UNIVERSITÉ DE LORRAINE | BIBLIOTHÈQUES UNIVERSITAIRES

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact bibliothèque : ddoc-theses-contact@univ-lorraine.fr
(Cette adresse ne permet pas de contacter les auteurs)

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Méthode adaptative en vitesse pour les codes Semi-Lagrangiens : application aux instabilités Weibel relativistes

THÈSE

présentée et soutenue publiquement le 25 septembre 2025

pour l'obtention du

Doctorat de l'Université de Lorraine

Spécialité: Physique des plasmas

par

Maxence ANTOINE

Composition du jury

Président du jury :	M. d'Humières Emmanuel	Professeur des Universités, CELIA, Université de Bordeaux
Rapporteur :	M. Crouseilles Nicolas	Directeur de Recherche, IRMAR, Université de Rennes
Examinateurs :	M. Daniele Del Sarto	Maître de Conférences, IJL, Université de Lorraine
	Mme. Firpo Marie-Christine	Directrice de recherche, LPP, Sorbonne Université
Encadrants :	M. Alain Ghizzo	Professeur des Universités, IJL, Université de Lorraine
	M. Erwan Deriaz	Chargé de recherche, ZFAI, Université de Beihang

Institut Jean Lamour | UMR CNRS 7198, Nancy, France,
Université de Lorraine

Table des matières

1 Introduction **5**

1.1 Introduction générale sur les plasmas 6

1.1.1 Définition 6

1.1.2 Classification des plasmas 6

1.2 Modèles en physique des plasmas 8

1.2.1 Modèle à N corps 9

1.2.2 Modèle cinétique 10

1.2.3 Théorie de la dérive cinétique et modèle gyrocinétique 12

1.2.4 Modèle fluide 12

1.2.5 Modèle de la magnétohydrodynamique 14

1.2.6 Modèles et simulations 14

1.3 Équation de Vlasov 14

1.3.1 Formulation relativiste 15

1.3.2 Invariants de Casimir 17

1.4 Simulation de l'équation de Vlasov : PIC VS Semi-Lagrangien 18

1.4.1 Codes PIC 19

1.4.2 Codes Semi-Lagrangiens 21

1.4.3 Comparaison des méthodes 22

1.4.4 Problème de la filamentation 23

1.5 Méthodes de raffinement de maillage 25

1.6 Instabilités de type Weibel 27

1.6.1 Instabilités en physique des plasmas 27

1.6.2 Instabilités Weibel 29

1.6.3 Instabilités CFI, TSI et Oblique . . . 30
1.6.4 Exemples physiques de ces instabilités . . . 32
1.7 Objectifs de la thèse . . . 36

2 Présentation du code VLEM **37**
2.1 Historique . . . 38
2.2 Méthode Semi-Lagrangienne . . . 39
2.2.1 Normalisation du code . . . 39
2.2.2 Problème du "time splitting" . . . 40
2.2.3 Schéma numérique . . . 42
2.2.4 Parallélisations et architecture de VLEM . . . 44
2.2.5 Rotation dans l'espace des vitesses . . . 45
2.2.6 Advections spatiales locales . . . 48
2.2.7 Évolution des champs électromagnétiques . . . 54
2.2.8 Résumé du schéma numérique global . . . 60
2.3 Modifications apportées au cours de la thèse . . . 61
2.3.1 Optimisation du temps de calcul . . . 61
2.3.2 Implémentation de nouveaux diagnostics . . . 63
2.4 Développement de VLEM3D3V . . . 67
2.4.1 Problème de la mémoire . . . 67
2.4.2 Modification à onze et dix points . . . 67
2.4.3 Modification de la méthode d'advection à six points . . . 68
2.4.4 Méthode de la quasi-interpolation . . . 69
2.4.5 Comparaison entre les méthodes . . . 73

3 Implémentation des grilles emboîtées dans le code VLEM **77**
3.1 Explication de la méthode . . . 78
3.1.1 Stratégie de raffinement . . . 78
3.1.2 Représentation des grilles raffinées . . . 80
3.2 Choix du critère de raffinement . . . 81
3.3 Implémentation dans VLEM . . . 83
3.3.1 Initialisation de la grille raffinée . . . 83

3.3.2 Évolution temporelle de la grille 84
3.3.3 Modification du schéma numérique global 89
3.3.4 Modification du calcul des intégrales 91
3.4 Validation de la méthode 97
3.4.1 Transition de OI à CFI avec VLEM2D2V 98
3.4.2 Transition de OI à CFI avec VLEM2D3V 110
3.4.3 Cas relativiste de l'instabilité oblique avec VLEM2D2V 115
3.4.4 Test de performance numérique 118
3.5 Possibilité d'amélioration du raffinement 121
3.5.1 Grille non rectangulaire 121
3.5.2 Modification spatiale de la grille raffinée 122
3.5.3 Raffinement en espace avec les ondelettes d'Arakawa 123

4 Études physiques avec le code VLEM_EG 125
4.1 Étude de l'instabilité oblique relativiste à 1010keV 126
4.1.1 Transition vers CFI 127
4.1.2 Impact de la température du faisceau froid 129
4.1.3 Évolution des énergies et de la fonction de distribution 130
4.1.4 Analyse du cas oblique 1010keV 133
4.2 Étude de l'instabilité oblique relativiste à 2000keV 139
4.2.1 Transition OI vers CFI 141
4.2.2 Évolution des énergies et de la fonction de distribution 142
4.3 Étude de l'instabilité Weibel relativiste 145
4.3.1 Comparaison de trois fonctions de distributions 146
4.3.2 Instabilité Weibel relativiste 146

5 Méthode multi-faisceaux 2D3V 155
5.1 Méthode de réduction Hamiltonienne 156
5.2 Modification du "time splitting" 158
5.3 Modification du schéma numérique 160
5.3.1 Modification de la décomposition de domaine MPI 161
5.4 Calcul des coefficients pour déterminer les moments 166

5.5 Transition de OI à CFI avec VLEM2D3V multi-faisceaux 168
5.5.1 Nature électrostatique-électromagnétique de la perturbation initiale 168
5.5.2 Processus de saturation non linéaire 170
5.6 Discussion sur la méthode et compatibilité avec l'AMR 175
5.6.1 Avantages principaux du modèle 175
5.6.2 Limites et ouverture du modèle 176
5.6.3 Raffinement de maillage multi-faisceaux 177

6 Conclusions et perspectives **179**
6.1 Conclusions 179
6.2 Perspectives 180

A Démonstration de l'ordre 2 du schéma numérique **183**
A.1 Formulation Lagrangienne en utilisant les caractéristiques de l'équation de Vlasov 183
A.2 Formulation Eulérienne 187

B Complément sur VLEM3D3V **191**
B.1 Valeurs des $\breve{\omega}$ 191
B.2 Coefficients ω sous forme rationnelle 192
B.2.1 Méthode de la dérivée première $\partial_x s(x_i)$ 192
B.2.2 Méthode de la quasi-interpolation 193
B.3 Estimation de l'erreur L^∞ des méthodes 193
B.3.1 Méthode de la quasi-interpolation 193
B.3.2 Méthode d'estimation de la dérivée première 194

C Description des méthodes de raffinement par ondelettes **197**
C.1 Rappel sur les ondelettes 197
C.2 Description de VLEM_EG par ondelettes 198
C.3 Description des ondelettes splines 199
C.4 Description des ondelettes d'Arakawa 199

D Études des conditions aux bords **201**

D.1 Méthodes des conditions aux bords lors de la rotation 201

D.1.1 Schéma centré 202

D.1.2 Dérivée seconde nulle 202

D.1.3 Schéma décentré 202

D.1.4 Méthode des splines 203

D.1.5 Méthode de la quasi-interpolation 204

D.2 Comparaison numérique des méthodes d'estimation de la dérivée seconde 205

D.2.1 Grille évolutive 205

D.2.2 Grille fixe 206

E Valeur du paramètre C dans le cas 3V **209**

Bibliographie **215**

Remerciements

Comme le veut la coutume, je vais commencer ce manuscrit par remercier les gens sans qui cette thèse aurait très certainement eu une autre saveur. Le lecteur qui souhaite uniquement en apprendre plus sur les méthodes adaptatives en vitesse ou sur les codes Semi-Lagrangiens peut directement se rendre au chapitre 1. Pour les autres, soit vous espérez voir votre nom, soit vous aimez beaucoup en apprendre plus sur la vie privée des gens.

Tout d'abord, j'aimerais remercier *Nicolas Crouseilles* et *Emmanuel d'Humières* d'avoir été les rapporteurs de cette thèse, et également *Marie-Christine Firpo* et *Daniele Del Sarto* de les avoir rejoints dans le jury. Les échanges et les questions au cours de la soutenance ont été très agréables et me poussent à vouloir modifier encore et encore le code.

J'aimerais également remercier *Alain Ghizzo*, le directeur de cette thèse, qui a été un excellent encadrant, avec des échanges très complets, tant sur la physique que sur des points techniques qui concernent VLEM, que ce soit au cours de réunions organisées ou à l'improviste. Merci également à *Erwan Deriaz* d'avoir encadré cette thèse, d'abord depuis Strasbourg, puis depuis la Chine, mais toujours avec des réunions en visio très bénéfiques. Merci aussi à toi pour l'invitation en Chine, qui a été très enrichissante, tant pour ma thèse que pour ma culture personnelle. Merci également à toutes les personnes que j'ai rencontrées là-bas, et qui ont rendu le séjour agréable.

Revenons en France, et plus particulièrement à l'IJL. Même si je n'ai pas travaillé avec grand monde (et non, pas de remerciement aux CC ici), je tiens à remercier tous ceux qui ont rendu les déambulations dans les couloirs distrayantes et qui font partie du charme du laboratoire. Que ce soient les permanents qui procrastinent autant que leurs étudiants, ou les fortes têtes qui crient tout le temps.

Plus précisément, merci à tous les membres de l'équipe 107, particulièrement *Stéphane* pour avoir été mon premier contact à Nancy, *Étienne* pour toujours avoir ta porte ouverte pour toutes mes questions, et merci encore à *Daniele* d'avoir un peu été comme mon deuxième directeur de thèse pour les points physiques et pour l'administratif le jour de la soutenance.

J'aimerais également remercier ma famille de s'être déplacée en nombre le jour de ma soutenance. J'ai essayé d'être pédagogue, donc j'espère que vous ne vous êtes pas ennuyés. Merci à tous ceux qui n'ont pas pu venir, que ce soit par empêchement professionnel ou par souci de santé, et particulièrement *Mamielou*, qui, j'en suis sûr, aurait adoré être présente le jour de ma soutenance ou m'aider à relire ma thèse.

Merci aussi à toutes mes connaissances d'Amiens, toujours prêtes à aller boire un coup quand je reviens pour les vacances. Je vais pouvoir commencer à faire une liste non exhaustive, sinon je vais me faire réprimander, merci donc à *Alexandre* (*Mhbb*), *Simon*, *Bastien*, *Nico*, *Youn*, *Pauline*, *Twingue*, les connaissances du lycée, les gens qui

squattaient à la coloc de *Simon* ou les connaissances de bar.

Merci aussi à tous les gens avec qui j'ai grandi et me suis amusé, que ce soit depuis la fac, le lycée, le collège et même la primaire.

Pour les copains du labo, je vais faire une petite énumération, dont l'ordre est non exhaustif, donc merci à :

- ◇ *Agathe* (et *Timothounet* d'amour) pour les afters gin-Fortnite jusqu'à 5h, les potins et les rires honnêtes.
- ◇ *Alan*, pour les JDR, les poulets rôtis, les gâteaux et les dates devant la LEC.
- ◇ *Anna*, pour les bières le jour du pot et pour avoir appris à *Loris* que le puits était optimal au pierre-feuille-ciseaux.
- ◇ *Ce mec*, qui vient toujours pour faire des trucs bizarres, comme mordre le cactus.
- ◇ *Clavel*, pour m'avoir fait découvrir toutes sortes de plats et de boissons (toutes alcoolisées).
- ◇ *Clélia*, pour m'avoir appris à entraîner la souplesse dans mes jambes (j'ai réussi une hyperextension !).
- ◇ *Clovis*, pour ce super séjour à Reims qui m'a fait découvrir la merveilleuse histoire de *frère Clément* et de son invention.
- ◇ *Corentin*, pour les afters chez toi pendant que tu dors (tout en étant avec nous !) et les pizzas qui rendent malade.
- ◇ *Cypriiienhhh*, pour être le plus grand fournisseur de vélos de Nancy et pour m'avoir fait découvrir les bienfaits de l'imprimante.
- ◇ ***Guillaume***, moi je te réponds ***feur***.
- ◇ *Jean*, pour toutes ces heures à réfléchir et travailler ensemble.
- ◇ *Justine*, rendez-vous mercredi prochain à phi (valable peu importe la date de lecture).
- ◇ *Killian*, pour cette fameuse soirée marcel et pour l'invitation au brassage (promis on fait ça cette année).
- ◇ *Loris*, qui est toujours le premier à être motivé et toujours le dernier à partir qu'il s'agisse de nouba, de partie de cartes, de parler aux futures GGF ou tout simplement de thèse.
- ◇ *Lucie*, pour les petits mots à la craie devant chez moi (merci aussi pour les craies), et pour avoir été la meilleure voisine que j'aie jamais eue.
- ◇ *Merlin*, pour la résolution de tous ces mystères et la soirée dans les Vosges avec *Alain*.
- ◇ *Nathan* (qui n'a remercié presque personne dans sa thèse et m'a quand même battu ... bref), pour m'avoir montré tes games de TFT et les films de ton enfance pour que je critique.
- ◇ *Niamh* et *Vital*, mes deux super co-bureaux, arrivés tout deux pendant ma dernière année. Vous avez été une super découverte toujours plein de joie et très silencieux.

◇ *PA*, pour toujours savoir quand partir du groupe parce que tu as dit une bêtise.
◇ *Rasmus* et toute l'équipe pour les travaux.
◇ *Romain*, pour être mon identité officielle au CNRS.
◇ *Sebastien* (lequel suspense...), pour l'aide lors de la soutenance et pour nous montrer à tous ce qu'est une vraie journée de travail.
◇ *Theo*, pour ces multiples soirées chez toi (les voisinous attendent encore pour l'anniversaire d'Agathe) qu'elles soient cocktail, LEC ou JDR.
◇ *Thomas*, qui n'est jamais vraiment parti du labo et qui est toujours là pour le café (sauf visiblement pour ma soutenance).
◇ *Timothé*, pour les heures de procrastination offertes chaque semaine depuis le stage de M1, tu seras toujours le numéro 1 sur mes échelles de comparaison.

Désolé à tous ceux qui n'ont pas eu de blagues, j'avais soit pas d'inspiration, soit pas la place mais voici quand même vos noms : *Adèle*, *Antoine*, *Arthur*, *Leïla*, *Leonel*, *Mauricio*, *Romain*, *Roua*, *Vivien* et tous ceux qui espéraient voir leurs noms.

Merci à *Mariam* qui a certainement dû penser que je l'ai oubliée.

Chapitre 1

Introduction

Sommaire

1.1 Introduction générale sur les plasmas . . . 6
1.1.1 Définition . . . 6
1.1.2 Classification des plasmas . . . 6
1.2 Modèles en physique des plasmas . . . 8
1.2.1 Modèle à N corps . . . 9
1.2.2 Modèle cinétique . . . 10
1.2.3 Théorie de la dérive cinétique et modèle gyrocinétique . . . 12
1.2.4 Modèle fluide . . . 12
1.2.5 Modèle de la magnétohydrodynamique . . . 14
1.2.6 Modèles et simulations . . . 14
1.3 Équation de Vlasov . . . 14
1.3.1 Formulation relativiste . . . 15
1.3.2 Invariants de Casimir . . . 17
1.4 Simulation de l'équation de Vlasov : PIC VS Semi-Lagrangien 18
1.4.1 Codes PIC . . . 19
1.4.2 Codes Semi-Lagrangiens . . . 21
1.4.3 Comparaison des méthodes . . . 22
1.4.4 Problème de la filamentation . . . 23
1.5 Méthodes de raffinement de maillage . . . 25
1.6 Instabilités de type Weibel . . . 27
1.6.1 Instabilités en physique des plasmas . . . 27
1.6.2 Instabilités Weibel . . . 29
1.6.3 Instabilités CFI, TSI et Oblique . . . 30
1.6.4 Exemples physiques de ces instabilités . . . 32
1.7 Objectifs de la thèse . . . 36

1.1 Introduction générale sur les plasmas

1.1.1 Définition

La première utilisation du mot « plasma » pour désigner un état de la matière a été faite par Irving Langmuir en 1928 dans l'article [1]. Les raisons de ce choix ne sont pas connues, mais il existe deux pistes. La première serait qu'il viendrait du mot grec "plasma", qui signifie "modelable" [2]. La deuxième hypothèse serait qu'il proviendrait du mot médical désignant le plasma sanguin, les deux étant constitués de plusieurs espèces/corpuscules [2, 3]. On peut cependant le définir de la manière suivante :

Définition : Un plasma est un fluide composé d'ions positifs, d'électrons et de neutres.

Dans cette définition, les neutres sont des molécules ou des atomes de gaz, tandis que les ions positifs désignent des molécules ou des atomes ayant reçu suffisamment d'énergie pour pouvoir être ionisés une ou plusieurs fois, c'est-à-dire qu'ils ont perdu un ou plusieurs électrons.

Le plasma est donc un état de la matière différent des trois principaux états connus (solide, liquide et gazeux), ce qui explique qu'il soit souvent considéré comme le "quatrième état de la matière". Actuellement, on estime que le plasma compose 99% de l'univers observable [4], même s'il est très rare sur Terre.

On peut séparer les plasmas en deux catégories en fonction de la proportion de neutres qui les compose, que l'on quantifie par le paramètre α de la manière suivante :

$$\alpha = \frac{n_e}{n_e + n}, \tag{1.1}$$

avec n_e la densité des électrons et n la densité de neutres. Usuellement, en physique des plasmas, la densité s'exprime comme le nombre de particules par unité de volume en m^{-3}.

Si $\alpha \sim 1$, on est dans le cas d'un plasma dit "chaud", qui sera le cadre d'étude de cette thèse. Tandis que si $\alpha \sim 0$ on est dans le cas d'un plasma dit "froid". Cette terminologie "chaud/froid" est principalement française et, dans les autres pays, on préfère parler de plasma hautement ou faiblement ionisé.

1.1.2 Classification des plasmas

Afin de différentier et de caractériser les plasmas, on va introduire d'autres variables qui définiront les échelles de temps et d'espace de notre système.

En l'absence de collisions, si l'on perturbe un plasma stable et homogène en déplaçant une couche d'électrons, on peut retrouver une pulsation d'oscillation de ces

électrons, que l'on appelle fréquence plasma électronique :

$$\omega_{p_e} = \sqrt{\frac{n_e e^2}{m_e \epsilon_0}}, \tag{1.2}$$

avec e et m_e respectivement la charge et la masse de l'électron et ϵ_0 la permittivité du vide. Cette pulsation, va donner l'ordre de grandeur des échelles de temps de notre système.

Afin de donner un ordre de grandeur des échelles d'espace de notre système, on peut s'intéresser à la longueur typique d'écrantage d'une charge dans un plasma. C'est-à-dire, la longueur caractéristique à partir de laquelle le potentiel électrique créé par cette charge devient négligeable. Cette longueur s'appelle longueur de Debye et s'écrit dans sa version électronique comme :

$$\lambda_{D_e} = \sqrt{\frac{\epsilon_0 k_B T_e}{e^2 n_e}}, \tag{1.3}$$

avec T_e la température des électrons et k_B la constante de Boltzmann.

La fréquence plasma et la longueur de Debye ont été données ici pour les électrons, mais peuvent être écrites pour chaque espèce en remplaçant n_e, m_e, T_e et e respectivement par la densité, la masse, la température et la charge de l'espèce considérée. Afin de les exprimer de manière générale pour tout type de particules, on utilisera les indices s.

Si on s'intéresse maintenant au nombre de particules contenues dans un volume de Debye, on trouve :

$$N_D = n_s \lambda_{D_s}^3, \tag{1.4}$$

en prenant l'inverse de cette quantité, on trouve le paramètre de grain g :

$$g = \frac{1}{n_s \lambda_{D_s}^3}. \tag{1.5}$$

Ce paramètre de grain va permettre d'estimer le taux de collisionalité tel qu'un plasma très peu collisionnel aura $g \simeq 0$ tandis qu'un plasma collisionnel aura $g \simeq 1$. La figure 1.1 présente différents types de plasmas en fonction de leur densité et de leur température. On peut considérer que les plasmas situés au dessus de la ligne $g \simeq 10^{-4}$ sont peu collisionnels et sont donc dominés par des effets collectifs. Tandis que ceux situés en dessous de la ligne $g \simeq 1$ sont dominés par les collisions et donc par des effets individuels.

Dans cette thèse, on va s'intéresser à des plasmas peu collisionnels.

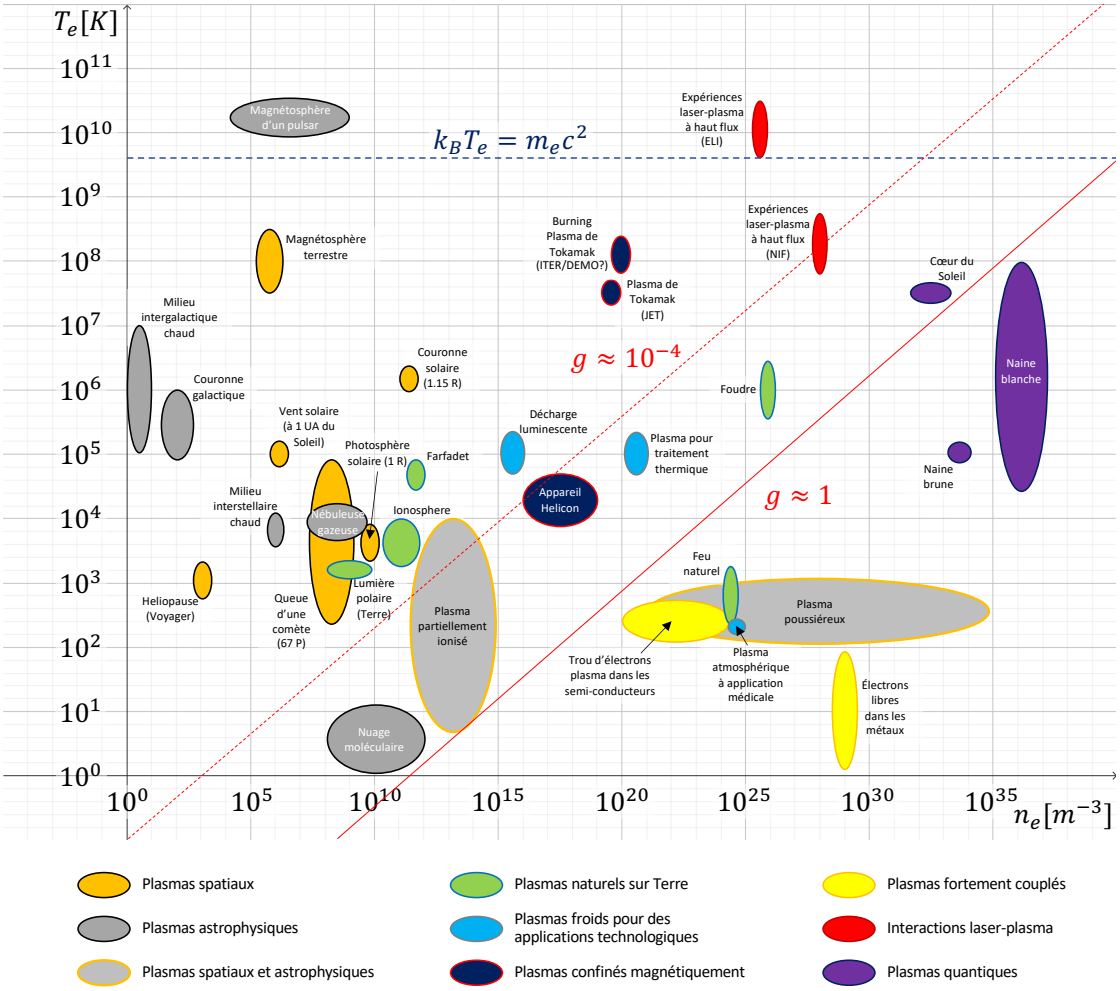

FIGURE 1.1 – Figure issue de [5] représentant une variété de plasmas en fonction de leur densité et de leur température. Les indications entre parenthèses désignent l'objet ou l'endroit de référence. La ligne rouge continue et la ligne rouge pointillée représentent respectivement le paramètre de grain fixé à $g \simeq 1$ et $g \simeq 10^{-4}$. La ligne bleue pointillée représente la température $T_e = m_e c^2/k_B$.

1.2 Modèles en physique des plasmas

Afin de comprendre l'évolution d'un plasma et les phénomènes qui lui sont associés il est utile de le décrire par un modèle physique adapté. Ces modèles sont plus ou moins précis et plus ou moins complexes, et le choix parmi ces derniers dépendra du problème à étudier.

1.2.1 Modèle à N corps

Le choix le plus proche de la réalité, et probablement le plus intuitif, pour comprendre l'évolution d'un plasma est de regarder la trajectoire de chaque particule et les forces qu'elles exercent les unes sur les autres. On définit la fonction de distribution décrivant ces N_s particules, avec l'indice s désignant le type de particule ($s = e$ pour les électrons), comme :

$$F_s = \sum_{n=1}^{N_s} \delta(\mathbf{r} - \mathbf{r_n}(\mathbf{t}))\delta(\mathbf{v} - \mathbf{v_n}(\mathbf{t})), \tag{1.6}$$

avec δ la distribution de Dirac, $\mathbf{r_n}(\mathbf{t})$ la position dans l'espace de la particule n et $\mathbf{v_n}(\mathbf{t})$ la vitesse qui lui est associée. Cette fonction de distribution localise donc chaque particule par ses coordonnées de position et de vitesse, c'est-à-dire par ses coordonnées dans l'espace des phases (**r**,**v**).

On va maintenant chercher une équation permettant de décrire l'évolution temporelle de cette distribution. En appliquant la dérivée temporelle, on obtient [6] :

$$\frac{\partial F_s}{\partial t} = \sum_{n=1}^{N_s} \frac{d\mathbf{r_n}}{dt} \cdot \frac{\partial F_s}{\partial \mathbf{r_n}} + \sum_{n=1}^{N_s} \frac{d\mathbf{v_n}}{dt} \cdot \frac{\partial F_s}{\partial \mathbf{v_n}}. \tag{1.7}$$

En utilisant le fait que $d\mathbf{r_n}/dt = \mathbf{v_n}$ et $d\mathbf{v_n}/dt = \mathbf{a_n} = \hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{n}}/m_n$, avec $\mathbf{a_n}$ l'accélération de la particule, $\hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{n}}$ la force qu'elle subit et m_n sa masse, on peut réécrire :

$$\begin{aligned} \frac{\partial F_s}{\partial t} = &\sum_{n=1}^{N_s} \mathbf{v_n} \cdot \frac{\partial}{\partial \mathbf{r_n}} \left(\delta(\mathbf{r} - \mathbf{r_n}(\mathbf{t}))\delta(\mathbf{v} - \mathbf{v_n}(\mathbf{t}))\right) \\ &+ \sum_{n=1}^{N_s} \frac{\hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{n}}}{m_n} \cdot \frac{\partial}{\partial \mathbf{v_n}} \left(\delta(\mathbf{v} - \mathbf{v_n}(\mathbf{t}))\delta(\mathbf{r} - \mathbf{r_n}(\mathbf{t}))\right). \end{aligned} \tag{1.8}$$

En appliquant les propriétés de la distribution de Dirac, on trouve :

$$\begin{aligned} \frac{\partial F_s}{\partial t} = &\sum_{n=1}^{N_s} \mathbf{v} \cdot \frac{\partial}{\partial \mathbf{r_n}} \left(\delta(\mathbf{r} - \mathbf{r_n}(\mathbf{t}))\delta(\mathbf{v} - \mathbf{v_n}(\mathbf{t}))\right) \\ &+ \sum_{n=1}^{N_s} \frac{\hat{\boldsymbol{\mathcal{F}}}}{m} \cdot \frac{\partial}{\partial \mathbf{v_n}} \left(\delta(\mathbf{v} - \mathbf{v_n}(\mathbf{t}))\delta(\mathbf{r} - \mathbf{r_n}(\mathbf{t}))\right). \end{aligned} \tag{1.9}$$

Comme F_s est une somme de distribution de Dirac, on peut appliquer les formules suivantes [6] :

$$\sum_{n=1}^{N_s} \frac{\partial F_s}{\partial \mathbf{r_n}} = -\frac{\partial F_s}{\partial \mathbf{r}}, \qquad \sum_{n=1}^{N_s} \frac{\partial F_s}{\partial \mathbf{v_n}} = -\frac{\partial F_s}{\partial \mathbf{v}}. \tag{1.10}$$

On peut donc retrouver l'équation de Klimontovich [7] :

$$\frac{\partial F_s}{\partial t} + \mathbf{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{r}} F_s + \frac{\hat{\boldsymbol{\mathcal{F}}}}{m} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}} F_s = 0. \tag{1.11}$$

Pour un plasma, les forces exercées sont les forces électromagnétiques tel que l'on puisse écrire :

$$\hat{\boldsymbol{\mathcal{F}}} = q_s \left(\hat{\mathbf{E}}(\mathbf{r}, t) + \mathbf{v} \times \hat{\mathbf{B}}(\mathbf{r}, t) \right), \tag{1.12}$$

avec $\hat{\mathbf{E}}$ et $\hat{\mathbf{B}}$ respectivement les champs électrique et magnétique au point $\mathbf{r}$ et au temps t.

L'équation de Klimontovich a donc besoin d'être couplée aux équations de Maxwell :

$$\begin{aligned} \boldsymbol{\nabla} \times \hat{\mathbf{E}} &= -\frac{\partial \hat{\mathbf{B}}}{\partial t}, \qquad & \boldsymbol{\nabla} \cdot \hat{\mathbf{E}} &= \frac{\rho}{\epsilon_0}, \\ \boldsymbol{\nabla} \times \hat{\mathbf{B}} &= \mu_0 \hat{\mathbf{J}} + \frac{1}{c^2} \frac{\partial \hat{\mathbf{E}}}{\partial t}, \qquad & \boldsymbol{\nabla} \cdot \hat{\mathbf{B}} &= 0, \end{aligned} \tag{1.13}$$

avec ρ la densité de charge, μ_0 la perméabilité du vide, c la vitesse de la lumière et $\hat{\mathbf{J}}$ le courant au point $\mathbf{r}$ et au temps t.

L'équation de Klimontovich est donc la représentation la plus proche de la réalité pour décrire un plasma. Cependant, pour décrire des cas réalistes, elle est beaucoup trop complexe, puisqu'elle nécessite de suivre la trajectoire de N_s particules qui interagissent toutes entre elles. Or dans les plasmas le nombre de particules est souvent très important. En effet, si on se réfère à la figure 1.1, on remarque que les plasmas de plus faible densité s'étendent sur des volumes très importants (héliopause, milieu inter-galactique, magnétosphère ...) et donc le nombre de particules pour décrire ces systèmes est énorme. A l'inverse, là où le volume est plus faible, comme par exemple ITER avec un volume estimé en début 2025 de $830m^3$ [8], la densité de particules est plus importante ($10^{20}m^{-3}$), ce qui donne une estimation de l'ordre de $N_s \sim 10^{23}$. On note que même si la densité du plasma dans ITER est plus importante que dans certains milieux astrophysiques, elle reste très faible, environ six ordres de grandeur en dessous de la densité de l'air.

Le nombre de particules dans les plasmas est donc trop important pour pouvoir être étudié directement. Il est donc nécessaire de trouver un modèle plus simple pour les décrire.

1.2.2 Modèle cinétique

Une option permettant de simplifier l'équation de Klimontovich consiste à lisser cette somme de particules localisées en faisant la moyenne sur un petit volume 6D de l'espace des phases [9]. L'espace des phases désigne un espace dans lequel la vitesse $\mathbf{v}$

est une dimension. On peut écrire la moyenne d'une quantité A sur un volume $\delta^3 r \times \delta^3 v$ tel que :

$$\langle A \rangle = \frac{1}{\delta^3 r \delta^3 v} \int \int W\left(\frac{\mathbf{r} - \mathbf{r}'}{\delta^3 r}\right) W\left(\frac{\mathbf{v} - \mathbf{v}'}{\delta^3 v}\right) A d\mathbf{r}' d\mathbf{v}', \tag{1.14}$$

avec W une fonction rectangle à trois dimensions telle que :

$$W(\mathbf{z} = z_1, z_2, z_3) = \begin{cases} 1 & \text{si} |z_i| < 1, \forall i = 1,2,3 \\ 0 & \text{sinon} \end{cases}. \tag{1.15}$$

À l'intérieur d'un tel volume, les particules peuvent être considérées comme non corrélées et exprimées comme une densité de distribution à une seule particule [9].

En appliquant cette moyenne à l'équation de Klimontovich (1.11), on obtient la forme suivante :

$$\frac{\partial \langle F_s \rangle}{\partial t} + \langle \mathbf{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{r}} F_s \rangle + \left\langle \frac{\hat{\boldsymbol{\mathcal{F}}}}{m} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}} F_s \right\rangle = 0, \tag{1.16}$$

que l'on peut réécrire comme :

$$\frac{\partial \langle F_s \rangle}{\partial t} + \mathbf{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{r}} \langle F_s \rangle + \frac{\left\langle \hat{\boldsymbol{\mathcal{F}}} \right\rangle}{m} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}} \langle F_s \rangle = \mathcal{C}, \tag{1.17}$$

avec $\mathcal{C}$ la variable qui représente les fluctuations de vitesses et de forces par rapport à la moyenne [9]. Comme on néglige la corrélation entre les particules en faisant une approximation de champ moyen, on peut prendre $C = 0$. On retrouve donc l'équation de Vlasov classique comme :

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{r}} f_s + \frac{\boldsymbol{\mathcal{F}}}{m} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}} f_s = 0, \tag{1.18}$$

avec $\boldsymbol{\mathcal{F}} = \left\langle \hat{\boldsymbol{\mathcal{F}}} \right\rangle$ et $f_s = \langle F_s \rangle$. Comme la corrélation entre les particules, et donc les effets de collision, ont été négligés, on se trouve désormais désormais dans le cadre des plasmas non collisionnels, c'est-à-dire avec un paramètre de grain $g \to 0$.

Une autre méthode permettant de retrouver l'équation de Vlasov à partir de celle de Klimontovich consiste à intégrer étape par étape le volume associé à chaque particule. Puis, de la même manière qu'ici, les termes de corrélation entre les particules sont négligés. Cette méthode est décrite dans les références [10, 11].

Cette équation de Vlasov (Eq. (1.18)) nécessite toujours d'être couplée aux équations de Maxwell, dans lesquelles les champs sont également moyennés dans le cadre de l'approximation de champ moyen. C'est-à-dire qu'il faut reprendre les équations (1.13) en modifiant les variables tel que $\hat{\mathbf{E}} \to \mathbf{E}$, $\hat{\mathbf{B}} \to \mathbf{B}$, $\hat{\mathbf{J}} \to \mathbf{J}$, avec $\mathbf{E}$, $\mathbf{B}$ et $\mathbf{J}$ respectivement les champs électrique, magnétique et les courants moyennés sur un volume de l'espace des phases.

On verra par la suite, que cette dernière peut être résolue numériquement. C'est d'ailleurs cette équation , une fois réécrite dans sa version relativiste (voir sous-partie 1.3.1), qui constituera l'équation principale de cette thèse.

1.2.3 Théorie de la dérive cinétique et modèle gyrocinétique

En présence d'un fort champ magnétique, il est possible de réduire le nombre de dimensions de l'équation de Vlasov en se basant sur la trajectoire d'une particule chargée. En effet, dû à la force de Lorentz, une particule va effectuer un mouvement de rotation autour des lignes de champs magnétiques, ce mouvement s'appelle le mouvement cyclotron. Le rayon de cette rotation est le rayon de Larmor r_L et la fréquence de rotation est la fréquence cyclotron ω_c. On appelle centre-guide la position autour de laquelle la particule va tourner.

Lorsque le champ magnétique est important, le rayon de Larmor est très petit devant les longueurs caractéristiques du plasma. On va donc pouvoir approximer le mouvement de la particule par le mouvement de son centre-guide. C'est la théorie de la dérive cinétique, où on va étudier comment se déplace ce centre permettant de réduire le nombre de variables de l'équation de Vlasov de sept ($\mathbf{r}$, $\mathbf{v}$, t) à six ($\mathbf{r}$, $v_{//}$, μ,t), avec $v_{//}$ la vitesse de la particule parallèle à la ligne de champ magnétique et $\mu = m_s v_\perp^2/(2B)$ le moment magnétique.

On va donc considérer que l'on étudie uniquement la vitesse de ce centre-guide. Cette dernière est composée de la vitesse de dérive v_D, orthogonale aux lignes de champs magnétiques, et $v_{//}$ pour la direction parallèle. On peut écrire l'équation de dérive cinétique à partir de l'équation de Vlasov comme :

$$\frac{\partial f_s}{\partial t} + \mathbf{v_{D_s}} \cdot \boldsymbol{\nabla}_r f_s + \frac{dv_{//}}{dt}\frac{\partial}{\partial v_{//}} f_s = 0, \tag{1.19}$$

cette dernière est couplée à l'invariance adiabatique du moment magnétique μ.

L'inconvénient de ce modèle est qu'il est trop restrictif. En effet, supposer que la trajectoire de la particule est identique à celle du centre-guide conduit à des erreurs dans les résultats qui faussent la physique du problème.

Un modèle plus complexe est le modèle gyrocinétique, qui fait intervenir un opérateur de gyro-moyenne afin de prendre en compte les effets de rayon de Larmor fini, c'est-à-dire le fait que la particule n'est pas située sur son centre-guide. La gyromoyenne doit donc être effectuée sur toutes les équations du système, les rendant plus complexes mais permettant de garder seulement cinq variables (plus le temps) pour le système [12].

Le modèle gyrocinétique est très souvent utilisé pour décrire les plasmas de réacteurs dans lesquels la condition du fort champ magnétique est imposée. C'est par exemple le cas de GENE [13], Gysela [14] ou encore Teresa [15].

1.2.4 Modèle fluide

L'équation de Vlasov reste assez complexe à résoudre puisque la fonction de distribution dépend des six variables de l'espace des phases ($\mathbf{r}$, $\mathbf{v}$) plus le temps (t) et

même si le modèle gyrocinétique permet de le réduire à cinq (plus t), cela reste coûteux numériquement. Afin de réduire ce nombre de variables, il est possible de moyenner sur l'espace des vitesses. Chaque espèce composant le plasma aura donc une vitesse moyenne et pourra être comparée à un fluide.

On va donc multiplier l'équation de Vlasov par une fonction $\Psi(\mathbf{v})$ et intégrer cela sur l'espace des vitesses ($\mathbf{v}$) :

$$\int \Psi \frac{\partial f_s}{\partial t} d\mathbf{v} + \int \Psi \mathbf{v} \cdot \boldsymbol{\nabla}_{\boldsymbol{r}} f_s d\mathbf{v} + \int \Psi \frac{\boldsymbol{\mathcal{F}}}{m} \cdot \boldsymbol{\nabla}_{\boldsymbol{v}} f_s d\mathbf{v} = 0. \tag{1.20}$$

En fonction du choix de $\Psi(\mathbf{v})$, on va pouvoir retrouver un ensemble d'équation. En prenant $\Psi = 1$, c'est-à-dire le premier moment, et en considérant qu'il n'y a ni création ni destruction de particules, on trouve l'équation de continuité :

$$\frac{\partial n_s}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{r}} \cdot (n_s \mathbf{u_s}) = 0, \tag{1.21}$$

avec n_s la densité de l'espèce s dans le plasma, que l'on définit comme $n_s = \int f_s d\mathbf{v}$ et $\mathbf{u_s}$ la vitesse moyenne du fluide de l'espèce s définie comme $\mathbf{u_s} = \langle v \rangle = \int f_s \mathbf{v} d\mathbf{v} / n_s$, avec la moyenne sur les vitesses qui s'écrit $\langle A \rangle = \int f_s A d\mathbf{v} / n_s$. La moyenne utilisée ici est différente de celle de la sous-partie 1.2.2.

Cette équation seule n'est pas suffisante pour décrire entièrement l'évolution de notre espèce s, puisque l'on a quatre variables (n_s, u_{x_s}, u_{y_s} et u_{z_s}). On va donc remplacer dans l'équation (1.20) $\Psi = \mathbf{v_i}$, en utilisant la notation d'Einstein. Ψ est ici le deuxième moment et on trouve l'équation fluide du mouvement :

$$n_s m_s \left[\frac{\partial \mathbf{u_s}}{\partial t} + (\mathbf{u_s} \cdot \boldsymbol{\nabla}) \mathbf{u_s} \right] = -\boldsymbol{\nabla} \cdot \boldsymbol{\Pi} + n_s q_s (\mathbf{E} + \mathbf{u_s} \times \mathbf{B}) + \cdots, \tag{1.22}$$

avec $\boldsymbol{\Pi}$ le tenseur de pression qui s'écrit comme $\Pi_{ij} = n_s m_s \langle (v_i - u_{i_s})(v_j - u_{j_s}) \rangle$ et "$\cdots$" permettant de prendre en compte l'impact de forces supplémentaires.

À nouveau, afin de décrire le tenseur de pression, il faut trouver l'équation utilisant un moment d'ordre plus élevée. En prenant le troisième moment, c'est-à-dire $\Psi = v_i v_j$, on trouve l'équation du transport de l'énergie :

$$\begin{aligned} \frac{\partial \Pi_{s_{ij}}}{\partial t} + \frac{\partial Q_{s_{ijk}}}{\partial x_k} + \frac{\partial (\Pi_{s_{ij}} u_{k_s})}{\partial x_k} + \Pi_{s_{jk}} \frac{\partial (u_{i_s})}{\partial x_k} + \Pi_{s_{ik}} \frac{\partial (u_{j_s})}{\partial x_k} \\ = \frac{q_s}{m_s} \left[\epsilon_{jlm} \Pi_{s_{li}} + \epsilon_{ilm} \Pi_{s_{lj}} \right] B_m, \end{aligned} \tag{1.23}$$

avec $Q_{ijk} = n_s m_s \langle (v_i - u_{i_s})(v_j - u_{j_s})(v_k - u_{k_s}) \rangle$ le flux de chaleur et ϵ_{ijk} le tenseur de Levi-Civita.

On observe que à chaque fois que l'on introduit un nouvelle ordre, un tenseur d'ordre plus élevé apparaît. Il faut donc introduire une condition de fermeture sur notre

ensemble d'équations afin de pouvoir les résoudre. Cette condition va dépendre du problème que l'on étudie.

On note que les équations obtenues ici sont très proches des équations obtenues en mécanique des fluides excepté que les forces exercées sont les forces électromagnétiques et non celle de gravitation.

1.2.5 Modèle de la magnétohydrodynamique

Le modèle fluide est très utile pour résoudre analytiquement, et même numériquement, une variété de problèmes. Cependant, ce dernier nécessite un groupe d'équations fluide pour chaque espèce de particules. C'est pourquoi, une simplification supplémentaire va être de rassembler les équations fluides des ions ($s = i$) et des électrons ($s = e$) afin de décrire le plasma comme une seule espèce. Cette simplification s'appelle la magnétohydrodynamique (MHD).

1.2.6 Modèles et simulations

Les modèles présentés dans ce sous-chapitre ont été introduits du plus détaillé (N corps) au moins détaillé (MHD), mais aussi du plus lourd numériquement au plus léger. En effet, le nombre de dimensions et d'équations des modèles va fortement influer sur le coût numérique ou la difficulté analytique de leur utilisation.

Il existe cependant des modèles potentiellement plus simples que le modèle MHD, appelés modèles réduits. Ces derniers sont souvent basés sur des conservations ou des géométries particulières et ne peuvent pas être appliqués à tous les phénomènes.

Il faut donc adapter le modèle au problème étudié et à la précision souhaitée. Dans cette thèse, on va s'intéresser à l'équation de Vlasov, donc à une description cinétique, afin de pouvoir décrire l'évolution de la fonction de distribution dans l'espace des phases. On va par ailleurs utiliser la version relativiste de cette équation permettant ainsi de décrire les phénomènes à haute énergie.

1.3 Équation de Vlasov

Le sous-chapitre précèdent a présenté les principaux types de modèles utilisés en physique des plasmas. On va dans ce sous-chapitre regarder de plus près un modèle cinétique au travers de l'équation de Vlasov relativiste et des invariants qui l'accompagnent. Cette équation s'avère nécessaire si on souhaite décrire des fonctions de distributions relativistes pour décrire, par exemple, de fortes accélérations de particules.

1.3.1 Formulation relativiste

Afin de retrouver l'équation de Vlasov dans sa version relativiste, plutôt que de moyenner l'équation de Klimontovich (Eq. (1.11)) sur un volume de l'espace des phases, on va utiliser le formalisme Hamiltonien. À partir de maintenant, on ne considérera plus qu'une seule espèce : les électrons. Les indices s ont donc été retirés. Les calculs vont être faits pour un espace à cinq dimensions, une pour le temps (t), deux pour l'espace (x,y) et deux pour les quantités de mouvements (p_x, p_y) qui remplacent dans la version relativiste les vitesses (v_x, v_y). Cette réduction du nombre de dimensions du système est souvent réalisée afin d'étudier des processus physiques ne nécessitant pas un espace complet. Cela permet de réduire la difficulté numérique et analytique du problème.

On donne la différentielle de la fonction de distribution que l'on écrit comme :

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{dx}{dt}\frac{\partial f}{\partial x} + \frac{dy}{dt}\frac{\partial f}{\partial y} + \frac{dp_x}{dt}\frac{\partial f}{\partial p_x} + \frac{dp_y}{dt}\frac{\partial f}{\partial p_y} = 0. \tag{1.24}$$

On va chercher les valeurs de dx/dt, dy/dt, dp_x/dt et dp_y/dt. Pour cela, on prend l'Hamiltonien relativiste pour une particule de charge e et de masse m dans un potentiel électrique qui s'écrit comme :

$$H = \gamma m c^2 + e\phi, \tag{1.25}$$

avec ϕ le potentiel électrique et γ le facteur relativiste. Ce facteur relativiste est aussi appelé facteur de Lorentz et s'écrit comme :

$$\gamma = \frac{1}{\sqrt{1 - \frac{v_x^2 + v_y^2}{c^2}}}, \tag{1.26}$$

en utilisant le fait que $v_\alpha = \frac{p_\alpha}{\gamma m}$, on retrouve γ en fonction des quantités de mouvements :

$$\gamma = \sqrt{1 + \frac{p_x^2 + p_y^2}{m^2 c^2}}. \tag{1.27}$$

On rappelle les équations canoniques d'Hamilton :

$$\begin{aligned} \frac{dx}{dt} &= \frac{\partial H}{\partial P_x} \qquad \frac{dy}{dt} = \frac{\partial H}{\partial P_y} \\ \frac{dP_x}{dt} &= -\frac{\partial H}{\partial x} \quad \frac{dP_y}{dt} = -\frac{\partial H}{\partial y} \end{aligned} \tag{1.28}$$

avec $\mathbf{P}$ le moment canonique qui est lié à la quantité de mouvement par la relation :

$$\mathbf{P} = \mathbf{p} + e\mathbf{A}, \tag{1.29}$$

avec $\mathbf{A}$ le potentiel vecteur tel que $\mathbf{B} = \boldsymbol{\nabla} \times \mathbf{A}$.

D'un coté, pour la dérivée temporelle des coordonnées spatiales on a :

$$\frac{dx}{dt} = \frac{\partial H}{\partial P_x} = mc^2 \frac{\partial}{\partial P_x} \sqrt{1 + \frac{(P_x - eA_x)^2 + p_y^2}{m^2 c^2}} = \frac{(P_x - eA_x)}{\gamma m} = \frac{p_x}{\gamma m}, \tag{1.30}$$

$$\frac{dy}{dt} = \frac{\partial H}{\partial P_y} = mc^2 \frac{\partial}{\partial P_y} \sqrt{1 + \frac{p_x^2 + (P_y - eA_y)^2}{m^2 c^2}} = \frac{(P_y - eA_y)}{\gamma m} = \frac{p_y}{\gamma m}. \tag{1.31}$$

De l'autre, pour les dérivées temporelles des moments canoniques, on a :

$$\frac{dP_x}{dt} = -\frac{\partial H}{\partial x} = e\frac{\partial A_x}{\partial x}\frac{p_x}{\gamma m} + e\frac{\partial A_y}{\partial x}\frac{p_y}{\gamma m} - e\frac{\partial \phi}{\partial x}, \tag{1.32}$$

$$\frac{dP_y}{dt} = -\frac{\partial H}{\partial y} = e\frac{\partial A_x}{\partial y}\frac{p_x}{\gamma m} + e\frac{\partial A_y}{\partial y}\frac{p_y}{\gamma m} - e\frac{\partial \phi}{\partial y}. \tag{1.33}$$

En utilisant le fait qu'on puisse aussi les écrire à partir de l'équation (1.29), on a :

$$\frac{dP_x}{dt} = \frac{dp_x}{dt} + e\frac{\partial A_x}{\partial t} + e\frac{\partial A_x}{\partial x}\frac{p_x}{\gamma m} + e\frac{\partial A_x}{\partial y}\frac{p_y}{\gamma m}, \tag{1.34}$$

$$\frac{dP_y}{dt} = \frac{dp_y}{dt} + e\frac{\partial A_y}{\partial t} + e\frac{\partial A_y}{\partial x}\frac{p_x}{\gamma m} + e\frac{\partial A_y}{\partial y}\frac{p_y}{\gamma m}. \tag{1.35}$$

En combinant les équations (1.32) et (1.34) et les équations (1.33) et (1.35), on trouve :

$$\frac{dp_x}{dt} = -e\frac{\partial \phi}{\partial x} - e\frac{\partial A_x}{\partial t} + e\left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}\right)\frac{p_y}{\gamma m}, \tag{1.36}$$

$$\frac{dp_y}{dt} = -e\frac{\partial \phi}{\partial y} - e\frac{\partial A_y}{\partial t} + e\left(\frac{\partial A_x}{\partial y} - \frac{\partial A_y}{\partial x}\right)\frac{p_x}{\gamma m}. \tag{1.37}$$

En utilisant les relations suivantes :

$$\mathbf{E} = -\boldsymbol{\nabla}\phi - \frac{\partial \mathbf{A}}{\partial t}, \qquad \mathbf{B} = \boldsymbol{\nabla} \times \mathbf{A}, \tag{1.38}$$

on trouve la forme suivante pour les dérivées temporelles des quantités de mouvement :

$$\frac{dp_x}{dt} = e(E_x + \frac{p_y}{\gamma m} B_z), \tag{1.39}$$

$$\frac{dp_y}{dt} = e(E_y - \frac{p_x}{\gamma m} B_z). \tag{1.40}$$

En injectant les équations (1.30), (1.31), (1.39) et (1.40) dans l'équation (1.24), on retrouve l'équation de Vlasov relativiste :

$$\frac{\partial f}{\partial t} + \frac{\boldsymbol{p}}{m\gamma} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} f + e\left(\boldsymbol{E} + \frac{\boldsymbol{p} \times \boldsymbol{B}}{m\gamma}\right) \cdot \boldsymbol{\nabla}_{\boldsymbol{p}} f = 0. \tag{1.41}$$

Cette dernière est aussi valable pour les problèmes à sept dimensions (t, x, y, z, p_x, p_y, p_z).

L'équation de Vlasov n'est pas résolvable seule puisqu'elle dépend des champs électromagnétiques et doit donc être couplée aux équations de Maxwell. Ce couplage se fera par l'intermédiaire de la densité n et des densités de courant $\boldsymbol{J}$ que l'on peut écrire, en réintroduisant les indices s pour les différentes espèces, comme :

$$n_s(\boldsymbol{x}) = \iiint f_s(\boldsymbol{x}, \boldsymbol{p}) d^3p, \tag{1.42}$$

$$\boldsymbol{J}(\boldsymbol{x}) = \sum_s q_s \iiint \frac{\boldsymbol{p}}{m\gamma} f_s(\boldsymbol{x}, \boldsymbol{p}) d^3p. \tag{1.43}$$

1.3.2 Invariants de Casimir

Une fonction de distribution dont l'évolution est définie par l'équation de Vlasov (Eq.(1.41)) possèdent des quantités qui dépendent de la fonction de distribution et qui sont invariantes dans le temps. Ces quantités s'appellent des invariants de Casimir. On peut par exemple citer la densité n ou l'entropie S et leurs conservations lors d'une simulation numérique permettra de vérifier que le problème est bien résolu.

On propose ici une démonstration de la conservation temporelle de ces invariants. On considère une fonction $F(f)$ et on définit les invariants de Casimir comme :

$$C(t) = \int F(f) d^3x d^3p, \tag{1.44}$$

avec $C(t)$ ne dépendant que du temps car f dépend de $\mathbf{x}$ et $\mathbf{p}$.

On montre la conservation de ces invariants en écrivant :

$$\frac{dC(t)}{dt} = \int \frac{dF(f)}{df} \frac{\partial f}{\partial t} d^3x d^3p. \tag{1.45}$$

En remplaçant $\frac{\partial f}{\partial t}$ dans l'équation de Vlasov (Eq. (1.41)), on peut écrire :

$$\frac{dC(t)}{dt} = \int \frac{dF(f)}{df} \left[-\frac{\mathbf{p}}{m\gamma} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} f - e(\mathbf{E} + \frac{\boldsymbol{p} \times \boldsymbol{B}}{m\gamma}) \cdot \boldsymbol{\nabla}_{\boldsymbol{p}} f \right] d^3x d^3p, \tag{1.46}$$

$$\frac{dC(t)}{dt} = -\int \frac{\mathbf{p}}{m\gamma} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} F d^3x d^3p - \int e\left(\mathbf{E} + \frac{\boldsymbol{p} \times \boldsymbol{B}}{m\gamma}\right) \cdot \boldsymbol{\nabla}_{\boldsymbol{p}} F d^3x d^3p. \tag{1.47}$$

On va étudier les deux intégrales séparément. D'un coté on a :

$$\int \frac{\mathbf{p}}{m\gamma} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} F d^3x d^3p = \int \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\frac{\mathbf{p}}{m\gamma} F \right) d^3x d^3p, \tag{1.48}$$

car $\mathbf{p}/m\gamma$ ne dépend pas de l'espace. En utilisant le théorème de la divergence sur une surface sphérique Ω de rayon infini on a :

$$\int \frac{\mathbf{p}}{m\gamma} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} F d^3x d^3p = \int \left(\int_\Omega \left(\frac{\mathbf{p}}{m\gamma} F \right) \cdot d^2S \right) d^3p = 0, \tag{1.49}$$

car la fonction $F(f)$ est nulle à l'infini.

De l'autre coté, on a :

$$\begin{aligned} \int e \left(\mathbf{E} + \frac{\mathbf{p} \times \mathbf{B}}{m\gamma} \right) \cdot \boldsymbol{\nabla}_{\boldsymbol{p}} F d^3x d^3p = & \left[e \left(\mathbf{E} + \frac{\mathbf{p} \times \mathbf{B}}{m\gamma} \right) F \right]_{-\infty}^{\infty} \\ & - \int F \boldsymbol{\nabla}_{\boldsymbol{p}} \cdot e \left(\mathbf{E} + \frac{\mathbf{p} \times \mathbf{B}}{m\gamma} \right) d^3x d^3p = 0. \end{aligned} \tag{1.50}$$

De la même manière, le premier terme de droite est nul car $F(f)$ est nulle pour une quantité de mouvement à l'infini. Le deuxième est nul car E ne dépend pas de $\mathbf{p}$ et que $\nabla_p \cdot \frac{\mathbf{p} \times \mathbf{B}}{m\gamma} = 0$.

On a donc, en rassemblant les équations (1.49) et (1.50) :

$$\frac{dC(t)}{dt} = 0. \tag{1.51}$$

$C(t)$ est donc bien un invariant temporel pour une fonction de distribution dont l'évolution est décrite par l'équation de Vlasov.

On note cependant que si on prend en compte les collisions, il faut rajouter un terme dans l'équation (1.46). Ce dernier va donner la possibilité à certains invariants $F(f)$ d'évoluer dans le temps.

La démonstration a pour but de justifier l'existence des invariants de Casimir. Mais ces derniers peuvent aussi être justifiés par le théorème de Liouville qui impose la réversibilité de la fonction de distribution dont l'évolution est décrite par l'équation de Vlasov.

1.4 Simulation de l'équation de Vlasov : PIC VS Semi-Lagrangien

L'équation de Vlasov, couplée aux équations de Maxwell, permet de décrire l'évolution d'une fonction de distribution dans l'espace des phases. Cependant, en raison

de la complexité de cette résolution due au caractère non linéaire de ce couplage, les simulations numériques sont souvent privilégiées.

Deux des principales méthodes pour résoudre le système Vlasov/Maxwell sont les codes Lagrangiens à particules appelés PIC [16, 17] pour "Particles In Cell" en anglais, que nous présenterons dans la prochaine sous-partie, et les codes Eulériens [18], qui résolvent l'évolution de la fonction de distribution sur une grille de l'espace des phases.

Il existe plusieurs types de méthodes Eulériennes comme les méthodes de résolution par éléments finis [19, 20], les méthodes spectrales [21, 22], dans lesquelles la fonction de distribution en vitesse est écrite à l'aide de fonctions de base (comme Fourier ou Hermite) afin d'obtenir un système d'équations de moment. Il existe également la méthode des volumes finis qui consiste à résoudre le problème sur des volumes de l'espace des phases au lieu de le résoudre sur des points. Elle prend la forme d'une balance de flux. Cette méthode a été introduite pour la première fois pour l'équation de Vlasov par Fijalkow (1999) [23]. Une version améliorée est la méthode de flux conservatif et positif (PFC) [24] ou plus récemment [25, 26], qui est non seulement conservative, mais qui préserve également la positivité et le maximum de la fonction de distribution. Enfin, la dernière méthode Eulérienne qui va nous intéresser est la méthode Semi-Lagrangienne [27, 28, 29, 30] qui utilise les caractéristiques de l'équation de Vlasov pour résoudre l'évolution de la fonction de distribution sur les points. Nous allons développer cette méthode dans la sous-partie 1.4.2.

Nous avons choisi de comparer les codes PIC, qui sont à ce jour les plus utilisés dans la communauté de la physique des interactions laser-plasma, aux codes Semi-Lagrangiens, qui sont l'objet d'étude de cette thèse.

1.4.1 Codes PIC

Afin de résoudre l'évolution de la fonction de distribution cinétique, les codes PIC utilisent des macro-particules représentant un certain nombre de particules données. Ces codes suivent alors la trajectoire de ces macro-particules dans l'espace des phases, en prenant en compte l'accélération subie en raison des forces électromagnétiques, ainsi que leurs déplacements dus à leurs quantités de mouvement (ou vitesses). La figure 1.2 représente à gauche le mouvement de ces macro-particules et leurs interactions avec les champs électromagnétiques.

Ces codes vont discrétiser les champs magnétique et électrique sur une grille de l'espace [16, 17]. Afin de déterminer les forces exercées sur chaque particule, qui sont de taille finie, des fonctions de formes ("shape function") sont utilisées, ce qui permet de prendre en compte la géométrie des particules. Les modifications de la quantité de mouvement de la particule ainsi que de sa trajectoire vont être effectuées à l'aide d'une méthode appelée "particle pusher" qui utilise les caractéristiques de l'équation de Vlasov (Eqs.(1.30-1.31) et Eqs.(1.39-1.40)). Ces fonctions de formes vont ensuite être utilisées pour déterminer la densité de charge et les courants sur les points de grille à

partir de la position des particules, ce qui permet de résoudre les équations de Maxwell (flèches rouges sur la figure 1.2).

La plupart des codes PIC sont basés sur une approche explicite en temps, c'est-à-dire que la trajectoire des particules et l'évolution des champs vont être résolues itérativement les unes après les autres. On peut citer par exemple OSIRIS [31], EPOCH [32], SHARP [33], SMILEI [17], VPIC [34], WARPX [35] ou PHARE [36]. Il existe cependant de grosses limitations pour les codes explicites sur les pas de temps et d'espace [37], dont les valeurs dépendent de l'épaisseur de peau locale ($d_e = c/\omega_p$) et de l'inverse de la fréquence plasma (ω_p^{-1}). En présence de forts gradients de densité ces valeurs peuvent varier considérablement à l'intérieur du domaine. Ainsi seules ces régions nécessiteraient des grilles plus fines et des pas de temps plus petits, mais les méthodes explicites ont l'inconvénient d'imposer des paramètres de simulation restrictifs dans l'ensemble du domaine. Pour ces raisons, des recherches approfondies ont été consacrées au développement de codes PIC implicites. Ceux-ci impliquent essentiellement une discrétisation implicite des équations de Maxwell, ainsi qu'un "particle pusher" qui peut être couplé de manière non linéaire (ce qui donne le solveur PIC implicite complet [38, 39]) ou ne pas être couplé au solveur de champs électromagnétiques (ce qui conduit dans ce dernier cas aux codes PIC semi-implicites tels que ceux de [40, 37]).

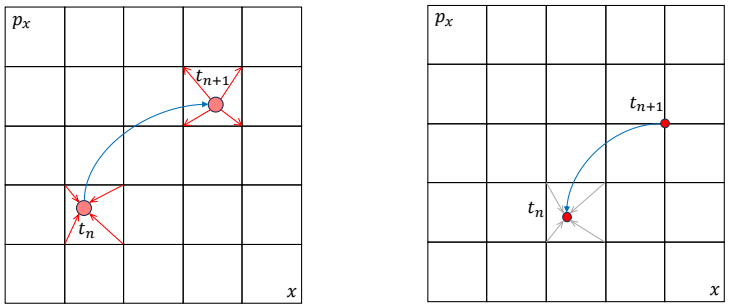

FIGURE 1.2 – Schématisation des méthodes PIC (à gauche) et SL (à droite). Pour les codes PIC, les cercles représentent une macro-particule dont on va suivre la trajectoire afin de résoudre l'évolution de la fonction de distribution. Les flèches rouges indiquent les interactions entre la particule et les champs électromagnétiques définis sur la grille uniforme. Ces derniers entraînent des forces sur la particule, ce qui la met en mouvement. La particule va ensuite faire évoluer les champs électromagnétiques au travers de la densité de charge et des courants. Tandis que pour les codes Semi-Lagrangiens, la fonction de distribution est définie sur les points de grille (cercle rouge à t_{n+1}) et ses valeurs évoluent à l'aide des caractéristiques de l'équation de Vlasov vers l'arrière suivies d'une interpolation (cercle rouge à t_n et flèches grises).

La simplicité de fonctionnement des codes PIC, tant dans la théorie que dans l'implémentation, ajoutée à leur faible coût numérique, explique pourquoi cette méthode est l'une des plus utilisées en physique des plasmas. Elle présente toutefois un inconvénient : le bruit numérique dû à la taille finie des macro-particules [41].

1.4.2 Codes Semi-Lagrangiens

Les codes PIC sont des codes Lagrangiens, puisqu'ils suivent la trajectoire des particules, à l'opposé des méthodes Eulériennes qui observent l'évolution de la fonction de distribution sur une grille fixe et résolvent directement l'équation de Vlasov couplée aux équations de Maxwell à l'aide de méthodes comme les différences finies ou les volumes finis. Ces types de méthodes Eulériennes ont donc des conditions CFL qui restreignent le pas de temps.

Les codes Semi-Lagrangiens, quant à eux, couplent les deux méthodes. La fonction de distribution est définie sur des points de grille fixes (Eulérien) et elle est résolue à l'aide des caractéristiques de l'équation de Vlasov (Lagrangien).

L'algorithme original des méthodes Semi-Lagrangiennes a été introduit par Chen et Knorr en 1976 [42] puis a été intégré dans un cadre plus général par Sonnendrücker et al. en 1998 [30], en utilisant une interpolation par spline cubique, et par Nakamura et Yabe en 1999 [43], en utilisant une méthode CIP de l'anglais "Cubic interpolated Propagation". Des méthodes de "splitting" Hamiltonien [44] et de réduction Hamiltonienne ont par la suite été utilisées, mais ce sujet sera traité dans le chapitre 5.

On peut résumer ce schéma Semi-Lagrangien par deux étapes principales : tout d'abord, le transport de la distribution f est effectué à l'aide des caractéristiques de l'équation de Vlasov vers l'arrière, (les codes PIC impliquant plutôt une résolution des caractéristiques vers l'avant). C'est-à-dire que l'on va chercher la position dans l'espace des phases de notre point de grille à l'itération précédente. Ensuite la valeur de f au pied des caractéristiques est calculée en reconstruisant f à partir des points du maillage à l'aide d'une technique d'interpolation. On peut résumer cette étape à l'aide de la figure 1.2 (à droite).

Une explication détaillée de la méthode sera fournie dans le chapitre 2, avec le cas particulier du code VLEM. Elle présente l'avantage, par rapport aux méthodes Eulériennes, de ne pas être soumise aux conditions CFL sur l'évolution de la fonction de distribution. Il y aura cependant des conditions CFL dues à la méthode de calcul des champs électrique et magnétique mais cette dernière est moins forte (voir sous-partie 2.2.7). De plus, les méthodes SL n'ont pas le bruit statistique des codes PIC et possèdent seulement un bruit numérique, bien plus faible, dû à la précision machine. Elles ont cependant l'inconvénient d'être très coûteuses en terme de temps de calcul CPU et/ou en mémoire, surtout lorsque l'on étudie des phénomènes à petite échelle qui nécessitent une haute résolution ou des phénomènes à haute dimensionnalité, c'est-à-dire avec une fonction de distribution qui a un nombre de paramètres importants.

1.4.3 Comparaison des méthodes

Les avantages et les inconvénients des méthodes PIC et des méthodes Semi -Lagrangiennes ont été rappelés dans les deux sous-partie précédentes. Cependant, les efforts numériques à fournir peuvent être un bon indicateur pour choisir entre un code PIC ou un code Semi-Lagrangien. Tout d'abord, pour ce qui est du calcul des champs électromagnétiques les efforts numériques sont les mêmes pour les deux types de méthodes [45]. En effet, pour chaque méthode les champs sont décrits sur une grille spatiale et sont résolus en utilisant les équations de Maxwell (voir sous-partie 2.2.7). Pour un code Vlasov Semi-Lagrangien, on peut écrire l'effort numérique comme le nombre total de points dans l'espace des phases [45, 46] :

$$N_{SL} = N_x^{d_x} N_p^{d_p} = \left(\frac{L}{\Delta x}\right)^{d_x} N_p^{d_p}, \tag{1.52}$$

avec N_x et N_p respectivement le nombre de points dans une direction spatiale et dans une direction de quantité de mouvement. d_x et d_p représentent le nombre de dimensions respectivement spatiales et de quantité de mouvement.

L'effort numérique d'un code PIC peut directement être relié au nombre de particules (N_{PIC}) présentes dans l'espace. On peut l'écrire en fonction de la taille de la boîte de côté L, de la densité des macro-particules n_0 et du pas d'espace Δx :

$$N_{PIC} = n_0 L^{d_x} = n_0 (\Delta x)^{d_x} \left(\frac{L}{\Delta x}\right)^{d_x}, \tag{1.53}$$

Δx est en générale de l'ordre de la longueur de Debye (Eq.(1.3)). On peut donc faire apparaître un paramètre de grain (Eq.(1.5)) dû à la taille finie des macro-particules (g_{PIC}) qui doit être le plus petit possible pour décrire au mieux un plasma non collisionnel [45, 46] ($g_{PIC} \to 0$).

En supposant un échantillonnage équivalent de la coordonnée spatiale et en considérant que l'effort nécessaire pour modifier un point de grille avec la méthode Semi-Lagrangienne est équivalent au déplacement d'une particule avec un code PIC, le rapport entre les efforts numériques d'un code Semi-Lagrangien et d'un code PIC peut être estimé par :

$$\frac{N_{SL}}{N_{PIC}} = g_{PIC} N_p^{d_p}. \tag{1.54}$$

En prenant un nombre de points dans la direction des quantités de mouvement typique et équivalent à ce qui sera étudié par la suite, on choisit $N_p = 100$. Les résultats obtenus en fonction du nombre de dimensions et du paramètre de grain peuvent être représentés sous la forme du tableau 1.1.

Bien que les efforts numériques soient équivalents sur la diagonale, nous voyons clairement que le code Semi-Lagrangien est plus coûteux dans l'espace des quantités

	$d_p = 1$	$d_p = 2$	$d_p = 3$
$g_{PIC} = 10^{-2}$	1	10^2	10^4
$g_{PIC} = 10^{-4}$	10^{-2}	1	10^2
$g_{PIC} = 10^{-6}$	10^{-4}	10^{-2}	1

TABLE 1.1 – Rapport N_{SL}/N_{PIC} pour comparer les codes Semi-Lagrangiens aux codes PIC. Le paramètre de grain associé aux différents phénomènes physiques est représenté dans la figure 1.1.

de mouvement 2D et 3D. Dans le cas 3D, le traitement des effets cinétiques peut être étudié par les deux méthodes avec un coût de calcul équivalent, mais seulement pour une plus petite valeur de g_{PIC}, c'est-à-dire $g_{PIC} \sim 10^{-6}$, qui correspond à un grand nombre de macro-particules, rarement utilisé dans les codes PIC.

Alors qu'en une dimension dans l'espace des quantités de mouvement, le code Semi-Lagrangien offre de meilleures capacités en termes de précision et de description des phénomènes plasma. Le traitement de la dynamique des plasmas pour une dimension supérieure dans l'espace des quantités de mouvement constitue un obstacle technique et un véritable défi pour la physique numérique.

On peut également noter que les codes PIC sont plus difficilement parallélisables que les codes Semi-Lagrangiens puisqu'ils nécessitent un algorithme de balance de charge [17] afin de répartir les particules de manière optimale entre les processus. Tandis que pour les codes Semi-Lagrangiens, un simple découpage spatial entre les processus suffit.

1.4.4 Problème de la filamentation

Les approches PIC et SL sont donc deux techniques complémentaires, qui résolvent toutes deux l'équation de Vlasov en supposant que le paramètre de grain tend vers zéro. Elles décrivent aussi toutes deux une propriété fondamentale de l'équation de Vlasov : la filamentation de f dans l'espace des quantités de mouvement. Alors que les codes PIC sont peu affectés par le problème de la filamentation (mais au prix d'une augmentation considérable du bruit thermique du plasma due à la réintroduction des effets de grain (avec un facteur $g_{PIC} \neq 0$) lié à la taille finie des macro-particules), les codes SL nécessitent généralement un échantillonnage élevé dans l'espace des quantités de mouvement, pour suivre la dynamique des filaments de plus en plus fins qui y apparaissent.

Le problème de la filamentation est une propriété intrinsèque de l'équation de Vlasov contenue dans le terme de "free-streaming" [10] qui s'écrit pour une version non relativiste à trois dimensions (t, x, v) : $\frac{\partial f}{\partial t} + v \cdot \nabla_x f = 0$, et dont la solution exacte dans

l'espace de Fourier s'écrit comme :

$$\hat{f}\left(k, v, t\right) = \hat{f}\left(k, v, 0\right) \exp\left(ikvt\right), \tag{1.55}$$

avec k l'espace de Fourier de la variable x.

La fonction $\hat{f}\left(k, v, t\right)$ oscille donc à une fréquence kt qui augmente linéairement avec le temps. Lorsque cette fréquence atteint la taille d'une cellule Δv, on perd de l'information et la fonction de distribution n'est plus définie précisément [10, 47].

On observe cela sur la figure 1.3, où on a représenté la filamentation obtenue à l'aide d'un code Semi-Lagrangien en négligeant les effets dus aux champs électrique et magnétique et en prenant en compte seulement le "free-streaming". La filamentation va donner des structures de plus en plus fines jusqu'à ce que, en raison de la grille numérique qui devient alors trop grossière, l'on retrouve la fonction de distribution initiale à partir d'un temps de récurrence $T_R = 2\pi/(k_0 \Delta v) \simeq 362,5\omega_p^{-1}$ [10] correspondant à la périodicité de (1.55).

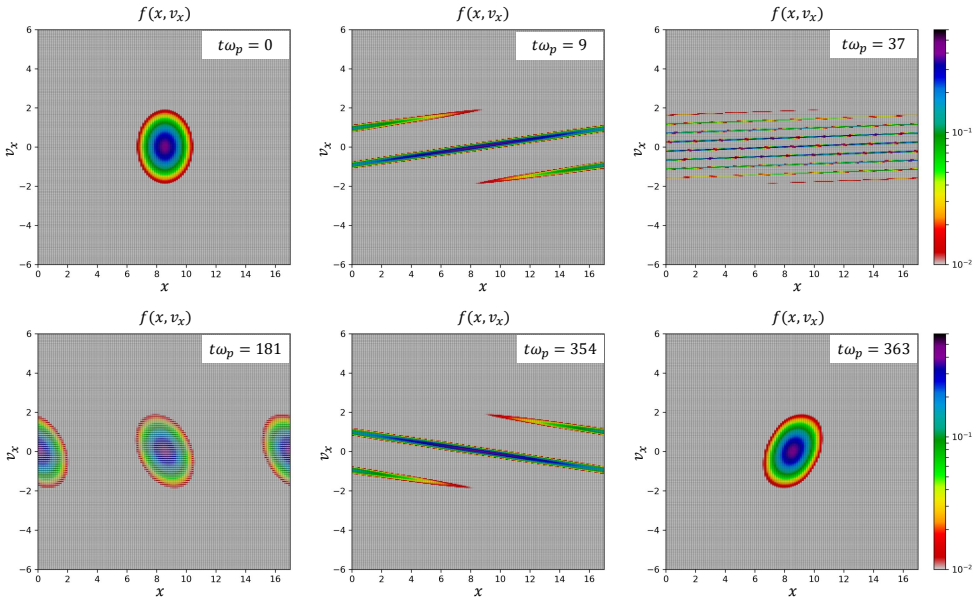

FIGURE 1.3 – Représentation de la filamentation obtenue à l'aide d'un code Vlasov-Poisson Semi-Lagrangien en prenant une distribution initiale Gaussienne en espace et en vitesse.

Même si ce processus est ici "forcé" puisque le "free-streaming" est représenté seul, il est toujours présent dans la résolution de l'équation de Vlasov même dans sa version relativiste. Le couplage de ce "free-streaming" avec les forces électromagnétiques lors de l'évolution de la fonction de distribution peut créer des structures très filamentées dans l'espace des phases et provoquer un chauffage non collisionnel [48]. Il peut également y avoir une perte d'informations irréversible sur ces structures, en raison de la

taille finie de la grille. Certaines instabilités peuvent même conduire à la possibilité de renforcer le processus de filamentation [48].

Une bonne description de la filamentation est donc nécessaire aux codes Semi-Lagrangiens afin de décrire de manière correcte les transferts d'énergie vers les échelles microscopiques. C'est un processus que l'on retrouve dans une variété de plasmas comme dans les instabilités de type Weibel [48], que nous allons introduire par la suite, ou encore pour la reconnexion magnétique [49, 50].

Pour résumer cette partie 1.4, les codes PIC ont l'avantage pour des phénomènes à haute dimensionnalité d'être moins coûteux numériquement que les codes Semi-Lagrangiens. De plus, les codes PIC n'ont aucune difficulté à suivre la filamentation de la fonction de distribution contrairement aux codes SL qui doivent avoir un échantillonage élevé dans l'espace des quantités de mouvement pour pouvoir la suivre. Les codes PIC ont cependant l'inconvénient d'avoir du bruit statistique qui n'est pas présent dans les codes Semi-Lagrangiens. L'absence de bruit des codes SL joue un rôle majeur pour la compréhension exacte des phénomènes physiques mis en jeu surtout lorsque l'on étudie des instabilités qui peuvent être très sensibles à de petites excitations.

Une solution que l'on propose dans cette thèse pour combler la lacune qu'ont les codes Semi-Lagrangiens pour décrire l'espace des vitesses/quantités de mouvement et d'y implémenter le raffinement de maillage.

1.5 Méthodes de raffinement de maillage

Les méthodes de raffinement de maillage adaptatif (AMR, de l'anglais Adaptive Mesh Refinement) sont des techniques numériques de modification du maillage d'un code afin de s'adapter à un problème donné. On les qualifie de dynamique lorsque le maillage évolue au cours du temps.

Les méthodes AMR visent à diminuer le temps de calcul et les ressources en mémoire nécessaires pour simuler un système. Souvent, la situation se résume à un compromis entre un volume numérique (mémoire de l'ordinateur, nombre d'opérations élémentaires, etc.) et une augmentation de la complexité de l'implémentation (structures de données ou algorithmes). L'objectif va être d'affiner localement le maillage d'un problème numérique uniquement là où une plus grande précision est nécessaire. Si la région du domaine où le raffinement est nécessaire est suffisamment petite par rapport à l'ensemble du domaine, l'effort de calcul pour effectuer l'AMR peut être largement compensé par le gain associé au plus petit nombre de points nécessaire pour atteindre une résolution équivalente dans un maillage uniforme. Cette technique est couramment utilisée et s'adapte à plusieurs domaines scientifiques comme la mécanique [51], la physique des plasmas [52, 53] ou encore l'astrophysique [54, 55]. Pour la physique des plasmas et l'étude de l'équation de Vlasov, certains codes ont déjà appliqué des méthodes AMR pour Vlasov-Poisson [56, 57], et pour les codes Vlasov-Maxwell

non relativistes et plus récemment aussi pour les codes Vlasov-Maxwell relativistes [45, 58, 59]. On peut aussi citer des codes utilisant des méthodes hybrides [36, 54, 60].

La méthode AMR va dépendre du type de maillage utilisé dans le code. Dans cette thèse on s'intéressera principalement à une grille cartésienne qui sera affinée localement en une grille cartésienne plus fine. Mais il est aussi possible d'appliquer le raffinement de maillage à des grilles triangulaires (ou tétrahédrique à 3D) [61, 62].

Nous pouvons distinguer deux approches différentes parmi les méthodes AMR. La première est une méthode basée sur les patchs. Les données seront structurées en une collection de grilles de tailles et de précisions variables, qui sont incluses les unes dans les autres [36, 47, 53]. La seconde est l'AMR par arborescence entièrement filée où les points, ou les cellules, sont stockés dans les noeuds ou les feuilles d'une grande structure arborescente [51, 57]. On donne un exemple de ces deux approches différentes sur la figure 1.4 pour le cas d'un trou d'électron que l'on cherche à raffiner dans l'espace des phases.

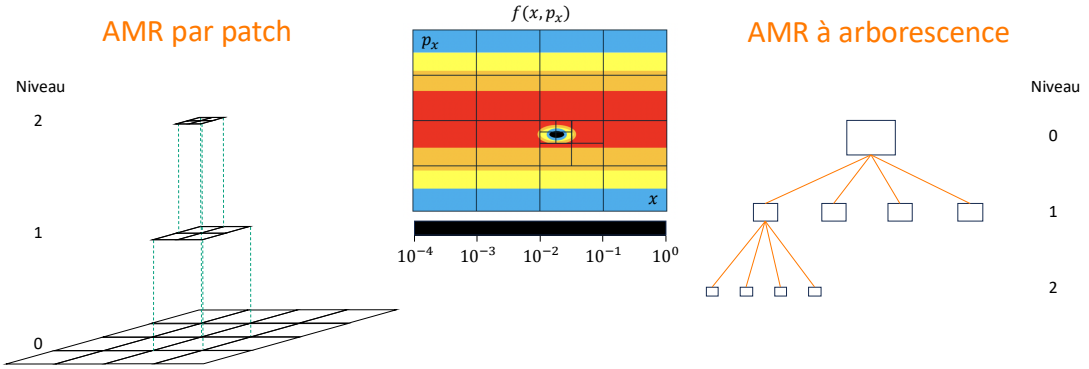

FIGURE 1.4 – Représentation des différents types de raffinement de maillage pour un exemple schématisé d'un trou d'électron que l'on raffine dans l'espace des phases (x, p_x). Ici le raffinement de maillage se fait dans toutes les directions. A gauche la méthode AMR par patch est représentée sur trois niveaux de raffinement. A droite la méthode à arborescence est représentée pour le cas spécifique de la cellule mère étant la seule raffinée.

On appellera niveau de raffinement le groupe de cellules ou de points qui partage la même précision. Ainsi la grille initiale sans raffinement représentera le niveau de raffinement 0. Pour les méthodes par arborescence, on appellera cellule mère, par rapport à une autre cellule, la cellule située sur le niveau grossier le plus proche et cellule fille la cellule sur le niveau raffiné le plus proche. Par exemple pour la figure 1.4 la cellule sur le niveau 1 qui est raffinée aura quatre cellules filles sur le niveau 2 et une cellule mère sur le niveau 0. Par abus de langage, on utilisera parfois ces termes pour désigner le lien entre des points situés au même endroit de l'espace des phases mais

sur des niveaux de grilles différents avec la méthode par patch.

Le raffinement de maillage dynamique n'est pas forcément nécessaire lorsque l'on étudie un problème physique. En effet, si on étudie par exemple l'interaction du vent solaire avec le champ magnétique terrestre, comme c'est le cas pour Vlasiator [60], on connaît la géométrie de notre système et les interfaces qui ont un intérêt à être raffinées. Cependant, si la structure n'est pas prévisible ou si elle est amenée à évoluer au cours du temps, comme c'est le cas par exemple des instabilités Weibel qui seront discutées dans la prochaine partie, un raffinement de maillage dynamique est préférable.

Les techniques de raffinement de maillage adaptatif (AMR) s'avèrent particulièrement utiles dans l'espace des quantités de mouvement, notamment en ce qui concerne le gain en temps de calcul qu'elles peuvent apporter à un solveur Vlasov par rapport à un solveur PIC, avec une résolution spatiale équivalente et un nombre fixe de macroparticules. En effet, l'équation (1.54) montre que le rapport des efforts numériques, à g_{PIC} fixé, dépend uniquement du nombre de points dans l'espace des quantités de mouvement dans le code Semi-Lagrangien. De plus la difficulté qu'ont les codes Semi-Lagrangiens à suivre la filamentation est principalement liée à la précision dans l'espace des quantités de mouvement.

C'est pour ces raisons que l'on a choisi dans cette thèse d'implémenter un raffinement de maillage dans un code Semi-Lagrangien uniquement dans l'espace des quantités de mouvement. Ce choix permet également d'éviter les discontinuités dans l'espace, et donc dans les résolutions des équations de Maxwell. Le sujet d'étude physique principal sera les instabilités de type Weibel car la filamentation dans l'espace des quantités de mouvement peut jouer un rôle primordial dans les échanges d'énergies [48].

1.6 Instabilités de type Weibel

Les instabilités de type Weibel seront donc les phénomènes étudiés au cours de cette thèse. Une introduction sera faite dans cette partie afin de comprendre les raisons physiques de leurs existences et les phénomènes dans lesquels on les retrouve.

1.6.1 Instabilités en physique des plasmas

Avant toute chose, il faut introduire le concept d'instabilité en physique des plasmas.

Un plasma est à l'équilibre s'il reste dans le même état en l'absence de perturbation. Lorsque l'on vient perturber ce système, la perturbation va soit pouvoir croître, dans ce cas le système est instable, soit être amortie, soit le système va osciller. Pour comprendre ce phénomène, on peut comparer un plasma à l'équilibre à une boule immobile et la perturbation à une légère impulsion sur cette boule, et donc une légère vitesse initiale. Si la boule est au sommet d'une pente, on va considérer le système

comme instable puisque la vitesse va fortement croître. A l'inverse si la boule est dans un trou, la boule va osciller ou avoir un mouvement amortie si on prend en compte les frottements.

On peut écrire la perturbation d'une quantité ϵ mathématiquement sous la forme suivante :

$$\epsilon = \epsilon_0 + \delta\epsilon \tag{1.56}$$

avec ϵ_0 la quantité à l'équilibre et $\delta\epsilon$ la quantité perturbée que l'on peut approximer dans un premier temps par la perturbation à l'ordre 1 $\delta\epsilon = \epsilon_1$. Cette dernière peut s'écrire sous la forme d'une onde comme :

$$\epsilon_1 = \tilde{\epsilon}_1 e^{i(\mathbf{k}\cdot\mathbf{x}-\omega t)}, \tag{1.57}$$

où $\tilde{\epsilon}_1$ est le coefficient d'amplitude initiale de la perturbation, $\mathbf{k}$ le vecteur d'onde de la perturbation et ω est la pulsation de la perturbation dont on va séparer la forme complexe de la forme réelle $\omega = \omega_R + i\Gamma$.

C'est le paramètre Γ qui va donc décrire l'évolution en amplitude de la quantité ϵ. Si $\Gamma > 0$ alors l'instabilité va croître et, à l'inverse, si $\Gamma < 0$ alors elle va s'amortir. Ce paramètre Γ est appelé taux de croissance lorsqu'il est positif. On comparera souvent les taux de croissance des différentes instabilités afin de savoir laquelle est la plus instable pour un problème donné.

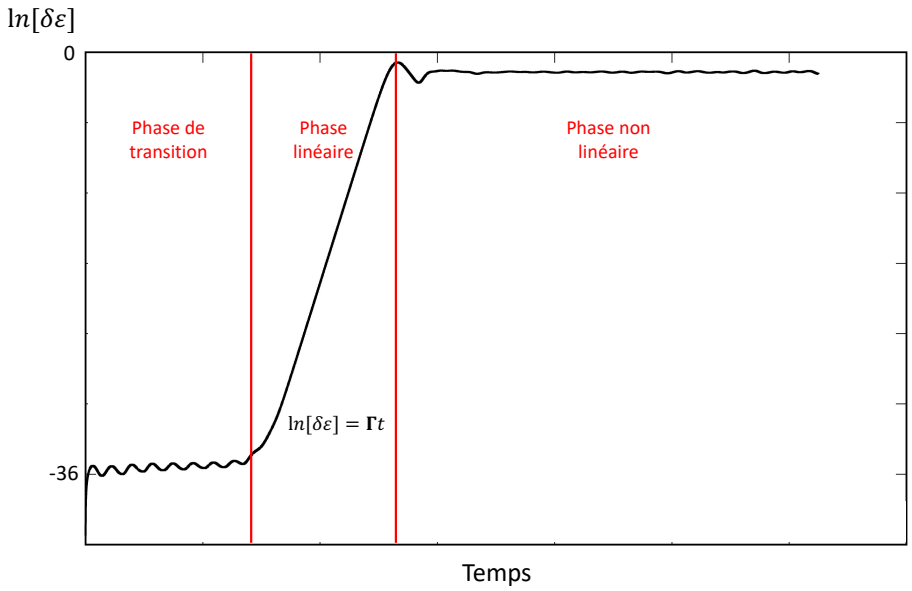

FIGURE 1.5 – Tracé de $\ln(\epsilon_1)$ en fonction du temps afin de montrer les trois phases d'une instabilité plasma. Au début, la phase transitoire ("transient time"). Puis la phase linéaire, où l'on a $\epsilon_1 \propto e^{(\Gamma t)}$. Puis la phase non linéaire où l'on a atteint la saturation de l'instabilité et où d'autres instabilités secondaires peuvent apparaître.

