Schémas numériques adaptatifs pour les équations de Vlasov-Poisson

Éric Madaule

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Schémas numériques adaptatifs pour les équations de Vlasov-Poisson

THÈSE

présentée et soutenue publiquement le 14 octobre 2016

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention mathématiques appliquées)

par

Éric MADAULE

Composition du jury

Président :	Stéphane COLOMBI	Directeur de recherche au CNRS, Institut d'Astrophysique de Paris
Rapporteurs :	Philippe HELLUY	Professeur, Université de Strasbourg
	Valérie PERRIER	Professeur, Grenoble INP
Examinateurs :	Virginie GRANDGIRARD	Ingénieur CEA, CEA-Cadarache
	Simon LABRUNIE	Maître de conférence (HDR), Université de Lorraine
Directeur :	Nicolas BESSE	Professeur, Observatoire de la Côte d'Azur
Co-directeur :	Erwan DERIAZ	Chargé de recherches CNRS, Institut Jean Lamour

I.J.L. – Faculté des Sciences et technologies, BP 70239, F-54506 Vandoeuvre lès Nancy cedex
Université de Lorraine - Pôle M4 : matière, matériaux, métallurgie, mécanique

Mis en page avec la classe thesul.

Remerciements

Je tiens à remercier Erwan Deriaz et Nicolas Besse qui ont accepté de m'encadrer pendant ces trois années de thèse. Je remercie en particulier Erwan pour sa présence, son soutien et les discussions sur l'algorithme et l'implémentation.

Merci à toutes les personnes de l'ANR VLASIX sans qui ma thèse n'aurait pas été possible. Je tiens à remercier en particulier Stéphane Colombi qui m'a encouragé et éclairé dans la publication de mon code. Merci également à Thierry Sousbie pour ses explications sur l'utilisation d'OpenMP.

J'exprime également mes remerciements à l'ensemble de mon jury, Philippe Helluy et Valérie Perrier qui ont été mes rapporteurs, Virginie Grandgirard, Simon Labrunie, et Stéphane Colombi qui a présidé le jury. Je les remercie pour leur temps et leur bienveillance.

Je tiens également à remercier toute l'équipe plasma chaud de l'institut Jean Lamour (Nancy). Merci à tous mes collègues de bureau (ou peu s'en faut), David Coulette, Thomas Drouot, Mégane Collard, Mathieu Sarrat, Julien Médina et Maxime Lesur. Merci également à Étienne Gravier, Stéphane Heuraux et Stéphane Deveaux, Jérôme Moritz et Thierry Réveillé pour leur conversations et leur bonne humeur. Merci également à Anne-Sophie sans qui l'équipe aurait du mal à fonctionner.

Ces remerciements ne serait pas complet si je ne citais pas la communauté du site stackexchange.com, et en particulier les communautés des sites stackoverflow.com, superuser.com, askubuntu.com et tex.stackexchange.com qui m'ont énormément aidé en matière de programmation, de gestion de Linux en général et d'Ubuntu en particulier, et pour les astuces avec LaTeX.

Je remercie CENS, l'association des étudiants de la faculté de science de Nancy, qui m'a ouvert ses portes et a égayé mes midis et aidé à me sentir encore un peu étudiant. Je remercie également le club Légendes, et en particulier Vincent Plée, docteur qui était à Légendes avant moi et à soutenu pendant ma thèse, et Fabien Uhrig, l'actuel président de l'association. Merci également à mes amis de Bordeaux et de Lyon qui m'ont soutenu.

Je remercie enfin ma famille qui, bien que géographiquement éloignée, m'a toujours soutenu et encouragé.

Sommaire

Table des figures

Chapitre 1

Contextes et objectifs

1.1 Physique des plasmas et astrophysique

Cette thèse s'inscrit dans une longue lignée de travaux déjà réalisés, en cours ou à venir ayant trait à la physique des plasmas, et notamment aux plasmas de fusion. Je pourrais vous parler ici de ce qu'est un plasma et des enjeux physiques et technologiques que notre monde actuel fait graviter autour de la maîtrise des plasmas de fusion, mais de nombreux docteurs l'ont déjà fait avant moi et je ne suis pas spécialiste du domaine. Si le sujet vous intéresse, je vous invite à lire les introductions des thèses de M. Besse Nicolas, M. Steiner Christophe, M. Coulette David, M. Huot Fabien, M. Drouot Thomas et bien d'autres encore. Le Commissariat à l'énergie atomique a également publié de nombreux articles de vulgarisation permettant de se renseigner sur le sujet en présentant les enjeux, les risques, les défis et de nombreux aspects intéressants le curieux non spécialiste du domaine. L'ANR Vlasix [4] ayant financé cette thèse inclut de nombreux astrophysiciens. Certaines simulations présentées plus loin relèveront de ce domaine. Là aussi, nombre de thèses dans le domaine, comme celle de M. Colombi Stéphane, Mme Hallé Anaëlle, M. Sousbie Thierry et bien d'autres, vous présenteront une solide introduction. Le numéro 58 des Clefs du CEA, Dans les secrets de l'Univers, permet lui aussi une approche plus complète du sujet. Mon travail a principalement porté sur quelques algorithmes et leur programmation, je vais donc vous parler plus longuement de l'aspect numérique.

Le système d'équation de Vlasov-Poisson est un problème de Cauchy avec une donnée initiale infiniment dérivable [34, 39]. Le théorème de Cauchy stipule que pour une donnée initiale

$$f_0 \in \mathcal{C}_c^m(\mathbb{R}_x^3 \times \mathbb{R}_v^3) \tag{1.1}$$

où $\mathcal{C}_c^m(\mathbb{R}_x^3 \times \mathbb{R}_v^3)$ est l'espace des fonctions m-fois dérivables à support compact, il existe une unique solution en temps f telle que

$$f \in \mathcal{C}^m(0, T\ ;\ \mathcal{C}_c^m(\mathbb{R}_x^3 \times \mathbb{R}_v^3)). \tag{1.2}$$

Une bonne description numérique doit donc avoir la plus grande régularité possible tout en étant capable de capturer les détails locaux. Dans cette perspective, l'utilisation de méthodes d'ordres élevés en espace est un atout.

1.2 La modélisation numérique

Que ce soit pour des raisons techniques ou financières, le recours à la modélisation numérique est très fréquente dans la recherche contemporaine. Quand l'expérimentation est possible, la simulation numérique est un complément utile et efficace. L'expérimentation et le calcul analytique valident les modèles numériques à l'aide de quelques cas tests aux solutions connues. La simulation permet d'accéder ensuite à de nombreux résultats en s'épargnant les coûts liés aux expérimentations, qu'ils soient dûs au prix des matériaux ou à la construction des dispositifs expérimentaux, ou aux temps d'expérimentation, ces derniers étant souvent liés au temps nécessaire à la construction du dispositif. Dans certains contextes, notamment en astrophysique, l'expérimentation n'est même pas accessible. La simulation numérique permet alors de valider (ou d'invalider) les modèles proposés.

Si l'idée initiale de la simulation numérique est simple – résoudre la formulation mathématique d'un problème physique grâce à la puissance de calcul d'un ordinateur –, l'obtention rapide de résultats précis l'est rarement. Les limites liées à l'utilisation de mémoires ou à la capacité de calcul sont régulièrement repoussées par la production à grande échelle d'ordinateurs toujours plus puissants, toujours plus performants. L'ordinateur embarqué à bord des missions Apollo, la pointe de la technologie à la fin des années 1960, bénéficiait de 4 ko de RAM, 72 ko de ROM, et d'un processeur avec une fréquence de 1 MHz (mais dont la fréquence de calcul est à diviser par deux pour obtenir la synchronisation entre les deux ordinateurs embarqués) [1]. Aujourd'hui, un ordinateur de bureau dispose en moyenne d'un processeur à quatre cœurs travaillant chacun à une fréquence de plus de 2 GHz, dispose de 4 Go de RAM et de plus de 500 Go de ROM. Les centres de calculs vont bien au-delà de ces capacités en offrant à leurs utilisateurs des capacités de RAM de l'ordre de la dizaine ou de la centaine de Go et des dizaines de cœurs de calcul. Néanmoins, de telles performances de calcul ont une accessibilité limitée. Le problème de Vlasov en dimension six requiert environ 10 To de mémoire par pas de temps, ce qui n'est actuellement pas traitable en temps raisonnable. Il reste donc important de travailler sur les méthodes numériques mises en œuvre afin de les optimiser, tant en termes de précision qu'en termes de coûts de calcul.

1.2.1 Maillages adaptatifs

Une possibilité afin d'optimiser les calculs est d'adopter une description non-uniforme du problème. Cela présente l'avantage de réduire les coûts de calcul en se concentrant sur la résolution numérique du problème aux points où les informations utiles sont concentrées. Néanmoins, tout gain a un prix en programmation, et ce prix peut aller de la lisibilité du code à l'utilisation de méthodes moins intuitives et plus difficiles à appréhender. L'utilisation de grilles non-uniformes est un premier pas mais il est possible d'aller encore plus loin. L'utilisation d'une grille statique non-uniforme implique de connaître à l'avance l'évolution de la quantité observée. De plus, la grille étant statique, elle comporte des points qui ne sont pas utiles au calcul à chaque instant. Afin d'aller encore plus loin, on demande à l'algorithme de posséder un mécanisme lui permettant au cours de la simulation d'ajouter lui même des points là où des détails significatifs seront transportés et de supprimer des points aux endroits où aucun détail significatif n'est présent. La définition

de ce qu'est un détail significatif est laissée à l'appréciation de l'utilisateur. Ce type de maillage est dit maillage adaptatif, parce qu'il s'adapte de lui même à l'évolution de la quantité observée.

Un maillage adaptatif s'approche donc du maillage optimal en terme de coût de calcul et d'écart entre la solution réelle et la solution numérique. Néanmoins, comme indiqué plus haut, tout gain a un coût en programmation informatique. Certaines méthodes doivent être adaptées pour fonctionner sur des grilles non-uniformes et mouvantes. Dans certains cas, le surcoût apporté à la méthode de résolution par la non-uniformité du maillage est plus grand que le gain apporté par le fait de réduire le nombre de points de calcul. Certaines méthodes peuvent aussi réduire l'adaptativité en nécessitant un raffinement graduel du maillage, c'est à dire qu'un point de niveau de raffinement donné ne peut pas avoir plus d'un niveau de raffinement d'écart avec ses voisins. Enfin, le parcours des points du maillage est plus complexe dans le cas d'un maillage adaptatif que dans le cas d'un maillage fixe. Pour un maillage fixe, les points peuvent être listés dès le début du calcul. Dans le cas d'un maillage adaptatif, il faut établir une nouvelle liste de points à chaque fois que le maillage est modifié.

1.2.2 Méthodes semi-lagrangiennes

En première approximation, on peut répartir les schémas numérique en deux catégories : ceux basés sur des résolutions eulériennes et ceux basés sur des résolutions lagrangiennes du problème.

Dans une méthode eulérienne, la donnée n'est connue qu'en certains points fixés et on regarde son évolution au cours du temps grâce à ses voisins. Prenons l'exemple suivant

$$\partial_t f(X,t) = S(f,X,t), \tag{1.3}$$

où f est la quantité que l'on souhaite suivre, S est un terme d'évolution, X une coordonnée et t le temps. L'exemple le plus simple de résolution eulérienne est le schéma d'Euler explicite. On écrit alors

$$f(X,t+\delta t) = f(X,t) + \delta t\, S(f,X,t). \tag{1.4}$$

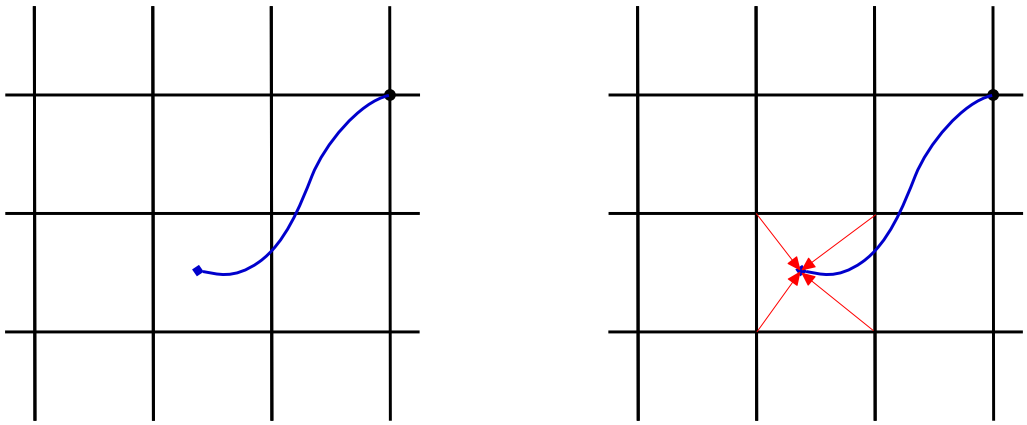
Cette écriture est la plus simple possible, mais de nombreuses variantes existent pour en améliorer l'ordre ou la stabilité (schéma d'Euler implicite, méthodes de Runge-Kutta, etc.). Néanmoins, pour les schémas explicites, cette méthode nécessite de lier la taille du pas de temps à celle du pas d'espace par une condition de Courant–Friedrichs–Lewy (condition de CFL) spécifiques des équations hyperboliques,

$$\partial_t f(x,t) + \partial_x F(f,x,t) = 0, \tag{1.5}$$

ce qui aboutit en général à l'utilisation de pas de temps très courts et à la nécessité de procéder à de (très) nombreuses itérations.

La description lagrangienne consiste à suivre le déplacement d'une particule au cours du temps. Ce type de description est notamment utilisé en dynamique des fluides. La coordonnée d'une particule est alors décrite par

$$\frac{d}{dt}X(t) = U[f](X(t),t), \tag{1.6}$$



(a) Calcul du pied de la caractéristique.

(b) Calcul de la fonction au pied de la caractéristique.

FIGURE 1.1 – Illustration de la méthode semi-lagrangienne arrière avec $\frac{d}{dt}f(X(t),t) = 0$.

où U dépend de f via un opérateur non-local en espace. La fonction f obéit alors à l'équation

$$\frac{d}{dt}f(X(t),t) = S(X(t),t) \tag{1.7}$$

où S est un terme d'évolution. Ce type de méthode sans pas d'espace ne souffre par conséquent pas de condition de CFL limitant la taille du pas de temps. En revanche, le description du domaine peut être très hétérogène à l'état final, c'est à dire que certaines régions seront décrites par de nombreux points tandis que d'autres régions seront décrites par très peu de points, et ne permet pas d'obtenir des informations en tout temps sur tout le domaine.

Les méthodes semi-lagrangiennes combinent le maillage des méthodes eulériennes et l'utilisation des caractéristiques faite dans les méthodes lagrangiennes. J'utilise dans ma thèse des méthodes semi-lagrangiennes arrière (appelées en anglais backward semi-Lagrangian method et abrégé BSL) dont le principe est illustré par la figure 1.1 avec un terme d'évolution $S = 0$. Dans un premier temps, on cherche les pieds des caractéristiques qui au temps $n+1$ arrivent aux nœuds du maillage. Une fonction étant constante le long de ses caractéristiques, la valeur calculée au pied de la caractéristique est donc la nouvelle valeur au nœud correspondant. Cette étape est illustrée par la figure 1.1a. Le point noir (en haut à droite) est le point d'arrivée de la caractéristique tracée en bleu. Le carré correspond au pied de la caractéristique. La valeur de la fonction est ensuite calculée au pied de la caractéristique à partir des valeurs de la fonction aux nœuds les plus proches, comme montré sur la figure 1.1b. Un important travail de développement d'une bibliothèque de résolution par des méthodes semi-lagrangiennes est notamment effectué dans le cadre du projet SeLaLib [3].

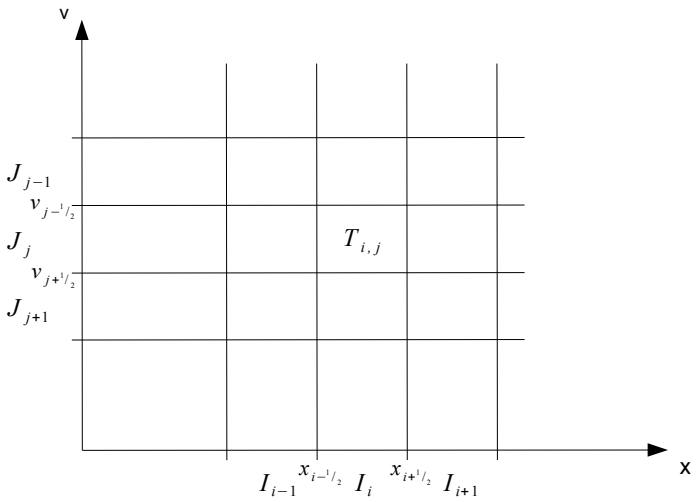


FIGURE 1.2 – Représentation du maillage et notations.

1.2.3 Travaux existants

Le problème de Vlasov-Poisson est un système très étudié (mais pas complètement compris) de la physique des plasmas et de l'astrophysique. C'est notamment un problème majeur pour les futures simulations de plasmas à grandes échelles. La résolution eulérienne du système de Vlasov-Poisson avec une description spatiale utilisant des méthodes Galerkin discontinues a fait l'objet de nombreuses publications, comme celles d'Ayuso *et al.* [8, 9], Cheng *et al.* [16, 17], Heath *et al.* [29], Rossamnith *et al.* [38], Restelli *et al.* [32], etc. Des méthodes semi-lagrangiennes ont également été largement étudiées, avec notamment les publications de Guo *et al.* [28], Qiu et Shu [36], Crouseilles *et al.* [21], Bokanowski and Simarta [13], etc. L'utilisation des multi-ondelettes a également fait l'objet de nombreuses études. Leurs propriétés mathématiques ont été largement étudiées par Alpert *et al.* [5, 6]. Leur utilisation dans le domaine des lois de conservation hyperboliques non linéaires a fait l'objet de plusieurs publications par Müller *et al.* [27, 30]. Gerhard *et al.* ont également publié une étude dans le cas des écoulements compressibles [26]. L'utilisation de méthodes Galerkin discontinues et des multi-ondelettes pour différents problèmes de convection est également présentée par Archibald *et al.* [7]. Besse *et al.* [12] ont également présenté l'avantage d'un maillage adaptatif dans le cadre de Vlasov-Poisson relativiste.

1.3 Notations

Sauf indications contraires, on utilisera les notations définies dans cette section.

Pour décrire le schéma en espace, le maillage en deux dimensions est découpé en cellules $T_{i,j} = I_i \times J_j$ qui est le produit tensoriel de la cellule $I_i = [x_{i-1/2} ; x_{i+1/2}]$ dans la direction x et de la cellule $J_j = [v_{j-1/2} ; v_{j+1/2}]$ dans la direction v (ou p quand on considère la quantité de mouvement dans la direction orthogonale à x plutôt que la vitesse). L'intervalle I_i a une longueur $x_{i+1/2} - x_{i-1/2} = \Delta x_i$ et l'intervalle J_j a une longueur $v_{j+1/2} - v_{j-1/2} = \Delta v_j$.

Ce découpage est illustré par la figure 1.2. Dans le cas d'un maillage adaptatif non-conforme (c'est à dire que les sommets d'une cellule peuvent se trouver le long de l'arrête d'une cellule voisine) cette représentation n'est plus applicable. Néanmoins, comme le maillage reste cartésien, les notations seront conservées. On considérera simplement qu'il existe plusieurs niveaux de cellules I_i et J_j. Cette notation est étendue dans le cas de la dimension quatre à des cellules notées $T_{i,j,k,l}$ et les directions sont respectivement x, y, v_x et v_y.

Les points de Gauss-Legendre sont notés x_{i_g} s'ils sont pris dans dans la direction x et v_{j_g} s'ils sont pris dans la direction v. Les poids associés sont respectivement w_{i_g} et w_{j_g}. Dans le cas du produit tensoriel, on obtient $w_{i_g,j_g} = w_{i_g} w_{j_g}$. En dimension quatre, les indices sont notés i_g, j_g, k_g et l_g dans les directions respectivement données dans le paragraphe précédent.

L'intervalle de référence est $[-1\ ;\ 1]$ pour les cas à une dimension, $[-1\ ;\ 1] \times [-1\ ;\ 1]$ en dimension deux, et $[-1\ ;\ 1]^4$ en dimension quatre. On pose $\mathbb{P}^K$ l'espace des polynômes de degré inférieur ou égal à K sur l'intervalle de référence $[-1\ ;\ 1]$. On définit $p_k(x)$ le k-ième polynôme de Legendre normalisé en norme L^2, c'est à dire

$$\int_{-1}^{1} \left(p_k(x)\right)^2 \, dx = 1. \tag{1.8}$$

L'ensemble des $\{p_k\}_{k=0}^{K}$ forme une base orthonormée et échelonnée de $\mathbb{P}^K$. La construction de ces polynômes est expliquée dans l'annexe A. Pour un problème en dimension d, on note $\boldsymbol{k} = (k_1, \ldots, k_d)$ un vecteur d'indices sur $\mathbb{N}^d$. En dimension multiple, la notation $\mathbb{P}^K$ est étendue sans que cela ne pose d'ambiguïtés de sens par

$$\mathbb{P}^K = \mathrm{Vect}\left(p_{k_1}(x_1) p_{k_2}(x_2) \ldots p_{k_d}(x_d), \quad \text{tels que } k_i \leq K, \ \forall i\right) \tag{1.9}$$

pour obtenir tous les polynômes dont le degré par direction est inférieur ou égal à K. On pose également

$$\mathfrak{P}^K = \mathrm{Vect}\left(p_{k_1}(x_1) p_{k_2}(x_2) \ldots p_{k_d}(x_d), \quad \text{tels que } \sum_{i=0}^{d} k_i \leq K\right) \tag{1.10}$$

l'ensemble des polynômes dont le degré total est inférieur ou égal à K. L'opérateur $\mathrm{Vect}(A)$ indique l'espace vectoriel généré par A.

Dans la continuité de ce qui vient d'être introduit, en dimension deux, k_1 est l'indice dans la direction x et k_2 est l'indice dans la direction v. En dimension quatre, k_1, k_2, k_3 et k_4 sont respectivement les indices dans les directions x, y, v_x et v_y.

On utilise fréquemment durant cette thèse une formule de quadrature utilisant les points de Gauss. La formule s'écrit

$$\int_{-1}^{1} p(x)\, dx \approx \sum_{i_g=0}^{n} w_{i_g} p(x_{i_g}) \tag{1.11}$$

et est exacte si p est un polynôme de degré inférieur ou égal à $2n-1$ pou la formule à n points de Gauss.

Dans la suite de la thèse, sauf précision contraire, les coefficients, fonctions ou polynômes dont le nom est surmonté d'un tilde sont exprimés sur l'intervalle de référence. En l'absence de tilde la valeur est exprimée sur l'intervalle du maillage. Il suffit d'une homothétie pour passer de l'un à l'autre. Les poids des formules de quadrature échappent à cette règle, ils sont toujours exprimés sur l'intervalle de référence. Un facteur de dilatation ou de contraction sera ajouté quand une formule de quadrature sera utilisée sur un intervalle qui n'est pas l'intervalle de référence. Sauf mention contraire explicite, l'intervalle de référence est $[-1\ ;\ 1]$. (Ce dernier point sera rappelé dans le chapitre sur les multi-ondelettes, chapitre 3.)

La lettre h en indice d'une fonction indique la projection de la fonction sur un espace de dimension finie qui est l'espace de travail et de calcul. On a donc $f(x,v)$ la distribution physique et

$$f_h(x,v) = \sum_{T_{i,j}} \sum_{k_1,k_2} f_{i,j,k_1,k_2} p_{k_1}(x) p_{k_2}(v) \tag{1.12}$$

est l'approximation de f sur la base polynomiale.

On s'intéresse dans cette thèse à des méthodes semi-lagrangiennes, ce qui implique la présence de courbes caractéristiques. L'exposant $*$ est utilisé pour indiquer le pied de la caractéristique issue du point portant l'exposant.

Sauf précision contraire la norme considérée est toujours la norme L^2. Pour des fonctions il s'agit donc de

$$\left\| f \right\|_{L^2} = \left(\int_{-1}^{1} (f(x))^2 \, dx \right)^{1/2}, \tag{1.13}$$

pour des vecteurs ou des fonctions décomposées sur des bases orthonormées, il s'agit de la norme euclidienne

$$\left\| \vec{B} \right\| = \left(\sum_j B_j^2 \right)^{1/2}, \tag{1.14}$$

$$\left\| f_h \right\|_{L^2} = \left(\sum_j f_j^2 \right)^{1/2}. \tag{1.15}$$

Pour les fonctions L^2, le produit scalaire $\langle ., . \rangle$ est celui associé au calcul de la norme, c'est à dire

$$\langle f, g \rangle = \int_{-1}^{1} f(x) g(x) \, dx. \tag{1.16}$$

Bien que l'immense majorité du travail se fasse avec des variables scalaires, l'introduction du problème à quatre dimensions fait intervenir des champs à plusieurs dimensions. Les variables multi-dimensionnelles sont notées en gras. Le champ $E_i(x,t)$ est donc un champ scalaire alors que $\boldsymbol{E}(x,t)$ est un champ vectoriel sous la forme $\boldsymbol{E}(x,t) = \sum_i E_i(x,t) \boldsymbol{e}_i$, où les $\boldsymbol{e}_i$ sont les vecteurs de base de l'espace considéré.

Chapitre 2

Système d'équations de Vlasov-Poisson et schémas numériques

2.1 Le modèle Vlasov-Poisson

Dans cette thèse on s'intéresse plus particulièrement à l'équation de Vlasov. Cette équation est utilisée en physique des plasmas pour décrire l'évolution d'un plasma sans collision et en astrophysique pour décrire une distribution de matière non collisionnelle. L'équation de Vlasov décrit l'évolution temporelle d'une distribution dans un espace des phases `espace` × `vitesse`. On s'intéresse principalement dans cette thèse au cas où le couplage entre la distribution et le champ qu'elle génère se fait par l'équation de Poisson. Dans le cadre de la physique des plasmas, on suit l'évolution d'une distribution d'électrons avec en fond des ions uniformément répartis. Les électrons n'étant pas répartis de façon homogène, un champ électrique apparaît. Dans le cas d'un problème d'astrophysique, la présence de matière génère un champ gravitationnel. Ces champs sont obtenus par la résolution de l'équation de Poisson.

Jusqu'au chapitre 7 le travail est réalisé dans un espace des phases à une dimension d'espace et une dimension de vitesse. (La dimension totale est donc deux.) Tout le travail et les exemples présentés avant seront donc présentés dans le cadre de problèmes à une ou deux dimensions. Le passage en dimension quatre se fera ensuite presque naturellement en étendant ce qui aura été posé pour des dimensions inférieures, principalement par l'utilisation du produit tensoriel.

Les équations de Vlasov-Poisson adimensionnées dans un espaces des phases à d dimensions en espace et d dimensions en vitesse s'écrivent

$$\partial_t f(\boldsymbol{x},\boldsymbol{v},t) + \boldsymbol{v}\cdot\nabla_{\boldsymbol{x}} f(\boldsymbol{x},\boldsymbol{v},t) + \boldsymbol{E}(\boldsymbol{x},t)\cdot\nabla_{\boldsymbol{v}} f(\boldsymbol{x},\boldsymbol{v},t) = 0, \quad (\boldsymbol{x},\boldsymbol{v}) \in \Omega_{\boldsymbol{x}} \times \mathbb{R}^d_v, \tag{2.1}$$

où $f(\boldsymbol{x},\boldsymbol{v},t)$ est la distribution dépendant des coordonnées d'espace $\boldsymbol{x}$, des coordonnées de vitesse $\boldsymbol{v}$ et du temps t. $\Omega_{\boldsymbol{x}}$ est le domaine de définition en $\boldsymbol{x}$ de f. En physique des plasmas $\boldsymbol{E}(\boldsymbol{x},t)$ est le champ électrique obtenu en résolvant l'équation de Poisson adimensionnée

$$-\Delta_{\boldsymbol{x}}\Phi(\boldsymbol{x},t) = \rho(\boldsymbol{x},t) - 1, \quad \boldsymbol{x} \in \Omega_{\boldsymbol{x}} \tag{2.2a}$$

$$\boldsymbol{E}(\boldsymbol{x},t) = -\nabla_{\boldsymbol{x}}\Phi(\boldsymbol{x},t) \tag{2.2b}$$

avec $\rho(\boldsymbol{x},t)$ la densité de charge

$$\rho(\boldsymbol{x},t) = \int_{\mathbb{R}^d} f(\boldsymbol{x},\boldsymbol{v},t)\,d\boldsymbol{v} \tag{2.3}$$

et $\Phi(\boldsymbol{x},t)$ est le potentiel électrique. L'équation (2.2a) est présentée dans le cas d'une distribution d'électrons non constante sur un fond d'ions équi-répartis et statiques qui se traduit par la présence d'une constante unitaire.

Dans le cadre du problème en dimension une d'espace et en dimension une de vitesse, l'équation de Vlasov adimensionnée (2.1) devient

$$\partial_t f(x,v,t) + v\partial_x f(x,v,t) + E(x,t)\partial_v f(x,v,t) = 0, \quad (x,v) \in \Omega_x \times \mathbb{R}, \tag{2.4}$$

et le système d'équations de Poisson (2.2) s'écrit

$$-\partial_x^2\Phi(x,t) = \rho(x,t) - 1, \quad x \in \Omega_x, \tag{2.5a}$$
$$E(x,t) = -\partial_x\Phi(x,t) \tag{2.5b}$$

avec $\rho(x,t)$ la densité de charges

$$\rho(x,t) = \int_{\mathbb{R}} f(x,v,t)\,dv, \quad x \in \Omega_x. \tag{2.6}$$

Ω_x est un domaine compact de $\mathbb{R}$ avec des conditions périodiques aux bords. Pour des raisons numériques, le domaine de vitesse doit être fini et est souvent choisi sous la forme $\Omega_v = [-L\ ;\ L]$.

Si on se place dans le contexte de l'astrophysique, alors f devient la fonction de distribution de la matière et E un champ de gravité (souvent noté G en astrophysique), et l'équation de Poisson devient

$$\partial_x^2\Phi(x,t) = \rho(x,t), \quad x \in \Omega_x, \tag{2.7a}$$
$$E(x,t) = -\partial_x\Phi(x,t). \tag{2.7b}$$

On notera la disparition du champ de fond et le signe de l'équation (2.7a) qui est l'opposé de l'équation (2.5a). Néanmoins, pour respecter la périodicité imposée de E au bord de Ω_x, l'équation (2.7a) sera utilisée sous la forme

$$-\partial_x^2\Phi(x,t) = \rho(x,t) - \frac{\int_{\Omega_x}\rho(\xi,t)\,d\xi}{\Gamma(\Omega_x)}, \quad x \in \Omega_x \tag{2.8}$$

où $\Gamma(\Omega_x)$ est la longueur de l'intervalle Ω_x.

On impose des conditions aux limites de Dirichlet en vitesse qui correspondent à interdire des vitesses infinies au plasma. Les conditions aux limites en espace sont choisies périodiques et on fixe E à moyenne nulle (ce qui est équivalent à fixer $\Phi(a,t) = \Phi(b,t)$ pour $\Omega_x = [a\ ;\ b]$).

On notera pour la suite l'on peut dans le cas du système de Vlasov-Poisson passer de façon immédiate de la formulation advective (2.4) à la formulation conservative

$$\partial_t f(x,v,t) + \partial_x(v f(x,v,t)) + \partial_v(E(x,t) f(x,v,t)) = 0, \quad (x,v) \in \Omega_x \times \mathbb{R}. \tag{2.9}$$

Ce passage est immédiat car v est indépendant de x et $E(x,t)$ est indépendant de v. Ce point est important car les schémas numériques mis en œuvre utilisent les formulations conservatives des équations.

Le système d'équations de Vlasov-Poisson conserve une infinité de quantités. Il conserve au cours du temps le minimum et le maximum de la distribution, c'est à dire

si $\min\limits_{(x,v)\in\Omega_x\times[-L\ ;\ L]} f(x,v,0) = A,$

alors $\min\limits_{(x,v)\in\Omega_x\times[-L\ ;\ L]} f(x,v,t) = A, \quad \forall t \in [0\ ;\ +\infty[,$

et

si $\max\limits_{(x,v)\in\Omega_x\times[-L\ ;\ L]} f(x,v,0) = B,$

alors $\max\limits_{(x,v)\in\Omega_x\times[-L\ ;\ L]} f(x,v,t) = B, \quad \forall t \in [0\ ;\ +\infty[.$

Il conserve également la quantité de mouvement du système

$$\frac{d}{dt}\left(\int_{\Omega_x\times[-L\ ;\ L]} v\, f(x,v,t)\, dxdv\right) = 0, \tag{2.10}$$

l'énergie totale du système

$$\frac{d}{dt}\left(\int_{\Omega_x\times[-L\ ;\ L]} \frac{1}{2} v^2 f(x,v,t)\, dxdv + \int_{\Omega_x} \frac{1}{2} E(x,t)^2\, dx\right) = 0, \tag{2.11}$$

et toutes les normes L^p de f, $1 \le p \le \infty$

$$\frac{d}{dt}\left(\int_{\Omega_x\times[-L\ ;\ L]} |f(x,v,t)|^p\, dxdv\right)^{\frac{1}{p}} = 0, \quad 1 \le p < \infty, \tag{2.12}$$

et

$$\frac{d}{dt}\|f\|_\infty = \frac{d}{dt}\left(\sup_{(x,v)\in\Omega_x\times\mathbb{R}} |f(x,v,t)|\right) = 0. \tag{2.13}$$

Ceci reste bien sûr vrai en dimension supérieure et pour des supports plus généraux. Ces conservations sont intéressantes puisqu'elles nous permettent d'évaluer et de comparer les schémas numériques. On notera en plus qu'aussi bien en physique des plasmas qu'en astrophysique, la distribution initiale est toujours positive. (Il s'agit d'une distribution de matière, elle ne peut donc physiquement pas être négative.) En pratique, la conservation des extremums arrive rarement naturellement et est souvent imposée au détriment d'autres conservations. La positivité de la fonction est souvent perdue, ce qui induit un écart entre la quantité appelée masse numérique

$$\int_{\Omega_x\times[-L\ ;\ L]} f_h(x,v,t)\, dxdv$$

et la norme L^1

$$\int_{\Omega_x\times[-L\ ;\ L]} |f_h(x,v,t)|\, dxdv.$$

Il est donc intéressant de suivre les deux, cela nous permet notamment d'observer à quel point la distribution devient négative. Dans la suite de cette thèse, et notamment au chapitre 6, quand on parlera de masse, il s'agira toujours de la masse numérique prenant en compte le signe de la distribution.

2.2 La méthode Galerkin discontinue

2.2.1 Présentation de la méthode

L'essence de la méthode Galerkin discontinue est de proposer une formulation faible du problème et d'en faire une résolution locale. Prenons l'équation de conservation

$$\partial_t f(x,t) + \partial_x \left(a(x,t) f(x,t)\right) = 0. \tag{2.14}$$

On la multiplie alors par une fonction test quelconque φ appartenant à un espace choisi ultérieurement $\mathbb{V}$. De façon générale, l'espace $\mathbb{V}$ est choisi en fonction des propriétés que l'on souhaite utiliser ou mettre en avant. Dans le cadre de la méthode Galerkin discontinue l'espace $\mathbb{V}$ est un espace de polynômes par morceaux de degré fini fixé qui correspondra à l'ordre spatial de la méthode. L'équation est alors intégrée sur chaque élément $I_i = [x_{i-1/2}\ ;\ x_{i+1/2}]$ du maillage.

$$\int_{I_i} \Big(\varphi(x)\partial_t f(x,t) + \varphi(x)\partial_x \left(a(x,t) f(x,t)\right)\Big)\, dx = 0. \tag{2.15}$$

On procède ensuite à une intégration par parties de la dérivée en espace

$$\int_{I_i} \varphi(x)\partial_t f(x,t)\, dx = \int_{I_i} a(x,t) f(x,t) \partial_x \varphi(x)\, dx - \left[a(x,t) f(x,t) \varphi(x)\right]_{x^+_{i-1/2}}^{x^-_{i+1/2}}, \tag{2.16}$$

où $x^+_{i-1/2}$ (respectivement $x^-_{i+1/2}$) désigne la limite en $x_{i-1/2}$ (respectivement $x_{i+1/2}$) par valeur supérieure (respectivement inférieure). La trace aux bords n'ayant pas de sens dans le cadre d'une représentation discontinue, la partie entre crochets est remplacée par un flux numérique adapté à la méthode de résolution temporelle. La méthode Galerkin discontinue fait le lien entre les méthodes de volumes finis et d'éléments finis [25].

Dans le cas des équations de Vlasov-Poisson, on décompose notre domaine à deux dimensions en un ensemble $T_{i,j} = I_i \times J_j$ où I_i est utilisé pour la direction x et J_j pour la direction v. La fonction test est aussi une fonction de x et v. En prévoyant une résolution en temps par splitting on peut écrire

$$\int_{T_{i,j}} \varphi(x,v)\partial_t f(x,v,t)\, dxdv = \int_{T_{i,j}} v f(x,v,t) \partial_x \varphi(x,v)\, dxdv - \int_{J_j} \left[v f \varphi\right]_{x^+_{i-1/2}}^{x^-_{i+1/2}} dv, \tag{2.17a}$$

$$\int_{T_{i,j}} \varphi(x,v)\partial_t f(x,v,t)\, dxdv = \int_{T_{i,j}} E(x,t) f(x,v,t) \partial_v \varphi(x,v)\, dxdv - \int_{I_i} \left[E f \varphi\right]_{v^+_{j-1/2}}^{v^-_{j+1/2}} dx. \tag{2.17b}$$

2.2.2 Projection de la fonction de distribution

Dans la suite on considérera l'espace des fonctions tests $\mathbb{P}^n$ définies dans la section 1.3 comme l'espace engendré par le produit tensoriel des polynômes de Legendre

$$\mathbb{P}^n = \left\{ f(x,v) = \sum_{(k_1,k_2)} a_{k_1,k_2} p_{k_1}(x) p_{k_2}(v), \quad \forall k_1, k_2 \leq n \right\}.$$

Les fonctions tests seront prises parmi la base

$$\{p_{k_1}(x)p_{k_2}(v), \quad \forall\ k_1, k_2 \leq n\}$$

et on pose $\boldsymbol{k} = (k_1, k_2)$. La fonction de distribution au temps initial est décomposée sur cette base comme décrit par les formules (2.2.21), (2.2.22) et (2.2.23) de [14], reprises ici dans les formules (2.18) et (2.19)

$$f(x_{i_g}, v_{j_g}) = \sum_{\boldsymbol{k}} f_{\boldsymbol{k}} P_{\boldsymbol{k}}(x_{i_g}, v_{j_g}), \quad \forall i_g, j_g \tag{2.18}$$

où $f_{\boldsymbol{k}}$ sont les coefficients de la décomposition de f sur la base des $\{P_{\boldsymbol{k}}\}_{\boldsymbol{k}} = \{(p_{k_1}, p_{k_2})\}_{(k_1,k_2)}$ obtenus par la relation

$$f_{\boldsymbol{k}} = \frac{\Delta x_i \Delta v_j}{4} \sum_{i_g, j_g} f(x_{i_g}, v_{j_g}) P_{\boldsymbol{k}}(x_{i_g}, v_{j_g}) w_{i_g,j_g} \approx \int f(x,v) P_{\boldsymbol{k}}(x,v)\, dxdv. \tag{2.19}$$

La formule (2.2.23) de [14] est une formule de normalisation. L'utilisation d'une base orthonormée fait qu'il n'est pas nécessaire ici de passer par cette étape. La norme des polynômes de la base n'apparaît donc pas dans le calcul. Adapté à notre formulation, en n'utilisant que des projections vers l'élément de référence, la formule (2.18) devient

$$f(x_{i_g}, v_{j_g}) = \sum_{\boldsymbol{k}} \tilde{f}_{\boldsymbol{k}} \tilde{P}_{\boldsymbol{k}}(\tilde{x}_{i_g}, \tilde{v}_{j_g}), \tag{2.20}$$

et la formule (2.19) s'écrit

$$\tilde{f}_{\boldsymbol{k}} = \sum_{i_g, j_g} f(x_{i_g}, v_{j_g}) \tilde{P}_{\boldsymbol{k}}(\tilde{x}_{i_g}, \tilde{v}_{j_g}) w_{i_g,j_g}. \tag{2.21}$$

Dans le cas de la méthode Galerkin discontinue, les fonctions tests sont prises dans la base sur laquelle est décomposée la distribution. En d'autres termes, ici, la base primale et la base duale sont la même.

Les sections 2.3.1 et 2.4.1 expliquent les schémas numériques utilisés avec la même équation d'advection. Les sections 2.3.2 et 2.4.2 appliquent les schémas numériques à l'équation de Vlasov.

2.2.3 Résolution du problème de Poisson

Le problème de Poisson à une dimension que nous voulons résoudre est le système présenté précédemment, équation (2.5) Dans notre cadre de travail,la densité de charge

$\rho(x,t)$, le potentiel électrique $\Phi(x,t)$ et le champ électrique $E(x,t)$ sont des polynômes par morceaux, et ρ est de degré n fixé par le degré de la distribution. En une dimension il est possible de résoudre directement l'équation équivalente

$$\partial_x E(x,t) = \rho(x,t) - 1 \tag{2.22}$$

par l'intégration directe

$$E(x,t) = \int \rho(x,t) - 1\, dx, \tag{2.23a}$$

$$\int_{\Omega_x} E(x,t)\, dx = 0. \tag{2.23b}$$

Le champs E est calculé de façon exacte avec un degré $n+1$, ce qui offre un degré de liberté supplémentaire pour imposer E continu à moyenne nulle, ce qui correspond aux contraintes physique du problème.

Le champ E est ensuite projeté sur l'espace des polynômes de degré inférieur ou égal à n sur chaque cellule par la formule

$$\tilde{E}_k(t) = \sum_{i_g} E(x_{i_g},t)\tilde{p}_k(\tilde{x}_{i_g})w_{i_g}, \quad k = 0..n. \tag{2.24}$$

La résolution est détaillée dans la section 4.2, après que toutes les notations et les notions nécessaires ont été introduites.

