL sur le pas de temps. Pour la méthode semi-lagrangienne, le pas de temps n'est pas restreint. On remarque que toute fois un maillage dense donne de meilleur résultat. En effet,

lors que l'on remonte les caractéristique, l'on cherche à être au plus près d'une maille du maillage dans le but de transporter le plus exactement possible la distribution.

3.3 Cas test : Champ électrique connu

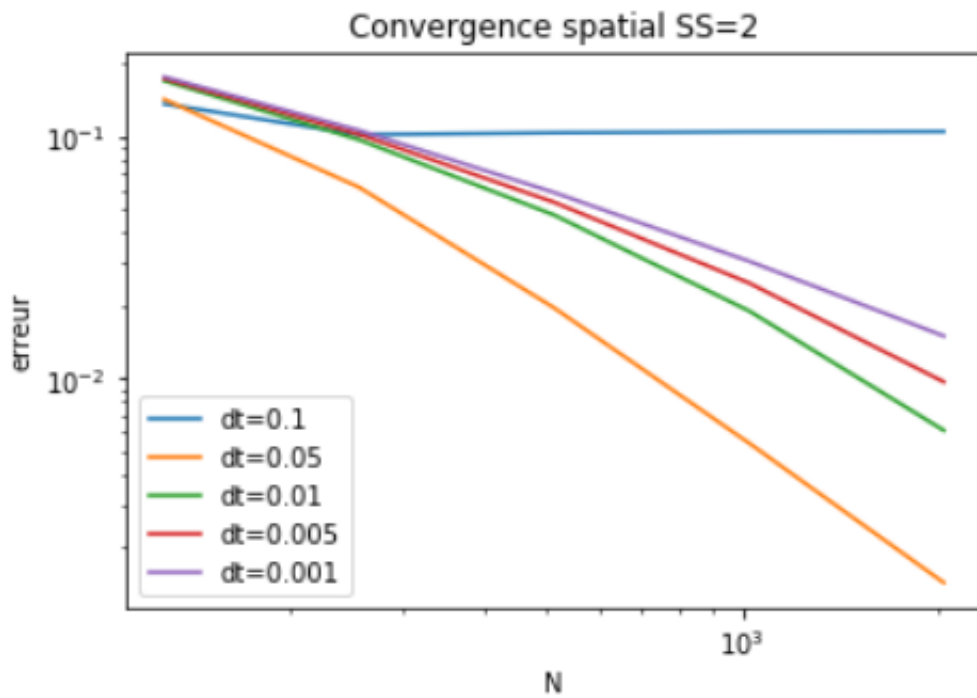
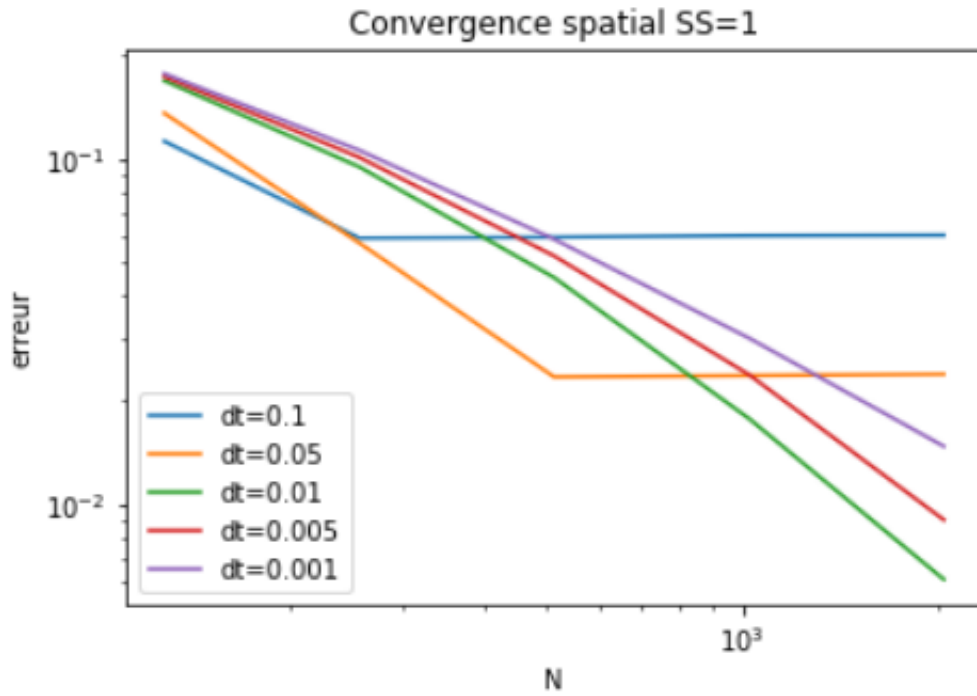
Pour pouvoir étudier la convergence de la méthode implémentée, on cherche à résoudre un problème avec une solution connue. On étudie alors le problème suivant :