On a représenté sur la figure 1.5 les différentes phases de l'évolution du paramètre $\delta\epsilon$ pour un cas instable.

La première phase est appelée phase de transition ("transient time" en anglais). C'est un artefact numérique dû à une mauvaise initialisation du code. En effet, cette dernière dépend fortement de la cohérence entre les paramètres initiaux. On verra par exemple dans le chapitre 5 que si les champs magnétique et électrique sont excités de manière cohérente, la durée de cette phase est beaucoup plus courte.

La seconde phase est la phase linéaire, ici l'instabilité va croître avec un taux de croissance Γ. En l'absence d'excitation, pour un code sans bruit statistique comme les codes SL, cette croissance va démarrer du bruit numérique de la machine qui est de l'ordre de 10^{-16} pour de la double précision.

À la fin de la phase linéaire, on atteint la saturation de l'instabilité, c'est-à-dire que le paramètre ϵ_1 atteint sa valeur maximale. On rentre ensuite dans le régime non-linéaire, où d'autres instabilités, dites secondaires peuvent faire leur apparition.

Dans la réalité des simulations, le paramètre ϵ représente une énergie et sa croissance est généralement dûe à un échange avec un autre type d'énergie.

1.6.2 Instabilités Weibel

L'instabilité Weibel a pour la première fois était démontrée par E.S. Weibel en 1958 [63]. Cette dernière se produit lorsqu'il existe une anisotropie de température électronique initiale suffisamment importante. C'est-à-dire lorsque $T_\perp > T_{//}$, les symboles $\perp$ et $//$ désignant la direction par rapport au vecteur d'onde $\mathbf{k}$ et T est la température électronique.

Elle va conduire à la création (ou l'amplification) de l'énergie magnétique au dépend de l'énergie cinétique (ou thermique) au travers la création de courants dans la direction perpendiculaire. Ce transfert d'énergie va amener à une régulation de l'anisotropie initiale de température.

Cette instabilité est non-propagative et se développe même en l'absence de champ magnétique extérieur. À l'opposé, en présence d'un champ magnétique parallèle au vecteur d'onde, on peut avoir des instabilités "Whistler" ou modes siffleurs. Cette dernière aide aussi à réguler l'anisotropie initiale et nécessite d'avoir une anisotropie de température plus importante que l'instabilité Weibel [64] ($T_\perp > T_{//}(1+(|\omega_c/\omega_r|\text{-}1)^{\text{-}1})$, avec ω_r la fréquence réelle de l'onde "Whistler").

Encore à ce jour, l'instabilité Weibel est étudiée, principalement pour des cas relativistes où la résolution de la relation de dispersion reste encore complexe [65, 66, 67]. Les instabilités Weibel relativistes étudiées avec plusieurs fonctions de distribution seront le sujet d'étude du chapitre 4.

1.6.3 Instabilités CFI, TSI et Oblique

Afin de représenter par un modèle simplifié l'instabilité Weibel et de déterminer les mécanismes responsable de cette dernière, B.D. Fried propose de décrire cette anisotropie de température par une anisotropie de quantités de mouvements à l'aide de deux faisceaux d'électrons froids se propageant dans des directions opposés [68]. On peut schématiser ceci à l'aide de la figure 1.6, où la distribution initiale Gaussienne utilisée par Weibel [63] est approchée par deux distributions de Dirac à trois dimensions, remplaçant ainsi l'anisotropie de température par une anisotropie de quantité de mouvement (ou de vitesse).

On a donc une distribution froide dans la direction x et z, mais dans la direction y il va exister une "température", avec y la direction perpendiculaire à $\mathbf{k}$ et x et z les deux directions parallèles. Cette température existe puisque les deux faisceaux reproduisent l'écart type de la distribution Gaussienne. En effet, la distribution double faisceau décrit de manière identique le premier (vitesse moyenne) et le second (température) moment fluides de la distribution Gaussienne.

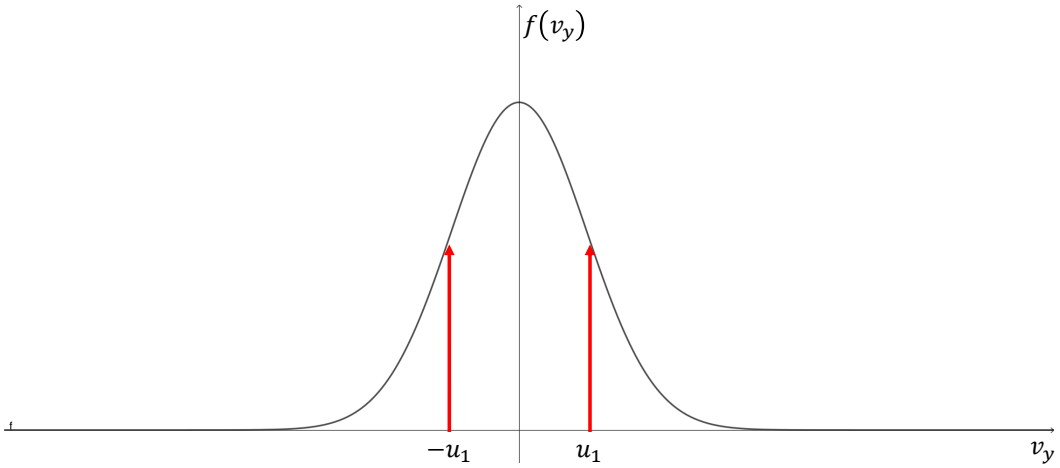

FIGURE 1.6 – Tracé de la fonction de distribution Gaussienne utilisée par Weibel dans [63] (en noir) qui peut être décrite comme deux distributions froides avec une vitesse moyenne (en rouge). Cette représentation a été utilisée par B.D. Fried dans [68] en prenant comme fonction de distribution $f(v_x, v_y, v_z) = \delta(v_x)\delta(v_y^2 - u_1^2)\delta(v_z)$.

On peut justifier ce mécanisme d'augmentation du champ magnétique à l'aide de la figure 1.7. Sur la figure de gauche, deux faisceaux d'électrons se propagent dans la direction y dans des sens opposés. Il y a une perturbation du champ magnétique dans la direction x, donc la direction de $\mathbf{k}$. La force de Lorentz va dévier les électrons créant ainsi des courants qui oscillent de la même manière que le champ magnétique mais qui atteignent leurs maximums lorsque le champ magnétique est nul. De même, il va y avoir une densité d'électrons qui oscille avec une période deux fois plus importante que

le champ magnétique. Ces courants vont ainsi venir renforcer la perturbation initiale du champ magnétique. C'est ainsi que l'énergie magnétique va augmenter et l'énergie cinétique diminuer réduisant ainsi l'anisotropie.

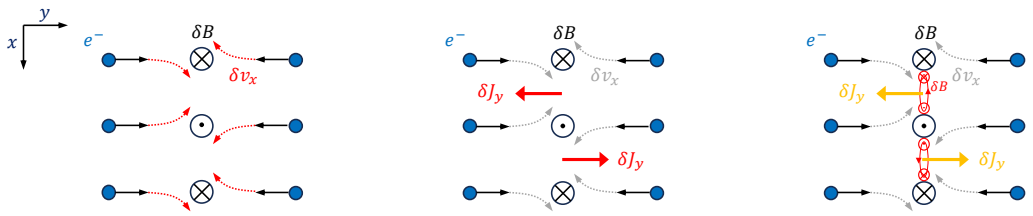

FIGURE 1.7 – Schématisation du processus d'amplification de champ magnétique initial dans une instabilité de type Weibel. Ce schéma a été repris de la thèse de M. Sarrat [69] et de l'article [67]. A gauche, on a représenté les électrons de vitesse opposés qui vont être déviés en raison de la force de Lorentz induite par le champ magnétique oscillant. Cette déviation va entraîner la production de courant qui oscille avec la même longueur d'onde que le champ magnétique mais avec les maximums et minimums des courants correspondant aux valeurs nulles des champs magnétiques et inversement. De la même manière, les régions où la densité électronique est plus importante sont les régions de faible champ magnétique et la densité électronique est plus faible dans les régions à fort champ. Cela va entraîner une oscillation de longueur d'onde deux fois plus importante pour la densité. Les courants créés vont quant à eux produire un champ magnétique qui va venir renforcer le champ magnétique déjà présent et donc amplifier ce processus jusqu'à la saturation.

On appelle aujourd'hui cette instabilité dû à l'anisotropie des quantités de mouvement : instabilité de filamentation de courant (CFI) de l'anglais "Current Filamentation Instability".

Les deux instabilités (CFI et Weibel) ont beaucoup de similitudes. Elles sont toutes deux dûes à une anisotropie de la distribution en vitesses. Elles sont non-propagatives ($\omega_R = 0$), à condition que la CFI ait des faisceaux parfaitement symétriques. Elles vont toutes deux entraîner une réduction de l'anisotropie initiale et la production d'un fort champ magnétique qui piège les électrons.

Il existe cependant quelques différences, les instabilités Weibel sont vraiment produites par une anisotropie de température tandis que l'instabilité CFI est due à une anisotropie des quantités de mouvement entre deux faisceaux d'électrons. Cette différence entraîne la possibilité pour le système de faisceaux d'électrons d'être instable pour des valeurs de $\mathbf{k}$ qui ne sont pas forcément orthogonales.

En effet, lorsque le vecteur $\mathbf{k}$ est parallèle à la direction des faisceaux on va avoir une instabilité dite instabilité double faisceaux (TSI) de l'anglais "two-stream instability" [70, 71]. Cette dernière va entraîner un échange entre l'énergie cinétique et

l'énergie électrostatique, on la qualifie donc d'instabilité électrostatique. Pour une direction arbitraire de **k**, c'est-à dire ni entièrement parallèle ni entièrement perpendiculaire, on aura une instabilité oblique (OI) [29, 72, 73, 74]. L'énergie cinétique va alors être transférée en énergie magnétique et électrostatique puisque cette dernière est un couplage entre la CFI et la TSI.

Sur la figure 1.8, on donne un exemple du taux de croissance des instabilités CFI, TSI et OI pour un cas relativiste qui sera présenté dans le chapitre 4. Dans ce cas à deux dimensions spatiales (x et y), les faisceaux se propagent dans la direction de y, et on observe la CFI pour $\mathbf{k} = (k_x, 0)$, la TSI pour $\mathbf{k} = (0, k_y)$ et l'OI pour $\mathbf{k} = (k_x, k_y)$ avec k_x et k_y non nul.

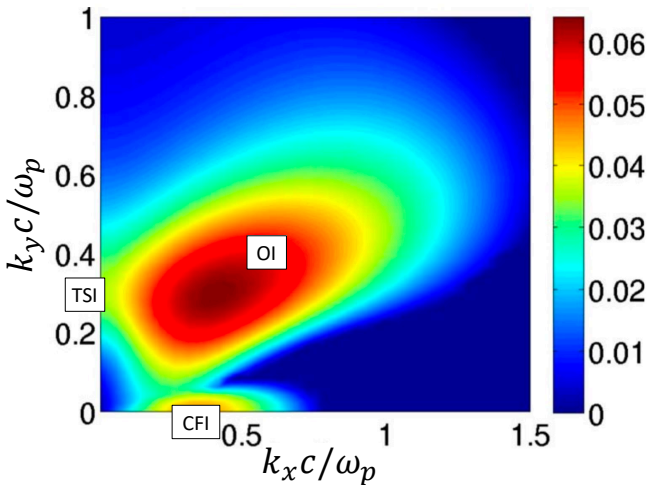

FIGURE 1.8 – Représentation du taux de croissance des instabilités CFI, TSI et OI pour un système de deux faisceaux relativistes se propageant dans des directions opposées. L'image est issue de [66] et les détails de ce cas seront présentés dans le chapitre 4.

Elles sont toutes de nature cinétique et amènent à une modification de la fonction de distribution dans l'espace des phases complet. C'est pourquoi elles nécessitent une description cinétique et sont souvent étudiées par des codes PIC ou SL.

1.6.4 Exemples physiques de ces instabilités

Ces instabilités ne sont pas seulement théoriques, mais elles peuvent être observées dans une variété de phénomènes physiques, tant dans la nature que dans un laboratoire. Nous dressons une liste non exhaustive dans cette sous-partie.

En astrophysique : les sursauts gamma

Les sursauts gamma ou "gamma ray-burst" (GRB) en anglais sont des phénomènes astrophysiques caractérisés par l'apparition d'un rayonnement électromagnétique, principalement composé de rayons gamma, très intenses et brefs (de quelques secondes à quelques minutes) suivi d'une émission rémanente sur tout le spectre électromagnétique pouvant durer plusieurs mois. On les considère comme les événements les plus énergétiques de l'univers même si leur origine n'est pas encore certaine.

En fonction de la durée du premier rayonnement, on distingue les sursauts longs des sursauts courts. Actuellement, on suppose que l'origine des sursauts courts est l'implosion d'une étoile en fin de vie qui donne lieu à une supernova. Tandis que pour les sursauts longs, ce serait plutôt dû à la fusion de deux étoiles à neutrons.

On peut expliquer les sursauts gamma par le modèle de la boule de feu ("fireball"). Des électrons, des photons et des positrons vont être éjectés par "spasmes" à des vitesses ultra-relativistes (pouvant dépasser une énergie de $10^{19}eV$). Il va alors y avoir deux types de phénomènes. Lorsqu'une couche d'électrons (ou coquille) plus rapide va rattraper une couche moins rapide il va y avoir un choc dit interne. On parle de choc externe lorsque les couches d'électrons vont interagir avec le milieu interstellaire environnant que l'on peut considérer comme un plasma froid.

Ces deux types de choc peuvent être décrits comme deux faisceaux plasmas qui ont une quantité de mouvement moyenne différente, c'est exactement le cadre de la CFI, de la TSI et de l'instabilité oblique. Des études numériques et théoriques qui se sont penchées sur le phénomène ont montré que les instabilités de type Weibel était cohérentes en terme d'intensité et de durée et peuvent être à l'origine de ce rayonnement électromagnétique intense [75, 76, 77].

En astrophysique : une variété de phénomène

De manière plus générale, de nombreuses impulsions électromagnétiques provenant de l'univers ont pu être observées. Les raisons de ces rayonnements peuvent être variées mais souvent ils seront dûs à des jets de particules relativistes qui peuvent provenir des trous noirs centraux de certaines galaxies (AGN de l'anglais "Active Galactive Nucleus"), des pulsars, des quasars, des étoiles à neutrons ou, si le jet est non relativiste, du Soleil avec le vent solaire par exemple.

Ces jets de particules vont produire des chocs pas forcément collisionnels en rencontrant d'autres particules souvent beaucoup moins rapides. Les chocs concernent parfois des interactions ions-ions, électrons-électrons, électrons-positrons...

Ces chocs décrivent à nouveau un système physique de deux faisceaux ayant des quantités de mouvement différentes. C'est pourquoi les instabilités de type Weibel sont souvent considérées comme étant le mécanisme principal de génération de ces impulsions électromagnétiques [71, 78].

Sur Terre : la fusion inertielle

L'un des grands axes de recherche actuelle dans le domaine de la physique des plasmas chauds est la fusion nucléaire. Cette dernière a pour objectif de reproduire les mécanismes mis en jeu dans les étoiles afin de produire de l'énergie qui pourrait être utilisée afin de produire de l'électricité.

La fusion nucléaire consiste à fusionner des atomes légers afin de produire des atomes plus lourd avec un gain d'énergie cinétique dû au défaut de masse entre les réactifs et les produits. Cependant, pour pouvoir avoir lieu, il faut que les atomes soit ionisés, donc à l'état plasma, et qu'ils aient suffisamment d'énergie pour pouvoir franchir la barrière de répulsion coulombienne.

Dans les étoiles, cette énergie, ou ce chauffage, est amené par la contraction de l'astre et le confinement de ce plasma est dû à la force de gravitation, c'est un confinement gravitationnel. Sur terre, un tel confinement est impossible, d'autres méthodes sont donc actuellement privilégiées, le confinement magnétique et le confinement inertiel.

La première, le confinement magnétique, se base sur la trajectoire que prennent les particules chargées dans un fort champ magnétique, c'est-à-dire le mouvement cyclotron introduit dans la sous-partie 1.2.3. L'idée actuelle est de piéger le plasma, à l'aide de ces lignes de champs magnétique, dans une cavité torique. Cette géométrie a l'avantage d'éviter des collisions évidentes avec la paroi mais produit un gradient de champ magnétique vers le centre du tore. Ce gradient de champ va amener les particules à être déviées de leurs trajectoires. Il existe deux écoles principales dans le confinement magnétique pour supprimer cette dérive de particules. Le premier, le Tokamak, induit un courant dans le plasma afin d'entraîner une rotation de ce dernier. C'est le cas des réacteurs comme JET [79] ou ITER [8] par exemple. Le second, le Stellarator, courbe directement les lignes de champs magnétiques à l'aide des bobines. On peut citer Asdex-U [80] ou Weindelstein 7-X [81]. Le chauffage des réacteurs par confinement magnétique se fait souvent par injection d'onde dans le plasma et par effet Joule (pour les Tokamaks).

La deuxième méthode de confinement du plasma pour la fusion sur terre est le confinement inertiel, on peut par exemple citer les installations suivantes : le Laser Mega Joule (LMJ) [82] et la National Ignition Facility (NIF) [83]. Une capsule de deutérium-tritium va être comprimée de manière uniforme par des faisceaux laser très puissants permettant ainsi de produire un plasma et de le chauffer. Cette technique n'est cependant pas suffisante pour pouvoir atteindre des énergies importantes au coeur de la capsule [84]. C'est pourquoi, une technique appelée fusion inertielle par allumage rapide a été développée. Cette dernière fonctionne initialement de la même manière avec une compression homogène par faisceaux laser de la capsule. Puis l'idée va être de créer un chemin d'accès pour atteindre le coeur de cette capsule comprimée qui est maintenant entourée de plasma. La première méthode explorée pour faire cela

est appelé "hole boring" [85] et consiste à envoyer une impulsion laser courte ($\sim 100ps$) et ultra-intense ($\sim 10^{19}W/cm^2$). Cette impulsion laser va creuser un trou qui va rapprocher la densité critique, c'est-à-dire celle à partir de laquelle les lasers sont absorbés ou réfléchis, du coeur de la capsule. La deuxième méthode appelée "cone-in-shell" [86] consiste à initialement positionner la capsule sur un cône métallique qui permettra d'atteindre directement le coeur du plasma. Ainsi ces deux méthodes permettent d'accéder directement au coeur de la capsule afin d'envoyer un faisceau laser ultra court ($\sim 10ps$) et intense $\sim 10^{20}W/cm^2$) permettant de chauffer le coeur de la capsule et de réaliser les réactions de fusion Deuterium-Tritium.

Ce dernier faisceau laser va accélérer des électrons vers la cible, laissant ainsi derrière eux un surplus d'ions. L'attraction qu'auront ces ions sur le plasma va produire un courant de retour produisant ainsi un faisceau d'électrons plus froid. Ces deux faisceaux d'électrons vont avoir pour effet de produire des instabilités CFI, TSI et OI [87, 88] dont il est nécessaire de comprendre l'impact afin de réaliser au mieux l'allumage et la fusion de la capsule.

Sur Terre : les accélérateurs par champ de sillage

Un autre secteur de recherche pour l'application sur Terre de la physique des plasmas chauds est l'utilisation de plasma comme accélérateur de particules, appelé accélérateur par champ de sillage. Les accélérateurs de particules actuels utilisés en physique des particules, ou pour l'étude de divers matériaux via les synchrotrons sont souvent considérables en terme de taille et de coût de production. Le plus grand à l'heure actuelle est le Grand collisionneur de hadrons (LHC) [89] avec une taille de 27 km.

L'idée des accélérateurs par champ de sillage va être d'envoyer une onde laser ultra intense dans un plasma sous-dense afin de ioniser les espèces neutres restantes, accélérer les électrons par la force pondéromotrice et ainsi perturber le profil de densité. Cette séparation de charge entre électrons accélérés et ions va produire une onde de Langmuir (ou onde plasma électronique) que l'on appellera onde de sillage puisqu'elle va suivre l'impulsion laser. Cette onde va générer un champ électrique ultra-intense qui va pouvoir accélérer des faisceaux d'électrons à des vitesses ultra-relativistes pour des énergies allant jusqu'au GeV.

Une autre méthode permettant de créer ce champ de sillage est d'envoyer des faisceaux de protons rapides dans le plasma qui auront le même effet que l'onde laser ultra intense précédemment décrite. C'est le cas d'une expérience actuellement en place au CERN : AWAKE [90].

Dans les deux méthodes, le faisceau d'électrons ultra-relativiste va être sujet à des instabilités CFI puisque, de la même manière que dans la fusion inertielle, un courant de retour opposé à ce faisceau d'électrons accéléré va se créer. Ces instabilités CFI vont donc produire des champs magnétiques intenses dans l'une des directions orthogonales à l'accélération, ce qui peut perturber le fonctionnement de l'accélérateur [91, 92].

La filamentation dans les instabilités de type Weibel

Ces instabilités physiques vont être liées à l'émergence d'une filamentation spatiale de la fonction de distribution f, qui a un impact sur la propriété de filamentation cinétique de l'équation de Vlasov. Ainsi, une cascade turbulente peut avoir lieu, qui produit des filaments de plus en plus fins, d'abord dans l'espace des configurations, et finalement dans l'espace des phases. Un tel processus induit une augmentation de la filamentation de f (une sorte de filamentation « augmentée » [48]) qui conduit à de fortes modifications dans le transfert d'énergie et peut conduire à un chauffage stochastique du plasma. L'émergence de ces effets non linéaires introduit des asymétries (élément d'irréversibilité) entre deux aspects fondamentaux de l'équation de Vlasov : la filamentation de la fonction de distribution liée à la réversibilité temporelle de l'équation de Vlasov et le transfert de l'énergie magnétique en énergie cinétique, un processus déjà observé dans la reconnexion magnétique [49, 50].

1.7 Objectifs de la thèse

L'objectif de cette thèse va donc être d'implémenter une méthode de raffinement de maillage uniquement dans l'espace des quantités de mouvement dans le code Semi-Lagrangien VLEM.

Dans un premier temps nous allons présenter le code Semi-Lagrangien VLEM qui servira de point de départ à cette thèse. Nous décrirons donc dans le chapitre 2, le fonctionnement théorique et numérique du code ainsi que les améliorations qui lui ont été apportées au cours de cette thèse.

Dans le chapitre 3 nous présenterons en détail la méthode de raffinement de maillage qui a été développée au cours de cette thèse, puis nous comparerons les résultats avec l'ancienne version afin de valider le nouveau code.

Après avoir montré le bon fonctionnement du code, on l'appliquera à des cas d'instabilités obliques très relativistes ainsi que pour des instabilités Weibel sans excitations initiales avec plusieurs fonctions de distribution dans le chapitre 4.

Enfin, nous terminerons dans le chapitre 5 par présenter une autre méthode de réduction du temps de calcul basée sur une approche de réduction Hamiltonienne qui est la méthode multi-faisceau. Cette dernière a été en partie développée, validée et améliorée au cours de cette thèse.

Chapitre 2

Présentation du code VLEM

Sommaire

2.1 Historique 38
2.2 Méthode Semi-Lagrangienne 39
2.2.1 Normalisation du code 39
2.2.2 Problème du "time splitting" 40
2.2.3 Schéma numérique 42
2.2.4 Parallélisations et architecture de VLEM 44
2.2.5 Rotation dans l'espace des vitesses 45
2.2.6 Advections spatiales locales 48
2.2.7 Évolution des champs électromagnétiques 54
2.2.8 Résumé du schéma numérique global 60
2.3 Modifications apportées au cours de la thèse 61
2.3.1 Optimisation du temps de calcul 61
2.3.2 Implémentation de nouveaux diagnostics 63
2.4 Développement de VLEM3D3V 67
2.4.1 Problème de la mémoire 67
2.4.2 Modification à onze et dix points 67
2.4.3 Modification de la méthode d'advection à six points 68
2.4.4 Méthode de la quasi-interpolation 69
2.4.5 Comparaison entre les méthodes 73

Le code qui va être utilisé au cours de cette thèse est le code Semi-Lagrangien VLEM, qui est un acronyme de l'anglais pour "*Vlasov Electromagnetic solver*". Ce chapitre a pour but d'expliquer son fonctionnement ce qui sera la clé de voûte de la thèse qui consiste à implémenter le raffinement de maillage dans VLEM. La fin de ce chapitre sera, quant à elle, dédiée aux modifications et améliorations qui ont été apportées à ce code durant la thèse.

2.1 Historique

Dans ce sous-chapitre, on va s'intéresser à l'historique des différentes améliorations du code VLEM ainsi que celui des super-calculateurs sur lesquels les calculs étaient effectués.

La première version qui peut être considérée comme l'ancêtre de VLEM a été présentée à la $11^{\text{ème}}$ conférence européenne sur les simulations dans les plasmas en 1986 [93] et le premier article l'utilisant a été publié en 1987 [94]. Le code était alors non relativiste, avec une dimension pour les vitesses et une dimension pour l'espace. Les simulations étaient effectuées sur Cray I, un super-calculateur vectoriel d'une puissance de 160 MFLOPS [95].

Le FLOPS est une unité de mesure qui permet de transcrire la puissance d'un calculateur puisqu'il désigne le nombre d'opérations en virgule flottante que peuvent réaliser l'ensemble des processeurs de la machine. Même si un utilisateur n'a pas accès à tous les processeurs du calculateur en même temps, la comparaison entre différentes puissances permet de comprendre l'évolution des technologies.

En 1990, le code devient relativiste et utilise une méthode multi-faisceaux [96], que l'on développera dans le chapitre 5 de cette thèse. L'équation de Vlasov est donc résolue dans sa version relativiste sur grille 1D1V avec une méthode de réduction Hamiltonienne pour traiter une direction de quantité de mouvement supplémentaire comme une classe de solutions exactes et permettre de faire des simulations 1D2V. A l'époque, le code fonctionnait sur Cray 2, un super-calculateur avec une puissance de $1,9$ GFLOPS [95].

Puis la version de VLEM avec 2 dimensions d'espace et 2 dimensions de quantité de mouvement (2D2V) est décrite en 1999 dans un article [97]. Il introduit aussi la parallélisation MPI (Message Passing Interface) du code, c'est-à-dire qu'il va effectuer les calculs sur plusieurs processeurs qui ne partagent pas la même mémoire. Les simulations sont donc lancées sur un calculateur de plus de 1 TFLOPS permettant ainsi la parallélisation de codes : Cray T3E [98].

Récemment, en 2017, le code a amélioré la parallélisation MPI en introduisant des méthodes locales (voir sous-partie 2.2.6). L'article présentant cette version de VLEM [28] a aussi introduit la version avec 2 dimensions d'espace et 3 dimensions de quantité de mouvement (2D3V). Le code tournait alors sur Turing un super calculateur avec une

puissance de 836 TFLOPS [99].

Le calculateur qui a été utilisé pour tous les calculs avec VLEM dans cette thèse est Jean-Zay qui depuis son amélioration en Juin 2022 atteint une puissance maximale de $36,8$ PFLOPS.

Ce résumé historique permet de comprendre les différentes étapes d'amélioration du code, qui a principalement été développé par Alain Ghizzo, le directeur de cette thèse, et plusieurs collaborateurs. On peut aussi observer que l'amélioration du code nécessite une amélioration technologique qui se traduit par un augmentation de la puissance des calculateurs.

2.2 Méthode Semi-Lagrangienne

Le code VLEM est un code Semi-Lagrangien qui résout l'équation de Vlasov relativiste à plusieurs dimensions pour les électrons, et dont la base provient du code centre-guide Gysela [14]. Les ions seront considérés comme un fond neutralisant, c'est-à-dire que l'on pourra les considérer comme fixes.

2.2.1 Normalisation du code

Le code VLEM va être normalisé, c'est-à-dire que les variables habituellement utilisées en physique vont être modifiées afin de devenir sans dimensions. Cela permet de simplifier les équations mais aussi d'obtenir un résultat général qui pourrait être associé à plusieurs cas particuliers.

Dans VLEM, le choix de la normalisation s'est fait sur la vitesse (v) et le temps (t) respectivement par rapport à la vitesse de la lumière (c) et à l'inverse de la fréquence plasma électronique ($\omega_p = \omega_{p_e}$) qui est un paramètre déjà défini dans la sous-partie 1.1.2 et qui s'écrit :

$$\omega_p = \sqrt{\frac{n_0 e^2}{m_e \epsilon_0}}, \tag{2.1}$$

avec e et m_e la charge et la masse de l'électron et ϵ_0 la permittivité du vide. n_0 est la densité des ions qui est supposé constante et qui initialement est égale à la densité électronique moyenne n.

De ce choix de normalisation vont découler celles de toutes les autres variables qui sont résumées dans le tableau 2.1 en écrivant $\hat{A}$ la nouvelle variable normalisée de la variable A. Dans la suite de ce manuscrit, sauf indication contraire, les variables qui seront utilisées vont être normalisées sans utiliser l'indication $\hat{}$.

Il faut aussi noter que la fonction de distribution est normalisée à $(mc)^{d_p}/n_0$ et va dépendre de la dimension de l'espace des quantités de mouvement (d_p). Dans le cas 3V, la fonction de distribution est normalisée à m^3c^3/n_0 ce qui permet la description d'une

$\hat{t}$	$t\omega_p$
$\hat{\boldsymbol{x}}$	$\boldsymbol{x}\omega_p/\mathrm{c}$
$\hat{\boldsymbol{p}}$	$\boldsymbol{p}/(m\mathrm{c})$
$\hat{\boldsymbol{E}}$	$\boldsymbol{E}e/(m\omega_p\mathrm{c})$
$\hat{\boldsymbol{B}}$	$\boldsymbol{B}e/(m\omega_p)$
$\hat{n}$	n/n_0

TABLE 2.1 – Normalisation des variables dans VLEM.

fonction de distribution du type Maxwell-Jüttner dont la forme générale 3V s'écrit :

$$f(\mathbf{p}) = \frac{n_0}{4\,\pi\,m^3\,c^3\,\theta\,K_2(\frac{1}{\theta})}\exp\left(-\frac{\gamma(\mathbf{p})}{\theta}\right), \tag{2.2}$$

avec $\gamma(\mathbf{p})$ le facteur de Lorentz, K_2 la fonction de Bessel modifiée de deuxième type et $\theta = k_B T/(mc^2)$.

On peut donc, en se servant des normalisations, réécrire l'équation de Vlasov relativiste comme :

$$\frac{\partial f}{\partial t} + \frac{\boldsymbol{p}}{\gamma}\cdot\boldsymbol{\nabla}_{\boldsymbol{x}} f + \left(\boldsymbol{E} + \frac{\boldsymbol{p}\times\boldsymbol{B}}{\gamma}\right)\cdot\boldsymbol{\nabla}_{\boldsymbol{p}} f = 0, \tag{2.3}$$

avec $\gamma = \sqrt{1+\boldsymbol{p}\cdot\boldsymbol{p}}$, le facteur relativiste normalisé.

2.2.2 Problème du "time splitting"

Afin de résoudre numériquement l'équation de Vlasov, on choisit de la découper en plusieurs équations plus simples, d'où le terme "splitting". Pour l'équation de Vlasov dans sa version non-relativiste, on va pouvoir découper l'équation de Vlasov en autant d'équations qu'il y a de variables. Cependant, on va voir dans cette sous-partie que ceci n'est plus vrai lorsque l'on passe en relativiste dû à l'interdépendance des variables de quantité de mouvement au travers de γ. Le problème a été indiqué dans [100] et on va ici le ré-expliquer pour un cas avec deux dimensions d'espace et deux dimensions de quantité de mouvement.

On rappelle que la résolution de l'équation de Vlasov est couplée aux champs électromagnétiques dont la résolution va se faire dans VLEM par la résolution des équations de Maxwell (voir sous-partie 2.2.7).

On commence d'abord par réécrire l'équation de Vlasov à deux dimensions d'espace et deux dimensions de quantité de mouvement comme :

$$\frac{\partial f}{\partial t} + \frac{p_x}{\gamma}\frac{\partial f}{\partial x} + \frac{p_y}{\gamma}\frac{\partial f}{\partial y} + \left(E_x + \frac{p_y}{\gamma}B_z\right)\frac{\partial f}{\partial p_x} + \left(E_y - \frac{p_x}{\gamma}B_z\right)\frac{\partial f}{\partial p_y} = 0, \tag{2.4}$$

$$\frac{\partial f}{\partial t} + \mathbf{U}(\mathbf{X}, t) \cdot \boldsymbol{\nabla}_{\boldsymbol{X}} f = 0, \tag{2.5}$$

avec $\mathbf{X} = (x, y, p_x, p_y)$, le vecteur contenant les variables, et

$$\mathbf{U} = \begin{pmatrix} \frac{p_x}{\gamma} \\ \frac{p_y}{\gamma} \\ E_x + \frac{p_y}{\gamma} B_z \\ E_y - \frac{p_x}{\gamma} B_z \end{pmatrix} = \begin{pmatrix} U_{11} \\ U_{12} \\ U_{21} \\ U_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{U_1} \\ \mathbf{U_2} \end{pmatrix}, \tag{2.6}$$

avec $\mathbf{U}_\alpha = \begin{pmatrix} U_{\alpha 1} \\ U_{\alpha 2} \end{pmatrix}$.

Sachant que les champs E_x, E_y et B_z ne dépendent que des variables d'espace (x,y) et que $\gamma = \sqrt{1 + p_x^2 + p_y^2}$, on peut en déduire que $\mathbf{U}$ est à divergence nulle. On peut donc réécrire 2.5 dans une forme conservative comme :

$$\begin{aligned} \frac{\partial f}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{X}} \cdot (\mathbf{U}(\mathbf{X}, t) f) = \frac{\partial f}{\partial t} &+ \boldsymbol{\nabla}_{\boldsymbol{X_1}} \cdot (\mathbf{U_1}(\mathbf{X_1}, \mathbf{X_2}, t) f) \\ &+ \boldsymbol{\nabla}_{\boldsymbol{X_2}} \cdot (\mathbf{U_2}(\mathbf{X_1}, \mathbf{X_2}, t) f) = 0, \end{aligned} \tag{2.7}$$

avec $\mathbf{X_1} = \begin{pmatrix} X_{11} \\ X_{12} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix}$ et $\mathbf{X_2} = \begin{pmatrix} X_{21} \\ X_{22} \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \end{pmatrix}$.

On peut résoudre cette équation en conservant la précision au second ordre [101] si on résout alternativement les deux équations suivantes :

$$\frac{\partial f}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{X_1}} \cdot (\mathbf{U_1}(\mathbf{X}, t) f) = 0, \tag{2.8}$$

$$\frac{\partial f}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{X_2}} \cdot (\mathbf{U_2}(\mathbf{X}, t) f) = 0, \tag{2.9}$$

c'est le fameux "splitting".

Comme on a $\boldsymbol{\nabla}_{\boldsymbol{X_1}} \cdot \mathbf{U_1}(\mathbf{X}, \mathbf{X}, t) = 0$ et $\boldsymbol{\nabla}_{\boldsymbol{X_2}} \cdot \mathbf{U_2}(\mathbf{X}, \mathbf{X}, t) = 0$, on peut réécrire ces deux équations sous forme advective comme :

$$\frac{\partial f}{\partial t} + \mathbf{U_1}(\mathbf{X}, t) \cdot \boldsymbol{\nabla}_{\boldsymbol{X_1}} f = 0, \tag{2.10}$$

$$\frac{\partial f}{\partial t} + \mathbf{U_2}(\mathbf{X}, t) \cdot \boldsymbol{\nabla}_{\boldsymbol{X_2}} f = 0. \tag{2.11}$$

Si on recommence le "splitting" sur Eq.(2.10), comme $\nabla_{X_{11}} U_{11} = \partial_x (p_x/\gamma) = 0$ et $\nabla_{X_{12}} U_{12} = \partial_y (p_y/\gamma) = 0$, on peut résoudre l'équation (2.10) en alternant la résolution des deux équations :

$$\frac{\partial f}{\partial t} + U_{11}(\mathbf{X}, t) \frac{\partial f}{\partial X_{11}} = 0, \tag{2.12}$$

$$\frac{\partial f}{\partial t} + U_{12}(\mathbf{X}, t)\frac{\partial f}{\partial X_{12}} = 0, \tag{2.13}$$

que l'on peut réécrire en développant les variables comme :

$$\frac{\partial f}{\partial t} + \frac{p_x}{\gamma}\frac{\partial f}{\partial x} = 0, \tag{2.14}$$

$$\frac{\partial f}{\partial t} + \frac{p_y}{\gamma}\frac{\partial f}{\partial y} = 0. \tag{2.15}$$

Si on essaye maintenant de faire la même chose avec l'équation (2.11), on remarque qu'il n'est plus possible de l'écrire sous forme advective. En effet, dû à la dépendance du facteur relativiste $\gamma(p_x, p_y)$, on a :

$$\nabla_{X_{21}}.U_{21} = \frac{\partial}{\partial p_x}\left(E_x + \frac{p_y}{\gamma}B_z\right) = -\frac{p_y p_x}{\gamma^3}B_z, \tag{2.16}$$

$$\nabla_{X_{22}}.U_{22} = \frac{\partial}{\partial p_y}\left(E_y - \frac{p_x}{\gamma}B_z\right) = \frac{p_y p_x}{\gamma^3}B_z. \tag{2.17}$$

On ne peut donc pas séparer l'équation d'advection (2.11) en une succession d'équations d'advections à une seule variable. Ce problème étant dû au facteur γ, il existe uniquement pour la résolution de l'équation de Vlasov relativiste. En effet pour un code non-relativiste, l'équation (2.11) est séparable en deux autres équations d'advection à une dimension.

On peut redémontrer facilement que le problème existe de la même manière à trois dimensions dans l'espace et/ou dans les vitesses. A chaque fois, les advections des quantités de mouvement devront être effectuées ensemble.

2.2.3 Schéma numérique

En reprenant les résultats de la sous-partie précédente, on voit qu'afin de résoudre numériquement l'équation de Vlasov, il faut résoudre en alternant les équations (2.14), (2.15) et (2.11). Afin d'avoir un schéma numérique à l'ordre 2 en temps (voir annexe A), on va effectuer des séquences successives de décalages et de rotation sous la forme $\left(\frac{\mathcal{Y}}{2}\frac{\mathcal{X}}{2}\mathcal{R}\frac{\mathcal{X}}{2}\frac{\mathcal{Y}}{2}\right)^n$ ce qui est équivalent à :

$$\frac{\mathcal{Y}}{2}\frac{\mathcal{X}}{2}\left(\mathcal{R}\frac{\mathcal{X}}{2}\mathcal{Y}\frac{\mathcal{X}}{2}\right)^{n-1}\mathcal{R}\frac{\mathcal{X}}{2}\frac{\mathcal{Y}}{2}.$$

Où $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{R}$ correspondent respectivement à la résolution sur un pas de temps Δt des équations (2.14), (2.15) et (2.11).

En ignorant les extrémités de la séquence, on peut résumer les étapes principales de résolution, non normalisée, comme :

$$\text{La rotation } \mathcal{R} : \quad f^*(x,y,\boldsymbol{p}) = f^n\left(x,y,\boldsymbol{P}(x,y,\boldsymbol{p})\right), \tag{2.18}$$

$$\text{Le décalage } \mathcal{X}/2 : \quad f^{**}(x,y,\boldsymbol{p}) = f^*\left(x - \frac{p_x}{m\gamma}\frac{\Delta t}{2}, y, \boldsymbol{p}\right), \tag{2.19}$$

$$\text{Le décalage } \mathcal{Y} : \quad f^{***}(x,y,\boldsymbol{p}) = f^{**}\left(x, y - \frac{p_y}{m\gamma}\Delta t, \boldsymbol{p}\right), \tag{2.20}$$

$$\text{Le décalage } \mathcal{X}/2 : \quad f^{n+1}(x,y,\boldsymbol{p}) = f^{***}\left(x - \frac{p_x}{m\gamma}\frac{\Delta t}{2}, y, \boldsymbol{p}\right), \tag{2.21}$$

avec f^n la fonction de distribution à la $n^{\text{ème}}$ itération temporelle. On réécrit donc le schéma en temps comme $((\mathcal{R})\,(\mathcal{X}/2)\,(\mathcal{Y})\,(\mathcal{X}/2))$.

Il est possible de découpler les champs électrique et magnétique dans l'équation (2.11), on se retrouve avec deux équations d'advection non normalisées sous la forme $\frac{\partial f}{\partial t} + e\boldsymbol{E}\cdot\boldsymbol{\nabla}_{\boldsymbol{p}} f = 0$ et $\frac{\partial f}{\partial t} + e\frac{\boldsymbol{p}\times\boldsymbol{B}}{m\gamma}\cdot\boldsymbol{\nabla}_{\boldsymbol{p}} f = 0$. Cependant il n'est toujours pas possible de découpler les différentes composantes des quantités de mouvement dans l'équation des champs magnétiques puisqu'elle reste couplée par le terme γ. On peut donc développer la rotation effectuée à l'équation (2.18) dans le cas 3V général comme :

$$P_x = -e\frac{\Delta t E_x}{2} + p_x^\delta R_{xx} + p_y^\delta R_{xy} + p_z^\delta R_{xz}, \tag{2.22}$$

$$P_y = -e\frac{\Delta t E_y}{2} + p_x^\delta R_{yx} + p_y^\delta R_{yy} + p_z^\delta R_{yz}, \tag{2.23}$$

$$P_z = -e\frac{\Delta t E_z}{2} + p_x^\delta R_{zx} + p_y^\delta R_{zy} + p_z^\delta R_{zz}, \tag{2.24}$$

avec

$$\boldsymbol{p}^{\boldsymbol{\delta}} = \boldsymbol{p} - e\frac{\Delta t \boldsymbol{E}}{2}, \tag{2.25}$$

et avec R la matrice de rotation [102] :

$$R = \begin{pmatrix} b_x^2+(1-b_x^2)\cos\phi & b_x b_y(1-\cos\phi)-b_z\sin\phi & b_x b_z(1-\cos\phi)+b_y\sin\phi \\ b_x b_y(1-\cos\phi)+b_z\sin\phi & b_y^2+(1-b_y^2)\cos\phi & b_y b_z(1-\cos\phi)-b_x\sin\phi \\ b_x b_z(1-\cos\phi)-b_y\sin\phi & b_y b_z(1-\cos\phi)+b_x\sin\phi & b_z^2+(1-b_z^2)\cos\phi \end{pmatrix}, \tag{2.26}$$

avec $b_\alpha = B_\alpha/|B|$ et $\phi = \omega_c \Delta t$, en prenant $\omega_c = e|B|/(m\gamma)$ la fréquence cyclotron. Cette matrice est celle utilisée pour le code VLEM2D3V, mais pour VLEM2D2V, on va utiliser une version simplifiée où $b_x = b_y = 0$ et $b_z = 1$. En effet, on verra dans la sous-partie 2.2.7 que lorsqu'on a 2 dimensions de vitesses (p_x, p_y) on ne peut pas avoir de courants J_z et donc on impose $B_x = B_y = E_z = 0$.

Ici P_α est la solution obtenue à partir des caractéristiques de l'équation de Vlasov dans l'espace des quantités de mouvement, ce n'est pas le moment canonique.

Pour résoudre numériquement les équations (2.18-2.21) on va avoir besoin de méthodes d'interpolation 1D et 3D (ou 2D en fonction des dimensions du système) qui seront développées dans les sous-parties 2.2.5 et 2.2.6 suivantes.

2.2.4 Parallélisations et architecture de VLEM

Le code VLEM est donc un code Semi-Lagrangien qui va approcher la fonction de distribution pour un nombre de dimensions donné avec des conditions aux bords périodiques en espace et avec des conditions aux bords nulles pour les quantités de mouvement. Cette fonction de distribution va être représentée numériquement comme un tenseur d'ordre $d_x \times d_p$ avec d_x le nombre de dimensions d'espace et d_p le nombre de dimensions de quantité de mouvement. Il existe donc plusieurs versions de VLEM en fonction de la dimension des problèmes que l'on souhaite étudier. Celles qui seront utilisées au cours de la thèse seront le 2D2V ($d_x = 2, d_p = 2$ pour un tenseur d'ordre 4), le $2D3V$ ($d_x = 2, d_p = 3$ pour un tenseur d'ordre 5) et on introduira le 3D3V ($d_x = 3, d_p = 3$ pour un tenseur d'ordre 6).

On peut donc écrire la fonction de distribution, qui est définie sur des points de grilles telle que à 6 dimensions on ait $f(x, y, z, p_x, p_y, p_z) = f(i_x, i_y, i_z, j_x, j_z)$ avec les indices i_α numérotant les points spatiaux et les indices j_α les points des quantités de mouvement. Les valeurs des indices i_α vont varier de 0 à $N_\alpha - 1$ et les indices j_α de 0 à $N_{p_\alpha} - 1$, avec N_α et N_{p_α} respectivement le nombre de points spatiaux et des quantités de mouvement dans la direction α.

Comme annoncé dans la partie historique, le code VLEM est parallélisé MPI. C'est-à-dire que la résolution du code va être effectuée sur différents processus (groupements de processeurs) qui ne partagent pas la même mémoire. Pour se faire, un découpage spatial du domaine de simulation a été fait. Ce découpage va se faire sur toutes les variables qui évoluent spatialement dans le code et donc aussi sur la fonction de distribution (on verra par la suite que c'est aussi le cas pour les champs électromagnétiques et les courants).

Ce découpage spatial va donc séparer le domaine entier formé par les i_α en sous domaines de taille 16^{d_x} chacun indépendants en terme de mémoire. En 3D, comme on a 3 indices pour les 3 composantes spatiales, on choisit de découper le cube de dimensions $N_x \times N_y \times N_z$ en cubes de dimensions $16 \times 16 \times 16$ (en 2D ce sont des carrés de dimensions 16×16). Le positionnement des cubes est fait de manière simple et ordonnée afin que chaque cube/processus ait 6 voisins, soit un par face. Pour le cas 2D, les carrés ont donc 4 voisins comme montré sur la figure 2.1.

Le choix de 16 dans les dimensions du cube est fait de manière à être plus facilement parallélisable en OpenMP par des multiples de 4. En effet, la parallélisation OpenMP permet, quant à elle, de séparer le travail au sein d'un cube/processus entre différents

processeurs. Ces derniers vont, quant à eux, partager leur mémoire. Une autre raison pour ce choix de 16 sera expliquée dans la sous-partie 2.2.6.

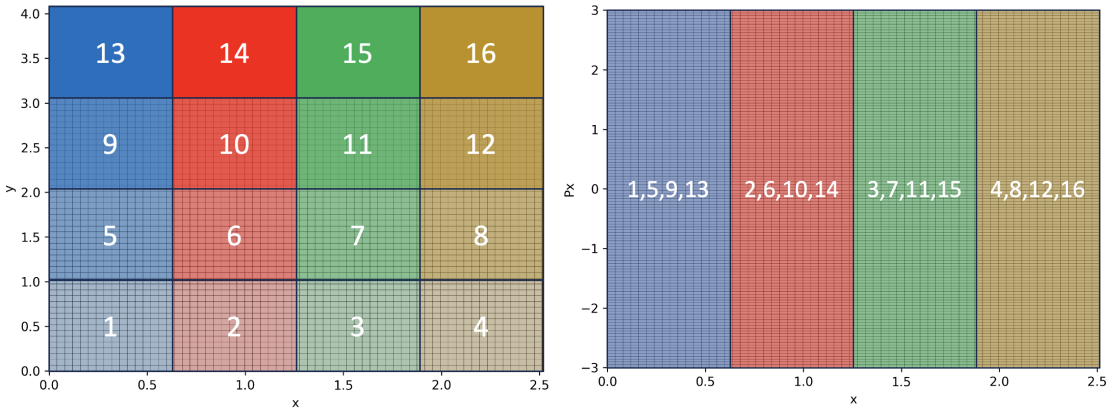

FIGURE 2.1 – Représentation de la décomposition MPI spatiale du code VLEM. Les processus sont numérotés de 1 à 16. A gauche la fonction de distribution est représentée dans l'espace (x, y) et le découpage entre les processus est fait sur des carrés de dimensions 16×16 points. A droite, la fonction de distribution est représentée sur l'espace (x, p_x) et la décomposition MPI se fait uniquement spatialement.

Par la suite, on utilisera les termes voisins de "gauche" et "droite" pour parler des processus voisins suivant x, "bas" et "haut pour la direction y et "derrière" et "devant" pour la direction z. Sachant que le code VLEM est périodique, on a par exemple pour le processus 1 (en s'appuyant sur la figure 2.1), les processus 2 et 4 comme voisins respectivement de "droite" et de "gauche" et les processus 5 et 13 comme voisins du "haut" et du "bas". On utilisera aussi $i_{\alpha_{start}}$ et $i_{\alpha_{end}}$, ou plus simplement i_{start} et i_{end}, qui désignent respectivement le premier et le dernier point spatiaux, dans la direction α, d'un sous domaine.

2.2.5 Rotation dans l'espace des vitesses

La méthode d'interpolation utilisée pour effectuer la rotation (Eq.(2.18)) est basée sur la fonction B-spline cubique en 3 ou 2 dimensions en fonction du nombre de composantes des quantités de mouvement (d_p). Cette méthode a été décrite pour le cas 2V dans l'article de Sonnendrücker et al. [30]. L'explication que je vais donner dans cette partie est une extension au cas 3V avec une clarification de certains points.

À une position donnée dans l'espace $(\mathbf{x})$ la fonction de distribution $f : \mathbb{R}^3 \longrightarrow \mathbb{R}_+, \mathbf{p} \mapsto f(\mathbf{x}, \mathbf{p})$ est interpolée par une fonction spline cubique s qui peut s'écrire comme :

$$s(p_x, p_y, p_z) = \sum_{-1 \leq \kappa \leq N_{p_x}} \sum_{-1 \leq \mu \leq N_{p_y}} \sum_{-1 \leq \nu \leq N_{p_z}} \eta_{\kappa,\mu,\nu} B_\kappa(p_x) B_\mu(p_y) B_\nu(p_z), \tag{2.27}$$

avec N_{p_α} le nombre de points dans la direction p_α et avec $B_\kappa(p_x)$ la fonction B-spline cubique pour des points espacés régulièrement [103] telle que :

$$B_\kappa(p_x) = \frac{1}{6h^3} \begin{cases} (p_x - p_{x_{\kappa-2}})^3 & p_{x_{\kappa-2}} \leq p_x < p_{x_{\kappa-1}} \\ h^3 + 3h^2(p_x - p_{x_{\kappa-1}}) + 3h(p_x - p_{x_{\kappa-1}})^2 - 3(p_x - p_{x_{\kappa-1}})^3 & p_{x_{\kappa-1}} \leq p_x < p_{x_\kappa} \\ h^3 + 3h^2(p_{x_{\kappa+1}} - p_x) + 3h(p_{x_{\kappa+1}} - p_x)^2 - 3(p_{x_{\kappa+1}} - p_x)^3 & p_{x_\kappa} \leq p_x < p_{x_{\kappa+1}} \\ (p_{x_{\kappa+2}} - p_x)^3 & p_{x_{\kappa+1}} \leq p_x < p_{x_{\kappa+2}} \\ 0 & \text{sinon} \end{cases} \tag{2.28}$$

On a tracé graphiquement cette fonction B-spline cubique $B\kappa(p_x)$ sur la figure 2.2.

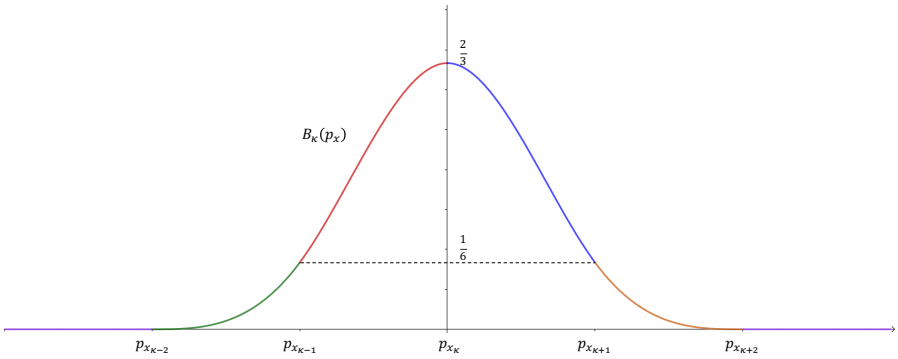

FIGURE 2.2 – Tracé de la fonction B-spline cubique $B_\kappa(p_x)$. Les différents intervalles sont représentés en couleur. Les valeurs de $B_\kappa(p_\kappa) = \frac{2}{3}$ et $B_\kappa(p_{\kappa\pm1}) = \frac{1}{6}$ sont représentées.

Cette écriture (Eq.(2.28)) et ce tracé graphique (Fig.2.2) sont aussi valides pour les couples de paramètres μ/p_y et ν/p_z avec $h = \Delta p_\alpha = \frac{p_{\alpha_{max}} - p_{\alpha_{min}}}{N_{p\alpha} - 1}$ correspondant au pas des quantités de mouvement.

On peut réécrire Eq.(2.27) comme :

$$s(p_x, p_{y_j}, p_{z_k}) = \sum_{-1 \leq \kappa \leq N_{px}} \gamma_\kappa(p_{y_j}, p_{z_k}) B_\kappa(p_x), \tag{2.29}$$

pour $j = 0, ..., N_{p_y} - 1$ et $k = 0, ..., N_{p_z} - 1$, avec

$$\gamma_\kappa(p_y, p_{z_k}) = \sum_{-1 \leq \mu \leq N_{py}} \gamma^*_{\kappa,\mu}(p_{z_k}) B_\mu(p_y), \tag{2.30}$$

et

$$\gamma^*_{\kappa,\mu}(p_z) = \sum_{-1 \leq \nu \leq N_{pz}} \eta_{\kappa,\mu,\nu} B_\nu(p_z). \tag{2.31}$$

Dans un premier temps, on va résoudre Eq.(2.29). On a donc $(N_{p_y} * N_{p_z})$ systèmes composés chacun de $N_{p_x} + 2$ inconnues. Pour un système situé en j et k fixés, on doit

imposer $N_{p_x}+2$ conditions. La première est que la fonction interpolée s doit valoir la valeur de la fonction f aux points, tel que :

$$f(p_{x_i},p_{y_j},p_{z_k}) = s(p_{x_i},p_{y_j},p_{z_k}). \tag{2.32}$$

En développant la valeur de la fonction interpolée s à l'aide de l'équation (2.29) et de la fonction B-spline (Eq.(2.28)) on peut réécrire ces conditions comme :

$$f(p_{x_i},p_{y_j},p_{z_k}) = \frac{1}{6}\gamma_{i-1}(p_{y_j},p_{z_k}) + \frac{4}{6}\gamma_i(p_{y_j},p_{z_k}) + \frac{1}{6}\gamma_{i+1}(p_{y_j},p_{z_k}). \tag{2.33}$$

Il reste 2 conditions à imposer pour pouvoir résoudre le système. On choisit 2 conditions aux bords sur la dérivée seconde telles que :

$$\begin{aligned}\partial^2_{p_x} f(p_{x_i},p_{y_j},p_{z_k}) = \partial^2_{p_x} s(p_{x_i},p_{y_j},p_{z_k}) &= \sum_{-1\leq\kappa\leq N_{p_x}} \gamma_\kappa(p_{y_j},p_{z_k})\partial^2_{p_x}B_\kappa(p_{x_i}) \\ &= \frac{1}{h^2}\gamma_{i-1}(p_{y_j},p_{z_k}) - \frac{2}{h^2}\gamma_i(p_{y_j},p_{z_k}) + \frac{1}{h^2}\gamma_{i+1}(p_{y_j}.p_{z_k}),\end{aligned} \tag{2.34}$$

avec $i=0$ ou $i=N_{p_x}-1$.

On peut réécrire le système sous forme matricielle tel que :

$$\boldsymbol{M}\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} = \begin{pmatrix}\mathbf{b}\\ \mathbf{c}\end{pmatrix}. \tag{2.35}$$

Où dans un premier temps $\mathbf{x}$ et $\mathbf{y}$ correspondent respectivement aux vecteurs $(\gamma_{N_{p_x}},\gamma_{-1})^T$ et $(\gamma_0,\gamma_1,...,\gamma_{N_{p_x}-1})^T$. Et où $\mathbf{b}$ et $\mathbf{c}$ correspondent à $(\partial^2_{p_x}f(p_{x_{N_{p_x}\text{-}1}},p_{y_j},p_{z_k})$,$\partial^2_{p_x}f(p_{x_0},p_{y_j},p_{z_k}))^T$ et $(f(p_{x_0},p_{y_j},p_{z_k}),f(p_{x_1},p_{y_j},p_{z_k}),...,f(p_{x_{N_{p_x}\text{-}1}},p_{y_j},p_{z_k}))^T$. Avec $(\ )^T$ la transposée du vecteur.

On a donc M la matrice quasi-tridiagonale de dimension $(N_{p_x}+2)^2$:

$$\mathbf{M} = \frac{1}{6}\begin{pmatrix} \frac{6}{h^2} & 0 & \cdots & \cdots & 0 & \frac{6}{h^2} & \frac{-12}{h^2} \\ 0 & \frac{6}{h^2} & \frac{-12}{h^2} & \frac{6}{h^2} & 0 & \cdots & 0 \\ \vdots & 1 & 4 & 1 & 0 & \cdots & \vdots \\ \vdots & 0 & 1 & 4 & 1 & 0 & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \vdots & \cdots & \ddots & 1 & 4 & 1 \\ 1 & 0 & \cdots & \cdots & 0 & 1 & 4 \end{pmatrix}, \tag{2.36}$$

que l'on va séparer en blocs tels que :

$$\mathbf{M} = \left(\begin{array}{c|c} \boldsymbol{\delta} & \boldsymbol{\zeta} \\ \hline \boldsymbol{\Gamma} & \mathbf{A} \end{array}\right), \tag{2.37}$$

où $\boldsymbol{A}$ est une matrice tridiagonale avec 1/6 dans la sous-diagonale et 4/6 dans la diagonale et $\boldsymbol{\delta}$ la matrice diagonale (2×2) avec comme coefficient $1/h^2$.

Dans VLEM, la dérivée seconde aux bords de la boîte est prise comme nulle, telle que $\mathbf{b} = (0,0)^T$. C'est la condition aux bords naturelle des splines [103].

Résoudre l'équation (2.35) amène à 2 équations pour $\mathbf{x}$ et $\mathbf{y}$:

$$(\boldsymbol{\delta} - \boldsymbol{\zeta}\boldsymbol{A}^{-1}\boldsymbol{\Gamma})\mathbf{x} = \mathbf{b} - \boldsymbol{\zeta}\boldsymbol{A}^{-1}\mathbf{c}, \tag{2.38}$$

et

$$\mathbf{y} = \boldsymbol{A}^{-1}\mathbf{c} - \boldsymbol{A}^{-1}\boldsymbol{\Gamma}\mathbf{x}. \tag{2.39}$$

Pour résoudre ces équations (Eqs.(2.38) et (2.39)) on procède aux étapes suivantes [30] :

- Calcul de $\boldsymbol{A}^{-1}\boldsymbol{\Gamma}$.
- Calcul du terme de gauche de l'équation Eq.(2.38) $(\boldsymbol{\delta} - \boldsymbol{\zeta}\boldsymbol{A}^{-1}\boldsymbol{\Gamma})$.
- Calcul de $\boldsymbol{A}^{-1}\mathbf{c}$ avec $\mathbf{c}$ la fonction de distribution $f(p_x, p_{y_j}, p_{z_k})$.
- Calcul de $\mathbf{x} = (\boldsymbol{\delta} - \boldsymbol{\zeta}\boldsymbol{A}^{-1}\boldsymbol{\Gamma})^{-1}(\mathbf{b} - \boldsymbol{\zeta}\boldsymbol{A}^{-1}\mathbf{c})$.
- Résolution de l'équation (2.39) pour trouver $\mathbf{y}$.

On a désormais les valeurs de $\gamma_\kappa(p_{y_j}, p_{z_k})$ et on va résoudre de la même manière Eq.(2.30) mais avec $\mathbf{x}$ et $\mathbf{y}$ correspondant ici respectivement aux vecteurs $(\gamma^*_{\kappa,N_{p_y}}, \gamma^*_{\kappa,-1})^T$ et $(\gamma^*_{\kappa,0}, \gamma^*_{\kappa,1}, \ ..., \gamma^*_{\kappa,N_{p_y}-1})^T$. Et où $\mathbf{b}$ et $\mathbf{c}$ correspondent à $\Big(\partial^2_{p_y}\gamma_\kappa(p_{y_{N_{p_y}-1}}, p_{z_k})$, $\partial^2_{p_y}\gamma_\kappa(p_{y_0}, p_{z_k})\Big)^T = (0,0)^T$ et $(\gamma_\kappa(p_{y_0}, p_{z_k}), \gamma_\kappa(p_{y_1}, p_{z_k}), ..., \gamma_\kappa(p_{y_{N_{p_y}-1}}, p_{z_k}))^T$. On obtient ainsi les valeurs des $\gamma^*_{\kappa,\mu}(p_{z_k})$ et on peut reprendre les mêmes étapes pour trouver les coefficients de splines $\eta_{\kappa,\mu,\nu}$.

On peut donc désormais résoudre la rotation (Eq.(2.18)) pour chaque point de l'espace (x, y) en posant pour le cas particulier du code 2D $f^n(x, y, \boldsymbol{P}(x, y, \boldsymbol{p})) = s^n(x, y, \boldsymbol{P}(x, y, \boldsymbol{p}))$.

2.2.6 Advections spatiales locales

Cette sous-partie est expliquée dans l'article [104] mais elle va être développée ici puisqu'une étude sur la modification de cette méthode va être effectuée par la suite dans la sous-partie 2.4.3.

Comme vu dans la sous-partie 2.2.3, les advections spatiales peuvent être résolues séparément. La méthode d'interpolation utilisée sera la même pour chaque direction et ce seront, comme dans la partie précédente des B-splines cubiques mais à une dimension.

Cependant, la décomposition de domaine spatiale due à la parallélisation MPI que nous avons abordée dans la sous-partie 2.2.4 va nécessiter d'utiliser des splines locaux. C'est-à-dire que plutôt que d'être calculés afin d'interpoler une ligne complète allant

de 0 à N_α-1 en prenant en compte les conditions périodiques, ils seront calculés uniquement sur les 16 points spatiaux que contient le processus ainsi que le dernier point du processus voisin "gauche" et les deux premiers points du processus voisin "droite" (c.à.d. entre $i_{start}-1$ et $i_{end}+2$). Cela va permettre de limiter les communications aux 2 voisins proches de la direction de l'advection plutôt qu'entre tous les processus d'une même ligne.

La dissymétrie entre gauche et droite a été constatée au cours de la thèse. En effet, on va chercher un point à droite de plus que à gauche. Cette dissymétrie ne semble pas poser de problème et a été conservée pour des raisons historiques. Elle sera cependant réparée lors des modifications futures.

On va prendre le cas particulier de l'advection suivant x, mais les résultats et la méthode peuvent être appliqués à y et z. On peut écrire la fonction interpolante en un point fixe $(y, z, \mathbf{p})$ comme :

$$s(x) = \sum_{i_{start}-2\leq\kappa\leq i_{end}+3} \eta_\kappa B_\kappa(x), \tag{2.40}$$

avec i_{start} et i_{end} les points initiaux et finaux définis par notre processus. La fonction cubique B-spline à pas constant est la même que pour la partie de rotation (Eq.(2.43)) :

$$B_\kappa(x) = \frac{1}{6h^3}\begin{cases} (x-x_{\kappa-2})^3 & x_{\kappa-2}\leq x<x_{\kappa-1} \\ h^3+3h^2(x-x_{\kappa-1})+3h(x-x_{\kappa-1})^2-3(x-x_{\kappa-1})^3 & x_{\kappa-1}\leq x<x_\kappa \\ h^3+3h^2(x_{\kappa+1}-x)+3h(x_{\kappa+1}-x)^2-3(x_{\kappa+1}-x)^3 & x_\kappa\leq x<x_{\kappa+1} \\ (x_{\kappa+2}-x)^3 & x_{\kappa+1}\leq x<x_{\kappa+2} \\ 0 & \text{sinon} \end{cases} \tag{2.41}$$

avec $h = \Delta x$ pour ce cas particulier.

Le calcul de la fonction interpolante (Eq.(2.40)) est donc un système de $i_{end}-i_{start}+6$ inconnues donc 21 donc le cas particulier de VLEM. Il va donc falloir, de la même manière que pour la rotation, fixer des conditions. On fixe donc la condition que la fonction interpolante $s(x)$ est égale à la fonction $f(x)$ sur les points de grille (de $i_{start}-1$ à $i_{end}+2$) :

$$f(x_i) = s(x_i), \tag{2.42}$$

$$f(x_i) = \frac{1}{6}\eta_{i-1} + \frac{4}{6}\eta_i + \frac{1}{6}\eta_{i+1}. \tag{2.43}$$

Il reste donc 2 conditions à imposer et contrairement à l'interpolation pour les quantités de mouvement (pour la rotation $\mathcal{R}$), ce n'est pas sur la dérivée seconde aux bords mais sur la dérivée première aux points i_{start} et $i_{end}+1$ qu'elles seront imposées. On pose donc :

$$\begin{aligned} \partial_x f(x_{i_{start}}) = \partial_x s(x_{i_{start}}) &= \sum_{i_{start}-2\leq\kappa\leq i_{end}+3} \eta_\kappa \partial_x B_\kappa(x_{i_{start}}) \\ &= -\frac{1}{2h}\eta_{i_{start}-1} + \frac{1}{2h}\eta_{i_{start}+1}, \end{aligned} \tag{2.44}$$

$$\partial_x f(x_{i_{end}+1}) = \partial_x s(x_{i_{end}+1}) = -\frac{1}{2h}\eta_{i_{end}} + \frac{1}{2h}\eta_{i_{end}+2}. \tag{2.45}$$

Le fait d'avoir choisi la dérivée première en i_{start} et $i_{end}+1$ plutôt qu'en $i_{start}-1$ et $i_{end}+2$ va changer la manière de calculer les coefficients splines associés aux bords. En effet, ces derniers seront calculés après avoir déterminé les autres coefficients splines η_κ en utilisant la formule (2.43) tel que :

$$\begin{aligned} \eta_{i_{start}-2} &= 6f(x_{i_{start}-1}) - 4\eta_{i_{start}-1} - \eta_{i_{start}}, \\ \eta_{i_{end}+3} &= 6f(x_{i_{end}+2}) - 4\eta_{i_{end}+2} - \eta_{i_{end}+1}. \end{aligned} \tag{2.46}$$

Pour d'abord trouver les autres coefficients de spline, on va écrire sous forme matricielle le système :

$$\mathbf{M} * \mathbf{H} = \mathbf{F}. \tag{2.47}$$

Avec $\mathbf{H}$ correspondant à $(\eta_{istart-1}, \eta_{istart}, \cdots, \eta_{iend+2})^T$ et $\mathbf{F}$ correspondant à $(f'(x_{istart}), f(x_{istart}), \cdots, f(x_{iend+1}), f'(x_{iend+1}))^T$. Contrairement à la rotation qui a des conditions aux bords naturelles (c.à.d. dérivée seconde nulle aux bords), on va ici choisir les conditions aux bords d'Hermite ($s' = f'$ aux bords) [103]. La manière de calculer la dérivée première de f aux bords sera donnée par la suite.

La matrice $\mathbf{M}$ carrée de dimension $i_{end} - i_{start} + 4$:

$$\mathbf{M} = \frac{1}{6}\begin{pmatrix} \frac{-3}{h} & 0 & \frac{3}{h} & 0 & \cdots & 0 \\ 1 & 4 & 1 & 0 & \cdots & \vdots \\ 0 & 1 & 4 & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & 1 & 4 & 1 \\ 0 & 0 & \cdots & \frac{-3}{h} & 0 & \frac{3}{h} \end{pmatrix}. \tag{2.48}$$

Cette matrice va être décomposée à l'aide d'une décomposition LU de telle sorte que $\mathbf{L} * \mathbf{U} = \mathbf{M}$ avec :

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ -h/3 & 1 & 0 & \cdots & \cdots & \vdots \\ 0 & l_1 & 1 & \ddots & \cdots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & l_N & 1 & 0 \\ 0 & \cdots & 0 & -3l_N/h & 3l_{N+1}/h & 1 \end{pmatrix}, \tag{2.49}$$

et

$$\mathbf{U} = \frac{1}{6}\begin{pmatrix} \frac{-3}{h} & 0 & \frac{3}{h} & 0 & \cdots & 0 \\ 0 & d_1 & 2 & 0 & \cdots & \vdots \\ 0 & 0 & d_2 & 1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & d_{N+1} & 1 \\ 0 & 0 & \cdots & \cdots & 0 & 3d_{N+2}/h \end{pmatrix}. \tag{2.50}$$

Dans notre exemple, on a $N = i_{end} - i_{start} + 1$. On peut définir $d_1 = 4$, $l_1 = 1/4$, $d_2 = 4 - 2l_1$ et les relations de récurrences suivantes $l_i = 1/d_i$ et $d_{i+1} = 4 - l_i$, avec i allant de 2 à N et $l_{N+1} = 1/(d_N d_{N+1})$ et $d_{N+2} = 1 - l_{N+1}$.

Le système (2.47) est ensuite résolu avec les étapes suivantes :

- $\boldsymbol{\varphi} = \boldsymbol{L}^{-1}\boldsymbol{F}$.
- $\boldsymbol{H} = \boldsymbol{U}^{-1}\boldsymbol{\varphi}$.

Les résolutions sont faites numériquement en utilisant le pivot de Gauss. On a donc trouvé les coefficients de spline entre $i_{start} - 2$ et $i_{end} + 3$, il ne reste plus qu'à appliquer les formules (2.46) pour trouver les deux derniers.

La méthode de résolution a bien été définie, il faut maintenant bien définir la valeur de la dérivée première f' aux points i_{start} et $i_{end} + 1$. Ce résultat est encore une fois issu de [104] mais certaines étapes intermédiaires ont été développées et corrigées. En écrivant de manière généralisée pour les deux points tels que $i = i_{start}$ ou $i = i_{end} + 1$ on a :

$$f'(x_i) = s'(x_i) = -\frac{1}{2h}\eta_{i-1} + \frac{1}{2h}\eta_{i+1}. \tag{2.51}$$

En utilisant l'équation (2.43) pour isoler η_i et en remplaçant η_{i-1} et η_{i+1} dans l'équation précédente on trouve :

$$\begin{aligned} f'(x_i) = s'(x_i) =& \frac{1}{2h}\left(\frac{3}{2}f(x_{i+1}) - \frac{1}{4}\eta(x_i) - \frac{1}{4}\eta(x_{i+2})\right. \\ & \left. -\frac{3}{2}f(x_{i-1}) + \frac{1}{4}\eta(x_{i-2}) + \frac{1}{4}\eta(x_i)\right) \\ =& \frac{3}{4h}\left(f(x_{i+1}) - f(x_{i-1})\right) - \frac{1}{4}\left(s'(x_{i-1}) + s'(x_{i+1})\right). \end{aligned} \tag{2.52}$$

En remplaçant la valeur des $s'(x_{i-1})$ et $s'(x_{i+1})$ par cette même formule (2.52) on trouve :

$$\begin{aligned} f'(x_i) = s'(x_i) =& \frac{3}{4h}(f(x_{i+1}) - f(x_{i-1})) - \frac{3}{16h}(f(x_{i+2}) - f(x_{i-2})) \\ & + \frac{1}{16}(2s'(x_i) + s'(x_{i-2}) + s'(x_{i+2})). \end{aligned} \tag{2.53}$$

En passant le terme en $s'(x_i)$ du terme de droite au terme de gauche on retrouve :

$$\begin{aligned} f'(x_i) = s'(x_i) =& \frac{6}{7h}(f(x_{i+1}) - f(x_{i-1})) - \frac{3}{14h}(f(x_{i+2}) - f(x_{i-2})) \\ &+ \frac{1}{14}(s'(x_{i-2}) + s'(x_{i+2})). \end{aligned} \tag{2.54}$$

En reprenant le même processus qu'avant et en remplaçant $s'(x_{i-2})$ et $s'(x_{i+2})$ par (2.54) et en regroupant les termes en $s'(x_i)$ à gauche, on trouve :

$$\begin{aligned} f'(x_i) = s'(x_i) =& \frac{84}{97h}(f(x_{i+1}) - f(x_{i-1})) - \frac{21}{97h}(f(x_{i+2}) - f(x_{i-2})) \\ &+ \frac{6}{97h}(f(x_{i+3}) - f(x_{i+1}) + f(x_{i-1}) - f(x_{i-3})) \\ &- \frac{3}{194h}(f(x_{i+4}) - f(x_{i-4})) + \frac{1}{194}(s'(x_{i-4}) + s'(x_{i+4})). \end{aligned} \tag{2.55}$$

En reprenant une dernière fois le procédé pour $s'(x_{i-4})$ et $s'(x_{i+4})$, on a :

$$\begin{aligned} \frac{18817}{18818}s'(x_i) =& \frac{84}{97h}(f(x_{i+1}) - f(x_{i-1})) - \frac{21}{97h}(f(x_{i+2}) - f(x_{i-2})) \\ &+ \frac{6}{97h}(f(x_{i+3}) - f(x_{i+1}) + f(x_{i-1}) - f(x_{i-3})) - \frac{3}{194h}(f(x_{i+4}) - f(x_{i-4})) \\ &+ \frac{42}{9409h}(f(x_{i+5}) - f(x_{i+3}) + f(x_{i-3}) - f(x_{i-5})) \\ &- \frac{21}{18818h}(f(x_{i+6}) - f(x_{i+2}) + f(x_{i-2}) - f(x_{i-6})) \\ &+ \frac{3}{9409h}(f(x_{i+7}) - f(x_{i+5}) + f(x_{i+3}) - f(x_{i+1}) \\ &\qquad + f(x_{i-1}) - f(x_{i-3}) + f(x_{i-5}) - f(x_{i-7})) \\ &- \frac{3}{37636h}(f(x_{i+8}) - f(x_{i-8})) + \frac{1}{37636}(s'(x_{i-8}) + s'(x_{i+8})). \end{aligned} \tag{2.56}$$

En se servant de la différence finie au 4 ème ordre, on peut réécrire :

$$s'(x_{i\pm8}) = f'(x_{i\pm8}) = \pm\frac{-f(x_{i\pm10}) + 8f(x_{i\pm9}) - 8f(x_{i\pm7}) + f(x_{i\pm6})}{12h}. \tag{2.57}$$

Ce qui donne l'écriture compacte :

$$\partial_x s(x_i) = \frac{1}{h}\left(\sum_{j=-10}^{-1} \omega_j^- f_{i+j} + \sum_{j=1}^{10} \omega_j^+ f_{i+j}\right), \tag{2.58}$$

avec $\omega_j^- = -\omega_j^+$.

ω^-_{-10}	ω^-_{-9}	ω^-_{-8}	ω^-_{-7}	ω^-_{-6}
$0,22143E^{-5}$	$-1,77144E^{-5}$	$7,9715E^{-5}$	$-3,01146E^{-4}$	$1,11379E^{-3}$
ω^-_{-5}	ω^-_{-4}	ω^-_{-3}	ω^-_{-2}	ω^-_{-1}
$-4,1452E^{-3}$	$1,546474E^{-2}$	$-5,771377E^{-2}$	$0,21539034$	$-0,80384758$

TABLE 2.2 – Valeurs des coefficients ω^- pour l'estimation de la dérivée avec 20 points.

Dans l'article de Crouseilles et al. [104], les auteurs comparent cette méthode avec le fait de directement prendre la dérivé à l'ordre 4 aux points i_{start} et $i_{end}+1$. Ils observent que la conservation de la densité est bien meilleure avec cette méthode. Dans une partie ultérieure, on reviendra sur cette méthode pour regarder l'impact de la réduction du nombre de points et aussi proposer une autre méthode utilisant une quasi-interpolation sur les bords.

Pour résumer, le calcul de la fonction interpolante s par B-spline cubique va permettre de réaliser les advections spatiales (2.19-2.21) de la fonction f en chaque point de l'espace des phases. Cette dernière va être réalisée de manière indépendante pour chaque processus avec une communication entre les processus voisins :

- Envoie au processus de "gauche" $f(x_{i_{start}})$, $f(x_{i_{start}+1})$ et $\sum_{j=1}^{10}\omega_j^+ f(x_{i_{start}+j})$.
- Envoie au processus de "droite" $f(x_{i_{end}})$ et $\sum_{j=-10}^{-1}\omega_j^- f(x_{i_{end}+1+j})$.
- Réception du processus de "droite" de $f(x_{i_{end}+1})$, $f(x_{i_{end}+2})$ et $\sum_{j=1}^{10}\omega_j^+ f(x_{i_{end}+1+j})$.
- Réception du processus de "gauche" de $f(x_{i_{start}-1})$ et $\sum_{j=-10}^{-1}\omega_j^- f(x_{i_{start}+j})$.

Les expressions "droite" et "gauche" peuvent être remplacée respectivement par "bas" et "haut" pour les advections suivant y et "derrière" et "devant" pour les advections suivant z (voir sous-partie 2.2.4). On voit aussi que le nombre de points compris dans une direction d'un processus doit être de $N_\alpha \geq 11$ avec cette méthode (i_{start} à $i_{start}+10$).

On peut remarquer que le fait que les coefficients de splines soient définis entre $i_{start}-2$ et $i_{end}+3$ permet de donner une interpolation de la fonction $f(x_i)$ uniquement entre $(x_{i_{start}-1})$ et $(x_{i_{end}+2})$. En effet, si on regarde les équations (2.40) et (2.41), on s'aperçoit que si $(x_{i_{start}-2}) < x < (x_{i_{start}-1})$, on a besoin de la valeur de $\eta_{i_{start}-3}$ pour estimer $s(x)$ correctement. C'est-à-dire que l'on a une condition sur le décalage de l'advection tel que $|\frac{p_x}{m\gamma}|\Delta t \leq \Delta x$. Sachant que $\frac{p_x}{m\gamma} = v_x$ et que en relativité restreinte la vitesse de la lumière c est la vitesse maximale, on a donc la condition, non normalisée, suivante pour que l'interpolation fonctionne :

$$c\Delta t \leq \Delta x. \tag{2.59}$$

On verra par la suite que cette condition doit aussi être respectée dans le code afin

de respecter une condition CFL qui n'est, quant à elle, pas liée à la relativité. C'est la raison pour laquelle on a choisi de limiter l'interpolation aux points $x_{i_{start}-1}$ et $x_{i_{end}+2}$. Dans un cadre non relativiste, on voit que les advections locales peuvent poser des problèmes pour les particules très rapides qui ne sont plus limitées à la vitesse de la lumière.

L'extension des splines locaux à deux ou trois dimensions peut être utile pour des schémas numériques différents de celui présenté ici. Cependant, leurs implémentations se complexifient avec le nombre de dimensions puisqu'elles impliquent de plus en plus de communications avec les processus voisins (8 voisins à 2 dimensions et 26 voisins à 3 dimensions).

2.2.7 Évolution des champs électromagnétiques

VLEM combine les différentes méthodes d'advection avec deux des équations de Maxwell (respectivement Maxwell-Ampère (2.60) et Maxwell-Faraday (2.61)) afin de résoudre l'évolution temporelle des champs électromagnétiques ($\boldsymbol{E}$, $\boldsymbol{B}$). On rappelle leurs expressions non normalisées :

$$\frac{\partial \boldsymbol{E}}{\partial t} = \mathrm{c}^2 \mathrm{rot}(\boldsymbol{B}) - \frac{\boldsymbol{J}}{\epsilon_0}, \tag{2.60}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = -\mathrm{rot}(\boldsymbol{E}), \tag{2.61}$$

avec ϵ_0 la permittivité du vide et où le courant $\mathbf{J}$ va être calculé par l'intégrale suivante :

$$\boldsymbol{J} = e \iiint \frac{\boldsymbol{p}}{m\gamma} f d^3p. \tag{2.62}$$

En plus des deux équations de Maxwell ci-dessus (2.60,2.61), qui assurent le calcul de l'évolution temporelle des champs, la résolution de l'équation de Poisson (2.63) va être réalisée afin d'assurer au code une meilleure conservation de l'énergie. Elle s'écrit comme :

$$\mathrm{div}\boldsymbol{E} = \frac{\rho}{\varepsilon_0}, \tag{2.63}$$

avec ρ la densité de charge qui est calculée en supposant que les ions forment un fond neutralisant de densité constante n_0 :

$$\rho = e \int f d^3p - e n_0. \tag{2.64}$$

Ce n'est pas directement l'équation de Poisson, mais plutôt l'erreur qui est faite dessus, que l'on va regarder :

$$\epsilon_E = \mathrm{div}\boldsymbol{E} - \frac{\rho}{\varepsilon_0}. \tag{2.65}$$

On calcule le potentiel électrique lié à cette erreur à partir de $\nabla^2\phi = \epsilon_E$ comme :

$$\phi = TF^{-1}\left(\frac{TF(\epsilon_E)}{k_x^2 + k_y^2}\right), \tag{2.66}$$

avec TF la transformée de Fourier et TF^{-1} la transformée de Fourier inverse. Le facteur $k_x^2 + k_y^2$ vient de la transformée de Fourier de -$\nabla^2\phi$. On peut ainsi corriger les champs électriques avec :

$$\boldsymbol{E}_{corrige} = \boldsymbol{E} - (-\nabla\phi). \tag{2.67}$$

Idéalement, la résolution des équations (2.60-2.61) devrait conserver l'équation de Poisson ainsi que l'équation de Maxwell-Thomson :

$$\mathrm{div}\boldsymbol{B} = 0, \tag{2.68}$$

si ces dernières sont initialement conservées. Cependant, il a été observé que dû aux erreurs du code, on devait rajouter cette résolution de l'équation de Poisson afin de garantir une meilleure conservation des énergies.

On va maintenant décrire les équations pour l'évolution temporelle de $\boldsymbol{E}$ et $\boldsymbol{B}$ pour un système 3D3V puisque c'est le modèle le plus général. On peut donc expliciter et normaliser les équations (2.60-2.61) comme :

$$\frac{\partial E_x}{\partial t} = \left(\frac{\partial B_z}{\partial y} - \frac{\partial B_x}{\partial z}\right) - J_x, \tag{2.69}$$

$$\frac{\partial E_y}{\partial t} = \left(\frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x}\right) - J_y, \tag{2.70}$$

$$\frac{\partial E_z}{\partial t} = \left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}\right) - J_z, \tag{2.71}$$

$$\frac{\partial B_x}{\partial t} = \frac{\partial E_y}{\partial z} - \frac{\partial E_z}{\partial y}, \tag{2.72}$$

$$\frac{\partial B_y}{\partial t} = \frac{\partial E_z}{\partial x} - \frac{\partial E_x}{\partial z}, \tag{2.73}$$

$$\frac{\partial B_z}{\partial t} = \frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x}. \tag{2.74}$$

Numériquement, ces équations (2.69-2.74) sont résolues en utilisant un schéma explicite appelé schéma de Yee [105]. Ce type de schéma est basé sur une méthode saute-mouton (ou leap-frog) qui décale dans l'espace et dans le temps les différentes composantes de façon à avoir un ordre de résolution du schéma numérique égal à 2. Les champs sont donc répartis comme sur la figure 2.3 avec les champs électriques E_α décalés d'un demi point dans la direction α et les champs magnétiques B_α décalés d'un

demi-point dans les deux autres directions spatiales. L'évolution temporelle est aussi décalée avec les champs électriques calculés aux temps $n\Delta t$ et les champs magnétiques aux temps $(n+\frac{1}{2})\Delta t$. Cela permet de calculer les champs comme :

$$\begin{aligned} E^{n+1}_{x\, i+\frac{1}{2},j,k} =& E^{n}_{x\, i+\frac{1}{2},j,k} + \frac{\Delta t}{\Delta y}\left(B^{n+\frac{1}{2}}_{z\, i+\frac{1}{2},j+\frac{1}{2},k} - B^{n+\frac{1}{2}}_{z\, i+\frac{1}{2},j-\frac{1}{2},k}\right) \\ & - \frac{\Delta t}{\Delta z}\left(B^{n+\frac{1}{2}}_{y\, i+\frac{1}{2},j,k+\frac{1}{2}} - B^{n+\frac{1}{2}}_{y\, i+\frac{1}{2},j,k-\frac{1}{2}}\right) - \Delta t J^{n+\frac{1}{2}}_{x\, i+\frac{1}{2},j,k}, \end{aligned} \tag{2.75}$$

$$\begin{aligned} E^{n+1}_{y\, i,j+\frac{1}{2},k} =& E^{n}_{y\, i,j+\frac{1}{2},k} + \frac{\Delta t}{\Delta z}\left(B^{n+\frac{1}{2}}_{x\, i,j+\frac{1}{2},k+\frac{1}{2}} - B^{n+\frac{1}{2}}_{x\, i,j+\frac{1}{2},k-\frac{1}{2}}\right) \\ & - \frac{\Delta t}{\Delta x}\left(B^{n+\frac{1}{2}}_{z\, i+\frac{1}{2},j+\frac{1}{2},k} - B^{n+\frac{1}{2}}_{z\, i-\frac{1}{2},j+\frac{1}{2},k}\right) - \Delta t J^{n+\frac{1}{2}}_{y\, i,j+\frac{1}{2},k}, \end{aligned} \tag{2.76}$$

$$\begin{aligned} E^{n+1}_{z\, i,j,k+\frac{1}{2}} =& E^{n}_{z\, i,j,k+\frac{1}{2}} + \frac{\Delta t}{\Delta x}\left(B^{n+\frac{1}{2}}_{y\, i+\frac{1}{2},j,k+\frac{1}{2}} - B^{n+\frac{1}{2}}_{y\, i-\frac{1}{2},j,k+\frac{1}{2}}\right) \\ & - \frac{\Delta t}{\Delta y}\left(B^{n+\frac{1}{2}}_{x\, i,j+\frac{1}{2},k+\frac{1}{2}} - B^{n+\frac{1}{2}}_{x\, i,j-\frac{1}{2},k+\frac{1}{2}}\right) - \Delta t J^{n+\frac{1}{2}}_{z\, i,j,k+\frac{1}{2}}, \end{aligned} \tag{2.77}$$

$$\begin{aligned} B^{n+\frac{1}{2}}_{x\, i,j+\frac{1}{2},k+\frac{1}{2}} =& B^{n-\frac{1}{2}}_{x\, i,j+\frac{1}{2},k+\frac{1}{2}} + \frac{\Delta t}{\Delta z}(E^{n}_{y\, i,j+\frac{1}{2},k+1} - E^{n}_{y\, i,j+\frac{1}{2},k}) \\ & - \frac{\Delta t}{\Delta y}(E^{n}_{z\, i,j+1,k+\frac{1}{2}} - E^{n}_{z\, i,j,k+\frac{1}{2}}), \end{aligned} \tag{2.78}$$

$$\begin{aligned} B^{n+\frac{1}{2}}_{y\, i+\frac{1}{2},j,k+\frac{1}{2}} =& B^{n-\frac{1}{2}}_{y\, i+\frac{1}{2},j,k+\frac{1}{2}} + \frac{\Delta t}{\Delta x}(E^{n}_{z\, i+1,j,k+\frac{1}{2}} - E^{n}_{z\, i,j,k+\frac{1}{2}}) \\ & - \frac{\Delta t}{\Delta z}(E^{n}_{x\, i+\frac{1}{2},j,k+1} - E^{n}_{x\, i+\frac{1}{2},j,k}), \end{aligned} \tag{2.79}$$

$$\begin{aligned} B^{n+\frac{1}{2}}_{z\, i+\frac{1}{2},j+\frac{1}{2},k} =& B^{n-\frac{1}{2}}_{z\, i+\frac{1}{2},j+\frac{1}{2},k} + \frac{\Delta t}{\Delta y}(E^{n}_{x\, i+\frac{1}{2},j+1,k} - E^{n}_{x\, i+\frac{1}{2},j,k}) \\ & - \frac{\Delta t}{\Delta x}(E^{n}_{y\, i+1,j+\frac{1}{2},k} - E^{n}_{y\, i,j+\frac{1}{2},k}). \end{aligned} \tag{2.80}$$

Les courants J_α sont calculés en normalisant l'équation (2.62) puis en la réécrivant de manière discrète :

$$\begin{aligned} J_\alpha(x,y,z) &= \int \frac{p_\alpha}{\gamma(\boldsymbol{p})} f(x,y,z,\boldsymbol{p})\boldsymbol{dp} \\ &= \sum_{j_x=j_{x_{min}}}^{j_{x_{max}}} \sum_{j_y=j_{y_{min}}}^{j_{y_{max}}} \sum_{j_z=j_{z_{min}}}^{j_{z_{max}}} \frac{p_\alpha}{\gamma(p_x,p_y)} f(x,y,z,j_x,j_y,j_z)\Delta p_x \Delta p_y \Delta p_z, \end{aligned} \tag{2.81}$$

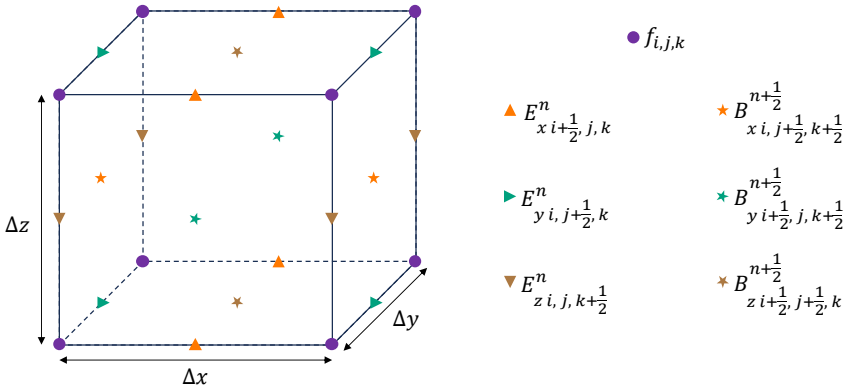

FIGURE 2.3 – Représentation d'une cellule élémentaire de la méthode de Yee. La fonction de distribution est définie sur les points de grille et est représentée par des ronds violets. Les champs électriques sont décalés d'un demi point dans la direction du champ et sont représentés par des triangles. Les champs magnétiques sont décalés d'un demi point dans les deux directions opposées au champ et sont représentés pas des étoiles. Pour les champs électriques et magnétiques, les couleurs orange, verte et marron représentent respectivement la composante x, y ou z.