2.3 La méthode Galerkin discontinue semi-lagrangienne

2.3.1 Présentation de la méthode

La méthode Galerkin discontinue semi-lagrangienne ou GDSL (abrégé SLDG en anglais) est basée sur le travail de Qiu et Shu [36] et la formulation conservative de l'équation de Vlasov

$$\partial_t f + \partial_x(vf) + \partial_v(Ef) = 0. \tag{2.25}$$

Dans ce contexte, l'indice h indique une distribution obtenue par résolution numérique. L'indice k indique un coefficient obtenu par projection sur une base de polynômes. L'exposant n réfère au temps $t^n = n\Delta t$. Cette section présente la méthode dans le cadre général et la section suivante l'applique au cas particulier de l'équation de Vlasov. On part de l'équation

$$\partial_t f + \partial_x(af) = 0, \tag{2.26}$$

où a est une fonction pouvant dépendre de x et de t. On applique ensuite la méthode Galerkin discontinue à cette équation pour aboutir à l'équation suivante,

$$\int_{I_i} \varphi \partial_t f\, dx = \int_{I_i} af\partial_x\varphi\, dx - \left[af\varphi\right]_{x^+_{i-1/2}}^{x^-_{i+1/2}}. \tag{2.27}$$

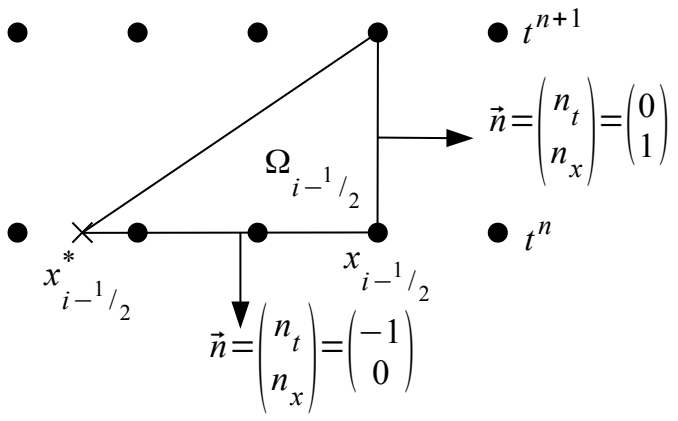


FIGURE 2.1 – Schéma des caractéristiques pour la méthode SLDG

Cette nouvelle équation est alors intégrée en temps entre t^n et t^{n+1}

$$\int_{I_i} f_h^{n+1}(x)\varphi(x)\,dx = \underbrace{\int_{I_i} f_h^n(x)\varphi(x)\,dx}_{T_0} + \underbrace{\int_{t^n}^{t^{n+1}} \int_{I_i} a f_h(x,t)\varphi'(x)\,dx\,dt}_{T_1}$$
$$\underbrace{- \int_{t^n}^{t^{n+1}} \left((a(x,t) f_h(x,t)\varphi(x))|_{x^-_{i+1/2}} - (a(x,t) f_h(x,t)\varphi(x))|_{x^+_{i-1/2}} \right) dt}_{T_2}, \quad \varphi \in \mathbb{V}, \quad (2.28)$$

avec $f_h^n(x) = f_h(x, t^n)$. On appelle respectivement les trois intégrales à droite de l'égalité de l'équation (2.28) T_0, T_1 et T_2. La notation introduite est ainsi

$$\int_{I_i} f_h^{n+1}(x)\varphi(x)\,dx = T_0 + T_1 - T_2. \quad (2.29)$$

On notera ici que T_2 ne contient que l'intégrale et qu'il est effectivement précédé d'un signe moins.

Le terme T_0 ne pose aucun soucis puisqu'il s'agit uniquement d'intégrer au temps n sur un domaine connu. On notera au passage que si φ est le k-ème vecteur de la base sur laquelle est décomposée la distribution, alors

$$T_0 = \int_{I_i} f_h^n(x)\varphi(x)\,dx = f_k^n, \quad (2.30)$$

et

$$\int_{I_i} f_h^{n+1}(x)\varphi(x)\,dx = f_k^{n+1} \quad (2.31)$$

Afin de calculer le terme T_2, on pose

$$\boldsymbol{F}(x,t) = \begin{pmatrix} f(x,t) \\ a(x,t) f(x,t) \end{pmatrix}. \quad (2.32)$$

On a alors

$$\nabla_{t,x} \cdot \boldsymbol{F} = \partial_t f + \partial_x (af) = 0. \quad (2.33)$$

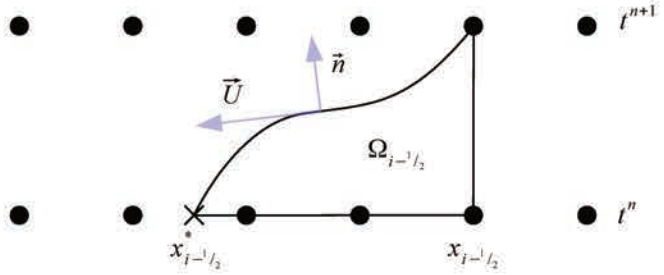


FIGURE 2.2 – Schéma du vecteur directeur des caractéristiques pour la méthode SLDG

On utilise la figure 2.1. Le point $x^*_{i-1/2}$ est le pied de la caractéristique passant par $x_{i-1/2}$ au temps t^{n+1}. Sur le volume $\Omega_{i-1/2}$, on a par définition

$$\iint_{\Omega_{i-1/2}} \nabla_{t,x} \cdot \boldsymbol{F}\, dx dv = 0. \tag{2.34}$$

On applique alors le théorème de la divergence à l'équation (2.34)

$$\begin{aligned} 0 &= \oint_{\partial\Omega_{i-1/2}} \boldsymbol{F}\cdot\boldsymbol{n}\, ds \\ &= \underbrace{\int_{t^n}^{t^{n+1}} a(x_{i-1/2},t) f(x_{i-1/2},t)\, dt}_{T_3} + \underbrace{\int_{x^*_{i-1/2}}^{x_{i-1/2}} -f(x,t^n)\, dx}_{T_4} + \\ &\quad \underbrace{\int_{(x^*_{i-1/2},t^n)}^{(x_{i-1/2},t^{n+1})} \boldsymbol{F}\cdot\boldsymbol{n}\, ds}_{T_5}. \end{aligned} \tag{2.35}$$

Le terme T_3 est similaire aux termes contenus dans T_2. Le terme T_4 peut être calculé si le point $x^*_{i-1/2}$ est connu. Nous allons à présent montrer que le terme T_5 est nul.

L'intégrale T_5 est calculée entre $(x^*_{i-1/2}, t^n)$ et $(x_{i-1/2}, t^{n+1})$ le long d'une caractéristique associée à l'opérateur de transport

$$\partial_t + a\partial_x\ . \tag{2.36}$$

Par définition, les caractéristiques associées à l'équation (2.26) sont définies par l'équation

$$\frac{d}{dt} X(x^*_{i-1/2}, t) = a(X(x^*_{i-1/2}, t), t), \tag{2.37a}$$
$$X(x^*_{i-1/2}, t^n) = x^*_{i-1/2}, \tag{2.37b}$$
$$X(x_{i-1/2}, t^{n+1}) = x_{i-1/2}. \tag{2.37c}$$

Sur la figure 2.2, $\boldsymbol{U}$ est le vecteur directeur de la caractéristique et $\boldsymbol{n}$ est la normale unitaire à $\boldsymbol{U}$. On a donc

$$\boldsymbol{n}\cdot\boldsymbol{U} = 0, \tag{2.38a}$$

$$\|\boldsymbol{n}\| = 1, \tag{2.38b}$$

et on pose

$$d\boldsymbol{l} = \boldsymbol{U}\,dt, \tag{2.39}$$

$$ds = \left\|d\boldsymbol{l}\right\| = \left\|\boldsymbol{U}\right\| dt. \tag{2.40}$$

Par définition, on a

$$\boldsymbol{U} = \frac{d\boldsymbol{l}}{dt} = \frac{d}{dt}\begin{pmatrix} t \\ X(t) \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{dX}{dt} \end{pmatrix} = \begin{pmatrix} 1 \\ a \end{pmatrix}, \tag{2.41}$$

et par conséquent,

$$ds = \sqrt{1+a^2}dt. \tag{2.42}$$

On réécrit alors T_5 pour obtenir

$$\begin{aligned} T_5 &= \int_{(x^*_{i-1/2},t^n)}^{(x_{i-1/2},t^{n+1})} \boldsymbol{F}(x,t)\cdot\boldsymbol{n}(x,t)\,ds \\ &= \int_{(x^*_{i-1/2},t^n)}^{(x_{i-1/2},t^{n+1})} f(x,t)\begin{pmatrix} 1 \\ a(x,t) \end{pmatrix}\cdot\begin{pmatrix} a(x,t) \\ -1 \end{pmatrix}\frac{1}{\sqrt{1+a^2}}\,ds \\ &=0. \end{aligned} \tag{2.43}$$

En utilisant $T_5 = 0$, (2.35) donne

$$\int_{t^n}^{t^{n+1}} a(x_{i-1/2},t)f(x_{i-1/2},t)\,dt = \int_{x^*_{i-1/2}}^{x_{i-1/2}} f(x,t^n)\,dx. \tag{2.44}$$

On utilise l'équation (2.44) pour calculer les termes de T_2 et ainsi obtenir

$$T_2 = -\varphi(x^+_{i-1/2})\int_{x^*_{i-1/2}}^{x_{i-1/2}} f_h^n(x)\,dx + \varphi(x^-_{i+1/2})\int_{x^*_{i+1/2}}^{x_{i+1/2}} f_h^n(x)\,dx. \tag{2.45}$$

Comme pour le terme T_0, il ne s'agit plus alors que d'intégrer au temps n la distribution sur un domaine connu.

Le terme T_1 est similaire au terme T_2 mais contient en plus une intégrale en espace. Cette intégrale est donc résolue en utilisant une formule de quadrature pour se ramener au cas précédent.

$$\begin{aligned} T_1 &= \int_{t^n}^{t^{n+1}}\int_{I_i} af_h(x,t)\varphi'(x)\,dx\,dt \\ &= \int_{t^n}^{t^{n+1}} \frac{\Delta x_i}{2}\sum_{i_g} af_h(x_{i_g},t)\varphi'(x_{i_g})w_{i_g}\,dt \\ &= \frac{\Delta x_i}{2}\sum_{i_g}\varphi'(x_{i_g})w_{i_g}\int_{x^*_{i_g}}^{x_{i_g}} f_h^n(x)\,dx. \end{aligned} \tag{2.46}$$

où $\{x_{i_g}, w_{i_g}\}_{i_g}$ sont respectivement les points et les poids associés à la formule de quadrature choisie. On voit que comme la fonction test φ ne dépend pas du temps elle n'intervient pas dans le processus de réécriture de T_1 et T_2.

On rappelle que la formule de quadrature de Gauss à n points pour des polynômes p de degré inférieur à $2n-1$ est

$$\int_{-1}^{1} p(x)\,dx = \sum_{i_g=0}^{n} w_{i_g} p(x_{i_g})$$

et que cette méthode sur un intervalle de longueur Δx_i devient

$$\int p(x)\,dx = \frac{\Delta x_i}{2} \sum_{i_g=0}^{n} w_{i_g} p(x_{i_g}).$$

Le pied des caractéristiques est obtenu en résolvant par la méthode désirée le système (2.37). La résolution mise en œuvre est décrite dans la section 4.3. Dans ce système on connaît le point d'arrivée des caractéristiques. La méthode est donc une méthode semi-lagrangienne en arrière, appelée en anglais « backward semi-Lagrangian ». De plus amples détails sont indiqués dans la section 4.3, notamment en matière de programmation.

2.3.2 Application de la méthode GDSL à l'équation de Vlasov

On considère le maillage $T_{i,j} = I_i \times J_j$. Cette description correspond à un maillage en deux dimensions, cartésien, conforme et structuré. L'utilisation d'un maillage conforme simplifie les écritures mais n'est absolument pas nécessaire au schéma numérique, la généralisation à un maillage non conforme est immédiate puisque le calcul des intégrales en espace n'est pas explicité ici.

On considère les espaces

$$\mathcal{P}^n = \left\{ f \text{ telle que } f|_{T_{i,j}} \in \mathbb{P}^n \right\}$$

et

$$\mathcal{P}^n_{i,j} = \left\{ f \text{ telle que } f|_{T_{i,j}} \in \mathcal{P}^n,\ f|_{\mathbb{R}^2 \setminus T_{i,j}} = 0 \right\}.$$

On peut alors définir $\{P_{\boldsymbol{k}}|_{T_{i,j}}\}_{\boldsymbol{k}}$ comme base de $\{\bigcup_{i,j,\boldsymbol{k}} P_{\boldsymbol{k}}|_{T_{i,j}}\}$ comme base de $\mathcal{P}^n$. Les formules (2.18) et (2.19) sont généralisées à l'utilisation de l'espace $\mathcal{P}^n$ avec sa base associée. On utilise les polynômes de Legendre normalisés comme base de polynômes sur un intervalle de référence, comme indiqué à la section 1.3. On notera dans la suite de cette section par un tilde les fonctions et les points pris sur l'intervalle de référence. Il suffira d'effectuer une homothétie pour passer de la fonction ou de la variable définie sur l'intervalle de travail (cas par défaut) à son équivalent (même nom avec un tilde) sur l'intervalle de référence. Les indices et polynômes en dimension deux et en dimension une sont toujours liés par les relations $\boldsymbol{k} = (k_1, k_2)$ et $P_{\boldsymbol{k}}(x,v) = p_{k_1}(x) p_{k_2}(v)$.

Comme on le voit dans la formule (2.46), la dérivée première de la fonction test intervient. En une dimension la dérivée du k-ième polynôme de Legendre au i_g-ème point de quadrature de Gauss-Legendre sur l'intervalle I_i de longueur Δx_i peut s'écrire $p'_k(x_{i_g}) = \frac{2}{\Delta x_i} \tilde{p}'_k(\tilde{x}_{i_g})$. Dans cette expression, le 2 au numérateur est la longueur de l'intervalle de référence.

Le système de Vlasov-Poisson est un système à deux dimensions et les équations présentées précédemment l'ont été dans le cadre de problème à une dimension. L'extension aux dimensions supérieures se fait par une méthode de splitting (cf. section 4.5). Les notations f_k^n et f_k^{n+1} sont des abus de langage à but explicatif. Ces notations désignent ici la distribution avant et après chaque étape du splitting. On peut dans ces conditions écrire dans la direction x

$$\begin{aligned} f_{\boldsymbol{k},T_{i,j}}^{n+1} &= \int_{T_{i,j}} f_h^{n+1}(x,v) P_{\boldsymbol{k}}(x,v)\, dxdv \\ &= \int_{T_{i,j}} f_h^{n}(x,v) P_{k}(x,v)\, dxdv \\ &\quad + \int_{J_j} \frac{\Delta x_i}{2} \sum_{i_g} w_{i_g} \int_{x_{i_g}^*}^{x_{i_g}} f_h^{n+1}(x,v)\, dx\, p'_{k_1}(x_{i_g}) p_{k_2}(v)\, dv \\ &\quad + \int_{J_j} \left(\int_{x^*_{i-1/2}}^{x_{i-1/2}} f_h^n(x,v)\, dx\, p_{k_1}(x^+_{i-1/2}) - \right. \\ &\qquad \left. \int_{x^*_{i+1/2}}^{x_{i+1/2}} f_h^n(x,v)\, dx\, p_{k_1}(x^-_{i+1/2}) \right) p_{k_2}(v)\, dv. \end{aligned} \tag{2.47}$$

On reconnaît dans cette équation le terme $f^n_{\boldsymbol{k},T_{i,j}}$. Comme les dérivées sont calculées sur l'intervalle de référence la formule peut être réécrite avec la substitution.

$$\begin{aligned} f_{\boldsymbol{k},T_{i,j}}^{n+1} =& f_{\boldsymbol{k},T_{i,j}}^{n} \\ &+ \int_{J_j} \sum_{i_g} w_{i_g} \int_{x_{i_g}^*}^{x_{i_g}} f_h^{n+1}(x,v)\, dx\, \tilde{p}'_{k_1}(\tilde{x}_{i_g}) p_{k_2}(v)\, dv \\ &+ \int_{J_j} \left(\int_{x^*_{i-1/2}}^{x_{i-1/2}} f_h^n(x,v)\, dx\, p_{k_1}(x^+_{i-1/2}) - \int_{x^*_{i+1/2}}^{x_{i+1/2}} f_h^n(x,v)\, dx\, p_{k_1}(x^-_{i+1/2}) \right) p_{k_2}(v)\, dv. \end{aligned} \tag{2.48}$$

Les termes $f^{n+1}_{k,T_{i,j}}$ et $f^n_{k,T_{i,j}}$ sont ici exprimés sur $T_{i,j}$, ce qui implique que la base de polynômes soit aussi exprimée sur $T_{i,j}$. Pour des raisons pratiques, il est plus simple de travailler uniquement avec la distribution décomposée sur l'intervalle de référence. Les seules grandeurs à projeter entre le domaine de calcul et l'intervalle de référence sont alors les points de calcul. La base de polynôme n'a ainsi besoin d'être exprimée que sur l'intervalle de référence, et pas sur chaque cellule du domaine de calcul. On obtient alors l'écriture suivante.

$$\begin{aligned} \frac{\Delta x_i \Delta v_j}{4} \tilde{f}_{\boldsymbol{k},T_{i,j}}^{n+1} =& \frac{\Delta x_i \Delta v_j}{4} \tilde{f}_{\boldsymbol{k},T_{i,j}}^{n} \\ &+ \int_{J_j} \sum_{i_g} w_{i_g} \int_{x_{i_g}^*}^{x_{i_g}} f_h^{n+1}(x,v)\, dx\, \tilde{p}'_{k_1}(\tilde{x}_{i_g}) p_{k_1}(v)\, dv \\ &+ \int_{J_j} \left(\int_{x^*_{i-1/2}}^{x_{i-1/2}} f_h^n(x,v)\, dx\, p_{k_1}(x^+_{i-1/2}) - \right. \\ &\qquad \left. \int_{x^*_{i+1/2}}^{x_{i+1/2}} f_h^n(x,v)\, dx\, p_{k_1}(x^-_{i+1/2}) \right) p_{k_2}(v)\, dv, \end{aligned} \tag{2.49}$$

ou encore

$$\begin{aligned}\tilde{f}^{n+1}_{\boldsymbol{k},T_{i,j}} =& \tilde{f}^{n}_{\boldsymbol{k},T_{i,j}} \\ &+ \frac{4}{\Delta x_i \Delta v_j} \int_{J_j} \sum_{i_g} w_{i_g} \int_{x^*_{i_g}}^{x_{i_g}} f_h^{n+1}(x,v)\,dx\,\tilde{p}'_{k_1}(\tilde{x}_{i_g}) p_{k_2}(v)\,dv \\ &+ \frac{4}{\Delta x_i \Delta v_j} \int_{J_j} \left(\int_{x^*_{i-1/2}}^{x_{i-1/2}} f_h^{n}(x,v)\,dx\, p_{k_1}(x^+_{i-1/2}) - \right. \\ &\qquad \left. \int_{x^*_{i+1/2}}^{x_{i+1/2}} f_h^{n}(x,v)\,dx\, p_{k_1}(x^-_{i+1/2}) \right) p_{k_2}(v)\,dv,\end{aligned} \tag{2.50}$$

Les intégrales sur J_j sont calculées par une formule de quadrature. Les extrémités de l'intervalle $[x_{i-1/2}\,;\,x_{i+1/2}]$ projetées sur l'intervalle de référence sont respectivement -1 et 1. Le facteur de dilatation $\frac{\Delta v_j}{2}$ se simplifie avec les facteurs déjà présents et on arrive à la formule

$$\begin{aligned}\tilde{f}^{n+1}_{\boldsymbol{k},T_{i,j}} =& \tilde{f}^{n}_{\boldsymbol{k},T_{i,j}} \\ &+ \frac{2}{\Delta x_i} \sum_{j_g} w_{j_g} \left(\sum_{i_g} w_{i_g} \int_{x^*_{i_g}}^{x_{i_g}} f_h^{n+1}(x,v_{j_g})\,dx\,\tilde{p}'_{k_1}(\tilde{x}_{i_g}) \right) \tilde{p}_{k_2}(\tilde{v}_{j_g}) \\ &+ \frac{2}{\Delta x_i} \sum_{j_g} w_{j_g} \left(\int_{x^*_{i-1/2}}^{x_{i-1/2}} f_h^{n}(x,v_{j_g})\,dx\,\tilde{p}_{k_1}(-1) - \right. \\ &\qquad \left. \int_{x^*_{i+1/2}}^{x_{i+1/2}} f_h^{n}(x,v_{j_g})\,dx\,\tilde{p}_{k_1}(1) \right) \tilde{p}_{k_2}(\tilde{v}_{j_g})\end{aligned} \tag{2.51}$$

qui est la formule utilisée dans l'algorithme. Les intégrales restantes dans la direction x sont calculées selon le principe

$$\int_a^b f(x)\,dx = \sum_i \int_{[a\,;\,b]\cap I_i} f(x)dx. \tag{2.52}$$

Le calcul numérique de cette intégrale est expliqué au chapitre 4.

De façon totalement similaire, dans la direction v on obtient la formule équivalente.

$$\begin{aligned}\tilde{f}^{n+1}_{\boldsymbol{k},T_{i,j}} =& \tilde{f}^{n}_{\boldsymbol{k},T_{i,j}} \\ &+ \frac{2}{\Delta v_j} \sum_{i_g} w_{i_g} \left(\sum_{j_g} w_{j_g} \int_{v^*_{j_g}}^{v_{j_g}} f_h^{n+1}(x_{i_g},v)\,dv\,\tilde{p}'_{k_2}(\tilde{v}_{j_g}) \right) \tilde{p}_{k_1}(\tilde{x}_{i_g}) \\ &+ \frac{2}{\Delta v_j} \sum_{i_g} w_{i_g} \left(\int_{v^*_{j-1/2}}^{v_{j-1/2}} f_h^{n}(x_{i_g},v)\,dv\,\tilde{p}_{k_2}(-1) - \right. \\ &\qquad \left. \int_{v^*_{j+1/2}}^{v_{j+1/2}} f_h^{n}(x_{i_g},v)\,dv\,\tilde{p}_{k_2}(1) \right) \tilde{p}_{k_1}(\tilde{x}_{i_g}).\end{aligned} \tag{2.53}$$

2.4 La méthode caractéristiques-Galerkin discontinue

2.4.1 Présentation de la méthode

La méthode caractéristiques-Galerkin discontinue (CGD, ou CDG en anglais) est basée sur l'algorithme de [28]. Cependant, cette méthode avait déjà été décrite dans [18] sous le nom de « Lagrange-characteristic » et encore avant dans [22] et [35] sous le nom de projection de Lagrange dans des contextes différents. Le principe de ce schéma est de transporter les cellules dans l'espace pour appliquer la distribution au temps t^n sur la base de polynômes au temps t^{n+1}. Avant d'entrer dans les détails du schéma, posons quelques notations utiles. Soit Ψ une fonction polynomiale par morceaux telle que le degré des polynômes soit toujours inférieur ou égal à k. Soit ψ satisfaisant le système

$$\begin{cases} \partial_t \psi(x,t) + a(x,t)\partial_x \psi(x,t) = 0, \\ \psi(x,t^{n+1}) = \Psi(x). \end{cases} \tag{2.54}$$

On remarque que dans l'équation de transport (2.54), la solution reste constante le long des caractéristiques, ce qui n'est pas le cas pour l'équation de conservation (2.26).

Soit $I(t) = [x^*(x_{i-1/2},t)\ ;\ x^*(x_{i+1/2},t)]$, l'intervalle tel que $I(t^n) = I_i = [x_{i-1/2}\ ;\ x_{i+1/2}]$. Les points $x^*(x_{i\pm 1/2})$ sont alors les pieds des caractéristiques issues de $x_{i\pm 1/2}$. On a alors

$$\begin{aligned} \frac{d}{dt}\int_{I_i(t)} f_h(x,t)\psi(x,t)\,dx =& \frac{dx}{dt}|_{x_{i+1/2}} f(x^*(x_{i+1/2},t),t)\psi(x^*(x_{i+1/2},t),t) \\ &- \frac{dx}{dt}|_{x_{i-1/2}} f(x^*(x_{i-1/2},t),t)\psi(x^*(x_{i-1/2},t),t) \\ &+ \int_{I(t)} \partial_t \left(f(x,t)\psi(x,t)\right)\,dx \\ =& a(x^*(x_{i+1/2},t),t)f(x^*(x_{i+1/2},t),t)\psi(x^*(x_{i+1/2},t),t) \\ &- a(x^*(x_{i-1/2},t),t)f(x^*(x_{i-1/2},t),t)\psi(x^*(x_{i-1/2},t),t) \\ &+ \int_{I(t)} \psi(x,t)\partial_t f(x,t)\,dx + \int_{I(t)} f(x,t)\partial_t\psi(x,t)\,dx \\ =& (af\psi)(x^*(x_{i+1/2},t),t) - (af\psi)(x^*(x_{i-1/2},t),t) \\ &- \int_{I(t)} \psi(x,t)\partial_x(a(x,t)f(x,t))\,dx + \int_{I(t)} f(x,t)\partial_t\psi(x,t)\,dx \\ =& (af\psi)(x^*(x_{i+1/2},t),t) - (af\psi)(x^*(x_{i-1/2},t),t) \\ &+ \int_{I(t)} f(x,t)a(x,t)\partial_x\psi(x,t)\,dx + \int_{I(t)} f(x,t)\partial_t\psi(x,t)\,dx \\ &- a(x^*(x_{i+1/2},t),t)f(x^*(x_{i+1/2},t),t)\psi(x^*(x_{i+1/2},t),t) \\ &+ a(x^*(x_{i-1/2},t),t)f(x^*(x_{i-1/2},t),t)\psi(x^*(x_{i-1/2},t),t) \\ =& 0. \end{aligned} \tag{2.55}$$

Cela qui signifie que

$$\int_{I_i(t_1)} f_h(x,t_1)\psi(x,t_1)\,dx = \int_{I_i(t_2)} f_h(x,t_2)\psi(x,t_2)\,dx, \quad \forall\{t_1,t_2\}, \tag{2.56}$$

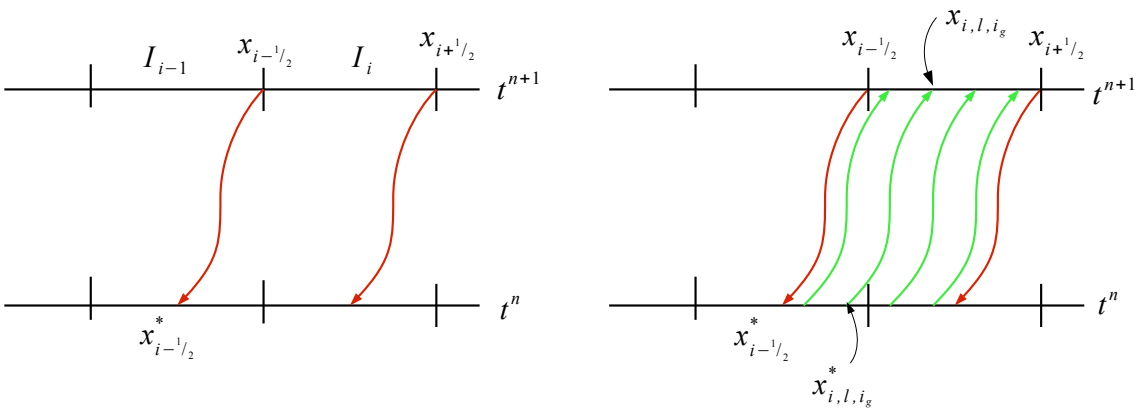


FIGURE 2.3 – Schéma des étapes 1 (à gauche) et 3–4 (à droite) de l'algorithme GC avec deux points de quadrature.

et en particulier

$$\int_{I_i(t^n)} f_h^n(x)\psi(x,t^n)\,dx = \int_{I_i(t^{n+1})} f_h^{n+1}(x)\Psi(x)\,dx. \tag{2.57}$$

On note alors $I_i(t^n) = [x^*_{i-1/2}\,;\, x^*_{i+1/2}]$. Le schéma pour calculer f_h^{n+1} est alors le suivant.

Étape 1 : déterminer $x^*_{i\pm1/2}$, pieds des caractéristiques issues respectivement de $x_{i\pm1/2}$. Dans le cas de l'advection à vitesse constante uniforme (comme c'est le cas pour Vlasov-Poisson avec résolution par splitting) $x^*_{i\pm1/2} = x_{i\pm1/2} - a\Delta t$. C'est l'illustration de gauche sur la figure 2.3.

Étape 2 : détecter l'ensemble des sous-intervalles $I^*_{i,l} = I_l \cap I_i(t^n)$ intersections de l'intervalle translaté avec le maillage.

Étape 3 : sur chaque intervalle $I^*_{i,l}$, placer les points de quadrature de la formule choisie (par homothétie des points depuis l'intervalle de référence considéré). Ces points sont notés x^*_{i,l,i_g}. On voit donc qu'il s'agit de pieds de caractéristiques.

Étape 4 : avancer les caractéristiques partant des points x^*_{i,l,i_g} pour trouver les points x_{i,l,i_g} au temps t^{n+1}. Ces points doivent se trouver sur l'intervalle I_i. L'équation (2.54) nous donne

$$\psi(x^*_{i,l,i_g}, t^n) = \Psi(x_{i,l,i_g}). \tag{2.58}$$

Ce sont les courbes vertes sur la figure 2.3. L'équation de conservation (2.26) nous permet également d'aller plus loin avec l'égalité

$$f_h^{n+1}(x) = f_h^n(x^*)\left|\frac{\partial x^*}{\partial x}\right| \tag{2.59}$$

ou $\left|\frac{\partial x^*}{\partial x}\right|$ est le jacobien et

$$dx^* = \left|\frac{\partial x^*}{\partial x}\right| dx. \tag{2.60}$$

On peut alors écrire

$$\int_{I_i(t^{n+1})} f_h^{n+1}(x)\Psi(x)\,dx = \int_{I_i(t^n)} f_h^n(x^*)\Psi(x(x^*))\,dx^*. \tag{2.61}$$

On remarque que dans le membre de droite, la variable d'intégration est x^*, exprimée au temps t^n mais la fonction test Ψ dépend de x exprimé au temps t^{n+1}. Les deux sont liées par la relation (2.37).

Étape 5 : utiliser la formule de quadrature pour résoudre l'équation (2.57)

$$\begin{aligned}\int_{I_i(t^{n+1})} f_h^{n+1}(x)\Psi(x)\,dx &= \int_{I_i(t^n)} f_h^n(x)\psi(x,t^n)\,dx \\ &= \int_{I_i(t^n)} f_h^n(x^*)\Psi(x(x^*))\,dx^* \\ &\approx \sum_l \left(\sum_{i_g} w_{i_g} f_h^n(x^*_{i,l,i_g})\Psi(x_{i,l,i_g})\frac{\Gamma(I^*_{i,l})}{2} \right),\end{aligned} \tag{2.62}$$

où les w_{i_g} sont les poids de la formule de quadrature, $\Gamma(I^*_{i,l})$ est la longueur de l'intervalle $I^*_{i,l}$ et $\frac{\Gamma(I^*_{i,l})}{2}$ est le facteur d'échelle. La somme sur l correspond au travail réalisé à l'étape 2.

Les étapes 1 et 3–4 sont illustrées sur la Figure 2.3. À gauche les caractéristiques permettent de trouver les points $x^*_{i\pm 1/2}$. À droite deux points de quadrature x^*_{i,l,i_g} sont placés sur chacun des intervalles $I^*_{i,l}$ et les caractéristiques permettent de trouver les points x_{i,l,i_g} correspondants.

On constate lors de la cinquième et dernière étape du schéma que si Ψ est le k-ième polynôme de la base sur laquelle est décomposée la distribution, alors

$$f_k^{n+1} = \int_{I_i(t^{n+1})} f_h^{n+1}(x)\Psi(x)\,dx = \sum_l \left(\sum_{i_g} w_{i_g} f_h^n(x^*_{i,l,i_g}) p_k(x_{i,l,i_g})\frac{\Gamma(I^*_{i,l})}{2} \right). \tag{2.63}$$

2.4.2 Application de la méthode CGD à l'équation de Vlasov

On applique l'algorithme décrit dans la section précédente avec une intégration dans l'espace des phases à deux dimensions en s'appuyant sur ce qui a déjà été fait dans la section 2.3.1. On procède toujours par splitting directionnel, la notation f_k^{n+1} est donc un abus de langage à but explicatif. On peut alors écrire dans la direction x

$$\begin{aligned}f^{n+1}_{\boldsymbol{k},T_{i,j}} &= \int_{T_{i,j}} f_h^{n+1}(x,v)P_{\boldsymbol{k}}(x,v)\,dxdv \\ &= \int_{J_j}\int_{I_i^*} f_h^n(x^*,v)P_{\boldsymbol{k}}(x,v)\,dxdv \\ &= \frac{1}{2}\int_{J_j} \sum_l \sum_{i_g} w_{i_g} f_h^n(x_{i_g,l},v)P_{\boldsymbol{k}}(x_{i_g},l,v)\Gamma(I^*_{i,l}(v)))\,dv \\ &= \frac{\Delta v_j}{4}\sum_{j_g} w_{j_g} \sum_l \sum_{i_g} w_{i_g} f_h^n(x^*_{i_g,l,j_g},v_{j_g})P_{\boldsymbol{k}}(x_{i_g,l,j_g},v_{j_g})\Gamma(I^*_{i,l,j_g}).\end{aligned} \tag{2.64}$$

Comme pour passer des équations (2.49) à (2.50) on ramène la décomposition de la distribution $\{f_k\}_k$ à l'intervalle de référence pour obtenir la formule finale

$$\tilde{f}^{n+1}_{k,T_{i,j}} = \sum_{j_g} w_{j_g} \sum_l \sum_{i_g} w_{i_g} f^n_h(x^*_{i_g,l,j_g}, v_{j_g}) P_{\boldsymbol{k}}(x_{i_g,l,j_g}, v_{j_g}) \frac{\Gamma(I^*_{i,l,j_g})}{\Delta x_i}. \tag{2.65}$$

On remarque qu'ici Γ exprime une dépendance dans la direction transverse puisque les intervalles d'intégration dépendent de la coordonnée dans la seconde direction.

On procède de même dans la direction v pour obtenir la formule

$$\tilde{f}^{n+1}_{k,T_{i,j}} = \sum_{i_g} w_{i_g} \sum_l \sum_{j_g} w_{j_g} f^n_h(x_{i_g}, v^*_{j_g,l,i_g}) P_{\boldsymbol{k}}(x_{i_g}, v_{j_g,l,i_g}) \frac{\Gamma(J^*_{j,l,i_g})}{\Delta v_j}. \tag{2.66}$$

Chapitre 3

Représentation en multi-ondelettes et adaptativité

3.1 Principe des multi-ondelettes

Le principe des multi-ondelettes est expliqué dans la section 3 de [5] et repris ici. Dans un soucis de cohérence avec l'article de référence, les calculs sont ici faits sur l'intervalle $[0\ ;\ 1]$. Les valeurs seront rapportées à l'intervalle de référence $[-1\ ;\ 1]$ lors du calcul des filtres. La norme est donc aussi calculée sur $[0\ ;\ 1]$. Les multi-ondelettes ont depuis été largement étudiées et utilisées. Une analyse approfondie de leurs propriétés se trouve dans [5]. Des exemples illustrent l'utilisation des multi-ondelettes dans [30] et [27]. Un algorithme pour la construction des bases de multi-ondelettes est détaillé dans [6] et repris à la section 3.4.

On considère l'espace de fonction à une variable $\mathcal{V}_n^k$ tel que

$$\mathcal{V}_n^k = \left\{ f|_{[2^{-n}l\ ;\ 2^{-n}(l+1)]} \in \mathbb{P}^k([2^{-n}l\ ;\ 2^{-n}(l+1)]) \quad \forall l \in [\![0\ ;\ 2^n-1]\!] \right\},$$

avec $\mathbb{P}^k([2^{-n}l\ ;\ 2^{-n}(l+1)])$ l'ensemble des polynômes de degré inférieur ou égal à k sur $[2^{-n}l\ ;\ 2^{-n}(l+1)]$. L'espace $\mathcal{V}_n^k$ est donc de dimension $2^n(k+1)$ et par construction,

$$\mathcal{V}_0^k \subset \mathcal{V}_1^k \subset \cdots \subset \mathcal{V}_n^k \subset \cdots \subset L^2([0\ ;\ 1]). \tag{3.1}$$

$\mathcal{V}_n^k$ est l'espace des fonctions d'échelles. L'ensemble des $\{\mathcal{V}_n^k\}_n$ est dense dans $L^2([0\ ;\ 1])$, c'est à dire

$$\overline{\bigcup_{n=0}^{\infty} \mathcal{V}_n^k} = L^2\left([0\ ;\ 1]\right). \tag{3.2}$$

On définit $\mathcal{W}_n^k$ l'espace des multi-ondelettes tel que

$$\mathcal{V}_n^k \oplus \mathcal{W}_n^k = \mathcal{V}_{n+1}^k, \quad \mathcal{V}_n^k \perp \mathcal{W}_n^k. \tag{3.3}$$

Par extension de la formule (3.3) on peut écrire

$$\mathcal{V}_n^k = \mathcal{V}_0^k \oplus \mathcal{W}_0^k \oplus \mathcal{W}_1^k \oplus \cdots \oplus \mathcal{W}_{n-1}^k. \tag{3.4}$$

On note $\{\phi_j\}$ une base orthonormée de $\mathcal{V}_0^k$ et par dilatation et translation les $2^n(k+1)$ fonctions de base de $\mathcal{V}_n^k$ sont

$$\phi_{j,l}^n(x) = 2^{n/2}\phi_j(2^n x - l), \quad j \in [\![0\ ;\ k]\!], \quad l \in [\![0\ ;\ 2^n - 1]\!]. \tag{3.5}$$

De même, on note $\{\psi_j\}$ une base orthonormée de $\mathcal{W}_0^k$ et sur le principe de l'équation (3.5) cette base peut être étendue aux niveaux supérieurs :

$$\psi_{j,l}^n(x) = 2^{n/2}\psi_j(2^n x - l), \quad j \in [\![0\ ;\ k]\!], \quad l \in [\![0\ ;\ 2^n - 1]\!]. \tag{3.6}$$

Par définition on a $\mathcal{V}_n^k \perp \mathcal{W}_n^k$, ce qui signifie que

$$\int_0^1 \psi_j(x) x^i \, dx = 0, \quad \forall i, j \in [\![0\ ;\ k]\!]. \tag{3.7}$$

Cette propriété sera exploitée lors de la construction de la base de multi-ondelettes. De toutes ces propriétés découle une propriété supplémentaire,

$$\int_0^1 \psi_{i,l}^n(x)\psi_{j,m}^{n'}(x)\, dx = \delta_{i,j}\delta_{l,m}\delta_{n,n'} \tag{3.8}$$

où δ est le symbole de Kronecker.

3.2 Équations multi-échelles

On utilise les polynômes de Legendre non orthonormés $\{\widehat{P}_j(x)\}_j$ dans un premier temps et on définit les fonctions d'échelle $\{\phi_j(x)\}_j$ à partir des polynômes de Legendre par

$$\phi_j(x) = \begin{cases} \sqrt{2j+1}\widehat{P}_j(2x-1) & \text{si } x \in [0\ ;\ 1], \\ 0 & \text{sinon.} \end{cases} \tag{3.9}$$

On peut lier les polynômes de Legendre $\widehat{P}_j(x)$ aux polynômes de Legendre normalisés par la formule

$$\tilde{P}_j(x) = \sqrt{j + 1/2}\,\widehat{P}_j(x). \tag{3.10}$$

On en déduit immédiatement que $\phi_j(x)$ correspond aux polynômes de Legendre normalisés sur l'intervalle $[0\ ;\ 1]$, c'est à dire

$$\int_0^1 (\phi_j(x))^2 = 1. \tag{3.11}$$

On peut donc trouver des filtres $\{h_{i,j}^{(0)}\}_{i,j}$, $\{h_{i,j}^{(1)}\}_{i,j}$, $\{g_{i,j}^{(0)}\}_{i,j}$ et $\{g_{i,j}^{(1)}\}_{i,j}$ dépendant de l'ordre n d'interpolation et permettant d'exprimer les relations décrites par les équations (3.1) et (3.3) dans une opération de transition entre deux échelles consécutives

$$\phi_i(x) = \sqrt{2}\sum_{j=0}^{k}\left(h_{i,j}^{(0)}\phi_j(2x) + h_{i,j}^{(1)}\phi_j(2x-1)\right), \qquad i \in [\![0\ ;\ k]\!], \tag{3.12a}$$

$$\psi_i(x) = \sqrt{2}\sum_{j=0}^{k}\left(g_{i,j}^{(0)}\phi_j(2x) + g_{i,j}^{(1)}\phi_j(2x-1)\right), \qquad i \in [\![0\ ;\ k]\!]. \tag{3.12b}$$

On pose alors les matrices de coefficients

$$H^{(0)} = \{h_{i,j}^{(0)}\}, \quad H^{(1)} = \{h_{i,j}^{(1)}\}, \quad G^{(0)} = \{g_{i,j}^{(0)}\}, \quad G^{(1)} = \{g_{i,j}^{(1)}\}. \tag{3.13}$$

Par construction, en utilisant le produit scalaire

$$\langle f, g\rangle = \int_0^1 f(x)g(x)\,dx,$$

on a $\langle \phi_i, \phi_j\rangle = \delta_{i,j}$, $\langle \psi_i, \psi_j\rangle = \delta_{i,j}$ et $\langle \phi_i, \psi_j\rangle = 0$, quels que soient i et j. En utilisant ces relations d'orthogonalité et les équations (3.12) on obtient les relations matricielles suivantes

$$H^{(0)}H^{(0)T} + H^{(1)}H^{(1)T} = I, \tag{3.14a}$$
$$G^{(0)}G^{(0)T} + G^{(1)}G^{(1)T} = I, \tag{3.14b}$$
$$H^{(0)}G^{(0)T} + H^{(1)}G^{(1)T} = 0. \tag{3.14c}$$

On introduit

$$U = \begin{pmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{pmatrix}.$$

On note alors que $UU^T = I$. Par conséquent U est une matrice unitaire, c'est-à-dire $UU^T = U^TU = I$. Ceci peut également se traduire avec les matrices $H^{(0)}$, $H^{(1)}$, $G^{(0)}$ et $G^{(1)}$ sous la forme

$$H^{(0)T}H^{(0)} + G^{(0)T}G^{(0)} = I, \tag{3.15a}$$
$$H^{(1)T}H^{(1)} + G^{(1)T}G^{(1)} = I, \tag{3.15b}$$
$$H^{(0)T}H^{(1)} + G^{(0)T}G^{(1)} = 0. \tag{3.15c}$$

Cette dernière équation nous permet d'écrire la deuxième relation de transition entre deux échelles consécutive

$$\phi_i(2x) = \frac{1}{\sqrt{2}}\sum_{j=0}^{k}\left(h_{j,i}^{(0)}\phi_j(x) + g_{j,i}^{(0)}\psi_j(x)\right), \tag{3.16a}$$
$$\phi_i(2x-1) = \frac{1}{\sqrt{2}}\sum_{j=0}^{k}\left(h_{j,i}^{(1)}\phi_j(x) + g_{j,i}^{(1)}\psi_j(x)\right). \tag{3.16b}$$

Si on multiplie l'équation (3.12a) par $\sqrt{2}\phi_j(2x)$ puis qu'on intègre sur $[0\ ;\ 1]$, on obtient alors

$$h_{i,j}^{(0)} = \sqrt{2}\int_0^{1/2}\phi_i(x)\phi_j(2x)\,dx, \tag{3.17}$$

qui peut être calculé exactement en utilisant une formule de quadrature

$$h_{i,j}^{(0)} = \frac{1}{\sqrt{2}}\sum_{m=0}^{k} w_m\phi_i\left(\frac{x_m}{2}\right)\phi_j(x_m). \tag{3.18a}$$

En faisant de même pour toutes les équations (3.12) on obtient finalement

$$h_{i,j}^{(1)} = \frac{1}{\sqrt{2}} \sum_{m=0}^{k} w_m \phi_i\left(\frac{x_m+1}{2}\right)\phi_j(x_m), \tag{3.18b}$$

$$g_{i,j}^{(0)} = \frac{1}{\sqrt{2}} \sum_{m=0}^{k} w_m \psi_i\left(\frac{x_m}{2}\right)\phi_j(x_m), \tag{3.18c}$$

$$g_{i,j}^{(1)} = \frac{1}{\sqrt{2}} \sum_{m=0}^{k} w_m \psi_i\left(\frac{x_m+1}{2}\right)\phi_j(x_m). \tag{3.18d}$$

On notera qu'ici les points et poids de quadrature sont donnés sur l'intervalle $[0\ ;\ 1]$ et non sur l'intervalle de référence habituel $[-1\ ;\ 1]$.

