

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies
de l'Information et de la Communication
Spécialité : *Mathématiques et leurs Interactions*

Par

Josselin MASSOT

Modélisation hybride fluide-cinétique de plasmas

Je suis sous le titre

Thèse présentée et soutenue à 127.0.0.1, le un jour
Unité de recherche : IRMAR (UMR CNRS 6625)

Rapporteurs avant soutenance :

Prénom NOM Fonction et établissement d'exercice
Prénom NOM Fonction et établissement d'exercice
Prénom NOM Fonction et établissement d'exercice

Composition du Jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer qu'elle est conforme et devra être répercutée sur la couverture de thèse

Président :	Prénom	NOM	Fonction et établissement d'exercice (à préciser après la soutenance)
Examinateurs :	Prénom	NOM	Fonction et établissement d'exercice
	Prénom	NOM	Fonction et établissement d'exercice
	Prénom	NOM	Fonction et établissement d'exercice
	Prénom	NOM	Fonction et établissement d'exercice
Dir. de thèse :	Nicolas	CROUSEILLES	Directeur de Recherches, Inria Bretagne Atlantique
Co-dir. de thèse :	Anaïs	CRESTETTO	Maître de Conférences, Université de Nantes

REMERCIEMENTS

Mer sea

TABLE DES MATIÈRES

0 Introduction	1
0.1 Hiéarchie des modèle du problème général	1
0.1.1 Dérivation du modèle de Vlasov-Maxwell hybride linéarisé	2
0.1.2 Structure hamiltonienne	4
Appendices	5
0.A Dérivation du modèle de Vlasov-Maxwell hybride linéarisé	5
0.A.1 Dérivation du modèle de Vlasov-Maxwell hybride	5
0.A.2 Dérivation du modèle de Vlasov-Maxwell hybride linéarisé	8
1 Méthodes exponentielles appliquées aux équations cinétiques	11
1.1 Introduction	11
1.2 Exponential integrators and Lawson methods	14
1.3 Linear analysis	16
1.3.1 Lawson methods	17
1.3.2 Exponential integrators	18
1.3.3 Phase space discretization	20
1.3.4 Computing the CFL condition	23
1.4 Numerical simulation: Vlasov-Poisson equations	28
1.4.1 Landau damping test.	30
1.4.2 Bump on tail test.	31
1.5 Numerical simulation: drift-kinetic equations	36

TABLE DES MATIÈRES

1.5.1	Numerical discretization	37
1.5.2	Numerical results	38
Acknowledgement		48
Appendices		49
1.A	Butcher tableaus	49
1.B	WENO5 scheme	51
2 Modèle hybride linéarisé dans le cas $1dx - 1dv$		53
2.1	Introduction	53
2.2	Hiérarchie des modèles	56
2.2.1	Dérivation du modèle de Vlasov hybride linéarisé	57
2.2.2	Quelques propriétés du modèle hybride linéarisé	62
2.3	Structure géométrique du modèle hybride linéarisé VHL	63
2.4	Schémas numériques	65
2.4.1	Méthode de <i>splitting</i> hamiltonien	66
2.4.2	Méthode de Lawson sur le modèle hybride	71
2.4.3	Méthode de pas de temps adaptatif	75
2.5	Relations de dispersion	78
2.5.1	Relations de dispersion dans le cas cinétique	79
2.5.2	Relations de dispersion dans le cas hybride	82
2.5.3	Expression du champ électrique linéarisé	85
2.5.4	Applications	88
2.6	Limite du modèle cinétique vers le modèle hybride	96
2.6.1	Convergence en énergie totale	100
2.6.2	Convergence en température à l'aide des relations de dispersion . .	103
2.6.3	Évolution avec la densité de particules chaudes	106
2.7	Comparaison des deux résolutions hybrides	107

TABLE DES MATIÈRES

2.7.1	Comparaison des deux résolutions hybrides à pas de temps fixe	107
2.7.2	Comparaison des deux méthodes de pas de temps adaptatif	113
2.7.3	À propos des temps de calcul	116
Appendices		118
2.A	Résultats sur les relations de dispersion	118
3 Modèle hybride linéarisé dans le cas $1dz - 3dv$		127
3.1	Introduction	127
3.2	Présentation du modèle	128
3.3	Schémas numériques	130
3.3.1	Méthode de <i>splitting</i> hamiltonien	130
3.3.2	Méthode de Lawson sur le modèle hybride	139
3.4	Résultats numériques	151
3.4.1	Comparaison des solveurs à pas de temps constant	153
3.4.2	Étude à pas de temps adaptatif	156
3.5	Approximation de la partie linéaire	158
3.5.1	Troncature de la série de Taylor	159
3.5.2	Approximant de Padé	160
3.6	Résultats numériques des schémas de Lawson approchés	164
3.7	Optimisations et performances	164
3.7.1	Génération automatique de code	164
3.7.2	Parallélisation	166
3.7.3	Performances	166
Conclusion		167
Première section de la conclusion	167	
Bibliographie		169

INTRODUCTION

Objectif de la thèse

Plasma, contexte physique

Modèles

Méthodes numériques

Rappel des objectifs et plan

Plan :

chap 1 : schémas expo-Lawson pour le cinétique

chap 2-3 : cadre général (6D) modèle hybride

0.1 Hiérarchie des modèle du problème général

Les modèles étudiés dans cette partie sont présentés dans un cadre général multi-dimensionnel en espace et en vitesse. Nous commençons par présenter les équations de Vlasov-Maxwell $3dx - 3dv$:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \frac{q_e}{m_e} (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f = 0 \quad (0.1)$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla_{\mathbf{x}} \times \mathbf{E} \quad (0.2)$$

$$\frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} = \nabla_{\mathbf{x}} \times \mathbf{B} - \mu q_e \int \mathbf{v} f d\mathbf{v} \quad (0.3)$$

$$\nabla_{\mathbf{x}} \cdot \mathbf{E} = \frac{1}{\epsilon_0} \left(q_i \rho_i + q_e \int f d\mathbf{v} \right) \quad (0.4)$$

où $f = f(t, \mathbf{x}, \mathbf{v})$ représente la densité de particules dans l'espace des phases $\{(\mathbf{x}, \mathbf{v}) \in \Omega \times \mathbb{R}^3\}$ avec $\Omega \subset \mathbb{R}^3$, au temps $t \geq 0$, $\mathbf{E} = \mathbf{E}(t, \mathbf{x}) \in \mathbb{R}^3$ représente le champ électrique,

$\mathbf{B} = \mathbf{B}(t, \mathbf{x}) \in \mathbb{R}^3$ représente le champ magnétique, et $\mathbf{B}_0 = (0, 0, B_0)^\top$ est un champ magnétique extérieur supposé constant et d'intensité $B_0 \in \mathbb{R}$. Le système (0.1)-(0.4) forment les équations de Vlasov-Maxwell, et modélisent le transport non-linéaire d'une quantité f dans l'espace des phases $\Omega \times \mathbb{R}^3$. On considérera des conditions périodiques en espace et nulles à l'infini en vitesse.

0.1.1 Dérivation du modèle de Vlasov-Maxwell hybride linéarisé

Le modélisation hybride vient de la volonté de réduire le temps de calcul en considérant la dynamique d'une population de particules comme fluide et non plus cinétique. En suivant cette stratégie, dans le modèle de Vlasov-Maxwell (0.1)-(0.4), on distingue la population de particules f en deux : un premier groupe de particules froides f_c dont la vitesse thermique est faible, et un second groupe de particules, dites chaudes, f_h dont la vitesse thermique est grande. Ces deux populations sont supposées indépendantes et n'interagissent ensemble que via le champ électrique E . Considérant cela, l'équation de Vlasov (0.1) devient :

$$\begin{aligned}\frac{\partial f_c}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_c + \frac{q_e}{m_e} (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_c &= 0 \\ \frac{\partial f_h}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_h + \frac{q_e}{m_e} (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_h &= 0\end{aligned}$$

et l'équation de Maxwell-Ampère (0.3) et de Maxwell-Gauss (0.4) deviennent :

$$\begin{aligned}\frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} &= \nabla_{\mathbf{x}} \times \mathbf{B} - \mu q_e \int \mathbf{v} f_c \, d\mathbf{v} - \mu q_e \int \mathbf{v} f_h \, d\mathbf{v} \\ \nabla_{\mathbf{x}} \cdot \mathbf{E} &= \frac{1}{\epsilon_0} \left(q_i \rho_i + q_e \int f_c \, d\mathbf{v} + q_e \int f_h \, d\mathbf{v} \right)\end{aligned}$$

La population de particules dont la vitesse thermique est suffisamment faible pour être approchée par un fluide est la population des particules dites froides f_c , que l'on souhaite rendre indépendante de la vitesse \mathbf{v} , et seulement dépendante du temps t et de la position \mathbf{x} . En effet, elle représente une densité de particules froides, de faible vitesse thermique, dont on peut supposer qu'elles restent proches d'un équilibre thermodynamique. Pour obtenir le modèle hybride il est nécessaire de calculer les moments de f_c et d'introduire

la densité $\rho_c = \rho_c(t, \mathbf{x})$ et la vitesse moyenne $\mathbf{u}_c = \mathbf{u}_c(t, \mathbf{x})$ des particules froides :

$$\begin{pmatrix} \rho_c \\ \rho_c \mathbf{u}_c \end{pmatrix} = \int_{\mathbb{R}^3} \begin{pmatrix} 1 \\ \mathbf{v} \end{pmatrix} f_c \, d\mathbf{v}$$

Proposition 0.1. *En suppose l'approximation dite de plasma froid utilisée dans la littérature ([86, 49]) qui suppose $f_c(t, \mathbf{x}, \mathbf{v}) = \rho_c(t, \mathbf{x})\delta_{\{\mathbf{v}=\mathbf{u}_c(t,\mathbf{x})\}}(\mathbf{v})$, le modèle hybride s'écrit :*

$$\partial_t f_h + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_h + \frac{q_e}{m_e} (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_h = 0 \quad (0.5)$$

$$\partial_t \rho_c + \frac{1}{q_e} \nabla_{\mathbf{x}} \cdot (\mathbf{j}_c) = 0 \quad (0.6)$$

$$\partial_t \mathbf{j}_c + \nabla_{\mathbf{x}} \cdot \frac{\mathbf{j}_c \otimes \mathbf{j}_c}{q_e \rho_c} - \frac{q_e}{m} (q_e \rho_c \mathbf{E} + \mathbf{j}_c \times (\mathbf{B} + \mathbf{B}_0)) = 0 \quad (0.7)$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla_{\mathbf{x}} \times \mathbf{E} \quad (0.8)$$

$$\frac{1}{c^2} \partial_t \mathbf{E} = \nabla_{\mathbf{x}} \times \mathbf{B} - \mu \mathbf{j}_c - \mu q_e \int \mathbf{v} f_h \, d\mathbf{v} \quad (0.9)$$

$$\nabla_{\mathbf{x}} \cdot \mathbf{E} = \frac{1}{\varepsilon_0} \left(q_i \rho_i + q_e \rho_c + q_e \int f_h \, d\mathbf{v} \right) \quad (0.10)$$

Démonstration. Voir annexe 0.A.1. □

Ce système (0.5)-(0.10) n'est pas nécessairement plus simple à résoudre numériquement que le système cinétique à cause de la non-linéarité introduite par le calcul des moments, mais celui-ci peut être initialisé avec une condition initiale raide de la forme $f_c(t, \mathbf{x}, \mathbf{v}) = \rho_c(t, \mathbf{x})\delta_{\{\mathbf{v}=\mathbf{u}_c(t,\mathbf{x})\}}(\mathbf{v})$. La littérature physique propose de linéarisé la partie fluide (voir [49]). Ainsi on considère maintenant la linéarisation du modèle (0.5)-(0.10) satisfait par $(\rho_c, \mathbf{j}_c, \mathbf{E}, \mathbf{B}, f_h)$ autour de l'équilibre donné par $(\rho_c^{(0)}, 0, 0, 0, f_h^{(0)}(\mathbf{v}))$, avec $f_h^{(0)}(\mathbf{v})$ une fonction telle que $\int \mathbf{v} f_h^{(0)} \, d\mathbf{v} = 0$. L'objectif est d'obtenir un modèle dans lequel la partie fluide est linéaire, tout en conservant la non-linéarité dans l'équation cinétique permettant le couplage des différentes quantités. On écrit alors :

$$\begin{aligned} \rho_c(t, \mathbf{x}) &= \rho_c^{(0)}(\mathbf{x}) + \varepsilon \rho_c^{(1)}(t, \mathbf{x}) \\ \mathbf{j}_c(t, \mathbf{x}) &= \varepsilon \mathbf{j}_c^{(1)}(t, \mathbf{x}) \\ \mathbf{E}(t, \mathbf{x}) &= \varepsilon \mathbf{E}^{(1)}(t, \mathbf{x}) \\ \mathbf{B}(t, \mathbf{x}) &= \varepsilon \mathbf{B}^{(1)}(t, \mathbf{x}) \\ f_h(t, \mathbf{v}, \mathbf{x}) &= f_h^{(0)}(\mathbf{v}) + \varepsilon f_h^{(1)}(t, \mathbf{v}, \mathbf{x}) \end{aligned} \quad (0.11)$$

Proposition 0.2. *Le système (0.5)-(0.10) peut être linéarisé autour de l'état d'équilibre $(\rho_c^{(0)}, 0, 0, 0, f_h^{(0)}(\mathbf{v}))$, avec $f_h^{(0)}(\mathbf{v})$ une fonction telle que $\int \mathbf{v} f_h^{(0)} d\mathbf{v} = 0$, par :*

$$\partial_t f_h + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_h + \frac{q_e}{m_e} (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_h = 0 \quad (0.12)$$

$$\partial_t j_c = \varepsilon_0 \Omega_{pe}^2 \mathbf{E} + \frac{q}{m_e} \mathbf{j}_c \times \mathbf{B}_0 \quad (0.13)$$

$$\partial_t \mathbf{B} = -\nabla_{\mathbf{x}} \times \mathbf{E} \quad (0.14)$$

$$\frac{1}{c^2} \partial_t \mathbf{E} = \nabla_{\mathbf{x}} \times \mathbf{B} - \mu_0 \mathbf{j}_c - \mu_0 q_e \int \mathbf{v} f_h d\mathbf{v} \quad (0.15)$$

Démonstration. Voir annexe 0.A.2. □

0.1.2 Structure hamiltonienne

L'énergie du système (0.12)-(0.15), ou hamiltonien, est :

$$\mathcal{H} = \underbrace{\frac{\varepsilon_0}{2} \int_{\Omega} |\mathbf{E}|^2 d\mathbf{x}}_{\mathcal{H}_E} + \underbrace{\frac{1}{2\mu_0} \int_{\Omega} |\mathbf{B}|^2 d\mathbf{x}}_{\mathcal{H}_B} + \underbrace{\frac{1}{2\varepsilon_0} \int_{\Omega} \frac{1}{\Omega_{pe}^2} |\mathbf{j}_c|^2 d\mathbf{x}}_{\mathcal{H}_{j_c}} + \underbrace{\frac{m_e}{2} \int_{\Omega} \int_{\mathbb{R}^3} |\mathbf{v}|^2 f_h d\mathbf{x} d\mathbf{v}}_{\mathcal{H}_{f_h}} \quad (0.16)$$

APPENDICES

0.A Dérivation du modèle de Vlasov-Maxwell hybride linéarisé

0.A.1 Dérivation du modèle de Vlasov-Maxwell hybride

On s'intéresse ici à démontrer la proposition 0.1.

Démonstration. Pour écrire le modèle hybride, on distingue deux populations de particules dans le modèle de Vlasov-Maxwell (0.1)-(0.4), une population de particules froides f_c dont la vitesse thermique est faible, et une seconde population de particules chaudes f_h dont la vitesse thermique est grande. Ces populations sont supposées indépendantes, l'équation de Vlasov devient :

$$\begin{aligned}\frac{\partial f_c}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_c + \frac{q_e}{m_e} (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_c &= 0 \\ \frac{\partial f_h}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_h + \frac{q_e}{m_e} (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_h &= 0\end{aligned}$$

et l'équation de Maxwell-Ampère (0.3) et de Maxwell-Gauss (0.4) deviennent :

$$\begin{aligned}\frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} &= \nabla_{\mathbf{x}} \times \mathbf{B} - \mu q_e \int \mathbf{v} f_c d\mathbf{v} - \mu q_e \int \mathbf{v} f_h d\mathbf{v} \\ \nabla_{\mathbf{x}} \cdot \mathbf{E} &= \frac{1}{\epsilon_0} \left(q_i \rho_i + q_e \int f_c d\mathbf{v} + q_e \int f_h d\mathbf{v} \right)\end{aligned}$$

On introduit la densité $\rho_c = \rho_c(t, \mathbf{x})$ et la vitesse moyenne $\mathbf{u}_c = \mathbf{u}_c(t, \mathbf{x})$ des particules froides :

$$\begin{pmatrix} \rho_c \\ \rho_c \mathbf{u}_c \end{pmatrix} = \int_{\mathbb{R}^3} \begin{pmatrix} 1 \\ \mathbf{v} \end{pmatrix} f_c d\mathbf{v}$$

Maintenant on calcule les moments de l'équation de Vlasov sur les particules chaudes,

c'est-à-dire que l'on multiplie l'équation par $(1, \mathbf{v})^\top$ puis on intègre en \mathbf{v} :

$$\begin{aligned} \partial_t \int_{\mathbb{R}^3} \left(\frac{1}{\mathbf{v}} \right) f_c + \int_{\mathbb{R}^3} \left(\frac{1}{\mathbf{v}} \right) (\mathbf{v} \cdot \nabla_{\mathbf{x}} f_c) d\mathbf{v} \\ + \frac{q_e}{m_e} \int_{\mathbb{R}^3} \left(\frac{1}{\mathbf{v}} \right) (\mathbf{E} \cdot \nabla_{\mathbf{v}} f_c) d\mathbf{v} + \frac{q_e}{m_e} \int_{\mathbb{R}^3} \left(\frac{1}{\mathbf{v}} \right) (\mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_c d\mathbf{v} = 0. \end{aligned}$$

On réécrit cela comme deux équations, une par moment calculé :

$$\begin{aligned} \partial_t \rho_c + \nabla_{\mathbf{x}} \cdot (\rho_c \mathbf{u}_c) &= 0 \\ \partial_t \rho_c \mathbf{u}_c + \int \mathbf{v} (\mathbf{v} \cdot \nabla_{\mathbf{x}} f_c) d\mathbf{v} + \frac{q_e}{m_e} \int \mathbf{v} (\mathbf{E} \cdot \nabla_{\mathbf{v}} f_c) d\mathbf{v} \\ &\quad + \frac{q_e}{m_e} \int \mathbf{v} ((\mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_c) d\mathbf{v} = 0 \end{aligned}$$

Cette deuxième équation contient plusieurs termes que l'on peut traiter individuellement en les regardant par composante :

$$\begin{aligned} \left(\int \mathbf{v} (\mathbf{v} \cdot \nabla_{\mathbf{x}} f_c) d\mathbf{v} \right)_i &= \int v_i \sum_{j=1}^3 v_j \partial_{x_j} f_c dv_i \\ &= \sum_{j=1}^3 \partial_{x_j} \int v_i v_j f_x dv_i \\ &= \sum_{j=1}^3 \partial_{x_j} \int (\mathbf{v} \otimes \mathbf{v})_{ij} f_x dv_i \\ &= \left(\nabla_{\mathbf{x}} \cdot \int \mathbf{v} \otimes \mathbf{v} f_c d\mathbf{v} \right)_i \\ &= (\nabla_{\mathbf{x}} \cdot (\rho_c \mathbf{u}_c \otimes \mathbf{u}_c))_i \end{aligned}$$

où on note \otimes le produit tensoriel : $(\alpha \otimes \beta)_{i,j} = \alpha_i \beta_j$. Le terme suivant :

$$\begin{aligned} \left(\frac{q_e}{m_e} \int \mathbf{v} (\mathbf{E} \cdot \nabla_{\mathbf{v}} f_c) d\mathbf{v} \right)_i &= \frac{q_e}{m_e} \int v_i \sum_{j=1}^3 E_j \partial_{v_j} f_c dv_i \\ &= -\frac{q_e}{m_e} \sum_{j=1}^3 e_j \int \partial_{v_j}(v_i) f_c dv_i \\ &= -\frac{q_e}{m_e} E_i \int f_c dv_i \\ &= -\frac{q_e}{m_e} (\rho_c E)_i. \end{aligned}$$

Et enfin le troisième terme :

$$\begin{aligned}
 \left(\frac{q_e}{m_e} \int \mathbf{v} ((\mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_c) d\mathbf{v} \right)_i &= \frac{q_e}{m_e} \int v_i \sum_{j=1}^3 (\mathbf{v} \times (\mathbf{B} + \mathbf{B}_0))_j \partial_{v_j} f_c d\mathbf{v}_i \\
 &= -\frac{q_e}{m_e} \sum_{j=1}^3 \int \partial_{v_j}(v_i) (\mathbf{v} \times (\mathbf{B} + \mathbf{B}_0))_j f_c d\mathbf{v}_i \\
 &= -\frac{q_e}{m_e} \int (\mathbf{v} \times (\mathbf{B} + \mathbf{B}_0))_i f_c d\mathbf{v}_i \\
 &= \left[-\frac{q_e}{m_e} \left(\int \mathbf{v} f_c d\mathbf{v} \right) \times (\mathbf{B} + \mathbf{B}_0) \right]_i \\
 &= \left[-\frac{q_e}{m_e} (\rho_c \mathbf{u}_c) \times (\mathbf{B} + \mathbf{B}_0) \right]_i.
 \end{aligned}$$

On peut ainsi réécrire les moments comme :

$$\begin{aligned}
 \partial_t \rho_c + \nabla_{\mathbf{x}} \cdot (\rho_c \mathbf{u}_c) &= 0 \\
 \partial_t (\rho_c \mathbf{u}_c) + \nabla_{\mathbf{x}} \cdot (\rho_c \mathbf{u}_c \otimes \mathbf{u}_c) - \frac{q}{m} (\rho_c \mathbf{E} + (\rho_c \mathbf{u}_c) \times (\mathbf{B} + \mathbf{B}_0)) &= 0.
 \end{aligned}$$

On définit maintenant le courant induit par les particules froides $\mathbf{j}_c = \mathbf{j}_c(t, \mathbf{x})$ comme étant une renormalisation du courant $\rho_c \mathbf{u}_c$:

$$\mathbf{j}_c = q_e(\rho_c \mathbf{u}_c)$$

ce qui nous permet de réécrire l'équation du deuxième moment de f_c comme :

$$\partial_t \mathbf{j}_c + \nabla_{\mathbf{x}} \cdot \frac{\mathbf{j}_c \otimes \mathbf{j}_c}{q_e \rho_c} = \frac{q_e}{m_e} (q_e \rho_c \mathbf{E} + \mathbf{j}_c \times (\mathbf{B} + \mathbf{B}_0))$$

On écrit ainsi les équations de Vlasov-Maxwell hybride :

$$\partial_t f_h + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_h + \frac{q_e}{m_e} (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_h = 0 \quad (0.17)$$

$$\partial_t \rho_c + \frac{1}{q_e} \nabla_{\mathbf{x}} \cdot (\mathbf{j}_c) = 0 \quad (0.18)$$

$$\partial_t \mathbf{j}_c + \nabla_{\mathbf{x}} \cdot \frac{\mathbf{j}_c \otimes \mathbf{j}_c}{q_e \rho_c} - \frac{q_e}{m} (q_e \rho_c \mathbf{E} + \mathbf{j}_c \times (\mathbf{B} + \mathbf{B}_0)) = 0 \quad (0.19)$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla_{\mathbf{x}} \times \mathbf{E} \quad (0.20)$$

$$\frac{1}{c^2} \partial_t \mathbf{E} = \nabla_{\mathbf{x}} \times \mathbf{B} - \mu \mathbf{j}_c - \mu q_e \int \mathbf{v} f_h d\mathbf{v} \quad (0.21)$$

$$\nabla_{\mathbf{x}} \cdot \mathbf{E} = \frac{1}{\epsilon_0} \left(q_i \rho_i + q_e \rho_c + q_e \int f_h d\mathbf{v} \right) \quad (0.22)$$

□

0.A.2 Dérivation du modèle de Vlasov-Maxwell hybride linéarisé

On s'intéresse ici à démontrer la proposition 0.2.

Démonstration. On souhaite linéariser le système (0.5)-(0.10) autour de l'état d'équilibre $(\rho_c^{(0)}, 0, 0, 0, f_h^{(0)}(\mathbf{v}))$, avec $f_h^{(0)}(\mathbf{v})$ une fonction telle que $\int \mathbf{v} f_h^{(0)} d\mathbf{v} = 0$. En réutilisant la réécriture des inconnues (0.11), on peut réécrire le système (0.5)-(0.10) sous la forme :

$$\partial_t f_h + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_h + \frac{q_e}{m_e} (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_h = 0 \quad (0.23)$$

$$\varepsilon \partial_t \rho_c^{(1)} + \frac{\varepsilon}{q_e} \nabla_{\mathbf{x}} \cdot (\mathbf{j}_c^{(1)}) = 0 \quad (0.24)$$

$$\varepsilon \partial_t \mathbf{j}_c^{(1)} + \varepsilon^2 \nabla_{\mathbf{x}} \cdot \frac{\mathbf{j}_c^{(1)} \otimes \mathbf{j}_c^{(1)}}{q_e (\rho_c^{(0)} + \varepsilon \rho_c^{(1)})} - \frac{q_e}{m_e} (q_e (\rho_c^{(0)} + \varepsilon \rho_c^{(1)}) \varepsilon \mathbf{E} + \varepsilon \mathbf{j}_c^{(1)} \times (\varepsilon \mathbf{B}^{(1)} + \mathbf{B}_0)) = 0 \quad (0.25)$$

$$\varepsilon \partial_t \mathbf{B}^{(1)} = -\varepsilon \nabla_{\mathbf{x}} \times \mathbf{E}^{(1)} \quad (0.26)$$

$$\frac{\varepsilon}{c^2} \partial_t \mathbf{E}^{(1)} = \varepsilon \nabla_{\mathbf{x}} \times \mathbf{B}^{(1)} - \varepsilon \mu \mathbf{j}_c^{(1)} - \varepsilon \mu q_e \int \mathbf{v} f_h^{(1)} d\mathbf{v} \quad (0.27)$$

$$\varepsilon \nabla_{\mathbf{x}} \cdot \mathbf{E}^{(1)} = \frac{1}{\epsilon_0} \left(q_i \rho_i + q_e \rho_c + q_e \int f_h d\mathbf{v} \right) \quad (0.28)$$

Nous souhaitons négliger les termes non-linéaires d'ordre $\mathcal{O}(\varepsilon^2)$. Dans le système (0.23)-(0.27) On remarque alors que (0.24) est la seule équation faisant intervenir $\rho_c^{(1)}$, et cette

variable peut être recalculer à partir de (0.28) si besoin. L'équation (0.24) pourra donc être omise du système par la suite. Par abus de notation, pour la lisibilité, on retirera les indices de variables linéarisées lorsqu'il n'y a pas d'ambiguïté, par conséquent les variables \mathbf{j}_c , \mathbf{E} , \mathbf{B} qui apparaissent par la suite, donc d'ordre ε et correspondent à leurs perturbations par rapport à un équilibre nul. On introduit aussi la fréquence de plasma des particules froides $\Omega_{pe}^2 = \frac{q^2 \rho_c^{(0)}}{\varepsilon_0 m_e}$. On peut alors réécrire le système (0.23)-(0.27) comme :

$$\partial_t f_h + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_h + \frac{q_e}{m_e} (\mathbf{E} + \mathbf{v} \times (\mathbf{B} + \mathbf{B}_0)) \cdot \nabla_{\mathbf{v}} f_h = 0 \quad (0.29)$$

$$\partial_t j_c = \varepsilon_0 \Omega_{pe}^2 \mathbf{E} + \frac{q}{m_e} \mathbf{j}_c \times \mathbf{B}_0 \quad (0.30)$$

$$\partial_t \mathbf{B} = -\nabla_{\mathbf{x}} \times \mathbf{E} \quad (0.31)$$

$$\frac{1}{c^2} \partial_t \mathbf{E} = \nabla_{\mathbf{x}} \times \mathbf{B} - \mu_0 \mathbf{j}_c - \mu_0 q_e \int \mathbf{v} f_h \, d\mathbf{v} \quad (0.32)$$

□

Est-ce qu'on ajoute ici l'adimensionnement du système (0.12)-(0.15) ou un lien vers l'annexe où cela sera fait ?

MÉTHODES EXPONENTIELLES POUR RÉSOUDRE UN PROBLÈME HYPERBOLIQUE AVEC APPLICATION AUX ÉQUATIONS CINÉTIQUES

Ce chapitre a pour objectif de présenter les outils de calcul de stabilité, les schémas exponentielles, et se familiariser avec des méthodes d'ordre élevé comme le schéma WENO5. Le calcul de stabilité pour une méthode WENO a été initié dans [88] et [71], le calcul de condition de CFL a été poursuivit dans [64] ; ce chapitre continue ce travail et propose un algorithme d'estimation de condition de CFL entre un schéma WENO et une méthode Runge-Kutta donnée, en plus d'étudier la stabilité des méthodes exponentielles en temps pour une équation de transport, avec une application a des équations cinétiques que sont les équations de la physique des plasmas. Ce travail est synthétisé par cet article qui est un travail collaboratif avec Nicolas Crouseilles¹ et Lukas Einkemmer² qui a mené a un article *Exponential methods for solving hyperbolic problems with application to kinetic equations* publié dans *Journal of Computational Physics* en 2020 [19].

1.1 Introduction

The goal of this work is to develop high order and efficient numerical methods for nonlinear collisionless kinetic models, such as the Vlasov-Poisson equations or drift-kinetic models. In most situations, the nonlinearity in the transport term originates from the

1. Univ Rennes, Inria Bretagne Atlantique (MINGuS) & ENS Rennes
 2. Department of Mathematics, Univeristy of Innsbruck, Austria

coupling with a Poisson type problem that is used to compute the electric field.

Historically, particle in cell methods have been extensively used to treat kinetic problems. In this approach, the unknown is sampled by discrete particles which are advanced in time using an ODE solver, whereas the electric field is computed on a spatial grid. For some problems, these methods can tackle high dimensional kinetic problems with a relatively low computational cost. However, they also suffer from numerical noise which pollutes the accuracy in low density regions of phase space. Moreover, as the number of particles is increased the error only decreases as the inverse of the square of the number of particles. Thus, convergence is slow. For a review of particle methods we refer the reader to [87].

On the other hand, Eulerian methods (*e.g.* finite volumes or finite differences), which directly discretize the phase space, are able to reach high order accuracy in time, space, and velocity. However, in addition to the fact that they are costly, these methods usually suffer from stability constraints that force a relation between the time step and the phase space grid sizes, the so-called Courant–Friedrichs–Lewy (CFL) condition. Hence, a large number of time steps is required to reach the long times that are often required in plasma physics applications.

To overcome this CFL condition, semi-Lagrangian methods have been developed during the last decades. These methods realize a compromise between the Lagrangian (*i.e.* particle in cell) and Eulerian approaches by exploiting the characteristics equations to overcome the CFL condition, while still performing computations on a grid in both space and velocity [13, 79, 38]. This approach is usually combined with splitting methods to avoid a costly multidimensional interpolation step. This allows for a separate treatment of the terms in the equation and the corresponding characteristic curves can then, at least in some situations, be computed analytically. For purely hyperbolic problems, the setting we consider here, it is also possible to construct high order splitting schemes [10].

Splitting results in a very accurate and efficient scheme for the Vlasov-Poisson equation. This is the case because the problem is only split into two parts, the characteristics of which can then be solved exactly in time. However, this is not necessarily true for more complicated equations such as gyrokinetic or drift-kinetic models. Indeed, for the drift-kinetic model, a three terms splitting has to be performed so that a relatively large number of stages are required to reach high order accuracy in time. In addition, some stages can not be solved exactly in time and thus require additional numerical work to

approximate them.

In [20] an alternative approach based on exponential integrators was proposed. These schemes exploit the fact that in many applications where (gyro)kinetic models are used, the most stringent CFL condition is associated with the linear part of the model. This observation serves as the basis for the numerical methods we will consider in this work. Starting from the variation of constant formula, the linear part of the model will be solved exactly as part of an exponential integrator, whereas the nonlinear part, which is very often orders of magnitudes less stiff than the linear part, will be treated explicitly in time. In practice, the linear part can then be solved in phase space by using Fourier techniques or semi-Lagrangian schemes and the nonlinear part is approximated by standard finite difference/finite volume/discontinuous Galerkin techniques.

The numerical results presented in [20] were generally very favorable. The authors were able to take larger time steps compared to what has been reported for splitting methods in the literature and the computational cost was significantly reduced. In addition, since exponential integrators treat the nonlinear part explicitly, they can be adapted much more easily to different models. Despite these many favorable properties, the largest stable time step size was difficult to predict and varied significantly from method to method. Moreover, as we will see, many exponential integrators can behave rather erratically depending on the specific configuration of the simulation.

Thus, the main goal in this paper is to understand the stability of exponential integrators when applied to purely hyperbolic problems. While there is a large literature and well established theory for exponential integrators applied to parabolic problems (see [48] and references therein), we will see that for purely hyperbolic problems many surprises are encountered. Based on this analysis we will then propose to use a class of exponential methods, Lawson methods, that do not suffer from the described deficiency. Our analysis explains the efficient and robust behavior of Lawson integrators for this kind of problems. We will also present numerical results for both the Vlasov–Poisson equations and a drift-kinetic model that confirm the expected behavior and shows that using this approach significant performance improvements compared to the exponential integrators used in [20], and by extension compared to splitting methods, can be attained.

The paper is organized as follows. First, we offer a brief introduction to exponential methods (section 1.2). Then, in section 1.3 a linear stability analysis is performed for both the time and phase space discretization. For the explicit part, we consider both centered

differences (such as Arakawa’s method) and weighted essentially non-oscillatory schemes (WENO) schemes. In sections 1.4 and 1.5 we investigate the performance of these methods for the Vlasov–Poisson equations and a four-dimensional drift-kinetic model, respectively.

1.2 Exponential integrators and Lawson methods

Exponential methods are a class of time integration schemes that are applied to differential equations of the form

$$\dot{u} = Au + F(u), \quad (1.1)$$

where A is a matrix and F is a, in general nonlinear, function of u . Usually, both A and F are the result of a spatial semi-discretization of a partial differential equation. Exponential methods are applied to problems where A is stiff or otherwise poses numerical challenges, while F can be treated explicitly. For the hyperbolic case, a prototypical example is the Vlasov equation (1.11). Exponential methods are advantageous if the largest velocity is large compared to the electric field. Then the linear part has a much more stringent CFL condition than the nonlinear part of the equation. We will consider this example in some detail later in the paper.

In this paper, we will consider two types of exponential methods. The idea of *exponential integrators* is to use the variation of constants formula to rewrite equation (1.1) in the following form

$$u(t_n + \Delta t) = \exp(\Delta t A)u(t_n) + \int_0^{\Delta t} \exp((\Delta t - s)A)F(u(t_n + s)) ds,$$

where we denote the time step size by $\Delta t > 0$ and $t_n = n\Delta t$ with $n \in \mathbb{N}$. This expression is still exact; *i.e.* no approximation has been made. Note, however, that this can not be used as a numerical method as evaluating the integral would require the knowledge of $u(t_n + s)$, which is not available. The idea of an exponential integrator is to approximate the nonlinear part $F(u(t_n + s))$ in terms of the available data. In the simplest case we just evaluate it at the left endpoint. That is, we use $F(u(t_n + s)) \approx F(u^n)$. Then we can integrate the term $\exp((\Delta t - s)A)$ exactly and obtain.

$$u(t_n + \Delta t) \approx u^{n+1} = \exp(\Delta t A)u^n + \Delta t \varphi_1(\Delta t A)F(u^n),$$

where $\varphi_1(z) = (e^z - 1)/z$ is an entire function. This is the first order exponential Euler method. In a similar way exponential Runge–Kutta methods can be constructed. We refer to the literature, in particular the review article [48], for more details.

Another ansatz to remove the stiff linear term from equation (1.1) is to introduce the change of variable

$$v(t) := \exp(-tA)u(t).$$

Plugging this into equation (1.1) yields

$$\dot{v}(t) = \exp(-tA)F(\exp(tA)v(t)). \quad (1.2)$$

Now we apply an explicit Runge–Kutta method to the transformed equation. In the simplest case, applying the explicit Euler scheme yields

$$v(t_n + \Delta t) \approx v^{n+1} = v^n + \Delta t \exp(-t_n A)F(\exp(t_n A)v^n).$$

Reversing the change of variables yields

$$u^{n+1} = \exp(\Delta t A)u^n + \Delta t \exp(\Delta t A)F(u^n).$$

This is the Lawson–Euler method, also a method of order one. Lawson methods are also commonly referred to as integrating factor methods. We immediately see that any explicit Runge–Kutta method applied to equation (1.2) uniquely determines a Lawson scheme. We call the chosen Runge–Kutta method the *underlying Runge–Kutta method*. For more details we refer the reader to [61, 8, 84, 67].

The example of the Lawson–Euler method already shows the similarity between Lawson schemes and exponential integrators. In fact, Lawson methods can be considered a subclass of exponential integrators. That is, they are a type of exponential integrators that only involve the exponential, but no other matrix functions. For the purpose of this paper we keep the nomenclature distinct. A Lawson scheme is a numerical method obtained as described above, while an exponential integrator is a numerical scheme that, in addition to the matrix exponential, uses other matrix functions.

The efficiency of exponential methods crucially depends on a good method to evaluate the application of the required matrix functions to a vector. A range of methods has been developed to accomplish this. For example, Krylov methods or interpolation at Leja points

can be used for a wide range of problems; see, for example, [44, 46, 68, 6, 7]. However, often the most efficient approach is to exploit particular knowledge about the differential equation under consideration. For example, in the hyperbolic case A might be a linear advection operator. In this case the application of $\exp(\Delta t A)$ can be computed by using Fourier techniques or semi-Lagrangian schemes. Much research effort has been dedicated towards improving spectral and semi-Lagrangian schemes for kinetic problems [23, 33, 27, 38, 40, 54, 70, 72, 79, 13, 75, 17, 18, 30, 35, 34] and obtaining good performance on state of the art HPC systems [73, 36, 4, 58, 66, 28, 22, 31].

Before proceeding, let us note that for parabolic problems, *i.e.* where A is an elliptic operator, a mature theory for exponential integrators is available. We again refer the reader to the review article [48]. In this setting there are relatively few surprises with respect to stability and even rigorous convergence results are available. In addition, exponential integrators have been considered for problems that include both hyperbolic and parabolic terms (see, for example [65, 82, 37, 32]). An interesting point to make is that in this community Lawson methods have all but lost their appeal. In fact, there are many reasons why exponential integrators are to be preferred. For example, if a Krylov method is used to compute the matrix functions, the φ_1 function usually converges faster than the exponential. In addition, exponential integrators that retain their full order for non-homogeneous boundary conditions have been constructed [47]. It has been shown that this property can not be achieved for Lawson methods [45]. However, the situation for purely hyperbolic problems is markedly different. Most of the theoretical results that have been obtained in an abstract framework do not apply and there is relatively little literature available. We will see that the stability for exponential integrators in the fully hyperbolic setting is full of surprises. Moreover, since for kinetic problems we usually have efficient methods to compute the matrix exponential and complicated boundary conditions are rather rare, Lawson methods are an attractive choice due to their improved stability, as we will see.

1.3 Linear analysis

Determining the stability of a numerical scheme by conducting an analysis of a linear and scalar test equation is very well established in the literature. Usually, the Dahlquist test equation $\dot{u} = \lambda u$ is considered. The justification for this is that a linear ODE can

be written as $\dot{u} = Au$. Once the matrix A is diagonalized we essentially obtain the test equation. For linear PDEs we first perform a space discretization. Then the same argument can be applied to the resulting differential equation and stability constraints, such as the famous CFL condition, can be deduced. In the nonlinear case this, of course, only gives an indication for stability. Nevertheless, in many practical problems the theory derived in this fashion agrees very well with what is observed in numerical experiments.

The situation for exponential integrators and Lawson methods is more complicated as we separate two parts of the differential equation. Thus, we will consider the following test equation

$$\dot{u} = iau + \lambda u, \quad a \in \mathbb{R}, \lambda \in \mathbb{C}, \quad u(0) = u_0 \in \mathbb{C}. \quad (1.3)$$

We note that we are here exclusively interested in equations with two hyperbolic parts. The reason why we allow λ to lie in the complex plane is that some space discretization schemes introduce numerical diffusion. Thus, the discretization moves the eigenvalues from the imaginary axis to the left half complex plane.

Although this test equation is used frequently in the literature, its use is also frequently criticized. The reason for this criticism is that in the linear case the equation $\dot{u} = Au + Bu$ can only be transformed to the form given in equation (1.3) if A and B are simultaneously diagonalizable. This is a severe restriction which is usually not true in practice. Thus, the test equation, even in the linear case, gives only a necessary condition for stability. While this argument is certainly correct, we emphasize that if a numerical integrator does not work for the test equation (1.3) there is not much hope that it would work for more complicated problems. Therefore, the test equation is still useful and in fact we will see that many of the deficiencies of exponential integrators observed in practice can be illustrated well using the test equation (1.3).

1.3.1 Lawson methods

Applying a Lawson method to the test equation (1.3) proceeds as follows. First, we introduce the change of variable

$$v(t) = e^{-iat}u(t).$$

which yields the equation

$$\dot{v} = e^{-iat} \lambda(e^{iat} v) = \lambda v.$$

Thus, we precisely obtain the Dahlquist test equation. We now apply an explicit Runge–Kutta method to that equation. It is well known that this results in

$$v^{n+1} = \phi(z)v^n, \quad z = \lambda\Delta t,$$

where ϕ is the so-called stability function. Reversing the change of variable we obtain

$$u^{n+1} = e^{ia\Delta t} \phi(z)u^n, \quad z = \lambda\Delta t.$$

The condition for stability is $|e^{ia\Delta t} \phi(z)| = |\phi(z)| \leq 1$. Thus, the linear stability characteristics of a Lawson scheme is identical to that of its underlying Runge–Kutta method.

This makes the problem rather easy as the stability constraint for explicit Runge–Kutta methods has been extensively studied in the literature. In our present application we are primarily interested in obtaining numerical methods that maximize the part of the imaginary axis that is included in the domain of stability. It is well known that an s stage method can include at most $i[-(s - 1), s - 1]$. This is, for example, stated as an exercise in [42, Chapter. IV.2, exercise 3]. Thus, unfortunately, there is no analog to Runge–Kutta–Chebyshev methods for hyperbolic problems.

For the sake of completeness we plot in Figure 1.1 the curve given by $|\phi(z)| = 1$ for different Runge–Kutta methods. The only non-standard method here is *RK(3,2) best* which is a three stage second order method that has been purposefully constructed to enhance stability on the imaginary axis (see Appendix 1.A for its Butcher tableau). We also emphasize that the stability domain of the classic four stage fourth order Runge–Kutta method is quite close to the theoretical bound $i[-(s - 1), s - 1]$, $s = 4$.

1.3.2 Exponential integrators

We now apply commonly used exponential integrators to the test equation (1.3). In this work we will consider the following methods: ExpRK22 (a classic two stage second order method), the method of Cox–Matthews [15], the method of Hochbruck–Ostermann [47], and the method of Krogstad [57]. We refer to [48] for more details and to Appendix 1.A for the Butcher tableaus of these methods.

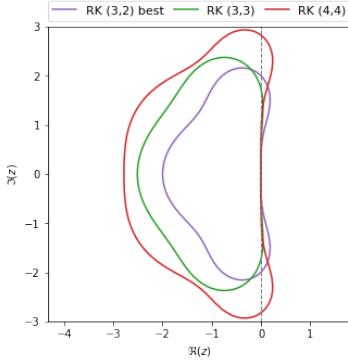


Figure 1.1 – The domain of stability for some classic explicit Runge–Kutta methods is shown. The nomenclature $RK(s,p)$ denotes a method with s stages that is of order p . The Butcher tableaus of these methods are given in Appendix 1.A.

For the sake of brevity we will only detail the calculation for the ExpRK22 scheme. Applying this method to the test equation we obtain

$$\begin{aligned} k_1 &= e^{ia\Delta t} u^n + \Delta t \varphi_1(i a \Delta t) \lambda u^n \\ u^{n+1} &= e^{ia\Delta t} u^n + \Delta t \left[(\varphi_1(i a \Delta t) - \varphi_2(i a \Delta t)) \lambda u^n + \varphi_2(i a \Delta t) \lambda k_1 \right], \end{aligned}$$

where $\varphi_1(z) = (e^z - 1)/z$ and $\varphi_2(z) = (e^z - 1 - z)/z^2$ are entire functions. This yields the stability function

$$\phi(z) = e^{ia\Delta t} + \left(\varphi_1(i a \Delta t) - \varphi_2(i a \Delta t) + e^{ia\Delta t} \varphi_2(i a \Delta t) \right) z + \varphi_1(i a \Delta t) \varphi_2(i a \Delta t) z^2,$$

where, as before, we use $z = \lambda \Delta t$.

Our first observation is that, in contrast to Lawson methods, the behavior of this stability function can not be understood by the domain of stability of the underlying $RK(2,2)$ method, *i.e.* the explicit method we obtain if we take $a \rightarrow 0$. In fact, as we vary a the domain of stability changes drastically. The domain of stability for the four exponential integrators (ExpRK22, Cox–Matthews, Hochbruck–Ostermann, and Krogstad) is plotted in Figure 1.2 for $a\Delta t = 1.1$ and $a\Delta t = 3.4$. It is most striking that for large $|a\Delta t|$ the domain of stability does not contain a symmetric interval of the imaginary axis. It should be evident that this has the potential to cause severe stability issues.

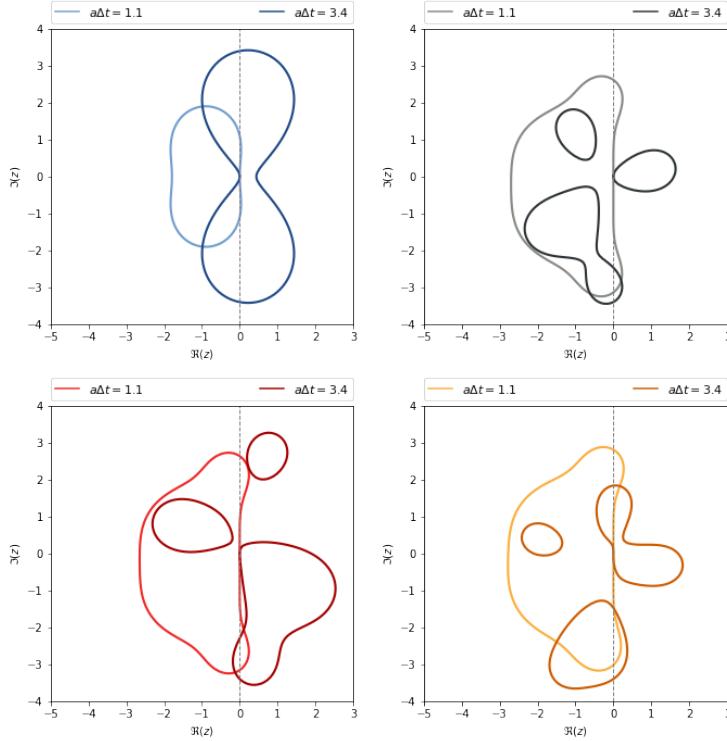


Figure 1.2 – Stability domain of exponential integrators for two different values of $a\Delta t \in \{1.1, 3.4\}$. From top left to bottom right: ExpRK22, Krogstad, Cox–Matthews and Hochbruck–Ostermann.

1.3.3 Phase space discretization

We start from the two-dimensional linear transport equation

$$\partial_t f + d\partial_x f + b\partial_v f = 0, \quad d, b \in \mathbb{R}, \quad x \in [0, 2\pi], \quad v \in [-v_{\max}, v_{\max}], \quad (1.4)$$

where $v_{\max} > 0$ refers to the truncated velocity domain. The sought-after distribution function is $f(t, x, v)$ and we impose periodic boundary conditions in the x -direction. We assume that d and b are constants and thus the corresponding operators commute. This is an idealization of the Vlasov equation we will consider in the next section. In preparation for that example it is most useful to think that d is large and thus would induce a stringent CFL condition if discretized by an explicit scheme.

We now have to discretize this equation both in the x and the v direction. In the spatial direction x , we will consider a spectral approximation. Performing a Fourier trans-

formation of equation (1.4) yields

$$\partial_t \hat{f}_k + idk \hat{f}_k + b \partial_v \hat{f}_k = 0, \quad (1.5)$$

where $\hat{f}_k(t, v)$ denotes the Fourier transform of $f(t, x, v)$ with respect to x . The corresponding frequency is denoted by k .

We now perform the discretization in the v direction. The grid points are denoted by $v_j = -v_{\max} + j\Delta v$, with $\Delta v = 2v_{\max}/N_v$, where N_v is the number of points. We will consider two options here. Namely, either using a centered difference scheme or an upwind scheme.

Centered scheme in v .

The classic centered scheme is obtained by approximating the velocity derivative in equation (1.5) by

$$(\partial_v \hat{f}_k)(v_j) \approx \frac{\hat{f}_{k,j+1} - \hat{f}_{k,j-1}}{2\Delta v},$$

where $\hat{f}_{k,j}$ is an approximation of $\hat{f}_k(v_j)$. Inserting this centered approximation in (1.5) yields

$$\partial_t \hat{f}_{k,j} + idk \hat{f}_{k,j} + b \frac{\hat{f}_{k,j+1} - \hat{f}_{k,j-1}}{2\Delta v} = 0. \quad (1.6)$$

The system is already diagonal with respect to the index k . We now also diagonalize it with respect to the index j . To do that we express the function in terms of its Fourier modes with respect to v . That is,

$$\hat{f}_{k,j} = \sum_m \bar{f}_{k,m} \exp\left(i \frac{2\pi m}{2v_{\max}} v_j\right),$$

where $\bar{f}_{k,m}$ denotes the (double) Fourier transform of f with frequency in space k and frequency in velocity m . Inserting this into equation (1.6) yields

$$\partial_t \bar{f}_{k,m} + idk \bar{f}_{k,m} + b \frac{i \sin(2\pi m \Delta v / (2v_{\max}))}{\Delta v} \bar{f}_{k,m} = 0. \quad (1.7)$$

We immediately see that this equation is precisely in the form of equation (1.3) as studied in the previous section. We also observe that $\lambda \in i\mathbb{R}$. That is, the eigenvalues for the centered difference approximation lie exclusively on the imaginary axis. One immediate consequence is that for Lawson methods the CFL condition is given by $b\Delta t < C\Delta v$,

where C is chosen such that $i[-C, C]$ lies in the domain of stability of the underlying Runge–Kutta method.

Linearized WENO approximation in v .

A common technique to discretize hyperbolic partial differential equations is to use the so-called weighted essentially non-oscillatory schemes (WENO) schemes. These are nonlinear schemes that limit oscillations in regions where sharp gradients occur, but still yield high order accuracy in smooth regions of the phase space. In the linear case WENO schemes reduce to upwind discretizations. Here, we will consider the LW5 scheme (the linearized version of the WENO5 scheme as considered in [3, 64, 71, 88]) that is given by (from now on we assume w.l.o.g. that $b > 0$)

$$(\partial_v \hat{f}_k)(v_j) \approx \frac{1}{\Delta v} \left(-\frac{1}{30} \hat{f}_{k,j-3} + \frac{1}{4} \hat{f}_{k,j-2} - \hat{f}_{k,j-1} + \frac{1}{3} \hat{f}_{k,j} + \frac{1}{2} \hat{f}_{k,j+1} - \frac{1}{20} \hat{f}_{k,j+2} \right).$$

We now perform the same analysis as for the centered scheme (see [3, 16]). This yields

$$\begin{aligned} \left(\frac{2\pi}{2v_{\max}} im \bar{f}_{k,m} \approx \right) \mu_m \bar{f}_{k,m} := \frac{\bar{f}_{k,m}}{\Delta v} \left(-\frac{1}{30} e^{-\frac{3im\pi\Delta v}{v_{\max}}} + \frac{1}{4} e^{-\frac{2im\pi\Delta v}{v_{\max}}} - e^{-\frac{im\pi\Delta v}{v_{\max}}} \right. \\ \left. + \frac{1}{3} + \frac{1}{2} e^{\frac{im\pi\Delta v}{v_{\max}}} - \frac{1}{20} e^{\frac{2im\pi\Delta v}{v_{\max}}} \right). \end{aligned} \quad (1.8)$$

We then obtain the equation

$$\partial_t \bar{f}_{k,m} + idk \bar{f}_{k,m} + b \mu_m \bar{f}_{k,m} = 0. \quad (1.9)$$

Once again this is precisely the form of equation (1.3), where $a = dk \in \mathbb{R}$ and $\lambda = b\mu_m \in \mathbb{C}$. The main difference to the centered difference scheme is that λ is not necessarily on the imaginary axis. In fact, the eigenvalues acquire a negative real part which stabilizes the scheme and avoids spurious oscillations, but also adds unphysical dissipation to the numerical method.

Remarque 1.1. Let us remark that performing a Fourier approximation in v is also possible. Using the same notation as before, the counterpart of (1.7) and (1.9) in that case is

$$\partial_t \bar{f}_{k,m} + idk \bar{f}_{k,m} + bi \frac{2\pi m}{2v_{\max}} \bar{f}_{k,m} = 0.$$

It is worth mentioning that this last equation can be obtained by considering the limit

Δv goes to zero in (1.7) or (1.9). The stability condition can be computed as for the centered difference case, since the eigenvalues of the Fourier approximation also lies on the imaginary axis. The CFL condition is given by $b\pi < C\Delta v$, where C is chosen such that $i[-C, C]$ lies in the domain of stability of the underlying Runge–Kutta method.

1.3.4 Computing the CFL condition

Equipped with the knowledge of the domain of stability for the time discretization and the eigenvalues of the space discretization, we are now in a position to determine the CFL condition for the linear transport equation (1.4). This task will be rather easy to accomplish for the Lawson schemes, where the stability does not depend on the advection speed for the transport in the x direction. However, for exponential integrators even this linear analysis is rather complicated, as we will see.

1.3.4.1 Centered scheme in v .

In the case of centered approximation of the velocity derivative, the Fourier multiplier is a pure imaginary complex number (see equation (1.7)). We thus look for $y_{\max} \in \mathbb{R}_+$ such that the interval $i(-y_{\max}, y_{\max}) \subset \mathcal{D}$, where \mathcal{D} is the domain of stability for the chosen time integrator.

Lawson integrators.

We simply look for the largest value y_{\max} such that $i(-y_{\max}, y_{\max}) \subset \mathcal{D}$. The corresponding values for a number of schemes are listed in Table 1.1. These values have to be understood in the following way: they induce the CFL condition $b\Delta t \leq y_{\max}\Delta v$ for the discretized equation (1.7), where Δt denotes the time step size and Δv is the velocity mesh size.

Methods	Lawson($RK(3, 2)$ best)	Lawson($RK(3, 3)$)	Lawson($RK(4, 4)$)
y_{\max}	2	$\sqrt{3}$	$2\sqrt{2}$

Table 1.1 – CFL number for some Lawson schemes applied to (1.7).

Exponential integrators.

For the exponential integrators the domain of stability is very sensitive to the value of $(a\Delta t)$. To get an idea of what we can expect, we consider the quantity

$$y_{\max} = \min_{(a\Delta t) \in \mathbb{R}} y_{\max}^{\exp}(a\Delta t).$$

As before, $y_{\max}^{\exp}(a\Delta t)$ is the largest value such that :

$$i(-y_{\max}^{\exp}(a\Delta t), y_{\max}^{\exp}(a\Delta t)) \subset \mathcal{D},$$

where \mathcal{D} is the domain of stability for the chosen exponential time integrator for a given $(a\Delta t)$. Even for relatively simple numerical methods it is not possible to compute this quantity analytically. Thus, we resort to numerical approximations. Unfortunately, it turns out that for most exponential integrators this value is zero. This can be appreciated by considering Figure 1.2 once more. Clearly, there are values of $(a\Delta t)$ such that no relevant part of the imaginary axis (or only half the imaginary axis) is part of the domain of stability. Thus, most exponential integrators are unstable in the von Neumann sense. However, this is not what we observe in practice. In fact, already the results presented in [20] indicate that we can successfully run numerical simulations using, for example, the Cox–Matthews scheme. There are two major points to consider here

- The y_{\max} obtained is a worst case estimate. In fact, we know that for $\Delta t \rightarrow 0$ we regain the stability of the underlying Runge–Kutta method. Thus, for small $(a\Delta t)$ the methods is expected to work well.
- As is usually done we have mandated that $|\phi(z)| \leq 1$. However, strictly speaking this is not necessary for practical simulation. If we assume that $|\phi(z)| \leq 1 + \varepsilon$ and we take n steps the amplification of the error is given by $(1 + \varepsilon)^n$. In the limit $n \rightarrow +\infty$ this quantity diverges. However, since we usually do not take infinitely small time steps we still can hope to obtain a relatively accurate approximation, especially if ε is small. In particular, if $\varepsilon = C\Delta t = Ct_{final}/n$ (where t_{final} denotes the final time and n the number of iterations) we have $(1 + \varepsilon)^n = (1 + \frac{Ct_{final}}{n})^n \leq \exp(Ct_{final})$ and thus the scheme is stable, while the error constant is increased by $\exp(Ct_{final})$.

To investigate this further, we propose to relax the stability condition by introducing a threshold $\varepsilon > 0$ in the definition of the stability domain

$$\mathcal{D}_\varepsilon = \{z \in \mathbb{C} : |\phi(z)| \leq 1 + \varepsilon\}. \quad (1.10)$$

In Figure 1.3, we plot the domain of stability for the Cox–Matthews method and $a\Delta t = 3.4$ for $\varepsilon = 0$ and $\varepsilon = 10^{-2}$. One can observe that in the latter case a non-zero $y_{\max}^{\exp}(3.4)$ is obtained. We also call attention to the fact that the part of the imaginary axis included in this relaxed stability domain is not symmetric.

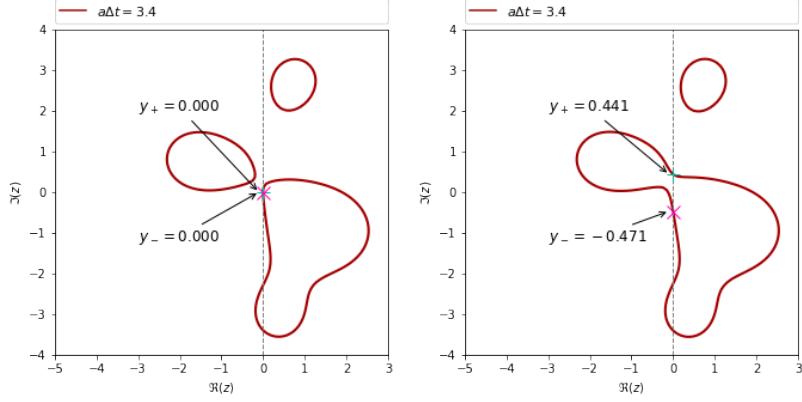


Figure 1.3 – Example of variation of y_+ and y_- when we relax the stability condition for the Cox–Matthews scheme. We represent \mathcal{D}_0 on the left, and $\mathcal{D}_{10^{-2}}$ on the right with the values y_+ and y_- such that $i(y_-, y_+) \subset \mathcal{D}_\varepsilon$.

In Figure 1.4, we plot the dependence of y_{\max}^{\exp} as a function of $(a\Delta t)$ for $\varepsilon = 10^{-2}$ and the ExpRK22 method. Let us recall that for $\varepsilon = 0$, the method gives $y_{\max} = 0$. One can observe that the domain of stability \mathcal{D}_ε of this method is still symmetric with respect to the real axis. In addition, the method becomes more stable as $|a\Delta t|$ increases. Thus, the behavior of the method is completely different from the configuration with $\varepsilon = 0$.

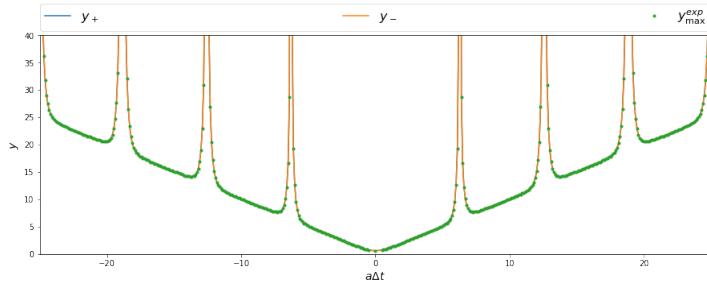


Figure 1.4 – y_{\max}^{\exp} , $|y_+|$ and $|y_-|$ as a function of $a\Delta t$ for the ExpRK22 method with $\varepsilon = 10^{-2}$.

In Figure 1.5, we plot y_{\max}^{\exp} as a function of $(a\Delta t)$ for the Hochbruck–Ostermann method (once again for $\varepsilon = 10^{-2}$). This schemes also gives $y_{\max} = 0$ for $\varepsilon = 0$. In this case the domain of stability is not symmetric and the stability depends quite erratically

Methods	ExpRK22	Krogstad	Cox–Matthews	Hochbruck–Ostermann
$y_{\max}^{\exp}(\varepsilon = 10^{-3})$	0.300	0.100	0.150	0.250
$y_{\max}^{\exp}(\varepsilon = 10^{-2})$	0.551	0.200	0.450	0.501
$y_{\max}^{\exp}(\varepsilon = 10^{-1})$	1.001	0.601	1.351	1.702

Table 1.2 – CFL number, assuming the relaxed stability constraint, for some exponential integrators applied to (1.7).

on the value of $(a\Delta t)$. In Table 1.2 we have summarized the values of y_{\max} for the four

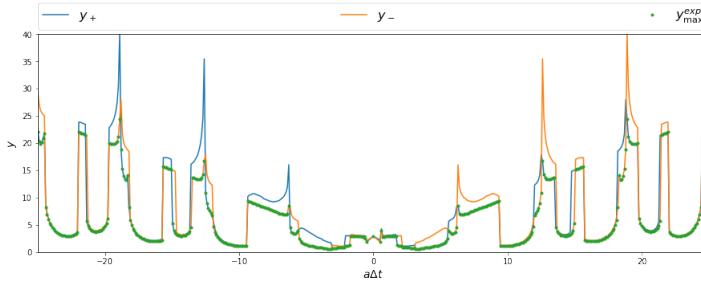


Figure 1.5 – y_{\max}^{\exp} , $|y_+|$ and $|y_-|$ as a function of $a\Delta t$ for the Hochbruck–Ostermann method with $\varepsilon = 10^{-2}$.

exponential integrators considered in this paper.