Cette méthode d'intégrale est pour des pas de quantité de mouvement Δp_α constants. La fonction de distribution pouvant être considérée comme périodique (grâce aux bords nuls du domaine des quantités de mouvement) et étant C^∞, l'erreur faite par cette méthode d'intégrale est nulle [106].

La fonction de distribution étant définie sur les points de grille (i, j, k), le courant obtenu par l'équation précédente est $J_{\alpha\, i,j,k}$. Il faut donc les reconstruire spatialement mais aussi temporellement puisque l'on va effectuer une moyenne sur une itération complète pour avoir leurs valeurs en $(n + \frac{1}{2})\Delta t$ (voir figure 2.4) [28]. On aura donc :

$$J^{n+\frac{1}{2}}_{x\, i+\frac{1}{2},j,k} = \frac{1}{4}\left(J^n_{x\, i,j,k} + J^n_{x\, i+1,j,k} + J^{n+1}_{x\, i,j,k} + J^{n+1}_{x\, i+1,j,k}\right). \tag{2.82}$$

De même, comme les éléments des champs ne sont pas définis sur les points de grille spatiaux, il faut les reconstruire spatialement mais aussi temporellement pour les champs magnétiques. En effet, les champs qui interviennent dans la rotation $\mathcal{R}$ sont $E^n_{\alpha\, i,j,k}$ et $B^n_{\alpha\, i,j,k}$. On les détermine donc comme :

$$E^n_{x\, i,j,k} = \frac{1}{2}\left(E^n_{x\, i-\frac{1}{2},j,k} + E^n_{x\, i+\frac{1}{2},j,k}\right), \tag{2.83}$$

$$E^n_{y\,i,j,k} = \frac{1}{2}\left(E^n_{y\,i,j-\frac{1}{2},k} + E^n_{y\,i,j+\frac{1}{2},k}\right), \tag{2.84}$$

$$E^n_{z\,i,j,k} = \frac{1}{2}\left(E^n_{z\,i,j,k-\frac{1}{2}} + E^n_{z\,i,j,k+\frac{1}{2}}\right), \tag{2.85}$$

$$\begin{aligned} B^n_{x\,i,j,k} = &\frac{1}{8}\left(B^{n-\frac{1}{2}}_{x\,i,j-\frac{1}{2},k-\frac{1}{2}} + B^{n-\frac{1}{2}}_{x\,i,j-\frac{1}{2},k+\frac{1}{2}} + B^{n-\frac{1}{2}}_{x\,i,j+\frac{1}{2},k-\frac{1}{2}} + B^{n-\frac{1}{2}}_{x\,i,j+\frac{1}{2},k+\frac{1}{2}}\right) \\ &+ \frac{1}{8}\left(B^{n+\frac{1}{2}}_{x\,i,j-\frac{1}{2},k-\frac{1}{2}} + B^{n+\frac{1}{2}}_{x\,i,j-\frac{1}{2},k+\frac{1}{2}} + B^{n+\frac{1}{2}}_{x\,i,j+\frac{1}{2},k-\frac{1}{2}} + B^{n+\frac{1}{2}}_{x\,i,j+\frac{1}{2},k+\frac{1}{2}}\right), \end{aligned} \tag{2.86}$$

$$\begin{aligned} B^n_{y\,i,j,k} = &\frac{1}{8}\left(B^{n-\frac{1}{2}}_{y\,i-\frac{1}{2},j,k-\frac{1}{2}} + B^{n-\frac{1}{2}}_{y\,i-\frac{1}{2},j,k+\frac{1}{2}} + B^{n-\frac{1}{2}}_{y\,i+\frac{1}{2},j,k-\frac{1}{2}} + B^{n-\frac{1}{2}}_{y\,i+\frac{1}{2},j,k+\frac{1}{2}}\right) \\ &+ \frac{1}{8}\left(B^{n+\frac{1}{2}}_{y\,i-\frac{1}{2},j,k-\frac{1}{2}} + B^{n+\frac{1}{2}}_{y\,i-\frac{1}{2},j,k+\frac{1}{2}} + B^{n+\frac{1}{2}}_{y\,i+\frac{1}{2},j,k-\frac{1}{2}} + B^{n+\frac{1}{2}}_{y\,i+\frac{1}{2},j,k+\frac{1}{2}}\right), \end{aligned} \tag{2.87}$$

$$\begin{aligned} B^n_{z\,i,j,k} = &\frac{1}{8}\left(B^{n-\frac{1}{2}}_{z\,i-\frac{1}{2},j-\frac{1}{2},k} + B^{n-\frac{1}{2}}_{z\,i+\frac{1}{2},j-\frac{1}{2},k} + B^{n-\frac{1}{2}}_{z\,i-\frac{1}{2},j+\frac{1}{2},k} + B^{n-\frac{1}{2}}_{z\,i+\frac{1}{2},j+\frac{1}{2},k}\right) \\ &+ \frac{1}{8}\left(B^{n+\frac{1}{2}}_{z\,i-\frac{1}{2},j-\frac{1}{2},k} + B^{n+\frac{1}{2}}_{z\,i+\frac{1}{2},j-\frac{1}{2},k} + B^{n+\frac{1}{2}}_{z\,i-\frac{1}{2},j+\frac{1}{2},k} + B^{n+\frac{1}{2}}_{z\,i+\frac{1}{2},j+\frac{1}{2},k}\right). \end{aligned} \tag{2.88}$$

En reprenant les équations (2.60-2.61), on peut réécrire l'évolution des champs normalisés comme :

$$\frac{\partial^2 \boldsymbol{E}}{\partial t^2} = \left(\frac{\partial^2 \boldsymbol{E}}{\partial x^2} + \frac{\partial^2 \boldsymbol{E}}{\partial y^2} + \frac{\partial^2 \boldsymbol{E}}{\partial z^2}\right) - \frac{\partial \boldsymbol{J}}{\partial t}, \tag{2.89}$$

$$\frac{\partial^2 \boldsymbol{B}}{\partial t^2} = \left(\frac{\partial^2 \boldsymbol{B}}{\partial x^2} + \frac{\partial^2 \boldsymbol{B}}{\partial y^2} + \frac{\partial^2 \boldsymbol{B}}{\partial z^2}\right) + \mathrm{rot}\,(\boldsymbol{J})\,. \tag{2.90}$$

En négligeant les courants et en sachant que le schéma qui est utilisé est un schéma saute-mouton explicite, on retrouve les conditions CFL normalisées [107, 108] :

$$\Delta t^2(\Delta x^{-2} + \Delta y^{-2} + \Delta z^{-2}) \leq 1, \tag{2.91}$$

et dans sa version non normalisée 1D, on a :

$$c\Delta t \leq \Delta x. \tag{2.92}$$

Dans cette réécriture, on a enlevé les termes en Δy et Δz afin d'avoir une vision à une dimension de ces conditions CFL, cependant l'inéquation reste valable dans tous les cas où une dimension x existe, elle est juste plus restrictive. Cette forme permet cependant de faire une analogie avec la relativité restreinte qui est le cadre physique dans lequel le code VLEM fonctionne.

En effet, l'inéquation (2.92) (qui n'est pas liée à la relativité mais au schéma numérique explicite) sous entend que le pas d'espace doit être plus petit que la distance

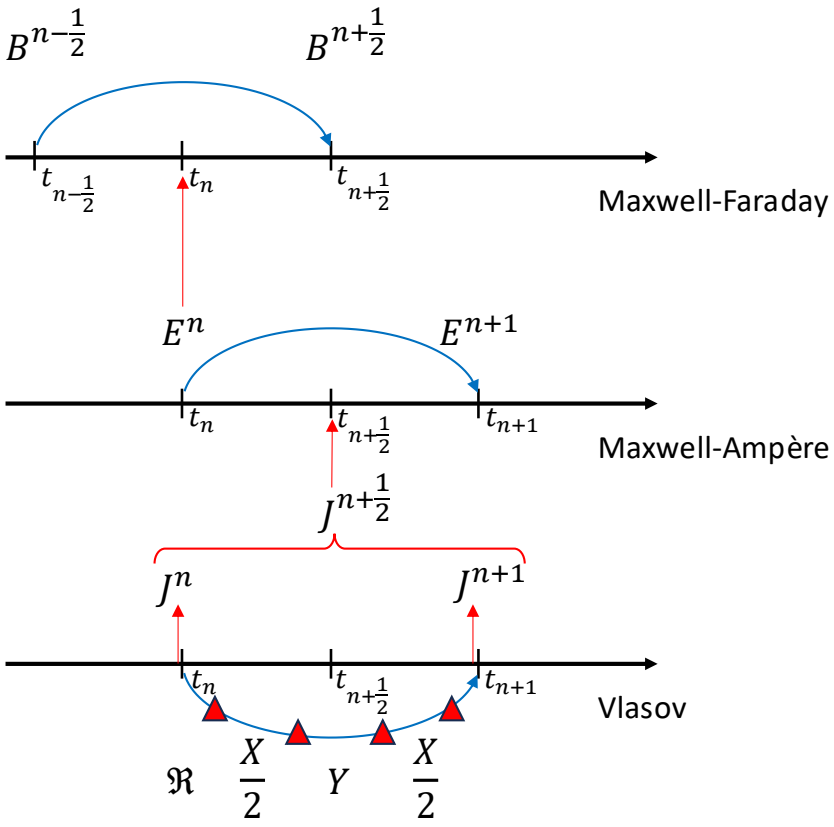

FIGURE 2.4 – Schéma numérique utilisé dans VLEM pour la résolution itérative temporelle utilisant les advections et la méthode de Yee. Les triangles rouges représentent le transfert (descente) d'information entre les grilles dans VLEM_EG et seront explicités dans le chapitre 3.

parcourue par un élément allant à la vitesse de la lumière sur un pas de temps. On rappelle que la vitesse de la lumière est, en relativité restreinte, la vitesse limite maximale que peuvent atteindre les particules (atteinte seulement par les particules de masse nulle). Cela sous entend que pour une advection spatiale temporelle ($\mathcal{X}$, $\mathcal{Y}$ ou $\mathcal{Z}$) on aura $|\frac{p_\alpha}{m\gamma}|\Delta t \leq c\Delta t \leq \Delta x$. Cela justifie le fait de calculer les coefficients de splines seulement jusqu'à $\eta_{istart-2}$ et $\eta_{i_{end}+2}$ pour décrire l'évolution des points entre i_{start} et i_{end} (voir Eq.(2.43)).

On peut aussi faire le rapprochement de la condition (2.92) avec le cône de lumière de la relativité. En relativité, le cône de lumière est une représentation dans l'espace temps des zones qui peuvent ou qui pourront influencer ou être influencées par un point donné (voir partie bleu sur la figure 2.5). Or la condition CFL va désigner un cône plus large ou aussi large que le cône de lumière, voir partie droite de la figure 2.5. Il faut donc que le domaine numérique soit plus large que le domaine physique afin de

s'assurer d'une convergence exacte de la solution et donc de pouvoir décrire tous les phénomènes physiques [107].

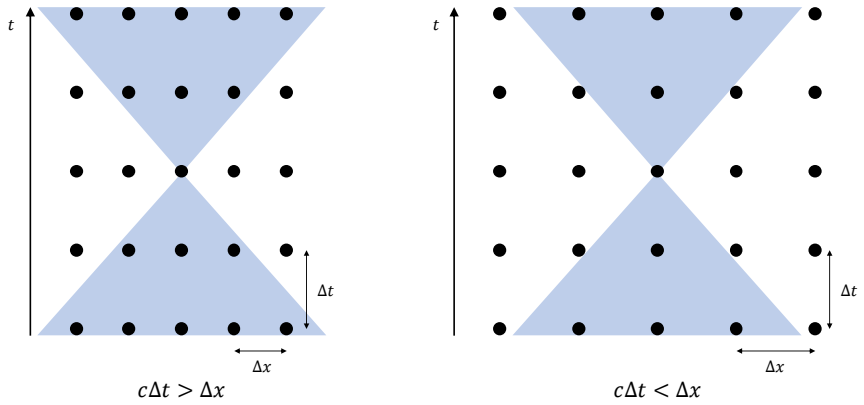

FIGURE 2.5 – Représentation du lien entre condition CFL et cône de lumière (en bleu). A gauche, pour $c\Delta t > \Delta x$ le domaine de dépendance numérique est plus fin que le domaine physique et le système est instable. A droite, à l'inverse les conditions CFL sont respectées ($c\Delta t < \Delta x$) et le système est stable.

Les informations et les calculs présentés dans cette sous-partie 2.2.7 sont valables pour un code 3D3V. Pour retrouver les équations dans le cas où il y a moins de dimensions spatiales, il suffit de considérer les dérivées partielles des directions manquantes comme nulle ($\frac{\partial}{\partial \alpha} = 0$ et donc de considérer $\frac{\Delta t}{\Delta \alpha} = 0$).

Si une variable de vitesse manque, il n'est plus possible de calculer le courant J_α associé. On considérera donc le champ électrique associé comme nul $E_\alpha = 0$ et aussi tous les champs qui ont besoin de E_α pour être calculés. C'est pourquoi dans les cas 2V, on a que les variables E_x, E_y et B_z.

2.2.8 Résumé du schéma numérique global

Afin de résumer tout ce sous-chapitre 2.2, on va présenter sous forme d'algorithme le schéma numérique global d'une itération temporelle du code VLEM3D3V. A 3 dimensions, le schéma numérique des advections à l'ordre 2 peut être résumé par la séquence $\left(\mathcal{R}\frac{\mathcal{X}}{2}\frac{\mathcal{Y}}{2}\mathcal{Z}\frac{\mathcal{Y}}{2}\frac{\mathcal{X}}{2}\right)^n$ et l'algorithme qui couple cela à la méthode de Yee et à la résolution de l'équation de Poisson est le suivant :

Toutes les parties sont parallélisées en MPI sauf l'équation (2.66) due aux transformées de Fourier qui sont réalisées de manière séquentielle sur un seul processus.

La figure 2.4 représente aussi le schéma global du code VLEM avec le couplage entre les advections et la méthode de Yee. C'est une version schématique de l'algorithme 1.

while $n \leq n_{max}$ **do**
  Résolution des Eqs.(2.78-2.80) $\left(\mathbf{B}(t_{n-1/2}) \rightarrow \mathbf{B}(t_{n+1/2})\right)$
  Calcul de $\mathbf{E_{i,j,k}}(t_n)$ et $\mathbf{B_{i,j,k}}(t_n)$ avec les Eqs. (2.83-2.88)
  Rotation $\mathcal{R}$
  Advection locale $\mathcal{X}/2$
  Advection locale $\mathcal{Y}/2$
  Advection locale $\mathcal{Z}$
  Advection locale $\mathcal{Y}/2$
  Advection locale $\mathcal{X}/2$
  Résolution de l'équation de Poisson (Eqs.(2.63-2.67)) (correction de $\mathbf{E}$)
  Résolution des Eqs.(2.75-2.77) $(\mathbf{E}(t_n) \rightarrow \mathbf{E}(t_{n+1}))$
  $n = n + 1$
end

Algorithme 1 : Algorithme simplifié de la résolution itérative temporelle de VLEM avec les advections et la méthode de Yee. n_{max} représente le nombre maximum d'itérations.

2.3 Modifications apportées au cours de la thèse

Une partie de ce travail de thèse a été de comprendre et d'améliorer le fonctionnement théorique et l'implémentation numérique du code VLEM. Ainsi, de nombreuses modifications ont été effectuées, sans prendre en compte la version avec raffinement de maillage. Ce sous-chapitre a pour objectif de citer les points les plus importants.

2.3.1 Optimisation du temps de calcul

Comme expliqué dans la sous-partie 2.2.4, le code est parallélisé MPI et OpenMP. Des optimisations sur la parallélisation de VLEM ont été effectuées principalement sur le calcul des courants, de la densité et des champs électriques et magnétiques.

Le rajout de la parallélisation OpenMP dans les calculs des courants et de la densité a été le plus important. En effet, ces derniers étant des intégrales (Eqs.(2.64) et (2.62)) et donc des sommes sur l'espace des phases complet, ils peuvent ralentir le code s'ils ne sont pas parallélisés.

On va donc comparer les temps de calcul et l'optimisation de la parallélisation OpenMP entre l'ancienne version du code et la nouvelle version datant de la fin de la thèse. On a choisi le code VLEM2D2V avec $N_x \times N_y \times = 256^2$ et $N_{p_x} \times N_{p_y} = 32 \times 64$, sur 500 itérations temporelles et avec 4 processus MPI pour toutes les simulations. On a fait varier le nombre de taches OpenMP de 1 à 32, cette limite maximale étant fixée par le calculateur qui a un maximum de 40 coeurs par noeud (les taches OpenMP devant être dans le même noeud pour éviter des augmentations du temps d'accès mémoire).

On résume les résultats des simulations dans les tableaux 2.3 et 2.4.

Les tests ont été réalisés sur le supercalculateur Jean Zay de l'IDRIS sur la partition CPU_P1 qui contient 720 noeuds de calculs scalaires avec 2 processeurs Intel Cascade Lake 6248 (20 coeurs à 2,5 GHz), c'est-à-dire 40 coeurs par noeud et 192 GB de mémoire par noeud [109].

Pour plus d'informations, on peut lire dans [109] : "Jean Zay est un supercalculateur comportant un partie Eviden BullSequana XH3000 et une partie HPE SGI 8600 formant un total de cinq partitions : une partition contenant des nœuds scalaires (ayant uniquement des CPU) et quatre partitions contenant des nœuds accélérés (nœuds hybrides équipés à la fois de CPU et de GPU). Les nœuds de calcul HPE SGI 8600 sont interconnectés par un réseau Intel Omni-PAth et ceux de l'extension Eviden BullSequana XH3000 sont interconnectés par un réseau Infiniband. L'ensemble des nœuds accède à un système de fichiers partagé à très forte bande passante."

p (Nombre de taches OpenMP)	$T_{CPU}(s)$	$T_{\text{Écoulé}}(s)$	$T_{CPU}/T_{\text{Écoulé}}$	$S(p)$	$E(p)$
1	5071,373	5090,478	0,996	1,000	1,000
2	5389,727	3204,263	1,682	1,589	0,794
4	6111,275	2282,969	2,677	2,230	0,557
8	7644,662	1838,616	4,158	2,769	0,346
16	11288,342	1652,444	6,831	3,081	0,193
32	16668,359	1476,781	11,287	3,447	0,108

TABLE 2.3 – Accélération $S(p)$ et efficacité $E(p) = S(p)/p$ en fonction du nombre p de taches OpenMP pour l'ancienne version du code VLEM2D2V. $S(p) = T_{\text{Écoulé}}(1)/T_{\text{Écoulé}}(p)$.

p (Nombre de taches OpenMP)	$T_{CPU}(s)$	$T_{\text{Écoulé}}(s)$	$T_{CPU}/T_{\text{Écoulé}}$	$S(p)$	$E(p)$
1	4719,814	4737,061	0,996	1,000	1,000
2	4876,828	2446,933	1,993	1,936	0,968
4	4996,249	1254,155	3,984	3,777	0,944
8	5503,171	691,510	7,958	6,850	0,856
16	6108,353	384,874	15,871	12,308	0,769
32	7903,385	250,254	31,581	18,929	0,592

TABLE 2.4 – Accélération $S(p)$ et efficacité $E(p) = S(p)/p$ en fonction du nombre p de taches OpenMP pour la nouvelle version du code VLEM2D2V. $S(p) = T_{\text{Écoulé}}(1)/T_{\text{Écoulé}}(p)$.

Les temps CPU (T_{CPU}) représentent la somme moyenne des temps de calcul OpenMP pour une tâche MPI. Les temps écoulés $T_{\text{Écoulé}}$ représentent le temps moyen des taches OpenMP.

En traçant l'accélération $S(p)$ en fonction de p pour les deux versions du code et en comparant avec l'accélération théorique $S(p) = p$ (Fig.2.6), on se rend compte que la nouvelle version parallélise beaucoup mieux en OpenMP que l'ancienne version. Cela revient, pour des nombres de taches OpenMP égal à 16 (32), à diviser par 4 (5) le temps de calcul par rapport à avant.

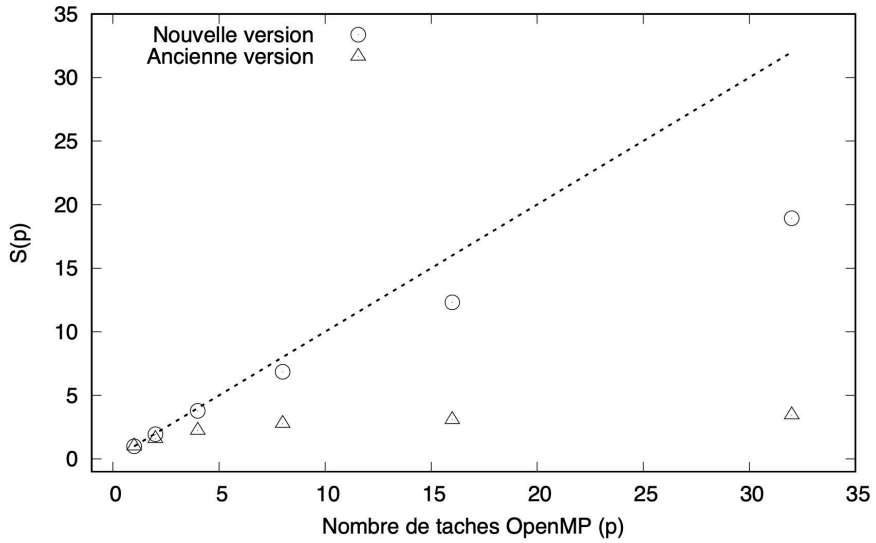

FIGURE 2.6 – Accélération $S(p)$ en fonction du nombre de taches OpenMP pour l'ancienne version du code (triangle) et la nouvelle version (rond).

Des temps de simulation plus longs auraient permis de mieux se rendre compte de la parallélisation de la boucle itérative temporelle en accordant moins d'importance à l'initialisation et la fin du code. Cependant les temps de calcul pour peu de taches OpenMP auraient été très importants.

2.3.2 Implémentation de nouveaux diagnostics

Les diagnostics représentent l'ensemble des parties du code ou post-code qui permettent de décrire les procédés qui se produisent lors de la simulation. Les deux diagnostics qui ont nécessité des modifications sont l'entropie et l'évolution temporelle des modes électromagnétiques.

Calcul de l'entropie

L'entropie (S) est un invariant de Casimir (voir sous-partie 1.3.2) qui peut s'exprimer dans la définition usuelle de Shannon comme :

$$S = -k_B \int \int f \ln(f) \boldsymbol{dxdp}, \tag{2.93}$$

avec k_B la constante de Boltzmann.

Si on cherche à réécrire cette intégrale de manière numérique pour des pas d'espace $\Delta \boldsymbol{x}$ et de quantité de mouvement $\Delta \boldsymbol{p}$ constants et en normalisant, on a :

$$S = -\sum_{\boldsymbol{x}} \sum_{\boldsymbol{p}} f \ln(f) \Delta \boldsymbol{x} \Delta \boldsymbol{p}. \tag{2.94}$$

Cette écriture est générale et peut être développée en fonction du nombre de dimensions spatiales ou des quantités de mouvement tel que l'on a $\Delta \boldsymbol{x} = \Delta x \Delta y \Delta z$ et $\Delta \boldsymbol{p} = \Delta p_x \Delta p_y \Delta p_z$.

L'un des problèmes de certains codes Semi-Lagrangiens est que la fonction de distribution peut prendre des valeurs négatives qui ne sont pas physiques. Ces dernières sont généralement très faibles et sont présentes dans les zones de petites oscillations proches de 0. Il existe des codes Semi-Lagrangiens qui imposent la positivité de la fonction de distribution [24, 110] mais cela nécessite souvent d'adapter les méthodes d'interpolation et on perd sur la conservation de certains invariants. Ce n'est cependant pas le cas du code VLEM qui autorise les valeurs négatives de la fonction de distribution.

Les valeurs négatives ne sont pas un problème pour les calculs de densité ou d'énergie puisque les oscillations positives vont en moyenne compenser les oscillations négatives. Le problème des valeurs négatives intervient surtout dans le calcul de l'entropie puisque la fonction $\ln(x)$ est une fonction $\mathbb{R}_+^* \longrightarrow \mathbb{R}$. Il faut donc numériquement choisir comment prendre en compte les valeurs négatives

Initialement, le choix a été fait de calculer l'entropie avec un seuil tel que si $f \leq 10^{-15}$, on fixe $\ln(f) = \ln(10^{-15})$. On peut donc résumer le calcul de l'entropie avec l'algorithme 1D1V suivant :

S=0
for $i_x = 0, N_x - 1$ **do**
  for $j_x = 0, N_{p_x} - 1$ **do**
    if $f \leq 10^{-15}$ **then**
      $S = S - f(i_x, j_x) \times 34,53877639491068 \times \Delta x \times \Delta p_x$
    else
      $S = S + f \times \ln(f) \times \Delta x \times \Delta p_x$
    end
  end
end

On voit dans cet algorithme que ce n'est pas l'entropie S mais plutôt $-S$ qui était calculée. La valeur $34,53877639491068$ est la valeur du seuil limite, c'est-à-dire $-\ln(10^{-15})$.

Le problème de cette définition est qu'elle entraîne une décroissance de l'entropie, ce qui n'est pas considéré usuellement comme physique. On a tracé sur la figure 2.7 (haut gauche) le type d'évolution qui pouvait apparaître. Si on considère que c'est $-S$ que l'on trace, on aperçoit donc une décroissance piquée aux alentours de $t\omega_p = 80$. On peut essayer de justifier cette décroissance piquée en regardant le nombre de points positifs et pourtant inférieurs à 10^{-15} (haut droite). Ces points ont une valeur de $\ln(f)$ qui est surestimée en comparaison de leur vraie valeur. Donc, quand aux alentours de $t\omega_p = 80$, l'ensemble de ces points est réduit à 0, il n'y a plus cette surestimation et donc la valeur de l'entropie (qui est censée être conservée) va diminuer. Cela va apparaître comme une augmentation piquée due au fait que l'on calcule $-S$. La diminution (et donc l'augmentation) de l'entropie qui survient après est "normale" numériquement et due à la conservation non exacte de l'entropie qui est principalement due à une précision trop faible de la grille.

Cette ancienne définition permet en fait de définir une variation d'entropie des paramètres de la fonction de distribution qui sont au dessus du seuil $S \to \Delta S = S_0 - S$, avec S_0 l'entropie des espèces sous le seuil et S l'entropie totale du système.

On a exploré deux pistes pour calculer différemment l'entropie. La première en négligeant les valeurs négatives (bas à gauche de Fig.2.7), la deuxième en calculant :

$$S = -\sum_{\boldsymbol{x}} \sum_{\boldsymbol{p}} f \ln(|f|) \Delta \boldsymbol{x} \Delta \boldsymbol{p}. \tag{2.95}$$

partout (bas à droite de Fig.2.7). Ces deux méthodes amènent à des résultats assez proches. C'est le deuxième cas qui a été retenu puisqu'il permet de compenser les petites oscillations non physiques de la fonction de distribution. On peut observer cela puisqu'il y a moins d'oscillations et de diminutions dans la valeur de l'entropie. L'algorithme en 1D1V qui a été retenu est donc le suivant :

S=0
for $i_x = 0, N_x - 1$ **do**
  for $j_x = 0, N_{p_x} - 1$ **do**
    $S = S - f \times \ln(|f|) \times \Delta x \times \Delta p_x$
  end
end

Représentation de l'évolution des différents modes

En physique des plasmas, il est intéressant de connaître le taux de croissance γ de l'instabilité d'un mode avec un vecteur d'onde $\boldsymbol{k}$ donné. Dans l'ancienne version du

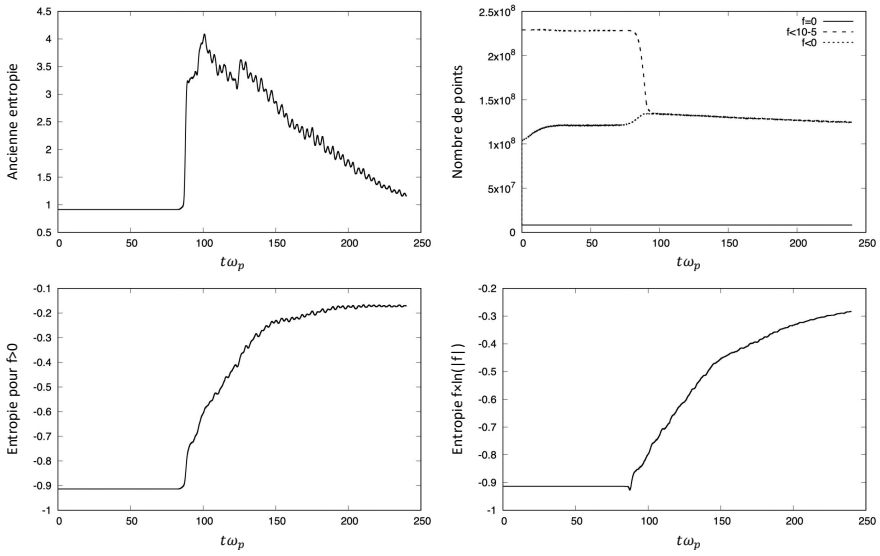

FIGURE 2.7 – Comparaison des méthodes de calcul de l'entropie pour un cas physique qui sera étudié dans la sous-partie 3.4.1 avec l'ancienne méthode (haut-gauche), une méthode qui néglige la contribution des valeurs négatives (bas-gauche) et la méthode de l'équation 2.95 (bas-droite). L'évolution du nombre de points ayant une valeur nulle, inférieure à 10^{-15} et inférieur à 0 est représentée en haut à droite.

code VLEM, l'information du taux de croissance était donnée par les énergies électrique ou magnétique calculées respectivement, de manière non normalisée et en 2D, comme $E_{e_\alpha} = \frac{\epsilon_0}{2L_xL_y}\int d^2x\ E_\alpha^2$ et $E_{m_\alpha} = \frac{\epsilon_0}{2L_xL_y}\int d^2x\ c^2B_\alpha^2$. Cependant ces énergies additionnent l'ensemble des modes contenus dans les champs et donc le taux de croissance qui en ressort représente la contribution de tous les taux de croissance.

Il est cependant possible d'avoir accès aux taux de croissance d'un seul mode en effectuant la transformé de Fourier sur toutes les dimensions spatiales des champs et en regardant l'évolution temporelle du mode d'intérêt. On rappelle la transformée de Fourier discrète à deux dimensions spatiales [111] :

$$\boldsymbol{A}(m,n) = \frac{1}{N_xN_y}\sum_{i_x=0,N_x-1}\ \sum_{i_y=0,N_y-1} A(i_x,i_y)e^{-i2\pi\left(\frac{mi_x}{N_x}+\frac{ni_y}{N_y}\right)}\ . \tag{2.96}$$

Cette formule est valable pour une variable A qui peut être les champs électriques $A = E_\alpha$ ou magnétiques $A = B_\alpha$.

Dans le code, cette transformée de Fourier est effectuée par une FFT (Fast Fourier Transform [112]) à 2 dimensions sur les champs et on retrouve les modes associés au vecteur d'onde en prenant $k_x = m\Delta k_x$ et $k_y = n\Delta k_y$ avec $\Delta k_x = 2\pi/L_x$ et $\Delta k_y = 2\pi/L_y$. Il suffit ensuite de tracer l'évolution temporelle de $\ln(A(m,n))$ et de regarder la pente pour avoir accès au taux de croissance γ du mode d'intérêt.

On peut envisager une amélioration de ce diagnostic permettant notamment de trouver le mode le plus instable ou celui avec le taux de croissance le plus élevé. Actuellement, cela se fait en comparant les valeurs "manuellement".

2.4 Développement de VLEM3D3V

La version 3D3V du code VLEM a été développée par Alain Ghizzo au cours de cette thèse mais le code nécessitant beaucoup de ressources numériques, il a fallu réfléchir à des techniques d'optimisation.

2.4.1 Problème de la mémoire

Le principal problème rencontré est la taille mémoire que nécessite un tel code. En effet, comme expliqué dans la sous-partie 2.2.4 le découpage en mémoire est spatial et à 3D, il se fait sur des cubes de 16^3 points spatiaux. La fonction de distribution est la partie qui va nécessiter le plus de mémoire puisqu'il faut en effet stocker des nombres réels proches de la double précision dans un tableau à 6 dimensions. Ces réels sont proches de la double précision puisqu'il sont définis sur 13 chiffres significatifs (contre 15 pour la double précision) et avec un exposant compris entre 10^{-200} et 10^{200} (contre 10^{-324} et 10^{308}) [113].

Après des tests, le nombre de points maximal toléré est $N_{p_x} \times N_{p_y} \times N_{p_z} = 32^3$, ce qui est très peu et ne donnerait pas une très bonne précision pour l'espace des quantités de mouvements. On va donc chercher à modifier le nombre de points spatiaux par processus MPI.

2.4.2 Modification à onze et dix points

La seule partie du code qui limite le nombre de points spatialement est celle qui traite les advections locales spatiales. Or, on a vu dans la sous-partie 2.2.6 que le nombre de points compris dans une direction d'un processus doit être de $N_\alpha \geq 11$. Il est donc possible de descendre à des cubes de 11^3 points, ce qui permettrait d'avoir environ $N_{p_x} \times N_{p_y} \times N_{p_z} = 46^3$.

Si on regarde encore ce qui a été dit au début de la sous-partie 2.2.6, c'est qu'il y a une dissymétrie entre "gauche" et "droite". En effet les coefficients de splines sont calculés entre $\eta_{i_{start}-2}$ et $\eta_{i_{end}+3}$ afin d'interpoler la fonction entre $f(x_{i_{start}-1})$ et $f(x_{i_{end}+2})$, or on a vu que dû aux conditions CFL (voir sous-partie 2.2.7) l'advection du point en $x_{i_{end}}$ ne pouvait pas aller plus loin que $x_{i_{end}} + \Delta x = x_{i_{end}+1}$. C'est-à-dire que l'on peut choisir de réduire de un à "droite" la taille de la méthode d'interpolation (on doit donc maintenant calculer de $\eta_{i_{start}-2}$ à $\eta_{i_{end}+2}$). Cela va réduire légèrement la précision de la

méthode d'interpolation mais va la rendre plus symétrique. On peut donc descendre à des cubes de 10^3 points et avoir $N_{p_x} \times N_{p_y} \times N_{p_z} = 51^3$.

2.4.3 Modification de la méthode d'advection à six points

Les résultats de la sous-partie précédente permettent d'augmenter le nombre de points maximums pour l'espace des quantités de mouvements, mais ne permettent pas encore d'atteindre une précision suffisante. On va donc reprendre le calcul des conditions aux bords de la méthode d'advection locale en s'arrêtant une itération de calcul plus tôt pour calculer la dérivée $f'(x_i)$. On reprend donc l'équation (2.55) :

$$\begin{aligned} f'(x_i) = s'(x_i) =& \frac{84}{97h}(f(x_{i+1}) - f(x_{i-1})) - \frac{21}{97h}(f(x_{i+2}) - f(x_{i-2})) \\ &+ \frac{6}{97h}(f(x_{i+3}) - f(x_{i+1}) + f(x_{i-1}) - f(x_{i-3})) \\ &- \frac{3}{194h}(f(x_{i+4}) - f(x_{i-4})) + \frac{1}{194}(s'(x_{i-4}) + s'(x_{i+4})), \end{aligned} \tag{2.97}$$

et on applique la dérivée d'ordre 4 sur $s'(x_{i-4})$ et $s'(x_{i+4})$. Ce qui donne une nouvelle écriture de $f'(x_i)$:

$$\partial_x s(x_i) = \frac{1}{h}\left(\sum_{j=-6}^{-1} \omega_j^- f_{i+j} + \sum_{j=1}^{6} \omega_j^+ f_{i+j}\right), \tag{2.98}$$

toujours avec $\omega_j^- = -\omega_j^+$.

ω_{-6}^- $4,295532646E^{-4}$	ω_{-5}^- $-3,436426117E^{-3}$	ω_{-4}^- $1,546391753E^{-2}$
ω_{-3}^- $-5,841924399E^{-2}$	ω_{-2}^- $0,2160652921$	ω_{-1}^- $-0,8041237113$

TABLE 2.5 – Valeurs des coefficients ω^- pour l'estimation de la dérivée avec 12 points.

On peut estimer la différence d'erreur entre la méthode actuelle et l'ancienne méthode. Ces erreurs sont calculées dans la dernière partie de l'annexe B. On a donc pour l'erreur L^∞ sur $f(x_{i_{start}-1})$ pour l'ancienne méthode d'advection à 10 points :

$$\epsilon_{f(x_{i_{start}-1})} = \frac{8h}{6}\epsilon_{s'_i} = 4,93305E^{-5}|f|_\infty. \tag{2.99}$$

Tandis que pour la nouvelle méthode d'advection à 6 points, on a :

$$\epsilon_{f(x_{i_{start}-1})} = \frac{8}{6}\epsilon_{s'_i} = 9,569614E^{-2}|f|_\infty. \tag{2.100}$$

Il y a trois ordres de grandeurs entre les deux erreurs, cependant avec la nouvelle méthode nous ne sommes plus limités à des cubes spatiaux de 10^3 mais nous pouvons descendre jusqu'à 6^3 ce qui fait 78% de gain mémoire. On peut donc atteindre $N_{p_x} \times N_{p_y} \times N_{p_z} = 85^3$ ce qui est une bonne précision.

En plus de fournir le calcul détaillé de l'erreur en norme L^∞ de cette méthode, l'annexe B contient une version des tableaux 2.2 et 2.5 en écriture fractionnaire.

2.4.4 Méthode de la quasi-interpolation

L'idée de cette méthode est de trouver une expression de $\eta_{i_{start}-1}$ et $\eta_{i_{end}+1}$ sans faire intervenir la dérivée ; pour cela on va utiliser une méthode de quasi-interpolation pour estimer 4 coefficients situés en $\eta_{i_{start}-1\pm j}$ et $\eta_{i_{end}+1\pm j}$. On va ensuite utiliser la méthode des splines cubiques pour retrouver la valeur de $\eta_{i_{start}-1}$ et $\eta_{i_{end}+1}$. On note que l'on a conservé la symétrisation introduite dans la sous-partie 2.4.2.

On peut réécrire l'expression de la fonction interpolante comme :

$$f(x) = s(x) = \sum_{k\in\mathbb{Z}} \eta_k \Phi\left(\frac{x-kh}{h}\right), \tag{2.101}$$

comme on veut une approximation interpolante, on peut écrire :

$$f(lh) = \sum_{k\in\mathbb{Z}} \eta_k \Phi\left(l-k\right). \tag{2.102}$$

On sait que pour les coefficients de B-spline, on a : $\Phi\left(1\right) = \Phi\left(-1\right) = 1/6$ et $\Phi\left(0\right) = 4/6$ (voir Eq.(2.41) ou Fig.2.2).

Or, on cherche à déterminer et réécrire les η_k en fonction de f aux points tel que :

$$\eta_l = \sum_{k=-n}^{n} a_k f(lh+kh), \tag{2.103}$$

avec les a_k des constantes que l'on va chercher à déterminer.

L'idée ici va être de choisir une petite valeur de n et de regarder les coefficients a_k minimisant l'erreur que l'on fait sur le coefficient η_l. On a donc choisi $n = 1$ et pour l'erreur (ϵ) on a :

$$\begin{aligned}
\epsilon &= f_0 - \sum_{k\in\mathbb{Z}} \eta_k \Phi\left(\text{-}k\right) \\
&= f_0 - \sum_{k\in\mathbb{Z}} \sum_{l=-1}^{1} a_l f(lh+kh) \Phi\left(\text{-}k\right) \\
&= f_0 - \left(\frac{a_{\text{-}1}}{6} f_{\text{-}2} + \frac{4a_{\text{-}1} + a_0}{6} f_{\text{-}1} + \frac{a_{\text{-}1} + 4a_0 + a_1}{6} f_0 + \frac{a_0 + 4a_1}{6} f_1 + \frac{a_1}{6} f_2\right).
\end{aligned} \tag{2.104}$$

On a simplifié l'écriture avec $f(lh) = f_l$. Comme on veut un système symétrique, on choisit $a_1 = a_{\text{-}1}$. On a donc :

$$\epsilon = -\frac{a_1}{6}f_{-2} - \frac{4a_1 + a_0}{6}f_{-1} + \frac{6 - 4a_0 - 2a_1}{6}f_0 - \frac{a_0 + 4a_1}{6}f_1 - \frac{a_1}{6}f_2. \tag{2.105}$$

On va maintenant chercher les valeurs des a_k qui minimisent l'erreur jusqu'à l'ordre le plus haut :

- Pour $f(x) = 1$, on veut $\epsilon = 0$:

$$\epsilon = 0 = 1 - 2a_1 - a_0. \tag{2.106}$$

- Pour $f(x) = x$, ϵ vaut forcément 0 comme pour tous les $x^{(2n+1)}$ puisque les coefficients sont symétriques.
- Pour $f(x) = x^2$, on veut $\epsilon = 0$:

$$\epsilon = 0 = -\frac{8}{3}a_1 - \frac{1}{3}a_0. \tag{2.107}$$

Le système donné par les équations (2.106-2.107), nous donne $a_{-1} = a_1 = -1/6$ et $a_0 = 4/3$, on peut donc écrire :

$$\eta_k = c_k = -\frac{1}{6}f_{k-1} + \frac{4}{3}f_k - \frac{1}{6}f_{k+1}. \tag{2.108}$$

En sachant que l'erreur que l'on commet sur ce terme en norme L^∞ est :

$$\epsilon = \left|\frac{1}{36}\right||f|_\infty + \left|-\frac{1}{9}\right||f|_\infty + \left|\frac{1}{6}\right||f|_\infty + \left|-\frac{1}{9}\right||f|_\infty + \left|\frac{1}{36}\right||f|_\infty = \frac{4}{9}|f|_\infty. \tag{2.109}$$

L'erreur obtenue est importante, on ne va donc pas se servir directement de ce terme pour calculer la valeur de $\eta_{i_{start}-1}$ et $\eta_{i_{end}+1}$.

Comme on ne détermine plus de la même manière les conditions aux bords, on doit réécrire le système de matrice de (2.47)en :

$$\widetilde{M} * \widetilde{H} = \widetilde{F}. \tag{2.110}$$

Avec $\widetilde{H}$ correspondant à $(\eta_{i_{start}}, \cdots, \eta_{i_{end}})^T$, $\widetilde{F}$ correspondant à $(f(x_{i_{start}})$-$\eta_{i_{start}-1}/6, f(x_{i_{start}+1}), \cdots, f(x_{i_{end}-1}), f(x_{i_{end}}) - \eta_{i_{end}+1}/6)^T$ et :

$$\widetilde{M} = \frac{1}{6}\begin{pmatrix} 4 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & 4 & 1 & 0 & \cdots & \vdots \\ 0 & 1 & 4 & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & 1 & 4 & 1 \\ 0 & 0 & \cdots & 0 & 1 & 4 \end{pmatrix}. \tag{2.111}$$

La nouvelle décomposition LU de cette méthode est :

$$\widetilde{L} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ l_1 & 1 & 0 & \cdots & \cdots & \vdots \\ 0 & l_2 & 1 & 0 & \cdots & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 & l_N & 1 \end{pmatrix}, \tag{2.112}$$

et

$$\widetilde{U} = \frac{1}{6}\begin{pmatrix} 4 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & d_2 & 1 & 0 & \cdots & \vdots \\ 0 & 0 & \vdots & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 1 \\ \vdots & \vdots & \ddots & \ddots & 0 & d_{N+1} \end{pmatrix}, \tag{2.113}$$

avec $N = i_{end} - i_{start}$ et les conditions $d_1 = 4, l_1 = 1/4, d_{i+1} = 4 - l_i$ et $l_i = 1/d_i \ \forall i > 1$. On peut par ailleurs faire remarquer que $d_{i+1} = 4 - 1/d_i$ est une suite convergente qui va tendre vers $d_\infty = 2+\sqrt{3}$ étant le seul point attractif. Comme dit précédemment, on ne va pas directement calculer $\eta_{i_{start}-1}$ et $\eta_{i_{end}+1}$ avec la formule (2.108) mais on va estimer à la place $\eta_{i_{start}-1-j}, \eta_{i_{start}-1+j}$,$\eta_{i_{end}+1-j}$ et $\eta_{i_{end}+1+j}$. Ces nouveaux coefficients vont nous permettre d'estimer de manière plus précise les coefficients $\eta_{i_{start}-1}$ et $\eta_{i_{end}+1}$ en utilisant un système identique à (2.110) mais de taille plus petit. On va donc encore avoir :

$$\breve{M} * \breve{H} = \breve{F}. \tag{2.114}$$

Mais cette fois ci, $\breve{H}$ va être $(\eta_{i_{start}-1-j}, \cdots, \eta_{i_{start}-1+j})^T$ ou $(\eta_{i_{end}+1+j}, \cdots, \eta_{i_{end}+1-j})^T$. Avec j un entier qui pourra être choisi en fonction de la taille et/ou de la précision souhaitée. On peut aussi écrire $\breve{F}$ respectivement comme $(f(x_{i_{start}-1-j})\text{-}\eta_{i_{start}-2-j}/6, f(x_{i_{start}-j}), \cdots, f(x_{i_{start}-2+j}), f(x_{i_{start}-1+j})\text{-}\eta_{i_{start}+j}/6)^T$ et $(f(x_{i_{end}+1+j})\text{-}\eta_{i_{end}+2+j}/6, f(x_{i_{end}+2+j}), \cdots, f(x_{i_{end}-j}), f(x_{i_{end}+1-j})\text{-}\eta_{i_{end}-j}/6)^T$. Il est important de préciser que pour la partie concernant i_{end}, on a écrit le vecteur dans l'ordre décroissant. Cela va venir de la méthode de résolution explicitée après.

En sachant que l'on peut réécrire, à l'aide de l'équation (2.108) :

$$\begin{aligned} \eta_{i_{start}-2-j} &= (-f(x_{i_{start}-3-j}) + 8f(x_{i_{start}-2-j}) - f(x_{i_{start}-1-j}))/6, \\ \eta_{i_{start}+j} &= (-f(x_{i_{start}-1+j}) + 8f(x_{i_{start}+j}) - f(x_{i_{start}+1+j}))/6, \\ \eta_{i_{end}-j} &= (-f(x_{i_{end}-1-j}) + 8f(x_{i_{end}-j}) - f(x_{i_{end}+1-j}))/6, \\ \eta_{i_{end}+2+j} &= (-f(x_{i_{end}+1+j}) + 8f(x_{i_{end}+2+j}) - f(x_{i_{end}+3+j}))/6. \end{aligned} \tag{2.115}$$

La partie qui nous intéresse n'est pas la résolution complète de ce problème, mais juste la valeur de $\eta_{i_{start}-1}$ et $\eta_{i_{end}+1}$. On va donc choisir un moyen qui va limiter le nombre d'opérations et de communications entre les processus. On choisit, plutôt que d'utiliser la décomposition LU pour calculer les $\eta_{i_{start}-1}$ et $\eta_{i_{end}+1}$, de regarder l'inverse de la matrice $\breve{M}$ afin de ressortir la valeur de $\breve{H}(j+1)$ correspondant soit à $\eta_{i_{start}-1}$ soit à $\eta_{i_{end}+1}$. On peut donc écrire :

$$\eta_{i_{start}-1} = 6 * \sum_{k=-j}^{j} \breve{\omega}_k \breve{F}(j+k+1), \tag{2.116}$$

avec $\breve{\omega}_k$ les valeurs du vecteur $\breve{M}^{-1}(j+1,k)$. On donne la valeur de ces derniers dans l'annexe B. En utilisant la formule 2.108 dans $\breve{F}(1)$ et $\breve{F}(2j+1)$ on peut donc écrire :

$$\eta_{i_{start}-1} = \sum_{k=-j-2}^{j+2} \omega_k f_{i_{start}-1+k}. \tag{2.117}$$

En appliquant le même raisonnement, on trouve :

$$\eta_{i_{end}+1} = \sum_{k=-j-2}^{j+2} \omega_k f_{i_{end}+1+k}. \tag{2.118}$$

On voit que, pour éviter les échanges avec les seconds voisins, il faut que chaque processus ait un nombre de points dans une direction de l'espace $N_\alpha \geq j+3$. Cependant, en comparaison de la méthode précédemment utilisée, on peut choisir n'importe quelle valeur de j et nous ne sommes plus limités à un nombre de points dépendant de la méthode itérative utilisée aux équations (2.52-2.56).

On donne dans les tableaux 2.6 et 2.7 la valeur des coefficients ω_i pour $j=5$ et $j=3$, en notant que $\omega_i = \omega_{-i}$.

ω_{-7}	ω_{-6}	ω_{-5}	ω_{-4}
$-6,168270417E^{-5}$	$4,934616334E^{-4}$	$-2,282260054E^{-3}$	$8,8823094E^{-3}$
ω_{-3}	ω_{-2}	ω_{-1}	ω_0
$-3,330866025E^{-2}$	$0,1243523316$	$-0,4641006662$	$1,732050333$

TABLE 2.6 – Valeurs des coefficients ω pour l'estimation des coefficients splines aux bords avec 15 points.

On choisit de résumer dans l'annexe B tous les coefficients ω en écriture fractionnaire des deux méthodes d'estimation de la dérivée première de la fonction interpolante ($s(x)$) et pour les 2 méthodes de quasi-interpolation avec $j=5$ et $j=3$.

ω_{-5}	ω_{-4}	ω_{-3}
$-8,591065292E^{-4}$	$6,872852234E^{-3}$	$-3,178694158E^{-2}$
ω_{-2}	ω_{-1}	ω_0
$0,1237113402$	$-0,4639175258$	$1,731958763$

TABLE 2.7 – Valeurs des coefficients ω pour l'estimation des coefficients splines aux bords avec 11 points.

On peut donc désormais calculer les coefficients $\eta_{i_{start}-1}$ et $\eta_{i_{end}+1}$ avec les équations (2.117-2.118). On peut ainsi calculer les coefficients de splines servant à l'interpolation locale pour calculer les advections spatiales en résolvant le système Eq.(2.110) avec la décomposition LU explicitée Eqs.(2.112-2.113).

Dans l'annexe B on fait une estimation de l'erreur en norme L^∞ de la méthode de quasi-interpolation.

Cette nouvelle méthode permet donc de descendre à des décompositions MPI en cubes spatiaux de taille 8^3 pour $j = 5$ et 6^3 pour $j = 3$. Cela permet donc respectivement d'atteindre un nombre de points $N_{p_x} \times N_{p_y} \times N_{p_z} = 64^3$ et $N_{p_x} \times N_{p_y} \times N_{p_z} = 85^3$.

2.4.5 Comparaison entre les méthodes

Dans cette sous-partie, on va comparer les 5 méthodes différentes pour le calcul des bords de l'advection locale présentées précédemment : la méthode de calcul de s' avec 11^2 points spatiaux par processus (sous-partie 2.4.2), la même méthode avec 10^2 (sous-partie 2.4.2) et 6^2 (sous-partie 2.4.3) et les méthodes de quasi-interpolation (sous-partie 2.4.4) avec 8^2 ($j = 5$) et 6^2 ($j = 3$).

Afin de valider le bon fonctionnement de ces méthodes, on a choisi de faire des cas tests similaires à ceux de l'article présentant la méthode initiale [104]. On a donc adapté les paramètres de simulation qui ont été initialement choisis pour un code 1D1V au code VLEM2D2V. Le cas test de l'amortissement Landau faible n'a pas amené de différence entre les méthodes puisqu'il perturbe peu la distribution, on va donc maintenant regarder uniquement l'amortissement Landau fort.

Pour cela on va choisir une fonction de distribution qui correspond à la version relativiste de la Maxwellienne. Cette fonction est la fonction de Maxwell-Jüttner en 2D2V que l'on va la perturber suivant x pour amener à l'amortissement Landau :

$$f(x, y, p_x, p_y) = A \exp(-\mu E)\,(1 + \epsilon \cos(k_x x)), \tag{2.119}$$

avec A une constante d'intégration telle que : $\iint f \boldsymbol{dxdp} = 1$, E l'énergie normalisée relativiste $E = (1 + p_x^2 + p_y^2)^{1/2}$ et μ la température normalisée du plasma $\mu = \frac{m_e c^2}{k_B T_e} = 511/T[keV]$.

On ajoute aussi une perturbation sur le champ électrique E_x telle que :

$$E_x(x, y) = \epsilon k_x \sin(k_x x). \tag{2.120}$$

On choisit des paramètres identiques à l'article [104] c'est-à-dire un vecteur d'onde suivant x de valeur $k_x = 0,5$, des longueurs de boites spatiales $L_x = L_y = 2\pi/k_x = 4\pi$, un nombre de points sur l'espace des quantités de mouvement $N_{p_x} = N_{p_y} = 128$ et une perturbation $\epsilon = 0,5$. Pour les autres paramètres qui changent par rapport à l'article, on va choisir $k_B T = 50keV$ et $p_x \times p_y \in [-6, 6] \times [-6, 6]$. Pour le nombre de points spatiaux, on va chercher à avoir $N_x = N_y$ avec la valeur la plus proche pour toutes les méthodes. On a choisi $N_x = N_y = 120$ qui est multiple de 6, 8 et 10. Cependant pour le cas avec 11^2 points spatiaux par processus, il faut choisir $N_x = N_y = 121$. Afin d'atteindre le même temps final, c'est-à-dire $t\omega_p = 60$, on choisit $\Delta t = 0,01$ sur 6000 itérations.

Les résultats obtenus pour les 5 simulations sont identiques pour la variation temporelle des énergies (E_e, E_k et E_m) et de l'entropie (S). La seule différence intervient sur la densité dont l'évolution est représentée l'évolution sur la figure 2.8 pour les 5 méthodes différentes.

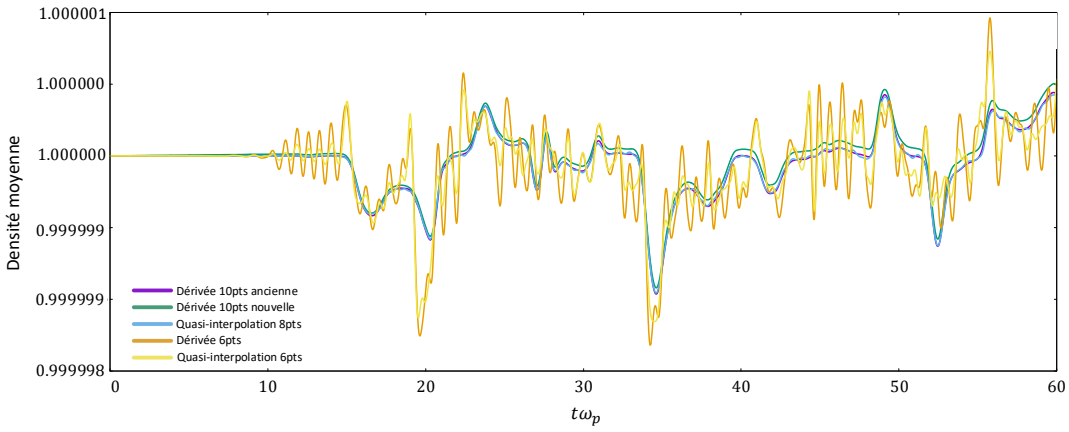

FIGURE 2.8 – Évolution temporelle de la densité pour une étude de l'amortissement Landau fort avec 5 méthodes d'advection locales différentes.

On observe que la densité est raisonnablement conservée dans la version initiale du code avec de légères variations de l'ordre de 10^{-6}.

Les deux méthodes ayant 6^2 points par domaine MPI conservent la densité dans les mêmes ordres de grandeurs que les trois autres méthodes. Cependant, on voit une perturbation supplémentaire sur cette densité qui est de plus haute fréquence que celle déjà présente. Même si cette perturbation semble être légèrement plus faible en intensité que celle déjà présente, on peut considérer que ces deux méthodes ne sont pas assez satisfaisantes ici.

La méthode qui sera donc préférée et qui est actuellement utilisée dans le code VLEM3D3V est la méthode de quasi-interpolation utilisant 8^3 points. Elle ne perturbe pas la conservation de la densité pour tous les cas qui ont été observés et de plus, elle permet de bien fonctionner avec la parallélisation OpenMP avec 2, 4 ou 8 taches.

Le code VLEM3D3V reste cependant encore en développement et les premières expériences numériques intéressantes devraient bientôt être regardées.

Chapitre 3

Implémentation des grilles emboîtées dans le code VLEM

Sommaire

3.1 Explication de la méthode 78
3.1.1 Stratégie de raffinement 78
3.1.2 Représentation des grilles raffinées 80
3.2 Choix du critère de raffinement 81
3.3 Implémentation dans VLEM 83
3.3.1 Initialisation de la grille raffinée 83
3.3.2 Évolution temporelle de la grille 84
3.3.3 Modification du schéma numérique global 89
3.3.4 Modification du calcul des intégrales 91
3.4 Validation de la méthode 97
3.4.1 Transition de OI à CFI avec VLEM2D2V 98
3.4.2 Transition de OI à CFI avec VLEM2D3V 110
3.4.3 Cas relativiste de l'instabilité oblique avec VLEM2D2V . . . 115
3.4.4 Test de performance numérique 118
3.5 Possibilité d'amélioration du raffinement 121
3.5.1 Grille non rectangulaire 121
3.5.2 Modification spatiale de la grille raffinée 122
3.5.3 Raffinement en espace avec les ondelettes d'Arakawa 123

Afin d'accélérer le code VLEM il est possible d'y implémenter le raffinement de maillage. Une première version utilisant les ondelettes permettant une analyse multi-résolution a été présentée dans [45]. Cependant cette dernière avait un coût numérique important et était restreinte à deux dimensions d'espace des phases (1D1V). De plus, elle raffinait aussi le domaine spatial ce qui ne va pas être le but de la nouvelle version.

Dans ce chapitre 3, on va présenter une nouvelle méthode de raffinement de maillage pour VLEM que l'on va appliquer uniquement sur l'espace des quantités de mouvement.

3.1 Explication de la méthode

3.1.1 Stratégie de raffinement

Comme on l'a vu dans l'introduction, les techniques de raffinement de maillage adaptatif (AMR) s'avèrent particulièrement utiles dans l'espace des quantités de mouvement, notamment en ce qui concerne le gain en temps de calcul qu'elles peuvent apporter à un code Semi-Lagrangien par rapport à un code PIC, avec une résolution spatiale équivalente et un nombre fixe de macro-particules (voir sous-partie 1.4.3). En outre, c'est dans l'espace des quantités de mouvement que se produit le phénomène de filamentation cinématique de f, qui constitue une contrainte forte puisqu'elle nécessite l'utilisation d'un échantillonnage élevé dans l'espace des $\mathbf{p}$ pour être correctement décrite. La filamentation de la fonction de distribution n'est pas un artefact numérique et peut être forcée ou même amplifiée par certaines instabilités physiques ou pendant les processus de reconnexion. Cela conduit à la nécessité d'ajouter des méthodes de raffinement de maillage à VLEM. Mais la complexité de ce code conduit au choix d'un type spécifique de raffinement de maillage à base de grilles emboîtées rectangulaires.

Cette technique subdivise les cellules qui satisfont le critère de raffinement (voir sous-chapitre 3.2), mais aussi celles qui permettent de compléter la grille pour former un rectangle. On utilise le terme de cellule que l'on définit comme l'espace formé par les 4 points en 2V (ou 8 points si 3V) : (j_x, j_y), $(j_x{+}1, j_y)$, $(j_x, j_y{+}1)$ et $(j_x{+}1, j_y{+}1)$. On va ainsi améliorer la résolution locale dans l'espace des quantités de mouvement en créant 2^{d_p} cellules "filles" faisant un quart (ou un huitième si 3V) de la surface (ou du volume) de la cellule "mère", où d_p est le nombre de dimensions de l'espace des quantités de mouvement. Le terme de cellules "filles" et cellules "mères" sont les mêmes que ceux définis dans le sous-chapitre 1.5.

Maintenant que l'on sait où raffiner, avec un critère qui sera défini plus tard, et comment subdiviser les cellules, on va réfléchir à une méthode AMR compatible avec VLEM. On rappelle que VLEM utilise une technique de décomposition de domaine pour répartir l'espace $\mathbf{x}$ entre les différents processus MPI (comme vu dans la sous-partie 2.2.4). Par conséquent, le choix d'un raffinement de maillage uniquement dans

l'espace des quantités de mouvement est compatible avec cette parallélisation et ne modifie pas l'architecture de VLEM. En effet, un maillage dans l'espace nécessiterait plus d'échanges entre processus et l'optimisation d'un tel code serait un véritable défi, notamment à 6D. De plus, si on souhaite uniquement améliorer la précision dans l'espace, c'est-à-dire augmenter le nombre de points, il suffit d'augmenter en même temps le nombre de processus. Le temps de calcul restera approximativement le même, grâce au bon "weak scalling" de VLEM [28], mais le coût de calcul augmentera.

Cette méthode permet, en plus, d'éviter de faire intervenir les techniques AMR dans les équations de Maxwell empêchant ainsi les instabilités numériques dans la méthode de Yee. En effet, dans ces articles de Fujimoto et al. [114, 115], les auteurs discutent le fait que la propagation des ondes électromagnétiques dans les zones de connexion entre les différentes grilles de raffinement peut produire des instabilités numériques.

Un autre critère à prendre en compte lors de la mise en œuvre du raffinement du maillage est la rotation $\mathcal{R}$ que l'on a décrite dans la sous-partie 2.2.5. En effet, l'équation (2.18) est résolue en utilisant la méthode des B-splines cubiques, qui nécessite, en un point fixe de l'espace x, d'avoir l'information sur l'entièreté du domaine des quantités de mouvement de la fonction de distribution. Numériquement, cette information est contenue dans un tenseur dont la dimension est égale au nombre de dimensions des quantités de mouvement (d_p). Ce tenseur est donc d'ordre trois ou deux, et on peut le comparer à une sorte de cuboïde rectangulaire ou à un rectangle. C'est pourquoi nous avons choisi des grilles emboîtées avec cette forme rectangulaire et pour généraliser, nous avons choisi le terme de grilles emboîtées "rectangulaires". Ce terme a été conservé pour la version 2D3V, même s'il n'est pas tout à fait exact. Cela signifie donc que la fonction de distribution peut être considérée comme un ensemble de rectangles emboîtées, un par niveau de raffinement, avec un nombre de points par volume de quantité de mouvement augmentant au fur et à mesure que l'on passe vers des rectangles plus internes.

Enfin, les advections spatiales Eqs.(2.19-2.21), explicitées dans la sous-partie 2.2.6, introduisent également une contrainte sur le raffinement. En effet, ces advections sont réalisées avec la méthode des splines locaux [28], ce qui signifie que nous avons besoin d'avoir l'information sur la fonction de distribution en un p_i donné pour tous les points x_i du processus et les 10 (ou 11) points les plus proches des deux processus voisins. Cela signifie que le « rectangle » d'un point donné dans (i_x, i_y) doit être le même pour toutes les valeurs de (i_x, i_y), c'est-à-dire pour l'ensemble de la grille spatiale.

De cette manière, on aboutit à l'idée d'une méthode de raffinement de maillage par rectangles emboîtés dans l'espace des quantités de mouvement, avec les mêmes grilles dans tout l'espace. On a représenté cela sur la figure 3.1.

La technique AMR va donc permettre une réduction du nombre total de points et des ressources informatiques associées à l'introduction de zones plus précises placées de manière hiérarchique uniquement dans les régions de l'espace des quantités de

mouvement où le piégeage des particules, la filamentation, l'accélération ou l'interaction onde-particule ont lieu, et où une résolution plus élevée est requise.

3.1.2 Représentation des grilles raffinées

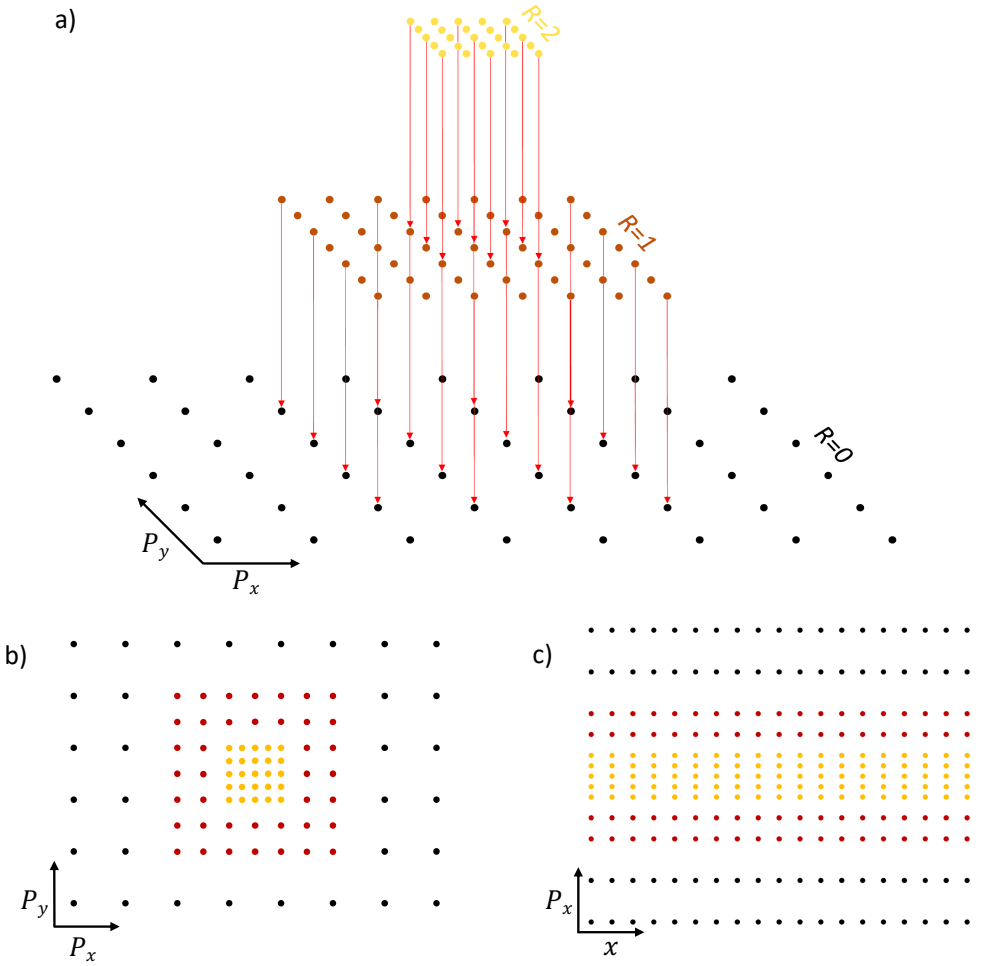

FIGURE 3.1 – Représentation du raffinement de maillage par grille emboîtée. (a) Visualisation des communications entre les différentes grilles en un point fixe (i_x, i_y). Les flèches rouges montrent la descente d'information entre les points situés au même endroit de l'espace des phases. (b) Visualisation de la fonction de distribution en (p_x, p_y) pour un point donné en (i_x, i_y). (c) Visualisation de la fonction de distribution en (x, p_x) pour un point donné en $(i_y = N_y/2, j_y)$.

Dans cette section, par souci de simplicité mais sans perte de généralité, toutes les explications et les figures se réfèreront à un code 2D2V.

La notion de grille emboîtée signifie que la fonction de distribution sera décrite à différents niveaux définis par leur niveau de raffinement R. Le niveau de base est R=0 et il aura des bords fixes qui sont les bords limites de la boîte : $[p_{x_{min}}(0), p_{x_{max}}(0)] \times [p_{y_{min}}(0), p_{y_{max}}(0)]$, ce qui correspondra aux dimensions maximales de notre système. Ensuite, les autres niveaux auront leurs propres $p_{x_{min}}(R), p_{y_{min}}(R), p_{x_{max}}(R), p_{y_{max}}(R)$ qui changeront au cours des itérations et qui peuvent aller jusqu'à atteindre les limites de la grille R=0 sous la condition $p_{\alpha_{min}}(R{+}1) \geq p_{\alpha_{min}}(R)$ et $p_{\alpha_{max}}(R{+}1) \leq p_{\alpha_{max}}(R)$. On défini R^{max} le niveau maximum de raffinement autorisé, qui peut être choisi initialement en fonction de la simulation, et R_n^{max} le niveau de raffinement au temps $t_n = n\Delta t$. On aura donc $R_n^{max}{+}1$ grilles rectangulaires discrétisant la fonction de distribution au temps t_n. Chacune d'entre elles est indépendante et elles ne communiqueront qu'en certains points spécifiques représentés par les flèches rouges sur la figure 3.1.

On remarque que R^{max} ne changera pas au cours de la simulation puisque c'est un paramètre d'entrée du code, cependant la valeur de R_n^{max} va pouvoir varier entre 0 et R^{max} en fonction du niveau de raffinement nécessaire au cours des itérations. Il y aura donc la possibilité de créer des niveaux de raffinement ou d'en détruire.

Le ratio de raffinement entre deux grilles est de 2. Ainsi, pour le cas 2V, une grille de niveau R=m+1 aura une densité de points dans l'espace des quantités de mouvement 4 fois supérieure à la grille R=m et pour un cas 3V, elle sera 8 fois supérieure. La figure 3.1 montre comment chaque grille est située l'une par rapport à l'autre. Les flèches montrent les points définis dans les deux grilles, situés au même endroit de l'espace des phases et qui communiqueront après chaque advections avec une "descente" de l'information des niveaux $R{+}1$ vers les niveaux R (voir sous-partie 3.3.3). On peut aussi commenter le fait que la frontière d'une grille de niveau $R{+}1$ doit se terminer sur un point décrit par le niveau R. Ceci sera justifié dans la sous-partie 3.3.3.

On utilisera i_x, i_y, j_x et j_y les indices respectivement de x, y, p_x et p_y. Un point situé en (i_x, i_y, j_x, j_y) de la grille R décrit la même position dans l'espace des phases que le point en (i_x, i_y, $2j_x$, $2j_y$) de la grille $R{+}1$. Les points spécifiques à la grille $R{+}1$ et qui ne sont pas définis sur la grille R seront les points qui contiennent une ou plusieurs valeurs impaires de j_α.

La version modifiée du code VLEM avec la technique de raffinement de maillage sera appelée VLEM_EG (pour "*Embedded Grid*") dans la suite de la thèse.

3.2 Choix du critère de raffinement

Dans cette sous-partie, on va expliciter le critère de raffinement que l'on utilise pour choisir si une cellule doit être raffinée ou non. Le critère est directement issu de [56], qui donne :

$$\sum_{k,l} |d_{kl}| > \epsilon\, 2^{(-s+\frac{d}{q})R}. \tag{3.1}$$

Ce critère permet de minimiser la norme L^q sur les dérivées $s^{\text{ième}}$ en dimension d, avec $\epsilon > 0$ un paramètre à fixer. Les $|d_{kl}|$ représentent les résidus que l'on va associer dans notre méthode à la différence entre la vraie valeur de la fonction de distribution sur le niveau R (f_R) et la valeur interpolée par la grille de niveau R-1, tel que :

$$\begin{aligned} &\sum_{k,l} |d_{kl}| = \\ &\sum_{k,l} \left| f_{\text{interpolée par R-1}}(i_x, i_y, j_x + \frac{k}{2}, j_y + \frac{l}{2}) - f_R(i_x, i_y, 2j_x + k, 2j_y + l) \right|. \end{aligned} \tag{3.2}$$

Dans l'article de Deriaz et al. [56], les auteurs ont choisi $k = l = (0, 1, 2)$. La signification de cette somme est illustrée dans la figure 3.2. Le critère nous dit donc que si la somme de ces différences est plus grande qu'une certaine valeur, dépendant de ϵ, que l'on va fixer en fonction de la précision souhaitée, il faut raffiner la grille est créer le niveau R+1.

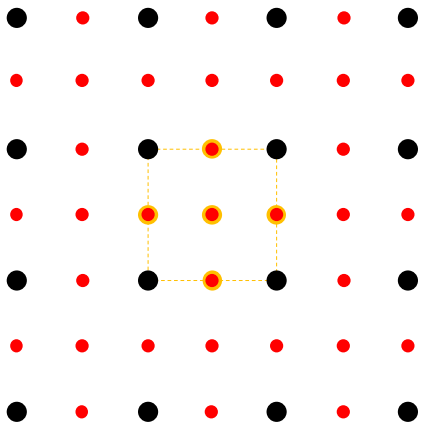

FIGURE 3.2 – Représentation du critère de raffinement. Les points de la grille R-1 sont en noir, ceux de la grille R sont en rouge et les points interpolés à partir de la grille R-1 qui serviront à calculer d_{kl} pour $k = l = (0, 1, 2)$ sont en orange. Pour le critère que l'on va choisir dans cette thèse, seul le point central en $k = l = 1$ sera comparé à sa valeur interpolée.

Pour notre part, on va se concentrer uniquement sur les points qui donnent la moins bonne interpolation qui sont les points au milieu d'une cellule [103] et donc on

choisira $k = l = 1$. Comme on veut minimiser l'erreur L^∞ sur la dérivée première des quantités de mouvement, le critère afin de raffiner sur la grille R+1 utilisé dans VLEM_EG devient :

$$|d_{11}| > \epsilon\, 2^{-(R-1)}. \tag{3.3}$$

On définit aussi un critère permettant de conserver la grille de niveau R qui est :

$$|d_{11}| > \epsilon\, 2^{-R}. \tag{3.4}$$

On rappelle que le choix de minimiser l'erreur L^∞ sur la première dérivée des quantités de mouvement permet de décrire au mieux le processus de filamentation qui va conduire à des gradients de quantités de mouvement de plus en plus forts.

Comme les paramètres d_{11} et ϵ sont des mesures de la variation de la fonction de distribution, ils partagent la même normalisation que f. Si on se réfère à la sous-partie 2.2.1, ces deux paramètres sont donc normalisés à $(mc)^{d_p}/n_0$ où d_p est la dimension de l'espace des quantités de mouvement. Dans le cas 3D, cette normalisation est m^3c^3/n_0.

Dans la suite le paramètre d_{11} sera appelé d.

3.3 Implémentation dans VLEM

3.3.1 Initialisation de la grille raffinée

La première étape avant de commencer l'itération temporelle est d'initialiser les grilles avec la fonction de distribution souhaitée. Les étapes pour initialiser sont représentées visuellement dans la figure 3.3 et vont être explicitées dans cette sous-partie :

- **Étape 1** : La grille de base du niveau (R=0), qui a une dimension égale à la taille de la boite, est définie de façon usuelle en associant à chaque point de la grille la valeur de la fonction de distribution normalisée. Les bords de la boîte de l'espace des quantités de mouvement (c'est-à-dire j_α=0 ou j_α=N_{p_α}-1) sont fixés à 0 comme dans VLEM.
- **Étape 2** : On interpole ensuite avec la méthode des splines cubiques, développée dans la sous-partie 2.2.5, la valeur de tous les points situés en $(j_x + \frac{1}{2}, j_y + \frac{1}{2})$ pour chaque point de l'espace (i_x, i_y). Cette interpolation est faite à partir de la grille R-1 (initialement R-1=0).
- **Étape 3** : Ces valeurs de la fonction de distribution interpolée vont être comparées aux vraies valeurs de la fonction de distribution initiale normalisée dont on a la forme analytique. Cette étape sera différente au cours de l'itération temporelle puisque la fonction de distribution analytique ne sera plus connue. On calcule donc $d = d_{11}$, la valeur absolue de la différence entre ces points, telle

que :

$$d = \left| f_{\text{interpolée par R-1}}(i_x, i_y, j_x + \frac{1}{2}, j_y + \frac{1}{2}) - f_{\text{réelle}}(i_x, i_y, j_x + \frac{1}{2}, j_y + \frac{1}{2}) \right|. \tag{3.5}$$

On applique ensuite le critère de raffinement introduit dans le sous-chapitre 3.2 (Eq. (3.4)). Si $d > \epsilon 2^{-R}$ la grille doit être raffinée sur le niveau R, sinon ce n'est pas nécessaire. En effet, dans la sous-partie précédente, cette condition permettait de maintenir le niveau de raffinement sur R mais comme il n'existe pas encore, ce critère va sélectionner les points qui doivent exister.

— **Étape 4** : On va ensuite sélectionner, dans tout l'espace des phases, les valeurs minimales et maximales des j_x et j_y qui doivent être raffinées. Cela permettra de raffiner tous les points contenus dans le rectangle de l'espace des quantités de mouvement dont les bornes sont ces valeurs minimales et maximales. S'il n'y a aucun point pour lequel la condition (3.4) est atteinte, aucune grille ne sera créée puisque cela signifie que l'interpolation est déjà suffisamment bonne pour décrire le problème et les étapes s'arrêtent ici. Sinon, les nouveaux points qui doivent être raffinés définiront la grille de niveau R et la valeur de la fonction de distribution en ces points sera la vraie valeur de la fonction analytique.

On remarque que les bords de la grille de niveau R vont toujours finir sur des valeurs paires de j_α.

On recommence ensuite de manière itérative ces étapes à partir de l'étape 2 en interpolant la grille de niveau R-1=m-1 afin de savoir si on crée la grille de niveau $R = m$ et ce jusqu'à ce qu'on atteigne m=R^{max} (le niveau maximal de raffinement que l'on souhaite) ou jusqu'à ce qu'aucune grille ne soit créée à l'étape 4. Le niveau de la dernière grille va fixer la valeur initiale de $R^{max}_{n=0}$.

Quand tous les niveaux de raffinement sont créés, on renormalise une dernière fois la densité à n_0 et on commence l'évolution temporelle.

3.3.2 Évolution temporelle de la grille

On rappelle que l'évolution temporelle à 2 dimensions d'espace suit la séquence $((\mathcal{R})\,(\mathcal{X}/2)\,(\mathcal{Y})\,(\mathcal{X}/2))^{n_{max}}$. Afin de maintenir une grille raffinée bien adaptée au problème, on va vérifier toutes les κ itérations si le raffinement doit être modifié. On choisit $\kappa = 5$ puisque c'est un bon compromis entre un suivi précis des fluctuations de la grille et un faible impact sur le temps de calcul. Dans les améliorations futures du code, on peut imaginer une valeur de κ qui évoluerait temporellement en fonction des fluctuations du problème, comme un fort champ électrique ou une accélération de particules importante.

Ce raffinement, qui a lieu toutes les κ itérations, va être effectué d'une manière assez similaire au raffinement initial présenté dans la sous-partie précédente 3.3.1. Des

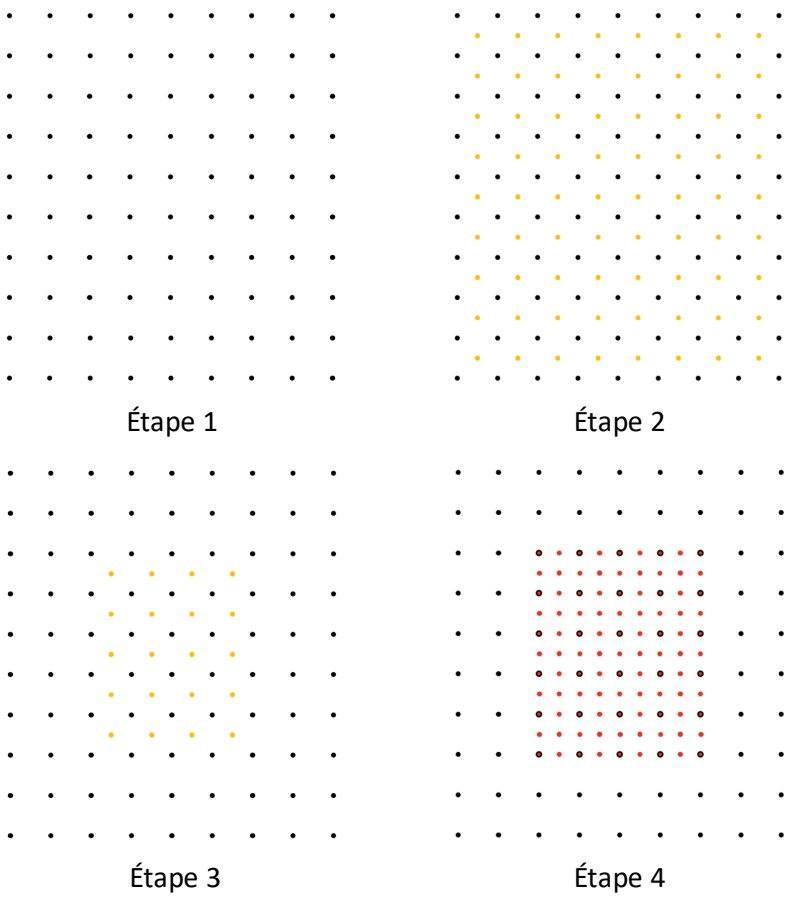

FIGURE 3.3 – Représentation des étapes d'initialisation de la grille en (p_x,p_y) pour tous les points de l'espace. Les étapes sont indiquées et sont expliquées dans la sous-partie 3.3.1.

étapes supplémentaires vont cependant être ajoutées, afin de prendre en compte le critère de raffinement Eq.(3.3) et afin d'anticiper les advections de la grille. On rappelle les étapes 1 à 4 (Fig.3.3) puisqu'il y aura de légères modifications comparé à la sous-partie précédente, puis on introduira les nouvelles étapes 5 à 8 (Fig.3.4) :

- **Étape 1** : Notre fonction de distribution est définie sur le niveau R-1.
- **Étape 2** : On utilise l'interpolation par splines cubiques pour estimer la valeur de la fonction de distribution au point $(j_x + \frac{1}{2}, j_y + \frac{1}{2})$.
- **Étape 3** : On calcule d (Eq.(3.6)) mais cette fois-ci $f_{\text{réelle}}$ n'est plus la fonction de distribution initiale analytique, puisque cette dernière a évolué, mais la valeur de la fonction de distribution sur le niveau R, c'est-à-dire :

$$d = \left| f_{\text{interpolée par R -1}}\left(i_x, i_y, j_x + \frac{1}{2}, j_y + \frac{1}{2}\right) - f_R(i_x, i_y, 2j_x{+}1, 2j_y{+}1)\right|. \quad (3.6)$$

De nouveau, si on a la condition $d > \epsilon 2^{-R}$ (Eq.(3.4)), cela veut dire que la grille de niveau R doit rester raffinée sur les cellules entourant ce point.

Il faut cependant ajouter une condition supplémentaire afin de pouvoir aussi créer la grille de niveau R+1. En effet, sans cette condition, il n'est pas possible que la grille s'étende au delà de ses limites initiales puisque les points de comparaison sont situés aux endroits où f_R existe. On verra à l'étape 5 qu'il est quand même possible sans ce critère que la grille s'étende mais moins rapidement qu'avec ce nouveau critère. On rajoute donc le critère introduit Eq.(3.3) : $d > \epsilon 2^{-(R\text{-}1)}$. Si la condition est remplie, le point doit aussi être raffiné sur le niveau R+1.

Afin que ce critère soit valide et fonctionnel pour créer la grille de niveau R=1, il faut créer une grille R=-1 factice. C'est-à-dire que l'étape 2 est aussi faite en interpolant les points en j_α impaire à l'aide des splines cubiques utilisant seulement la fonction de distribution sur les points j_α paire de R=0. Cela revient à faire une interpolation à partir de la grille R=-1 aillant les mêmes bords que R=0. La comparaison se fera avec les vrais points impaires de R=0. A noter que le critère $d > \epsilon 2^{-R}$ (Eq.(3.4)) avec R=0 ne va pas être utilisé puisque la grille conserve des limites minimales fixes.