En utilisant la parité des polynômes de Legendre et les équations (3.12) et (3.18) on peut remarquer que

$$h_{i,j}^{(1)} = (-1)^{i+j} h_{i,j}^{(0)}, \quad 0 \le i,j \le n, \tag{3.19a}$$

$$g_{i,j}^{(1)} = (-1)^{i+j+k+1} g_{i,j}^{(0)}, \quad 0 \le i,j \le n. \tag{3.19b}$$

3.3 Décomposition et reconstruction à l'aide des multi-ondelettes

On peut écrire $P_n f$ la projection au niveau n de f sur la base des fonctions d'échelle sous la forme

$$P_n f(x) = \sum_{l=0}^{2^n-1} \sum_{j=0}^{k} s_{j,l}^n \phi_{j,l}^n(x), \tag{3.20}$$

ce qui, par l'équation (3.4), peut aussi se traduire sous la forme

$$P_n f(x) = \sum_{j=0}^{k} s_{j,0}^0 \phi_j(x) + \sum_{j=0}^{k} \sum_{m=0}^{n-1} \sum_{l=0}^{2^m-1} d_{j,l}^m \psi_{j,l}^m(x), \tag{3.21}$$

où les coefficients $\{s_{i,l}^m\}_{i,l,m}$ sont obtenus en décomposant f sur la base des fonctions $\{\phi_{i,l}^m\}_{i,l,m}$,

$$s_{j,l}^n = \int_{2^{-n}l}^{2^{-n}(l+1)} f(x)\phi_{j,l}^n(x)\,dx, \tag{3.22}$$

et les coefficients $\{d_{i,l}^m\}_{i,l,m}$ sur la base $\{\psi_{i,l}^m\}_{i,l,m}$,

$$d_{j,l}^m = \int_{2^{-m}l}^{2^{-m}(l+1)} f(x)\psi_{j,l}^m(x)\,dx. \tag{3.23}$$

À partir des équations (3.12), (3.16) et (3.21), on peut définir une opération de restriction du niveau $m+1$ vers le niveau m par

$$s_{i,l}^m = \sum_{j=0}^{k} \left(h_{i,j}^{(0)} s_{j,2l}^{m+1} + h_{i,j}^{(1)} s_{j,2l+1}^{m+1} \right), \tag{3.24a}$$

$$d_{i,l}^{m} = \sum_{j=0}^{k} \left(g_{i,j}^{(0)} s_{j,2l}^{m+1} + g_{i,j}^{(1)} s_{j,2l+1}^{m+1} \right), \tag{3.24b}$$

et une opération de prédiction ou d'interpolation du niveau m vers le niveau $m+1$

$$s_{i,2l}^{m+1} = \sum_{j=0}^{k} \left(h_{j,i}^{(0)} s_{j,l}^{m} + g_{j,i}^{(0)} d_{j,l}^{m} \right), \tag{3.25a}$$

$$s_{i,2l+1}^{m+1} = \sum_{j=0}^{k} \left(h_{j,i}^{(1)} s_{j,l}^{m} + g_{j,i}^{(1)} d_{j,l}^{m} \right). \tag{3.25b}$$

Les coefficients $s_{\cdot,\cdot}^{m}$ sont les coefficients d'échelle au niveau m et les coefficients $d_{\cdot,\cdot}^{m}$ sont les détails stockés dans les multi-ondelettes du niveau m et qui permettent de calculer les coefficients d'échelle au niveau $m+1$.

Les équations (3.24) et (3.25) sont obtenues sur $[0\ ;\ 1]$, on les ramène à présent sur notre intervalle de référence $[-1\ ;\ 1]$ en conservant la norme L^2. En partant des équations (3.9), (3.10), (3.22) et (3.23), on écrit

$$\begin{aligned}
s_{j,l}^{m} &= \int_{2^{-n}l}^{2^{-n}(l+1)} f(x)\phi_{j,l}^{n}(x)\,dx \\
&= \int_{2^{-n}l}^{2^{-n}(l+1)} f(x)\phi_{i}(2^{m}x - l)2^{m/2}\,dx \\
&= \int_{-1}^{1} f(2^{-m-1}(X+2l+1))\phi_{i}\left(\frac{X+2l+1}{2} - l\right)2^{m/2}2^{-m-1}\,dX \\
&= \int_{-1}^{1} f(2^{-m-1}(X+2l+1))2^{1/2}\tilde{p}_{i}(X)2^{m/2}2^{-m-1}\,dX \\
&= 2^{-m/2-1/2}\int_{-1}^{1} f(2^{-m-1}(X+2l+1))\tilde{p}_{i}(X)\,dX \\
&= 2^{-m/2-1/2}\tilde{s}_{i,l}^{m}.
\end{aligned} \tag{3.26}$$

Le calcul est le même pour passer de $d_{j,l}^{m}$ à $\tilde{d}_{j,l}^{m}$. On substitue ensuite dans (3.24) pour obtenir

$$\tilde{s}_{i,l}^{m} = \sqrt{2}\sum_{j=0}^{k} \left(h_{i,j}^{(0)} \tilde{s}_{j,2l}^{m+1} + h_{i,j}^{(1)} \tilde{s}_{j,2l+1}^{m+1} \right), \tag{3.27a}$$

$$\tilde{d}_{i,l}^{m} = \sqrt{2}\sum_{j=0}^{k} \left(g_{i,j}^{(0)} \tilde{s}_{j,2l}^{m+1} + g_{i,j}^{(1)} \tilde{s}_{j,2l+1}^{m+1} \right), \tag{3.27b}$$

et (3.25) devient

$$\tilde{s}_{i,2l}^{m+1} = \frac{1}{\sqrt{2}}\sum_{j=0}^{k} \left(h_{j,i}^{(0)} \tilde{s}_{j,l}^{m} + g_{j,i}^{(0)} \tilde{d}_{j,l}^{m} \right), \tag{3.28a}$$

$$\tilde{s}_{i,2l+1}^{m+1} = \frac{1}{\sqrt{2}}\sum_{j=0}^{k} \left(h_{j,i}^{(1)} \tilde{s}_{j,l}^{m} + g_{j,i}^{(1)} \tilde{d}_{j,l}^{m} \right). \tag{3.28b}$$

3.4 Construction des multi-ondelettes à une dimension

La construction des multi-ondelettes se fait selon la méthode présentée dans la section 1.1 de [6]. Cette méthode est rappelée ici. Le but est de construire une base de $L^2([-1\ ;\ 1])$ par dilatation et translation d'une famille finie $\{\psi_i\}_{i=0}^k$ de polynômes par morceaux sur $[-1\ ;\ 1]$ et nuls en dehors. La famille des $\{\psi_i\}_i$ est orthogonale aux polynômes de degré inférieur ou égal à k sur $[-1\ ;\ 1]$, c'est à dire

$$\int_{-1}^{1} \psi_j(x) x^i \, dx = 0, \quad \forall i \in [\![0\ ;\ k]\!],\ \forall j \in [\![0\ ;\ k]\!]. \tag{3.29}$$

Les fonctions $\psi_0, \ldots, \psi_k\ :\ [-1\ ;\ 1] \to \mathbb{R}$ satisfont les propriétés suivantes :

1. $\psi_i|_{[0\ ;\ 1]} \in \mathbb{P}^k([0\ ;\ 1])$;
2. ψ_i a la parité de $i+k$ et est étendue à $[-1\ ;\ 0]$ par ce principe ;
3. la famille $\{\psi_i\}_i$ est orthonormée, c'est à dire

$$\langle \psi_i, \psi_j \rangle = \delta_{i,j}, \quad i, j \in [\![0\ ;\ k]\!]\ ;$$

4. chaque fonction ψ_j a tous ses moments nuls jusqu'à l'ordre $j+k$, c'est-à-dire

$$\int_{-1}^{1} \psi_j(x) x^i \, dx = 0, \quad \forall i \in [\![0\ ;\ j+k]\!]. \tag{3.30}$$

Pour construire la famille $\{\psi_j\}_j$ on se munit des $k+1$ polynômes de Legendre formant la base de $\mathbb{P}^k$ et on se donne $k+1$ fonctions $\psi_0^1, \psi_1^1, \ldots, \psi_k^1$ telles que

$$\psi_j^1(x) = \begin{cases} x^j & \text{si } x \in]0\ ;\ 1], \\ -x^j & \text{si } x \in [-1\ ;\ 0[, \\ 0 & \text{sinon.} \end{cases} \tag{3.31}$$

Toutes ces fonctions, au nombre de $2(k+1)$ (polynômes de Legendre de degré inférieur ou égal à k et $\{\psi_j^1\}_{j=0}^k$), sont linéairement indépendantes et forment une base de polynômes de degré inférieur ou égal à k sur $]-1\ ;\ 0[\ \cup\ ]0\ ;\ 1[$. Les fonctions ψ_j^1 satisfont les propriétés 1 et 2.

Étape 1 : on orthogonalise les ψ_j^1 aux polynômes de Legendre p_i par le procédé de Gram-Schmidt décrit dans l'algorithme 3.1. Il est important dans cette étape d'orthogonaliser les fonctions ψ_j^1 à une famille elle-même orthogonale, sans quoi l'orthogonalisation ne serait pas possible. On obtient alors la famille des $\{\psi_j^2\}_{j=0}^k$.

Étape 2 : il s'agit en fait d'une série d'étapes visant à obtenir la propriété 4. Les ψ_j^2 sont réordonnées telles que $\langle \psi_0^2, x^{k+1} \rangle \neq 0$. On définit ensuite

$$\psi_j^3 = \psi_j^2 - \frac{\langle \psi_j^2, x^{k+1} \rangle}{\langle \psi_0^2, x^{k+1} \rangle} \psi_0^2,$$

ce qui nous garantit pour j de 1 à k l'orthogonalité entre ψ_j^3 et x^{k+1}. On continue de façon similaire de x^{k+2} à x^{2k} pour obtenir finalement ψ_0^2, ψ_1^3, ψ_2^4 jusqu'à ψ_k^{k+2} telles que $\langle \psi_j^{j+2}, x^i \rangle = 0$ pour $i \leq j+k$.

Étape 3 : on orthogonalise la famille $\{\psi_j^{j+2}\}_{j=0}^{k}$ dans l'ordre des j décroissants par le procédé de Gram-Schmidt décrit dans l'algorithme 3.2 pour obtenir les fonctions ψ_k, ψ_{k-1},...,ψ_0, fonctions de base pour l'espace des multi-ondelettes sur $[-1\ ;\ 1]$.

Algorithme 3.1 Algorithme de Gram-Schmidt pour orthogonaliser une famille avec une autre famille orthogonale

Pour $j = 0..k$ **faire**
  $f = \psi_j^1$
  Pour $i = 0..k$ **faire**
    $a = \langle f, p_i \rangle$
    $f = f - a \times p_i$
  Fin pour
  $\psi_j^2 = f$
Fin pour

Algorithme 3.2 Algorithme de Gram-Schmidt pour orthogonaliser entre eux les membres d'une même famille

Pour $j = k..0$ **faire**
  $f = \psi_j$
  Pour $i = j + 1..k$ **faire**
    $f = f - \langle f, \psi_i \rangle \psi_i$
  Fin pour
  $\psi_j = \frac{f}{\|f\|}$
Fin pour

3.5 Multi-ondelettes en deux dimensions

En dimension deux, on définit les fonctions d'échelle par le produit tensoriel des fonctions d'échelle en dimension 1

$$\phi_{\boldsymbol{k}}(x, y) = \phi_{k_1}(x)\phi_{k_2}(y) \tag{3.32}$$

De la même façon, on obtient l'ensemble des multi-ondelettes par produit tensoriel,

$$\mathcal{W}_n^k = \big\{\{\phi_{i,l}^n(x)\psi_{j,l}^n(y)\},\ \{\psi_{i,l}^n(x)\phi_{j,l}^n(y)\},\ \{\psi_{i,l}^n(x)\psi_{j,l}^n(y)\},\ \forall i, j \in [\![0\ ;\ k]\!],\ \forall l \in [\![0\ ;\ 2^n - 1]\!]\big\}\,. \tag{3.33}$$

Les détails sont alors indicés « $[a]$ », « $[b]$ » et « $[c]$ » pour faire référence respectivement aux trois catégories de multi-ondelettes de l'espace $\mathcal{W}_n^k$ ainsi décrit.

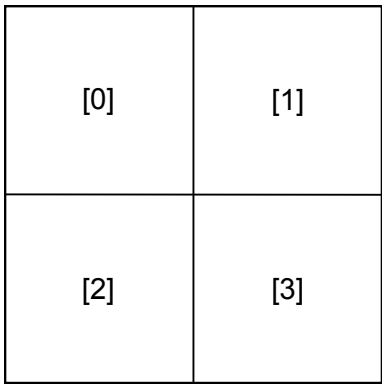


FIGURE 3.1 – Cellule de niveau m divisée en quatre cellules de niveau $m+1$ et numérotation des cellules filles.

On a pu voir grâce aux équations (3.27) et (3.28) que les calculs pour changer de niveau ne dépendent à chaque fois que d'une cellule mère et de ses cellules filles. Plutôt que d'avoir un indice l par direction, on s'intéresse dans cette partie au cas d'une cellule quelconque et l'indice l est remplacé comme suit. (Les notations utilisées ici font également référence au code.)

La figure 3.1 représente une cellule de niveau m qui a été raffinée en quatre cellules filles. Les numéros entre crochet indiquent la façon dont elles sont numérotées dans le code et dans l'algorithme. La notation $\tilde{s}^{m+1}_{p,[2]}$ réfère donc au coefficient d'échelle au niveau $m+1$, cellule fille de la cellule considérée au niveau m en position $[2]$ sur la figure, fonction d'échelle $\boldsymbol{p} = (p_1, p_2)$.

Avec ces notations, pour $K+1$ fonctions de base en dimension 2 (et donc $K+1$ coefficients), en utilisant la notation $\boldsymbol{q} = (q_1, q_2)$, les équations (3.27) deviennent en deux dimensions

$$\tilde{s}^m_{\boldsymbol{q}} = 2\sum_{\boldsymbol{p}=0}^{K} h^{(0)}_{q_1,p_1} h^{(0)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[2]} + h^{(0)}_{q_1,p_1} h^{(1)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[0]} + h^{(1)}_{q_1,p_1} h^{(0)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[3]} + h^{(1)}_{q_1,p_1} h^{(1)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[1]}, \quad (3.34a)$$

$$\tilde{d}^m_{\boldsymbol{q},[a]} = 2\sum_{\boldsymbol{p}=0}^{K} h^{(0)}_{q_1,p_1} g^{(0)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[2]} + h^{(0)}_{q_1,p_1} g^{(1)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[0]} + h^{(1)}_{q_1,p_1} g^{(0)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[3]} + h^{(1)}_{q_1,p_1} g^{(1)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[1]}, \quad (3.34b)$$

$$\tilde{d}^m_{\boldsymbol{q},[b]} = 2\sum_{\boldsymbol{p}=0}^{K} g^{(0)}_{q_1,p_1} h^{(0)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[2]} + g^{(0)}_{q_1,p_1} h^{(1)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[0]} + g^{(1)}_{q_1,p_1} h^{(0)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[3]} + g^{(1)}_{q_1,p_1} h^{(1)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[1]}, \quad (3.34c)$$

$$\tilde{d}^m_{\boldsymbol{q},[c]} = 2\sum_{\boldsymbol{p}=0}^{K} g^{(0)}_{q_1,p_1} g^{(0)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[2]} + g^{(0)}_{q_1,p_1} g^{(1)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[0]} + g^{(1)}_{q_1,p_1} g^{(0)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[3]} + g^{(1)}_{q_1,p_1} g^{(1)}_{q_2,p_2} \tilde{s}^{m+1}_{\boldsymbol{p},[1]}. \quad (3.34d)$$

Pour des polynômes en dimension un de degré inférieur ou égal à k, on a posé $K+1 = (k+1)^2$. Ici, p et q ne sont pas des vecteurs mais des indices allant chacun de 0 à K. Il y a $K+1$ coefficients impliquant chacun $4 \times (K+1)$ termes, ce qui signifie que le nombre d'opération total est de l'ordre de $4(K+1)^2$, ou encore, exprimé grâce au degré des polynômes en dimension 1, $4(k+1)^4$. Il est possible de réduire le nombre de calculs

pour améliorer les performances numériques. En procédant par direction il est possible de décomposer le calcul de la façon suivante.

$$c^m_{l,i,[0]} = \sum_{j=0}^{k} h^{(0)}_{l,j}\, \tilde{s}^{m+1}_{\{i,j\},[2]} + h^{(1)}_{l,j}\, \tilde{s}^{m+1}_{\{i,j\},[0]}, \tag{3.35a}$$

$$c^m_{l,i,[1]} = \sum_{j=0}^{k} h^{(0)}_{l,j}\, \tilde{s}^{m+1}_{\{i,j\},[3]} + h^{(1)}_{l,j}\, \tilde{s}^{m+1}_{\{i,j\},[1]}, \tag{3.35b}$$

$$c^m_{l,i,[2]} = \sum_{j=0}^{k} g^{(0)}_{l,j}\, \tilde{s}^{m+1}_{\{i,j\},[2]} + g^{(1)}_{l,j}\, \tilde{s}^{m+1}_{\{i,j\},[0]}, \tag{3.35c}$$

$$c^m_{l,i,[3]} = \sum_{j=0}^{k} g^{(0)}_{l,j}\, \tilde{s}^{m+1}_{\{i,j\},[3]} + g^{(1)}_{l,j}\, \tilde{s}^{m+1}_{\{i,j\},[1]}, \tag{3.35d}$$

où les $c^m_{l,i,[.]}$ sont des intermédiaires de calculs et $i,\ l \in [\![0\,;k]\!]$. Les $\tilde{s}^{m+1}_{\{i,j\},[.]}$ sont les coefficients en deux dimensions correspondant aux $\tilde{s}^{m+1}_{p,[2]}$ des équations (3.34) mais la décomposition par direction rend préférable cette notation.

$$\tilde{s}^m_{\{j,l\}} = 2\sum_{i=0}^{k} h^{(0)}_{j,i}\, c^m_{l,i,[0]} + h^{(1)}_{j,i}\, c^m_{l,i,[1]}, \tag{3.36a}$$

$$\tilde{d}^m_{\{j,l\},[a]} = 2\sum_{i=0}^{k} h^{(0)}_{j,i}\, c^m_{l,i,[2]} + h^{(1)}_{j,i}\, c^m_{l,i,[3]}, \tag{3.36b}$$

$$\tilde{d}^m_{\{j,l\},[b]} = 2\sum_{i=0}^{k} g^{(0)}_{j,i}\, c^m_{l,i,[0]} + g^{(1)}_{j,i}\, c^m_{l,i,[1]}, \tag{3.36c}$$

$$\tilde{d}^m_{\{j,l\},[c]} = 2\sum_{i=0}^{k} g^{(0)}_{j,i}\, c^m_{l,i,[2]} + g^{(1)}_{j,i}\, c^m_{l,i,[3]}. \tag{3.36d}$$

Le nombre de calculs est à présent en $O((k+1)^3)$. C'est cette deuxième méthode qui est la méthode utilisée dans le code.

De la même façon, les équations (3.28) peuvent se récrire

$$\tilde{s}^{m+1}_{\boldsymbol{q},[2]} = \frac{1}{2}\sum_{\boldsymbol{p}=0}^{K} h^{(0)}_{p_1,q_1} h^{(0)}_{p_2,q_2}\, \tilde{s}^m_{\boldsymbol{p}} + h^{(0)}_{p_1,q_1} g^{(0)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[a]} + g^{(0)}_{p_1,q_1} h^{(0)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[b]} + g^{(0)}_{p_1,q_1} g^{(0)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[c]}, \tag{3.37a}$$

$$\tilde{s}^{m+1}_{\boldsymbol{q},[0]} = \frac{1}{2}\sum_{\boldsymbol{p}=0}^{K} h^{(0)}_{p_1,q_1} h^{(1)}_{p_2,q_2}\, \tilde{s}^m_{\boldsymbol{p}} + h^{(0)}_{p_1,q_1} g^{(1)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[a]} + g^{(0)}_{p_1,q_1} h^{(1)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[b]} + g^{(0)}_{p_1,q_1} g^{(1)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[c]}, \tag{3.37b}$$

$$\tilde{s}^{m+1}_{\boldsymbol{q},[3]} = \frac{1}{2}\sum_{\boldsymbol{p}=0}^{K} h^{(1)}_{p_1,q_1} h^{(0)}_{p_2,q_2}\, \tilde{s}^m_{\boldsymbol{p}} + h^{(1)}_{p_1,q_1} g^{(0)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[a]} + g^{(1)}_{p_1,q_1} h^{(0)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[b]} + g^{(1)}_{p_1,q_1} g^{(0)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[c]}, \tag{3.37c}$$

$$\tilde{s}^{m+1}_{\boldsymbol{q},[1]} = \frac{1}{2}\sum_{\boldsymbol{p}=0}^{K} h^{(1)}_{p_1,q_1} h^{(1)}_{p_2,q_2}\, \tilde{s}^m_{\boldsymbol{p}} + h^{(1)}_{p_1,q_1} g^{(1)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[a]} + g^{(1)}_{p_1,q_1} h^{(1)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[b]} + g^{(1)}_{p_1,q_1} g^{(1)}_{p_2,q_2}\, \tilde{d}^m_{\boldsymbol{p},[c]}. \tag{3.37d}$$

De même que précédemment, le calcul peut être décomposé par direction, ce qui donne alors

$$c^{m+1}_{l,i,[0]} = \sum_{j=0}^{k} h^{(0)}_{j,l}\, \tilde{s}^m_{\{i,j\}} + g^{(0)}_{j,l}\, \tilde{d}^m_{\{i,j\},[a]}, \tag{3.38a}$$

$$c^{m+1}_{l,i,[1]} = \sum_{j=0}^{k} h^{(0)}_{j,l}\, \tilde{d}^{m}_{\{i,j\},[b]} + g^{(0)}_{j,l}\, \tilde{d}^{m}_{\{i,j\},[c]}, \tag{3.38b}$$

$$c^{m+1}_{l,i,[2]} = \sum_{j=0}^{k} h^{(1)}_{j,l}\, \tilde{s}^{m}_{\{i,j\}} + g^{(1)}_{j,l}\, \tilde{d}^{m}_{\{i,j\},[a]}, \tag{3.38c}$$

$$c^{m+1}_{l,i,[3]} = \sum_{j=0}^{k} h^{(1)}_{j,l}\, \tilde{d}^{m}_{\{i,j\},[b]} + g^{(1)}_{j,l}\, \tilde{d}^{m}_{\{i,j\},[c]}, \tag{3.38d}$$

puis

$$\tilde{s}^{m+1}_{\{j,l\},[2]} = \frac{1}{2} \sum_{i=0}^{k} h^{(0)}_{i,j}\, c^{m+1}_{l,i,[0]} + g^{(0)}_{i,j}\, c^{m+1}_{l,i,[1]}, \tag{3.39a}$$

$$\tilde{s}^{m+1}_{\{j,l\},[0]} = \frac{1}{2} \sum_{i=0}^{k} h^{(0)}_{i,j}\, c^{m+1}_{l,i,[2]} + g^{(0)}_{i,j}\, c^{m+1}_{l,i,[3]}, \tag{3.39b}$$

$$\tilde{s}^{m+1}_{\{j,l\},[3]} = \frac{1}{2} \sum_{i=0}^{k} h^{(1)}_{i,j}\, c^{m+1}_{l,i,[0]} + g^{(1)}_{i,j}\, c^{m+1}_{l,i,[1]}, \tag{3.39c}$$

$$\tilde{s}^{m+1}_{\{j,l\},[1]} = \frac{1}{2} \sum_{i=0}^{k} h^{(0)}_{i,j}\, c^{m+1}_{l,i,[2]} + g^{(1)}_{i,j}\, c^{m+1}_{l,i,[3]}. \tag{3.39d}$$

3.6 Le seuillage

L'opération de seuillage consiste à regarder l'importance des détails de la distribution afin de décider s'ils sont ou non significatifs et s'ils doivent ou non être conservés pour les calculs suivants. On procède comme expliqué ici. La distribution $f_h(x,v)$ peut se décomposer pour obtenir le niveau $m+1$ selon la formule

$$f_h(x,v) = \sum_{c=0,1,2,3} \sum_{q} \phi^{m+1}_{q,[c]}(x,v)\, s^{m+1}_{\boldsymbol{q},[c]} \tag{3.40}$$

en considérant directement le niveau $m+1$, ou

$$f_h(x,v) = \sum_{\boldsymbol{q}} \phi^{m}_{\boldsymbol{q}}(x,v)\, s^{m}_{\boldsymbol{q}} + \sum_{\Theta=a,b,c} \sum_{\boldsymbol{k}} \psi^{m}_{\boldsymbol{k},[\Theta]}(x,v)\, d^{m}_{\boldsymbol{k},[\Theta]} \tag{3.41}$$

en considérant le niveau m et les détails pour passer du niveau m au niveau $m+1$. Les formules (3.40) et (3.41) sont bien sûr équivalentes.

Le seuillage consiste à comparer la norme L^2 des détails $d^{m}_{\cdot,[\cdot]}$ à un seuil ϵ_0 donné et à les ignorer quand ils sont plus petits que ce seuil. Ceci peut s'écrire

$$\text{si } \left(\sum_{\Theta=a,b,c} \sum_{\boldsymbol{k}} \left(d^{m}_{\boldsymbol{k},[\Theta]} \right)^2 \right)^{1/2} \leq \epsilon_0, \text{ alors on ignore les détails de niveau } m. \tag{3.42}$$

Cette opération est répétée jusqu'à ce que tous les détails conservés soient significatifs en partant des niveaux de raffinement les plus élevés et en allant vers les niveaux de raffinement les plus grossiers. Le paramètre de seuillage ϵ_0 est présenté dans le cadre de

l'utilisation d'une norme L^p dans [12]. L'utilisation de la norme L^2 rend ce paramètre indépendant du niveau considéré. La valeur du paramètre ϵ_0 et le niveau maximum de raffinement sont choisis empiriquement pour permettre de suivre des détails fins et importants au cours de la simulation tout en ayant des temps de calcul raisonnables.

Dans le cas d'un problème à une dimension, pour des polynômes de degré inférieur ou égal à k, l'erreur commise est d'ordre $k+1$ et est portée par la dérivée k+1-ème de la fonction. Le niveau de raffinement local estimé par

$$m = \mathrm{E}\left(\frac{K \left\| f^{(k+1)} \right\|_{L^2}}{\epsilon_0^{\frac{1}{k+1}}} \right), \tag{3.43}$$

où m est le niveau local, ϵ_0 est le seuil, K est une constante indépendante de la fonction f et du seuil, et E indique la partie entière.

On utilise uniquement la norme L^2 dans le code pour l'opération de seuillage (3.42). Néanmoins, il est possible d'utiliser d'autres normes, et notamment la norme L^∞. Les équations (3.5) et (3.6) sont étendues en dimension deux. On a alors pour les équations en dimension d

$$\phi^m_{\boldsymbol{k},\boldsymbol{l}} = 2^{md/2} \phi_{\boldsymbol{k}}(2^n x_1 - l_1, \ldots, 2^n x_d - l_d). \tag{3.44}$$

Utilisée avec la norme L^∞, la relation (3.42) devient alors

$$\text{si } \max_{\Theta=a,b,c \ ; \ \boldsymbol{k}} d^m_{\boldsymbol{k},[\Theta]} \leq \frac{\epsilon_0}{2^m}, \text{ alors on ignore les détails de niveau } m. \tag{3.45}$$

Ces résultats peuvent aussi être retrouvés dans les analyses de Cohen *et al.* [19] ou de De Vore et Yu [37].

Chapitre 4

Programmation des méthodes

4.1 Le maillage

Cette section indique le vocabulaire utilisé pour parler du maillage, de son stockage et des opérations qui lui sont appliquées, ainsi que de sa structure au sein du code.

Quand le niveau est élevé, le maillage est fin et les détails sont conservés. Quand le niveau est faible, le maillage est grossier et les détails sont perdus. Le seuillage est expliqué dans la section 3.6. On raffine le maillage quand on augmente le niveau, on déraffine le maillage quand on diminue le niveau. Bien sûr le niveau est local et deux points du domaine numérique, proches ou éloignés, peuvent avoir des niveaux différents. On voit grâce aux équations (2.51), (2.53), (2.63), (3.34) et (3.37) que le maillage n'a pas besoin d'être gradé, c'est-à-dire que deux cellules adjacentes peuvent avoir plus d'un niveau d'écart. (Un maillage gradé est un maillage dans lequel deux cellules adjacentes ont au plus un niveau de raffinement d'écart.)

Le maillage est stocké en mémoire sous forme d'arbre. Chaque cellule de niveau $m > 0$ a une cellule mère de niveau $m - 1$ et peut avoir deux cellules filles par direction (donc quatre cellules filles en dimension 2) de niveau $m + 1$. Si une cellule n'a pas de cellule fille elle est appelée feuille. Un exemple d'arbre allant du niveau 0 au niveau 3 est représenté figure 4.1. L'encart « Code 1 » résume la structure de donnée et son contenu. (Le code est rédigé en langage c.)

Initialement le maillage disposait en plus d'une structure de graphe, c'est à dire que toutes les cellules possédaient des pointeurs vers leurs voisines, mais il s'est avéré à l'usage que parcourir l'arbre était beaucoup plus rapide que chercher la cellule voisine correspondante dans la liste de cellules voisines, même en rangeant les cellules voisines. Par conséquent cette structure a été abandonnée.

L'ensemble du domaine de simulation représente la cellule de niveau 0. Chaque niveau m comporte au maximum 4^m cellules. Cela signifie que par la suite, lorsqu'il sera fait mention du niveau maximum de raffinement M, le maillage le plus fin pourra comporter au plus 4^M cellules (2^M par direction). En pratique il est souhaitable que ce raffinement soit local. De plus, pour connaître le nombre de degrés de liberté, il faut prendre en compte le degré maximum des polynômes utilisés. Dans le cas où l'on utilise des polynômes de

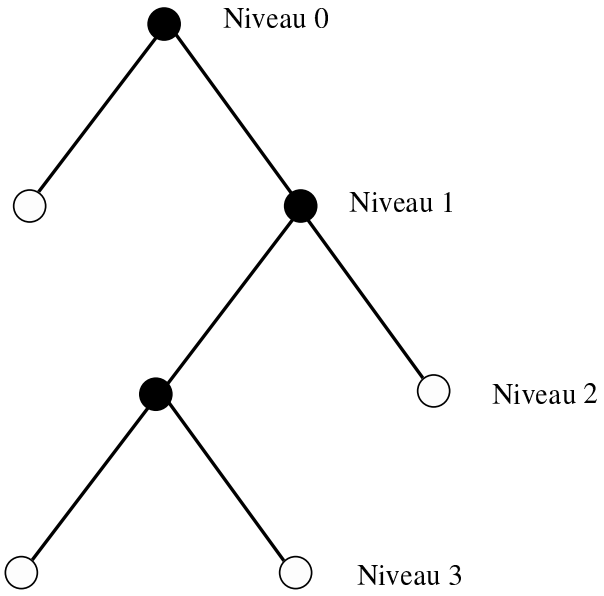


FIGURE 4.1 – Structure d'arbre du maillage 1D. Les ronds (pleins et creux) indiquent les cellules. Un rond creux indique que la cellule est une feuille de l'arbre.

degré $N-1$ par direction, cela donne N^2 degrés de liberté par cellule. Le nombre maximum de degrés de liberté est alors $4^M \times N^2$. Dans la pratique, beaucoup de mes simulations sont faites avec des polynômes de degré 2 et 8 niveaux de raffinement. Cela signifie que le nombre maximum de degrés de liberté par direction est $2^8 \times 3 = 768$, soit 589 824 degrés de libertés au niveau le plus fin.

4.2 Résolution de l'équation de Poisson

La résolution de l'équation de Poisson (2.5), que l'on rappelle ici

$$-\frac{\partial^2}{\partial x^2}\Phi(x,t) = \rho(x,t) - 1, \quad x \in \Omega_x, \tag{4.1a}$$

$$E(x,t) = -\partial_x \Phi(x,t) \tag{4.1b}$$

se fait en plusieurs étapes. La première étape consiste à calculer la densité $\rho(x)$. (Le temps n'intervient pas dans les calculs présentés ici, il est donc omis des écritures afin de les simplifier.) Parce que le coût numérique du calcul du champ est faible devant le coût numérique de la distribution dans l'espace des phases, la densité et le champ électrique ne sont considérés que sur le maillage le plus fin, qui, par conséquent, est uniforme. Afin de rester cohérent avec le choix d'utiliser les polynômes de Legendre et parce que ces polynômes sont faciles à intégrer, on souhaite décomposer $\rho(x)$ sur la base des polynômes de Legendre rapportée à chaque cellule, c'est à dire sous la forme

$$\rho(x) = \sum_{I_i} \sum_{k} \rho_{i,k} p_{i,k}(x), \tag{4.2}$$

Code 1 Structure du maillage

```
struct st_Mesh
{
  unsigned char m_lvl; // niveau de la cellule
  st_Mesh *m_parent; // pointeur vers la cellule mère. NULL si m_lvl==0
  st_Mesh *m_children[4]; // tableau de pointeurs vers les enfants
  char m_leaf; // 1 si la cellule est une feuille, 0 sinon

  double m_boundaries[2][2]; // extrémités de la cellule dans les deux
                             // directions
  double m_centre[2]; // centre de la cellule dans les deux directions
  double m_length[2]; // longueur de la cellule dans chaque direction

  double *m_dist, *m_distnp1; // tableaux de coefficients s_k^m à t^n et
                              // t^{n+1}
};
```

où $p_{i,k}(x)$ est le k-ième polynôme de Legendre orthonormé en norme L^2 et rapporté à la cellule I_i. Pour aboutir à ce résultat, pour chacune des cellules I_i, on calcule

$$\widehat{\rho}_{i_g,i} = \int_{\Omega_v} f_h(x_{i_g,i}, v)\, dv, \tag{4.3}$$

avec $\widehat{\rho}_{i_g,i} = \rho(x_{i_g,i})$ et $\{x_{i_g,i}\}$ les points de Gauss sur la cellule I_i. En utilisant ensuite l'équation (2.19) on obtient

$$\rho_{i,k} = \sum_{i_g} \widehat{\rho}_{i_g,i} p_{i,k}(x_{i_g,i}). \tag{4.4}$$

L'intégrale sur Ω_v se calcule en décomposant sur chacune des cellules $T_{i,j}$

$$\begin{aligned}\int_{\Omega_v} f_h(x_{i_g,i}, v)\, dv &= \sum_{i,j} \int_{T_{i,j}} f_h(x_{i_g,i}, v)\, dv \\ &= \sum_{i,j} \int_{T_{i,j}|_{J_j}} \sum_{\boldsymbol{k}} f_{\boldsymbol{k},T_{i,j}} P_{\boldsymbol{k},T_{i,j}}(x_{i_g,i}, v)\, dv \\ &= \sum_{i,j} \sum_{\boldsymbol{k}} f_{\boldsymbol{k},T_{i,j}} \int_{J_j} P_{\boldsymbol{k},T_{i,j}}(x_{i_g,i}, v)\, dv.\end{aligned} \tag{4.5}$$

Le calcul est alors immédiat. Comme énoncé précédemment à la section 2.2.3, on résout alors

$$\partial_x E(x) = \rho(x) - 1. \tag{4.6}$$

La structure polynomiale par morceaux du modèle de décomposition adopté est là aussi mise à profit. On commence par résoudre (4.6) sur chacune des cellules I_i. On a alors

$$\partial_x \left(\sum_{k=0}^{n+1} e_{ik} x^k \right) = \sum_{k=0}^{n} \rho_{ik} x^k, \tag{4.7}$$

où e_{ik} et ρ_{ik} désignent les coefficients des monômes de degré k de $E(x)$ et $\rho(x)$, et n est le degré de la reconstruction polynomiale. L'intégration de (4.7) conduit alors à

$$e_{ik} = \rho_{ik-1}/k, \quad \text{pour } k = 1 \text{ à } n+1. \tag{4.8}$$

On remarque qu'à ce stade, $E(x)$ est exprimé sur une base de degré $n+1$ alors qu'on désire son expression sur une base de degré n, que e_{m0} n'est pas fixé, et que la moyenne $E(x)$ n'est pour l'instant pas nulle. On commence par fixer pour chacune des cellules I_i à l'exception de la première les coefficients e_{m0} tels que $E(x)$ soit continu. Le champ est alors défini à une constante près. L'ensemble des coefficients est alors réajusté pour imposer

$$\int_{\Omega_x} E(x)\,dx = 0, \tag{4.9}$$

c'est à dire pour imposer $E(x)$ à moyenne nulle. On applique finalement (2.19) afin de projeter $E(x)$ sur la base des polynômes de Legendre de degré n.

4.3 Calcul de l'origine des caractéristiques

4.3.1 Calculs de bases

L'équation de Vlasov (2.9),

$$\partial_t f(x,v,t) + \partial_x(v f(x,v,t)) + \partial_v(E(x,t) f(x,v,t)) = 0, \quad (x,v) \in \Omega_x \times \mathbb{R}. \tag{4.10}$$

est résolue par un splitting directionnel. Cela signifie qu'on résout indépendamment

$$(\partial_t f(x,v,t) + \partial_x(v f(x,v,t)))\big|_v = 0 \tag{4.11}$$

à v fixé et

$$(\partial_t f(x,v,t) + \partial_v(E(x,t) f(x,v,t)))\big|_x = 0 \tag{4.12}$$

à x fixé. L'algorithme final et les étapes du splitting seront détaillés ultérieurement dans la section 4.5. On ne s'attache ici qu'au calcul des caractéristiques et aucunement aux détails du splitting. On note $X(t)$ les courbes caractéristiques telles que

$$\frac{d}{dt} X(t) = v. \tag{4.13}$$

L'équation (4.11) nous permet alors d'écrire

$$\frac{d}{dt} f(X(t),v,t)\bigg|_v = 0. \tag{4.14}$$

Dans les schémas GDSL et CGD, on connaît les points x_i, points d'arrivée des caractéristiques et on cherche l'origine associée. L'équation (4.13) nous permet de déterminer directement le pied des caractéristiques

$$x_i^* = x_i - v\,dt, \tag{4.15}$$

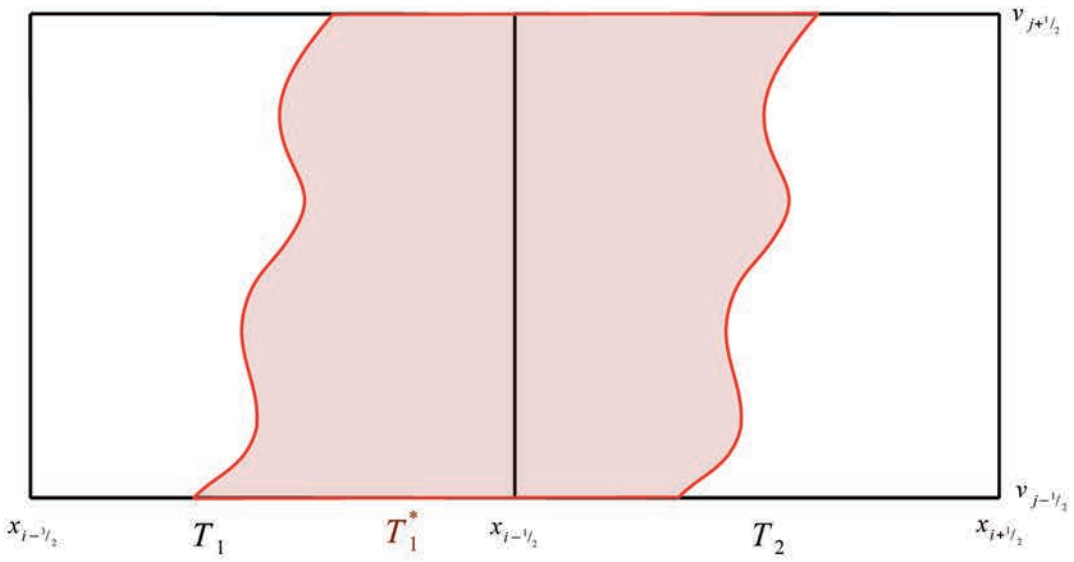


FIGURE 4.2 – Transport 1D d'une cellule 2D.

avec x_i^* le pied de la caractéristique au temps t^n et passant par x_i au temps t^{n+1}. Le schéma CGD nécessite en plus de calculer les points d'arrivée des caractéristiques en ne connaissant que leur origine $x^*_{i_g,l,j_g}$. En inversant l'équation (4.15) on trouve immédiatement

$$x_{i_g,l,j_g} = x^*_{i_g,l,j_g} + v_{j_g} dt. \tag{4.16}$$

En procédant de même dans la direction v, en remplaçant la vitesse v par le champ électrique $E(x,t)$, on obtient finalement

$$v_j^* = v_j - E(x,t)dt, \tag{4.17}$$

$$v_{j_g,l,i_g} = v^*_{j_g,l,i_g} + E(x_{i_g},t)dt. \tag{4.18}$$

On notera que dans ce cas, le champs $E(x,t)$ n'étant connu que de façon discrète en temps, il est pris à un temps donné selon le principe du splitting en temps. C'est ce qui ressort des étapes 3 et 4 de l'algorithme final présenté à la section 4.5.

4.3.2 Application à la méthode CGD

On s'attache ici à un problème soulevé par la résolution des intégrales par la formule de quadrature de Gauss. Ce problème apparaît dans les deux méthodes (GDSL et CGD) mais il est plus facile à illustrer dans le cadre de la méthode CGD. Cette difficulté existe sur maillage uniforme et est accrue par l'utilisation d'un maillage non-uniforme. Néanmoins, l'étude sur maillage uniforme suffit pour en comprendre et appréhender les principaux tenants.

On se place dans le cas particulier de la méthode CGD sur maillage uniforme pour l'équation

$$\frac{\partial f}{\partial t} + g(v)\frac{\partial f}{\partial x} = 0, \tag{4.19}$$

où $g(v)$ est un polynôme de degré inférieur ou égal au degré des polynômes utilisés dans la méthode Galerkin discontinue.