Finally, we give an illustration of the above comments by running (1.4) ($d = b = 1$ and $v_{\max} = 3$) with a discontinuous initial data to study the impact of high-frequency on the stability of exponential schemes coupled with a centered scheme in v . The initial condition is

$$f_0(x, v) = 1 \text{ if } \sqrt{(x - \pi)^2 + v^2} \leq 1, \text{ and } 0 \text{ elsewhere.}$$

From the stability analysis we know that for each k it holds that $\bar{f}_{k,m}^{n+1} = \phi(z)\bar{f}_{k,m}^n$, where ϕ is the stability function. Our study enables us to estimate the amplification factor $|\phi(z)|$ by $(1 + \varepsilon)$ (uniformly with respect to k) so that $|\bar{f}_{k,m}^{n+1}|^2 \leq (1 + \varepsilon)^2 |\bar{f}_{k,m}^n|^2$. Hence, for a given ε , we consider a time step according to Table 1.2 and run two exponential schemes, namely Hochbruck–Ostermann and Cox–Matthews, for 100 timesteps. After each time step we compute $\|f^n\|_{\ell^2}^2 / \|f^0\|_{\ell^2}^2$. The results can be found in Figure 1.6. First, we have a numerical confirmation that $(1 + \varepsilon)$ is an estimate of the amplification factor (the ratio $\|f^n\|_{\ell^2}^2 / \|f^0\|_{\ell^2}^2$ always lies under the curve $n \mapsto (1 + \varepsilon)^{2n}$). Second, since the number of time steps is fixed, for $\varepsilon = 0.1$ the simulation becomes unstable for $n \geq 20$. As soon as ε is small enough, the simulation is stable.

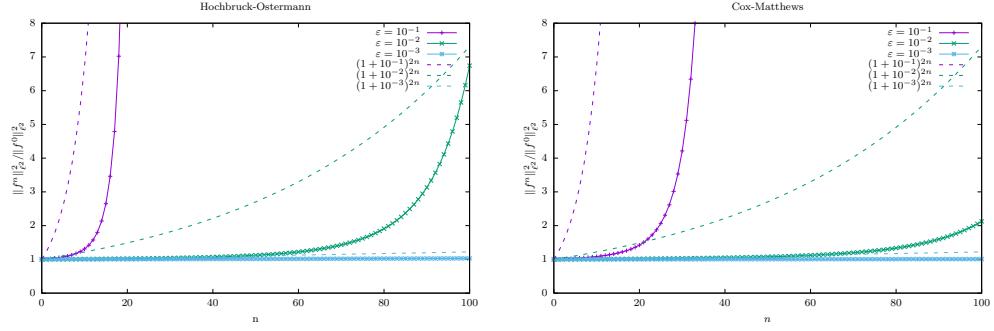


Figure 1.6 – Evolution of $\|f^n\|_{\ell^2}^2 / \|f^0\|_{\ell^2}^2$ and of $(1 + \varepsilon)^{2n}$ as a function of n for different values of ε . Left: Hochbrück–Ostermann scheme. Right: Cox–Matthews scheme.

1.3.4.2 Linearized WENO5 (LW5) scheme.

In the case of a WENO5 approximation of the velocity derivative, we can not easily find a Fourier multiplier because of its nonlinearity. Recent studies about stability of WENO5 [88, 71, 64] considered the linearized version of WENO schemes by freezing the nonlinear weights, so that WENO5 reduces to a high order (linear) upwind scheme called LW5. For this LW5 scheme we can compute the eigenvalues, see equation (1.8), and different time stepping schemes can be studied to determine the stability limit. We consider only Lawson methods here since we found that the exponential schemes we considered (*i.e.* ExpRK22, Krogstad, Cox–Matthews, Hochbrück–Ostermann) lead to unstable results when they are combined with LW5 (even in the weak sense considered in the previous section).

The goal is then to determine the largest non-negative real number $\sigma > 0$ such that the eigenvalues of the upwind scheme LW5 scaled by σ are contained in the domain of stability for the time integrator. Since the eigenvalues of LW5 are not as simple as in the case of the centered scheme, we determine σ numerically. The main idea of the algorithm to obtain an estimation of σ is:

1. The argument φ of the eigenvalues μ_m (normalized by Δv) given by (1.8) are discretized using a fine angular grid $\{\varphi_k\} \subset [-\pi/2, \pi/2]$, since the real part of μ_m is negative due to its diffusive character.
2. A discretized version of the boundary of the stability domain of the underlying Runge–Kutta method is computed.
3. For each discretized eigenvalue, we look for the closest boundary point of the Runge–Kutta stability domain. This enables us to compute the associated stretching factor

$\sigma(\varphi_k)$.

4. Taking the minimum over all the discretized eigenvalues yields $\sigma := \min_k \sigma(\varphi_k)$.

In Figure 1.7 (left), we plot the dependence of σ with respect to the angle $\varphi \in [-\pi/2, \pi/2]$ for Lawson($RK(4, 4)$) coupled with LW5. We also plot (Figure 1.7 (right)) the stability domain of Lawson($RK(4, 4)$), the eigenvalues for LW5 (normalized by Δv) and the eigenvalues for LW5 scaled by σ . The CFL number for some Lawson schemes is shown in Table 1.3. It is interesting to note that the time step size is reduced compared to the centered scheme.

Methods	Lawson($RK(3, 2)$ best)	Lawson($RK(3, 3)$)	Lawson($RK(4, 4)$)
σ	1.344	1.433	1.73

Table 1.3 – CFL number for some Lawson schemes applied to (1.9).

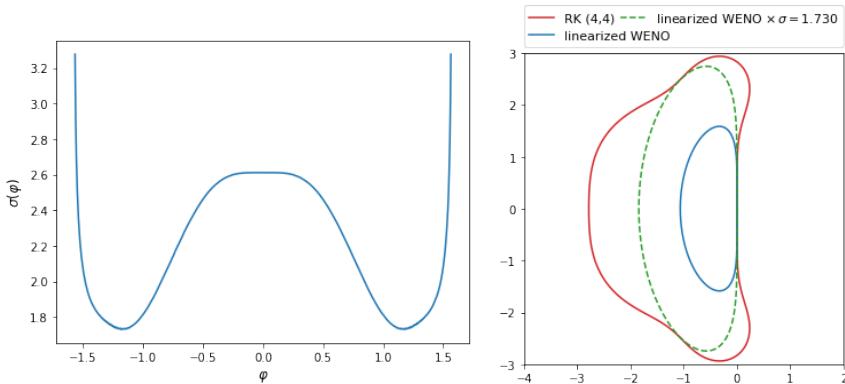


Figure 1.7 – Left: σ as a function of the angle φ . Right: stability domain of Lawson($RK(4, 4)$) (red), eigenvalues for LW5 normalized by Δv (blue) and eigenvalues for LW5 normalized by Δv stretched with factor $\sigma = 1.73$ (dashed green).

1.4 Numerical simulation: Vlasov-Poisson equations

In this section we apply Lawson methods and exponential integrators to the Vlasov-Poisson system. We will see that the linear theory developed in the last section gives a good indication of the stability even for this nonlinear problem. We consider a distribution function $f(t, x, v)$ depending on time $t \geq 0$, space x , with periodic boundary conditions,

and velocity v , which satisfies the Vlasov equation

$$\partial_t f(t, x, v) + v \partial_x f(t, x, v) + E(f)(t, x) \partial_v f(t, x, v) = 0 \quad (1.11)$$

coupled to a Poisson problem for the electric field $E(f)(t, x)$

$$\partial_x E(f)(t, x) = \int_{\mathbb{R}} f(t, x, v) dv - 1. \quad (1.12)$$

We employ a Fourier approximation in space. In velocity we either use a centered discretization

$$\partial_t \hat{f}_{k,j} + v_j ik \hat{f}_{k,j} + \widehat{\left(E \cdot \frac{f_{\cdot, j+1} - f_{\cdot, j-1}}{2\Delta v} \right)}_k = 0$$

or the WENO5 discretization

$$\partial_t \hat{f}_{k,j} + v_j ik \hat{f}_{k,j} + \widehat{\left(E^+ \frac{f_{\cdot, j+1/2}^+ - f_{\cdot, j-1/2}^+}{\Delta v} \right)}_k + \widehat{\left(E^- \frac{f_{\cdot, j+1/2}^- - f_{\cdot, j-1/2}^-}{\Delta v} \right)}_k = 0, \quad (1.13)$$

where $E^+ = \max(E, 0)$, $E^- = \min(E, 0)$ and $f_{j+1/2}^\pm$ denote the numerical fluxes (see Appendix 1.B for more details). Both of these phase space discretizations can be easily cast in the following form

$$\partial_t \hat{f}_{k,j} = -v_j ik \hat{f}_{k,j} + F(f)_{k,j},$$

for an appropriately defined F . We can now apply an exponential integrator or a Lawson scheme. To illustrate this let us consider the exponential Euler method. This gives

$$\hat{f}_{k,j}^{n+1} = \exp(-\Delta t v_j ik) \hat{f}_{k,j}^n + \Delta t \varphi_1(-\Delta t v_j ik) F(f^n)_{k,j}.$$

Since in Fourier space the exponential and φ_1 functions have only scalar arguments, their computation is easy and efficient (*i.e.* no matrix functions have to be computed). Due to the nonlinearity, it is favorable to compute $E \partial_v f$ in real space. This is done efficiently by using the fast Fourier transform. Generalizing this scheme to multiple dimensions in both space and velocity is straightforward.

To apply our theory from the linear analysis to the nonlinear Vlasov-Poisson case, we need a way to compute the CFL condition. Note that the coefficient of the v advection depends on E and thus implicitly on time. We choose the time step for the centered scheme as follows

$$\Delta t_n = \frac{y_{\max} \Delta v}{\|E^n\|_{L^\infty}}, \quad (1.14)$$

whereas for the WENO5 scheme we use the CFL condition computed from its linearized version (LW5)

$$\Delta t_n = \frac{\sigma \Delta v}{\|E^n\|_{L^\infty}}. \quad (1.15)$$

The value $\|E^n\|_{L^\infty}$ is just the maximal value of the electric field at time t_n . The values for y_{\max} and σ are given in Tables 1.1, 1.2, and 1.3 according to the chosen time integrator.

1.4.1 Landau damping test.

We present numerical results for the standard Landau damping test case. The initial condition is given by

$$f_0(x, v) = \frac{1}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} (1 + 0.001 \cos(0.5x)), \quad x \in [0, 4\pi], v \in \mathbb{R}.$$

The numerical parameters are chosen as follows: the number of points in space is $N_x = 81$ whereas the velocity domain is truncated to $[-v_{\max}, v_{\max}]$ with $v_{\max} = 8$ and is discretized with $N_v = 128$ grid points.

Let us remark that for the Landau damping test, the conditions (1.14) and (1.15) allow us to take very large time steps, since $\|E^n\|_{L^\infty} \leq \|E^0\|_{L^\infty} = 2 \cdot 10^{-3}$. Then, we get $\Delta t = C \Delta v / 0.5 \cdot 10^3 = 62.5C$, where C can be either y_{\max} or σ depending on the chosen time integrator. This means that in practice we can choose the time step Δt independently from the mesh. This is clearly a desirable feature of the time integrator.

In Figure 1.8, the time history of the electric energy $\|E^n\|_{L^2}$ (in semi-log scale) using Lawson($RK(4, 4)$)-WENO5 (with two different time steps $\Delta t = 1/8$ and $\Delta t = 1$), and using Hochbruck–Ostermann-CD2 (with $\Delta t = 1$). One can observe that the expected damping rate ($\gamma = -0.153$) is recovered for the three schemes. Although, the accuracy deteriorates for $\Delta t = 1$ Lawson($RK(4, 4)$)-WENO5, the Hochbruck–Ostermann-CD2 scheme gives very good results even with $\Delta t = 1$. We note that all the numerical schemes are clearly stable.

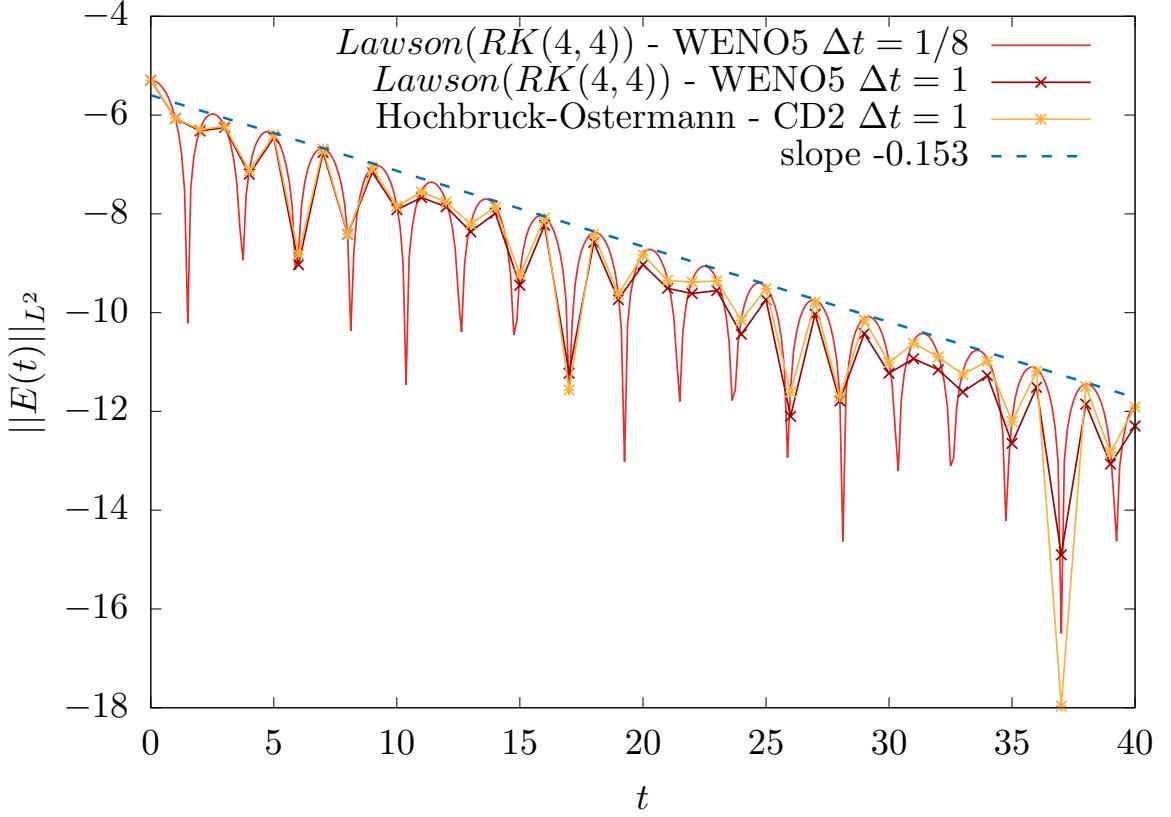


Figure 1.8 – Landau damping test: time history of $\|E(t)\|_{L^2}$ (semi-log scale) obtained with Lawson($RK(4,4)$) and WENO5 (with $\Delta t = 1/8$ and $\Delta t = 1$) and with Hochbruck-Ostermann and CD2 (with $\Delta t = 1$).

1.4.2 Bump on tail test.

Next, we consider the bump on tail test for which the initial condition is

$$f_0(x, v) = \left[\frac{0.9}{\sqrt{2\pi}} e^{-\frac{v^2}{2}} + \frac{0.2}{\sqrt{2\pi}} e^{-2(v-4.5)^2} \right] (1 + 0.04 \cos(0.3x)), \quad x \in [0, 20\pi], v \in \mathbb{R}.$$

The numerical parameters are chosen as follows: the number of points in space is $N_x = 135$ whereas the velocity domain $[-v_{\max}, v_{\max}]$ (with $v_{\max} = 8$) is discretized with $N_v = 256$ grid points. Concerning the time step, as in the Landau damping example, the conditions (1.14) and (1.15) turn out to be very light for Lawson schemes. Indeed, we found $\max_n \|E^n\|_{L^\infty} \approx 0.6$ so that, with the considered velocity grid, the time step has to be smaller than 0.14 in the worst case (Lawson($RK(3,2)$ best) combined with WENO5). To capture correctly the phenomena involved in the bump on tail test, we take the following

time step size

$$\Delta t_n = \min \left(0.1, \frac{C \Delta v}{\|E^n\|_{L^\infty}} \right), \quad (1.16)$$

with $C = y_{\max}$ or σ depending on the chosen scheme. Thus, also in this configuration we are mostly limited by the accuracy and not by the stability constraint.

In Figure 1.9, the full distribution function f is plotted at time $t = 40$ ($\Delta t = 0.05$) for different schemes (exponential or Lawson in time and WENO or centered differences in velocity). One can observe spurious oscillations when the centered differences scheme case is used (second and third rows) whereas the slope limiters of WENO5 (first line) are able to control this phenomena so that extrema are well preserved. This is consistent with what has been observed in the literature.

In Figure 1.10, we plot the time evolution of $(\mathcal{H}^n - \mathcal{H}(0))/\mathcal{H}(0)$, where $\mathcal{H}^n \approx \mathcal{H}(n\Delta t)$ and $\mathcal{H}(t)$ is the total energy defined by

$$\mathcal{H}(t) = \frac{1}{2} \int \int |v|^2 f(t, x, v) dx dv + \frac{1}{2} \int |E|^2(t, x) dx.$$

This quantity is known to be preserved with time at the continuous level. It is thus a useful metric to evaluate and compare the different numerical methods. At this stage, all the used numerical methods are stable and we now look at their accuracy with respect to conservation of energy. We observe that Lawson/centered schemes (referred as 'CD2' in the legend) preserve this quantity well. It is well known (see for example [21]) that centered schemes are better at preserving the total energy compared to upwind schemes. The reason is that upwind schemes introduce numerical diffusion. The exponential integrators that are considered show all very similar behavior with respect to energy conservation. They seem to include less drift than the Lawson methods, but for the time scales considered here their error is larger.

Although being able to choose the time step size independently of the mesh is a desirable feature, it makes checking the sharpness of the CFL estimate derived in the previous section more difficult. To accomplish this, we consider the same parameters as before, except for the phase space mesh which now uses $N_x = 81$ and $N_v = 512$ grid points. Then the maximum time step becomes $\Delta t = \min_n C \Delta v / \|E^n\|_{L^\infty} \approx 0.052C$ (since $\max_n \|E^n\|_{L^\infty} \approx 0.6$ and $\Delta v = 16/512 = 0.03125$). We consider two different time steps: $\Delta t = 0.052C$ (which satisfies the linearized CFL condition) and $\Delta t = 1.4 \times 0.052C$ (which violates the linearized CFL condition). The results are shown in Figure 1.11. There the

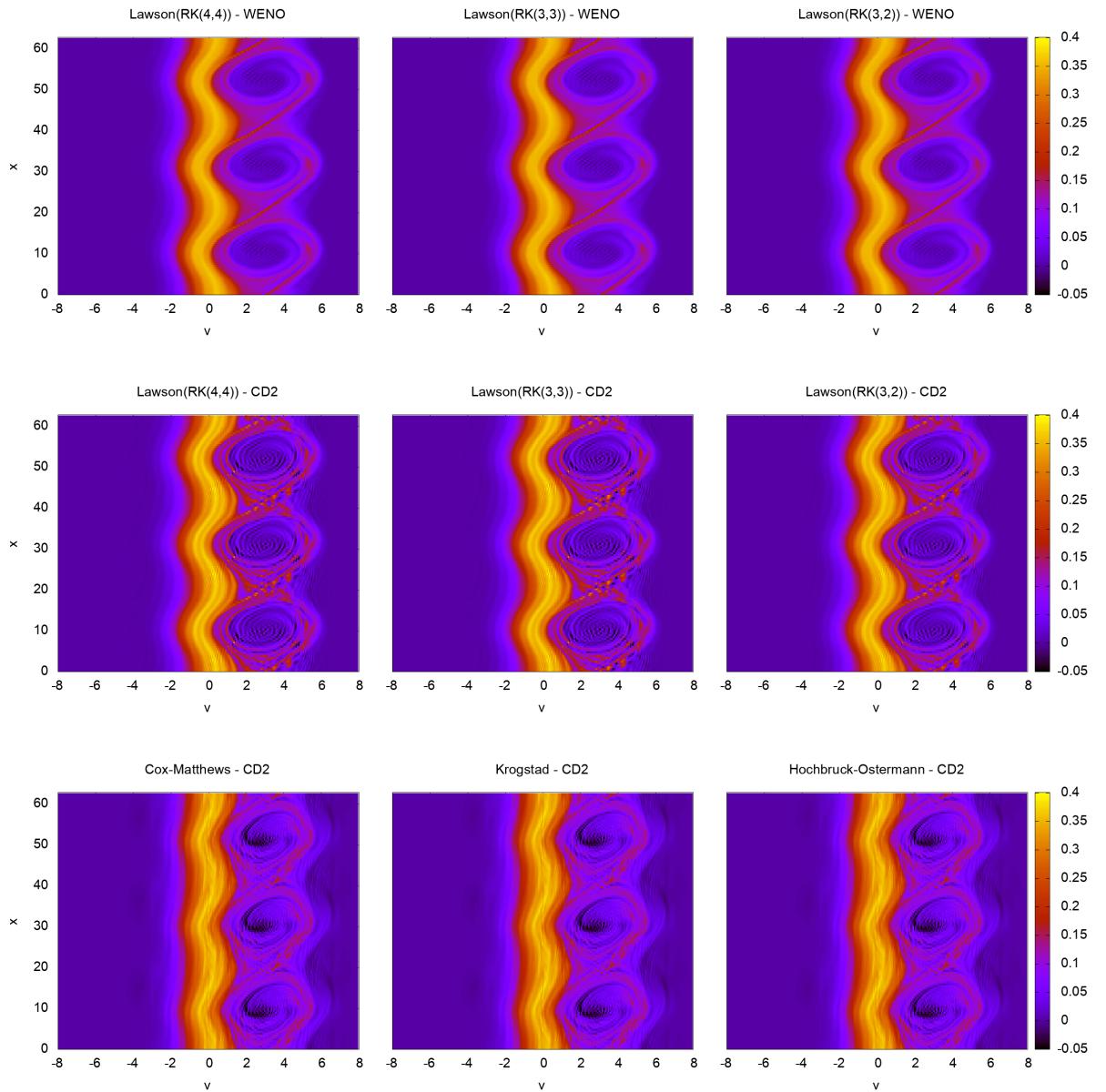


Figure 1.9 – Distribution function at time $t = 40$ as a function of x and v for: (i) Lawson schemes ($RK(4,4)$, $RK(3,3)$, $RK(3,2)$) + WENO5 (first row) ; (ii) Lawson schemes ($RK(4,4)$, $RK(3,3)$, $RK(3,2)$) + centered difference scheme (second row) ; (iii) exponential schemes (Cox-Matthews, Krogstad, Hochbrück–Ostermann) + centered difference scheme (third row).

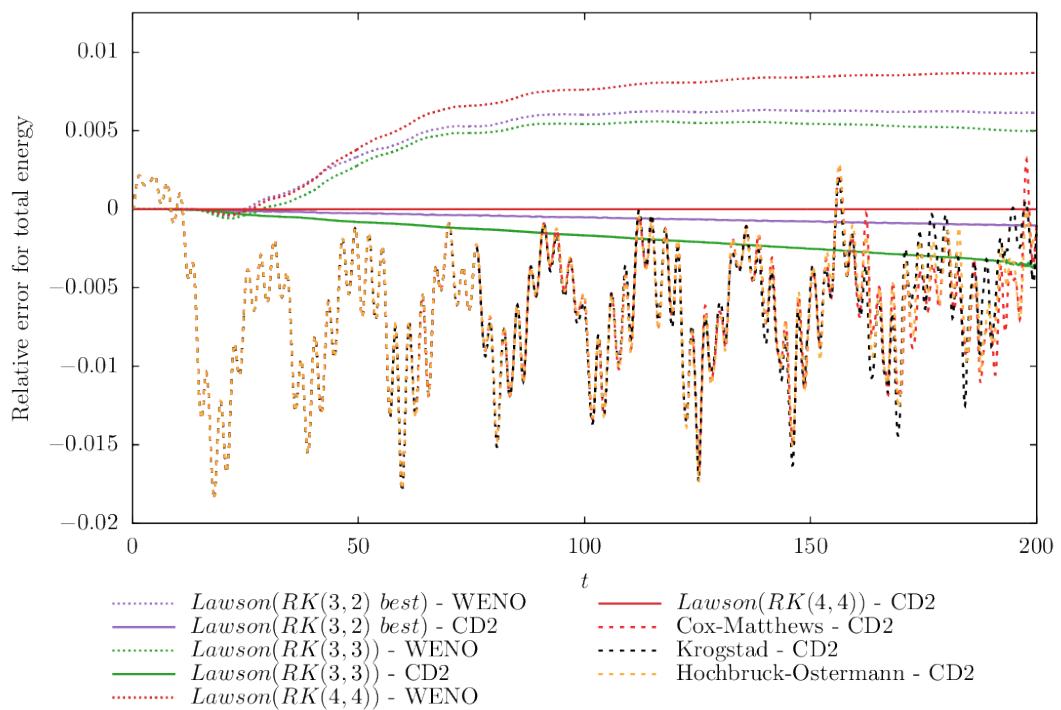


Figure 1.10 – Time evolution of the relative error of the total energy for the different methods.

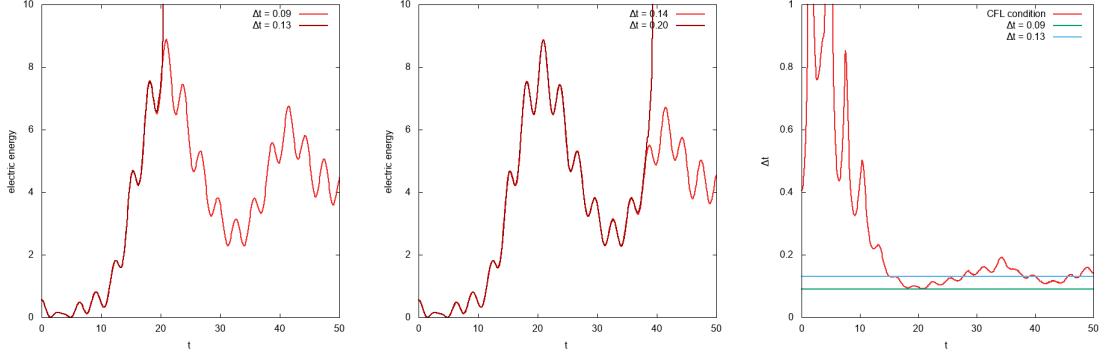


Figure 1.11 – Illustration of the accuracy of the CFL estimate obtained from the linear theory. History of electric energy with Lawson($RK(4,4)$) + WENO5 (left), Lawson($RK(4,4)$) + centered scheme (middle) and history of CFL condition for Lawson($RK(4,4)$) + WENO5 case (right)

Lawson($RK(4,4)$) method has been chosen for the time discretization whereas WENO5 and centered scheme are both considered for the velocity discretization. More specifically,

- for WENO5 we use $C = 1.73$ (obtained from the linearized version LW5) and we compare the results obtained with $\Delta t = 0.09$ (satisfies the CFL condition) and $\Delta t = 0.13$ (does not satisfy the CFL condition).
- for the centered scheme, we use $C = 2\sqrt{2}$ and we compare the results obtained with $\Delta t = 0.14$ (satisfies the CFL condition) with $\Delta t = 0.2$ (does not satisfy the CFL condition).

In Figure 1.11, the time evolution of the electric energy $\|E(t)\|_{L^2}^2$ is displayed for these two velocity discretizations. One can observe for the time step size that satisfies the CFL condition the simulation is stable and gives the expected results, whereas for the choice that violates the CFL condition the simulation blows up. Thus, the results confirm that the CFL condition obtained by the linear theory yields a good prediction for the nonlinear Vlasov–Poisson equation. On the right part of Figure 1.11, the time history of the quantity $C\Delta v/\|E^n\|_{L^\infty}$ is shown (red) together with the time step size considered for the WENO velocity discretization. The choice $\Delta t = 0.13$ (blue line) is larger than the allowed time step size around $t \approx 20$, which explains the numerical instability observed at that point in time.

1.5 Numerical simulation: drift-kinetic equations

In this section we will consider a model motivated by the simulation of strongly magnetized plasmas, such as those found in tokamaks. In this case the dynamics is governed by gyrokinetic equations. Gyrokinetics averages out the fast oscillatory motion of the charged particles around the magnetic field lines. In a simplified slab geometry, gyrokinetic models reduce to the drift-kinetic equation. In this case the unknown f depends on three cylindrical spatial coordinates (r, θ, z) and one velocity direction v . This model is composed of a guiding-center dynamics in the plane orthogonal to the magnetic field lines and of a Vlasov type dynamics in the direction parallel (to the magnetic field lines). In addition to its relevance in physics, it is also a good test case for stressing exponential methods. The latter is due to the fact that after some time the nonlinearity can become strong enough such that the time step size is dictated by stability constraints (especially for high order methods).

Our goal in this section is to find a numerical approximation of $f = f(t, r, \theta, z, v)$ satisfying the following 4D slab drift-kinetic equation (see [40])

$$\partial_t f - \frac{\partial_\theta \phi}{r} \partial_r f + \frac{\partial_r \phi}{r} \partial_\theta f + v \partial_z f - \partial_z \phi \partial_v f = 0, \quad (1.17)$$

for $(r, \theta, z, v) \in \Omega \times [0, L] \times \mathbb{R}$, $\Omega = [r_{\min}, r_{\max}] \times [0, 2\pi]$. The self-consistent potential $\phi = \phi(r, \theta, z)$ is determined by solving the quasi neutrality equation

$$\begin{aligned} & - \left[\partial_r^2 \phi + \left(\frac{1}{r} + \frac{\partial_r n_0(r)}{n_0(r)} \right) \partial_r \phi + \frac{1}{r^2} \partial_\theta^2 \phi \right] + \frac{1}{T_e(r)} (\phi - \langle \phi \rangle) \\ & = \frac{1}{n_0(r)} \int_{\mathbb{R}} f dv - 1, \end{aligned} \quad (1.18)$$

where $\langle \phi \rangle = \frac{1}{L} \int_0^L \phi(r, \theta, z) dz$ and the functions n_0 and T_e depend only on r and are given analytically.

In many situations the $v \partial_z f$ term yields the most restrictive CFL condition. In this setting exponential methods can be very successful as they remove the most stringent CFL condition, while still treating the remaining terms explicitly (which computationally is relatively cheap). The φ functions can be computed in Fourier space (as has been discussed in some detail for the Vlasov–Poisson system in the previous section) or using a semi-Lagrangian approach. Exponential integrators for the drift-kinetic model have been

proposed in [20]. They compare favorably to splitting schemes and have the advantage that they can be more easily adapted to different models. In [20] only a second order exponential integrator and the fourth order Cox–Matthews scheme have been considered. Due to the investigations in the present paper we now understand that this is not an ideal choice. Thus, the purpose of this section is to demonstrate that Lawson methods can be more efficient and to further corroborate the results obtained in the previous sections. The difference in stability for Lawson schemes and exponential integrators will be very evident in the numerical simulations that are presented.

1.5.1 Numerical discretization

First, we remark that z is a periodic variable which motivates us to consider the Fourier transform in this direction. The corresponding frequencies are denoted by k . Equation (1.17) then becomes

$$\partial_t \hat{f}_k - \partial_r \left(\widehat{\frac{\partial_\theta \phi}{r} f} \right)_k + \partial_\theta \left(\widehat{\frac{\partial_r \phi}{r} f} \right)_k + vik \hat{f}_k - \partial_v (\widehat{\partial_z \phi f})_k = 0,$$

Setting $F(t, f) = \partial_r \left(\widehat{\frac{\partial_\theta \phi}{r} f} \right) - \partial_\theta \left(\widehat{\frac{\partial_r \phi}{r} f} \right) + \partial_v (\widehat{\partial_z \phi f})$, this equation can be written as

$$\partial_t \hat{f} = -vik \hat{f} + F(t, f).$$

This is now precisely in the form to which we can apply an exponential method. In addition, computing the required matrix functions is very efficient as all the frequencies decouple (see the corresponding discussion in section 1.4).

To complete the numerical scheme, one has to detail the phase space approximation. As in [20] we will use Arakawa’s method to approximate the derivatives needed to compute F . Arakawa’s method is a centered difference scheme that conserves three invariants. More details can be found in [20].

1.5.2 Numerical results

In this section, we detail the physical parameters of the considered test case. The set up is identical to [20] (see also [14, 24]). The initial value is given by

$$f(t=0, r, \theta, z, v) = f_{\text{eq}}(r, v) \left[1 + \epsilon \exp\left(-\frac{(r - r_p)^2}{\delta r}\right) \cos\left(\frac{2\pi n}{L}z + m\theta\right) \right],$$

where the equilibrium distribution is given by

$$f_{\text{eq}}(r, v) = \frac{n_0(r) \exp\left(-\frac{v^2}{2T_i(r)}\right)}{(2\pi T_i(r))^{1/2}}. \quad (1.19)$$

The radial profiles T_i , T_e , and n_0 have the analytic expressions

$$\mathcal{P}(r) = C_{\mathcal{P}} \exp\left(-\kappa_{\mathcal{P}} \delta r_{\mathcal{P}} \tanh\left(\frac{r - r_p}{\delta r_{\mathcal{P}}}\right)\right), \quad \mathcal{P} \in \{T_i, T_e, n_0\}$$

with the constants defined as follows

$$C_{T_i} = C_{T_e} = 1, \quad C_{n_0} = \frac{r_{\max} - r_{\min}}{\int_{r_{\min}}^{r_{\max}} \exp\left(-\kappa_{n_0} \delta r_{n_0} \tanh\left(\frac{r - r_p}{\delta r_{n_0}}\right)\right) dr}.$$

Finally, we consider the parameters of [14] (MEDIUM case)

$$\begin{aligned} r_{\min} &= 0.1, \quad r_{\max} = 14.5, \\ \kappa_{n_0} &= 0.055, \quad \kappa_{T_i} = \kappa_{T_e} = 0.27586, \\ \delta r_{T_i} &= \delta r_{T_e} = \frac{\delta r_{n_0}}{2} = 1.45, \quad \epsilon = 10^{-6}, \quad n = 1, \quad m = 5, \\ L &= 1506.759067, \quad r_p = \frac{r_{\min} + r_{\max}}{2}, \quad \delta r = \frac{4\delta r_{n_0}}{\delta r_{T_i}}. \end{aligned}$$

and use a v -range of $v \in [-7.32, 7.32]$.

We consider two configurations. A direct formulation, where the boundary conditions are given by

$$f(r_{\min}, \theta, z, v) = f_{\text{eq}}(r_{\min}, v) \quad f(r_{\max}, \theta, z, v) = f_{\text{eq}}(r_{\max}, v).$$

Note that these are not homogeneous Dirichlet boundary conditions. It is well known (and

supported by [20]) that the Arakawa scheme works better for homogeneous boundary conditions. In addition to the direct formulation, we therefore also introduce a so-called perturbation formulation (see also [24, 59]). First, we note that the equilibrium function f_{eq} defined in (1.19) is a steady state for our problem. We therefore divide f into

$$f(t, r, \theta, v) = f_{eq}(r, v) + \delta f(t, r, \theta, v).$$

With this formulation, our problem (1.17) becomes

$$\partial_t \delta f + \frac{E_\theta}{r} \partial_r (f_{eq} + \delta f) - \frac{E_r}{r} \partial_\theta \delta f + v \partial_z \delta f + E_z \partial_v (f_{eq} + \delta f) = 0,$$

where $E_\theta = -\partial_\theta \phi$, $E_r = -\partial_r \phi$ and $E_z = -\partial_z \phi$. Expanding the various terms we obtain

$$\partial_t \delta f + \frac{E_\theta}{r} \partial_r \delta f - \frac{E_r}{r} \partial_\theta \delta f + v \partial_z \delta f + E_z \partial_v \delta f + \frac{E_\theta}{r} \partial_r f_{eq} + E_z \partial_v f_{eq} = 0$$

which can be written as

$$\partial_t \delta f + v \partial_z \delta f - F(\delta f) + \frac{E_\theta}{r} \partial_r f_{eq} + E_z \partial_v f_{eq} = 0.$$

Note that the equation is very similar to equation (1.17). We, however, have obtained two additional source terms, which depend on the equilibrium distribution f_{eq} as well as on the electric field. Furthermore, the right hand side of the quasi-neutrality equation (1.18) becomes

$$\frac{1}{n_0} \int f_{eq} dv + \frac{1}{n_0} \int \delta f dv - 1 = \frac{1}{n_0} \int \delta f dv.$$

Due to the similarity of the direct formulation and the perturbation formulation, the same code can be used for both by simply exchanging the right hand side of the quasi-neutrality equation, changing the boundary conditions, and adding the appropriate source terms. Thus, to implement the exponential integrator we consider the following equation

$$\partial_t \delta f + v \partial_z \delta f = F_{pert}(\delta f),$$

with

$$F_{pert}(\delta f) = F(\delta f) - \frac{E_\theta}{r} \partial_r f_{eq} - E_z \partial_v f_{eq},$$

and proceed as before (with F replaced by F_{pert}). The space discretization of the source terms can be done either analytically or using a numerical approximation. In our imple-

mentation we have used standard centered differences. The Arakawa scheme that is used to discretize $F(\delta f)$ now employs homogeneous Dirichlet boundary conditions for δf in the r -direction.

We have seen in section 1.4 that for the Vlasov–Poisson equation we can derive a constraint on the time step size which ensures stability. For Lawson methods this also gives a good estimate in practice. However, for exponential integrators the situation is far more complicated, see the discussion in section 1.3. Thus, a natural question that arises is how large time steps can we take in practice. To do that we will employ an adaptive step size controller that uses Richardson extrapolation to obtain an error estimate. By denoting a time step as follows $f^{n+1} = \varphi_{\Delta t_n}(f^n)$ and considering $\tilde{f}^{n+1} = \varphi_{\Delta t_n/2} \circ \varphi_{\Delta t_n/2}(f^n)$ we can construct the Richardson extrapolated numerical solution of a method of order p as follows $f_R^{n+1} = (2^{p+1}\tilde{f}^{n+1} - f^{n+1})/(2^{p+1} - 1)$, which turns out to be an approximation of order $(p + 1)$ of the exact solution. Then, it is possible to determine an estimate of the local error e_{n+1} of the time integrator through the following expression

$$e_{n+1} = \|f_R^{n+1} - f^{n+1}\|_{L^\infty} + \mathcal{O}(\Delta t_n^{p+2}),$$

where the L^∞ norm is considered in the r, θ, z, v variables. If the estimate for the error e_{n+1} is larger than a specified tol we reject the step and start again from time t_n . Otherwise, the step is accepted and we proceed with the time integration. In either case we then determine the new step size Δt_{new} such that the local error is smaller than the tolerance. That is, we choose

$$\Delta t_{new} = s\Delta t_n \left(\frac{\text{tol}}{e_{n+1}} \right)^{1/(p+1)}, \quad (1.20)$$

where tol is the prescribed tolerance, p is the order of the method, and $s = 0.8$ is a safety factor. This process is very well established in the literature and we refer the interested reader to [41, 41, 76, 77, 29]. Other strategies can also be considered such as embedded Lawson or exponential methods (see [43, 2]). Such methods may be more efficient but we restrict ourselves here to the strategy based on Richardson extrapolation since it can be applied to any time integrator.

An interesting property of this adaptive step size controller is that it forces the time step size to satisfy the stability constraint of the numerical method. This is perhaps surprising at first sight since the scheme only controls the local error. However, numerical instability are characterized by error amplification as integration proceeds in time. Thus,

a single step can violate the stability constraint, but later on the error amplification increases the local error in such a way that the adaptive step size controller is forced to reduce the time step size. Thus, the controller ensures that we obtain a stable numerical simulation for which the local error is below the specified tolerance.

This procedure allows us to perform a fair comparison between Lawson methods and exponential integrators. Since we are mainly interested in the stability of the methods and, particularly in the nonlinear regime, prescribing a stringent tolerance is infeasible in any case, we will choose a relatively large tolerance for our simulation ($\text{tol} = 10^{-2}$ for the perturbation formulation). To avoid the problem of too large time steps at the beginning of the simulation, where accuracy and not stability dictates the time step, we limit the maximal step size to $\Delta t = 11$ (coarse) and $\Delta t = 10$ (fine) for second order methods, $\Delta t = 30$ for third order methods and Lawson($RK(3, 2)$ best), and $\Delta t = 40$ for fourth order methods.

To evaluate the performances of the different time integrators, we consider the time evolution of the electric energy defined by

$$\mathcal{E}(t) = \left(\int_0^L \int_0^{2\pi} \phi^2(t, r_p, \theta, z) d\theta dz \right)^{1/2}, \quad \text{with } r_p = \frac{r_{\min} + r_{\max}}{2},$$

as well as the time evolution of the total mass and the total energy

$$\begin{aligned} \mathcal{M}(t) &= \int_{r_{\min}}^{r_{\max}} \int_0^L \int_0^{2\pi} \int_{\mathbb{R}} f(t, r, \theta, z, v) dv d\theta dz dr, \\ \mathcal{N}(t) &= \int_{r_{\min}}^{r_{\max}} \int_0^L \int_0^{2\pi} \int_{\mathbb{R}} \frac{v^2}{2} f(t, r, \theta, z, v) dv d\theta dz dr \\ &\quad + \int_{r_{\min}}^{r_{\max}} \int_0^L \int_0^{2\pi} \int_{\mathbb{R}} f(t, r, \theta, z, v) \phi(t, r, \theta, z) dv d\theta dz dr. \end{aligned}$$

The numerical results for the perturbation formulation are given in Figure 1.12. There the time history of the electric energy and the time step size as of function of time for two different discretizations in phase space, $32 \times 32 \times 32 \times 64$ and $64 \times 64 \times 64 \times 128$ grid points, are shown. We first see that all the time integrators agree very well; that is, we see an initial exponential growth (the rate is in good agreement with the linear theory; see, for example, [14]) in the electric energy. This phase is followed by saturation at very similar levels for all numerical methods used. We also observe that all exponential integrators, except the method of Krogstad, are forced to reduce their step size after

time $t \approx 5000$ for the fine, *i.e.* $64 \times 64 \times 64 \times 128$ grid points, case. This is particularly drastic for ExpRK33 and the Cox–Matthews method which suffer from stability issues even for the coarse discretization. In general, for the finer space discretization (see the right plot in Figure 1.12), the problem becomes significantly more severe. It is also worth mentioning that ExpRK33 leads to unstable results in spite of the step size controller, which clearly highlights the unstable nature of that integrator. Neither of the Lawson schemes have similar issues and the size of the stability domain on the imaginary axis gives a good indication of the relative time steps this methods can take. We also note that at later times Lawson schemes are able to take significantly larger time steps compared to exponential integrators. The only exponential integrator that performs well in this regime is the method of Krogstad. Thus, the numerical results agree well with what we would expect based on the theoretical analysis.

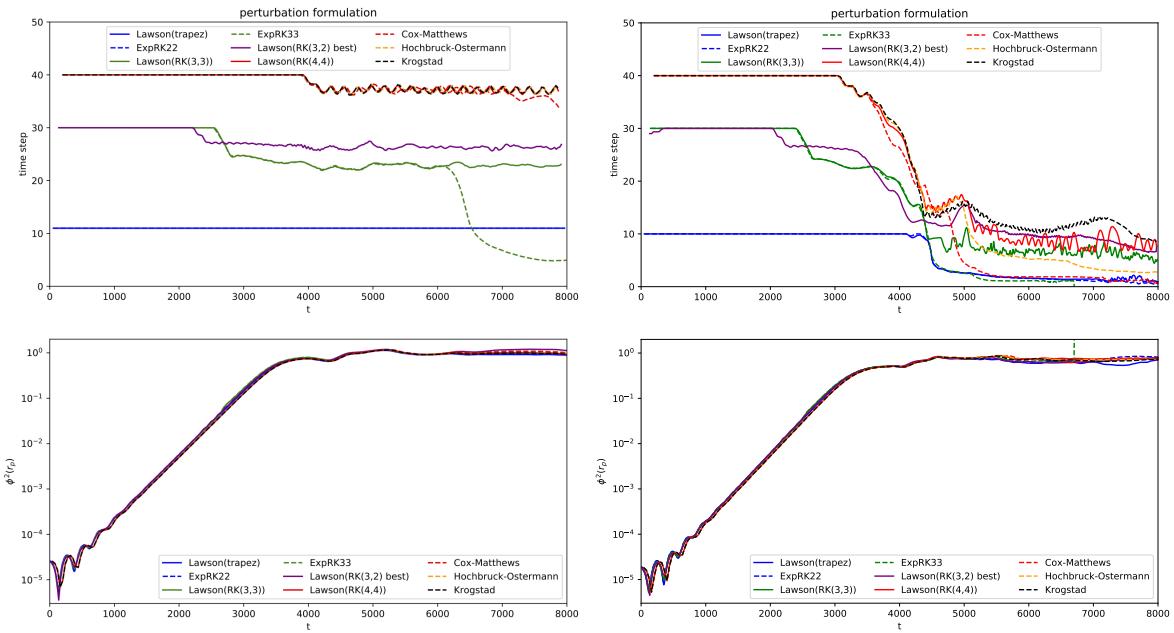


Figure 1.12 – Numerical simulation for a number of Lawson methods and exponential integrators for the drift-kinetic model (perturbation formulation). The upper plots show the time step size as a function of time. The lower plots show the time evolution of the electric energy. The configuration on the left uses $32 \times 32 \times 32 \times 64$ grid points and the configuration on the right uses $64 \times 64 \times 64 \times 128$ grid points.

The corresponding numerical results using the direct formulation are shown in Figure 1.13. The situation for the direct formulation is very similar to the perturbation formulation, even if one can observe that the time steps are slightly larger than in the perturbation

formulation. One explanation comes from the fact that the relative error computed in the perturbation case involves the norm of δf which can be quite different from the norm of f so that equation (1.20) leads to different value of the time step even if the accuracy of the solution is the same.

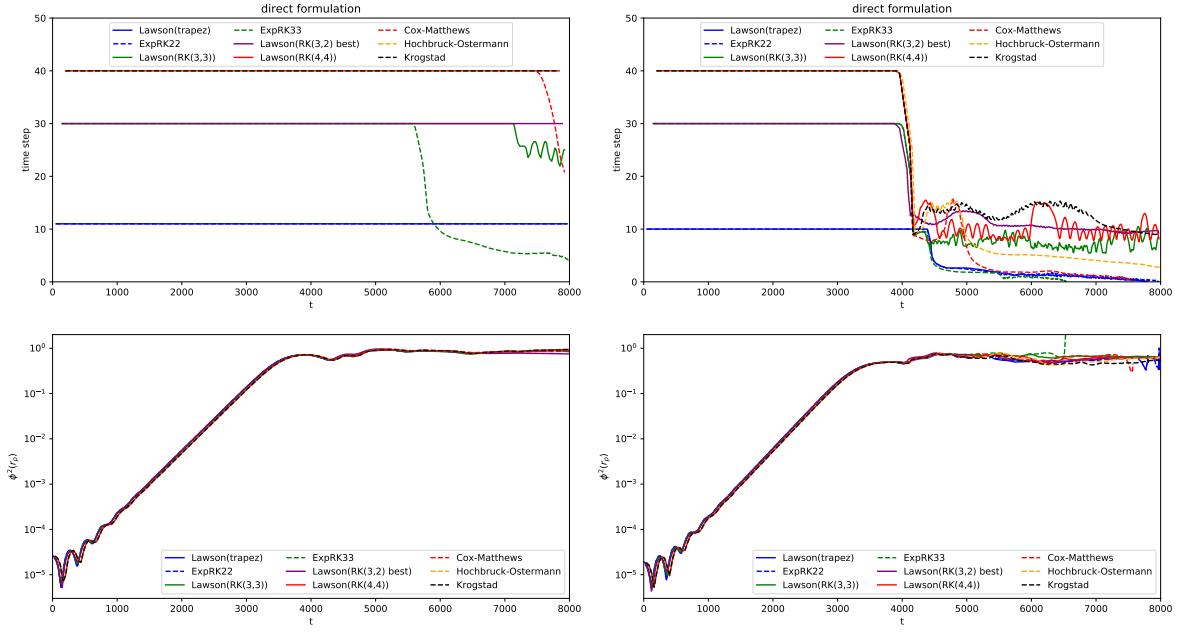


Figure 1.13 – Numerical simulation for a number of Lawson methods and exponential integrators for the drift-kinetic model (direct formulation). The upper plots show the time step size as a function of time. The lower plots show the time evolution of the electric energy. The configuration on the left uses $32 \times 32 \times 32 \times 64$ grid points and the configuration on the right uses $64 \times 64 \times 64 \times 128$ grid points.

In Figure 1.14, the time history of the relative error of the total mass and of the total energy are displayed with the phase space discretization $64 \times 64 \times 64 \times 128$ and using Lawson($RK(4,4)$) and Cox–Matthews time integrators (the perturbation formulation is used here). Since mass is a linear invariant it is preserved, up to machine precision, by the exponential integrator, see [36]. We also observe good conservation of energy even in the nonlinear phase, which confirms the excellent behavior of the methods.

Finally, we show slices of the distribution function and the density at different times. The simulation in Figure 1.15 is conducted with the Lawson($RK(4,4)$) scheme and the simulation in Figure 1.16 with the Cox–Matthews scheme (both with the perturbation formulation). In both cases the configuration of Figure 1.12 and the fine space resolution has been employed. As comparison, a reference solution computed with the Lawson($RK(4,4)$)

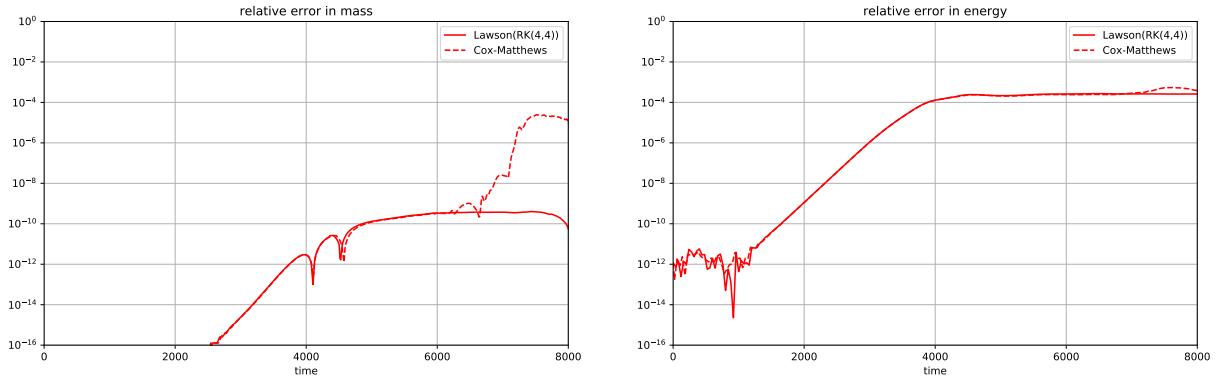


Figure 1.14 – Numerical simulation for Lawson(RK(4,4)) and the Cox–Matthews method for the drift-kinetic model (perturbation formulation). Left: time history of the error in total mass. Right: time history of the error in total energy. The configuration uses $64 \times 64 \times 64 \times 128$ grid points.

scheme and a step size controller that keeps the error below 10^{-5} per unit time step is shown in Figure 1.17. We remark that all simulations show good agreement: the $m = 5$ modes in the θ direction are recovered and after initial growth of the unstable mode, we can observe a shearing of the structures and the appearance of small scale structures which are typical for the nonlinear phase.

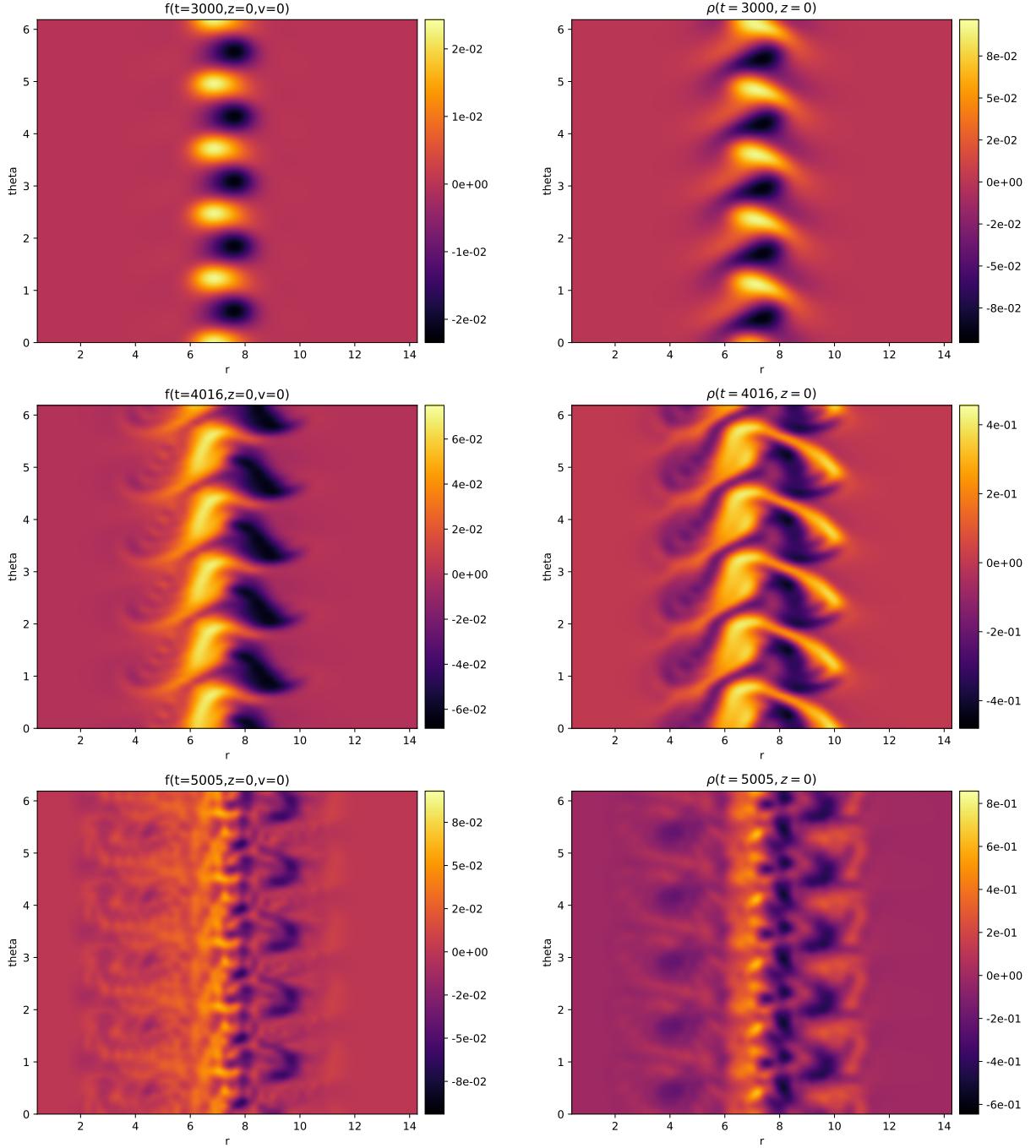


Figure 1.15 – A slices at $(z, v) = (0, 0)$ of the distribution function (on the left) and a slice at $z = 0$ of the density (on the right) are shown for times $t = 3000, 4000$, and 5000 . The Lawson($RK(4, 4)$) scheme, in the configuration described in section 1.5.2, with $64 \times 64 \times 64 \times 128$ grid points is used.

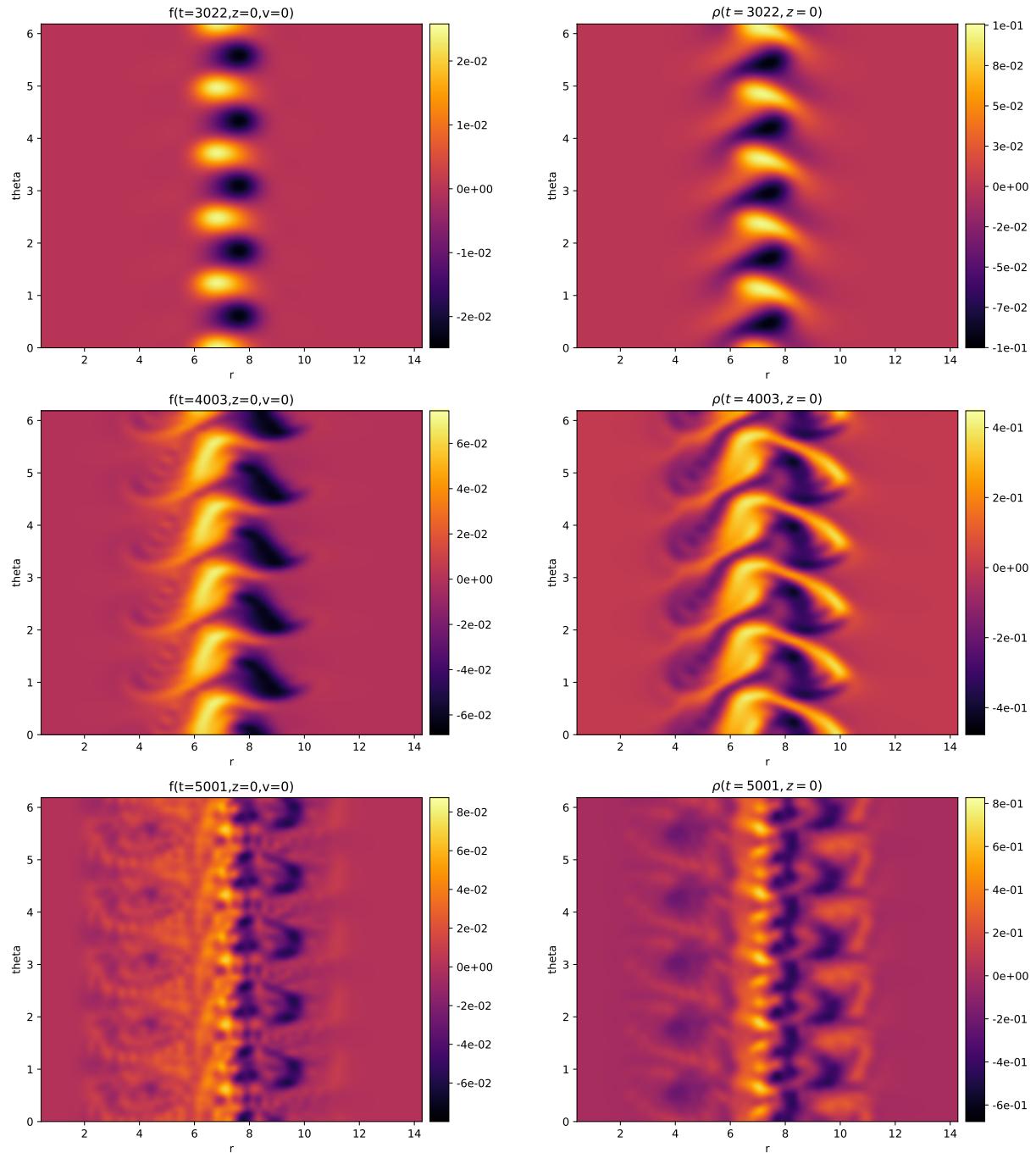


Figure 1.16 – A slices at $(z, v) = (0, 0)$ of the distribution function (on the left) and a slice at $z = 0$ of the density (on the right) are shown for times $t = 3000, 4000$, and 5000 . The Cox–Matthews scheme, in the configuration described in section 1.5.2, with $64 \times 64 \times 64 \times 128$ grid points is used.

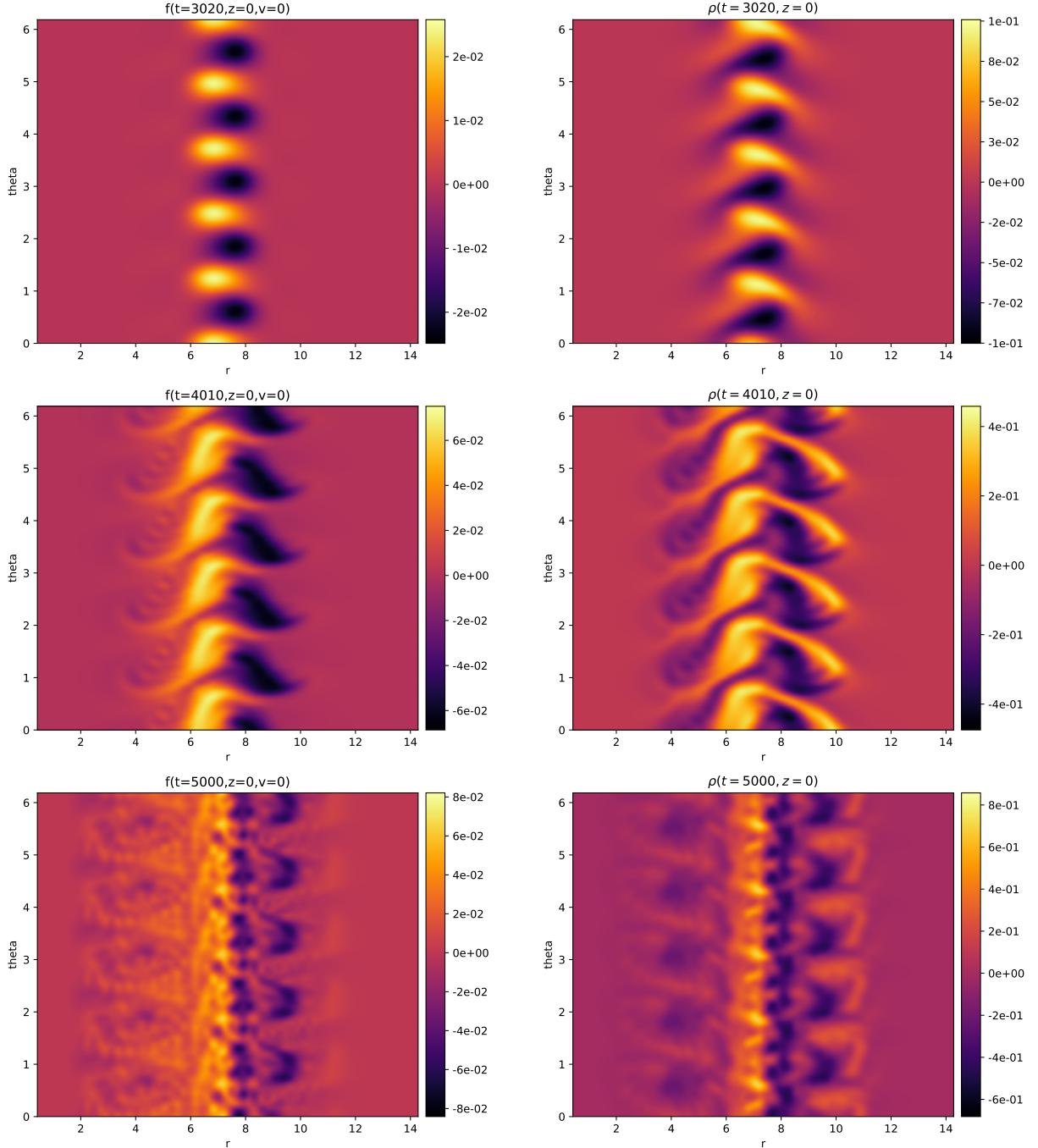


Figure 1.17 – A slices at $(z, v) = (0, 0)$ of the distribution function (on the left) and a slice at $z = 0$ of the density (on the right) are shown for times $t = 3000, 4000$, and 5000 . The Lawson($RK(4, 4)$) scheme with a tolerance of 10^{-5} per unit step and $64 \times 64 \times 64 \times 128$ grid points is used.