— **Étape 4** : Encore une fois, on garde les valeurs de j_α minimale et maximale afin de définir la grille rectangulaire. Si aucun point n'atteint la condition pour être créé, aucune grille n'est créée et la grille est détruite si elle existait.

— **Étape 5** : Pour tous les points qui doivent être raffinés, on va anticiper où ils seront situés dans κ itérations temporelles. Cela veut dire que l'on va faire une advection vers l'avant. Comme on a la même grille pour chaque point de l'espace (i_x,i_y), on ne va pas regarder les advections spatiales mais seulement la rotation vers l'avant, on a donc :

$$P_\alpha = \frac{e\Delta T E_\alpha}{2} + p_x^{\delta'} R_{\alpha x}^{-1} + p_y^{\delta'} R_{\alpha y}^{-1} + p_z^{\delta'} R_{\alpha z}^{-1}, \tag{3.7}$$

avec $\Delta T = \kappa * \Delta t = 5\Delta t$ et où $\boldsymbol{R^{-1}}$, l'inverse de la matrice $\boldsymbol{R}$ introduite Eq.(2.26), s'écrit :

$$\boldsymbol{R^{-1}} = \begin{pmatrix} b_x^2+(1-b_x^2)\cos\phi & b_x b_y(1-\cos\phi)+b_z\sin\phi & b_x b_z(1-\cos\phi)-b_y\sin\phi \\ b_x b_y(1-\cos\phi)-b_z\sin\phi & b_y^2+(1-b_y^2)\cos\phi & b_y b_z(1-\cos\phi)+b_x\sin\phi \\ b_x b_z(1-\cos\phi)+b_y\sin\phi & b_y b_z(1-\cos\phi)-b_x\sin\phi & b_z^2+(1-b_z^2)\cos\phi \end{pmatrix}, \tag{3.8}$$

avec $p_\alpha^{\delta'} = p_\alpha + \frac{e\Delta T E_\alpha}{2}$ et $\phi = \omega_c \Delta T$.

En utilisant l'équation (3.7), on peut maintenant choisir les valeurs minimales et maximales des j_α correspondant aux p_α et P_α afin d'avoir le plus gros rectangle possible. De cette manière, on est désormais sûr que la grille raffinée sera bien définie durant les κ itérations.

Cette étape d'advection vers l'avant est inspirée de la thèse d'Olivier Hoenen [116] et à été adaptée au code VLEM et à un système de grille emboîtées dans l'espace des quantités de mouvement.

— **Étape 6** : On va augmenter la taille de chaque rectangle d'un point dans chaque direction afin de créer des points "fantômes" dont l'utilité sera expliquée dans la sous-partie 3.3.3.
— **Étape 7** : On force les bords de la grille rectangulaire à terminer sur des valeurs paires de j_α.

On répète les étapes 1 à 7 jusqu'à ce qu'on atteigne $R = R^{max}$ ou jusqu'à ce qu'aucune autre grille ne soit nécessaire, puis on procède à l'étape 8 :

— **Étape 8** : Pour la méthode de raffinement de maillage que l'on utilise, on a besoin d'une grille graduée ("gradded grid"). Cela signifie que l'on doit avoir un écartement d'un certain nombre de points entre 2 grilles séparées de 2 niveaux de raffinement (R et R+2). On choisit d'avoir au moins 3 points de la grille R+1 entre la grille R et la grille R+2. Cela va permettre de réduire la propagation numérique d'erreur dans la fonction de distribution mais aussi d'être cohérent avec la méthode d'intégration que l'on présentera dans la sous-partie 3.3.4.

On va donc vérifier si cette condition est déjà remplie et si ce n'est pas le cas, on va augmenter la taille de la grille de niveau R+1 pour remplir cette condition. On vérifie aussi à la fin de cette étape que les points "fantômes", qui sont situés à la bordure de chaque grille, sont toujours situés sur des valeurs paires de j_α (voir point orange de la figure 3.5). Ces deux conditions sont forcément remplies si la grille est sur les bords de la boîte des quantités de mouvement puisque la fonction de distribution est fixée à 0 en dehors de la boite.

Toutes les étapes sont résumées dans la figure 3.4.

À la fin de toutes ces étapes, il y a 3 scénarios différents pour un point sur une grille de niveau $R > 0$:

— Le point n'existe plus, dans ce cas sa valeur est détruite.
— Le point reste actif, dans ce cas il conserve sa valeur.
— Le point a besoin d'être créé. Dans ce cas, on va estimer sa valeur à l'aide de l'interpolation de Lagrange avec 4 points. C'est l'une des raisons pour lesquels on a besoin d'avoir une grille graduée. Une option a aussi été implémentée pour remplacer cette interpolation de Lagrange par l'interpolation par splines cubique. Cependant cette dernière allonge la durée de l'algorithme et ne correspond plus au modèle présenté dans l'annexe C.2, où l'on va décrire le schéma de raffinement présenté ici par une méthode d'ondelette.

On peut résumer le critère de raffinement de l'étape 3 par la figure 3.6.

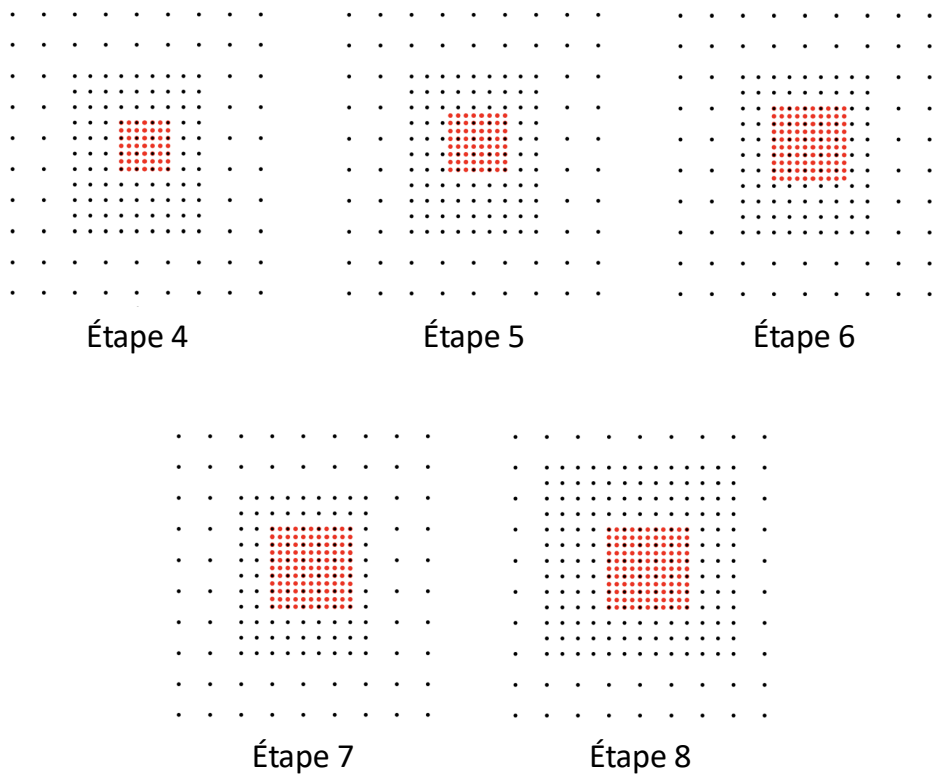

FIGURE 3.4 – Représentation des étapes du raffinement itératif de la grille en p_x/p_y pour tous les points de l'espace. Les étapes sont indiquées et expliquées dans la sous-partie 3.3.2. L'étape 4 est la même que dans la figure 3.3.

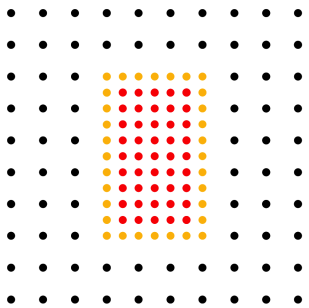

FIGURE 3.5 – Représentation des points "fantômes" entre deux grilles dans le plan (p_x, p_y). En orange sont représentés les points "fantômes" qui sont toujours localisés sur les points paires. En rouge sont représentés les points du niveau R+1 et en noir les points du niveau R.

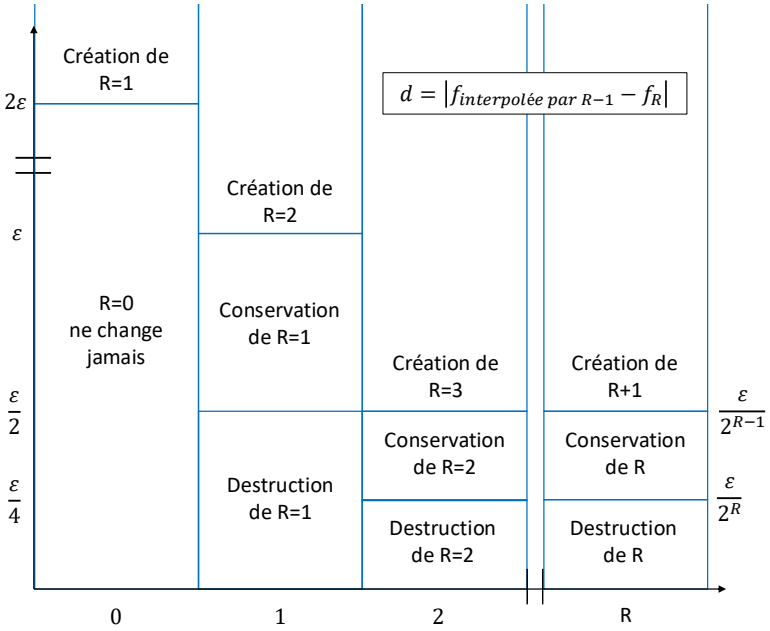

FIGURE 3.6 – Représentation du critère de raffinement en fonction de R. Conservation signifie que le point en $f_R(i_x, i_y, 2j_x{+}1, 2j_y{+}1)$ doit rester raffiné. Destruction signifie que le point en $f_R(i_x, i_y, 2j_x{+}1, 2j_y{+}1)$ doit être déraffiné, il peut cependant rester raffiné dû à d'autres conditions comme la complétion de la grille rectangle, la création de cellules fantômes ou les conditions de la grille graduée.

3.3.3 Modification du schéma numérique global

On va conserver le même schéma numérique que dans VLEM $((\mathcal{R})\,(\mathcal{X}/2)\,(\mathcal{Y})\,(\mathcal{X}/2))$. Cependant, pour chaque point $(i_x,\, i_y,\, j_x,\, j_y)$, on effectue les calculs d'advections sur la grille la plus fine sur laquelle il est défini. Cela permet d'éviter les répétitions dans le calcul et d'assurer une interpolation aussi précise que possible pour chaque point.

On effectue donc d'abord la rotation $(\mathcal{R})$ de manière séparée sur chaque grille en commençant par la grille la plus raffinée. On calcule pour chaque niveau les coefficients de splines en se servant de la fonction de distribution sur le niveau R comme expliqué dans la sous-partie 2.2.5. Cependant, la méthode de rotation présentée dans cette sous-partie a des conditions aux bords des splines naturelles ($s''{=}0$) qui sont vraies pour les bords de la grille $R{=}0$ mais qui peuvent être fausses pour les autres grilles qui n'ont pas $j_{\alpha_{min}} = 0$ et $j_{\alpha_{max}} = (N_{p_\alpha}(R{=}0) - 1) \times 2^R$. La condition aux bords naturelle pour les splines fait partie des 3 conditions (avec celle d'Hermite et la condition périodique) habituellement utilisées pour les splines cubiques puisque ce sont elles qui satisfont la relation d'intégrale (voir p234-236 de [103]). Cependant, on estime que dans le cas des

grilles qui sont dans le coeur de la fonction de distribution, la condition aux bords qui donne la dérivée seconde de s comme étant nulle semble fausse.

On choisit donc de modifier le vecteur $\mathbf{b}=(0,0)^T = (\partial^2_{p_x} f(j_{x_{max}}, p_{y_j}, p_{z_k}), \partial^2_{p_x} f(j_{x_{min}}, p_{y_j}, p_{z_k}))^T$ introduit dans (2.35) qui permet de calculer les $\gamma_\kappa(p_{y_j}, p_{z_k})$. Ce vecteur est aussi utilisé pour calculer les coefficients $\gamma^*_{\kappa,\mu}(p_{z_k})$ puis retrouver les coefficients $\eta_{\kappa,\mu,\nu}$ en prenant respectivement les valeurs $\mathbf{b} = \left(\partial^2_{p_y}\gamma_\kappa(j_{y_{max}}, p_{z_k}), \partial^2_{p_y}\gamma_\kappa(j_{y_{min}}, p_{z_k})\right)^T$ puis $\mathbf{b} = (\partial^2_{p_z}\gamma^*_{\kappa,\mu}(j_{z_{max}}), \partial^2_{p_z}\gamma^*_{\kappa,\mu}(j_{z_{min}}))^T$. Il faut donc, non seulement calculer la dérivée seconde de la fonction de distribution mais aussi celle des coefficients γ_κ et $\gamma^*_{\kappa,\mu}$. On calcule à l'aide d'un schéma décentré cette dérivée seconde comme :

$$\begin{aligned}
\partial^2_{p_x} f(j_{x_{min}}) = \frac{1}{h^2}\,&(45f(j_{x_{min}})\text{-}154f(j_{x_{min}}\text{+}1)\text{+}214f(j_{x_{min}}\text{+}2)\\
&\text{-}156f(j_{x_{min}}\text{+}3)\text{+}61f(j_{x_{min}}\text{+}4)\text{-}10f(j_{x_{min}}\text{+}5)) + O(h^4),\\
\partial^2_{p_x} f(j_{x_{max}}) = \frac{1}{h^2}\,&(\text{-}10f(j_{x_{max}}\text{-}5)\text{+}61f(j_{x_{max}}\text{-}4)\text{-}156f(j_{x_{max}}\text{-}3)\\
&\text{+}214f(j_{x_{max}}\text{-}2)\text{-}154f(j_{x_{max}}\text{-}1)\text{+}45f(j_{x_{max}})) + O(h^4).
\end{aligned} \tag{3.9}$$

avec $\partial^2_{p_x} f(j_x) = 0$ si $j_x = 0$ ou $j_x = (N_{p_x}(R\text{=}0) - 1) \times 2^R$ et $h = \Delta p_x$. On utilise les mêmes formules pour estimer la dérivée seconde de γ_κ et $\gamma^*_{\kappa,\mu}$ en remplaçant la variable correspondante pour la dérivée.

Actuellement dans le code il y a l'option de prendre tout le temps la dérivée seconde comme nulle ou de la calculer de cette manière. Une discussion sur ce choix, en comparant l'erreur obtenue avec différentes méthodes, est présentée dans l'annexe D. On y montre que cette méthode conserve le même ordre que le code sans raffinement de maillage.

Afin de réduire l'erreur faite sur les bords due à la mauvaise estimation de la dérivée seconde (soit avec Eq.(3.9) soit en la posant nulle), on choisit de calculer la rotation des bords de la grille de niveau R avec la grille R-1. C'est la raison pour laquelle ces points sont des points "fantômes", ils servent à l'interpolation mais ne sont donc pas calculés sur le niveau R. Cela permet aussi d'éviter d'avoir des problèmes dus à une décélération trop importante qui imposerait une advection en dehors des bords de l'interpolation de la grille R.

Le plan pour la nouvelle rotation est donc le suivant :

- Pour la grille R=R_n^{max}, on effectue la rotation et on remplace par les nouvelles valeurs les points de sa fonction de distribution sauf pour les points "fantômes".
- Pour les grilles de niveau R=R_n^{max}-$N > 0$, on effectue la rotation et on remplace par les nouvelles valeurs les points de sa fonction de distribution sauf pour les points "fantômes" et pour les points de l'espace des phases qui ont déjà été calculés par les niveaux plus raffinés. On utilise aussi les coefficients de splines pour calculer les points "fantômes" du niveau R_n^{max}-N+1.

— Pour la grille R=0, on effectue la rotation et on remplace par les nouvelles valeurs les points de sa fonction de distribution sauf pour les points de l'espace des phases qui ont déjà été calculés par les niveaux plus raffinés. On utilise aussi les coefficients de splines pour calculer les points "fantômes" du niveau R=1.

Après toutes ces étapes qui représentent la rotation, il y a une "descente" d'informations des grilles de niveau supérieur vers celles de niveau inférieur. C'est-à-dire que les points de la fonction de distribution définis sur la grille R+1 et qui existent aussi sur la grille R vont donner leurs valeurs à ces points de grille. Ce procédé de "descente" d'informations a aussi lieu après chaque advection spatiale. On représente cela avec les flèches rouges sur la figure 3.1.

On effectue donc ensuite la demi-advection spatiale $(\mathcal{X}/2)$. Dans VLEM, les splines locaux sont calculés pour chaque p_x mais dans la version par grilles emboîtées on va effectuer l'advection locale pour chaque p_x seulement sur la grille la plus raffinée pour laquelle le point est défini. Cela veut dire que le niveau R va effectuer l'advection seulement sur les points noirs sur la figure 3.5 mais pas sur les points intérieurs rouges et oranges qui seront calculés sur la grille R+1. La "descente" d'information qui va être faite juste après $(\mathcal{X}/2)$ va permettre de compléter les fonctions de distribution de chaque niveau.

Les mêmes étapes sont effectuées pour les deux autres advections $(\mathcal{Y})$ et $(\mathcal{X}/2)$. On résume toutes ces modifications dans l'algorithme 2 qui a été introduit pour VLEM dans la sous-partie 2.2.8.

Une justification de cette méthode de raffinement avec la méthode des ondelettes ainsi qu'une comparaison avec une méthode faisant intervenir les coefficients de spline sont présentées dans l'annexe C.

Les champs électriques et magnétiques ne sont pas directement impactés par le raffinement de maillage puisqu'ils sont définis seulement sur les variables d'espace. Cependant si on regarde leurs évolutions temporelles (Eqs.(2.75-2.80)), on remarque que celle du champ magnétique dépend uniquement du champ électrique mais pour l'évolution du champ électrique, il y a une dépendance du champ magnétique mais aussi des courants. Or ces derniers sont calculés par l'intégrale (2.62) qui peut s'écrire dans le code VLEM comme une simple sommation (Eq.(2.81)). On va voir que pour le code VLEM_EG ce n'est plus suffisant et il faut modifier la manière de calculer les intégrales.

3.3.4 Modification du calcul des intégrales

Dans le code VLEM, les intégrales sont calculées comme la somme des éléments de l'intégrale en chaque point multipliée par le volume de l'espace des phases associé, par

Raffinement initial avec les Étapes 1-4
while $n \leq n_{max}$ **do**
  Résolution de Eq.(2.80) $\left(B_z(t_{n-1/2}) \rightarrow B_z(t_{n+1/2})\right)$
  Calcul de $\mathbf{E_{i,j,k}}(t_n)$ et $B_{i,j,k}(t_n)$ avec les Eqs. (2.83-2.84) et (2.88)
  if $mod(n,5) = 1$ **then**
    Raffinement itératif avec les Étapes 1-8
  end
  Rotation $\mathcal{R}$
  Descente $(f_R(x,y,j_x,j_y) = f_{R+1}(x,y,2j_x,2j_y))$
  Advection locale $\mathcal{X}/2$
  Descente $(f_R(x,y,j_x,j_y) = f_{R+1}(x,y,2j_x,2j_y))$
  Advection locale $\mathcal{Y}$
  Descente $(f_R(x,y,j_x,j_y) = f_{R+1}(x,y,2j_x,2j_y))$
  Advection locale $\mathcal{X}/2$
  Descente $(f_R(x,y,j_x,j_y) = f_{R+1}(x,y,2j_x,2j_y))$
  Résolution de l'équation de Poisson (Eqs.(2.63-2.67)) (correction de E_x et E_y)
  Résolution des Eqs.(2.75-2.76) $\begin{pmatrix} E_x(t_n) \rightarrow E_x(t_{n+1}) \\ E_y(t_n) \rightarrow E_y(t_{n+1}) \end{pmatrix}$
  $n = n + 1$
end

Algorithme 2 : Algorithme simplifié de la résolution itérative temporelle de VLEM_EG, appliqué au cas 2D2V, avec les advections et la méthode de Yee (voir figure 2.4 en prenant en compte les triangles rouges). * * représentent les termes qui sont spécifique à VLEM_EG et qui n'existent pas dans VLEM. f_R est la fonction de distribution définie sur le niveau R et n_{max} représente le nombre maximum d'itérations. L'extension à 3V se fait en rajoutant les calculs des champs électriques et magnétiques manquant E_z, B_x et B_y. L'extension à 3D se fait en rajoutant les advections sur $\mathcal{Z}$ comme dans l'algorithme 1.

exemple pour le courant, dans le cas spécifique du code VLEM 2D2V, on a :

$$\begin{aligned} J_\alpha(x,y) &= \iint e\frac{p_\alpha}{m\gamma(\mathbf{p})}f(\mathbf{x},\mathbf{p})d^2p \\ &= \sum_{jx=0}^{Np_x\text{-}1}\sum_{jy=0}^{Np_y\text{-}1} e\frac{p_\alpha(j_\alpha)}{m\gamma(j_x,j_y)}f(x,y,j_x,j_y)\Delta p_x\Delta p_y. \end{aligned} \tag{3.10}$$

Cependant ceci n'est possible que si le pas de grille est constant. Dans le cas présent des grilles emboîtées, il faut revoir la méthode d'intégrale. On pose donc de manière générale l'intégrale I d'une fonction $F(\boldsymbol{x},\boldsymbol{p})$, dans un cas 2D2V, suivante :

$$\begin{aligned} I(x,y) &= \iint F(x,y,p_x,p_y)dp_xp_y \\ &= \sum_{jx=0}^{Np_x\text{-}1}\sum_{jy=0}^{Np_y\text{-}1} F(i_x,i_y,j_x,j_y)\Delta p_x\Delta p_y. \end{aligned} \tag{3.11}$$

L'idée qui a été retenue pour faire le calcul de ces intégrales est de supposer une grille homogène définie sur toute la boîte avec la précision du niveau R_n^{max} afin de pouvoir utiliser l'expression (3.11). Les points qui n'existent pas sur ce niveau de grille R_n^{max} vont être redistribués sur le niveau de grille R_n^{max}-1 en utilisant la méthode d'interpolation de Lagrange à 4 points. À deux dimensions de quantité de mouvement pour un point situé sur une seule valeur impaire, soit j_x soit j_y impaire, on utilise l'interpolation 1D suivante :

$$F_{R+1}(2j_x\text{+}1,2j_y) = \frac{1}{16}\left[\text{-}F_R(j_x\text{-}1,j_y)\text{+}9F_R(j_x,j_y)\text{+}9F_R(j_x\text{+}1,j_y)\text{-}F_R(j_x\text{+}2,j_y)\right], \tag{3.12}$$

tandis que pour un point situé sur une valeur impaire pour j_x et pour j_y on utilise l'interpolation 2D suivante :

$$\begin{aligned} F_{R+1}(2j_x\text{+}1,2j_y\text{+}1) = \frac{1}{16}\big[&-F_{R+1}(2j_x\text{+}1,2j_y\text{-}2) + 9F_{R+1}(2j_x\text{+}1,j_y) \\ &+9F_{R+1}(2j_x\text{+}1,2j_y\text{+}2) - F_{R+1}(2j_x\text{+}1,2j_y\text{+}4)\big], \end{aligned} \tag{3.13}$$

avec $F_R(j_x,j_y) = F_R(i_x,i_y,j_x,j_y)$ la fonction que l'on intègre et qui utilise la fonction de distribution définie sur le niveau R. On rappelle qu'un point situé en (i_x, i_y, j_x, j_y) sur la grille R décrit la même position dans l'espace des phases que le point (i_x, i_y, $2j_x$, $2j_y$) sur la grille R+1. Donc seuls les points qui n'existent pas sur R+1 et qui contiennent une ou plusieurs valeurs de j_α impaires seront redistribués sur les points de la grille R.

Ce processus est ensuite répété, si possible, sur les points qui n'existent pas sur le niveau R_n^{max}-1 afin de redistribuer leurs valeurs sur le niveau R_n^{max}-2. Lorsque l'on

atteint le niveau de grille R=0, qui est défini sur l'entièreté de la grille des quantités de mouvement, on arrête ce processus.

On va appliquer cela aux différents cas rencontrés dans l'intersection de nos grilles. Comme on n'a qu'une seule grille rectangulaire par niveau et qu'il y a une zone de 3 points au moins entre chaque grille (voir sous-partie 3.3.2) les zones d'intérêts sont limitées. On commence donc par étudier le cas de la bordure entre la grille R et R-1 située en n (voir tableau 3.1) dans un code 2V. Dans l'intégrale de VLEM, chaque point à un poids de 1, on peut donc écrire le poids de chaque point comme dans le tableau 3.1.

n-4	n-3	n-2	n-1	n	n+1	n+2	n+3	n+4	
1	1	1	1	1	(1)	1	(1)	1	2M
1	1	1	1	1	$*1*$	[1]	$*1*$	[1]	2M+1

TABLE 3.1

Les paramètres n et M vont permettre de développer le calcul de manière générale pour les 4 cotés du rectangle de la grille R. On a $n = j_{\alpha_{max}}(R)$ et $n+1 = j_{\alpha_{max}}(R)+1$ où α va représenter la direction orthogonale au côté. On a aussi $2M$ qui représente les valeurs paires et $2M$+1 les valeurs impaires de $j_\beta(R)$ avec β l'autre direction du système 2V, donc celle parallèle au côté. Les calculs que l'on va effectuer seront valides tant que l'on a $j_{\beta_{min}}(R)+2 < 2M < j_{\beta_{max}}(R)-2$. Pour les autres points manquants, il faudra regarder le calcul pour les coins.

Comme tous les points ne sont pas définis partout, on va développer la valeur de ceux qui ont été interpolés. Ils ont été représentés par une couleur et entourés de symboles en fonction de la manière dont ils ont été interpolés et donc de la manière dont ils vont repartir leurs poids, tel que l'on peut les développer :

$$\begin{vmatrix} / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & *1* & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \end{vmatrix} = \frac{1}{256} \begin{vmatrix} 1 & / & -9 & / & -9 & / & 1 \\ / & / & / & / & / & / & / \\ -9 & / & 81 & / & 81 & / & -9 \\ / & / & / & / & / & / & / \\ -9 & / & 81 & / & 81 & / & -9 \\ / & / & / & / & / & / & / \\ 1 & / & -9 & / & -9 & / & 1 \end{vmatrix}, \tag{3.14}$$

$$\begin{vmatrix} / & / & / & / & / & / & / \\ / & / & / & (1) & / & / & / \end{vmatrix} = \frac{1}{16} \begin{vmatrix} / & / & / & / & / & / & / \\ -1 & / & 9 & / & 9 & / & -1 \end{vmatrix}, \tag{3.15}$$

$$\begin{vmatrix} / & / \\ / & / \\ / & / \\ / & [1] \\ / & / \\ / & / \\ / & / \end{vmatrix} = \frac{1}{16} \begin{vmatrix} / & -1 \\ / & / \\ / & 9 \\ / & / \\ / & 9 \\ / & / \\ / & -1 \end{vmatrix}. \tag{3.16}$$

Les symboles / représentent un coefficient de 0.

Si on développe ces coefficients en prenant aussi en compte ceux qui existent mais qui ne sont pas représentés dans les limites du tableau 3.1, on trouve :

n-4	n-3	n-2	n-1	n	n+1	n+2	n+3	n+4	
1	1	$\frac{14}{16}$	1	2	/	$\frac{33}{8}$	/	4	2M
1	1	1	1	1	/	/	/	/	2M+1

Table 3.2

Si on fait maintenant la même chose pour les coins du rectangle, on peut écrire les poids comme :

n-2	n-1	n	n+1	n+2	
1	(1)	1	(1)	1	m+2
[1]	∗1∗	[1]	∗1∗	[1]	m+1
1	1	1	(1)	1	m
1	1	1	∗1∗	[1]	m-1
1	1	1	(1)	1	m-2

Table 3.3

et les développer de façon à trouver :

n-2	n-1	n	n+1	n+2	
$\frac{1057}{256}$	/	$\frac{131}{32}$	/	$\frac{1023}{256}$	m+2
/	/	/	/	/	m+1
$\frac{61}{32}$	1	$\frac{11}{4}$	/	$\frac{131}{32}$	m
1	1	1	/	/	m-1
$\frac{191}{256}$	1	$\frac{61}{32}$	/	$\frac{1057}{256}$	m-2

Table 3.4

En prenant en compte que $\Delta p_x(R)\Delta p_y(R) = \Delta p_x(R{+}1)\Delta p_y(R{+}1) \times 4$, on trouve pour le cas 2V les coefficients de C, utilisé dans Eq.(3.17), suivant :

n-2	n-1	n	n+1	n+2	
14	16	32	/	16,5	2M
16	16	16	/	/	2M+1

TABLE 3.5 – Valeur de $16 * C(j_x, j_y)$ sur un coté pour le cas 2V. Les points qui ne sont pas représentés et qui ne sont pas affectés par les autres interfaces entre les grilles ont une valeur $C(j_x, j_y) = 1$.

n-2	n-1	n	n+1	n+2	
264,25	/	262	/	255,75	m+2
/	/	/	/	/	m+1
488	256	704	/	262	m
256	256	256	/	/	m-1
191	256	488	/	264,25	m-2

TABLE 3.6 – Valeur de $256 * C(j_x, j_y)$ sur un coin pour le cas 2V. Les points qui ne sont pas représentés et qui ne sont pas affectés par les autres interfaces entre les grilles ont une valeur $C(j_x, j_y) = 1$.

Pour les points qui ne sont pas indiqués dans les tableaux 3.5 et 3.6 la valeur de C est de 1.

Chaque intégrale va donc pouvoir s'écrire comme la fonction à intégrer multipliée par un coefficient de poids C tel que :

$$\begin{aligned} I(x,y) = \sum_{R_n=0}^{R_n^{max}-1} \Bigg[\sum_{j_x,j_y \in S_{R_n} \setminus S_{R_{n+1}}} F_{R_n}(j_x,j_y) C(j_x,j_y) \Delta p_x(R) \Delta p_y(R) \Bigg] \\ + \sum_{j_x,j_y \in S_{R_n^{max}}} F_{R_n^{max}}(j_x,j_y) C(j_x,j_y) \Delta p_x(R_n^{max}) \Delta p_y(R_n^{max}), \end{aligned} \tag{3.17}$$

avec $S_{R_n} \setminus S_{R_{n+1}}$ l'ensemble des points de la grille R qui ne sont pas définis sur la grille $R+1$. On peut les représenter par les points noirs sur la figure 3.5.

Si on applique cela aux courants, on obtient :

$$J_\alpha^{VLEM_EG}(x,y) = \int e\frac{p_\alpha}{m\gamma(\mathbf{p})} f(x,y,\mathbf{p})\mathbf{dp} = \sum_{R_n=0}^{R_n^{max}-1} \left[\sum_{j_x,j_y \in S_{R_n}\setminus S_{R_{n+1}}} e\frac{p_\alpha(j_\alpha)}{m\gamma(j_x,j_y)} f_{R_n}(j_x,j_y) C(j_x,j_y)\Delta p_x(R)\Delta p_y(R) \right] + \sum_{j_x,j_y \in S_{R_n^{max}}} e\frac{p_\alpha(j_\alpha)}{m\gamma(j_x,j_y)} f_{R_n^{max}}(j_x,j_y) C(j_x,j_y)\Delta p_x(R_n^{max})\Delta p_y(R_n^{max}). \quad (3.18)$$

Les coefficients pour les versions 3V du code VLEM_EG sont donnés et explicités en annexe E.

Toutes les intégrales sur les composantes des quantités de mouvement faites avec VLEM_EG utilisent cette méthode. Ceci comprend aussi tous les invariants de Casimir (donc les moments de la fonction de distribution) et l'énergie cinétique.

On remarque aussi que le choix de 3 points avec la grille graduée est compatible avec cette méthode puisque la modification atteint seulement les 3 points les plus proches du bord de la grille.

3.4 Validation de la méthode

Cette section est directement tirée de l'article de présentation de VLEM_EG [47].

Afin de vérifier l'implémentation du code et de tester la méthode, il est nécessaire d'en comparer les résultats à ceux obtenus avec le code d'origine. Dans cette section on va tester l'accélération et la précision du code pour VLEM2D2V et VLEM2D3V avec et sans raffinement par grilles emboîtées. Le choix d'expériences numériques s'est porté sur les instabilités de type Weibel, déjà introduites dans le sous-chapitre 1.6, dont on va rappeler quelques notions.

Les études linéaires des instabilités de type Weibel ont permis d'identifier plusieurs classes d'instabilité électronique faisceau-plasma :

- L'instabilité de filamentation de courant (CFI) [68, 117], due à une anisotropie des courants avec le vecteur d'onde $\boldsymbol{k}$ perpendiculaire à la direction des faisceaux.
- L'instabilité électrostatique double faisceaux (TSI), avec $\boldsymbol{k}$ parallèle à la direction des faisceaux.
- L'instabilité oblique (OI) [72, 73], avec une direction arbitraire de $\boldsymbol{k}$.

La CFI et l'OI peuvent amener à l'émergence d'une cascade directe vers les grands nombres d'ondes lors des processus turbulents, c'est-à-dire qu'il va y avoir un transfert de l'énergie des grandes échelles vers les petites échelles. C'est pourquoi, ces instabilités sont un sujet idéal pour l'étude de ces processus et pour montrer l'efficacité de la méthode AMR.

Nous allons donc regarder le problème de deux faisceaux d'électrons de sens opposés pour la CFI et l'OI, en prenant en compte la possibilité d'une TSI électrostatique. Nous examinerons la transition entre l'OI et la CFI d'abord pour un cas peu relativiste avec le code 2D2V puis le code 2D3V et ensuite pour un cas antisymétrique relativiste afin de montrer que le code est adapté même dans le cas d'une dissymétrie de température entre les faisceaux. Les trois simulations peuvent être schématisées par la figure 3.7 en adaptant la température des faisceaux aux problèmes décrits dans les sous-parties suivantes : 3.4.1, 3.4.2 et 3.4.3.

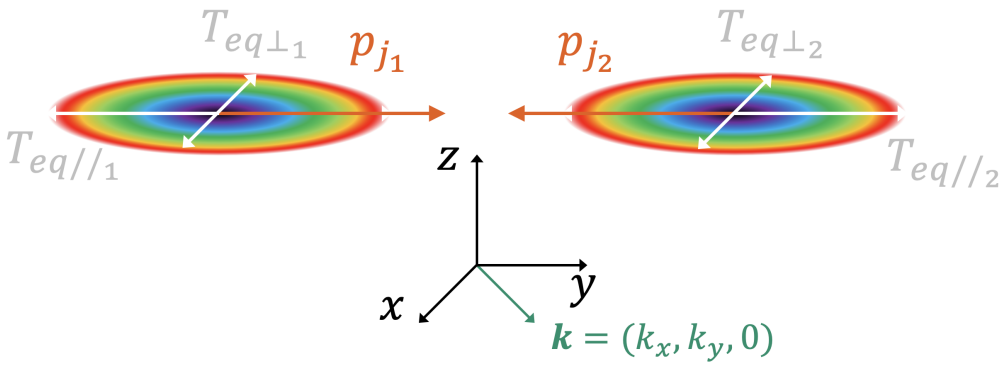

FIGURE 3.7 – Schéma de la configuration initiale avec deux faisceaux contra-propageants dans la direction y. La perturbation linéaire a un vecteur d'onde $\boldsymbol{k}$ dans le plan x, y. Pour les sous-parties 3.4.1 et 3.4.2 on a $T_{eq\perp_1} = T_{eq//_1} = T_{eq\perp_2} = T_{eq//_2} = 6keV/k_B$. Pour la sous-partie 3.4.3 on a $T_{eq\perp_1} = T_{eq//_1} = 2MeV/k_B$ et $T_{eq\perp_2} = T_{eq//_2} = 5keV/k_B$. Les sous-parties 3.4.1 et 3.4.2 seront décrites par une double Maxwellienne tandis que la sous-partie 3.4.3 qui est beaucoup plus relativiste sera décrite par deux distributions de Maxwell-Jüttner.

3.4.1 Transition de OI à CFI avec VLEM2D2V

Performances

On étudie le problème de deux faisceaux d'électrons contra-propageants dans la direction des y comme représenté sur la figure 3.7. On choisit de décrire ce phénomène avec une fonction de distribution de Maxwell-Botzmann non relativiste et que l'on peut écrire comme $f(x, y, \mathbf{p}, t = 0) = \sum_{j=1,2} n_j F_{0j}(\mathbf{p})$ avec :

$$F_{0j} = \frac{1}{2\pi m^2 c^2 \beta_{th_j}^2} \exp\left(-\frac{p_x^2 + (p_y - p_j)^2}{2m^2 c^2 \beta_{th_j}^2}\right), \tag{3.19}$$

où $\beta_{th_j} = \sqrt{k_B T_{eq_j}/mc^2}$ est la vitesse thermique normalisée et p_j la quantité de mouvement moyenne normalisée du faisceau (flèche orange de la figure 3.7).

Dans la situation de Eq.(3.19), en fonction de la direction du vecteur d'onde **k** dans le plan (x,y), la nature de l'instabilité va être modifiée amenant soit à l'instabilité double faisceaux ($\mathbf{k} = (0, k_y)$), soit à l'instabilité de filamentation de courant ($\mathbf{k} = (k_x, 0)$) ou plus généralement à l'instabilité "Weibel" oblique ($\mathbf{k} = (k_x, k_y)$). Dans [74], les auteurs ont résolus la relation de dispersion et ont trouvé deux branches différentes pour l'instabilité oblique, les branches "haute" ("high") et "basse" ("low") fréquence. On a repris la figure 2 de [74] dans la figure 3.8 afin de montrer ces deux branches et les taux de croissances associés. Nous allons nous concentrer ici sur la branche "basse" fréquence déjà étudiée dans le cas B de [29]. Cette branche est propagative et pour $k_B T_{eq_1} = k_B T_{eq_2} = 6keV$, $p_1 = -p_2 = 0,9mc$ et $n_1 = n_2 = \frac{n_0}{2}$ elle peut être déclenchée en excitant le mode dominant $k_x d_e \approx 3,07$ et $k_y d_e \approx 5$ représenté par un cercle blanc sur la figure 3.8 (avec $d_e = c\omega_p^{-1}$, l'épaisseur de peau). Ce qui fixe la dimension normalisée de la boîte à $L_x = 4,0879 d_e = \frac{2\pi}{\Delta k_x}$ et $L_y = 2,513274 d_e = \frac{2\pi}{\Delta k_y}$, ce qui correspond au mode de Fourier $(n_x, n_y) = (2, 2)$. Afin d'exciter ce mode, nous allons injecter une perturbation initiale sur le champ magnétique sous la forme :

$$\delta B_z = \delta B_0 \sin(n_x \Delta k_x x + n_y \Delta k_y y). \tag{3.20}$$

On choisit $e\delta B_0/(m\omega_p) = 0,005$ comme perturbation initiale normalisée, $\Delta t\, \omega_p = 0,004$ pour le pas de temps avec 60 000 itérations, le nombre de points de grille en espace est $N_x N_y = 128 \times 128$ distribués sur 64 processus MPI avec 8 processus OpenMP par processus MPI, ce qui fait un total de 512 processus.

Comme on a réalisé la même expérience numérique que dans le cas B de l'article de Ghizzo et al. [29] on va pouvoir comparer nos résultats à cet article et même aller plus loin dans le temps. Cette simulation va conduire à une transition de l'OI à la CFI. Ces deux instabilités vont permettre la création de structures magnétiques amenant la fonction de distribution à s'étaler dans l'espace des quantités de mouvement. C'est pourquoi une méthode de raffinement de maillage semble nécessaire.

Nous allons regarder quatre cas différents avec VLEM_EG pour 2 valeurs de ϵ ($\epsilon = 10^{-5}$ et $\epsilon = 10^{-3}$) (Eqs.(3.3-3.4)) :

- **Cas 1**, sans raffinement avec une taille de boîte sur les quantités de mouvement de $\pm 3mc$ et avec 127 points répartis uniformément dans chaque direction des quantités de mouvement.
- **Cas 2**, avec un niveau de raffinement et la même taille de boîte que le **Cas 1** pour la grille R=0 composées de 64 points dans chaque directions, ce qui amène à un nombre maximum de 127 points dans chaque direction pour R=1.
- **Cas 3**, on ajoute un autre niveau de raffinement au **Cas 2**.
- **Cas 4**, on garde deux niveaux de raffinement mais on augmente la taille de la boîte jusqu'à $\pm 6mc$ pour les quantités de mouvement. On obtient donc la même précision que le **Cas 2** mais on va être en capacité de mieux suivre les pertes aux bords de la fonction de distribution due à la taille finie de la boite.

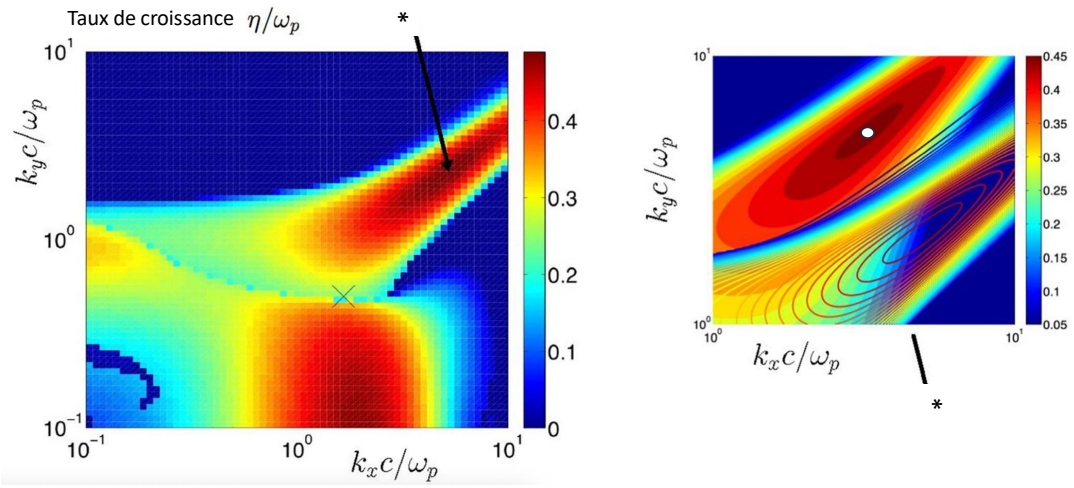

FIGURE 3.8 – Figure issue de [74] modifiée afin d'adapter la forme et la langue. Dans le panneau de gauche, le tracé des solutions avec les taux de croissance maximaux de la relation de dispersion cinétique qui est donné par (A1 dans [74]). Les deux classes principales de modes instables peuvent être reconnues : les modes TSI sont exactement longitudinaux ($k_x = 0$), tandis que les modes CFI sont purement perpendiculaires ($k_y = 0$) par rapport à la direction des faisceaux (ici le long de y). Si des perturbations à la fois parallèles et perpendiculaires à la propagation des faisceaux deviennent instables, l'instabilité est alors qualifiée d'oblique : cette solution se présente sous la forme d'un lobe dans une direction inclinée par rapport à l'axe k_x. La croix dans le panneau de gauche indique une région où les instabilités CFI et oblique peuvent être excitées simultanément. Dans le panneau de droite, un zoom plus détaillé est tracé dans l'intervalle $1 \leq d_e k_x \leq 10$ et $1 \leq d_e k_y \leq 10$, dans lequel des solutions propagatives et non propagatives de la relation de dispersion cinétique complète ont été trouvées. Un deuxième lobe, correspondant à des modes basse fréquence ayant un taux de croissance comparable aux autres, est clairement visible. Le lobe « haute fréquence » est ici superposé à l'aide d'une représentation en courbes de niveau. Les paramètres physiques correspondent à une température électronique de $k_B T_x = k_B T y = 6keV$ et à une quantité de mouvement moyenne normalisée $p_1 = -p_2 = 0,9mc$ avec une densité de faisceau de $n1 = n2 = n0/2$. Le cercle blanc représente le mode qui va être excité dans la sous-partie 3.4.1.

On résume tous ces paramètres dans le tableau 3.7.

On va comparer ces simulations avec VLEM dans 3 cas qui sont indiqués dans le tableau 3.8. Le premier cas (**Cas A**) a une taille de boîte pour les quantités de mouvement de $\pm 3mc$ et avec 127 points dans chaque direction. Le deuxième cas (**Cas B**) a la même taille de boîte avec 253 points dans chaque direction des quantités de mouvement. Le dernier (**Cas C**) avec 253 points dans chaque direction et avec une taille de boîte pour les quantités de mouvement de $\pm 6mc$.

VLEM_EG	**Cas 1**	**Cas 2**	**Cas 3**	**Cas 4**
R^{max}	0	1	2	2
Bords pour R=0 (en mc)	± 3	± 3	± 3	± 6
$\Delta p_{x/y}\,(R^{max})$ (en mc)	$4,76 \times 10^{-2}$	$4,76 \times 10^{-2}$	$2,38 \times 10^{-2}$	$4,76 \times 10^{-2}$

TABLE 3.7 – Différents cas étudiés avec VLEM2D2V_EG.

VLEM	**Cas A**	**Cas B**	**Cas C**
Bords ($-p_{x/y}/p_{x/y}$) (en mc)	± 3	± 3	± 6
$\Delta p_{x/y}$ (en mc)	$4,76 \times 10^{-2}$	$2,38 \times 10^{-2}$	$4,76 \times 10^{-2}$

TABLE 3.8 – Différents cas étudiés avec VLEM2D2V.

Les **Cas 1** et **2** de VLEM_EG seront comparés au **Cas A** de VLEM, le **Cas 3** avec le **Cas B** et le **Cas 4** avec le **Cas C**, cela permet de comparer des cas ayant la même précision. Le choix du nombre de points dans chaque direction vient du fait que pour les **Cas 2**, **3** et **4** le nombre de points en $R = 0$ est 64×64 qui est un multiple de 8, ce qui correspond bien au nombre de taches OpenMP.

Les résultats des temps de calcul de chaque simulation sont présentés dans le tableau 3.9 pour $\epsilon = 10^{-5}$ et dans le tableau 3.10 pour $\epsilon = 10^{-3}$ dans les Eqs.(3.3-3.4). On a aussi rajouté le temps théorique (τ_t) qui correspond au temps de calcul que mettrait le code par grille emboîtée si ce temps de calcul était proportionnel au nombre de points par rapport au code VLEM de référence. On peut déterminer ce temps théorique en utilisant le nombre moyen de points dans l'espace des quantités de mouvement dans VLEM_EG ($N_{p_{EG}}$), le nombre dans l'espace des quantités de mouvement dans VLEM ($N_{p_{VLEM}}$) et le temps de calcul de ce dernier (τ_{VLEM}) comme :

$$\tau_t = \frac{N_{p_{EG}}}{N_{p_{VLEM}}} \tau_{VLEM}. \tag{3.21}$$

Les résultats des temps de calcul pour les cas comparatifs avec VLEM sont présentés dans le tableau 3.11. On a tracé l'ensemble de ces résultats dans la figure 3.9 afin d'avoir une représentation plus visuelle.

Pour les deux valeurs de ϵ, on peut observer que le **Cas 1** donne une légère augmentation du temps de calcul d'environ 5% en comparaison du **Cas A**. Cela montre que les modifications apportées par la méthode AMR au code VLEM, qui ne sont pas liées à la création de grille, ont été bien implémentées. On observe cependant une légère augmentation de l'effort numérique due à l'implémentation de la méthode AMR.

	Cas 1	**Cas 2**	**Cas 3**	**Cas 4**
Temps de calcul (heure)	3,97	3,32	9,40	5,23
Temps théorique de calcul (heure)	3,77	2,92	8,17	4,83
Nombre moyen de points des quantités de mouvement	16129	12500	34008	19920
Temps de calcul moyen par point par itération (en 10^{-10}s)	9,00	9,72	10,13	9,63

TABLE 3.9 – Informations sur les temps de calcul pour les différents cas explicités dans le tableau 3.7 pour VLEM_EG avec $\epsilon = 10^{-5}$.

	Cas 1	**Cas 2**	**Cas 3**	**Cas 4**
Temps de calcul (heure)	3,97	2,88	6,68	3,63
Temps théorique de calcul (heure)	3,77	2,40	6,17	3,38
Nombre moyen de points des quantités de mouvement	16129	10282	25668	13954
Temps de calcul moyen par point par itération (en 10^{-10}s)	9,00	10,26	9,52	9,56

TABLE 3.10 – Informations sur les temps de calcul pour les différents cas explicités dans le tableau 3.7 pour VLEM_EG avec $\epsilon = 10^{-3}$.

	Cas A	**Cas B**	**Cas C**
Temps de calcul (heure)	3,77	15,38	15,51
Nombre de points des quantités de mouvement	16129	64009	64009
Temps de calcul moyen par point par itération (en 10^{-10}s)	8,56	8,80	9,07

TABLE 3.11 – Informations sur les temps de calcul pour les différents cas explicités dans le tableau 3.8 pour VLEM.

Pour les trois autres Cas on observe que les deux valeurs de ϵ vont amener à une accélération (ou une diminution) du temps de calcul. Le **Cas 2** montre une faible accélération de 12% pour $\epsilon = 10^{-5}$ et de 24% pour $\epsilon = 10^{-3}$ en comparant avec le **Cas A**. Cela est dû à la faible précision de la grille R=0 dès le début de la simulation et donc à la nécessité d'avoir une grille raffinée presque dans tout l'espace des quantités de mouvement.

Pour le **Cas 3** et encore plus pour le **Cas 4**, le gain en temps de calcul devient beaucoup plus important. Il est respectivement de 39% et 66% pour $\epsilon = 10^{-5}$ et de 57% et 76% pour $\epsilon = 10^{-3}$. Les résultats pour le **Cas 3** montrent qu'une précision importante

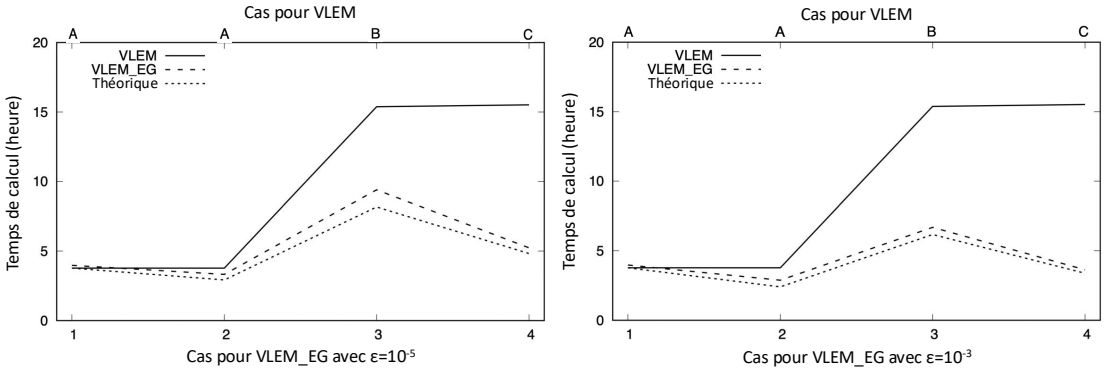

FIGURE 3.9 – Comparaison des temps de calcul pour différents cas explicités dans les tableaux 3.7-3.8 pour VLEM, VLEM_EG et le temps théorique avec $\epsilon = 10^{-5}$ (gauche) et $\epsilon = 10^{-3}$ (droite). Les valeurs tracées sont présentées dans les tableaux 3.9, 3.10 et 3.11.

est nécessaire principalement au centre de la grille des quantités de mouvement (voir Fig.3.12). Cela donne une bonne réduction du temps de calcul en comparaison de VLEM qui a besoin de garder la même précision même là où ce n'est pas nécessaire. Pour le **Cas 4**, le temps de calcul de VLEM_EG est plus grand que pour le **Cas 2** d'environ 50%, et ce même si la précision de la grille en quantité de mouvement est la même. Cela est principalement dû aux limites de la boîte qui sont trop rapprochées de la fonction de distribution pour le **Cas 2**. Cela sera rediscuté plus tard lorsque l'on décrira l'évolution de la fonction de distribution dans l'espace des quantités de mouvement à l'aide de la figure 3.13.

Les tableaux 3.9, 3.10 et 3.11 montrent aussi le temps de calcul moyen par point de l'espace des phases (i_x,i_y,j_x,j_y) par itération (τ_N). Cette valeur est calculée en ignorant les points identiques entre les grilles. On peut formuler son calcul comme :

$$\tau_N = \frac{\tau}{N_x\, N_y\, N_p\, n_{max}}, \tag{3.22}$$

avec τ le temps écoulé moyen de toutes les taches OpenMP de la simulation avec VLEM (aussi appelé $T_{\text{Écoulé}}$), n_{max} le nombre d'itérations et N_p le nombre moyen de points sur l'espace des quantités de mouvement avec VLEM_EG. Ce dernier est calculé dans le code et ne prend en compte chaque point qu'une seule fois, même s'ils sont définis sur plusieurs grilles.

Plus les résultats pour τ_N avec le code VLEM_EG seront proches de ceux de VLEM et plus l'implémentation sera bonne. En regardant pour les **Cas 2**, **3** et **4**, on trouve une valeur moyenne de ce paramètre de $9,60 \times 10^{-10} s$ contre $8,81 \times 10^{-10} s$ pour VLEM. Cette différence est d'environ 9%, ce qui est assez faible et montre que l'ensemble des modifications apportées par VLEM_EG au code VLEM (voir entre autre algorithme 2)

est bien implémenté et peu coûteux en terme de temps de calcul.

Afin de vérifier les résultats de VLEM_EG, on va comparer la fonction de distribution sur (x, p_x) (Fig.3.12) ainsi que l'évolution spatiale de B_z (Fig.3.11) et E_x (Fig.3.10) entre VLEM et VLEM_EG avec $\epsilon = 10^{-5}$. On choisit de représenter le **Cas 3** et le **Cas B** car ce sont les cas avec la plus grande précision. On vérifie facilement sur les figures 3.10-3.12 que les deux simulations nous donnent les mêmes résultats pour la phase de transition (gauche), la phase linéaire (milieu) et la saturation (droite). On retrouve aussi sur la figure 3.12 les structures de piégeages électrostatique et magnétique qui ont été observées dans [29]. Comme on est allé plus loin sur les temps de simulation, on peut observer un état stationnaire qui est atteint à partir $t\omega_p \simeq 122$ et qui continue jusqu'à la fin de la simulation. Sur les images de droite, à $t\omega_p \simeq 86$, des figures 3.10 et 3.11, , on observe que l'état stationnaire est déjà atteint pour les champs. Cependant, pour que la fonction de distribution entre également dans cet état, il faut attendre $t\omega_p \simeq 122$ (voir figure 3.12 à droite à $t\omega_p \simeq 134$). Dans cet état, les électrons vont être piégés dans les régions de faible champ magnétique ($B_z \approx 0$) en raison de la force pondéromotrice qui les repousse hors des régions de fort champ magnétique. Cette localisation des électrons va générer un champ électrique oscillant deux fois plus en espace que le champ magnétique. Cette évolution est typique de la CFI et a déjà été discutée pour le cas non-relativiste dans [67].

Dans d'autres simulations test, on a pu observer que l'accélération en temps de calcul CPU à ϵ fixé est de plus en plus importante lorsqu'on augmente le nombre maximum de points de grille. C'est logique puisque, pour une fonction plutôt lisse, plus le nombre de points de grille sera important sur la grille R-1, plus la précision sera élevée pour l'estimation des points de grille en R, et donc plus la grille en R sera petite.

Impact de la filamentation

L'un des points importants dans le développement d'une méthode AMR est le choix du critère de raffinement. Ce critère doit être choisi afin de décrire avec soin le procédé physique que l'on observe. Dans les instabilités obliques, il existe plusieurs modes, comme ceux de la branche "haute" fréquence, qui vont croître et amener à une cascade directe vers les petites échelles. Dans les articles [48, 118], il a été montré que ces procédés peuvent amener à un renversement du transfert d'énergie.

De plus, la CFI va renforcer la production de filaments de plus en plus fins dans l'espace des phases qui vont se coupler avec la filamentation de la fonction de distribution déjà abordée dans la sous-partie 1.4.4. Ce processus a été appelé filamentation renforcée ("reinforced filamentation"). Ce mécanisme va induire de profondes modifications dans le transfert d'énergie et va amener à un fort chauffage stochastique.

Ici, le critère de raffinement défini dans les équations (3.3) et (3.4), permet de prendre en compte ce type de processus physique, surtout si la taille de la cellule dans l'espace des quantités de mouvement peut parfaitement suivre la filamentation. C'est d'autant

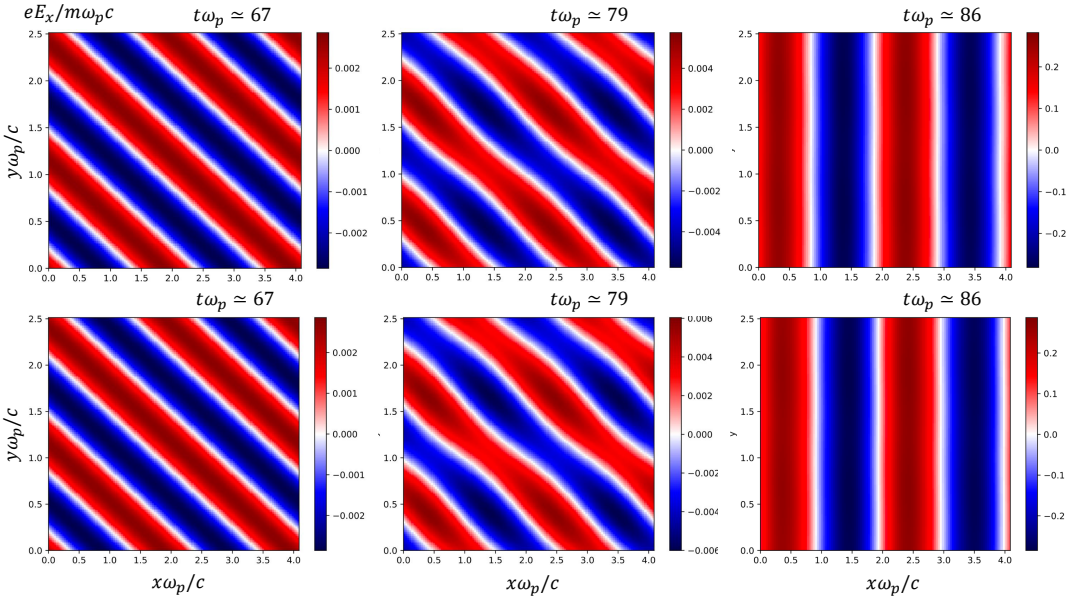

FIGURE 3.10 – Tracés comparatifs de l'amplitude normalisée de E_x pour le **Cas 3** (haut) avec $\epsilon = 10^{-5}$ et le **Cas B** (bas) à $t\omega_p \simeq 67$, 79 et 86. La transition entre OI et CFI est décrite de la même manière dans les deux codes.

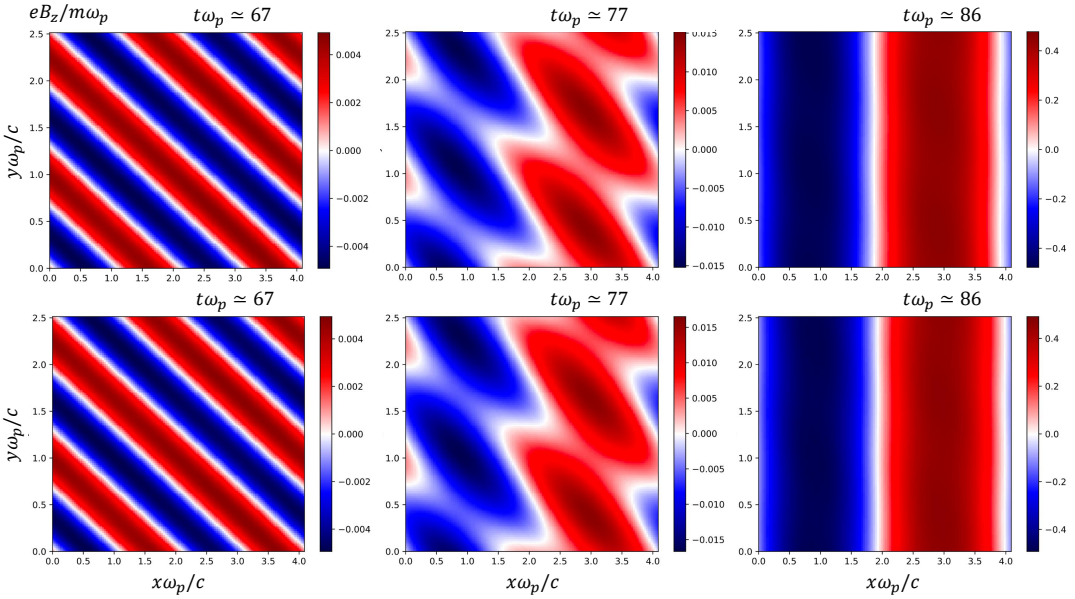

FIGURE 3.11 – Tracés comparatifs de l'amplitude normalisée de B_z pour le **Cas 3** (haut) avec $\epsilon = 10^{-5}$ et le **Cas B** (bas) à $t\omega_p \simeq 67$, 77 et 86. La transition entre OI et CFI est décrite de la même manière dans les deux codes.

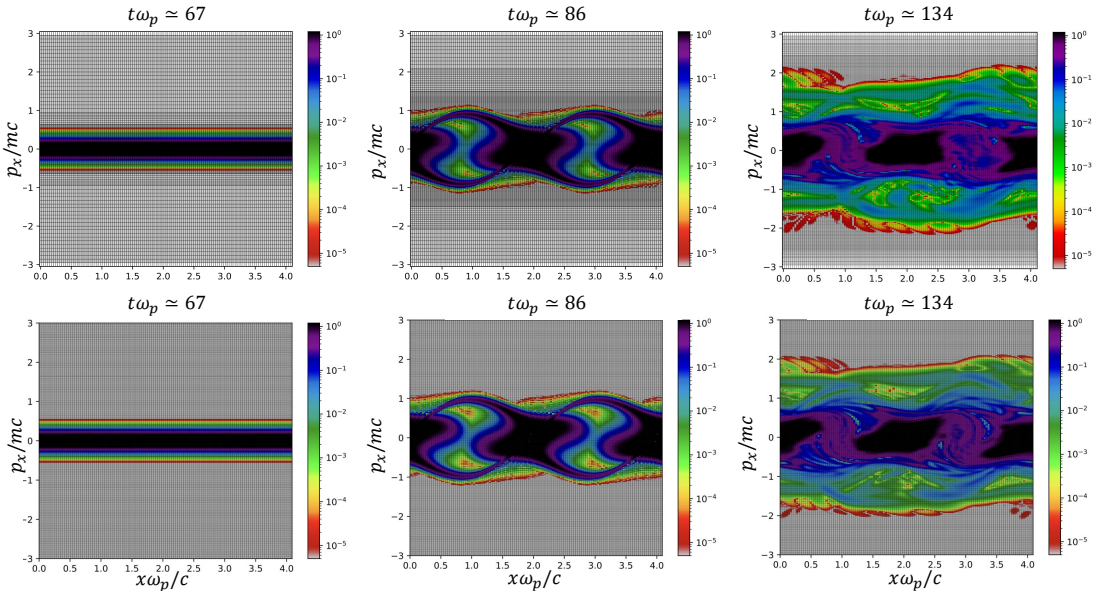

FIGURE 3.12 – Représentations comparatives de la fonction de distribution dans l'espace des phases $x - p_x$, moyennée suivant p_y, pour le **Cas 3** avec $\epsilon = 10^{-5}$ (haut) et le **Cas B** (bas) à $t\omega_p \simeq 67$, 86 et 134 pour un point situé en ($i_y = N_y/2$). Pour le **Cas 3** avec VLEM_EG, la grille est représentée et suit bien la structure.

plus le cas lorsque ϵ est petit comme pour $\epsilon = 10^{-5}$. La technique AMR, restreinte à l'espace des quantités de mouvement est donc parfaitement adaptée pour suivre la filamentation renforcée.

A noter que ce type de processus est présent dans la figure 3.13 qui représente l'évolution de la fonction de distribution dans l'espace des quantités de mouvement (p_x, p_y). Les conditions initiales choisies correspondent à l'excitation d'un mode localisé sur la branche "basse" fréquence, minimisant ce type de processus qui persiste quand même.

En effet, si on observe la figure 3.13 qui représente l'évolution de la grille pour le Cas 4 pour $\epsilon = 10^{-5}$, on observe le "chauffage" de la fonction de distribution qui va occuper plus de place dans l'espace des quantités de mouvement. Sur l'image en bas à droite, à $t\omega_p = 134$, la grille interne est plus large (environ $p_{\alpha_{min/max}}(R{=}2) = \pm 4mc$) que le maximum de la grille pour le Cas 2 ($p_{\alpha_{min/max}}(R{=}0) = \pm 3mc$). Cela montre que la taille de la boîte en quantité de mouvement doit être augmentée pour le **Cas 2** à la fois pour avoir une meilleure interpolation mais aussi pour suivre certains éléments de la queue de la fonction de distribution qui pourront être perdus. C'est le type de scénario où le raffinement de maillage est bien adapté. On note que même si de l'information va être perdue dû à la taille de boîte restreinte pour les **Cas 1**, **2** et **3**, les simulations donnent toujours des résultats corrects puisque l'impact de ces pertes va être minime.

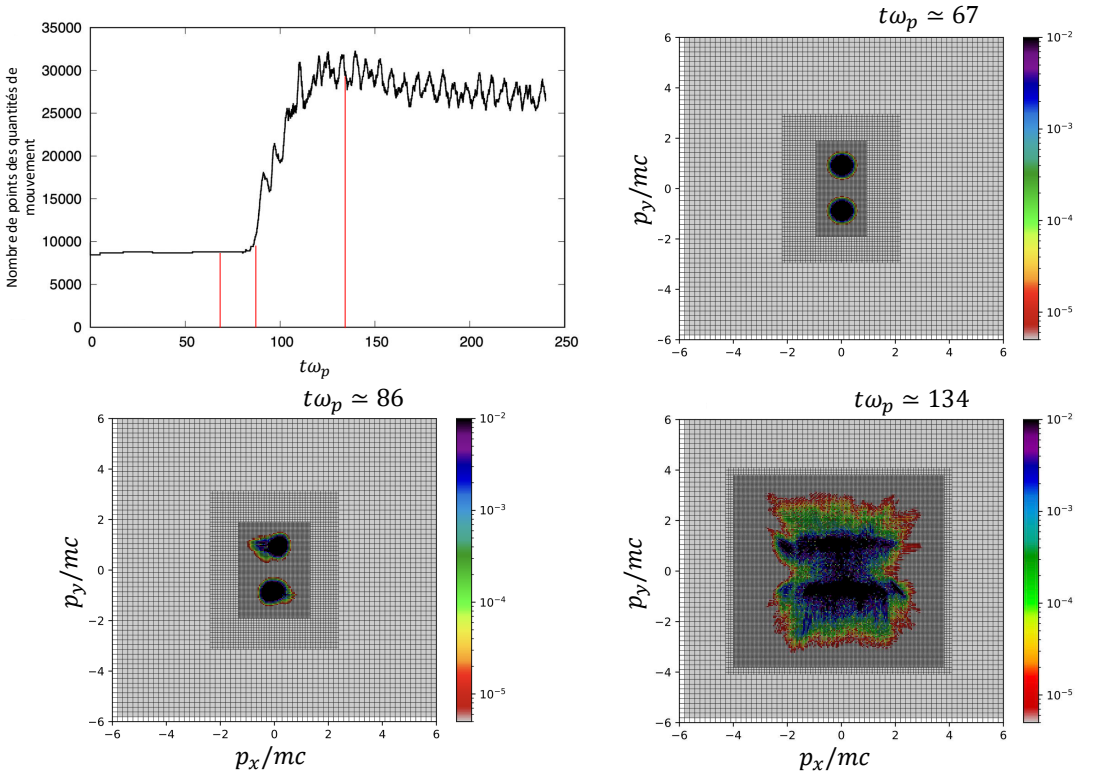

FIGURE 3.13 – (haut-gauche) Évolution temporelle du nombre de points dans l'espace des quantités de mouvement sur la grille la plus fine $R = R_n^{max}$ pour le **Cas 4** avec $\epsilon = 10^{-5}$. Les lignes rouges verticales montrent les différents temps auxquels sont représentées les autres figures. Représentation de la fonction de distribution dans l'espace des phases $p_x - p_y$ pour le **Cas 4** avec $\epsilon = 10^{-5}$ aux temps $t\omega_p \simeq 67$ (haut-droite), 86 (bas-gauche) et 134 (bas-droite) pour un point localisé en $(i_x = N_x/2, i_y = N_y/2)$. Les grilles emboîtées sont représentées et suivent bien l'évolution de la fonction de distribution.

La figure 3.13 montre aussi que l'AMR est bien adaptée pour suivre la phase transitoire, ici localisée à $t\omega_p \leq 75$. En effet, pendant cette phase les champs électrique et magnétique augmentent doucement en amenant de légères perturbations sur la fonction de distribution. La grille raffinée va donc nécessiter moins de points, ce qui va conduire à réduire le coût numérique de la simulation.

La figure 3.14 montre les mêmes résultats que la figure 3.13 mais pour une valeur de epsilon plus grande ($\epsilon = 10^{-3}$). Afin de mieux comparer entre les deux valeurs de ϵ, on a repris sur la figure 3.15 les deux fonctions de distribution suivant $p_x - p_y$ au temps $t\omega_p \simeq 134$. On peut voir que le nombre de point des quantités de mouvement est toujours plus petit avec $\epsilon = 10^{-3}$ que pour le cas avec $\epsilon = 10^{-5}$ ce qui justifie que le

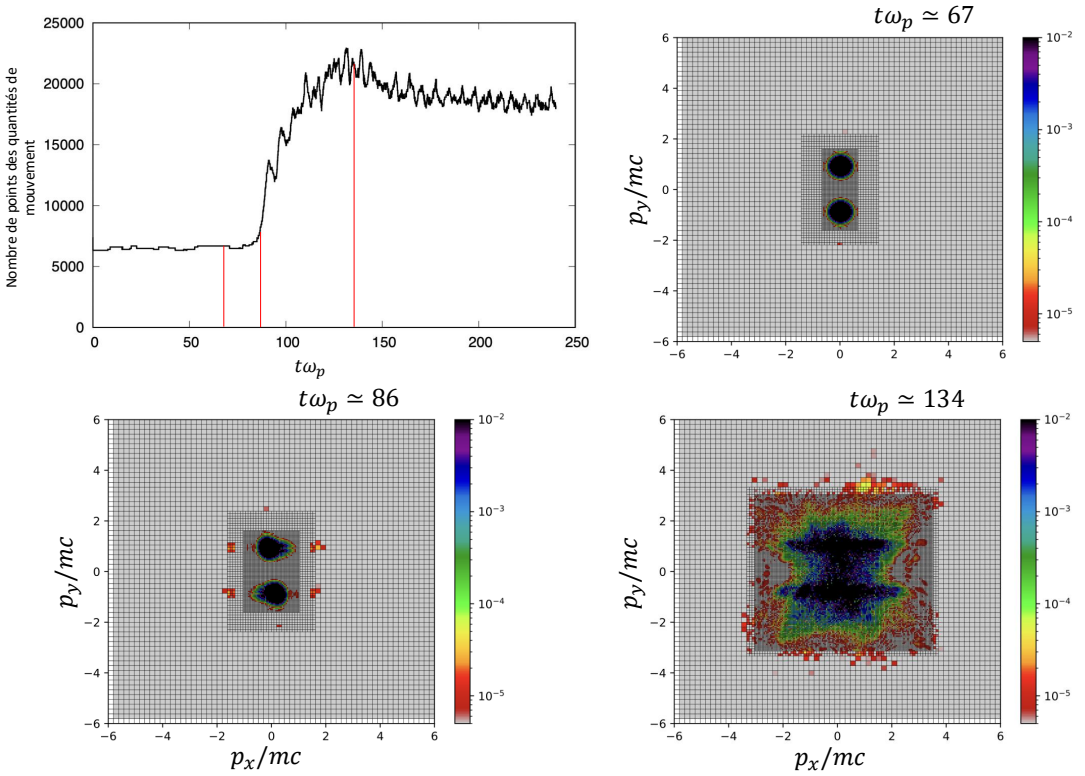

FIGURE 3.14 – (haut-gauche) Évolution temporelle du nombre de points dans l'espace des quantités de mouvement sur la grille la plus fine $R = R_n^{max}$ pour le **Cas 4** avec $\epsilon = 10^{-3}$. Les lignes rouges verticales montrent les différents temps auxquels sont représentées les autres figures. Représentation de la fonction de distribution dans l'espace des phases $p_x - p_y$ pour le **Cas 4** avec $\epsilon = 10^{-3}$ aux temps $t\omega_p \simeq 67$ (haut-droite), 86 (bas-gauche) et 134 (bas-droite) pour un point localisé en $(i_x = N_x/2, i_y = N_y/2)$. Les grilles emboîtées sont représentées et on observe l'apparition d'erreurs sur les bords due à la trop grande valeur de ϵ.

temps de calcul est plus court. Cependant, le faible nombre de points entre deux grilles, que l'on a fixé à 3 pour la grille graduée, et la proximité entre la grille et la fonction de distribution vont amener à une erreur jusqu'à un ordre de 5×10^{-5} dans plusieurs cellules (voir figure 3.15). Ces erreurs sont directement liées au choix de $\epsilon = 10^{-3}$ et à la méthode de grille emboîtée, cela est discuté dans l'annexe D. C'est pourquoi nous choisirons un paramètre $\epsilon = 10^{-5}$ dans la prochaine simulation.

A noter que l'erreur est visible puisqu'on a choisi de tracer la fonction de distribution jusqu'à une intensité de 5×10^{-6}. Mais si on choisit de la tracer jusqu'à des valeurs beaucoup plus faible, on verra l'apparition d'erreur aussi dans la grille $\epsilon = 10^{-5}$. Cependant, on est sûr que pour cette valeur de ϵ la conservation de la densité ne sera

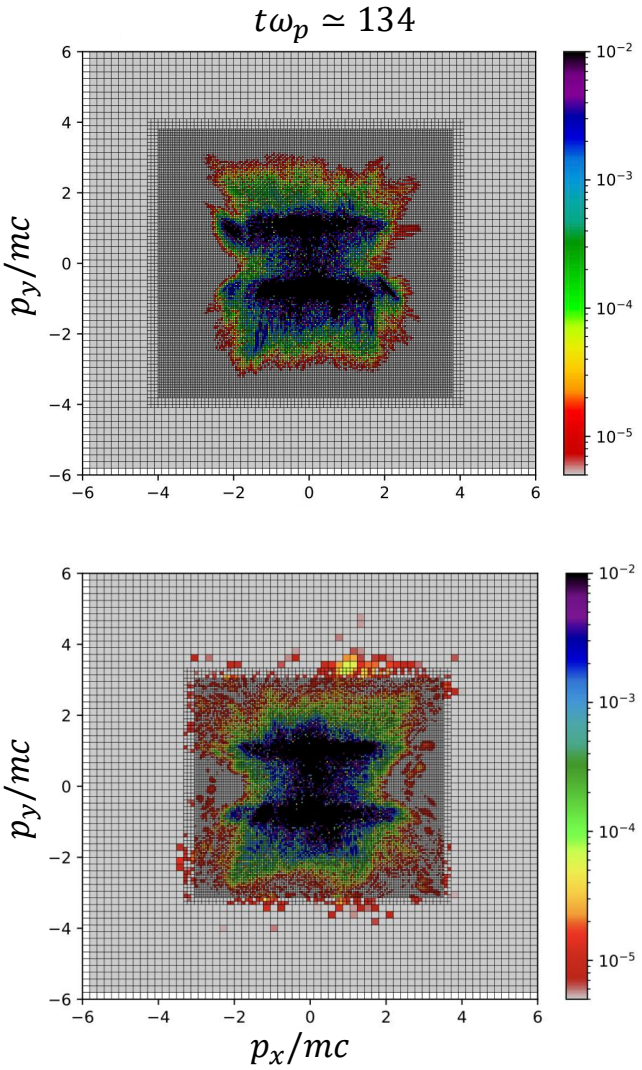

FIGURE 3.15 – Représentation de la fonction de distribution dans l'espace des phases $p_x - p_y$ pour le **Cas 4** au temps $t\omega_p \simeq 134$ pour un point localisé en $(i_x = N_x/2, i_y = N_y/2)$. En haut, $\epsilon = 10^{-5}$ et en bas avec $\epsilon = 10^{-3}$. On observe la création d'erreur allant jusqu'à des valeurs de 5×10^{-5} dans le cas $\epsilon = 10^{-3}$. Pour $\epsilon = 10^{-5}$, on se rend compte de la nécessité d'avoir une grille aussi raffinée pour décrire au mieux les fluctuations de la fonction de distribution.

pas atteinte. En effet, on pourra vérifier cela en plus de la conservation des différentes énergies dans la sous-partie suivante 3.4.2.

Le choix de la valeur ϵ va donc être un compromis entre la précision et le temps de calcul. Afin d'éviter de voir l'apparition d'erreurs, une valeur de $\epsilon = 10^{-5}$ est suffisante,

mais il est possible de diminuer cette valeur si l'on souhaite augmenter la précision (voir sous-partie 3.4.3).

3.4.2 Transition de OI à CFI avec VLEM2D3V

Afin de valider aussi la version à 3 dimensions de quantité de mouvement, on va regarder la même instabilité que la sous-partie précédente mais avec VLEM2D3V et VLEM_EG2D3V. Rajouter une troisième dimension donne la possibilité d'avoir des densités de courant dans la direction z (J_z) et donc d'avoir les champs électromagnétiques associés : E_z, B_x and B_y. La reconnexion magnétique est désormais possible dans le plan $x - y$.

Les simulations qui vont être comparées auront une résolution dans l'espace des phases plus petite que précédemment dans le but de réduire le coup numérique. On choisit donc $\Delta t\omega_p = 0,008$ avec 24000 pas de temps sur 8 intervalles de temps, chacun après 3000 pas de temps, $N_x N_y = 64 \times 64$ sur 128 processus en ayant choisi 8 taches OpenMP par processus MPI. Le choix de séparer en intervalle de temps est dû à une contrainte du centre de calcul sur la durée des simulations et nécessite de relancer la simulation à partir d'une sauvegarde. Il y a donc eu 7 "restarts" pour effectuer cette simulation. Cette partie a nécessité d'adapter la technique de sauvegarde pour relancer les simulations afin qu'elle prenne en compte les différents niveaux de grilles de la fonction de distribution. La sauvegarde des champs magnétique et électrique n'a pas nécessité de modification.

Tous les autres paramètres qui décrivent la perturbation et la fonction de distribution initiale sont les mêmes que dans la sous-partie précédente. On va choisir pour VLEM_EG un seul niveau de raffinement avec une taille de boîte sur les quantités de mouvement de $\pm 3mc$ dans les 3 directions, un nombre de points sur le niveau R=1 pouvant aller jusqu'à 79^3 et $\epsilon = 10^{-5}$. On va comparer cela avec VLEM sur 79^3 points pour l'espace des quantités de mouvement, ce qui donne aux deux simulations $\Delta p_x = \Delta p_y = \Delta p_z = 0,0769mc$.

Si on regarde le temps de calcul entre chaque intervalle de temps (Tableau.3.12 et Fig.3.16), on peut à nouveau comparer VLEM, VLEM_EG et le temps théorique. Ce n'est pas surprenant de voir que pour VLEM le temps de calcul reste quasiment le même pour chaque intervalle de temps dû au nombre fixe de points. Pour VLEM_EG, on trouve, en comparant le temps de calcul avec celui de VLEM, une réduction moyenne du temps de calcul sur les 7 "restarts" de 54%. Ce résultat est fortement influencé par le nombre de points à chaque instant de la simulation (voir Fig.3.18). En effet, avant $t\omega_p \simeq 80$, le nombre total de points des quantités de mouvement est fixé à 193 605. Si on compare cela avec l'évolution de la fonction de distribution (p_x, p_y) dans le cas 2D2V (Fig.3.13) ou en (p_y, p_z) (Fig.3.17) on est toujours dans la phase de transition, là où il n'y a pas d'évolution visible de la forme de la fonction de distribution. Cela veut dire que le nombre de points de grille pour R=1 est au plus bas et donc le temps de calcul aussi.

Intervalles de temps (en $t\omega_p$)	$0-24$	$24-48$	$48-72$	$72-96$
VLEM	7,33	7,33	7,33	7,33
VLEM_EG	3,03	3,07	3,05	3,77
Théorique	2,88	2,88	2,88	3,48
Intervalles de temps (en $t\omega_p$)	$96-120$	$120-144$	$144-168$	$168-192$
VLEM	7,33	7,33	7,33	7,33
VLEM_EG	4,83	4,75	4,67	4,63
Théorique	4,36	4,31	4,25	4,23

TABLE 3.12 – Temps de calcul (en heures) pour VLEM2D3V entre les différents intervalles de temps. Le temps théorique est calculé à l'aide de l'équation (3.21) où le nombre de points des quantités de mouvement est donné sur la figure 3.18.

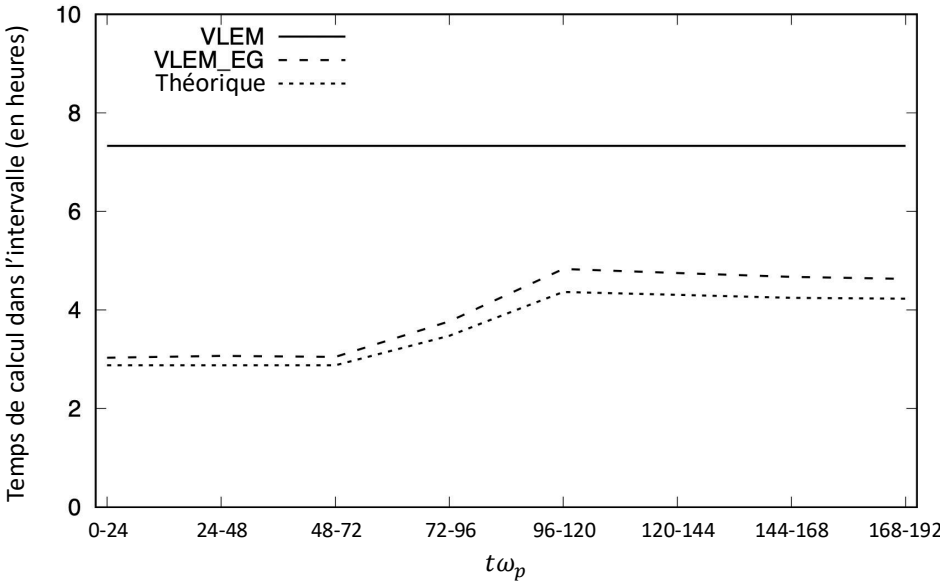

FIGURE 3.16 – Évolutions du temps de calcul de VLEM2D3V, VLEM_EG2D3V et théorique entre chaque intervalles de temps présentés dans le tableau 3.12.

Il y a ensuite une augmentation du nombre de points jusqu'à 303 575 avec une oscillation temporelle. Cela correspond, pour la première partie, à la phase linéaire et donc à l'expansion/le chauffage de la fonction de distribution dans l'espace des quantités de mouvement (Figs.3.13 et 3.17). Pour la deuxième partie, nous sommes au début de la phase de saturation juste avant l'état stationnaire final, ce qui explique les oscillations. Pour ces deux parties, la fonction de distribution est définie sur un nombre plus grand de points qui varie dans le temps. Cela va amener à une augmentation du temps de calcul avec avec un maximum atteint, entre $t\omega_p = 96$ et $t\omega_p = 120$ (Fig.3.16). Ensuite les oscillations ralentissent et s'arrêtent à 284 409 points pour le dernier intervalle de

temps, ce qui amène à une légère diminution du temps de calcul. Cet état final correspond à un équilibre où l'instabilité est atteinte et le piégeage électrostatique garde un mouvement régulier dans le temps (Fig.3.12).

C'est aussi intéressant de se concentrer sur la raison de ce faible coût numérique. En effet, le nombre de points maximum dans chaque direction est plus faible que pour le **Cas 2** de la sous-partie précédente. Cela veut dire que la grille R=1 va recouvrir entièrement le domaine (p_x, p_y) au moins à la saturation. Mais le gain en temps de calcul se fait suivant la direction p_z où il n'y a quasiment pas de fluctuations ou d'expansion de la fonction de distribution (Fig.3.17). La grille en p_z est aux alentours de $p_{z_{min/max}}(R = 1) = \pm 1, 8mc$ ce qui amène à approximativement 60% d'accélération théorique. En regardant la figure 3.17 on peut noter qu'il y a de faibles différences entre les deux résultats numériques à $t\omega_p = 134$ mais ces différences sont principalement des artefacts due aux bords proches de la boîte qui ont un faible effet sur la fonction de distribution. En effet, on voit en comparant les invariants (Figs.3.18 et 3.19) que ces différences n'ont que peu d'effets sur le code.

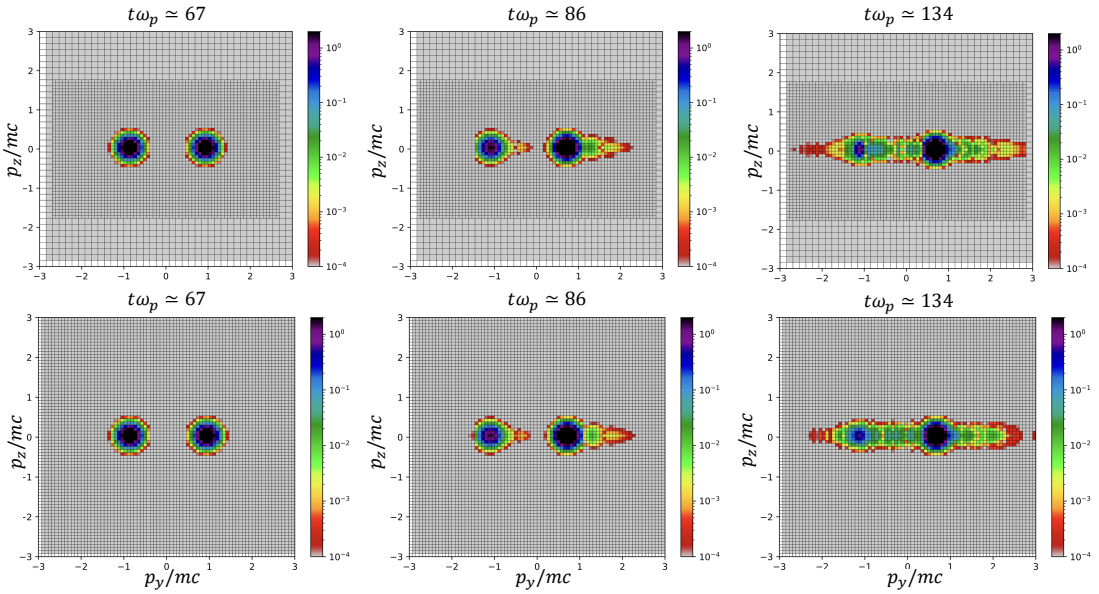

FIGURE 3.17 – Représentations comparatives de la fonction de distribution dans l'espace des phases $p_y - p_z$, moyennée suivant p_x, pour VLEM_EG2D3V avec $\epsilon = 10^{-5}$ (haut) et VLEM2D3V (bas) à $t\omega_p \simeq 67$, 86 et 134 pour un point situé en $(i_x = N_x/2, i_y = N_y/2)$. Les grilles emboîtées de VLEM_EG sont représentées expliquant le gain en temps de calcul dû à la non-expansion en p_z.