$$\begin{aligned}\partial_t f + v \partial_x f - x \partial_v f &= 0, \forall x \in [-5, 5], v \in [-5, 5] \\ f_0(x, v) &= \exp(-10x^2) \exp(-v^2)\end{aligned}$$

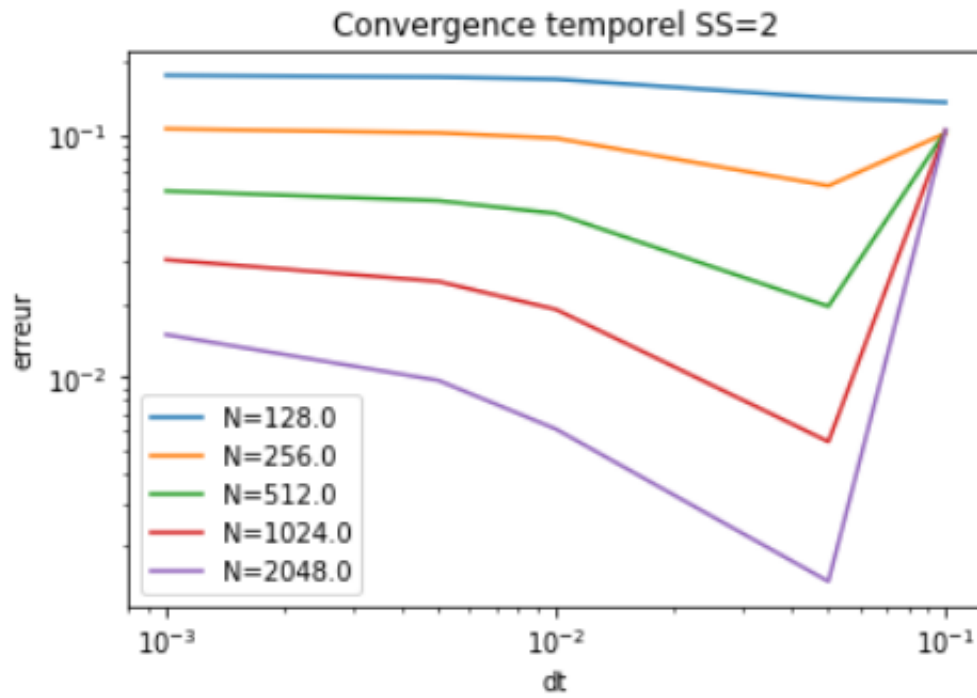
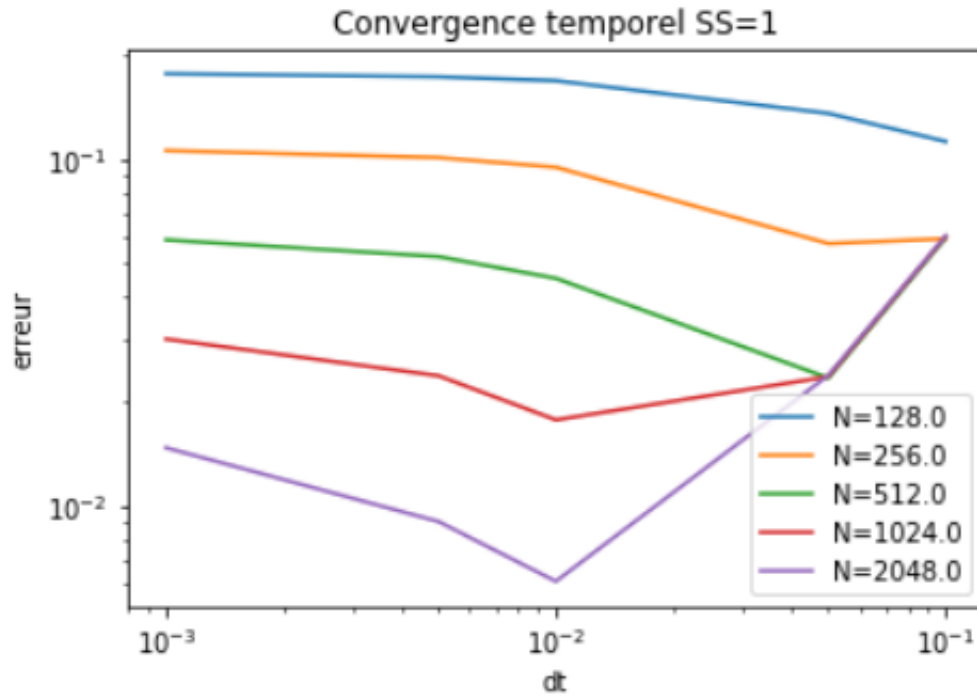
On utilise des conditions au bord de type dirichlet homogène. Ce problème admet la solution analytique suivante :