Acknowledgement

We would like to thank David C. Seal (U.S. Naval Academy) and Sigal Gottlieb (University of Massachusetts, Dartmouth) for the helpful discussion.

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014- 2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. The work has been supported by the French Federation for Magnetic Fusion Studies (FR-FCM) and by the Austrian Science Fund (FWF): project number P 32143-N32.

APPENDICES

1.A Butcher tableaus

In this section, we write down the different numerical methods used in this work. As in section 1.2, we consider the following equation

$$\dot{u} = Au + F(u),$$

where A is a matrix and F a general nonlinear function of u . The Butcher tableaus for the Lawson integrators used in the main text are stated in this section. A Lawson method is uniquely determined by the underlying (explicit) Runge–Kutta methods and can be written as follows

$$u^{(\ell)} = e^{c_\ell \Delta t A} u^n + \Delta t \sum_{j=1}^s a_{\ell,j} e^{-(c_j - c_\ell) \Delta t A} F(u^{(j)}),$$

$$u^{n+1} = e^{\Delta t A} u^n + \Delta t \sum_{j=1}^s b_j e^{(1-c_j) \Delta t A} F(u^{(j)}),$$

where the coefficients $a_{\ell,j}$ and b_j are given by the Butcher tableaus. The Butcher tableaus for the $RK(2,2)$ best, $RK(3,3)$ (the classic method of order 3), and $RK(4,4)$ (the classic method of order 4) are shown in Table 1.4.

0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	0
$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	-1	2	$\frac{1}{2}$
$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$	$\frac{1}{2}$
1	0	0	1	0	0	1
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$		

Table 1.4 – Butcher tableaus for $RK(4,4)$ (left), $RK(3,3)$ (middle) and $RK(3,2)$ best (right).

A general exponential integrator can be written as

$$u^{(\ell)} = u^n + \Delta t \sum_{j=1}^s a_{\ell,j}(\Delta t A) (F(u^{(j)}) + Au^n)$$

$$u^{n+1} = u^n + \Delta t \sum_{j=1}^s b_j(\Delta t A) (F(u^{(j)}) + Au^n),$$

where the coefficients $a_{\ell,j}(\Delta t A)$ and $b_j(\Delta t A)$ can be written as a linear combination of φ_ℓ and $\varphi_{\ell,j}$ (see [48])

$$\varphi_\ell(z) = \frac{e^z - \sum_{k=0}^{\ell-1} \frac{1}{k!} z^k}{z^\ell},$$

and we use the notations $\varphi_\ell := \varphi_\ell(\Delta t A)$ and $\varphi_{\ell,j} := \varphi_\ell(c_j \Delta t A)$. The coefficients are collected in tableau form, see Table 1.5. The Butcher tableaus for the exponential integrators

0			
c_2	$a_{2,1}$		
\vdots	\vdots \ddots		
c_s	a_{s1}	\cdots	$a_{s,s-1}$
	b_1	\cdots	b_{s-1} b_s

Table 1.5 – Butcher tableau of a general exponential integrators

used in the main text are given in Tables 1.6, 1.7, 1.8 and 1.9.

0			
1	$\varphi_{1,2}$		
	$\varphi_1 - \varphi_2$	φ_2	

Table 1.6 – Butcher tableau of ExpRK22.

0			
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2}$		
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,3} - \varphi_{2,3}$	$\varphi_{2,3}$	
1	$\varphi_{1,4} - 2\varphi_{2,4}$	0	$2\varphi_{2,4}$
	$\varphi_1 - 3\varphi_2 + 4\varphi_3$	$2\varphi_2 - 4\varphi_3$	$2\varphi_2 - 4\varphi_3$ $-\varphi_2 + 4\varphi_3$

Table 1.7 – Butcher tableau of the Krogstad method.

0					
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2}$				
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,3} - \varphi_{2,3}$	$\varphi_{2,3}$			
1	$\varphi_{1,4} - 2\varphi_{2,4}$	$\varphi_{2,4}$	$\varphi_{2,4}$		
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,5} - 2a_{5,2} - a_{5,4}$	$a_{5,2}$	$a_{5,2}$	$\frac{1}{4}\varphi_{2,5} - a_{5,2}$	
	$\varphi_1 - 3\varphi_2 + 4\varphi_3$	0	0	$-\varphi_2 + 4\varphi_3$	$2\varphi_2 - 8\varphi_3$

$$a_{5,2} = \frac{1}{2}\varphi_{2,5} - \varphi_{3,4} + \frac{1}{4}\varphi_{2,4} - \frac{1}{2}\varphi_{3,5}$$

$$a_{5,4} = \frac{1}{4}\varphi_{2,5} - a_{5,2}$$

Table 1.8 – Butcher tableau of the Hochbruck–Ostermann method.

0					
$\frac{1}{2}$	$\frac{1}{2}\varphi_{1,2}$				
$\frac{1}{2}$	0	$\frac{1}{2}\varphi_{1,3}$			
1	$\frac{1}{2}\varphi_{1,3}(\varphi_{0,3} - 1)$	0	$\varphi_{1,3}$		
	$\varphi_1 - 3\varphi_2 + 4\varphi_3$	$2\varphi_2 - 4\varphi_3$	$2\varphi_2 - 4\varphi_3$	$4\varphi_3 - \varphi_2$	

Table 1.9 – Butcher tableau of the Cox–Matthews method.

1.B WENO5 scheme

The different ingredients of the WENO5 scheme used in (1.13) are detailed here. First the fluxes are given by

$$\begin{aligned} f_{j+\frac{1}{2}}^+ &= w_0^+ \left(\frac{2}{6}f_{j-2} - \frac{7}{6}f_{j-1} + \frac{11}{6}f_j \right) + w_1^+ \left(-\frac{1}{6}f_{j-1} + \frac{5}{6}f_j + \frac{2}{6}f_{j+1} \right) \\ &\quad + w_2^+ \left(\frac{2}{6}f_j + \frac{5}{6}f_{j+1} - \frac{1}{6}f_{j+2} \right) \end{aligned}$$

and

$$\begin{aligned} f_{j+\frac{1}{2}}^- &= w_2^- \left(-\frac{1}{6}f_{j-1} + \frac{5}{6}f_j + \frac{2}{6}f_{j+1} \right) + w_1^- \left(\frac{2}{6}f_j + \frac{5}{6}f_{j+1} - \frac{1}{6}f_{j+2} \right) \\ &\quad + w_0^- \left(\frac{11}{6}f_{j+1} - \frac{7}{6}f_{j+2} + \frac{2}{6}f_{j+3} \right). \end{aligned}$$

The weights are defined through the β coefficients

$$\begin{aligned}\beta_0^+ &= \frac{13}{12} \underbrace{(f_{j-2}^+ - 2f_{j-1}^+ + f_j^+)^2}_{\Delta x^2(f_j'' + \mathcal{O}(\Delta x))} + \frac{1}{4} \underbrace{(f_{j-2}^+ - 4f_{j-1}^+ + 3f_j^+)^2}_{2\Delta f'_j + \mathcal{O}(\Delta x^2))} \\ \beta_1^+ &= \frac{13}{12} \underbrace{(f_{j-1}^+ - 2f_j^+ + f_{j+1}^+)^2}_{\Delta x^2(f_j'' + \mathcal{O}(\Delta x^2))} + \frac{1}{4} \underbrace{(f_{j-1}^+ - f_{j+1}^+)^2}_{2\Delta x f'_j + \mathcal{O}(\Delta x^2))} \\ \beta_2^+ &= \frac{13}{12} \underbrace{(f_j^+ - 2f_{j+1}^+ + f_{j+2}^+)^2}_{\Delta x^2(f_j'' + \mathcal{O}(\Delta x))} + \frac{1}{4} \underbrace{(3f_j^+ - 4f_{j+1}^+ + f_{j+2}^+)^2}_{-2\Delta f'_j + \mathcal{O}(\Delta x^2))}\end{aligned}$$

with

$$\begin{aligned}\beta_0^- &= \frac{13}{12} (f_{j+1}^- - 2f_{j+2}^- + f_{j+3}^-)^2 + \frac{1}{4} (3f_{j+1}^- - 4f_{j+2}^- + f_{j+3}^-)^2 \\ \beta_1^- &= \frac{13}{12} (f_j^- - 2f_{j+1}^- + f_{j+2}^-)^2 + \frac{1}{4} (f_j^- - f_{j+2}^-)^2 \\ \beta_2^- &= \frac{13}{12} (f_{j-1}^- - 2f_j^- + f_{j+1}^-)^2 + \frac{1}{4} (f_{j-1}^- - 4f_j^- + 3f_{j+1}^-)^2\end{aligned}$$

Then, the normalized weights are

$$\alpha_i^\pm = \frac{\gamma_i}{(\varepsilon + \beta_i^\pm)^2}, \quad i = 0, 1, 2,$$

where ε is a numerical regularization parameter set to 10^{-6} and $\gamma_0 = \frac{1}{10}$, $\gamma_1 = \frac{6}{10}$ and $\gamma_2 = \frac{3}{10}$. Finally the weights are given by

$$w_i^\pm = \frac{\alpha_i^\pm}{\sum_m \alpha_m^\pm}, \quad i = 0, 1, 2.$$

MODÈLE HYBRIDE LINÉARISÉ DANS LE CAS $1dx - 1dv$

Dire que c'est le début d'un article soumis, et qu'il y a des choses complémentaires (préciser les détails supplémentaires).

Ce chapitre est la première partie d'une étude du modèle hybride linéarisé (0.12)-(0.15) dans le cadre restreint $1dx - 1dv$. Cela permet de mettre en place certaines routines d'analyse, et d'utiliser les résultats du chapitre précédent dans le contexte d'un modèle hybride linéarisé. Ce travail collaboratif avec Anaïs Crestetto¹, Nicolas Crouseilles² et Yingzhe Li³ a mené à un article *Comparison of high-order Eulerian methods for electron hybrid model* soumis dans *Journal of Computational Physics* en 2021.

2.1 Introduction

L'objectif de ce chapitre est d'introduire et de simuler numériquement une hiérarchie de modèles permettant de décrire des systèmes de particules chargées où une population de particules chaudes interagit avec un plasma ambiant plus froid. Une telle configuration physique peut par exemple être étudiée dans les plasmas de tokamak où les particules alpha (générées par les réactions de fusion) interagissent avec le plasma ambiant. Un autre exemple se trouve dans la haute atmosphère où les électrons énergétiques du vent solaire interagissent avec la magnétosphère terrestre. Des modèles adaptés à ces configurations ont ainsi été obtenus par exemple dans les deux contextes (voir [49], [12] [52] [83] [85] [86]). Le modèle de départ qui servira de référence repose sur une description cinétique pour

1. Université de Nantes, laboratoire Jean Leray
 2. Univ Rennes, Inria Bretagne Atlantique (MINGuS) & ENS Rennes
 3. Max Planck Institute, Institut Für Plasmaphysik, Germany

l’ensemble du plasma considéré. On introduit alors la fonction de distribution des électrons $f(t, x, v) \in \mathbb{R}_+$ solution du modèle de Vlasov-Poisson (les ions sont considérées immobiles, comme étant un fond neutralisant). En supposant que la population électronique peut être séparée entre une population "froide" f_c et une population délectrons énergétiques f_h , une première étape consiste à écrire f comme la somme de ces deux fonctions de distribution $f = f_c + f_h$. Une seconde étape consiste à supposer que les particules froides restent proches d’un équilibre thermodynamique de température $T_c \approx 0$ et peuvent donc être représentées par un modèle fluide. On obtient le modèle hybride fluide/cinétique où la partie fluide (linéaire) décrit la dynamique des particules froides alors que les particules chaudes sont décrites à l'aide d'un modèle cinétique. Ce modèle hybride peut encore être simplifié en considérant des perturbations de type ondes de faible amplitude. Les termes non linéaires de la partie fluide sont donc négligés alors que la partie cinétique reste non linéaire. Le modèle ainsi obtenu (voir [49]) est le modèle hybride linéarisé VHL (Vlasov Hybrid Linearized).

Du fait de la forte disparité des vitesses thermiques entre les particules froides et chaudes, le modèle cinétique est très coûteux à résoudre numériquement, notamment car le maillage en vitesse doit être choisi très fin pour capturer la vitesse thermique des particules froides. Cela implique en outre, pour les schémas numériques classiques, une condition restrictive sur le pas de temps et donc des simulations coûteuses. La dérivation de modèles simplifiés moins coûteux à résoudre numériquement est donc d'un grand intérêt. Parmi ces modèles simplifiés, nous considérerons ici le modèle hybride linéarisé VHL étudié dans [49]. Afin d'effectuer une étude comparative entre le modèle VHL et le modèle cinétique original et de tester les schémas numériques, nous nous placerons dans le cas de la dimension 1 en espace et en vitesse. Ce cadre nous permettra aussi de poser les bases de l'étude du cas $1d-3v$ pour lequel il est beaucoup plus complexe d'effectuer ces comparaisons et ces tests. Ce type d'étude permettra enfin de comprendre le domaine de validité du modèle VHL.

Pour résoudre numériquement le modèle VHL, nous proposons deux méthodes. La première repose sur le fait que le modèle VHL possède une structure géométrique [70][85], [86]. Plus précisément, le modèle VHL possède une structure Hamiltonienne non canonique, ce qui signifie que les équations peuvent être obtenues à partir d'un crochet de Poisson et d'un Hamiltonien. Cette structure garantit la préservation d'invariants, comme l'énergie totale. L'objectif est d'exploiter cette structure pour construire des schémas numériques qui possèdent un bon comportement en temps long. Le schéma utilisé est un schéma de

type splitting construit à partir d'un *splitting* du Hamiltonien. Cette approche permet de combiner astucieusement certains termes du modèle et on est alors amené à résoudre trois sous-systèmes simples (comme dans [17], [10], [62]). Une propriété remarquable est que chacun des sous-systèmes peut être résolu exactement en temps, l'erreur en temps de la méthode provient donc uniquement de la méthode de *splitting* utilisée. Des méthodes d'ordre arbitraire en temps peuvent être obtenues par composition [43]. La deuxième méthode est basée sur un schéma exponentiel [48], [47], [61], [51], [60], [19]. En exploitant le fait que la partie linéaire du modèle VHL peut être résolue exactement et efficacement, on construit alors des schémas de type Lawson d'ordre élevé. Les résultats du chapitre précédent et de [19] sont donc repris et étendus au cas du système VHL.

Pour les deux méthodes en temps (splitting et Lawson), nous avons introduit une technique de pas de temps adaptatif. Pour les méthodes de type Lawson, le cadre des méthodes *embedded* [26][25] [2][1] permet de calculer l'erreur locale facilement. Dans le cas des méthodes de splitting, nous utiliserons le travail récent [5] qui propose des méthodes de splitting *embedded*. Des méthodes d'ordre 4(3) seront utilisées dans le cadre de la comparaison (ordre 3 et ordre 4 pour estimer l'erreur locale). Pour l'approximation de l'espace des phases, nous avons choisi une méthode spectrale en espace et une approximation type différences finies d'ordre élevé (ordre 5 en pratique) pour la direction en vitesse.

La première approche (splitting Hamiltonien) comporte des similarités avec les approches proposées dans [56] et [49] ; néanmoins, ces méthodes reposent sur une approximation de type Particle-In-Cell de l'espace des phases alors que nous utilisons des méthodes eulériennes. Ainsi, on est plus dans l'esprit de [17], [62] où on effectue un splitting puis on discrétise alors que dans [56] et [49], on discrétise l'espace des phases puis on discrétise en temps.

Afin de valider les résultats numériques, une étude approfondie des relations de dispersion est effectuée. Ces relations de dispersion sont obtenues par la résolution du modèle VHL linéarisé. À l'aide de transformée de Fourier en espace, de transformée de Laplace en temps, il est en effet possible de déterminer très précisément la phase linéaire des simulations de modèle non linéaire ; on peut calculer le taux d'amortissement ou d'instabilité d'un équilibre perturbé [78], [39], mais aussi reconstruire le mode fondamental du champ électrique. En plus de fournir des informations pour valider de manière quantitative les codes développés, cette analyse nous permet de faire le lien entre les modèles. En effet, en faisant tendre T_c vers zéro dans la relation de dispersion du modèle de Vlasov original,

il est possible de retrouver la relation de dispersion du modèle VHL.

Le chapitre est organisé comme suit : nous présentons tout d'abord la hiérarchie de modèles que nous souhaitons étudier, depuis le modèle cinétique jusqu'au modèle hybride linéarisé. La structure géométrique de ce modèle est exhibée en section 2.3. La section 2.4 est dédiée à la présentation des méthodes numériques construites pour la résolution du modèle hybride linéarisé. Dans la section 2.5, les relations de dispersion sont introduites et étudiées. Les sections 2.6 et 2.7 contiennent de nombreuses illustrations numériques. La section 2.6 se concentre sur la comparaison du modèle cinétique avec le modèle hybride linéarisé, alors que dans la section 2.7, nous étudions les avantages et les inconvénients des deux méthodes numériques pour le modèle hybride linéarisé.

2.2 Hiérarchie des modèles

Les modèles étudiés dans cette partie sont présentés dans le cas uni-dimensionnel en espace et en vitesse mais les dérivations peuvent être généralisées au cas multi-dimensionnel. Ainsi, notre point de départ est l'équation de Vlasov $1dx - 1dv$ suivante :

$$\begin{cases} \partial_t f + v\partial_x f + E\partial_v f = 0 \\ f(t = 0, x, v) = f^0(x, v) \end{cases}, \quad (2.1)$$

où $f = f(t, x, v)$ représente la densité de particules dans l'espace des phases $\{(x, v) \in \Omega \times \mathbb{R}\}$ avec $\Omega \subset \mathbb{R}$, au temps $t \geq 0$, f^0 est la condition initiale et $E(t, x)$ désigne le champ électrique qui est obtenu soit par l'équation de Poisson :

$$\partial_x E = \int_{\mathbb{R}} f \, dv - 1 \quad (2.2)$$

ou de manière équivalente par l'équation d'Ampère :

$$\partial_t E = - \int_{\mathbb{R}} v f \, dv + \frac{1}{|\Omega|} \int_{\Omega} \int_{\mathbb{R}} v f \, dv \, dx, \quad (2.3)$$

couplée à une condition initiale $E(t = 0, x) = E^0(x)$ qui vérifie l'équation de Poisson (2.2) au temps initial. Ces deux dernières équations font des systèmes de Vlasov-Poisson (2.1)-(2.2) et de Vlasov-Ampère (2.1)-(2.3) des équations de transport non linéaire d'une quantité f dans l'espace des phases $\Omega \times \mathbb{R}$. On considérera des conditions périodiques en espace

et nulles à l'infini en vitesse.

2.2.1 Dérivation du modèle de Vlasov hybride linéarisé

Partons du modèle de Vlasov-Ampère (2.1)-(2.3) :

$$\begin{cases} \partial_t f + v \partial_x f + E \partial_v f = 0 \\ \partial_t E = - \int_{\mathbb{R}} v f \, dv + \frac{1}{|\Omega|} \int_{\Omega} \int_{\mathbb{R}} v f \, dv \, dx, \end{cases}$$

avec la condition initiale $(f^0(x, v), E^0(x))$ vérifiant $\partial_x E^0(x) = \int_{\mathbb{R}} f^0(x, v) \, dv - 1$. On souhaite distinguer la population de particules f en deux : un premier groupe de particules froides f_c dont la vitesse thermique est faible et un second groupe de particules dites chaudes f_h , dont la vitesse thermique est grande. Le modèle de Vlasov-Ampère en considérant ces deux espèces indépendamment s'écrit :

$$\begin{cases} \partial_t f_c + v \partial_x f_c + E \partial_v f_c = 0, \\ \partial_t f_h + v \partial_x f_h + E \partial_v f_h = 0, \\ \partial_t E = - \int_{\mathbb{R}} v f_c \, dv - \int_{\mathbb{R}} v f_h \, dv + \frac{1}{|\Omega|} \int_{\Omega} \int_{\mathbb{R}} v f \, dv \, dx, \end{cases}$$

avec la condition initiale $(f_c^0(x, v), f_h^0(x, v), E^0(x))$ vérifiant

$$\partial_x E^0(x) = \int_{\mathbb{R}} f_c^0(x, v) \, dv + \int_{\mathbb{R}} f_h^0(x, v) \, dv - 1.$$

La compatibilité $f^0 = f_c^0 + f_h^0$ est nécessaire pour garantir l'équivalence entre les deux modèles.

On souhaite essentiellement travailler sur la variable f_c pour la considérer non plus comme une inconnue cinétique mais fluide (donc ne dépendant plus de la vitesse v mais seulement du temps t et de la position x). En effet, elle représente des particules froides, de faible vitesse, dont on peut supposer qu'elles restent proches d'un équilibre thermodynamique. Pour cela calculons les moments de la première équation en multipliant celle-ci par $(1, v)^T$ puis en intégrant par rapport à v :

$$\int_{\mathbb{R}} \binom{1}{v} \partial_t f_c \, dv + \int_{\mathbb{R}} \binom{1}{v} v \partial_x f_c \, dv + \int_{\mathbb{R}} \binom{1}{v} E \partial_v f_c \, dv = 0.$$

On introduit la densité $\rho_c(t, x)$ et la vitesse moyenne $u_c(t, x)$ des particules froides

$$\begin{pmatrix} \rho_c(t, x) \\ \rho_c(t, x)u_c(t, x) \end{pmatrix} = \int_{\mathbb{R}} \begin{pmatrix} 1 \\ v \end{pmatrix} f_c(t, x, v) dv,$$

de sorte que le système aux moments se réécrire

$$\partial_t \begin{pmatrix} \rho_c \\ \rho_c u_c \end{pmatrix} + \partial_x \begin{pmatrix} \rho_c u_c \\ \int_{\mathbb{R}} v^2 f_c dv \end{pmatrix} - \begin{pmatrix} 0 \\ \rho_c E \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (2.4)$$

Le système (2.4) n'étant pas fermé, il faut faire une hypothèse sur la répartition en vitesse des particules froides. On utilisera l'approximation "*plasma froid*" utilisé dans la littérature ([86], [49]) qui suppose l'approximation $f_c(t, x, v) = \rho_c(t, x)\delta_{\{v=u_c(t, x)\}}(v)$, ce qui nous permet d'obtenir le système :

$$\begin{cases} \partial_t \rho_c + \partial_x (\rho_c u_c) = 0 \\ \partial_t (\rho_c u_c) + \partial_x (\rho_c u_c^2) - \rho_c E = 0, \end{cases}$$

puisque $\int_{\mathbb{R}} v^2 \rho_c(t, x)\delta_{\{v=u_c(t, x)\}}(v) dv = \rho_c(t, x)u_c^2(t, x)$. Ce modèle est connu dans la littérature sous le nom d'équations d'Euler sans pression.

En considérant le couplage avec le modèle de Vlasov pour les particules chaudes, l'équation d'Ampère et l'équation de Poisson, on obtient ainsi le modèle de Vlasov-Ampère hybride non-linéaire

$$\begin{cases} \partial_t \rho_c + \partial_x (\rho_c u_c) = 0, \\ \partial_t (\rho_c u_c) + \partial_x (\rho_c u_c^2) - \rho_c E = 0, \\ \partial_t f_h + v \partial_x f_h + E \partial_v f_h = 0, \\ \partial_t E = -\rho_c u_c - \int_{\mathbb{R}} v f_h dv + \frac{1}{|\Omega|} \int_{\Omega} \int_{\mathbb{R}} v f_h dv dx + \frac{1}{|\Omega|} \int_{\Omega} \rho_c u_c dx, \\ \partial_x E = \rho_c + \int_{\mathbb{R}} f_h dv - 1, \end{cases} \quad (2.5)$$

avec les conditions initiales $(\rho_c^0, \rho_c^0 u_c^0, f_h^0, E^0)$ vérifiant

$$\partial_x E^0(x) = \rho_c^0(x) + \int_{\mathbb{R}} f_h^0(x, v) dv - 1.$$

La compatibilité $\int_{\mathbb{R}} (f^0(x, v) - f_h^0(x, v)) dv = \rho_c^0(x)$ est nécessaire pour garantir le lien avec

le modèle hybride non-linéaire (2.5) et le modèle cinétique de départ.

Le modèle (2.5) peut être récrit de manière équivalente sous la forme d'un modèle plus simple. C'est l'objet de la proposition suivante.

Proposition 2.1. *Le modèle (2.5) se réécrit sous la forme ($x \in \Omega \subset \mathbb{R}$ and $v \in \mathbb{R}$)*

$$\begin{cases} \partial_t u_c + \frac{1}{2} \partial_x u_c^2 - E = 0, \\ \partial_t f_h + v \partial_x f_h + E \partial_v f_h = 0, \\ \partial_t E = -\rho_c u_c - \int_{\mathbb{R}} v f_h \, dv + \frac{1}{|\Omega|} \int_{\Omega} \int_{\mathbb{R}} v f_h \, dv \, dx + \frac{1}{|\Omega|} \int_{\Omega} \rho_c u_c \, dx, \end{cases}$$

avec les conditions initiales $(u_c^0, f_h^0, E^0, \rho_c^0)$ vérifiant

$$\partial_x E^0(x) = \rho_c^0(x) + \int_{\mathbb{R}} f_h^0(x, v) \, dv.$$

La densité ρ_c est obtenue pour tout temps $t \geq 0$ par

$$\rho_c(t, x) = \partial_x E(t, x) - \int_{\mathbb{R}} f_h(t, x, v) \, dv + 1.$$

Démonstration. À partir de la deuxième équation de (2.5), on écrit

$$\rho_c \partial_t u_c + u_c \partial_t \rho_c + \rho_c u_c \partial_x u_c + u_c \partial_x (\rho_c u_c) - \rho_c E = 0.$$

Grâce à l'équation de continuité (première équation de (2.5)), on obtient alors, après simplification par ρ_c

$$\partial_t u_c + u_c \partial_x u_c - E = 0.$$

Prenons l'équation d'Ampère et dérivons-la par rapport à la variable x :

$$\partial_x \partial_t E = -\partial_x (\rho_c u_c) - \int_{\mathbb{R}} v \partial_x f_h \, dv.$$

Or le modèle (2.5) nous donne

$$-\partial_x (\rho_c u_c) = \partial_t \rho_c \text{ et } v \partial_x f_h = -\partial_t f_h - E \partial_v f_h,$$

ce qui permet d'écrire :

$$\partial_t \partial_x E = \partial_t \rho_c + \int_{\mathbb{R}} (\partial_t f_h + E \partial_v f_h) \, dv = \partial_t \rho_c + \partial_t \int_{\mathbb{R}} f_h \, dv.$$

Après intégration en temps entre 0 et t , on obtient :

$$\partial_x E - \partial_x E^0 = \rho_c + \int_{\mathbb{R}} f_h dv - \rho_c^0 - \int_{\mathbb{R}} f_h^0 dv.$$

Ayant supposé que initialement on a $\partial_x E^0 = \rho_c^0 + \int_{\mathbb{R}} f_h^0 dv - 1$, on obtient finalement :

$$\partial_x E = \rho_c + \int_{\mathbb{R}} f_h dv - 1,$$

ce qui signifie que si l'équation de Poisson est vérifiée initialement, elle est vérifiée pour tout $t \geq 0$. \square

Dans la littérature physique, le modèle hybride non linéaire est encore simplifié et c'est la version linéarisée de la partie fluide qui est étudiée (voir [49], [86]). Ainsi, on considère maintenant la linéarisation du modèle (2.1) satisfait par (ρ_c, u_c, E, f_h) autour de l'équilibre donné par $(\rho_c^{(0)}(x), 0, 0, f_h^{(0)}(v))$ avec $f_h^{(0)}(v)$ une fonction paire. L'objectif est d'obtenir un modèle dans lequel la partie fluide est linéaire alors que l'équation cinétique non linéaire est conservée pour les particules chaudes. Remarquons que le terme $\frac{1}{|\Omega|} \int_{\Omega} \int_{\mathbb{R}} v f_h dv dx + \frac{1}{|\Omega|} \int_{\Omega} \rho_c u_c dx$ dans l'équation d'Ampère ne sera pas pris en compte dans la suite pour alléger les notations. On écrit alors

$$\begin{aligned} \rho_c(t, x) &= \rho_c^{(0)}(x) + \varepsilon \rho_c^{(1)}(t, x) \\ u_c(t, x) &= \varepsilon u_c^{(1)}(t, x) \\ E(t, x) &= \varepsilon E^{(1)}(t, x), \\ f_h(t, x, v) &= f_h^{(0)}(v) + \varepsilon f_h^{(1)}(t, x, v), \end{aligned} \tag{2.6}$$

et on insère dans les équations fluides du modèle hybride non linéaire pour obtenir (la partie cinétique n'est pas modifiée) :

$$\begin{cases} \varepsilon \partial_t u_c^{(1)} + \frac{\varepsilon^2}{2} \partial_x \left(u_c^{(1)} \right)^2 - \varepsilon E^{(1)} = 0 \\ \varepsilon \partial_t E^{(1)} = - \int_{\mathbb{R}} v (f_h^{(0)} + \varepsilon f_h^{(1)}) dv - \varepsilon \rho_c^{(0)} u_c^{(1)} - \varepsilon^2 \rho_c^{(1)} u_c^{(1)}. \end{cases}$$

On néglige maintenant les termes non linéaire (en ε^2), ce qui nous permet, sous l'hypothèse

$f_h^{(0)}$ paire d'écrire le système suivant :

$$\begin{cases} \partial_t u_c^{(1)} - E^{(1)} = 0 \\ \partial_t E^{(1)} = - \int_{\mathbb{R}} v f_h^{(1)} dv - \rho_c^{(0)} u_c^{(1)} \end{cases}$$

soit encore, avec les notations (2.6) et en utilisant que $\varepsilon \int_{\mathbb{R}} v f_h^{(1)} dv = \int_{\mathbb{R}} v f_h dv$ (puisque $f_h^{(0)}(v)$ paire implique $\int_{\mathbb{R}} v f_h^{(0)} dv = 0$)

$$\begin{cases} \partial_t u_c - E = 0 \\ \partial_t E = - \int_{\mathbb{R}} v f_h dv - \rho_c^{(0)} u_c. \end{cases}$$

Le système d'équations de Vlasov hybride linéarisé s'écrit donc finalement :

$$\begin{cases} \partial_t u_c = E \\ \partial_t E = -\rho_c^{(0)} u_c - \int_{\mathbb{R}} v f_h dv \\ \partial_t f_h + v \partial_x f_h + E \partial_v f_h = 0, \end{cases} \quad (2.7)$$

avec la condition initiale (u_c^0, E^0, f_h^0) et $\rho_c^{(0)}(x)$ tels que $\partial_x E^0(x) = \rho_c^{(0)}(x) + \int_{\mathbb{R}} f_h^0(x, v) dv - 1$.

Remarque 2.1. *Tous les calculs précédents se généralisent au cas multi-dimensionnel $x, v \in \mathbb{R}^d$ et le modèle hybride non linéaire s'écrit alors*

$$\begin{cases} \partial_t u_c + (u_c \cdot \nabla_x) u_c = E \\ \partial_t E = -\rho_c u_c - \int_{\mathbb{R}^d} v f_h dv \\ \partial_t f_h + v \cdot \nabla_x f_h + E \cdot \nabla_v f_h = 0, \end{cases} \quad (2.8)$$

et le modèle hybride linéarisé s'écrit

$$\begin{cases} \partial_t u_c = E \\ \partial_t E = -\rho_c^{(0)} u_c - \int_{\mathbb{R}^d} v f_h dv \\ \partial_t f_h + v \cdot \nabla_x f_h + E \cdot \nabla_v f_h = 0, \end{cases} \quad (2.9)$$

avec les conditions initiales (u_c^0, E^0, f_h^0) et $\rho_c^{(0)}(x)$ tels que $\nabla_x \cdot E^0(x) = \rho_c^{(0)}(x) + \int_{\mathbb{R}} f_h^0(x, v) dv - 1$.

2.2.2 Quelques propriétés du modèle hybride linéarisé

Dans cette partie, quelques propriétés du modèle hybride linéarisé sont exhibées.

Proposition 2.2. *Le modèle hybride linéarisé VHL (2.7) assure la conservation de la masse totale et de l'énergie totale, c'est-à-dire*

$$\begin{aligned} 0 &= \frac{d}{dt} \int_{\Omega} \int_{\mathbb{R}} f_h dx dv \left(= \frac{d}{dt} \int_{\Omega} \rho_c^{(0)} dx \right), \\ 0 &= \frac{d}{dt} \left[\int_{\Omega} \int_{\mathbb{R}} f_h v^2 dx dv + \int_{\mathbb{R}} \rho_c u_c^2 dx + \int_{\Omega} E^2 dx \right]. \end{aligned}$$

Remarque 2.2. *À partir de la conservation de l'énergie totale pour l'équation cinétique*

$$\frac{d}{dt} \left[\int_{\Omega} \int_{\mathbb{R}} f v^2 dx dv + \int_{\Omega} E^2 dx \right] = 0,$$

et l'approximation de $f_c(t, x, v)$ par $\rho_c(t, x) \delta_{v=u_c(t,x)}(v) + f_h(t, x, v)$, on peut retrouver la conservation de l'énergie totale pour le modèle VHL grâce au calcul suivant

$$\begin{aligned} 0 &= \frac{d}{dt} \left[\iint_{\Omega \times \mathbb{R}} (f_c + f_h) v^2 dx dv + \int_{\Omega} E^2 dx \right] \\ &= \frac{d}{dt} \left[\iint_{\Omega \times \mathbb{R}} (\rho_c \delta_{v=u_c(t,x)}(v) + f_h) v^2 dx dv + \int_{\Omega} E^2 dx \right] \\ &= \frac{d}{dt} \left[\int_{\Omega} \rho_c u_c^2 dx + \iint_{\Omega \times \mathbb{R}} f_h v^2 dx dv + \int_{\Omega} E^2 dx \right]. \end{aligned}$$

Démonstration.

- La conservation de la masse s'obtient en intégrant tout d'abord l'équation de Vlasov en espace et en vitesse :

$$\frac{d}{dt} \int_{\Omega} \int_{\mathbb{R}} f_h dx dv + \int_{\Omega} \int_{\mathbb{R}} E \partial_v f_h dv dx = 0.$$

Or, avec les conditions aux bords choisies, on a $\frac{d}{dt} \int_{\Omega} \int_{\mathbb{R}} f_h dx dv = 0$ et comme $\rho_c^{(0)}$ ne dépend pas du temps, on déduit la conservation de la masse totale. Notons que $\int_{\Omega} E dx = \int_{\Omega} u dx = 0$.

- Pour la conservation de l'énergie, on multiplie l'équation de Vlasov par v^2 et on

intègre en x, v pour obtenir, après intégration par partie

$$\frac{d}{dt} \int_{\Omega} \int_{\mathbb{R}} v^2 f_h dx dv - 2 \int_{\Omega} E \left(\int_{\mathbb{R}} v f_h dv \right) dx = 0.$$

En utilisant l'équation d'Ampère, le deuxième terme devient

$$\begin{aligned} 2 \int_{\Omega} E \left(\partial_t E + \rho_c^{(0)} u_c \right) dx &= \frac{d}{dt} \int_{\Omega} E^2 dx + 2 \int_{\Omega} E \rho_c^{(0)} u_c dx, \\ &= \frac{d}{dt} \int_{\Omega} E^2 dx + 2 \int_{\Omega} (\partial_t u_c) \rho_c^{(0)} u_c dx, \\ &= \frac{d}{dt} \int_{\Omega} E^2 dx + \frac{d}{dt} \int_{\Omega} \rho_c^{(0)} u_c^2 dx, \end{aligned}$$

où on a utilisé l'équation sur u_c et le fait que $\rho_c^{(0)}$ ne dépende pas du temps.

□

2.3 Structure géométrique du modèle hybride linéarisé VHL

Dans cette partie, nous présentons la structure du modèle hybride linéarisé VHL (2.7), à savoir son Hamiltonien et son crochet de Poisson. Pour simplifier les notations, nous notons dans cette section $f = f_h$, $\rho_c = \rho_c^{(0)}$ et $u = u_c$. Cette structure permet notamment d'assurer la préservation de nombreux invariants (énergie totale et opérateurs de Casimir entre autres) mais sera à la base d'un *splitting* en temps, dans l'esprit de [17], [10], [56] [62]. Nous aurons besoin d'introduire certaines notations pour pouvoir introduire la structure.

Tout d'abord, rappelons que pour une fonctionnelle donnée $\mathcal{G}(f)$, la dérivée de Fréchet de la distribution $\frac{\delta \mathcal{G}}{\delta f}(f)$ évaluée au point f , est définie par

$$\mathcal{G}(f + \delta f) - \mathcal{G}(f) = \int_{\Omega \times \mathbb{R}} \frac{\delta \mathcal{G}}{\delta f}(f)(x, v) \delta f(x, v) dx dv + \mathcal{O}(\delta f^2), \quad (2.10)$$

pour toute variation régulière δf . On définit le Hamiltonien associé au modèle VHL (2.7)

$$\mathcal{H} = \frac{1}{2} \int_{\mathbb{R}} E^2 dx + \frac{1}{2} \int_{\mathbb{R}} \rho_c u^2 dx + \frac{1}{2} \int_{\mathbb{R}} \int_{\mathbb{R}} v^2 f dx dv, \quad (2.11)$$

$$= \mathcal{H}_E + \mathcal{H}_u + \mathcal{H}_f. \quad (2.12)$$

Les trois termes correspondent respectivement à l'énergie électrique, l'énergie cinétique des particules froides et l'énergie cinétique des particules chaudes. Pour une fonctionnelle $\mathcal{G}(E, u, f)$, on notera $\delta\mathcal{G}/\delta f$, $\delta\mathcal{G}/\delta E$ et $\delta\mathcal{G}/\delta u$ les dérivées de Fréchet de \mathcal{G} par rapport à f , E et u respectivement. On introduit à présent le crochet de Poisson de deux fonctionnelles $\mathcal{F}(E, u, f)$ et $\mathcal{G}(E, u, f)$

$$\begin{aligned} \{\mathcal{F}, \mathcal{G}\}(u, E, f) &= \int_{\mathbb{R}} \int_{\mathbb{R}} f \left(\partial_x \frac{\delta \mathcal{F}}{\delta f} \partial_v \frac{\delta \mathcal{G}}{\delta f} - \partial_v \frac{\delta \mathcal{F}}{\delta f} \partial_x \frac{\delta \mathcal{G}}{\delta f} \right) dv dx \\ &\quad + \int_{\mathbb{R}} \left(\frac{\delta \mathcal{F}}{\delta u} \frac{\delta \mathcal{G}}{\delta E} - \frac{\delta \mathcal{F}}{\delta E} \frac{\delta \mathcal{G}}{\delta u} \right) dx \\ &\quad + \int_{\mathbb{R}} \int_{\mathbb{R}} \left(\frac{\delta \mathcal{F}}{\delta E} \partial_v f \frac{\delta \mathcal{G}}{\delta f} - \frac{\delta \mathcal{G}}{\delta E} \partial_v f \frac{\delta \mathcal{F}}{\delta f} \right) dv dx \end{aligned}$$

Avec cette notation, le modèle hybride linéarisé (2.7) se réécrit alors, avec $U = (u, E, f)$ et \mathcal{H} donné par (2.11)

$$\partial_t U = \{U, \mathcal{H}\}. \quad (2.13)$$

Dans la suite, on vérifie que la réécriture (2.13) est bien équivalente au modèle VHL. Pour cela, on a besoin des relations suivantes

$$\frac{\delta \mathcal{H}}{\delta f} = \frac{v^2}{2}, \quad \frac{\delta \mathcal{H}}{\delta u} = \rho_c u, \quad \frac{\delta \mathcal{H}}{\delta E} = E.$$

De plus, par abus de notation, on notera la fonctionnelle associée à la fonction comme suit (par exemple pour u) $u(t, z) = \int_{\mathbb{R}} u(t, x) \delta(x - z) dx$, de sorte que $\frac{\delta u}{\delta u} = \delta(x - z)$. Pour f , on notera $f(t, x, w) = \int_{\mathbb{R}} f(t, z, v) \delta(x - z) \delta(w - v) dx dv$, de sorte que $\frac{\delta f}{\delta f} = \delta(x - z) \delta(w - v)$.

- On calcule dans un premier temps $\{u, \mathcal{H}\}$

$$\partial_t u(t, z) = \{u, \mathcal{H}\} = 0 + \int_{\mathbb{R}} \delta(x - z) E(t, x) dx + 0 = E(t, z)$$

- Puis on considère $\{E, \mathcal{H}\}$

$$\begin{aligned} \partial_t E(t, z) = \{E, \mathcal{H}\} &= 0 - \int_{\mathbb{R}} \delta(x - z) \rho_c u dx + \int_{\mathbb{R}} \int_{\mathbb{R}} \left(\delta(x - z) \partial_v f \frac{v^2}{2} \right) dv dx \\ &= -\rho_c u(t, z) - \int_{\mathbb{R}} f(t, z, v) v dv \end{aligned}$$

- Finalement, $\{f, \mathcal{H}\}$ donne

$$\begin{aligned} \partial_t f(t, z, w) = \{f, \mathcal{H}\} &= \int_{\mathbb{R}} \int_{\mathbb{R}} f \left(\partial_x (\delta(x-z)\delta(w-v)) \partial_v \frac{v^2}{2} \right) dv dx \\ &\quad + 0 - \int_{\mathbb{R}} \int_{\mathbb{R}} (E \partial_v f) \delta(x-z) \delta(w-v) dv dx \\ &= (-v \partial_x f - E \partial_v f)(t, z, w). \end{aligned}$$

Enfin, on peut vérifier que le crochet de Poisson satisfait les propriétés suivantes

- anti-symétrie : $\{F, G\} = -\{G, F\}$
- bilinéarité : $\{F + G, H\} = \{F, H\} + \{G, H\}$
- identité de Jacobi : $\{\{F, G\}, H\} + \{\{G, H\}, F\} + \{\{H, F\}, G\} = 0$.

Les deux premières propriétés sont simples alors que la dernière est habituellement plus compliquée. On utilise les calculs de [62], [69].

2.4 Schémas numériques

Nous allons maintenant présenter les schémas numériques développés pour approcher la solution du modèle hybride linéarisé (2.7). Notre but est de comparer deux approches pour la discrétisation temporelle. D'une part, pour profiter de la structure hamiltonienne présentée dans la section 2.3, nous proposons une méthode de *splitting* hamiltonien en temps, couplée à une méthode de composition d'ordre élevé, en l'occurrence la méthode de Suzuki ([81], [43], [5]). Concernant la discrétisation en espace, nous utilisons la transformée de Fourier discrète, alors que la discrétisation en vitesse est effectuée par une méthode semi-Lagrangienne d'ordre 5. Nous détaillons cette approche dans la sous-section 2.4.1. D'autre part, le modèle VHL (2.7) peut être récrit sous la forme

$$\partial_t U = AU + N(U),$$

avec A une matrice 3×3 et N représente les termes non linéaires. On reconnaît une structure particulière qu'un intégrateur exponentiel peut exploiter, en particulier pour éviter une condition CFL trop restrictive induite par le terme de transport en espace (souvent la plus restrictive [19]). Nous présentons dans la sous-section 2.4.2 la discrétisation en temps de Lawson d'ordre 4 que nous avons choisi d'implémenter. Celle-ci est couplée à une méthode de transformée de Fourier discrète en espace et à une méthode WENO d'ordre 5

en vitesse. Les deux schémas numériques obtenus sont donc d'ordre élevé dans toutes les variables. Pour optimiser le temps de calcul des deux schémas, nous proposons d'utiliser une méthode de pas de temps adaptatif, présentée dans la sous-section 2.4.3. Pour le cas des schémas exponentiels, cette approche de pas de temps adaptatif est motivée par le fait que le terme non linéaire $E\partial_v f$ ne va pas induire de grand déplacement au moins dans la phase linéaire (car $|E| \ll 1$ dans ce régime), ce qui permettra d'utiliser de grands pas de temps ; dans le cas où E est plus grand, cela signifie que le système tente de reproduire des phénomènes complexes non linéaires, ainsi un pas de temps plus petit devra être considéré pour capturer ces phénomènes.

2.4.1 Méthode de *splitting* hamiltonien

On construit une méthode numérique de type *splitting* à partir de la décomposition du Hamiltonien

$$\mathcal{H} = \mathcal{H}_E + \mathcal{H}_u + \mathcal{H}_f.$$

Ainsi, avec $U = (u, E, f)$, le *splitting* en temps se déduit de

$$\partial_t U = \{U, \mathcal{H}_E + \mathcal{H}_u + \mathcal{H}_f\} = \{U, \mathcal{H}_E\} + \{U, \mathcal{H}_u\} + \{U, \mathcal{H}_f\}. \quad (2.14)$$

Dans la suite, on nommera les solutions correspondantes aux trois différentes parties $\varphi^{[E]}$, $\varphi^{[u]}$ et $\varphi^{[f]}$. On commence par $\varphi^{[E]}(U)$ solution de $\partial_t U = \{U, \mathcal{H}_E\}$,

$$\begin{aligned} \partial_t u &= \{u, \mathcal{H}_E\} = E \\ \partial_t E &= \{E, \mathcal{H}_E\} = 0 \\ \partial_t f &= \{f, \mathcal{H}_E\} = -E\partial_v f \end{aligned} \quad (2.15)$$

Puis, on considère $\partial_t U = \{U, \mathcal{H}_u\}$ (dont la solution est $\varphi^{[u]}(U)$)

$$\begin{aligned} \partial_t u &= \{u, \mathcal{H}_u\} = 0 \\ \partial_t E &= \{E, \mathcal{H}_u\} = -\rho_c u \\ \partial_t f &= \{f, \mathcal{H}_u\} = 0 \end{aligned} \quad (2.16)$$

Enfin, on écrit les équations associées à $\partial_t U = \{U, \mathcal{H}_f\}$ (dont la solution est $\varphi^{[f]}(U)$),

$$\begin{aligned}\partial_t u &= \{u, \mathcal{H}_f\} = 0 \\ \partial_t E &= \{E, \mathcal{H}_f\} = - \int_{\mathbb{R}} v f \, dv \\ \partial_t f &= \{f, \mathcal{H}_f\} = -v \partial_x f\end{aligned}\tag{2.17}$$

Ainsi, la solution $\varphi(U)$ de (2.14) sera approchée par la composition de $\varphi^{[u]}(U)$, $\varphi^{[E]}(U)$ et $\varphi^{[f]}(U)$. Par exemple, un splitting de Lie-Trotter permet d'approcher $\varphi(U)$ grâce à $\varphi(U) \approx \varphi^{[E]} \circ \varphi^{[u]} \circ \varphi^{[f]}(U)$. Une remarque importante est que chaque sous-système peut être résolu exactement en temps de sorte que l'erreur en temps ne provient que du splitting.

2.4.1.1 Résolution de chaque sous-système

Nous nous intéressons à la semi-discrétisation en temps du *splitting* hamiltonien donné par les trois systèmes (2.15)-(2.16)-(2.17). Soit $\Delta t > 0$ un pas de temps, on définit $t^n = n\Delta t$ pour $n \geq 0$ et on note U^n l'approximation de $U(t^n)$. Connaissant U^n , nous souhaitons calculer U^{n+1} . Pour ne pas multiplier les notations, nous notons dans chaque étape du *splitting* \tilde{U}^n le résultat de l'étape précédente (pour la première étape $\tilde{U}^n = U^n$). Il est important dans une méthode de *splitting* de résoudre exactement chaque sous-système. Nous résolvons chaque système de la manière suivante.

- $U_{\Delta t}^{[E]} = \varphi_{\Delta t}^{[E]}(\tilde{U}^n)$: dans cette étape, E est constant, par conséquent l'intégration en temps de l'équation $\partial_t u_c = E$ ne pose pas de problème :

$$u_{c\Delta t}^{[E]} = \tilde{u}_c^n + \Delta t \tilde{E}^n$$

La deuxième équation $\partial_t E = 0$ se résout par une simple copie des données :

$$E_{\Delta t}^{[E]} = \tilde{E}^n$$

La troisième équation $\partial_t f_h + E \partial_v f_h = 0$ est un transport en v , on pourrait souhaiter résoudre cette équation par une transformée de Fourier en v , mais pour effectuer une comparaison avec la méthode de Lawson que nous verrons par la suite, nous utiliserons ici une méthode semi-lagrangienne, par conséquent le problème se résout en remontant

une caractéristique.

$$f_h^{[E]} = \tilde{f}_h^n(x, v - \Delta t \tilde{E}^n)$$

On synthétise cela avec :

$$U_{\Delta t}^{[E]} = \varphi_{\Delta t}^{[E]}(\tilde{U}^n) = \begin{pmatrix} \tilde{u}_c^n + \Delta t \tilde{E}^n \\ \tilde{E}^n \\ \tilde{f}_h(x, v - \Delta t \tilde{E}^n) \end{pmatrix}$$

- $U_{\Delta t}^{[u]} = \varphi_{\Delta t}^{[u]}(\tilde{U}^n)$: dans cette étape les variables u_c et f_h n'évoluent pas au cours du temps, il n'y a qu'une équation différentielle sur E que l'on peut résoudre de façon exacte. Pour la variable u_c , respectivement f_h , on a l'équation $\partial_t u_c = 0$, respectivement $\partial_t f_h = 0$, ce qui nous donne :

$$u_{c\Delta t}^{[u]} = \tilde{u}_c^n \quad f_{h\Delta t}^{[u]} = \tilde{f}_h^n$$

Enfin l'équation sur E : $\partial_t E = -\rho_c^0 u_c$:

$$E_{\Delta t}^{[u]} = \tilde{E}^n - \Delta t \rho_c^0 \tilde{u}_c^n$$

On résumera cette étape par :

$$U_{\Delta t}^{[u]} = \varphi_{\Delta t}^{[u]}(\tilde{U}^n) = \begin{pmatrix} \tilde{u}^n \\ \tilde{E}^n - \Delta t \rho_c \tilde{u}_c^n \\ \tilde{f}_h^n \end{pmatrix}$$

- $U_{\Delta t}^{[f]} = \varphi_{\Delta t}^{[f]}(\tilde{U}^n)$: pour résoudre cette étape, la première équation $\partial_t u_c = 0$ ne présente pas de difficulté :

$$u_{c\Delta t}^{[f]} = \tilde{u}_c^n$$

la troisième équation $\partial_t f_h + v \partial_x f_h = 0$ se résout simplement après une transformée de Fourier en x , et elle peut se résoudre exactement pour tout $s \in [t^n, t^{n+1}]$:

$$f_{h\Delta t}^{[f]} = f_h^{[f]}(\Delta t) \quad \hat{f}_h^{[f]}(s) = e^{-ikv(s-t^n)} \hat{\tilde{f}}_h^n,$$

où $\hat{\tilde{f}}_h^n$ désigne la transformée de Fourier de \tilde{f}_h^n en x et k désigne la variable de Fourier. La deuxième équation $\partial_t E = -\int v f_h dv$ profite de la connaissance exacte pour tout temps de f_h sur l'intervalle de temps considéré, cela permet d'effectuer une intégration en temps

sans difficulté, en effet on a :

$$\begin{cases} \partial_t E = - \int v f_h \, dv \\ \hat{f}_h^{[f]}(s) = e^{-ikv(s-t^n)} \hat{\tilde{f}}_h^n, \quad \forall k \in \mathbb{Z} \end{cases}$$

On insère l'équation sur $f_h^{[f]}(s)$, pour tout $s \in [t^n, t^n + \Delta t]$, on travaille sur les modes de Fourier, une intégration en temps sur l'intervalle $[t^n, t^n + \Delta t]$ nous permet d'obtenir ($\hat{E}_{\Delta t}^{[f]}$ et \tilde{E}^n désignent les transformées de Fourier de $E_{\Delta t}^{[f]}$ et \tilde{E}^n)

$$\begin{aligned} \hat{E}_{\Delta t}^{[f]} &= \tilde{E}^n - \int_{t^n}^{t^n + \Delta t} \int_{\mathbb{R}} v e^{-ikv(s-t^n)} \hat{\tilde{f}}_h^n \, dv \, ds \\ &= \tilde{E}^n - \int_{\mathbb{R}} v \hat{\tilde{f}}_h^n \int_{t^n}^{t^n + \Delta t} e^{-ikv(s-t^n)} \, ds \, dv \\ &= \tilde{E}^n - \int_{\mathbb{R}} v \hat{\tilde{f}}_h^n \left[\frac{-1}{ikv} (e^{-ikv\Delta t} - 1) \right] \, dv \\ &= \hat{E}^n - \frac{i}{k} \int_{\mathbb{R}} (e^{-ikv\Delta t} - 1) \hat{\tilde{f}}_h^n \, dv. \end{aligned}$$

On synthétise cela avec :

$$U_{\Delta t}^{[f]} = \varphi_{\Delta t}^{[f]}(\tilde{U}^n) = \begin{pmatrix} \tilde{u}_c^n \\ \hat{E}^n - \frac{i}{k} \int_{\mathbb{R}} (e^{-ikv\Delta t} - 1) \hat{\tilde{f}}_h^n \, dv \\ e^{-ikv\Delta t} \hat{\tilde{f}}_h^n \end{pmatrix}$$

On a ainsi chacune de nos 3 étapes qui est résolue de manière exacte. Le pas de temps d'intégration Δt est à voir comme un paramètre de la résolution de chaque sous étape, ce qui permet en les réalisant successivement sur un pas de temps Δt , d'obtenir un *splitting* de Lie :

$$U^{n+1} = \mathcal{L}_{\Delta t}(U^n) = \varphi_{\Delta t}^{[E]} \circ \varphi_{\Delta t}^{[u]} \circ \varphi_{\Delta t}^{[f]}(U^n). \quad (2.18)$$

Mais nous pouvons les concaténer différemment, avec des pas d'intégration différents, pour construire la méthode de Strang ou une méthode d'ordre plus élevé que nous allons voir dans la sous-section suivante, la méthode de Suzuki.

2.4.1.2 Composition d'ordre élevé

On s'intéresse à une méthode en temps d'ordre élevé, la méthode de Suzuki [81]. Celle-ci se construit à partir de la méthode de Strang [80] dont la formulation à 3 étapes s'écrit comme suit :

$$U^{n+1} = S_{\Delta t}(U^n) = \varphi_{\Delta t/2}^{[E]} \circ \varphi_{\Delta t/2}^{[u]} \circ \varphi_{\Delta t}^{[f]} \circ \varphi_{\Delta t/2}^{[E]} \circ \varphi_{\Delta t/2}^{[u]}(U^n) \quad (2.19)$$

La méthode de Suzuki est une composition de 5 méthodes de Strang, donc un total de 25 étapes. Celle-ci s'écrit :

$$U^{n+1} = \mathcal{S}_{\Delta t}(U^n) = S_{\alpha_1 \Delta t} \circ S_{\alpha_2 \Delta t} \circ S_{\alpha_3 \Delta t} \circ S_{\alpha_2 \Delta t} \circ S_{\alpha_1 \Delta t}(U^n) \quad (2.20)$$

où les constantes α_i sont définies par :

$$\alpha_1 = \alpha_2 = \frac{1}{4 - \sqrt[3]{4}} \quad \alpha_3 = \frac{1}{1 - 4^{\frac{2}{3}}}$$

Pour rappel, la méthode de Strang est une composition d'ordre 2 en temps, la méthode de Suzuki ainsi construite est une méthode d'ordre 4. On réfère à [9] pour d'autres méthodes de composition basées sur une décomposition en trois parties.

2.4.1.3 Discréétisation de l'espace des phases

Avec les conditions périodiques en espace, il paraît naturel d'effectuer la résolution du système (2.17) en espace grâce aux transformées de Fourier en x . Cette étape sera effectuée par l'algorithme de transformée de Fourier rapide (FFT) qui effectue une transformée de Fourier discrète. Ainsi l'équation sur E et le transport en x de la variable f_h s'effectuent dans l'espace de Fourier discret. Pour le système (2.15) où nous résolvons l'équation de transport en v de la quantité f_h en utilisant le fait que f est constante le long des caractéristiques, nous utiliserons une interpolation à l'aide d'un polynôme par morceaux de Lagrange de degré 5 (voir [11]).

2.4.2 Méthode de Lawson sur le modèle hybride

Nous présentons une seconde approche pour la discréétisation en temps du modèle (2.7), que nous comparerons à la méthode de splitting présentée dans la sous-section 2.4.1. Il s'agit de la méthode de Lawson [60], qui fait partie de la classe des méthodes de type exponentielle.

2.4.2.1 Présentation de la méthode de Lawson

Le système VHL (2.7) s'écrit, pour un mode k après une transformée de Fourier en x sur l'équation de Vlasov sur f_h :

$$\begin{cases} \partial_t u_c = E \\ \partial_t E = -\rho_c^{(0)} u_c - \int_{\mathbb{R}} v f_h \, dv \\ \partial_t \hat{f}_h + ikv \hat{f}_h + \widehat{E \partial_v f_h} = 0 \end{cases} \quad (2.21)$$

où $\hat{f}_h := \hat{f}_h(t, k, v)$ désigne la transformée de Fourier de $f_h(t, x, v)$ par rapport à x , k étant la variable de Fourier. Ce modèle peut se réécrire sous la forme suivante :

$$\partial_t \begin{pmatrix} u_c \\ E \\ \hat{f}_h \end{pmatrix} + \begin{pmatrix} 0 & -1 & 0 \\ \rho_c^{(0)} & 0 & 0 \\ 0 & 0 & ikv \end{pmatrix} \begin{pmatrix} u_c \\ E \\ \hat{f}_h \end{pmatrix} + \begin{pmatrix} 0 \\ \int_{\mathbb{R}} v f_h \, dv \\ \widehat{E \partial_v f_h} \end{pmatrix} = 0.$$

On pose $U = (u_c, E, \hat{f}_h)^T$, ainsi que :

$$A = \begin{pmatrix} 0 & -1 & 0 \\ \rho_c & 0 & 0 \\ 0 & 0 & ikv \end{pmatrix}, \quad N(U) = \begin{pmatrix} 0 \\ \int_{\mathbb{R}} v f_h \, dv \\ \widehat{E \partial_v f_h} \end{pmatrix}$$

pour écrire (2.21) sous une forme plus compacte suivante :

$$\partial_t U + AU + N(U) = 0.$$

Cette formulation est propice à l'utilisation d'intégrateurs exponentiels dont le point de départ est la réécriture suivante

$$\partial_t(e^{tA}U) + e^{tA}N(U) = 0.$$

Puis, en effectuant le changement d'inconnue $V = e^{tA}U$ et avec $\tilde{N} : (t, V) \mapsto e^{tA}N(e^{-tA}V)$, on peut écrire :

$$\partial_t V + \tilde{N}(t, V) = 0$$

Cette équation peut se résoudre numériquement avec une méthode de type Runge-Kutta. Cette méthode Runge-Kutta sur V peut se réécrire en méthode sur U , la méthode ainsi obtenue sur U est appelée méthode de Lawson induite par la méthode Runge-Kutta choisie, présentée initialement dans [61].

Ainsi, à partir d'une méthode de Runge-Kutta explicite⁴ définie par un tableau de Butcher

0			
c_2	$a_{2,1}$		
\vdots	\vdots \ddots		
c_s	$a_{s,1}$	\cdots	$a_{s,s-1}$
	b_1	\cdots	b_{s-1} b_s

on peut écrire le schéma sur V

$$V^{(i)} = v^n + \Delta t \sum_j a_{ij} \tilde{N}(t^n + c_j \Delta t, V^{(j)})$$

$$V^{n+1} = v^n + \Delta t \sum_i b_i \tilde{N}(t^n + c_i \Delta t, V^{(i)})$$

avec la convention $V^{(0)} = V^n$. Exprimé avec la variable U le schéma s'écrit alors :

$$U^{(s)} = e^{c_s \Delta t A} U^n + \Delta t \sum_{j=0}^{s-1} a_{s,j} e^{-(c_j - c_s) \Delta t A} N(U^{(j)}),$$

$$U^{n+1} = e^{\Delta t A} U^n + \Delta t \sum_{i=0}^{s-1} b_i e^{(1 - c_i) \Delta t A} N(U^{(i)})$$

4. Nous ne nous intéresserons ici qu'à des méthodes Runge-Kutta explicites, ce qui explique que le tableau de Butcher est triangulaire strictement inférieur, ce choix est fait pour des raisons de résolution numérique, en effet nous souhaitons mettre en place des méthodes d'ordre élevé au plus faible coût de calcul possible.