Si on détermine aussi le temps de calcul moyen par point et par itération, on trouve $4,35 \times 10^{-9}s$ pour VLEM contre $4,71 \times 10^{-9}s$ pour VLEM_EG. Cela montre que la technique AMR est mieux implémentée sur cette version puisque la différence n'est

plus que de 8% entre les deux codes.

La faible expansion de la fonction de distribution amène à des faibles courants dans la direction z et donc E_z, B_x et B_y n'auront qu'une faible contribution au problème. Ceci explique pourquoi on obtient des résultats similaires entre les cas 2V et 3V.

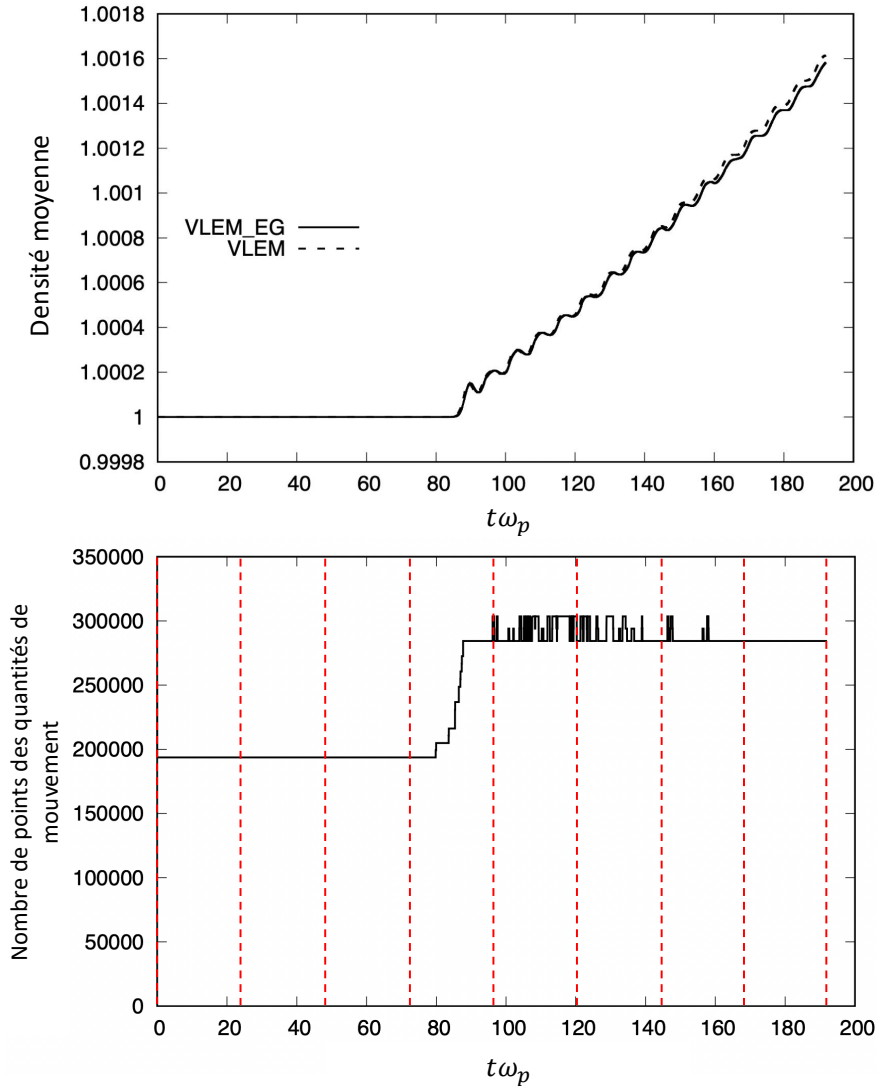

FIGURE 3.18 – Comparaison de l'évolution de la densité moyenne (haut) entre VLEM2D3V et VLEM_EG2D3V. (bas) Évolution temporelle du nombre de points des quantités de mouvement pour VLEM_EG2D3V. Les lignes pointillées rouges représentent les différents intervalles de temps utilisés dans le tableau 3.12.

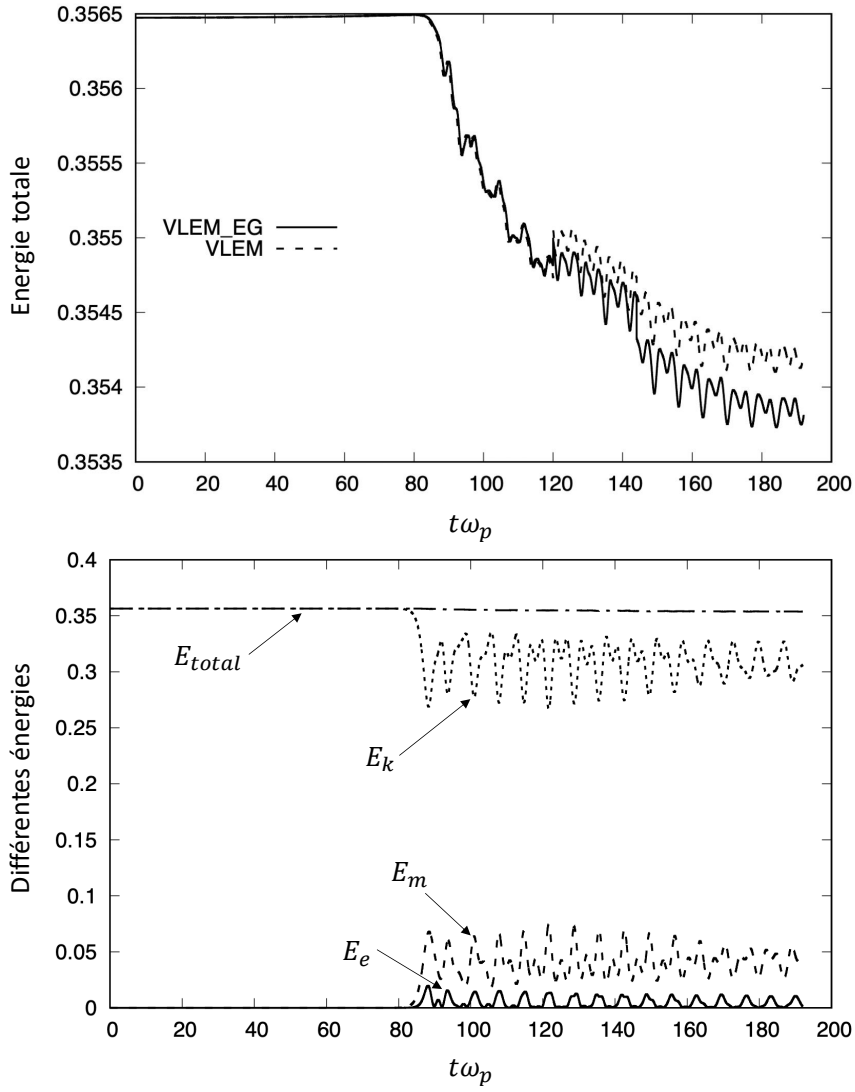

FIGURE 3.19 – Comparaison de l'évolution totale de l'énergie (haut) entre VLEM2D3V et VLEM_EG2D3V. (bas) Évolution des énergies cinétique (ligne en pointillé), magnétique (ligne en tiret), électrique (ligne pleine) et totale (ligne en pointillé et tiret) pour VLEM_EG2D3V.

On va maintenant regarder et comparer les différentes énergies entre VLEM et VLEM_EG (Fig.3.19). A cause des similitudes entre les simulations, les évolutions des énergies seront très proches du cas 2V. On calcule l'énergie cinétique, magnétique et électrique respectivement comme : $E_k = \frac{1}{L_x L_y} \int d^2x \int d^3p \ mc^2(\gamma - 1)f$, $E_m = \frac{\epsilon_0}{2L_x L_y} \int d^2x \ c^2 B^2$ et $E_e = \frac{\epsilon_0}{2L_x L_y} \int d^2x \ E^2$. On observe qu'il n'y a quasiment pas de différences entre les deux codes pour l'évolution de chaque énergie. C'est aussi le cas pour

les cas 2D2V. Le transfert entre énergie magnétique et cinétique est bien calculé et est caractéristique de la CFI. On note aussi que l'énergie totale ($E_{tot} = E_k + E_m + E_e$) et la densité moyenne $\left(n = \frac{1}{L_x L_y} \int d^3p \int d^2x f\right)$ sont bien conservées (environ 1% de variation) et sont quasi-similaires entre les deux codes. On rappelle que les intégrales pour le calcul de la densité et l'énergie cinétique avec VLEM_EG utilisent la méthode présentée dans la sous-partie 3.3.4 tandis que pour VLEM, c'est la sommation standard. La similitude entre ces deux méthodes permet de se rendre compte que la méthode d'intégrale fonctionne bien.

Ces cas valident la bonne implémentation et le gain en temps de calcul de la méthode AMR pour VLEM2D2V et VLEM2D3V.

3.4.3 Cas relativiste de l'instabilité oblique avec VLEM2D2V

On va maintenant tester le code avec une interaction plus relativiste et asymétrique entre deux faisceaux d'électrons. Afin de mieux représenter la fonction de distribution on va maintenant décrire les faisceaux avec une fonction de distribution Maxwell-Jüttner [119] décalée. La représentation sur la figure 3.7 est toujours valide mais la fonction de distribution de Maxwell-Jüttner ne permet pas d'avoir des températures différentes entre la direction x et y, elles doivent être identiques. La forme 3D de la fonction est la suivante :

$$F_{0j} = \frac{1}{4\pi m^3 c^3 \theta_j K_2\left(\frac{1}{\theta_j}\right)} \exp\left(-\frac{(\gamma(\mathbf{p}) - \beta_j p_y)}{\theta_j}\right), \tag{3.23}$$

avec β_j la vitesse de dérive normalisée dans la direction y, $\gamma(\mathbf{p})$ le facteur relativiste,$1/\theta_j = mc^2/(k_B T_j)$ l'inverse de la température normalisée et K_2 la fonction de Bessel modifiée de deuxième type. Pour le problème que l'on va étudier, on peut réécrire l'Eq.(3.23), qui est valide pour un cas 3D3V général, pour décrire un problème 2D2V qui aura une meilleure précision pour les quantités de mouvement. La fonction en 2D2V s'écrit comme $F_{0j} = A\exp\left(-\theta_j^{-1}(\gamma(\mathbf{p}) - \beta_j p_y)\right)$, avec A le facteur de normalisation différent du cas 3D.

On perturbe le système dans le but d'exciter l'instabilité oblique mais cette fois ci avec une asymétrie de température entre un faisceau froid et un faisceau chaud. Pour ce cas, on choisit $L_x = 15,707 d_e$ et $L_y = 20,943 d_e$ pour la taille normalisée de la boite, ce qui correspond à $\Delta k_x d_e = 0,40$, $\Delta k_y d_e = 0,30$. Pour le faisceau froid, on choisit une température $k_B T_1 = 50 keV$ et pour le faisceau chaud relativiste $k_B T_2 = 1010 keV$. On garde $n_1 = n_2 = n_0/2$ mais on va choisir $\beta_1 = -\beta_2 = -0,745$. Ces paramètres correspondent à ce qui a été utilisé dans Ghizzo et al. [120]. La perturbation initiale est encore appliquée à la composante B_z de la même manière que précédemment (Eq.(3.20)) avec $(n_x, n_y) = (1, 1)$ ce qui correspond au mode le plus instable. On choisit pour les autres

paramètres normalisés : $e\delta B_0/(m\omega_p) = 0,005$, $\Delta t\omega_p = 0,01$ avec 30 000 pas de temps, $N_x N_y = 256^2$ sur 2048 processus (256 processus MPI avec 8 taches OpenMP).

Ce type de simulation a déjà été réalisé dans Ghizzo et al. [120] où les paramètres choisis étaient les mêmes sauf pour la température. Dans un premier temps, ils ont choisi $k_B T_1 = 71keV$ et $k_B T_2 = 1010keV$ puis $k_B T_1 = 101keV$ et $k_B T_2 = 875,5keV$. Ces choix ont été faits pour des raisons de ressources numériques. En effet, comme le faisceau chaud occupe une grande place dans la boîte des quantités de mouvement, la taille de la boîte doit être choisie large. Mais comme le faisceau froid est très petit en comparaison, il faut aussi une précision importante pour bien le décrire. C'est pourquoi, ils avaient pris un faisceau froid plus chaud que le cas que l'on va étudier. Pour ce type de simulation, la méthode AMR est bien appropriée puisqu'elle va augmenter la précision seulement là où c'est nécessaire dans l'espace des quantités de mouvement.

L'idée de l'article de Ghizzo et al. [120] était de comparer avec la figure 3 de l'article de Bret et al. [66] pour un faisceau ayant des températures $k_B T_1 = 5keV$ et $k_B T_2 = 2MeV$. Cependant, la fonction de Maxwell Jüttner contenait une coquille et $1/\theta_j$ était défini comme $\sqrt{mc^2/(k_B T_j)}$. Dans cette sous-partie, on a réécrit les vraies températures correspondant à la vraie fonction de distribution de Maxwell Jüttner. C'est pourquoi les températures ne sont plus $5keV$, $10keV$, $20keV$, $1,5MeV$ et $2MeV$ mais sont maintenant respectivement $50keV$, $71keV$, $101keV$, $875,5keV$ et $1010keV$. On verra cependant dans le prochain chapitre que les résultats obtenus avec ces températures modifiées sont très similaires à ceux du cas où les températures sont $k_B T_1 = 5keV$ et $k_B T_2 = 2000keV$.

On lance ces simulations avec VLEM2D2V avec $p_{x_{min}}/p_{x_{max}} = -30mc/30mc$ et $p_{y_{min}}/p_{y_{max}} = -20mc/80mc$ et pour le nombre de points $N_{p_x} = 255$ et $N_{p_y} = 511$. Pour la comparaison, on choisit VLEM2D2V_EG avec un niveau de raffinement, avec la même taille de boîte et un nombre de points maximum pour R=1 égal à $N_{p_x} = 255$ et $N_{p_y} = 511$. On choisit de diminuer le paramètre ϵ à 10^{-7} afin de mieux suivre les micro-oscillations de la fonction de distribution.

Comme dans les sous-parties précédentes, on obtient le même résultat entre les deux codes tant sur l'évolution de la fonction de distribution que sur l'évolution des champs. Le code VLEM prend 19 heures et 37 minutes contre 8 heures et 56 minutes pour la méthode AMR, soit environ 55% de gain en temps de calcul.

On peut voir sur la figure 3.20 que pendant les premiers pas de temps, au début de la phase transitoire, la grille raffinée est bien concentrée sur le faisceau froid ce qui va conduire à un faible nombre de point et donc à un faible coût numérique. Ensuite, quand l'instabilité commence à apparaître de manière oblique dans les régions $p_x < 0$ la grille va bien suivre la filamentation et l'accélération des particules. Le fait qu'à la fin de la simulation la grille occupe moins de 60% de la grille totale justifie la bonne accélération obtenue. Ceci est dû à la fonction en $p_x > 0$ qui est plutôt lisse et donc avec des gradients précisément décrits.

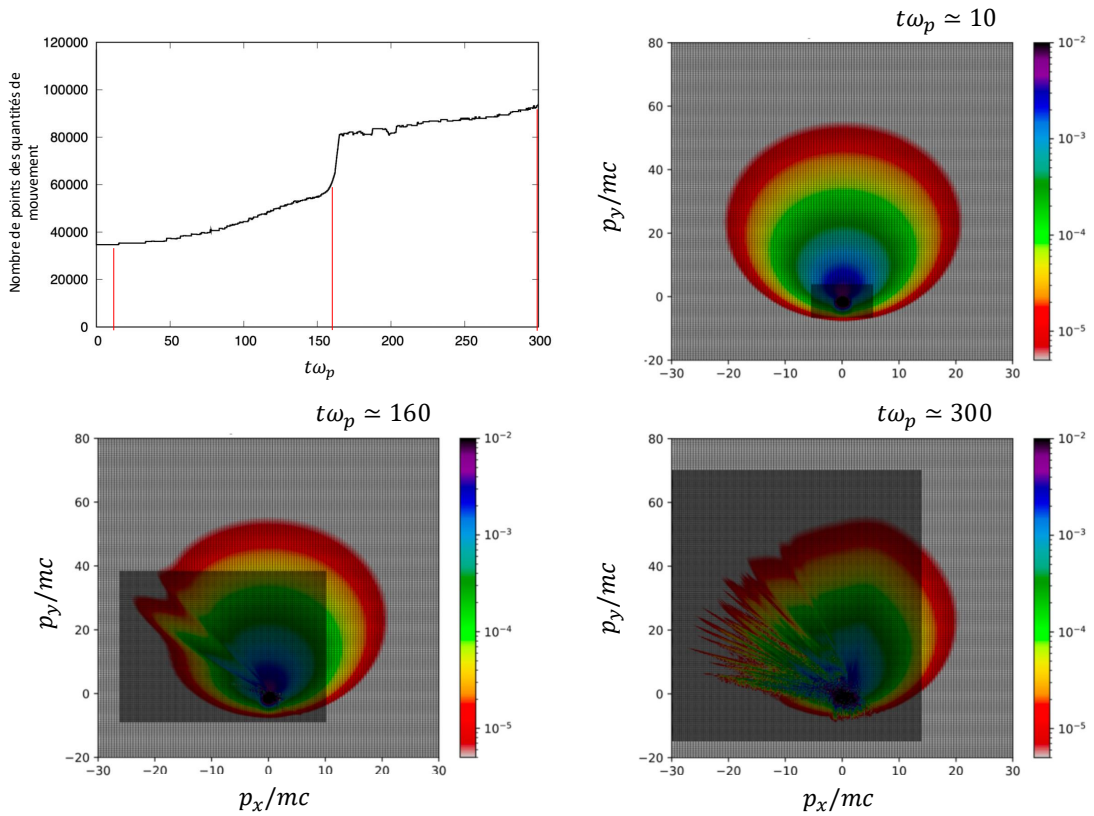

FIGURE 3.20 – (haut-gauche) Évolution temporelle du nombre de points dans l'espace des quantités de mouvement sur la grille la plus fine $R = R_n^{max}$ pour VLEM_EG2D2V avec $\epsilon = 10^{-7}$. Les lignes rouges verticales montrent les différents temps auxquels sont représentées les autres figures. Représentation de la fonction de distribution en $p_x - p_y$ pour VLEM_EG2D2V avec $\epsilon = 10^{-7}$ aux temps $t\omega_p \simeq 10$, 160 et 300 pour un point localisé en $(i_x = 218 \times N_x/255, i_y = N_y/2)$. Les grilles emboîtées sont représentées montrant une bonne description initiale du faisceau froid puis des fluctuations de la fonction de distribution.

Le temps moyen de calcul par point par itération est de $2,75 \times 10^{-10}$ pour VLEM contre $2,62 \times 10^{-10}$ pour VLEM_EG, avec $N_{p_{EG}}$ =62 546 dans l'équation (3.21). Ce résultat peut sembler illogique puisque VLEM_EG a un algorithme plus complexe que VLEM et donc doit avoir une valeur plus importante pour ce paramètre. Mais il est cependant possible que le faible nombre de points dans l'espace des quantités de mouvement permet un accès plus direct et rapide en mémoire accélérant ainsi la simulation par point et par itération. Une meilleure parallélisation du code VLEM_EG peut aussi être à l'origine de cette différence et cela sera observé dans la prochaine sous-partie.

A la fin de cette simulation, on perçoit une transition de OI à CFI qui commence à apparaître. Cette dernière sera étudiée dans le chapitre 4.

3.4.4 Test de performance numérique

Dans cette section, on va mesurer le niveau de parallélisme, MPI et aussi OpenMP du code VLEM_EG en le comparant à la version de VLEM après modification (discutée dans la sous-partie 2.3.1).

Comme dans la sous-partie 2.3.1, les tests ont été réalisés sur le supercalculateur Jean Zay de l'IDRIS sur la partition CPU_P1 qui contient 720 noeuds de calculs scalaires avec 2 processeurs Intel Cascade Lake 6248 (20 coeurs à 2,5 GHz), c'est-à-dire 40 coeurs par noeuds et 192 GB de mémoire par noeud.

Performance, accélération et MPI.

La résolution de l'équation de Vlasov ainsi que les équations de Maxwell est réalisée de manière parallèle sur p processus MPI. Cependant, l'équation de Poisson est calculée en utilisant seulement la parallélisation OpenMP, rendant plus simple l'implémentation de la conservation de la charge avec une transformée de Fourier rapide (voir sous-partie 2.2.7) en utilisant une bibliothèque fft3w.

En général, on peut s'attendre à une accélération $1 \leq S(p) \leq p$. En effet, si on a p processus MPI, on peut diviser le nombre d'opérations séquentielles en p taches amenant à une accélération du temps CPU jusqu'à p. Cependant, il est parfois possible d'avoir $S(p) > p$.

Différentes techniques d'optimisation sont utilisées dans le code : les parallélisations OpenMP et MPI. On choisit de regarder le code VLEM_EG2D2V avec $N_x N_y = 256^2$ dans l'espace des positions et $N_{p_x} N_{p_y} = 32 \times 64$ points de grille pour R=0, avec trois niveaux de raffinement dans l'espace des quantités de mouvement, sur 500 itérations temporelles. On fixe pour cette partie le nombre de taches OpenMP à 4. Ce cas test est donc le même que celui des test sur VLEM dans la sous-partie 2.3.1 excepté que l'on a rajouté avec le code VLEM_EG deux niveaux de raffinement.

Les résultats sont présentés sur la figure 3.21 et dans les tableaux 3.13 et 3.14. L'accélération MPI théorique est tracée en ligne pointillée sur la figure 3.21, tandis que les valeurs numériques sont représentées par des cercles pour VLEM_EG ou des triangles pour VLEM.

Comme dans cette partie, on s'intéresse à l'efficacité MPI on la mesure comme $S(p) = T_{CPU}(1)/T_{CPU}(p)$, avec T_{CPU} qui représente la somme moyenne des temps de calcul OpenMP pour une tache MPI. Les temps écoulés $T_{\text{Écoulé}}$ représentent le temps moyen des taches OpenMP.

Pour VLEM_EG, on observe un bon comportement de la parallélisation jusqu'à 74,6% pour 256 processus (avec $N_{thread} = 4$ taches OpenMP, c'est-à-dire 1024 processeurs). La raison pour la faible perte de performance est liée au traitement séquentiel de l'équation de Poisson utilisé pour maintenir la conservation de la charge (on attribue cette perte d'efficacité à la loi d'Amdahl [121]).

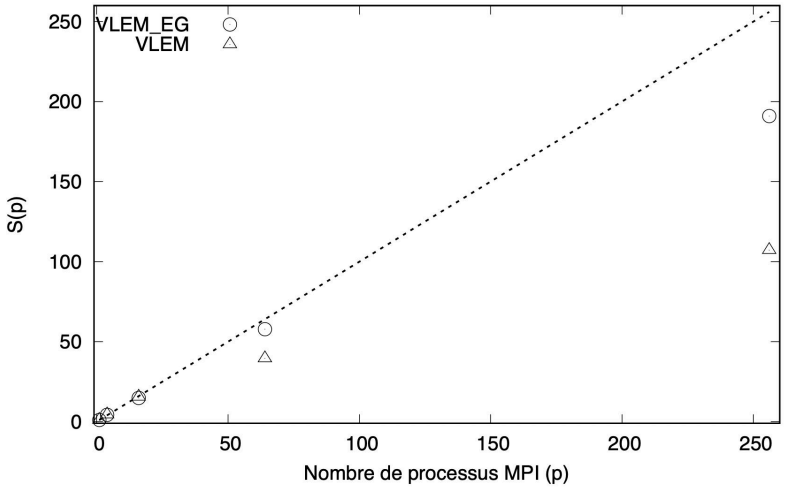

FIGURE 3.21 – Accélération numérique $S\left(p\right)$ en fonction du nombre de processus MPI p pour les codes VLEM_EG2D2V (rond) et VLEM (triangle). La ligne pointillée représente $S(p) = p$, l'accélération théorique. $S\left(p\right) = T_{CPU}(1)/T_{CPU}\left(p\right)$. L'efficacité du code est proche de $E\left(256\right) \simeq 77\%$.

p (Nombre de processus MPI)	$T_{CPU}(s)$	$T_{\text{Écoulé}}(s)$	$T_{CPU}/T_{\text{Écoulé}}$	$S\left(p\right)$	$E\left(p\right)$
1	131447,992	33178,260	3,962	1,000	1,000
4	31154,384	7868,918	3,959	4,219	1,055
16	8877,642	2229,722	3,982	14,807	0,925
64	2274,475	576,961	3,942	57,793	0,903
256	688,682	182,404	3,776	190,869	0,746

TABLE 3.13 – Accélération $S\left(p\right)$ et efficacité $E\left(p\right) = S\left(p\right)/p$ en fonction du nombre de processus MPI p pour le code VLEM_EG2D2V. $S\left(p\right) = T_{CPU}(1)/T_{CPU}\left(p\right)$.

p (Nombre de processus MPI)	$T_{CPU}(s)$	$T_{\text{Écoulé}}(s)$	$T_{CPU}/T_{\text{Écoulé}}$	$S\left(p\right)$	$E\left(p\right)$
1	22142,004	5594,606	3,958	1,000	1,000
4	4996,249	1254,155	3,984	4,432	1,108
16	1430,195	370,733	3,858	15,482	0,968
64	560,597	260,520	2,152	39,497	0,617
256	206,569	105,547	1,957	107,189	0,419

TABLE 3.14 – Accélération $S\left(p\right)$ et efficacité $E\left(p\right) = S\left(p\right)/p$ en fonction du nombre de processus MPI p pour le code VLEM2D2V. $S\left(p\right) = T_{CPU}(1)/T_{CPU}\left(p\right)$. On observe une diminution importante de l'efficacité pour $p \geq 64$, ce qui indique un problème de communication entre les processus.

Pour VLEM, l'efficacité est moins bonne que VLEM_EG dès 64 taches MPI. Cela peut en partie être dû au nombre de points des quantités de mouvement plus faible même si on observe de manière générale une meilleure implémentation de la parallélisation avec la méthode AMR. La raison est que les nouvelles parties du code spécifiques au raffinement de maillage sont toutes parallélisées MPI.

Ces résultats sont une bonne indication que l'implémentation de la parallélisation dans la technique AMR conserve un haut niveau d'efficacité pour la parallélisation MPI du code.

Performances OpenMP

On va maintenant lancer le code VLEM_EG sur plusieurs processus OpenMP en fixant 4 processus MPI et en utilisant le même échantillonnage et le même nombre de pas de temps que dans la sous-partie précédente. On donne les résultats mesurés dans le tableau 3.15. On lance les simulations jusqu'à un maximum de 32 taches OpenMP. Pour les résultats du code VLEM, on va pouvoir reprendre ceux du tableau 2.4.

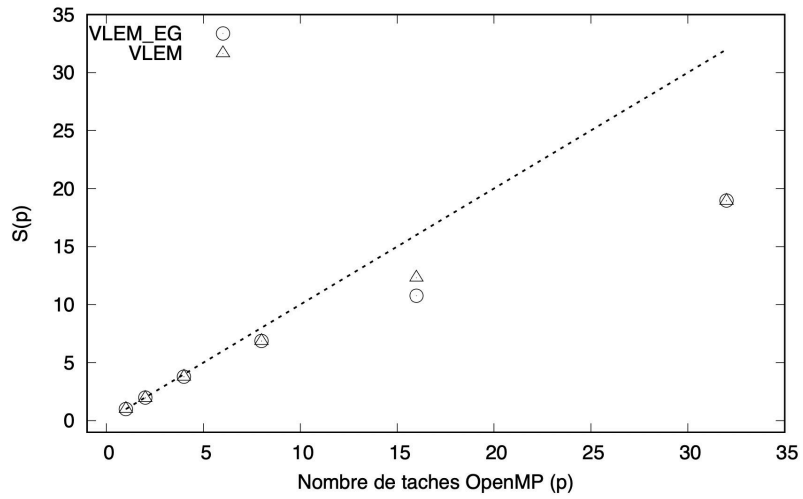

FIGURE 3.22 – Accélération $S\,(p)$ en fonction du nombre de taches OpenMP (p) pour les codes VLEM_EG2D2V (rond) et VLEM(triangle). La ligne pointillées représente $S(p) = p$, l'accélération théorique. $S\,(p) = T_{\text{Écoulé}}(1)/T_{\text{Écoulé}}\,(p)$.

On a tracé figure 3.22, la nouvelle accélération $S\,(p)$ maintenant déteminée à partir du temps écoulé (au lieu du temps CPU) en fonction du nombre de taches OpenMP p. La courbe théorique est encore montrée en ligne pointillée, tandis que les résultats de $S\,(p)$ obtenu à partir de VLEM_EG sont présentés avec des ronds contre des triangles pour VLEM.

Les résultats obtenus avec les codes sont identiques ce qui montre la bonne implémentation de la parallélisation OpenMP dans VLEM_EG.

p (Nombre de taches OpenMP)	$T_{CPU}(s)$	$T_{\text{Écoulé}}(s)$	$T_{CPU}/T_{\text{Écoulé}}$	$S(p)$	$E(p)$
1	29683,132	29788,133	0,996	1,000	1,000
2	29863,300	14994,382	1,992	1,987	0,993
4	31154,384	7868,918	3,959	3,786	0,946
8	34405,181	4345,308	7,918	6,855	0,857
16	43481,328	2764,443	15,729	10,775	0,673
32	48972,385	1570,436	31,184	18,968	0,593

TABLE 3.15 – Accélération $S(p)$ et efficacité $E(p) = S(p)/p$ en fonction du nombre de taches Open-MP p pour le code VLEM_EG2D2V. $S(p) = T_{\text{Écoulé}}(1)/T_{\text{Écoulé}}(p)$.

On peut aussi observer que le parallélisme reste optimal dans la quatrième colonne du tableau 3.15, où l'on a calculé le rapport $T_{CPU}/T_{\text{Écoulé}}$, puisque les valeurs restent très proches du nombre de taches. La chute de l'efficacité que l'on voit surtout apparaître à partir de 16 taches OpenMP est probablement due au nombre total de processeurs qui dépasse le nombre de processeurs dans un coeur (40 processeurs dans un coeur). Cette chute d'efficacité est donc potentiellement due à des communications MPI qui sont plus longues dû aux processeurs qui ne sont plus sur le même coeur.

3.5 Possibilité d'amélioration du raffinement

Dans cette partie, on va donner des pistes d'améliorations du code dont les implémentations ne sont pas prévues mais qui ont été étudiées et dont les limites ont été soulevées.

3.5.1 Grille non rectangulaire

La première amélioration possible pourrait être de sortir de la limitation d'une seule grille rectangle par niveau de raffinement R sur un point de l'espace $\mathbf{x}$ donné. Ces grilles ont été choisies pour fonctionner avec la méthode des splines cubiques à plusieurs dimensions qui a été présentée dans la sous-partie 2.2.5.

Si on choisit de totalement modifier la forme des grilles dans l'espace des quantités de mouvement, c'est-à-dire de ne plus les limiter à des rectangles, il faudra modifier la méthode d'interpolation. Soit en interpolant les points manquants pour retrouver une forme de rectangle, soit en choisissant une méthode d'interpolation plus locale. Il faudra aussi modifier la méthode d'intégration qui est actuellement définie uniquement pour des interfaces entre 2 grilles rectangles.

Une autre option, qui semble plus facilement implémentable serait d'avoir une grille sur le niveau R composées de plusieurs rectangles. Chaque rectangle pourrait

alors effectuer les splines cubiques pour la rotation sans la modifier. Quant aux intégrales, il faudrait réfléchir aux nouvelles interfaces possibles, mais une version automatique du calcul des C existe déjà et permettrait de s'adapter à toutes les configurations sans avoir à regarder à la main au cas par cas. On peut imaginer que ce type de configuration serait idéale pour s'adapter aux instabilités double faisceaux. En effet, si on regarde la figure 3.13, on peut se dire qu'avec des distributions plus éloignées sur la grille des quantités de mouvement (c'est-à-dire un $|p_j|$ plus grand), on pourrait avoir une grille rectangle par faisceau sur le niveau R.

3.5.2 Modification spatiale de la grille raffinée

Une idée qui n'a pas été retenue dans le développement de la technique AMR consistait à utiliser une grille rectangulaire qui change pour chaque point de l'espace (i_x, i_y) et à utiliser une interpolation pour récupérer les points manquants lors des advections spatiales. En effet, pour une advection dans la direction $\mathcal{X}$, il est nécessaire, pour un ensemble de points partageant les mêmes coordonnées (i_y, j_x, j_y), de connaître la fonction de distribution sur l'ensemble des i_x d'un processus (de $i_{x_{start}}$ à $i_{x_{end}}$) et aussi sur quelques points appartenant aux voisins (voir les sous-parties 2.2.6, 2.4.2 et 2.4.3). Il faudrait donc interpoler la valeur de la fonction de distribution sur les points de l'espace des phases où elle n'est pas définie. En effet, avec ce type de raffinement et si l'on se trouve sur un niveau raffiné, la fonction peut exister en un point (i_x, i_y, j_x, j_y) mais pas sur son point voisin $(i_x + 1, i_y, j_x, j_y)$. La nouvelle interpolation pour ces points, ainsi que le fait que les processus auraient dû attendre leurs voisins pour chaque échange d'information, auraient conduit à un temps de calcul égal ou supérieur à celui requis par la méthode de raffinement que nous implémentons ici.

Le problème principal serait donc la parallélisation de ce type de méthode. En effet, on peut garder la méthode de parallélisation spatiale actuelle, mais si un cube MPI spatial (un processus) a moins de points dans l'espace des quantités de mouvement qu'un autre, il sera plus rapide pour effectuer les intégrations et la rotation. Cependant, pour les advections spatiales comme il y a le partage d'information entre processus voisins, il faut qu'ils s'attendent et donc il n'y a pas de gain de temps de calcul dans cette partie. Cette méthode pourrait donc créer un déséquilibre de charge numérique. C'est l'argument qui avait été soulevé dans la sous-partie 3.1.1. Les advections spatiales seraient donc toujours aussi longue que la méthode actuelle avec la grille rectangle la plus grosse sur tout l'espace.

On pourrait, pour résoudre ces problèmes, optimiser le code avec une répartition de charge entre les processus. C'est ce qui est fait dans les codes PIC quand des particules se retrouve au même endroit de l'espace. Cela nécessiterait de modifier la décomposition spatiale et complexifierait l'ensemble du code. De plus avec le code périodique que l'on a et le type de simulation qui sont étudiée, il y a rarement des situations où le nombre de points des quantités de mouvement dans un processus doit

être beaucoup plus important que dans le processus voisin. Cette étape serait donc superflue. Ce n'est cependant pas le cas dans d'anciennes versions de VLEM qui traitaient la propagation d'ondes laser à la surface d'un plasma [45, 122]. Dans ces cas, le code n'était plus périodique et le plasma était localisé spatialement.

3.5.3 Raffinement en espace avec les ondelettes d'Arakawa

Même si l'intérêt d'un raffinement spatial semble limité par rapport aux instabilités induite par les ondes électromagnétiques aux interfaces de grille [114, 115]. On s'est intéressé à une technique qui pourrait fonctionner avec la méthode de saute-mouton (ou leap-frog) de Yee.

En effet, dans la méthode de Yee, les champs ne sont pas positionnés au même endroit spatialement (voir figure 2.3). Or un raffinement comme effectué actuellement dans l'espace des quantités de mouvement, c'est-à-dire en créant un point entre deux points existants, ne fonctionnerait pas. En effet, on peut voir sur la gauche de la figure 3.23 qu'avec un tel raffinement pour tous les points on n'améliore pas le schéma de Yee.

Il faudrait en effet plutôt créer des points situés en $-1/4$ et $1/4$ comme à droite de la figure 3.23, on conserverait ainsi facilement le schéma saute-mouton. Ces points définiraient alors la nouvelle grille raffinée R+1 mais ils ne seraient pas situés au même endroit de l'espace que la grille R.

Pour la sélection du critère de raffinement en espace, on pourrait s'inspirer de Vlasiator qui compare les gradients des champs électromagnétiques [60].

Pour créer les points, situé en $1/4$, on pourrait par exemple prendre, la fonction d'ordre 3 suivante :

$$f\left(\frac{h}{4}\right) = \frac{1}{32}\left(-3f(-h) + 30f(0) + 5f(h)\right) + \frac{5}{128}h^3 f^{(3)} + O(h^4), \tag{3.24}$$

avec h le pas d'espace d'une des dimensions spatiales.

Comme les points sur le niveau R ne sont plus situés au même endroit de l'espace que les points plus grossier R-1, on ne peut plus remonter l'information directement. On va utiliser la formule d'interpolation suivante pour partager l'information sur les points les plus grossiers :

$$\begin{aligned} f\left(h\right) = \frac{1}{224}\Bigg(&-3f\left(-\frac{h}{4}\right) - 5f\left(\frac{h}{4}\right) + 120f\left(\frac{3h}{4}\right) \\ &+ 120f\left(\frac{5h}{4}\right) - 5f\left(\frac{7h}{4}\right) - 3f\left(\frac{9h}{4}\right)\Bigg) + \frac{45}{14336}h^4 f^{(4)} + O(h^6). \end{aligned} \tag{3.25}$$

On peut justifier cela par une description utilisant les ondelettes d'Arakawa explicité dans l'annexe C.4.

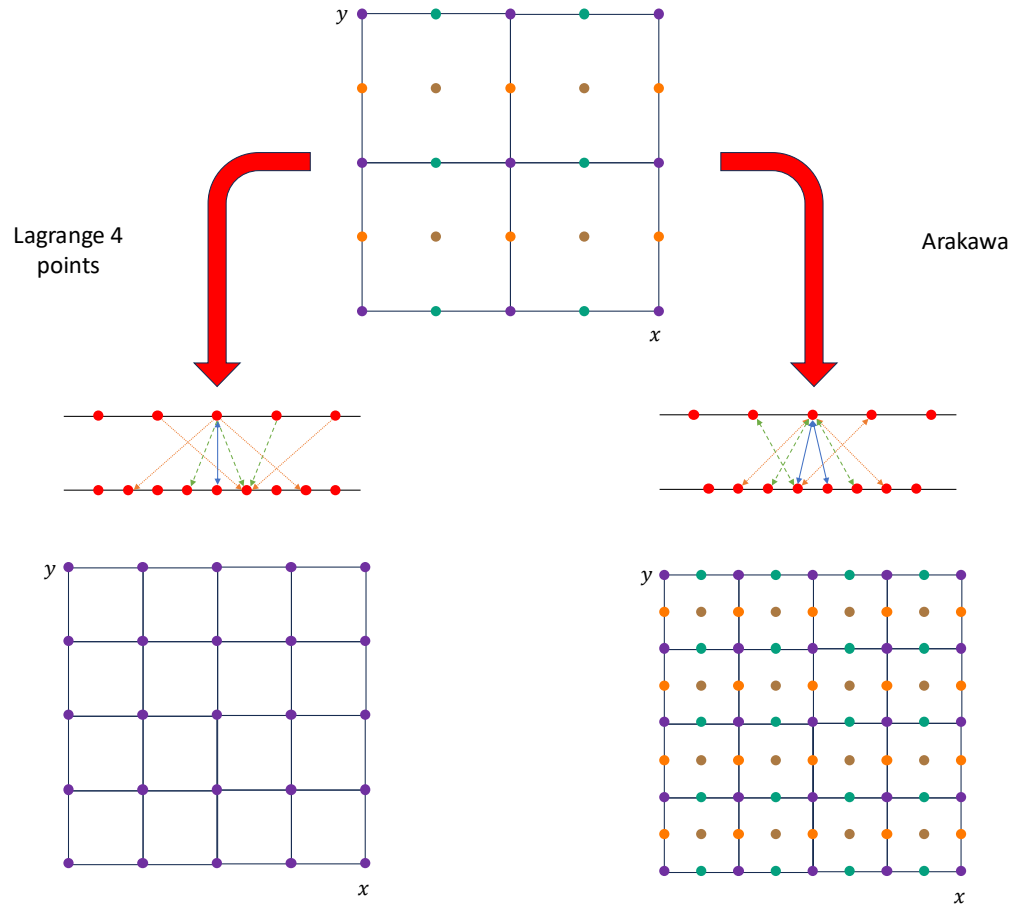

FIGURE 3.23 – Schéma du raffinement spatiale avec la méthode utilisée pour les quantités de mouvement (gauche) décrite dans ce chapitre et avec la méthode d'Arakawa. On peut associer les points violet avec la fonction de distribution, les points verts, oranges et marrons respectivement à E_x, E_y et B_z pour un code 2D2V.

Cependant, cela ne règle pas les problèmes qui peuvent être rencontrés dans les codes sur les instabilités concernant les ondes électromagnétiques à l'interface de grille [114, 115].

Chapitre 4

Études physiques avec le code VLEM_EG

Sommaire

4.1 Étude de l'instabilité oblique relativiste à 1010keV 126
4.1.1 Transition vers CFI 127
4.1.2 Impact de la température du faisceau froid 129
4.1.3 Évolution des énergies et de la fonction de distribution . . . 130
4.1.4 Analyse du cas oblique 1010keV 133
4.2 Étude de l'instabilité oblique relativiste à 2000keV 139
4.2.1 Transition OI vers CFI 141
4.2.2 Évolution des énergies et de la fonction de distribution . . . 142
4.3 Étude de l'instabilité Weibel relativiste 145
4.3.1 Comparaison de trois fonctions de distributions 146
4.3.2 Instabilité Weibel relativiste 146

4.1 Étude de l'instabilité oblique relativiste à 1010keV

Comme annoncé à la fin de la sous-partie 3.4.3, on va s'intéresser à la transition de OI à CFI que l'on commence à percevoir à la fin de la simulation. On rappelle que le cas étudié est inspiré de l'article de Ghizzo et al. [120] où les auteurs se sont intéressés à l'instabilité oblique qui devait être la plus instable en s'inspirant des travaux de Bret et al. [66]. On va donc, dans cette partie, étudier ce système de deux faisceaux d'électrons avec un faisceau froid et un faisceau relativiste.

On relance donc la simulation de la sous-partie 3.4.3 avec les mêmes paramètres mais avec un niveau de grille grossière en plus tel que $R_{max} = 2$, ce dernier va permettre d'accélérer le temps de calcul du code. On garde $\epsilon = 10^{-7}$ dans les équations (3.4) et (3.3). On rappelle que la fonction de distribution est une fonction de Maxwell-Jüttner décalée [119] qui à deux dimensions d'espace et deux dimensions de quantité de mouvement s'écrit pour le faisceau d'électron j :

$$F_{0j} = A\exp\left(-\theta_j^{-1}(\gamma(\mathbf{p}) - \beta_j p_y)\right), \tag{4.1}$$

avec A le facteur de normalisation, β_j la vitesse de dérive normalisée dans la direction y, $\gamma(\mathbf{p})$ le facteur relativiste et $1/\theta_j = mc^2/(k_B T_j)$.

Le système que l'on étudie est composé d'un faisceau froid et d'un faisceau chaud se propageant dans des directions opposées suivant y. Pour le faisceau froid, on choisit une température $k_B T_1 = 50keV$ et pour le faisceau chaud relativiste on choisit $k_B T_2 = 1010keV$, avec une densité équivalente entre les deux faisceaux ($n_1 = n_2 = n_0/2$) et $\beta_1 = -\beta_2 = -0,745$.

On perturbe le système dans le but d'exciter l'instabilité oblique pour cela, on choisit $L_x = 15,707 d_e$ et $L_y = 20,943 d_e$ pour la taille normalisée de la boîte, ce qui correspond à $\Delta k_x d_e = 0,40$ et $\Delta k_y d_e = 0,30$. La perturbation initiale est appliquée à la composante B_z sous la forme :

$$\delta B_z = \delta B_0 \sin(n_x \Delta k_x x + n_y \Delta k_y y), \tag{4.2}$$

avec $e\delta B_0/(m\omega_p) = 0,005$ et $(n_x, n_y) = (1,1)$. On choisit $\Delta t\omega_p = 0,01$ mais cette fois-ci, à la différence de la sous-partie 3.4.3, on prend 50 000 pas de temps afin d'aller plus loin dans les itérations temporelles. Pour la grille spatiale, on a $N_x N_y = 256^2$ sur 2048 processus (256 processus MPI avec 8 taches OpenMP). On choisit un nombre de points maximum pour la grille $R = R^{max} = 2$ égal à $N_{p_x} = 253$ et $N_{p_y} = 509$ avec comme limite de boîte des quantités de mouvement $p_{x_{min}}/p_{x_{max}} = -30mc/30mc$ et $p_{y_{min}}/p_{y_{max}} = -20mc/80mc$.

La simulation étant identique à celle du cas de la sous-partie 3.4.3, on observe bien, à gauche de la figure 4.1, que l'instabilité oblique démarre dès le début de la simulation. On observe aussi, sur cette représentation spatiale des champs B_z et E_x, qu'il existe une transition du mode oblique (1,1) vers le mode CFI (1,0) aux alentours de $t\omega_p \simeq 280$.

Cette transition est accompagnée d'une augmentation de l'intensité des champs E_x et B_z. On discutera plus en détail les échanges d'énergie dans la sous-partie 4.1.4.

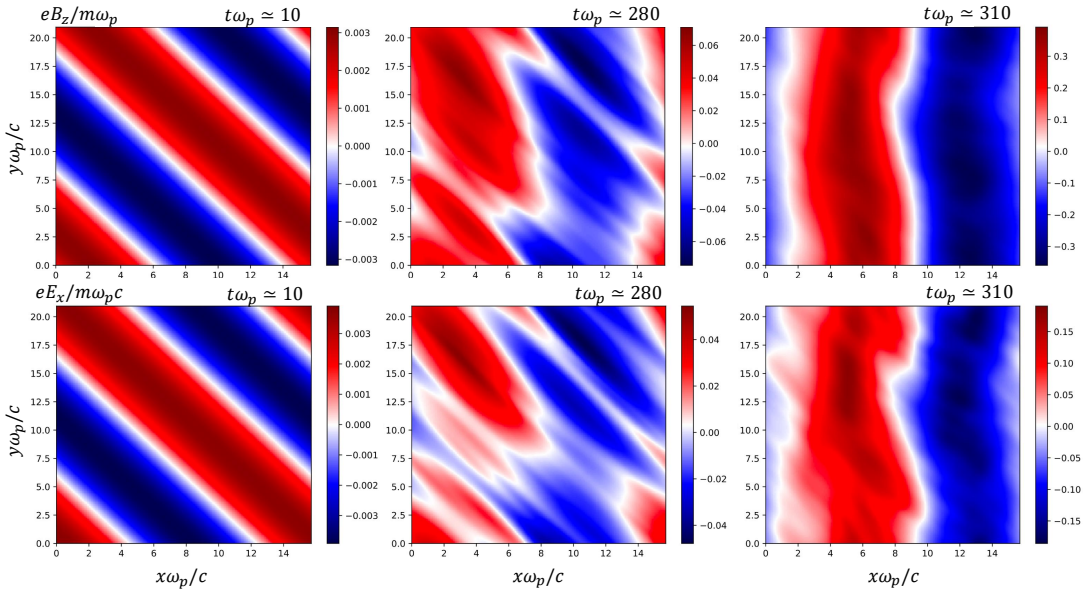

FIGURE 4.1 – Représentation de l'amplitude normalisée des champs B_z (haut) et E_x (bas) à $t\omega_p \simeq 10$, 280 et 310. La transition entre OI et CFI se fait aux alentours de $t\omega_p \simeq 280$.

4.1.1 Transition vers CFI

Dans cette sous-partie, on va analyser cette transition afin de déterminer si elle est due à la phase de saturation de l'instabilité oblique qui, en modifiant la fonction de distribution, entraînerait une transition vers CFI, ou si, au contraire, la CFI était le mode le plus instable depuis le début.

On choisit donc de tracer sur la figure 4.2 l'évolution de l'intensité des modes, une option qui a été ajoutée dans le code et expliquée dans la sous-partie 2.3.2. Sur cette figure, on a représenté les modes les plus instables présents dans la simulation qui sont (n_x,n_y)=(1,1), (1,0), (2,0) et (0,1) qui correspondent respectivement à l'OI, la CFI et son harmonique, et la TSI. On observe, comme ce qui est prévu dans les codes Semi-Lagrangien, que à part pour le mode (1, 1) pour lequel on retrouve bien l'intensité initiale avec laquelle il a été excité ($\ln(0,005) \simeq -5,30$) sur B_z, les autres modes démarrent à un niveau beaucoup plus faible proche de l'erreur d'approximation de l'ordinateur [123].

La phase de transition est bien visible entre OI et CFI lorsque l'intensité du mode

(1,0) devient plus importante que celle du mode (1,1). Cette transition se fait aux alentours de $t\omega_p \simeq 280$ comme ce qui a pu être observé précédemment sur la figure 4.1.

Le taux de croissance numérique calculé pour l'instabilité oblique ($k_x d_e = 0,40$, $k_y d_e = 0,30$) est de $\Gamma_{11}^{num}/\omega_p = 0,057$. En comparaison, dans l'article de Ghizzo et al. [120], pour une température légèrement différente du faisceau froid ($k_B T_1 = 71,5 keV$) ils avaient $\Gamma_{11}^{comp}/\omega_p = 0,08$. Cependant, l'intensité de la composante $\hat{B}_z$ du champ magnétique ne varie que peu et semble donc être saturée presque initialement. Il est donc possible que le taux de croissance mesuré ici soit différent de celui que l'on aurait pû mesurer sans excitation initiale.

Le taux de croissance de la CFI, quant à lui, est de $\Gamma_{10}^{num}/\omega_p = 0,08$ entre $t\omega_p = 0$ et 200 et $\Gamma_{10}^{num}/\omega_p = 0,11$ entre $t\omega_p = 200$ et 300. La CFI présente donc une rupture de pente aux alentours de $t\omega_p = 200$ qui pourrait être due aux modifications de la fonction de distribution, ces modifications étant causées par la saturation de l'OI.

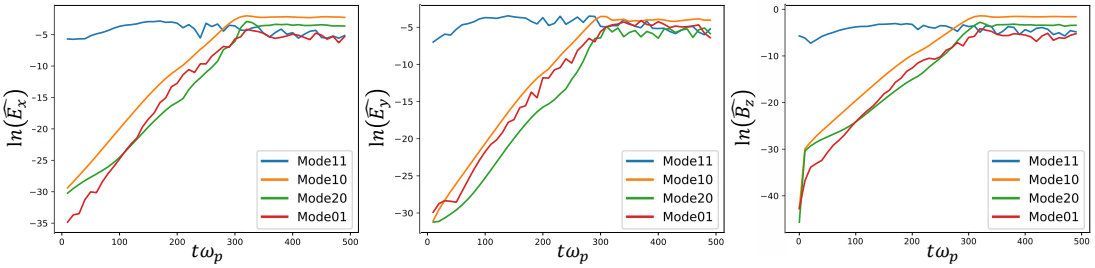

FIGURE 4.2 – Évolution de l'intensité des champs électriques et magnétique $\hat{E}_x(n_x\Delta k_x, n_y\Delta k_y)$ (à gauche), $\hat{E}_y(n_x\Delta k_x, n_y\Delta k_y)$ (au milieu) et $\hat{B}_z(n_x\Delta k_x, n_y\Delta k_y)$ (à droite) pour les modes (n_x,n_y)=(1,1), (1,0), (2,0) et (0,1) correspondant respectivement aux instabilités OI, CFI, CFI (harmonique) et TSI. Les résultats ont été obtenus pour une perturbation initiale sur le mode (1,1).

On peut cependant s'interroger au vu des résultats si le mode le plus instable pour ce cas est vraiment l'OI. On relance donc la même simulation mais en choisissant d'exciter plusieurs modes tel que :

$$\delta B_z = \delta B_0(\sin(\Delta k_x x + \Delta k_y y) + \sin(\Delta k_x x) + \sin(\Delta k_y y)), \tag{4.3}$$

avec $e\delta B_0/(m\omega_p) = 0,005$. Cela correspond à exciter simultanément l'OI, la CFI et TSI respectivement sur les modes (n_x,n_y)=(1,1), (1,0) et (0,1). De cette façon, les modes commenceront au même niveau et la croissance de la CFI sera moins affectée par la modification de la fonction de distribution due à la phase de saturation de l'OI.

On trace à nouveau l'évolution de l'intensité des modes les plus instables sur la figure 4.3. On observe maintenant que les trois modes excités initialement commencent à $-5,30$. La première chose que l'on observe sur cette figure 4.3 est que la CFI va

saturer avec une intensité plus importante que l'OI. On en déduit donc que, dans ce cas physique, la CFI est plus instable que l'OI ce qui explique la transition observée précédemment sur les figures 4.1 et 4.2. En effet, lorsque l'on a initialement excité uniquement l'OI dans la sous-partie 3.4.3, le mode a pu saturer plus rapidement que les autres et avoir un impact sur les échanges d'énergies avant le mode le plus instable (1,0) de la CFI. Cependant, la saturation de l'OI dans le cas précédent semble avoir que peu affecté la croissance de la CFI puisque le taux de croissance est maintenant de $\Gamma_{10}^{num}/\omega_p = 0,11$ (figure 4.3), valeur identique à celle trouvée précédemment après la rupture de pente.

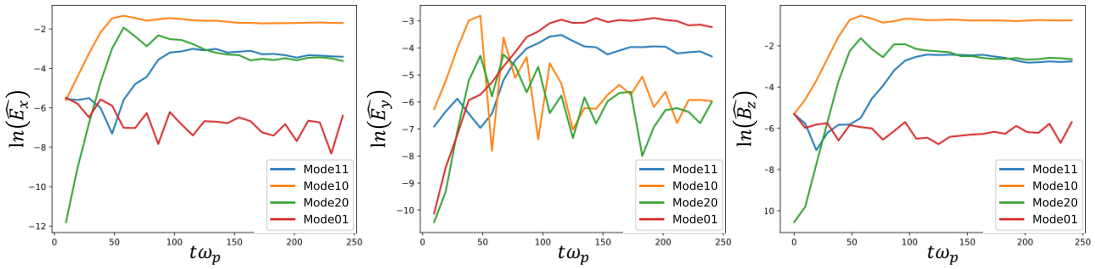

FIGURE 4.3 – Évolution de l'intensité des champs électriques et magnétique $\hat{E}_x(n_x\Delta k_x, n_y\Delta k_y)$ (à gauche), $\hat{E}_y(n_x\Delta k_x, n_y\Delta k_y)$ (au milieu) et $\hat{B}_z(n_x\Delta k_x, n_y\Delta k_y)$ (à droite) pour les modes (n_x,n_y)=(1,1), (1,0), (2,0) et (0,1) correspondant respectivement aux instabilités OI, CFI, CFI (harmonique) et TSI. Les résultats ont été obtenus pour une perturbation initiale sur les trois modes (1,1), (1,0), et (0,1).

4.1.2 Impact de la température du faisceau froid

Dans cette sous-partie, on va s'intéresser à l'impact qu'a la température du faisceau froid sur les instabilités mises en jeu et sur cette transition CFI à OI. En effet, dans l'article de Bret et al. [66] qui nous sert de référence pour la résolution analytique des instabilités OI, CFI et TSI relativistes avec deux faisceaux de Maxwell-Jüttner, la variation de température à été uniquement effectuée sur le faisceau chaud mais pas sur le faisceau froid. Lorsque le faisceau froid est fixé à $k_BT_1 = 50keV$ et que l'on augmente la température du faisceau chaud, Bret et al. [66] prédisent qu'en gardant $n_1 = n_2 = n_0/2$, la CFI sera la plus instable pour les basses valeurs de température du faisceau chaud $k_BT_2 < 300keV$ tandis que l'OI dominera pour les hautes valeurs de température $k_BT_2 > 400keV$. Autrement dit, lorsque l'anisotropie de température est plus importante, c'est l'OI qui est le mode le plus instable.

On a choisi de reprendre le cas initial où l'on perturbe uniquement le mode (1,0) afin d'observer comment cette transition va évoluer. On trace, sur la figure 4.4,

l'évolution de l'intensité des modes les plus instables pour le champ magnétique pour trois températures différentes $k_BT_1 = 32keV$, $k_BT_1 = 50keV$ et $k_BT_1 = 71,5keV$.

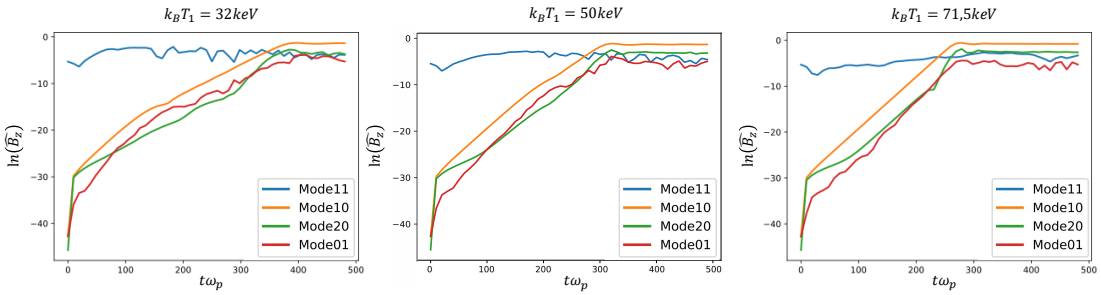

FIGURE 4.4 – Évolution de l'intensité du champ magnétique $\hat{B}_z(n_x\Delta k_x, n_y\Delta k_y)$ pour les modes (n_x,n_y)=(1,1), (1,0), (2,0) et (0,1) correspondant respectivement aux instabilités OI, CFI, CFI (harmonique) et TSI. Les résultats ont été obtenus pour une perturbation initiale sur le mode (1,1) en modifiant la température du faisceau froid entre $k_BT_1 = 32keV$ (à gauche), $k_BT_1 = 50keV$ (au milieu) et $k_BT_1 = 71,5keV$ (à droite).

On observe sur la figure 4.4 que pour les trois températures, l'OI sature presque initialement, ce qui rend impossible la définition d'un taux de croissance pour cette instabilité. Cependant, lorsque la température du faisceau froid augmente, c'est-à-dire que l'anisotropie de température diminue, le taux de croissance de la CFI augmente. On peut observer cela au travers de la diminution de la durée avant d'atteindre la saturation. Ce résultat est donc cohérent avec ceux obtenus dans [66] où l'on observe que lorsque la température du faisceau chaud augmente, c'est-à-dire que l'anisotropie de température augmente, le taux de croissance de la CFI diminue et l'OI devient l'instabilité avec le taux de croissance le plus important.

Lorsque l'on monte à une température équivalente entre faisceau froid et chaud, donc $k_BT_1 = k_BT_2 = 1010keV$, on n'observe plus d'instabilités sur les pas de temps observés.

4.1.3 Évolution des énergies et de la fonction de distribution

Un phénomène que l'on retrouve dans toutes les simulations des instabilités de type Weibel est la diminution de l'énergie magnétique dans la phase non linéaire au début de la saturation de l'instabilité. Une explication apportée dans Polomarov et al. [124] est que cette dernière provient de la fusion dans l'espace (x,y) de filaments ayant un courant d'intensité supérieur au courant d'Alfvén. La résultante de cette fusion va entraîner un courant total moins important que la somme des deux courants initiaux, entraînant ainsi une diminution de l'énergie magnétique. Cependant, cette fusion de filaments dans l'espace n'est possible que si la configuration le permet et dans nos simulations ce n'est pas le cas. En effet, dans un cas à deux dimensions d'espace et deux

dimensions de quantité de mouvement, le champ magnétique est seulement suivant z. Cela empêche la reconnexion magnétique mais aussi la fusion de filaments de courant comme ce qui est représenté dans l'article de Polomarov et al. [124]. On va donc chercher une autre hypothèse, au travers de la synchronisation, permettant de justifier la diminution de l'énergie magnétique dans nos simulations.

Phasestrophie

Un paramètre que l'on va introduire et qui permet de décrire la filamentation est la phasestrophie [48, 125, 126, 127]. C'est un paramètre inspiré de l'enstrophie de la mécanique des fluides [125, 128], qui est la moyenne du carré des perturbations de la vorticité. La phasestrophie a été introduite dans le cadre de la turbulence gyrocinétique [125] et reprise dans le code VLEM récemment au travers du flux de phasestrophie [48]. C'est un paramètre proche de la norme L^2 mais qui n'est pas un invariant de Casimir, il va donc pouvoir varier au cours de la simulation. On définit le flux de phasestrophie en 2D2V comme :

$$\epsilon_\varphi = -\int (d^2k/V)\epsilon_{\varphi,\mathbf{k}}, \tag{4.4}$$

avec $V = L_x \times L_y$ le "volume" de la boîte en espace et $\epsilon_{\varphi,\mathbf{k}}$ la densité de flux de phasestrophie que l'on peut écrire comme :

$$\epsilon_{\varphi,\mathbf{k}} = -\int d^2p \frac{|\delta f_\mathbf{k}|^2}{2\frac{\partial F_0}{\partial p_x}} \frac{p_x}{m\gamma}, \tag{4.5}$$

avec F_0 la fonction de distribution initiale et $\delta f_\mathbf{k}$ la variation de la fonction de distribution dans l'espace de Fourier tel que l'on a $f(\mathbf{x},\mathbf{p},t) = F_0(\mathbf{p}) + \delta f_\mathbf{k}(\mathbf{p},t)e^{i\mathbf{k}\cdot\mathbf{x}} + c.c.$, avec $c.c.$ le complexe conjugué.

Le flux de phasestrophie, en plus de permettre de décrire la croissance des filaments et le transfert de quantités de mouvement dans le système [48], est directement lié à la variation de la densité spectrale d'énergie magnétique ($\epsilon_{m,\mathbf{k}}$) [48] via la relation suivante :

$$\partial_t \left(\epsilon_{m,\mathbf{k}} + \epsilon_{\varphi,\mathbf{k}}\right) = 0. \tag{4.6}$$

Afin de donner une définition de cette densité spectrale d'énergie magnétique, il est nécessaire d'introduire le modèle de Kuramoto [129] pour l'équation de Vlasov.

Modèle de Kuramoto

Le modèle de Kuramoto [129] est un modèle permettant d'expliquer pourquoi des oscillateurs de toutes sortes finissent par se mettre en phase (le cri des criquets, des personnes qui marchent, des horloges). C. Huygens est le premier à avoir remarqué ce phénomène en observant deux horloges sur le même mur qui se sont mises à battre

en phase. Le modèle de Kuramoto [129] a par la suite été introduit afin de donner une description mathématique du phénomène. En écrivant :

$$\dot{\theta_n} = \omega_n + Kr\sin\Big(\bar{\theta} - \theta_n\Big), \tag{4.7}$$

$$\dot{J_n} = KrJ_n\cos\Big(\bar{\theta} - \theta_n\Big), \tag{4.8}$$

avec $n = 1, \cdots, N$ pour décrire les N oscillateurs de fréquence propre ω_n, de phase θ_n et d'action J_n. $\bar{\theta}$ désigne la moyenne de la phase de tous les oscillateurs, K est le paramètre de couplage et r est le paramètre d'ordre qui varie entre 0 et 1 et qui mesure la cohérence du système. Si $r = 1$, alors il y aura une synchronisation globale du système, c'est-à-dire que tous les oscillateurs vont tendre vers la même phase. À l'inverse, si $r = 0$, les oscillateurs ne s'affecteront pas entre eux.

Il est possible, en réécrivant l'équation de Vlasov dans l'espace de Fourier spatial, de retrouver une forme similaire à l'équation de Kuramoto [48, 130]. L'équation de Vlasov devient alors :

$$\partial_t \hat{f}_{\mathbf{k}}(\mathbf{p}, t) = -i\mathbf{k}\cdot\mathbf{p}\frac{\hat{f}_{\mathbf{k}}}{m\gamma} + R_{\mathbf{k}}(\mathbf{p}, t), \tag{4.9}$$

avec $R_{\mathbf{k}}(\mathbf{p}, t) = |R_{\mathbf{k}}|e^{i\Theta_{\mathbf{k}}}$ qui est la transformée de Fourier de $R(\mathbf{x}, \mathbf{p}, t) = -e(\mathbf{E} + \frac{\mathbf{p}\times\mathbf{B}}{m\gamma})\cdot\boldsymbol{\nabla}_{\boldsymbol{p}} f$.

En séparant la phase (φ_k) de l'amplitude ($|\hat{f}_{\mathbf{k}}|$) dans la fonction de distribution $\hat{f}_{\mathbf{k}}(\mathbf{p}, t) = |\hat{f}_{\mathbf{k}}|e^{i\varphi_k}$, on peut retrouver à partir de l'équation (4.9) les équations suivantes :

$$\partial_t\varphi_{\mathbf{k}} = -\frac{\mathbf{k}\cdot\mathbf{p}}{m\gamma} + \frac{|R_{\mathbf{k}}|}{|\hat{f}_{\mathbf{k}}|}\sin(\Theta_{\mathbf{k}} - \varphi_{\mathbf{k}}), \tag{4.10}$$

$$\partial_t|f_{\mathbf{k}}| = |R_{\mathbf{k}}|\cos(\Theta_{\mathbf{k}} - \varphi_{\mathbf{k}}). \tag{4.11}$$

On observe que ce système d'équations a une forme similaire à celui du modèle de Kuramoto (Eqs.(4.7-4.8)).

Si on écrit maintenant la transformée de Fourier des champs électromagnétiques, on peut les réécrire comme $\hat{E_{x,\mathbf{k}}} = |E_{x,\mathbf{k}}|e^{i\theta_x}$, $\hat{E_{y,\mathbf{k}}} = |E_{y,\mathbf{k}}|e^{i\theta_y}$ et $\hat{B_{z,\mathbf{k}}} = |B_{z,\mathbf{k}}|e^{i\theta_z}$. On donne ainsi la définition de la variation de la densité spectrale d'énergie magnétique comme :

$$\epsilon_{m,\mathbf{k}} \equiv \frac{1}{2}(|E_{x,\mathbf{k}}|^2 + |E_{y,\mathbf{k}}|^2 - c^2|B_{z,\mathbf{k}}|^2), \tag{4.12}$$

ce qui permet d'écrire :

$$\epsilon_{m,\mathbf{k}}\frac{\partial\Theta_{\mathbf{k}}}{\partial t} = -\int d^2p\frac{|\delta f_{\mathbf{k}}|^2}{2\frac{\partial F_0}{\partial p_x}}\frac{p_x}{m\gamma}\left(\frac{\partial\varphi_{\mathbf{k}}}{\partial t} + \frac{\mathbf{k}\cdot\mathbf{p}}{m\gamma}\right). \tag{4.13}$$

On reconnaît dans le terme de gauche de l'intégrale la densité de flux de phasestrophie (Eq.(4.5)).

Lorsque les modes se synchronisent, le terme de droite $\left(\frac{\partial \varphi_{\mathbf{k}}}{\partial t} + \frac{\mathbf{k}\cdot\mathbf{p}}{m\gamma}\right)$ dans l'intégrale de l'équation (4.13) devient proche de zéro car dans l'équation (4.10), $\varphi_{\mathbf{k}}$ va tendre vers la phase $\Theta_{\mathbf{k}}$. Cependant, le terme de gauche de l'équation (4.13) peut être non nul et va être fixé par l'instabilité qui est en place. Par exemple, pour l'OI, si la fréquence réelle est non nulle on aura $\omega_{OI} \simeq \frac{\partial \Theta_{\mathbf{k}}}{\partial t} \neq 0$. Ainsi, le premier terme de droite qui correspond au flux de phasestrophie (Eq.(4.5)) devra être non nul dans le cas d'une synchronisation et va diverger.

Il y a donc deux raisons principales à l'augmentation du flux de phasestrophie (Eq.(4.4)), une instabilité qui va produire et amener à la croissance de filaments ou une synchronisation des modes de bases du plasma. Ces deux phénomènes se produisent lors de deux phases différentes. En règle générale, on remarque que le flux de phasestrophie est lié à un transfert de quantité de mouvement dans le système [48, 126].

Plusieurs articles ont déjà utilisé le modèle de Kuramoto en physique des plasmas [48, 130, 131, 132], notamment pour décrire des phénomènes liés à la synchronisation pour l'amortissement Landau ou pour les modes de Van Kampen [133]. En effet, le terme correspondant à la fréquence propre de l'oscillateur (ω_n) devient dans le cadre de l'équation de Vlasov $-\frac{\mathbf{k}\cdot\mathbf{p}}{m\gamma}$ (Eq.(4.10)). Dans sa version non relativiste et à une dimension de vitesse, il devient $-kv$, ce qui correspond à la fréquence des modes de Van Kampen. Un tableau comparatif entre Kuramoto et Vlasov est proposé dans Xu et al. [130].

4.1.4 Analyse du cas oblique 1010keV

On va maintenant analyser plus en détail les résultats obtenus pour la simulation avec $k_B T_1 = 50keV$ et $k_B T_2 = 1010keV$, pour une excitation initiale sur le mode oblique (1,1) (Eq.(4.2)). On a représenté pour ce cas les différentes énergies sur la figure 4.5. On observe bien qu'il y a un premier transfert entre énergie cinétique $\left(E_k = \frac{1}{L_x L_y}\int d^2x \int d^2p\ mc^2(\gamma - 1)f\right)$ et énergie magnétique ($E_m = \frac{\epsilon_0}{2L_x L_y}\int d^2x\ c^2 B_z^2$) et électrique ($E_e = \frac{\epsilon_0}{2L_x L_y}\int d^2x\ E_x^2 + E_y^2$) qui commence aux alentours de $t\omega_p \simeq 50$ et un second transfert d'énergie vers $t\omega_p \simeq 280$.

Sur le premier transfert, on observe un gain en énergie électrique, presque identique à celui en énergie magnétique, de l'ordre de $E \simeq 0,002$. Dans le second transfert, aux alentours de $t\omega_p \simeq 280$, les énergies mises en jeu sont environ dix fois plus importantes que dans le premier transfert (figure 4.5). À l'inverse du premier, ce transfert se fait principalement de l'énergie cinétique vers magnétique ce qui est typique de la CFI.

Ces transferts d'énergie sont aussi accompagnés d'un chauffage. En effet, toujours sur la figure 4.5, on a représenté la moyenne spatiale des composantes $\tilde{\Pi}_{xx}$ et $\tilde{\Pi}_{yy}$ du

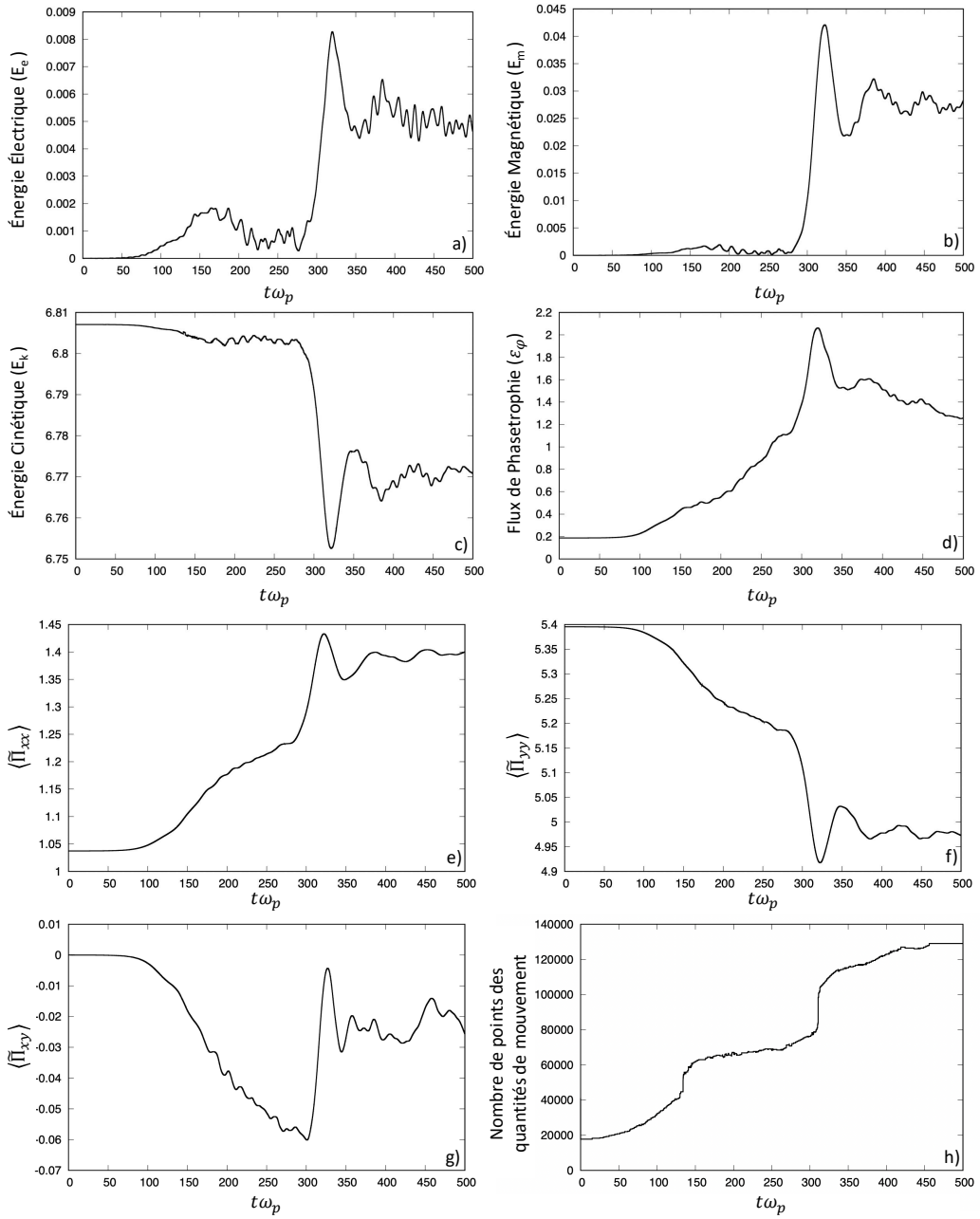

FIGURE 4.5 – Évolution temporelle des énergies électrique E_e (a), magnétique E_m (b) et cinétique E_k (c). Évolution temporelle du flux de phasestrophie ϵ_φ(d), de la moyenne des composantes $\tilde{\Pi}_{xx}$ (e), $\tilde{\Pi}_{yy}$ (f) et $\tilde{\Pi}_{xy}$ (g) du tenseur matériel de contrainte de densité $\tilde{\mathbf{\Pi}}$. Évolution temporelle du nombre de points des quantités de mouvement avec la méthode de grille emboîtée (h).

tenseur matériel de contrainte de densité que l'on peut écrire comme :

$$\tilde{\Pi}_{ij} = \iint \frac{p_i p_j}{m\gamma(x,y,p_x,p_y)} f(x,y,p_x,p_y) dp_x dp_y. \tag{4.14}$$

Ces composantes du tenseur permettent de décrire le chauffage dans leurs directions respective des quantités de mouvement. On a aussi représenté le terme hors diagonale $\tilde{\Pi}_{xy}$ qui va fournir une indication sur la présence d'anisotropie, même si cette dernière sera discutée plus en détail par la suite.

Il est important de mentionner que ce tenseur est différent du tenseur de pression déjà présenté dans la sous-partie 1.2.4 et que l'on peut écrire dans un cadre non relativiste comme $\Pi = nm < (\mathbf{v}-\mathbf{u})(\mathbf{v}-\mathbf{u}) >$ avec $<>$ la moyenne dans l'espace. En effet, contrairement au tenseur de pression qui prend en compte l'énergie interne du système, c'est-à-dire les effets thermiques, le tenseur matériel de contrainte de densité prend en compte l'énergie interne relativiste du système et aussi l'énergie cinétique associée au mouvement.

Si on observe les variations de $< \tilde{\Pi}_{xx} >$ et de $< \tilde{\Pi}_{yy} >$ (partie (e) et (f) de la figure 4.5), on observe bien qu'il y a un chauffage dans la direction p_x et un refroidissement dans la direction p_y. Si on se rappelle que la direction p_y est la direction de propagation des faisceaux, et donc de l'anisotropie, cela veut dire que le système cherche bien à réguler cette anisotropie. On observe par ailleurs que ce transfert d'énergie se fait aussi en deux temps correspondant respectivement à l'OI puis à la CFI. Si on s'intéresse maintenant à $< \tilde{\Pi}_{xy} >$, on voit qu'il varie avec des valeurs beaucoup plus faibles que les variations de $< \tilde{\Pi}_{xx} >$ et $< \tilde{\Pi}_{yy} >$. En effet, il atteint son minimum aux alentours de $t\omega_p \simeq 300$ pour une valeur de $| < \tilde{\Pi}_{xy} > | \simeq 0,06$ contre une variation des deux autres composantes du tenseur de l'ordre de $0,47$. La variable $< \tilde{\Pi}_{xy} >$ montre aussi que l'instabilité oblique est beaucoup plus anisotrope que la CFI et cela sera aussi observé par la suite.

Le flux de phasestrophie (partie (d) de la figure 4.5) quand à lui montre bien les échanges de quantité de mouvement qui ont lieu entre les éléments du tenseur matériel de contrainte de densité. Cela montre aussi le rôle des filaments dans ces transferts de quantité de mouvement (voir figure 4.6). Le maximum du flux de phasestrophie est atteint aux alentours de $t\omega_p \simeq 320$, ce qui correspond au maximum de l'énergie magnétique et électrique et aussi de $< \tilde{\Pi}_{xx} >$, ainsi qu'au minimum de l'énergie cinétique et de $< \tilde{\Pi}_{yy} >$. Ce maximum décrit donc la CFI. On peut par ailleurs remarquer que contrairement aux résultats de Ghizzo et al. [48], il ne décroît que peu après avoir atteint le maximum. Une explication possible est que dans sa phase non linéaire, le système continue de produire des filaments et donc la phasestrophie ne peut pas être nulle.

On a vu que le paramètre $< \tilde{\Pi}_{xy} >$ nous indique qu'il y a une anisotropie, et donc d'une "brisure de symétrie", dans l'espace des quantités de mouvement du système

principalement lorsque l'OI domine. On choisit donc de représenter la fonction de distribution suivant p_x et p_y sur la figure 4.6 avec à gauche la représentation montrant les différents niveaux de grille et à droite une interpolation des résultats de la grille la plus raffinée afin de voir plus précisément les fluctuations.

L'une des premières structures que l'on perçoit sur la fonction de distribution est l'apparition vers $t\omega_p \simeq 120$ d'une direction privilégiée d'accélération qui vient briser la symétrie (voir figure 4.6). Cette dernière apparaît pour le faisceau chaud mais aussi, dans les p_y négatifs, pour le faisceau froid. Même si ici l'image de référence de la figure 4.6 est pour un point fixe en $i_x = N_x/2$ et $i_y = N_y/2$, cette accélération apparaît dans tout l'espace (x,y) avec un angle assez similaire. Pour étudier plus en détail cette brisure de symétrie et cette accélération, on a représenté sur la figure 4.7 la fonction de distribution en $p_x - p_y$ moyennée sur l'espace pour différentes températures. Le cas que nous sommes en train d'étudier représente la partie (b) de cette figure. Tous les cas possèdent la même excitation initiale sur le mode oblique (1,1) avec la même taille de boîte en espace, il y a cependant le cas (d) qui possède en plus une excitation sur le mode $(1, -1)$ afin d'avoir une excitation symétrique par rapport à x et le cas (e) qui a une excitation uniquement sur le mode $(1, -1)$. On remarque justement que pour ce cas (d), on arrive à retrouver la symétrie du problème et que lorsque l'on excite uniquement le mode (1,−1) sur le cas (e), on retrouve la même anisotropie mais vers les p_x positifs. En sachant que le mode (1,1) correspond à exciter le vecteur mode de vecteur d'onde $\mathbf{k}d_e = (k_x d_e = 0,4, k_y d_e = 0,3)$, on remarque que cette accélération se fait dans une direction orthogonale à ce vecteur d'onde, c'est-à-dire dans la direction $\mathbf{k}^* d_e = (k_x d_e = -0,3, k_y d_e = 0,4)$ et, pour le cas (d) et (e), aussi dans la direction $\mathbf{k}^{**} d_e = (k_x d_e = 0,3, k_y d_e = 0,4)$. On a représenté cette direction k^* mais dans l'espace des quantités de mouvement en pointillés blancs sur la figure 4.7. On observe bien que pour les six cas, avec des températures différentes (figure 4.7), cette direction semble bien correspondre à la direction d'accélération des électrons. On note par ailleurs que cette accélération d'électrons est accompagnée d'une décélération d'électrons provenant d'une région avec un k_y plus grand. Cela reflète que cette accélération d'électrons cherche à réduire l'anisotropie du système en transférant de la quantité de mouvement de la direction p_y à p_x entraînant ainsi un chauffage dans la direction p_x.

Si on reprend la figure 4.6 pour $t\omega_p \simeq 260$, on observe que de plus en plus de filaments se forment aux alentours de cet angle k^*, principalement pour des angles plus faibles (c'est-à-dire plus proches de $(k_x d_e = -1, k_y d_e = 0)$). Cette production de filaments peut aussi être observée au travers du flux de phasestrophie qui est non nul (voir (d) sur la figure 4.5). Le couplage avec la rotation due à la force de Lorentz va amener ces filaments à changer d'angle en fonction de l'intensité du champ magnétique et donc de leur position dans l'espace. L'ensemble de ces deux processus (production de filaments et rotation) va conduire à un chauffage dans la direction p_x et à une augmentation de l'anisotropie entre les p_x positifs et négatifs comme le montre respectivement $< \tilde{\Pi}_{xx} >$

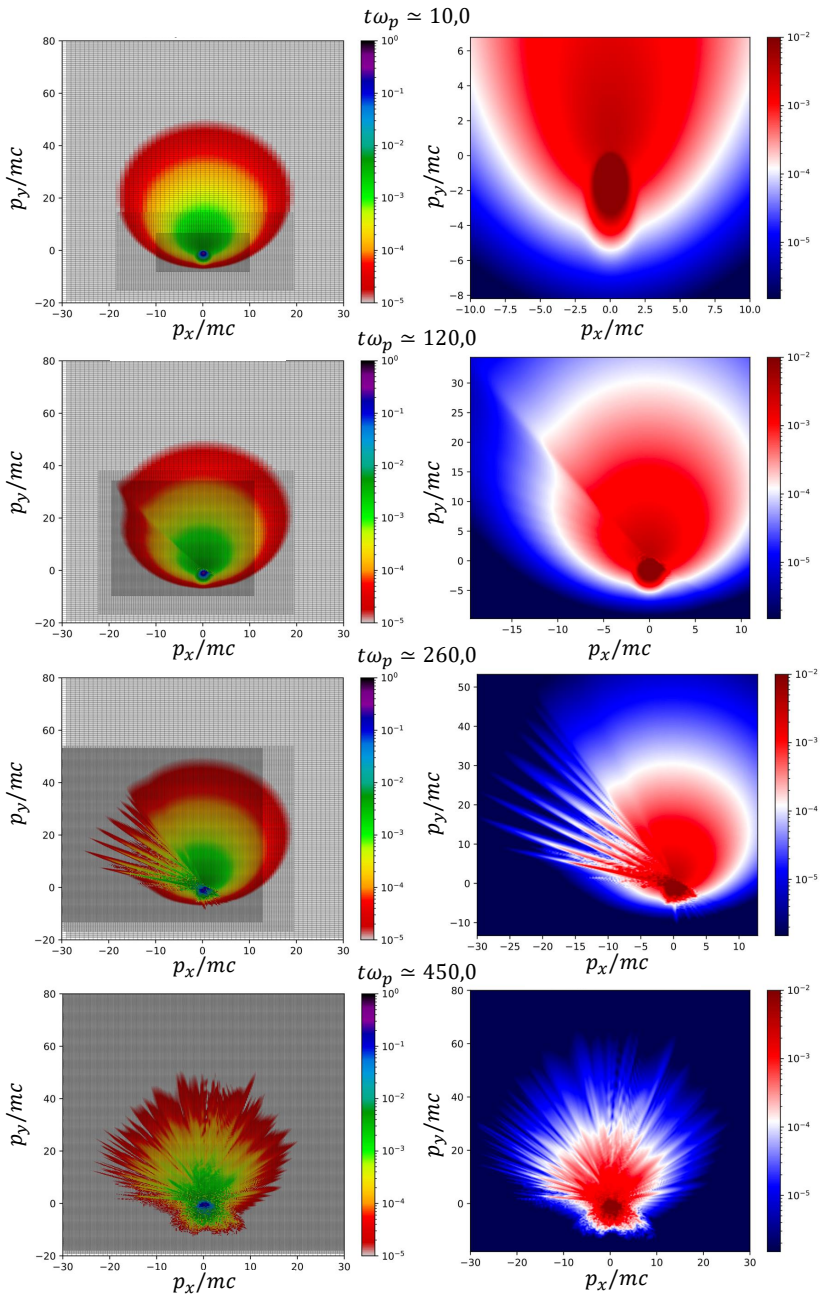

FIGURE 4.6 – Représentation de la fonction de distribution en $p_x - p_y$ pour VLEM_EG2D2V avec $\epsilon = 10^{-7}$ aux temps $t\omega_p \simeq 10$, 120, 260 et 450 pour un point localisé en $(i_x = N_x/2, i_y = N_y/2)$ (à gauche) avec un zoom sur la grille la plus raffinée $R = R^{max} = 2$ (à droite).