$$f(t, x, v) = \exp(-10(x \cos(t) - v \sin(t))^2) \exp(-(x \sin(t) + v \cos(t))^2)$$

L'évolution de la solution numérique au cours du temps est fourni sous forme d'une vidéo. Intéressons nous alors à la convergence.



On remarque que si la précision temporelle est trop faible, la précision spatiale n'apporte rien à un certain moment. On remarque que l'utilisation d'un schéma split-step d'ordre plus élevé permet de rétablir la convergence spatiale.



Ici, l'on étudie la convergence temporelle, c'est à dire N fixé. L'on remarque que si la résolution spatiale est faible, des petit pas de temps font augmenter l'erreur. On peut encore une fois observer l'amélioration de la convergence à l'aide de l'ordre plus élevé du split-step.

3.4 Cas test : Keen Waves

Pour cette section, on cherche à reproduire les résultats de l'article Vlasov on GPU (VOG Project) écrit par M. Mehrenberger, C. Steiner, L. Marradi, N. Crouseilles, E. Sonnendrucker, B. Afeyan. Au lieu d'ajouter une perturbation, on ajoute un champ électrique extérieur. On se propose alors d'étudier le modèle suivant :

$$\partial_t f + v \partial_x f + (E - E_{\text{app}}) \partial_v f = 0, \quad \partial_x E = \int_{\mathbb{R}} f dv - 1,$$

where $E_{\text{app}}(t, x)$ is of the form $E_{\text{app}}(t, x) = E_{\text{max}} k a(t) \sin(kx - \omega t)$, where

$$a(t) = \frac{0.5(\tanh(\frac{t-t_L}{t_{wL}}) - \tanh(\frac{t-t_R}{t_{wR}})) - \epsilon}{1 - \epsilon}, \quad \epsilon = 0.5(\tanh(\frac{t_0 - t_L}{t_{wL}}) - \tanh(\frac{t_0 - t_R}{t_{wR}}))$$

is the amplitude, $t_0 = 0$, $t_L = 69$, $t_R = 307$, $t_{wL} = t_{wR} = 20$, $k = 0.26$, $\omega = 0.37$ and $E_{\text{max}} = 0.2$. The initial condition is

$$f_0(x, v) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{v^2}{2}\right), \quad (x, v) \in [0, 2\pi/k] \times [-6, 6].$$

La condition initiale est l'équilibre utilisé pour montrer l'amortissement Landau. On remarque que l'on a du mal à retrouver les phénomènes que l'on retrouve dans la littérature (2 articles sont en référence pour les résultat). Les calculs étant long avec des échelles de temps petites et un nombre de point important, il serait intéressant d'augmenter l'ordre de nos méthode d'interpolation, comme pour le calcul de la valeur du pied de la caractéristique. Il serait intéressant d'ajouter cela de manière à améliorer la qualité des résultats. De même, un ordre pour le split-step plus élevé permettrait de garder pas temporel raisonnable.