Sur la figure 4.2 sont représentés deux domaines, T_1 et T_2, délimités par des bordures noires, et le domaine T_1^* qui est le domaine obtenu en transportant T_1 conformément à ce qui est expliqué dans la section 2.4.1. L'étude de $T_1^* \cap T_1$ est équivalente à l'étude de $T_1^* \cap T_2$, on se contente donc ici de discuter de $T_1^* \cap T_2$, et notamment de la validité de l'équation (2.65) sur ce domaine.

On pose

$$T_1^* \cap T_2 = [x_{i-1/2} \ ; \ \mathcal{P}(v)] \times [v_{j-1/2} \ ; \ v_{j+1/2}], \tag{4.20}$$

avec

$$\mathcal{P}(v) = x_{i-1/2} + g(v)dt. \tag{4.21}$$

La méthode CGD expliquée section 2.4.1 nécessite de résoudre

$$\int_{T_1^* \cap T_2} f_h^n(x^*, v) P_{\boldsymbol{k}}(x(x^*), v)\, dx^*\, dv = \int_{v_{j-1/2}}^{v_{j+1/2}} \left(\int_{x_{i-1/2}}^{\mathcal{P}(v)} f_h^n(x^*, v) P_{\boldsymbol{k}}(x(x^*), v)\, dx^* \right) dv. \tag{4.22}$$

On pose afin de simplifier les notations

$$\mathcal{F}(x^*, v) = f_h^n(x^*, v) P_{\boldsymbol{k}}(x(x^*), v). \tag{4.23}$$

On effectue le changement de variable

$$x = \frac{a - \mathcal{P}(v)}{a - b}\overline{x} + a\frac{\mathcal{P}(v) - b}{a - b} \tag{4.24}$$

pour se ramener à une intégrale sur le rectangle $(\overline{x}, v) \in [a \ ; \ b] \times [v_{j-1/2} \ ; \ v_{j+1/2}]$. L'équation (4.20) devient alors

$$\int_{T_1^* \cap T_2} \mathcal{F}(x, v)\, dx dv = \int_{v_1}^{v_2} \int_a^b \mathcal{F}\left(\frac{a - \mathcal{P}(v)}{a - b}\overline{x} + a\frac{\mathcal{P}(v) - b}{a - b}, v \right) \frac{a - \mathcal{P}(v)}{a - b}\, d\overline{x}\, dv. \tag{4.25}$$

On constate alors que le polynôme intégré dans le membre de droite est de degré $2n+1$ dans la direction d'intégration x et de degré $3n$ dans la direction orthogonale v. L'utilisation de n points de Gauss dans les formules de quadrature fait donc que l'intégration est toujours exacte dans la direction d'intégration mais n'est plus exacte dans la direction orthogonale.

4.4 Optimisations

L'algorithme initialement écrit suivait au plus près les équations (2.51), (2.53) et (2.65), (2.66). Néanmoins, ceci n'était absolument pas efficace, de nombreux calculs étaient faits plusieurs fois et les temps de calculs généraux étaient extrêmement longs (entre 2 et 10 fois le temps de calcul espéré), en particulier pour la méthode GDSL. Ce chapitre présente donc les optimisations que j'ai pu mettre en œuvre.

L'équation (2.51) nécessite de calculer l'intégrale de la distribution entre des points toujours différents. L'équation (2.52) permet de décomposer ces intégrales en une somme d'intégrales sur de multiples intervalles (en supposant $a < b$)

$$\int_a^b f(x)\,dx = \sum_i \int_{[a\ ;b]\cap I_i} f(x)\,dx. \tag{4.26}$$

On note x_a tel que $a \in [x_{a_{1/2}}\ ;\ x_{a+1/2}]$ et x_b tel que $b \in [x_{b-1/2}\ ;\ x_{b+1/2}]$. Dans ces conditions, l'équation (4.26) peut se réécrire sous la forme

$$\begin{aligned}\int_a^b f(x)\,dx = &\int_a^{x_{a+1/2}} f(x)\,dx + \\ &\sum_{i/x_a<x_i<x_b} \int_{x_{i-1/2}}^{x_{i+1/2}} f(x)\,dx + \\ &\int_{x_{b-1/2}}^{b} f(x)\,dx.\end{aligned} \tag{4.27}$$

Le premier et le dernier terme doivent être calculés pour chaque intervalle $[a\ ;\ b]$ mais le calcul des termes intermédiaires peut être simplifié. En effet, on a localement (sur chaque cellule)

$$f(x) = \sum_{k=0}^{n} f_k p_k(x), \tag{4.28}$$

où les p_k sont les polynômes de Legendre. Par conséquent, grâce aux propriétés d'orthogonalité des polynômes

$$\int_{-1}^{1} \tilde{p}_k(x)\,dx = \delta_{k,0} \tag{4.29}$$

où $\delta_{k,0}$ vaut 1 pour $k = 0$ et 0 sinon, on simplifie le calcul de la façon suivante :

$$\int_{I_i} f(x)\,dx = \frac{\Delta x_i}{2} \tilde{f}_0 \int_{-1}^{1} \tilde{p}_0(x)\,dx. \tag{4.30}$$

Comme j'utilise des polynômes orthonormés pour la norme L^2 (et pas simplement orthogonaux), le calcul devient

$$\int_{I_i} f(x)\,dx = \tilde{f}_0 \times (x_{i+1/2} - x_{i-1/2}) \times \frac{\sqrt{2}}{2}. \tag{4.31}$$

Dans la pratique, le problème se pose avec des polynômes à deux dimensions et cette méthode perd en efficacité. Une autre possibilité est de calculer initialement les primitives $\mathcal{P}_k$ des polynômes de Legendre p_k. On peut alors écrire

$$\int_{I_i} f(x)\,dx = \sum_{k=0}^{n} f_k \times \left(\mathcal{P}_k(x_{i+1/2}) - \mathcal{P}_k(x_{i-1/2})\right). \tag{4.32}$$

En deux dimensions l'équation devient

$$\begin{aligned}\int_{I_i} f(x,v)|_v\,dx = &\sum_{\boldsymbol{k}} f_{\boldsymbol{k}} p_{k_2}(v)|_v \left(\mathcal{P}_{k_1}(x_{i+1/2}) - \mathcal{P}_{k_1}(x_{i-1/2})\right), \\ = &\frac{\Delta x_i}{2} \sum_{\boldsymbol{k}} \tilde{f}_{\boldsymbol{k}} \tilde{p}_{k_2}(\tilde{v}) \left(\tilde{\mathcal{P}}_{k_1}(1) - \tilde{\mathcal{P}}_{k_2}(-1)\right).\end{aligned} \tag{4.33}$$

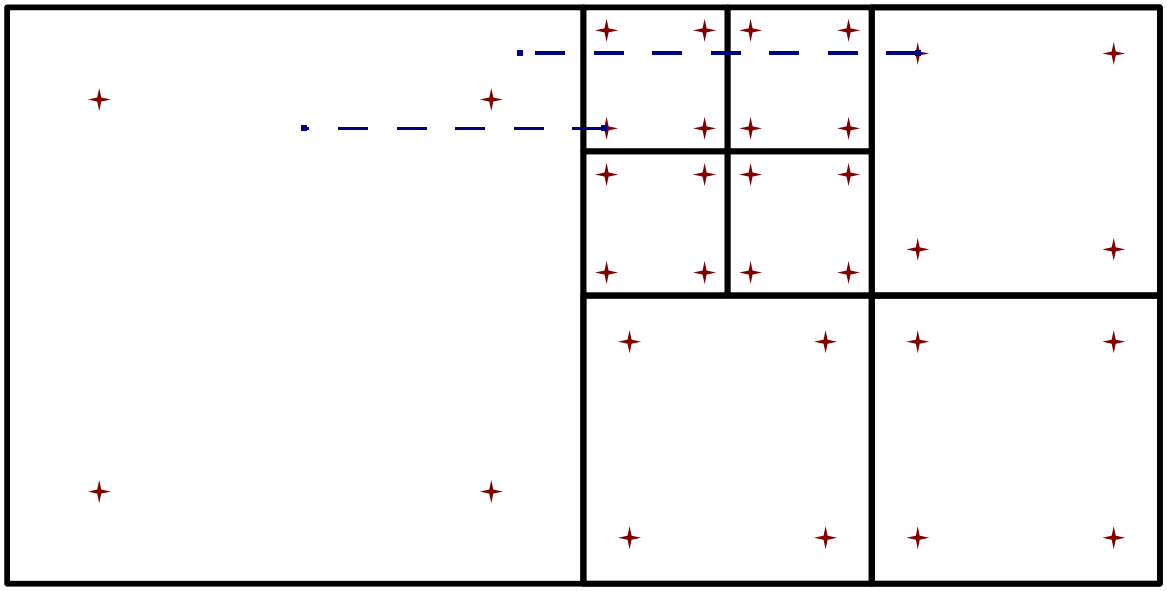
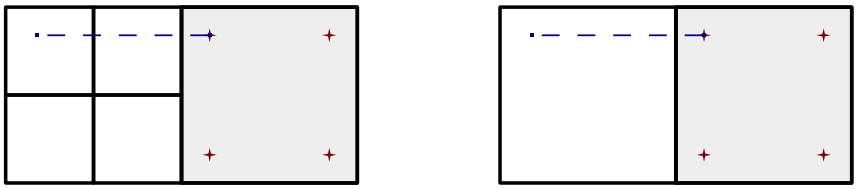


FIGURE 4.3 – Exemple d'interaction entre les différents niveaux. En noir les bords des cellules, en rouge les points de Gauss sur les cellules, en bleu quelques trajectoires d'intégration.

v étant fixé, les $\tilde{p}_{k_2}(\tilde{v})$ peuvent également être calculés avant la somme afin d'optimiser encore le calcul. La démarche se fait sur le même principe lorsqu'on inverse le rôle des directions x et v.
(On rappelle qu'un tilde au dessus d'une variable indique qu'elle est exprimée sur l'intervalle de référence $[-1\ ;\ 1]$ et que la décomposition de la distribution en coefficients sur la base des polynômes de Legendre se fait sur cet intervalle de référence. En tout temps, on connaît donc les $\{\tilde{f}_{\boldsymbol{k}}\}_{\boldsymbol{k}}$ uniquement.)

Sachant que dans notre cas, la direction orthogonale fixe (v dans le cas ci-dessus) correspond toujours à des points de Gauss sur une cellule à un niveau donné de raffinement, il serait possible de calculer tous les $p_{k_2}(v)$ possibles au début du programme pour aller ensuite chercher la valeur voulue à chaque fois qu'elle est nécessaire au calcul. En pratique, le niveau le plus grossier de représentation est souvent le niveau 2 et le niveau le plus fin de représentation est souvent fixé à 8. La figure 4.3 présente quelques interactions entre les différents niveaux par le biais des intégrales à calculer. On y voit que pour avoir l'ensemble des pré-calculs utiles il faudrait calculer les valeurs des p_k sur les points de Gauss correspondant à tous les niveaux possibles. Bien que cela soit possible, il est plus simple de calculer les valeurs des polynômes sur une discrétisation fine pour approcher la valeur de $p_k(x)$. En pratique, cette méthode est inefficace. Le temps passé à chercher le point d'approximation le plus proche du point voulu est supérieur au temps passé à résoudre l'équation (4.33).



(a) Maillage réel. (b) Maillage vu par la cellule grise.

FIGURE 4.4 – Illustration de l'utilisation des différents niveaux du maillage. En noir, les bords des cellules, en rouge, les points de Gauss sur la cellule grise, en bleu, une trajectoire d'intégration.

Une autre optimisation utilisée et validée par l'expérience consiste à limiter le niveau des cellules utilisées. Si le résultat doit être appliqué à une cellule de niveau l, on utilise des données de niveau l où elles sont disponibles et des données de niveau inférieur lorsque le niveau l n'est pas disponible. À la fin de chaque étape du splitting la distribution est calculée à tous les niveaux grossiers existants et ce sont ces valeurs qui sont utilisées. Ceci permet d'utiliser moins de données donc d'optimiser le calcul. En pratique, les résultats obtenus sont identiques. Ce processus est illustré par la figure 4.4.

Enfin, on n'aura pas manqué de remarquer dans les sections 2.3 et 2.4 que de nombreux calculs se font aux points de Gauss, au moins dans une direction. La méthode GDSL utilise en plus les valeurs au bord des cellules. Tous les polynômes sont donc évalués aux points de Gauss, en -1 et en 1 lors de l'initialisation du programme. Dans le cas de la méthode GDSL, les dérivées des polynômes de Legendre aux points de Gauss sont également calculées et stockées lors de l'initialisation.

4.5 Algorithme final

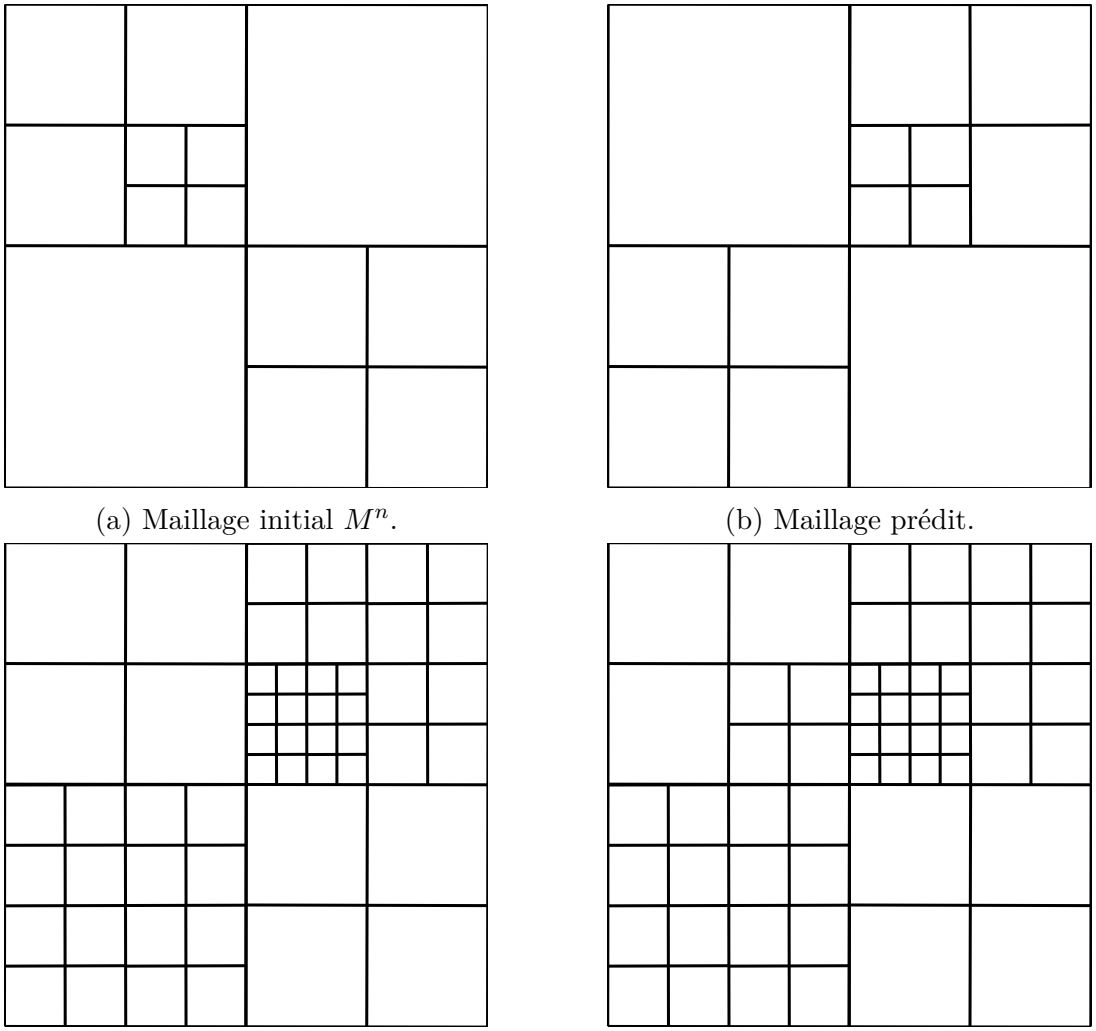
L'algorithme finalement implémenté est le suivant.

Étape 1 : initialisation. On projette la distribution f sur le niveau le plus grossier du maillage (habituellement le niveau 0) et on calcule les détails au même niveau par les formules (3.22) et (3.23). On regarde ensuite si les détails sont significatifs comme indiqué par l'équation (3.42). Tant que les détails sont significatifs, la cellule est raffinée et l'opération est répétée sur les cellules filles.

Étape 2 : première étape du splitting de Strang. La première étape du splitting de Strang est un demi pas de temps dans la direction x de l'espace des phases uniquement (de t^n à $t^n + dt/2$). On résout

$$\partial_t f + \partial_x(vf) = 0. \tag{4.34}$$

Chaque étape du splitting de Strang est divisée en trois phases.



(a) Maillage initial M^n.

(b) Maillage prédit.

(c) Maillage prédit après raffinement.

(d) Maillage final M^{n+1}.

FIGURE 4.5 – Étapes de la prédiction du maillage.

Prédiction. À partir du maillage M^n connu au temps t^n, on prédit le maillage M^{n*} au temps $t^n + dt/2$ à l'aide d'un schéma d'Euler explicite pour calculer les caractéristiques vers l'avant et en s'assurant que leurs points d'arrivée est au moins un niveau plus raffiné que le point de départ. Ce processus est illustré par la figure 4.5. Dans un premier temps, le maillage initial M^n (figure 4.5a) est transporté pour obtenir le maillage de la figure 4.5b. Le maillage de la figure 4.5c est obtenu en raffinant d'un niveau le maillage de la figure 4.5c pour anticiper la création de fines structures. Ce modèle de prédiction est tiré de [12]. Tandis que Besse *et al.* utilisent un critère pour savoir si ce raffinement supplémentaire est nécessaire, l'ajout d'un niveau de détail est ici automatique. Les discussions sur la précision et l'intérêt de ce critère sont laissées à [12]. Le maillage M^{n+1} de la figure 4.5d est obtenu en additionnant le maillage initial M^n (figure 4.5a) au maillage prédit et raffiné de la figure 4.5c.

La prédiction du maillage se fait en utilisant une structure indépendante de la structure utilisée lors de la phase de calcul. Cette structure de prédiction est un arbre similaire à ce qui est présenté dans l'extrait de code 1 mais ne contenant que la géométrie et aucune donnée de calcul. Cela signifie que l'étape de prédiction nécessite de manipuler deux structures d'arbre. La structure utilisée aux figures 4.5a et 4.5d contient les données utiles à la reconstruction de la distribution (ainsi que la géométrie des cellules décrites), tandis que la structure utilisée pour les figures 4.5b et 4.5c ne contient que la géométrie. Par conséquent, le maillage M^{n+1} (utilisé pour le calcul) doit être au moins aussi fin localement que le maillage initial M^n, sans quoi des détails de la distribution f_h^n non négligeables seraient perdues. La fonction de distribution f_h^n connue sur le maillage M^n (figure 4.5a) est projetée sur la maillage M^{n+1} (figure 4.5d) à l'aide des équations (3.38) et (3.39). On considère dans ce cas que les détails inconnus des niveaux plus fins sont nuls.

Calcul. Le schéma numérique choisi (GDSL ou CGD) est appliqué sur le maillage final pour avancer la distribution. On résout (4.34) entre t^n et $t^n + dt/2$ comme présenté aux sections 2.3.2 et 2.4.2.

Déraffinement. Les équation (3.35) et (3.36) sont appliquées récursivement à toutes les cellules afin de calculer la distribution à tous les niveaux. Dans le même temps, les détails sont comparés au seuil ϵ_0 afin de supprimer ceux qui ne sont pas significatifs en utilisant l'équation (3.42).

Étape 3 : résolution de l'équation de Poisson. On résout l'équation de Poisson (2.5) à partir de la distribution obtenue à l'issue de l'étape 2. Le champ est toujours calculé au niveau de raffinement le plus fin car son coût numérique est très faible devant le coût numérique de l'avancée de la distribution.

Étape 4 : deuxième étape du splitting de Strang. La deuxième étape du splitting de Strang est un pas de temps complet dans la direction v uniquement de l'espace des phases (de t^n à t^{n+1}). On résout

$$\partial_t f + \partial_v(Ef) = 0. \tag{4.35}$$

Cette étape suit le même cheminement que l'étape 2 et utilise les résultats des étapes 2 et 3.

Étape 5 : troisième étape du splitting de Strang. La troisième est dernière étape du splitting de Strang est identique à l'étape 2 (première étape du splitting de Strang) si ce n'est qu'elle utilise les données de l'étape 4 (deuxième étape du splitting de Strang) et que son résultat est la fonction de distribution au temps t^{n+1}.

Chapitre 5

Validation du code

Ce chapitre s'attache à valider les différentes parties du code. On y trouvera donc la validation du solveur de Poisson, des cas d'advection et de rotation avec les méthodes GDSL et CGD, et des décompositions-reconstructions à l'aide des multi-ondelettes.

5.1 Validation de la résolution de l'équation de Poisson

On s'attache ici à valider le code permettant de résoudre

$$\partial_x E_x(x,t) = \rho(x,t) - 1, \quad t \text{ fixé}, \tag{5.1}$$

avec des conditions aux limites périodiques.

On a par construction

$$\rho(x,t) = \sum_i \sum_{k=0}^{n} \rho_{I_i,k}(t) p_k(x) \tag{5.2}$$

où I_i désigne les cellules feuilles. Par simplicité, parce que le coût numérique du maillage en une dimension est faible devant celui du maillage en plusieurs dimensions, les champs sont toujours définis uniformément au niveau le plus fin utilisé dans la description de la distribution $f(x,v,t)$. Dans la suite de cette section on fera abstraction du temps pour simplifier la communication. Le temps est considéré fixe.

Les polynômes $p_k(x)$ étant les polynômes de Legendre, il est possible de les intégrer directement par intégration des monômes

$$\int k\, x^{k-1}\, dx = x^k. \tag{5.3}$$

Le champ E est alors exprimé exactement sur chaque cellule I_i sous la forme

$$\widehat{E}(x)|_{I_i} = \sum_{k=0}^{n+1} \widehat{e}_{i,k} x^k. \tag{5.4}$$

Ce champ pourrait être projeté et utilisé sur une base orthonormée de polynômes de degré $n+1$ mais il m'a paru plus logique de le projeter sur la base de polynômes de degré n.

Ce choix s'est fait en considérant d'une part le fait que la distribution est exprimée sur une base de degré n, et d'autre part le fait que pour des modèles différents (dans le cas de Vlasov-Maxwell relativiste par exemple) ou des dimensions plus élevées le champ ne pourrait pas être calculé sur une base de degré $n+1$ mais seulement sur une base de degré n. La projection se fait de façon similaire à la projection (2.19), ce qui permet d'obtenir

$$E_h(x)|_{I_i} = \sum_{n=0}^{n} e_{i,k} p_k(x). \tag{5.5}$$

Dans ces conditions, l'erreur L^2 entre le champ E analytique et le champ E_h numérique doit être d'ordre n. L'intégration exacte est d'ordre $n+1$ (ce qui correspond à l'ordre d'erreur pour f_h et ρ) mais la projection fait perdre un ordre de précision.

Pour ce test, on considère le domaine $[0\ ;\ 2\pi] \times [-8\ ;\ 8]$ et la distribution

$$f(x,v) = \frac{1+\cos(x)}{\sqrt{2\pi}} \exp\left(-\frac{v^2}{2}\right). \tag{5.6}$$

La densité est alors

$$\rho(x) = 1 + \cos(x), \tag{5.7}$$

et le champ électrique

$$E(x) = \sin(x). \tag{5.8}$$

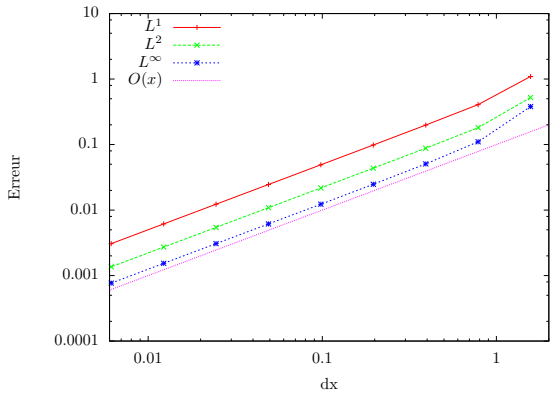
La figure 5.1 montre pour les degrés 1 à 6 la convergence de l'erreur sur le champ E à l'ordre n pour des polynômes de degré n, conformément à ce qui était attendu. L'erreur sur le champ E sature à 10^{-8}. L'erreur sur la densité ρ et sur la distribution f (non représentées ici) saturent à 10^{-10}. Cette erreur sur le calcul de la distribution numérique est le résultat de l'accumulation de toutes les erreurs numériques lors de la projection (2.19).

5.2 Validation des multi-ondelettes

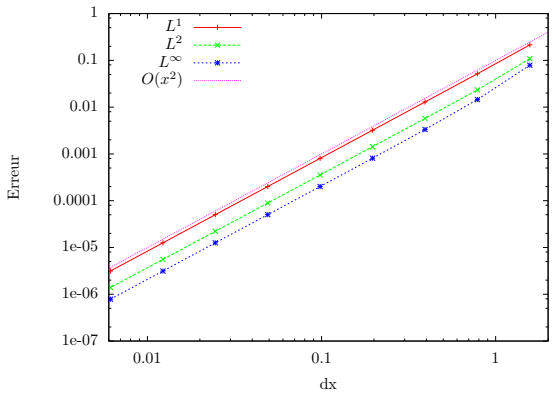
La validation des multi-ondelettes se fait en deux étapes. La première étape consiste à contrôler la construction des multi-ondelettes et des coefficients associés. La deuxième étape consiste à valider les processus de décomposition et de reconstruction.

Pour valider la construction des multi-ondelettes on compare les multi-ondelettes obtenues avec celles présentées dans la partie 3 de [6]. Comme la méthode de construction est la même, les polynômes par morceaux doivent être les mêmes, ce qui est bien le cas. Les multi-ondelettes sont tracées et détaillées dans l'Annexe B

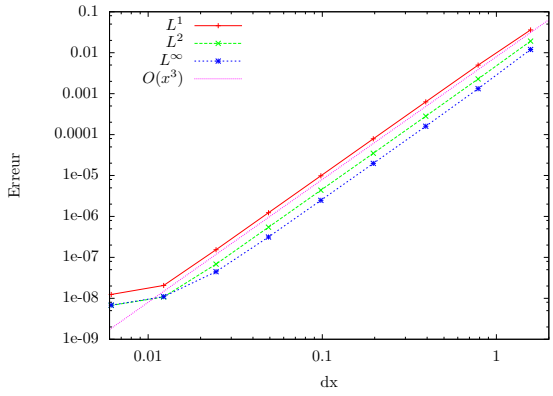
Afin de valider l'implémentation des multi-ondelettes on construit une distribution test sur un maillage grossier. Ce maillage est ensuite raffiné puis ramené au niveau grossier. La distribution finale doit alors être égale à la distribution initiale avec une précision de l'ordre de la précision de la machine. En effet, lors du raffinement les détails sont nuls, l'information ne peut donc pas être perdue lors du déraffinement.



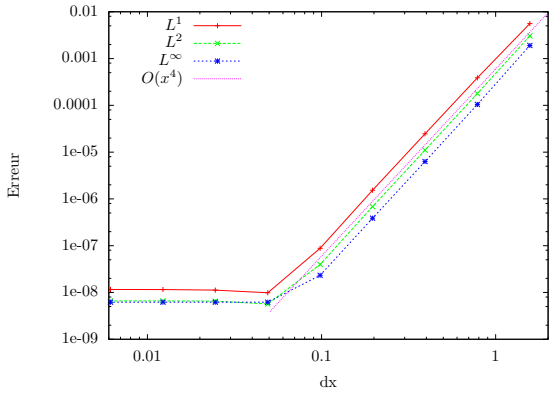
(a) Avec des polynômes de degré 1.



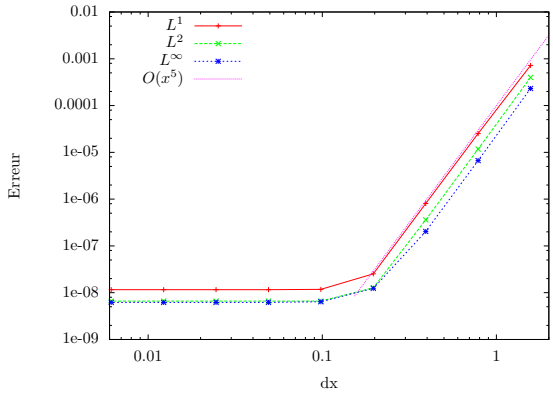
(b) Avec des polynômes de degré 2.



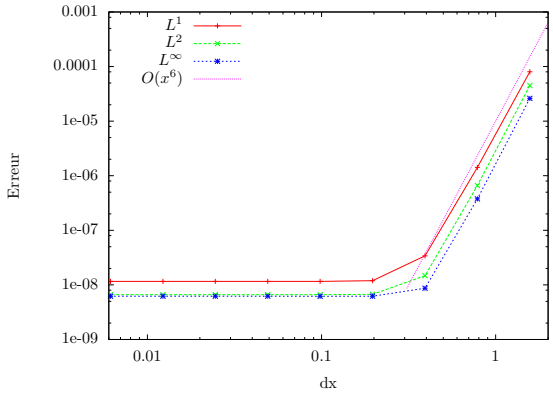
(c) Avec des polynômes de degré 3.



(d) Avec des polynômes de degré 4.

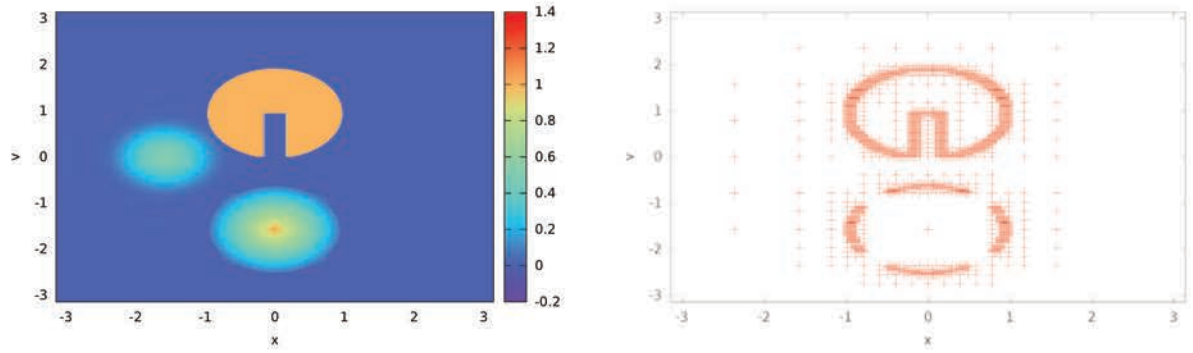


(e) Avec des polynômes de degré 5.



(f) Avec des polynômes de degré 6.

FIGURE 5.1 – Erreur sur E.



(a) Distribution initiale. (b) Maillage initial.

FIGURE 5.2 – Distribution et maillage initiaux pour le cas test de rotation déformation.

On utilise

$$f(x,v) = 2\cos(3x)\sin(\pi * v), \quad x \in [0\ ;\ 2\pi],\ v \in [-10\ ;\ 10] \tag{5.9}$$

pour valider la reconstruction. La fonction est initialement construite avec des polynômes de degré 2 au 5-ème niveau de raffinement, soit l'équivalent d'une construction avec 96 points par direction (2^5 cellules et trois points par cellule). On raffine ensuite jusqu'au niveau 10 puis on revient au niveau 5. la différence entre la fonction initiale et la fonction reconstruite est nulle.

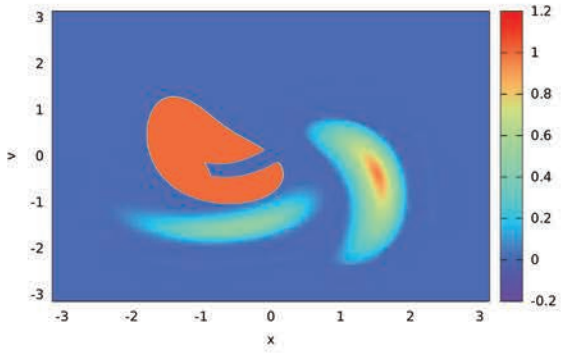
5.3 Validation des schémas en temps

Les schémas Galerkin discontinu semi-lagrangien et caractéristiques-Galerkin discontinu sont testés avec le cas test de rotation-déformation présenté dans l'exemple 5.5 de l'article [36]. la distribution initiale est composée d'un cylindre percé, d'un cône et d'un cosinus, comme indiqué dans l'équation (5.10).

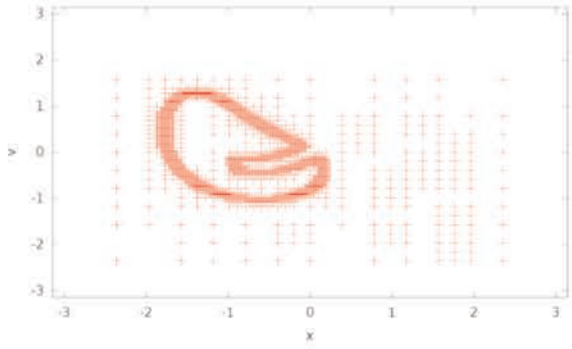
$$f_0(x,y) = \begin{cases} 1 & \text{si } x^2 + (y - 0.3\pi)^2 < (0.3\pi)^2 \text{ et} \\ & (|x| > 0.05\pi \text{ ou } y > 0.3 * \pi)\,, \\ 1 - \sqrt{x^2 + (y * 0.5\pi)^2} & \text{si } x^2 + (y + 0.5\pi)^2 < (0.3\pi)^2, \\ 0.25\left(1 + \cos(\pi\sqrt{(x + 0.5\pi)^2 + y^2})\right) & \text{si } (x + 0.5\pi)^2 + y^2 < (0.3\pi)^2, \\ 0 & \text{sinon.} \end{cases} \tag{5.10}$$

La distribution initiale et le maillage associés sont présentés sur la figure 5.2. Sur cette figure, les oscillations des polynômes aux bords du cylindre tronqué ont été coupées dans un soucis d'échelle et de lisibilité.

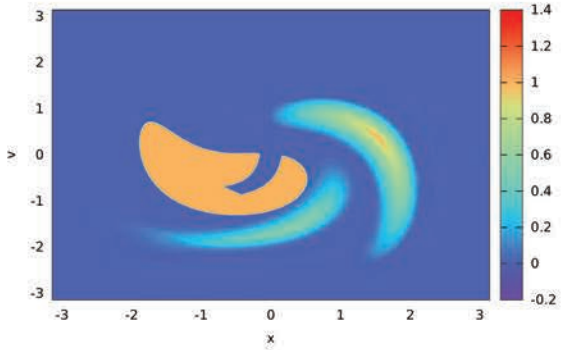
On résout l'équation



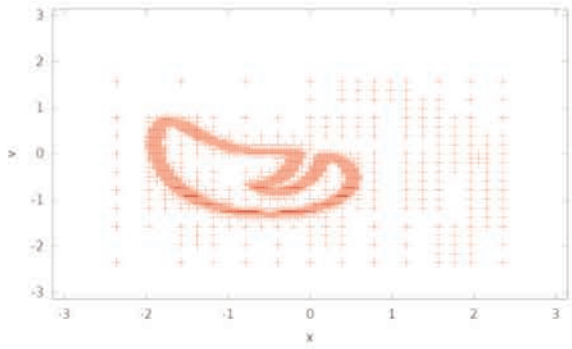
(a) Distribution à $t = 0,40$.



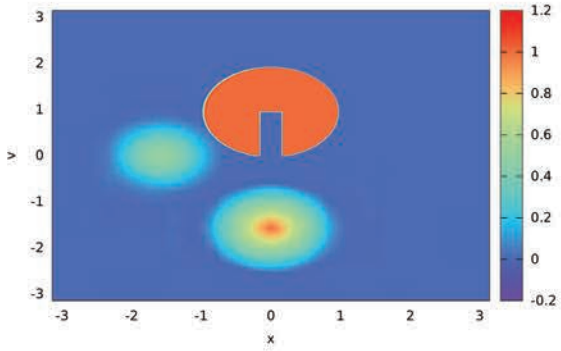
(b) Maillage à $t = 0,40$.



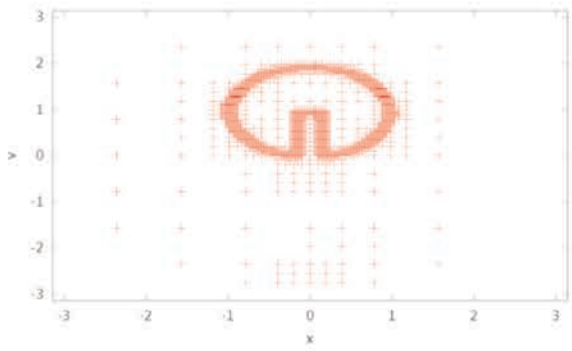
(c) Distribution à $t = 0,75$.



(d) Maillage à $t = 0,75$.

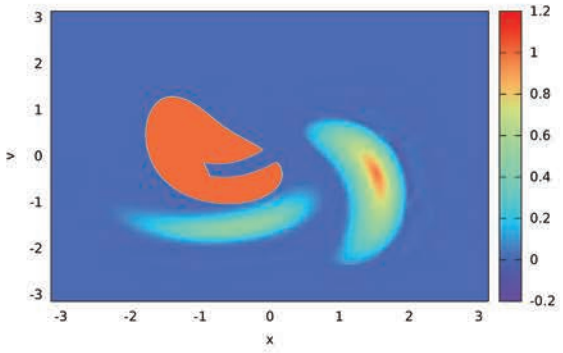


(e) Distribution à $t = 1,5$.

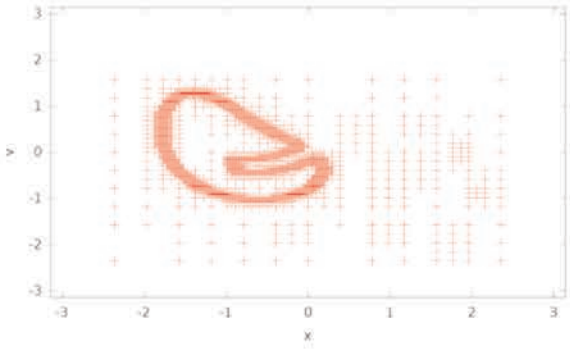


(f) Maillage à $t = 1,5$.

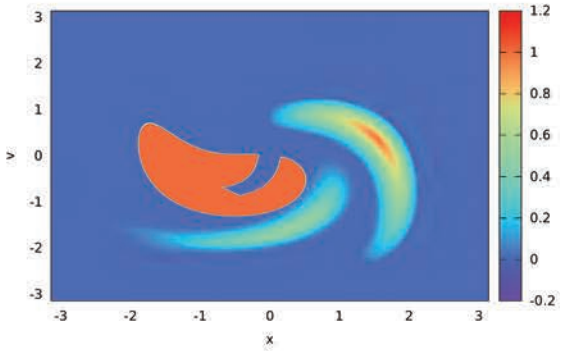
FIGURE 5.3 – Distribution et maillages pour la rotation déformation obtenu par la méthode CGD.



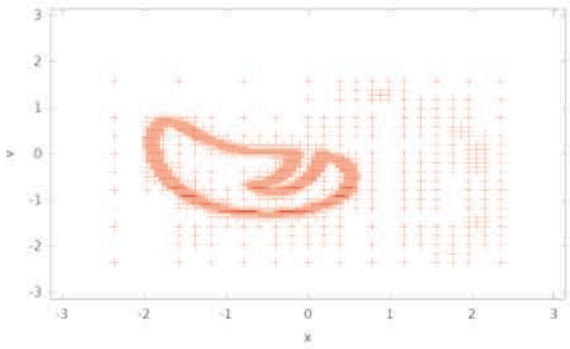
(a) Distribution à $t = 0,40$.



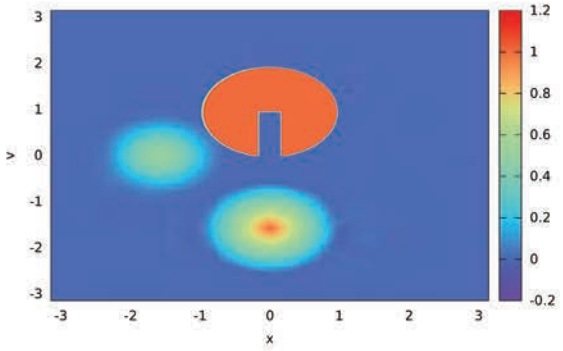
(b) Maillage à $t = 0,40$.



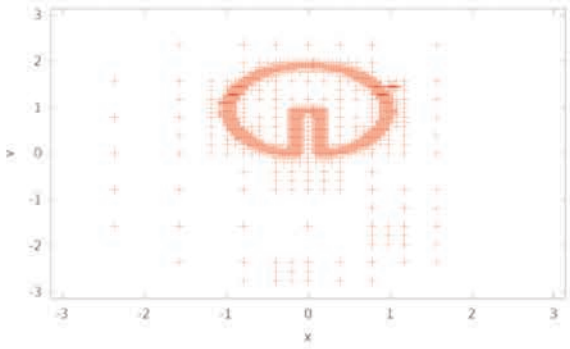
(c) Distribution à $t = 0,75$.



(d) Maillage à $t = 0,75$.



(e) Distribution à $t = 1,5$.



(f) Maillage à $t = 1,5$.

FIGURE 5.4 – Distribution et maillages pour la rotation déformation obtenu par la méthode GDSL.

$$\partial_t f - \partial_x \left(\cos^2\left(\frac{x}{2}\right) \sin(y) \cos\left(\frac{\pi t}{T}\right) \pi f \right) + \partial_y \left(\sin(x) \cos^2\left(\frac{y}{2}\right) \cos\left(\frac{\pi t}{T}\right) \pi f \right) = 0, \quad (x,\ y) \in [-\pi\ ;\ \pi]^2, \quad (5.11)$$

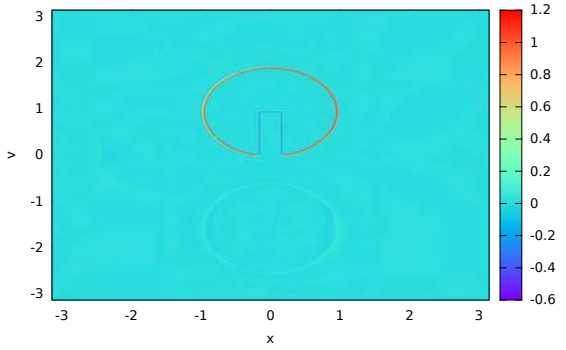
avec $T = 1.5$. La déformation est périodique. Elle est maximale à $t = 0.75$ et la distribution revient à son état initiale à $t = 1.5$. La distribution et le maillage à $t = 0.40$, à $t = 0.75$ (déformation maximale) et à $t = 1.5$ (retour à l'état initial) sont présentés sur la figure 5.3 dans le cadre de la méthode CGD et sur la figure 5.4 dans le cadre de la méthode GDSL. Ces figures sont obtenues en utilisant des polynômes de degré 3 dans chaque direction, 8 niveaux de raffinement et un seuil $\epsilon_0 = 0.01$. Le pas de temps vaut 0.01 pour un temps final $T = 1.5$, soit un total de 100 pas de temps.