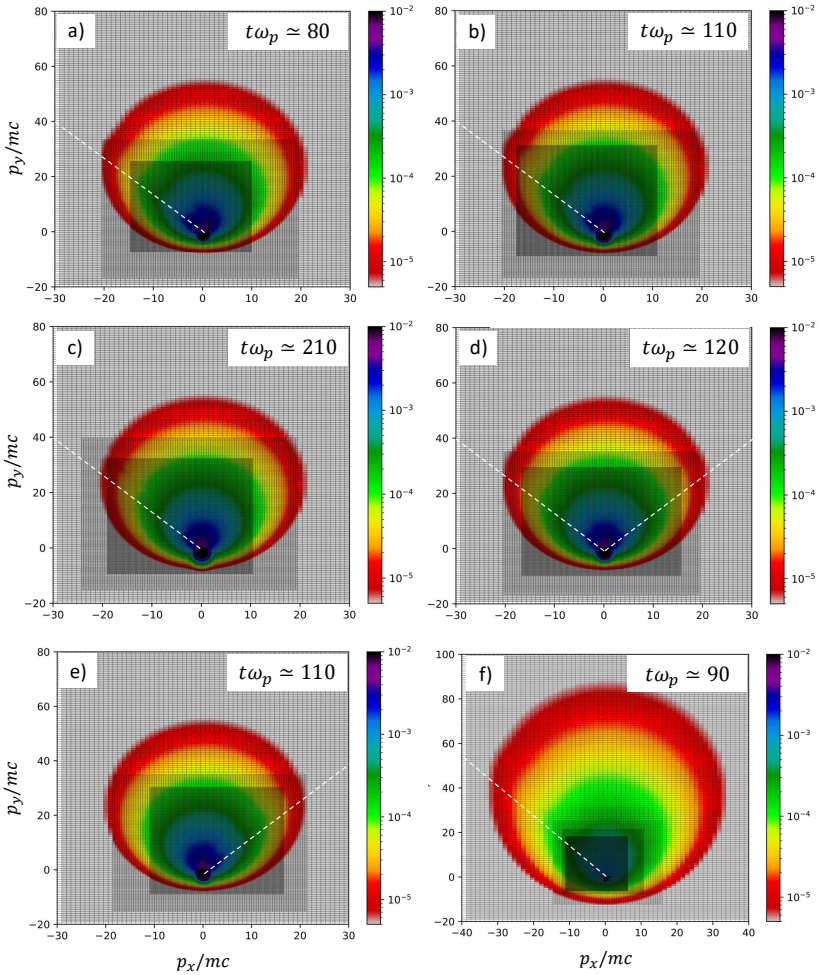

FIGURE 4.7 – Représentation de la fonction de distribution en $p_x - p_y$ moyennée sur (x, y) pour VLEM_EG2D2V. En pointillé blanc sur chaque figure est représenté l'angle orthogonal à $\mathbf{k}$ rapporté dans l'espace des quantités de mouvement. Les cas (a), (b) et (c) correspondent à une perturbation sur le mode oblique (1, 1) avec $k_B T_2 = 1010keV$ et respectivement $k_B T_1 = 32keV$, $k_B T_1 = 50keV$ et $k_B T_1 = 71,5keV$. Le cas (d) correspond à une perturbation sur les modes obliques (1, 1) et (1, −1) avec $k_B T_2 = 1010keV$ et $k_B T_1 = 50keV$. Le cas (e) correspond à une perturbation sur le mode oblique (1, −1) avec $k_B T_2 = 1010keV$ et $k_B T_1 = 50keV$, il est symétrique par rapport au cas (b). Le cas (f) correspond à une perturbation sur le mode oblique (1, 1) avec $k_B T_2 = 2000keV$ et $k_B T_1 = 5keV$, ce cas sera étudié en détail dans le sous-chapitre 4.2. On observe que pour les six cas, cet angle correspond à la direction d'accélération des électrons avec une décélération des électrons ayant un p_y plus important.

et $< \tilde{\Pi}_{xy} >$. Ce dernier atteint par ailleurs sa fluctuation maximale juste avant que la CFI devienne l'instabilité dominante vers $t\omega_p \simeq 300$.

Ensuite, la CFI (1,0) devient le mode dominant et les filaments commencent à se former dans toutes les directions de l'espace des quantités de mouvement (figure 4.6 à $t\omega_p \simeq 450$). Cela va avoir pour effet de chauffer encore plus la direction p_x et aussi de réduire l'anisotropie entre les p_x positifs et négatifs (voir $< \tilde{\Pi}_{xx} >$ et $< \tilde{\Pi}_{xy} >$ de la figure 4.5). On peut aussi noter qu'à la fin, le faisceau froid semble avoir chauffé et le faisceau chaud s'être refroidi même si les deux sont maintenant mélangés.

4.2 Étude de l'instabilité oblique relativiste à 2000keV

Dans le sous-chapitre précèdent, nous nous étions limités à un faisceau chaud ayant une température $k_B T_2 = 1010keV$ et à des faisceaux froids avec des températures comprises entre $k_B T_1 = 32keV$ et $k_B T_1 = 71,5keV$. Ces valeurs de températures avaient pour but de justifier la transition entre OI et CFI obtenue dans la sous-partie 3.4.3 en partant des simulations numériques réalisées dans Ghizzo et al. [120].

Dans ce sous-chapitre, on va désormais étudier le cas où la différence entre la température des faisceaux est beaucoup plus importante. On choisit une température de $k_B T_1 = 5keV$ pour le faisceau froid et pour le faisceau chaud relativiste $k_B T_2 = 2MeV$. Le but est de pouvoir comparer exactement avec les résultats théoriques obtenus dans l'article de Bret et al. [66] que l'on a repris sur la figure 4.8. Pour cela, on reprend $\beta_1 = -\beta_2 = -0,745$ et $n_1 = n_2 = n_0/2$. On garde la même taille normalisée de boîte en espace, c'est-à-dire $L_x = 15,707 d_e$ et $L_y = 20,943 d_e$, ce qui correspond à $\Delta k_x d_e = 0,40$, $\Delta k_y d_e = 0,30$. La perturbation initiale est appliquée de la même manière que dans l'équation (4.2), sur B_z, avec $e\delta B_0/(m\omega_p) = 0,005$ et $(n_x, n_y) = (1,1)$. On garde $\Delta t\omega_p = 0,01$ mais on va maintenant prendre 36 000 pas de temps afin de réduire le temps de calcul de la simulation. Pour la grille spatiale, on a toujours $N_x N_y = 256^2$ sur 2048 processus (256 processus MPI avec 8 taches OpenMP).

Contrairement à la sous-partie précédente, on va rajouter une précision plus importante dans l'espace des quantités de mouvement afin de décrire correctement le faisceau froid. En effet, ce dernier étant désormais de $5keV$, il va falloir une précision plus importante pour mieux le décrire que le faisceau de $50keV$ précédemment. De plus, le faisceau chaud étant maintenant de $2000keV$ contre $1010keV$ avant, il va falloir augmenter la taille de la boîte. La méthode de grilles emboîtées est donc parfaitement adaptée à ce type de situation.

On choisit quatre niveaux de raffinement, tel que $R^{max} = 3$, avec $\epsilon = 10^{-7}$ dans les équations (Eqs.(3.3) et (3.4)). Le nombre de points maximum pour la grille la plus raffinée est égal à $N_{p_x} = 505$ et $N_{p_y} = 1017$ avec comme limite de boîte des quantités de mouvement $p_{x_{min}}/p_{x_{max}} = -40mc/40mc$ et $p_{y_{min}}/p_{y_{max}} = -20mc/100mc$. On peut observer sur la figure 4.9 que ces paramètres sont suffisants pour décrire le

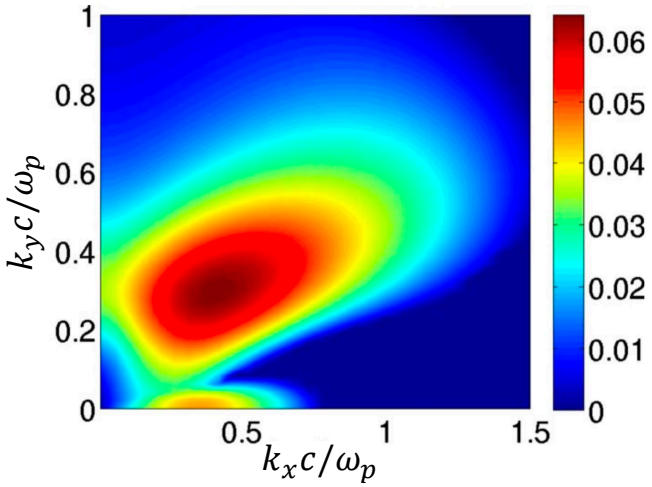

FIGURE 4.8 – Représentation du taux de croissance des instabilités CFI ($k_y = 0$), TSI ($k_x = 0$) et OI ($k_x \neq 0$ et $k_y \neq 0$) pour un système de deux faisceaux relativistes décrits par une distribution de Maxwell-Jüttner se propageant suivant y dans des directions opposées. L'image est issue de Bret et al. [66] pour un cas avec $k_B T_1 = 5keV$, $k_B T_2 = 2MeV$, $n_1 = n_2 = n_0/2$ et $\beta_1 = -\beta_2 = -0,745$.

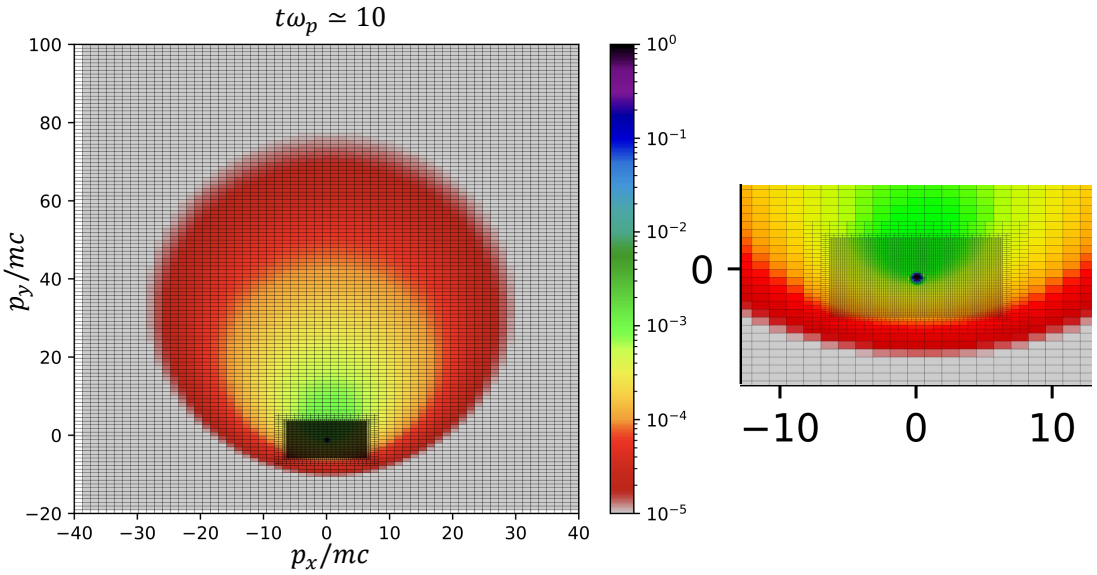

FIGURE 4.9 – Représentation de la fonction de distribution en $p_x - p_y$ pour VLEM_EG2D2V avec $\epsilon = 10^{-7}$ au temps $t\omega_p \simeq 10$ pour un point localisé en ($i_x = N_x/2, i_y = N_y/2$) (à gauche) avec un zoom sur la partie représentant le faisceau froid afin de montrer comment les grilles emboîtées décrivent le problème (à droite).

faisceau froid avec une dizaine de points pour les temps initiaux. Ce type de simulation est idéal pour être décrit par un raffinement de maillage en vitesse. On estime par ailleurs à la fin de la simulation 72% de gain en temps de calcul et jusqu'à 89% si on s'était arrêté à 18000 pas de temps, ce qui aurait été nécessaire pour voir la première saturation de l'instabilité.

4.2.1 Transition OI vers CFI

Tout comme dans la sous-partie 3.4.3 et dans le sous-chapitre précèdent, on va observer une transition de OI vers CFI dans la phase de saturation de l'OI (à gauche de la figure 4.10). On choisit donc de tracer sur la figure 4.10 (à gauche pour ce cas) l'intensité des modes les plus instables sur $\hat{B}_z$. Le taux de croissance mesuré à gauche sur la figure 4.10 pour l'OI est de $\Gamma_{11}^{num}/\omega_p = 0,060$ ce qui est assez proche du taux de croissance théorique $\Gamma_{11}^{theo}/\omega_p = 0,065$ (voir figure 4.8). De la même manière que dans le sous-chapitre précédent, l'OI semble avoir rapidement atteint sa saturation mais un taux de croissance est tout de même mesurable ici.

Pour la CFI, on observe un taux de croissance de $\Gamma_{10}^{num}/\omega_p = 0,055$ entre $t\omega_p \simeq 0$ et 150, puis $\Gamma_{10}^{num}/\omega_p = 0,078$ entre $t\omega_p \simeq 150$ et 250 et $\Gamma_{10}^{num}/\omega_p = 0,299$ entre $t\omega_p \simeq 250$ et 280. Tous ces taux de croissance sont supérieurs à celui théorique $\Gamma_{10}^{theo}/\omega_p = 0,045$ (voir figure 4.8), mais celui entre $t\omega_p \simeq 0$ et 150, qui est donc le moins impacté par la saturation de l'OI, a une valeur plus proche de la valeur théorique et plus faible que le taux de croissance de l'OI, ce qui correspond aux résultats théoriques de la figure 4.8.

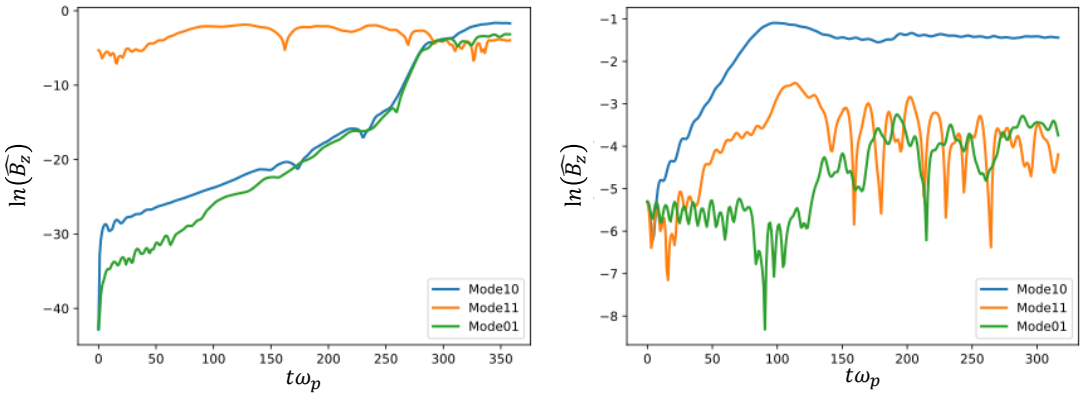

FIGURE 4.10 – Évolution de l'intensité du champ magnétique $\hat{B}_z(n_x\Delta k_x, n_y\Delta k_y)$ pour les modes (n_x,n_y)= (1,0), (1,1) et (0,1) correspondant respectivement aux instabilités CFI, OI et TSI. Comparaison entre une perturbation initiale seulement sur le mode (1,1) (à gauche) et sur les trois modes (1,0), (1,1) et (0,1) (à droite).

Le fait qu'il y ait trois taux de croissance différents interroge à nouveau sur le rôle de la saturation de l'OI dans la transition vers CFI. C'est pourquoi, de la même manière que dans la sous-partie 4.1.1 on va venir exciter les trois instabilités OI, CFI et TSI (Eq.(4.3)) afin de comparer hors des effets non linéaires les taux de croissance.

On représente à droite de la figure 4.10 les taux de croissance lorsque l'on perturbe ces trois instabilités. On trouve maintenant un taux de croissance $\Gamma_{10}^{num}/\omega_p = 0,048$ pour la CFI, ce qui est assez proche du taux théorique. Cependant pour l'OI, le taux de croissance $\Gamma_{11}^{num}/\omega_p = 0,035$ est plus faible que le taux théorique mais aussi plus faible que celui de la CFI. L'instabilité CFI est aussi celle qui sature avec une intensité plus élevée. On en déduit donc que, expérimentalement, la CFI est la plus instable pour ces paramètres.

Cette différence entre résultats numériques et théorique [66] (figure 4.8) peut être due à une perturbation initiale trop forte. Pour valider ces derniers, il faudrait soit relancer une expérience numérique sans perturbation initiale soit utiliser un code PIC.

4.2.2 Évolution des énergies et de la fonction de distribution

Dans cette sous-partie, on va s'intéresser à l'évolution des énergies et de la fonction de distribution pour le cas où on excite les trois modes (1,1), (1,0) et (0,1) correspondant respectivement à l'OI, la CFI et la TSI. En effet l'étude faite dans la sous-partie 4.1.3 pour l'excitation de l'OI est directement rapportable à notre cas, comme le montre la figure 4.7 sur le cas (f).

On a représenté l'évolution des énergies, du flux de phasestrophie, des paramètres moyenné du tenseur $\tilde{\Pi}$ et du nombre de points des quantités de mouvement sur la figure 4.11. A l'inverse de la figure 4.5 présentée dans la sous-partie 4.1.3 désormais on observe qu'un seul échange entre énergie cinétique et énergie magnétique et électrique qui atteint son maximum aux alentours de $t\omega_p \simeq 100$ correspondant au moment où la CFI sature (voir figure 4.10 à droite). Il y a ensuite un second échange d'énergie, de l'énergie électrique et magnétique en énergie cinétique. Ce dernier est assez important puisque la moitié de l'énergie électrique et magnétique gagnée va être reconvertie en énergie cinétique. Si on observe maintenant le flux de phasestrophie, on voit qu'il a augmenté de manière presque linaire jusqu'à $t\omega_p \simeq 105$ (première ligne pointillée sur la figure 4.11) mais qu'il atteint son maximum ensuite pour $t\omega_p \simeq 125$ (deuxième ligne pointillée sur la figure 4.11), soit au milieu du second transfert de l'énergie électromagnétique en cinétique. Cette première augmentation du flux de phasestrophie va correspondre à un transfert de quantité de mouvement de la direction p_y à la direction p_x. Cela va entraîner un chauffage dans la direction p_x que l'on perçoit sur les composantes $< \tilde{\Pi}_{xx} >$ et $< \tilde{\Pi}_{yy} >$ du tenseur matériel de contrainte de densité. Cependant, la deuxième augmentation du flux de phasestrophie n'est pas due à la CFI et elle peut être due à de la synchronisation des modes de Van Kampen du plasma. Cette synchronisation pourrait aussi être la cause de cet échange secondaire d'énergie.

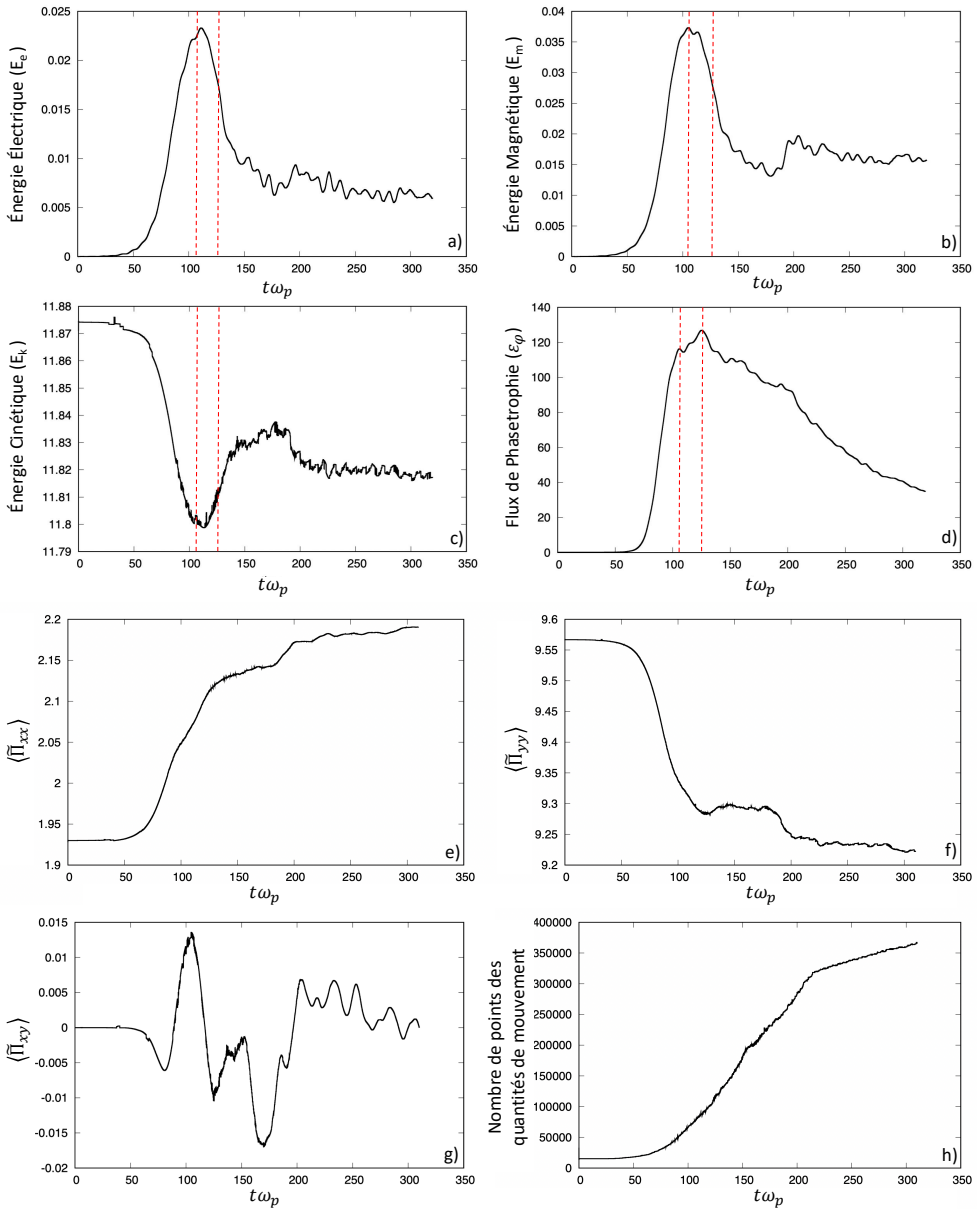

FIGURE 4.11 – Évolution temporelle des énergies électrique E_e (a), magnétique E_m (b) et cinétique E_k (c). Évolution temporelle du flux de phasestrophie ϵ_φ(d), de la moyenne des composantes $\tilde{\Pi}_{xx}$ (e), $\tilde{\Pi}_{yy}$ (f) et $\tilde{\Pi}_{xy}$ (g) du tenseur matériel de contrainte de densité $\tilde{\mathbf{\Pi}}$. Évolution temporelle du nombre de points des quantités de mouvement avec la méthode de grille emboîtée (h). Les lignes pointillées rouges sur (a), (b), (c) et (d) représentent les temps $t\omega_p \simeq 105$ et $t\omega_p \simeq 125$.

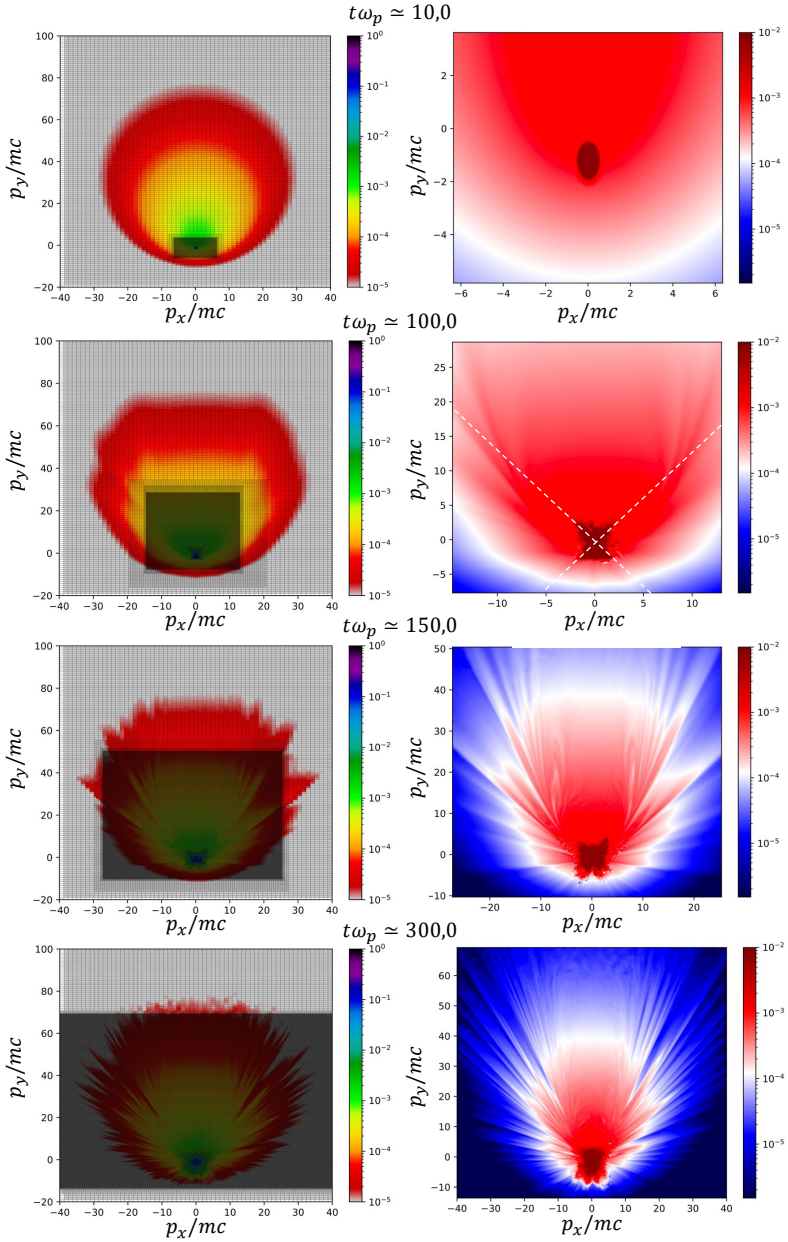

FIGURE 4.12 – Représentation de la fonction de distribution en $p_x - p_y$ pour VLEM_EG2D2V avec $\epsilon = 10^{-7}$ aux temps $t\omega_p \simeq 10$, 100, 150 et 300 pour un point localisé en $(i_x = N_x/2, i_y = N_y/2)$ (à gauche) avec un zoom sur la grille la plus raffinée $R = R^{max} = 3$ (à droite). Les pointillé blanc au temps $t\omega_p \simeq 100$ correspondent à $\mathbf{k}^* d_e = (k_x d_e = -0,3, k_y d_e = 0,4)$ et $\mathbf{k}^{**} d_e = (k_x d_e = 0,3, k_y d_e = 0,4)$ les directions orthogonales à la perturbation du mode oblique (1,1) et (1,−1).

On observe aussi un troisième transfert d'énergie aux alentours de $t\omega_p \simeq 200$, encore plus faible que les deux précèdents, cette fois-ci principalement entre énergie cinétique et énergie magnétique. Ce transfert conduit à un chauffage supplémentaire dans la direction p_x ($< \tilde{\Pi}_{xx} >$) et à un refroidissement dans la direction p_y ($< \tilde{\Pi}_{yy} >$). Quand à l'anisotropie, elle semble assez faible puisque $< \tilde{\Pi}_{xy} >$ varie sur des ordres de grandeurs dix à vingt fois plus faibles que $< \tilde{\Pi}_{xx} >$ et $< \tilde{\Pi}_{yy} >$.

Si on s'intéresse maintenant aux variations de la fonction de distribution dans l'espace $p_x - p_y$ représentée sur la figure 4.12, on peut retrouver le chauffage dans la direction p_x. En effet, dès $t\omega_p \simeq 100$, on observe que la fonction de distribution s'étend dans la direction des p_x et ce de manière symétrique, justifiant ainsi l'augmentation de la composante $< \tilde{\Pi}_{xx} >$ et les faibles variations de $< \tilde{\Pi}_{xy} >$. Ce chauffage symétrique est une signature de la CFI. On observe aussi, comme ce qui avait pu être déjà décrit dans la sous-partie 4.1.3, que le faisceau froid chauffe. En effet si on observe la différence de taille du faisceau froid d'intensité 10^{-2} entre $t\omega_p \simeq 10$ et $t\omega_p \simeq 100$, on voit qu'il s'est élargi. On observe par ailleurs qu'il prend une forme de X dont la direction des branches correspond au k^* et k^{**} de la sous-partie 4.1.3. Ce chauffage dans ces directions particulières pourraient être une preuve de l'OI dans l'excitation initiale, mais comme seul le vecteur d'onde correspondant au mode (1,1) à été excité cela ne justifierait pas la présence de k^{**}. On peut donc en déduire que la CFI possède les mêmes directions privilégiés que l'OI. La présence de l'OI dans l'excitation initiale se ressent cependant puisque l'accélération dans la direction k^* semble être plus importante que dans la direction k^{**}. On peut percevoir cela sur la fonction de distribution froide ou directement sur les limites de la grille de niveau $R = R^{max} = 3$. En effet, elle est un peu plus étendue vers la gauche avec un début de boîte en $p_x/mc = -14$ que sur la droite, avec une fin de boîte en $p_x/mc = 13$.

4.3 Étude de l'instabilité Weibel relativiste

Pour l'instant, la fonction de distribution que l'on a utilisée pour une distribution relativiste était la fonction de distribution de Maxwell-Jüttner décalée (Eq.(4.1)) [119]. Elle est bien adaptée afin de décrire une distribution relativiste pour un plasma isotrope [134]. Cependant elle a l'inconvénient de ne pas permettre d'avoir d'anisotropie de température [135]. Elle ne permet donc pas d'étudier des instabilités Weibel, c'est pourquoi jusqu'à présent on a étudié uniquement des instabilités avec deux faisceaux.

De plus, pour certains cas astrophysiques dans lesquels on peut retrouver les instabilités Weibel, CFI, OI et TSI (voir sous-partie 1.6.4), on ne peut pas mesurer la fonction de distribution de ces particules. C'est pourquoi, on considère qu'il existe plusieurs fonctions de distribution qui peuvent être appropriées [134, 135].

4.3.1 Comparaison de trois fonctions de distributions

Afin d'étudier les instabilités Weibel pures, c'est-à-dire dues à une anisotropie de température, il faut choisir une autre fonction de distribution que la Maxwell-Jüttner. On s'est donc intéressé à trois fonctions de distribution qui permettent cela : la bi-Maxwellienne, la fonction Kappa et la bi-Maxwellienne modifiée. Ces dernières ont déjà été étudiées dans plusieurs articles [65, 135] et nous allons les expliciter ici.

La première fonction de distribution est la fonction bi-Maxwellienne qui s'écrit comme :

$$f_{BM}(p_x, p_y) = A\exp\left(-\mu\left((\gamma(p_x, p_y) + \Psi p_x^2\right)\right), \tag{4.15}$$

avec $\mu = mc^2/(k_B T_y)$ le paramètre qui caractérise la température dans la direction y (perpendiculaire) du plasma, $\Psi = \left(\frac{T_y}{T_x} - 1\right)$ le paramètre qui caractérise l'anisotropie, γ le facteur relativiste et A le facteur de normalisation. Dans certaines références [135, 65], les variables γ, T_y et T_x sont appelées respectivement E, $T_\perp$ et $T_{//}$. Les directions parallèle $//$ et perpendiculaire $\perp$ se réfèrent au vecteur d'onde $\mathbf{k}$ qui pour l'instabilité Weibel est orthogonal à la température la plus élevée. Dans notre cas, afin de faire des analogies plus simples avec la CFI étudiée dans les deux sous-chapitres précédents, on impose $\mathbf{k} = (k_x, 0)$.

La deuxième fonction de distribution que l'on va étudier est la fonction de Kappa qui s'écrit comme :

$$f_\kappa(p_x, p_y) = B\left(1 + \frac{p_x^2}{\kappa m \theta_x^2} + \frac{p_y^2}{\kappa m \theta_y^2}\right)^{-\kappa+1}, \tag{4.16}$$

avec $\theta_\alpha^2 = \frac{k_B T_\alpha (2\kappa - 3)}{m\kappa}$ la vitesse thermique au carré et B un facteur de normalisation. Cette fonction impose la condition $\kappa \in]3/2, \infty[$.

Cette fonction Kappa peut aussi être réécrite sous forme exponentielle dans la limite $\kappa \to \infty$ [65]. On appellera cette fonction bi-Maxwellienne modifiée et elle s'écrit comme :

$$f_{mBM}(p_x, p_y) = C\exp\left(-\mu\left(\gamma^2(p_x, p_y) + \Psi p_x^2\right)\right). \tag{4.17}$$

En comparant l'expression des fonctions bi-Maxwellienne (Eq.(4.15)) et bi-Maxwellienne modifiée (Eq.(4.17)), on remarque que la différence est sur la puissance du facteur relativiste dans l'exponentielle. Cela veut dire que pour un cas peu relativiste ($\gamma \simeq 1$), les deux fonctions devrait être assez similaires, mais dans un cas très relativiste, la bi-Maxwellienne modifiée tendra beaucoup plus rapidement vers zéro, ce qui signifie qu'elle s'étendra moins dans l'espace des phases.

4.3.2 Instabilité Weibel relativiste

On s'est donc intéressé à comparer ces trois fonctions de distribution. Pour cela, on s'est inspiré de l'article de Schaefer-Rolffs et al. [65] dans lequel les auteurs proposent

une résolution de la relation de dispersion dans les cas limites ultrarelativistes, peu anisotrope et très anisotrope pour ces trois fonctions de distribution. Le cas auquel nous avons décidé de nous intéresser est le cas ultrarelativiste.

Ce cas ultrarelativiste, présent sur le cas d de la figure 2 de l'article [65], fixe les paramètres $\Psi = 0,3$ et $\mu = \mu_\kappa = 10^{-2}$. On note que la valeur de Ψ choisie correspond à un cas peu anisotrope ($\Psi < 1$) et que pour les températures, en reprenant l'expression de μ, on a $k_B T_y = 51,1 MeV$ et $k_B T_x \simeq 39,3 MeV$, ce qui correspond à des températures très relativistes.

En reprenant les résultats de l'article [65], on peut retrouver des formules analytiques pour le taux de croissance de l'instabilité Weibel dans un cas très relativiste ($\gamma >> 1$). Pour le taux de croissance de l'instabilité Weibel pour la fonction bi-Maxwellienne dans la limite ultrarelativiste on a :

$$\frac{\Gamma^{theo}_{biMax}}{\omega_p} = \frac{2}{3\pi}k - \frac{1}{3\sqrt{\pi}}\sqrt{\frac{\mu}{\Psi}}k^3, \tag{4.18}$$

et pour la fonction Kappa et la bi-Maxwellienne modifiée, qui partagent toutes deux la même formule du taux de croissance, on a dans la limite ultrarelativiste de l'instabilité Weibel :

$$\frac{\Gamma^{theo}_{Kappa/mbMax}}{\omega_p} = \frac{2}{3\pi}k - \frac{4}{3\pi}(1+\Psi)^{-\frac{1}{2}}\left(k^3 + \frac{1}{2}k\right). \tag{4.19}$$

Pour le cas qui va nous intéresser ($\Psi = 0,3$ et $\mu = \mu_\kappa = 10^{-2}$) les fonctions bi-Maxwellienne modifiée et Kappa ont un taux de croissance environ dix fois plus faible que la bi-Maxwellienne. Le taux de croissance maximal est atteint pour la bi-Maxwellienne modifiée et la Kappa aux alentours de $k_x d_e \simeq 0,15$ pour une valeur $\Gamma^{theo}_{Kappa/mbMax}/\omega_p \simeq 0,003$. Ce qui correspond à un taux de croissance pour la bi-Maxwellienne de $\Gamma^{theo}_{biMax}/\omega_p \simeq 0,032$. On choisit donc une taille de boîte qui correspond à cette valeur de k_x, c'est-à-dire $L_x = L_y = \frac{2\pi}{k_x} = 41,888 d_e$. Pour les autres paramètres de simulations, on pose $N_x N_y = 256^2$ sur 2048 processus (256 processus MPI avec 8 taches OpenMP). On choisit $R = R^{max} = 1$ et un nombre de points maximum pour la grille raffinée égal à $N_{p_x} = 511$ et $N_{p_y} = 255$ avec comme limite de boîte des quantités de mouvement $p_{x_{min}}/p_{x_{max}} = -200mc/200mc$ et $p_{y_{min}}/p_{y_{max}} = -400mc/400mc$. On choisit $\Delta t\omega_p = 0,05$ sur 25000 pas de temps afin d'aller sur des très longs temps ($t\omega_p = 1250$). On a choisi d'observer le système jusqu'à $t\omega_p = 1250$ puisqu'on ne va pas perturber la simulation. En effet, on a pu voir dans les deux sous-chapitres précédents que la perturbation initiale peut conduire à modifier les résultats de la simulation. On profite donc d'un des avantages des codes Semi-Lagrangien qui est l'absence de bruit statistique (voir sous-chapitre 1.4), ce qui nous permet de démarrer les instabilités à l'erreur machine [123].

Pour la fonction bi-Maxwellienne modifiée et la fonction Kappa, l'instabilité n'a jamais démarré sur la durée observée, et ce, même en adaptant la taille de la boîte

des quantités de mouvement. Cela peut être dû au fait que même si on a choisi le maximum du taux de croissance, il reste très faible ($\Gamma^{Kappa/mbMax}/\omega_p \simeq 0,003$) et donc l'instabilité n'a pas le temps de croître jusqu'à des valeurs significatives.

Cependant pour la fonction bi-Maxwellienne, l'instabilité arrive à démarrer avec un taux de croissance mesuré de $\Gamma^{num}_{biMax}/\omega_p \simeq 0,064$ ce qui est deux fois plus important que le taux théorique $\Gamma^{theo}_{biMax}/\omega_p \simeq 0,032$. On note que le taux de croissance pour le champ E_x est égal à $\Gamma^{num}_{biMax}/\omega_p \simeq 0,128$ ce qui deux fois plus important que celui pour les champs B_z et E_y. Cela correspond bien aux résultats théoriques des instabilités Weibel [67].

Évolution de la fonction bi-Maxwellienne

On va maintenant s'intéresser à l'évolution des champs et des énergies de cette instabilité Weibel avec la fonction bi-Maxwellienne. De la même manière que dans les deux sous-chapitres précédents, on va d'abord analyser l'évolution des énergies, du flux de phasestrophie et des composantes du tenseur matériel de contrainte de densité représentés sur la figure 4.13. On observe que l'instabilité commence à croître à des valeurs significatives aux alentours de $t\omega_p \simeq 500$ et elle sature vers $t\omega_p \simeq 530$ avec un transfert d'énergie cinétique principalement vers l'énergie magnétique, ce qui est caractéristique de l'instabilité Weibel. La différence majeure avec les deux sous-chapitres précédents est la quantité d'énergie transférée en fonction de l'énergie cinétique totale. En effet, pour ce cas le transfert représente environ 5% de l'énergie cinétique initiale, ce qui est beaucoup plus que les 0,7% de la figure 4.5 du sous-chapitre 4.1 ou les 0,6% sur la figure 4.11 du sous-chapitre 4.2. On note aussi que presque la totalité de l'énergie électrique gagnée à été reconvertie en énergie cinétique (environ 80%).

On observe aussi sur la figure 4.13 que le transfert d'énergie interne se fait d'une manière identique à la CFI, c'est-à-dire qu'on observe un chauffage dans la direction p_x, via la composante $< \tilde{\Pi}_{xx} >$ du tenseur matériel de contrainte de densité qui augmente, avec un refroidissement dans la direction p_y, via une diminution de $< \tilde{\Pi}_{yy} >$. Ce chauffage se fait de manière très isotrope puisque la composante $< \tilde{\Pi}_{xy} >$ fluctue avec des valeurs beaucoup plus faibles que $< \tilde{\Pi}_{xx} >$ et $< \tilde{\Pi}_{yy} >$. Si on regarde le flux de phasestrophie, on voit que le premier pic aux alentours de $t\omega_p \simeq 530$ semble correspondre à la saturation de la CFI et à ce chauffage (première ligne pointillée sur la figure 4.13). Cependant, il y a aussi la présence d'une série de pics qui ne correspondent pas à la saturation de la CFI. Le second pic (deuxième ligne pointillée sur la figure 4.13) est intéressant puisqu'il a une intensité assez similaire à l'intensité du premier pic et il est situé à $t\omega_p \simeq 580$, soit au moment du second transfert d'énergie magnétique à cinétique. Ce dernier peut donc être interprété comme de la synchronisation de modes et vient appuyer l'impact de cette synchronisation sur le second transfert d'énergie de magnétique à cinétique.

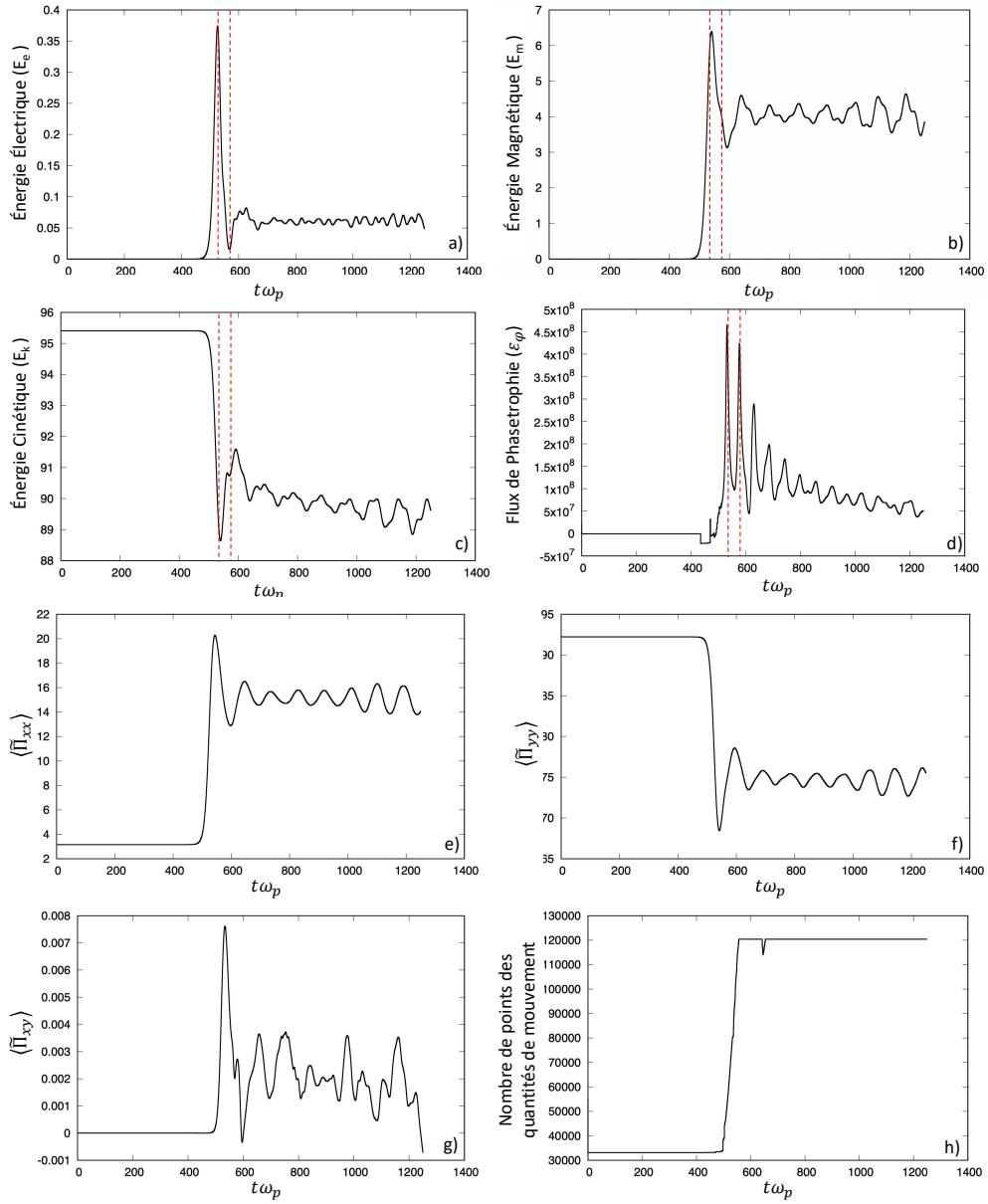

FIGURE 4.13 – Évolution temporelle des énergies électrique E_e (a), magnétique E_m (b) et cinétique E_k (c). Évolution temporelle du flux de phasestrophie ϵ_φ(d), de la moyenne des composantes $\tilde{\Pi}_{xx}$ (e), $\tilde{\Pi}_{yy}$ (f) et $\tilde{\Pi}_{xy}$ (g) du tenseur matériel de contrainte de densité $\tilde{\mathbf{\Pi}}$. Évolution temporelle du nombre de points des quantités de mouvement avec la méthode de grille emboîtée (h). Les lignes pointillées rouges sur (a), (b), (c) et (d) représentent les temps $t\omega_p \simeq 530$ et $t\omega_p \simeq 580$.

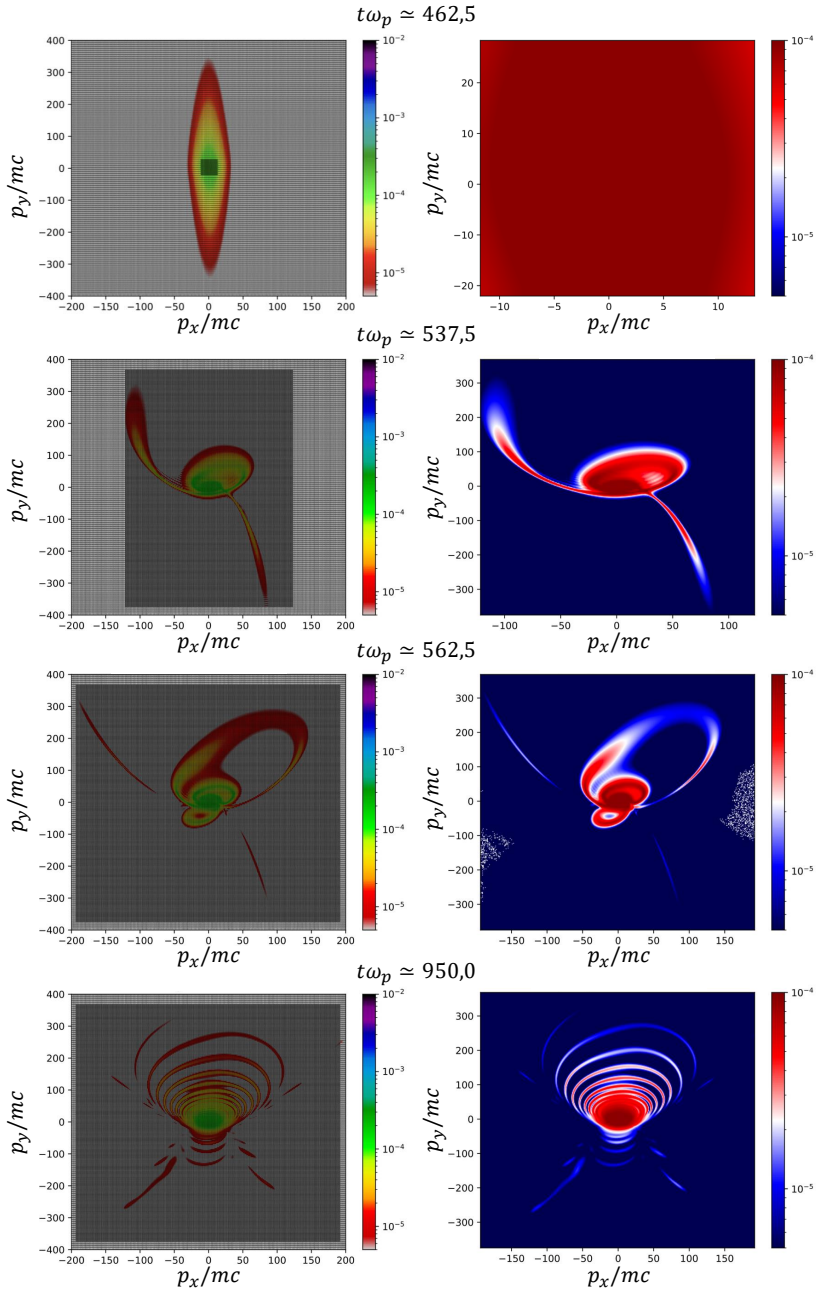

FIGURE 4.14 – Représentation de la fonction de distribution en $p_x - p_y$ pour VLEM_EG2D2V avec $\epsilon = 10^{-7}$ aux temps $t\omega_p \simeq 462{,}5$, 537,5 , 562,5 et 950,0 pour un point localisé en $(i_x = N_x/2, i_y = N_y/2)$ (à gauche) avec un zoom sur la grille la plus raffinée $R = R^{max} = 1$ (à droite).

Si on regarde maintenant l'évolution de la fonction de distribution dans l'espace des quantités de mouvement (p_x-p_y) sur la figure 4.14. On constate l'apparition de filaments accélérés sous la forme de bras. Cette accélération de particules se fait, pour le point situé en $i_x = N_x/2$ et $i_y = N_y/2$, uniquement dans la direction des p_y positifs. Cela est dû à sa localisation dans l'espace qui sera discutée par la suite en analysant la figure 4.16. En observant la figure 4.14 on s'aperçoit que cette accélération de particules continue jusqu'à la fin de la simulation et qu'elle se fait par vagues à fréquence régulière. La fréquence mesurée pour cette accélération est $\omega/\omega_p = 0,02$ ce qui correspond exactement à la fréquence des pics du flux de phasestrophie. On va essayer de comparer cela aux autres fréquences caractéristiques dans cette instabilité Weibel afin d'essayer d'apporter une explication à cette accélération.

On définit la fréquence de rebond magnétique comme [136] :

$$\frac{\omega_B}{\omega_p} = \sqrt{\frac{|\mathbf{k}|c}{\omega_p}\frac{p_\perp}{mc\gamma^2}\frac{eB_{max}}{m\omega_p}}, \tag{4.20}$$

avec $eB_{max}/m\omega_p$ le champ magnétique maximal. Pour notre simulation on a $\omega_B/\omega_p = 0,037$, ce qui est proche du double de la fréquence d'accélération des particules. On peut aussi regarder la fréquence de rebond électrostatique qui s'écrit comme [136] :

$$\frac{\omega_E}{\omega_p} = \sqrt{\frac{2|\mathbf{k}|c}{\omega_p}\frac{eE_{max}}{\gamma m\omega_p c}}, \tag{4.21}$$

$eE_{max}/m\omega_p c$ le module maximal du champ électrique $\left(E = \sqrt{E_x^2 + E_y^2}\right)$. Pour notre cas, on a $\omega_E/\omega_p = 0,037$ ce qui est identique à la fréquence de rebond magnétique et donc proche du double de la fréquence observée pour l'accélération de ces particules. Même si ces fréquences ne correspondent pas à la fréquence d'accélération des particules, on peut voir que le champ magnétique semble être fortement impliqué. En effet, sur la figure 4.14, vers la fin de la simulation ($t\omega_p \simeq 950$), on perçoit que la rotation due au champ magnétique semble avoir de l'importance puisque ces particules accélérées vont "tourner" dans l'espace p_x-p_y. Cette rotation justifie aussi la forme de "bras" pour l'accélération des premiers filaments, aux alentours de $t\omega_p \simeq 537,5$. De plus, si on compare maintenant l'intensité des champs, à l'aide de la figure 4.15, on remarque que le champ magnétique ($eB_{max}/m\omega_p \simeq 4$) est huit fois plus intense que le champ électrique ($eE_{max}/m\omega_p c \simeq 0,5$), cela était déjà observé sur les énergies sur la figure 4.13.

Si on observe la figure 4.15, on remarque que le champ B_z évolue sur le mode (1,0) et le champ E_x sur le mode (2,0). Le fait que le champ E_x soit sur un mode deux fois plus élevé que le champ magnétique est caractéristique de l'instabilité Weibel [67]. Cela est dû au fait que les électrons vont se localiser dans les régions de faible champ magnétique $B_z \simeq 0$ créant ainsi un surplus d'électrons qui va entraîner un champ électrique.

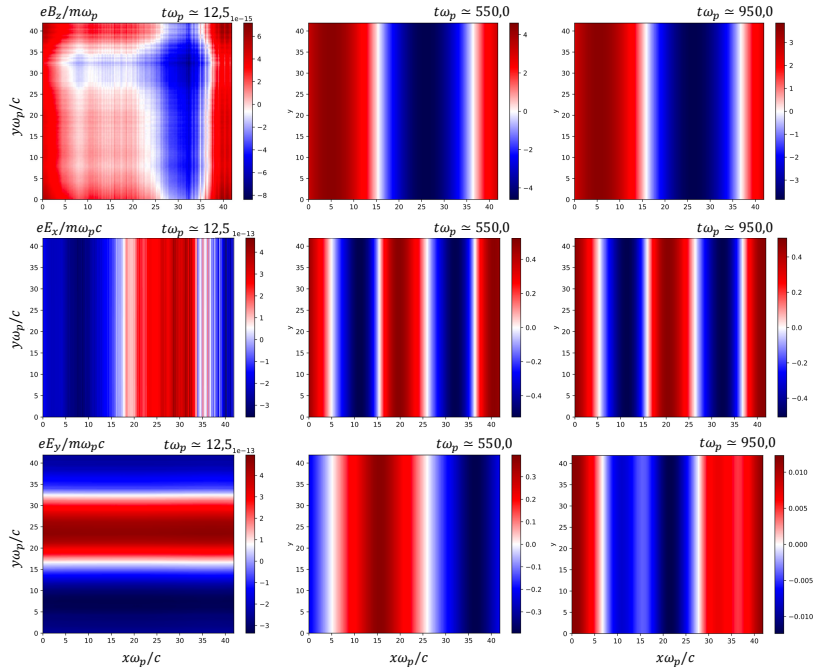

FIGURE 4.15 – Représentation de l'amplitude normalisée des champs B_z (haut), E_x (milieu) et E_y (bas) à $t\omega_p \simeq 12{,}5$, 550,0 et 950,0. Le champ E_y prend en général des valeurs plus faibles que E_x et il oscille dans le temps.

Si on observe le champ électrique E_y, on voit qu'il prend des valeurs en moyenne dix fois plus faibles que le champ E_x (voir figure 4.15). Par ailleurs, ce champ E_y va fluctuer dans le temps entre valeurs positives et négatives et la fréquence à laquelle il passe de positif à négatif est de $\omega/\omega_p = 0,02$. Ce qui semble correspondre avec la fréquence d'accélération des particules et de la variation du flux de phasestrophie (figure 4.13 et 4.14). Cependant, la fréquence entre deux maximums positifs est deux fois plus faible et est de $\omega/\omega_p = 0,01$. Ce champs E_y pourrait donc être lié à cette accélération de particules dans la direction des p_y et ce malgré ces faibles valeurs. Cependant, au vu de l'intensité et de la forme de l'accélération des particules en forme de bras, le champ magnétique semble être la raison principale à cette accélération.

Si on s'intéresse maintenant à l'évolution de la fonction de distribution suivant x, on peut tracer $f(x, p_x)$ (à droite) et $f(x, p_y)$ (à gauche) sur la figure 4.16. On observe que la fonction de distribution s'est séparée, en fonction du signe de p_y, spatialement suivant x en localisant le maximum de la fonction de distribution aux alentours des minimums du champ magnétique vers $x\omega_p/c \simeq 15,5$ et $x\omega_p/c \simeq 36,5$ (voir figure 4.15). Ce type de structure est typique de l'instabilité Weibel et du piégeage magnétique [67], on la retrouve aussi pour CFI, avec la séparation spatiale des deux faisceaux. Les particules les plus énergétiques vont pouvoir se déplacer librement suivant x puisqu'elles

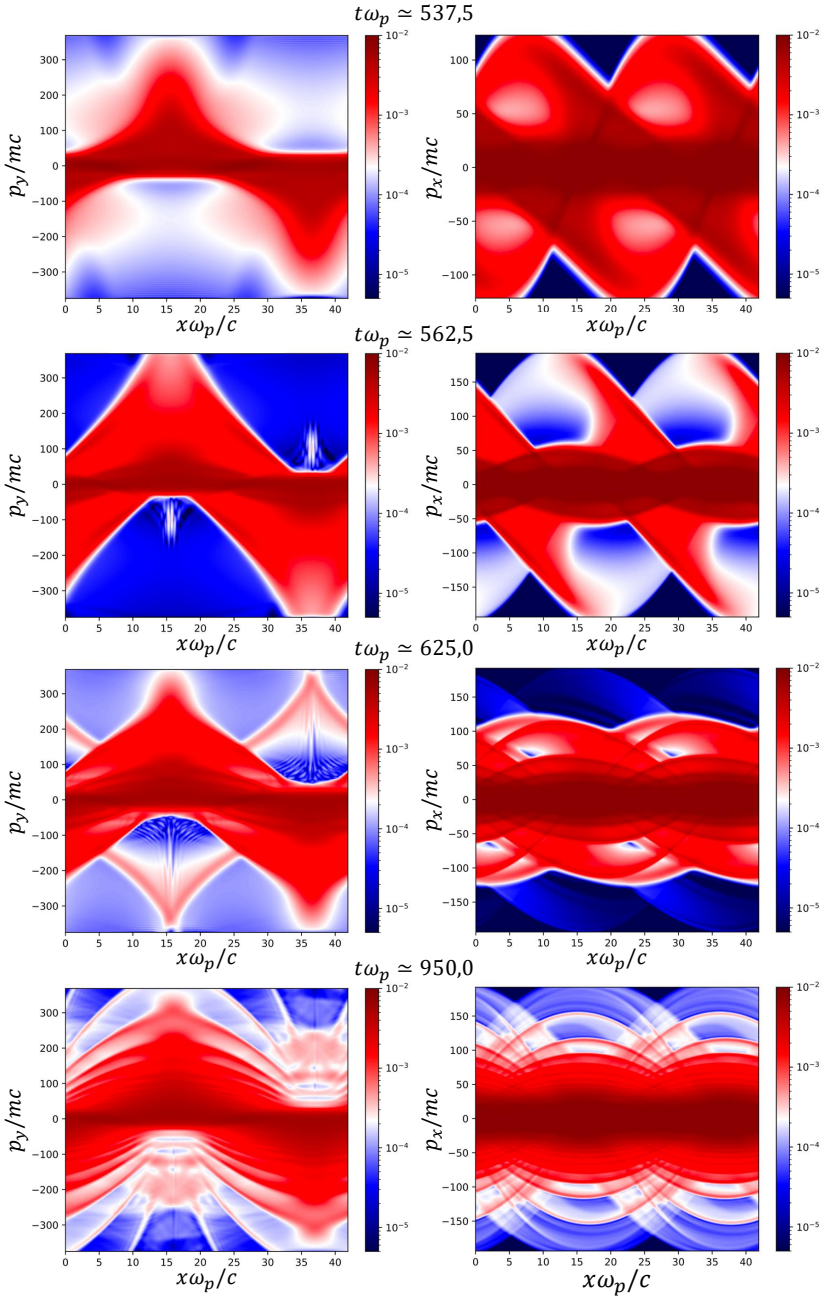

FIGURE 4.16 – Représentation de la fonction de distribution en $x-p_y$(à gauche), moyennée suivant p_x, et en $x-p_x$ (à droite), moyennée suivant p_y, pour VLEM_EG2D2V avec $\epsilon = 10^{-7}$ aux temps $t\omega_p \simeq 462{,}5$, $537{,}5$, $625{,}0$ et $950{,}0$ pour un point localisé en $(i_y = N_y/2)$.

subissent moins le piégeage magnétique. On observe cela sur la figure de gauche de 4.16, avec des "vagues" de particules, qui correspondent aux bras de la figure 4.14, qui vont se propager dans la direction x. Sur la figure de droite, on observe le piégeage magnétique sous forme de vortex bien caractéristiques de l'impact du champ B_z. Pour la direction y, les fonctions de distribution n'ont pas été représentées car il n'y pas de fluctuations spatiales dans cette direction.

La compréhension complète de cette accélération reste encore à étudier ainsi que l'impact de la synchronisation des modes de Van Kampen sur ce phénomène.

Chapitre 5

Méthode multi-faisceaux 2D3V

Sommaire

5.1 Méthode de réduction Hamiltonienne 156
5.2 Modification du "time splitting" 158
5.3 Modification du schéma numérique 160
5.3.1 Modification de la décomposition de domaine MPI 161
5.4 Calcul des coefficients pour déterminer les moments 166
5.5 Transition de OI à CFI avec VLEM2D3V multi-faisceaux 168
5.5.1 Nature électrostatique-électromagnétique de la perturbation initiale 168
5.5.2 Processus de saturation non linéaire 170
5.6 Discussion sur la méthode et compatibilité avec l'AMR 175
5.6.1 Avantages principaux du modèle 175
5.6.2 Limites et ouverture du modèle 176
5.6.3 Raffinement de maillage multi-faisceaux 177

Une autre méthode qui a été adaptée au code VLEM et qui permet d'accélérer le temps de calcul est la méthode multi-faisceaux. Cette dernière a commencé à être développée, peu de temps après la création de la première version de VLEM, dans les années 90, pour l'étude de la diffusion Raman [96], puis dans les années 2000, d'une part pour traiter des instabilités laser-plasma [137] et d'autre part pour étudier les instabilités de type Weibel [138, 139, 140, 141, 142, 143]. Cette méthode est basée sur une réduction Hamiltonienne liée à l'existence d'une invariance translationnelle, ce qui se traduit par une variable manquante, c'est-à-dire à l'absence d'une ou plusieurs dimensions de l'espace.

Implémenter le multi-faisceaux dans VLEM nécessite toutefois de modifier le schéma numérique ainsi que la méthode de parallélisation du code initial. Ces modifications sont compatibles avec la technique AMR présentées dans le chapitre 3, mais la combinaison de ces deux méthodes n'a pas encore été effectuée.

Au cours de cette thèse, des travaux d'amélioration et de développement du code VLEM multi-faisceaux 2D3V ont été effectués, aboutissant à la publication d'un article [144]. Les points clés seront présentés dans ce chapitre.

5.1 Méthode de réduction Hamiltonienne

Comme introduit au début de ce chapitre, la méthode multi-faisceaux nécessite l'absence d'une ou plusieurs dimensions spatiales. Cette absence va permettre, en utilisant les propriétés de l'Hamiltonien, de réduire le nombre de points dans une ou plusieurs dimensions de quantité de mouvement.

En effet, dans le cas du code VLEM2D3V multi-faisceaux, où on a deux dimensions d'espace (x, y), on peut écrire :

$$\frac{d\mathcal{P}_z}{dt} = -\frac{\partial H}{\partial z} = 0, \tag{5.1}$$

puisque H est indépendant de z. Or le moment canonique $\mathcal{P}_z$ est relié à la quantité de mouvement p_z par la relation :

$$\mathcal{P}_z = p_z + eA_z(x, y, t) = C_j, \tag{5.2}$$

avec A_z le potentiel vecteur du champ magnétique et C_j une constante.

On va donc pouvoir décomposer la fonction de distribution à 2 dimensions d'espace et 3 dimensions de quantité de mouvement comme une somme de fonctions de distribution à 2 dimensions d'espace et 2 dimensions de quantité de mouvement, telle que :

$$F(x, y, p_x, p_y, p_z) = \sum_{j=-N}^{N} \alpha_j f_j(x, y, p_x, p_y)\delta(p_z - (C_j - eA_z)), \tag{5.3}$$

avec α_j des coefficients qui déterminent la forme globale de F (voir figure 5.1) et qui peuvent être issus des moments de la fonction de distribution qui seront explicités dans le sous-chapitre 5.4.

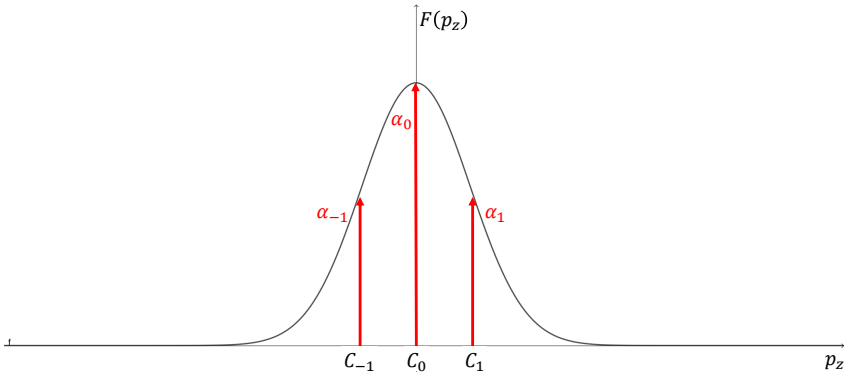

FIGURE 5.1 – Représentation d'une fonction de distribution $F(p_z)$ décrite par 3 faisceaux. Les C_j correspondent à la position sur la variable p_z tandis que les a_j représentent $F(C_j)$.

La fonction de distribution suivant p_z passe donc d'une description continue avec N_{p_z} points à une description avec un nombre $2N+1$ de faisceaux qui sont des distributions de Dirac. On va pouvoir suivre la dynamique de F à travers la dynamique des $2N+1$ fonctions de distributions f_j couplées à l'équation (5.2), avec les C_j connus initialement, ainsi qu'à l'équation donnant l'évolution du potentiel vecteur A_z :

$$\frac{\partial A_z}{\partial t} = -E_z, \tag{5.4}$$

et toujours couplées aux équations de Maxwell. On remarque qu'un faible nombre de faisceaux (3 ou 5) va suffire à apporter une bonne description de l'évolution de cette distribution [140, 143, 144].

Si on reprend la discussion de l'introduction sur la différence dans l'effort numérique entre un code PIC et un code Semi-Lagrangien (sous-partie 1.4.3), le code multi-faisceaux permet de remplacer $(N_v)^{d_v}$ par $(N_v)^{(d_v-1)} * N_{faisceau}$. Ainsi pour un modèle utilisant 5 faisceaux au lieu des 100 points ($N_v = 100$), on gagne un facteur 20 rendant ainsi un code Semi-Lagrangien multi-faisceaux plus compétitif face aux codes à particule (PIC). On peut cependant remarquer que la méthode peut aussi être appliquée aux codes PIC [145].

On peut donc réécrire l'Hamiltonien donné dans l'introduction (voir Eq.(1.25)) pour un seul faisceau comme :

$$H_j = \gamma_j mc^2 + e\phi, \tag{5.5}$$

avec γ_j le facteur relativiste associé au faisceau j qui va maintenant aussi dépendre des variables spatiales (x et y) et que l'on peut écrire :

$$\gamma_j = \sqrt{1 + \frac{p_x^2}{m^2c^2} + \frac{p_y^2}{m^2c^2} + \frac{\left(C_j - eA_z(x,y,t)\right)^2}{m^2c^2}}. \tag{5.6}$$

En reprenant les calculs fait dans la sous-partie 1.3.1, on peut réécrire l'équation de Vlasov comme :

$$\begin{aligned}\frac{\partial f_j}{\partial t} + \frac{p_x}{m\gamma_j}\frac{\partial f_j}{\partial x} + \frac{p_y}{m\gamma_j}\frac{\partial f_j}{\partial y} + \left[eE_x + e\frac{p_y}{m\gamma_j}B_z - \frac{\partial_x(C_j - eA_z)^2}{2m\gamma_j}\right]\frac{\partial f_j}{\partial p_x} \\ + \left[eE_y - e\frac{p_x}{m\gamma_j}B_z - \frac{\partial_y(C_j - eA_z)^2}{2m\gamma_j}\right]\frac{\partial f_j}{\partial p_y} = 0.\end{aligned} \tag{5.7}$$

En prenant en compte que le potentiel vecteur A_z dépend des variables d'espace et est lié au champ magnétique par $\mathbf{B} = \mathrm{rot}(\mathbf{A})$, on a :

$$\begin{aligned}\frac{\partial f_j}{\partial t} + \frac{p_x}{m\gamma_j}\frac{\partial f_j}{\partial x} + \frac{p_y}{m\gamma_j}\frac{\partial f_j}{\partial y} + \left[eE_x + e\frac{p_y}{m\gamma_j}B_z - \frac{B_y}{m\gamma_j}(C_j - eA_z)\right]\frac{\partial f_j}{\partial p_x} \\ + \left[eE_y - e\frac{p_x}{m\gamma_j}B_z + \frac{B_x}{m\gamma_j}(C_j - eA_z)\right]\frac{\partial f_j}{\partial p_y} = 0.\end{aligned} \tag{5.8}$$

Cette réécriture de l'équation de Vlasov va amener à modifier le "splitting" introduit dans la sous-partie 2.2.2.

5.2 Modification du "time splitting"

On va reprendre les calculs de la sous-partie 2.2.2 mais cette fois ci avec l'équation de Vlasov réécrite pour un faisceau (Eq.(5.8)). On donne la forme générale de Eq.(5.8) comme :

$$\frac{\partial f_j}{\partial t} + \mathbf{U}(\mathbf{X}, t) \cdot \boldsymbol{\nabla}_{\boldsymbol{X}} f_j = 0, \tag{5.9}$$

avec $\mathbf{X} = (x, y, p_x, p_y)$ et

$$\mathbf{U} = \begin{pmatrix} \frac{p_x}{m\gamma_j} \\ \frac{p_y}{m\gamma_j} \\ eE_x + e\frac{p_y}{m\gamma_j}B_z - \frac{B_y}{m\gamma_j}(C_j - eA_z) \\ eE_y - e\frac{p_x}{m\gamma_j}B_z + \frac{B_x}{m\gamma_j}(C_j - eA_z) \end{pmatrix} = \begin{pmatrix} U_{11} \\ U_{12} \\ U_{21} \\ U_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{U_1} \\ \mathbf{U_2} \end{pmatrix}. \tag{5.10}$$

Si U est à divergence nulle, on peut réécrire :

$$\frac{\partial f_j}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{X}} \cdot (\mathbf{U}(\mathbf{X}, t) f_j) = 0. \tag{5.11}$$

On vérifie donc la valeur de chacun des termes de la divergence séparément :

$$\frac{\partial U_{11}}{\partial X_{11}} = \frac{2ep_x}{m^3c^2\gamma_j^3}(C_j - eA_z)\frac{\partial A_z}{\partial x} = M, \tag{5.12}$$

$$\frac{\partial U_{12}}{\partial X_{12}} = \frac{2ep_y}{m^3c^2\gamma_j^3}(C_j - eA_z)\frac{\partial A_z}{\partial y} = N, \tag{5.13}$$

$$\frac{\partial U_{21}}{\partial X_{21}} = -\frac{ep_yB_z}{m\gamma_j^3}\frac{2p_x}{m^2c^2} - \frac{2ep_x}{m^3c^2\gamma_j^3}(C_j - eA_z)\frac{\partial A_z}{\partial x} = -O - M, \tag{5.14}$$

$$\frac{\partial U_{22}}{\partial X_{22}} = \frac{ep_xB_z}{m\gamma_j^3}\frac{2p_y}{m^2c^2} - \frac{2ep_y}{m^3c^2\gamma_j^3}(C_j - eA_z)\frac{\partial A_z}{\partial y} = O - N. \tag{5.15}$$

Les variables M, N et O ont été introduites afin de justifier les explications des calculs par la suite.

Dans la version non multi-faisceaux (sous-partie 2.2.2), les équations (5.12) et (5.13) sont nulles puisque γ_j ne dépend pas des variables d'espaces. On va voir par la suite que cela va entraîner une différence dans le schéma numérique à adopter.

On a donc bien $\boldsymbol{\nabla}_{\boldsymbol{X}} \cdot (\mathbf{U}(\mathbf{X}, t)) = 0$, l'équation (5.11) est donc correcte. Si on la réécrit en séparant les termes, on a :

$$\frac{\partial f_j}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{X_A}} \cdot (\mathbf{U_A}(\mathbf{X}, t) f_j) + \boldsymbol{\nabla}_{\boldsymbol{X_B}} \cdot (\mathbf{U_B}(\mathbf{X}, t) f_j) + \boldsymbol{\nabla}_{\boldsymbol{X_C}} \cdot (\mathbf{U_A}(\mathbf{X}, t) f_j) = 0, \tag{5.16}$$

avec

$$\mathbf{U_A} = \begin{pmatrix} \frac{p_x}{m\gamma_j} \\ 0 \\ -\frac{B_y}{m\gamma_j}(C_j - eA_z) \\ 0 \end{pmatrix}, \mathbf{X_A} = \begin{pmatrix} x \\ 0 \\ p_x \\ 0 \end{pmatrix}, \tag{5.17}$$

$$\mathbf{U_B} = \begin{pmatrix} 0 \\ \frac{p_y}{m\gamma_j} \\ 0 \\ \frac{B_x}{m\gamma_j}(C_j - eA_z) \end{pmatrix}, \mathbf{X_B} = \begin{pmatrix} 0 \\ y \\ 0 \\ p_y \end{pmatrix}, \tag{5.18}$$

$$\mathbf{U_C} = \begin{pmatrix} 0 \\ 0 \\ eE_x + e\frac{p_y}{m\gamma_j}B_z \\ eE_y - e\frac{p_x}{m\gamma_j}B_z \end{pmatrix}, \mathbf{X_C} = \begin{pmatrix} 0 \\ 0 \\ p_x \\ p_y \end{pmatrix}. \tag{5.19}$$

On peut résoudre l'équation (5.16) en conservant la précision au second ordre [101] en résolvant alternativement les équations suivantes :

$$\frac{\partial f_j}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{X_A}} \cdot (\mathbf{U_A}(\mathbf{X}, t) f_j) = 0, \tag{5.20}$$

$$\frac{\partial f_j}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{X_B}} \cdot (\mathbf{U_B}(\mathbf{X}, t) f_j) = 0, \tag{5.21}$$

$$\frac{\partial f_j}{\partial t} + \boldsymbol{\nabla}_{\boldsymbol{X_C}} \cdot (\mathbf{U_C}(\mathbf{X}, t) f_j) = 0. \tag{5.22}$$

Comme $\boldsymbol{\nabla}_{\boldsymbol{X_A}} \cdot (\mathbf{U_A}(\mathbf{X}, t)) = M - M = 0$, $\boldsymbol{\nabla}_{\boldsymbol{X_B}} \cdot (\mathbf{U_B}(\mathbf{X}, t)) = N - N = 0$ et $\boldsymbol{\nabla}_{\boldsymbol{X_C}} \cdot (\mathbf{U_C}(\mathbf{X}, t)) = -O + O$, on peut réécrire les équations (5.20), (5.21) et (5.22) sous la forme advective :

$$\frac{\partial f_j}{\partial t} + \frac{p_x}{m\gamma_j}\frac{\partial f_j}{\partial x} - \frac{B_y}{m\gamma_j}(C_j - eA_z)\frac{\partial f_j}{\partial p_x} = 0, \tag{5.23}$$

$$\frac{\partial f_j}{\partial t} + \frac{p_y}{m\gamma_j}\frac{\partial f_j}{\partial y} + \frac{B_x}{m\gamma_j}(C_j - eA_z)\frac{\partial f_j}{\partial p_y} = 0, \tag{5.24}$$

$$\frac{\partial f_j}{\partial t} + \left(eE_x + e\frac{p_y}{m\gamma_j}B_z\right)\frac{\partial f_j}{\partial p_x} + \left(eE_y - e\frac{p_x}{m\gamma_j}B_z\right)\frac{\partial f_j}{\partial p_y} = 0, \tag{5.25}$$

on appelle ces équations les "briques" élémentaires du modèle.

Afin de résoudre numériquement l'équation de Vlasov au second ordre pour le modèle multi-faisceaux, il va falloir résoudre les 3 équations d'advections (5.23-5.25) en les alternant. Chaque équation correspond à une advection bidimensionnelle qui est résolue globalement sans utiliser de technique de "splitting" supplémentaire.

5.3 Modification du schéma numérique

Séquence de résolution et advections

La principale modification apportée par ce nouveau modèle est que toutes les advections se feront à deux dimensions. On peut écrire la séquence de résolution comme $\left(\frac{\mathcal{X}}{2}\frac{\mathcal{Y}}{2}\mathcal{R}\frac{\mathcal{Y}}{2}\frac{\mathcal{X}}{2}\right)^n$ avec :

$$\frac{\mathcal{X}}{2} : \quad f^*(x, y, p_x, p_y) = f^n\left(x - \frac{p_x}{m\gamma}\frac{\Delta t}{2}, y, p_x + \frac{B_y}{m\gamma_j}(C_j - eA_z)\frac{\Delta t}{2}, p_y\right), \tag{5.26}$$

$$\frac{\mathcal{Y}}{2} : \quad f^{**}(x, y, p_x, p_y) = f^*\left(x, y - \frac{p_y}{m\gamma}\frac{\Delta t}{2}, p_x, p_y - \frac{B_x}{m\gamma_j}(C_j - eA_z)\frac{\Delta t}{2}\right), \tag{5.27}$$

$$\mathcal{R} : \quad f^{***}(x, y, \boldsymbol{p}) = f^{**}\left(x, y, P_x(x, y, \boldsymbol{p}), P_y(x, y, \boldsymbol{p})\right). \tag{5.28}$$

On a gardé les mêmes indices ($\mathcal{X}$, $\mathcal{Y}$ et $\mathcal{R}$) que ceux introduits dans la sous-partie 2.2.3 même si les advections sont différentes.

P_x et P_y sont les solutions obtenues à partir des caractéristiques de l'équation de Vlasov dans un espace à deux dimensions d'espace et deux dimensions de quantité de mouvement, telle que :

$$P_x = -e\frac{\Delta t E_x}{2} + p_x^\delta R_{xx} + p_y^\delta R_{xy}, \tag{5.29}$$

$$P_y = -e\frac{\Delta t E_y}{2} + p_x^\delta R_{yx} + p_y^\delta R_{yy}, \tag{5.30}$$

avec

$$\boldsymbol{p}^{\boldsymbol{\delta}} = \boldsymbol{p} - e\frac{\Delta t \boldsymbol{E}}{2}, \tag{5.31}$$

et avec R la matrice de rotation déjà introduite dans l'équation (2.26) mais ici simplifiée au cas 2D2V :

$$R = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}, \tag{5.32}$$

avec $\phi = \omega_c \Delta t$ et en prenant $\omega_c = eB_z/(m\gamma)$ la fréquence cyclotron.

5.3.1 Modification de la décomposition de domaine MPI

On a conservé les expressions de $\mathcal{X}$ et $\mathcal{Y}$ mais à la différence de la sous-partie 2.2.3 elles ont toutes deux une partie d'advection supplémentaire sur l'espace des quantités de mouvement. Cette différence va nécessiter, pour chaque advection, des interpolations à deux dimensions.

Pour les interpolations à deux dimensions, on va utiliser les B-splines cubiques introduits dans la sous-partie 2.2.5. On choisit d'utiliser une approche globale, ce qui nécessite d'utiliser l'inversion de la matrice tridiagonale. Ce choix permet de supprimer l'utilisation des splines "locaux" et d'éliminer la contrainte CFL qui en découle. Cependant, une approche globale requiert une transposition des variables sur lesquelles la parallélisation MPI est effectuée.

En effet, le découpage MPI introduit dans la sous-partie 2.2.4, qui était effectué sur des carrés de l'espace (x, y), va désormais être modifié au cours d'une itération temporelle. On garde un découpage en carré, voire en rectangle en fonction du nombre de points, cependant ce sont les variables sur lesquelles la décomposition MPI est effectuée qui vont changer. On écrit ses variables avec un symbole $\bar{}$ dans l'algorithme 3 représentant le nouveau schéma numérique.

On va donc, avant chaque advection, transposer les variables sur lesquelles la décomposition se fait afin d'avoir une advection à deux dimensions qui s'effectue sur le même processus.

Advections 2D $\mathcal{X}$ et $\mathcal{Y}$

En observant les deux advections spatiales ($\mathcal{X}$ et $\mathcal{Y}$, Eq.(5.26) et (5.27)), on remarque qu'elles couplent les variables d'advections entre elles. En effet, pour l'advection $\mathcal{X}$ en un point (y, p_y) fixé, on a le système suivant :

$$\begin{cases} x(i_x) - x - \frac{p_x}{m\gamma(x,p_x)}\frac{\Delta t}{2} = 0 \\ p_x(j_x) - p_x + \frac{B_y(x)}{m\gamma_j(x,p_x)}(C_j - eA_z(x))\frac{\Delta t}{2} = 0 \end{cases}, \tag{5.33}$$

avec $x(i_x)$ et $p_x(j_x)$ les points de grilles connus et x et p_x les variables de ce système qui représentent la position, dans l'espace des phases, du point de grille avant l'advection. Pour résoudre ce système, on utilise une méthode de Newton à deux dimensions avec cinq itérations (voir refs. [101, 146]) permettant de trouver les variables x et p_x. On peut ensuite effectuer l'advection qui nécessitera d'interpoler par B-splines cubiques la valeur de la fonction de distribution en ce point (x, p_x).

Les étapes de calculs des B-splines sont les mêmes que pour une méthode à deux dimensions de vitesse, excepté que la matrice pour les splines dans la direction spatiale (x ou y), va utiliser des conditions aux bords périodiques ($f'(x_0) = f'(x_{N_x})$ et $f''(x_0) = f''(x_{N_x})$) et donc va pouvoir s'écrire comme [103] :

$$\mathbf{M} = \frac{1}{6}\begin{pmatrix} \frac{-3}{h} & \frac{-3}{h} & 0 & \frac{3}{h} & 0 & \cdots & 0 & \frac{3}{h} & 0 \\ \frac{-6}{h^2} & \frac{6}{h^2} & \frac{-12}{h^2} & \frac{6}{h^2} & 0 & \cdots & 0 & \frac{-6}{h^2} & \frac{12}{h^2} \\ 0 & 1 & 4 & 1 & 0 & \vdots & \ddots & \ddots & 0 \\ \vdots & 0 & 1 & 4 & 1 & 0 & \ddots & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \vdots & \cdots & \cdots & \cdots & \ddots & 1 & 4 & 1 \\ 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 & 4 \end{pmatrix}, \tag{5.34}$$

avec $\mathbf{b} = (0, 0)$ et $\mathbf{c} = (f(x_0, p_{x_i}), ..., f(x_{N_x-1}, p_{x_i}), f(x_N = x_0, p_{x_i}))$. La première ligne de la matrice est l'égalité entre les dérivées premières et la deuxième celle entre les dérivées secondes issues des équations (2.40-2.41).

Modification du schéma de Yee

Concernant les champs électromagnétiques, les équations utilisées seront les mêmes que dans la sous-partie 2.2.7. Cependant, comme on souhaite garder la séquence de résolution $\left(\frac{\mathcal{X}}{2}\frac{\mathcal{Y}}{2}\mathcal{R}\frac{\mathcal{Y}}{2}\frac{\mathcal{X}}{2}\right)^n$, on doit modifier les positions temporelles des champs électriques et magnétiques. Les champs électriques seront désormais définis sur les quarts

de pas de temps ($t_{n+1/4}$ et $t_{n+3/4}$) tandis que les champs magnétiques et le potentiel vecteur seront définis sur les demis pas de temps (t_n et $t_{n+1/2}$). Le schéma de Yee est donc désormais résolus deux fois sur une advection temporelle. On a schématisé cela sur la figure 5.2.

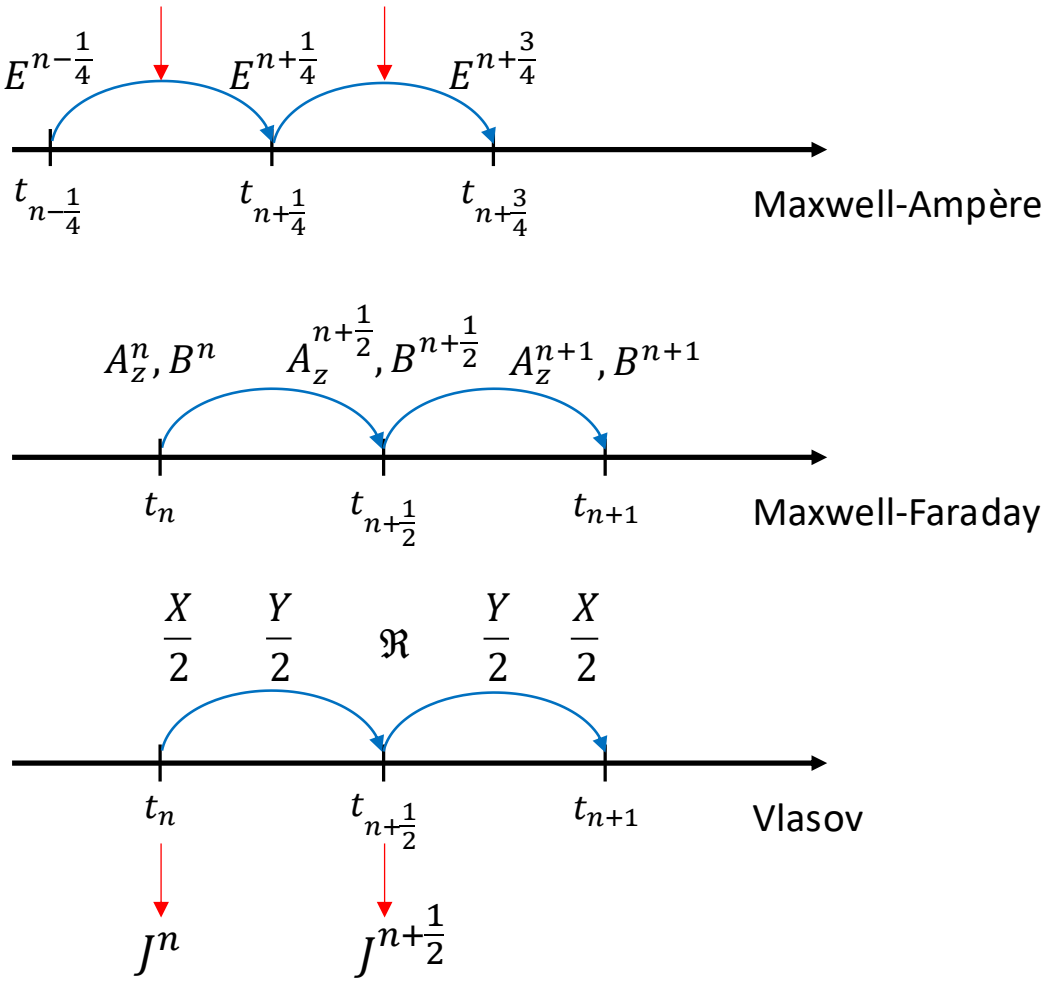

FIGURE 5.2 – Schéma numérique utilisé dans VLEM multi-faisceaux pour la résolution itérative temporelle utilisant les advections et la méthode de Yee.

Les nouvelles conditions CFL sont donc :

$$c^2\Delta t^2(\Delta x^{-2} + \Delta y^{-2}) \leq 4. \tag{5.35}$$

On peut trouver l'équation d'évolution temporelle pour le potentiel vecteur A_z à

partir de l'écriture du champs électrique :

$$\mathbf{E} = -\boldsymbol{\nabla}\phi - \frac{\partial \mathbf{A}}{\partial t}, \tag{5.36}$$

avec ϕ le potentiel électrique. En prenant la projection suivant z et sachant que la dérivée dans cette direction spatiale est nulle, on retrouve l'équation (5.4) :

$$E_z = -\frac{\partial A_z}{\partial t}. \tag{5.37}$$

On peut donc écrire l'évolution temporelle du potentiel électrique comme :

$$A_{z\,i,j}^{n+\frac{1}{2}} = A_{z\,i,j}^{n} - \frac{\Delta t}{2} E_{z\,i,j}^{n+\frac{1}{4}}, \tag{5.38}$$

avec $A_{z\,i,j}^{-\frac{1}{2}} = 0$ sauf si on ajoute une perturbation dans le système.