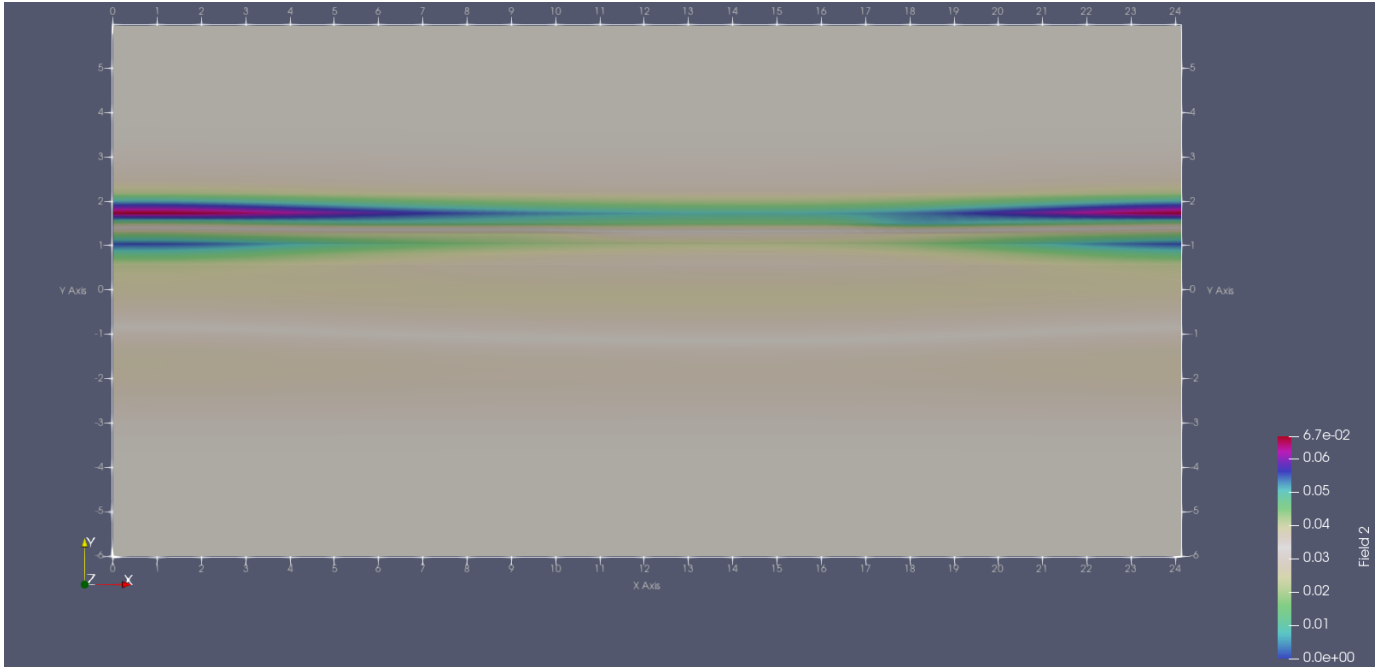


FIGURE 3 – KEEN N=512, dt=.5, T=500

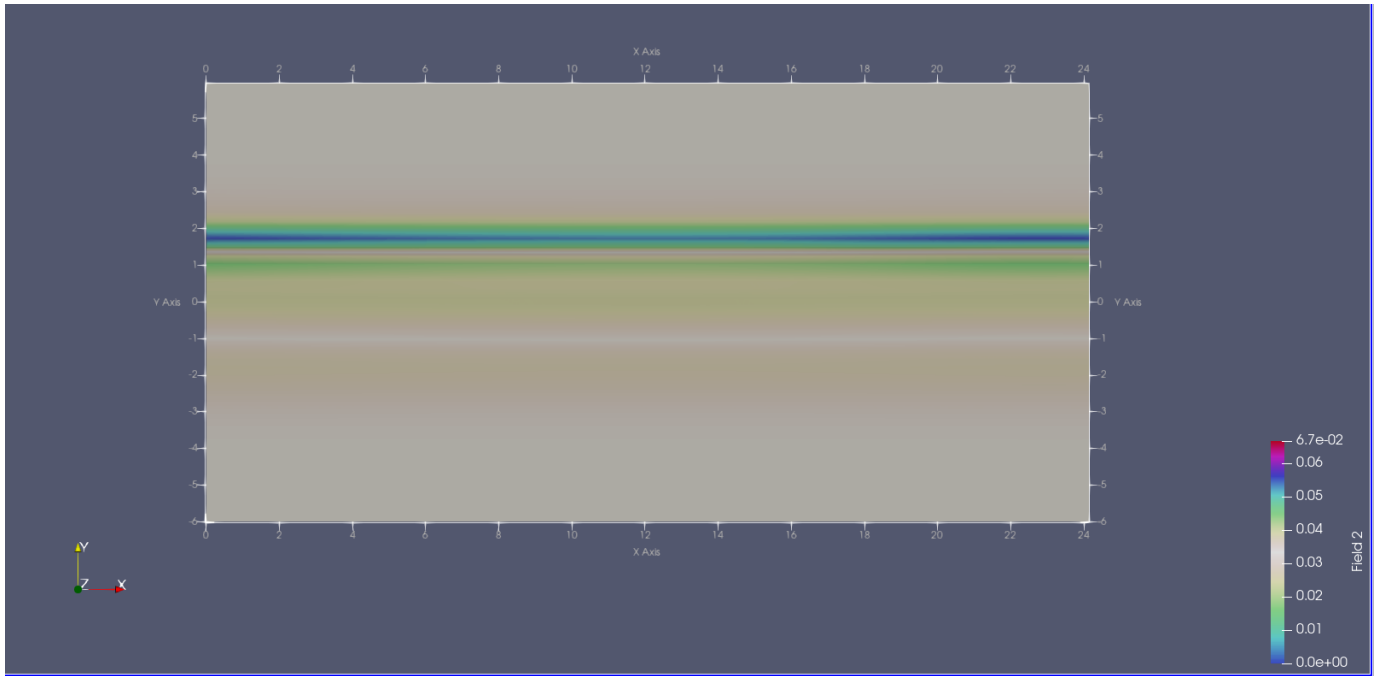


FIGURE 4 – KEEN $N=512$, $dt=.5$, $T=1000$

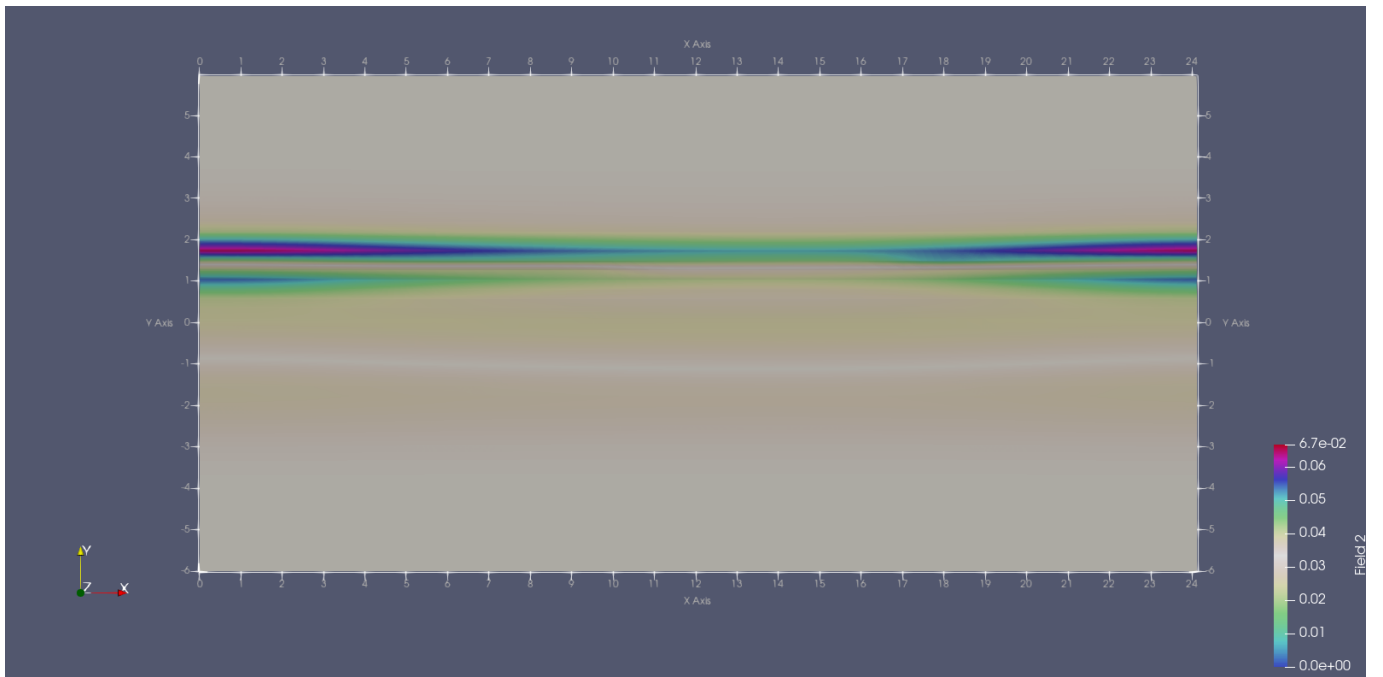


FIGURE 5 – KEEN $N=512$, $dt=.1$, $T=500$

3.5 Choix du splitting

Du fait de l'implémentation, il est facile d'adapter le code à un splitting d'ordre plus élevée. Pour ce faire dans la partie de résolution, il suffit d'appeler chaque résolution des sous équations avec le pas en argument. Il faut aussi tenir compte du calcul du champ électrique. De ce fait, on préférera minimiser la résolution du transport en espace. En effet, le transport en vitesse quand à lui ne modifie pas le champ électrique (condition périodique).

3.6 Difficultés rencontrés, Amélioration

L'implémentation de la résolution en elle même n'est pas très difficile. Ici la difficulté est placée au niveau de la grille. En effet, de manière à prendre en compte la périodicité de manière naturelle, on implémente des assesseurs et mutateurs recalculant les indice de manière modulaire. Un problème que l'on a rencontré est alors le modulo entre un *int* et un *unsigned int*. L'entier signé étant promu en non-signé, le résultat est alors faux dans le cas où le signé est négatif.

Une autre difficultés était de savoir pour la FFT, dans quel ordre était stockés les composantes fréquentielles du signal.