On peut voir sur les figures 5.3 et 5.4 les oscillations dues à la projection d'une fonction fortement discontinue sur un espace de polynômes. On constate que la diffusion amortit les oscillations aux bords du cylindre et du cône. On voit également que le maillage suit très bien les détails de la distribution.

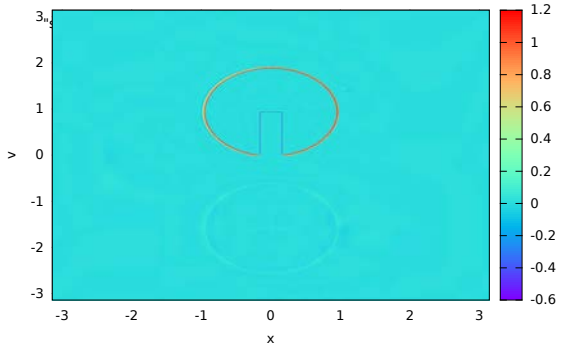
La figure 5.5 montre l'écart entre la solution numérique et la solution analytique au temps final, ainsi que l'écart entre les deux méthodes au temps final. On constate sur les figures 5.5a et 5.5b que l'écart est dominé par la discontinuité aux bords du cylindre tronqué. Ce résultat est cohérent avec les oscillations polynomiales engendrées par la forte discontinuité aux bords de ce cylindre. En limitant la gamme des écarts représentés, on obtient les figures 5.5c et 5.5d. On constate alors que un écart concentré aux bords du cône et de la cloche, là où la distribution n'est pas infiniment dérivable. Enfin, on compare les distributions finales obtenues via les deux méthodes sur les figures 5.5e et 5.5f. On constate alors que les deux méthodes ne divergent pas au niveau du cylindre. Cela indique que dans les deux cas, les oscillations polynomiales ont été transportées de façons similaires. L'écart entre les deux méthodes semble principalement lié à un artefact numérique que l'on voit apparaître également sur les figures 5.3 et 5.4, une différence de maillage entre les deux méthodes au bord du cône et la formation d'une sorte de traînée dans le cas de la méthode GDSL.

Ce cas test utilise la projection d'une fonction discontinue sur une base de polynômes, ce qui fait que la distribution initiale présente de fortes oscillations. Cet exemple peut donc difficilement être considéré comme un bon cas test pour les propriétés de conservation. Les variations relatives de la masse et des normes L^1, L^2 et L^∞ sont toutefois tracées sur la figure 5.6. La masse et les normes L^1 et L^2 sont bien conservées. La norme L^∞ augmente fortement, l'écart étant principalement localisé aux bords du cylindre tronqué.

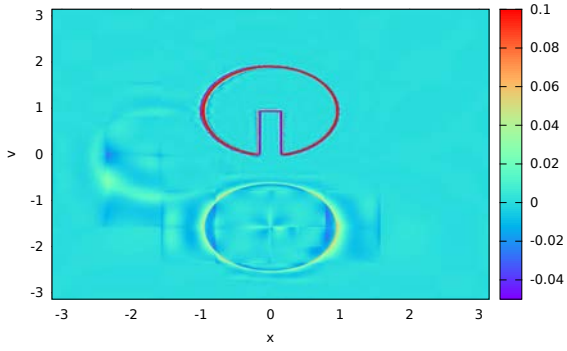
La figure 5.7 présente les erreurs L^1, L^2 et L^∞ obtenues par les deux méthodes en fonction du seuil et regardées uniquement sur la région initiale décrite par un cosinus $(x + 0.5\pi)^2 + y^2 < (0.3\pi)^2$. On constate que toutes les erreurs peuvent être approchées par des fonctions linéaires du seuil.



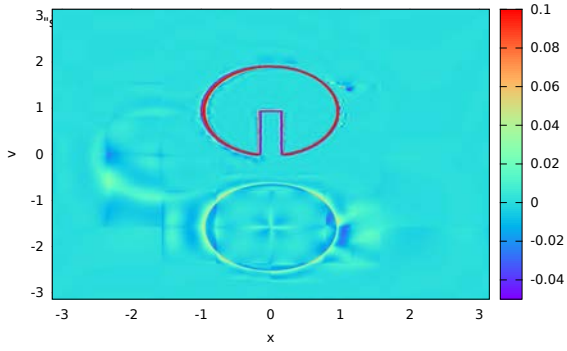
(a) Écart entre la solution obtenue par la méthode CGD et la solution exacte.



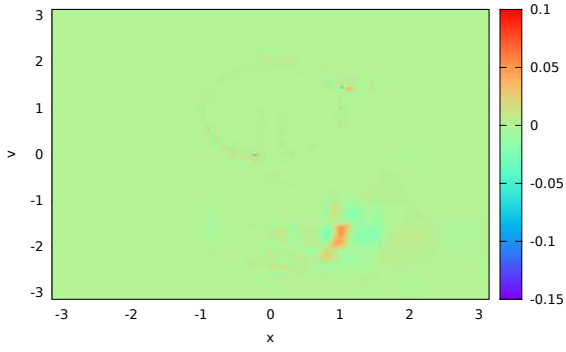
(b) Écart entre la solution obtenue par la méthode GDSL et la solution exacte.



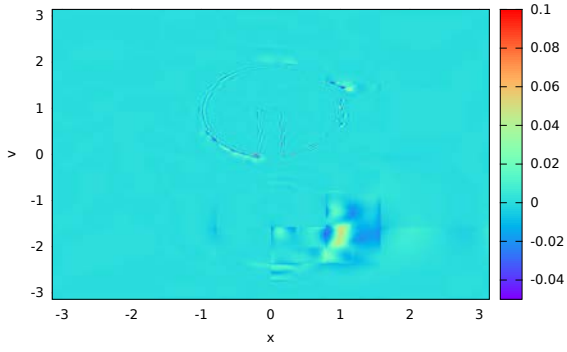
(c) Écart tronqué entre la solution obtenue par la méthode CGD et la solution exacte.



(d) Écart tronqué entre la solution obtenue par la méthode GDSL et la solution exacte.

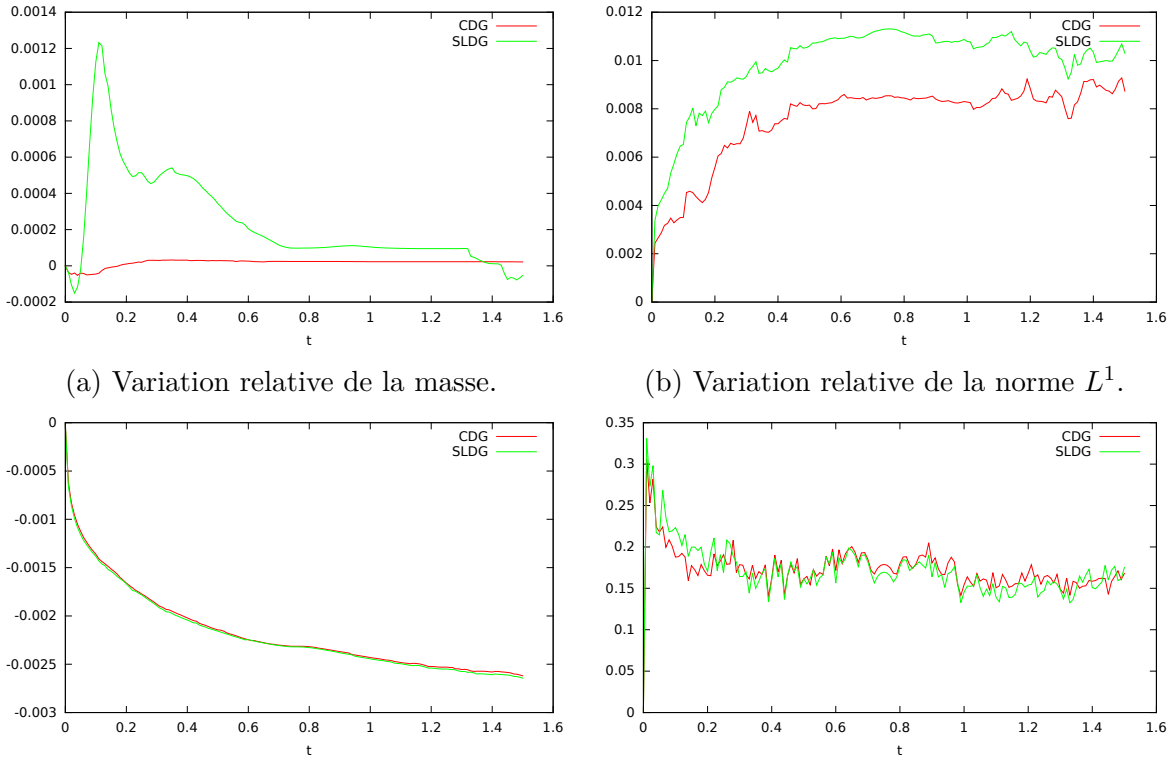


(e) Écart entre les solutions obtenues par les méthodes CGD et GDSL.



(f) Écart tronqué entre les solutions obtenues par les méthodes CGD et GDSL.

FIGURE 5.5 – Écarts entre la distribution finale numérique et la distribution finale analytique pour le cas test de rotation-déformation.



(a) Variation relative de la masse.

(b) Variation relative de la norme L^1.

(c) Variation relative de la norme L^2.

(d) Variation relative de la norme L^∞.

FIGURE 5.6 – Variations des principaux diagnostiques pour le cas test de rotation-déformation.

5.4 Équation de Burgers 1D

On s'intéresse dans cette section à l'équation de Burgers. Ceci s'inscrit dans le prolongement de la démonstration de la méthode GDSL, et en particulier l'équation (2.44). À la section 2.3.1, nous avons démontré que dans le cadre d'une équation de type

$$\partial_t f(x,t) + \partial_x \left(a(x,t) f(x,t)\right) = 0, \tag{5.12}$$

on a

$$\int_{(x_i^*,t^n)}^{(x_i,t^{n+1})} \begin{pmatrix} f(x,t) \\ a(x,t)f(x,t) \end{pmatrix} \cdot \boldsymbol{n}\, ds = 0 \tag{5.13}$$

le long de la coordonnée curviligne formée par une caractéristique.

Nous aimerions étendre ce résultat à l'équation de Burgers

$$\partial_t f(x,t) + f(x,t)\partial_x f(x,t) = 0. \tag{5.14}$$

Dans ce cadre, en reprenant l'analyse de la section 2.3.1, l'équation des caractéristiques devient

$$\frac{d}{dt} X(x^*_{i-1/2}, t) = f(X(x^*_{i-1/2}, t), t), \tag{5.15a}$$

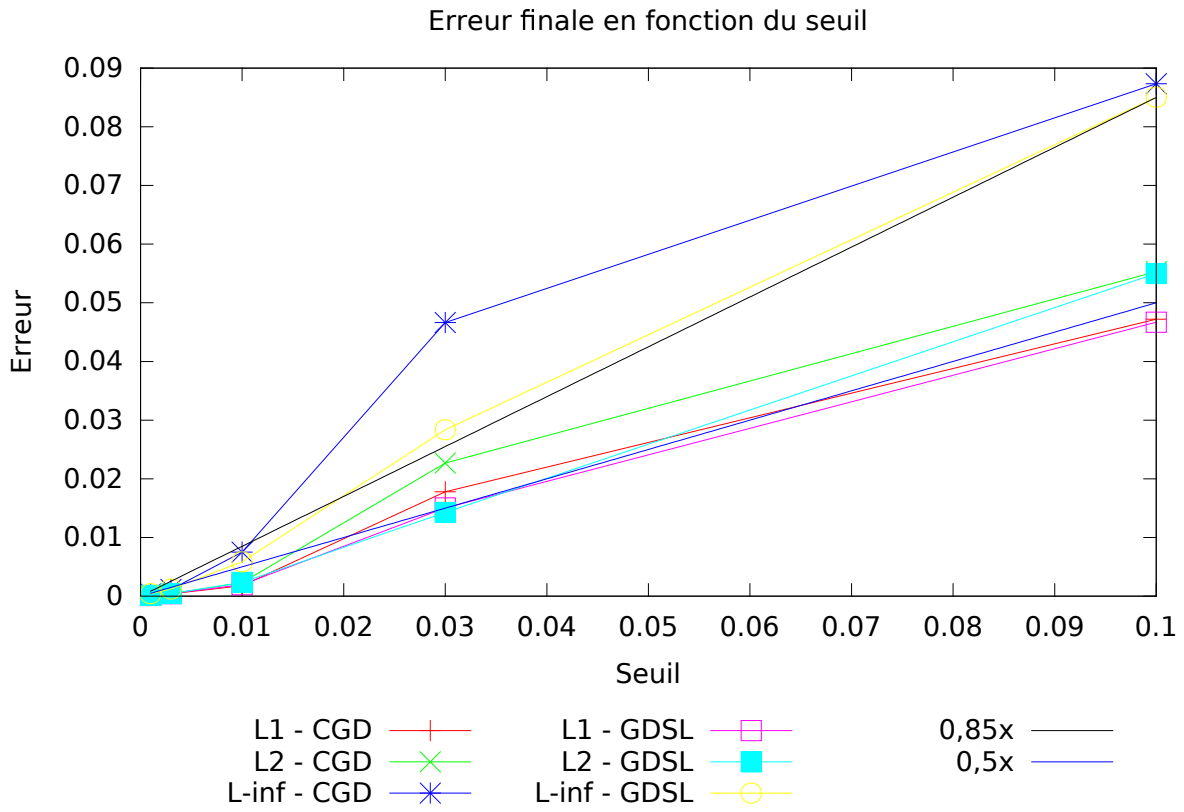


FIGURE 5.7 – Erreur L^1, L^2 et L^∞ pour la déformation à $t = 1,5$ en fonction du seuil.

$$X(x^*_{i-1/2}, t^n) = x^*_{i-1/2}, \tag{5.15b}$$
$$X(x_{i-1/2}, t^{n+1}) = x_{i-1/2}, \tag{5.15c}$$

l'équation (5.13) devient

$$\int_{(x_i^*,t^n)}^{(x_i,t^{n+1})} \begin{pmatrix} f(x,t) \\ (f(x,t))^2 \end{pmatrix} \cdot \boldsymbol{n}\, ds = 0. \tag{5.16}$$

D'après l'analyse de la section 2.3.1, l'équation (5.16) devient

$$\int_{(x_i^*,t^n)}^{(x_i,t^{n+1})} \begin{pmatrix} f(x,t) \\ (f(x,t))^2 \end{pmatrix} \cdot \begin{pmatrix} (f(x,t)) \\ -1 \end{pmatrix} ds = 0. \tag{5.17}$$

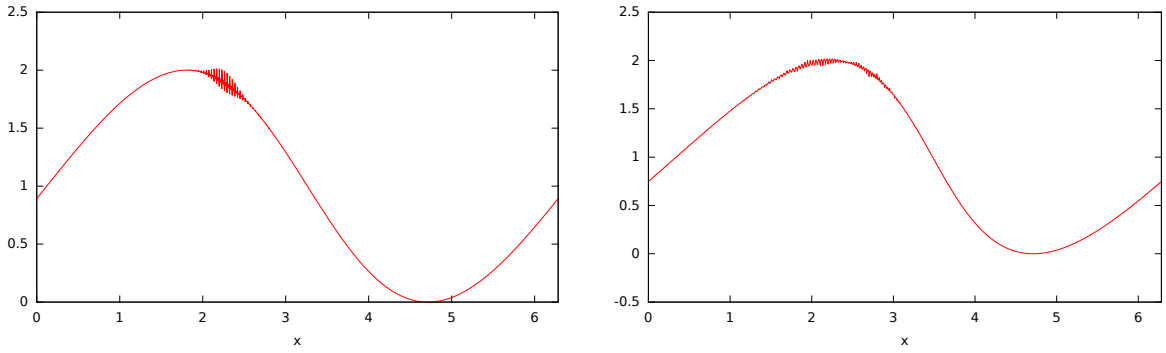
Ici aussi l'intégrale est nulle et la méthode GDSL peut être appliquée avant les chocs.

On teste donc les deux méthodes, GDSL et CGD, sur l'équation de Burgers à une dimension

$$\partial_t f(x,t) + f(x,t)\partial_x f(x,t) = 0. \tag{5.18}$$

On part de la distribution initiale

$$f(x,0) = 1 + sin(x), \quad x \in [0\ ;\ 2\pi] \tag{5.19}$$



(a) Distribution à $t = 0,06$ avec la méthode GDSL.

(b) Distribution à $t = 0,17$ avec la méthode CGD.

FIGURE 5.8 – Oscillations sur la résolution de l'équation de Burgers.

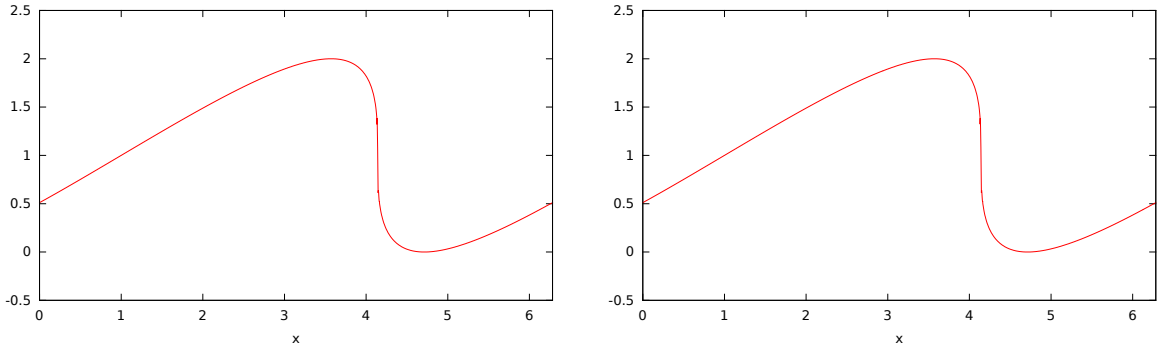
qui produit un choc à $T = 1$. On s'intéresse à ce qui se passe avant le choc, l'utilisation des caractéristiques n'étant valide que jusqu'au choc. Dans le cas de l'équation de Burgers, avant le choc, les caractéristiques sont résolues par

$$\frac{d}{dt}X(x^*_{i-1/2}, t) = f(X(x^*_{i-1/2}, t), t) \tag{5.20a}$$

$$X(x_{i-1/2}, t^{n+1}) = x_{i-1/2}. \tag{5.20b}$$

On effectue un premier test avec des polynômes de degré 3, un maillage uniforme de 200 cellules et un pas de temps de 10^{-3}. Les caractéristiques sont calculées par une méthode d'Euler implicite. On constate alors sur la figure 5.8 l'apparition d'oscillations là où la vitesse est maximale. Ces oscillations n'apparaissent pas de la même façon ni au même moment pour les deux méthodes. On retrouve le même type d'oscillations sur les cas tests d'astrophysique présentés à la section 6.5. Dans le cadre de l'équation de Burgers, ces oscillations s'amplifient et la solution numérique explose.

Des simulations réalisées avec des polynômes de degré 3, un maillage uniforme de 400 cellules et un pas de temps de 10^{-5} permettent de supprimer les oscillations. La figure 5.9 montre la distribution obtenue à $t = 0,5$ pour les deux méthodes. Dans ce cas, même avec un maillage uniforme de 200 cellules, les oscillations n'apparaissent pas. Le problème vient alors de la mauvaise résolution spatiale de la zone de choc. La pente devient très raide très localement et on constate alors l'apparition d'oscillations polynomiales. Ce cas test fait apparaître une limitation de la taille du pas de temps dans certains cas. Cette limitation apparaît clairement dans les cas tests de Vlasov-Poisson pseudo polaire exposés à la section 6.4.



(a) Distribution à $t = 0,50$ avec la méthode GDSL.

(b) Distribution à $t = 0,50$ avec la méthode CGD.

FIGURE 5.9 – Formation du choc lors de la résolution de l'équation de Burgers.

Chapitre 6

Résultats numériques

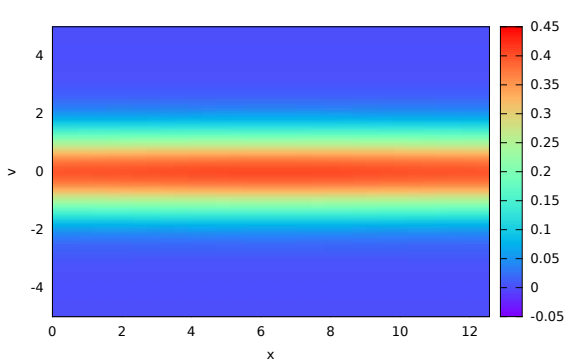
Ce chapitre présente les résultats numériques obtenus pour divers cas tests ainsi que l'évolution de différentes grandeurs significatives. Lorsqu'ils sont donnés, les maximums et minimums sont calculés aux points de Gauss sur chaque cellule.

6.1 Amortissement Landau

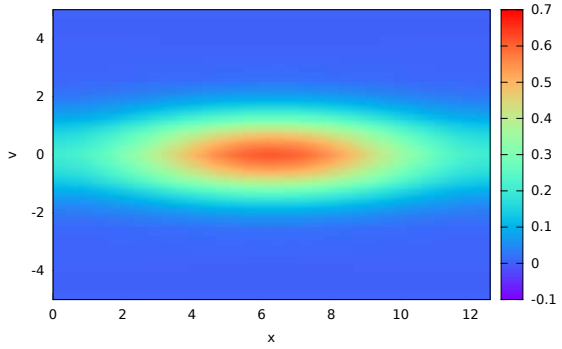
Le cas test d'amortissement Landau est un cas test de Vlasov-Poisson permettant de poursuivre la validation du code. On considère la distribution initiale

$$f_0(x,v) = \frac{(1-\alpha\cos(x/2))}{\sqrt{2\pi}}\exp\left(\frac{-v^2}{2}\right), \quad (x,v) \in [0\ ;\ 4\pi]\times[-L_v\ ;\ L_v]. \tag{6.1}$$

L_v est choisi suffisamment grand pour assurer des conditions de bord de type Dirichlet dans la direction v. Habituellement $L_v = 5$ est suffisant et nous utiliserons cette valeur. On trouve parfois dans la littérature des cas où L_v descend jusqu'à 4 et des cas où L_v monte jusqu'à 10. Le choix de α est beaucoup plus déterminant et deux valeurs sont



(a) Distribution initiale pour le cas linéaire.



(b) Distribution initiale pour le cas non-linéaire.

FIGURE 6.1 – Distributions initiales pour le cas Landau.

principalement utilisées. Tout d'abord, $\alpha = 0.01$ correspond au cas d'amortissement linéaire. Dans ce cas le taux de décroissance du champ électrique et la période sont connus analytiquement (bien qu'en pratique la période soit rarement étudiée). Ensuite, $\alpha = 0.5$ correspond à l'amortissement non-linéaire. Dans ce cas ni le taux de décroissance, ni le taux de croissance ne sont connus mais ils peuvent être comparés à de nombreuses sources dans la littérature. Les distributions initiales sont présentées sur la figure 6.1. On rappelle que l'étude du champ électrique consiste à étudier la partie électrique de l'énergie totale (2.11), c'est-à-dire

$$\frac{1}{2}\int_{\Omega_x} |E(x,t)|^2\, dx. \tag{6.2}$$

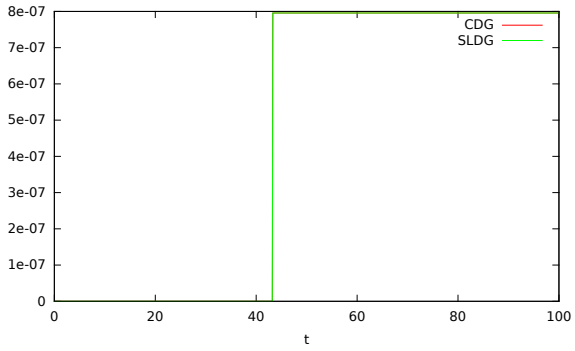
Ces deux cas sont étudiés jusqu'à un temps final $T = 100$ périodes plasma et un pas de temps de 0.1 période plasma, pour un total de 1000 pas de temps. Les polynômes utilisés sont de degré 2 par direction. On utilise au maximum 8 niveaux de raffinement. Afin d'obtenir des résultats cohérents, un seuil $\epsilon_0 = 10^{-3}$ est nécessaire dans le cas linéaire, tandis qu'un seuil de $\epsilon_0 = 10^{-2}$ est suffisant dans le cas non linéaire. L'unité de temps pour les figures représentant la distribution et le maillage au cours du temps est la période plasma abrégée pp.

6.1.1 L'amortissement Landau linéaire

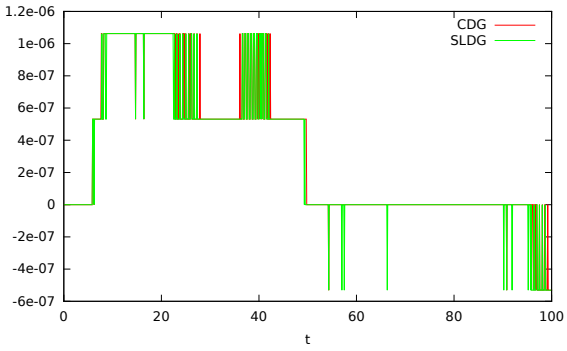
Maillage non limité.

On utilise dans cette section les paramètres présentés précédemment pour l'amortissement Landau linéaire. dans cette section, le niveau maximum de raffinement autorisé n'est pas atteint.

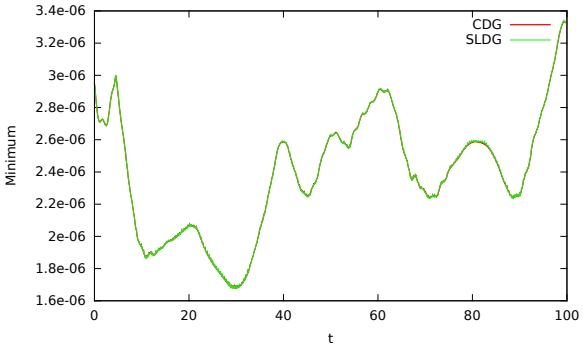
La figure 6.2 présente l'évolution des principales valeurs de diagnostique pour le cas Landau linéaire. Les variations des normes L^1 et L^2 en forme de créneaux sont typiques des variations d'ordre de l'erreur machine amplifiées par différents facteurs. Bien qu'importante en proportion, la variation du minimum est en réalité faible devant l'amplitude maximale de la distribution. Les schémas ne conservant ni les maximums, ni les minimums, les variations de faible amplitude observées sur ces quantités sont cohérentes. On constate néanmoins que pour le cas test d'amortissement Landau linéaire, la distribution reste positive. La variation de l'énergie totale est faible mais on la constate dominée par la variation de l'énergie électrique. La quantité de mouvement initialement nulle a une variation totale qui est aussi de l'ordre de l'erreur machine. La valeur qui nous intéresse le plus dans ce cas est l'évolution du champ électrique figure 6.2f. On peut voir jusqu'à $t = 30$ périodes plasmas la décroissance linéaire du champ électrique avec le taux attendu, -0.153. Néanmoins ce taux n'est conservé que sur un temps relativement court. Ce taux pourrait être conservé sur des temps plus long par l'utilisation de polynômes de degré plus élevés et d'un seuil plus petit accompagné d'un plus grand niveau de raffinement. Néanmoins, cette étude peut être réalisée en utilisant simplement un maillage fixe structuré non-uniforme et non-conforme. Le but de cette thèse étant l'utilisation de maillages adaptatifs, j'ai jugé ces premiers résultats satisfaisants pour le but à atteindre.



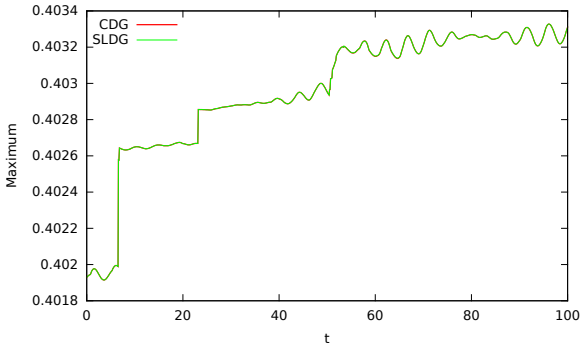
(a) Variation relative de la norme L^1.



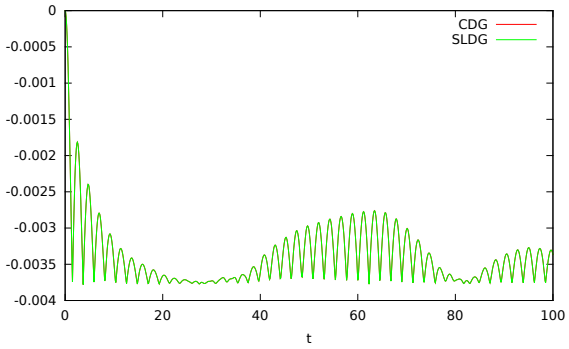
(b) Variation relative de la norme L^2.



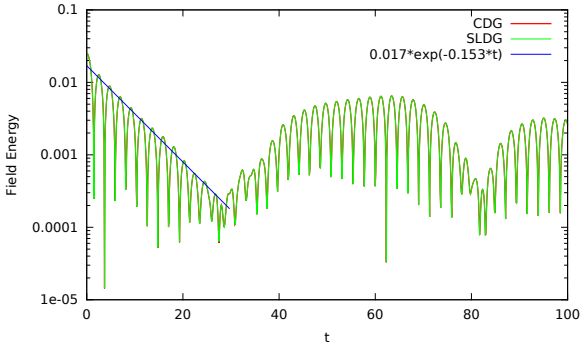
(c) Évolution du minimum.



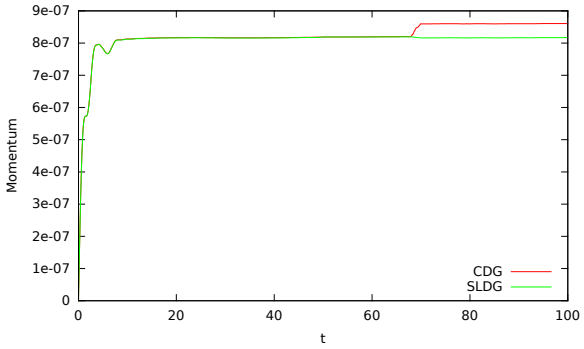
(d) Évolution du maximum.



(e) Variation relative de l'énergie totale.

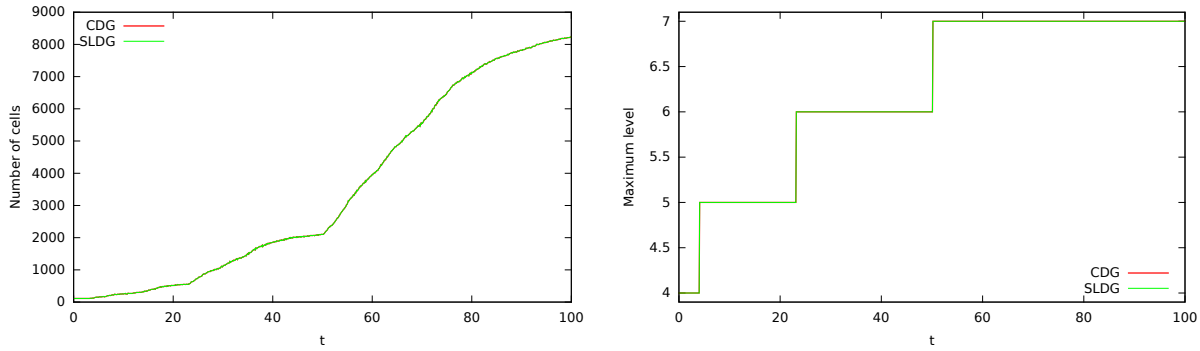


(f) Évolution de l'énergie électrique.



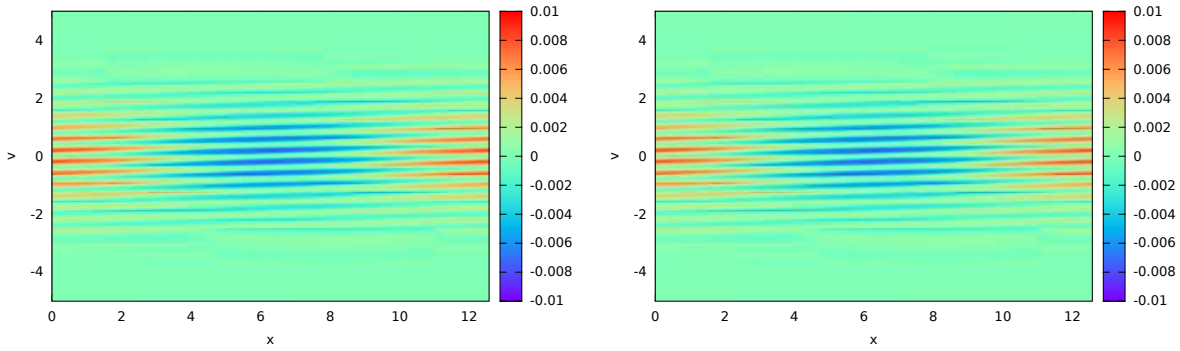
(g) Évolution de la quantité de mouvement.

FIGURE 6.2 – Évolution des quantités de références pour le cas Landau linéaire.



(a) Évolution du nombre de cellules. (b) Évolution du niveau maximum.

FIGURE 6.3 – Évolution du maillage pour l'amortissement Landau linéaire.



(a) Avec la méthode GDSL. (b) Avec la méthode CGD.

FIGURE 6.4 – Différence $f(x, v, 30) - f_0(x, v)$ pour le cas test Landau linéaire.

Les figures 6.3 présentent l'évolution du nombre de cellules feuilles et du niveau maximum de calcul au cours du temps. Le maillage initial est très grossier car la distribution présente peu de petits détails. C'est avec la formation de petites oscillations que le maillage va progressivement se raffiner et que des cellules plus petites vont être crées. Ces petites oscillations sont mises en évidence sur la figure 6.4 en traçant l'écart entre la distribution initiale $f_0(x, v)$ et la distribution à $t = 30$ périodes plasma. On remarque que bien que le niveau de raffinement maximum autorisé soit 8, il n'est ici jamais atteint. Cela signifie qu'avec les critères utilisés, les détails sont suffisamment petits pour être considérés comme négligeables et ignorés.

Un dernier point important mis en évidence par les figures 6.2 et 6.3 est la similitude entre les résultats obtenus par la méthode GDSL et la méthode CGD. L'équivalence des méthodes dans le cas du transport linéaire est prouvée à la section 4.1 de l'article [36]. Les équations résolues ici ne correspondent pas à un transport linéaire, pour cette raison les résultats ne sont pas absolument égaux.

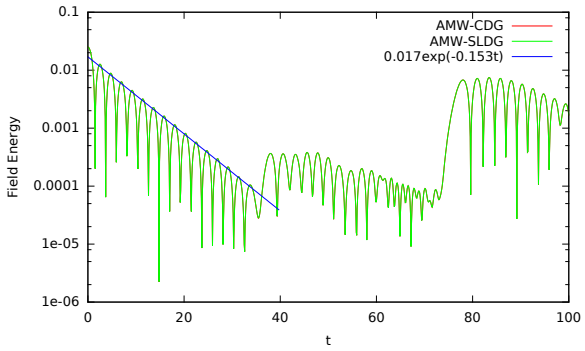
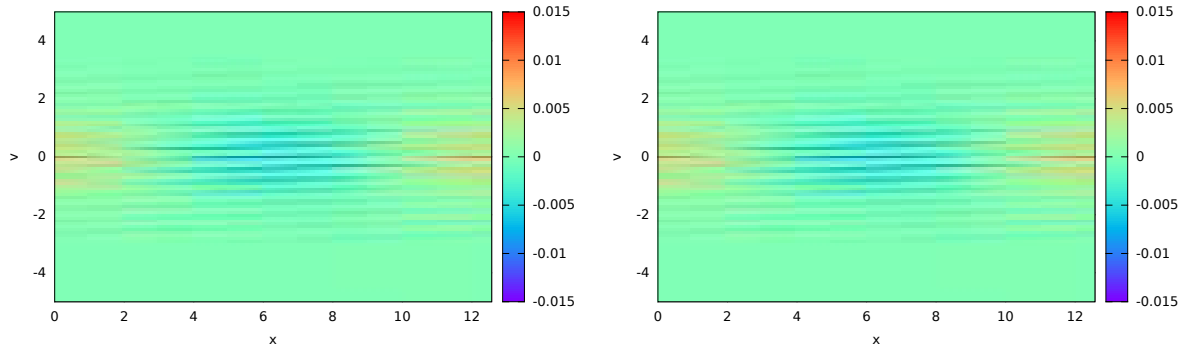


FIGURE 6.5 – Évolution de l'énergie électrique pour le cas Landau linéaire avec maillage limité.



(a) Avec la méthode GDSL. (b) Avec la méthode CGD.

FIGURE 6.6 – Différence $f(x, v, 30) - f_0(x, v)$ pour le cas test Landau linéaire avec maillage limité.

Maillage limité

On considère dans cette section les mêmes paramètres de discrétisation en temps et en espace que précédemment mais on utilise des critères d'adaptativité différents. Le seuil de raffinement utilisé est $\epsilon_0 = 10^{-4}$ et on ne permet ici que cinq niveaux de raffinement. Le but est ici de forcer le raffinement du maillage afin de s'approcher d'un maillage uniforme.

La figure 6.5 montre la décroissance linéaire du champs électrique suivie de deux rebonds. La première récurrence apparaît à $t = 38$ pp et la deuxième récurrence est observée à $t = 76$ pp. Ce phénomène est clairement dû au phénomène de récurrence décrit par Cheng et Knorr [15]. Dans notre cas, le temps de récurrence est $T_r = 38$ pp. Dans le cas d'un maillage uniformément raffiné au plus fin niveau autorisé, le temps de récurrence est $T_r = 2\pi/(k\Delta v) = 40.21$ pp. L'erreur relative entre les deux temps de récurrence est d'environ 5.5%. Comme le maillage utilisé ici est dynamique et n'est pas uniformément raffiné au niveau le plus fin, l'écart est cohérent.

Les figures 6.6 mettent en évidences les petites oscillations de la fonctions de distributions. Comparées aux figures 6.4, on constate immédiatement qu'un maillage beaucoup

plus grossier entraîne un suivit moins précis des petites structures, même si le seuil de raffinement est plus petit.

6.1.2 L'amortissement Landau non-linéaire

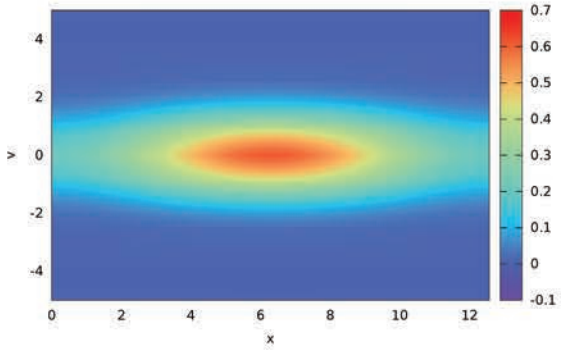
La figure 6.7 présente l'évolution de la distribution et du maillage dans le cas de l'amortissement Landau non-linéaire. On y voit très nettement le processus de filamentation et le maillage qui suit les filaments jusqu'à la figure 6.7f. Ensuite, à partir de la figure 6.7g, les filaments sont plus resserrés et le maillage est bloqué au niveau maximum autorisé pour finir par ne former qu'un maillage semblable à un maillage uniforme au plus fin niveau de raffinement autorisé. On voit néanmoins que même en temps long le raffinement du maillage suit les détails de la distribution jusqu'aux limites qui lui sont imposées.

Les figures 6.8 présentent l'évolution temporelle des principaux diagnostiques utilisés dans le cas Landau non-linéaire. La masse numérique est calculée comme

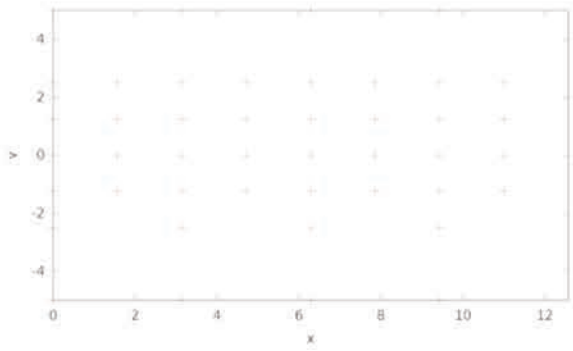
$$\iint_{\Omega_x \times \Omega_v} f(x, v, t)\, dxdv. \tag{6.3}$$

On note ici l'absence de valeur absolue alors que la distribution numérique peut être négative. Bien que les propriétés du système d'équations indiquent que la masse et la norme L^1 devraient référer à la même quantité, parce que les schémas ne conservent pas le signe de la distribution il est intéressant de suivre les deux. On constate ainsi que pour les deux schémas, la variation relative de masse est de l'ordre de 2×10^{-4} alors que la variation relative de norme L^1 est de l'ordre de 2×10^{-3}, soit dix fois plus importante. Pour ce cas test la masse numérique est donc mieux conservée grâce à l'apparition de valeurs négatives pour la fonction de distribution. On constate également que les courbes des méthodes GDSL et CGD ont des profils très similaires avec une amplitude de variation légèrement moindre pour la méthode GDSL. Bien que ce cas soit non-linéaire et que le maillage soit très vite amené à son raffinement maximal sur une large part du domaine, la norme L^2 décroit très peu, ce qui indique une faible dissipation pour ce cas test. Les maximums et minimums varient plus fortement, notamment en temps long. Ils permettent également d'observer un écart important entre les méthode GDSL et CGD sans pour autant signifier qu'une méthode aurait un comportement plus physique que l'autre.

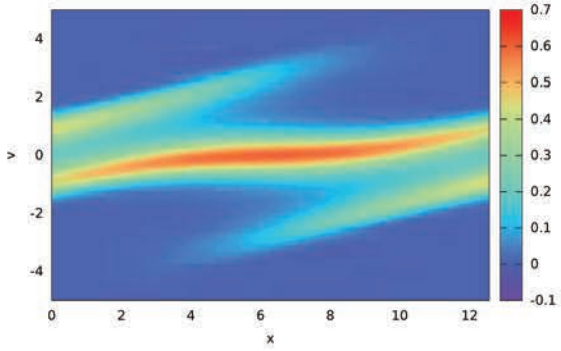
On constate au cours des premiers pas de temps une forte augmentation de l'énergie totale alors que l'énergie électrique décroit. Cela indique qu'au début de la simulation l'énergie cinétique augmente fortement. Cette augmentation se fait durant les entre $t = 0$ et $t = 1.5$ période plasma. Si une explication liée à l'utilisation de points de calcul aux bords des cellules peut être avancée dans le cas de la méthode GDSL, cette explication n'est pas valide dans le cas de la méthode CGD qui n'utilise que les points de Gauss-Legendre qui ne se trouvent par conséquent pas aux bords des cellules. L'énergie reste ensuite quasi-constante autour de la valeur atteinte. On observe un phénomène similaire pour la quantité de mouvement bien qu'à une échelle moindre (6×10^{-3} dans le cas de la quantité de mouvement contre une variation de 26% dans le cas de l'énergie totale). Il est donc



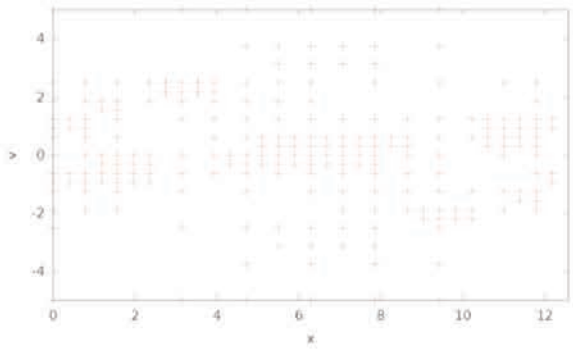
(a) Distribution initiale.