On va aussi déterminer la valeur de A_z aux temps $t_{n+1/4}$ et $t_{n+3/4}$ par une moyenne afin de pouvoir être utilisée dans $\mathcal{X}/2$ et $\mathcal{Y}/2$ (5.26 et 5.27).

Le courant J_z est aussi modifié, puisqu'il n'y a plus directement la variable p_z. On le calcul pour $2N+1$ faisceaux, comme :

$$\begin{aligned} J_z(x,y) &= \sum_{j=-N}^{N} \alpha_j \int \frac{e(C_j - A_z(x,y))}{m\gamma(x,y,p_x,p_y)} f_j(x,y,p_x,p_y) dp_x dp_y \\ &= \sum_{j=-N}^{N} \alpha_j \sum_{j_x=j_{x_{min}}}^{j_{x_{max}}} \sum_{j_y=j_{y_{min}}}^{j_{y_{max}}} \frac{e(C_j - A_z(x,y))}{m\gamma(x,y,p_x,p_y)} f_j(x,y,j_x,j_y) \Delta p_x \Delta p_y, \end{aligned} \tag{5.39}$$

On remarque que comme on est dans un cadre avec deux dimensions d'espace, on va avoir deux groupes de variables qui sont découplés l'un de l'autre. (E_x, E_y, B_z) que l'on va appeler les composantes "transverses magnétiques" et (E_z, B_x, B_y) que l'on va appeler les composantes "transverses électriques".

Algorithme du schéma numérique

On peut résumer l'ensemble de ces modifications en représentant l'algorithme du schéma numérique dans l'algorithme 3. Il peut être comparé aux algorithmes 1 et 2, correspondant respectivement aux codes VLEM et VLEM_EG.

while $n \leq n_{max}$ **do**
- Résolution des Eqs.(2.69-2.71) $\left(\mathbf{E}(t_{n-1/4}) \rightarrow \mathbf{E}(t_{n+1/4})\right)$
- Résolution des Eqs.(2.72-2.74, 5.38) $\begin{pmatrix} \mathbf{B}(t_n) \rightarrow \mathbf{B}(t_{n+1/2}) \\ A_z(t_n) \rightarrow A_z(t_{n+1/2}) \end{pmatrix}$
- Calcul de $\mathbf{B_{i,j,k}}(t_{n+1/4})$ et $A_z(t_{n+1/4})$
- Transposition $f(\bar{x}, \bar{y}, p_x, p_y) \rightarrow f(x, \bar{y}, p_x, \bar{p_y})$
- Advection 1D1V $\mathcal{X}/2$
- Transposition $f(x, \bar{y}, p_x, \bar{p_y}) \rightarrow f(\bar{x}, y, \bar{p_x}, p_y)$
- Advection 1D1V $\mathcal{Y}/2$
- Transposition $f(\bar{x}, y, \bar{p_x}, p_y) \rightarrow f(\bar{x}, \bar{y}, p_x, p_y)$
- Résolution des Eqs.(2.69-2.71) $\left(\mathbf{E}(t_{n+1/4}) \rightarrow \mathbf{E}(t_{n+3/4})\right)$
- Calcul de $\mathbf{E_{i,j,k}}(t_{n+1/2})$ et $\mathbf{B_{i,j,k}}(t_{n+1/2})$
- Rotation $\mathcal{R}$
- Résolution des Eqs.(2.72-2.74,5.38) $\begin{pmatrix} \mathbf{B}(t_{n+1/2}) \rightarrow \mathbf{B}(t_{n+1}) \\ A_z(t_{n+1/2}) \rightarrow A_z(t_{n+1}) \end{pmatrix}$
- Calcul de $\mathbf{B_{i,j,k}}(t_{n+3/4})$ et $A_z(t_{n+3/4})$
- Transposition $f(\bar{x}, \bar{y}, p_x, p_y) \rightarrow f(\bar{x}, y, \bar{p_x}, p_y)$
- Advection 1D1V $\mathcal{Y}/2$
- Transposition $f(\bar{x}, y, \bar{p_x}, p_y) \rightarrow f(x, \bar{y}, p_x, \bar{p_y})$
- Advection 1D1V $\mathcal{X}/2$
- Transposition $f(x, \bar{y}, p_x, \bar{p_y}) \rightarrow f(\bar{x}, \bar{y}, p_x, p_y)$
- Résolution de l'équation de Poisson (Eqs.(2.63-2.67)) (correction de $\mathbf{E}$)
- $n = n + 1$

end

Algorithme 3 : Algorithme simplifié de la résolution itérative temporelle de VLEM avec les advections et la méthode de Yee. n_{max} représente le nombre maximum d'itérations.

5.4 Calcul des coefficients pour déterminer les moments

Maintenant que l'on a décrit le schéma numérique, il faut déterminer les coefficients α_j et C_j qui interviendront dans le code. Ces coefficients sont fixés à l'aide de la fonction de distribution initiale et sont des constantes au cours de la simulation.

Pour cela, on rappelle que la fonction de distribution initiale est normalisée à n_0 qui est fixée à 1. En supposant une fonction de distribution qui ne dépend initialement pas de l'espace $F(x, y, p_x, p_y, p_z) = F(p_x, p_y, p_z)$, on peut prendre la moyenne sur p_x et p_y pour trouver :

$$h(p_z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(p_x, p_y, p_z)dp_x dp_y = \sum_{j=-N}^{N} \alpha_j \delta(p_z - C_j). \tag{5.40}$$

On a tracé cette fonction de distribution sur la figure 5.3 afin de montrer la signification des C_j et des α_j, les coefficients qui permettent de retrouver la fonction de distribution moyennées.

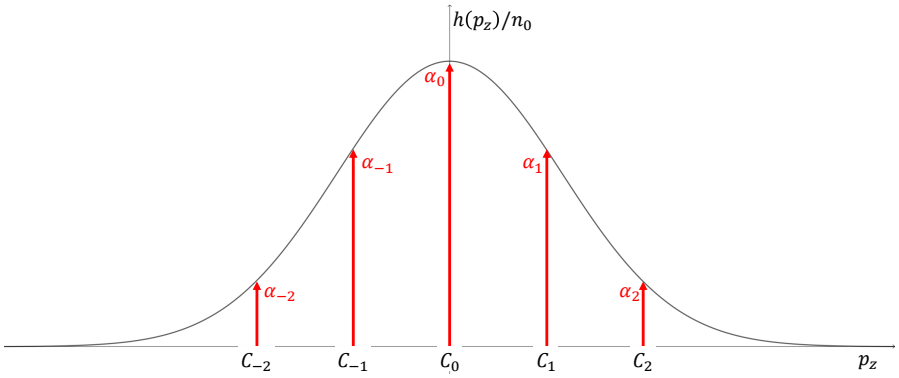

FIGURE 5.3 – Représentation de la fonction de distribution moyennée suivant p_x et p_y. Les C_j sont représentés et ont été choisis symétriques.

On va calculer ses coefficients en supposant que la fonction de distribution s'écrit sous la forme d'une Maxwelienne :

$$F(p_x, p_y, p_z) = \frac{1}{(2\pi m k_B T)^{\frac{3}{2}}} exp\left(-\frac{p_x^2 + p_y^2 + p_z^2}{2mk_BT}\right). \tag{5.41}$$

La moyenne des moments d'ordre $2n$ s'écrit donc comme :

$$< p_z^{2n} >= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_z^{2n} F(p_x, p_y, p_z)dp_x dp_y dp_z. \tag{5.42}$$

D'une part, en prenant la définition de la fonction de distribution définie Eq.(5.3), on a :

$$< p_z^{2n} >= \int_{-\infty}^{\infty} p_z^{2n} \sum_{j=-N}^{N} \alpha_j \delta(p_z - C_j) dp_z = \sum_{j=-N}^{N} \alpha_j C_j^{2n}, \tag{5.43}$$

et d'autre part, en utilisant la définition de la Maxwellienne (Eq.(5.41)), on trouve :

$$< p_z^{2n} >= \frac{1}{(2\pi m k_B T)^{\frac{1}{2}}} \int_{-\infty}^{\infty} p_z^{2n} exp\left(-\frac{p_z^2}{2mk_BT}\right). \tag{5.44}$$

En utilisant la formule générale de l'intégrale de Gauss [147] :

$$\int_{-\infty}^{\infty} x^{2n} e^{-ax^2} = \sqrt{\frac{\pi}{a}} \frac{(2n-1)!!}{(2a)^n}, \tag{5.45}$$

avec $(2n-1)!! = \Pi_{i=1}^{n}(2i-1)$, on trouve pour Eq.(5.44) :

$$< p_z^{2n} >= (2n-1)!!(mk_BT)^n. \tag{5.46}$$

En combinant les deux équations (Eq.(5.43) et Eq. (5.46)), on obtient l'équation pour le moment d'ordre 2n :

$$\sum_{j=-N}^{N} \alpha_j C_j^{2n} = (2n-1)!!(mk_BT)^n. \tag{5.47}$$

On retrouve pour le moment d'ordre 0 : $\sum_{j=-N}^{j=N} \alpha_j = 1$, ce qui redonne la valeur de la densité normalisée.

Comme cette fonction de distribution est symétrique par rapport à $p_z = 0$, on va choisir des coefficients tels que $C_0 = 0$, $C_j = -C_{-j}$ et $\alpha_j = \alpha_{-j}$. Si on choisit une représentation sur cinq faisceaux ($N = 5$), on trouve le système suivant :

$$\alpha_0 + 2\alpha_1 + 2\alpha_2 = 1, \tag{5.48}$$

$$2\alpha_1 C_1^2 + 2\alpha_2 C_2^2 = m^2 v_{th,z}^2, \tag{5.49}$$

$$2\alpha_1 C_1^4 + 2\alpha_2 C_2^4 = 3m^4 v_{th,z}^4. \tag{5.50}$$

En imposant $C_0 = 0$, $C_1 = m v_{th,z}$ et $C_2 = 2m v_{th,z}$, avec $v_{th,z} = \sqrt{k_B T_z / m}$ la vitesse thermique suivant z, on obtient pour les coefficients $\alpha_0 = 1/2$, $\alpha_1 = 1/6$ et $\alpha_2 = 1/12$. Ce choix a été fait afin d'avoir un espacement régulier entre les faisceaux.

5.5 Transition de OI à CFI avec VLEM2D3V multi-faisceaux

Dans cette partie, nous allons reprendre les résultats présentés dans l'article [144] dans la sous-partie 4.5.

On va appliquer le code multi-faisceaux en 2D3V à l'expérience numérique déjà présentée dans la sous-partie 3.4.2 afin de vérifier si le code conserve bien les invariants et si les advections 2D, $\mathcal{X}$, $\mathcal{Y}$ et $\mathcal{R}$ sont correctes.

5.5.1 Nature électrostatique-électromagnétique de la perturbation initiale

On rappelle que l'on étudie le cas de deux faisceaux se propageant en sens opposé, représentés par deux fonctions de distribution de Maxwell-Boltzmann décalées que l'on écrit comme $f\left(\boldsymbol{x},\boldsymbol{p},t=0\right)=\sum_{j=1,2}n_jF_{0j}\left(\boldsymbol{p}\right)$ où :

$$F_{0j}=\frac{1}{\left(2\pi m^2c^2\beta_{th_j}^2\right)^{3/2}}\exp\left(-\frac{p_x^2+\left(p_y-p_j\right)^2+p_z^2}{2m^2c^2\beta_{th_j}^2}\right), \tag{5.51}$$

et avec $\beta_{th_j}=\sqrt{k_BT_{eq_j}/mc^2}$ la vitesse thermique normalisée, $p_j=m\gamma_{0j}\beta_jc$ la quantité de mouvement moyenne normalisée du faisceau et $\gamma_{0j}=1/\sqrt{1-\beta_j^2}$ le facteur de Lorentz correspondant.

Comme on choisit d'étudier le même problème que dans la sous-partie 3.4.2, on prend un cas symétrique avec $p_1=-p_2=0,9mc$, ce qui correspond à $\beta_1=-\beta_2=0,67$. En reprenant les résultats de Ghizzo et al. [29, 74], pour $k_BT_{eq_1}=k_BT_{eq_2}=6keV$ et $n_1=n_2=\frac{n_0}{2}$, on peut déclencher l'instabilité oblique "basse" fréquence en excitant le mode $k_xd_e\approx3,07$ et $k_yd_e\approx5$ (avec $d_e=c\omega_p^{-1}$, l'épaisseur de peau). Ce qui fixe la dimension normalisée de la boîte à $L_x=4,0879d_e=\frac{2\pi}{\Delta k_x}$ et $L_y=2,513274d_e=\frac{2\pi}{\Delta k_y}$, où l'instabilité va correspondre au mode de Fourier $(n_x,n_y)=(2,2)$ ($\Delta k_xd_e\approx1,54$ et $\Delta k_yd_e\approx2,50$). On choisit d'échantillonner l'espace avec $N_xN_y=256^2$, on a donc $\Delta x=0,0160d_e$ and $\Delta y=0,00985d_e$. On rappelle que, pour ce cas, la figure 3.8 donne les taux de croissance et que l'on a choisi des paramètres afin d'exciter dans le cercle blanc correspondant à l'instabilité oblique « basse » fréquence.

Pour le code VLEM, on prend un échantillonnage $N_{p_x}N_{p_y}N_{p_z}=64^3$ dans l'espace des quantités de mouvement et un pas de temps $\Delta t\omega_p=0,008$ et de la même manière que précédemment (sous-partie 3.4.2), on va perturber le champs magnétique suivant z, $\delta B_z=\delta B_0\sin\left(n_x\Delta k_xx+n_y\Delta k_yy\right)$ afin d'exciter le mode le plus instable. On choisit une perturbation $e\delta B_0/m\omega_p=0,005$.

Pour le modèle multi-faisceaux, on résout l'équation de Vlasov réduite (Eq.(5.8)), avec cinq faisceaux, couplés de manière auto-cohérente avec l'Eq.(5.37) et les équations de Maxwell. Pour la fonction de distribution initiale, on choisit un cas similaire au code VLEM et on va modéliser la température suivant p_z en utilisant les équations (5.48-5.50). Il faut cependant pour ce modèle, en plus de la perturbation $\delta B_z = \delta B_0 \sin\left(n_x \Delta k_x x + n_y \Delta k_y y\right)$, ajouter une perturbation sur les deux autres composantes transverses magnétiques telle que :

$$\begin{aligned} E_x &= -\frac{k_y \delta B_0 \omega_p}{k_x^2 + k_y^2} \sin\left(n_x \Delta k_x x + n_y \Delta k_y y\right), \\ E_y &= \frac{k_x \delta B_0 \omega_p}{k_x^2 + k_y^2} \sin\left(n_x \Delta k_x x + n_y \Delta k_y y\right). \end{aligned} \tag{5.52}$$

Le choix de cette perturbation est dû au fait que la CFI, quasi-statique si elle est excitée seule avec une propagation perpendiculaire par rapport à la propagation des faisceaux, se couple avec la TSI et conduit à l'instabilité oblique. Par conséquent, le mode combiné CFI-TSI confère une nature propagative à la perturbation du champ magnétique, impliquant ainsi les composantes E_x et E_y.

Alors qu'une simple perturbation des composantes du champ magnétique suffit à déclencher l'instabilité dans le code VLEM2D3V, une perturbation cohérente doit être introduite dans le code multi-faisceaux, affectant à la fois les composantes électromagnétiques et le potentiel vecteur, ainsi que la fonction de distribution. Cette perturbation cohérente va permettre un démarrage plus rapide de l'instabilité, mais doit respecter l'équilibre initial, c'est-à-dire les équations de Maxwell. Cette différence entre les deux codes est liée à la construction du multi-faisceaux, qui "découple" quelque peu la dynamique de la fonction de distribution f obtenue à partir des advections $\mathcal{X}$ et $\mathcal{Y}$ (Eqs.(5.23) et (5.24)) et le potentiel vecteur calculé à partir de l'équation (5.37), du comportement de f décrit par la rotation globale (Eq.(5.25)).

Ainsi, pour les composantes transverses électrostatiques, on a :

$$\begin{aligned} E_z &= \delta E_0 \sin\left(n_x \Delta k_x x + n_y \Delta k_y y\right), \\ B_x &= \frac{k_y \delta E_0}{\omega_0} \sin\left(n_x \Delta k_x x + n_y \Delta k_y y\right), \\ B_y &= -\frac{k_x \delta E_0}{\omega_0} \sin\left(n_x \Delta k_x x + n_y \Delta k_y y\right), \end{aligned} \tag{5.53}$$

où $\omega_0^2 = \omega_p^2 + \left(k_x^2 + k_y^2\right) c^2$. Une perturbation correspondante ainsi qu'une perturbation de plus faible amplitude permettant d'exciter un large spectre de modes instables sont ajoutées sur le potentiel vecteur A_z tel que :

$$A_z = -\frac{\delta E_0}{\omega_0} \cos\left(k_x x + k_y y\right) + \frac{\delta E_0}{\omega_0} \times 10^{-3} \sum_{n_x, n_y = 1}^{15} \sin\left(n_x \Delta k_x x + n_y \Delta k_y y\right). \tag{5.54}$$

En ce qui concerne la fonction de distribution, les perturbations de la densité et de la quantité de mouvement sont introduites de manière cohérente avec les perturbations données sur les composantes du champ électromagnétique. On a donc :

$$\begin{gathered} f\left(\boldsymbol{x}, \boldsymbol{p}, t=0\right) = \\ \left(1+\delta n \cos\left(n_x \Delta k_x x + n_y \Delta k_y y\right)\right) \sum_{j=1,2} n_j F_{0j}\left(p_x - P_x, p_y - P_y\right), \end{gathered} \tag{5.55}$$

où les quantités de mouvement perturbées sont :

$$\begin{aligned} P_x &= \frac{e k_y \delta B_0}{k_x^2 + k_y^2} \cos\left(n_x \Delta k_x x + n_y \Delta k_y y\right), \\ P_y &= \frac{e k_x \delta B_0}{k_x^2 + k_y^2} \cos\left(n_x \Delta k_x x + n_y \Delta k_y y\right), \end{aligned} \tag{5.56}$$

avec $\delta n/n_0 = 0,005$, $e\delta B_0/m\omega_p = 0,005$ et $e\delta E_0/m\omega_p c = 0,05$. On a pour l'échantillonnage $N_{p_x} N_{p_y} = 128^2$, le pas de temps utilisé dans les simulations est $\Delta t \omega_p = 0,005$.

5.5.2 Processus de saturation non linéaire

La CFI est l'un des principaux processus de base du plasma qui génère un fort champ magnétique B_z. Dans sa version oblique (OI), étudiée ici, un couplage avec l'instabilité électrostatique à double faisceaux (TSI) est également possible, permettant ainsi la croissance d'une composante électrostatique du champ électrique. Cette configuration va convertir l'énergie cinétique $E_k = \frac{1}{L_x L_y} \int d^2x \int d^3p\, mc^2(\gamma - 1) f$ en énergie magnétique $E_m = \frac{\epsilon_0}{2 L_x L_y} \int d^2x\, c^2 B^2$ et en énergie électrique $E_e = \frac{\epsilon_0}{2 L_x L_y} \int d^2x\, E^2$. D'un point de vue numérique, les différentes énergies sont normalisées à $n_0 mc^2$.

Dans la figure 5.4, on a superposé l'évolution temporelle de l'énergie magnétique en échelle logarithmique obtenue à partir du code VLEM2D3V (en noir), et celle obtenue à partir du code multi-faisceaux (en bleu). Les courbes montrent le même taux de croissance, proche de $\Gamma^{num}/\omega_p = 0,405 \pm 0,003$, en accord avec le taux de croissance maximum théorique $\Gamma^{theo}/\omega_p \sim 0,40$ prédit par la relation de dispersion obtenue dans [74]. Les courbes montrent un retard dans le démarrage de l'instabilité (proche du décalage $t_{\text{shift}}\omega_p \sim 60$), l'instabilité OI dans le code VLEM2D3V démarrant plus tard, ce décalage ne change pas le taux de croissance de l'OI. Ce décalage temporel est attribué à une différence entre les perturbations introduites dans les deux codes.

Dans les deux approches numériques (entièrement cinétique avec le code VLEM et réduite avec le code multi-faisceaux), l'équation de Vlasov est résolue à l'aide d'un schéma Semi-Lagrangien, qui présente un très faible niveau de bruit numérique. Le démarrage de l'instabilité est très sensible au type de perturbation introduit initialement.

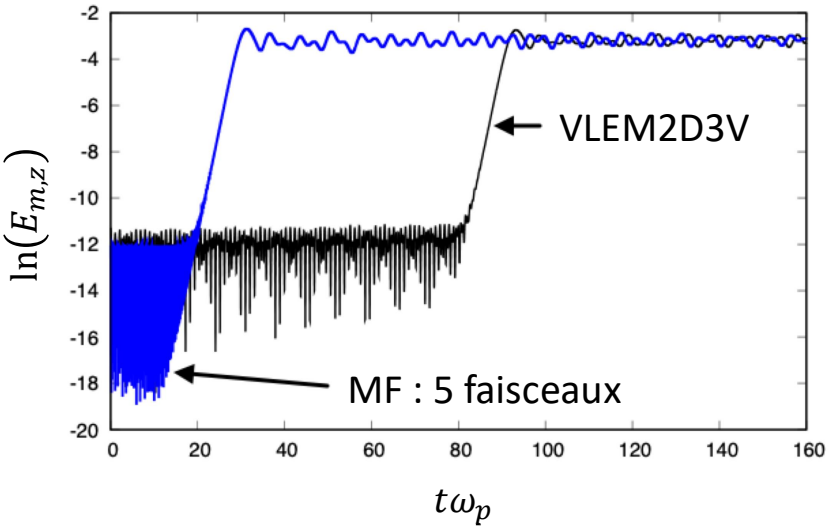

FIGURE 5.4 – Évolution temporelle de la composante z de l'énergie magnétique $E_{m,z} = \frac{\epsilon_0}{2L_xL_y} \int d^2x\ c^2B_z^2$, en échelle logarithmique, dans le cas d'une instabilité OI à caractère hybride, à la fois électromagnétique (CFI) et électrostatique (TSI), pour un système constitué de deux faisceaux d'électrons de sens opposés. La courbe noire correspond au code VLEM2D3V, tandis que la courbe bleue a été obtenue à partir du code multi-faisceaux (MF) (avec 5 faisceaux). Les paramètres physiques sont identiques dans les deux simulations.

Ainsi, le code multi-faisceaux nécessite l'introduction d'une perturbation cohérente non seulement dans les composantes des champs électromagnétiques $(\boldsymbol{E}, \boldsymbol{B})$ et pour le potentiel vectoriel $\boldsymbol{A}$, mais aussi dans la fonction de distribution, qui tient compte des perturbations de la densité et des quantités de mouvement induites par le champ électromagnétique initial. Une telle perturbation est nécessaire pour déclencher l'instabilité Weibel, car le code commence dans un état sans bruit, les composantes électrostatiques et électromagnétiques du champ électrique étant découplées dès le départ. En revanche, le code VLEM2D3V ne nécessite pas de telles modifications dans les conditions initiales ; seule une perturbation de la composante du champ magnétique est nécessaire.

La figure 5.5 montre l'évolution temporelle de l'énergie magnétique $E_{m,z}$ calculée à partir de la composante B_z (à gauche) ainsi que l'évolution de la densité moyenne (à droite) avec le code multi-faisceaux dans le cas où une perturbation est introduite uniquement sur les champs pour un cas similaire à la figure 5.4.

Dans ce cas, la densité des particules est très bien conservée (la variation de la densité relative est ici limitée à 10^{-8}). La simulation illustrée dans cette figure 5.5 montre l'excellente stabilité du code, puisque la condition initiale d'équilibre métastable est

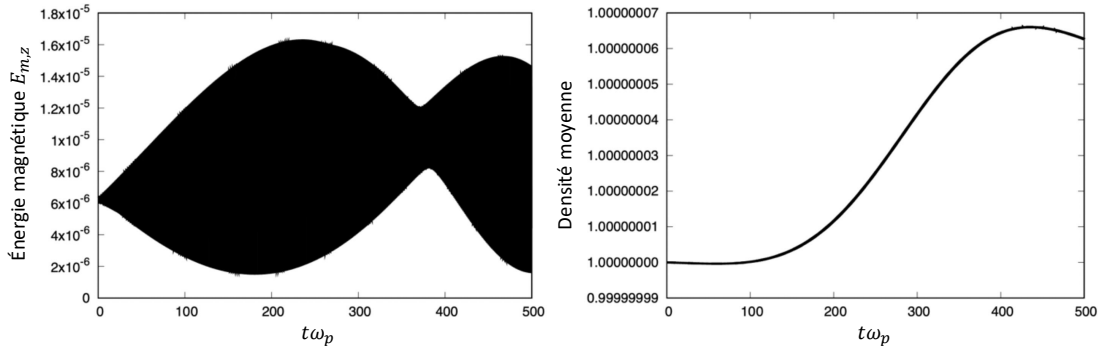

FIGURE 5.5 – Exemple avec le code multi-faisceaux d'une instabilité OI dans laquelle le couplage entre la CFI et TSI n'a pas lieu. La perturbation initiale n'est introduite que dans les champs, mais pas dans la fonction de distribution. En raison de la manière dont l'algorithme fait progresser les quantités, cela induit un temps de transition beaucoup plus long avant que le mode propre OI ne soit formé et ne commence à croître. Par conséquent, le système reste stable pendant un long intervalle de temps. A gauche, l'évolution temporelle de la composante z de l'énergie magnétique $E_{m,z}$, à droite l'évolution temporelle de la densité moyenne qui présente une bonne conservation.

préservée jusqu'à environ 10^5 itérations. La perte de particules ($\sim 10^{-8}$) de la densité relative commence à environ $t\omega_p \simeq 200$ en raison de l'apparition de l'instabilité à partir d'un long temps transitoire (la perturbation donnée permet néanmoins l'apparition d'une OI de faible amplitude). La conservation de la densité pourrait en principe être améliorée en augmentant la précision de l'espace des phases. Cependant, sa conservation imparfaite ne peut être évitée puisqu'elle est principalement due aux conditions aux limites ouvertes dans l'espace des quantités de mouvement : ceci, combiné à l'élargissement de la fonction de distribution (visible, dans le cas d'un OI, sur la composante p_x dans la figure 5.7, suivante), fait inévitablement que les particules situées sur la queue de la distribution "s'échappent" de la boîte de simulation.

En revanche, lorsque la perturbation initiale est introduite dans le modèle multi-faisceaux à la fois sur les composantes du champ électromagnétique et sur la fonction de distribution initiale (pour assurer le démarrage en phase du champ électromagnétique et du potentiel vecteur), le mode propre oblique se développe sur un temps transitoire beaucoup plus court. Cependant, le taux de croissance de l'instabilité reste inchangé, et le niveau de fluctuation du champ magnétique ainsi que la fréquence de rebond magnétique restent similaires dans les deux approches numériques, même dans le régime de saturation de l'instabilité.

La figure 5.6 montre la conservation de la masse et de l'énergie pour le code multi-faisceaux avec une perturbation initiale sur les champs et sur la fonction de distribution. La masse totale est conservée à 0,4% en appliquant les méthodes d'advection $\mathcal{X}$

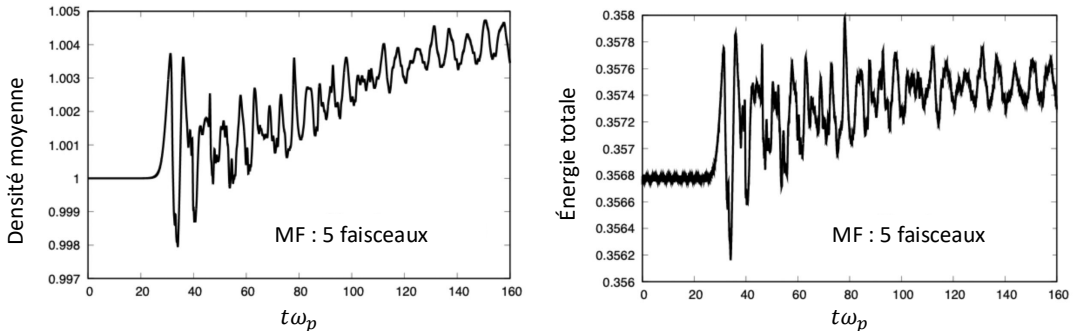

FIGURE 5.6 – A gauche, représentation de l'évolution temporelle de la densité moyenne (masse totale) obtenue dans le cas du code multi-faisceaux (MF) (avec 5 faisceaux) en utilisant le schéma numérique de l' algorithme 3, basé sur l'utilisation alternée d'advections 2D. A droite, l'énergie totale correspondante en fonction du temps. On note que la densité moyenne est maintenue à 0,8 % pour 20 000 itérations en moyenne.

(x, p_x), $\mathcal{Y}$ (y, p_y) et $\mathcal{R}$ (p_x, p_y). La conservation totale de l'énergie est atteinte à 0,3 % sur 20 000 itérations. Cependant, la conservation de la masse est moins efficace que les résultats de la figure précédente 5.5, en raison de l'instabilité qui provoque un échauffement important de la fonction de distribution, ce qui élargit la distribution sur les variables de quantité de mouvement p_x et p_y, comme le montre la figure 5.7.

En haut de la figure 5.7, le tracé de la fonction de distribution correspond au résultat obtenu avec le code entièrement cinétique VLEM2D3V, qui rend compte de la dynamique complète de f dans l'espace des quantités de mouvement (c'est-à-dire qu'il n'y a pas de faisceaux ici). Cette courbe est tracée à l'instant $t\omega_p = 400$, lorsque la saturation de l'instabilité oblique est bien établie. Le panneau du milieu montre une dynamique similaire obtenue avec le code multi-faisceaux à un instant équivalent (c'est-à-dire lorsque l'instabilité a atteint le régime de saturation). Dans le régime de saturation, la dynamique de f est caractérisée par l'émergence de deux structures de piégeage magnétique. En conséquence, la fonction de distribution subit un élargissement significatif en p_x, indiquant un processus de chauffage.

Sur les cadres du milieu et du bas, dans la figure 5.7, on montre les résultats obtenus avec le code multi-faisceaux utilisant 5 faisceaux : le cadre du milieu correspond à un tracé de la fonction de distribution du faisceau central, c'est-à-dire situé à $p_z \sim C_0 = 0$. Ce tracé met en évidence la dynamique de f pour le faisceau central, en particulier pour la population d'électrons qui appartiennent au coeur de la distribution (avec une quantité de mouvement p_z de zéro).

Le modèle multi-faisceaux fournit une description détaillée des interactions onde-particule, même dans les régions de l'espace des phases où les densités sont très faibles. Un exemple est illustré en bas sur la figure 5.7, qui montre le comportement de la fonction de distribution pour une valeur de quantité de mouvement de

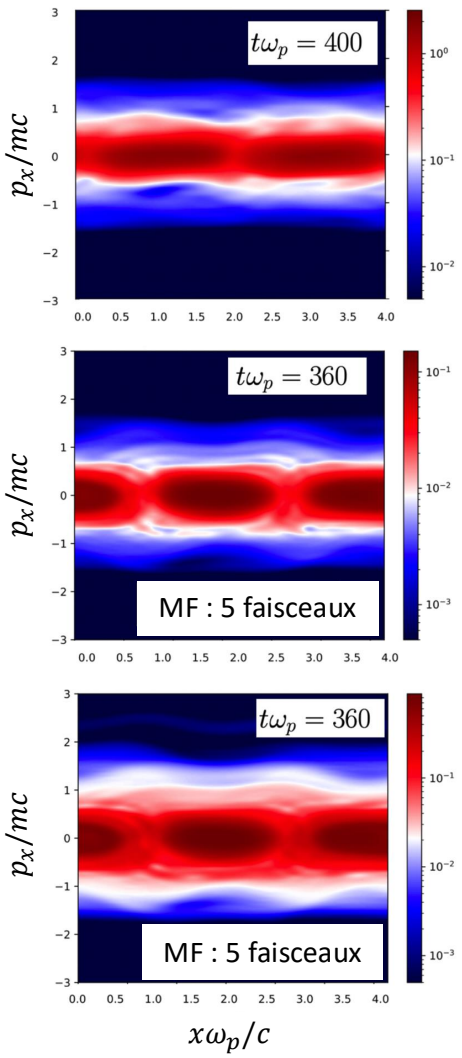

FIGURE 5.7 – Représentation de la fonction de distribution en $x-p_x$, moyennée suivant p_y et p_z, obtenue à partir du code cinétique complet VLEM2D3V (haut) pour un point situé en ($i_y = N_y/2$). Les tracés du milieu et du bas correspondent aux distributions du faisceau central (positionné en $p_z \sim C_0 = 0$) et du dernier faisceau (positionné en $p_z \sim C_2 = 2p_{\mathrm{th},z}$) obtenues à partir du code multi-faisceaux (avec 5 faisceaux). Notons que le dernier faisceau (en bas) correspondant à une quantité de mouvement de $p_z \sim 2p_{\mathrm{th},z}$ (deux fois la quantité de mouvement thermique) est constitué d'une population d'électrons de plus faible densité, et que le code multi-faisceaux (MF) est parfaitement adapté à une description fine de l'interaction onde-particule, y compris dans les régions de la queue de la fonction de distribution.

$p_z \sim C_2 = 2p_{\mathrm{th},z}$ (deux fois la quantité de mouvement thermique), correspondant à la fonction de distribution du cinquième faisceaux (avec une densité initiale beaucoup plus faible par rapport au faisceau central). C'est l'un des points forts de cette méthode puisqu'une telle analyse nécessiterait un nombre exorbitant de « particules » dans un code PIC et/ou un échantillonnage très élevé de f dans l'espace des quantités de mouvement dans le cas d'une approche directe utilisant le code VLEM2D3V. Les processus liés aux queues de la fonction de distribution sont ainsi décrits avec précision. On observe que le processus de piégeage, qui est de nature mixte - à la fois électrostatique (induit par TSI) et magnétique (induit par CFI) - est plus important dans les régions de très faibles densités.

5.6 Discussion sur la méthode et compatibilité avec l'AMR

5.6.1 Avantages principaux du modèle

Le modèle multi-faisceaux présente trois avantages principaux. D'un point de vue physique, il est toujours possible de décrire tout effet de température perpendiculaire (c'est-à-dire le long de p_z). Trois faisceaux sont suffisants pour représenter avec précision toute fonction de distribution jusqu'à son moment du second ordre. En augmentant le nombre de faisceaux et en modifiant leurs α_j et C_j, il est alors possible d'établir une correspondance initiale jusqu'à un moment fluide arbitrairement élevé de la fonction de distribution [143]. D'un point de vue numérique, cette méthode permet également une réduction drastique du temps de calcul, théoriquement de l'ordre de $N_{p_z}/N_{faisceaux}$, c'est-à-dire le rapport de l'échantillonnage utilisé dans la direction p_z sur le nombre total de faisceaux. Enfin, le fait de disposer d'un modèle mathématique plus simple permet également de mieux comprendre et de mieux décrire les processus physiques décrits.

En effet, un point qui a été présenté dans l'article [144] mais sur lequel nous n'avons pas voulu appuyer dans ce chapitre est la possibilité de calculer plus précisément les intégrales sur p_z. En effet, ces dernières étant liées à une somme de Dirac, on peut les considérer comme "semi exactes".

On peut donc étudier plus précisément les processus de reconnexion magnétique dont on donne un exemple sur la figure 5.8. La simulation présentée utilise également le fait que le modèle multi-faisceaux permet de représenter facilement une anisotropie de température dans la direction T_z en choisissant les C_j et α_j adéquats (voir sous-chapitre 5.4).

Comme le modèle multi-faisceaux permet un calcul plus exact des intégrales suivant p_z, le tenseur matériel de contrainte de densité est aussi impacté. Il s'écrit dans

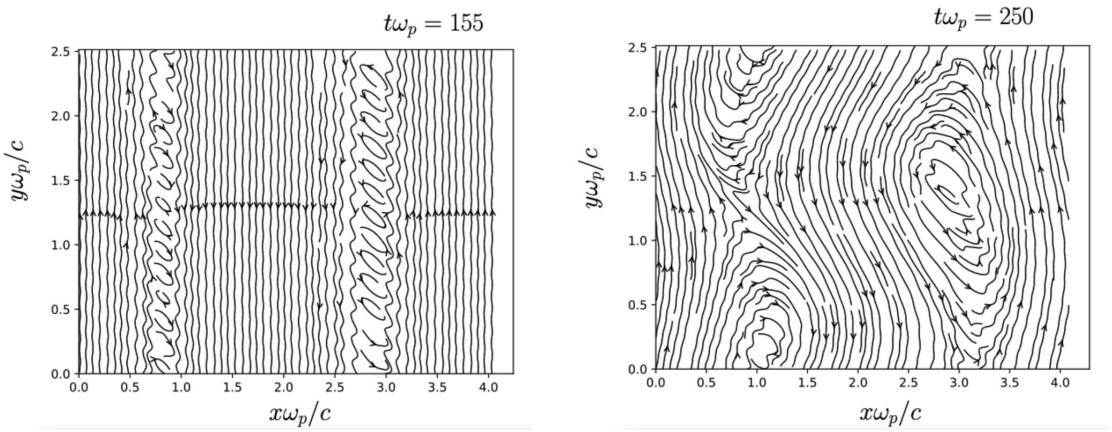

FIGURE 5.8 – Illustration de l'auto-organisation des lignes de champs magnétiques (B_x, B_y) dans le plan (x,y) obtenu à partir du modèle multi-faisceau. Une quinzaine de sites de reconnexion sont observés constituant deux chaînes d'îlots magnétique à $t\omega_p = 155$ (gauche). L'auto-organisation du système conduit à des processus de fusion des îlots jusqu'à l'apparition de deux îlots mésoscopiques à $t\omega_p = 250$ (droite). La simulation a été réalisée en utilisant cinq faisceaux qui ont été choisis de manière à avoir une anisotropie de température ($T_z > T_x = T_y$).

un cas purement cinétique et à trois dimensions de quantité de mouvement comme :

$$\tilde{\Pi}_{ij} = \iiint \frac{p_i p_j}{m\gamma(x, y, p_x, p_y)} f(x, y, p_x, p_y, p_z) dp_x dp_y dp_z, \tag{5.57}$$

et dans le cadre multi-faisceaux :

$$\tilde{\Pi}_{ik} = \sum_{j=-N}^{N} \alpha_j \iint dp_x dp_y \frac{p_i p_k}{m\gamma_j\,(x, y, p_x, p_y)} f_j. \tag{5.58}$$

où on remplace la quantité p_z par $p_z \equiv C_j - eA_z\,(x, y, t)$.

Ce tenseur englobe la contribution de l'énergie cinétique et de l'énergie interne du système. Ces composantes diagonales ($\tilde{\Pi}_{xx}$, $\tilde{\Pi}_{yy}$, $\tilde{\Pi}_{zz}$) permettent de décrire le chauffage dans chaque direction des quantités de mouvement. Tandis que les termes hors diagonale peuvent fournir une indication sur la présence d'anisotropie, ce qui permet d'identifier les couplages potentiels entre les diverses instabilités.

5.6.2 Limites et ouverture du modèle

Cependant, la méthode multi-faisceaux nécessite qu'on ait au moins une dimension d'espace manquante. On ne peut donc pas l'utiliser pour simplifier des cas 2D2V ou

3D3V, par exemple. Elle reste néanmoins très efficace pour décrire les codes 2D3V qui nécessite des ressources numériques encore importantes.

L'idée d'utiliser l'invariance du moment canonique pour l'intégration numérique de l'équation de Vlasov a probablement été introduite pour la première fois par Dickman et al [148], qui l'ont utilisée pour étudier les instabilités dans les dispositifs de confinement magnétique. Morse et Nielson [145] l'ont utilisée dans les codes PIC sur les instabilités Weibel. Mais le fait de regrouper des particules selon l'invariance du moment canonique n'est pas si différent que de le faire sur les invariants adiabatiques, sur lesquels sont fondés les notions de « structures zonales de l'espace-phase » [149] ou de « modes de particules » [150], introduites plus tard dans les modèles gyro-cinétiques. Dans un cadre gyrocinétique, une technique de réduction multi-faisceaux a également été utilisée par Manfredi et al. [151].

Le principe de cette méthode peut aussi être adapté à des modèles plus éloignés qui sont décrits par l'équation de Vlasov comme les semi-conducteurs ou les plasmas quantiques (équation de Wigner [152]). On peut aussi rajouter les degrés de libertés du spin afin d'avoir des codes jusqu'à 8 dimensions (avec 2 dimensions de plus dues à l'angle du spin dans une sphère de Riemman) [153, 154]. Ou encore en relativité en prenant en compte les quatre-vecteurs [155, 156, 157].

5.6.3 Raffinement de maillage multi-faisceaux

La méthode AMR présentée dans le chapitre 3 est compatible avec le modèle multi-faisceaux présenté dans ce chapitre mais à condition d'y apporter des modifications. En effet, comme expliqué dans le sous-chapitre 5.3, les variables sur lesquelles on va effectuer la parallélisation vont changer au cours d'une itération. Ce qui veut dire qu'un raffinement de maillage dans l'espace des quantités de mouvement pourrait amener à des dissymétries dans les temps de calculs entre les processus. Par exemple lorsque les variables de la parallélisation seraient $(\bar{x},\bar{p_x})$, les processus dans les zones où p_x est moins raffiné calculeraient beaucoup plus rapidement que les processus complètement raffinés.

Cependant, les méthodes d'advection à deux dimensions resteraient compatibles car le rectangle de la grille emboîtée est le même pour chaque point de l'espace et donc chaque advection va pouvoir se faire d'une manière identique à la rotation dans VLEM_EG (voir sous-partie 3.3.3). Il faudra cependant modifier la matrice $\mathbf{M}$ pour l'adapter aux conditions périodiques spatiales.

Une idée pour rendre compatible cette méthode serait soit d'implémenter une technique de répartition du poids entre les processus comme c'est effectué dans différents codes [60, 116]. Soit, et cette dernière est plus beaucoup plus simple, effectuer la parallélisation MPI uniquement sur une seule variable d'espace lors des advections spatiales $\mathcal{X}$ et $\mathcal{Y}$. On devrait donc remplacer dans l'algorithme 3, avant $\mathcal{X}/2$, $f(x,\bar{y},p_x,\bar{p_y})$ par $f(x,\bar{y},p_x,p_y)$, et avant $\mathcal{Y}/2$, $f(\bar{x},y,\bar{p_x},p_y)$ par $f(\bar{x},y,p_x,p_y)$.

L'inconvénient est que le nombre de point spatiaux dans chaque processus peut être faible. En effet, pour l'exemple choisi dans ce chapitre, on a $N_x * N_y = 256^2$ et 256 processus. Une parallélisation suivant une seule variable d'espace donnerait donc un seul point spatial par processus. Cela pourrait avoir pour effet de réduire la qualité de la parallélisation. Cependant, comme il n'y a plus d'advections locales, nous ne sommes plus limités dans le nombre de point par processus qui devait être dans la version initiale de VLEM supérieur à 11. On peut donc s'assurer d'en avoir toujours un nombre convenable pour la parallélisation.

L'implémentation de ce type de raffinement à la méthode multi-faisceaux peut être intéressant afin de rendre plus concurrent les codes Semi-Lagrangiens par rapport aux codes à particules. Cependant, cela reviendrait à modifier entièrement un code "nouveau" avec la méthode AMR et cela nécessiterait plus de temps.

Chapitre 6

Conclusions et perspectives

6.1 Conclusions

Les codes Semi-Lagrangiens permettent de décrire les plasmas de manière cinétique et sans bruit. Cependant, ils présentent l'inconvénient, à haute dimension, d'être coûteux en mémoire et en temps de calcul. La première solution proposée dans cette thèse consiste à implémenter une technique de raffinement de maillage, uniquement dans l'espace des quantités de mouvement, dans le code Semi-Lagrangien relativiste VLEM. La seconde méthode que l'on a étudiée, et qui permet de réduire les problèmes de mémoire et de temps de calcul, est la méthode multi-faisceaux.

Après une introduction, dans le chapitre 1, sur tous les éléments qui seront utilisés dans cette thèse, on a explicité en détail le fonctionnement du code Semi-Lagrangien VLEM dans le chapitre 2. Les méthodes de "splitting" de l'équation de Vlasov, de calcul des champs électromagnétiques, d'interpolations pour les advections ainsi que la parallélisation numérique du code ont ainsi été discutées. Ce chapitre a également permis de mettre en valeur les modifications apportées au code VLEM durant cette thèse, qu'il s'agisse de l'ajout ou de la modification de diagnostics, de l'amélioration de la parallélisation ou encore de l'étude de nouvelles conditions aux bords pour les splines locaux afin de réduire les problèmes de mémoire de la version 3D3V.

L'introduction détaillée du code VLEM au cours du chapitre 2 a aussi permis de présenter les éléments clés qui ont été modifiés dans la nouvelle version avec raffinement de maillage. Ainsi le chapitre 3 a permis d'introduire la méthode de grilles emboîtées adaptatives uniquement dans l'espace des quantités de mouvement qui a été choisie pour le code VLEM_EG. Les points à raffiner sont choisis afin de minimiser l'erreur L^∞ de la dérivée première sur les quantités de mouvement de la fonction distribution permettant ainsi de suivre la filamentation. L'implémentation d'une telle méthode nécessite l'ajout de coefficients de poids lors du calcul des intégrales ainsi qu'une réflexion sur les conditions aux bords de l'interpolation pour la rotation.

Cependant, cette méthode possède l'avantage de ne pas créer d'instabilités numériques sur les champs comme peuvent le faire les méthodes de raffinement dans l'espace. Elle permet par ailleurs de bien suivre les structures dans l'espace des quantités de mouvement, que ce soit de la filamentation, de l'accélération, de la décélération ou du chauffage. On a montré que le code était efficace pour réduire le temps de calcul principalement lors de la phase transitoire. Cela permet donc de réaliser, avec un bien plus faible coût numérique, des expériences sans perturbation initiale et de profiter de l'absence de bruit des codes Semi-Lagrangiens.

On a ensuite utilisé ce code VLEM_EG, dans le chapitre 4, pour étudier des instabilités dues à une anisotropie de quantité de mouvement entre deux faisceaux d'électrons de températures différentes dans un cadre relativiste. Cela a permis de s'intéresser à la saturation des modes instables et d'observer que, pour le problème que l'on a étudié, la CFI semble être l'instabilité qui sature avec la valeur la plus élevée. On a aussi pu mettre en avant les directions privilégiées d'accélérations de la fonction de distribution dans l'espace des quantités de mouvement. Puis on s'est intéressé à une instabilité Weibel purement due à une anisotropie de température et sans perturbation initiale. Cette dernière amène à une série d'accélérations de particules qui se produisent à une fréquence régulière. Ces trois simulations ont aussi permis, au travers du flux de phasestrophie d'étudier les échanges de quantités de mouvement et l'impact de la synchronisation des modes de bases du plasma sur les échanges d'énergie.

Le dernier chapitre 5 a présenté une méthode basée sur une réduction Hamiltonienne due à une invariance dans une direction de l'espace. Cette méthode permet de représenter dans un espace 2D, une fonction de distribution comme une somme de faisceaux. Cela va entraîner une réduction du temps de calcul ainsi qu'un gain en mémoire mais elle nécessite de modifier le "splitting" utilisé dans l'équation de Vlasov. Cette méthode avait déjà été présentée pour des cas 1D et l'extension y a été faite ici en 2D autorisant désormais la reconnexion magnétique.

6.2 Perspectives

On a présenté dans cette thèse deux méthodes différentes permettant de réduire le temps de calcul des codes Semi-Lagrangiens. Comme déjà introduit à la fin du chapitre 5 on pourrait combiner la méthode des grilles emboîtées avec la méthode multi-faisceaux. Cette version nécessiterait de modifier la parallélisation de la méthode multi-faisceaux mais elle pourrait permettre, en supposant que la réduction du temps de calcul se fait dans un cas idéal, de traiter un problème en 2D3V pour le coût numérique d'un problème 2D2V sans raffinement.

Une autre amélioration possible du code VLEM_EG, qui permettrait de réduire le coût actuel en mémoire sans modifier les techniques de résolution, consisterait à retirer de la mémoire de la grille de niveau R les zones déjà définies par d'autres grilles de

niveau $R^* > R$. Une idée serait de décomposer chaque grille, sauf la grille la plus raffinée, en quatre rectangles. Cette méthode nécessiterait néanmoins de rajouter des échanges d'informations entre les grilles au moment de la rotation.

Il est aussi possible d'envisager une variation des grilles emboîtées en fonction de la position (x,y) dans l'espace. Cependant, avec la méthode d'advection locale, il faudrait reconstruire la valeur des points qui n'existent pas lors du calcul des splines. Cette méthode devrait être couplée à une méthode de répartition du poids entre les processus MPI afin d'optimiser le temps de calcul de chacun d'entre eux.

La forme des grilles pourrait aussi être modifiée afin de ne plus se limiter à un seul rectangle. La version la plus simple serait de pouvoir avoir plusieurs rectangles définis sur le même niveau de précision. Cela pourrait par exemple donner un rectangle par faisceaux lors du cas des instabilités CFI, TSI et OI. Cette modification nécessiterait de rendre automatique le calcul du poids dans les intégrales, mais cela a déjà été fait dans le code.

Le passage du code sur GPU pourrait également permettre d'accélérer le code, mais cela nécessiterait une réécriture complète. À plus long terme, on peut aussi envisager un portage sur ordinateurs quantiques, mais cela nécessiterait non seulement une réécriture du code, mais aussi une nouvelle réflexion sur la méthode.

Pour étudier d'autres aspects physiques, et non plus accélérer le code, on pourrait ajouter la contribution des ions. Une première étape consisterait à ajouter des positrons pour étudier les instabilités liées aux chocs dans les pulsars, par exemple. Les positrons, qui ont la même masse que les électrons et une charge de signe opposé, peuvent ainsi être facilement implémentés dans le code. La seconde étape consisterait à ajouter des ions de masse et de charge variables afin d'étudier les instabilités liées aux chocs dans les lasers de laboratoire ou dans le vent solaire, par exemple. L'ajout de positrons au code fait actuellement l'objet d'une thèse qui a débuté en 2024.

D'autres améliorations permettant d'observer de nouveaux effets physiques pourraient également être envisagées, comme la prise en compte des collisions, soit avec des particules neutres à l'aide d'un opérateur de collision de Krook, soit avec des collisions coulombiennes en résolvant l'équation de Fokker-Planck. La prise en compte des effets quantiques (QED) pourrait également être intéressante si l'on souhaite étudier les interactions entre électrons et positrons qui peuvent se produire dans les cascades.

Toutes ces propositions d'amélioration pour étudier les aspects physiques existent déjà dans les codes PIC. Il est cependant intéressant d'étudier ces aspects avec des méthodes différentes, afin de pouvoir comparer les résultats et de tirer parti de la complémentarité de chaque méthode. C'est pourquoi cela reste d'intérêt d'étudier les codes Semi-Lagrangiens.

Annexe A

Démonstration de l'ordre 2 du schéma numérique

Dans cette annexe, on va démontrer l'ordre 2 du code relativiste Semi-Lagrangien VLEM à 2 dimensions d'espace, l'extension à 3 dimensions se faisant de la même manière, et 3 dimensions de quantité de mouvement. Pour ce faire, on va comparer à l'ordre 2 l'évolution temporelle des variables entre la résolution directe des caractéristiques de l'équation de Vlasov et la méthode Eulérienne. La comparaison va donner des résultats similaires à cet ordre. On peut donc en déduire que le schéma Eulérien utilisé dans VLEM résout les équations des caractéristiques de Vlasov à l'ordre 2 en Δt.

A.1 Formulation Lagrangienne en utilisant les caractéristiques de l'équation de Vlasov

En partant de la trajectoire d'une particule qui suit les caractéristiques de l'équation de Vlasov, il est possible de déterminer la position (x^n, y^n) et la vitesse (v_x^n, v_y^n, v_z^n) d'une particule à l'instant $t_n = n\Delta t$. On considère les équations qui décrivent cette trajectoire dans un système 2D tel que :

$$\frac{dx}{dt} = v_x(t) = \frac{p_x(t)}{m\gamma(t)}, \tag{A.1}$$

$$\frac{dy}{dt} = v_y(t) = \frac{p_y(t)}{m\gamma(t)}, \tag{A.2}$$

$$\frac{d}{dt}(mv_x\gamma) = e(E_x + v_yB_z - v_zB_y), \tag{A.3}$$

$$\frac{d}{dt}\left(mv_y\gamma\right) = e\left(E_y - v_xB_z + v_zB_x\right), \tag{A.4}$$

$$\frac{d}{dt}\left(mv_z\gamma\right) = e\left(E_z + v_xB_y - v_yB_x\right), \tag{A.5}$$

et où le facteur de Lorentz γ dans les équations (A.3-A.5) est défini par :

$$\gamma = \frac{1}{\sqrt{1 - \frac{v_x^2}{c^2} - \frac{v_y^2}{c^2} - \frac{v_z^2}{c^2}}}. \tag{A.6}$$

A partir de cette définition, on peut exprimer les termes de gauche des Eqs.(A.3-A.5), sans prendre en compte la masse qui est une constante, comme :

$$\frac{d}{dt}\left(\gamma v_x\right) = \gamma\left(1 + \frac{v_x^2\gamma^2}{c^2}\right)\frac{dv_x}{dt} + \frac{v_xv_y\gamma^3}{c^2}\frac{dv_y}{dt} + \frac{v_xv_z\gamma^3}{c^2}\frac{dv_z}{dt}, \tag{A.7}$$

$$\frac{d}{dt}\left(\gamma v_y\right) = \frac{v_xv_y\gamma^3}{c^2}\frac{dv_x}{dt} + \gamma\left(1 + \frac{v_y^2\gamma^2}{c^2}\right)\frac{dv_y}{dt} + \frac{v_yv_z\gamma^3}{c^2}\frac{dv_z}{dt}, \tag{A.8}$$

$$\frac{d}{dt}\left(\gamma v_z\right) = \frac{v_xv_z\gamma^3}{c^2}\frac{dv_x}{dt} + \frac{v_yv_z\gamma^3}{c^2}\frac{dv_y}{dt} + \gamma\left(1 + \frac{v_z^2\gamma^2}{c^2}\right)\frac{dv_z}{dt}. \tag{A.9}$$

Le système donné par les Eqs.(A.7-A.9) est un système d'équation linéaire. En utilisant la méthode de Cramer [158], on retrouve le système suivant pour l'évolution temporelle des vitesses :

$$\frac{dv_x}{dt} = \frac{1}{\gamma^3}\left(1 + \frac{v_y^2\gamma^2}{c^2} + \frac{v_z^2\gamma^2}{c^2}\right)\frac{d}{dt}\left(\gamma v_x\right) - \frac{v_xv_y}{\gamma c^2}\frac{d}{dt}\left(\gamma v_y\right) - \frac{v_xv_z}{\gamma c^2}\frac{d}{dt}\left(\gamma v_z\right), \tag{A.10}$$

$$\frac{dv_y}{dt} = -\frac{v_xv_y}{\gamma c^2}\frac{d}{dt}\left(\gamma v_x\right) + \frac{1}{\gamma^3}\left(1 + \frac{v_x^2\gamma^2}{c^2} + \frac{v_z^2\gamma^2}{c^2}\right)\frac{d}{dt}\left(\gamma v_y\right) - \frac{v_yv_z}{\gamma c^2}\frac{d}{dt}\left(\gamma v_z\right), \tag{A.11}$$

$$\frac{dv_z}{dt} = -\frac{v_xv_z}{\gamma c^2}\frac{d}{dt}\left(\gamma v_x\right) - \frac{v_yv_z}{\gamma c^2}\frac{d}{dt}\left(\gamma v_y\right) + \frac{1}{\gamma^3}\left(1 + \frac{v_x^2\gamma^2}{c^2} + \frac{v_y^2\gamma^2}{c^2}\right)\frac{d}{dt}\left(\gamma v_z\right). \tag{A.12}$$

En écrivant l'évolution temporelle de x^n et de y^n à l'ordre 2 avec la formule des trapèzes (on choisit d'exprimer α^n puisque par la suite on comparera cela avec les méthodes d'advection vers l'arrière) on a :

$$x^n = x^{n+1} - v_x^{n+\frac{1}{2}}\Delta t = x^{n+1} - \left(\frac{v_x^{n+1} + v_x^n}{2}\right)\Delta t + O\left(\Delta t^3\right), \tag{A.13}$$

$$y^n = y^{n+1} - v_y^{n+\frac{1}{2}}\Delta t = y^{n+1} - \left(\frac{v_y^{n+1} + v_y^n}{2}\right)\Delta t + O\left(\Delta t^3\right). \tag{A.14}$$

En introduisant la force de Lorentz sous la forme :

$$\begin{aligned} F_x^{n+\frac{1}{2}} &= e\left(E_x^{n+\frac{1}{2}} + v_y^{n+\frac{1}{2}}B_z^{n+\frac{1}{2}} - v_z^{n+\frac{1}{2}}B_y^{n+\frac{1}{2}}\right), \\ F_y^{n+\frac{1}{2}} &= e\left(E_y^{n+\frac{1}{2}} - v_x^{n+\frac{1}{2}}B_y^{n+\frac{1}{2}} + v_z^{n+\frac{1}{2}}B_x^{n+\frac{1}{2}}\right), \\ F_z^{n+\frac{1}{2}} &= e\left(E_z^{n+\frac{1}{2}} + v_x^{n+\frac{1}{2}}B_z^{n+\frac{1}{2}} - v_y^{n+\frac{1}{2}}B_x^{n+\frac{1}{2}}\right). \end{aligned} \tag{A.15}$$

On trouve en remplaçant dans les équations (A.10-A.12) les termes $d(\gamma v_\alpha)/dt$, pour l'évolution temporelle des vitesses :

$$\begin{aligned} v_x^n = v_x^{n+1} + \Delta t \Bigg[&-\frac{F_x^{n+\frac{1}{2}}}{m\gamma^3}\left(1 + \frac{(v_y^{n+1})^2\gamma^2}{c^2} + \frac{(v_z^{n+1})^2\gamma^2}{c^2}\right) \\ &+ \frac{v_x^{n+1}v_y^{n+1}F_y^{n+\frac{1}{2}}}{mc^2\gamma} + \frac{v_x^{n+1}v_z^{n+1}F_z^{n+\frac{1}{2}}}{mc^2\gamma}\Bigg] + O\left(\Delta t^2\right), \end{aligned} \tag{A.16}$$

$$\begin{aligned} v_y^n = v_y^{n+1} + \Delta t \Bigg[&\frac{v_x^{n+1}v_y^{n+1}F_x^{n+\frac{1}{2}}}{mc^2\gamma} \\ &-\frac{F_y^{n+\frac{1}{2}}}{m\gamma^3}\left(1 + \frac{(v_x^{n+1})^2\gamma^2}{c^2} + \frac{(v_z^{n+1})^2\gamma^2}{c^2}\right) + \frac{v_y^{n+1}v_z^{n+1}F_z^{n+\frac{1}{2}}}{mc^2\gamma}\Bigg] + O\left(\Delta t^2\right), \end{aligned} \tag{A.17}$$

$$\begin{aligned} v_z^n = v_z^{n+1} + \Delta t \Bigg[&\frac{v_x^{n+1}v_z^{n+1}F_x^{n+\frac{1}{2}}}{mc^2\gamma} + \frac{v_y^{n+1}v_z^{n+1}F_y^{n+\frac{1}{2}}}{mc^2\gamma} \\ &-\frac{F_z^{n+\frac{1}{2}}}{m\gamma^3}\left(1 + \frac{(v_x^{n+1})^2\gamma^2}{c^2} + \frac{(v_y^{n+1})^2\gamma^2}{c^2}\right)\Bigg] + O\left(\Delta t^2\right). \end{aligned} \tag{A.18}$$

En injectant cela dans les équations (A.13) et (A.14), on a alors :

$$\begin{aligned} x^n = x^{n+1} - v_x^{n+1}\Delta t + \Delta t^2 \Bigg[&\frac{F_x^{n+\frac{1}{2}}}{2m\gamma^3}\left(1 + \frac{\left(v_y^{n+1}\right)^2\gamma^2}{c^2} + \frac{(v_z^{n+1})^2\gamma^2}{c^2}\right) \\ &-\frac{v_x^{n+1}v_y^{n+1}F_y^{n+\frac{1}{2}}}{2mc^2\gamma} - \frac{v_x^{n+1}v_z^{n+1}F_z^{n+\frac{1}{2}}}{2mc^2\gamma}\Bigg] + O\left(\Delta t^3\right), \end{aligned} \tag{A.19}$$

$$
\begin{aligned}
y^n = y^{n+1} - v_y^{n+1}\Delta t + \Delta t^2 \Bigg[& -\frac{v_x^{n+1}v_y^{n+1}}{2mc^2\gamma}F_x^{n+\frac{1}{2}} \\
& + \frac{F_y^{n+\frac{1}{2}}}{2m\gamma^3}\left(1 + \frac{\left(v_x^{n+1}\right)^2\gamma^2}{c^2} + \frac{\left(v_z^{n+1}\right)^2\gamma^2}{c^2}\right) - \frac{v_y^{n+1}v_z^{n+1}F_z^{n+\frac{1}{2}}}{2mc^2\gamma}\Bigg] + O\left(\Delta t^3\right).
\end{aligned}
\tag{A.20}
$$

En raisonnant de la même manière pour les quantités de mouvement p_x, p_y et p_z et en développant les vitesses dans $F_\alpha^{n+\frac{1}{2}}$ on trouve :

$$
\begin{aligned}
p_x^n = p_x^{n+1} - \mathcal{F}_x^{n+\frac{1}{2}}\Delta t - \frac{eB_z^{n+\frac{1}{2}}\Delta t^2}{2m\gamma} \Bigg[& \frac{v_x^{n+1}v_y^{n+1}F_x^{n+\frac{1}{2}}}{c^2} \\
& - \frac{F_y^{n+\frac{1}{2}}}{\gamma^2}\left(1 + \frac{\left(v_x^{n+1}\right)^2\gamma^2}{c^2} + \frac{\left(v_z^{n+1}\right)^2\gamma^2}{c^2}\right) + \frac{v_y^{n+1}v_z^{n+1}F_z^{n+\frac{1}{2}}}{c^2}\Bigg] \\
+ \frac{eB_y^{n+\frac{1}{2}}\Delta t^2}{2m\gamma} \Bigg[& \frac{v_x^{n+1}v_z^{n+1}F_x^{n+\frac{1}{2}}}{c^2} + \frac{v_y^{n+1}v_z^{n+1}F_y^{n+\frac{1}{2}}}{c^2} \\
& - \frac{F_z^{n+\frac{1}{2}}}{\gamma^2}\left(1 + \frac{\left(v_x^{n+1}\right)^2\gamma^2}{c^2} + \frac{(v_y^{n+1})^2\gamma^2}{c^2}\right)\Bigg] + O\left(\Delta t^3\right),
\end{aligned}
\tag{A.21}
$$

$$
\begin{aligned}
p_y^n = p_y^{n+1} - \mathcal{F}_y^{n+\frac{1}{2}}\Delta t - \frac{eB_x^{n+\frac{1}{2}}\Delta t^2}{2m\gamma} \Bigg[& \frac{v_x^{n+1}v_z^{n+1}F_x^{n+\frac{1}{2}}}{c^2} \\
& + \frac{v_y^{n+1}v_z^{n+1}F_y^{n+\frac{1}{2}}}{c^2} - \frac{F_z^{n+\frac{1}{2}}}{\gamma^2}\left(1 + \frac{\left(v_x^{n+1}\right)^2\gamma^2}{c^2} + \frac{\left(v_y^{n+1}\right)^2\gamma^2}{c^2}\right)\Bigg] \\
+ \frac{eB_z^{n+\frac{1}{2}}\Delta t^2}{2m\gamma} \Bigg[& -\frac{F_x^{n+\frac{1}{2}}}{\gamma^2}\left(1 + \frac{\left(v_y^{n+1}\right)^2\gamma^2}{c^2} + \frac{\left(v_z^{n+1}\right)^2\gamma^2}{c^2}\right) \\
& + \frac{v_x^{n+1}v_y^{n+1}F_y^{n+\frac{1}{2}}}{c^2} + \frac{v_x^{n+1}v_z^{n+1}F_z^{n+\frac{1}{2}}}{c^2}\Bigg] + O\left(\Delta t^3\right),
\end{aligned}
\tag{A.22}
$$

$$p_z^n = p_z^{n+1} - \mathcal{F}_z^{n+\frac{1}{2}}\Delta t - \frac{eB_y^{n+\frac{1}{2}}\Delta t^2}{2m\gamma}\left[-\frac{F_x^{n+\frac{1}{2}}}{\gamma^2}\left(1+\frac{\left(v_y^{n+1}\right)^2\gamma^2}{c^2}+\frac{\left(v_z^{n+1}\right)^2\gamma^2}{c^2}\right)+\frac{v_x^{n+1}v_y^{n+1}F_y^{n+\frac{1}{2}}}{c^2}+\frac{v_x^{n+1}v_z^{n+1}F_z^{n+\frac{1}{2}}}{c^2}\right]+\frac{eB_x^{n+\frac{1}{2}}\Delta t^2}{2m\gamma}\left[\frac{v_x^{n+1}v_y^{n+1}F_x^{n+\frac{1}{2}}}{c^2}-\frac{F_y^{n+\frac{1}{2}}}{\gamma^2}\left(1+\frac{\left(v_x^{n+1}\right)^2\gamma^2}{c^2}+\frac{\left(v_z^{n+1}\right)^2\gamma^2}{c^2}\right)+\frac{v_y^{n+1}v_z^{n+1}F_z^{n+\frac{1}{2}}}{c^2}\right]+O\left(\Delta t^3\right), \tag{A.23}$$

avec $\mathcal{F}_\alpha$ les forces de Lorentz où les vitesses moyennes $\frac{v_\alpha^n+v_\alpha^{n+1}}{2}$ sont remplacées par v_α^{n+1}. À noter que pour les équations (A.19-A.23), on peut remplacer toutes les forces F_α par $\mathcal{F}_\alpha$ en restant au même ordre car on a : $\mathcal{F}_\alpha = F_\alpha + O(\Delta t)$.

A.2 Formulation Eulérienne

On va maintenant regarder le point de vue des méthodes Eulériennes, donc celle utilisée dans VLEM. On suit la séquence $(\mathcal{X}/2)(\mathcal{Y}/2)(\mathcal{R})(\mathcal{Y}/2)(\mathcal{X}/2)$ qui est appliquée à la fonction de distribution $f^n(x, y, p_x, p_y, p_z, t = n\Delta t)$.

On a donc :

$$f^{n+1}(x, y, p_x, p_y, p_z) = f^n(\overline{x}, \overline{y}, \overline{p_x}, \overline{p_y}, \overline{p_z}). \tag{A.24}$$

Pour comparer avec la sous-partie précédente, les variables $(\overline{x}, \overline{y}, \overline{p_x}, \overline{p_y}, \overline{p_z})$ représentent l'ancienne position de la "particule" (α^n) et les variables (x, y, p_x, p_y, p_z) la nouvelle position de la "particule" (α^{n+1}). Si on reprend les équations d'advection du schéma, on a donc :

$$\overline{x} = x - \frac{p_x}{m\gamma}\frac{\Delta t}{2} - \frac{\overline{p_x}}{m\overline{\gamma}}\frac{\Delta t}{2} + O\left(\Delta t^3\right), \tag{A.25}$$

$$\overline{y} = y - \frac{p_y}{m\gamma}\frac{\Delta t}{2} - \frac{\overline{p_y}}{m\overline{\gamma}}\frac{\Delta t}{2} + O\left(\Delta t^3\right), \tag{A.26}$$

avec $\overline{\gamma} = \overline{\gamma}(\overline{p_x}, \overline{p_y}, \overline{p_z})$ et

$$\overline{p_x} = p_x - e\left(E_x^* + \frac{v_y + \overline{v_y}}{2}B_z^* - \frac{v_z + \overline{v_z}}{2}B_y^*\right)\Delta t + O\left(\Delta t^3\right), \tag{A.27}$$

$$\overline{p_y} = p_y - e\left(E_y^* + \frac{v_z + \overline{v_z}}{2}B_x^* - \frac{v_x + \overline{v_x}}{2}B_z^*\right)\Delta t + O\left(\Delta t^3\right), \tag{A.28}$$

$$\overline{p_z} = p_z - e\left(E_z^* + \frac{v_x + \overline{v_x}}{2}B_y^* - \frac{v_y + \overline{v_y}}{2}B_x^*\right)\Delta t + O\left(\Delta t^3\right). \tag{A.29}$$

E_i^* et B_i^* les champs électrique et magnétique à $n + \frac{1}{2}$. On introduit la contribution de la force de Lorentz sous la forme :

$$\begin{aligned} F_x^* &= e\left(E_x^* + v_y B_z^* - v_z B_y^*\right), \\ F_y^* &= e\left(E_y^* - v_x B_z^* + v_z B_x^*\right), \\ F_z^* &= e\left(E_z^* + v_x B_y^* - v_y B_x^*\right). \end{aligned} \tag{A.30}$$

En exprimant le facteur de Lorentz à la fin de la séquence, on trouve :

$$\frac{1}{\overline{\gamma}_j} = \frac{1}{\gamma} + \left[p_x F_x^* + p_y F_y^* + p_z F_z^*\right]\frac{\Delta t}{m^2c^2\gamma^3} + O\left(\Delta t^2\right). \tag{A.31}$$

On peut donc développer les équations (A.25-A.26) avec les équations (A.27-A.28) et (A.31) :

$$\overline{x} = x - \frac{p_x}{m\gamma}\Delta t - \frac{\Delta t^2}{2m\gamma^3}\left[F_x^*\left(\frac{p_x^2}{m^2c^2} - \gamma^2\right) + \frac{p_x p_y F_y^*}{m^2c^2} + \frac{p_x p_z F_z^*}{m^2c^2}\right] + O\left(\Delta t^3\right), \tag{A.32}$$

$$\overline{y} = y - \frac{p_y}{m\gamma}\Delta t - \frac{\Delta t^2}{2m\gamma^3}\left[\frac{p_x p_y F_x^*}{m^2c^2} + F_y^*\left(\frac{p_y^2}{m^2c^2} - \gamma^2\right) + \frac{p_y p_z F_z^*}{m^2c^2}\right] + O\left(\Delta t^3\right). \tag{A.33}$$

En exprimant les vitesses par rapport aux quantités de mouvement et en utilisant les Eqs.(A.27-A.29) et (A.31) :

$$\overline{v_x} \equiv \frac{\overline{p_x}}{m\overline{\gamma}} = \frac{p_x}{m\gamma} + \frac{\Delta t}{m\gamma^3}\left[F_x^*\left(\frac{p_x^2}{m^2c^2} - \gamma^2\right) + \frac{p_x p_y F_y^*}{m^2c^2} + \frac{p_x p_z F_z^*}{m^2c^2}\right] + O\left(\Delta t^2\right), \tag{A.34}$$

$$\overline{v_y} \equiv \frac{\overline{p_y}}{m\overline{\gamma}} = \frac{p_y}{m\gamma} + \frac{\Delta t}{m\gamma^3}\left[\frac{p_x p_y F_x^*}{m^2c^2} + F_y^*\left(\frac{p_y^2}{m^2c^2} - \gamma^2\right) + \frac{p_y p_z F_z^*}{m^2c^2}\right] + O\left(\Delta t^2\right), \tag{A.35}$$

$$\overline{v_z} \equiv \frac{\overline{p_z}}{m\overline{\gamma}} = \frac{p_z}{m\gamma} + \frac{\Delta t}{m\gamma^3}\left[\frac{p_x p_z F_x^*}{m^2c^2} + \frac{p_y p_z F_y^*}{m^2c^2} + F_z^*\left(\frac{p_z^2}{m^2c^2} - \gamma^2\right)\right] + O\left(\Delta t^2\right). \tag{A.36}$$

Enfin, en injectant l'expression de ces vitesses dans Eqs.(A.27-A.29) on trouve pour les quantités de mouvement :

$$\begin{aligned}\overline{p_x} = p_x - F_x^*\Delta t - \Delta t^2\frac{eB_z^*}{2m\gamma^3}\left[\frac{p_xp_y}{m^2c^2}F_x^* + F_y^*\left(\frac{p_y^2}{m^2c^2}-\gamma^2\right) + \frac{p_yp_z}{m^2c^2}F_z^*\right]\\ +\Delta t^2\frac{eB_y^*}{2m\gamma^3}\left[\frac{p_xp_z}{m^2c^2}F_x^* + \frac{p_yp_z}{m^2c^2}F_y^* + F_z^*\left(\frac{p_z^2}{m^2c^2}-\gamma^2\right)\right] + O\left(\Delta t^3\right),\end{aligned} \tag{A.37}$$

$$\begin{aligned}\overline{p_y} = p_y - F_y^*\Delta t - \Delta t^2\frac{eB_x^*}{2m\gamma^3}\left[\frac{p_xp_z}{m^2c^2}F_x^* + \frac{p_yp_z}{m^2c^2}F_y^* + F_z^*\left(\frac{p_z^2}{m^2c^2}-\gamma^2\right)\right]\\ +\Delta t^2\frac{eB_z^*}{2m\gamma^3}\left[F_x^*\left(\frac{p_x^2}{m^2c^2}-\gamma^2\right) + \frac{p_xp_y}{m^2c^2}F_y^* + \frac{p_xp_z}{m^2c^2}F_z^*\right] + O\left(\Delta t^3\right),\end{aligned} \tag{A.38}$$

$$\begin{aligned}\overline{p_z} = p_z - F_z^*\Delta t - \Delta t^2\frac{eB_y^*}{2m\gamma^3}\left[F_x^*\left(\frac{p_x^2}{m^2c^2}-\gamma^2\right) + \frac{p_xp_y}{m^2c^2}F_y^* + \frac{p_xp_z}{m^2c^2}F_z^*\right]\\ +\Delta t^2\frac{eB_x^*}{2m\gamma^3}\left[\frac{p_xp_y}{m^2c^2}F_x^* + F_y^*\left(\frac{p_y^2}{m^2c^2}-\gamma^2\right) + \frac{p_yp_z}{m^2c^2}F_z^*\right] + O\left(\Delta t^3\right).\end{aligned} \tag{A.39}$$

En comparant les équations entre la résolution des caractéristiques Eqs.(A.19-A.23) et celles de la méthode Eulérienne Eqs.(A.31-A.32) et (A.37-A.39), on voit sans difficulté que la méthode de séparation intègre la fonction de distribution en suivant les caractéristiques à l'ordre 2 en Δt.

Annexe B

Complément sur VLEM3D3V

B.1 Valeurs des $\breve{\omega}$

On rappelle l'équation pour déterminer les conditions aux bords de la quasi-interpolation :

$$\eta_{i_{start}-1} = 6 * \sum_{k=-j}^{j} \breve{\omega}_k \breve{F}(j+k+1), \tag{B.1}$$

avec $\breve{\omega}_k$ les valeurs du vecteur $\breve{M}^{-1}(j+1,k)$. On a $\breve{\omega}_i = \breve{\omega}_{-i}$. On donne les valeurs des $\breve{\omega}_k$ dans les tableaux suivants pour $j = 7$, 5 et 3. Le premier cas n'a pas été abordé dans le manuscrit mais va servir de comparatif avec l'ancienne méthode puisqu'elle nécessite 10 points (j+3).

$\breve{\omega}_{-7}$	$\breve{\omega}_{-6}$	$\breve{\omega}_{-5}$	$\breve{\omega}_{-4}$
$-\frac{1}{37634}$	$\frac{4}{37634}$	$-\frac{15}{37634}$	$\frac{56}{37634}$
$\breve{\omega}_{-3}$	$\breve{\omega}_{-2}$	$\breve{\omega}_{-1}$	$\breve{\omega}_{0}$
$-\frac{209}{37634}$	$\frac{780}{37634}$	$-\frac{2911}{37634}$	$\frac{10864}{37634}$

TABLE B.1 – Valeurs des $\breve{\omega}$ pour $j = 7$.

$\breve{\omega}^{-}_{-5}$	$\breve{\omega}^{-}_{-4}$	$\breve{\omega}^{-}_{-3}$
$-\frac{1}{2702}$	$\frac{4}{2702}$	$-\frac{15}{2702}$
$\breve{\omega}^{-}_{-1}$	$\breve{\omega}^{-}_{-1}$	$\breve{\omega}^{-}_{0}$
$\frac{56}{2702}$	$-\frac{209}{2702}$	$\frac{780}{2702}$

TABLE B.2 – Valeurs des $\breve{\omega}$ pour $j = 5$.

$\breve{\omega}_{-3}^{-}$	$\breve{\omega}_{-2}^{-}$
$-\frac{1}{194}$	$\frac{4}{194}$
$\breve{\omega}_{-1}^{-}$	$\breve{\omega}_{0}^{-}$
$-\frac{15}{194}$	$\frac{56}{194}$

TABLE B.3 – Valeurs des $\breve{\omega}$ pour $j = 3$.

B.2 Coefficients ω sous forme rationnelle

Dans cette partie on va fournir l'écriture fractionnaire des paramètres ω pour la méthode d'estimation de la dérivée première et pour la quasi-interpolation. Cette écriture permet de donner la valeur exacte des coefficients sans être limité par le nombre de chiffres significatifs.

B.2.1 Méthode de la dérivée première $\partial_x s(x_i)$

On rappelle la forme générale de l'estimation de la dérivée aux bords de la fonction interpolante s :

$$\partial_x s(x_i) = \frac{1}{h}\left(\sum_{j=-N}^{-1} \omega_j^- f_{i+j} + \sum_{j=1}^{N} \omega_j^+ f_{i+j}\right). \tag{B.2}$$

On pose $2N$ le nombre de points nécessaires pour estimer la dérivée. On rappelle $\omega_j^- =$ -ω_j^+ On donne les tableaux suivants pour l'écriture fractionnaire des paramètres ω_j^-.

ω_{-10}^{-}	ω_{-9}^{-}	ω_{-8}^{-}	ω_{-7}^{-}	ω_{-6}^{-}
$\frac{1}{451608}$	$-\frac{8}{451608}$	$\frac{36}{451608}$	$-\frac{136}{451608}$	$\frac{503}{451608}$
ω_{-5}^{-}	ω_{-4}^{-}	ω_{-3}^{-}	ω_{-2}^{-}	ω_{-1}^{-}
$-\frac{1872}{451608}$	$\frac{6984}{451608}$	$-\frac{26064}{451608}$	$\frac{97272}{451608}$	$-\frac{363024}{451608}$

TABLE B.4 – Valeur des ω^- pour $N = 10$.

ω_{-6}^{-}	ω_{-5}^{-}	ω_{-4}^{-}
$\frac{1}{2328}$	$-\frac{8}{2328}$	$\frac{36}{2328}$
ω_{-3}^{-}	ω_{-2}^{-}	ω_{-1}^{-}
$-\frac{136}{2328}$	$\frac{503}{2328}$	$-\frac{1872}{2328}$

TABLE B.5 – Valeur des ω^- pour $N = 6$.

B.2.2 Méthode de la quasi-interpolation

On rappelle les formules pour estimer les coefficients splines aux bords avec la méthode de la quasi-interpolation :

$$\begin{aligned}\eta_{i_{start}-1} &= \sum_{k=-j-2}^{j+2} \omega_k f_{i_{start}-1+k},\\ \eta_{i_{end}+1} &= \sum_{k=-j-2}^{j+2} \omega_k f_{i_{end}+1+k},\end{aligned} \tag{B.3}$$

avec $\omega_i = \omega_{-i}$.

ω_{-9}	ω_{-8}	ω_{-7}	ω_{-6}	ω_{-5}
$-\frac{1}{225804}$	$\frac{8}{225804}$	$-\frac{37}{225804}$	$\frac{144}{225804}$	$-\frac{540}{225804}$
ω_{-4}	ω_{-3}	ω_{-2}	ω_{-1}	ω_0^-
$\frac{2016}{225804}$	$-\frac{7524}{225804}$	$\frac{28080}{225804}$	$-\frac{104796}{225804}$	$\frac{391104}{225804}$

TABLE B.6 – Valeur des ω pour $j = 7$.

ω_{-8}	ω_{-7}	ω_{-6}	ω_{-5}
$-\frac{1}{16212}$	$\frac{8}{16212}$	$-\frac{37}{16212}$	$\frac{144}{16212}$
ω_{-3}	ω_{-2}	ω_{-1}	ω_0
$-\frac{540}{16212}$	$\frac{2016}{16212}$	$-\frac{7524}{16212}$	$\frac{28080}{16212}$

TABLE B.7 – Valeur des ω pour $j = 5$.

ω_{-5}	ω_{-4}	ω_{-3}
$-\frac{1}{1164}$	$\frac{8}{1164}$	$-\frac{37}{1164}$
ω_{-2}	ω_{-1}	ω_0
$\frac{144}{1164}$	$-\frac{540}{1164}$	$\frac{2016}{1164}$

TABLE B.8 – Valeur des ω pour $j = 3$.

B.3 Estimation de l'erreur L^∞ des méthodes

B.3.1 Méthode de la quasi-interpolation

On a déjà estimé l'erreur L^∞ sur $\eta_{i_{start}-2-j}$ et $\eta_{i_{end}+2+j}$ à l'équation (2.109). On va regarder le cas $\eta_{i_{start}-2-j}$ et on pourra en déduire l'erreur découlant de $\eta_{i_{end}-2-j}$.

On a donc :

$$\epsilon_{\eta_{i_{start}-2-j}} = \frac{4}{9}|f|_\infty. \tag{B.4}$$

Sachant que $\breve{F}$ vaut $(f(x_{i_{start}-1-j})\text{-}\eta_{i_{start}-2-j}/6, f(x_{i_{start}-j}), \cdots, f(x_{i_{start}-2+j}), f(x_{i_{start}-1+j})\text{-}\eta_{i_{start}+j}/6)^T$, on a :

$$\epsilon_{\breve{F}(1)=\breve{F}(2j+1)} = \frac{1}{6} \times \frac{4}{9}|f|_\infty. \tag{B.5}$$

En utilisant la formule que l'on a rappelée à l'équation (B.1), on a :

$$\epsilon_{\eta_{i_{start}-1}} = 6 \times (\breve{\omega_{-j}}\epsilon_{\breve{F}(1)} + \breve{\omega_j}\epsilon_{\breve{F}(2j+1)}). \tag{B.6}$$

La fonction la plus proche qui va être calculée, et donc celle sur laquelle l'erreur sera la plus importante, est $f(x_{i_{start}-1})$ avec l'équation (2.43). On calcule son erreur, en négligeant celle provenant de $\eta_{i_{end}+1}$ qui aura un impact bien plus faible car suffisamment éloigné et on trouve :

$$\epsilon_{f(x_{i_{start}-1})} = \frac{4 \times 6}{6} \times (\breve{\omega_{-j}}\epsilon_{\breve{F}(1)} + \breve{\omega_j}\epsilon_{\breve{F}(2j+1)}). \tag{B.7}$$

On trouve pour $j = 7$:

$$\epsilon_{f(x_{i_{start}-1})} = \frac{8}{508059}|f|_\infty \sim 1,575E^{-5}|f|_\infty, \tag{B.8}$$

pour $j = 5$:

$$\epsilon_{f(x_{i_{start}-1})} = \frac{8}{36477}|f|_\infty \sim 2,19E^{-4}|f|_\infty, \tag{B.9}$$

et pour $j = 3$:

$$\epsilon_{f(x_{i_{start}-1})} = \frac{8}{2619}|f|_\infty \sim 3,05E^{-3}|f|_\infty. \tag{B.10}$$

B.3.2 Méthode d'estimation de la dérivée première

On cherche à déterminer l'erreur en norme L^∞ faite sur $s'_{i\pm8}$ quand on utilise la formule :

$$s'_{i+8} = \frac{f_{i+6} - 8f_{i+7} + 8f_{i+9} - f_{i+10}}{12h}. \tag{B.11}$$

La formule la plus exacte que l'on a trouvée pour exprimer s'_i est celle que l'on a rappelée équation (B.2), on peut donc écrire l'erreur sur $s'_{i\pm8}$ comme :

$$\begin{aligned}\epsilon_{s'_{i+8}} =& \frac{1}{h}\left(\sum_{j=-10}^{-1} \omega_j^- f_{i+j+8} + \sum_{j=1}^{10} \omega_j^+ f_{i+j+8}\right) \\ &- \frac{f_{i+6} - 8f_{i+7} + 8f_{i+9} - f_{i+10}}{12h} + \frac{2}{37636}\epsilon_{s'_{i+8}},\end{aligned} \tag{B.12}$$

avec le facteur $\frac{2}{37636}\epsilon$ qui vient des expressions de s'_0 et s'_{16} qui ont permis de trouver l'équation (B.2).

On peut donc réécrire :

$$\begin{aligned}\frac{37634h}{37636}\epsilon_{s'_{i+8}} = &\left(\sum_{j=-10}^{-3}\omega_j^- f_{i+j+8} + \sum_{j=3}^{10}\omega_j^+ f_{i+j+8}\right)\\ &+ f_{i+6}\left(-\frac{1}{12}+\omega_{-2}^-\right) + f_{i+7}\left(\frac{8}{12}+\omega_{-1}^-\right)\\ &+ f_{i+10}\left(\frac{1}{12}+\omega_2^+\right) + f_{i+9}\left(-\frac{8}{12}+\omega_1^+\right),\end{aligned} \tag{B.13}$$

$$\begin{aligned}\frac{37634h}{37636}\epsilon_{s'_{i+8}} = &\left[\left(\sum_{j=-10}^{-3}|\omega_j^-| + \sum_{j=3}^{10}|\omega_j^+|\right)\right.\\ &+ \left|\frac{1}{12}+\omega_{-2}^-\right| + \left|-\frac{8}{12}+\omega_{-1}^-\right|\\ &\left.+ \left|-\frac{1}{12}+\omega_2^+\right| + \left|\frac{8}{12}+\omega_1^+\right|\right]|f|_\infty,\end{aligned} \tag{B.14}$$

$$\epsilon_{s'_{i+8}} = 0,696189415\frac{|f|_\infty}{h}. \tag{B.15}$$

Or dans l'équation (2.56), on peut déterminer $\epsilon_{s'_i}$ comme :

$$\epsilon_{s'_i} = \frac{18818}{18817}\frac{2}{37636}\epsilon_{s'_{i+8}} = 3,69979E^{-5}\frac{|f|_\infty}{h}. \tag{B.16}$$

L'erreur que l'on recherche est celle faite sur f_i en prenant les équations (2.43) et (2.44), on trouve :

$$\epsilon_{f(x_{i_{start}-1})} = \frac{8h}{6}\epsilon_{s'_i} = 4,93305E^{-5}|f|_\infty. \tag{B.17}$$

Pour la deuxième méthode présentée dans la sous-partie 2.4.3 avec la dérivée utilisant 6 points dans chaque direction on trouve, en utilisant l'erreur exprimée Eq.(B.15) pour $\epsilon_{s'_{i+4}}$:

$$\epsilon_{s'_i} = \frac{2}{194}\epsilon_{s'_{i+8}} = 7,17721E^{-3}\frac{|f|_\infty}{h}, \tag{B.18}$$

et donc

$$\epsilon_{f(x_{i_{start}-1})} = \frac{8h}{6}\epsilon_{s'_i} = 9,569614E^{-2}|f|_\infty. \tag{B.19}$$

On observe une différence d'un multiple de 4 entre les deux méthodes ce qui rend légèrement plus avantageuse la méthode de la quasi-interpolation. Mais le plus gros avantage de cette méthode est qu'elle n'est pas limitée dans le nombre de points puisque l'on peut choisir la valeur de j que l'on souhaite.