Pour un comparatif d'ordre équivalent à celui de la méthode de splitting présentée dans la sous-section 2.4.1, la méthode de Lawson que nous choisissons est la méthode de Lawson sous-jacente à la méthode Runge-Kutta d'ordre 4 : $RK(4, 4)$:

$$\begin{array}{c|ccccc} & 0 & & & & \\ \frac{1}{2} & & \frac{1}{2} & & & \\ \frac{1}{2} & & 0 & \frac{1}{2} & & \\ 1 & & 0 & 0 & 1 & \\ \hline 1 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & \end{array}$$

dont le schéma est :

$$\begin{aligned} U^{(1)} &= e^{\frac{\Delta t}{2}A}U^n + \frac{\Delta t}{2}e^{\frac{\Delta t}{2}A}N(U^n) \\ U^{(2)} &= e^{\frac{\Delta t}{2}A}U^n + \frac{\Delta t}{2}N(U^{(1)}) \\ U^{(3)} &= e^{\Delta t A}U^n + \Delta t e^{\frac{\Delta t}{2}A}N(U^{(2)}) \\ U^{n+1} &= -\frac{1}{3}e^{\Delta t A}U^n + \frac{1}{3}e^{\frac{\Delta t}{2}A}U^{(1)} + \frac{2}{3}e^{\frac{\Delta t}{2}A}U^{(2)} + \frac{1}{3}U^{(3)} + \frac{\Delta t}{6}N(U^{(3)}) \end{aligned}$$

Les méthodes de Lawson sont particulièrement intéressantes dans notre cadre car l'exponentielle de la matrice A est connue explicitement et peut donc être calculée très efficacement

$$e^{tA} = \begin{pmatrix} \cos(\sqrt{\rho_c^{(0)}}t) & -\frac{\sin(\sqrt{\rho_c^{(0)}}t)}{\sqrt{\rho_c^{(0)}}} & 0 \\ \sqrt{\rho_c^{(0)}} \sin(\sqrt{\rho_c^{(0)}}t) & \cos(\sqrt{\rho_c^{(0)}}t) & 0 \\ 0 & 0 & e^{ikvt} \end{pmatrix}.$$

2.4.2.2 Discrétisation spatiale

Il est maintenant nécessaire de présenter les méthodes de discrétisation dans l'espace des phases. Dans le modèle (2.21) que nous résolvons il n'y a que l'équation de Vlasov qui présente des dérivées spatiales. La dérivée spatiale dans la direction x , symbolisée par le $ikv\hat{f}_h$, sera approchée par une méthode pseudo-spectrale faisant appel en pratique à l'algorithme de transformée de Fourier rapide (FFT). La dérivée dans la direction v , $E\partial_v f_h$, nécessite une méthode d'ordre élevé pour bien capturer la filamentation produite dans les solutions du modèle de Vlasov-Poisson ou Vlasov-Ampère. Nous utilisons pour

ce fait la méthode WENO (*Weighted Essentially Non-oscillatory*) d'ordre 5 [74],[63],[88]. Cette méthode se présente comme suit (voir aussi [19]) :

$$\partial_t \hat{f}_{h,k,\ell} + v_{\ell} i k \hat{f}_{h,k,\ell} + \left(E^+ \frac{\widehat{f_{h \cdot, \ell+1/2}^+} - f_{h \cdot, \ell-1/2}^+}{\Delta v} \right)_k + \left(E^- \frac{\widehat{f_{h \cdot, \ell+1/2}^-} - f_{h \cdot, \ell-1/2}^-}{\Delta v} \right)_k = 0$$

où $\hat{f}_{h,k,\ell} \approx \hat{f}_h(k, v_\ell)$, $v_\ell = \ell \Delta v + v_{\min}$, $E^+ = \max(E, 0)$, $E^- = \min(E, 0)$ et $f_{h \cdot, \ell \pm 1/2}^\pm$ représente le flux numérique donné par la méthode de WENO5.

La méthode WENO est une famille de schémas volumes finis non-linéaires ayant une interprétation en tant que méthode aux différences finies. La méthode consiste à utiliser 3 interpolations pondérées par des poids non-linéaires issus des approximations des dérivées successives de f . L'écriture des poids s'effectue comme suit :

$$\begin{aligned} \beta_0^+ &= \frac{13}{12} \underbrace{(f_{h,i,j-2}^+ - 2f_{h,i,j-1}^+ + f_{h,i,j}^+)^2}_{\Delta x^2(f_{h,i,j}'' + \mathcal{O}(\Delta x))} + \frac{1}{4} \underbrace{(f_{h,i,j-2}^+ - 4f_{h,i,j-1}^+ + 3f_{h,i,j}^+)^2}_{2\Delta x(f_{h,i,j}' + \mathcal{O}(\Delta x^2))} \\ \beta_1^+ &= \frac{13}{12} \underbrace{(f_{h,i,j-1}^+ - 2f_{h,i,j}^+ + f_{h,i,j+1}^+)^2}_{\Delta x^2(f_{h,i,j}'' + \mathcal{O}(\Delta x^2))} + \frac{1}{4} \underbrace{(f_{h,i,j-1}^+ - f_{h,i,j+1}^+)^2}_{2\Delta x(f_{h,i,j}' + \mathcal{O}(\Delta x^2))} \\ \beta_2^+ &= \frac{13}{12} \underbrace{(f_{h,i,j}^+ - 2f_{h,i,j+1}^+ + f_{h,i,j+2}^+)^2}_{\Delta x^2(f_{h,i,j}'' + \mathcal{O}(\Delta x))} + \frac{1}{4} \underbrace{(3f_{h,i,j}^+ - 4f_{h,i,j+1}^+ + f_{h,i,j+2}^+)^2}_{-2\Delta x(f_{h,i,j}' + \mathcal{O}(\Delta x^2))} \end{aligned}$$

et de manière similaire :

$$\begin{aligned} \beta_0^- &= \frac{13}{12} (f_{h,i,j+1}^- - 2f_{h,i,j+2}^- + f_{h,i,j+3}^-)^2 + \frac{1}{4} (3f_{h,i,j+1}^- - 4f_{h,i,j+2}^- + f_{h,i,j+3}^-)^2 \\ \beta_1^- &= \frac{13}{12} (f_{h,i,j}^- - 2f_{h,i,j+1}^- + f_{h,i,j+2}^-)^2 + \frac{1}{4} (f_{h,i,j}^- - f_{h,i,j+2}^-)^2 \\ \beta_2^- &= \frac{13}{12} (f_{h,i,j-1}^- - 2f_{h,i,j}^- + f_{h,i,j+1}^-)^2 + \frac{1}{4} (f_{h,i,j-1}^- - 4f_{h,i,j}^- + 3f_{h,i,j+1}^-)^2 \end{aligned}$$

Ce qui nous permet de calculer les poids définis par :

$$\alpha_i^\pm = \frac{\gamma_i}{(\varepsilon + \beta_i^\pm)^2}, \quad i = 0, 1, 2$$

où ε est un paramètre numérique pour assurer la non nullité du dénominateur, il sera pris à 10^{-6} ; et avec $\gamma_0 = \frac{1}{10}$, $\gamma_1 = \frac{6}{10}$ et $\gamma_2 = \frac{3}{10}$. La normalisation des poids s'effectue comme

suit :

$$w_i^\pm = \frac{\alpha_i^\pm}{\sum_m \alpha_m^\pm}, \quad i = 0, 1, 2$$

Nous pouvons ensuite calculer les flux numériques pour WENO5 [74], donnés par :

$$\begin{aligned} \hat{f}_{i,j+\frac{1}{2}}^+ &= w_0^+ \left(\frac{2}{6} f_{h,i,j-2}^+ - \frac{7}{6} f_{h,i,j-1}^+ + \frac{11}{6} f_{h,i,j}^+ \right) + w_1^+ \left(-\frac{1}{6} f_{h,i,j-1}^+ + \frac{5}{6} f_{h,i,j}^+ + \frac{2}{6} f_{h,i,j+1}^+ \right) \\ &\quad + w_2^+ \left(\frac{2}{6} f_{h,i,j}^+ + \frac{5}{6} f_{h,i,j+1}^+ - \frac{1}{6} f_{h,i,j+2}^+ \right) \end{aligned}$$

et

$$\begin{aligned} \hat{f}_{i,j+\frac{1}{2}}^- &= w_2^- \left(-\frac{1}{6} f_{h,i,j-1}^- + \frac{5}{6} f_{h,i,j}^- + \frac{2}{6} f_{h,i,j+1}^- \right) + w_1^- \left(\frac{2}{6} f_{h,i,j}^- + \frac{5}{6} f_{h,i,j+1}^- - \frac{1}{6} f_{h,i,j+2}^- \right) \\ &\quad + w_0^- \left(\frac{11}{6} f_{h,i,j+1}^- - \frac{7}{6} f_{h,i,j+2}^- + \frac{2}{6} f_{h,i,j+3}^- \right) \end{aligned}$$

La méthode WENO5 prend la forme finale :

$$\partial_v f_h(x_i, v_j) \approx \frac{1}{\Delta v} \left[\left(\hat{f}_{i,j+\frac{1}{2}}^+ - \hat{f}_{i,j-\frac{1}{2}}^+ \right) + \left(\hat{f}_{i,j+\frac{1}{2}}^- - \hat{f}_{i,j-\frac{1}{2}}^- \right) \right]$$

2.4.3 Méthode de pas de temps adaptatif

Nous terminons cette section en présentant des méthodes de pas adaptatifs qui seront incorporées aux intégrateurs en temps précédents. Ce type d'approche est importante lorsqu'on souhaite effectuer des simulations dédiées à la physique des plasmas. En effet, lors d'instabilités, une phase linéaire peut être décrite à l'aide de grands pas de temps alors que dans la phase non linéaire, de petits pas de temps sont nécessaires pour capturer les phénomènes physiques complexes.

Pour une équation différentielle scalaire donnée $du(t)/dt = f(t, u(t))$, $u(0) = u_0 \in$, une méthode à pas de temps adaptatif consiste à effectuer 2 estimations numériques de la solution $u(t^{n+1})$ au temps t^{n+1} . On note Δt^n le pas de temps utilisé pour calculer $u_{[p]}^{n+1}$ et $u_{[p+1]}^{n+1}$ telles que :

$$u_{[p]}^{n+1} = u(t^{n+1}) + \mathcal{O}((\Delta t^n)^{p+1}) \quad u_{[p+1]}^{n+1} = u(t^{n+1}) + \mathcal{O}((\Delta t^n)^{p+2})$$

c'est-à-dire que $u_{[p]}^{n+1}$ est d'ordre p et $u_{[p+1]}^{n+1}$ d'ordre $p+1$. On peut alors effectuer une

estimation de l'erreur locale faite sur la solution d'ordre p :

$$L_{[p]}^{n+1} = |u_{[p+1]}^{n+1} - u_{[p]}^{n+1}|. \quad (2.22)$$

Etant donnée une tolérance tol (fixée par l'utilisateur), si l'erreur locale est supérieure à la tolérance alors l'itération est rejetée, on recommence l'itération avec u^n et un nouveau pas de temps Δt^n plus petit. Sinon l'itération est acceptée et $u^{n+1} = u_{[p]}^{n+1}$, car c'est sur l'estimation d'ordre p que l'on contrôle l'erreur, dans la pratique l'approximation d'ordre $p + 1$ est souvent celle qui finalement est conservée.

Pour l'itération suivante, le nouveau pas de temps optimal est calculé par :

$$\Delta t_{\text{opt}} = \sqrt[p]{\frac{\text{tol}}{L_{[p]}^{n+1}}} \Delta t^n \quad (2.23)$$

Il est possible de limiter l'évolution du pas de temps optimal en évitant une trop grande volatilité de celui-ci :

$$\Delta t^{n+1} = \max \left(\frac{1}{2}, \min \left(2, \sqrt[p]{\frac{\text{tol}}{L_{[p]}^{n+1}}} \right) \right) \Delta t^n$$

Les méthodes de pas de temps adaptatifs que nous présenterons ici sont des méthodes multi-étages. Pour limiter le coût de calcul, ces méthodes sont basées sur des intégrateurs d'ordre $p + 1$, auxquels on ajoute une pondération des étages pour dégrader cette solution et construire une méthode d'ordre p . De plus, ayant présenté l'approche dans le cas d'une équation différentielle, nous devons définir une norme en x et v pour donner un sens à l'erreur locale.

2.4.3.1 Méthode de pas de temps adaptatif avec la méthode de Suzuki

Pour utiliser la méthode de *splitting* de Suzuki présentée dans la sous-section 2.4.1 avec une méthode de pas de temps adaptatif [5], on définit les sous-étapes $U^{(m)}$, $m = 1, \dots, 4$,

comme ceci :

$$U_{[4]}^{n+1} = \mathcal{S}_{\Delta t}(U^n) = S_{\alpha_1 \Delta t} \circ S_{\alpha_2 \Delta t} \circ S_{\alpha_3 \Delta t} \circ S_{\alpha_4 \Delta t} \circ \underbrace{S_{\alpha_1 \Delta t}(U^n)}_{U^{(1)}}.$$

$$\underbrace{\quad\quad\quad}_{U^{(2)}}$$

$$\underbrace{\quad\quad\quad}_{U^{(3)}}$$

$$\underbrace{\quad\quad\quad}_{U^{(4)}}$$

On obtient, par pondération des $(U^{(s)})_{s \in \llbracket 1,4 \rrbracket}$ une approximation d'ordre 3 de $U(t^{n+1})$, donnée par :

$$U_{[3]}^{n+1} = -U^n + w_1(U^{(1)} + U^{(4)}) + w_2(U^{(2)} + U^{(3)})$$

avec :

$$w_1 = \frac{g_2(1-g_2)}{g_1(g_1-1)-g_2(g_2-1)} \quad w_2 = 1-w_1$$

où $g_1 = \alpha_1$ et $g_2 = \alpha_1 + \alpha_2$.

Ensuite on effectue l'estimation de l'erreur suivante : $L_{[3]}^{n+1} = \|U_{[4]}^{n+1} - U_{[3]}^{n+1}\|_2$.

La norme que nous utiliserons sur $U^n = (u_c^n, E^n, \hat{f}_h^n)$ pour estimer l'erreur locale est la somme des normes L^2 de chaque variable :

$$L_{[3]}^{n+1} = \left(\sum_i (u_c^{[4]} - u_c^{[3]})^2 \Delta x \right)^{\frac{1}{2}} \\ + \left(\sum_i (E_i^{[4]} - E_i^{[3]})^2 \Delta x \right)^{\frac{1}{2}} \\ + \left(\sum_j \sum_i |f_h^{[4]} - f_h^{[3]}|^2 \Delta x \Delta v \right)^{\frac{1}{2}}, \quad (2.24)$$

où u_c , E et f_h sont les inconnues discrètes associés au point i, j de grille de l'espace des phases.

2.4.3.2 Méthode de pas de temps adaptatif avec la méthode de Lawson

Nous présentons une méthode dite de Runge-Kutta *embedded*, qui est une méthode de pas de temps adaptatif pour les méthodes de Runge-Kutta. La littérature sur le sujet est relativement riche, nous avons voulu ici présenter une méthode du même ordre que la

méthode de Suzuki à pas de temps adaptatif pour effectuer une comparaison entre ces 2 méthodes de résolution. La méthode que nous avons retenue est aussi appelée la méthode de Dormand-Prince 4(3), abrégée en DP4(3) [25],[26]. Cette méthode a pour tableau de Butcher :

	0			
	$\frac{1}{2}$	$\frac{1}{2}$		
	$\frac{1}{2}$	0	$\frac{1}{2}$	
	1	0	0	1
	1	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$
		$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$
		$\left(\frac{1}{6} - \lambda\right)$	λ	

avec λ un paramètre fixé à $\frac{1}{10}$ et où l'estimateur d'ordre 4, $U_{[4]}^{n+1}$, est donné par l'avant dernière ligne, et l'estimateur d'ordre 3, $U_{[3]}^{n+1}$, est donné par la dernière ligne, l'avant dernière ligne se lisant alors comme une ligne classique du tableau de Butcher.

Comme dans l'approche précédente, on calcule l'estimation de l'erreur locale $L_{[3]}^{n+1} = \|U_{[4]}^{n+1} - U_{[3]}^{n+1}\|_2$ dont la définition est la même que (2.24), et on adapte le pas de temps comme expliqué plus haut.

Dans le tableau de Butcher, le paramètre λ peut être optimisé selon certains critères. En effet, si on note $R(\lambda)$ la fonction de stabilité de la méthode d'ordre 3, on obtient

$$R(\lambda) = \frac{\lambda z^5}{24} + z^4 \left(\frac{1}{24} - \frac{\lambda}{12} \right) + \frac{z^3}{6} + \frac{z^2}{2} + z + 1$$

Idéalement $\lambda = 0$ permet d'obtenir une méthode d'ordre 4 (ce qui est déjà effectuée dans l'étage précédent du tableau de Butcher). On cherche donc à trouver le $\lambda \neq 0$ tel que le domaine de stabilité soit le plus large possible ou que ce schéma minimise l'erreur tout en restant d'ordre 3.

2.5 Relations de dispersion

Cette section est dédiée à l'étude des relations de dispersion relatives aux modèles cinétique (2.1)-(2.2) et hybride linéarisé (2.7). Il s'agit de linéariser le modèle étudié puis d'exprimer le mode fondamental du champ électrique linéarisé. Cela permet d'obtenir une très bonne approximation de la phase linéaire de l'énergie électrique. Cette approche, complètement indépendante des schémas numériques utilisés pour résoudre le modèle de

départ, sera utilisée comme outil de validation des codes présentés dans la section 2.4.

Nous allons présenter les relations de dispersion de nos deux modèles, puis nous expliquerons comment reconstruire l'approximation linéaire de l'énergie électrique. Enfin, nous détaillerons les calculs des relations de dispersion pour le cas test qui nous intéressera dans les simulations numériques (sections 2.6 et 2.7).

2.5.1 Relations de dispersion dans le cas cinétique

Nous nous intéressons d'abord aux relations de dispersion du modèle cinétique de Vlasov-Poisson (2.1)-(2.2), en nous appuyant sur [78]. Pour obtenir les relations de dispersion, il est nécessaire de linéariser le système autour d'un équilibre, pour cela rappelons les équations de Vlasov-Poisson (2.1)-(2.2) :

$$\begin{cases} \partial_t f + v \partial_x f + E \partial_v f = 0 \\ \partial_x E = \int_{\mathbb{R}} f \, dv - 1 \\ f(t=0, x, v) = f^0(x, v) \end{cases} \quad (2.25)$$

Dans un premier temps, nous nous intéressons à la linéarisation de ce modèle cinétique autour d'un état d'équilibre donné par $(f(t, x, v))_{eq} = f^{(0)}(v)$ et $(E(t, x))_{eq} = 0$, on considère le développement suivant :

$$\begin{cases} f(t, x, v) = f^{(0)}(v) + \varepsilon f^{(1)}(t, x, v) + \mathcal{O}(\varepsilon^2) \\ E(t, x) = 0 + \varepsilon E^{(1)}(t, x) + \mathcal{O}(\varepsilon^2) \end{cases} \quad (2.26)$$

La densité de particules est définie par $\rho_0 = \rho_{0,c} + \rho_{0,h} = \int f^{(0)} \, dv$. On injecte (2.26) dans (2.25) pour obtenir :

$$\begin{cases} \varepsilon \partial_t f^{(1)} + v \varepsilon \partial_x f^{(1)} + \varepsilon E^{(1)} (\partial_v f^{(0)} + \varepsilon \partial_v f^{(1)}) = \mathcal{O}(\varepsilon^2) \\ \varepsilon \partial_x E^{(1)} = \int f^{(0)} + \varepsilon \int f^{(1)} - n_0 + \mathcal{O}(\varepsilon^2) \end{cases}$$

ce qui nous permet d'obtenir, en négligeant les termes d'ordre ε^2 , le système de Vlasov-Poisson linéarisé :

$$\begin{cases} \partial_t f^{(1)} + v \partial_x f^{(1)} + E^{(1)} \partial_v f^{(0)} = 0 \\ \partial_x E^{(1)} = \int f^{(1)} \, dv \end{cases} \quad (2.27)$$

Pour un état d'équilibre connu $f^{(0)}(v)$, habituellement une distribution gaussienne, les inconnues de (2.27) sont $f^{(1)}(t, x, v)$ et $E^{(1)}(t, x)$.

Nous souhaitons dériver l'expression générale de la relation de dispersion associée au modèle cinétique linéarisé (2.27). Afin de simplifier la lecture, nous supprimons l'index (1) sur nos inconnues $f^{(1)}$ et $E^{(1)}$. Nous supposons que les fonctions $f^{(1)}$ et $E^{(1)}$ sont L -périodiques en x dans le domaine $\Omega = [0, L]$, nous allons, successivement, appliquer une transformée de Fourier en x et une transformée de Laplace en t sur le système (2.27).

Tout d'abord, nous effectuons une transformée de Fourier en x , définie pour une fonction $f(x)$ comme :

$$\hat{f}(k) = \frac{1}{L} \int_0^L f(x) e^{-ikx} dx, \quad k = \frac{2\pi}{L} n, n \in \mathbb{Z}$$

Nous obtenons :

$$\begin{cases} \partial_t \hat{f} + ikv \hat{f} + \hat{E} \partial_v f^{(0)} = 0 \\ ik \hat{E} = \int \hat{f}(t, k, v) dv \end{cases} \quad (2.28)$$

Maintenant, nous utilisons la transformée de Laplace définie pour une fonction $f(t)$ par :

$$\tilde{f}(\omega) = \int_0^{+\infty} f(t) e^{i\omega t} dt$$

et, si elle est définie, la transformée de Laplace inverse est donnée par :

$$f(t) = \frac{1}{2i\pi} \int_{u-i\infty}^{u+i\infty} \tilde{f}(\omega) e^{-i\omega t} d\omega$$

Appliquons la transformée de Laplace à la première équation du système (2.28) :

$$\int_0^{+\infty} \partial_t \hat{f}(t) e^{i\omega t} dt + \int_0^{+\infty} ikv \hat{f}(t) e^{i\omega t} dt + \int_0^{+\infty} \hat{E}(t) \partial_v f^{(0)} e^{i\omega t} dt = 0$$

et en utilisant une intégration par partie dans la première intégrale nous obtenons :

$$\begin{aligned} -\hat{f}(t=0, k, v) - i\omega \int_0^{+\infty} \hat{f}(t) e^{i\omega t} dt + ikv \int_0^{+\infty} \hat{f}(t) e^{i\omega t} dt \\ + \partial_v f^{(0)} \int_0^{+\infty} \hat{E}(t) e^{i\omega t} dt = 0 \end{aligned}$$

et donc :

$$(ikv - i\omega) \tilde{\hat{f}}(\omega, k, v) + \partial_v f^{(0)} \tilde{\hat{E}}(\omega, k) = \hat{f}_0(k, v), \quad (2.29)$$

où $\hat{f}_0(k, v) = \hat{f}(t = 0, k, v)$ correspond à la condition initiale. En appliquant maintenant la transformée de Laplace à la seconde équation de (2.28) nous obtenons :

$$\int_0^{+\infty} ik\hat{E}(t, k)e^{i\omega t} dt = \int_0^{+\infty} \int_{-\infty}^{+\infty} \hat{f}(t, k, v) dv e^{i\omega t} dt$$

ce qui nous donne :

$$\tilde{\hat{E}}(\omega, k) = -\frac{i}{k} \int_{-\infty}^{+\infty} \tilde{\hat{f}}(\omega, k, v) dv \quad (2.30)$$

Maintenant, nous souhaitons injecter l'équation (2.29) dans (2.30). Nous devons prêter attention aux pôles $\omega = kv$. En fait, si $\text{Im}(\omega) > 0$ et pour une fonction analytique $g(v)$, alors l'intégrale $\int_{-\infty}^{+\infty} \frac{g(v)}{ikv - i\omega} dv$ est analytique. Lorsque $\text{Im}(\omega) \leq 0$, nous devons construire un prolongement analytique et remplacer l'intégrale par $\int_{\gamma} \frac{g(v)}{ikv - i\omega} dv$ avec γ un contour ouvert parallèle à l'axe réel à l'infini et qui passe par le pôle $\omega = kv$ (voir [78]). Par la suite, nous utiliserons la notation γ soit pour l'axe réel $]-\infty, +\infty[$ quand $\text{Im}(\omega) > 0$, ou pour un chemin ouvert bien choisi lorsque $\text{Im}(\omega) \leq 0$.

Avec cette notation, le résultat de l'injection de (2.29) dans (2.30) nous donne :

$$\begin{aligned} \tilde{\hat{E}} &= -\frac{i}{k} \int_{\gamma} \frac{1}{ikv - i\omega} \left(\hat{f}_0(k, v) - \partial_v f^{(0)} \tilde{\hat{E}}(\omega, k) \right) dv \\ &= -\frac{1}{k} \int_{\gamma} \frac{\hat{f}_0(k, v)}{kv - \omega} dv + \frac{1}{k} \int_{\gamma} \frac{\partial_v f^{(0)} \tilde{\hat{E}}(\omega, k)}{kv - \omega} dv \\ &= -\frac{1}{k^2} \int_{\gamma} \frac{\hat{f}_0(k, v)}{v - \frac{\omega}{k}} dv + \frac{1}{k^2} \tilde{\hat{E}}(\omega, k) \int_{\gamma} \frac{\partial_v f^{(0)}}{v - \frac{\omega}{k}} dv \end{aligned}$$

donc :

$$\left(1 - \frac{1}{k^2} \int_{\gamma} \frac{\partial_v f^{(0)}}{v - \frac{\omega}{k}} dv \right) \tilde{\hat{E}}(\omega, k) = -\frac{1}{k^2} \int_{\gamma} \frac{\hat{f}_0(k, v)}{v - \frac{\omega}{k}} dv$$

En introduisant :

$$D(k, \omega) = 1 - \frac{1}{k^2} \int_{\gamma} \frac{\partial_v f^{(0)}}{v - \frac{\omega}{k}} dv \quad (2.31)$$

et

$$N(k, \omega) = -\frac{1}{k^2} \int_{\gamma} \frac{\hat{f}_0(k, v)}{v - \frac{\omega}{k}} dv \quad (2.32)$$

nous pouvons définir $\tilde{\hat{E}}(\omega, k)$ comme :

$$\tilde{\hat{E}}(\omega, k) = \frac{N(k, \omega)}{D(k, \omega)}$$

L'équation (2.31) est appelée relation de dispersion du modèle cinétique.

2.5.2 Relations de dispersion dans le cas hybride

Intéressons-nous dans cette section à dériver l'expression générale de la relation de dispersion associée au modèle hybride linéarisé (2.7). Pour cela, nous allons repartir du modèle hybride non-linéaire (2.5) et linéariser à la fois les équations fluides et l'équation cinétique. On injecte le développement (2.6) dans (2.5). Les mêmes calculs que dans la section 2.2 et l'approximation en ε^2 y compris dans l'équation cinétique sur f_h conduisent au modèle

$$\begin{cases} \partial_t u_c^{(1)} = E^{(1)} \\ \partial_t E^{(1)} = -\rho_c^{(0)} u_c^{(1)} - \int v f_h^{(1)} dv \\ \partial_t f_h^{(1)} + v \partial_x f_h^{(1)} + E^{(1)} \partial_v f_h^{(0)} = 0 \end{cases}$$

d'inconnues $E^{(1)}$, $u_c^{(1)}$ et $f_h^{(1)}$ que nous noterons dans la suite, respectivement, E , u_c et f_h , solutions du système hybride linéarisé dans toutes les inconnues

$$\begin{cases} \partial_t u_c = E \\ \partial_t E = -\rho_c^{(0)} u_c - \int v f_h dv \\ \partial_t f_h + v \partial_x f_h + E \partial_v f_h^{(0)} = 0 \end{cases} \quad (2.33)$$

Nous insistons sur le fait que le modèle (2.33) correspond à une linéarisation de la partie cinétique (ou chaude) du modèle hybride (2.7) que nous avons étudié précédemment. Par la suite, nous supposerons que la densité de particules froides $\rho_c^{(0)}$ est une constante (en temps t et espace x) et que la fonction $f_h^{(0)}$ est une fonction paire en v et ne dépend que de cette variable. Nous supposons que les fonctions f_h , E et u_c sont L -périodiques en x sur le domaine spatial $\Omega = [0, L]$ et nous appliquons la transformée de Fourier en x puis une transformée de Laplace en t .

Tout d'abord, nous appliquons la transformée de Fourier en x :

$$\begin{cases} \partial_t \hat{u}_c = \hat{E} \\ \partial_t \hat{E} = -\rho_c^{(0)} \hat{u}_c - \int \hat{f}_h dv \\ \partial_t \hat{f}_h + ikv \hat{f}_h + \hat{E} \partial_v f_h^{(0)} = 0 \end{cases} \quad (2.34)$$

Alors, nous multiplions par $e^{i\omega t}$ et nous intégrons en temps. L'équation sur \hat{u}_c nous donne :

$$\begin{aligned} \int_0^{+\infty} \partial_t \hat{u}_c e^{i\omega t} dt &= \int_0^{+\infty} \hat{E} e^{i\omega t} dt \\ - \int_0^{+\infty} i\omega \hat{u}_c e^{i\omega t} dt - \hat{u}_c(t=0, k) &= \int_0^{+\infty} \hat{E} e^{i\omega t} dt \\ -i\omega \tilde{\hat{u}}_c(\omega, k) - \hat{u}_c(t=0, k) &= \tilde{\hat{E}}(\omega, k) \\ \tilde{\hat{u}}_c(\omega, k) + \frac{1}{i\omega} \tilde{\hat{E}}(\omega, k) &= -\frac{1}{i\omega} \hat{u}_c(t=0, k) \end{aligned} \quad (2.35)$$

Les mêmes opérations sur l'équation sur \hat{E} nous donnent :

$$-i\omega \tilde{\hat{E}}(\omega, k) - \hat{E}(t=0, k) = -\rho_c^{(0)} \tilde{\hat{u}}_c(\omega, k) - \int_{-\infty}^{+\infty} v \tilde{\hat{f}}_h(\omega, k) dv \quad (2.36)$$

Tandis que l'équation sur $\tilde{\hat{f}}_h$ nous donne :

$$\begin{aligned} -i\omega \tilde{\hat{f}}_h(\omega, k, v) - \hat{f}_h(t=0, k, v) + ikv \tilde{\hat{f}}_h(\omega, k, v) + \tilde{\hat{E}}(\omega, k) \partial_v f_h^{(0)}(v) &= 0 \\ \tilde{\hat{f}}_h(\omega, k, v) (ikv - i\omega) &= \hat{f}_h(t=0, k, v) - \tilde{\hat{E}}(\omega, k) \partial_v f_h^{(0)}(v) \\ \tilde{\hat{f}}_h(\omega, k, v) &= -\frac{i}{k} \frac{\hat{f}_h(t=0, k, v)}{v - \frac{\omega}{k}} + \frac{i}{k} \frac{\tilde{\hat{E}}(\omega, k) \partial_v f_h^{(0)}(v)}{v - \frac{\omega}{k}}. \end{aligned} \quad (2.37)$$

Nous injectons l'expression (2.37) dans (2.36) :

$$\begin{aligned} -i\omega \tilde{\hat{E}}(\omega, k) - \hat{E}(t=0, k) &= -\rho_c^{(0)} \tilde{\hat{u}}_c(\omega, k) + \frac{i}{k} \int_{\gamma} v \frac{\hat{f}_h(t=0, k, v)}{v - \frac{\omega}{k}} dv \\ &\quad - \frac{i}{k} \int_{\gamma} v \frac{\tilde{\hat{E}}(\omega, k) \partial_v f_h^{(0)}(v)}{v - \frac{\omega}{k}} dv \end{aligned}$$

soit :

$$\begin{aligned} \tilde{\hat{E}}(\omega, k) \left(1 - \frac{1}{\omega k} \int_{\gamma} v \frac{\partial_v f_h^{(0)}(v)}{v - \frac{\omega}{k}} dv \right) - \frac{\rho_c^{(0)}}{i\omega} \tilde{\hat{u}}_c(\omega, k) &= -\frac{1}{i\omega} \hat{E}(t=0, k) \\ &\quad - \frac{1}{\omega k} \int_{\gamma} v \frac{\hat{f}_h(t=0, k, v)}{v - \frac{\omega}{k}} dv \end{aligned}$$

Nous injectons maintenant l'expression (2.35) dans (2.38) pour obtenir le problème sui-

vant :

$$\begin{aligned}\tilde{\hat{E}}(\omega, k) & \left(1 - \frac{1}{\omega k} \int_{\gamma} v \frac{\partial_v f_h^{(0)}(v)}{v - \frac{\omega}{k}} dv \right) + \frac{\rho_c^{(0)}}{i\omega} \left(\frac{1}{i\omega} \tilde{\hat{E}}(\omega, k) + \frac{1}{i\omega} \hat{u}_c(t=0, k) \right) \\ & = -\frac{1}{i\omega} \hat{E}(t=0, k) - \frac{1}{\omega k} \int_{\gamma} v \frac{\hat{f}_h(t=0, k, v)}{v - \frac{\omega}{k}} dv\end{aligned}$$

soit :

$$\begin{aligned}\tilde{\hat{E}}(\omega, k) & \left(1 - \frac{1}{k^2} \left(\rho_c \frac{k^2}{\omega^2} + \int_{\gamma} \frac{\partial_v f_h^{(0)}(v)}{v - \frac{\omega}{k}} dv \right) \right) \\ & = \frac{\rho_c^{(0)}}{\omega^2} \hat{u}_c(t=0, k) - \frac{1}{i\omega} \hat{E}(t=0, k) - \frac{1}{\omega k} \int_{\gamma} v \frac{\hat{f}_h(t=0, k, v)}{v - \frac{\omega}{k}} dv\end{aligned}$$

Nous introduisons :

$$D(k, \omega) = 1 - \frac{1}{k^2} \left(\rho_c^{(0)} \frac{k^2}{\omega^2} + \int_{\gamma} \frac{\partial_v f_h^{(0)}(v)}{v - \frac{\omega}{k}} dv \right) \quad (2.38)$$

et

$$N(k, \omega) = \frac{\rho_c^{(0)}}{\omega^2} \hat{u}_c(t=0, k) - \frac{1}{i\omega} \hat{E}(t=0, k) - \frac{1}{\omega k} \int_{\gamma} v \frac{\hat{f}_h(t=0, k, v)}{v - \frac{\omega}{k}} dv, \quad (2.39)$$

nous pouvons alors définir $\tilde{\hat{E}}(\omega, k)$ comme :

$$\tilde{\hat{E}}(\omega, k) = \frac{N(k, \omega)}{D(k, \omega)}$$

Remarque 2.3. Comme nous le verrons dans la sous-section suivante, pour retrouver la pente de la partie linéaire de l'énergie électrique, il est suffisant de trouver les racines de $D(k, \omega)$, ou les pôles de $\tilde{\hat{E}}(\omega, k)$. Si seule la pente de la partie linéaire nous intéresse, un autre moyen de la retrouver est de réécrire les équations (2.35)-(2.38) comme le système suivant :

$$\begin{pmatrix} 1 & \frac{1}{i\omega} \\ -\frac{\rho_c}{i\omega} & 1 - \frac{1}{\omega k} \int_{\gamma} v \frac{\partial_v f_h^{(0)}(v)}{v - \frac{\omega}{k}} dv \end{pmatrix} \begin{pmatrix} \tilde{\hat{u}}_c(\omega, k) \\ \tilde{\hat{E}}(\omega, k) \end{pmatrix} = \begin{pmatrix} -\frac{1}{i\omega} \hat{u}_c(0, k) \\ -\frac{1}{i\omega} \hat{E}(0, k) - \frac{1}{\omega k} \int_{\gamma} v \frac{\hat{f}_h(0, k, v)}{v - \frac{\omega}{k}} dv \end{pmatrix} \quad (2.40)$$

Le problème revient alors à trouver les racines du déterminant de ce système, qui s'écrit

$$\begin{aligned} \text{Det}(k, \omega) &= 1 - \frac{1}{\omega k} \int_{\gamma} v \frac{\partial_v f_h^{(0)}(v)}{v - \frac{\omega}{k}} dv - \frac{\rho_c^{(0)}}{\omega^2} \\ &= 1 - \frac{1}{k^2} \left(\rho_c^{(0)} \frac{k^2}{\omega^2} + \int_{\gamma} \frac{\partial_v f_h^{(0)}(v)}{v - \frac{\omega}{k}} dv \right) \end{aligned}$$

On retrouve bien (2.38). La connaissance de (2.39) nous donnera, en plus de la pente, la phase de l'énergie électrique dans sa partie linéaire.

2.5.3 Expression du champ électrique linéarisé

Dans cette sous-section, nous considérons un prolongement analytique continu de $N(k, \omega)$ et $D(k, \omega)$, et nous supposons que la transformée de Laplace et de Fourier de \tilde{E} sont bien définies pour obtenir une approximation du champ électrique linéarisé.

La transformée de Laplace inverse peut être calculée à l'aide du théorème des résidus :

$$\hat{E}(t, k) = \frac{1}{2i\pi} \int_{u-i\infty}^{u+i\infty} \tilde{E}(\omega, k) e^{-i\omega t} d\omega = \sum_j \text{Res}_{\omega=\omega^{k,j}} \left(\tilde{E}(\omega, k) e^{-i\omega t} \right)$$

où $\omega^{k,j}$ sont les pôles de $\tilde{E}(\omega, k)$. Nous rappelons que si $\omega^{k,j}$ est un pôle simple, alors :

$$\begin{aligned} \text{Res}_{\omega=\omega^{k,j}} \left(\tilde{E}(\omega, k) e^{-i\omega^{k,j} t} \right) &= \lim_{\omega \rightarrow \omega^{k,j}} (\omega - \omega^{k,j}) \tilde{E}(\omega, k) e^{-i\omega t} \\ &= \lim_{\omega \rightarrow \omega^{k,j}} (\omega - \omega^{k,j}) \frac{N(k, \omega)}{D(k, \omega)} e^{-i\omega t} \end{aligned}$$

Maintenant, un développement de Taylor de $D(k, \omega)$ nous donne :

$$D(k, \omega) = \underbrace{D(k, \omega^{k,j})}_0 + (\omega - \omega^{k,j}) \frac{\partial D}{\partial \omega}(k, \omega^{k,j}) + \mathcal{O}((\omega - \omega^{k,j})^2)$$

donc, le passage à la limite nous donne :

$$\text{Res}_{\omega=\omega^{k,j}} \left(\tilde{E}(\omega, k) e^{-i\omega^{k,j} t} \right) = \frac{N(k, \omega^{k,j})}{\frac{\partial D}{\partial \omega}(k, \omega^{k,j})} e^{-i\omega^{k,j} t}. \quad (2.41)$$

Remarque 2.4. En fait, pour un k fixé, on obtient une très bonne approximation de

$\hat{E}(t, k)$ (excepté pour des temps courts) en considérant seulement la fréquence principale. Soient les deux racines $\omega^{k,j_0\pm} = \pm\omega_r + i\omega_i$ de $D(k, \omega)$ (où $\omega_r \in \mathbb{R}^+$, $\omega_i \in \mathbb{R}$) qui ont la plus grande partie imaginaire ω_i : pour toute autre racine $\omega^{k,j}$, on a $\text{Im}(\omega^{k,j}) < \omega_i$. Les autres pôles peuvent être négligés. En effet, nous avons :

$$\begin{aligned}\hat{E}(t, k) &= \sum_j C_j e^{-i\omega^{k,j} t} = C_{j_0^+} e^{-i\omega^{k,j_0^+} t} + C_{j_0^-} e^{-i\omega^{k,j_0^-} t} + \sum_{j \neq j_0^\pm} C_j e^{-i\omega^{k,j} t} \\ &= e^{\omega_i t} \left(C_{j_0^+} e^{-i\omega_r t} + C_{j_0^-} e^{i\omega_r t} + \sum_{j \neq j_0^\pm} C_j e^{-i\text{Re}(\omega^{k,j}) t} e^{(\text{Im}(\omega^{k,j}) - \omega_i) t} \right)\end{aligned}$$

et par hypothèse, $\text{Im}(\omega^{k,j}) - \omega_i < 0 \ \forall j \neq j_0^\pm$, nous pouvons conclure que la somme tend vers zéro lorsque $t \rightarrow +\infty$.

Lemme 2.5.1. Si $f^{(0)}(v)$ (respectivement $f_h^{(0)}(v)$) est une fonction paire, alors pour D défini par (2.31) (respectivement (2.38)) nous avons $D(k, \omega_r + i\omega_i) = 0 \Leftrightarrow D(k, -\omega_r + i\omega_i) = 0$.

Démonstration. Voir en Annexe 2.A. (à voir si on le laisse en annexe ou si on l'enlève complètement) \square

En considérant seulement les deux racines principales $\pm\omega_r + i\omega_i$ de $D(k, \omega)$, supposés pôles simples de $\tilde{E}(\omega, k)$, nous avons l'approximation :

$$\hat{E}(t, k) \approx \text{Res}_{\omega=\omega_r+i\omega_i} \left(\tilde{E}(\omega, k) e^{-i\omega t} \right) + \text{Res}_{\omega=-\omega_r+i\omega_i} \left(\tilde{E}(\omega, k) e^{-i\omega t} \right)$$

où les résidus sont définis par (2.41). Notons r^\pm le module de $\frac{N(k, \pm\omega_r + i\omega_i)}{\frac{\partial D}{\partial \omega}(k, \pm\omega_r + i\omega_i)}$ et ϕ^\pm son argument, nous avons alors :

$$\hat{E}(t, k) \approx r^+ e^{i\phi^+} e^{-i(\omega_r + i\omega_i)t} + r^- e^{i\phi^-} e^{-i(-\omega_r + i\omega_i)t}. \quad (2.42)$$

Dans plusieurs cas test classiques, nous avons une symétrie entre les racines, qui dépend de la perturbation initiale de l'équilibre. Par la suite la perturbation initiale de l'équilibre sera toujours une fonction cosinus.

Hypothèse 2.1. Le module et l'argument de $\frac{N(k, \pm\omega_r + i\omega_i)}{\frac{\partial D}{\partial \omega}(k, \pm\omega_r + i\omega_i)}$ vérifient $r^+ = r^-$ (noté r par la suite) et $\phi^+ = -\phi^-$ (noté simplement ϕ).

Sous l'hypothèse 2.1, nous obtenons :

$$\begin{aligned}\hat{E}(t, k) &\approx re^{i\phi}e^{-i(\omega_r+i\omega_i)t} + re^{-i\phi}e^{-i(-\omega_r+i\omega_i)t} \\ &= re^{\omega_i t} \left(e^{i(\omega_r t - \phi)} + e^{-i(\omega_r t - \phi)} \right) \\ &= 2re^{\omega_i t} \cos(\omega_r t - \phi).\end{aligned}\quad (2.43)$$

Maintenant, si nous considérons la définition des coefficients de Fourier, nous avons :

$$\hat{E}(t, k) = \frac{1}{L} \int_0^L E(t, x) e^{-ikx} dx = \frac{1}{L} \int_0^L E(t, x) \cos(-kx) dx + i \frac{1}{L} \int_0^L E(t, x) \sin(-kx) dx$$

et :

$$\begin{aligned}\hat{E}(t, -k) &= \frac{1}{L} \int_0^L E(t, x) e^{ikx} dx = \frac{1}{L} \int_0^L \cos(kx) dx + i \frac{1}{L} \int_0^L E(t, x) \sin(kx) dx \\ &= \frac{1}{L} \int_0^L E(t, x) e^{ikx} dx = \frac{1}{L} \int_0^L \cos(-kx) dx - i \frac{1}{L} \int_0^L E(t, x) \sin(-kx) dx \\ &= \overline{\hat{E}(t, k)}\end{aligned}$$

Hypothèse 2.2. $N(k, \omega) = 0$ si $k \notin \left\{ \pm \frac{2\pi}{L} \right\}$.

Sous l'hypothèse 2.2, avec l'approximation des coefficients de Fourier (qui sont tous réels) données par (2.43) et avec $l = \frac{2\pi}{L}$, nous obtenons l'approximation du champ électrique suivante :

$$\begin{aligned}E(t, x) &\approx \varepsilon E^{(1)}(t, x) \approx \varepsilon \left(\hat{E}(t, l) e^{ikx} + \overline{\hat{E}(t, l)} e^{-ilx} \right) \\ &\approx 2\varepsilon \hat{E}(t, l) \cos(lx) \\ &\approx 4\varepsilon r e^{\omega_i t} \cos(\omega_i t - \phi) \cos\left(\frac{2\pi}{L}x\right)\end{aligned}$$

Ce qui nous permet d'obtenir une approximation de l'énergie électrique, définie par :

$$\begin{aligned}\mathcal{E}(t) &:= \left(\int_0^L E^2(t, x) dx \right)^{\frac{1}{2}} \\ &\approx 4\varepsilon r e^{\omega_i t} |\cos(\omega_i t - \phi)| \left(\int_0^L \cos^2\left(\frac{2\pi}{L}x\right) dx \right)^{\frac{1}{2}} \\ &\approx 2\sqrt{2L} \varepsilon r e^{\omega_i t} |\cos(\omega_i t - \phi)|\end{aligned}\quad (2.44)$$

puisque :

$$\begin{aligned} \int_0^L \cos^2 \left(\frac{2\pi}{L} x \right) dx &= \int_0^L \frac{1}{2} dx + \int_0^L \frac{1}{2} \cos \left(\frac{4\pi}{L} x \right) dx \\ &= \frac{L}{2} + \left[\frac{L}{8\pi} \sin \left(\frac{4\pi}{L} x \right) \right]_0^L = \frac{L}{2} \end{aligned}$$

Remarque 2.5. Il est possible de mener une étude similaire pour une perturbation donnée par une fonction sinus. Nous obtenons alors des résultats similaires en remplaçant dans l'approximation de $\hat{E}(t, k)$, $E(t, x)$ et $\mathcal{E}(t)$ les fonctions cosinus par des fonctions sinus.

Il est à noter que ces approximations ne prennent en compte que les racines dominantes de $D(\frac{2\pi}{L}, \omega)$, les deux ayant la plus grande partie imaginaire. Cette approximation devient valable en temps t suffisamment long.

La partie imaginaire ω_i nous donne le comportement global des coefficients de Fourier du champ électrique, et donc de l'énergie électrique comme une fonction du temps. Nous obtenons un amortissement de l'énergie électrique si $\omega_i < 0$, ou une instabilité si $\omega_i > 0$. Lorsque nous traçons l'énergie électrique en fonction du temps en échelle logarithmique, nous pouvons observer les comportements suivants :

- un amortissement avec un taux $\omega_i < 0$, le taux indiquant la pente globale de l'amortissement,
- quelques oscillations stables, suivies du développement d'une instabilité avec un taux $\omega_i > 0$, jusqu'à la saturation recherchée.

2.5.4 Applications

Dans cette sous-section, nous nous intéressons au calcul de $D(k, \omega)$ pour le modèle cinétique (2.31) ou hybride (2.38), dans le cadre des cas tests qui nous intéressent. Pour le modèle cinétique la distribution initiale est donnée par :

$$f_0(x, v) = \mathcal{M}_{1-\alpha, 0, T_c}(v) + (\mathcal{M}_{\alpha/2, v_0, 1}(v) + \mathcal{M}_{\alpha/2, -v_0, 1}(v)) (1 + \epsilon \cos(kx))$$

avec α la densité de particules chaudes, centrées en $\pm v_0 \in \mathbb{R}$, et les particules froides sont caractérisées par une température T_c , et où l'on note :

$$\mathcal{M}_{\rho, u, T}(v) := \frac{1}{(2\pi T)^{\frac{1}{2}}} \exp \left(-\frac{|v - u|^2}{2T} \right)$$

la distribution maxwellienne de densité ρ , centrée en la vitesse u et de température T . Cette distribution initiale f_0 nous permet de construire une condition initiale compatible pour le modèle hybride, données par la limite $T_c \rightarrow 0$:

$$\begin{aligned} f_{h,0}(x, v) &= \left(\mathcal{M}_{\alpha/2, v_0, 1}(v) + \mathcal{M}_{\alpha/2, -v_0, 1}(v) \right) (1 + \epsilon \cos(kx)) \\ u_{c,0} &= 0 \end{aligned} \quad (2.45)$$

le champ électrique à l'instant initial E_0 est donné par la résolution de l'équation de Poisson avec la condition initiale :

$$\partial_x E_0(x) = (1 - \alpha) + \int_{\mathbb{R}} f_{h,0}(x, v) dv - 1$$

Nous cherchons ensuite les racines en ω de la fonction $D(k, \omega)$ pour k fixé. Celles-ci sont approchées numériquement à l'aide d'une méthode de Newton, la dérivée $\frac{\partial D}{\partial \omega}(k, \omega)$ est alors nécessaire. La racine ayant la plus grande partie imaginaire, dans la pratique nous ne conservons que celle avec une partie réelle positive, nous donne des informations sur l'évolution de l'énergie électrique au cours du temps (taux d'amortissement et taux d'instabilité en échelle logarithmique). De plus, le calcul de $N(k, \omega)$ nous permet d'obtenir plus d'informations sur le mode dominant $\hat{E}(t, k)$ donné par (2.42) dans le cas général, ou par (2.43) sous l'hypothèse 2.1. Nous en déduisons notamment la phase des oscillations de l'énergie électrique dans sa partie linéaire.

2.5.4.1 Quelques propriétés de la fonction de dispersion du plasma

Dans le calcul de $D(k, \omega)$ et $N(k, \omega)$ apparaît la fonction de dispersion du plasma, aussi appelée fonction de Fried-Conte [39] :

$$Z(\xi) := \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}}{z - \xi} dz \quad (2.46)$$

Tout d'abord on rappelle que :

$$Z'(\xi) = -2(1 + \xi Z(\xi)). \quad (2.47)$$

Nous allons maintenant établir quelques propriétés utiles pour vérifier l'hypothèse 2.1 dans différents cas test classiques.

Lemme 2.5.2. La fonction $Z_\alpha^0(\omega) : \omega \in \mathbb{C} \mapsto Z(\alpha\omega) \in \mathbb{C}$, avec $\alpha \in \mathbb{R}$ fixé, est telle que : $Z_\alpha^0(-\bar{\omega}) = -\overline{Z_\alpha^0(\omega)}$.

Lemme 2.5.3. La fonction $Z_{\alpha,\beta}^+(\omega) : \omega \in \mathbb{C} \mapsto Z(\alpha\omega - \beta) + Z(\alpha\omega + \beta) \in \mathbb{C}$, avec $\alpha \in \mathbb{R}$, $\beta \in \mathbb{R}$ fixés, est telle que : $Z_{\alpha,\beta}^+(-\bar{\omega}) = -\overline{Z_{\alpha,\beta}^+(\omega)}$.

Lemme 2.5.4. La fonction $Z_{\alpha,\beta}^-(\omega) : \omega \in \mathbb{C} \mapsto Z(\alpha\omega - \beta) - Z(\alpha\omega + \beta) \in \mathbb{C}$, avec $\alpha \in \mathbb{R}$, $\beta \in \mathbb{R}$ fixés, est telle que : $Z_{\alpha,\beta}^-(-\bar{\omega}) = \overline{Z_{\alpha,\beta}^-(\omega)}$.

La démonstration de ces lemmes est proposée dans l'annexe 2.A.

L'introduction de la fonction Z provient de la nécessité dans les relations de dispersion définies en (2.31)-(2.32) et (2.38)-(2.39) d'intégrer une distribution maxwellienne qui est une distribution gaussienne renormalisée :

$$\mathcal{M}_{\rho,u,T} = \frac{\rho}{\sqrt{2\pi T}} e^{-\frac{(v-u)^2}{2T}}$$

Rappelons le résultat :

$$\partial_v \mathcal{M}_{\rho,u,T}(v) = -\frac{v-u}{T} \mathcal{M}_{\rho,u,T}(v)$$

Ainsi, avant de passer à l'application de ces résultats sur le cas test qui nous intéresse, calculons une intégrale qui intervient dans le calcul de $D(k, \omega)$:

$$\begin{aligned} \int_{\gamma} \frac{\partial_v \mathcal{M}_{\rho,u,T}}{v - \frac{\omega}{k}} dv &= -\frac{\rho}{\sqrt{2\pi T} T} \int_{\gamma} \frac{(v - \frac{\omega}{k} + \frac{\omega}{k} - u)e^{-\frac{(v-u)^2}{2T}}}{v - \frac{\omega}{k}} dv \\ &= -\frac{\rho}{\sqrt{2\pi T} T} \left(\int_{\gamma} e^{-\frac{(v-u)^2}{2T}} dv + \left(\frac{\omega}{k} - u \right) \int_{\gamma} \frac{e^{-\frac{(v-u)^2}{2T}}}{v - \frac{\omega}{k}} dv \right) \end{aligned}$$

Dans la première intégrale, on utilise le changement de variable $w = \frac{v-u}{\sqrt{T}}$, $dw = \frac{dv}{\sqrt{T}}$, dans la seconde intégrale, nous utilisons le changement de variable suivant : $w = \frac{v-u}{\sqrt{2T}}$, $dw = \frac{dv}{\sqrt{2T}}$. Nous obtenons :

$$\begin{aligned} &-\frac{\rho}{\sqrt{2\pi T} T} \left(\int_{\gamma} e^{-\frac{w^2}{2}} \sqrt{T} dw + \left(\frac{\omega}{k} - u \right) \int_{\gamma} \frac{e^{-w^2}}{\sqrt{2T} w + u - \frac{\omega}{k}} \sqrt{2T} dw \right) \\ &= \frac{\rho}{T} \left(1 + \frac{1}{\sqrt{2\pi T}} \left(\frac{\omega}{k} - u \right) \int_{\gamma} \frac{e^{-w^2}}{w - \frac{1}{\sqrt{2T}} \left(\frac{\omega}{k} - u \right)} dw \right) \end{aligned}$$

et enfin nous obtenons :

$$\int_{\gamma} \frac{\partial_v \mathcal{M}_{\rho,u,T}(v)}{v - \frac{\omega}{k}} dv = -\frac{\rho}{T} \left(1 + \frac{1}{\sqrt{2T}} \left(\frac{\omega}{k} - u \right) Z \left(\frac{\frac{\omega}{k} - u}{\sqrt{2T}} \right) \right) \quad (2.48)$$

où Z est la fonction de diffusion de plasma (2.46).

Le calcul de la fonction $N(k, \omega)$ demande l'évaluation d'une intégrale pour laquelle on utilise le changement de variable $w = \frac{v-u}{\sqrt{2T}}$,

$$\begin{aligned} \int_{\gamma} \frac{\mathcal{M}_{\rho,u,T}(v)}{v - \frac{\omega}{k}} dv &= \frac{\rho}{\sqrt{2\pi T}} \int_{\gamma} \frac{e^{-\frac{(v-u)^2}{2T}}}{v - \frac{\omega}{k}} dv \\ &= \frac{\rho}{\sqrt{2\pi T}} \int_{\gamma} \frac{e^{-w^2}}{\sqrt{2T}w + u - \frac{\omega}{k}} \sqrt{2T} dw \\ &= \frac{\rho}{\sqrt{2\pi T}} \int_{\gamma} \frac{e^{-w^2}}{w - \frac{1}{\sqrt{2T}} \left(\frac{\omega}{k} - u \right)} dw \end{aligned}$$

soit :

$$\int_{\gamma} \frac{\mathcal{M}_{\rho,u,T}(v)}{v - \frac{\omega}{k}} dv = \frac{\rho}{\sqrt{2T}} Z \left(\frac{\frac{\omega}{k} - u}{\sqrt{2T}} \right) \quad (2.49)$$

2.5.4.2 Application à la modélisation hybride

La condition initiale du cas test du modèle hybride nous donne comme état d'équilibre (état perturbé) pour les particules chaudes :

$$f_h^{(0)}(v) = \mathcal{M}_{\alpha/2, v_0, 1}(v) + \mathcal{M}_{\alpha/2, -v_0, 1}(v) = \frac{\alpha}{2\sqrt{2\pi}} \left(e^{-\frac{(v-v_0)^2}{2}} + e^{-\frac{(v+v_0)^2}{2}} \right)$$

avec une vitesse des particules chaudes $v_0 \in \mathbb{R}$ fixée et une densité de particules chaudes $\alpha \in \mathbb{R}$. Les particules froides n'étant pas perturbées, l'état d'équilibre est l'état initial caractérisé par une densité $\rho_c^{(0)} = 1 - \alpha$, et une vitesse moyenne $u_c(t = 0, x) = 0$. L'ex-

pression (2.38) nous donne à l'aide de (2.48) :

$$\begin{aligned} D(k, \omega) &= 1 - \frac{1}{k^2} \left((1-\alpha) \frac{k^2}{\omega^2} + \int_{\gamma} \frac{\partial_v f_h^{(0)}(v)}{v - \frac{\omega}{k}} dv \right) \\ &= 1 - \frac{1}{k^2} \left[(1-\alpha) \frac{k^2}{\omega^2} - \frac{\alpha}{2} \left(1 + \frac{1}{\sqrt{2}} \left(\frac{\omega}{k} - v_0 \right) Z \left(\frac{1}{\sqrt{2}} \left(\frac{\omega}{k} - v_0 \right) \right) \right) \right. \\ &\quad \left. - \frac{\alpha}{2} \left(1 + \frac{1}{\sqrt{2}} \left(\frac{\omega}{k} + v_0 \right) Z \left(\frac{1}{\sqrt{2}} \left(\frac{\omega}{k} + v_0 \right) \right) \right) \right]. \end{aligned} \quad (2.50)$$

On dérive $D(k, \omega)$ à l'aide de (2.47) :

$$\begin{aligned} \frac{\partial D(k, \omega)}{\partial \omega} &= 2 \frac{(1-\alpha)}{\omega^3} + \frac{1}{\sqrt{2}k^3} \frac{\alpha}{2} \left[(1 - 2\tilde{\omega}_-^2) Z(\tilde{\omega}_-) \right. \\ &\quad \left. + (1 - 2\tilde{\omega}_+^2) Z(\tilde{\omega}_+) - 2\tilde{\omega}_- - 2\tilde{\omega}_+ \right] \end{aligned} \quad (2.51)$$

où $\tilde{\omega}_{\pm} = \frac{1}{\sqrt{2}} \left(\frac{\omega}{k} \pm v_0 \right)$.

Maintenant, remarquons que :

$$\hat{f}_h(t=0, k, v) = \hat{g}(k) \frac{\alpha}{2\sqrt{2}\pi} \left(e^{-\frac{(v-v_0)^2}{2}} e^{-\frac{(v+v_0)^2}{2}} \right), \quad g(x) = \cos \left(\frac{2\pi}{L} x \right)$$

ce qui nous permet de simplifier ce calcul de $N(k, \omega)$ en utilisant (2.39) et (2.49) :

$$\begin{aligned} N(k, \omega) &= \frac{(1-\alpha)}{\omega^2} \hat{u}(t=0, k) - \frac{1}{i\omega} \hat{E}(t=0, k) - \frac{\hat{g}(k)}{2\omega k} \left(\int_{\gamma} v \frac{\mathcal{M}_{\alpha/2, v_0, 1}}{v - \frac{\omega}{k}} dv + \int_{\gamma} v \frac{\mathcal{M}_{\alpha/2, -v_0, 1}}{v - \frac{\omega}{k}} dv \right) \\ &= \frac{(1-\alpha)}{\omega^2} \hat{u}(t=0, k) - \frac{1}{i\omega} \hat{E}(t=0, k) - \frac{\hat{g}(k)}{2\omega k} \left(\int_{\gamma} \mathcal{M}_{\alpha/2, v_0, 1} dv + \frac{\omega}{k} \int_{\gamma} \frac{\mathcal{M}_{\alpha/2, v_0, 1}}{v - \omega k} dv \right. \\ &\quad \left. + \int_{\gamma} \mathcal{M}_{\alpha/2, -v_0, 1} dv + \frac{\omega}{k} \int_{\gamma} \frac{\mathcal{M}_{\alpha/2, -v_0, 1}}{v - \omega k} dv \right) \\ &= \frac{(1-\alpha)}{\omega^2} \hat{u}(t=0, k) - \frac{1}{i\omega} \hat{E}(t=0, k) - \frac{\hat{g}(k)}{2\omega k} \left[\alpha + \frac{\omega}{k} \frac{\alpha}{2\sqrt{2}} \left(Z \left(\frac{\frac{\omega}{k} - v_0}{\sqrt{2}} \right) \right. \right. \\ &\quad \left. \left. + Z \left(\frac{\frac{\omega}{k} + v_0}{\sqrt{2}} \right) \right) \right] \end{aligned}$$

soit finalement :

$$N(k, \omega) = \frac{(1-\alpha)}{\omega^2} \hat{u}(t=0, k) - \frac{1}{i\omega} \hat{E}(t=0, k) - \frac{\hat{g}(k)}{k^2} \left[\alpha \frac{k}{\omega} + \frac{\alpha}{2\sqrt{2}} \left(Z\left(\frac{\omega}{k} - v_0\right) + Z\left(\frac{\omega}{k} + v_0\right) \right) \right] \quad (2.52)$$

où $\hat{g}(k)$ est donnée par :

$$\hat{g}\left(\frac{2\pi}{L}\right) = \hat{g}\left(-\frac{2\pi}{L}\right) = \frac{1}{2}, \quad \hat{g}(k) = 0, k \notin \left\{-\frac{2\pi}{L}, \frac{2\pi}{L}\right\} \quad (2.53)$$

Lemme 2.5.5. *Sous l'hypothèse $\hat{u}(t=0, k) = 0$, pour $\frac{\partial D(k, \omega)}{\partial \omega}$ donnée par (2.51) et $N(k, \omega)$ par (2.52), l'hypothèse 2.1 est satisfaite.*

La démonstration de ce lemme est effectuée dans l'annexe 2.A. Elle permet de justifier l'écriture (2.43) du mode fondamental du champ électrique linéarisé puis l'approximation (2.44) de l'énergie électrique linéarisée.

2.5.4.3 Application à la modélisation cinétique

La densité de particules initiale de la modélisation cinétique peut se réécrire comme la somme de la densité de particules froides et de la densité de particules chaudes, avec pour les particules froides une simple distribution maxwellienne non perturbée, et pour les particules chaudes une bi-maxwellienne dont l'intégration a déjà été traitée dans le cas hybride :

$$f_0(x, v) = \mathcal{M}_{1-\alpha, 0, T_c}(v) + f_{h,0}(x, v) \quad (2.54)$$

avec $f_{h,0}(x, v)$ donnée par (2.45). L'expression de $D(k, \omega)$ s'obtient à partir de (2.31) et (2.48) :

$$D(k, \omega) = 1 - \frac{1}{k^2} \left[-\frac{1-\alpha}{T_c} \left(1 + \frac{1}{\sqrt{2T_c}} \frac{\omega}{k} Z\left(\frac{1}{\sqrt{2T_c}} \frac{\omega}{k}\right) \right) - \frac{\alpha}{2} \left(1 + \frac{1}{\sqrt{2}} \left(\frac{\omega}{k} - v_0 \right) Z\left(\frac{1}{\sqrt{2}} \left(\frac{\omega}{k} - v_0 \right)\right) \right) - \frac{\alpha}{2} \left(1 + \frac{1}{\sqrt{2}} \left(\frac{\omega}{k} + v_0 \right) Z\left(\frac{1}{\sqrt{2}} \left(\frac{\omega}{k} + v_0 \right)\right) \right) \right] \quad (2.55)$$

Expression que l'on peut dériver et simplifier à l'aide de (2.47) :

$$\frac{\partial D(k, \omega)}{\partial \omega} = \frac{1}{\sqrt{2}k^3} \left[\frac{1-\alpha}{\sqrt{T_c T_c}} \left((1-2\tilde{\omega}_0^2)Z(\tilde{\omega}_0) - 2\tilde{\omega}_0 \right) + \frac{\alpha}{2} \left((1-2\tilde{\omega}_-^2)Z(\tilde{\omega}_-) + (1-2\tilde{\omega}_+^2)Z(\tilde{\omega}_+) - 2\tilde{\omega}_- - 2\tilde{\omega}_+ \right) \right] \quad (2.56)$$

où $\tilde{\omega}_0 = \frac{1}{\sqrt{2T_c}} \frac{\omega}{k}$ et $\tilde{\omega}_\pm = \frac{1}{\sqrt{2}} (\frac{\omega}{k} \pm v_0)$. Maintenant, pour le calcul de $N(k, \omega)$, on remarque que l'on a :

$$\hat{f}(t=0, k, v) = \hat{g}(k) \frac{\alpha}{2\sqrt{2\pi}} \left(e^{-\frac{(v-v_0)^2}{2}} + e^{-\frac{(v+v_0)^2}{2}} \right)$$

avec la fonction $g(x)$ qui vérifie :

$$\hat{g}\left(\frac{2\pi}{L}\right) = \hat{g}\left(-\frac{2\pi}{L}\right) = \frac{1}{2}, \quad \hat{g}(k) = 0, k \notin \left\{-\frac{2\pi}{L}, \frac{2\pi}{L}\right\}$$

ce qui nous permet, en utilisant les équations (2.32) et (2.49) d'obtenir :

$$N(k, \omega) = -\frac{\hat{g}(k)}{k^2} \frac{\alpha}{2\sqrt{2}} \left(Z\left(\frac{\frac{\omega}{k} - v_0}{\sqrt{2}}\right) + Z\left(\frac{\frac{\omega}{k} + v_0}{\sqrt{2}}\right) \right) \quad (2.57)$$

Nous avons donc le lemme suivant.