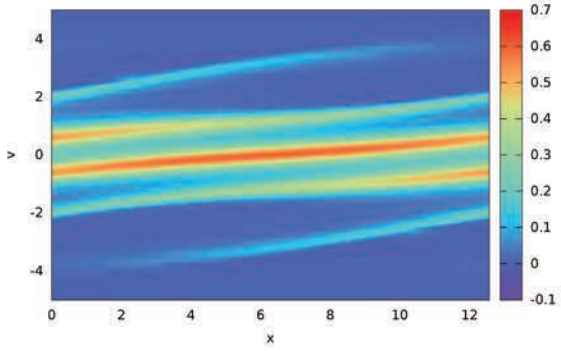
(b) Maillage initial.



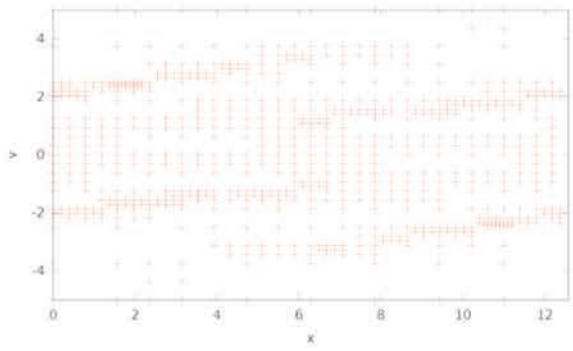
(c) Distribution à $t = 4$ pp.



(d) Maillage à $t = 4$ pp.

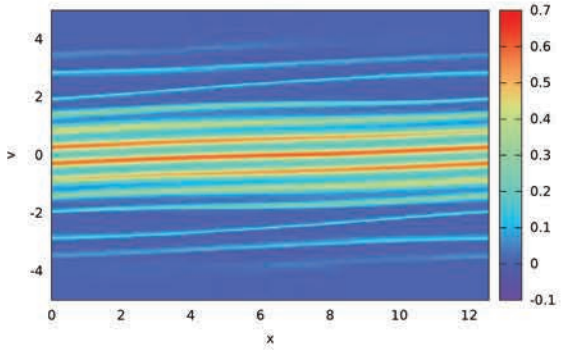


(e) Distribution à $t = 8$ pp.

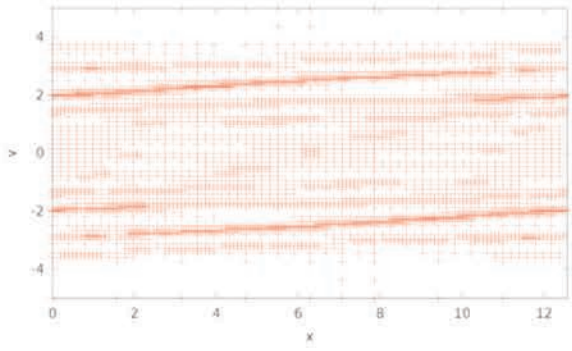


(f) Maillage à $t = 8$ pp.

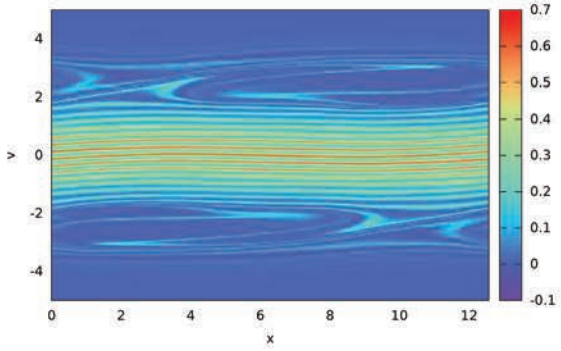
FIGURE 6.7 – Évolution de la distribution et du maillage pour l'amortissement Landau non-linéaire par la méthode CGD.



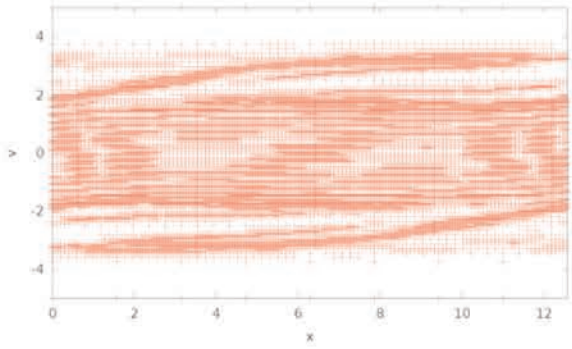
(g) Distribution à $t = 20$ pp.



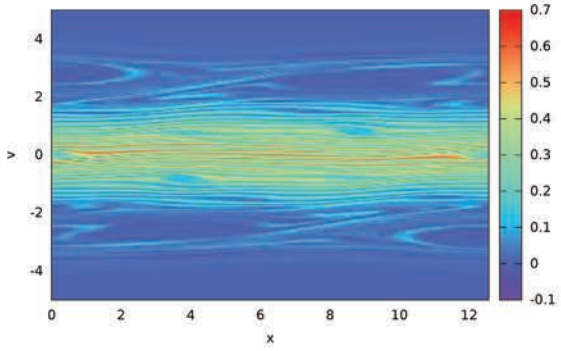
(h) Maillage à $t = 20$ pp.



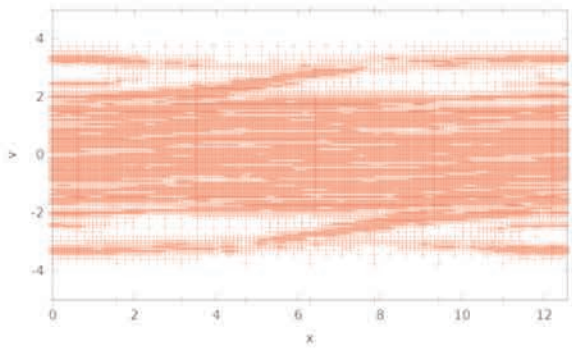
(i) Distribution à $t = 50$ pp.



(j) Maillage à $t = 50$ pp.

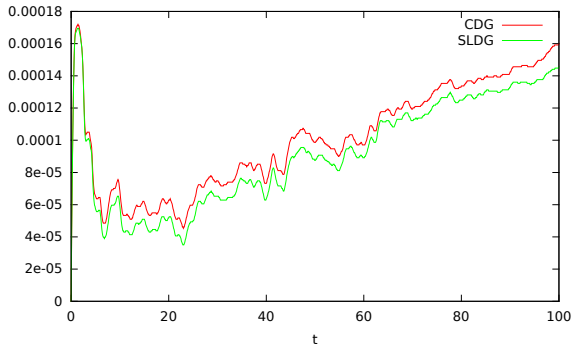


(k) Distribution à $t = 80$ pp.

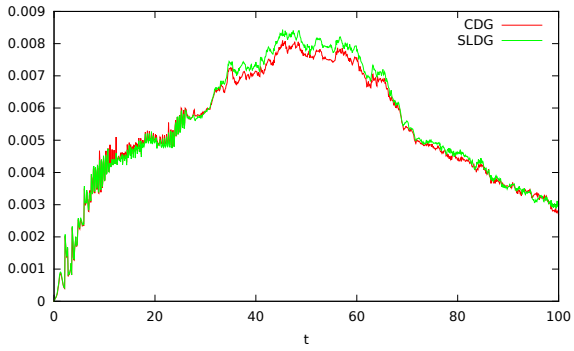


(l) Maillage à $t = 80$ pp.

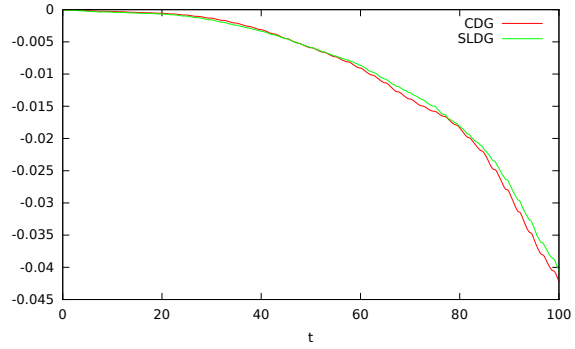
FIGURE 6.7 – Évolution de la distribution et du maillage pour l'amortissement Landau non-linéaire par la méthode CGD.



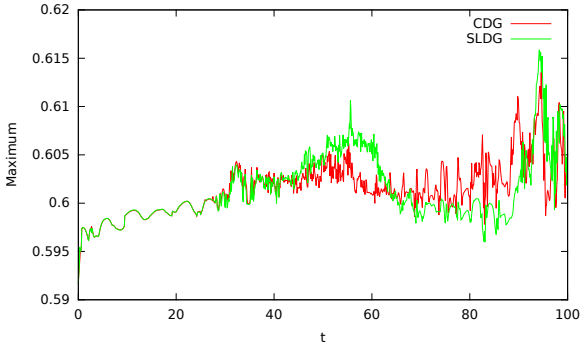
(a) Variation relative de la masse numérique.



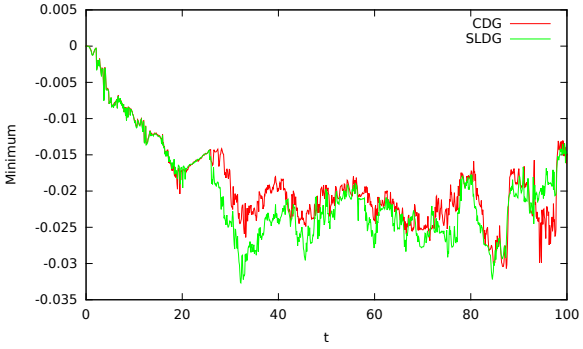
(b) Variation relative de la norme L^1.



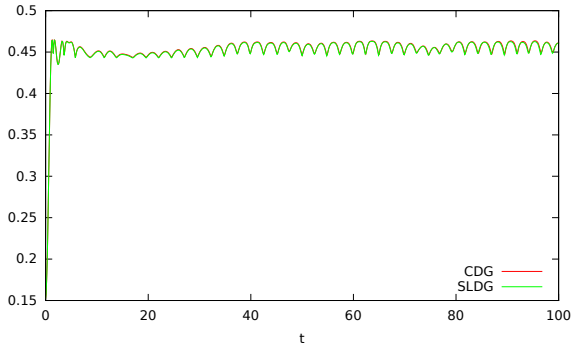
(c) Variation relative de la nome L^2.



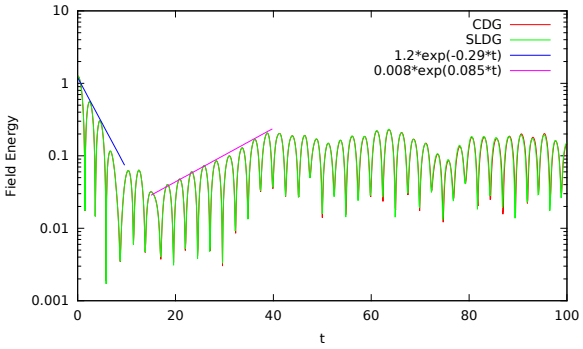
(d) Évolution du maximum.



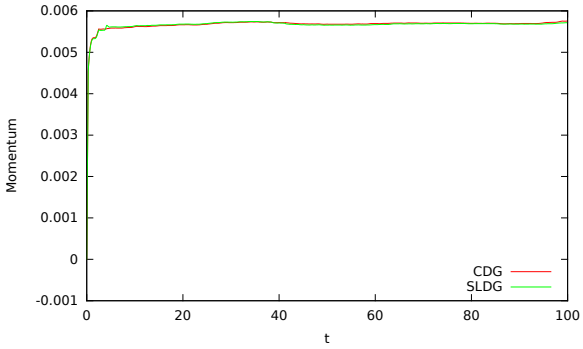
(e) Évolution du minimum.



(f) Variation relative de l'énergie totale.

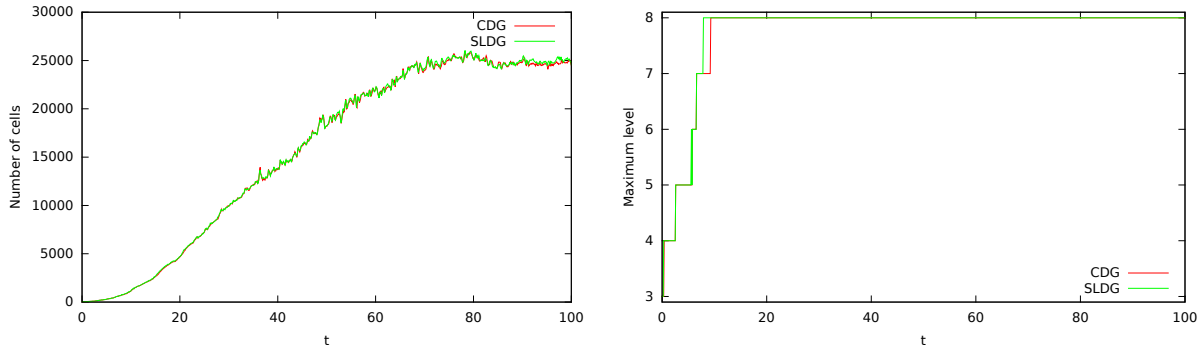


(g) Évolution du champ électrique.



(h) Évolution de la quantité de mouvement.

FIGURE 6.8 – Évolution des quantités de référence pour le cas Landau non-linéaire.



(a) Évolution du nombre de cellules. (b) Évolution du niveau maximum.

FIGURE 6.9 – Évolution du maillage pour l'amortissement Landau non-linéaire.

probable qu'un effet numérique que je n'ai pas identifié soit à l'origine de cette variation. La figure 6.8g présente l'évolution du champ électrique. On constate une décroissance avec un taux de -0.29 durant les dix premières périodes plasma, puis entre 20 et 40 périodes plasmas une croissance linéaire avec un taux de 0.085. Ces taux sont en adéquation avec ceux que l'on peut trouver dans la littérature, notamment dans [10, 21, 38].

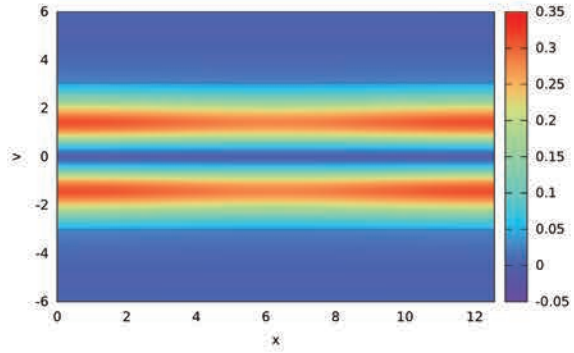
La figure 6.9 illustre l'évolution du nombre de cellules et du niveau de calcul le plus fin utilisé au cours du temps. On constate que dans ce cas, le huitième niveau de raffinement (qui est le niveau de raffinement le plus fin autorisé ici) est très rapidement atteint. On peut voir sur les figures 6.7 que dès $t = 8$ périodes plasma le maillage est très fin. Les détails étant de plus en plus fins et occupant de plus en plus d'espace, le nombre de cellules continue d'augmenter jusqu'à atteindre un état quasi-stationnaire où le maillage pourrait être remplacé par un maillage uniforme constant et visible à $t = 80$ périodes plasma sur les figures 6.7.

6.2 Instabilité double faisceau

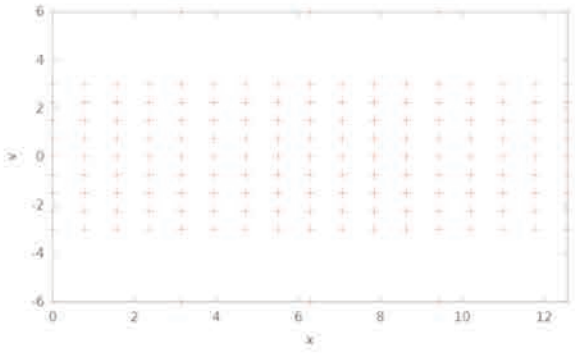
Il existe plusieurs cas d'instabilité double faisceau pour Vlasov-Poisson. On choisit ici les conditions initiales suivantes.

$$f_0(x,v) = \frac{v^2}{\sqrt{2\pi}}(1 + 0.05\cos(0.5x))\exp\left(\frac{-v^2}{2}\right), \quad (x,v) \in [0\ ;\ 4\pi] \times [-6\ ;\ 6]. \tag{6.4}$$

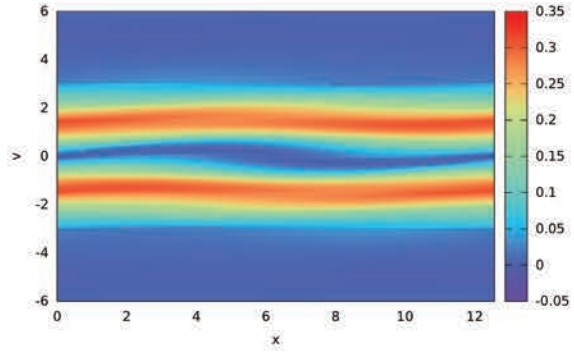
Ce cas test est le même que celui présenté section 3.7.3 de la thèse [11]. On l'étudie jusqu'à $T = 100$ périodes plasma avec un pas de temps de 0.1 période plasma pour un total de 1000 pas de temps. Les polynômes utilisés sont de degré 2 par direction. Le niveau de raffinement maximum est 8 et le seuil utilisé est $\epsilon_0 = 0.01$. Ce cas test développe un vortex et des filaments.



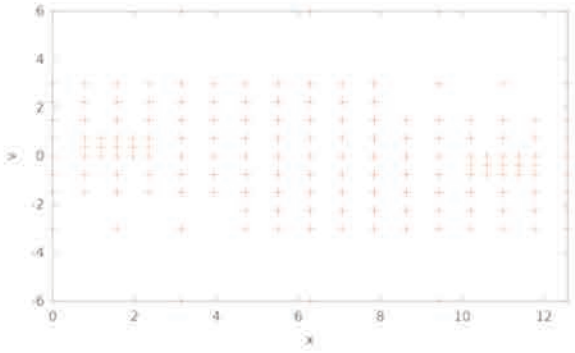
(a) Distribution initiale.



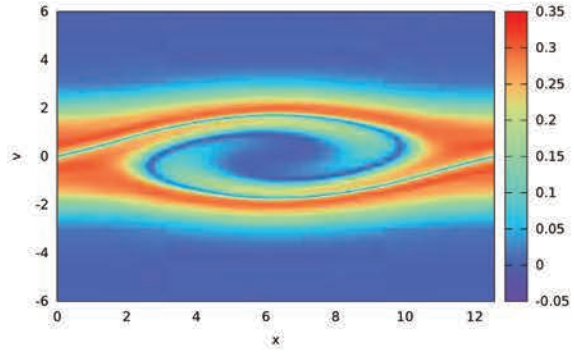
(b) Maillage initial.



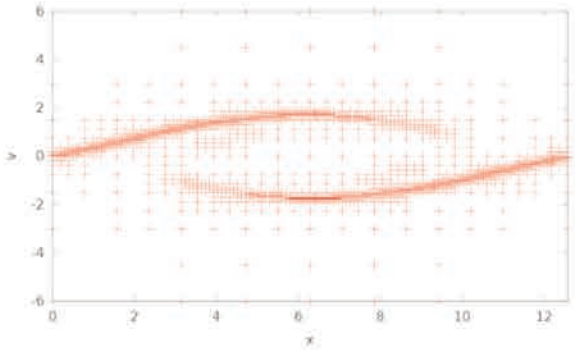
(c) Distribution à $t = 8$ pp.



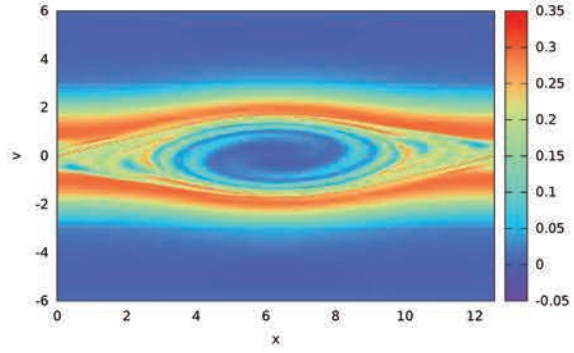
(d) Maillage à $t = 8$ pp.



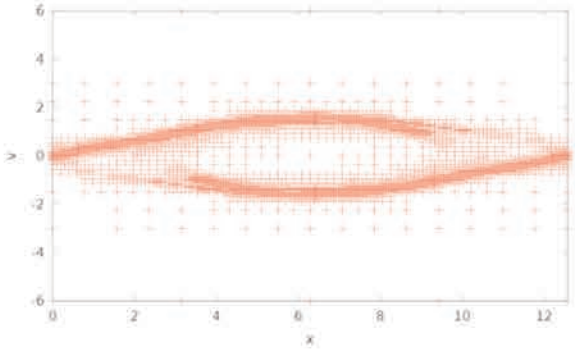
(e) Distribution à $t = 20$ pp.



(f) Maillage à $t = 20$ pp.

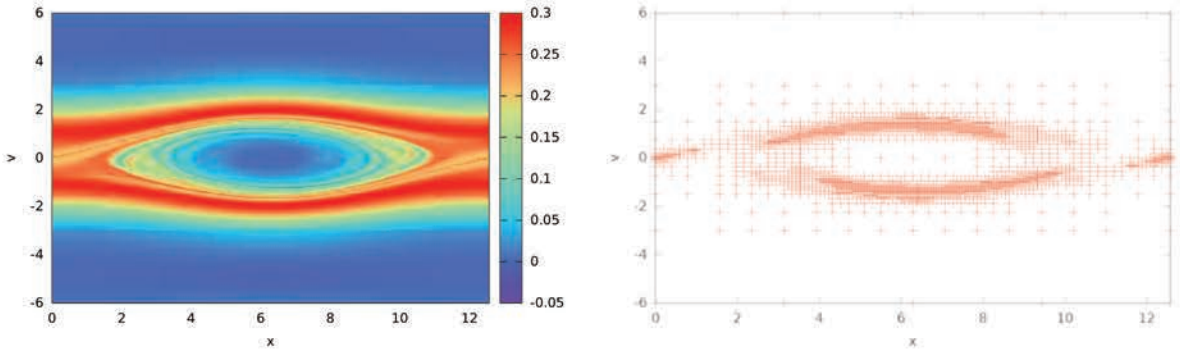


(g) Distribution à $t = 60$ pp.



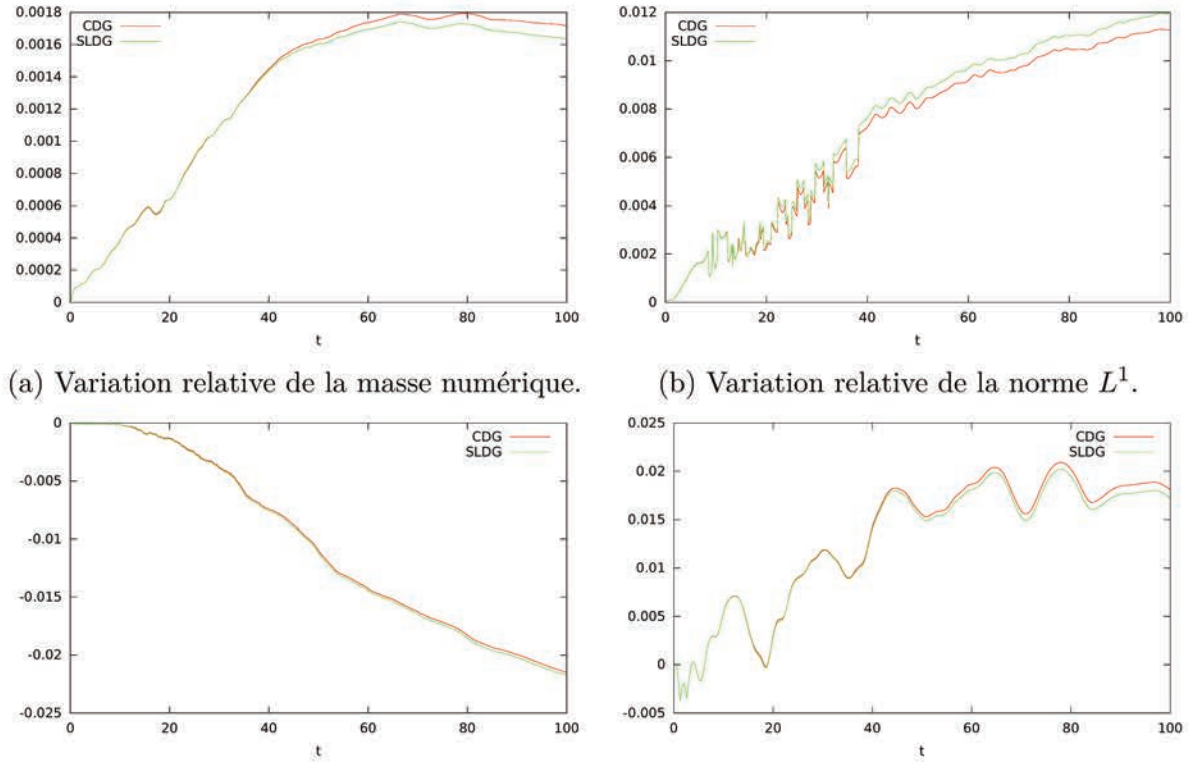
(h) Maillage à $t = 60$ pp.

FIGURE 6.10 – Évolution de la distribution et du maillage pour l'instabilité double faisceau par la méthode GDSL.



(i) Distribution à $t = 100$ pp. (j) Maillage à $t = 100$ pp.

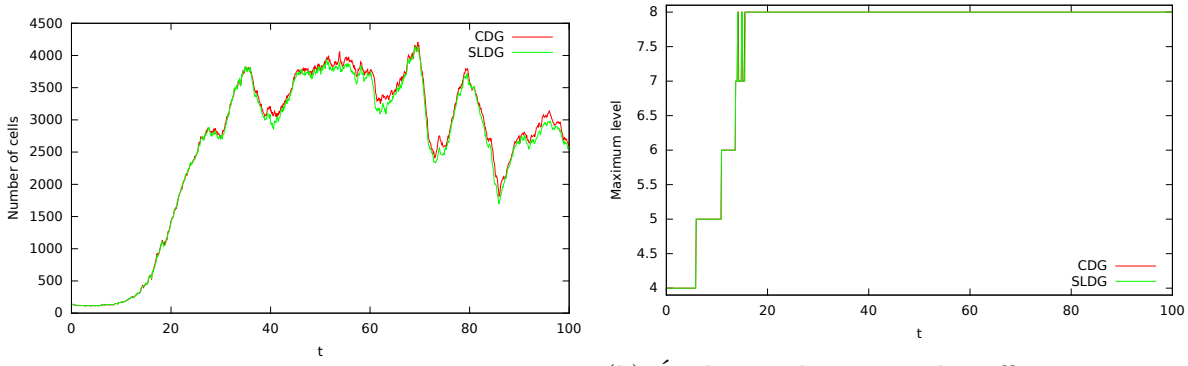
FIGURE 6.10 – Évolution de la distribution et du maillage pour l'instabilité double faisceau par la méthode GDSL.



(a) Variation relative de la masse numérique. (b) Variation relative de la norme L^1.

(c) Variation relative de la norme L^2. (d) Variation relative de l'énergie totale.

FIGURE 6.11 – Évolution des principaux diagnostiques pour l'instabilité double faisceau.



(a) Évolution du nombre de cellules.

(b) Évolution du niveau de raffinement maximum utilisé.

FIGURE 6.12 – Évolution du maillage pour l'instabilité double faisceau.

La figure 6.10 montre l'évolution de la distribution et le maillage associé obtenus par la méthode GDSL. Les résultats obtenus par la méthode CGD sont en tout point similaires. On voit la formation d'un tourbillon et de petites échelles à partir de $t = 20$ périodes plasma, ainsi que le raffinement du maillage au cours du temps pour suivre ces petites échelles.

La figure 6.11 indique les variations relatives de la masse numérique, des normes L^1 et L^2 et de l'énergie totale au cours du temps. L'écart entre la masse numérique et la norme L^1 indique que la distribution ne reste pas positive. Les discontinuités de la norme L^1 avant $t = 40$ périodes plasma indiquent des pics de minimum probablement liés à la fine structure que l'on observe sur la figure 6.10 à $t = 20$ périodes plasma. La norme L^2 décroit légèrement au cours du temps, ce qui indique une faible dissipation. Pour les quatre quantités observées on constate encore une fois la proximité des résultats donnés par les deux méthodes.

La figure 6.12 montre l'évolution du nombre de cellules et du niveau maximum de raffinement utilisé au cours du temps. On voit que le niveau de raffinement maximum est rapidement atteint avec la formation des fines structures déjà mentionnées. Le nombre de cellules en revanche continue d'augmenter sur un temps plus long puis oscille. Cela signifie que des détails fins apparaissent au cours du temps tandis que d'autres disparaissent, possiblement parce qu'ils deviennent trop fins pour être résolus sur la grille de niveau de raffinement maximum.

6.3 Bump-on-tail

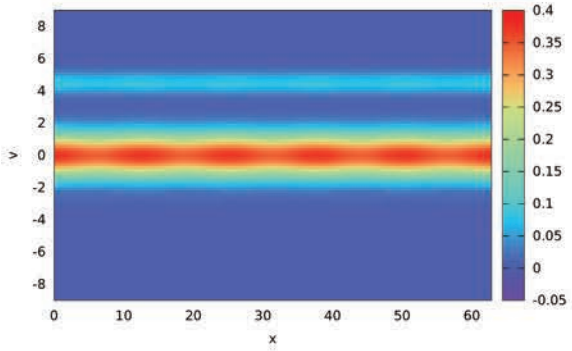
Le cas test dit bump-on-tail consiste à étudier un faisceau d'électrons avec une vitesse fixée au sein d'une population quasi statique. On considère les conditions initiales

$$f_0(x,v) = \frac{(1+0.04\cos(0.5x))}{10\sqrt{2\pi}}\left(9\exp\left(\frac{-v^2}{2}\right)+2\exp\left(-2(v-4.5)^2\right)\right),$$

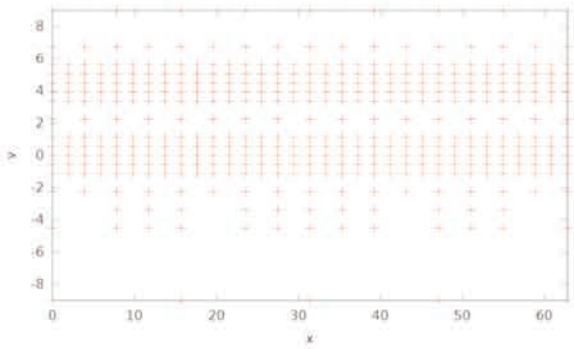
$$(x, v) \in [0\ ;\ 20\pi] \times [-9\ ;\ 9]. \quad (6.5)$$

Ce cas test est celui présenté dans [33] et dans [32]. Le temps final est $T = 300$ périodes plasma et le pas de temps est 0.1 périodes plasma, pour un total de 3000 pas de temps. Les polynômes utilisés sont de degré inférieur ou égal à 2 dans chaque direction. Le niveau maximum de raffinement autorisé est 8 et le seuil choisi est $\epsilon_0 = 3.10^{-3}$.

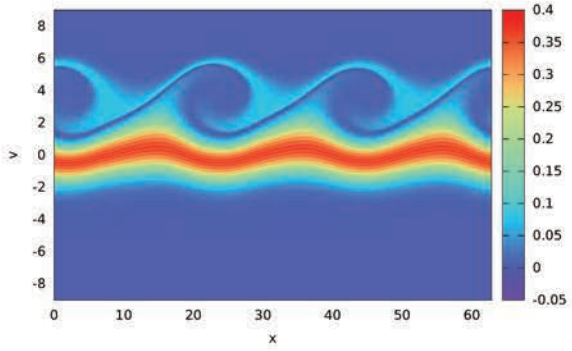
Nakamura et Yabe [33] présentent une résolution du problème par une méthode de propagation interpolée cubique sur maillage uniforme. C'est une méthode semi-lagrangienne nodale avec interpolation par des polynômes de degré 3. Dans [32], la décomposition spatiale se fait par la méthode Galerkin discontinue avec des polynômes de Lagrange aux points de Gauss-Lobatto. La résolution en temps se fait avec la méthode de Runge-Kutta à l'ordre 4.



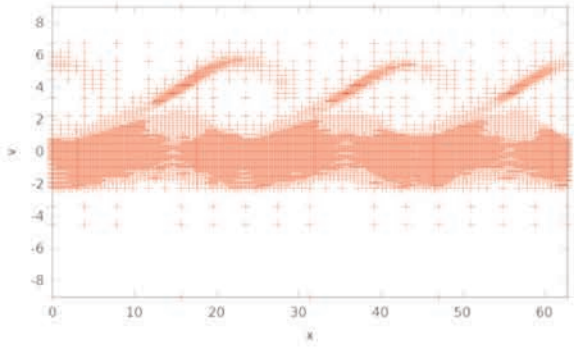
(a) Distribution initiale.



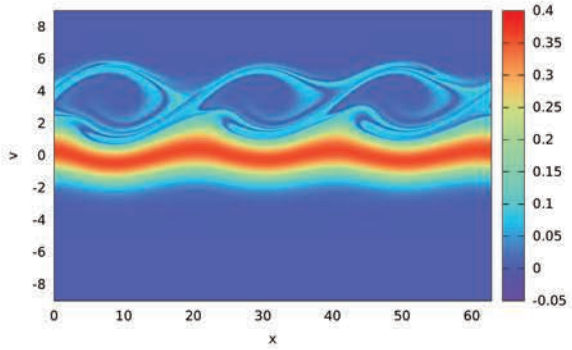
(b) Maillage initiale.



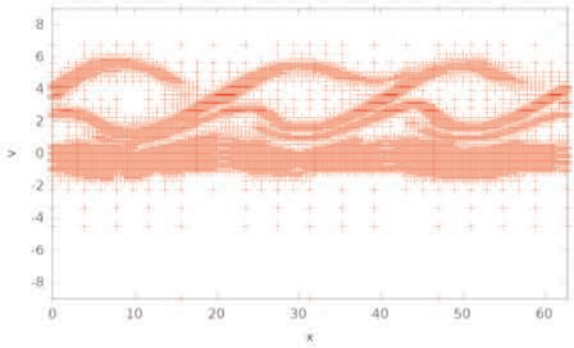
(c) Distribution à $t = 60$ pp.



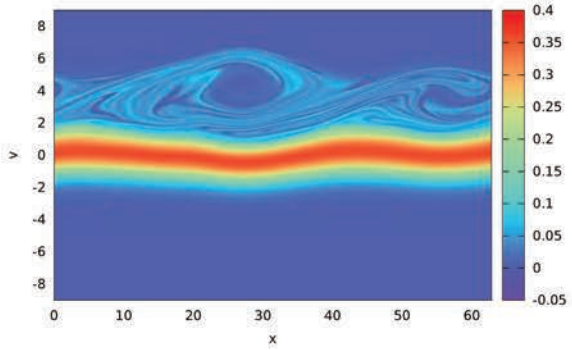
(d) Maillage à $t = 60$ pp.



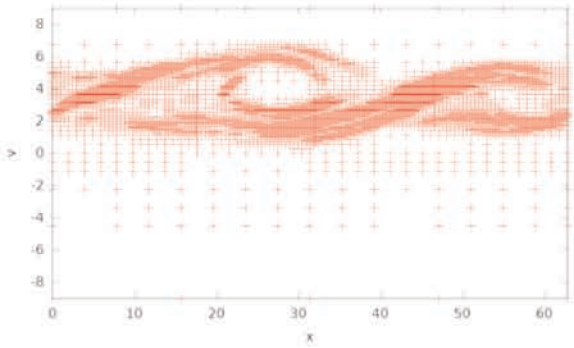
(e) Distribution à $t = 80$ pp.



(f) Maillage à $t = 80$ pp.

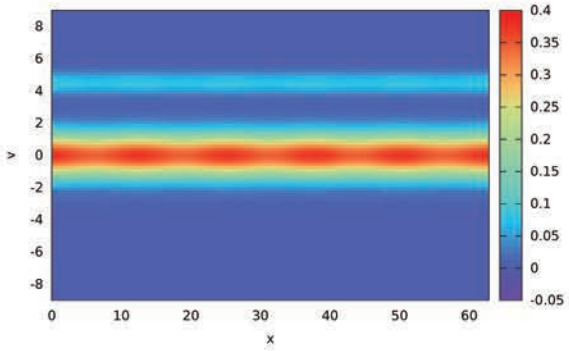


(g) Distribution à $t = 200$ pp.

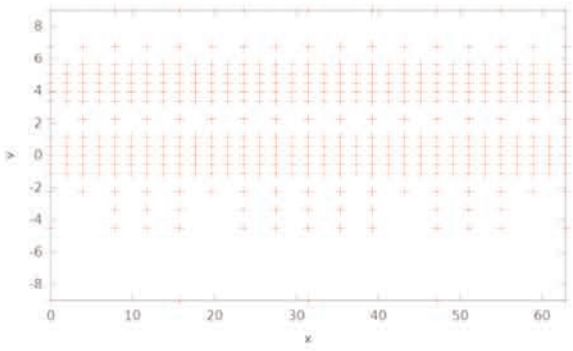


(h) Maillage à $t = 200$ pp.

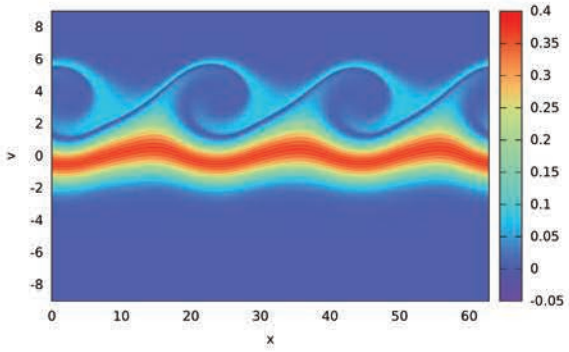
FIGURE 6.13 – Distribution et Maillage pour bump-on-tail avec CGD.



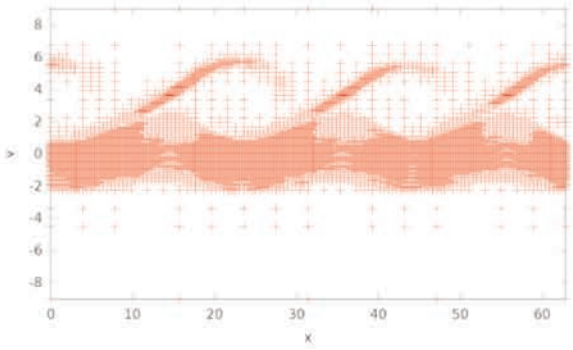
(a) Distribution initiale.



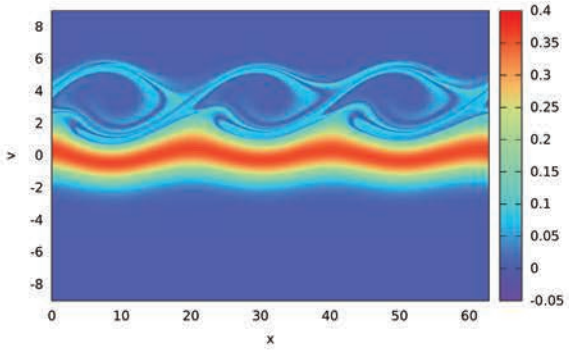
(b) Maillage initiale.



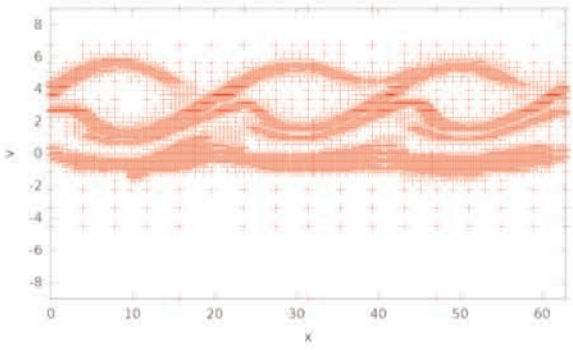
(c) Distribution à $t = 60$ pp.



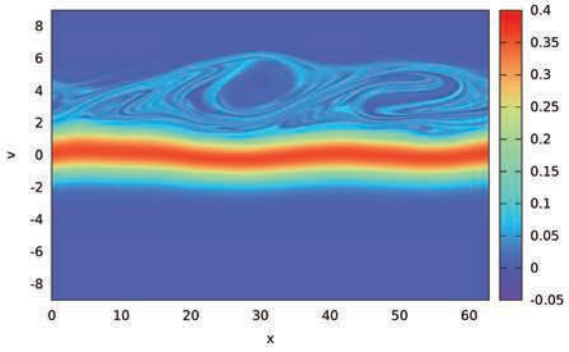
(d) Maillage à $t = 60$ pp.



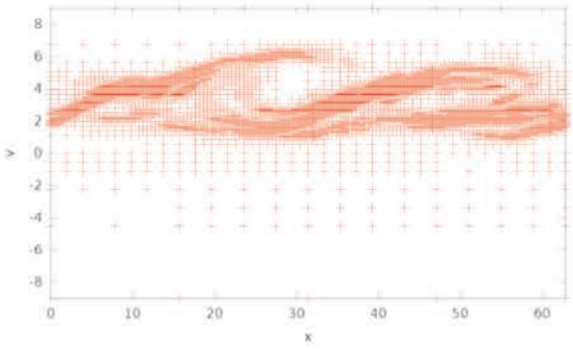
(e) Distribution à $t = 80$ pp.



(f) Maillage à $t = 80$ pp.