Annexe C

Description des méthodes de raffinement par ondelettes

C.1 Rappel sur les ondelettes

Dans cette partie, on va rappeler les éléments sur la théorie des ondelettes qui vont nous être utiles par la suite. On définit la fonction d'échelle, de filtre a_k comme :

$$\varphi\left(\frac{x}{2}\right) = \sum_{k\in\mathbb{Z}} a_k \varphi\left(x-k\right), \tag{C.1}$$

avec $\sum_{k\in\mathbb{Z}} a_k = 2$ en normalisation L^∞. Cette fonction est directement liée à la méthode d'interpolation et d'estimation des points (voir figure C.1) tel que :

$$\begin{aligned} f_{R+1}\left(2i_x\right) &= \sum_{k\in\mathbb{Z}} a_{2k} f_R\left(i_x-k\right), \\ f_{R+1}\left(2i_x+1\right) &= \sum_{k\in\mathbb{Z}} a_{2k+1} f_R\left(i_x-k\right). \end{aligned} \tag{C.2}$$

On associe à cette fonction d'échelle (φ) une fonction d'échelle duale (φ^*) telle que :

$$\varphi^*\left(\frac{x}{2}\right) = \sum_{k\in\mathbb{Z}} a_k^* \varphi^*\left(x-k\right), \tag{C.3}$$

qui doit vérifier :

$$\sum a_k a^*_{k-2p} = 2\delta(p). \tag{C.4}$$

Cette fonction d'échelle duale est liée à la remontée d'information des niveaux fins vers les niveaux grossiers (voir figure C.1) tel que :

$$f_R\left(i_x\right) = \frac{1}{2}\sum_{k\in\mathbb{Z}} a_k^* f_{R+1}\left(2i_x-k\right), \tag{C.5}$$

avec le facteur $1/2$ qui provient de la normalisation L^∞ des a_k^*.

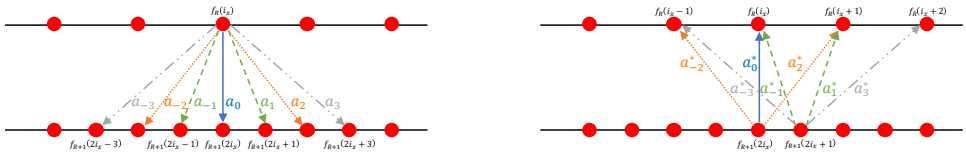

FIGURE C.1 – Représentation de la création et de la remontée d'informations entre la grille R et R+1 pour un raffinement quelconque.

C.2 Description de VLEM_EG par ondelettes

Lorsque l'on utilise les ondelettes pour interpoler aux points, on veut que $a_0 = 1$ et $a_{2k} = 0 \; \forall k \neq 0$. En effet, la valeur que prendra la grille raffinée sur un point déjà existant lorsqu'elle est créée est égale à la valeur du point de la grille grossière.

Comme on choisit d'interpoler les points milieux par la méthode Lagrange 4 points d'ordre 4. On peut écrire :

$$a_k = (a_{-3}, a_{-2}, a_{-1}, a_0, a_1, a_2, a_3) = \frac{1}{16}(-1, 0, 9, 16, 9, 0, -1). \tag{C.6}$$

En effet, on voit directement en remplaçant dans (C.2) que l'on retrouve bien l'interpolation de Lagrange à 4 points utilisée dans VLEM_EG (voir figure C.2).

On cherche donc maintenant les coefficients de la duale a_k^*. Pour cela, on va utiliser l'équation (C.4). Les coefficients a_k^* les plus simples qui sont résultats de ce système d'équations, c'est-à-dire ceux de la fonction d'échelle duale minimale, sont $a_0^* = 2$ et $a_k^* = 0 \; \forall k \neq 0$. Ce qui revient à dire que la fonction d'échelle duale minimale est la distribution de Dirac centrée en 0. En effet, c'est la seule fonction qui correspond à $\varphi^*(x) = 2\varphi^*(x/2)$. Donc pour remonter l'information il suffit juste de la communiquer entre les points situés au même endroit de l'espace des phases ($f_R(i_x) = f_{R+1}(2i_x)$, voir l'équation (C.5) et la figure C.2).

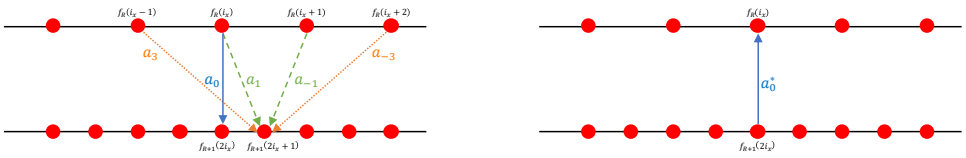

FIGURE C.2 – Représentation de la création et de la remontée d'informations entre la grille R et $R+1$ pour le raffinement dans VLEM_EG utilisant l'interpolation de Lagrange avec 4 points.

On peut aussi chercher à trouver une duale faisant intervenir plus de coefficients, par exemple :

$$a_k^* = (a_{-4}^*, a_{-3}^*, a_{-2}^*, a_{-1}^*, a_0^*, a_1^*, a_2^*, a_3^*, a_4^*) = \frac{1}{32}(1, 0, -8, 16, 46, 16, -8, 0, 1). \quad \text{(C.7)}$$

Cependant cette dernière n'améliorerait pas forcément la remontée d'informations. De plus, elle complexifierait le code au niveau des calculs des advections et des intégrales puisque la fonction de distribution n'aurait pas la même valeur entre 2 niveaux de grille au même endroit de l'espace des phases.

C.3 Description des ondelettes splines

Une des pistes explorées plutôt que d'utiliser l'interpolation de Lagrange sur la fonction de distribution était d'utiliser directement les splines cubiques.

On trouve dans la littérature [159] la formule suivante pour les splines cubiques :

$$m(z) = \sum_k a_k z^k = 2z^{\lfloor -\frac{n}{2} \rfloor} \left(\frac{1+z}{2} \right)^{n+1}, \quad \text{(C.8)}$$

où $\lfloor x \rfloor$ désigne l'arrondie à l'entier inférieur de x et avec n le degré de la spline. On peut noter que pour la spline n=0, on retrouve les ondelettes de Harr avec $a_0 = a_1 = 1$ mais nous allons nous intéresser aux spline cubiques (n=3) pour lesquelles on a :

$$a_k = (a_{-2}, a_{-1}, a_0, a_1, a_2) = \frac{1}{8}(1, 4, 6, 4, 1). \quad \text{(C.9)}$$

On peut trouver facilement la fonction d'échelle minimale à l'aide de l'équation (C.4) comme $a_k^* = (a_{-1}^*, a_0^*, a_1^*) = (-1, 4, -1)$ et une autre fonction d'échelle duale faisant intervenir plus de coefficients comme $a_k^* = (3, -12, 5, 40, 5, -12, 3)/16$. Cependant, aucune de ces deux fonctions d'échelle n'est de norme L^2.

Un autre problème de ces méthodes et qu'elles fonctionnent sur les coefficients de splines. C'est-à-dire que la fonction f dans les équations (C.2) et (C.5) représente η les coefficients de spline. Un raffinement de maillage faisant intervenir ces fonctions seraient donc plus complexe puisqu'il faudrait recalculer les coefficients η afin de trouver les valeurs de la fonction de distribution à chaque fois que l'on remonte l'information. Le temps de calcul du code serait donc plus long.

C.4 Description des ondelettes d'Arakawa

On rappelle que pour le raffinement avec les ondelettes d'Arakawa, les points raffinés sont situé en $1/4$. On a donc $f_{R+1}(2i_x)$ ($f_{R+1}(2i_x+1)$) qui est reculé (avancé) de $1/4$ de $\Delta x(R)$ par rapport à $f_R(i_x)$ (voir figure C.3).

Pour créer les points, situé en $1/4$, on a choisi de prendre, la fonction d'ordre 3 suivante :

$$f\left(\frac{h}{4}\right) = \frac{1}{32}\left(-3f(-h) + 30f(0) + 5f(h)\right) + \frac{5}{128}h^3 f^{(3)} + O(h^4). \tag{C.10}$$

On peut donc écrire la fonction d'échelle comme :

$$\begin{aligned}\varphi\left(\frac{x}{2}\right) = \frac{1}{32}\big(&- 3\varphi(x+2) + 5\varphi(x+1) + 30\varphi(x) \\ &+ 30\varphi(x-1) + 5\varphi(x-2) - 3\varphi(x-3)\big),\end{aligned} \tag{C.11}$$

avec les coefficients a_k :

$$a_k = (a_{-2}, a_{-1}, a_0, a_1, a_2, a_3) = \frac{1}{32}(-3, 5, 30, 30, 5, -3). \tag{C.12}$$

En utilisant l'équation (C.4), on trouve :

$$a_k^* = (a_{-2}^*, a_{-1}^*, a_0^*, a_1^*, a_2^*, a_3^*) = \frac{1}{112}(-3, -5, 120, 120, -5, -3). \tag{C.13}$$

En injectant les a_k et les a_k^* dans les équations (C.2) et (C.5) on retrouve les formules pour créer la grille raffinée et remonter l'information sur la grille la plus grossière :

$$\begin{aligned} f_{R+1}(2i_x) &= \frac{5}{32}f_R(i_x - 1) + \frac{30}{32}f_R(i_x) - \frac{3}{32}f_R(i_x + 1), \\ f_{R+1}(2i_x + 1) &= -\frac{3}{32}f_R(i_x - 1) + \frac{30}{32}f_R(i_x) + \frac{5}{32}f_R(i_x + 1), \end{aligned} \tag{C.14}$$

$$\begin{aligned} f_R(i_x) = &- \frac{3}{224}f_{R+1}(2i_x - 2) - \frac{5}{224}f_{R+1}(2i_x - 1) + \frac{120}{224}f_{R+1}(2i_x) \\ &+ \frac{120}{224}f_{R+1}(2i_x + 1) - \frac{5}{224}f_{R+1}(2i_x + 1) - \frac{3}{224}f_{R+1}(2i_x + 2). \end{aligned} \tag{C.15}$$

Ces dernières sont représentées dans la figure C.3.

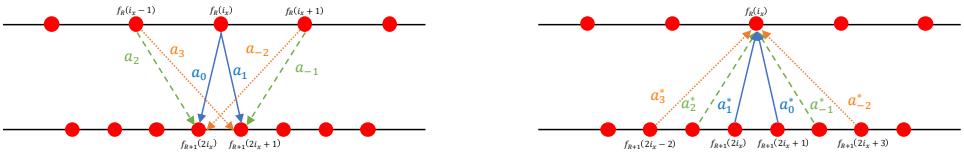

FIGURE C.3 – Représentation de la création et de la remontée d'informations entre la grille R et $R+1$ pour un raffinement avec les ondelettes d'Arakawa compatible avec la méthode de Yee.

Annexe D

Études des conditions aux bords

Dans cette annexe, nous allons étudier l'erreur engendrée par la méthode de grille emboîtée en utilisant différentes techniques pour calculer les conditions aux bords de la grille lors de la rotation (voir sous-partie 2.2.5).

Dans le code VLEM, c'est la condition aux bords naturelle qui est utilisée pour les splines cubiques servant à la rotation. Cette dernière pose la dérivée seconde de la fonction interpolée aux bords comme nulle ($s^{''}(j_{x_{min}}) = s^{''}(j_{x_{max}}) = 0$). Pour le code VLEM, cette condition est respectée, puisque la fonction est presque nulle et n'évolue que peu aux extrémités de la boîte. Cependant, ce n'est pas forcément le cas pour une grille interne dans le cadre de la méthode AMR. Nous allons donc vérifier la véracité de cette condition et en proposer d'autres.

Le choix a été fait de conserver les splines cubiques avec conditions aux bords sur la dérivée seconde afin de ne pas modifier l'implémentation de cette méthode dans le code. En effet, il aurait aussi été possible de modifier la méthode en estimant la dérivée première, mais les résultats obtenus avec la dérivée seconde ne nous ont pas incités à le faire. Au lieu de choisir la condition naturelle $s^{''}(j_{x_{min}}) = s^{''}(j_{x_{max}}) = 0$, on va chercher à estimer $s^{''}(j_{x_{min}})$ et $s^{''}(j_{x_{max}})$ pour un niveau de grille $R \neq 0$. La discussion se fera à une dimension, car l'extension à deux ou trois se fait simplement (voir sous-partie 2.2.5). Les tests numériques seront cependant réalisés avec le code 2D2V.

D.1 Méthodes des conditions aux bords lors de la rotation

Cette partie présente les différentes méthodes qui seront étudiées par la suite. Les deux dernières sous-parties seront simplement discutées et présentées, mais aucune implémentation numérique n'a été réalisée.

D.1.1 Schéma centré

Dans cette première méthode, on va chercher à estimer la valeur de la dérivée seconde de s aux bords en prenant un schéma centré pour calculer la dérivée seconde de f. Cependant, comme f est au bord de la grille R, les informations sur la fonction de distribution en dehors de la grille ne sont pas directement sur le niveau R. Une solution consisterait à utiliser la grille R-1 pour interpoler les informations manquantes. Cependant, pour deux (ou trois) dimensions, l'interpolation devrait être effectuée sur les coefficients de splines γ (et γ^*). Or, cela nécessiterait d'effectuer les interpolations entre les grilles simultanément ou de sauvegarder les valeurs des coefficients de splines. Ces pistes n'ont pas été explorées, car les résultats obtenus sans elles sont déjà très satisfaisants.

Afin d'estimer la dérivée de f par un schéma centré, on choisit donc :

$$\begin{aligned} s''(j_{x_{min}}) =& f''(j_{x_{min}}\text{+2}) = \\ & \frac{1}{12h^2}\,(\text{-}f(j_{x_{min}}) + 16f(j_{x_{min}}\text{+1}) \\ & \text{-}30f(j_{x_{min}}\text{+2}) + 16f(j_{x_{min}}\text{+3})\text{-}f(j_{x_{min}}\text{+4})) + O(h^4), \\ s''(j_{x_{max}}) =& f''(j_{x_{max}}\text{-2}) = \\ & \frac{1}{12h^2}\,(\text{-}f(j_{x_{max}}\text{-4}) + 16f(j_{x_{max}}\text{-3}) \\ & \text{-}30f(j_{x_{max}}\text{-2}) + 16f(j_{x_{max}}\text{-1})\text{-}f(j_{x_{max}})) + O(h^4). \end{aligned} \tag{D.1}$$

D.1.2 Dérivée seconde nulle

La seconde méthode consiste à ne pas modifier le code est à considérer que même si $f''(j_{x_{min}}) \neq 0$ et $f''(j_{x_{max}}) \neq 0$, on a :

$$s''(j_{x_{min}}) = s''(j_{x_{max}}) = 0. \tag{D.2}$$

D.1.3 Schéma décentré

Le schéma centré ne semble pas adapté à notre problème, car sur un bord de grille, on connaît l'information sur la fonction de distribution que d'un seul côté.

Nous allons donc choisir un schéma décentré, tel que l'on a :

$$\begin{aligned} s''(j_{x_{min}}) &= f''(j_{x_{min}}) = \\ &\frac{1}{h^2}\,(45f(j_{x_{min}})\text{-}154f(j_{x_{min}}\text{+}1)\text{+}214f(j_{x_{min}}\text{+}2) \\ &\text{-}156f(j_{x_{min}}\text{+}3)\text{+}61f(j_{x_{min}}\text{+}4)\text{-}10f(j_{x_{min}}\text{+}5)) + O(h^4), \\ s''(j_{x_{max}}) &= f''(j_{x_{max}}) = \\ &\frac{1}{h^2}\,(\text{-}10f(j_{x_{max}}\text{-}5)\text{+}61f(j_{x_{max}}\text{-}4)\text{-}156f(j_{x_{max}}\text{-}3) \\ &\text{+}214f(j_{x_{max}}\text{-}2)\text{-}154f(j_{x_{max}}\text{-}1)\text{+}45f(j_{x_{max}})) + O(h^4). \end{aligned} \tag{D.3}$$

C'est cette méthode qui va être conservée par la suite dans le code. On peut aussi l'écrire avec un ordre moins élevé :

$$\begin{aligned} s''(j_{x_{min}}) &= f''(j_{x_{min}}) = \\ &\frac{1}{h^2}\,(2f(j_{x_{min}})\text{-}5f(j_{x_{min}}\text{+}1)\text{+}4f(j_{x_{min}}\text{+}2)\text{-}f(j_{x_{min}}\text{+}3)) + O(h^2), \\ s''(j_{x_{max}}) &= f''(j_{x_{max}}) = \\ &\frac{1}{h^2}\,(\text{-}f(j_{x_{max}}\text{-}3)\text{+}4f(j_{x_{max}}\text{-}2)\text{-}5f(j_{x_{max}}\text{-}1)\text{+}2f(j_{x_{max}})) + O(h^2). \end{aligned} \tag{D.4}$$

D.1.4 Méthode des splines

L'idée va être comme pour les advections locales (voir sous-partie 2.2.6) d'estimer de manière itérative $s''_i = s''(j_x)$, on rappelle les équations (2.33) et (2.34) :

$$f_i = \frac{1}{6}(\eta_{i-1} + 4\eta_i + \eta_{i+1}), \tag{D.5}$$

$$s''_i = \frac{1}{h^2}(\eta_{i-1} - 2\eta_i + \eta_{i+1}). \tag{D.6}$$

En combinant ces deux équations, on retrouve :

$$s''_i = \frac{1}{h^2}(6f_i - 6\eta_i), \tag{D.7}$$

et donc on a

$$\eta_i = f_i - \frac{h^2}{6}s''_i. \tag{D.8}$$

On peut aussi écrire à partir de (D.5) :

$$\eta_i = \frac{6}{4}f_i - \frac{\eta_{i-1}}{4} - \frac{\eta_{i+1}}{4}. \tag{D.9}$$

En injectant cette équation dans (D.6) et en remplaçant η_{i-1} et η_{i+1} par (D.8), on retrouve :

$$s_i^{"} = \frac{1}{h^2}\left(-3f_i + \frac{3}{2}(f_{i-1} + f_{i+1}) - \frac{h^2}{4}(s_{i-1}^{"} + s_{i+1}^{"})\right). \tag{D.10}$$

On peut donc désormais reprendre d'une manière similaire à la méthode sur la dérivée première pour les splines locaux (voir la sous-partie 2.2.6 et l'article [104]). On réinjecte donc cette équation dans elle-même pour trouver :

$$\begin{aligned} s_i^{"} =& \frac{1}{h^2}\left(-3f_i + \frac{3}{2}(f_{i-1} + f_{i+1})\right. \\ & \left. -\frac{1}{4}\left(-3f_{i-1} + \frac{3}{2}(f_{i-2} + f_i) - 3f_{i+1} + \frac{3}{2}(f_i + f_{i+2})\right) + \frac{h^2}{16}(s_{i-2}^{"} + 2s_i^{"} + s_{i+2}^{"})\right), \end{aligned} \tag{D.11}$$

on a ainsi :

$$s_i^{"} = \frac{1}{h^2}\left(-\frac{30}{7}f_i + \frac{18}{7}(f_{i-1} + f_{i+1}) - \frac{3}{7}(f_{i-2} + f_{i+2})) + \frac{h^2}{14}(s_{i-2}^{"} + s_{i+2}^{"})\right). \tag{D.12}$$

On peut répéter cela itérativement, de la même manière que pour la dérivée première pour les splines locaux. Cependant, à la différence des splines locaux, nous n'avons l'information sur la fonction de distribution que d'un seul côté.

Une idée serait de calculer $s''(j_{x_{min}} + 1)$ en connaissant l'information sur $f(j_{x_{min}})$ grâce à la cellule fantôme et donc utiliser l'équation (D.10). Il faudrait choisir une formule pour exprimer $s_{i-1}^{"}$ et $s_{i+1}^{"}$, on peut par exemple citer les trois exemples précédents. En augmentant le nombre de cellules fantômes, on pourrait ainsi aller plus loin dans l'itération de $s_i^{"}$.

L'augmentation du nombre de cellules fantômes aurait cependant un coût en temps de calcul et en mémoire.

D.1.5 Méthode de la quasi-interpolation

De la même manière que pour les splines locaux, on pourrait utiliser la formule de quasi-interpolation donnée dans l'équation (2.108) afin d'estimer directement les coefficient splines $\eta_{i_{min}-1\pm j}$ et $\eta_{i_{max}+1\pm j}$. On rappelle cette équation :

$$\eta_k = c_k = -\frac{1}{6}f_{k-1} + \frac{4}{3}f_k - \frac{1}{6}f_{k+1}. \tag{D.13}$$

Ensuite, une résolution similaire à celle présentée dans la sous-partie 2.4.4 pourrait être effectuée afin d'obtenir directement une estimation de $\eta_{i_{min}-1}$ et $\eta_{i_{max}+1}$.

De la même manière que pour la méthode des splines présentée précédemment, cette dernière nécessiterait d'augmenter le nombre de cellules fantômes, ce qui engendrerait un coût en termes de temps de calcul et de mémoire.

Ces deux dernières méthodes n'ont pas été implémentées dans le code, car la méthode du schéma décentré avec une seule cellule fantôme s'est avérée suffisante. Il s'agit toutefois d'une amélioration possible du code qui pourrait être apportée dans le futur si besoin.

D.2 Comparaison numérique des méthodes d'estimation de la dérivée seconde

Pour estimer l'erreur commise avec la méthode, on utilise les splines cubiques 2V pour calculer l'advection d'une distribution Gaussienne isotrope que l'on va déplacer de $\Delta p_x = \Delta p_y = 0,1mc$ dans la direction positive de p_x et p_y. Cette distribution, qui a été advectée et interpolée, sera ensuite comparée au résultat théorique de la Gaussienne déplacée. On choisit $p_{x_{min/max}} = p_{y_{min/max}} = \mp 20mc$.

On va étudier cela avec deux cas, la méthode classique de VLEM_EG où les grilles vont s'adapter au problème (à gauche de la figure D.1) et un cas où on va fixer une seule grille raffinée sur un quart de la Gaussienne (à droite de la figure D.1).

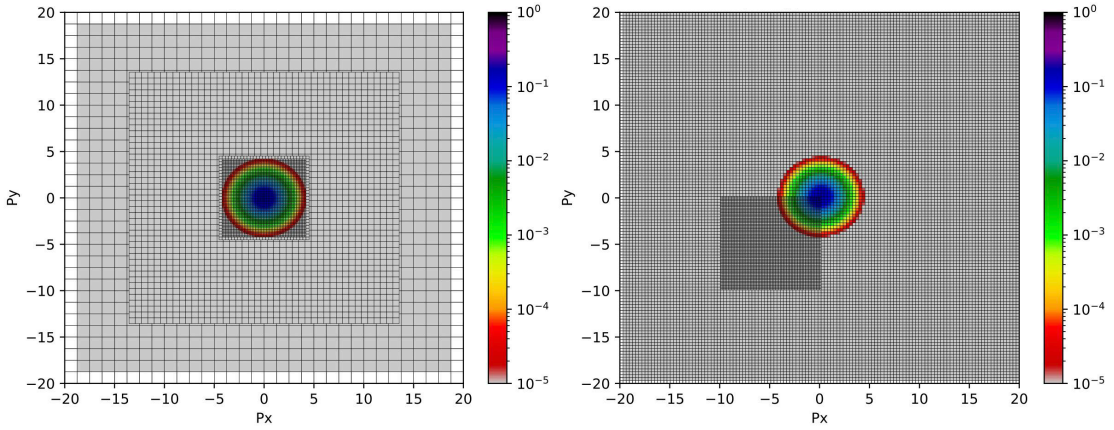

FIGURE D.1 – À droite : Représentation de la fonction de distribution Gaussienne en vitesse avec une grille adaptative de paramètre $\epsilon = 10^{-7}$ et $R^{max} = 3$ pour le cas qui sera étudié dans la sous-partie D.2.1. À gauche : Représentation de la fonction de distribution Gaussienne en vitesse avec une grille aux bords fixes qui sera étudiée dans la sous-partie D.2.2.

D.2.1 Grille évolutive

Dans un premier temps, on va augmenter le nombre de niveaux de raffinement de $R^{max} = 0$ à $R^{max} = 3$ avec un nombre de points pour la grille $R = 0$ de

$N_{p_x}(R=0) = N_{p_y}(R=0) = 32$ pour deux valeurs de ϵ ($\epsilon = 10^{-7}$ et $\epsilon = 10^{-5}$). Une représentation de la fonction de distribution étudiée est présentée à gauche de la figure D.1. On va étudier cela avec les trois conditions aux bords présentées dans les sous-parties D.1.2, D.1.1 et D.1.3 qui porteront respectivement le nom de Test 1, Test 2 et Test 3. On va rajouter un quatrième test en ajoutant une cellule fantôme supplémentaire au cas Test 3, on appellera cela Test 4.

Les résultats pour $\epsilon = 10^{-7}$ sont présentés à gauche de la figure D.2. On observe que les quatre conditions aux bords fournissent la même erreur L^∞ (l'erreur maximale) qui est aussi presque identique à celle de la grille homogène avec le code VLEM.

Lorsque $\epsilon = 10^{-5}$, l'erreur reste la même pour le code VLEM et pour les quatre conditions aux bords jusqu'à $R^{max} = 2$. Cependant, lorsque $R^{max} = 3$, les quatre conditions aux bords vont fournir une erreur différente du code VLEM et différente entre elles. En effet, à cette valeur le code VLEM fournit une erreur de $1,33 \times 10^{-6}$. Cependant comme la limite de ϵ tolère une erreur sur l'interpolation par $R = 0$ inférieure à $\frac{\epsilon}{2} = 5 \times 10^{-6}$ pour le point milieu (voir sous-partie 3.2 et figure 3.6), on peut retrouver une erreur avec cet ordre de grandeur. C'est ce qui apparaît sur la figure 3.14 pour un $\epsilon = 10^{-3}$.

L'erreur est cependant moins importante pour Test 3 et Test 4 que pour Test 1 et Test 2. Autrement dit, la condition aux bords utilisant la dérivée avec un schéma décentré semble être meilleure. Test 4 donne une erreur moins importante que Test 3 car les premiers coefficients splines qui entrent en jeu pour l'advection sont plus éloignés des bords. Cela va toutefois conduire à une augmentation plus importante du temps de calcul et de la mémoire en comparaison de Test 3.

D.2.2 Grille fixe

Nous allons maintenant étudier l'erreur qui est faite si le bord de la grille se termine dans une zone où la fonction de distribution a des valeurs plus importantes. Pour cela, on fixe les bords de la grille de façon à ce qu'ils se terminent en $p_{x_{max}}(R=1) = p_{y_{max}}(R=1) = 0mc$ et on impose aussi $p_{x_{min}}(R=1) = p_{y_{min}}(R=1) = -10mc$. On conserve $R^{max} = 1$ mais on va faire varier $N_{p_x}(R=0) \times N_{p_y}(R=0)$ de 16^2 à 128^2. Une représentation de la fonction de distribution étudiée et de la grille fixée est présentée à droite de la figure D.1. On va comparer les résultats avec le code VLEM et on trace aussi une erreur comparative avec une dépendance en $\Delta p_x^2 \Delta p_y^2$ afin de voir si la méthode conserve l'ordre de l'interpolation des B-splines cubiques [103].

On observe sur la figure D.3 que l'erreur maximale faite sur Test 3 et Test 4 est supérieure à l'erreur faite sur VLEM. Cela est dû au fait que l'erreur maximale est réalisée lors des interpolations de la grille $R = 0$. On observe cela puisque l'on retrouve avec la méthode AMR la même erreur que le code VLEM lorsque la grille $R = 0$ à la précision de la grille homogène de VLEM. Par exemple $N_{p_x}(R=0) \times N_{p_y}(R=0) \times 2^2 = 64^2$ retrouve le même résultat que lorsque VLEM a $N_{p_x} \times N_{p_y} = 32^2$.

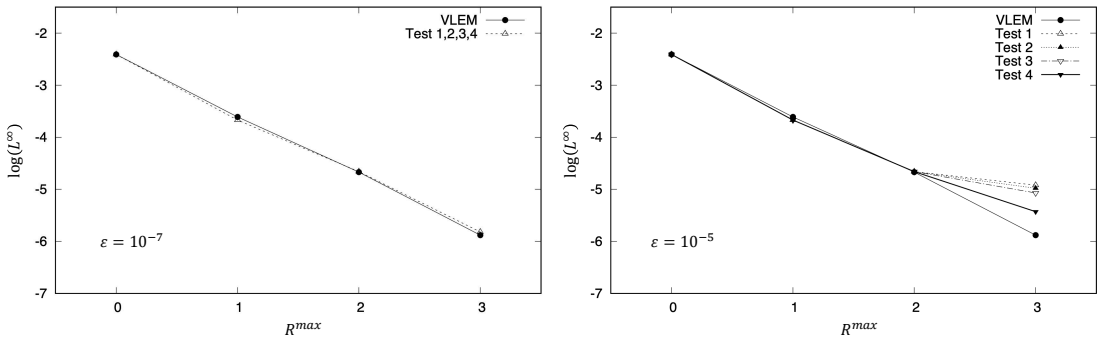

FIGURE D.2 – Tracé du logarithme de l'erreur L^∞ obtenue avec le code VLEM et VLEM_EG pour l'advection 2V d'une Gaussienne avec grille adaptative et $\epsilon = 10^{-7}$ (gauche) et $\epsilon = 10^{-5}$ (droite). Test 1 représente la condition aux bords de la sous-partie D.1.2 ($s''(j_x) = f''(j_x \pm 2)$). Test 2 représente la condition aux bords de la sous-partie D.1.1 ($s''(j_x) = 0$). Test 3 représente la condition aux bords de la sous-partie D.1.3 ($s''(j_x) = f''(j_x)$). Test 4 a les mêmes conditions aux bords que Test 3 mais avec une cellule fantôme supplémentaire. Pour le code VLEM, le nombre de points sur la grille uniforme correspond au nombre maximum de point sur R^{max}.

Les résultats des deux méthodes sont les mêmes sauf pour le premier point où $N_{p_x} N_{p_y}(R = 1) = 32^2$. Dans ce cas, la méthode avec deux cellules fantômes (Test 4) présente une erreur légèrement moins importante que pour le cas d'une seule cellule fantôme (Test 3). Ces résultats sont toutefois obtenus pour un cas où le nombre de points est vraiment faible, ce qui n'est donc pas un cas réaliste pour les applications physiques habituelles.

On observe également sur la figure D.3 que la décroissance de l'erreur lorsque l'on divise par deux Δp_x et Δp_y suit bien la même décroissance que $h^4 = \Delta p_x^2 \Delta p_y^2$. On peut donc en déduire que l'ordre de la méthode de rotation est bien conservé dans la méthode AMR pour la rotation par spline cubique.

Dans le code, ce sont les conditions aux bords du Test 3, présentées dans la sous-partie D.1.3 qui ont été gardées car les cellules fantômes représentent un coût en mémoire important, surtout pour le cas 3V. On peut également noter qu'aucune différence dans les résultats n'a été observée entre le schéma décentré d'ordre deux et quatre. Celui de quatre a été gardé dans le code même s'il présente une erreur L^∞ théorique plus importante.

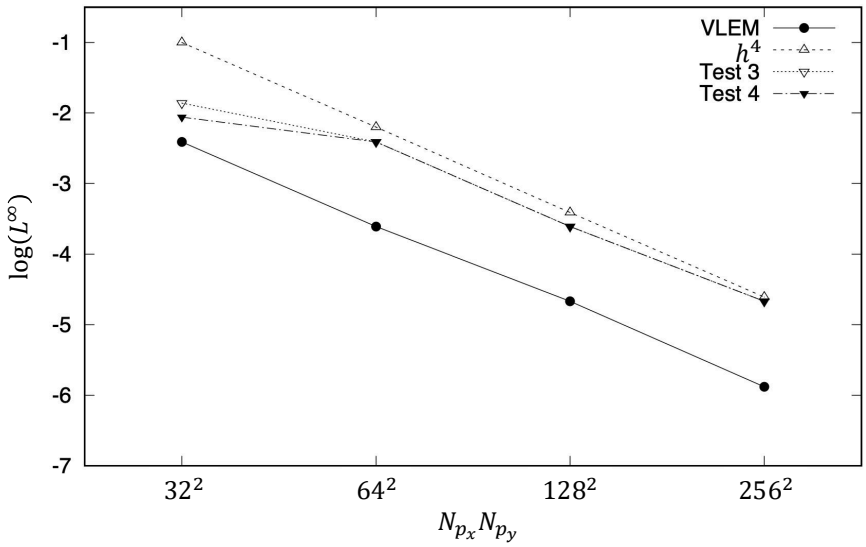

FIGURE D.3 – Tracé du logarithme de l'erreur L^∞ obtenue avec le code VLEM et VLEM_EG pour l'advection 2V d'une Gaussienne avec un seul niveau de raffinement et une grille dont les bords sont limités à un quart de la Gaussienne. Test 3 représente la condition aux bords de la sous-partie D.1.3 ($s''(j_x) = f''(j_x)$). Test 4 a les mêmes conditions aux bords que Test 3 mais avec une cellule fantôme supplémentaire. Pour le code VLEM_EG, $N_{p_x} \times N_{p_y}$ correspond au nombre maximum de points que pourrait avoir la grille de niveau $R^{max} = 1$ c'est-à-dire $N_{p_x}(R = 0) \times N_{p_y}(R = 0) \times 2^2$

Annexe E

Valeur du paramètre C dans le cas 3V

Comme pour le cas 2V, on va regarder les zones d'intérêts pour les grilles rectangles à 3V. On va donc avoir l'intersection entre la grille R et R-1 sur une face, une arête et un coin. Si on écrit initialement les coefficients de poids pour les 3 zones d'intersections, on trouve :

n-2	n-1	n	n+1	n+2		
1	1	1	(1)	1	2M	2L
1	1	1	[1]	(1)	2M+1	
1	1	1	[1]	(1)	2M	2L+1
1	1	1	∗1∗	[1]	2M+1	

Table E.1

n-2	n-1	n	n+1	n+2		
1	(1)	1	(1)	1	m+2	
(1)	[1]	(1)	[1]	(1)	m+1	
1	1	1	(1)	1	m	2L
1	1	1	[1]	(1)	m-1	
1	1	1	(1)	1	m-2	
(1)	[1]	(1)	[1]	(1)	m+2	
[1]	∗1∗	[1]	∗1∗	[1]	m+1	
1	1	1	[1]	(1)	m	2L+1
1	1	1	∗1∗	[1]	m-1	
1	1	1	[1]	(1)	m-2	

Table E.2

n-2	n-1	n	n+1	n+2		
1	(1)	1	(1)	1	m+2	
(1)	[1]	(1)	[1]	(1)	m+1	
1	(1)	1	(1)	1	m	l+2
(1)	[1]	(1)	[1]	(1)	m-1	
1	(1)	1	(1)	1	m-2	
(1)	[1]	(1)	[1]	(1)	m+2	
[1]	∗1∗	[1]	∗1∗	[1]	m+1	
(1)	[1]	(1)	[1]	(1)	m	l+1
[1]	∗1∗	[1]	∗1∗	[1]	m-1	
(1)	[1]	(1)	[1]	(1)	m-2	
1	(1)	1	(1)	1	m+2	
(1)	[1]	(1)	[1]	(1)	m+1	
1	1	1	(1)	1	m	l
1	1	1	[1]	(1)	m-1	
1	1	1	(1)	1	m-2	
(1)	[1]	(1)	[1]	(1)	m+2	
[1]	∗1∗	[1]	∗1∗	[1]	m+1	
1	1	1	[1]	(1)	m	l-1
1	1	1	∗1∗	[1]	m-1	
1	1	1	[1]	(1)	m-2	
1	(1)	1	(1)	1	m+2	
(1)	[1]	(1)	[1]	(1)	m+1	
1	1	1	(1)	1	m	l-2
1	1	1	[1]	(1)	m-1	
1	1	1	(1)	1	m-2	

TABLE E.3

Comme pour le cas 2V, on a dans le tableau E.1, $n = j_{\alpha_{max}}(R)$ et $n+1 = j_{\alpha_{max}}(R)+1$ où α va représenter la direction orthogonale au coté. On a aussi $2M$ et $2L$ qui représentent les valeurs paires et $2M$+1 et $2L$+1 les valeurs impaires de $j_\beta(R)$ et $j_\gamma(R)$ avec β et γ les deux autres directions du système 3V.

Les calculs que l'on va effectuer pour une face seront valides tant que l'on a $j_{\beta_{min}}(R) + 2 < 2M < j_{\beta_{max}}(R) - 2$ et $j_{\gamma_{min}}(R) + 2 < 2L < j_{\gamma_{max}}(R) - 2$ et ceux que l'on va effectuer pour une arête sont valides tant que l'on a $j_{\gamma_{min}}(R) + 2 < 2L < j_{\gamma_{max}}(R) - 2$. En remplaçant cette fois les valeurs des points interpolés en prenant aussi en compte la $3^{\text{ème}}$ direction,

on doit utiliser les répartitions suivantes :

$$\left|\begin{array}{ccccccc} / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & *1* & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \end{array}\right| = \frac{1}{4096} \left|\begin{array}{ccccccc} -1 & / & 9 & / & 9 & / & -1 \\ / & / & / & / & / & / & / \\ 9 & / & -81 & / & -81 & / & 9 \\ / & / & / & / & / & / & / \\ 9 & / & -81 & / & -81 & / & 9 \\ / & / & / & / & / & / & / \\ -1 & / & 9 & / & 9 & / & -1 \end{array}\right| \left|\begin{array}{ccccccc} / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \end{array}\right| \left|\begin{array}{ccccccc} 9 & / & -81 & / & -81 & / & 9 \\ / & / & / & / & / & / & / \\ -81 & / & 729 & / & 729 & / & -81 \\ / & / & / & / & / & / & / \\ -81 & / & 729 & / & 729 & / & -81 \\ / & / & / & / & / & / & / \\ 9 & / & -81 & / & -81 & / & 9 \end{array}\right| \left|\begin{array}{ccccccc} / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \end{array}\right| \left|\begin{array}{ccccccc} 9 & / & -81 & / & -81 & / & 9 \\ / & / & / & / & / & / & / \\ -81 & / & 729 & / & 729 & / & -81 \\ / & / & / & / & / & / & / \\ -81 & / & 729 & / & 729 & / & -81 \\ / & / & / & / & / & / & / \\ 9 & / & -81 & / & -81 & / & 9 \end{array}\right| \left|\begin{array}{ccccccc} / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \end{array}\right| \left|\begin{array}{ccccccc} -1 & / & 9 & / & 9 & / & -1 \\ / & / & / & / & / & / & / \\ 9 & / & -81 & / & -81 & / & 9 \\ / & / & / & / & / & / & / \\ 9 & / & -81 & / & -81 & / & 9 \\ / & / & / & / & / & / & / \\ -1 & / & 9 & / & 9 & / & -1 \end{array}\right|, \tag{E.1}$$

$$\begin{vmatrix} / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & [1] & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \end{vmatrix} = \frac{1}{256} \begin{vmatrix} 1 & / & -9 & / & -9 & / & 1 \\ / & / & / & / & / & / & / \\ -9 & / & 81 & / & 81 & / & -9 \\ / & / & / & / & / & / & / \\ -9 & / & 81 & / & 81 & / & -9 \\ / & / & / & / & / & / & / \\ 1 & / & -9 & / & -9 & / & 1 \end{vmatrix}, \tag{E.2}$$

$$\begin{vmatrix} / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & (1) & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \end{vmatrix} = \frac{1}{16} \begin{vmatrix} / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ -1 & / & 9 & / & 9 & / & -1 \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \\ / & / & / & / & / & / & / \end{vmatrix}. \tag{E.3}$$

En remplaçant et en utilisant le fait que l'on a $\Delta p(x)(R)\Delta p(y)(R)\Delta p(z)(R) = \boldsymbol{\Delta p}(R) = \boldsymbol{\Delta p}(R+1) \times 8$ dans l'équation (3.17) que l'on développe à 3V :

$$\begin{aligned} I(x,y) = & \sum_{R_n=0}^{R_n^{max}-1} \left[\sum_{j_x,j_y,j_z \in S_{R_n} \setminus S_{R_{n+1}}} F_{R_n}(j_x,j_y,j_z) C(j_x,j_y,j_z) \boldsymbol{\Delta p}(R) \right] \\ & + \sum_{j_x,j_y,j_z \in S_{R_n^{max}}} F_{R_n^{max}}(j_x,j_y,j_z) C(j_x,j_y,j_z) \boldsymbol{\Delta p}(R_n^{max}). \end{aligned} \tag{E.4}$$

on trouve pour le cas 3V les coefficients de C suivants :

n-2	n-1	n	n+1	n+2		
12	16	48	/	16,5	2M	2L
16	16	16	/	/	2M+1	
16	16	16	/	/	2M	2L+1
16	16	16	/	/	2M+1	

TABLE E.4 – Valeur de $16 * C(j_x, j_y, j_z)$ sur une face pour le cas 3V. Les points qui ne sont pas représentés et qui ne sont pas affectés par les autres interfaces entre les grilles ont une valeur $C(j_x, j_y, j_z) = 1$.

n-2	n-1	n	n+1	n+2		
264,25	/	262	/	255,75	m+2	
/	/	/	/	/	m+1	
720	256	1152	/	262	m	2L
256	256	256	/	/	m-1	
126	256	720	/	264,25	m-2	
/	/	/	/	/	m+2	
/	/	/	/	/	m+1	
256	256	256	/	/	m	2L+1
256	256	256	/	/	m-1	
256	256	256	/	/	m-2	

TABLE E.5 – Valeur de $256 * C(j_x, j_y, j_z)$ sur une arête pour le cas 3V. Les points qui ne sont pas représentés et qui ne sont pas affectés par les autres interfaces entre les grilles ont une valeur $C(j_x, j_y, j_z) = 1$.

n-2	n-1	n	n+1	n+2		
4091,875	/	4093	/	4096,125	m+2	
/	/	/	/	/	m+1	
4888	/	4672	/	4093	m	l+2
/	/	/	/	/	m-1	
5185	/	4888	/	4091,875	m-2	
/	/	/	/	/	m+2	
/	/	/	/	/	m+1	
/	/	/	/	/	m	l+1
/	/	/	/	/	m-1	
/	/	/	/	/	m-2	
4195	/	4168	/	4093	m+2	
/	/	/	/	/	m+1	
17856	4096	23040	/	4168	m	l
4096	4096	4096	/	/	m-1	
10728	4096	17856	/	4195	m-2	
/	/	/	/	/	m+2	
/	/	/	/	/	m+1	
4096	4096	4096	/	/	m	l-1
4096	4096	4096	/	/	m-1	
4096	4096	4096	/	/	m-2	
4232,125	/	4195	/	4091,875	m+2	
/	/	/	/	/	m+1	
10728	4096	17856	/	4195	m	l-2
4096	4096	4096	/	/	m-1	
-927	4096	10728	/	4232,125	m-2	

TABLE E.6 – Valeur de $4096 * C(j_x, j_y, j_z)$ sur un coin pour le cas 3V. Les points qui ne sont pas représentés et qui ne sont pas affectés par les autres interfaces entre les grilles ont une valeur $C(j_x, j_y, j_z) = 1$.

Bibliographie

[1] I. Langmuir, “Oscillations in Ionized Gases,” *Proc. Natl. Acad. Sci. U.S.A.*, vol. 14, no. 8, pp. 627–637, 1928, doi :10.1073/pnas.14.8.627.

[2] R. N. Franklin, N. St J. Braithwaite, “80 Years of Plasma,” *Plasma Sources Science and Technology*, vol. 18, no. 1, p. 010201, 2009, doi :10.1088/0963-0252/18/1/010201.

[3] L. Tonks, “The Birth of "Plasma",” *Am. J. Phys.*, vol. 35, no. 9, pp. 857–858, 1967, doi :10.1119/1.1974266.

[4] R. Paul Drake, *High Energy-Density-Physics : Fundamentals, Inertial Fusion, and Experimental Astrophysics.* Springer, 2006.

[5] D. Del Sarto, *Contribution to the study of energy conversion mechanisms in collisionless plasmas.* HDR thesis, Université de Lorraine, 2025.

[6] K. Nishikawa, M. Wakatani, *Plasma Physics : Basic Theory with Fusion Applications.* Springer-Verlag, 1990.

[7] Yu. L. Klimontovich, *The Statistical Theory of non-equilibrium processes in a plasma.* Pergamon press, 1967.

[8] ITER organization :https://www.iter.org/fr/facts-figures, “Iter en chiffres,” 2025.

[9] A. Gabrielli, M. Joyce, J. Morand, “Finite-N corrections to Vlasov dynamics and the range of pair interactions,” *Phys. Rev. E.*, vol. 90, no. 6, p. 062910, 2016, doi :10.1103/PhysRevE.90.062910.

[10] P. Bertrand, D. Del Sarto , A. Ghizzo, *The Vlasov Equation I : History and General Properties, Chapter 5.4.* ISTE Ltd, 2019.

[11] N. A. Krall, A. W. Trivelpiece, *Principles of Plasma Physics.* San Francisco Pr, 1986.

[12] A. J. Brizard, T. S. Hahm, "Foundations of nonlinear gyrokinetic theory," *REVIEWS OF MODERN PHYSICS*, vol. 79, no. 4, p. 421, 2007, doi :10.1103/RevModPhys.79.421.

[13] F. Jenko, W. Dorland, M. Kotschenreuther, B. N. Rogers, "Electron temperature gradient driven turbulence," *Phys. Plasmas*, vol. 7, no. 5, pp. 1904–1910, 2000, doi :10.1063/1.874014.

[14] V. Grandgirard, J. Abiteboul, J. Bigot, T. Cartier-Michaud, N. Crouseilles, G. Dif-Pradalier, Ch. Ehrlacher, D. Esteve, X. Garbet, Ph. Ghendrih, G. Latu, M. Mehrenberger, C. Norscini, Ch. Passeron, F. Rozar, Y. Sarazin, E. Sonnendrücker, A. Strugarek, D. Zarzoso, "A 5D gyrokinetic full-f global semi-Lagrangian code for flux-driven ion turbulence simulations," *Computer Physics Communications*, vol. 207, no. 1, pp. 35–68, 2016, doi :10.1016/j.cpc.2016.05.007.

[15] Y Sarazin, V Grandgirard, E Fleurence, X Garbet, Ph Ghendrih, P Bertrand, G Depret, "Kinetic features of interchange turbulence," *Plasma Phys. Control. Fusion*, vol. 47, no. 10, p. 1817, 2005, doi :10.1088/0741-3335/47/10/013.

[16] J. M. Dawson, "Particle simulation of plasmas," *Reviews of Modern Physics*, vol. 55, no. 2, pp. 403–447, 1983, doi :10.1103/RevModPhys.55.403.

[17] J. Derouillat, A. Beck, F. Pérez, T. Vinci, M. Chiaramello, A. Grassi, M. Flé, G. Bouchard, I. Plotnikov, N. Aunai, J. Dargent, C. Riconda, M. Grech, "Smilei : a collaborative, open-source, multi-purpose particle-in-cell code for plasma simulation," *Computer Physics Communications*, vol. 222, no. 1, pp. 351–373, 2018, doi :10.1016/j.cpc.2017.09.024.

[18] F. Filbet, E. Sonnendrücker, "Comparison of eulerian vlasov solvers," *Computer Physics Communications*, vol. 150, no. 3, pp. 247–266, 2003, doi :10.1016/S0010-4655(02)00694-X.

[19] J. A. Rossmanith, D. C. Seal, "A positivity-preserving high-order semi-lagrangian discontinuous Galerkin scheme for the Vlasov–Poisson equations," *J. Comput. Phys.*, vol. 230, no. 16, pp. 6203–6232, 2011, doi :10.1016/j.jcp.2011.04.018.

[20] S. I. Zaki, L. R. T. Gardner, T. J. M. Boyd, "A finite element code for the simulation of one-dimensional vlasov plasmas. I. Theory," *Journal of Computational Physics*, vol. 79, no. 1, pp. 184–199, 1988, doi :10.1016/0021-9991(88)90010-1.

[21] A. J. Klimas, "A method for overcoming the velocity space filamentation problem in collisionless plasma model solutions," *Journal of Computational Physics*, vol. 68, no. 1, pp. 202–226, 1987, doi :10.1016/0021-9991(87)90052-0.

[22] G.L. Delzanno, "Multi-dimensional, fully-implicit, spectral method for the Vlasov-Maxwell equations with exact conservation laws in discrete form," *Europhysics News*, vol. 301, no. 15, pp. 338–356, 2015, doi :10.1016/j.jcp.2015.07.028.

[23] E. Fijalkow, "A numerical solution to the Vlasov equation," *J. Comput. Phys.*, vol. 116, no. 2, pp. 319–328, 1999, doi :10.1016/S0010-4655(98)00146-5.

[24] F. Filbet, E. Sonnendrücker, P. Bertrand, "Conservative numerical schemes for the vlasov equation," *Journal of Computational Physics*, vol. 172, no. 1, pp. 166–187, 2001, doi :10.1006/jcph.2001.6818.

[25] H. Liu , X. Cai , Y. Cao , G. Lapenta , "An efficient energy conserving semi-Lagrangian kinetic scheme for the Vlasov-Ampere system," *J. Comput. Phys.*, vol. 492, no. 1, p. 112412, 2023, doi :10.1016/j.jcp.2023.112412.

[26] N. Schild, M. Raath, S. Eibl, K. Hallatschek, K. Kormann, "A performance portable implementation of the semi-Lagrangian algorithm in six dimensions," *Computer Physics Communications*, vol. 295, no. 1, p. 108973, 2024, doi :10.1016/j.cpc.2023.108973.

[27] O. Coulaud, E. Sonnendrucker, E. Dillon, P. Bertrand, A. Ghizzo, "Parallelisation of Semi-Lagrangian Vlasov Codes," *INRIA*, vol. 61, no. 3, pp. 435–448, 1998, doi :10.1017/S0022377899007527.

[28] M. Sarrat, A. Ghizzo, D. Del Sarto, L. Serrat, "Parallel implementation of a relativistic semi-lagrangian Vlasov-Maxwell solver," *The European Physical Journal D*, vol. 71, no. 1, p. 271, 2017, doi :10.1140/epjd/e2017-80188-4.

[29] A. Ghizzo, D. Del Sarto, "Low- and high-frequency nature of oblique filamentation modes. II. Vlasov-Maxwell simulations of collisionless heating process," *Physics of Plasma*, vol. 27, no. 7, p. 072104, 2020, doi :10.1063/5.0003698.

[30] E. Sonnendrucker, J. Roche, P. Bertrand, A. Ghizzo, "The Semi-Lagrangian Method for the Numerical Resolution of the Vlasov Equation," *Journal of Computational Physics*, vol. 149, no. 2, pp. 201–220, 1999, doi :10.1006/jcph.1998.6148.

[31] R. A. Fonseca, L. O. Silva, F. S. Tsung, V. K. Decyk, W. Lu, C. Ren, W. B. Mori, S. Deng, S. Lee, T. Katsouleas, J. C. Adam, "OSIRIS : A Three-Dimensional, Fully Relativistic Particle in Cell Code for Modeling Plasma Based Accelerators," vol. 2331, pp. 342–351, 2002, doi :10.1007/3-540-47789-6_36.

[32] T. D. Arber, K. Bennett, C. S. Brady, A. Lawrence-Douglas, M. G. Ramsay, N. J. Sircombe, P. Gillies, R. G. Evans, H. Schmitz, A. R. Bell, C. P. Ridgers, "Contemporary particle-in-cell approach to laser-plasma modelling," *Plasma Phys. Control. Fusion*, vol. 57, no. 11, p. 113001, 2015, doi :10.1088/0741-3335/57/11/113001.

[33] M. Shalaby, A. E. Broderick, P. Chang, C. Pfrommer, A. Lamberts, E. Puchwein, "SHARP : A Spatially Higher-order, Relativistic Particle-in-cell Code," *The Astrophysical Journal*, vol. 841, no. 1, p. 52, 2017, doi :10.3847/1538-4357/aa6d13.

[34] R. Bird, N. Tan, S. V. Luedtke , S. L. Harrell , M. Taufer , B. Albright , "VPIC 2.0 : Next Generation Particle-in-Cell Simulations," *IEEE Transactions on parallel and distributed systems*, vol. 33, no. 4, pp. 952–963, 2022, doi :10.1109/TPDS.2021.3084795.

[35] L. Fedeli, A. Huebl, F. Boillod-Cerneux,T. Clark, K. Gott, C Hillairet, S. Jaure, A. Leblanc, R. Lehe, A. C. Myers, C. Piechurski, M. Sato, N. Zaïm, W. Zhang, J.L. Vay, H. Vincenti , "Pushing the Frontier in the Design of Laser-Based Electron Accelerators with Groundbreaking Mesh-Refined Particle-In-Cell Simulations on Exascale-Class Supercomputers," 2022, doi :10.1109/SC41404.2022.00008.

[36] N. Aunai, R. Smets, A. Ciardi, P. Deegan, A. Jeandet, T. Payet, N. Guyot, L. Darrieumerlou , "PHARE : Parallel hybrid particle-in-cell code with patch-based adaptive mesh refinement," *Computer Physics Communications*, vol. 295, no. 1, p. 108966, 2024, doi :10.1016/j.cpc.2023.108966.

[37] F. Bacchini, "RelSIM : A Relativistic Semi-implicit Method for Particle-in-Cell Simulations," *The astrophysical Journal*, vol. 268, no. 2, p. 60, 2023, doi :10.3847/1538-4365/acefba.

[38] F. Bacchini, J. Amaya, G. Lapenta, "The relativistic implicit particle-in-cell method," *Journal of Physics : Conference Series*, vol. 1225, no. 1, p. 012011, 2019, doi :10.1088/1742-6596/1225/1/012011.

[39] A. Ghizzo, D. Del Sarto, "J. r. angus, w. farmer, a. friedman, d. ghosh,d. grote, d. larson, a. link," *Journal of Computational Physics*, vol. 491, no. 1, p. 112383, 2023, doi :10.1016/j.jcp.2023.112383.

[40] G. Lapenta, "Exactly energy conserving semi-implicit particle in cell formulation," *J. Comput. Phys.*, vol. 334, no. 1, p. 349, 2017, doi :10.1016/J.JCP.2017.01.002.

[41] G. Lehmann, "Efficient Semi-Lagrangian Vlasov-Maxwell Simulations of High Order Harmonic Generation from Relativistic Laser-Plasma Interactions," *Commun. Comput. Phys.*, vol. 20, no. 3, pp. 583–602, 2016, doi :10.4208/cicp.OA-2015-0019.

[42] C.Z. Cheng, G. Knorr, "The Integration of the Vlasov Equation in Configuration Space," *Journal of Computational Physics*, vol. 22, no. 3, pp. 330–351, 1976, doi :10.1016/0021-9991(76)90053-X.

[43] T. Yakamura, T. Yabe , "Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space," *Computer Physics Communications*, vol. 120, no. 2, pp. 122–154, 1999, doi :10.1016/S0010-4655(99)00247-7.

[44] N. Crouseilles, L. Einkemmer, E. Faou , "Hamiltonian splitting for the Vlasov-Maxwell equations," *J. Comput. Phys.*, vol. 283, no. 15, pp. 224–240, 2015, doi :10.1016/j.jcp.2014.11.029.

[45] N. Besse, G. Latu, A. Ghizzo, E. Sonnendrucker, P. Bertrand, "A wavelet-MRA-based adaptive semi-Lagrangian method for the relativistic Vlasov-Maxwell system," *Journal of Computational Physics*, vol. 227, no. 16, pp. 7889–7916, 2008, doi :10.1016/j.jcp.2008.04.031.

[46] F. Huot, A. Ghizzo, P. Bertrand, E. Sonnendrucker, O. Coulaud, "Study of Propagation of Ultraintense Electromagnetic Wave Through Plasma Using Semi-Lagrangian Vlasov Codes," *IEEE Transactions on Plasma Science*, vol. 28, no. 4, pp. 1170–1184, 2000, doi :10.1109/27.893305.

[47] M. Antoine , A. Ghizzo, E. Deriaz, D. Del Sarto, "Embedded grid refinement for semi-lagrangian parallelized relativistic vlasov-maxwell solver," *Phys. Plasmas*, vol. 32, no. 4, p. 043905, 2025, doi :10.1063/5.0252704.

[48] A. Ghizzo, D. Del Sarto, H. Betar, "Collisionless heating driven by vlasov filamentation in a counterstreaming beams configuration," *Phys. Rev. Lett.*, vol. 131, no. 3, p. 035101, 2023, doi :10.1103/PhysRevLett.131.035101.

[49] H. J. Ziegler, H. Wiechen, "Mixing and relaxation in ideal incompressible fluids," *Phys. Scr.*, vol. 1998, no. T74, p. 50, 1998, doi :10.1088/0031-8949/1998/T74/009.

[50] H. Wiechen, "New aspects of plasma sheet dynamics - MHD and kinetic theory," *Annales Geophysicae*, vol. 17, no. 5, pp. 595–603, 1999, doi :10.1007/s00585-999-0595-2.

[51] S. Popinet, "Gerris : a tree-based adaptive solver for the incompressible Euler equations in complex geometries," *Journal of computational physics*, vol. 190, no. 2, pp. 572–600, 2003, doi :10.1016/S0021-9991(03)00298-5.

[52] D. Del Sarto, E. Deriaz, "A multigrid AMR algorithm for the study of magnetic reconnection," *Journal of computational physics*, vol. 351, no. 1, pp. 511–533, 2017, doi :10.1016/j.jcp.2017.08.046.

[53] J. A. F. Hittinger, J. W. Banks, "Block-Structured Adaptive Mesh Refinement Algorithms for Vlasov Simulation," *Journal of computational physics*, vol. 241, no. 1, pp. 118–140, 2013, doi :10.1016/j.jcp.2013.01.030.

[54] L. Kotipalo, M. Battarbee, Y. Pfau-Kempf, M. Palmroth, "Physics-motivated Cell-octree Adaptive Mesh Refinement in the Vlasiator 5.3 Global Hybrid-Vlasov Code," *Geosci. Model Dev.*, vol. 17, no. 1, pp. 6401–6413, 2024, doi :10.5194/egusphere-2024-301.

[55] R. Teyssier, "Cosmological Hydrodynamics with Adaptive Mesh Refinement : a new high resolution code called RAMSES," *Astronomy and Astophysics*, vol. 385, no. 1, pp. 337–364, 2002, doi :10.1051/0004-6361:20011817.

[56] E. Deriaz, S. Peirani, "Six-dimensional adaptive simulation of the vlasov equations using a hierarchical basis," *Multiscale Modeling and Simulation :A SIAM Interdisciplinary Journal*, vol. 16, no. 2, pp. 583–614, 2018, doi :10.1137/16M1108649.

[57] É.Madaule, *Schemas numériques adaptatifs pour les equations de Vlasov-Poisson*. PhD thesis, Université de Lorraine, 2016.

[58] B. Svedung Wettervik, E. Siminos, T. Fülöp, "Relativistic Vlasov-Maxwell modelling using finite volumes and adaptive mesh refinement," *Eur. Phys. J. D*, vol. 71, no. 1, p. 157, 2017, doi :10.1140/epjd/e2017-80102-2.

[59] T. Pollinger , J. Rentrop , D. Pfluuger , K. Kormann , "A stable and mass-conserving sparse grid combination technique with biorthogonal hierarchical basis functions for kinetic simulations," *J. Comput. Phys.*, vol. 491, no. 15, p. 112338, 2023, doi :10.1016/j.jcp.2023.112338.

[60] U. Ganse, T. Koskela, M. Battarbee, Y. Pfau-Kempf, K. Papadakis, M. Alho, M. Bussov, G. Cozzani, M. Dubart, H. George, E. Gordeev, M. Grandin, K. Horaites, J. Suni, V. Tarvus, F Tesema Kebede, L. Turc, H. Zhou, M. Palmroth, "Enabling technology for global 3D + 3V hybrid-Vlasov simulations of near-Earth space," *Phys. Plasmas*, vol. 30, no. 4, p. 042902, 2023, doi :10.1063/5.0134387.

[61] D. A. Ibanez, E. Love, T. E. Voth, J. R. Overfelt, N. V. Roberts, G. A. Hansen, "Tetrahedral mesh adaptation for lagrangian shock hydrodynamics," *Computers & Mathematics with Applications*, vol. 78, no. 2, pp. 402–416, 2019, doi :10.1016/j.camwa.2018.06.013.

[62] S. Del Pino, I. Marmajou, "Triangular metric-based mesh adaptation for compressible multi-material flows in semi-lagrangian coordinates," *Journal of Computational Physics*, vol. 478, no. 1, p. 111975, 2023, doi :10.1016/j.jcp.2023.111975.

[63] E.S. Weibel, "Spontaneously growing transverse waves in a plasma due to an anisotropic velocity distribution," *Phys. Rev. Lett.*, vol. 2, no. 3, pp. 83–84, 1958, doi :10.1103/PhysRevLett.2.83.

[64] C. F. Kennel, H. E. Petschek, "Limit on Stably Trapped Particle Fluxes," *Journal of Geophysical Research*, vol. 71, no. 1, p. 1, 1966, doi :10.1029/JZ071i001p00001.

[65] U. Schaefer-Rolffs, R. C. Tautz, "The relativistic kinetic Weibel instability : Comparison of different distribution functions," *Phys. Plasmas*, vol. 15, no. 1, p. 062105, 2008, doi :10.1063/1.2932106.

[66] A. Bret, L. Gremillet, D. Bénisti, "Exact relativistic kinetic theory of the full unstable spectrum of an electron-beam–plasma system with Maxwell-Jüttner distribution functions," *Physical Review E*, vol. 81, no. 3, p. 036402, 2010, doi :10.1103/PhysRevE.81.036402.

[67] D. Del Sarto, A. Ghizzo, M. Sarrat, "Nonlinear coupling of electromagnetic and electrostatic modes via density and pressure fluctuations : the case of Weibel instabilities," *American Institute of Physics*, vol. 31, no. 7, p. 072113, 2024, doi :10.1063/5.0207974.

[68] B.D. Fried, "Mechanism for instability of transverse plasma waves," *Phys. Fluids*, vol. 2, no. 1, p. 337, 1959, doi :10.1063/1.1705933.

[69] M.Sarrat, *Physique des instabilités de type Weibel.* PhD thesis, Université de Lorraine, 2017.

[70] D. Bohm, I. P. Gross, "Theory of plasma oscillations. a. origin of medium-like behavior," *Phys. Rev.*, vol. 75, no. 12, pp. 1851–1864, 1958, doi :10.1103/PhysRev.75.1851.

[71] H. Che, "Electron two-stream instability and its application in solar and heliophysics," *Modern Physics Letters A*, vol. 31, no. 19, p. 1630018, 2016, doi :10.1142/S0217732316300184.

[72] A. Bret, M.E. Dieckmann, C. Deutsch, "Oblique electromagnetic instabilities for a hot relativistic beam interacting with a hot and magnetized plasma," *Phys. Plasmas*, vol. 13, no. 8, p. 082109, 2006, doi :10.1063/1.2335414.

[73] L. Gremillet, D. Benisti, E. Lefebvre, A. Bret , "Linear and nonlinear development of oblique beam-plasmas instabilities in the relativistic regime," *Phys. Plasmas*, vol. 14, no. 4, p. 040704, 2007, doi :10.1063/1.2714509.

[74] A. Ghizzo, D. Del Sarto, M. Sarrat, "Low- and high-frequency nature of oblique filamentation modes. I.linear theory," *Physics of Plasmas*, vol. 27, no. 7, p. 072103, 2020, doi :10.1063/5.0003697.

[75] M.V. Medvedev, A. Loeb, "Generation of magnetic fields in relativistic shock of gamma-ray burst sources," *Astrophys. J.*, vol. 526, no. 2, pp. 697–706, 1999, doi :10.1086/308038.

[76] C. H. Jaroschek, H. Lesch, R. A. Treumann, "Self-consistent Diffusive Lifetimes of Weibel Magnetic Fields in Gamma-Ray Bursts," *The Astrophysical Journal*, vol. 616, no. 2, p. 1065, 2004, doi :10.1086/424923.

[77] M. V. Medvedev, O. V. Zakutnyaya, "Magnetic fields and cosmic rays in grbs : A self-similar collisionless foreshock," *The Astrophysical Journal*, vol. 696, no. 2, p. 2269, 2009, doi :10.1088/0004-637X/696/2/2269.

[78] G.F. Swadling, C. Bruulsema, F. Fiuza, D.P. Higginson, C.M. Huntington, H.S. Park, B.B. Pollock, W. Rozmus, H.G. Rinderknecht, J. Katz, A. Birkel, J.S. Ross, "Measurement of kinetic-scale current filamentation dynamics and associated magnetic fields in interpenetrating plasmas," *Phys. Rev. Letters*, vol. 124, no. 21, p. 215001, 2020, doi :10.1103/PhysRevLett.124.215001.

[79] UK Atomic Energy Authority :https://ccfe.ukaea.uk/programmes/joint-european-torus/, "Jet : the joint european torus," 2025.

[80] Max Planck Institute for Plasma Physics :https://www.ipp.mpg.de/16195/asdex, "Asdex upgrade," 2025.

[81] Max Planck Institute for Plasma Physics :https://www.ipp.mpg.de/16931/einfuehrung, "Introduction – the wendelstein 7-x stellarator," 2025.

[82] CEA DAM :https://www-lmj.cea.fr/, "Laser mega joule," 2025.

[83] Lawrence Livermore National Laboratory :https://lasers.llnl.gov/about/how-nif-works, "How nif works," 2025.

[84] G. Schurtz, X. Ribeyre, M. Lafon, "Un nouveau procede d'allumage pour la fusion inertielle par laser," *Chocs avancees*, vol. 1, no. 1, pp. 12–13, 2008.

[85] M. Tabak, James Hammer, M. E. Glinsky, W. L. Kruer, S. C. Wilks, J. Woodworth, E. M. Campbell, M. D. Perry, R. J. Mason, "Ignition and high gain with ultrapowerful lasers," *Phys. Plasmas*, vol. 1, no. 1, pp. 1626–1634, 1994, doi :10.1063/1.870664.

[86] P. A. Norreys, R. Allott, R. J. Clarke, J. Collier, D. Neely, S. J. Rose, M. Zepf, M. Santala, A. R. Bell, K. Krushelnick, A. E. Dangor, N. C. Woolsey, R. G. Evans, H. Habara, T. Norimatsu, R. Kodama, "Experimental studies of the advanced fast ignitor scheme," *Phys. Plasmas*, vol. 7, no. 1, pp. 3721–3726, 2000, doi :10.1109/27.893305.

[87] J. Martin Hill, M. H. Key, S. P. Hatchett, R. R. Freeman, "Beam-Weibel filamentation instability in near-term and fast-ignition experiments," *Physics of Plasmas*, vol. 12, no. 8, p. 082304, 2005, doi :10.1063/1.1986988.

[88] A. Karmakar, N. Kumar, A. Pukhov, O. Polomarov, G. Shvets, "Three-dimensional filamentary structures of a relativistic electron beam in fast ignition plasmas," *Phys. Plasmas*, vol. 15, no. 1, p. 120702, 2008, doi :10.1063/1.3042208.

[89] CERN :https://www.home.cern/fr/science/accelerators/large-hadron-collider, "Grand collisionneur de hadrons," 2025.

[90] E. Gschwendtner, E. Adli, L. Amorim, R. Apsimon, R. Assmann, A.-M. Bachmann, F. Batsch, J. Bauche, V.K. Berglyd Olsen et al., "AWAKE, the advanced proton driven plasma wakefield acceleration experiment at CERN," *Nuclear Instruments and Methods in Physics Research Section A : Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 829, no. 1, pp. 76–82, 2016, doi :10.1016/j.nima.2016.02.026.

[91] T. V. Liseikina, F. Califano, V. A. Vshivkov, F. Pegoraro, S. V. Bulanov, "Small-scale electron density and magnetic-field structures in the wake of an ultraintense laser pulse," *Phys. Rev. E*, vol. 60, no. 5, pp. 5991–5997, 1999, doi :10.1063/1.873742.

[92] C. M. Huntington, A. G. R. Thomas, C. McGuffey, T. Matsuoka, V. Chvykov, G. Kalintchenko, S. Kneip, Z. Najmudin, C. Palmer, V. Yanovsky, A. Maksimchuk, R. P. Drake, T. Katsouleas, K. Krushelnick, "Current Filamentation Instability in Laser Wakefield Accelerators," *Phys. Rev. Lett.*, vol. 106, no. 4, p. 105001, 2011, doi :10.1103/PhysRevLett.106.105001.

[93] A. Ghizzo, B. Izrar, P. Bertrand, M.R. Feix, E. Fijalkow, M. Shoucri in *11th European Conf on Plasma simulation*, 1986.

[94] A. Ghizzo, B. Izrar, P. Bertrand, M.R. Feix, E. Fijalkow, M. Shoucri, "BGK STRUCTURES AS QUASI-PARTICLES," *Physics Letters A*, vol. 120, no. 4, pp. 191–195, 1987, doi :10.1016/0375-9601(87)90334-3.

[95] Wikipedia :Contributions Multiples :https://en.wikipedia.org/wiki/Cray-1, "Cray-1," 2024.

[96] A. Ghizzo, P. Bertrand, M.M. Shoucri, T.W. Johnston, E. Fijalkow, M.R. Feix, "A Vlasov Code for the Numerical Simulation of Stimulated Raman Scattering," *Journal of Computational Physics*, vol. 90, no. 2, pp. 431–457, 1990, doi :10.1016/0021-9991(90)90174-Y.

[97] M. L. Bégué, A. Ghizzo, P. Bertrand, "Two-Dimensional Vlasov Simulation of Raman Scattering and Plasma Beatwave Acceleration on Parallel Computers," *Journal of Computational Physics*, vol. 151, no. 2, pp. 458–478, 1999, doi :10.1006/jcph.1999.6193.

[98] Wikipedia :Contributions Multiples :https://en.wikipedia.org/wiki/Cray_T3E, "Cray-T3E," 2024.

[99] Wikipedia :Contributions Multiples :https://fr.wikipedia.org/wiki/Ada_et_Turing, "Ada et turing," 2024.

[100] F. Huot, A. Ghizzo, P. Bertrand, E. Sonnendrucker, O. Coulaud, "Instability of the time splitting scheme for the one-dimensional and relativistic Vlasov-Maxwell system," *Journal of computational physics*, vol. 185, no. 2, pp. 512–531, 2001, doi :10.1016/S0021-9991(02)00079-7.

[101] G. I. Marchuk, *Methods of Numerical Mathematics*. Springer, 1983.

[102] H. Cheng, K. C. Gupta, "An historical note on finite rotations," *Journal of Applied Mechanics*, vol. 56, no. 1, pp. 139–145, 1989, doi :10.1115/1.3176034.

[103] G. Hämmerlin, K.-H. Hoffmann, *Numerical Mathematics*. Springer, 1989.

[104] N. Crouseilles, G. Latu, E. Sonnendrucker, "Hermite spline interpolation on patches for parallelly solving the Vlasov-Poisson equation," *International Journal of Applied Mathematics and Computer Science*, vol. 17, no. 3, pp. 335–349, 2007, doi :10.2478/v10006-007-0028-x.

[105] K. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 3, pp. 302–307, 1966, doi :10.1109/TAP.1966.1138693.

[106] M. Abramowitz, I.A. Stegun, *Handbook of Mathematical Functions*. DOVER PUBLICATIONS, 2020.

[107] M. Alcubierre, *Introduction to Numerical Relativity*. Instituto de Ciencias Nucleares Universidad Nacional Autonoma de Mexico, 2006.

[108] Wikipedia :Contributions Multiples :https://en.wikipedia.org/wiki/Courant-Friedrichs-Lewy_condition, "Courant–friedrichs–lewy condition," 2025.

[109] CNRS :http://www.idris.fr/jean-zay/cpu/jean-zay-cpu-hw.html, "Jean zay : calculateur eviden bullsequana xh3000, hpe sgi 8600," 2024.

[110] M. Brunetti, V. Grandgirard, P. Bertrand, O. Sauter, J. Vaclabik, L. Villard, "Fine-Scale Structures and Negative-Density Regions : Comparison of Numerical Methods for Solving the Advection Equation," *Transport Theory and Statistical Physics*, vol. 34, no. 3, pp. 261–274, 2005, doi :10.1080/00411450500274451.

[111] Wikipedia :Contributions Multiples :https://fr.wikipedia.org/wiki/Transformation_de_Fourier_discrète, "Transformation de fourier discrète," 2025.

[112] Wikipedia :Contributions Multiples :https://fr.wikipedia.org/wiki/Transformation_de_Fourier_rapide, "Transformation de fourier rapide," 2025.

[113] Paul GABORIT :https://perso.imt-mines-albi.fr/∼gaborit/lang/CoursDeFortran/Fortran-node4.html, "Cours de fortran," 1999.

[114] K. Fujimoto, "A new electromagnetic particle-in-cell model with adaptive mesh refinement for high-performance parallel computation," *J. Comput. Phys.*, vol. 230, no. 23, pp. 8508–8526, 2011, doi :10.1016/j.jcp.2011.08.002.

[115] K. Fujimoto, S. Machida, "Electromagnetic full particle code with adaptive mesh refinement technique : Application to the current sheet evolution," *J. Comput. Phys.*, vol. 214, no. 2, pp. 550–566, 2006, doi :10.1016/j.jcp.2005.10.003.

[116] O. Hoenen, *Parallélisation de Méthodes Adaptatives Semi-Lagrangiennes pour la Résolution de l'Équation de Vlasov.* PhD thesis, ULP, 2008.

[117] G. Raj, O. Kononenko, M. F. Gilljohann, A. Doche, X. Davoine, C. Caizergues, Y. Y. Chang, J. P. Couperus Cabadag, A. Debus et al., "Probing ultrafast magnetic-field generation by current filamentation instability in femtosecond relativistic laser-matter interactions," *American Physical Society*, vol. 2, no. 2, p. 023123, 2020, doi :10.1103/PhysRevResearch.2.023123.

[118] A. Ghizzo, D. Del Sarto, H. Betar, "Collisionless heating in Vlasov plasma and turbulence-driven filamentation aspects," *Phys. Plasmas*, vol. 31, no. 7, p. 072109, 2024, doi :10.1063/5.0205253.

[119] F. Jüttner, "Das maxwellsche gesetz der geschwindigkeitsverteilung in der relativtheorie," *Annalen der Physik*, vol. 339, no. 5, pp. 856–882, 1911, doi :10.1002/andp.19113390503.

[120] A. Ghizzo, D. Del Sarto, "Momentum transfer driven by fluctuations in relativistic counter-propagating electron beams," *Plasma Physics and Controlled Fusion*, vol. 63, no. 5, p. 055007, 2021, doi :10.1088/1361-6587/abe695.

[121] Wikipedia :Contributions Multiples :https://en.wikipedia.org/wiki/Amdahl's_ law, "Amdahl's law," 2025.

[122] A. Ghizzo, F. Huot, P. Bertrand, "A non-periodic 2D semi-Lagrangian Vlasov code for laser-plasma interaction on parallel computer," *Journal of computational physics*, vol. 186, no. 1, pp. 47–69, 2003, doi :10.1016/S0021-9991(03)00010-X.

[123] Wikipedia :Contributions Multiples :https://fr.wikipedia.org/wiki/Epsilon_ d'une_ machine, "Epsilon d'une machine," 2025.

[124] O. Polomarov, I. Kaganovich, G. Shvets, "Merging of Super-Alfvénic Current Filaments during Collisionless Weibel Instability of Relativistic Electron Beams," *Phys. Rev. Lett.*, vol. 101, no. 17, p. 175001, 2008, doi :10.1103/PhysRevLett.101.175001.

[125] P. H. Diamond, S. I. Itoh, K. Itoh, *Modern Plasma Physics Vol. 1, Physical Kinetics of Turbulent Plasmas.* Cambridge University Press, 2010.

[126] M. Lesur, P. H. Diamond, "Nonlinear instabilities driven by coherent phase-space structures," *Phys. Rev. E*, vol. 87, no. 3, p. 031101, 2013, doi :10.1103/PhysRevE.87.031101.

[127] M. L. Nastac, R. J. Ewart, J. Juno, M. Barnes, A. A. Schekochihin, "Universal fluctuation spectrum of vlasov-poisson turbulence," *Preprint*, 2025, doi :10.48550/arXiv.2503.17278.

[128] Wikipedia :Contributions Multiples :https://fr.wikipedia.org/wiki/Enstrophie, "Enstrophie," 2025.

[129] Y. Kuramoto, *Chemical Oscillations, Waves and Turbulence.* Springer-Verlag, 1983.

[130] S. K. Xu, Z. B. Guo, Ö. D. Gürcan, "Transition from linear landau damping to nonlinear bernstein-greene-kruskal modes via phase synchronization," *Phys. Rev. E*, vol. 103, no. 3, p. 023208, 2021, doi :10.1103/PhysRevE.103.023208.

[131] S. H. Strogatz, R. E. Mirollo, P. C. Matthews, "Coupled nonlinear oscillators below the synchronization threshold : Relaxation by generalized Landau damping," *Phys. Rev. Lett.*, vol. 68, no. 18, pp. 2730–2733, 1992, doi :10.1103/PhysRevLett.68.2730.

[132] Z. B. Guo, Y. H. Wang,S. K. Xu, "Phase condensation and evaporation : Another look at wave-particle interactions and landau damping," *Phys. Rev. E*, vol. 101, no. 3, p. 030201, 2020, doi :10.1103/PhysRevE.101.030201.

[133] N. G. Van Kampen, "On the theory of stationary waves in plasmas," *Physica*, vol. 21, no. 10, pp. 949–963, 1958, doi :10.1016/S0031-8914(55)93068-8.

[134] P. H. Yoon, "Electromagnetic weibel instability in a fully relativistic bi-maxwellian plasma," *Phys. Fluids B*, vol. 1, no. 6, p. 1336–1338, 1989, doi :10.1063/1.858961.

[135] R. Schlickeiser, "Covariant kinetic dispersion theory of linear waves in anisotropic plasmas. i. general dispersion relations, bi-maxwellian distributions and nonrelativitic limits," *Phys. Plasmas*, vol. 11, no. 12, pp. 5532–5546, 2004, doi :10.1063/1.1806828.

[136] A. Ghizzo, "On the multistream approach of relativistic Weibel instability. III. Comparison with full-kinetic Vlasov simulations," *Physics of Plasmas*, vol. 20, no. 8, p. 082111, 2013, doi :10.1063/1.4817752.

[137] P. Bertand, M. Albrecht-Marc, T. Réveillé, A. Ghizzo, "Vlasov Models for Laser-Plasma Interaction," *Transport Theory and Statistical Physics*, vol. 34, no. 1, pp. 103–126, 2005, doi :10.1080/00411450500255310.

[138] A. Inglebert, A. Ghizzo, T. Reveille, D. Del Sarto, P. Bertrand, F. Califano, "A multi-stream Vlasov modeling unifying relativistic Weibel-type instabilities," *Europhysics Letters Association*, vol. 95, no. 4, p. 45002, 2011, doi :10.1209/0295-5075/95/45002.

[139] A. Inglebert, A. Ghizzo, T. Reveille, D. Del Sarto, F. Califano, "Multi-stream Vlasov model for the study of relativistic Weibel-type instabilities," *Plasma Physics and Controlled Fusion*, vol. 54, no. 8, p. 085004, 2012, doi :10.1088/0741-3335/54/8/085004.

[140] A. Inglebert, A. Ghizzo, T. Reveille, P. Bertrand, F. Califano, "Electron temperature anisotropy instabilities represented by superposition of streams," *Phys. Plasmas*, vol. 19, no. 1, p. 122109, 2012, doi :10.1063/1.4772770.

[141] A. Ghizzo, P. Bertrand, "On the multistream approach of relativistic Weibel instability. I. Linear analysis and specific illustrations," *Physics of Plasmas*, vol. 20, no. 8, p. 082109, 2013, doi :10.1063/1.4817750.

[142] A. Ghizzo, "On the multistream approach of relativistic Weibel instability. II. Bernstein-Greene-Kruskal-type waves in magnetic trapping," *Physics of Plasmas*, vol. 20, no. 8, p. 082110, 2013, doi :10.1063/1.4817751.

[143] A. Ghizzo, M. Sarrat, D. Del Sarto, "Vlasov models for kinetic Weibel-type instabilities," *Journal of Plasma Physics*, vol. 83, no. 1, p. 705830101, 2017, doi :10.1017/S0022377816001215.

[144] M. Antoine , A. Ghizzo, D. Del Sarto, E. Deriaz, "The multi-stream vlasov scheme, a hamiltonian model to reduce the dimensionality of problem in phase space," *Preprint*, 2025, doi :.

[145] R.L. Morse, C.W. Nielson, "Numerical Simulation of the Weibel Instability in One and Two dimensions," *Phys. Fluids*, vol. 14, no. 1, pp. 830–840, 1971, doi :10.1063/1.1693518.

[146] Wikipedia :Contributions Multiples :https://en.wikipedia.org/wiki/Newton's_method, "Newton's method," 2025.

[147] Wikipedia :Contributions Multiples :https://en.wikipedia.org/wiki/Gaussian_integral, "Gaussian integral," 2025.

[148] D.O. Dickman, R.L. Morse, C.W. Nielson, "Numerical Simulation of Axisymmetric, Collisionless, Finite-β Plasma," *Phys. Fluids*, vol. 12, no. 1, pp. 1708–1716, 1969, doi :10.1063/1.1692730.

[149] M.V. Falessi, F. Zonca, "Transport theory of phase space zonal structures," *Phys. Plasmas*, vol. 26, no. 1, p. 022305, 2019, doi :10.1063/1.5063874.

[150] A. Ghizzo, D. Del Sarto, "Low-frequency turbulence suppression by amplification of synchronized zonal flow with energetic particle driven modes," *Nucl. Fusion*, vol. 63, no. 10, p. 104002, 2023, doi :10.1088/1741-4326/acee95.

[151] G. Manfredi, M. Shoucri, P. Bertrand, E. Fijalkow, A. Ghizzo, "The numerical integration of the Vlasov equation possessing an invariant," *Journal of Computational Physics*, vol. 121, no. 2, pp. 298–313, 1995, doi :10.1016/S0021-9991(95)90136-1.

[152] Wikipedia :Contributions Multiples :https://en.wikipedia.org/wiki/Wigner_quasiprobability_distribution, "Wigner quasiprobability distribution," 2025.

[153] G. Brodin, M. Marklund, J. Zamanian, A. Ericsson, P. L. Mana, "Effects of the g factor in semiclassical kinetic plasma theory," *Phys. Rev. Lett.*, vol. 101, no. 1, p. 245002, 2008, doi :10.1103/PhysRevLett.101.245002.

[154] J. Lundin, G. Brodin, "Linearized kinetic theory of spin-1/2 particles in magnetized plasmas," *Physical Review E*, vol. 82, no. 1, p. 056407, 2010, doi :10.1103/PhysRevE.82.056407.

[155] A.J. Brizard, A.A. Chan, "Nonlinear relativistic gyrokinetic vlasov-maxwell equations," *Phys. Plasmas*, vol. 6, no. 1, pp. 4548–4558, 1999, doi :10.1063/1.873742.

[156] J.E. Marsden, R. Montgomery, P.J. Morrison, W.B. Thompson, "Covariant poisson brackets for classical fields," *Annals of Physics*, vol. 169, no. 1, pp. 29–47, 1986, doi :10.1016/0003-4916(86)90157-0.

[157] R.L. Liboff, *Kinetic Theory, classical, quantum and relativistic description.* Springer-Verlag, 2003.

[158] Wikipedia :Contributions Multiples :https://fr.wikipedia.org/wiki/Règle_de_Cramer, "Règle de cramer," 2025.

[159] J.P. Kahane, P.G. Lemarié, *Fourier series and wavelets.* Gordon and Breach Publishers, 1995.

Résumé

En physique des plasmas, il existe plusieurs modèles pour décrire une situation : N-corps, cinétique, gyrocinétique, fluides, magnétohydrodynamique... Les modèles cinétiques sont décrits par l'équation de Vlasov et sont, à l'heure actuelle, principalement résolus à l'aide de codes à particules qui présentent l'inconvénient de contenir du bruit numérique. Une autre méthode cinétique est la méthode Semi-Lagrangienne qui est sans bruit mais plus coûteuse en temps de calcul. L'objectif principal de cette thèse est de mettre en œuvre une technique de raffinement de maillage adaptatif dans l'espace des quantités de mouvement afin de réduire le temps de calcul des codes Semi-Lagrangiens. La première partie de cette thèse détaillera les méthodes numériques de résolution de l'équation de Vlasov et de raffinement de maillage ainsi que la validation du nouveau code. La seconde partie utilisera ce code pour étudier les instabilités de type Weibel dans un cadre relativiste. Ces instabilités sont dues à une anisotropie de température (instabilités Weibel) ou de quantité de mouvement (instabilités de filamentation de courant, oblique et double faisceau) et conduisent à la génération de champs électromagnétiques. La dernière partie présentera une méthode de réduction Hamiltonienne appelée multi-faisceaux qui permet de résoudre l'équation de Vlasov 2D sous la forme d'une somme de faisceaux. Cette méthode permet de réduire le temps de calcul et d'obtenir une estimation plus précise des courants.

Mots-clés: Physique des plasmas, Modèle de Vlasov-Maxwell, Raffinement de maillage adaptatif, Instabilité Weibel, Modèle multi-faisceaux.

Abstract

In plasma physics, there are several models for describing a situation : N-body, kinetic, gyrokinetic, fluid, magnetohydrodynamic... Kinetic models are described by the Vlasov equation and are currently mainly solved using particle codes, which have the disadvantage of containing numerical noise. Another kinetic method is the Semi-Lagrangian method, which is noise-free but more costly in terms of computational time. The main objective of this thesis is to implement an adaptive mesh refinement technique in momentum space in order to reduce the computational time of Semi-Lagrangian codes. The first part of this thesis will detail the numerical methods for solving the Vlasov equation and for mesh refinement, as well as the validation of the new code. The second part will use this code to study Weibel-type instabilities in a relativistic framework. These instabilities are due to temperature anisotropy (Weibel instabilities) or momentum anisotropy (current filamentation, oblique and double beam instabilities) and lead to the generation of electromagnetic fields. The final section will present a Hamiltonian reduction method called multi-stream, which allows the 2D Vlasov equation to be solved as a sum of beams. This method reduces the computational time and gives a more accurate estimate of the currents.

Keywords: Plasma physics, Vlasov-Maxwell model, Adaptive mesh refinement method, Weibel instability, Multi-stream model.