Lemme 2.5.6. Pour $\frac{\partial D(k, \omega)}{\partial \omega}$ donnée par (2.56) et $N(k, \omega)$ par (2.57), l'hypothèse 2.1 est satisfaite.

La démonstration de ce lemme est effectuée dans l'annexe 2.A. Elle permet de justifier l'écriture (2.43) du mode fondamental du champ électrique linéarisé puis l'approximation (2.44) de l'énergie électrique linéarisée.

2.5.4.4 Consistance des relations de dispersion

Dans les sous-sections précédentes, nous avons obtenu les relations de dispersion des modèles cinétique et VHL correspondant à la condition initiale (2.54). Une première validation va consister à vérifier que les relations de dispersion du modèle cinétique données par (2.55)-(2.56)-(2.57) sont consistantes, quand $T_c \rightarrow 0$, avec les relations de dispersion

du modèle hybride données par (2.50)-(2.51)-(2.52). Pour cela, rappelons que

$$Z(z) = \sqrt{\pi} \exp(-z^2)(i - \operatorname{erfi}(z))$$

et qu'à la limite $z \rightarrow +\infty$, nous avons le développement asymptotique suivant

$$\operatorname{erfi}(z) = -i + \exp(-z^2)/\sqrt{\pi} \left(\frac{1}{z} + \frac{1}{2z^3} + \frac{3}{4z^5} + \mathcal{O}(z^{-7}) \right).$$

Ainsi, nous avons $Z(z) = 2i\sqrt{\pi} \exp(-z^2) - \frac{1}{z} - \frac{1}{2z^3} - \frac{3}{4z^5} + \mathcal{O}(z^{-7})$ ou encore $Z(z) = -\frac{1}{z} - \frac{1}{2z^3} - \frac{3}{4z^5} + \mathcal{O}(z^{-7})$ et donc

$$zZ(z) = -1 - \frac{1}{2z^2} + \mathcal{O}(z^{-4}).$$

Commençons par regarder la consistance en $D(k, \omega)$. Avec $z = \frac{1}{\sqrt{2T_c}} \frac{\omega}{k}$ quand $T_c \rightarrow 0$, le terme correspondant aux particules froides de (2.55) s'écrit

$$\begin{aligned} -\frac{1-\alpha}{T_c} \left(1 + \frac{1}{\sqrt{2T_c}} \frac{\omega}{k} Z \left(\frac{1}{\sqrt{2T_c}} \frac{\omega}{k} \right) \right) &= -\frac{1-\alpha}{T_c} \left(1 - 1 - \frac{1}{2 \left(\frac{1}{\sqrt{2T_c}} \frac{\omega}{k} \right)^2} + \mathcal{O} \left(\left(\frac{1}{\sqrt{2T_c}} \frac{\omega}{k} \right)^{-4} \right) \right) \\ &= (1-\alpha) \frac{k^2}{\omega^2} + \mathcal{O}(T_c). \end{aligned}$$

C'est le terme correspondant à la partie fluide (froide) de (2.50). Les autres termes (venant des particules chaudes) sont les mêmes dans les deux expressions, donc $D(k, \omega)$ donné par le modèle cinétique est consistant, à la limite $T_c \rightarrow 0$, avec celui donné par le modèle hybride (avec un taux $\mathcal{O}(T_c)$). Regardons ensuite la consistance en $\frac{\partial D(k, \omega)}{\partial \omega}$. Les termes venant des particules chaudes sont les mêmes dans les modèles cinétique (2.56) et hybride (2.51). Nous ne nous intéressons qu'aux termes venant des particules froides. De (2.56), nous avons

$$\begin{aligned} &\frac{1}{\sqrt{2}k^3} \frac{1-\alpha}{T_c \sqrt{T_c}} \left(Z(\tilde{\omega}_0) - 2\tilde{\omega}_0^2 Z(\tilde{\omega}_0) - 2\tilde{\omega}_0 \right) \\ &= \frac{1}{\sqrt{2}k^3} \frac{1-\alpha}{T_c \sqrt{T_c}} \left(-\frac{1}{\tilde{\omega}_0} - \frac{1}{2\tilde{\omega}_0^3} + 2\tilde{\omega}_0 + \frac{1}{\tilde{\omega}_0} + \frac{3}{2\tilde{\omega}_0^3} - 2\tilde{\omega}_0 \right) + \mathcal{O}(\tilde{\omega}_0^{-5}) \\ &= \frac{1}{\sqrt{2}k^3} \frac{1-\alpha}{T_c \sqrt{T_c}} \frac{1}{\tilde{\omega}_0^3} + \mathcal{O}(\tilde{\omega}_0^{-5}) \end{aligned}$$

donc pour $\tilde{\omega}_0 = \frac{1}{\sqrt{2T_c}} \frac{\omega}{k}$, nous avons

$$\frac{1}{\sqrt{2}k^3} \frac{1-\alpha}{T_c\sqrt{T_c}} \frac{2T_c\sqrt{2T_c}k^3}{\omega^3} = 2 \frac{1-\alpha}{\omega^3}$$

qui est le terme fluide de (2.51). Regardons enfin la consistance en $N(k, \omega)$. Là encore, les termes venant des particules chaudes sont les mêmes dans les modèles cinétique (2.57) et hybride (2.52). Les termes supplémentaires dans le modèle hybride s'annulent sous l'hypothèse $\hat{u}(t=0, k) = 0$ et avec $\hat{g}(k)$ donné par (2.53) et $\hat{E}(t=0, k)$ obtenu à partir de l'équation de Poisson

$$\begin{aligned}\partial_x E(t=0, x) &= \rho_c(t=0, x) + \int f^h(t=0, x, v) dv - 1 \\ &= (1-\alpha) + \alpha \left(1 + \varepsilon \cos \left(\frac{2\pi}{L} x \right) \right) - 1 \\ &= \alpha \varepsilon \cos \left(\frac{2\pi}{L} x \right)\end{aligned}$$

soit

$$\hat{E}(t=0, k) = -\frac{i\alpha}{2k}, \quad k \in \left\{ -\frac{2\pi}{L}, \frac{2\pi}{L} \right\}, \quad \hat{E}(k) = 0, \quad k \notin \left\{ -\frac{2\pi}{L}, \frac{2\pi}{L} \right\}. \quad (2.58)$$

La consistance du modèle cinétique, à la limite $T_c \rightarrow 0$, vers le modèle hybride est établie sur les relations de dispersion.

2.6 Limite du modèle cinétique vers le modèle hybride

Il s'agit ici d'étudier numériquement la convergence du modèle cinétique (2.1)-(2.2) vers le modèle VHL (2.7), lorsque la température T_c des particules froides tend vers 0. Une première étude de consistance est effectuée sur les relations de dispersion. Une seconde étude, numérique, montre la convergence de différentes quantités obtenues par les schémas proposés dans la section 2.4. Ces deux études complémentaires ont pour but de justifier l'utilisation de la modélisation hybride linéarisée lorsque les particules se répartissent en deux faisceaux : l'un de particules chaudes (rapides) et l'autre de particules froides (de température $T_c \ll 1$, lentes). Pour cela, le modèle cinétique (2.1)-(2.2) sera initialisé avec

$$f^0(x, v) = \mathcal{M}_{1-\alpha, 0, T_c}(v) + (1 + \epsilon \cos(kx)) \left(\mathcal{M}_{\alpha/2, v_0, 1}(v) + \mathcal{M}_{\alpha/2, -v_0, 1}(v) \right) \quad (2.59)$$

avec $k = 0.5$, $v_0 = 3.4$, $\alpha = 0.2$, $x \in [0, L]$, $L = \frac{2\pi}{k} = 2\pi$, $v \in [-v_{\max}, v_{\max}]$ avec $v_{\max} = 12$ et la perturbation des particules chaudes $\epsilon = 10^{-2}$. Le paramètre T_c prendra différentes valeurs selon les résultats que nous souhaitons illustrer. Comme dans la sous-section 2.5.4, on a noté $\mathcal{M}_{\rho,u,T}(v)$ la distribution Maxwellienne :

$$\mathcal{M}_{\rho,u,T}(v) := \frac{1}{(2\pi T)^{\frac{1}{2}}} \exp\left(-\frac{|v - u|^2}{2T}\right)$$

Pour la condition initiale des simulations avec le modèle hybride linéarisé (2.7), nous considérerons :

$$\begin{aligned} u_c(x) &= 0 \\ f_h(x, v) &= (1 + \epsilon \cos(kx)) (\mathcal{M}_{\alpha/2, v_0, 1}(v) + \mathcal{M}_{\alpha/2, -v_0, 1}(v)) \end{aligned} \tag{2.60}$$

où k , v_0 , α et ϵ sont pris identiques au modèle cinétique, le domaine en x et en v reste inchangé. $E(t = 0, x)$ est obtenu en résolvant l'équation de Poisson sur notre condition initiale, comme indiqué dans la proposition 2.2 :

$$\partial_x E(t = 0) = (1 - \alpha) + \int (1 + \epsilon \cos(kx)) (\mathcal{M}_{\alpha/2, v_0, 1}(v) + \mathcal{M}_{\alpha/2, -v_0, 1}(v)) dv - 1$$

Avant une étude plus détaillée, nous donnons un premier aperçu des solutions des deux modèles pour le choix $T_c = 0.05$. Sur la figure 2.1, sont tracées la condition initiale $f^0(x, v)$ du modèle cinétique (gauche), la solution numérique $f(T_f = 300, x, v)$ au temps final du modèle cinétique (milieu) et la solution numérique $f_h(T_f = 300, x, v)$ des particules chaudes pour le modèle hybride ainsi que la vitesse moyenne des particules froides $u_c(T_f = 300, x)$ (courbe cyan) au temps final (droite). La bande jaune correspond à la population de particules froides, absente dans la modélisation hybride. On observe une bonne corrélation des vortex dans l'espace des phases dans la population de particules froides, et une bonne reconstruction de la population de particules chaudes à partir de la vitesse moyenne u_c . De plus, sur la figure 2.2 est tracée l'évolution de l'énergie électrique $\|E(t, \cdot)\|_{L_2}$ au cours du temps pour ces deux modèles avec les mêmes paramètres numériques (échelle semi-logarithmique) pour différentes valeurs de $T_c = 0.05, 0.1, 0.2, 0.4$ pour le modèle cinétique. On observe une convergence de l'énergie électrique du modèle cinétique vers le modèle hybride lorsque T_c tend vers 0. La première observation est que les résultats proches de ceux obtenus par le modèle hybride linéarisé (2.7) sont très proches de ceux obtenus par le modèle cinétique (2.1)-(2.2), ce qui valide la modélisation. La perturbation des particules

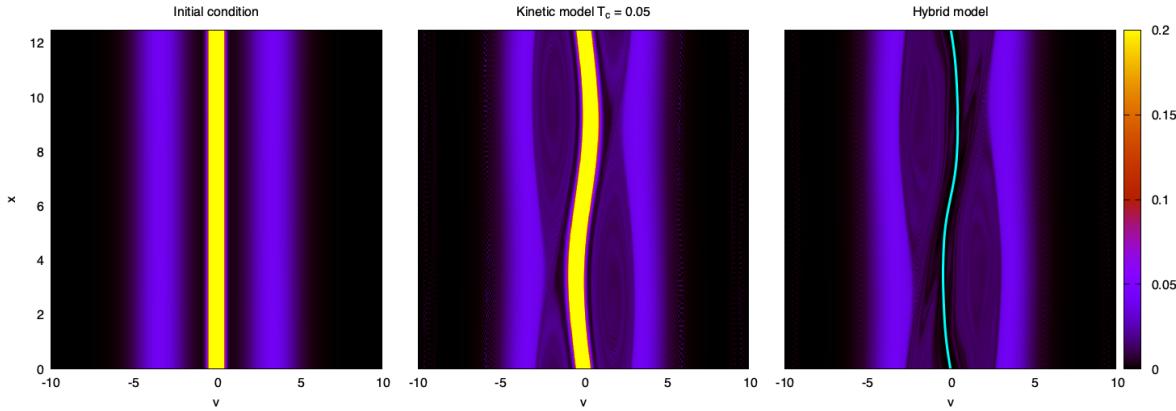


FIGURE 2.1 – Représentation de la condition initiale du modèle cinétique à gauche et la solution obtenue au temps final $T_f = 300$ avec le modèle cinétique avec $T_c = 0.05$ (au milieu) et la densité de particules chaudes obtenue avec modèle hybride linéarisé ainsi que la vitesse moyenne des particules froides (courbe cyan) (à droite).

chaudes induit une instabilité (l'équilibre étant du type double Gaussienne) qu'on voit se développer jusqu'au temps $t = 75$ (voir figure 2.2) et deux vortex sont alors créés autour de la vitesse $v \approx 2$, au centre desquels de fines structures se développent.

Dans la suite, nous allons approfondir cette étude en comparant les résultats obtenus aux relations de dispersion des deux modèles puis en essayant de déterminer le domaine de validité du modèle VHL.

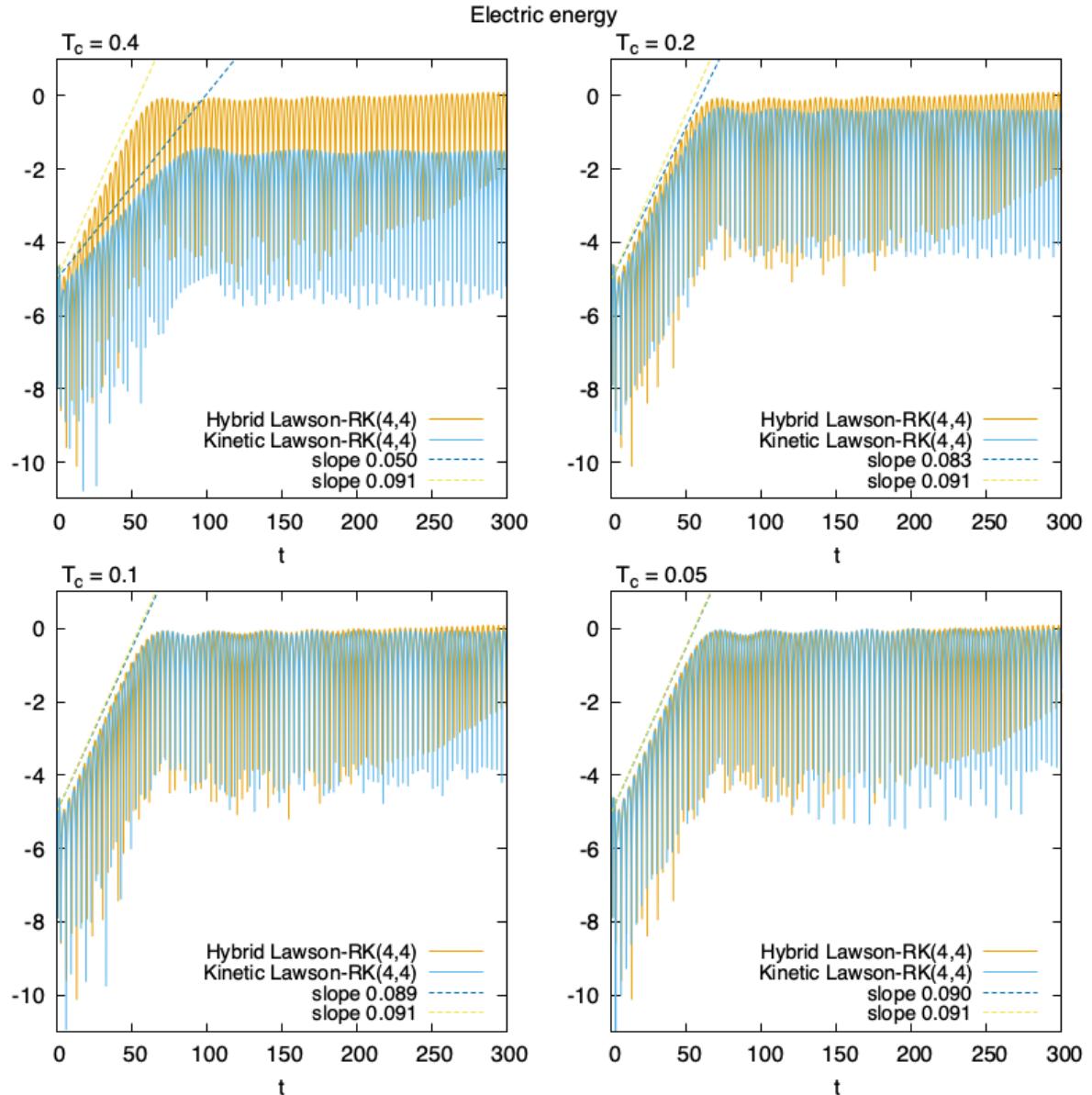


FIGURE 2.2 – Énergie électrique donnée pour le modèle cinétique avec $T_c = 0.05, 0.1, 0.2, 0.4$ et le modèle hybride linéarisé.

2.6.1 Convergence en énergie totale

Nous nous intéresserons ici à une grandeur conservée qu'est l'énergie totale, celle-ci est la somme de l'énergie cinétique et de l'énergie électrique. Pour le modèle cinétique elle se calcule comme :

$$\mathcal{E}_K(t) = \iint_{\Omega \times \mathbb{R}} v^2 f \, dx dv + \int_{\Omega} E^2 \, dx.$$

Pour le modèle VHL, l'énergie cinétique comporte deux termes, un terme fluide pour les particules froides, et un terme cinétique pour les particules chaudes :

$$\mathcal{E}_{VHL}(t) = \int_{\Omega} \rho_c u_c^2 \, dx + \iint_{\Omega \times \mathbb{R}} v^2 f_h \, dx dv + \int_{\Omega} E^2 \, dx$$

Proposition 2.3. *La différence en énergie totale entre le modèle cinétique et le modèle hybride linéarisé pour des conditions initiales données par (2.59) et (2.60) converge en $(1 - \alpha)T_c|\Omega|$.*

Démonstration. Pour le choix de f^0 , l'énergie totale du modèle cinétique vaut :

$$\begin{aligned} \mathcal{E}_K(t) &= \mathcal{E}_K(0) = \iint_{\Omega \times \mathbb{R}} v^2 f^0(x, v) \, dx dv + \int_{\Omega} E^2(t = 0, x) \, dx \\ &= [(1 - \alpha)T_c + \alpha v_0^2 + \alpha] |\Omega| \end{aligned}$$

On remarque que lorsque $T_c \rightarrow 0$, on obtient $\lim_{T_c \rightarrow 0} \mathcal{E}_K(t) = (\alpha v_0^2 + \alpha) |\Omega|$. L'énergie totale dans le cadre du modèle hybride se calcule comme suit :

$$\mathcal{E}_{HL}(t) = \int_{\Omega} \rho_c^{(0)} u_c^2 \, dx + \iint_{\Omega \times \mathbb{R}} v^2 f_h \, dx dv + \int_{\Omega} E^2 \, dx$$

ce qui nous donne, avec le choix de condition initiale $\rho_c^{(0)} = 1 - \alpha$, $u_c^0 = 0$ et $f_h^0(v) = \mathcal{M}_{\alpha/2, v_0, 1}(v) + \mathcal{M}_{\alpha/2, -v_0, 1}(v)$, conformément à (2.60) :

$$\mathcal{E}_{HL}(t) = (\alpha v_0^2 + \alpha) |\Omega|$$

qui est bien compatible avec $\lim_{T_c \rightarrow 0} \mathcal{E}_K(t)$. De plus on peut calculer :

$$\mathcal{E}_K(t) - \mathcal{E}_{HL}(t) = (1 - \alpha)T_c |\Omega|$$

c'est-à-dire que la convergence du modèle hybride est liée à la pression $\rho_c^{(0)}T_c$ des particules froides. \square

Pour vérifier numériquement cette proposition nous effectuons un jeu de simulations. Le modèle cinétique de Vlasov-Poisson (2.1)-(2.2) est simulé à l'aide d'une méthode en temps de type Lawson basée sur une méthode de Runge-Kutta d'ordre 4, la méthode WENO d'ordre 5 pour approcher la dérivée dans la direction v et l'algorithme de FFT pour la dérivée dans la direction x . Il s'agit ainsi du même schéma que celui utilisé pour la fonction de distribution f_h des particules chaudes du modèle hybride linéarisé. Nous choisissons la condition initiale (2.59) avec $\alpha = 0.2$, $T_c \in \{0.05, 0.1, 0.2, 0.4\}$, la discréétisation du domaine $\Omega = [0, 4\pi]$ s'effectue avec $N_x = 135$ points, la discréétisation du domaine en vitesse $[-v_{\max}, v_{\max}]$ nécessite de capturer la gaussienne représentant les particules froides pour différentes valeurs de T_c , nous choisissons donc d'adapter le nombre de points de discréétisation en vitesse N_v à T_c , $N_v \in \{1431, 1011, 715, 505\}$, ceux-ci correspondant à une quinzaine de points de discréétisation pour capturer la gaussienne de température T_c . Nous avons une condition CFL sur le schéma WENO utilisé dans la direction v , nous nous assurons d'être sous cette condition quelle que soit l'évolution de E en prenant $\Delta t = 0.5\Delta v$. Ce jeu de simulations s'arrête au temps 7, or le choix des différents Δt implique des données à des temps différents, nous choisissons d'effectuer une interpolation polynomiale de Lagrange d'ordre 5 pour exploiter les données au temps $T^* = 6.5$. Nous obtenons ainsi l'énergie totale pour différentes températures sur la figure 2.3, bien que le schéma de type Runge-Kutta ne conserve pas exactement l'énergie, celle-ci est bien préservée en temps court et reste sous l'erreur machine en simple précision jusqu'au temps 50. Après l'interpolation au temps $T^* = 6.5$ on obtient la convergence vers le modèle hybride sur la figure 2.3 (figure de droite) où l'on observe bien l'ordre 1 en température.

Nous effectuons le même type d'analyse sur l'énergie électrique, à partir des données des simulations précédentes, en sachant qu'il n'existe pas de résultat théorique sur sa convergence. L'énergie électrique pour les différents choix de T_c est représentée sur la figure 2.4 (gauche), cette figure illustre mieux la nécessité d'effectuer une interpolation pour extraire les données. Une convergence est observée sur la figure 2.4 (droite).

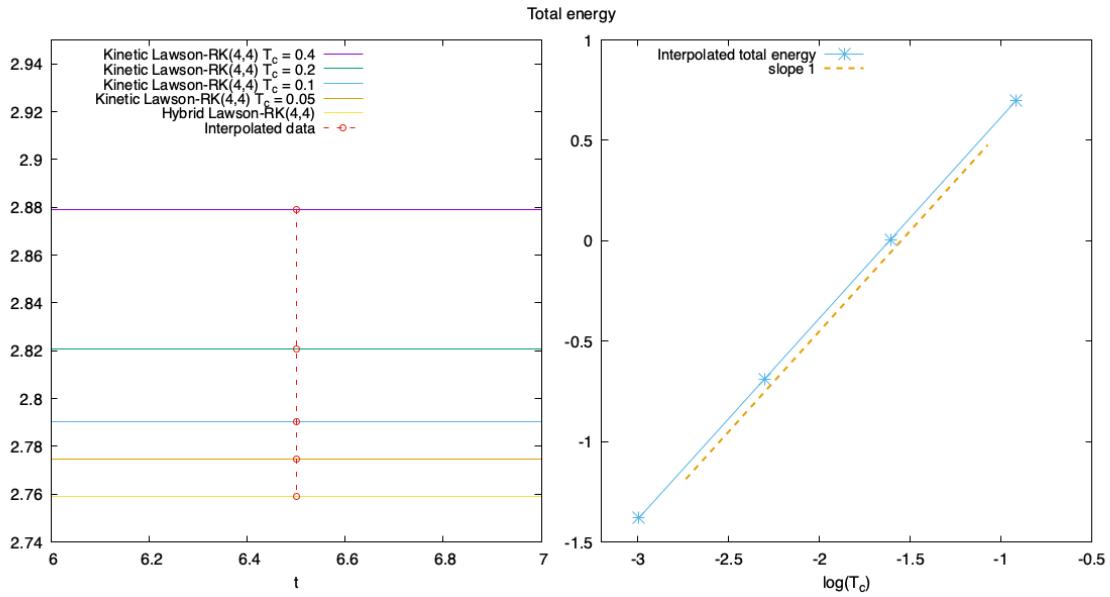


FIGURE 2.3 – Énergie totale avec les différents modèles en échelle semi-log (gauche) et convergence de l'énergie totale du modèle cinétique vers le modèle hybride quand T_c tend vers 0 en échelle log (droite).

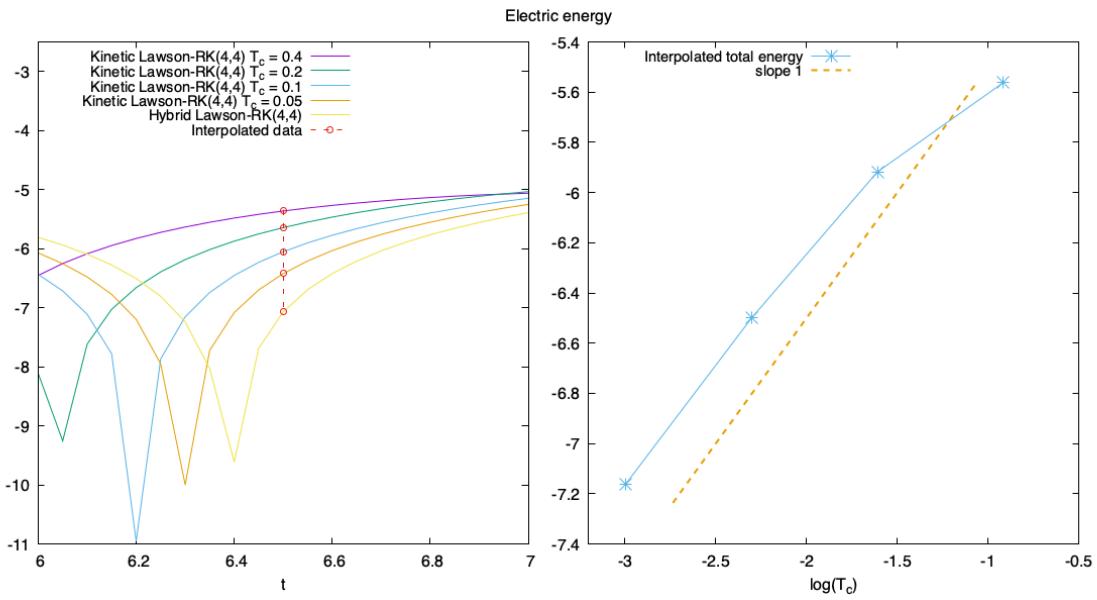


FIGURE 2.4 – Énergie électrique avec les différents modèles en échelle semi-log (gauche) et convergence de l'énergie totale du modèle cinétique vers le modèle hybride quand T_c tend vers 0 en échelle log (droite).

2.6.2 Convergence en température à l'aide des relations de dispersion

Nous étudions numériquement la convergence des racines de la relation de dispersion quant T_c tend vers 0. Pour cela, on note $D_{[T_c]}^K(\omega, k)$ la relation de dispersion du modèle cinétique (2.55) et $D^H(\omega, k)$ la relation de dispersion du modèle VHL (2.50). Pour k fixé, on note $\omega \in \mathbb{C}$ la racine de plus grande partie imaginaire. Cette racine est calculée numériquement à l'aide d'une méthode de Newton. On étudie maintenant la convergence des ω_K (zéro de $(D^K(\omega, k))$) vers ω_H (zéro de $(D^H(\omega, k))$). La convergence de $\omega_K(T_c)$ vers ω_H est visible sur la figure 2.5, où l'on représente, en échelle log-log le module de la différence des 2 zéros $\Delta\omega = |\omega_K - \omega_H|$. On observe une convergence d'ordre 1 en T_c des

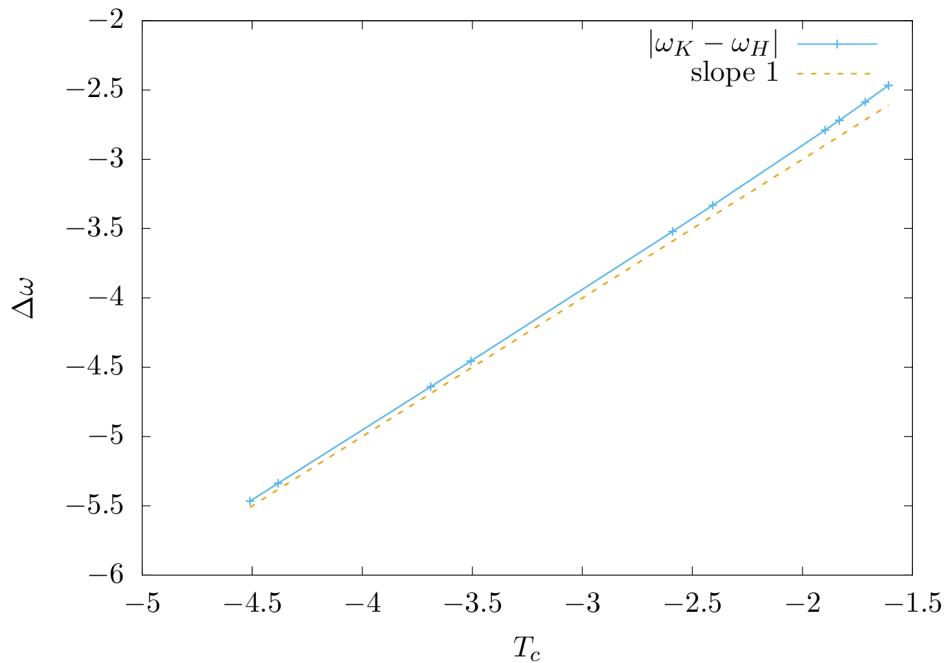
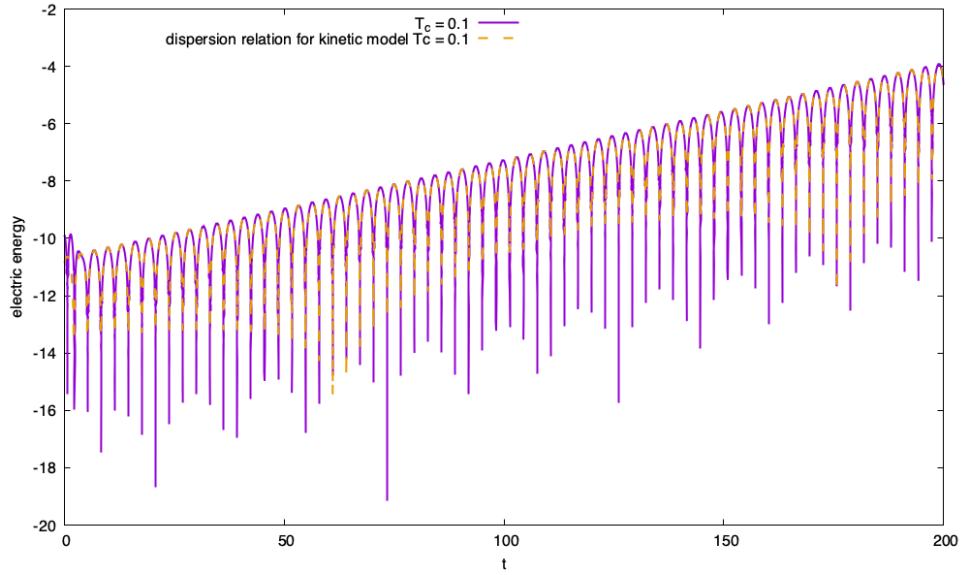


FIGURE 2.5 – Convergence des zéros de la relation de dispersion cinétique vers la solution hybride

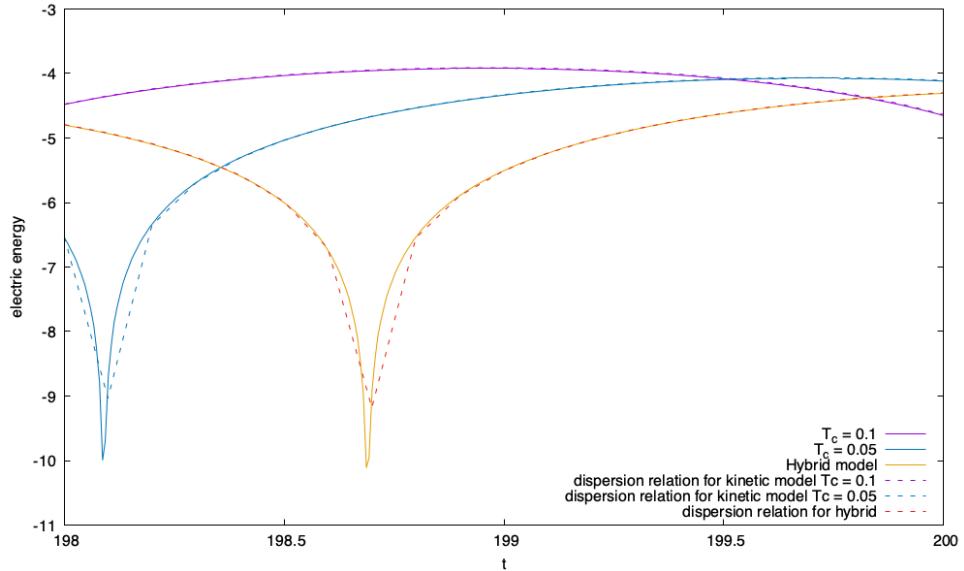
zéros de la relation de dispersion, aucun argument théorique sur les fonctions holomorphes ne vient appuyer ce résultat, contrairement à ce qui a été énoncé pour l'énergie totale.

La racine de plus grande partie imaginaire permettent de valider la phase linéaire du code. Cette phase linéaire peut être rendue plus longue en considérant une valeur très faible de la perturbation $\epsilon = 10^{-4}$ dans les conditions initiales (2.59) et (2.60).

Ceci va nous permettre de vérifier non seulement le taux d'instabilité, mais aussi, grâce aux calculs de la section 2.5 de l'énergie électrique. Nous pouvons donc comparer pour $\alpha = 0.1$, $T_c = 0.1$, $N_x = 135$, $N_v = 1200$, $T_f = 200$ et $\Delta t = 0.5\Delta x$ ce régime linéaire sur la figure 2.6a. Un résultat similaire est observable pour différentes températures ainsi que sur le modèle hybride linéarisé, comme l'illustre la figure 2.6b. Les reconstructions de l'énergie électrique se font à partir des relations de dispersion, avec l'équation (2.44). En plus du taux d'instabilité de l'énergie électrique, il est possible à l'aide des relations de dispersion d'obtenir une très bonne approximation de l'énergie électrique dans la phase linéaire. On peut voir que l'étude des relations de dispersion ne permet pas d'obtenir des résultats fiables en début de simulation, où d'autres modes que le mode principal sont encore visibles (modes évanescents). De même, comme on peut le voir sur la figure 2.2, la phase non-linéaire où l'énergie électrique atteint une saturation mélange de nombreux modes, c qui est incompatible avec l'étude du linéarisé. Néanmoins, même au temps $t \approx 200$, les résultats du code sont en excellent accord avec ceux obtenus grâce aux relations de dispersion (voir figure 2.6b).



(a) Énergie électrique jusqu'au temps 200 avec un régime linéaire très long, et comparaison avec les résultats données par les relations de dispersion.



(b) Énergie électrique entre les temps 198 et 200 pour les températures $T_c = 0.1, 0.05$ et le modèle hybride, et comparaison avec les résultats des relations de dispersion.

FIGURE 2.6 – Évolution de l'énergie électrique dans une longue phase linéaire et comparaison avec les relations de dispersion.

2.6.3 Évolution avec la densité de particules chaudes

Nous avons validé les modèles et les relations de dispersions lorsque la température des particules froides T_c tend vers 0, la proposition 2.3 nous indique que la convergence s'effectue en $(1 - \alpha)T_c|\Omega|$ où α est la densité des particules chaudes. Nous traçons sur la figure 2.7 l'évolution du taux d'instabilité donnée par les relations de dispersions (racine de plus grande partie imaginaire) en fonction de α et pour différentes valeurs de T_c . Cette évolution est représentée pour le modèle cinétique avec différentes températures, et pour le modèle hybride, avec comme condition initiale pour les particules chaudes :

$$f_h^0(x, v) = (\mathcal{M}_{\alpha/2, 4, 1}(v) + \mathcal{M}_{\alpha/2, -4, 1}(v)) (1 + \epsilon \cos(kx)), \quad x \in [0, 4\pi].$$

La condition initiale du modèle cinétique est donnée par : $f^0(x, v) = \mathcal{M}_{1-\alpha, 0, T_c}(v) + f_h^0(x, v)$. On retrouve sur la figure 2.7 la convergence en température du modèle cinétique

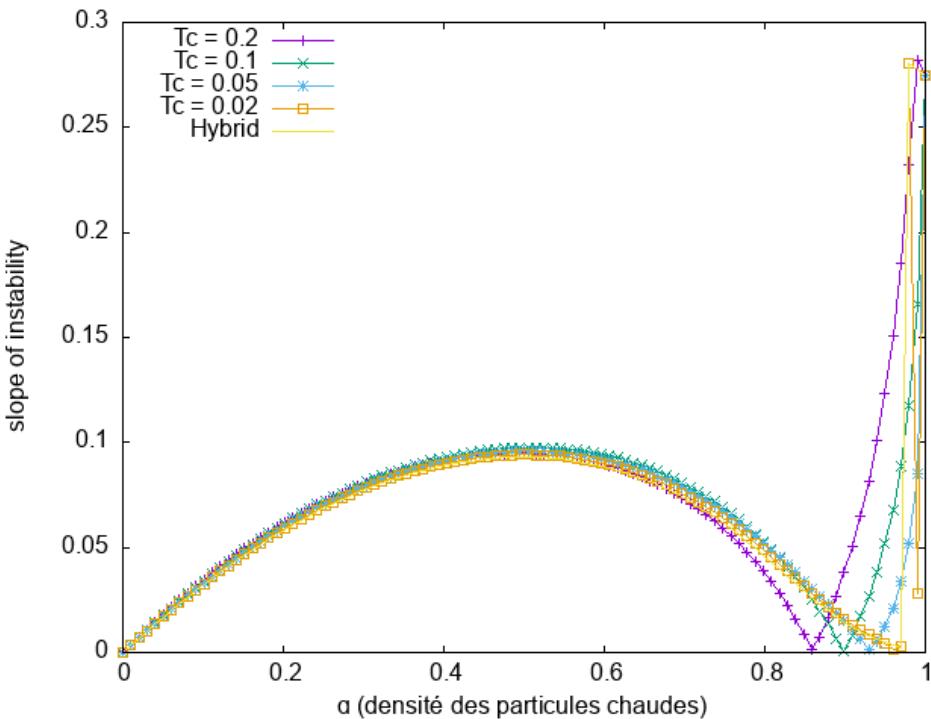


FIGURE 2.7 – Évolution de la pente du développement de l'instabilité (ou taux d'instabilité) donnée par les relations de dispersion en fonction de la densité de particules chaudes α

vers le modèle hybride. Pour $\alpha = 0$ la condition initiale se restreint aux particules froides,

qui ne sont pas perturbées, il est donc normal d'obtenir une pente nulle ; pour $\alpha = 1$, il n'y a que des particules chaudes et on retrouve l'instabilité double faisceaux (TSI) avec le bon taux d'instabilité. On peut enfin observer que pour ce choix de T_c , les taux d'instabilité obtenus restent proches de ceux du modèle VHL pour $0 \leq \alpha \leq 0.5$ (qui correspond à une population identique de particules chaude et froide).

2.7 Comparaison des deux résolutions hybrides

Dans cette section on s'intéressera à la comparaison des méthodes de simulation présentées dans la section 2.4 pour approcher numériquement le modèle VHL. On étudiera en particulier les méthodes de pas de temps adaptatif associées. Nous utilisons dans cette section la condition initiale suivante :

$$u_c(x) = 0$$

$$f_h(x, v) = (\mathcal{M}_{\alpha/2, v_0, 1}(v) + \mathcal{M}_{\alpha/2, -v_0, 1}(v)) (1 + \epsilon \cos(kx))$$

avec $k = 0.5$, $\alpha = 0.2$, $v_0 = 3.4$, $x \in [0, L]$, $L = 4\pi$, $v \in [-12, 12]$, et la perturbation $\epsilon = 10^{-2}$. Le champ électrique initial $E(t = 0, x)$ est obtenu en résolvant l'équation de Poisson sur notre condition initiale, comme indiqué dans la proposition 2.2 :

$$\partial_x E(t = 0) = (1 - \alpha) + \int (1 + \epsilon \cos(kx)) (\mathcal{M}_{\alpha/2, v_0, 1}(v) + \mathcal{M}_{\alpha/2, -v_0, 1}(v)) dv - 1$$

La discrétisation du domaine s'effectue avec $N_x = 27$ dans la direction x , et $N_v = 128$ points dans la direction v .

Nous allons effectuer deux types de comparaisons entre les méthodes de *splitting* hamiltonien et de Lawson, une comparaison à pas de temps fixe, où on illustrera l'absence de condition de CFL des méthodes de *splitting* ; puis une comparaison des méthodes de pas de temps adaptatif associés à ces méthodes présentées dans la section 2.4.3 avec une tolérance $tol = 2 \times 10^{-5}$.

2.7.1 Comparaison des deux résolutions hybrides à pas de temps fixe

Cette section est dédié à la comparaison entre la méthode de *splitting* hamiltonien, présenté dans la sous-section 2.4.1, et la méthode de Lawson présenté dans la sous-

section 2.4.2, pour la résolution du modèle hybride linéarisé (2.7).

Nous considérerons trois pas de temps différents :

- $\Delta t = 0.1 \approx 0.5\Delta v$, il s'agit d'une condition de CFL classique pour des méthodes de volumes finis ;
- $\Delta t = 0.5 \approx \sigma \frac{\Delta v}{\|E^n\|_\infty} = 0.54$, with $\sigma \approx 1.732$, il s'agit de la condition de CFL entre WENO5 et RK(4,4) calculé dans le chapitre précédent ou [19], et calculé à partir d'une précédente estimation numérique $\|E^n\|_\infty = \max_{i,n} |E_i^n| \approx 0.6$;
- $\Delta t = 0.7$, il s'agit dans ce cas d'un cas test avec un pas de temps plus grand que la CFL de la méthode de Lawson, pour illustrer que la méthode de *splitting* n'a pas de contrainte de stabilité sur le pas de temps.

Sur la figure 2.8, on trace l'évolution de l'énergie électrique (en échelle semi-log) calculée par deux méthodes d'ordre 4 (méthode de Suzuki et de Lawson-RK(4,4)), avec différentes valeurs de pas de temps Δt , 0.1, 0.5 et 0.7. On note que toutes les simulations capturent correctement l'énergie électrique dans la phase linéaire, jusqu'au temps 60, même lorsque la méthode est instable. Pour $\Delta t = 0.1, 0.5$, on vérifie la stabilité prévue des méthodes, qui donnent des résultats très similaires. Dans cas $\Delta t = 0.7$, la méthode de Lawson-RK(4,4) devient instable dans la phase non-linéaire, c'est en fait dans cette phase que l'amplitude du champ électrique atteint son maximum, et le paramètre $\Delta t = 0.7$ viole la condition de CFL, alors que la méthode de *splitting* de Suzuki reste stable comme prévu.

Sur la figure 2.9, on observe l'évolution de l'erreur relative sur l'énergie totale, calculée par :

$$\frac{\mathcal{H}^n}{\mathcal{H}^0} - 1 \tag{2.61}$$

avec :

$$\mathcal{H}^n = \frac{1}{2} \iint v^2 f_h^n dv dx + \frac{1}{2} \int \rho_c^{(0)} (u_c^n)^2 dx + \frac{1}{2} \int (E^n)^2 dx .$$

On considère les mêmes paramètres numériques et on compare l'effet de la méthode d'intégration en temps (méthode de *splitting* hamiltonien : Lie (2.18), Strang (2.19) et Suzuki (2.20), et la méthode Lawson-RK(4,4)) sur la préservation de l'énergie totale. On observe que les méthodes géométriques de *splitting* hamiltonien, préservent très bien l'énergie totale, en particulier, l'erreur relative oscille autour d'une constante en temps long, ce qui est un comportement typique d'une méthode géométrique. On observe sur la méthode de Lawson, si le pas de temps est pris sous la condition de CFL, l'erreur est proche de 4%, ce

qui est acceptable. Évidemment, comme évoqué précédemment sur l'énergie électrique, on observe un problème lorsque $\Delta t = 0.7$, l'erreur donnée par la méthode Lawson-RK(4,4) diverge dans la phase non-linéaire, à cause de l'instabilité numérique ; avant cela, l'erreur est autour des 2%. Le tableau 2.1 résume le maximum de l'erreur relative $\max_n |\mathcal{H}^n / \mathcal{H}^0 - 1|$ pour les différentes méthodes et les différents pas de temps considérés.

	0.1	0.5	0.7
Lie	0.0036	0.0187	0.0394
Strang	0.0001	0.0019	0.0109
Suzuki	3×10^{-8}	0.0001	0.0028
Lawson-RK(4,4)	0.0372	0.0331	NaN

TABLE 2.1 – Maximum de l'erreur relative donnée sur la figure 2.9.

Dans le cas $\Delta t = 0.1$, on compare sur la figure 2.10 la distribution de particules chaudes f_h calculée par les méthodes de Suzuki et de Lawson-RK(4,4) au temps $t = 100$, sur laquelle on ajoute la vitesse moyenne des particules froides u_c . Les solutions numériques ainsi obtenues sont très proches, la position des vortex et l'allure de la vitesse moyenne des particules froides sont très similaires entre les deux méthodes. On observe aussi que la méthode Lawson-RK(4,4) introduit plus de diffusion numérique que la méthode de Suzuki, en effet les vortex semblent avoir une meilleure résolution. Cela peut s'expliquer par la discrétisation dans l'espace des phases dans la direction v , une méthode WENO5 (avec limiteurs de pente) est utilisé avec la méthode Lawson-RK(4,4), contre une interpolation d'ordre 5 par des polynômes de Lagrange (sans limiteurs de pente) avec la méthode de Suzuki.

Pour compléter cette étude, on trace sur la figure 2.11 l'ordre en temps des différents intégrateurs en temps utilisés pour résoudre le modèle hybride : les méthodes de *splitting* hamiltonien (Lie, Strang et Suzuki), les méthodes de Lawson (RK(4,4) et RK(3,3)). Pour cela on calcule le maximum de l'erreur relative sur l'énergie totale : $\max_n |\mathcal{H}^n / \mathcal{H}^0 - 1|$, jusqu'au temps $t = 15$, en fonction du pas de temps $\Delta t \in [0.01, 0.125]$, avec $N_x = 243$ et $N_v = 512$. Tous les ordres théoriques des méthodes en temps sont bien reconstruits. On remarque que la constante d'erreur entre la méthode de Suzuki et de Lawson-RK(4,4) sont très proches, mais que cette première est légèrement plus coûteuse que la méthode de Lawson (plus de détails dans la section 2.7.3).

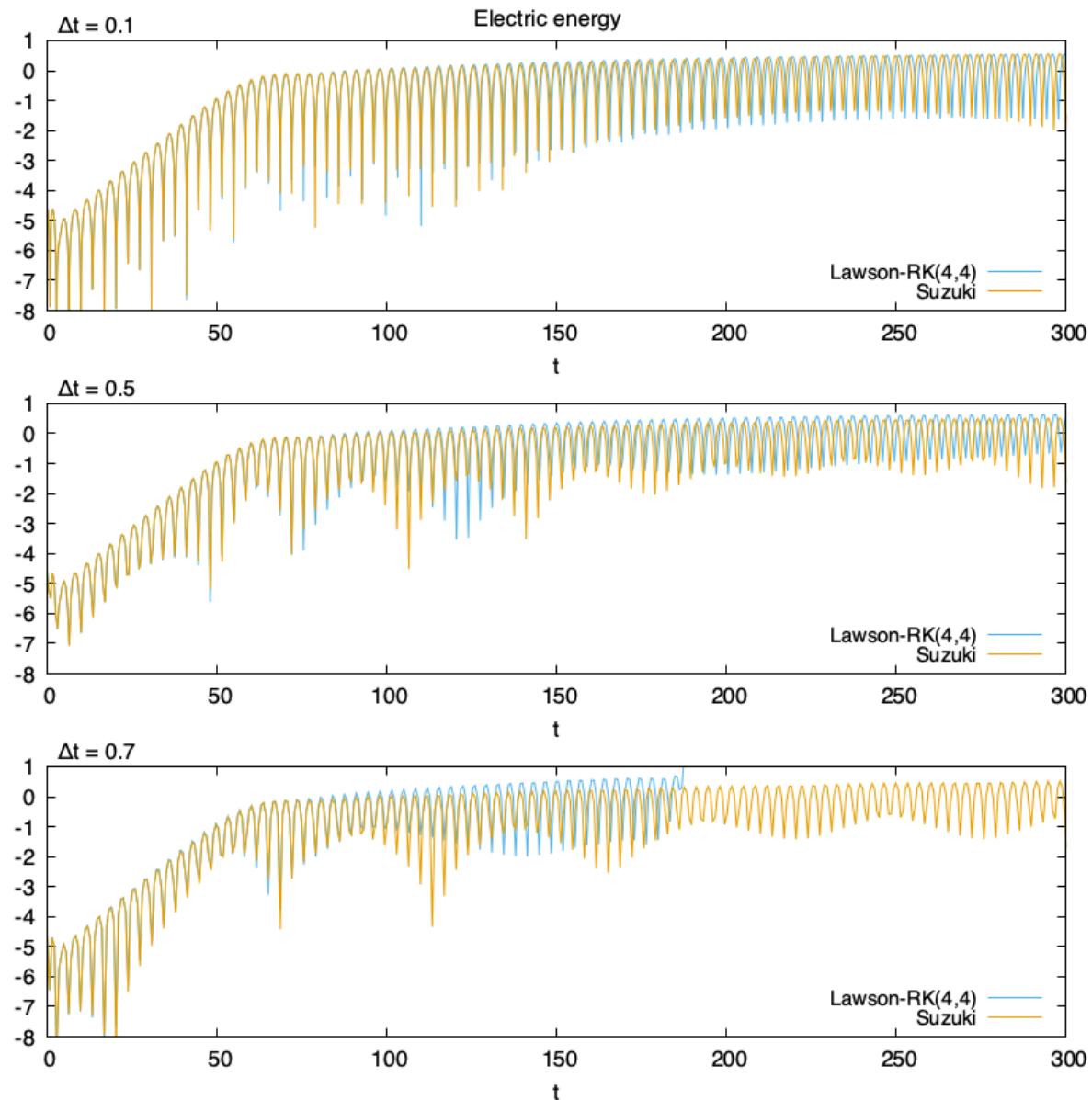


FIGURE 2.8 – Évolution de l'énergie électrique pour le modèle hybride (résolu avec la méthode de Lawson et de Suzuki) pour différentes valeurs de pas de temps $\Delta t = 0.1, 0.5, 0.7$.

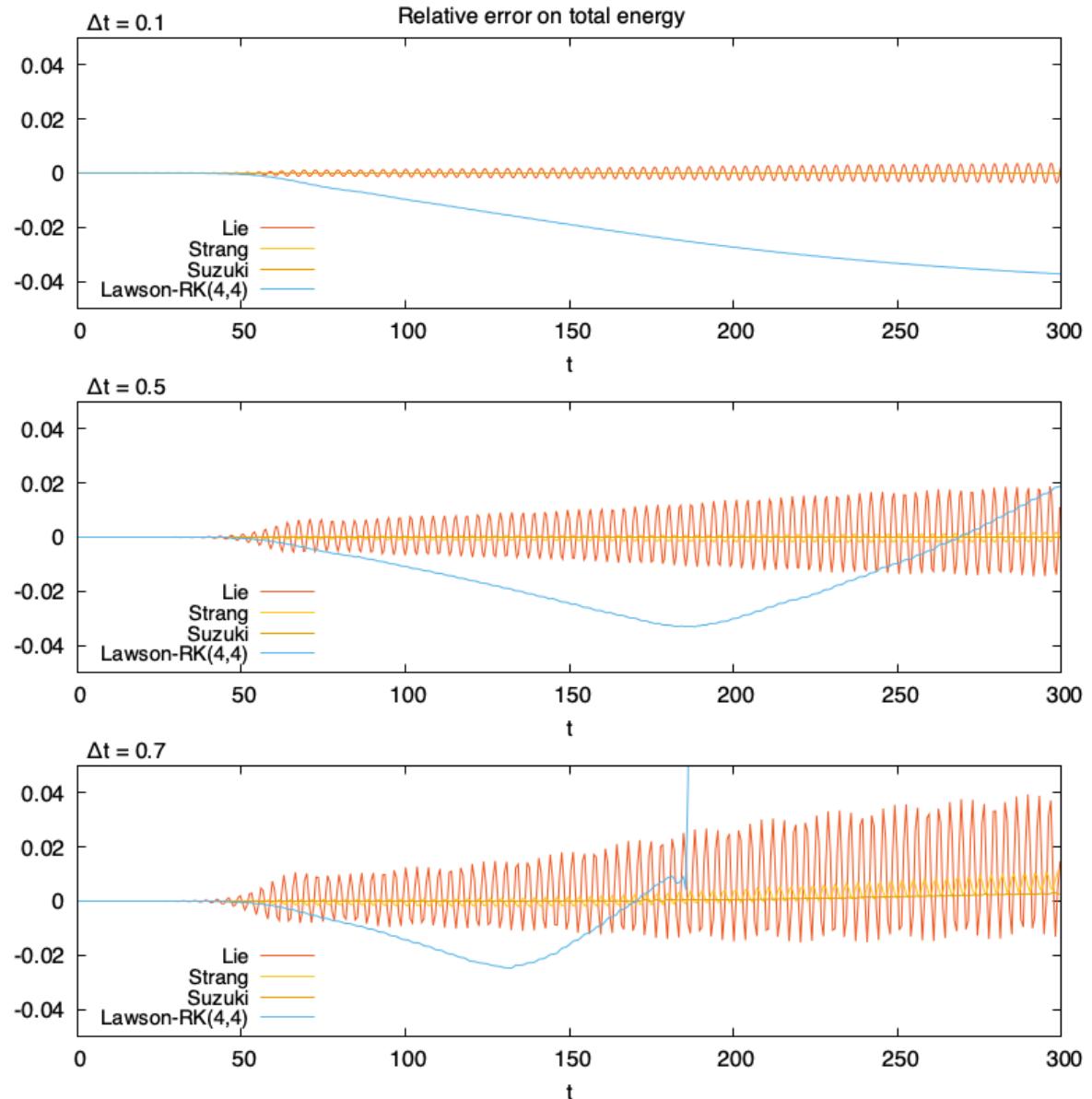


FIGURE 2.9 – Évolution de l'erreur relative sur l'énergie totale pour le modèle hybride (résolu avec la méthode de Lawson et de Suzuki) pour différentes valeurs de pas de temps $\Delta t = 0.1, 0.5, 0.7$.

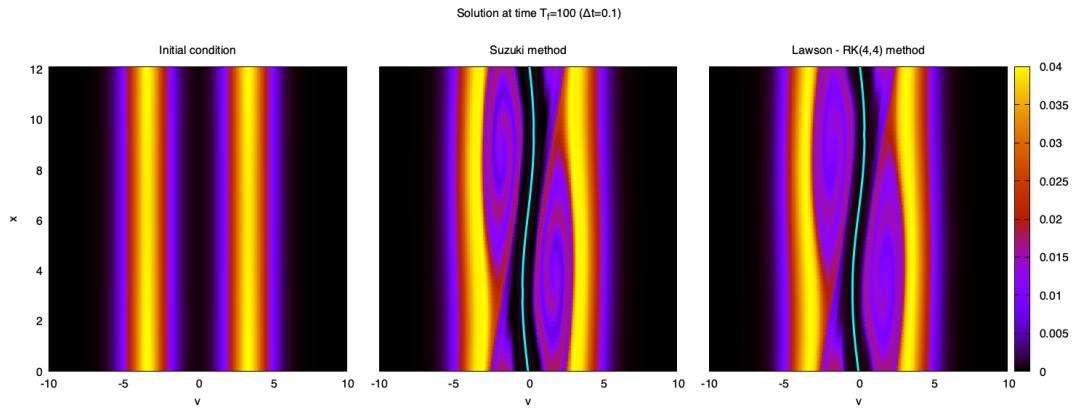


FIGURE 2.10 – Densité des particules chaudes f_h et vitesse moyenne des particules froides u_c (en cyan) la condition initiale (gauche), au temps $t = 100$ calculé par la méthode de Suzuki (milieu) et calculé par la méthode de Lawson-RK(4,4) (droite).

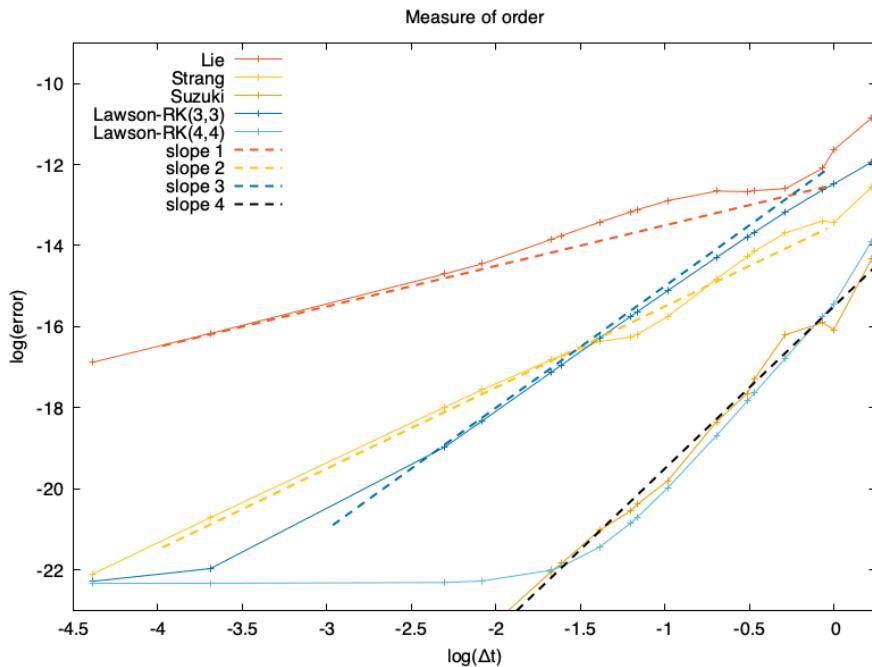


FIGURE 2.11 – Étude de l'ordre en temps des différentes méthodes numériques pour résoudre le modèle hybride (méthodes de Lawson et de *splitting*). L'erreur est calculée à partir du maximum de l'erreur relative sur l'énergie totale.

2.7.2 Comparaison des deux méthodes de pas de temps adaptatif

Cette section est dédiée à l'étude des méthodes de Suzuki et de Lawson avec leur stratégie de pas de temps adaptatif associée, présentée dans la section 2.4.3. Pour toutes les simulations nous nous intéressons à l'estimateur d'erreur qui dans ce cas s'écrit :

$$\begin{aligned} L_{[3]}^{n+1} &= \left(\sum_{i=0}^{N_x-1} (u_{ci}^{n_1,[4]} - u_{ci}^{n_1,[3]})^2 \Delta x \right)^{\frac{1}{2}} + \left(\sum_{i=0}^{N_x-1} (E_i^{n_1,[4]} - E_i^{n_1,[3]})^2 \Delta x \right)^{\frac{1}{2}} \\ &\quad + \left(\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_v-1} |f_{hi,j}^{n_1,[4]} - f_{hi,j}^{n_1,[3]}|^2 \Delta v \Delta x \right)^{\frac{1}{2}} \\ &= L_{u_c}^{n+1} + L_E^{n+1} + L_{f_h}^{n+1}, \end{aligned} \quad (2.62)$$

où $u_{ci}^{n+1,[p]}$, $E_i^{n+1,[p]}$ et $f_{hi,j}^{n+1,[p]}$ sont les inconnues discrétisées calculées avec une méthode d'ordre p en temps et associée au temps t^{n+1} et au point $x_i = i\Delta x$, $i = 0, \dots, N_x$ et $v_j = -v_{\max} + j\Delta v$, $j = 0, \dots, N_v$ de l'espace des phases. Pour les deux méthodes, si le critère d'erreur $\|L_{[3]}^{n+1}\| < tol$ est satisfait, alors l'itération est acceptée et le temps incrémenté, sinon l'itération est rejetée et reprend au temps t^n . Dans les deux cas, le pas de temps suivant est calculé en utilisant (2.23). Cela nous permet de comparer l'estimateur d'erreur avec la même tolérance tol (prise arbitrairement à $tol = 2 \times 10^{-5}$) entre les deux intégrateurs en temps : la méthode de Suzuki et DP4(3). Nous regarderons aussi la taille des pas de temps proposés par les deux méthodes et le nombre d'itérations nécessaires pour finir la simulation.

method	nombre d'itérations	nombre d'itérations acceptées	ratio
Suzuki	23895	23849	0.998
Lawson-DP4(3)	2288	2192	0.958

TABLE 2.2 – Comparaison du nombre d'itération pour résoudre le problème jusqu'au temps $t = 300$, le nombre d'itérations acceptées de la méthode de pas de temps adaptatif et le ratio entre le nombre d'itérations acceptées et le nombre totale d'itérations.

Le tableau 2.2 présente le nombre d'itérations nécessaires pour atteindre le temps final $t = 300$ pour les deux méthodes de pas de temps adaptatif considérées (méthode de Suzuki et de Lawson-DP4(3)) avec les paramètres numériques suivants $N_x = 81$, $N_v = 128$. Le nombre d'itérations acceptées (itérations où le critères d'erreur $\|L_{[3]}^{n+1}\| < tol$ est satisfait), et le ratio entre le nombre d'itérations acceptées et le nombre total d'itérations est aussi présenté. On observe qu'une très large majorité des itérations sont acceptées pour les deux

méthodes, ce qui signifie que l'estimateur d'erreur est un bon indicateur. Pour la méthode de Lawson-DP4(3) le ratio d'itérations acceptées est légèrement plus faible que pour la méthode de Suzuki, ce qui signifie que la stratégie de pas de temps adaptatif essaie des pas de temps plus larges qui sont parfois rejetés. La très faible proportion d'itérations rejetées indique qu'il est très rarement nécessaire de recalculer une itération avec un pas de temps plus petit. Le surcoût engendré par la méthode de pas de temps adaptatif est donc négligeable. La seconde remarque que l'on peut faire est à propos de la méthode de Suzuki, qui nécessite 10 fois plus d'itérations que la méthode de Lawson-DP4(3) pour atteindre le temps final $t = 300$, ce qui signifie que la méthode de Suzuki nécessite de plus petits pas de temps pour satisfaire le critère d'erreur $\|L_{[3]}^{n+1}\| < tol$.