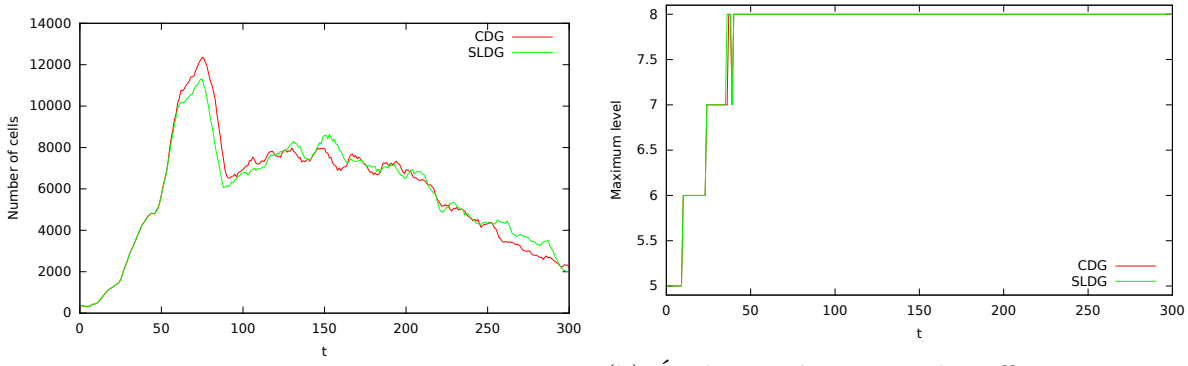


(g) Distribution à $t = 200$ pp.



(h) Maillage à $t = 200$ pp.

FIGURE 6.14 – Distribution et Maillage pour bump-on-tail avec GDSL.



(a) Évolution du nombre de cellules.

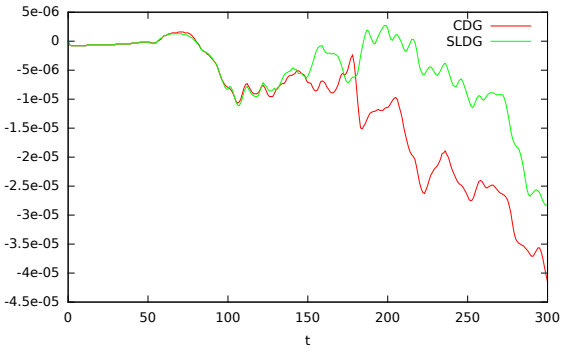
(b) Évolution du niveau de raffinement maximum utilisé.

FIGURE 6.15 – Évolution du maillage pour le cas bump-on-tail.

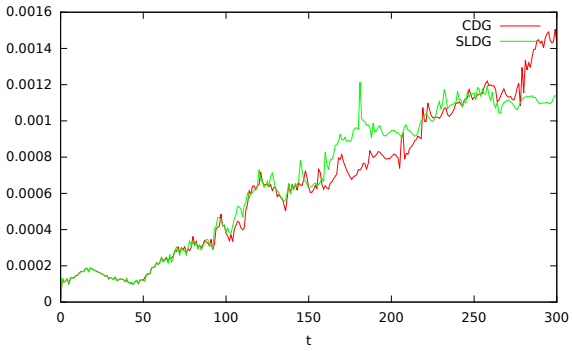
La figure 6.13 représente l'évolution temporelle de la distribution et du maillage obtenus avec la méthode CGD. La figure 6.14 est obtenue en utilisant la méthode GDSL. On observe un léger écart de la distribution et du maillage entre les deux méthodes, ce qui montre la sensibilité du cas test. On constate que pour les deux méthodes, que le maillage suit les détails des tourbillons lors de leur création et de leur existence (avant 100 périodes plasma) et lors de l'effondrement des tourbillons sous l'effet de la dissipation numérique sur des temps plus longs.

La figure 6.15 montre l'évolution temporelle du nombre de cellules feuilles et du niveau de raffinement le plus fin utilisé. On observe une augmentation rapide du nombre de cellules et du niveau de raffinement durant la création des tourbillons. Ensuite, lors de leur effondrement, les détails sont mois répartis, la distribution se lisse, et le nombre global de cellules feuilles diminue, bien que le niveau de raffinement maximum soit inchangé.

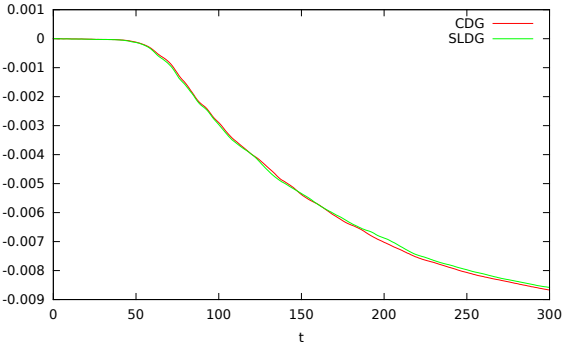
La figure 6.16 indique l'évolution relative de la masse numérique, des normes L^1 et L^2 et de l'énergie totale. On constate que toutes ces quantités sont très bien conservées jusqu'à 50 périodes plasma. Ce temps correspond à la création des trois tourbillons que l'on observe à $t = 80$ périodes plasma sur les figures 6.13 et 6.14. Néanmoins, la plus forte variation observée pour l'énergie totale est inférieure à 3%, ce qui reste faible. On constate que les variations de la masse numérique ne correspondent pas aux variations de la norme L^1. Les variations de la masse numérique sont d'ordre 10^{-5} et la masse numérique tend à diminuer alors que les variations de la norme L^1 sont d'ordre 10^{-3} et la norme L^1 augmente, ce qui signifie qu'il n'y a pas conservation de la positivité. La distribution numérique devient négative. La norme L^2 montre au temps final une diminution relative d'ordre 10^{-3}, ce qui indique une légère dissipation numérique. À l'exception de la masse numérique, on remarque également que les deux méthodes présentent des résultats très similaires.



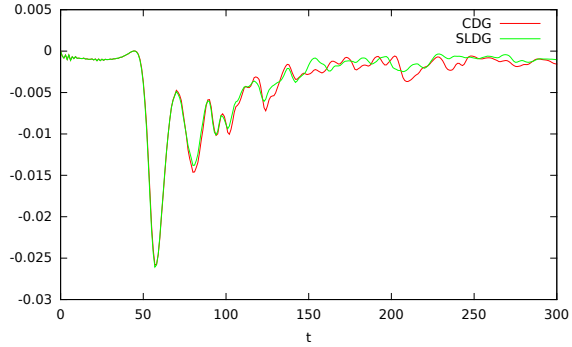
(a) Variation relative de la masse.



(b) Variation relative de la norme L^1.



(c) Variation relative de la norme L^2.



(d) Variation relative de l'énergie totale.

FIGURE 6.16 – Évolution des principaux diagnostiques pour le cas bump-on-tail.

6.4 Vlasov-Poisson pseudo-polaire

On s'intéresse dans cette section aux cas test de Vlasov-Poisson pseudo-polaires présentés dans [24] et dans [23]. Ces deux articles présentent des cas tests similaires résolus par des méthodes PIC. L'article [23] utilise un intégrateur exponentiel pour la résolution en temps, et l'article [24] analyse le cas test de faisceau focalisant. Ce sont des cas tests avec une importante filamentation et pour lesquels l'adaptatif est utile, au moins avant que les filaments ne soient répartis dans presque tout l'espace. La formulation de l'équation de Vlasov utilisée est alors

$$\partial_t f(r,v,t) + \partial_r\left(\frac{v}{\varepsilon} f(r,v,t)\right) + \partial_v\Big(\left(E_\varepsilon(r,t) + \Xi_\varepsilon(r,t)\right) f(r,v,t)\Big) = 0, \tag{6.6}$$

où la constante ε et les champs $E_\varepsilon(r,t)$ et $\Xi_\varepsilon(r,t)$ seront définis pour chaque cas test. L'équation de Poisson est également reformulée sous la forme

$$\frac{1}{r}\partial_r(rE_\varepsilon(r,t)) = \rho(r,t). \tag{6.7}$$

L'équation (6.7) a comme condition aux limites $E_\varepsilon(0,t) = 0$. Ceci est nécessaire afin de pouvoir la résoudre en $r = 0$.

Ces cas sont qualifiés de pseudo-polaires car si la formulation des équations de Vlasov-Poissons et les notations utilisées (nommer la variable d'espace r plutôt que x) rappellent

la formulation en coordonnées polaires du problème, la variable d'angle est absente (on reste sur une formulation 1D × 1V), et le problème est résolu sur des domaines en espace de type $\Omega_r = [-L_r\ ;\ L_r]$, donc notamment en $r = 0$. La variable d'angle est éliminée car la distribution initiale est invariante par rotation. Si la distribution initiale est invariante par rotation, l'invariance est conservée.

6.4.1 Avec un champ extérieur non auto-cohérent

Pour ce cas test, on prend $\varepsilon = 0.01$, $E_\varepsilon(r,t) = -r^3$ et $\Xi_\varepsilon(r,t) = -r/\varepsilon$. L'équation de Poisson n'intervient pas ici. La condition initiale est

$$f_0(r,v) = \frac{1}{\sqrt{2\pi}v_{th}} \exp\left(-\frac{v^2}{2v_{th}^2}\right) \mathbb{1}_{[-0.75\ ;0.75]}(r), \quad (r,v) \in [-1\ ;\ 1]^2, \tag{6.8}$$

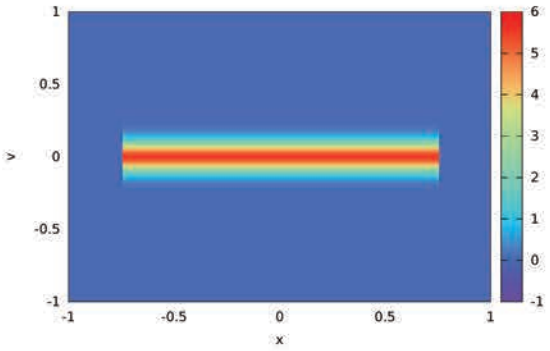
avec $v_{th} = 0.0727518214392$ et

$$\mathbb{1}_{[a\ ;b]}(r) = \begin{cases} 1 & \text{si } a \le r \le b, \\ 0 & \text{sinon.} \end{cases}$$

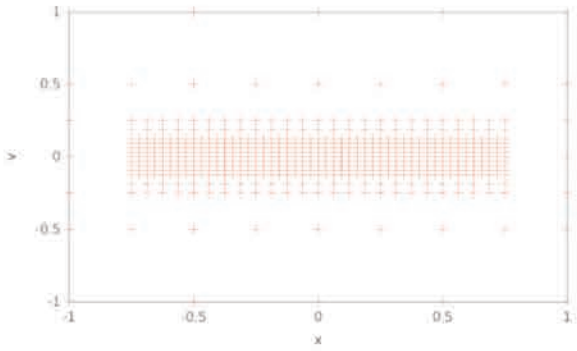
On considère un temps final $T = 100$ périodes plasma et un pas de temps de 0.1 périodes plasma, soit 1000 pas de temps. Les polynômes utilisés sont de degré inférieur ou égal à 2 dans chaque direction. Le niveau maximum de raffinement autorisé est 8 et le seuil choisi est $\epsilon_0 = 10^{-2}$.

La figure 6.17 montre la distribution et le maillage au cours du temps obtenus par la méthode GDSL. La méthode CGD donne des résultats extrêmement similaires. On peut suivre l'allongement des filaments et la diminution de l'espace entre les filaments aussi bien sur la distribution que sur le maillage, ce qui confirme la capacité des méthodes mises en œuvre à suivre des détails fins et localisés. La figure 6.18 reproduit de façon plus visible le maillage à $t = 80$ périodes plasma en utilisant des points aux sommets cellules. La figure 6.19 présente l'évolution du nombre de cellules feuilles au cours du temps. On voit que celui-ci augmente linéairement au cours du temps, à mesure que les filaments s'allongent et se resserrent, ce qui est cohérent avec les observations précédentes. Le niveau maximum utilisé n'est ici pas représenté car la distribution initiale est discontinue avec une discontinuité qui se situe à l'intérieur de plusieurs cellules. La distribution initiale est décrite avec 6 niveaux de raffinement mais dès le premier pas de temps des cellules de niveau 8 sont crées et ce niveau est conservé jusqu'à la fin de la simulation.

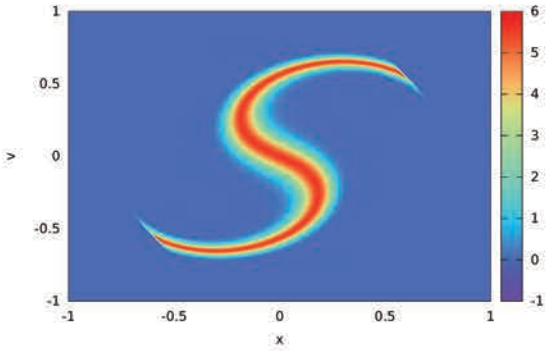
Sur la figure 6.20 sont tracées les variations relatives de la masse numérique et des normes L^1 et L^2. L'évolution du maximum et du minimum de la fonction de distribution est indiquée sur la figure 6.21. On remarque sur le minimum et sur le maximum les effets de la discontinuité projetée sur des polynômes dès le premier pas de temps. La valeur du minimum est importante, de l'ordre de -0.5, mais la masse numérique est très bien conservée, avec une variation relative de l'ordre de 10^{-4}. On constate une stabilisation du maximum à 5.5 (inférieure à la valeur de départ) autour de $t = 30$ périodes plasma suivie par une augmentation de la norme L^1 et une diminution de la norme L^2. La diminution de la norme L^2 indique la présence de dissipation numérique.



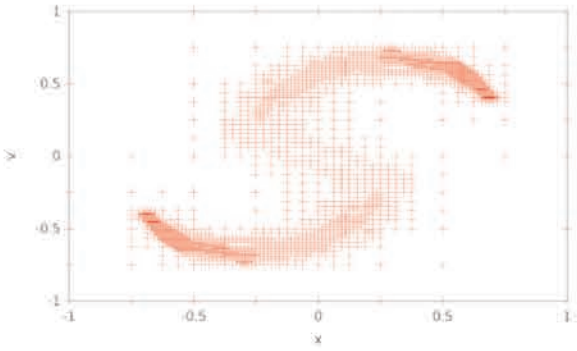
(a) Distribution initiale.



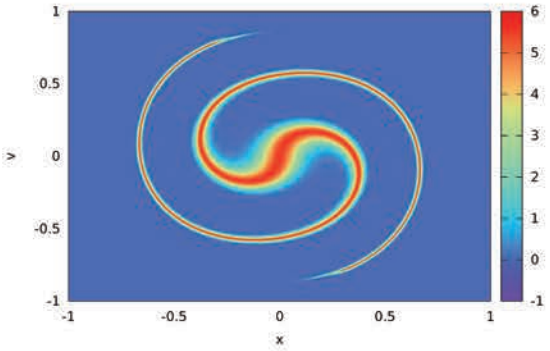
(b) Maillage initial.



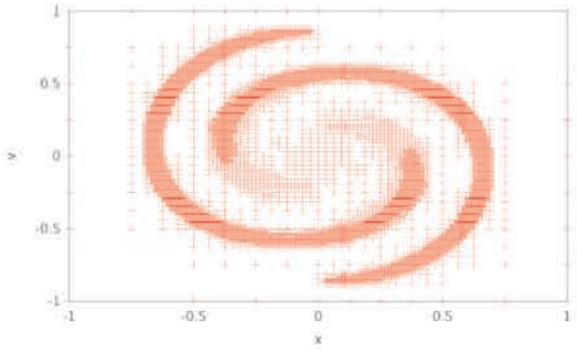
(c) Distribution à $t = 10$ pp.



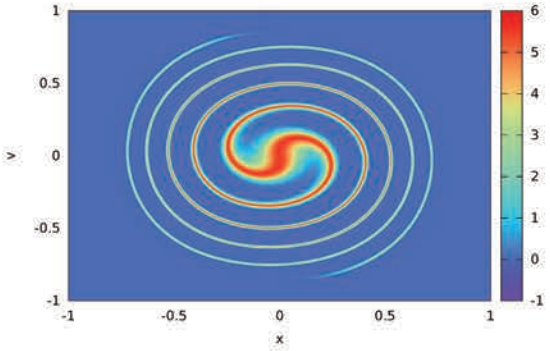
(d) Maillage à $t = 10$ pp.



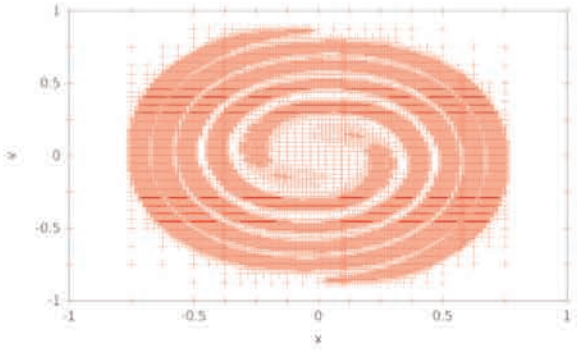
(e) Distribution à $t = 30$ pp.



(f) Maillage à $t = 30$ pp.



(g) Distribution à $t = 80$ pp.



(h) Maillage à $t = 80$ pp.

FIGURE 6.17 – Distribution et maillage pour le cas test pseudo-polaire avec champs forcés avec GDSL.

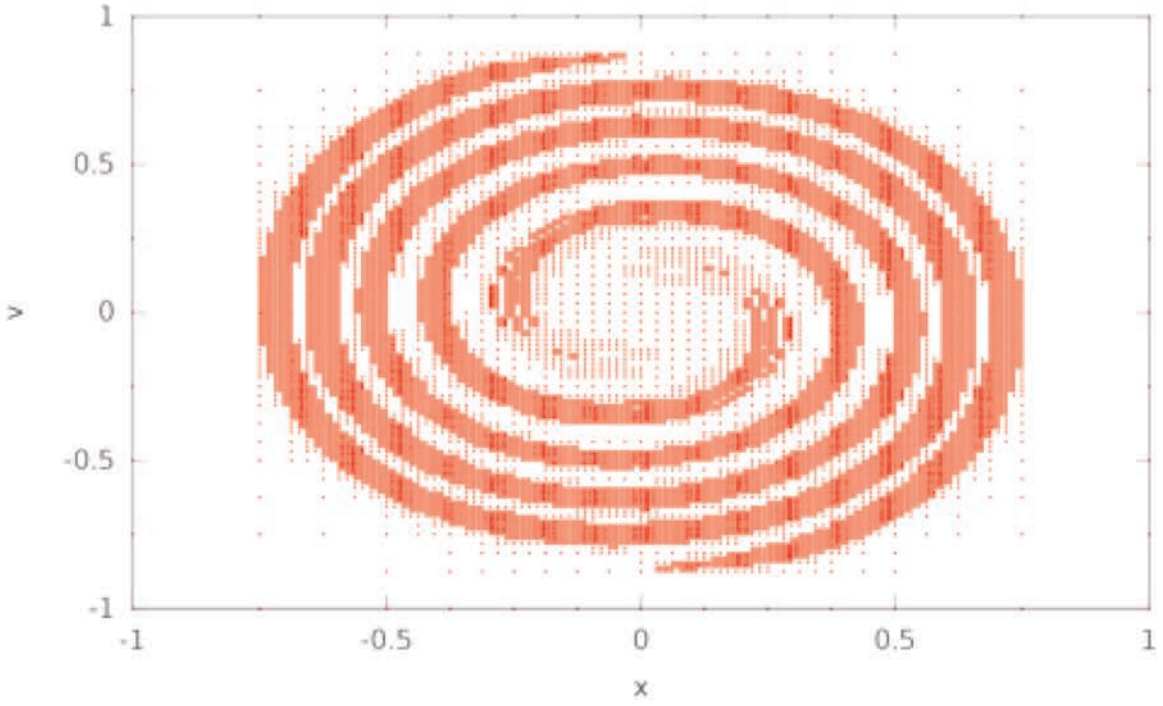


FIGURE 6.18 – Maillage à $t = 80$ pp pour le cas test pseudo-polaire avec champs forcés obtenu par la méthode GDSL.

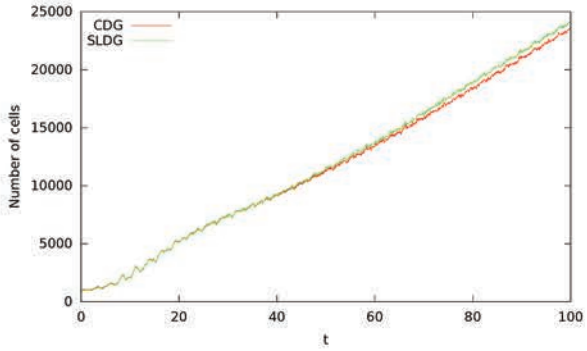
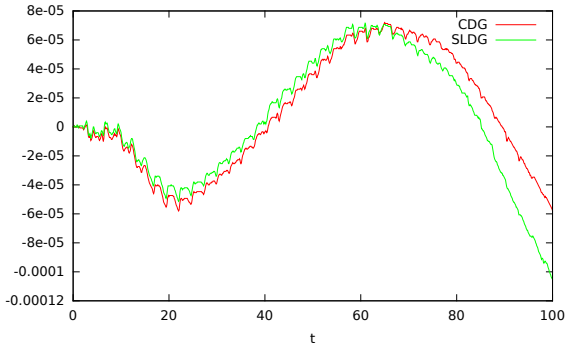
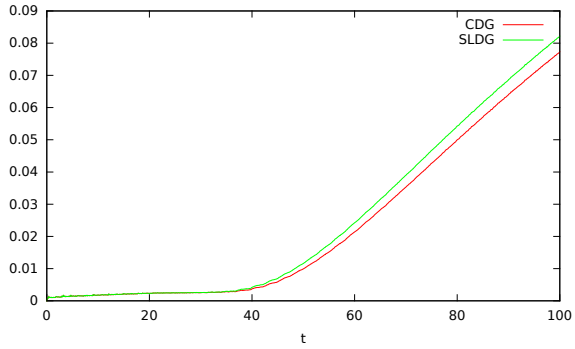


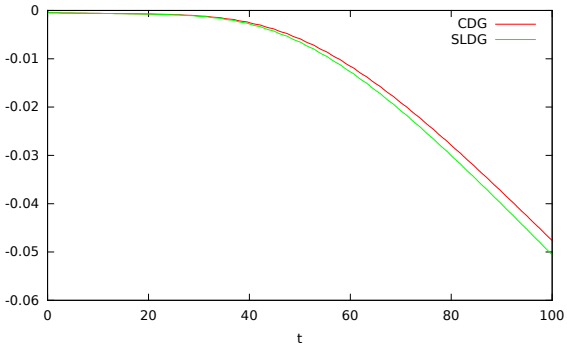
FIGURE 6.19 – Nombre de cellules pour le cas test pseudo-polaire avec champs forcés.



(a) Variation relative de la masse.

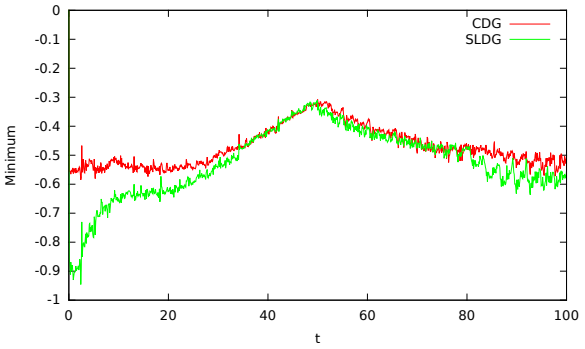


(b) Variation relative de la norme L^1.

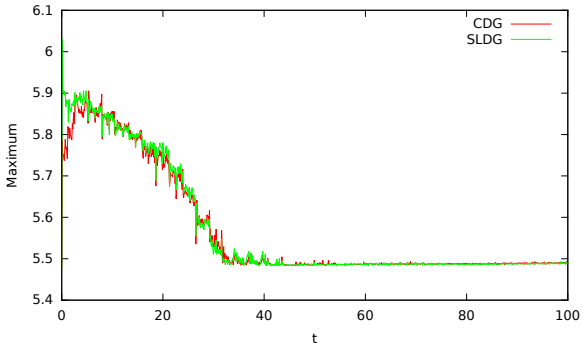


(c) Variation relative de la norme L^2.

FIGURE 6.20 – Évolution des principaux diagnostiques pour le cas pseudo-polaire avec champs forcés.



(a) Minimum de la distribution.



(b) Maximum de la distribution.

FIGURE 6.21 – Évolution du minimum et du maximum pour le cas test pseudo-polaire avec champs forcés.

6.4.2 Avec un champ auto-cohérent

On considère dans ce cas test $\varepsilon = 0.01$, $E_\varepsilon(r,t)$ est obtenu en résolvant l'équation de Poisson (6.7), et $\Xi_\varepsilon(r,t) = -r/\varepsilon$. La condition initiale est

$$f_0(r,v) = \frac{1}{\sqrt{2\pi}v_{th}} \exp\left(-\frac{v^2}{2v_{th}^2}\right) \mathbb{1}_{[-0.75\ ;0.75]}(r), \quad (r,v) \in [-1\ ;\ 1] \times [-1.2\ ;\ 1.2], \tag{6.9}$$

avec $v_{th} = 0.0727518214392$. Il s'agit de la même condition initiale que pour le cas de champs forcés avec simplement un domaine légèrement étendu en vitesse afin de satisfaire les conditions de Dirichlet homogènes aux bords, $f(x, \pm 1.2, t) = 0$.

On considère un temps final $T = 100$ périodes plasma et un pas de temps de 0.1 période plasma pour un total de 1000 pas de temps. Le système est résolu en utilisant des polynômes de degré inférieur ou égal à 2 dans chaque direction. Le niveau de raffinement maximum est 8 et le seuil choisi est $\epsilon_0 = 3.10^{-3}$.

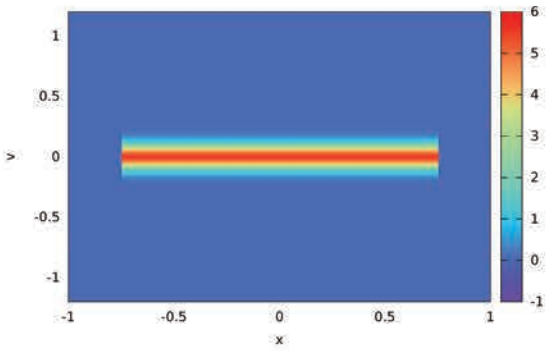
La distribution et le maillage obtenus par la méthode CGD sont représentés sur la figure 6.22. La méthode GDSL présente pour ce cas test une spirale dont l'extérieur prend la forme d'un hexagone. Cette forme particulière est probablement la conséquence d'un pas de temps moins adapté à la méthode. Bien que les conditions initiales soient presque les mêmes que dans le cas avec champs forcés, section 6.4.1, l'utilisation d'un seuil plus petit implique un maillage initial plus raffiné. Le maillage suit toujours la formation des filaments, mais ceux-ci étant plus resserrés que dans le cas des champs forcés et l'utilisation d'un seuil plus petit font que le maillage tend finalement vers un maillage uniformément raffiné. L'augmentation régulière du nombre de cellules feuilles présentée sur la figure 6.23 est cohérente avec l'évolution de la distribution présentée sur la figure 6.22.

La figure 6.24 montre la variation relative de la masse numérique et des normes L^1, L^2 et L^∞ pour ce cas test. La comparaison des évolutions de la variation de masse et de la variation de la norme L^1 indique que la distribution ne reste pas positive. Les deux schémas numériques conservent ici aussi la masse (variation relative de l'ordre de 10^{-4}) mais pas la norme L^1. La décroissance de la norme L^2 indique la présence de dissipation numérique. L'évolution de la norme L^∞ est frappante par l'augmentation extrêmement importante dès le premier pas de temps puis la baisse régulière observée pour revenir à la valeur initiale. Si on regarde la distribution initiale, on constate que la discontinuité à $|x| = 0.75$ coïncide avec la frontière des cellules de niveau 3 et plus. Par conséquent, il n'y a initialement pas d'oscillations polynomiales au temps initial. En revanche, dès le premier pas de temps, la discontinuité ne coïncide plus avec le bord des cellules et des oscillations apparaissent. Ces oscillations disparaissent ensuite avec la formation des filaments et la dissipation numérique.

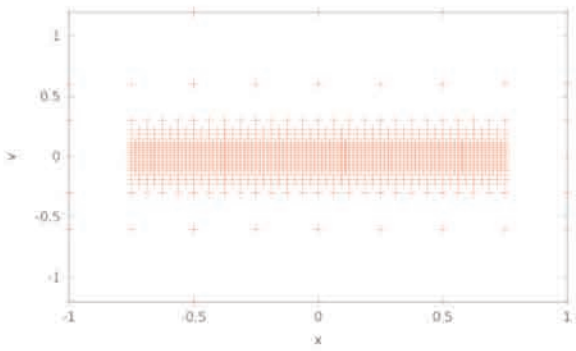
6.4.3 Faisceau focalisant

Le cas test de faisceau focalisant est expliqué et détaillé dans [24]. On considère $\varepsilon = 0.1$, $E_\varepsilon(r,t)$ est calculé en résolvant l'équation de Poisson (6.7) et

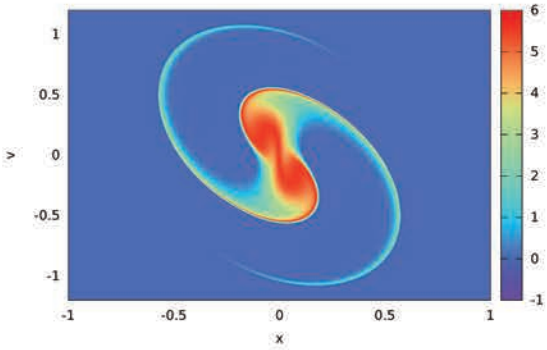
$$\Xi_\varepsilon(r,t) = r\left(\frac{-1}{\varepsilon} + \cos^2\left(\frac{t}{\varepsilon}\right)\right).$$



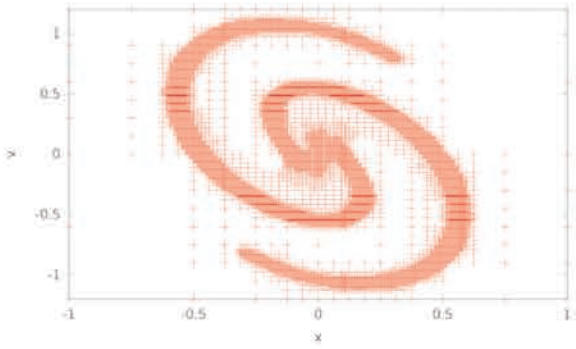
(a) Distribution initiale.



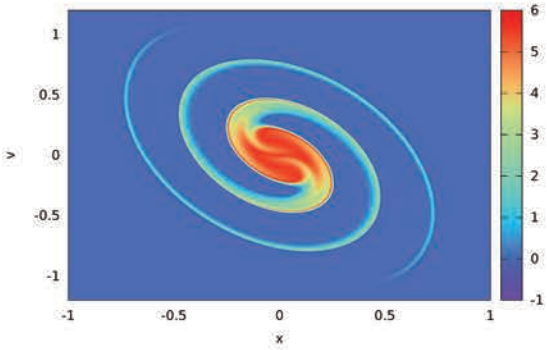
(b) Maillage initial.



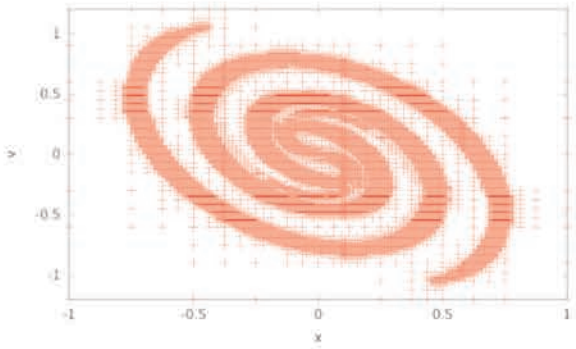
(c) Distribution à $t = 30$ pp.



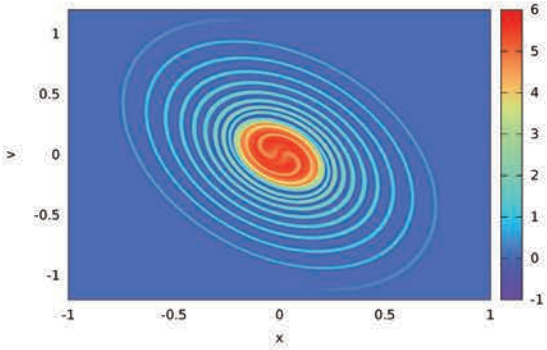
(d) Maillage à $t = 30$ pp.



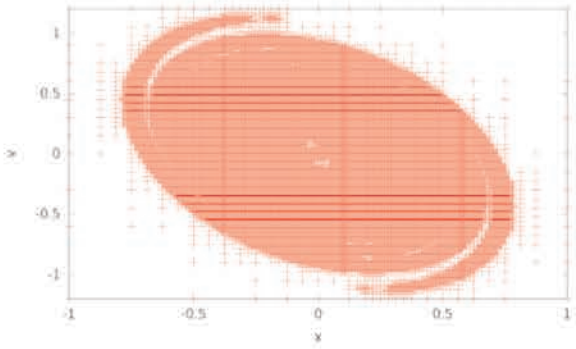
(e) Distribution à $t = 40$ pp.



(f) Maillage à $t = 40$ pp.



(g) Distribution à $t = 100$ pp.



(h) Maillage à $t = 100$ pp.

FIGURE 6.22 – Distribution et Maillage pour le cas Vlasov-Poisson pseudo-polaire avec CGD.

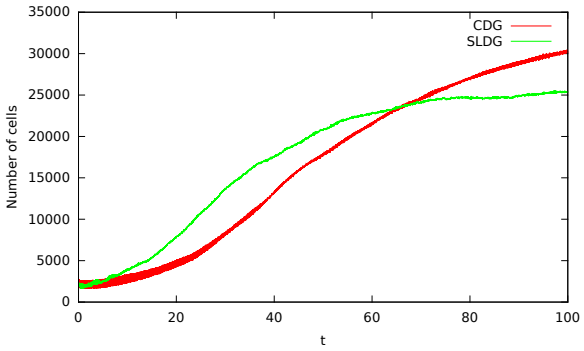


FIGURE 6.23 – Nombre de cellules pour le cas test Vlasov-Poisson pseudo-polaire.

La condition initiale est

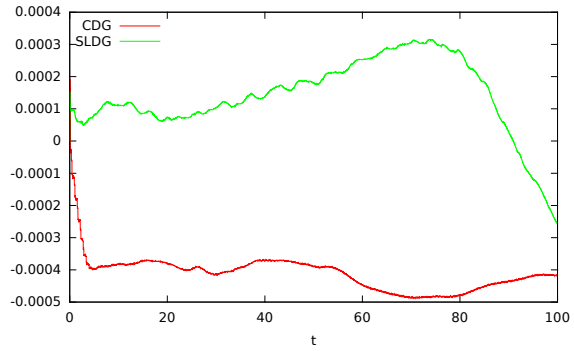
$$f_0(r,v) = \frac{3}{4v_{th}} \exp\left(\frac{-v^2}{2v_{th}^2}\right) \mathbb{1}_{[-1.83271471003\ ;\ 1.83271471003]}(r), \quad (r,v) \in [-3\ ;\ 3]^2, \tag{6.10}$$

avec $v_{th} = 0.0727518214392$.

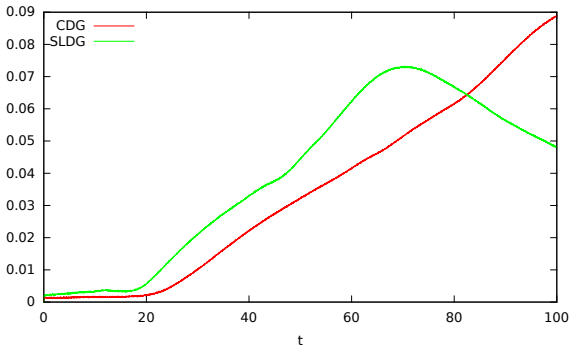
Le temps final considéré est $T = 100$ périodes plasma avec un pas de temps de 0.02 périodes plasma pour un total de 5000 pas de temps. Le pas de temps pour ce cas test est très court comparé au pas de temps de cas test précédents car le champ $\Xi_\varepsilon(r,t)$ varie rapidement en temps et il est nécessaire que ses variations soient bien représentées au niveau des caractéristiques pour que les résultats obtenus aient un sens. Une alternative possible serait d'utiliser des schémas d'ordres plus élevés pour le calcul des caractéristiques. Les polynômes utilisés sont de degré inférieur ou égal à 2 dans chaque direction. On utilise 8 niveaux de raffinement et un seuil $\epsilon_0 = 10^{-2}$.

La figure 6.25 montre l'évolution temporel du maillage et de la distribution obtenus par la méthode GDSL. La distribution et le maillage obtenus par la méthode CGD sont indiscernables de ceux présentés ici. Le maillage suit très bien les tourbillons qui se forment aux extrémités du faisceau initial, le filament qui les relie et les deux filaments qui se forment à l'extérieur. La distribution et le maillage ne sont présentés que jusqu'à $t = 20$ périodes plasma parce qu'au-delà les oscillations polynomiales et la dissipation numérique nuisent à la qualité des résultats, bien que le maillage suive toujours la distribution. L'évolution du nombre de cellules feuilles est tracé sur la figure 6.26. Le nombre de cellules augmente régulièrement jusqu'à ce que la dissipation devienne trop importante et les résultats non physiques. Il est tout de même intéressant de constater que même en temps long, les deux méthodes continuent de produire des résultats similaires.

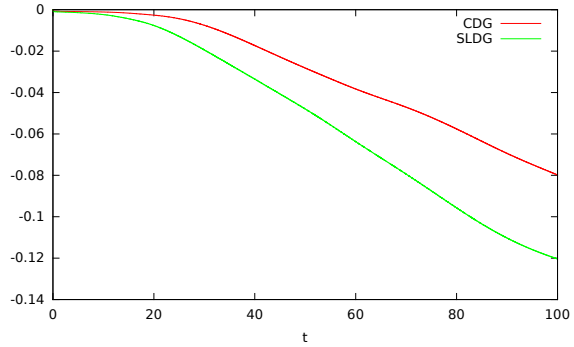
La figure 6.27 représente l'évolution temporelle de la masse numérique, des normes L^1, L^2 et L^∞ et du minimum de la fonction de distribution. On remarque une chute brutale de la masse numérique, environ 3% entre $t = 37$ et $t = 38$ périodes plasma. Cette chute s'observe avec la même intensité sur la norme L^1. On observe simultanément une diminution de la norme L^2, mais aucune variation sensible du minimum et du maximum de la fonction de distribution. Cette chute semble être liée à la dissipation brutale de certains filaments. La norme L^2 décroit beaucoup plus que pour les autres cas tests, près de 30% en moins à la fin de la simulation. La distribution admet des minimums très bas



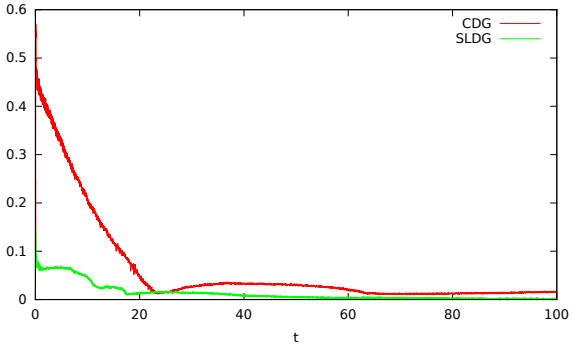
(a) Variation relative de la masse numérique.



(b) Variation relative de la norme L^1.



(c) Variation relative de la norme L^2.



(d) Variation relative de la norme L^∞.

FIGURE 6.24 – Évolution des principaux diagnostiques pour le cas test Vlasov-Poisson pseudo-polaire.

avec moins d'un ordre de grandeur d'écart avec le maximum initial. Le pas de temps choisi (2.10^{-2}) permet un suivi en temps acceptable du champ $\Xi_\varepsilon(r,t)$.

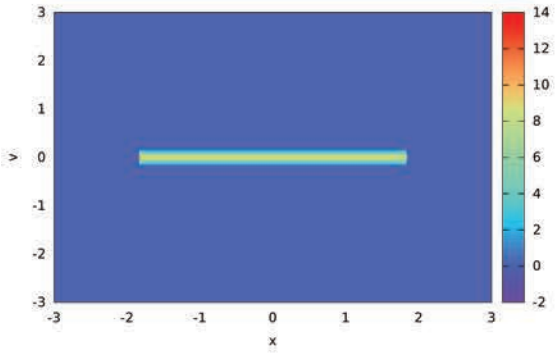
6.5 Cas test d'astrophysique

On s'intéresse dans cette section à des cas tests issus de l'astrophysique (2.4) et (2.7) plutôt que de la physique des plasmas. Les cas tests de couche froide perturbée, sections 6.5.1 et 6.5.2, souffrent d'effets indésirables dues aux conditions périodiques en espace. Le cas test avec comme condition initiale une gaussienne, section 6.5.3, est adapté pour éviter les artefacts numériques liés à la périodicité. Le temps n'est pas exprimé en périodes plasma mais en unité de temps astrophysique, abrégé uta, puisque l'on travaille toujours avec des équations adimensionnées.

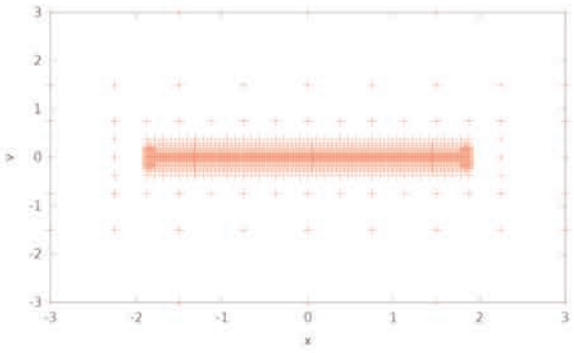
6.5.1 Couche froide avec perturbation en espace

On considère une fonction de distribution initiale semblable au cas test d'amortissement Landau :

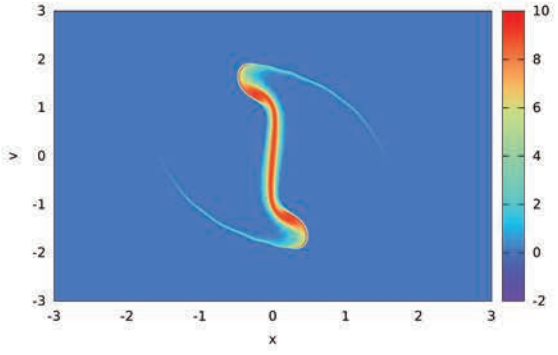
$$f_0(x,v) = \frac{(1-0.1\cos(x))}{\sqrt{2\pi}v_{th}} \exp\left(-\frac{v^2}{2v_{th}^2}\right), \quad (x,v) \in [0\ ;\ 2\pi] \times [-10\ ;\ 10], \tag{6.11}$$



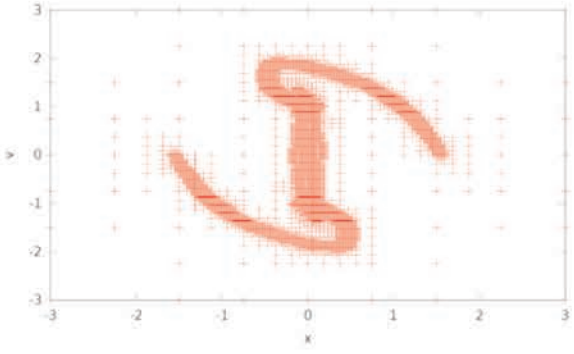
(a) Distribution initiale.