D'un point de vue sur les améliorations possible, il est envisageable d'améliorer la parallélisation avec OpenMP, en effet certaine routine pourrait être réécrite avec un coup mémoire plus élevée mais plus facilement distribuable. Il serait aussi possible d'utiliser MPI de manière à réduire les contrainte de mémoire.

Au niveau delà FFT, des améliorations au niveau de l'étape de planification sont possible, mais nos tentative n'étaient pas fructueuse.

La gestion du cas test analytique, peut être réécrite proprement à l'aide de l'héritage. En effet, cela permettrai de récrire une fonction donnant le champ électrique sans avoir à trop modifier le reste du code.

3.7 Utilisation du code

Ce code est constitué de plusieurs fichier source et d'une dépendance à *FFTW3*. Pour faciliter l'installation de cette librairie, nous utilisons Conan. C'est un gestionnaire de librairie permettant une installation locale des librairies requises pour un code C++. Cela permet de récupérer une librairie construite, si disponible pour la plateforme de l'utilisateur, ou d'automatiser la construction de facon à la rendre transparente à l'utilisateur. Conan utilise le fichier 'conanfile.txt' pour lister les librairies nécessaire et l'outil que l'on utilise pour générer notre Makefile (ici CMake).

Ayant traité la partie librairie, on cherche maintenant à compiler notre code. Pour ce faire, on utilise CMake, il permet de générer notre Makefile dans notre cas. L'intégration de OpenMP et des librairies installés par Conan est fait de manière simple, comme vous pouvez le voir dans le fichier 'CMakeLists.txt'. CMake se charge alors de créer le fichier nécessaire pour make. De plus, CMake est bien supporté par les IDE adaptés au C++, comme Qt Creator ou JetBrains CLion.

Il est alors possible d'utiliser make pour pouvoir construire le programme et ensuite être en mesure de l'exécuter. Il est aussi possible de régler dans le shell les variables d'environnements liés à OpenMP, de manière à limiter le nombre de threads utiliser par exemple.

Finalement, les étapes nécessaire pour pouvoir créer les exécutables sont les suivantes :

```
$ cd Projet
(Projet) $ mkdir build && cd build
(Projet/build) $ conan install ..
(Projet/build) $ cmake ..
(Projet/build) $ make
(Projet/build) $ ./Projet
```

Référence

- Cheng, Y., Gamba, I.M. Morrison, P.J. Study of conservation and recurrence of Runge–Kutta discontinuous Galerkin schemes for Vlasov–Poisson systems. J Sci Comput 56, 319–349 (2013). <https://doi.org/10.1007/s10915-012-9680-x>
- Vlasov on GPU (VOG Project), M. Mehrenberger, C. Steiner, L. Marradi, N. Crouseilles, E. Sonnendrucker, B. Afeyan.