Sur la figure 2.12 est représentée l'évolution de la taille du pas de temps au cours du temps, en effet celui-ci suit l'équation (2.23) et est donc recalculé à chaque itération. Les itérations rejetées, celles où le critère d'erreur n'est pas satisfait, sont représentées avec des carrés. On remarque tout d'abord que dans la phase linéaire (jusqu'au temps $t \approx 50$) des pas de temps plus grands sont pris. Pendant la phase non-linéaire, le pas de temps oscille près d'une constante, qui permet de capturer les effets non-linéaires (vortex dans la distribution de particules) et les forts gradients (filamentation issue de la formation des votex). On remarque ensuite, que pour une même tolérance ($tol = 2 \times 10^{-5}$), la méthode de Suzuki nécessite de plus petits pas de temps, comparé à la méthode DP4(3), pour satisfaire le critère d'erreur, comme remarqué dans le tableau 2.2. On remarque que les variations du pas de temps sont relativement importantes ; il est possible de contrôler ces oscillations en limitant l'évolution des pas de temps en prenant par exemple $\Delta t^{n+1} \in [0.5\Delta t^n, 2\Delta t^n]$, comme proposé dans [1]. Sur la figure 2.13 est tracée l'évolution de l'erreur local $L_{[3]}^n$ comme une fonction du temps pour la méthode de Lawson (gauche) et de Suzuki (droite), avec une stratégie de pas de temps adaptatif un pas de temps constant. Pour un grand pas de temps ($\Delta t = 0.5$), les méthodes sont stables, mais le pas de temps dépasse largement la tolérance fixée à $tol = 2 \times 10^{-5}$. Les méthodes à pas de temps adaptatif, choisissent automatiquement un pas de temps permettant de garantir une erreur locale sous la tolérance. La méthode de Lawson avec un pas de temps adaptatif (DP4(3)) et avec un pas de temps fixe à $\Delta t = 0.1$ (RK(4,4)) calculent une erreur locale très proche, pourtant la méthode DP4(3) optimise son pas de temps pour assurer une erreur locale sous la tolérance, et permet d'obtenir des pas de temps plus grands que $\Delta t = 0.1$ comme on peut le remarquer sur la figure 2.12. Assurer une erreur sous une tolérance avec les plus grands pas de temps possible est une heuristique intéressante. En ce qui concerne les résultats avec la méthode de Suzuki, on

remarque de nouveau que la stratégie requiert de plus petits pas de temps, comparé à DP4(3), pour assurer une estimation de l'erreur locale sous la tolérance. De plus, avec un pas de temps constant à $\Delta t = 0.1$, la méthode de Suzuki génère des erreurs locales bien plus importante, alors que la méthode DP4(3) était presque sous la tolérance.

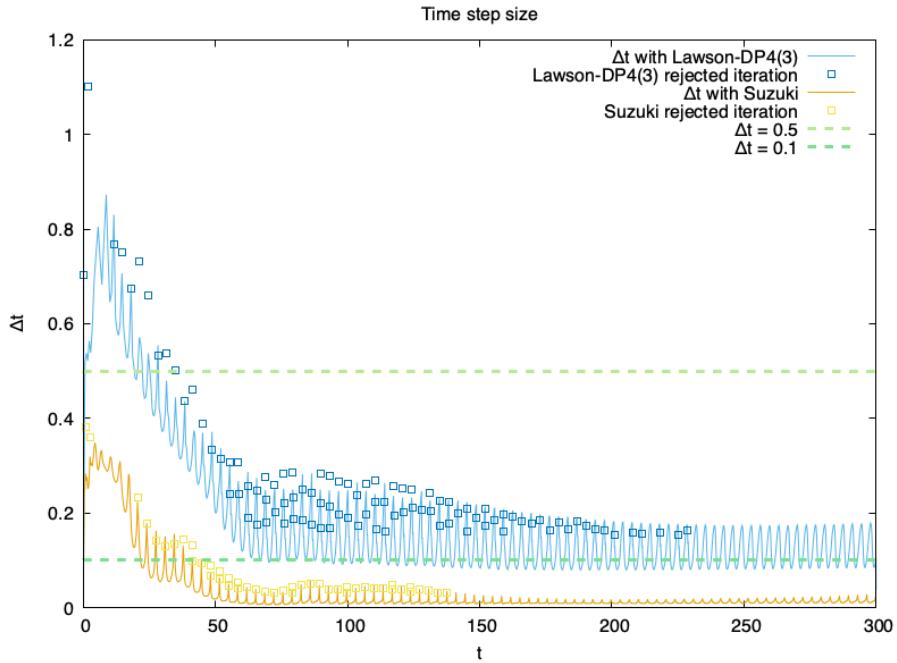


FIGURE 2.12 – Évolution du pas de temps Δt_n (les itérations rejetées sont représentées par des carrés) pour la modèle hybride avec des méthodes de pas de temps adaptatif.

Pour les deux méthodes à pas de temps adaptatif, on s'intéresse maintenant à l'évolution de l'estimateur d'erreur locale au cours du temps, en prenant en compte les différentes contributions de $L_{[3]}^n$, à savoir $L_{u_c}^n$, L_E^n et $L_{f_h}^n$ définies dans (2.62). Les résultats sont visibles sur la figure 2.14. L'erreur de la méthode DP4(3) et de ces différentes contributions sont visible en haut, alors que celles de la méthode de Suzuki sont tracées en bas. Pour la méthode DP4(3), on remarque tout d'abord que la contribution venant de $L_{u_c}^n$ est négligeable (autour de l'erreur machine), ce qui peut s'expliquer par le fait que dans la méthode de Lawson la partie linéaire est résolue exactement. Puisque la partie non-linéaire de la méthode de Lawson comprend le calcul du courant chaud $\int v f_h \, dv$ qui va impacter L_E^n et le transport dans la direction v qui va impacter $L_{f_h}^n$, les contributions à l'estimateur d'erreur de ces deux contributions reste prépondérante tout au long de la simulation. Pour la méthode de Suzuki, l'erreur provient essentiellement de $L_{f_h}^n$, erreur venant de l'interpolation du transport dans la direction v . Il est à noter, que les erreurs

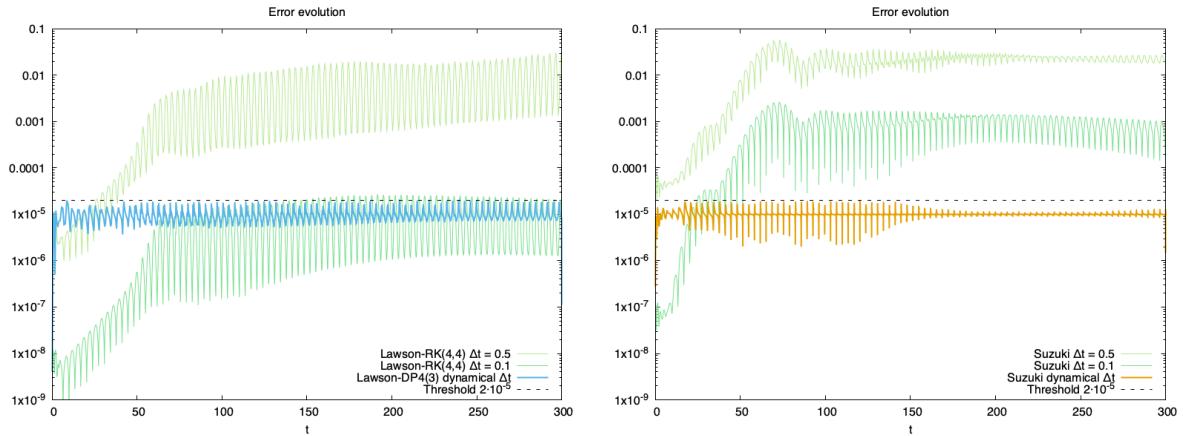


FIGURE 2.13 – Comparaison de l'évolution en temps de l'estimateur d'erreur pour une méthode à pas de temps constant et à pas de temps adaptatif. La méthode de Lawson est à gauche. La méthode de Suzuki est à droite. Les résultats sont en échelle semi-log.

$L_{u_c}^n$, L_E^n ne sont pas nulles au delà du temps $t \approx 50$, mais sont respectivement de l'ordre de 10^{-10} et 10^{-8} .

2.7.3 À propos des temps de calcul

Pour finir la comparaison entre les méthodes de *splitting* et de Lawson, nous comparerons leurs temps de calcul. Sur la figure 2.15 (gauche), on représente la valeur moyenne ainsi que les quartiles du temps de calcul d'une itération de RK(4,4), DP4(3) et de la méthode de Suzuki. Nous rappelons que la méthode RK(4,4) est constituée de 4 étages, alors que les méthodes DP4(3) et de Suzuki en comprennent 5. Cela explique pourquoi une itération de la méthode RK(4,4) coûte moins qu'une itération des deux autres méthodes. Sur la partie de droite de la figure 2.15 on compare le temps de calcul de chaque étape des différentes méthodes. On rappelle que la méthode DP4(3) est formée à partir des 4 étages de la méthode RK(4,4) plus un étage supplémentaire. Comme espéré, chaque étape de la méthode de Suzuki a le même coût (puisque la méthode de Suzuki est une composition de 5 méthodes de Strang). Au contraire, on observe que les deux premiers étages des méthodes RK(4,4) ou DP4(3) sont moins coûteux que les autres étages, ces deux premiers contenant moins d'opérations.

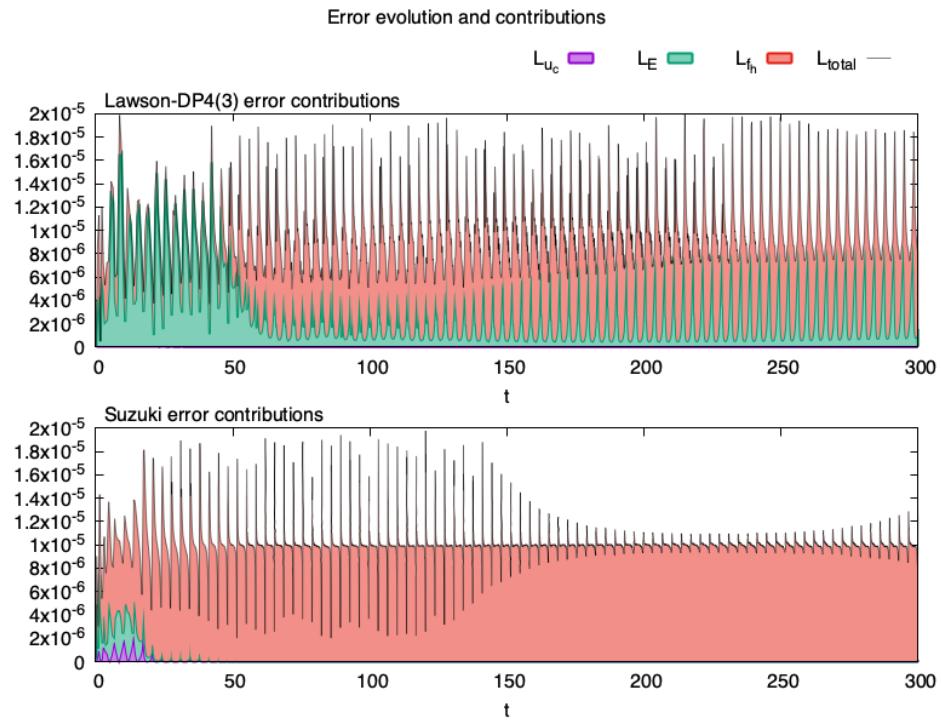


FIGURE 2.14 – Comparaison de la contribution de chaque composante de l'estimateur d'erreur locale en fonction du temps, pour la méthode de Lawson (haut) et de Suzuki (bas).

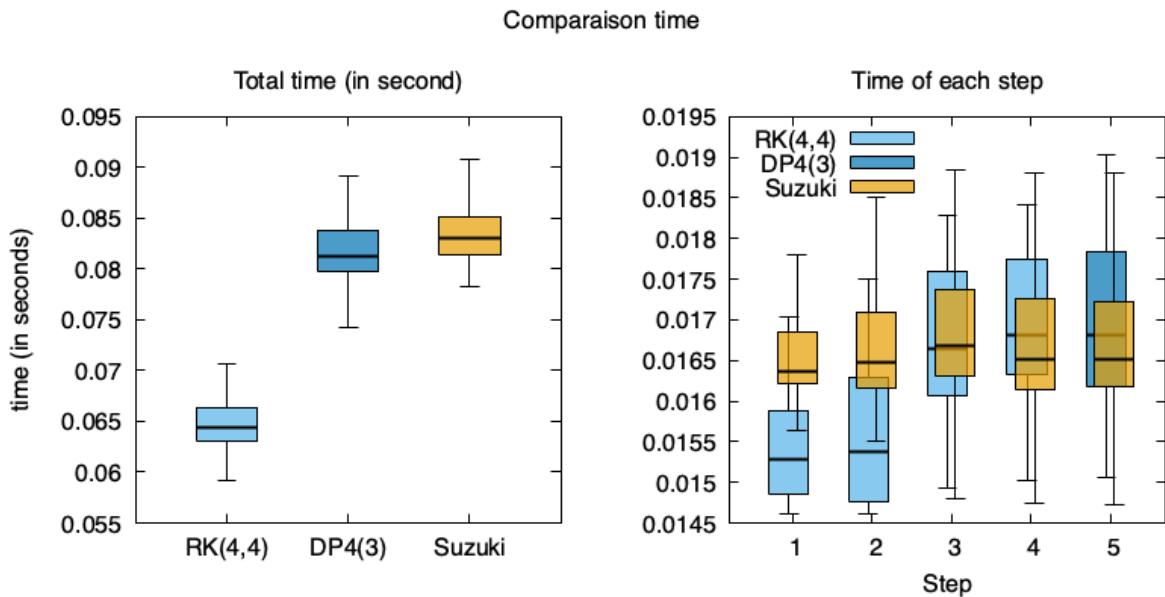


FIGURE 2.15 – Valeur moyenne et quartiles du temps de calcul sur une itération (gauche) et pour chaque étape d'une itération (droite).

APPENDICES

2.A Résultats sur les relations de dispersion

Cette annexe est dédiée aux démonstrations des propriétés énoncées dans la section 2.5 sur les relations de dispersion.

Nous démontrons tout d'abord le lemme 2.5.1, qui concerne la symétrie des racines de $D(k, \omega)$, et dont l'énoncé est rappelé ci-dessous.

Lemme 2.A.1. *Si $f^{(0)}(v)$ (respectivement $f_h^{(0)}(v)$) est une fonction paire, alors pour $D(k, \omega)$ défini par (2.31) (respectivement (2.38)) nous avons $D(k, \omega_r + i\omega_i) = 0 \Leftrightarrow D(k, -\omega_r + i\omega_i) = 0$.*

Démonstration. Nous le vérifions dans le cas cinétique, les calculs étant similaires dans le cas hybride. Avec la définition (2.31) de $D(k, \omega)$, nous avons

$$D(k, \omega_r + i\omega_i) = 0 \Leftrightarrow \operatorname{Re} \left(\frac{1}{k^2} \int_{\gamma} \frac{\partial_v f^0}{v - \frac{\omega_r + i\omega_i}{k}} dv \right) = 1, \quad \operatorname{Im} \left(\frac{1}{k^2} \int_{\gamma} \frac{\partial_v f^0}{v - \frac{\omega_r + i\omega_i}{k}} dv \right) = 0.$$

Distinguons les parties réelles et imaginaires :

$$\begin{aligned} \int_{\gamma} \frac{\partial_v f^0(v)}{v - \frac{\omega_r + i\omega_i}{k}} dv &= \int_{\gamma} \frac{\partial_v f^0(v)}{\left(v - \frac{\omega_r + i\omega_i}{k}\right) \left(v - \frac{\omega_r - i\omega_i}{k}\right)} \left(v - \frac{\omega_r - i\omega_i}{k}\right) dv \\ &= \int_{\gamma} \frac{\partial_v f^0(v)}{\left(v - \frac{\omega_r}{k}\right)^2 + \left(\frac{\omega_i}{k}\right)^2} \left(v - \frac{\omega_r}{k}\right) dv + i \int_{\gamma} \frac{\partial_v f^0(v)}{\left(v - \frac{\omega_r}{k}\right)^2 + \left(\frac{\omega_i}{k}\right)^2} \frac{\omega_i}{k} dv. \end{aligned}$$

Maintenant, considérons $\omega = -\omega_r + i\omega_i$ et rappelons qu'on a supposé que $f^0(v)$ était une

fonction paire. Nous obtenons

$$\begin{aligned}
 & \int_{\gamma} \frac{\partial_v f^0(v)}{v - \frac{-\omega_r + i\omega_i}{k}} dv \\
 &= \int_{\gamma} \frac{\partial_v f^0(v)}{\left(v + \frac{\omega_r}{k}\right)^2 + \left(\frac{\omega_i}{k}\right)^2} \left(v + \frac{\omega_r}{k}\right) dv + i \int_{\gamma} \frac{\partial_v f^0(v)}{\left(v + \frac{\omega_r}{k}\right)^2 + \left(\frac{\omega_i}{k}\right)^2} \frac{\omega_i}{k} dv \\
 &= - \int_{\gamma} \frac{\partial_v f^0(-v)}{\left(-v + \frac{\omega_r}{k}\right)^2 + \left(\frac{\omega_i}{k}\right)^2} \left(v - \frac{\omega_r}{k}\right) dv + i \int_{\gamma} \frac{\partial_v f^0(-v)}{\left(-v + \frac{\omega_r}{k}\right)^2 + \left(\frac{\omega_i}{k}\right)^2} \frac{\omega_i}{k} dv \\
 &= \int_{\gamma} \frac{\partial_v f^0(v)}{\left(v - \frac{\omega_r}{k}\right)^2 + \left(\frac{\omega_i}{k}\right)^2} \left(v - \frac{\omega_r}{k}\right) dv - i \int_{\gamma} \frac{\partial_v f^0(v)}{\left(v - \frac{\omega_r}{k}\right)^2 + \left(\frac{\omega_i}{k}\right)^2} \frac{\omega_i}{k} dv \quad .
 \end{aligned}$$

D'où

$$\begin{aligned}
 & \operatorname{Re} \left(\frac{1}{k^2} \int_{\gamma} \frac{\partial_v f^0}{v - \frac{-\omega_r + i\omega_i}{k}} dv \right) = 1, \quad \operatorname{Im} \left(\frac{1}{k^2} \int_{\gamma} \frac{\partial_v f^0}{v - \frac{-\omega_r + i\omega_i}{k}} dv \right) = 0 \\
 \Leftrightarrow & \operatorname{Re} \left(\frac{1}{k^2} \int_{\gamma} \frac{\partial_v f^0}{v - \frac{-\omega_r + i\omega_i}{k}} dv \right) = 1, \quad \operatorname{Im} \left(\frac{1}{k^2} \int_{\gamma} \frac{\partial_v f^0}{v - \frac{-\omega_r + i\omega_i}{k}} dv \right) = 0
 \end{aligned}$$

et

$$D(k, \omega_r + i\omega_i) = 0 \Leftrightarrow D(k, -\omega_r + i\omega_i) = 0.$$

□

Nous allons maintenant démontrer les lemmes 2.5.2, 2.5.3 et 2.5.4, dont les énoncés sont rappelés ci-dessous, qui donnent des propriétés de la fonction de Fried-Conte (2.46).

Lemme 2.A.2. *La fonction $Z_{\alpha}^0(\omega) : \omega \in \mathbb{C} \mapsto Z(\alpha\omega) \in \mathbb{C}$, avec $\alpha \in \mathbb{R}$ fixé, est telle que : $Z_{\alpha}^0(-\bar{\omega}) = -\overline{Z_{\alpha}^0(\omega)}$.*

Démonstration. Par définition de la fonction de Fried-Conte, et avec la notation $\omega = \omega_r + i\omega_i$, nous avons

$$Z(\alpha(\omega_r + i\omega_i)) = \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}}{z - \alpha(\omega_r + i\omega_i)} dz = \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega_r + i\alpha\omega_i)}{(z - \alpha\omega_r)^2 + (\alpha\omega_i)^2} dz$$

d'où

$$\begin{aligned}\operatorname{Re}(Z(\alpha(\omega_r + i\omega_i))) &= \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega_r)}{(z - \alpha\omega_r)^2 + (\alpha\omega_i)^2} dz \\ \operatorname{Im}(Z(\alpha(\omega_r + i\omega_i))) &= \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\alpha\omega_i}{(z - \alpha\omega_r)^2 + (\alpha\omega_i)^2} dz.\end{aligned}$$

Maintenant, $-\bar{\omega} = -\omega_r + i\omega_i$, implique

$$\begin{aligned}Z(\alpha(-\omega_r + i\omega_i)) &= \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}}{z - \alpha(-\omega_r + i\omega_i)} dz \\ &= \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z + \alpha\omega_r + i\alpha\omega_i)}{(z + \alpha\omega_r)^2 + (\alpha\omega_i)^2} dz \\ &= \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(-z + \alpha\omega_r + i\alpha\omega_i)}{(-z + \alpha\omega_r)^2 + (\alpha\omega_i)^2} dz \\ &= -\frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega_r)}{(z - \alpha\omega_r)^2 + (\alpha\omega_i)^2} dz + i \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\alpha\omega_i}{(z - \alpha\omega_r)^2 + (\alpha\omega_i)^2} dz\end{aligned}$$

d'où

$$\begin{aligned}\operatorname{Re}(Z(\alpha(-\omega_r + i\omega_i))) &= -\operatorname{Re}(Z(\alpha(\omega_r + i\omega_i))) \\ \operatorname{Im}(Z(\alpha(-\omega_r + i\omega_i))) &= \operatorname{Im}(Z(\alpha(\omega_r + i\omega_i))),\end{aligned}$$

ce qui termine la preuve. □

Lemme 2.A.3. La fonction $Z_{\alpha,\beta}^+(\omega) : \omega \in \mathbb{C} \mapsto Z(\alpha\omega - \beta) + Z(\alpha\omega + \beta) \in \mathbb{C}$, avec $\alpha \in \mathbb{R}$, $\beta \in \mathbb{R}$ fixés, est telle que : $Z_{\alpha,\beta}^+(-\bar{\omega}) = -\overline{Z_{\alpha,\beta}^+(\omega)}$.

Démonstration. Nous avons par définition de la fonction de Fried-Conte

$$\begin{aligned}Z(\alpha\omega - \beta) + Z(\alpha\omega + \beta) &= \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}}{z - \alpha\omega + \beta} + \frac{e^{-z^2}}{z - \alpha\omega - \beta} dz \\ &= \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega - \beta) + e^{-z^2}(z - \alpha\omega + \beta)}{(z - \alpha\omega)^2 - \beta^2} dz \\ &= \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega)}{(z - \alpha\omega)^2 - \beta^2} dz.\end{aligned}$$

Maintenant, avec la notation $\omega = \omega_r + i\omega_i$, nous avons

$$\begin{aligned}
 & Z(\alpha(\omega_r + i\omega_i) - \beta) + Z(\alpha(\omega_r + i\omega_i) + \beta) = \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega_r - i\alpha\omega_i)}{(z - \alpha\omega_r - i\alpha\omega_i)^2 - \beta^2} dz \\
 &= \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega_r - i\alpha\omega_i)}{(z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2 - 2i\alpha\omega_i(z - \alpha\omega_r)} dz \\
 &= \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega_r - i\alpha\omega_i)((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2 + 2i\alpha\omega_i(z - \alpha\omega_r))}{((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z - \alpha\omega_r)^2} dz \\
 &= \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}((z - \alpha\omega_r)((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2) + 2(\alpha\omega_i)^2(z - \alpha\omega_r))}{((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z - \alpha\omega_r)^2} dz \\
 &+ i \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(2\alpha\omega_i(z - \alpha\omega_r)^2 - \alpha\omega_i((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2))}{((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z - \alpha\omega_r)^2} dz \\
 &= \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega_r)((z - \alpha\omega_r)^2 + (\alpha\omega_i)^2 - \beta^2)}{((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z - \alpha\omega_r)^2} dz \\
 &+ i \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\alpha\omega_i((z - \alpha\omega_r)^2 + (\alpha\omega_i)^2 + \beta^2)}{((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z - \alpha\omega_r)^2} dz.
 \end{aligned}$$

Par ailleurs, en considérant $-\bar{\omega} = -\omega_r + i\omega_i$, nous avons

$$\begin{aligned}
 & Z(\alpha(-\omega_r + i\omega_i) - \beta) + Z(\alpha(-\omega_r + i\omega_i) + \beta) \\
 &= \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z + \alpha\omega_r)((z + \alpha\omega_r)^2 + (\alpha\omega_i)^2 - \beta^2)}{((z + \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z + \alpha\omega_r)^2} dz \\
 &+ i \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\alpha\omega_i((z + \alpha\omega_r)^2 + (\alpha\omega_i)^2 + \beta^2)}{((z + \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z + \alpha\omega_r)^2} dz.
 \end{aligned}$$

La seule fonction impaire en z est $(z + \alpha\omega_r)$, qui apparaît dans la partie réelle, ainsi

$$\begin{aligned}
 & Z(\alpha(-\omega_r + i\omega_i) - \beta) + Z(\alpha(-\omega_r + i\omega_i) + \beta) \\
 &= -\frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega_r)((z - \alpha\omega_r)^2 + (\alpha\omega_i)^2 - \beta^2)}{((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z - \alpha\omega_r)^2} dz \\
 &+ i \frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\alpha\omega_i((z - \alpha\omega_r)^2 + (\alpha\omega_i)^2 + \beta^2)}{((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z - \alpha\omega_r)^2} dz.
 \end{aligned}$$

L'identification des parties réelles et imaginaires de $Z(\alpha\omega - \beta) + Z(\alpha\omega + \beta)$ et $Z(-\alpha\bar{\omega} - \beta) + Z(-\alpha\bar{\omega} + \beta)$ achève la preuve. \square

Lemme 2.A.4. *La fonction $Z_{\alpha,\beta}^-(\omega) : \omega \in \mathbb{C} \mapsto Z(\alpha\omega - \beta) - Z(\alpha\omega + \beta) \in \mathbb{C}$, avec*

$\alpha \in \mathbb{R}$, $\beta \in \mathbb{R}$ fixés, est telle que : $Z_{\alpha,\beta}^-(-\bar{\omega}) = \overline{Z_{\alpha,\beta}^-(\omega)}$.

Démonstration. Nous avons par définition de la fonction de Fried-Conte

$$\begin{aligned} Z(\alpha\omega - \beta) - Z(\alpha\omega + \beta) &= \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}}{z - \alpha\omega + \beta} - \frac{e^{-z^2}}{z - \alpha\omega - \beta} dz \\ &= \frac{1}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}(z - \alpha\omega - \beta) - e^{-z^2}(z - \alpha\omega + \beta)}{(z - \alpha\omega)^2 - \beta^2} dz \\ &= -\frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\beta}{(z - \alpha\omega)^2 - \beta^2} dz. \end{aligned}$$

Maintenant, avec la notation $\omega = \omega_r + i\omega_i$, nous avons

$$\begin{aligned} Z(\alpha(\omega_r + i\omega_i) - \beta) - Z(\alpha(\omega_r + i\omega_i) + \beta) &= -\frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\beta}{(z - \alpha\omega_r - i\alpha\omega_i)^2 - \beta^2} dz \\ &= -\frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\beta}{(z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2 - 2i\alpha\omega_i(z - \alpha\omega_r)} dz \\ &= -\frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\beta((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2 + 2i\alpha\omega_i(z - \alpha\omega_r))}{((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z - \alpha\omega_r)^2} dz \end{aligned}$$

Par ailleurs, avec $-\bar{\omega} = -\omega_r + i\omega_i$, nous avons

$$\begin{aligned} Z(\alpha(-\omega_r + i\omega_i) - \beta) - Z(\alpha(-\omega_r + i\omega_i) + \beta) \\ = -\frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\beta((z + \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2 + 2i\alpha\omega_i(z + \alpha\omega_r))}{((z + \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z + \alpha\omega_r)^2} dz \end{aligned}$$

La seule fonction impaire en z est $(z + \alpha\omega_r)$, apparaissant dans la partie imaginaire, d'où

$$\begin{aligned} Z(\alpha(-\omega_r + i\omega_i) - \beta) - Z(\alpha(-\omega_r + i\omega_i) + \beta) \\ = -\frac{2}{\sqrt{\pi}} \int_{\gamma} \frac{e^{-z^2}\beta((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2 - 2i\alpha\omega_i(z - \alpha\omega_r))}{((z - \alpha\omega_r)^2 - (\alpha\omega_i)^2 - \beta^2)^2 + 4(\alpha\omega_i)^2(z - \alpha\omega_r)^2} dz \end{aligned}$$

L'identification des parties réelles et imaginaires de $Z(\alpha\omega - \beta) - Z(\alpha\omega + \beta)$ et $Z(-\alpha\bar{\omega} - \beta) - Z(-\alpha\bar{\omega} + \beta)$ achève la preuve. \square

Nous pouvons enfin démontrer les lemmes 2.5.5 et 2.5.6 concernant la vérification de l'hypothèse 2.1, qui conduit à l'expression (2.43) du mode fondamental du champ électrique linéarisé puis à l'approximation (2.44) de l'énergie électrique linéarisée. Ces

lemmes sont rappelés ci-dessous.

D'une part, nous rappelons le résultat 2.5.6 dans le cas cinétique.

Lemme 2.A.5. Pour $\frac{\partial D(k, \omega)}{\partial \omega}$ donnée par (2.56) et $N(k, \omega)$ par (2.57), l'hypothèse 2.1 est satisfaite.

Démonstration. En utilisant (2.56) et les lemmes 2.5.2, 2.5.3, 2.5.4 avec $\delta = \frac{1}{\sqrt{2T_c}k}$, $\eta = \frac{1}{\sqrt{2}k}$ et $\beta = \frac{v_0}{\sqrt{2}}$, nous avons

$$\begin{aligned}\frac{\partial D}{\partial \omega}(k, \omega) &= \frac{1}{\sqrt{2}k^3} \left[\frac{1-\alpha}{T_c\sqrt{T_c}} \left(\left(1 - \frac{\omega^2}{T_ck^2}\right) Z_\delta^0(\omega) - 2\frac{\omega}{\sqrt{2T_c}k} \right) \right. \\ &\quad + \frac{\alpha}{2} \left(\left(1 - \left(\frac{\omega}{k} - v_0\right)^2\right) Z\left(\frac{1}{\sqrt{2}}\left(\frac{\omega}{k} - v_0\right)\right) \right. \\ &\quad \left. + \left(1 - \left(\frac{\omega}{k} + v_0\right)^2\right) Z\left(\frac{1}{\sqrt{2}}\left(\frac{\omega}{k} + v_0\right)\right) \right. \\ &\quad \left. - \frac{2}{\sqrt{2}}\left(\frac{\omega}{k} - v_0\right) - \frac{2}{\sqrt{2}}\left(\frac{\omega}{k} + v_0\right) \right) \Big] \\ &= \frac{1}{\sqrt{2}k^3} \left[\frac{1-\alpha}{T_c\sqrt{T_c}} \left(\left(1 - \frac{\omega^2}{T_ck^2}\right) Z_\delta^0(\omega) - 2\frac{\omega}{\sqrt{2T_c}k} \right) - 2\sqrt{2}\frac{\omega}{k} \right. \\ &\quad \left. + \frac{\alpha}{2} \left((1 - v_0^2)Z_{\eta,\beta}^+(\omega) - \frac{\omega^2}{k^2}Z_{\eta,\beta}^+(\omega) + 2v_0\frac{\omega}{k}Z_{\eta,\beta}^-(\omega) \right) \right]\end{aligned}$$

et

$$\begin{aligned}\frac{\partial D}{\partial \omega}(k, -\bar{\omega}) &= \frac{1}{\sqrt{2}k^3} \left[\frac{1-\alpha}{T_c\sqrt{T_c}} \left(\left(1 - \frac{(-\bar{\omega})^2}{T_ck^2}\right) Z_\delta^0(-\bar{\omega}) + 2\frac{\bar{\omega}}{\sqrt{2T_c}k} \right) + 2\sqrt{2}\frac{\bar{\omega}}{k} \right. \\ &\quad \left. + \frac{\alpha}{2} \left((1 - v_0^2)Z_{\eta,\beta}^+(-\bar{\omega}) - \frac{(-\bar{\omega})^2}{k^2}Z_{\eta,\beta}^+(-\bar{\omega}) - 2v_0\frac{\bar{\omega}}{k}Z_{\eta,\beta}^-(-\bar{\omega}) \right) \right] \\ &= \frac{1}{\sqrt{2}k^3} \left[\frac{1-\alpha}{T_c\sqrt{T_c}} \left(-\left(1 - \frac{\bar{\omega}^2}{T_ck^2}\right) \overline{Z_\delta^0(\omega)} + 2\frac{\bar{\omega}}{\sqrt{2T_c}k} \right) + 2\sqrt{2}\frac{\bar{\omega}}{k} \right. \\ &\quad \left. + \frac{\alpha}{2} \left(-(1 - v_0^2)\overline{Z_{\eta,\beta}^+(\omega)} + \frac{\bar{\omega}^2}{k^2}\overline{Z_{\eta,\beta}^+(\omega)} - 2v_0\frac{\bar{\omega}}{k}\overline{Z_{\eta,\beta}^-(\omega)} \right) \right] \\ &= -\overline{\frac{\partial D}{\partial \omega}(k, \omega)}.\end{aligned}$$

Maintenant, en utilisant (2.57) et le lemme 2.5.3 avec $\eta = \frac{1}{\sqrt{2k}}$ et $\beta = \frac{v_0}{\sqrt{2}}$, nous avons

$$N(k, \omega) = -\frac{\hat{g}(k)}{k^2} \left(\frac{\alpha}{2\sqrt{2}} Z_{\eta, \beta}^+(\omega) \right)$$

et

$$\begin{aligned} N(k, -\bar{\omega}) &= -\frac{\hat{g}(k)}{k^2} \left(\frac{\alpha}{2\sqrt{2}} Z_{\eta, \beta}^+(-\bar{\omega}) \right) \\ &= -\frac{\hat{g}(k)}{k^2} \left(-\frac{\alpha}{2\sqrt{2}} \overline{Z_{\eta, \beta}^+(\omega)} \right) = -\overline{N(k, \omega)}. \end{aligned}$$

Ainsi, nous obtenons

$$\frac{N(k, -\bar{\omega})}{\frac{\partial D}{\partial \omega}(k, -\bar{\omega})} = \overline{\left(\frac{N(k, \omega)}{\frac{\partial D}{\partial \omega}(k, \omega)} \right)}.$$

Autrement dit, $\frac{N(k, \omega)}{\frac{\partial D}{\partial \omega}(k, \omega)} = re^{i\phi}$ si et seulement si $\frac{N(k, -\bar{\omega})}{\frac{\partial D}{\partial \omega}(k, -\bar{\omega})} = re^{-i\phi}$. \square

D'autre part, nous rappelons le résultat 2.5.5 dans le cas hybride.

Lemme 2.A.6. *Sous l'hypothèse $\hat{u}(t = 0, k) = 0$, pour $\frac{\partial D(k, \omega)}{\partial \omega}$ donnée par (2.51) et $N(k, \omega)$ par (2.52), l'hypothèse 2.1 est satisfaite.*

Démonstration. Regardons d'abord $\frac{\partial D(k, \omega)}{\partial \omega}$. Les termes en facteur de α (venant de la partie chaude cinétique) se comportent comme dans la preuve du lemme 2.5.5 (voir la preuve ci-dessus). Les termes en facteur de $1 - \alpha$ sont tels que

$$\frac{1}{(-\bar{\omega})^3} = -\frac{1}{\bar{\omega}^3} = -\frac{\overline{1}}{\omega^3}.$$

Nous en déduisons $\frac{\partial D}{\partial \omega}(k, -\bar{\omega}) = -\overline{\frac{\partial D}{\partial \omega}(k, \omega)}$.

Regardons ensuite $N(k, \omega)$. Sous l'hypothèse $\hat{u}(t = 0, k) = 0$ et avec les notations $\eta = \frac{1}{\sqrt{2k}}$ et $\beta = \frac{v_0}{\sqrt{2}}$, nous avons

$$N(k, -\bar{\omega}) = -\frac{1}{-i\bar{\omega}} \hat{E}(t = 0, k) - \frac{\hat{g}(k)}{k^2} \left[\alpha \frac{k}{-\bar{\omega}} + \frac{\alpha}{2\sqrt{2}} Z_{\eta, \beta}^+(-\bar{\omega}) \right].$$

Nous rappelons que $\hat{E}(t = 0, k)$ est un imaginaire pur (éventuellement nul) donné par

(2.58). Ceci implique

$$\begin{aligned} N(k, -\bar{\omega}) &= \frac{1}{i\bar{\omega}} \hat{E}(t = 0, k) + \frac{\hat{g}(k)}{k^2} \left[\alpha \frac{k}{\bar{\omega}} + \frac{\alpha}{2\sqrt{2}} \overline{Z_{\eta,\beta}^+(\omega)} \right] \\ &= \frac{1}{i\omega} \overline{\hat{E}(t = 0, k)} + \frac{\hat{g}(k)}{k^2} \left[\alpha \frac{\bar{k}}{\omega} + \frac{\alpha}{2\sqrt{2}} \overline{Z_{\eta,\beta}^+(\omega)} \right] = -\overline{N(k, \omega)}. \end{aligned}$$

La preuve est terminée. \square

MODÈLE HYBRIDE LINÉARISÉ DANS LE CAS $1dz - 3dv$

Ce chapitre est la seconde partie d'une étude du modèle hybride linéarisé (0.12)-(0.15) dans le cadre restreint $1dz - 3dv$. Il s'agit d'utiliser les concepts du chapitre précédent et de les étendre au cas multi-dimensionnel en vitesse pour prendre en compte les effets du champ magnétique. La manipulation de variables vectoriels impose une généralisation du cadre scalaire étudié dans le chapitre 1. Ce travail collaboratif avec Anaïs Crestetto¹, Nicolas Crouseilles² et Yingzhe Li³ a mené à un article *Comparison of high-order Eulerian methods for electron hybrid model* soumis dans *Journal of Computational Physics* en 2021.

3.1 Introduction

L'objectif de ce chapitre est d'étendre les stratégies développées pour la résolution d'un modèle hybride linéarisé, présentées dans le chapitre 2, au cas $1dz - 3dv$ permettant de prendre en compte, entre autre, les effets du champ magnétique sur la dynamique des particules. Le modèle que nous étudions ici s'applique au cas où la distribution de particules est composée de deux populations, une première ayant une vitesse thermique faible, et considérées comme *froide*, dont on approximera la dynamique comme celle d'un fluide ; une seconde population ayant une vitesse thermique plus élevée, considérées comme *chaude*, mais contrairement au chapitre 2, celle-ci n'a pas besoin d'être distribuée selon une bi-maxwellienne. Ce type de modèle peut être utilisé pour modéliser des particules d'un plasma dans un tokamak, ou des particules du vent solaire interagissant avec la

1. Université de Nantes, laboratoire Jean Leray
 2. Univ Rennes, Inria Bretagne Atlantique (MINGuS) & ENS Rennes
 3. Max Planck Institute, Institut Für Plasmaphysik, Germany

magnétosphère terrestre. Dans ces contextes les particules se déplacent de manière hélicoïdale selon les lignes du champ extérieur $\mathbf{B}_0 = (0, 0, B_0)^\top$, et seul le déplacement dans cette direction sera étudié ici, pour le déplacement hélicoïdale complet il faudrait ajouter ici une référence que je n'ai pas car je ne connais pas trop ces modèles, de mémoire c'est ce que faisait Xiaofei. Cela justifie la réduction du modèle à une seule dimension spatiale, ce qui permet de réduire le coût de calcul et ainsi tester différentes stratégies de simulation. Les deux méthodes de simulation que nous nous proposons de comparer sont les générations des deux stratégies développées dans le chapitre 2, une méthode de *splitting* et une méthode de Lawson.

3.2 Présentation du modèle

$$\frac{\partial j_{c,x}}{\partial t} = \Omega_{pe}^2 E_x - j_{c,y} B_0, \quad (3.1)$$

$$\frac{\partial j_{c,y}}{\partial t} = \Omega_{pe}^2 E_y + j_{c,x} B_0, \quad (3.2)$$

$$\frac{\partial B_x}{\partial t} = \partial_z E_y, \quad (3.3)$$

$$\frac{\partial B_y}{\partial t} = -\partial_z E_x, \quad (3.4)$$

$$\frac{\partial E_x}{\partial t} = -\partial_z B_y - j_{c,x} + \int v_x f_h \, d\mathbf{v}, \quad (3.5)$$

$$\frac{\partial E_y}{\partial t} = \partial_z B_x - j_{c,y} + \int v_y f_h \, d\mathbf{v}, \quad (3.6)$$

$$\frac{\partial f_h}{\partial t} + v_z \partial_z f_h - (E_x + v_y B_0 - v_z B_y) \partial_{v_x} f_h \quad (3.7)$$

$$- (E_y - v_x B_0 + v_z B_x) \partial_{v_y} f_h - (v_x B_y - v_y B_x) \partial_{v_z} f_h = 0.$$

Cela nous permet de définir l'hamiltonien :

$$\mathcal{H} = \underbrace{\frac{1}{2} \int_{\mathbb{R}} (E_x^2 + E_y^2) \, dz}_{\mathcal{H}_E} + \underbrace{\frac{1}{2} \int_{\mathbb{R}} (B_x^2 + B_y^2) \, dz}_{\mathcal{H}_B} + \underbrace{\frac{1}{2} \int_{\mathbb{R}} \frac{1}{\Omega_{pe}^2} (j_{c,x}^2 + j_{c,y}^2) \, dz}_{\mathcal{H}_{j_c}} + \underbrace{\frac{1}{2} \int_{\mathbb{R}} \int_{\mathbb{R}^2} |\mathbf{v}|^2 f_h \, d\mathbf{v} \, dz}_{\mathcal{H}_{f_h}} \quad (3.8)$$

On définit le crouchet suivant, défini pour deux fonctionnelles \mathcal{F} et \mathcal{G} :

$$\begin{aligned}
 \{\mathcal{F}, \mathcal{G}\}[j_{c,x}, j_{c,y}, B_x, B_y, E_x, E_y, f_h] &= \int_{\mathbb{R}} \int_{\mathbb{R}^3} f_h \left(\partial_z \frac{\delta \mathcal{F}}{\delta f_h} \partial_{v_z} \frac{\delta \mathcal{G}}{\delta f_h} - \partial_{v_z} \frac{\delta \mathcal{F}}{\delta f_h} \partial_z \frac{\delta \mathcal{G}}{\delta f_h} \right) d\mathbf{v} dz \\
 &\quad + \int_{\mathbb{R}} \int_{\mathbb{R}^3} f_h \left(\partial_{v_x} \frac{\delta \mathcal{F}}{\delta f_h} \frac{\delta \mathcal{G}}{\delta E_x} + \partial_{v_y} \frac{\delta \mathcal{F}}{\delta f_h} \frac{\delta \mathcal{G}}{\delta E_y} - \partial_{v_x} \frac{\delta \mathcal{G}}{\delta f_h} \frac{\delta \mathcal{F}}{\delta E_x} - \partial_{v_y} \frac{\delta \mathcal{G}}{\delta f_h} \frac{\delta \mathcal{F}}{\delta E_y} \right) d\mathbf{v} dz \\
 &\quad + \int_{\mathbb{R}} \int_{\mathbb{R}^3} f_h (\mathbf{B} + \mathbf{B}_0) \cdot \left(\nabla_{\mathbf{v}} \frac{\delta \mathcal{F}}{\delta f_h} \times \nabla_{\mathbf{v}} \frac{\delta \mathcal{G}}{\delta f_h} \right) d\mathbf{v} dz \\
 &\quad + \int_{\mathbb{R}} \left(-\partial_z \frac{\delta \mathcal{F}}{\delta E_y} \frac{\delta \mathcal{G}}{\delta B_x} + \partial_z \frac{\delta \mathcal{F}}{\delta E_x} \frac{\delta \mathcal{G}}{\delta B_y} + \partial_z \frac{\delta \mathcal{G}}{\delta E_y} \frac{\delta \mathcal{F}}{\delta B_x} - \partial_z \frac{\delta \mathcal{G}}{\delta E_x} \frac{\delta \mathcal{F}}{\delta B_y} \right) dz \\
 &\quad + \int_{\mathbb{R}} \Omega_{pe}^2 \left(\frac{\delta \mathcal{F}}{\delta j_{c,x}} \frac{\delta \mathcal{G}}{\delta E_x} + \frac{\delta \mathcal{F}}{\delta j_{c,y}} \frac{\delta \mathcal{G}}{\delta E_y} - \frac{\delta \mathcal{G}}{\delta j_{c,x}} \frac{\delta \mathcal{F}}{\delta E_x} - \frac{\delta \mathcal{G}}{\delta j_{c,y}} \frac{\delta \mathcal{F}}{\delta E_y} \right) dz \\
 &\quad + \int_{\mathbb{R}} \Omega_{pe}^2 B_0 \left(\frac{\delta \mathcal{F}}{\delta j_{c,x}} \frac{\delta \mathcal{G}}{\delta j_{c,y}} - \frac{\delta \mathcal{F}}{\delta j_{c,y}} \frac{\delta \mathcal{G}}{\delta j_{c,x}} \right) dz.
 \end{aligned} \tag{3.9}$$

Cela nous permet de réécrire le système (3.1)-(3.7) comme

$$\partial_t U = \{U, \mathcal{H}\}$$

où $U(t, z, \mathbf{v}) = (j_{c,\perp}(t, z), B_{\perp}(t, z), E_{\perp}(t, z), f)h(t, z, \mathbf{v})^{\top}$ où \mathcal{H} est donné par (3.8) et $j_{c,\perp} = (j_{c,x}, j_{c,y})^{\top}$, $B_{\perp} = (B_x, B_y)^{\top}$ et $E_{\perp} = (E_x, E_y)^{\top}$. Par la suite, nous utiliserons également les notations $v_{\perp} = (v_x, v_y)^{\top} \in \mathbb{R}^2$ ainsi que $\mathbf{v} = (v_{\perp}, v_z)^{\top} \in \mathbb{R}^3$ et nous définissons la matrice symplectique

$$J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

3.3 Schémas numériques

3.3.1 Méthode de *splitting* hamiltonien

Nous utilisons ici une méthode de *splitting* hamiltonien. Pour le modèle $1dz - 3dv$ celui-ci se décompose en 4 étapes, et il s'écrit sous la forme :

$$\begin{aligned}\partial_t U &= \{U, \mathcal{H}_{j_c}\} + \{U, \mathcal{H}_B\} + \{U, \mathcal{H}_E\} + \{U, \mathcal{H}_{f_h}\} \\ U(t=0) &= U_0\end{aligned}\tag{3.10}$$

Nous allons nous intéresser au calcul de $\varphi_t^{[j_c]}$, $\varphi_t^{[B]}$, $\varphi_t^{[E]}$ et $\varphi_t^{[f_h]}$ les solutions correspondant à chaque étape de sorte que $\varphi(U_0)$ de (3.10) peut être approximé au temps t avec une composition des sous-flux $\varphi_t^{[j_c, B, E, f_h]}$.

Étape \mathcal{H}_{j_c} : Pour obtenir $\varphi_t^{[j_c]}$, solution du sous-flux \mathcal{H}_{j_c} :

$$\begin{cases} \partial_t j_{c,\perp} = -J j_{c,\perp} B_0 \\ \partial_t B_\perp = 0 \\ \partial_t E_\perp = -j_{c,\perp} \\ \partial_t f_h = 0 \end{cases}$$

nous calculons :

$$\varphi_t^{[j_c]}(U_0) = \begin{pmatrix} e^{-tJ} j_{c,\perp}(0) B_0 \\ B_\perp(0) \\ E_\perp(0) - J(e^{-tJ} - I) j_{c,\perp}(0) \\ f_h(0) \end{pmatrix}$$

Ce résultat s'obtient grâce à $\int_0^t \exp(-sJ) j_{c,\perp}(0) ds = J(\exp(-tJ) - I) j_{c,\perp}(0)$.

Nous présentons l'algorithme permettant d'effectuer cette étape dans l'extrait de pseudo-code 3.1. Dans les calculs de complexité que nous nous proposons de faire ici, seuls les parcours de tableaux nous intéressent, la complexité des fonctions mathématiques est supposée constante. Cette étape a une complexité de $\mathcal{O}(N_z)$ en temps.

Algorithme 3.1 Calcul de l'étape \mathcal{H}_{j_c}

```

1: fonction  $\mathcal{H}_{j_c}(j_{c,x}, j_{c,y}, B_x, B_y, E_x, E_y, f_h)$ 
2:   pour  $i = 0, \dots, N_z - 1$  faire :
3:      $\bar{j}_{c,x} \leftarrow j_{c,x,[i]} \cos(\Delta t) - j_{c,y,[i]} \sin(\Delta t)$ 
4:      $\bar{j}_{c,y} \leftarrow j_{c,x,[i]} \sin(\Delta t) + j_{c,y,[i]} \cos(\Delta t)$ 
5:      $E_{x,[i]} \leftarrow -j_{c,x,[i]} \sin(\Delta t) + j_{c,y,[i]} (1 - \cos(\Delta t))$ 
6:      $E_{y,[i]} \leftarrow j_{c,x,[i]} (\cos(\Delta t) - 1) - j_{c,y,[i]} \sin(\Delta t)$ 
7:      $j_{c,x,[i]} \leftarrow \bar{j}_{c,x}$ 
8:      $j_{c,y,[i]} \leftarrow \bar{j}_{c,y}$ 
9:   fin pour
10: fin fonction

```

Étape \mathcal{H}_B : Nous souhaitons calculer $\varphi_t^{[B]}$, correspondant à la solution du système :

$$\begin{cases} \partial_t j_{c,\perp} = 0 \\ \partial_t B_\perp = 0 \\ \partial_t E_\perp = -J \partial_z B_\perp \\ \partial_t f_h = 0 \end{cases}$$

qui s'obtient de la manière suivante :

$$\varphi_t^{[B]}(U_0) = \begin{pmatrix} j_{c,\perp}(0) \\ B_\perp(0) \\ E_\perp(0) - t J \partial_z B_\perp(0) \\ f_h(0) \end{pmatrix}$$

Cette étape est résolu dans l'espace de Fourier, on obtient alors l'algorithme 3.2. Pour obtenir la complexité de cette étape il faut connaître la complexité de l'algorithme de transformée de Fourier rapide : FFT que l'on effectue selon l'axe z . L'implémentation naïve de cet algorithme est en $\mathcal{O}(N_z^2)$, mais il est possible de descendre à une complexité de $\mathcal{O}(N_z \log(N_z))$ qui est la valeur que nous considérerons. La complexité de la transformée de Fourier inverse, iFFT, est la même. On obtient ainsi une complexité temporelle de $\mathcal{O}(6N_z \log(N_z) + N_z) = \mathcal{O}(N_z \log(N_z))$. Pour la complexité spatiale, il est à noter que cette étape nécessite l'allocation de 4 tableaux temporaires de taille N_z pour le stockage des transformée de Fourier, donc contenant des valeurs complexes (chaque nombre complexe nécessite le stockage de deux réels à virgule flottante).

Algorithme 3.2 Calcul de l'étape \mathcal{H}_B

```

1: fonction  $\mathcal{H}_B(j_{c,x}, j_{c,y}, B_x, B_y, E_x, E_y, \hat{f}_h)$ 
2:    $\hat{B}_x \leftarrow \text{FFT}_z(B_x)$ 
3:    $\hat{B}_y \leftarrow \text{FFT}_z(B_y)$ 
4:    $\hat{E}_x \leftarrow \text{FFT}_z(E_x)$ 
5:    $\hat{E}_y \leftarrow \text{FFT}_z(E_y)$ 
6:   pour  $j = 0, \dots, N_z - 1$  faire :
7:      $\hat{E}_{x,[j]} \leftarrow \hat{E}_{x,[j]} - i\Delta t \kappa_{[j]} \hat{B}_{y,[j]}$ 
8:      $\hat{E}_{y,[j]} \leftarrow \hat{E}_{y,[j]} + i\Delta t \kappa_{[j]} \hat{B}_{x,[j]}$ 
9:   fin pour
10:   $E_x \leftarrow i\text{FFT}_z(\hat{E}_x)$ 
11:   $E_y \leftarrow i\text{FFT}_z(\hat{E}_y)$ 
12: fin fonction

```

Étape \mathcal{H}_E : Pour calculer la solution du sous-flux correspondant à \mathcal{H}_E , nous devons résoudre le système suivant :

$$\begin{cases} \partial_t j_{c,\perp} &= \Omega_{pe}^2 E_\perp \\ \partial_t B_\perp &= J \partial_z E_\perp \\ \partial_t E_\perp &= (0, 0)^\top \\ \partial_t f_h &= E_\perp \cdot \nabla_{v_\perp} f_h \end{cases}$$

Avec la condition initiale donnée par $U(t = 0) = U_0 = (j_{c,\perp}, B_\perp, E_\perp, f_h)(t = 0)$, la solution au temps t est obtenue par :

$$\varphi_t^{[E]}(U_0) = \begin{pmatrix} j_{c,\perp} + t\Omega_{pe}^2 E_\perp(0) \\ B_\perp(0) + tJ \partial_z E_\perp(0) \\ E_\perp(0) \\ f_h(0, z, v_\perp + tE_\perp(0), v_z) \end{pmatrix}$$

Le calcul de $f_h(0, z, v_\perp + tE_\perp(0), v_z)$ s'effectue en utilisant deux interpolations polynomiale de Lagrange d'ordre 5 à une dimension (une dans la direction v_x et une autre dans la direction v_y). L'algorithme permettant de résoudre cette étape est présenté dans l'algorithme 3.3. Les sous-étapes \mathcal{H}_{E,v_x} et \mathcal{H}_{E,v_y} sont très similaires et seul la première est détaillée, seul change l'axe d'interpolation et le pied de la caractéristique. Ces sous-étapes correspondent aux deux interpolations avec un polynôme de Lagrange, faisant

appel à la fonction, que l'on ne définit pas ici, `LAGRANGE_GENERATOR()` qui permet d'obtenir un polynôme de Lagrange, de degré 5, à partir des 6 points d'interpolations donnés en argument. On considère comme constante la complexité du calcul d'un polynôme d'interpolation, on obtient alors une complexité algorithmique en $\mathcal{O}(N_{v_x} N_{v_y} N_{v_z} N_z)$ pour les sous-étapes \mathcal{H}_{E,v_x} et \mathcal{H}_{E,v_y} . Le calcul de la transformée de Fourier inverse de \hat{f}_h , à la ligne 16, nécessite $N_{v_x} \times N_{v_y} \times N_{v_z}$ transformées de Fourier selon l'axe z , soit une complexité totale de $\mathcal{O}(N_{v_x} N_{v_y} N_{v_z} N_z \log(N_z))$, de même pour la transformée de Fourier à la ligne 19. L'étape \mathcal{H}_E possède donc une complexité algorithmique de $\mathcal{O}(2N_{v_x} N_{v_y} N_{v_z} N_z + 2N_{v_x} N_{v_y} N_{v_z} N_z \log(N_z) + N_z) = \mathcal{O}(N_{v_x} N_{v_y} N_{v_z} N_z \log(N_z))$. Pour ce qui concerne l'utilisation de la mémoire, cette étape nécessite deux tableaux de valeurs réelles $f^{(a)}$ et $f^{(b)}$ temporaires de taille $N_{v_x} N_{v_y} N_{v_z}$.

Algorithme 3.3 Calcul de l'étape \mathcal{H}_E

```

1: fonction  $\mathcal{H}_{E,v_x}(f^{in}, f^{out}, E_x)$ 
2:   pour tout  $(k_x, k_y, k_z) \in \llbracket 0, N_{v_x} \rrbracket \times \llbracket 0, N_{v_y} \rrbracket \times \llbracket 0, N_{v_z} \rrbracket$  faire :
3:     pour  $i = 0, \dots, N_z - 1$  faire :
4:        $v^* \leftarrow v_x + \Delta t E_{x,[i]}$ 
5:        $k^* \leftarrow \lceil \frac{v^* - v_{x,\min}}{\Delta v_x} \rceil$ 
6:        $\mathcal{L}_{[5]} \leftarrow \text{LAGRANGE\_GENERATOR}(f^{in}_{[k^*-3:k^*+2, k_y, k_z, i]})$ 
7:        $f^{out}_{[k_x, k_y, k_z, i]} \leftarrow \mathcal{L}_{[5]}(v^*)$ 
8:     fin pour
9:   fin pour
10:  fin fonction

11: fonction  $\mathcal{H}_E(j_{c,x}, j_{c,y}, B_x, B_y, E_x, E_y, \hat{f}_h)$ 
12:   pour  $i = 0, \dots, N_z - 1$  faire :
13:      $j_{c,x,[i]} \leftarrow j_{c,x,[i]} + \Omega_{pe}^2 \Delta t E_{x,[i]}$ 
14:      $j_{c,y,[i]} \leftarrow j_{c,y,[i]} + \Omega_{pe}^2 \Delta t E_{y,[i]}$ 
15:   fin pour
16:    $f^{(a)} \leftarrow i\text{FFT}_z(\hat{f}_h)$             $\triangleright$  La transformée de Fourier inverse s'effectue  $\forall v_x, v_y, v_z$ 
17:    $\mathcal{H}_{E,v_x}(f^{(a)}, f^{(b)}, E_x)$ 
18:    $\mathcal{H}_{E,v_y}(f^{(b)}, f^{(a)}, E_y)$ 
19:    $\hat{f}_h \leftarrow \text{FFT}_z(f^{(a)})$             $\triangleright$  La transformée de Fourier s'effectue  $\forall v_x, v_y, v_z$ 
20: fin fonction

```

Étape \mathcal{H}_{f_h} : Pour la dernière étape, nous devons calculer une solution du sous-système :

$$\begin{cases} \partial_t j_{c,\perp} = 0 \\ \partial_t B_\perp = 0 \\ \partial_t E_\perp = \int v_\perp f_h \, dv \\ \partial_t f_h = -v_z \partial_z f_h + (v_y B_0 - v_z B_y) \partial_{v_x} f_h + (-v_x B_0 + v_z B_x) \partial_{v_y} f_h + (v_x B_y - v_y B_x) \partial_{v_z} f_h \end{cases}$$

Comme pour la résolution de l'équation de Vlasov-Maxwell ([62]), ce système ne peut être résolu exactement en temps. Mais en suivant [62], nous pouvons subdiviser encore l'hamiltonien \mathcal{H}_{f_h} en $\mathcal{H}_{f_h} = \mathcal{H}_{f_{h,x}} + \mathcal{H}_{f_{h,y}} \mathcal{H}_{f_{h,z}}$, où $\mathcal{H}_{f_{h,\star}} = \frac{1}{2} \int v_\star^2 f_h \, dv$, où $\star = x, y, z$. Cela conduit à résoudre les sous-système suivant :

— $\mathcal{H}_{f_{h,x}}$:

$$\begin{cases} \partial_t j_{c,\perp} = 0 \\ \partial_t B_\perp = 0 \\ \partial_t E_x = \int v_x f_h \, dv \\ \partial_t E_y = 0 \\ \partial_t f_h - (-v_x B_0 \partial_{v_y} f_h + v_x B_y \partial_{v_z} f_h) = 0 \end{cases}$$

On remarque tout d'abord que dans ce sous-système, $\int v_x f_h \, dv$ est constant en temps, donc l'équation d'Ampère peut être résolue aisément. L'équation de transport sur f_h , peut être résolue exactement en utilisant un *splitting* directionnel puisque les deux opérateurs commutent, on obtient alors :

$$\varphi_t^{[f_{h,x}]}(U_0) = \begin{pmatrix} j_{c,\perp}(0) \\ B_\perp(0) \\ E_x(0) + t \int v_x f_h(0) \, dv \\ E_y(0) \\ f_h(0, z, v_x, v_y - tv_x B_0, v_z + t B_y v_x) \end{pmatrix}$$

En pratique, nous utiliserons deux interpolations d'ordre 5 avec des polynômes de Lagrange de dimension 1 dans les directions v_y et v_z , comme présenté dans l'algorithme 3.4. La sous-étape $\mathcal{H}_{f_{h,x,v_y}}$ n'est qu'une interpolation polynomiale et ne sera pas détaillée ici. La sous-étape $\mathcal{H}_{f_{h,x,v_z}}$ présente en plus la résolution de l'équation d'Ampère, que nous détaillons ici. La complexité de cette étape est

$\mathcal{O}(2N_{v_x}N_{v_y}N_{v_z}N_z + N_z) = \mathcal{O}(N_{v_x}N_{v_y}N_{v_z}N_z)$. Pour la complexité spatiale, on remarque la nécessité d'un tableau temporaire de réels $f^{(a)}$ de taille $N_{v_x}N_{v_y}N_{v_z}$.

Algorithme 3.4 Calcul de l'étape $\mathcal{H}_{f_{h,x}}$

```

1: fonction  $\mathcal{H}_{f_{h,x,v_z}}(f^{in}, f^{out}, B_y, E_x)$ 
2:    $j_{h,x} \leftarrow 0$                                  $\triangleright$  Initialisation d'un tableau de taille  $N_z$  à 0
3:   pour tout  $(k_x, k_y, k_z) \in \llbracket 0, N_{v_x} \rrbracket \times \llbracket 0, N_{v_y} \rrbracket \times \llbracket 0, N_{v_z} \rrbracket$  faire :
4:     pour  $i = 0, \dots, N_z - 1$  faire :
5:        $v^* \leftarrow v_z + \Delta t v_x B_{y,[i]}$ 
6:        $k^* \leftarrow \lceil \frac{v^* - v_{z,\min}}{\Delta v_z} \rceil$ 
7:        $\mathcal{L}_{[5]} \leftarrow \text{LAGRANGE\_GENERATOR}(f^{in}_{[k_x, k_y, k^*-3:k^*+2, i]})$ 
8:        $f^{out}_{[k_x, k_y, k_z, i]} \leftarrow \mathcal{L}_{[5]}(v^*)$ 
9:        $j_{h,x,[i]} \leftarrow j_{h,x,[i]} + v_x f^{out}_{[k_x, k_y, k_z, i]} \Delta \mathbf{v}$ 
10:      fin pour
11:      fin pour
12:       $mean \leftarrow \frac{1}{N_z} \sum_i j_{h,x,[i]}$ 
13:      pour  $i = 0, \dots, N_z - 1$  faire :
14:         $E_{x,[i]} \leftarrow E_{x,[i]} + \Delta t (j_{h,x,[i]} - mean)$ 
15:      fin pour
16:    fin fonction

17: fonction  $\mathcal{H}_{f_{h,x}}(f^{in}, f^{out}, B_y, E_x)$ 
18:    $\mathcal{H}_{f_{h,x,v_y}}(f^{in}, f^{(a)})$ 
19:    $\mathcal{H}_{f_{h,x,v_z}}(f^{(a)}, f^{out}, B_y, E_x)$ 
20: fin fonction

```

— $\mathcal{H}_{f_{h,y}}$:

$$\begin{cases} \partial_t j_{c,\perp} = 0 \\ \partial_t B_\perp = 0 \\ \partial_t E_x = 0 \\ \partial_t E_y = \int v_y f_h \, d\mathbf{v} \\ \partial_t f_h - (v_y B_0 \partial_{v_x} f_h + v_y B_x \partial_{v_z} f_h) = 0 \end{cases}$$

Cette étape est très similaire à la précédente, le courant des particules chaudes $\int v_y f_h \, d\mathbf{v}$ y est une constante, et l'équation de transport est résolue en utilisant un

splitting directionnel. On a :

$$\varphi_t^{[f_{h,x}]}(U_0) = \begin{pmatrix} j_{c,\perp}(0) \\ B_\perp(0) \\ E_x(0) \\ E_y(0) + t \int v_y f_h(0) \, d\mathbf{v} \\ f_h(0, z, v_x - tv_y B_0, v_y, v_z - t B_x v_y) \end{pmatrix}$$

De la même façon, nous utiliserons pour la résolution deux interpolations de dimension 1, interpolation d'ordre 5 à l'aide de polynômes de Lagrange. Nous obtenons les mêmes complexités temporelle et spatiale que l'étape $\mathcal{H}_{f_{h,x}}$.

— $\mathcal{H}_{f_{h,z}}$:

$$\begin{cases} \partial_t j_{c,\perp} = 0 \\ \partial_t B_\perp = 0 \\ \partial_t E_\perp = 0 \\ \partial_t f_h + v_z \partial_z f_h - (-v_z B_y \partial_{v_x} f_h + v_z B_x \partial_{v_y} f_h) = 0 \end{cases}$$

Le calcul de l'étape $\mathcal{H}_{f_{h,z}}$ nécessite plus de travail. Tout d'abord nous introduisons la fonction $g(t, z, \mathbf{v}) := f(t, z + tv_z, \mathbf{v})$ qui satisfait :

$$\partial_t g + B_y(0, z + tv_z) v_z \partial_{v_x} g - v_z B_x(0, z + tv_z) \partial_{v_y} g = 0 \quad (3.11)$$

Cette équation de transport peut être résolue exactement en temps. Pour cela, dans un premier temps on regarde les caractéristiques de (3.11) :

$$\dot{v}_x(t) = B_y(0, z(0) + tv_z(0)) v_z(0), \quad \dot{v}_y(t) = -B_x(0, z(0) + tv_z(0)) v_z(0), \quad (3.12)$$

celles-ci peuvent être résolues exactement. Les variables z , v_z , B_x et B_y sont constantes en temps dans cette sous-étape, mais le changement de variable introduit une dépendance en temps. Cela peut être résolu en développant le champ magnétique B_\perp en série de Fourier dans la direction z , ce qui nous donne :

$$B_\perp(t, z) = B_\perp(0, z) = \sum_k \hat{B}_\perp(0, k) e^{ikz},$$

on a alors :

$$B_{\perp}(0, z + tv_z) = \sum_k \hat{B}_{\perp}(0, k) e^{ik(z+tv_z)}.$$

On obtient alors, en intégrant l'équation (3.11) en temps :

$$\begin{aligned} v_x(t) &= v_x(0) + v_z(0) \int_0^t \sum_k \hat{B}_y(0, k) e^{ik(z(0)+sv_z(0))} ds \\ &= v_x(0) + v_z(0) \sum_k \hat{B}_y(0, k) e^{ikz(0)} \int_0^t e^{iks v_z(0)} ds \\ &= v_x(0) + \sum_k \hat{B}_y(0, k) \frac{1}{ik} e^{ikz(0)} (e^{ikt v_z(0)} - 1), \end{aligned}$$

tandis que pour l'équation sur v_y , nous obtenons :

$$\begin{aligned} v_y(t) &= v_y(0) - v_z(0) \int_0^t \sum_k \hat{B}_x(0, k) e^{ik(z(0)+sv_z(0))} ds \\ &= v_y(0) - \sum_k \hat{B}_x(0, k) \frac{1}{ik} e^{ikz(0)} (e^{ikt v_z(0)} - 1). \end{aligned}$$

En remontant les caractéristiques, on peut calculer $g(t, z, \mathbf{v})$ comme :

$$\begin{aligned} g(t, z, \mathbf{v}) &= g \left(0, z, v_x - \sum_k \hat{B}_y(0, k) \frac{1}{ik} e^{ikz} (e^{ikt v_z} - 1), \right. \\ &\quad \left. v_y + \sum_k \hat{B}_x(0, k) \frac{1}{ik} e^{ikz} (e^{ikt v_z} - 1), v_z \right). \end{aligned}$$

Cela nous permet d'obtenir $f_h(t)$ en effectuant le changement de variable inverse :

$$f(t, z, \mathbf{v}) = g(t, z - tv_z, \mathbf{v}).$$

Ces différentes étapes permettent de construire l'algorithme 3.5 permettant la résolution de ce sous-système. Deux sous-étapes $\mathcal{H}_{f_{h,z,v_x}}$ et $\mathcal{H}_{f_{h,z,v_y}}$, effectuent les interpolations polynomiales d'ordre 5, respectivement dans les directions v_x et v_y , nous détaillerons ici que la première. La complexité algorithmique de l'étape correspond à deux fois celle de la sous-étape $\mathcal{H}_{f_{h,z,v_x}}$ (la sous-étape $\mathcal{H}_{f_{h,z,v_y}}$ étant très similaire). Cette étape consiste une transformée de Fourier suivant l'axe z (de complexité $\mathcal{O}(N_z \log(N_z))$), puis pour toute valeur de $(k_x, k_y, k_z, j) \in \llbracket 0, N_{v_x} \rrbracket \times \llbracket 0, N_{v_y} \rrbracket \times \llbracket 0, N_{v_z} \rrbracket$, une somme sur N_z et une interpolation polynomiale que l'on a supposé de complexité

constante. On obtient une complexité algorithmique pour les sous-étapes $\mathcal{H}_{f_{h,z,vx}}$ et $\mathcal{H}_{f_{h,z,vy}}$ de $\mathcal{O}(N_z \log(N_z) + N_{v_x} N_{v_y} N_{v_z} N_z^2)$. La sous-étape $\mathcal{H}_{f_{h,z,z}}$, consiste en une transformée de Fourier dans la direction z (de complexité $\mathcal{O}(N_z \log(N_z))$) et un parcours dans la direction z , pour toute valeur d'indices du quadruplet (k_x, k_y, k_z, j) , ce qui correspond à une complexité de $\mathcal{O}(N_{v_x} N_{v_y} N_{v_z} (N_z \log(N_z) + N_z))$. On obtient une complexité algorithmique de l'étape $\mathcal{H}_{f_{h,z}}$ de :

$$\mathcal{O}(2N_z \log(N_z) + N_{v_x} N_{v_y} N_{v_z} N_z (N_z + \log(N_z) + 1)) = \mathcal{O}(N_{v_x} N_{v_y} N_{v_z} N_z^2).$$

Algorithme 3.5 Calcul de l'étape $\mathcal{H}_{f_{h,z}}$

```

1: fonction  $\mathcal{H}_{f_{h,z,vx}}(f^{in}, f^{out}, B_y)$ 
2:    $\hat{B}_y \leftarrow \text{FFT}_z(B_y)$ 
3:   pour tout  $(k_x, k_y, k_z) \in \llbracket 0, N_{v_x} \rrbracket \times \llbracket 0, N_{v_y} \rrbracket \times \llbracket 0, N_{v_z} \rrbracket$  faire :
4:     pour  $j = 0, \dots, N_z - 1$  faire :
5:        $s \leftarrow -\sum_{\ell=-N_z/2}^{N_z/2} i \frac{\hat{B}_{y,[\ell]}}{\kappa_{[\ell]}} \exp(i\kappa_{[\ell]} z_{[j]})(\exp(i\kappa_{[\ell]} v_z \Delta t) - 1)$ 
6:        $v^* \leftarrow v_x - \text{Re}(s)$ 
7:        $k^* \leftarrow \lceil \frac{v^* - v_{x,\min}}{\Delta v_x} \rceil$ 
8:        $\mathcal{L}_{[5]} \leftarrow \text{LAGRANGE\_GENERATOR}(f^{in}_{[k^*-3:k^*+2,k_y,k_z,j]})$ 
9:        $f^{out}_{[k_x,k_y,k_z,j]} \leftarrow \mathcal{L}_{[5]}(v^*)$ 
10:      fin pour
11:    fin pour
12:  fin fonction

13: fonction  $\mathcal{H}_{f_{h,z,z}}(f^{in}, \hat{f}^{out})$ 
14:   pour tout  $(k_x, k_y, k_z) \in \llbracket 0, N_{v_x} \rrbracket \times \llbracket 0, N_{v_y} \rrbracket \times \llbracket 0, N_{v_z} \rrbracket$  faire :
15:      $\hat{f}_v \leftarrow \text{FFT}_z(f^{in}_{[k_x,k_y,k_z,:]})$ 
16:     pour  $j = 0, \dots, N_z - 1$  faire :
17:        $\hat{f}^{out}_{[k_x,k_y,k_z,j]} \leftarrow \hat{f}_{v,[j]} \exp(-iv_x \kappa_{[j]} \Delta t)$ 
18:     fin pour
19:   fin pour
20: fin fonction

21: fonction  $\mathcal{H}_{f_{h,z}}(f^{in}, B_x, B_y, \hat{f}^{out})$ 
22:    $\mathcal{H}_{f_{h,z,vx}}(f^{in}, f^{(a)}, B_y)$ 
23:    $\mathcal{H}_{f_{h,z,vy}}(f^{(a)}, f^{(b)}, B_x)$ 
24:    $\mathcal{H}_{f_{h,z,z}}(f^{(b)}, \hat{f}^{out})$ 
25: fin fonction
```

Finalement, la solution de \mathcal{H}_{f_h} est calculée dans l'algorithme 3.6, qui est une simple composition des sous-étapes $\mathcal{H}_{f_{h,x}}$, $\mathcal{H}_{f_{h,y}}$ et $\mathcal{H}_{f_{h,z}}$. La complexité algorithmique est donc la somme des complexités de chaque sous-étape, ce qui revient, de manière asymptotique, à la complexité de la sous-partie la plus coûteuse, c'est-à-dire $\mathcal{H}_{f_{h,z}}$. La complexité algorithmique de \mathcal{H}_{f_h} est $\mathcal{O}(N_{v_x} N_{v_y} N_{v_z} N_z^2)$. On note cependant la nécessité d'avoir des au moins 2 tableaux temporaires $f^{(a)}$ et $f^{(b)}$ de taille $N_{v_x} N_{v_y} N_{v_z}$ de réels, la variable $f^{(c)}$ peut être remplacée par $f^{(a)}$. Cette étape est la plus coûteuse en temps de calcul, elle sera donc placer au milieu de la méthode *splitting* pour former la méthode de Strang.

Algorithme 3.6 Calcul de l'étape \mathcal{H}_{f_h}

```

1: fonction  $\mathcal{H}_{f_h}(j_{c,x}, j_{c,y}, B_x, B_y, E_x, E_y, \hat{f}_h)$ 
2:    $f^{(a)} \leftarrow i\text{FFT}_z(\hat{f})$ 
3:    $\mathcal{H}_{f_{h,x}}(f^{(a)}, f^{(b)}, B_y, E_x)$ 
4:    $\mathcal{H}_{f_{h,y}}(f^{(b)}, f^{(c)}, B_x, E_y)$ 
5:    $\mathcal{H}_{f_{h,z}}(f^{(c)}, \hat{f}_h, B_x, B_y)$ 
6: fin fonction
```

Il est par la suite nécessaire de concaténer les étapes \mathcal{H}_{j_c} , \mathcal{H}_B , \mathcal{H}_E et \mathcal{H}_{f_h} pour construire une méthode de Lie, de Strang ou de Suzuki (voir section 2.4.1.2 ou [81] et [5]). Il est à noter que la méthode Lie possède déjà 4 étapes (11 sous-étapes) et la méthode de Strang 7 (avec 15 sous-étapes, les sous-étapes de \mathcal{H}_{f_h} n'ayant pas besoin d'être concaténées en palindrome). Pour des raisons de temps de calcul, la méthode de Suzuki, avec 35 étapes (soit 75 sous-étapes) ne sera pas considérée comme alternative valable aux méthodes de Lawson d'ordre élevé.

3.3.2 Méthode de Lawson sur le modèle hybride

Dans cette section nous allons présenter la méthode d'intégration exponentielle pour discrétiser le modèle VHL (3.1)-(3.7). Il est naturel de réécrire ce système, après une transformée de Fourier dans la direction z , sous la forme :

$$\partial_t U = LU + N(t, U)$$

où $\kappa = \frac{2j\pi}{N_z}$, $j \in \llbracket -\frac{N_z}{2}, \frac{N_z}{2} \rrbracket$ représente les modes de Fourier et avec :

$$L = \begin{pmatrix} 0 & -B_0 & 0 & 0 & \Omega_{pe}^2 & 0 & 0 \\ B_0 & 0 & 0 & 0 & 0 & \Omega_{pe}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & i\kappa & 0 \\ 0 & 0 & 0 & 0 & -ik & 0 & 0 \\ -1 & 0 & 0 & -ik & 0 & 0 & 0 \\ 0 & -1 & i\kappa & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -v_z \partial_z \end{pmatrix}, \quad N : t, U \mapsto \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \int v_x f \, dv \\ \int v_y f \, dv \\ E_\perp \cdot \nabla_{v_\perp} f - v \times B \nabla_v f \end{pmatrix} \quad (3.13)$$

Mais le calcul de la partie linéaire $e^{\tau L}$ nécessaire pour l'écriture du schéma de la méthode LRK n'est pas réalisable avec SymPy ou un autre logiciel de calcul formel. Cela vient de l'expression des valeurs propres, dépendant du temps (τ) et de la discréétisation dans l'espace de Fourier des dérivées spatiales ($\kappa \in \left\{ \frac{2\pi j}{L}, j \in \llbracket -N_z, N_z \rrbracket \right\}$). Pour résoudre ce problème, il est décidé dans un premier temps d'intégrer la partie provenant des équations de Maxwell de L dans la partie non-linéaire N . Ce qui nous mène à redéfinir L et N comme :

$$L = \begin{pmatrix} 0 & -B_0 & 0 & 0 & \Omega_{pe}^2 & 0 & 0 \\ B_0 & 0 & 0 & 0 & 0 & \Omega_{pe}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -v_z \partial_z \end{pmatrix}, \quad N : t, U \mapsto \begin{pmatrix} 0 \\ 0 \\ i\kappa E_y \\ -i\kappa E_x \\ -ikB_y + \int v_x f \, dv \\ i\kappa B_x + \int v_y f \, dv \\ E_\perp \cdot \nabla_{v_\perp} f - v \times B \nabla_v f \end{pmatrix} \quad (3.14)$$

Un logiciel de calcul formel, comme SymPy, permet d'obtenir $e^{\tau L}$ pour toute valeur $\tau \in \mathbb{R}$:

$$e^{\tau L} \approx \begin{pmatrix} \alpha \cos(\theta^-) + \beta \cos(\theta^+) & \alpha \sin(\theta^-) - \beta \sin(\theta^+) & 0 & 0 & \gamma \sin(\theta^-) + \gamma \sin(\theta^+) & -\gamma \cos(\theta^-) + \gamma \cos(\theta^+) & 0 \\ -\alpha \sin(\theta^-) + \beta \sin(\theta^+) & \alpha \cos(\theta^-) + \beta \cos(\theta^+) & 0 & 0 & \gamma \cos(\theta^-) - \gamma \cos(\theta^+) & \gamma \sin(\theta^-) + \gamma \sin(\theta^+) & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -\delta \sin(\theta^-) - \delta \sin(\theta^+) & \delta \cos(\theta^-) - \delta \cos(\theta^+) & 0 & 0 & \beta \cos(\theta^-) + \alpha \cos(\theta^+) & \beta \sin(\theta^-) - \alpha \sin(\theta^+) & 0 \\ -\delta \cos(\theta^-) + \delta \cos(\theta^+) & -\delta \sin(\theta^-) - \delta \sin(\theta^+) & 0 & 0 & -\beta \sin(\theta^-) + \alpha \sin(\theta^+) & \beta \cos(\theta^-) + \alpha \cos(\theta^+) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{-ikv_z \tau} \end{pmatrix} \quad (3.15)$$

avec $\alpha = 0.378732187481834$, $\beta = 0.621267812518167$, $\gamma = 0.970142500145332$, $\delta = 0.242535625036333$, et $\theta^\pm = \frac{\sqrt{2}}{2}\tau\sqrt{9 \pm \sqrt{17}}$.

C'est donc à partir de la matrice L sans les termes provenant des équations de Maxwell, et avec le terme non-linéaire N , contenant les termes des équations de Maxwell, définis dans (3.14), que nous allons construire notre premier schéma de Lawson. Il est important d'avoir une formulation explicite de l'exponentielle de la partie linéaire pour construire un schéma en temps efficace. En effet le calcul numérique de l'exponentielle d'une matrice pour une valeur de τ et de k données est élevé et ne peut être effectuer en amont de la simulation. Le calcul en amont de ces quantités impose une discréétisation en espace donnée, et empêche la mise en place de méthodes à pas de temps adaptatif. Nous verrons par la suite, dans la section 3.5, une méthode permettant d'intégrer les termes des équations de Maxwell dans la partie linéaire de la méthode de Lawson.

3.3.2.1 Filtrage de l'équation de Vlasov

L'équation de Vlasov contient le terme $(\mathbf{v} \times \mathbf{B}_0)$ qui introduit une condition de CFL restrictive qui peut être outrepassé en utilisant le changement de variable $w = e^{tB_0J}v$, avec $w = (w_x, w_y)^\top$ et $v = (v_x, v_y)^\top$, pour filtrer ce terme en influençant le moins possible le reste du système. Nous introduisons $g(t, z, w, v_z) = f_h(t, z, \exp(-tB_0J)w, v_z)$ avec :

$$e^{-tB_0J} = \begin{pmatrix} \cos(tB_0) & -\sin(tB_0) \\ \sin(tB_0) & \cos(tB_0) \end{pmatrix}, \quad J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

alors on a $\exp(-tB_0J)w = (\cos(tB_0)w_x - \sin(tB_0)w_y, \sin(tB_0)w_x + \cos(tB_0)w_y)^\top$. On peut désormais dériver l'équation sur g :

$$\partial_t g + v_z \partial_z g - e^{-tB_0J} \mathbf{E} \cdot \nabla_w g - \mathcal{T}_v^{[B_0]} g = 0, \quad (3.16)$$

où $\mathcal{T}_{\mathbf{v}}^{[B_0]}g$ est donné par :

$$\begin{aligned}
 \mathcal{T}_{\mathbf{v}}^{[B_0]}g &:= (\mathbf{v} \times \mathbf{B}) \cdot_{\mathbf{v}} f_h \\
 &= \begin{pmatrix} \cos(tB_0)w_x - \sin(tB_0)w_y \\ \sin(tB_0)w_x + \cos(tB_0)w_y \\ v_z \end{pmatrix} \times \begin{pmatrix} B_x \\ B_y \\ 0 \end{pmatrix} \cdot \begin{pmatrix} e^{-tB_0 J} \nabla_w g \\ \partial_{v_z} g \end{pmatrix} \\
 &= \begin{pmatrix} v_z(-\cos(tB_0)B_y + \sin(tB_0)B_x) \\ v_z(\sin(tB_0)B_y + \cos(tB_0)B_x) \\ (\cos(tB_0)w_x - \sin(tB_0)w_y)B_y - (\sin(tB_0)w_x + \cos(tB_0)w_y)B_x \end{pmatrix} \cdot \begin{pmatrix} \partial_{w_x} g \\ \partial_{w_y} g \\ \partial_{v_z} g \end{pmatrix}
 \end{aligned} \tag{3.17}$$

Le changement de variable impacte également l'équation d'Ampère (3.5)-(3.6), l'intégrale du courant des particules chaudes devient :

$$\int \mathbf{v} f \, d\mathbf{v} = \int (e^{-tB_0 J} w) g \, dw \, dv_z,$$

les équations d'Ampère peuvent alors se réécrire comme :

$$\partial_t E_{\perp} = -J \partial_z B_{\perp} - j_{c,\perp} + \int (e^{-tB_0 J} w) g \, dw \, dv_z. \tag{3.18}$$

Les autres équations restent inchangées. Les termes impactés sont dans la partie non-linéaire de la méthode de Lawson, il n'y a donc aucune implication sur le calcul de l'exponentielle de la matrice de la partie linéaire.

3.3.2.2 Calcul de stabilité avec les équations de Maxwell

Dans la méthode de Lawson explicite que nous souhaitons mettre en place, nous proposons de placer les équations de Maxwell, équations linéaires, dans la partie non-linéaire (voir (3.14)), entraînant par conséquent une condition de stabilité. Puisque nous résolvons de manière explicite des équations linéaires, il est possible d'effectuer un calcul de cette condition CFL de stabilité induite par les équations de Maxwell. Les termes non-linéaires de l'équation de Vlasov sur les particules chaudes, imposent également une condition de stabilité, que l'on se propose de se restreindre à la condition suivante :

$$\|\mathbf{E} + \mathbf{v} \times \mathbf{B}\|_{\infty} \Delta t \leq \Delta v,$$

condition que l'on peut estimer numériquement par une première simulation.

Estimation de la condition de stabilité à partir d'une méthode Runge-Kutta :

Il est possible de faire une première estimation de la condition de stabilité en ne considérant que l'équation :

$$\dot{U} = A \cdot U \quad (3.19)$$

avec $U = (B_x, B_y, E_x, E_y)^\top$ le vecteur des seules variables impliquées dans les équations de Maxwell, et A la matrice :

$$A = \begin{pmatrix} 0 & 0 & 0 & i\kappa \\ 0 & 0 & -i\kappa & 0 \\ 0 & -i\kappa & 0 & 0 \\ i\kappa & 0 & 0 & 0 \end{pmatrix}$$

On souhaite résoudre ce problème avec une méthode de type Runge-Kutta, en effet notre choix de partie linéaire dans la méthode de Lawson pour résoudre le système VMHL nous impose de résoudre les équations de Maxwell via une méthode de type Runge-Kutta. La fonction à intégrer en temps par la méthode Runge-Kutta étant linéaire en U , on sait que calculer U^{n+1} par une méthode Runge-Kutta revient à multiplier U^n par la fonction de stabilité de la méthode Runge-Kutta choisie, évaluée en $\Delta t A$. On a donc pour une méthode RK(3,3) :

$$U^{n+1} = \left(I + \Delta t A + \frac{\Delta t^2}{2} A^2 + \frac{\Delta t^3}{6} A^3 \right) U^n,$$

et pour une méthode RK(4,4) :

$$U^{n+1} = \left(I + \Delta t A + \frac{\Delta t^2}{2} A^2 + \frac{\Delta t^3}{6} A^3 + \frac{\Delta t^4}{24} A^4 \right) U^n.$$

On note $p_{(s,n)}$ la fonction de stabilité de la méthode Runge-Kutta à s étages et d'ordre n RK(s,n). On appellera par la suite *matrice d'amplification*, la matrice correspondant à l'évaluation de la fonction de stabilité de la méthode RK(s,n) évaluée en $\Delta t A$. Pour des méthodes de type Runge-Kutta à s étages et d'ordre n explicite on a :

$$p_{(s,n)}(\Delta t A) = \sum_{k=0}^n \frac{\Delta t^k}{k!} A^k + \sum_{k=n+1}^s c_k \Delta t^k A^k$$

Ce qui correspond au développement de l'exponentielle jusqu'à l'ordre n , plus des termes supplémentaires si $n \neq s$. Pour estimer la CFL de l'équation (3.19), il faut rechercher les valeurs propres de la matrice d'amplification $p_{(s,n)}(\Delta t A)$ de la méthode choisie. Celles-ci dépendent de Δt et de κ , avec les modes de Fourier κ qui dépendent directement de la discrétisation en espace choisie, car $\kappa = \frac{2\pi j}{L}$ avec $j \in \llbracket -\frac{N_z}{2}, \frac{N_z}{2} \rrbracket$. Pour trouver la condition de stabilité de cette équation il faut donc trouver le couple $(\Delta t, \kappa)$ permettant de garantir que les valeurs propres de $p_{(s,n)}(\Delta t A)$ soient inférieure à 1.

On rappelle que les valeurs propres de A sont : $i\kappa$ et $-i\kappa$, chacune de multiplicité 2, cela nous permet de déterminer les valeurs propres de $p_{(s,n)}(\Delta t A)$ qui sont :

$$\begin{aligned} p_{(s,n)}(i\kappa\Delta t) &= \sum_{k=0}^n i^k \frac{\Delta t^k \kappa^k}{k!} + \sum_{k=n+1}^s i^k c_k \Delta t^k \kappa^k \\ p_{(s,n)}(-i\kappa\Delta t) &= \sum_{k=0}^n (-1)^k i^k \frac{\Delta t^k \kappa^k}{k!} + \sum_{k=n+1}^s (-1)^k i^k c_k \Delta t^k \kappa^k \end{aligned}$$

Il s'agit de polynômes en $\Delta t \kappa$ que l'on peut réécrire :

$$\begin{aligned} p_{(s,n)}(i\kappa\Delta t) &= \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \frac{\Delta t^{2k} \kappa^{2k}}{(2k)!} + i \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor - 1} (-1)^k \frac{\Delta t^{2k+1} \kappa^{2k+1}}{(2k+1)!} \\ &\quad + \sum_{k=\lfloor \frac{n}{2} \rfloor + 1}^{\lfloor \frac{s}{2} \rfloor} (-1)^k c_{2k} \Delta t^{2k} \kappa^{2k} + i \sum_{k=\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{s}{2} \rfloor - 1} (-1)^k c_{2k+1} \Delta t^{2k+1} \kappa^{2k+1} \\ p_{(s,n)}(-i\kappa\Delta t) &= \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} (-1)^k \frac{\Delta t^{2k} \kappa^{2k}}{(2k)!} - i \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor - 1} (-1)^k \frac{\Delta t^{2k+1} \kappa^{2k+1}}{(2k+1)!} \\ &\quad + \sum_{k=\lfloor \frac{n}{2} \rfloor + 1}^{\lfloor \frac{s}{2} \rfloor} (-1)^k c_{2k} \Delta t^{2k} \kappa^{2k} - i \sum_{k=\lfloor \frac{n}{2} \rfloor}^{\lfloor \frac{s}{2} \rfloor - 1} (-1)^k c_{2k+1} \Delta t^{2k+1} \kappa^{2k+1} \end{aligned}$$

où l'on remarque que $p_{(s,n)}(-i\kappa\Delta t) = \overline{p_{(s,n)}(i\kappa\Delta t)}$, donc $|p_{(s,n)}(-i\kappa\Delta t)| = |p_{(s,n)}(i\kappa\Delta t)|$. On peut déterminer la condition de stabilité pour les méthodes RK(3,3) et RK(4,4), c'est-à-dire la valeur de $\Delta t \kappa$ telle que $|p_{(n,n)}(i\kappa\Delta t)| = 1$, on obtient ainsi respectivement les valeurs $\frac{\sqrt{3}}{\kappa}$ et $\frac{2\sqrt{2}}{\kappa}$ avec $\kappa = \frac{2j\pi}{L}$, $j \in \llbracket -\frac{N_z}{2}, \frac{N_z}{2} \rrbracket$.

Estimation de la condition de stabilité à partir d'une méthode de Lawson complète :

Une estimation plus précise de la condition de stabilité dû aux équations de Maxwell

peut être calculée en résolvant avec une méthode de Lawson le système :

$$\dot{U} = LU + NU$$

où $U = (j_{c,x}, j_{c,y}, B_x, B_y, E_x, E_y)^\top$, et L (partie linéaire de la méthode de Lawson) et N (partie considérée comme non-linéaire dans la méthode de Lawson) sont deux matrices données par :

$$L = \begin{pmatrix} 0 & -1 & 0 & 0 & 4 & 0 \\ 1 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad N = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & ik \\ 0 & 0 & 0 & 0 & -ik & 0 \\ 0 & 0 & 0 & -ik & 0 & 0 \\ 0 & 0 & ik & 0 & 0 & 0 \end{pmatrix}$$

On ne s'intéresse ici qu'à la condition de stabilité provenant des équations de Maxwell, dû à l'impossibilité de calculer formellement $e^{\tau L}$ avec les équations de Maxwell. La densité de particules f_h n'est donc pas considérée ici. De plus la condition de stabilité que nous cherchons est celle provenant des équations de Maxwell, donc une condition sur la discréétisation spatiale, la condition de stabilité induite par f_h ne porte que sur la discréétisation en vitesse et ne sera pas étudié ici car le terme est fortement non-linéaire. Les matrices L et N ne commutent pas, donc la fonction de stabilité de la méthode de Lawson ne peut pas être obtenu comme dans le cas scalaire du chapitre 1. Il est nécessaire pour chaque méthode de calculer sa fonction de stabilité. Nous n'étudierons ici que les méthodes de Lawson induites par la méthode RK(3,3) de Shu-Osher et celle induite par la méthode RK(4,4) dites classique.

Nous allons commencer l'étude par la méthode RK(3,3) définie par :

$$\begin{aligned} U^{(1)} &= e^{\Delta t L} U^n + \Delta t e^{\Delta t L} N U^n \\ U^{(2)} &= \frac{3}{4} e^{\frac{\Delta t}{2} L} U^n + \frac{1}{4} e^{-\frac{\Delta t}{2} L} U^{(1)} + \frac{\Delta t}{4} e^{-\frac{\Delta t}{2} L} N U^{(1)} \\ U^{n+1} &= \frac{1}{3} e^{\Delta t L} U^n + \frac{2}{3} e^{\frac{\Delta t}{2} L} U^{(2)} + \frac{2}{3} \Delta t e^{\frac{\Delta t}{2} L} N U^{(2)} \end{aligned}$$

On peut calculer la fonction de stabilité de la méthode, on obtient alors :

$$\begin{aligned} U^{n+1} = & \left[e^{\Delta t L} + \Delta t \left(\frac{2}{3} e^{\frac{\Delta t}{2} L} N e^{\frac{\Delta t}{2} L} + \frac{1}{6} e^{\Delta t L} N + \frac{1}{6} N e^{\Delta t L} \right) \right. \\ & + \frac{\Delta t^2}{2} \left(\frac{1}{3} N e^{\Delta t L} N + \frac{1}{3} e^{\frac{\Delta t}{2} L} N e^{\frac{\Delta t}{2} L} N + \frac{1}{3} e^{\frac{\Delta t}{2} L} N e^{-\frac{\Delta t}{2} L} N e^{\Delta t L} \right) \\ & \left. + \frac{\Delta t^3}{6} e^{\frac{\Delta t}{2} L} N e^{-\frac{\Delta t}{2} L} N e^{\Delta t L} N \right] U^n \end{aligned}$$

La matrice d'amplification n'est plus un polynôme, nous nous proposons donc de faire une étude numérique. Pour cela on choisit une discréétisation des modes de Fourier $\kappa \in \llbracket -27, 27 \rrbracket$, et pour les pas de temps $\Delta t \in [0, 0.125]$. Pour tout couple de cette discréétisation on calcule numériquement les valeurs propres de la matrice d'amplification, qui sont au nombre de 5, sans compter la multiplicité, que l'on tri par valeur absolue. Cela nous permet de représenter graphiquement, sur la figure 3.1, la plus grande valeur propre en chaque point $(\kappa, \Delta t)$ de la discréétisation. On remarque une symétrie, attendue, entre les modes de Fourier positifs et négatifs, et c'est le mode le plus élevé qui va contraindre le plus notre condition de stabilité entre κ et Δt . On peut regarder, sur la figure 3.2a, pour le seul mode $\kappa = 15$, la valeur absolue des différentes valeurs propres de la matrice d'amplification. On remarque la présence d'une valeur propre égale à 1 ainsi que d'autres valeurs propres, qui sont strictement inférieure à 1 pour des petites valeurs de Δt (jusqu'à 0.11 environ). Pour estimer une condition de stabilité pour $\kappa = 15$, il est nécessaire de connaître la valeur de Δt telle que la plus grande valeur propre soit de module supérieur à 1. On trace sur la figure 3.2b, le module de la plus grande valeur propre de la matrice d'amplification, pour différents mode de Fourier $\kappa \in \llbracket 0, 27 \rrbracket$. On remarque que plus la discréétisation en x est fine (grande valeur des modes de Fourier) la valeur de Δt permettant d'assurer la stabilité diminue. On peut ainsi estimer une condition de stabilité pour $N_z = 15, 27$ et on obtient : pour $N_z = 15$, $\Delta t < 0.115 \approx \frac{\sqrt{3}}{15}$, et pour $N_z = 27$, $\Delta t < 0.066 \approx \frac{\sqrt{3}}{27}$.

On effectue une analyse similaire pour la méthode de Lawson LRK(4,4) suivante :

$$\begin{aligned} U^{(1)} &= e^{\frac{\Delta t}{2} L} U^n + \frac{\Delta t}{2} e^{\frac{\Delta t}{2} L} N U^n \\ U^{(2)} &= e^{\frac{\Delta t}{2} L} U^n + \frac{\Delta t}{2} N U^{(1)} \\ U^{(3)} &= e^{\Delta t L} U^n + \Delta t e^{\frac{\Delta t}{2} L} N U^{(2)} \\ U^{n+1} &= -\frac{1}{3} e^{\Delta t L} U^n + \frac{1}{3} e^{\frac{\Delta t}{2} L} U^{(1)} + \frac{2}{3} e^{\frac{\Delta t}{2} L} U^{(2)} + \frac{1}{3} U^{(3)} \frac{\Delta t}{6} N U^{(3)} \end{aligned}$$

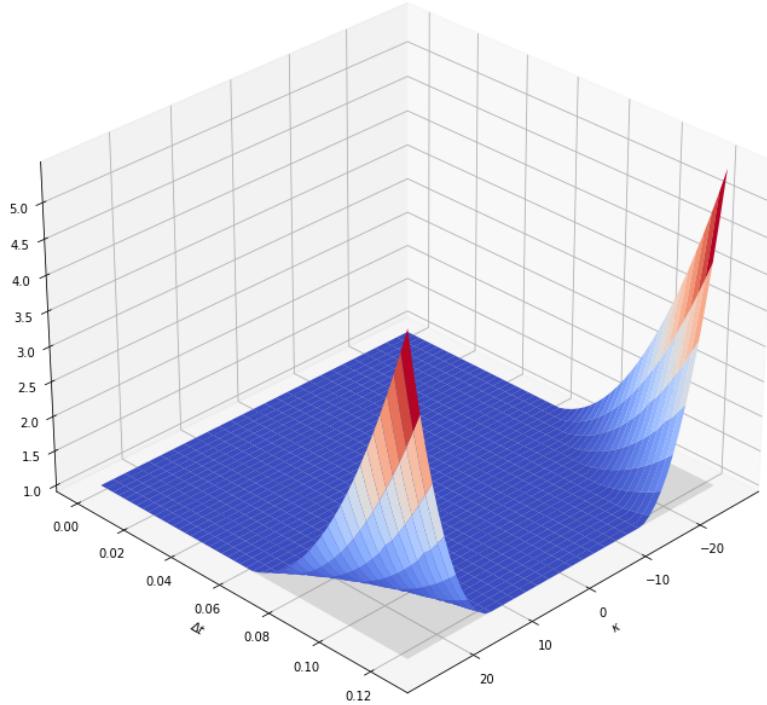
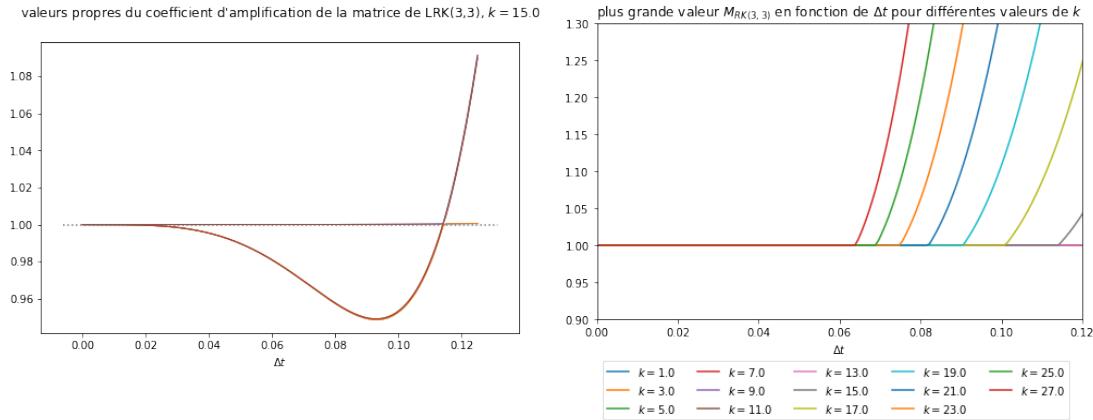


FIGURE 3.1 – Plus grande valeur propre en valeur absolue de la matrice d’amplification de la méthode LRK(3,3) sur l’espace discréteisé de $(\kappa, \Delta t) \in [-27, 27] \times [0, 0.125]$

Où l’on peut également calculer sa matrice d’amplification :

$$\begin{aligned} U^{n+1} = & \left[e^{\Delta t} + \Delta t \left(\frac{1}{6} e^{\Delta t L} N + \frac{2}{3} e^{\frac{\Delta t}{2} L} N e^{\frac{\Delta t}{2} L} + \frac{1}{6} N e^{\Delta t L} \right) \right. \\ & + \frac{\Delta t^2}{2} \left(\frac{1}{3} e^{\frac{\Delta t}{2} L} N e^{\frac{\Delta t}{2} L} N + \frac{1}{3} e^{\frac{\Delta t}{2} L} N^2 e^{\frac{\Delta t}{2} L} + \frac{1}{3} N e^{\frac{\Delta t}{2} L} N e^{\frac{\Delta t}{2} L} \right) \\ & + \frac{\Delta t^3}{6} \left(\frac{1}{2} e^{\frac{\Delta t}{2} L} N^2 e^{\frac{\Delta t}{2} L} N + \frac{1}{2} N e^{\frac{\Delta t}{2} L} N^2 e^{\frac{\Delta t}{2} L} \right) \\ & \left. + \frac{\Delta t^4}{24} N e^{\frac{\Delta t}{2} L} N^2 e^{\frac{\Delta t}{2} L} N \right] U^n \end{aligned}$$

Ce qui nous permet, avec la même discréétisation de κ et Δt de tracer les mêmes diagnostics que précédemment sur la figure 3.3. On représente ainsi à gauche le module des différentes valeurs propres de la matrice d’amplification de la méthode LRK(4,4) pour $\kappa = 15$, et à droite on ne s’intéresse qu’à la plus grande valeur propre pour différents modes de Fourier $\kappa \in [0, 27]$. Cela nous permet d’obtenir les conditions de stabilité pour différentes



- (a) Module des valeurs propres de la matrice d'amplification de LRK(3,3) pour le mode de Fourier $\kappa = 15$, $\Delta t \in [0, 0.12]$.
 (b) Module de la plus grande valeur propre de la matrice d'amplification de LRK(3,3) pour différents modes de Fourier $\kappa \in [0, 27]$ et $\Delta t \in [0, 0.12]$.

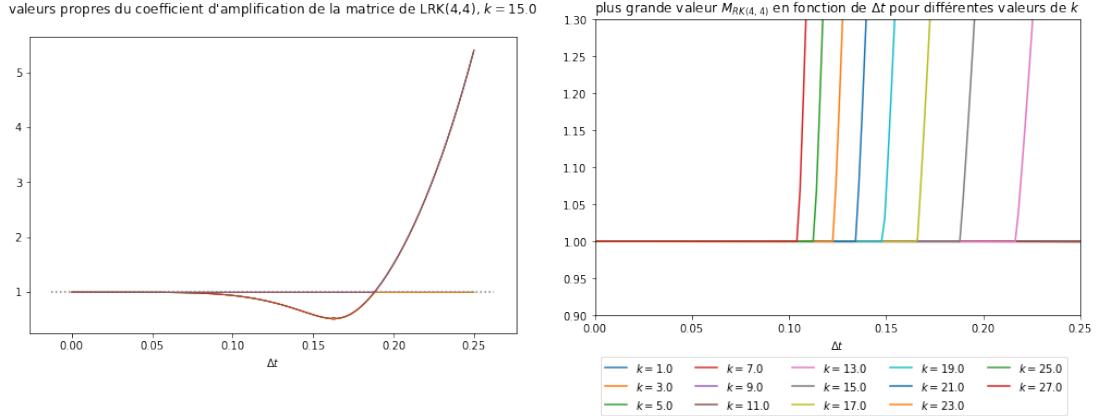
FIGURE 3.2 – Étude des valeurs propres de la matrice d'amplification de la méthode LRK(3,3) en fonction de Δt .

discrétisation en espace, et par exemple pour $N_z = 15$ on trouve $\Delta t < 0.188 \approx \frac{2\sqrt{2}}{15}$ et pour $N_z = 27$ on trouve $\Delta t < 0.104 \approx \frac{2\sqrt{2}}{27}$.

Cela conclut l'estimation de condition de stabilité en espace. On remarque que l'estimation numérique de la condition de stabilité de la méthode de Lawson est très proche de la condition de stabilité obtenue à l'aide de la restriction à la résolution de la méthode Runge-Kutta sur $(B_x, B_y, E_x, E_y)^\top$.

3.3.2.3 Synthèse d'un étage

Intéressons nous maintenant à l'implémentation de la méthode de Lawson. Les méthodes de type Runge-Kutta s'écrivent comme une succession d'étage et chaque étage possède des expressions différentes mais une structure similaire. Ces premières sont générées à l'aide d'outils de méta-programmation, présentés dans la sous-section 3.7.1, basés sur la bibliothèque Python de calcul symbolique SymPy. Le squelette d'un étage est présent dans l'algorithme 3.7. La résolution numérique des dérivées en espace z se font dans l'espace de Fourier, pour cette raison, les variables transmises entre étages ou itérations, sont des tableaux de complexes, correspondant aux modes de Fourier des différentes variables.



- (a) Module des valeurs propres de la matrice d'amplification de LRK(4,4) pour le mode de Fourier $\kappa = 15$, $\Delta t \in [0, 0.25]$.
 (b) Module de la plus grande valeur propre de la matrice d'amplification de LRK(4,4) pour différents modes de Fourier $\kappa \in [0, 27]$ et $\Delta t \in [0, 0.25]$.

FIGURE 3.3 – Étude des valeurs propres de la matrice d'amplification de la méthode LRK(4,4) en fonction de Δt .

L'algorithme d'un étage se comporte en deux grandes sections. La première partie permet de mettre à jour les variables spatiales $(j_{c,x}, j_{c,y}, B_x, B_y, E_x, E_y)$ dans l'espace de Fourier, qui commence à la ligne 1, qui commence par le calcul des courants induits par les particules chaudes $\int v_* f_h \, dv$ où $*$ $\in \{x, y\}$ dans l'espace de Fourier. Les lignes 7 à 12, sont générées à l'aide d'outils présentés dans la sous-section 3.7.1. La seconde partie permet la mise à jour de f_h dans l'espace de Fourier et commence à la lignes 14. Cette partie commence par le calcul de $\mathbf{E} + (\mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_h$ à l'aide de la méthode WENO, puis continue par la mise à jour de f_h dans l'espace de Fourier en effectuant la transformée de Fourier en z de $\mathbf{E} + (\mathbf{v} \times \mathbf{B}) \cdot \nabla_{\mathbf{v}} f_h$ pour une vitesse \mathbf{v} donnée. La ligne 27 est également générée automatiquement.

Étudions maintenant la complexité de l'algorithme 3.7. La première partie de l'algorithme (ligne 1 à 14) possède une complexité temporelle de $\mathcal{O}(2N_z)$ et nécessite la sauvegarde de deux tableaux, $\hat{j}_{j,x}, \hat{j}_{h,y}$, de taille N_z (tableaux de complexes). La deuxième partie de l'algorithme (ligne 14 à 29) nécessite tout d'abord le calcul de la transformée de Fourier inverse de tout \hat{f}_h , étape d'une complexité de $\mathcal{O}(N_{v_x} N_{v_y} N_{v_z} N_z \log(N_z))$. Il est ensuite nécessaire d'effectuer une boucle dans tout l'espace des phases pour calculer l'approximation $\partial_v f_h$, la méthode WENO peut se résumer à un calcul arithmétique à partir des 7 valeurs en entré et est donc considéré de complexité constante,

Algorithme 3.7 Squelette de l'algorithme d'un étage s d'une méthode LRK

```

1:  $\triangleright$  Calcul des variables  $\hat{j}_{c,x}^{(s)}$ ,  $\hat{j}_{c,y}^{(s)}$ ,  $\hat{B}_x^{(s)}$ ,  $\hat{B}_y^{(s)}$ ,  $\hat{E}_x^{(s)}$  et  $\hat{E}_y^{(s)}$ 
2: pour  $i = 0, \dots, N_z$  faire :
3:    $\hat{j}_{h,x,[i]} \leftarrow \sum_{k_x, k_y, k_z} v_{k_x} \hat{f}_{h,[i,k_x,k_y,k_z]}^{(s-1)} \Delta v$ 
4:    $\hat{j}_{h,y,[i]} \leftarrow \sum_{k_x, k_y, k_z} v_{k_y} \hat{f}_{h,[i,k_x,k_y,k_z]}^{(s-1)} \Delta v$ 
5: fin pour
6: pour  $i = 0, \dots, N_z$  faire :
7:    $\hat{j}_{c,x,[i]}^{(s)} \leftarrow \dots$   $\triangleright$  les expressions ici sont données par SymPy
8:    $\hat{j}_{c,y,[i]}^{(s)} \leftarrow \dots$ 
9:    $\hat{B}_{x,[i]}^{(s)} \leftarrow \dots$ 
10:   $\hat{B}_{y,[i]}^{(s)} \leftarrow \dots$ 
11:   $\hat{E}_{x,[i]}^{(s)} \leftarrow \dots$ 
12:   $\hat{E}_{y,[i]}^{(s)} \leftarrow \dots$ 
13: fin pour
14:  $\triangleright$  Calcul de la variable  $\hat{f}_h^{(s)}$ 
15:  $(f)_h_{\cdot, k_x, k_y, k_z} \leftarrow \text{iFFT}_z \left( \hat{f}_{h,\cdot,k_x,k_y,k_z}^{(s-1)} \right)$ 
16: pour tout  $(k_x, k_y, k_z) \in [0, N_x] \times [0, N_y] \times [0, N_z]$  faire :
17:   pour  $i = 0, \dots, N_z$  faire :
18:      $a_{v_x} \leftarrow E_{x,[i]} + v_{k_y} B_0 + v_{k_z} B_y, [i]$ 
19:      $a_{v_y} \leftarrow E_{y,[i]} + v_{k_x} B_0 + v_{k_z} B_x, [i]$ 
20:      $a_{v_z} \leftarrow v_{k_x} B_{y,[i]} + v_{k_y} B_{x,[i]}$ 
21:      $\partial_v f_{h,[i,k_x,k_y,k_z]} \leftarrow \text{WENO}(a_{v_x}, f_{h,[i,k_x-3:k_x+3,k_y,k_z]}) + \text{WENO}(a_{v_y}, f_{h,[i,k_x,k_y-3:k_y+3,k_z]})$ 
         $+ \text{WENO}(a_{v_z}, f_{h,[i,k_x,k_y,k_z-3:k_z+3]})$ 
22:   fin pour
23: fin pour
24: pour tout  $(k_x, k_y, k_z) \in [0, N_x] \times [0, N_y] \times [0, N_z]$  faire :
25:    $(\widehat{\partial_v f})_i \leftarrow \text{FFT}_z(\partial_v f_{\cdot, k_x, k_y, k_z})$ 
26:   pour  $i = 0, \dots, N_z$  faire :
27:      $\hat{f}_h^{(s)} \leftarrow \dots$   $\triangleright$  l'expression ici est donnée par SymPy
28:   fin pour
29: fin pour

```

ce calcul a donc une complexité de $\mathcal{O}(N_{v_x}N_{v_y}N_{v_z}N_z)$. Cette partie se termine par la mise à jour de \hat{f}_h de l'étage, il est possible de n'effectuer la transformée de Fourier pour une tranche de données au triplet (k_x, k_y, k_z) fixé, on obtient une complexité de $\mathcal{O}(N_{v_x}N_{v_y}N_{v_z}(N_z \log(N_z) + N_z)) = \mathcal{O}(N_{v_x}N_{v_y}N_{v_z}N_z \log(N_z))$. Un étage de la méthode de Lawson possède donc une complexité temporelle de $\mathcal{O}(N_{v_x}N_{v_y}N_{v_z}N_z \log(N_z))$ et nécessite 3 tableaux de taille N_z de valeurs complexes ($\hat{j}_{h,x}$, $\hat{j}_{h,y}$ et $\widehat{\partial_v f}$), et 2 tableaux de taille $N_{v_x}N_{v_y}N_{v_z}N_z$ (f_h de réels et $\partial_v f_h$ de complexes), ces tableaux temporaires peuvent être réutilisés par l'étage suivant sans nécessité de nouvelle allocation de mémoire.

Les résultats de chaque étage de la méthode de Lawson nécessitent d'être sauvegarder pour le calcul des étages suivants, ainsi chaque étage nécessite la sauvegarde de 6 tableaux de taille N_z de complexes et d'un tableau de taille $N_{v_x}N_{v_y}N_{v_z}N_z$ aussi de complexes. Ainsi pour une méthode de Lawson induite par une méthode RK(s,n) il sera nécessaire de multiplier l'espace mémoire nécessaire par le nombre d'étages s . Il est possible de minimiser l'espace mémoire nécessaire en utilisant des méthodes de type Runge-Kutta dites *low storage*, qui permettent de ne pas conserver les données de chaque étage pour calculer les suivants (voir [53]).

La mise en place de l'opération de filtrage dans le pseudo-code 3.7 nécessite seulement de modifier le calcul des variables de courants chauds $(\hat{j}_{h,x})_i$, $(\hat{j}_{h,y})_i$ (lignes 3 et 4) et des vitesses d'advection a_{v_x} , a_{v_y} et a_{v_z} (lignes 18 à 20). L'opération est présentée dans l'extrait d'algorithme 3.8. *\hat{j} ou \hat{j} ? telle est la question...(\jmathmath ou non)*

Algorithme 3.8 Modifications pour faire le filtrage dans l'algorithme 3.7

```

 $\tau^{n,s} \leftarrow t^n + c_s \Delta t$ 
3:  $\hat{j}_{h,x,[i]} \leftarrow \sum_{k_1,k_2,k_z} (w_1 \cos(B_0 \tau^{n,s}) - w_2 \sin(B_0 \tau^{n,s})) \hat{g}_{[i,k_1,k_2,k_z]} \Delta w \Delta v_z$ 
4:  $\hat{j}_{h,y,[i]} \leftarrow \sum_{k_1,k_2,k_z} (w_1 \sin(B_0 \tau^{n,s}) + w_2 \cos(B_0 \tau^{n,s})) \hat{g}_{[i,k_1,k_2,k_z]} \Delta w \Delta v_z$ 
18:  $a_{v_x} \leftarrow E_{x,[i]} \cos(B_0 \tau^{n,s}) + E_{y,[i]} \sin(B_0 \tau^{n,s}) + v_z B_{x,[i]} \sin(B_0 \tau^{n,s}) - v_z B_{y,[i]} \cos(B_0 \tau^{n,s})$ 
19:  $a_{v_y} \leftarrow -E_{x,[i]} \sin(B_0 \tau^{n,s}) + E_{y,[i]} \cos(B_0 \tau^{n,s}) + v_z B_{x,[i]} \cos(B_0 \tau^{n,s}) + v_z B_{y,[i]} \sin(B_0 \tau^{n,s})$ 
20:  $a_{v_z} \leftarrow -B_{x,[i]} (w_1 \sin(B_0 \tau^{n,s}) + w_2 \cos(B_0 \tau^{n,s})) + B_{y,[i]} (w_1 \cos(B_0 \tau^{n,s}) - w_2 \sin(B_0 \tau^{n,s}))$ 

```

3.4 Résultats numériques

Dans cette section on s'intéresse à des résultats numériques pour la résolution du modèle VHL par les méthodes présentées précédemment. Conformément à [50], nous consi-

dérons la condition initiale suivante :

$$f_h(t = 0, z, \mathbf{v}) = \frac{\rho_h}{(2\pi)^{3/2}\bar{v}_\parallel\bar{v}_\perp^2} \exp\left(-\frac{v_z^2}{2\bar{v}_\parallel^2} - \frac{(v_x^2 + v_y^2)^2}{2\bar{v}_\perp^2}\right)$$

avec $z \in [0, \frac{2\pi}{k}]$, $k = 2$, $\bar{v}_\parallel = 0.2$, $\bar{v}_\perp = 0.6$, $\rho_h = 0.2$ et $B_x(t = 0, z) = \epsilon \sin(kz)$. Les autres inconnues du système ($j_{c,x}, j_{c,y}, B_y, E_x, E_y$) sont initialisées à zéro. Le domaine en vitesse est restreint à $\mathbf{v} \in [-3.6, 3.6] \times [-3.6, 3.6] \times [-2.4, 2.4]$ et on note N_z , N_{v_x} , N_{v_y} , N_{v_z} le nombre de points de discréétisation dans chaque direction. Dans chacune des simulations, on s'intéresse à l'évolution temporelle des énergies suivantes :

Énergie magnétique :

$$\mathcal{H}_B(t) = \frac{1}{2} \int (B_x^2(t, z) + B_y^2(t, z)) dz \quad (3.20)$$

Énergie électrique :

$$\mathcal{H}_E(t) = \frac{1}{2} \int (E_x^2(t, z) + E_y^2(t, z)) dz \quad (3.21)$$

Énergie cinétique des particules froides :

$$\mathcal{H}_c(t) = \frac{1}{2\Omega_{pe}^2} \int (j_{c,x}^2(t, z) + j_{c,y}^2(t, z)) dz \quad (3.22)$$

Énergie cinétique des particules chaudes :

$$\mathcal{H}_h(t) = \frac{1}{2} \iint |\mathbf{v}|^2 f_h(t, z, \mathbf{v}) d\mathbf{v} dz \quad (3.23)$$

dont la somme, l'énergie totale, est préservée au cours du temps :

$$\frac{d\mathcal{H}}{dt} = \frac{d\mathcal{H}_B + \mathcal{H}_E + \mathcal{H}_c + \mathcal{H}_h}{dt} = 0.$$

Les diagnostiques que nous regarderons pour vérifier la validité des résultats sont les différentes énergies au cours du temps, en vérifiant le taux d'instabilité donné par les relations de dispersion, présentées dans [50]. La comparaison entre les méthodes de résolution se fera à partir de la préservation de l'énergie totale \mathcal{H} , et de l'erreur relative produite par chacune des méthodes, nous comparerons également les temps de calcul.

3.4.1 Comparaison des solveurs à pas de temps constant

Dans un premier temps nous considérons la méthode de *splitting* et de Lawson avec un pas de temps constant, comme présenté dans la section 3.3. Nous présentons trois méthodes de *splitting*, à savoir les méthodes de Lie, Strang et de Suzuki. La méthode de Suzuki ne sera présente qu'à titre d'illustration mais celle-ci n'est pas très intéressante dans un contexte multidimensionnel, en effet ses résultats sont très similaires à la méthode de Strang, mais son temps de calcul représente 5 fois celui de cette méthode d'ordre 2. Dès que l'hamiltonien se subdivise en de nombreuses parties (7 dans notre cas), le nombre d'étapes requises pour obtenir de l'ordre élevé augmente drastiquement (même si certaines stratégies peuvent être utilisées pour éviter cela, comme présentées dans [10]). La méthode de Strang nécessite à elle seule 15 étapes, ce qui représente $5 \times 15 = 75$ étapes par itération, ce qui est trop important pour que la méthode soit compétitive. Nous présenterons également deux méthodes de Lawson, la méthode LRK(3,3) et LRK(4,4) induites respectivement par la méthode RK(3,3) de Shu-Osher et la méthode RK(4,4) dite classique. Contrairement aux méthodes de *splitting*, les méthodes de Lawson ne sont pas trop impactées par la montée en ordre dans le sens où le nombre d'étages nécessaire correspond, *à peu près*, à l'ordre de la méthode. De plus, il existe une souplesse intrinsèque aux méthodes de Lawson qui réside dans le choix de la partie linéaire. Ce choix permet, par exemple, de se soustraire de certaines conditions de stabilités, sous contrainte du calcul de l'exponentielle de cette partie, problème contourné dans la section 3.5. De plus le filtrage possible du terme $\mathbf{v} \times \mathbf{B}$ est intéressant dans les méthodes de Lawson car il permet de relaxer la condition de stabilité de la méthode. Un filtrage similaire serait envisageable dans les méthodes de *splitting*, avec comme coût l'ajout de sous-étapes supplémentaires dans l'étape \mathcal{H}_{f_h} , étape déjà la plus coûteuse numériquement.

L'utilisation de schéma explicites introduit une condition de stabilité dans la résolution des équations de Maxwell, dans les deux approches (méthode de *splitting* et de Lawson), présentée dans [55]. Il est également possible de se concentrer uniquement sur les inconnues (B_x, B_y, E_x, E_y) pour déduire une condition de stabilité, comme vu dans la sous-section 3.3.2.2. Le tableau 3.1 résume des différentes conditions de stabilité. Pour les méthodes de Lawson une condition de CFL supplémentaire provient de la discréttisation en vitesse, mais on remarque numériquement que cette dernière est moins contraignante que la condition provenant des équations de Maxwell. Nous verrons dans la section 3.5 comment s'affranchir de cette condition en intégrant les équations de Maxwell dans la

partie linéaire de la méthode de Lawson.

methode	condition de CFL
Lie	$(\sqrt{2}/\pi)\Delta z$
Strang	$(2/\pi)\Delta z$
Lawson-RK(3, 3)	$(\sqrt{3}/\pi)\Delta z$
Lawson-RK(4, 4)	$(2\sqrt{2}/\pi)\Delta z$

TABLE 3.1 – Condition CFL de stabilité provenant des équations de Maxwell en utilisant différents intégrateurs.

Nous traçons, sur la figure 3.4 l'évolution des énergies électrique, magnétique et l'énergie cinétique provenant des particules froides, définies précédemment, en échelle semi-log, pour différentes méthodes de *splitting* (méthode de Lie, de Strang, et de Suzuki), et différentes méthodes de Lawson (LRK(3,3) et LRK(4,4)). On choisit le maillage $27 \times 32 \times 32 \times 41$ et $\Delta t = 0.05$ (ce qui assure la stabilité de toutes les méthodes présentées). On considère une faible perturbation $\epsilon = 10^{-5}$, pour assurer une longue phase linéaire (jusqu'au temps $t \approx 100$) tandis que la phase non-linéaire se développe jusqu'au temps final $t = 200$. Dans la partie linéaire, il est possible de comparer les résultats numériques avec les solutions des relations de dispersion présentées dans [50]. Comme attendu, les trois énergies suivent une croissance exponentielle, avec le taux prédit par la théorie linéaire, ce qui se traduit des transferts d'énergie des particules rapides (chaudes) vers l'énergie électro-magnétique et aux particules froides. Après la phase linéaire, les champs saturent en amplitude, ce qui marque l'effet des termes non-linéaires, la théorie linéaire n'est plus valide. On remarque que les deux classes de méthodes capture efficacement les différents phénomènes en jeu dans la phase linéaire, avec une très bonne corrélation avec les taux d'instabilités calculés à l'aide de la théorie linéaire. De plus les différentes méthodes saturent à des niveaux d'énergie très similaires.

La figure 3.5 représente l'erreur relative sur l'énergie totale, et permet d'étudier la conservation de celle-ci par les différentes méthodes. Bien que le maillage en espace soit très grossier ($\Delta z = \frac{2\pi}{27} \approx 0.23$) l'énergie totale est plutôt bien préservée : aux alentours de 8% pour les deux méthodes de Lawson présentées, et autour de 5% d'erreur pour les méthodes de Lie et de Strang. La méthode de Suzuki produit bien plus d'erreur (autour de 12%) très certainement à cause des multiples transformées de Fourier selon l'axe z effectuées avec peu de points de discréétisation. Comme attendu, et observé dans le cas $1dx - 1dv$, les méthodes basées sur un *splitting* hamiltonien ont une bonne préservation de

l'énergie totale, mais les qualités de leurs résultats, surtout observable avec la méthode de Suzuki, suggère la nécessité de raffiner le maillage. Il est à souligner que le coût numérique de la méthode de Strang, est environ deux fois plus important que celui de la méthode LRK(3,3).

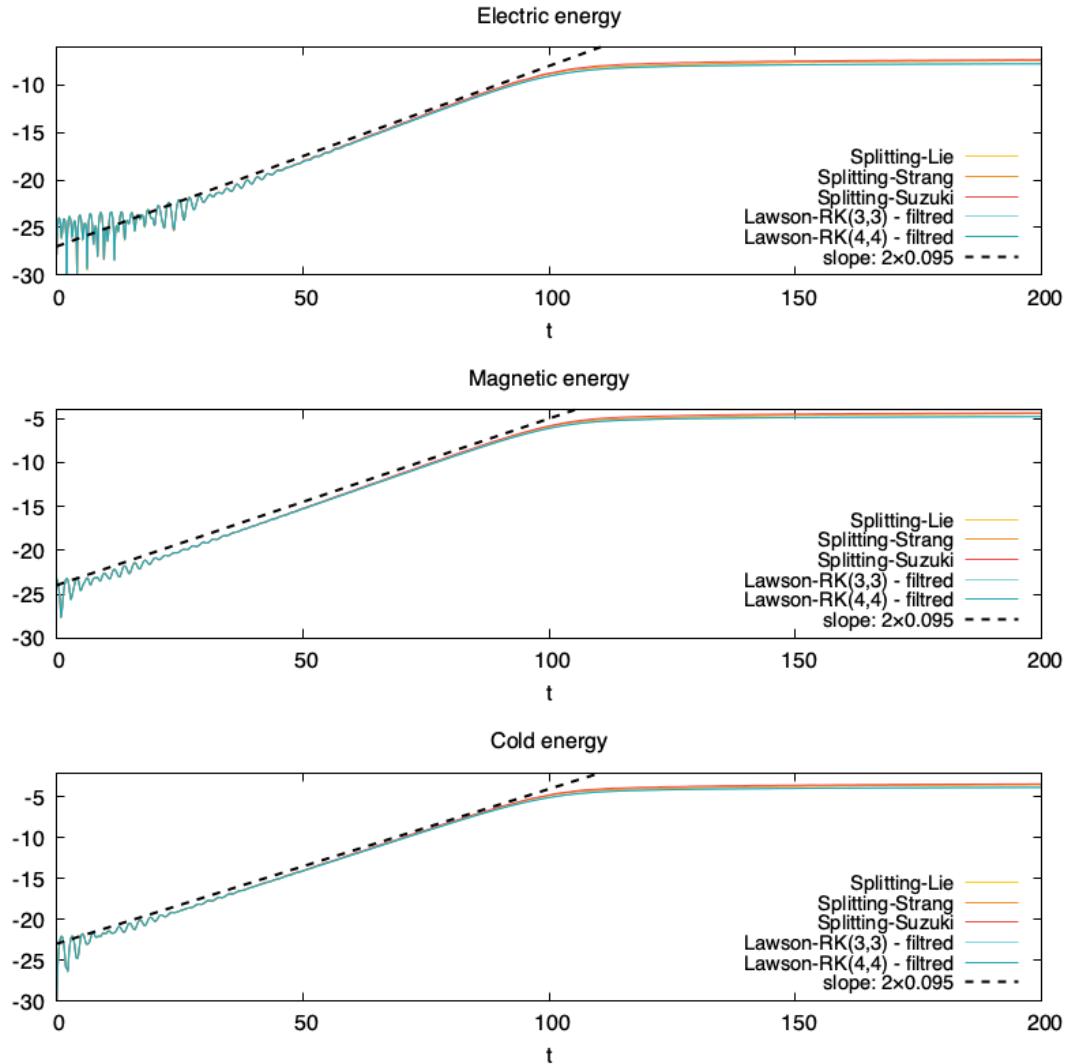


FIGURE 3.4 – Évolution de l'énergie électrique, magnétique et l'énergie cinétique des particules froides définies dans les équations 3.20-3.22, en échelle semi-log, pour les méthodes de *splitting* de Lie, Strang et Suzuki, et pour les méthodes de Lawson LRK(3,3) et LRK(4,4). $\Delta t = 0.05$, $N_x = 27$, $N_{v_x} = 32$, $N_{v_y} = 32$, $N_{v_z} = 41$.

On peut étudier l'importance du filtrage du terme $\mathbf{v} \times \mathbf{B}$ avec la figure 3.6, où l'on trace l'évolution de l'énergie électrique pour la méthode de Lawson LRK(3,3) avec ou sans

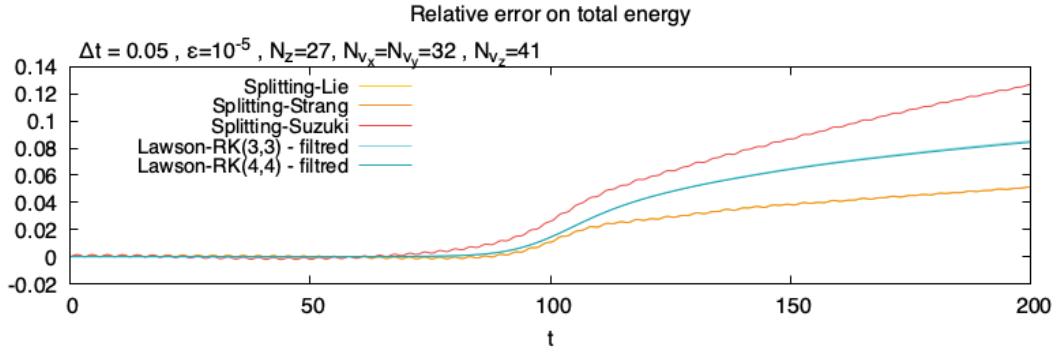


FIGURE 3.5 – Évolution de l’erreur relative sur l’énergie totale, pour les méthodes de *splitting* de Lie, Strang et Suzuki, et pour les méthodes de Lawson LRK(3,3) et LRK(4,4). $\Delta t = 0.05, N_x = 27, N_{v_x} = 32, N_{v_y} = 32, N_{v_z} = 41$.

le filtrage (voir l’algorithme 3.8 pour la différence algorithmique entre les deux méthodes) pour deux maillages différents de l’espace des phases ($15 \times 20 \times 20 \times 41$ à gauche et $15 \times 32 \times 32 \times 41$ à droite) avec un pas de temps $\Delta t = 0.1$. On peut observer l’impacte sur les résultats du raffinement dans la direction v_\perp sur la méthode non-filtrée, alors que sur la méthode filtrée on obtient un taux d’instabilité satisfaisant avec le maillage le plus grossier.

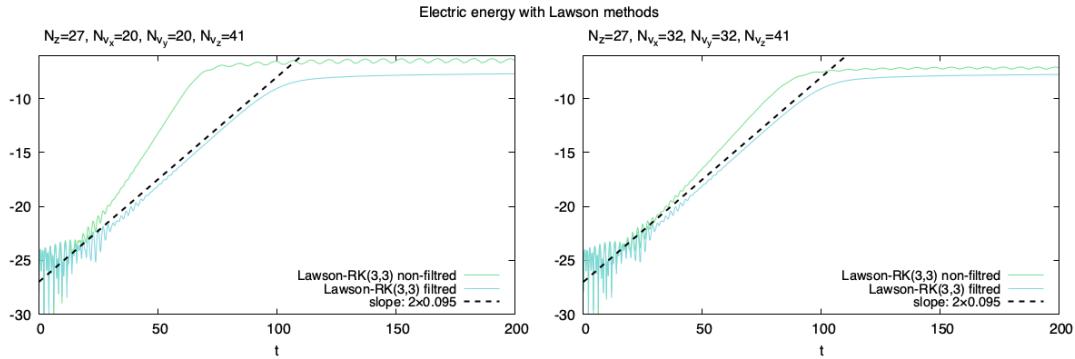


FIGURE 3.6 – Time evolution of the electric energy for the filtered and unfiltered versions of Lawson-RK(3, 3). Left : $\Delta t = 0.05, N_x = 27, N_{v_x} = 20, N_{v_y} = 20, N_{v_z} = 41$. Right : $\Delta t = 0.05, N_x = 27, N_{v_x} = 32, N_{v_y} = 32, N_{v_z} = 41$

3.4.2 Étude à pas de temps adaptatif

Le dernier test que nous souhaitons effectuer avec ces méthodes de résolution, est de considérer la méthode de Lawson à pas de temps adaptatif induite par le schéma DP4(3).

Pour que l'approche à pas de temps adaptatif soit intéressante dans notre cas il faut que :

- (i) Les champs électromagnétiques soient suffisamment faible dans la phase linéaire pour pouvoir prendre des grands pas de temps,
- (ii) La condition de stabilité provenant des équations de Maxwell ne doit pas trop restrictive,
- (iii) Dans la phase non-linéaire, l'estimateur d'erreur nous assure la stabilité.

En pratique, à cause de la discrétisation en espace la condition (ii) n'est pas satisfaite.

Pour la suite nous considérons les paramètres numériques suivant : $N_z = 27$, $N_{v_x} = 32$, $N_{v_y} = 32$, $N_{v_z} = 41$, $tol = 6 \times 10^{-5}$ et le pas de temps calculé par :

$$\Delta t_0 = \frac{C}{2} \Delta z, \quad \Delta t_{n+1} = \min \left(\max \left(\sqrt[p]{\frac{tol}{L_{[p]}^{n+1}}} \Delta t_n, \frac{C}{2} \Delta z \right), 3C \Delta z \right), n \geq 0$$

où $L_{[p]}^{n+1}$ est l'erreur locale calculée par 2.22, et C est donné par le tableau 3.1. La figure 3.7 montre l'énergie électrique en échelle semi-log. Comme pour les méthodes présentées dans la section 2.7.2 les résultats sur les énergies sont très similaires, on capture le bon taux d'instabilité. On regarde sur la figure 3.8 deux diagnostiques sur la méthode LDP4(3). Tout d'abord l'évolution de la taille des pas de temps au cours du temps (figure du haut). La simulation avec LDP4(3) est initialisée avec un pas de temps $\Delta t_0 = 0.05 \approx \frac{1}{2} \frac{2\sqrt{2}}{N_z}$, les 3 itérations suivantes sont effectuée avec le pas de temps $\Delta t_n = 3 \frac{2\sqrt{2}}{N_z}$ qui est la borne supérieure que peut prendre le pas de temps Δt_n (pour minimiser l'erreur générée en début de simulation, les champs électromagnétiques étant très faibles, l'estimateur d'erreur $L_{[p]}^{n+1}$ permet de prendre des pas de temps supérieur à 1); les autres itérations sont effectuées avec un pas de temps très proche de la condition de stabilité provenant des équations de Maxwell. Dans la phase linéaire, jusqu'au temps $t \approx 100$, la condition (ii) n'est pas satisfaite, la résolution des équations de Maxwell impose une condition CFL forte. Dans la phase non-linéaire, du temps $t \approx 100$ jusqu'au temps final $t = 200$, la condition de stabilité induite par le transport dans les directions (v_x, v_y, v_z) n'est pas trop contraignante, ce qui permet de satisfaire la condition (iii) avec une contrainte pas trop forte. On remarque tout de même que la méthode LDP4(3) permet de prendre des pas de temps légèrement supérieur à cette condition de stabilité, en effet 16% des itérations réussites ont un pas de temps Δt_n supérieur à la condition de stabilité $\frac{2\sqrt{2}}{N_z}$. La deuxième partie de la figure 3.8, celle-du bas, présente l'erreur locale au cours du temps, sont tracées également les itérations rejetées à l'aide des carrés. Contrairement au cas $1dx - 1dv$, où

le surcoût engendré par les itérations rejetées était négligeable, ici, seulement 71% des itérations sont acceptées par le critère d’erreur.

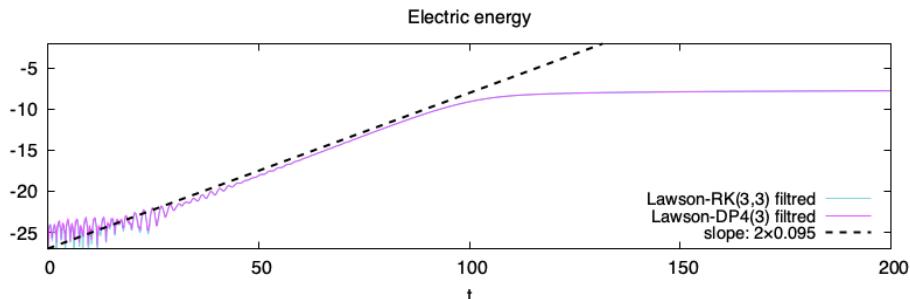


FIGURE 3.7 – Évolution de l’énergie électrique définie dans (3.21), en échelle semi-log pour les méthodes LRK(3,3) ($\Delta t = 0.05$) et LDP4(3). $N_x = 27$, $N_{v_x} = 32$, $N_{v_y} = 32$, $N_{v_z} = 41$.

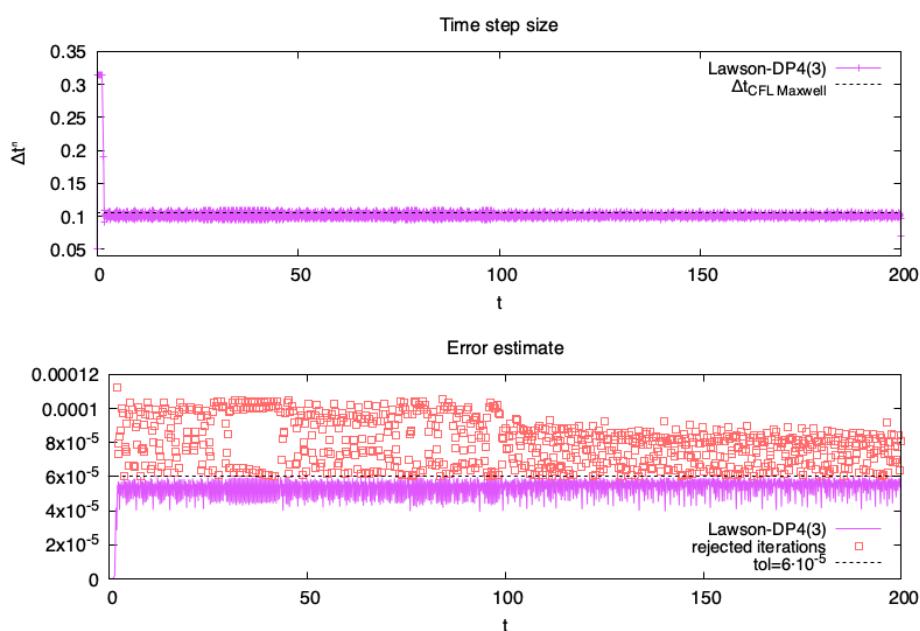


FIGURE 3.8 – Évolution au cours du temps de la taille du pas du temps (haut) et de l’erreur locale (bas) pour la méthode de Lawson LDP4(3). $N_x = 27$, $N_{v_x} = 32$, $N_{v_y} = 32$, $N_{v_z} = 41$.

3.5 Approximation de la partie linéaire

L’obtention, à l’aide d’un logiciel de calcul formel, de l’exponentielle de la partie linéaire n’est pas toujours envisageable. Il est possible de recourir à une méthode d’approximation.

mation pour obtenir une formulation formelle de celle-ci qui sera possible d'utiliser pour l'écriture du code de simulation. On s'intéressera dans cette section à la partie linéaire L définie par :

$$L = \begin{pmatrix} 0 & -B_0 & 0 & 0 & \Omega_{pe}^2 & 0 & 0 \\ B_0 & 0 & 0 & 0 & 0 & \Omega_{pe}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \partial_z & 0 \\ 0 & 0 & 0 & 0 & -\partial_z & 0 & 0 \\ -1 & 0 & 0 & -\partial_z & 0 & 0 & 0 \\ 0 & -1 & \partial_z & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -v_z \partial_z \end{pmatrix}$$

Cette matrice est de la forme :

$$L = \begin{pmatrix} A & 0 \\ 0 & -v_z \partial_z \end{pmatrix}$$

matrice diagonale par blocs, dont seul le bloc A pose problème pour calculer formellement l'exponentielle. Ainsi on s'intéressera surtout à la sous-matrice A obtenue après une transformée de Fourier en z du système :

$$A = \begin{pmatrix} 0 & -1 & 0 & 0 & 4 & 0 \\ 1 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & ik \\ 0 & 0 & 0 & 0 & -ik & 0 \\ -1 & 0 & 0 & -ik & 0 & 0 \\ 0 & -1 & ik & 0 & 0 & 0 \end{pmatrix}$$

Par abus de notation, nous noterons A_0 , la matrice A pour $k = 0$, ce qui revient à une partie linéaire sans les équations de Maxwell, ceci sera utile lors de la comparaison des résultats entre les méthodes.

3.5.1 Troncature de la série de Taylor

On peut définir e^{tA} par la série de Taylor :

$$e^{tA} = \sum_{n=0}^{\infty} \frac{t^n A^n}{n!}.$$

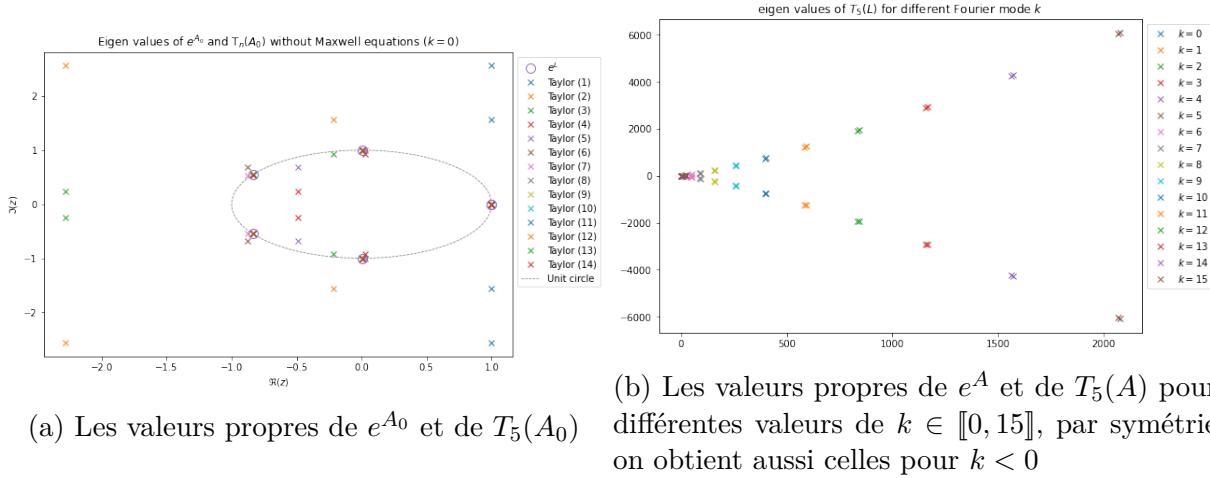


FIGURE 3.9 – Valeurs propres de e^A et de $T_5(A)$ pour $k = 0$ (sans les équations de Maxwell) à gauche, et pour différentes valeurs de $k \in [0, 15]$ à droite.

Une troncature d'ordre suffisamment élevé permet d'obtenir une approximation de l'exponentielle e^{tA} à un ordre plus élevé que la méthode LRK(s, n) où elle sera utilisé garantie que l'erreur de troncature reste inférieur à n , l'ordre de la méthode en temps. On définit la troncature de la série de Taylor à l'ordre p par :

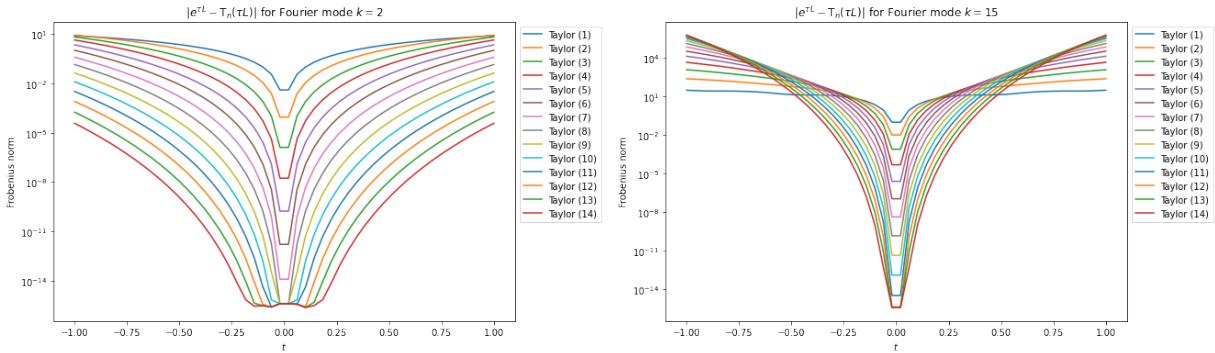
$$T_p(A) = \sum_{k=0}^p \frac{A^k}{k!}$$

On sait que les valeurs propres de A sont imaginaires pures, cela signifie que les valeurs propres de e^A sont de norme 1.

Je ne sais pas trop quoi dire sur les figures, donc je vais les mettre là un peu en vrac, on pourra discuter de leur intérêt plus tard, mais je pense qu'un petit calcul juste dire que les valeurs propres de $T_p(A)$ ne sont pas de module 1 serait plus intéressant.

3.5.2 Approximant de Padé

Pour approcher une fonction, au lieu d'utiliser un polynôme comme dans le cadre des séries des développements limités, il est possible de construire une fraction rationnelle. L'approximant de Padé de la fonction exponentielle est la meilleure approximation de la



(a) L'erreur absolue locale $\|e^{tA} - T_p(tA)\|$ pour le mode de Fourier $k = 2$ (b) L'erreur absolue locale $\|e^{tA} - T_p(tA)\|$ pour le mode de Fourier $k = 15$

FIGURE 3.10 – Erreur absolue locale $\|e^{tA} - T_p(tA)\|$ pour deux modes de Fourier $k = 2$ à gauche et $k = 15$ à droite.

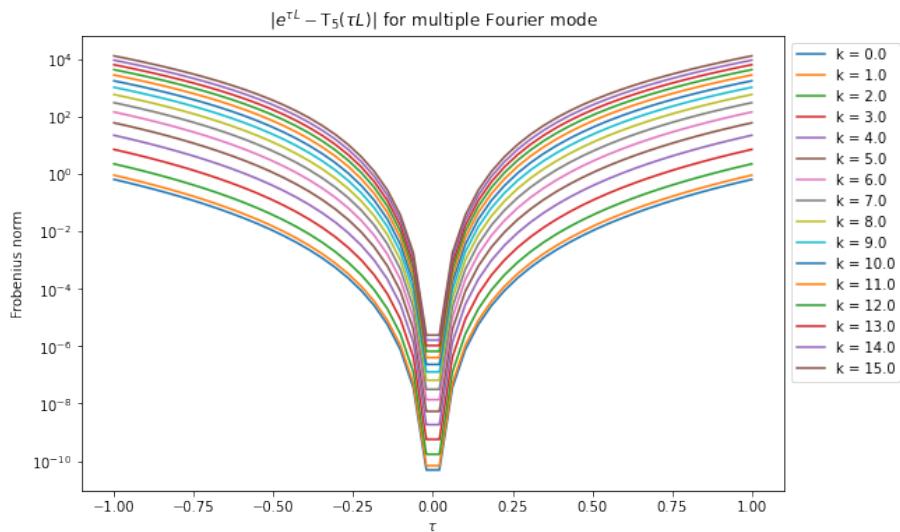
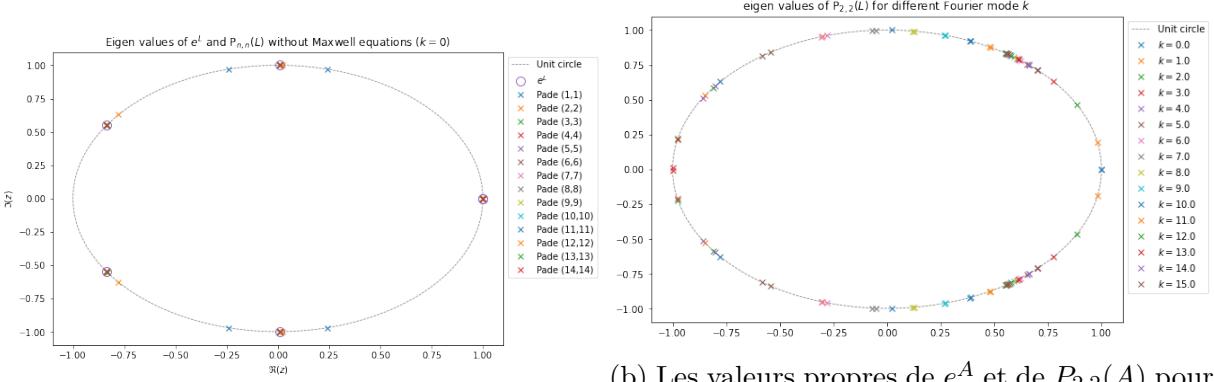


FIGURE 3.11 – L'erreur absolue locale $\|e^{tA} - T_5(tA)\|$ pour différents mode de Fourier



(a) Les valeurs propres de e^{A_0} et de $P_{n,n}(A_0)$ pour différentes valeurs de $k \in [0, 15]$, par symétrie on obtient aussi celles pour $k < 0$

(b) Les valeurs propres de e^A et de $P_{2,2}(A)$ pour

FIGURE 3.12 – Valeurs propres de e^A et de $P_{n,n}(A)$ pour $k = 0$ (sans les équations de Maxwell) à gauche, et pour différentes valeurs de $k \in [0, 15]$ à droite.

fonction exponentielle par une fraction rationnelle et est définie par :

$$h_{p,q}(x) = \sum_{i=0}^p \frac{\frac{p!}{(p-i)!}}{\frac{(p+q)!}{(p+q-i)!}} \frac{x^i}{i!}$$

$$k_{p,q}(x) = \sum_{j=0}^q (-1)^j \frac{\frac{q!}{(q-j)!}}{\frac{(p+q)!}{(p+q-j)!}} \frac{x^j}{j!}$$

$$p_{p,q}(x) = \frac{h_{p,q}(x)}{k_{p,q}(x)} \approx e^x$$

Pour utiliser cet approximant de Padé, qui est une fraction rationnelle, avec des matrices il faut utiliser la définition suivante :

$$e^M \approx P_{p,q}(M) = h_{p,q}(M) \cdot (k_{p,q}(M))^{-1}$$

On effectue la même étude qu'avec une troncature de la série de Taylor. On regarde donc dans un premier temps sur la figure 3.12 les valeurs propres dans le cas $k = 0$ et pour différentes valeurs de k .

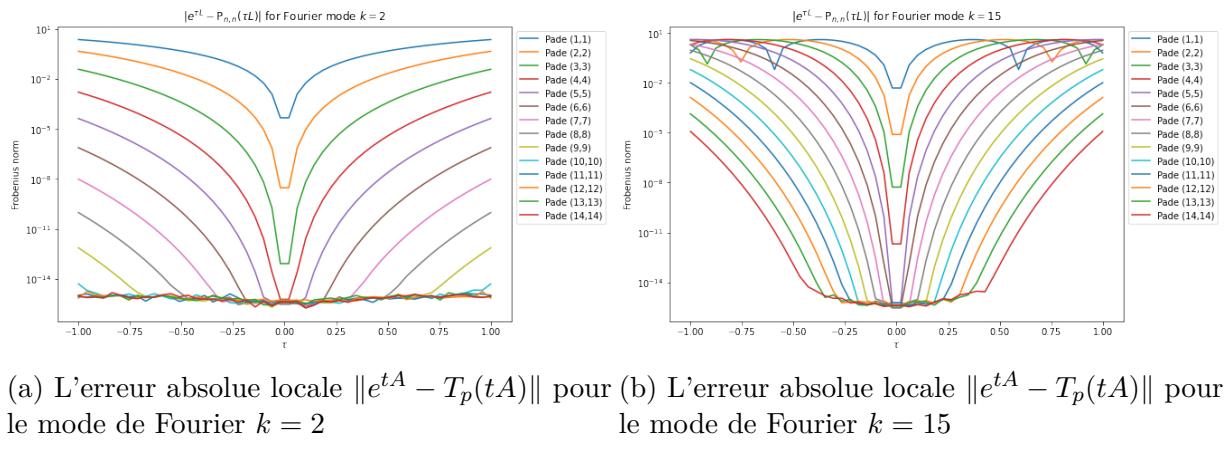


FIGURE 3.13 – Erreur absolue locale $\|e^{tA} - T_p(tA)\|$ pour deux modes de Fourier $k = 2$ à gauche et $k = 15$ à droite.

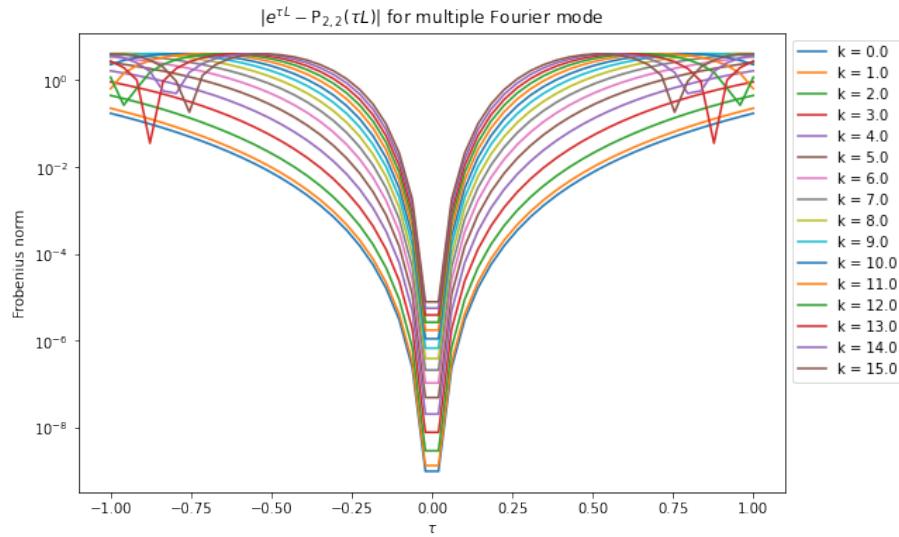


FIGURE 3.14 – L'erreur absolue locale $\|e^{tA} - T_5(tA)\|$ pour différents mode de Fourier

3.6 Résultats numériques des schémas de Lawson approchés

3.7 Optimisations et performances

3.7.1 Génération automatique de code

Je propose de mettre ceci sous forme d'une section ici, mais je ne sais pas trop quoi y dire. Il est compliqué de développer plus sans mettre d'extrait de code Python, et je ne sais pas si cela est nécessaire ou non (rentrer plus dans les détails nécessite de parler un peu plus de l'implémentation de SymPy). Je présente ici la génération de code de manière globale, sans parler des problèmes de minimisation des expressions nécessaire dans le cas Padé, et je ne fais que lister les bibliothèques Python que j'utilise. Sachant que ces outils ont déjà été utilisé pour la partie sans approximation de e^{tL} , seulement pour de l'aide à l'écriture.

La simulation d'un système à 7 variables, 6 variables à une dimension, et 1 variable à 4 dimensions, avec une méthode de type Lawson-Runge-Kutta (LRK) d'ordre élevé, nécessite de nombreuses lignes de code dont l'écriture peut s'avérer fastidieuse. Une part importante de l'analyse ayant été réalisée à l'aide de la bibliothèque de calcul symbolique Python : SymPy, il a été décidé de poursuivre son utilisation pour aider à l'écriture du code de simulation. Dans un premier temps cet usage s'est limité à une aide à l'écriture en générant chacune des 7 expressions pour chaque variable, et ce à chaque étage de la méthode LRK (3 étages pour RK(3,3), jusqu'à 5 étages pour une méthode comme DP4(3)). Des outils de métaprogrammation ont été utilisés pour obtenir une génération complète du code à partir d'un squelette de code et de l'écriture du schéma LRK que l'utilisateur souhaite utiliser.

Les expressions SymPy sont gérées comme des arbres syntaxiques dont les feuilles sont des nombres ou des symboles. Ces derniers vont servir à représenter des variables C++, il est donc nécessaire dans un premier temps de s'assurer que la conversion de ces symboles en chaînes de caractères assure des noms de variables valide en C++. En effet il est fréquent d'utiliser des symboles s'exportant facilement en L^AT_EX, or un tel symbole n'est pas utilisable de la sorte comme nom de variable, par exemple Δt sera s'exportera par défaut en chaîne de caractères en "\Delta\ t". Les noeuds de l'arbre syntaxique sont des fonctions,

il y a alors deux cas à distinguer, soit il s'agit d'une fonction dont la représentation en Python est la même qu'en C++, auquel cas aucune opération particulière n'est nécessaire, c'est le cas par exemple des opérations arithmétiques $+$, $-$, \times et \div qui sont représentées par les opérateurs binaires $+$, $-$, $*$ et $/$ en Python et C++ ; soit il s'agit d'une fonction dont la représentation Python et C++ diffère, auquel cas il est nécessaire de créer une fonction SymPy qui aura le même nom que la fonction C++ associée, et de substituer le noeud de l'arbre syntaxique par cette nouvelle fonction. La conversion en chaîne de caractère de l'arbre ainsi modifié sera une expression C++ valide. Il est possible d'améliorer l'expression C++ en faisant une évaluation numérique des nombres rationnels (et potentiellement aussi irrationnels) présents, pour limiter le nombre d'opérations dans l'expression finale. Ainsi l'expression $1/3$ sera substituée par 0.3333333333333333 , cela permet d'éviter des interprétation de fractions comme des divisions entières par le compilateur.

Pour chaque étage de la méthode LRK, il est ainsi possible d'obtenir une expression C++ valide par variable. L'étape supplémentaire pour assumer que l'on est un gros fainéant est d'utiliser un moteur de *template* pour insérer ces expressions dans un squelette de code qui s'adapte automatiquement au nombre d'étages de la méthode LRK, en initialisant et allouant les variables temporaires nécessaires. Ce travail est effectué par le moteur de *template* Jinja2 qui est une bibliothèque Python permettant d'ajouter des opérations logiques en plus d'une simple substitution de champs dans un squelette de code préexistant. Le squelette en pseudo-code d'un étage d'une méthode LRK est donné en exemple dans l'algorithme 3.7 dans la section 3.3.2.3.

Nota Bene : La bibliothèque SymPy contient des fonctions permettant la génération de code en C ou Fortran, mais le fonctionnement de celles-ci s'adapte mal à une intégration dans une boucle d'un code déjà existant. De plus les fonctions ainsi générées ne fonctionnent pas avec un code contenant des *template* C++, pour changer éventuellement de type pour de possibles optimisations. Elles ne prennent en paramètre que des valeurs par copie ou par pointeur, ce qui limite leur usage avec des structures de données évoluées proposées par les librairies C++. Il serait envisageable d'utiliser certains des mécanismes présents dans ces fonctions pour améliorer la génération de code proposé ci-dessus, en utilisant un parcours d'arbre syntaxique pour construire un *Abstract Syntax Tree* (AST) permettant la génération dans n'importe quel langage d'une expression. Et derniers points, ces fonctions sont très mal documentées (je les ai découverte alors que je générerais déjà les lignes de code pour le Lawson-RK(3,3) et que celui-ci tournait bien), et elles laissent des $1/2$, $1/3$

etc. qui peuvent valoir 0 selon les options de compilation ou les compilateurs.

3.7.2 Parallélisation

3.7.3 Performances

CONCLUSION

Première section de la conclusion

 Lorem ipsum dolor sit amet, « consectetuer » adipiscing elit. Maecenas fermentum, elit non lobortis cursus, orci velit suscipit est, id mollis turpis mi eget orci. Ut aliquam sollicitudin metus. Mauris at sapien sed sapien congue iaculis. Nulla lorem urna, bibendum id, laoreet iaculis, nonummy eget, massa. Phasellus ullamcorper commodo velit. Class aptent taciti sociosqu ad litora torquent per « conubia nostra », per inceptos hymenaeos. Phasellus est. Maecenas felis augue, gravida quis, porta adipiscing, iaculis vitae, felis. Nullam ipsum. Nulla a sem ac leo fringilla mattis. Phasellus egestas augue in sem. Etiam ac enim non mauris ullamcorper scelerisque. In wisi leo, malesuada vulputate, tempor sit amet, facilisis vel, velit. Mauris massa est, sodales placerat, luctus id, hendrerit a, urna. Nullam eleifend pede eget odio. Duis non erat. Nullam pellentesque.

BIBLIOGRAPHIE

- [1] Stéphane BALAC et Arnaud FERNANDEZ, « Mathematical analysis of adaptive step-size techniques when solving the nonlinear Schrödinger equation for simulating light-wave propagation in optical fibers », in : *Optics Communications* 329 (2014), DOI : 10.1016/j.optcom.2014.04.081 (cf. p. 55, 114).
- [2] Stéphane BALAC et Fabrice MAHÉ, « Embedded Runge-Kutta scheme for step-size control in the interaction picture method », in : *Computer Physics Communications* 184.4 (2013), p. 1211-1219, DOI : 10.1016/j.cpc.2012.12.020 (cf. p. 40, 55).
- [3] Michael BALDAUF, « Stability Analysis for linear discretisations of the advection equation with Runge-Kutta time integration », in : *Journal of Computational Physics* (2008) (cf. p. 22).
- [4] Julien BIGOT et al., « Scaling GYSELA code beyond 32K-cores on Blue Gene/Q », in : *ESAIM : PROCEEDINGS*, t. CEMRACS 2012, 43, Luminy, France, 2012, p. 117-135, DOI : 10.1051/proc/201343007 (cf. p. 16).
- [5] Sergio BLANES, Fernando CASAS et Mechthild THALHAMMER, « Splitting and composition methods with embedded error estimators », in : *Applied Numerical Mathematics* 146 (2019), p. 400-415, DOI : 10.1016/j.apnum.2019.07.022 (cf. p. 55, 65, 76, 139).
- [6] Marco CALIARI et al., « Comparison of software for computing the action of the matrix exponential », in : *BIT Numerical Mathematics* 54.1 (2014), p. 113-128, ISSN : 1572-9125, DOI : 10.1007/s10543-013-0446-0 (cf. p. 16).
- [7] Marco CALIARI et al., « The Leja Method Revisited : Backward Error Analysis for the Matrix Exponential », in : *SIAM Journal on Scientific Computing* 38.3 (2016), A1639-A1661, DOI : 10.1137/15M1027620, eprint : <https://doi.org/10.1137/15M1027620> (cf. p. 16).
- [8] Claudio CANUTO et al., *Spectral Methods in Fluid Dynamics*, Springer Berlin Heidelberg, 1988, DOI : 10.1007/978-3-642-84108-8 (cf. p. 15).

-
- [9] Fernando CASAS et Alejandro ESCORIHUELA-TOMÀS, « Composition Methods for Dynamical Systems Separable into Three Parts », in : *Mathematics* 8.4 (2020), ISSN : 2227-7390, DOI : 10.3390/math8040533 (cf. p. 70).
 - [10] Fernando CASAS et al., « High-order Hamiltonian splitting for Vlasov-Poisson equations », in : *Numerische Mathematik* 135.3 (2017), DOI : 10.1007/s00211-016-0816-z (cf. p. 12, 55, 63, 153).
 - [11] Frédérique CHARLES, Bruno DESPRÉS et Michel MEHRENBERGER, « Enhanced Convergence Estimates for Semi-Lagrangian Schemes Application to the Vlasov-Poisson Equation », in : *SIAM Journal on Numerical Analysis* 51.2 (2013), p. 840-863, DOI : 10.1137/110851511 (cf. p. 70).
 - [12] Liu CHEN et Fulvio ZONCA, « Physics of Alfvén waves and energetic particles in burning plasmas », in : *Rev. Mod. Phys.* 88 (2016), p. 015008, DOI : 10.1103/RevModPhys.88.015008 (cf. p. 53).
 - [13] C.Z CHENG et Georg KNORR, « The integration of the vlasov equation in configuration space », in : *Journal of Computational Physics* 22.3 (1976), p. 330-351, ISSN : 0021-9991, DOI : [https://doi.org/10.1016/0021-9991\(76\)90053-X](https://doi.org/10.1016/0021-9991(76)90053-X) (cf. p. 12, 16).
 - [14] David COULETTE et Nicolas BESSE, « Numerical comparisons of gyrokinetic multi-water-bag models », in : *Journal of Computational Physics* 248 (2013), p. 1-32, ISSN : 0021-9991, DOI : <https://doi.org/10.1016/j.jcp.2013.03.065> (cf. p. 38, 41).
 - [15] S.M. COX et P.C. MATTHEWS, « Exponential Time Differencing for Stiff Systems », in : *Journal of Computational Physics* 176.2 (2002), p. 430-455, ISSN : 0021-9991, DOI : <https://doi.org/10.1006/jcph.2002.6995> (cf. p. 18).
 - [16] N. CROUSEILLES et al., « Finite Volume Schemes for Vlasov », in : *Esaim : Proceedings* 38 (2012), p. 275-297 (cf. p. 22).
 - [17] Nicolas CROUSEILLES, Lukas EINKEMMER et Erwan FAOU, « Hamiltonian splitting for the Vlasov-Maxwell equations », in : *Journal of Computational Physics* 283 (2015), p. 224-240, DOI : 10.1016/j.jcp.2014.11.029 (cf. p. 16, 55, 63).

-
- [18] Nicolas CROUSEILLES, Lukas EINKEMMER et Erwan FAOU, « An asymptotic preserving scheme for the relativistic Vlasov–Maxwell equations in the classical limit », in : *Computer Physics Communications* 209 (2016), p. 13-26, ISSN : 0010-4655, DOI : <https://doi.org/10.1016/j.cpc.2016.08.001> (cf. p. 16).
 - [19] Nicolas CROUSEILLES, Lukas EINKEMMER et Josselin MASSOT, « Exponential methods for solving hyperbolic problems with application to kinetic equations », in : *Journal of Computational Physics* 420 (2020), DOI : 10.1016/j.jcp.2020.109688 (cf. p. 11, 55, 65, 74, 108).
 - [20] Nicolas CROUSEILLES, Lukas EINKEMMER et Martina PRUGGER, « An exponential integrator for the drift-kinetic model », in : *Computer Physics Communications* 224 (2018), p. 144-153, DOI : 10.1016/j.cpc.2017.11.003 (cf. p. 13, 24, 37-39).
 - [21] Nicolas CROUSEILLES et Francis FILBET, « Numerical approximation of collisional plasmas by high order methods », in : *Journal of Computational Physics* 201.2 (2004), p. 546-572, ISSN : 0021-9991, DOI : <https://doi.org/10.1016/j.jcp.2004.06.007> (cf. p. 32).
 - [22] Nicolas CROUSEILLES, Guillaume LATU et Eric SONNENDRÜCKER, « A parallel Vlasov solver based on local cubic spline interpolation on patches », in : *Journal of Computational Physics* 228.5 (2009), p. 1429-1446, ISSN : 0021-9991, DOI : <https://doi.org/10.1016/j.jcp.2008.10.041> (cf. p. 16).
 - [23] Nicolas CROUSEILLES, Michel MEHRENBERGER et Francesco VECIL, « Discontinuous Galerkin semi-Lagrangian method for Vlasov-Poisson », in : *ESAIM : Proceedings, CEMRACS 2010* (2011), p. 21, DOI : 10.1051/proc/2011022 (cf. p. 16).
 - [24] Nicolas CROUSEILLES et al., « A new fully two-dimensional conservative semi-Lagrangian method : applications on polar grids, from diocotron instability to ITG turbulence », in : *The European Physical Journal D : Atomic, molecular, optical and plasma physics* 68.9 (2014), article 252, p. 252-261, DOI : 10.1140/epjd/e2014-50180-9 (cf. p. 38, 39).
 - [25] J.R. DORMAND et P.J. PRINCE, « New Runge-Kutta algorithms for numerical simulation in dynamical astronomy », in : *Celestial mechanics* 18 (1978), p. 223-232, DOI : 10.1007/BF01230162 (cf. p. 55, 78).

-
- [26] J.R. DORMAND et P.J. PRINCE, « A family of embedded Runge-Kutta formulae », in : *Journal of Computational and Applied Mathematics* 6.1 (1980), p. 19-26, DOI : 10.1016/0771-050X(80)90013-3 (cf. p. 55, 78).
 - [27] L. EINKEMMER, « A study on conserving invariants of the Vlasov equation in semi-Lagrangian computer simulations », in : *Journal of Plasma Physics* 83.2 (2017), p. 705830203, DOI : 10.1017/S0022377817000216 (cf. p. 16).
 - [28] Lukas EINKEMMER, « A mixed precision semi-Lagrangian algorithm and its performance on accelerators », in : *International Conference on High Performance Computing Simulation (HPCS)*, 2016, p. 74-80, DOI : 10.1109/HPCSim.2016.7568318 (cf. p. 16).
 - [29] Lukas EINKEMMER, « An adaptive step size controller for iterative implicit methods », in : *Applied Numerical Mathematics* 132 (2018), p. 182-204, ISSN : 0168-9274, DOI : <https://doi.org/10.1016/j.apnum.2018.06.002> (cf. p. 40).
 - [30] Lukas EINKEMMER, « A performance comparison of semi-Lagrangian discontinuous Galerkin and spline based Vlasov solvers in four dimensions », in : *Journal of Computational Physics* 376 (2019), p. 937-951, ISSN : 0021-9991, DOI : <https://doi.org/10.1016/j.jcp.2018.10.012> (cf. p. 16).
 - [31] Lukas EINKEMMER, « Semi-Lagrangian Vlasov simulation on GPUs », in : *Computer Physics Communications* 254 (2020), p. 107351, ISSN : 0010-4655, DOI : <https://doi.org/10.1016/j.cpc.2020.107351> (cf. p. 16).
 - [32] Lukas EINKEMMER et Alexander OSTERMANN, « Exponential integrators on graphic processing units », in : *2013 International Conference on High Performance Computing Simulation (HPCS)*, 2013, p. 490-496, DOI : 10.1109/HPCSim.2013.6641458 (cf. p. 16).
 - [33] Lukas EINKEMMER et Alexander OSTERMANN, « A strategy to suppress recurrence in grid-based Vlasov solvers », in : *The European Physical Journal D* 68.7 (2014), p. 197, ISSN : 1434-6079, DOI : 10.1140/epjd/e2014-50058-x (cf. p. 16).
 - [34] Lukas EINKEMMER et Alexander OSTERMANN, « Convergence Analysis of a Discontinuous Galerkin/Strang Splitting Approximation for the Vlasov–Poisson Equations », in : *SIAM Journal on Numerical Analysis* 52.2 (2014), p. 757-778, DOI : 10.1137/120898620, eprint : <https://doi.org/10.1137/120898620> (cf. p. 16).

-
- [35] Lukas EINKEMMER et Alexander OSTERMANN, « Convergence Analysis of Strang Splitting for Vlasov-Type Equations », in : *SIAM Journal on Numerical Analysis* 52.1 (2014), p. 140-155, DOI : [10.1137/130918599](https://doi.org/10.1137/130918599), eprint : <https://doi.org/10.1137/130918599> (cf. p. 16).
 - [36] Lukas EINKEMMER et Alexander OSTERMANN, « A splitting approach for the Kadomtsev–Petviashvili equation », in : *Journal of Computational Physics* 299 (2015), p. 716-730, ISSN : 0021-9991, DOI : <https://doi.org/10.1016/j.jcp.2015.07.024> (cf. p. 16, 43).
 - [37] Lukas EINKEMMER, Mayya TOKMAN et John LOFFELD, « On the performance of exponential integrators for problems in magnetohydrodynamics », in : *Journal of Computational Physics* 330 (2017), p. 550-565, ISSN : 0021-9991, DOI : <https://doi.org/10.1016/j.jcp.2016.11.027> (cf. p. 16).
 - [38] F. FILBET et E. SONNENDRÜCKER, « Comparison of Eulerian Vlasov solvers », in : *Computer Physics Communications* 150.3 (2003), p. 247-266, ISSN : 0010-4655, DOI : [https://doi.org/10.1016/S0010-4655\(02\)00694-X](https://doi.org/10.1016/S0010-4655(02)00694-X) (cf. p. 12, 16).
 - [39] Burton D. FRIED et Samuel D. CONTE, *The Plasma Dispersion Function ; the Hilbert transform of the Gaussian*, Academic Press, 1961 (cf. p. 55, 89).
 - [40] V. GRANDGIRARD et al., « A drift-kinetic Semi-Lagrangian 4D code for ion turbulence simulation », in : *Journal of Computational Physics* 217.2 (2006), p. 395-423, ISSN : 0021-9991, DOI : <https://doi.org/10.1016/j.jcp.2006.01.023> (cf. p. 16, 36).
 - [41] Kjell GUSTAFSSON, Michael LUNDH et Gustaf SÖDERLIND, « API stepsize control for the numerical solution of ordinary differential equations », in : *BIT Numerical Mathematics* 28.2 (1988), p. 270-287, ISSN : 1572-9125, DOI : [10.1007/BF01934091](https://doi.org/10.1007/BF01934091) (cf. p. 40).
 - [42] Ernst HAIRER et Gerhard WANNER, *Solving Ordinary Differential Equations II : Stiff and Differential-Algebraic Problems (Springer Series in Computational Mathematics)*, Springer, 1996 (cf. p. 18).
 - [43] Ernst HAIRER, Gerhard WANNER et Christian LUBICH, *Geometric numerical integration : structure-preserving algorithms for ordinary differential equations*, Berlin New York : Springer, 2006, ISBN : 978-3-540-30663-4 (cf. p. 40, 55, 65).

-
- [44] Nicholas J. HIGHAM, *Functions of Matrices : Theory and Computation*, Society for Industrial et Applied Mathematics, 2008, ISBN : 978-0-898716-46-7 (cf. p. 16).
 - [45] Marlis HOCHBRUCK, Jan LEIBOLD et Alexander OSTERMANN, « On the convergence of Lawson methods for semilinear stiff problems », in : *Numerische Mathematik* 145.3 (2020), p. 553-580, ISSN : 0945-3245, DOI : 10.1007/s00211-020-01120-4 (cf. p. 16).
 - [46] Marlis HOCHBRUCK et Christian LUBICH, « On Krylov Subspace Approximations to the Matrix Exponential Operator », in : *SIAM Journal on Numerical Analysis* 34.5 (1997), p. 1911-1925, DOI : 10.1137/S0036142995280572 (cf. p. 16).
 - [47] Marlis HOCHBRUCK et Alexander OSTERMANN, « Explicit Exponential Runge–Kutta Methods for Semilinear Parabolic Problems », in : *SIAM Journal on Numerical Analysis* 43.3 (2005), p. 1069-1090, DOI : 10.1137/040611434 (cf. p. 16, 18, 55).
 - [48] Marlis HOCHBRUCK et Alexander OSTERMANN, « Exponential integrators », in : *Acta Numerica* 19 (2010), p. 209-286, DOI : 10.1017/S0962492910000048 (cf. p. 13, 15, 16, 18, 50, 55).
 - [49] Florian HOLDERIED, « Investigation of Finite Element Methods for a 4D Hyrbid Plasma Model », mém. de mast., Technische Universität München, 2019 (cf. p. 3, 53-55, 58, 60).
 - [50] Florian HOLDERIED et al., « Structure-preserving vs. standard particle-in-cell methods : The case of an electron hybrid model », in : *Journal of Computational Physics* 402 (2020), DOI : 10.1016/j.jcp.2019.109108 (cf. p. 151, 152, 154).
 - [51] Leah ISHERWOOD, Zachary J. GRANT et Sigal GOTTLIEB, « Strong Stability Preserving Integrating Factor Runge–Kutta Methods », in : *Journal on Numerical Analysis* 56.6 (2018), p. 3276-3307, DOI : 10.1137/17M1143290 (cf. p. 55).
 - [52] Yuto KATOH et Yoshiharu OMURA, « Computer simulation of chorus wave generation in the Earth’s inner magnetosphere », in : *Geophysical Research Letters* 34.3 (2007), DOI : 10.1029/2006GL028594 (cf. p. 53).
 - [53] David KETCHESON, *NodePy (Numerical ODEs in Python) softwate version 0.6*, 2015 (cf. p. 151).

-
- [54] A.J. KLIMAS et W.M. FARRELL, « A Splitting Algorithm for Vlasov Simulation with Filamentation Filtration », in : *Journal of Computational Physics* 110.1 (1994), p. 150-163, ISSN : 0021-9991, DOI : <https://doi.org/10.1006/jcph.1994.1011> (cf. p. 16).
- [55] Katharina KORMANN et Eric SONNENDRÜCKER, « Energy-conserving time propagation for a structure-preserving particle-in-cell Vlasov–Maxwell solver », in : *Journal of Computational Physics* 425 (2021), p. 109890, ISSN : 0021-9991, DOI : <https://doi.org/10.1016/j.jcp.2020.109890> (cf. p. 153).
- [56] Michael KRAUS et al., « GEMPIC : geometric electromagnetic particle-in-cell methods », in : *Journal of Plasma Physics* 83.4 (2017), p. 905830401, DOI : 10.1017/S002237781700040X (cf. p. 55, 63).
- [57] S. KROGSTAD, « Generalized integrating factor methods for stiff PDEs », in : *Journal of Computational Physics* 203.1 (2005), p. 72-88, ISSN : 0021-9991, DOI : <https://doi.org/10.1016/j.jcp.2004.08.006> (cf. p. 18).
- [58] G. LATU et al., « Gyrokinetic Semi-lagrangian Parallel Simulation Using a Hybrid OpenMP/MPI Programming », in : *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, Berlin, Heidelberg : Springer Berlin Heidelberg, 2007, p. 356-364, ISBN : 978-3-540-75416-9 (cf. p. 16).
- [59] Guillaume LATU et al., « Improving conservation properties of a 5D gyrokinetic semi-Lagrangian code », in : *The European Physical Journal D* 68.11 (2014), p. 345, ISSN : 1434-6079, DOI : 10.1140/epjd/e2014-50209-1 (cf. p. 39).
- [60] J. Douglas LAWSON, « An Order Six Runge-Kutta Process with Extended Region of Stability », in : *Journal on Numerical Analysis* (1967), DOI : 10.1137/0704056 (cf. p. 55, 71).
- [61] J. Douglas LAWSON, « Generalized Runge-Kutta Processes for Stable Systems with Large Lipschitz Constants », in : *SIAM Journal on Numerical Analysis* 4.3 (1967), p. 372-380, DOI : 10.1137/0704033, eprint : <https://doi.org/10.1137/0704033> (cf. p. 15, 55, 72).
- [62] Yingzhe LI, Nicolas CROUSEILLES et Yajuan SUN, « Numerical simulations of Vlasov–Maxwell equations for laser plasmas based on Poisson structure », in : *Journal of Computational Physics* 405 (2020), DOI : 10.1016/j.jcp.2019.109172 (cf. p. 55, 63, 65, 134).

-
- [63] Xu-Dong LIU, Stanley OSHER et Tony CHAN, « Weighted Essentially Non-oscillatory Schemes », in : *Journal of Computational Physics* 115.1 (1994), p. 200-212, ISSN : 0021-9991, DOI : <https://doi.org/10.1006/jcph.1994.1187> (cf. p. 74).
- [64] Thibaut LUNET et al., « Combination of WENO and Explicit Runge–Kutta Methods for Wind Transport in the Meso-NH Model », in : *Monthly Weather Review* 145.9 (2017), p. 3817-3838, DOI : 10.1175/MWR-D-16-0343.1 (cf. p. 11, 22, 27).
- [65] A. MARTINNEZ et al., « A massively parallel exponential integrator for advection-diffusion models », in : *Journal of Computational and Applied Mathematics* 231.1 (2009), p. 82-91, ISSN : 0377-0427, DOI : <https://doi.org/10.1016/j.cam.2009.01.024> (cf. p. 16).
- [66] Michel MEHRENBERGER et al., « Vlasov on GPU (VOG project) », in : *ESAIM : Proc.* 43 (2013), p. 37-58, DOI : 10.1051/proc/201343003 (cf. p. 16).
- [67] B. MINCHEV et W. WRIGHT, « A review of exponential integrators for first order semi-linear problems », in : 2005 (cf. p. 15).
- [68] Awad H. AL-MOHY et Nicholas J. HIGHAM, « Computing the Action of the Matrix Exponential, with an Application to Exponential Integrators », in : *SIAM Journal on Scientific Computing* 33.2 (2011), p. 488-511, DOI : 10.1137/100788860, eprint : <https://doi.org/10.1137/100788860> (cf. p. 16).
- [69] Philip J. MORRISON, « A general theory for gauge-free lifting », in : *Physics of Plasmas* 20 (2012), DOI : 10.1063/1.4774063 (cf. p. 65).
- [70] Philip J. MORRISON, « Structure and structure-preserving algorithms for plasma physics », in : *Physics of Plasmas* 24.5 (2017), p. 055502, DOI : 10.1063/1.4982054 (cf. p. 16, 54).
- [71] Mohammed MOTAMED, Colin B. MACDONALD et Steven J. RUUTH, « On the Linear Stability of the Fifth-Order WENO Discretization », in : *Journal of Scientific Computing* 47 (2010), p. 127-149, DOI : 10.1007/s10915-010-9423-9 (cf. p. 11, 22, 27).
- [72] James A. ROSSMANITH et David C. SEAL, « A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov–Poisson equations », in : *Journal of Computational Physics* 230.16 (2011), p. 6203-6232, ISSN : 0021-9991, DOI : <https://doi.org/10.1016/j.jcp.2011.04.018> (cf. p. 16).

-
- [73] Fabien ROZAR, Guillaume LATU et Jean ROMAN, « Achieving Memory Scalability in the Gysela Code to Fit Exascale Constraints », in : *Parallel Processing and Applied Mathematics*, 2014, p. 185-195, ISBN : 978-3-642-55195-6 (cf. p. 16).
- [74] Chi-Wang SHU, « High Order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD », in : *International Journal of Computational Fluid Dynamics* 17.2 (2003), p. 107-118, DOI : 10.1080/1061856031000104851 (cf. p. 74, 75).
- [75] N.J. SIRCOMBE et T.D. ARBER, « VALIS : A split-conservative scheme for the relativistic 2D Vlasov–Maxwell system », in : *Journal of Computational Physics* 228.13 (2009), p. 4773-4788, DOI : 10.1016/j.jcp.2009.03.029 (cf. p. 16).
- [76] Gustaf SÖDERLIND, « Automatic Control and Adaptive Time-Stepping », in : *Numerical Algorithms* 31.1 (2002), p. 281-301, DOI : 10.1023/A:1021160023092 (cf. p. 40).
- [77] Gustaf SÖDERLIND, « Time-step selection algorithms : Adaptivity, control, and signal processing », in : *Applied Numerical Mathematics* 56.3 (2006), p. 488-502, DOI : 10.1016/j.apnum.2005.04.026 (cf. p. 40).
- [78] Eric SONNENDRÜCKER, *Numerical Methods for the Vlasov-Maxwell equations*, Springer, 2015 (cf. p. 55, 79, 81).
- [79] Eric SONNENDRÜCKER et al., « The Semi-Lagrangian Method for the Numerical Resolution of the Vlasov Equation », in : *Journal of Computational Physics* 149.2 (1999), p. 201-220 (cf. p. 12, 16).
- [80] Gilbert STRANG, « On the Construction and Comparison of Difference Schemes », in : *SIAM Journal on Numerical Analysis* 5.3 (1968), p. 506-517, DOI : 10.1137/0705041 (cf. p. 70).
- [81] Masuo SUZUKI, « Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations », in : *Physics Letters A* 146.6 (1990), p. 319-323, ISSN : 0375-9601, DOI : [https://doi.org/10.1016/0375-9601\(90\)90962-N](https://doi.org/10.1016/0375-9601(90)90962-N) (cf. p. 65, 70, 139).
- [82] A. TAMBUE, G.J. LORD et S. GEIGER, « An exponential integrator for advection-dominated reactive transport in heterogeneous porous media », in : *Journal of Computational Physics* 229.10 (2010), p. 3957-3969, DOI : 10.1016/j.jcp.2010.01.037 (cf. p. 16).

-
- [83] X. TAO, « A numerical study of chorus generation and the related variation of wave intensity using the DAWN code », in : *Journal of Geophysical Research : Space Physics* (2014), DOI : 10.1002/2014JA019820 (cf. p. 53).
 - [84] Lloyd N. TREFETHEN, *Spectral Methods in MATLAB*, Society for Industrial et Applied Mathematics, 2000, DOI : 10.1137/1.9780898719598 (cf. p. 15).
 - [85] Cesare TRONCI, « Hamiltonian approach to hybrid plasma models », in : *Journal of Physics A : Mathematical and Theoretical* 43.37 (2010), p. 375501, DOI : 10.1088/1751-8113/43/37/375501 (cf. p. 53, 54).
 - [86] Cesare TRONCI et al., « Hybrid Vlasov-MHD models : Hamiltonian vs. non-Hamiltonian », in : *Plasma Physics and Controlled Fusion* 56.9 (2014), p. 095008, DOI : 10.1088/0741-3335/56/9/095008 (cf. p. 3, 53, 54, 58, 60).
 - [87] J P VERBONCOEUR, « Particle simulation of plasmas : review and advances », in : *Plasma Physics and Controlled Fusion* (2005), DOI : 10.1088/0741-3335/47/5a/017 (cf. p. 12).
 - [88] Rong WANG et Raymond J. SPITERI, « Linear instability of the fifth-order WENO method », in : *Journal on Numerical Analysis* 45.5 (2007), p. 1871-1901, DOI : 10.1137/050637868 (cf. p. 11, 22, 27, 74).

Titre : Modélisation hybride fluide-cinétique de plasmas

Mot clés : de 3 à 6 mots clefs

Résumé : Eius populus ab incunabulis primis ad usque pueritiae tempus extremum, quod annis circumcluditur fere trecentis, circummu-rana pertulit bella, deinde aetatem ingressus adultam post multiplices bellorum aerum-nas Alpes transcendit et fretum, in iuvenem erectus et virum ex omni plaga quam orbis ambit inmensus, reportavit laureas et triumphos, iamque vergens in senium et nomine solo aliquotiens vincens ad tranquilliora vitae discessit. Hoc inmaturo interitu ipse quoque sui pertaesus excessit e vita aetatis nono anno atque vicensimo cum quadriennio imperasset. natus apud Tuscos in Massa Vett-ernensi, patre Constantio Constantini fratre imperatoris, matreque Galla. Thalassius vero

ea tempestate praefectus praetorio praesens ipse quoque adrogantis ingenii, considerans incitationem eius ad multorum augeri dis-crmina, non maturitate vel consiliis mitiga-bat, ut aliquotiens celsae potestates iras prin-cipum molliverunt, sed adversando iurgan-doque cum parum congrueret, eum ad rabiem potius evibrabat, Augustum actus eius exag-gerando creberrime docens, idque, incertum qua mente, ne lateret adfectans. quibus mox Caesar acrius efferatus, velut contumaciae quoddam vexillum altius erigens, sine res-pectu salutis alienae vel suae ad vertenda op-posita instar rapidi fluminis irrevocabili impetu ferebatur. Hae duae provinciae bello quandam piratico catervis mixtae praedonum.

Title: Fluid-kinetic hybrid model of plasmas

Keywords: de 3 à 6 mots clefs

Abstract: Eius populus ab incunabulis primis ad usque pueritiae tempus extremum, quod annis circumcluditur fere trecentis, circummu-rana pertulit bella, deinde aetatem ingressus adultam post multiplices bellorum aerum-nas Alpes transcendit et fretum, in iuvenem erectus et virum ex omni plaga quam orbis ambit inmensus, reportavit laureas et triumphos, iamque vergens in senium et nomine solo aliquotiens vincens ad tranquilliora vitae discessit. Hoc inmaturo interitu ipse quoque sui pertaesus excessit e vita aetatis nono anno atque vicensimo cum quadriennio imperasset. natus apud Tuscos in Massa Vett-ernensi, patre Constantio Constantini fratre imperatoris, matreque Galla. Thalassius vero

ea tempestate praefectus praetorio praesens ipse quoque adrogantis ingenii, considerans incitationem eius ad multorum augeri dis-crmina, non maturitate vel consiliis mitiga-bat, ut aliquotiens celsae potestates iras prin-cipum molliverunt, sed adversando iurgan-doque cum parum congrueret, eum ad rabiem potius evibrabat, Augustum actus eius exag-gerando creberrime docens, idque, incertum qua mente, ne lateret adfectans. quibus mox Caesar acrius efferatus, velut contumaciae quoddam vexillum altius erigens, sine re-

spectu salutis alienae vel suaे ad vertenda petu ferebatur. Hae duae provinciae bello opposita instar rapidi fluminis irrevocabili im- quondam piratico catervis mixtae praedonum.