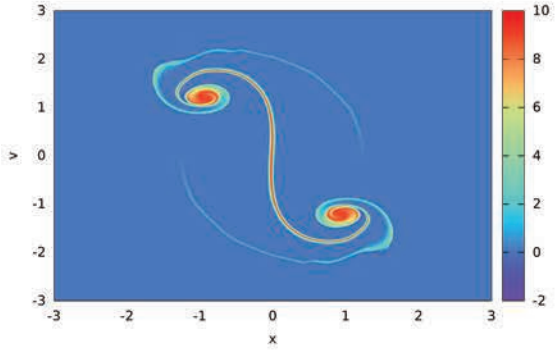
(b) Maillage initial.



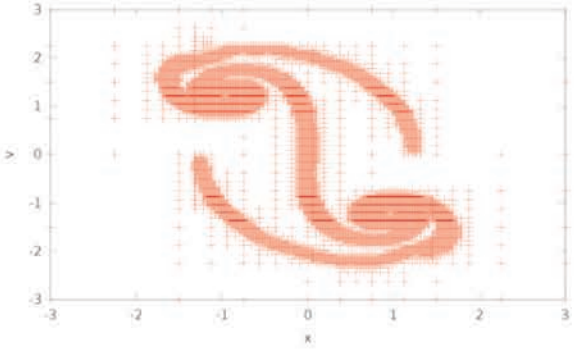
(c) Distribution à $t = 10$ pp.



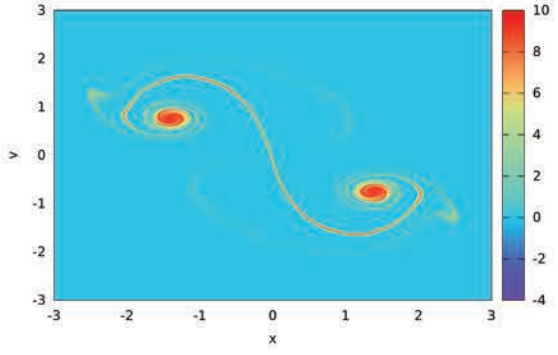
(d) Maillage à $t = 10$ pp.



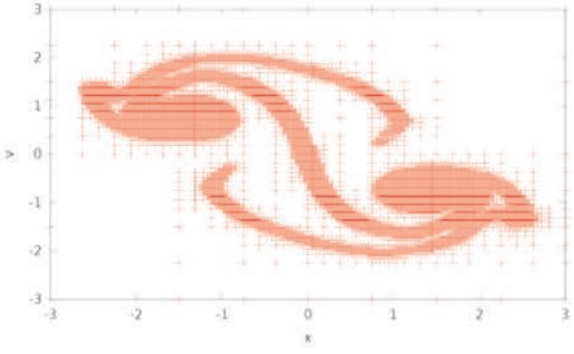
(e) Distribution à $t = 16$ pp.



(f) Maillage à $t = 16$ pp.



(g) Distribution à $t = 20$ pp.



(h) Maillage à $t = 20$ pp.

FIGURE 6.25 – Distribution et maillage pour le faisceau focalisant avec SLDG.

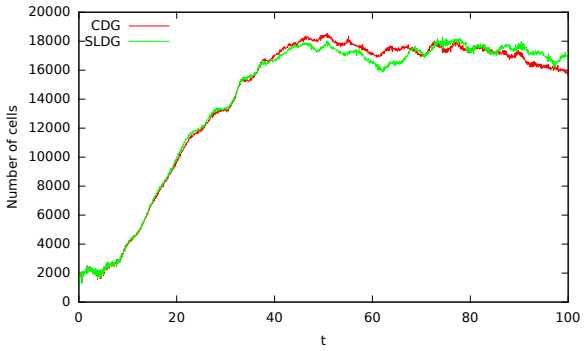


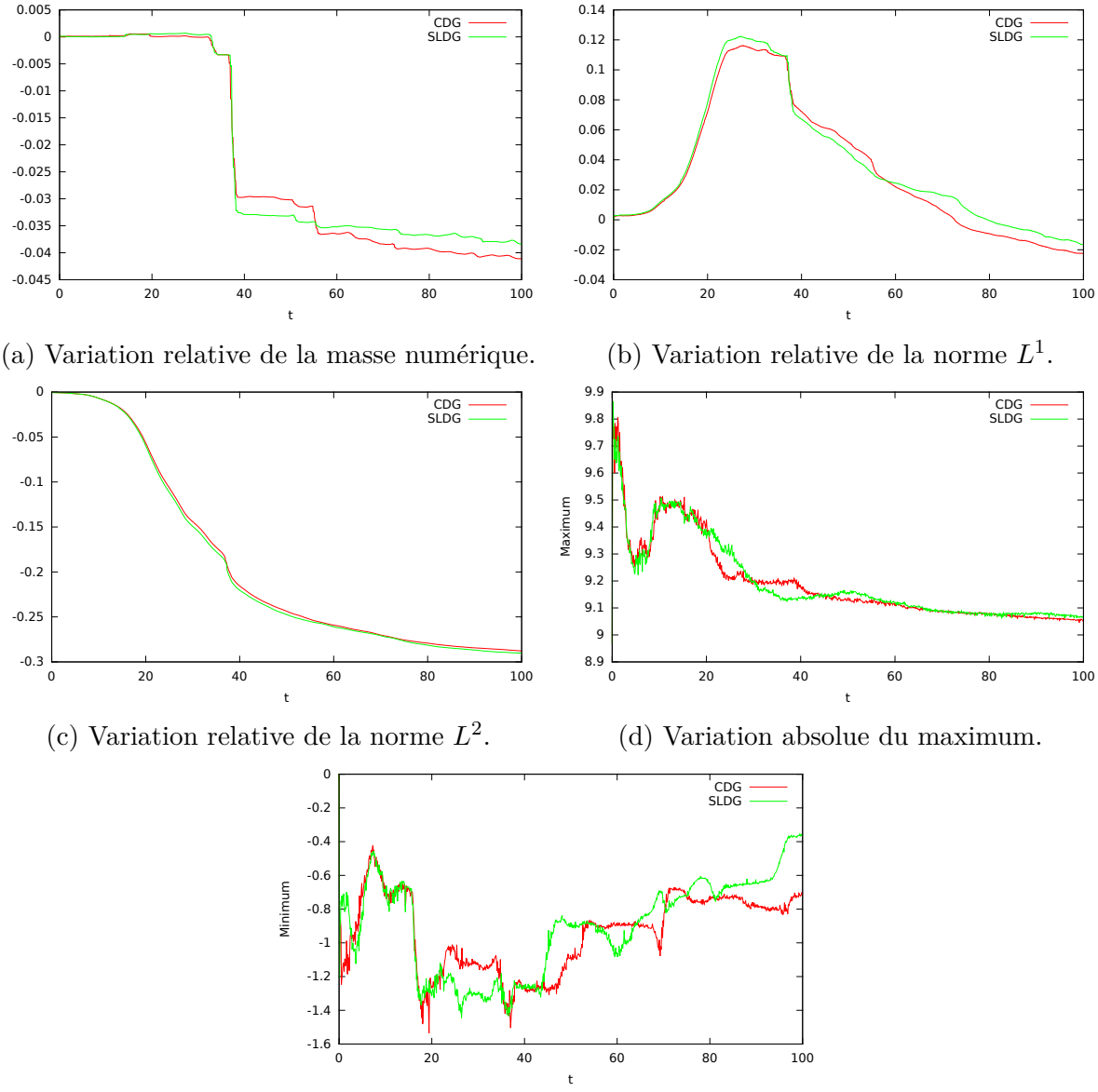
FIGURE 6.26 – Nombre de cellules pour le cas test faisceau focalisant.

avec $v_{th} = 0.15$.

Le temps final considéré est $T = 100$ unités de temps avec un pas de temps de 0.02 unité de temps pour un total de 5000 pas de temps. Les polynômes sont de degré inférieur ou égal à 2 dans chaque direction. On utilise 9 niveaux de raffinements et un seuil $\epsilon_0 = 3.10^{-3}$. Ce cas test est conçu pour toujours avoir des cellules au niveau de raffinement le plus fin autorisé.

La figure 6.28 représente la distribution et le maillage obtenus par la méthode CGD. Les simulations utilisant la méthode GDSL donnent les mêmes profils de distribution et des maillages semblables. Jusqu'à $t = 2$ uta on voit la fonction de distribution s'enrouler en spiral. À $t = 2$ uta on voit que les filaments commencent à être moins bien définis à cause de l'erreur d'approximation numérique. Dès $t = 3$ uta on observe les effets de la périodicité avec les filaments qui sont sortis d'un côté de la boîte pour entrer de l'autre côté et continuent le processus de filamentation. Ce phénomène, ainsi que l'augmentation de l'erreur de calcul des filaments sont visibles jusqu'à $t = 6$ uta. Les oscillations visibles sur les filaments à $t = 4$ uta et $t = 6$ uta sont dues à une mauvaise résolution en temps du problème. Ce sont les mêmes oscillations que celles observées lors de la résolution de l'équation de Burgers. À partir de $t = 8$ uta on observe une dégradation rapide de la résolution des filaments loin du centre, suivie d'un lissage de la distribution qui se poursuit jusqu'à la fin de la simulation. La filamentation puis la dissipation sont également visibles sur la figure 6.29. On constate que le nombre de cellules augmente très rapidement tant que dure le processus de filamentation puis qu'il diminue graduellement durant le lissage de la fonction de distribution et la dissipation des détails fins.

Il est important de rappeler qu'à partir du moment ou la périodicité intervient (entre $t = 2$ et $t = 3$ uta) ce cas test n'est plus physique. L'évolution de la masse numérique des normes L^1, L^2 et L^∞, de l'énergie totale et de la quantité de mouvement n'aurait aucun sens. Les variations relatives de la masse et des normes L^1, L^2 et L^∞ sont néanmoins tracées sur la figure 6.30 jusqu'à $t = 20$ uta, c'est à dire jusqu'au début du lissage de la distribution. Ce cas test nous montre simplement une fois de plus la capacité du maillage à suivre des détails fins et localisés.



(a) Variation relative de la masse numérique.

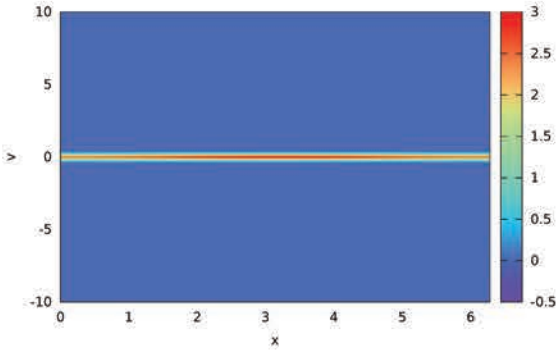
(b) Variation relative de la norme L^1.

(c) Variation relative de la norme L^2.

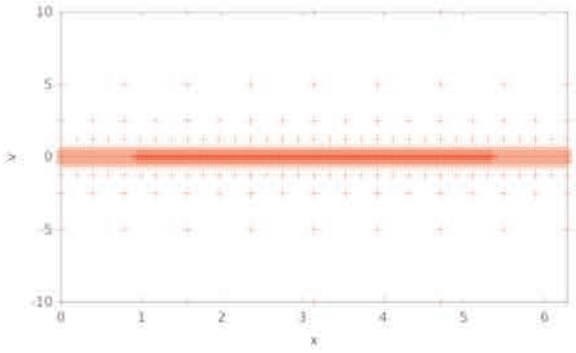
(d) Variation absolue du maximum.

(e) Variation absolue du minimum.

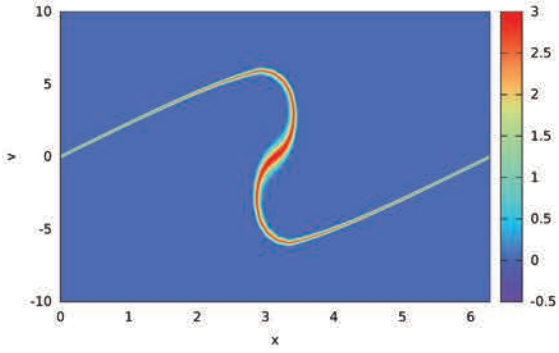
FIGURE 6.27 – Évolution des principaux diagnostiques pour le cas test du faisceau focalisant.



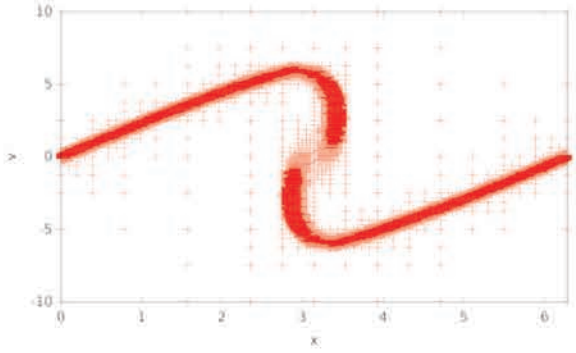
(a) Distribution initiale.



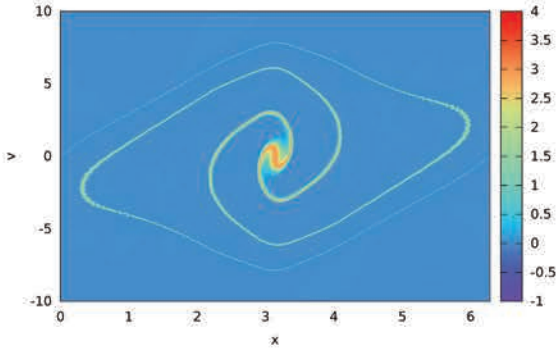
(b) Maillage initial.



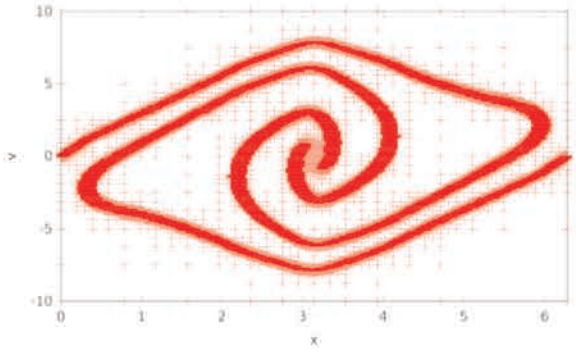
(c) Distribution à $t = 1$ uta.



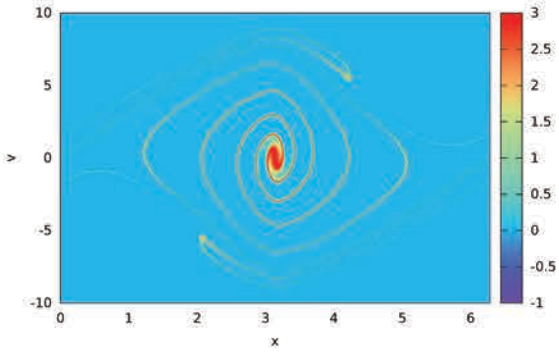
(d) Maillage à $t = 1$ uta.



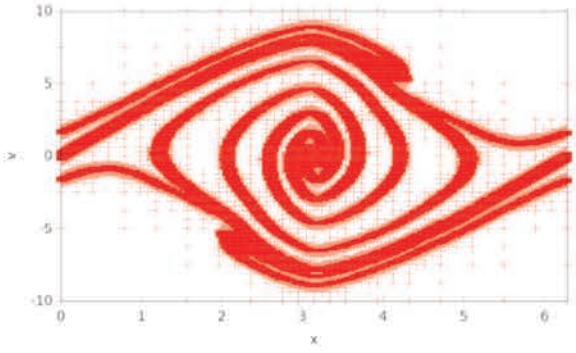
(e) Distribution à $t = 2$ uta.



(f) Maillage à $t = 2$ uta.

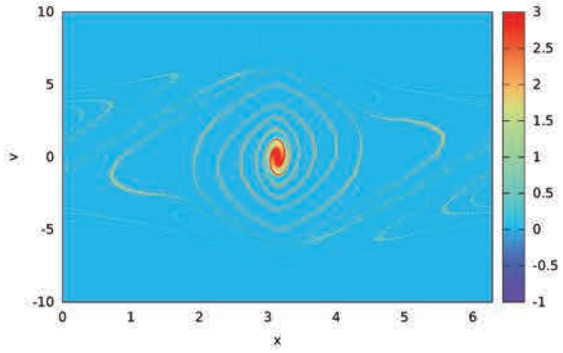


(g) Distribution à $t = 3$ uta.

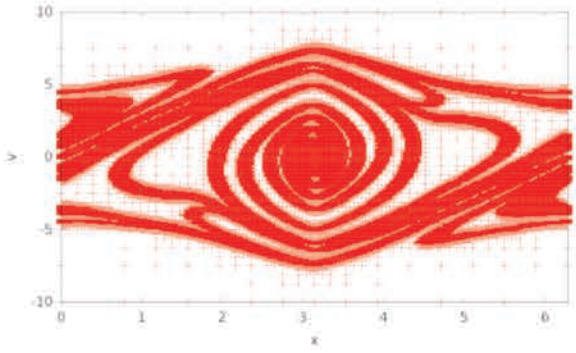


(h) Maillage à $t = 3$ uta.

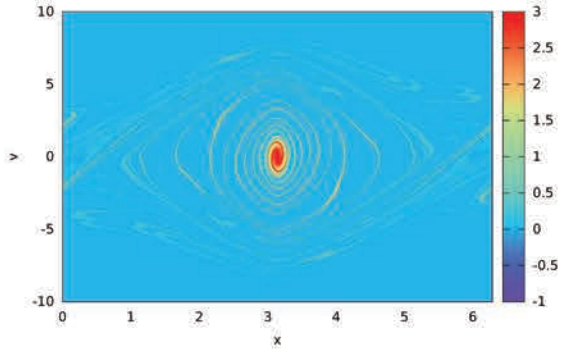
FIGURE 6.28 – Distribution et maillage pour le cas astrophysique de couche froide avec perturbation en espace par la méthode CGD.



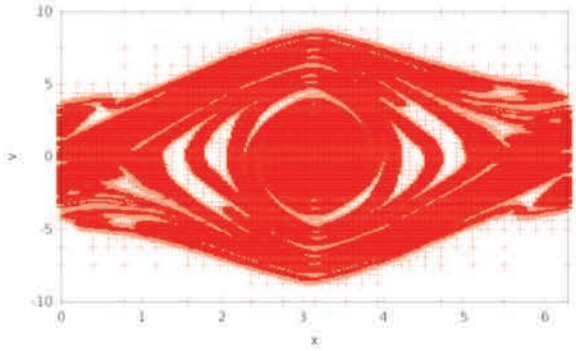
(i) Distribution à $t = 4$ uta.



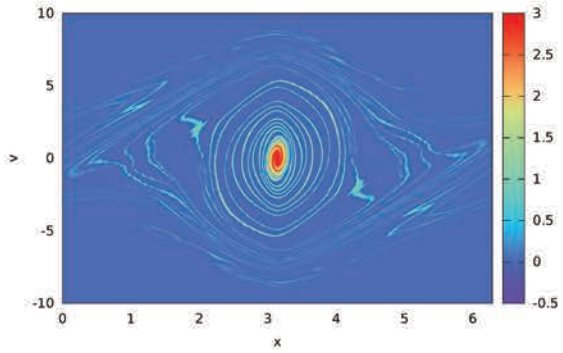
(j) Maillage à $t = 4$ uta.



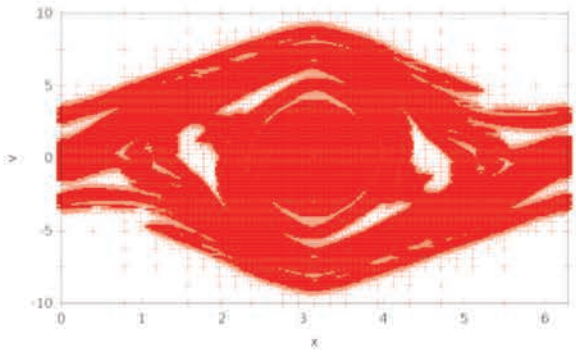
(k) Distribution à $t = 6$ uta.



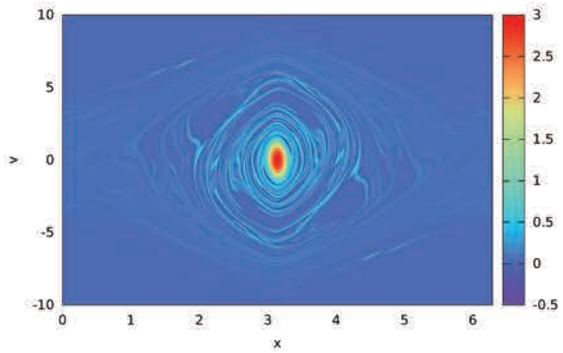
(l) Maillage à $t = 6$ uta.



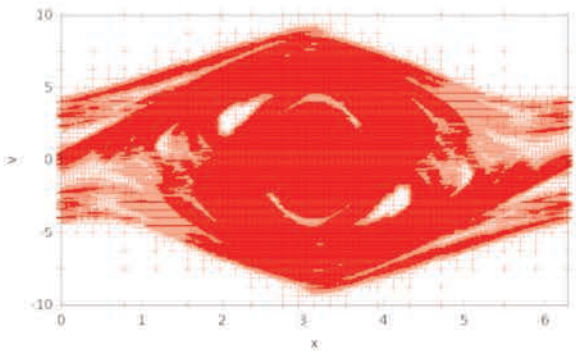
(m) Distribution à $t = 8$ uta.



(n) Maillage à $t = 8$ uta.



(o) Distribution à $t = 20$ uta.



(p) Maillage à $t = 20$ uta.

FIGURE 6.28 – Distribution et maillage pour le cas astrophysique de couche froide avec perturbation en espace par la méthode CGD.

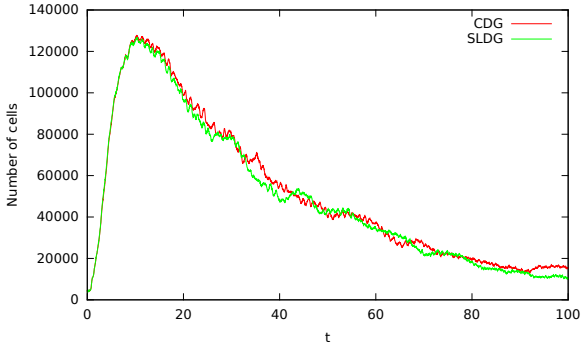
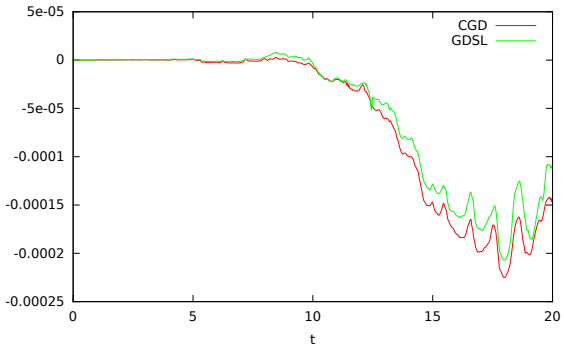
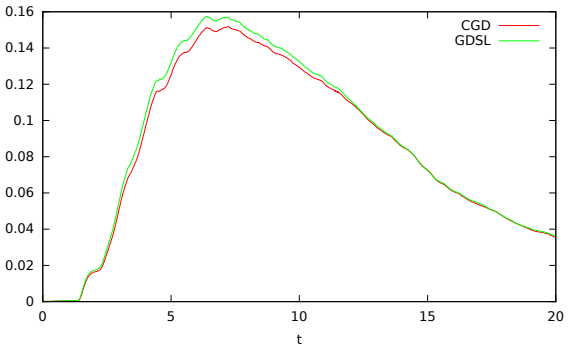


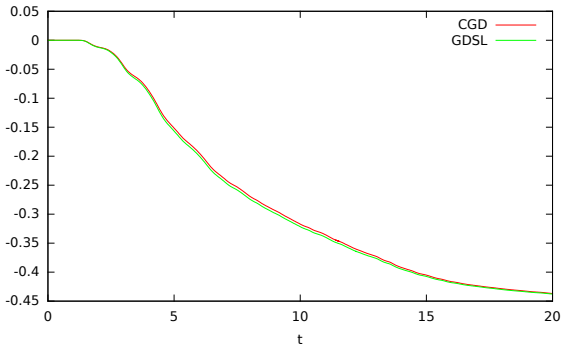
FIGURE 6.29 – Évolution du nombre de cellules pour le cas test astrophysique de couche froide avec perturbation en espace.



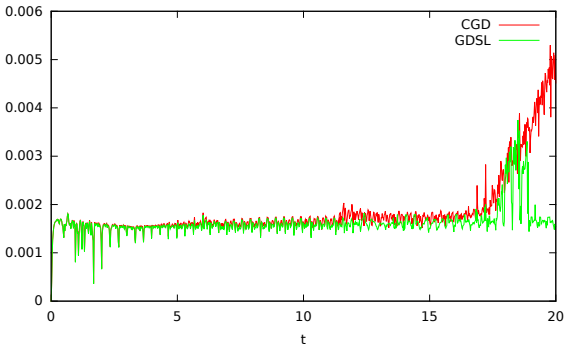
(a) Variation relative de la masse.



(b) Variation relative de la norme L^1.

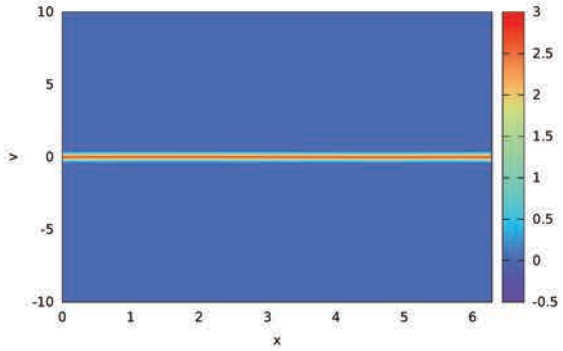


(c) Variation relative de la norme L^2.

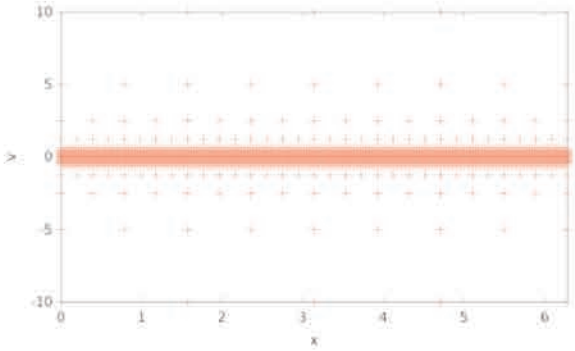


(d) Variation relative de la norme L^∞.

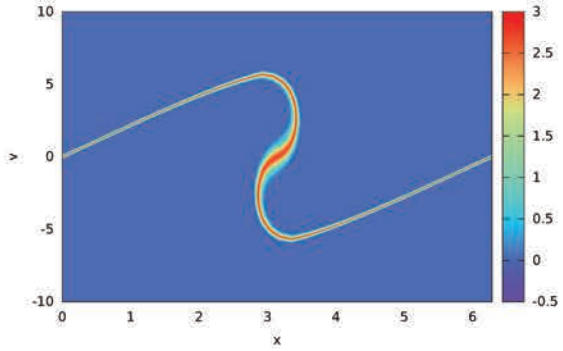
FIGURE 6.30 – Évolution des principales normes pour le cas test de couche froide avec perturbation en espace jusqu'à $t = 20$ uta.



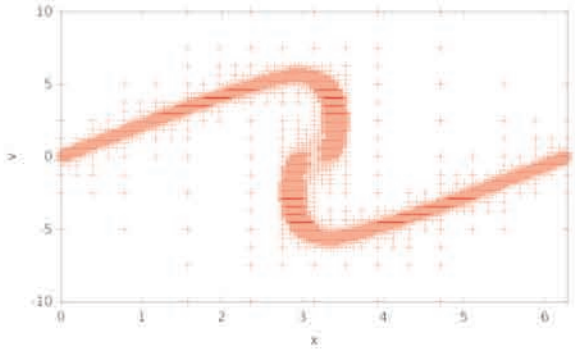
(a) Distribution initiale.



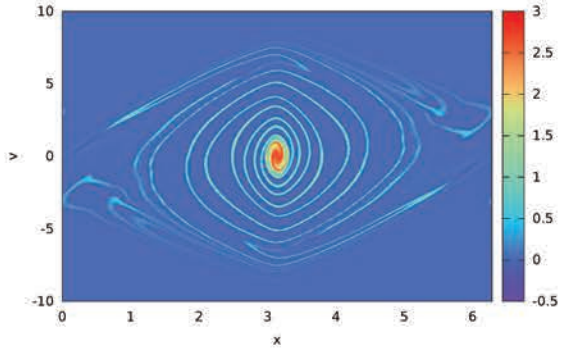
(b) Maillage initial.



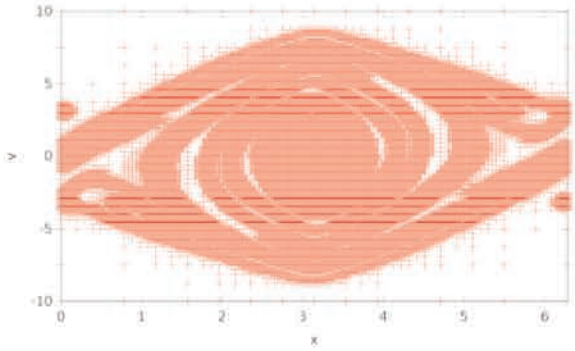
(c) Distribution à $t = 2$ uta.



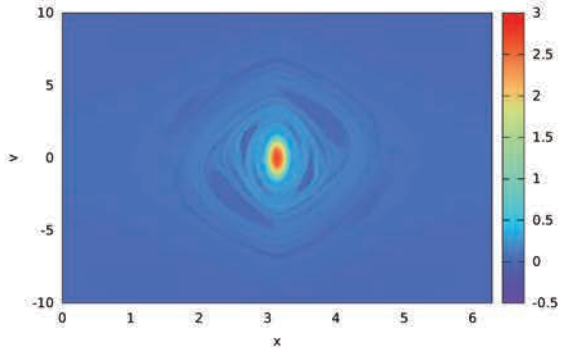
(d) Maillage à $t = 2$ uta.



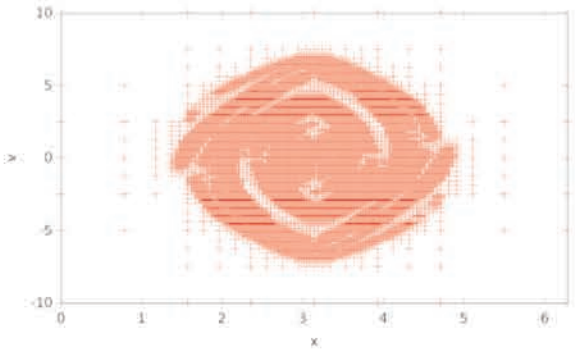
(e) Distribution à $t = 6$ uta.



(f) Maillage à $t = 6$ uta.

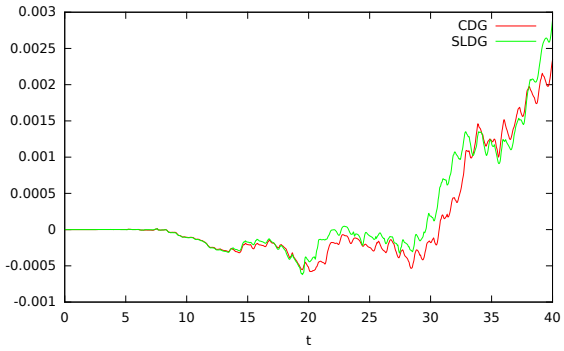


(g) Distribution à $t = 40$ uta.

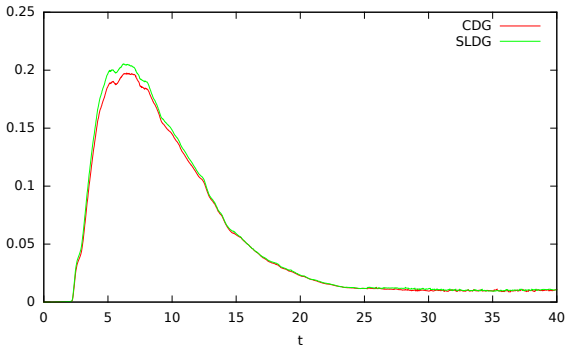


(h) Maillage à $t = 40$ uta.

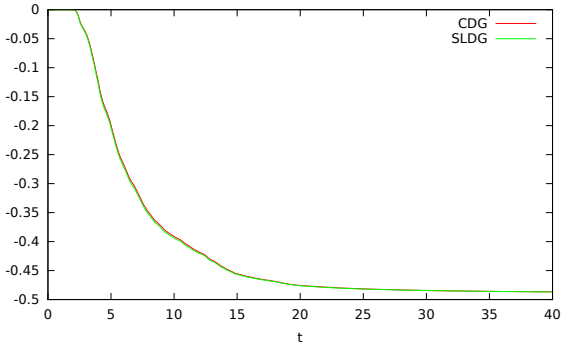
FIGURE 6.31 – Distribution et maillage pour le cas test astrophysique de couche froide avec perturbation en vitesse par la méthode GDSL.



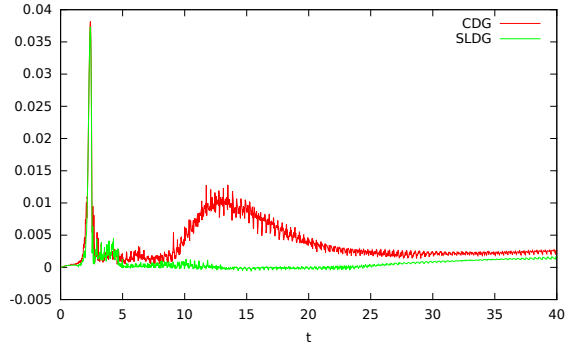
(a) Variation relative de la masse.



(b) Variation relative de la norme L^1.



(c) Variation relative de la norme L^2.



(d) Variation relative de la norme L^∞.

FIGURE 6.32 – Évolution des principales normes pour le cas test de couche froide avec perturbation en vitesse jusqu'à $t = 40$ uta.

6.5.2 Couche froide avec perturbation en vitesse

Ce cas test est fortement similaire au cas test précédent (section 6.5.1) mais la fonction de distribution initiale est perturbée en vitesse et non en espace, c'est à dire

$$f_0(x,v) = \frac{1}{\sqrt{2\pi}v_{th}} \exp\left(-\frac{(v-u(x))^2}{2v_{th}^2}\right), \quad (x,v) \in [0\ ;\ 2\pi] \times [-10\ ;\ 10], \tag{6.12}$$

avec $v_{th} = 0.15$ et

$$u(x) = 0.02\sin(x).$$

On considère un temps final $T = 40$ unité de temps astrophysique et un pas de temps de 0.02 unité de temps pour un total de 2000 pas de temps. On utilise des polynômes de degré inférieur ou égal à 2 par direction. Le niveau maximum de raffinement est fixé à 8 et le seuil est $\epsilon_0 = 3.10^{-3}$. Tout comme le cas précédent, ce cas test est conçu pour toujours utiliser le niveau le plus fin autorisé.

L'évolution de la fonction de distribution et du maillage avec la méthode GDSL est représentée sur la figure 6.31. Des résultats en tous points similaires sont obtenus par la méthode CGD. Comme annoncé précédemment, le profil de la distribution est similaire à celui obtenu avec une perturbation en espace dans la section précédente. On retrouve les mêmes phénomènes de filamentation, les mêmes artefacts liés à la périodicité et à la dissipation, et le même lissage de la distribution en temps long.

La figure 6.32 illustre les variations relatives de la masse et des norme L^1, L^2 et L^∞. On remarque à l'aide des variations de la masse et des variations de la norme L^1 que la distribution devient fortement négative. Les parties négatives sont visibles sur la figure 6.31e aux extrémités des filaments, aux bords du domaine.

6.5.3 Distribution initiale gaussienne

On s'intéresse ici au cas test décrit dans la section 2.1.1 et illustré par la figure 2 de [20]. L'article présente l'application d'une méthode de waterbags à un modèle astrophysique à une dimension d'espace et de vitesse. Ce cas test y est décrit avec des conditions de bords ouverts. Mon code est écrit avec des conditions aux limites de Dirichlet homogènes en vitesse et des conditions aux limites périodiques en espace. En se basant sur les résultats présentés dans [20], il a suffi d'adapter le domaine. Les conditions initiales considérées sont donc

$$f_0(x,v) = 4\exp\left(-\frac{(x^2+v^2)}{0.08}\right), \quad (x,v) \in [-2\ ;\ 2]^2. \tag{6.13}$$

Dans ses conditions, l'équation de Poisson que l'on résout est

$$-\frac{\partial E(x,t)}{\partial x} = \rho(x,t) - \frac{2\pi}{25}. \tag{6.14}$$

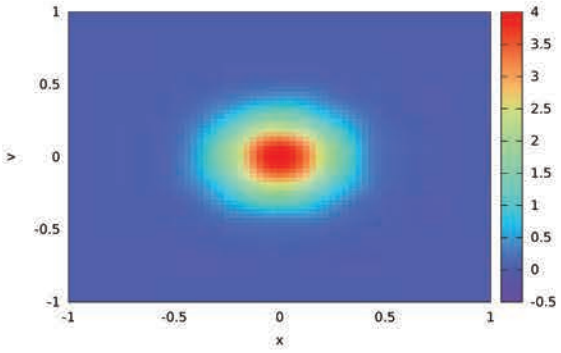
La constante introduite dans l'équation (6.14) assure la résolution dans le cas périodique et est obtenue par

$$\frac{1}{4}\iint_{\mathbb{R}^2} f_0(x,v)\,dxdv \;=\; \frac{1}{4}\iint_{\mathbb{R}^2} 4\exp\left(-\frac{(x^2+v^2)}{0,08}\right)dxdv \;=\; \frac{2\pi}{25}, \tag{6.15}$$

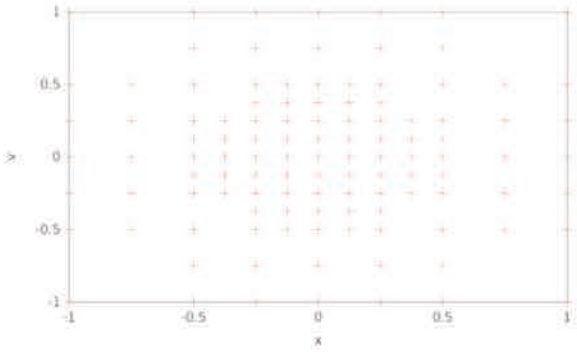
où le facteur 1/4 est la longueur du domaine dans la direction x. L'erreur relative commise en intégrant sur $\mathbb{R}^2$ et non sur $[-2\ ;\ 2]^2$ est de l'ordre de 10^{-24}, donc négligeable devant l'erreur machine relative qui est de l'ordre de 10^{-16} en double précision.

Le temps final considéré est $T = 100$ unités de temps avec un pas de temps de 0.1 unité de temps, soit un total de 1000 pas de temps. Les polynômes utilisés sont de degré inférieur ou égal à 2 par direction. On permet jusqu'à 9 niveaux de raffinements avec un seuil $\epsilon_0 = 3.10^{-3}$.

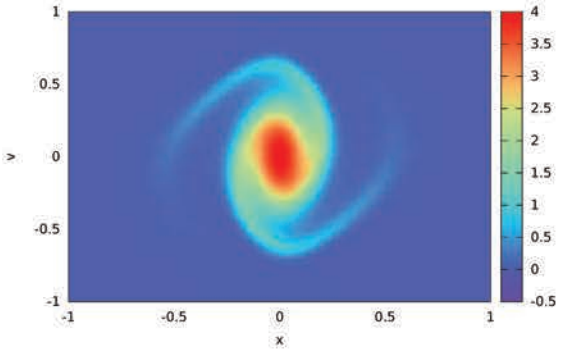
La distribution et le maillage obtenus par la méthode CGD sont représentés sur la figure 6.33. Le domaine de simulation a été fixé à $[-2\ ;\ 2]$ pour être certain d'éviter les interactions avec les bords, mais en pratique la distribution est quasi nulle en dehors de $[-1\ ;\ 1]$. La distribution et le maillage obtenus par la méthode GDSL sont presque identiques à ceux obtenus par la méthode CGD. La figure 6.34 indique les contours de la distribution. La figure 6.34a associée au fait que les cellules feuilles hors de $[-1\ ;\ 1]$ sont des cellules de niveau 2 (non représentées sur la figure 6.33) confirme que la distribution est quasi-uniforme en dehors de $[-1\ ;\ 1]$. On discerne sur la figure 6.33 le maillage qui se raffine le long des filaments. Le raffinement est beaucoup plus visible sur la figure 6.35 qui illustre le nombre de cellules et le niveau maximum de raffinement utilisé au cours de la simulation. Le niveau maximum augmente rapidement jusqu'à son maximum mais le nombre de cellules augmente beaucoup plus longtemps à mesure que les filaments



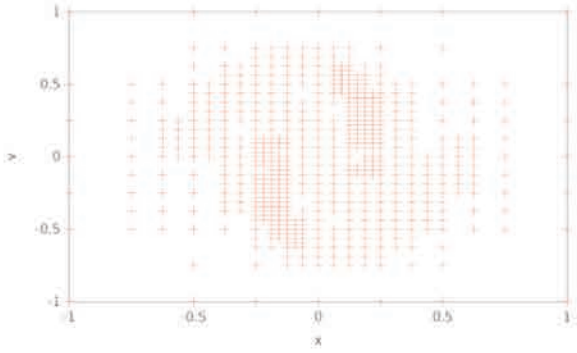
(a) Distribution initiale.



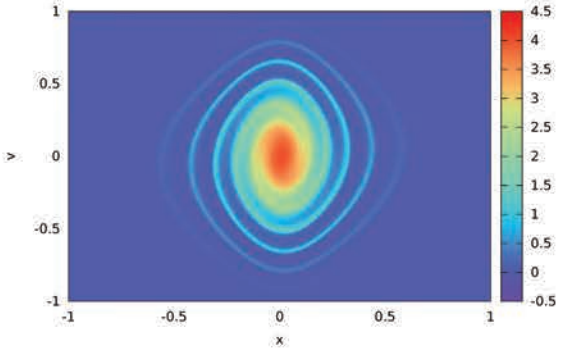
(b) Maillage initial.



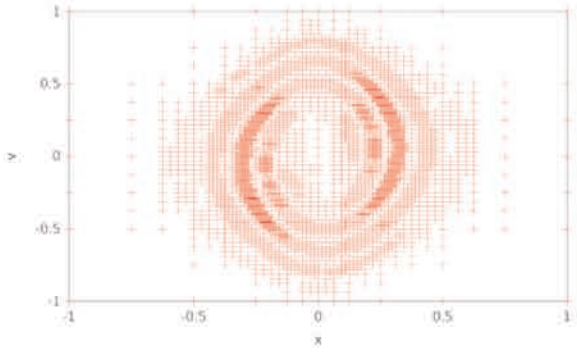
(c) Distribution à $t = 5$ uta.



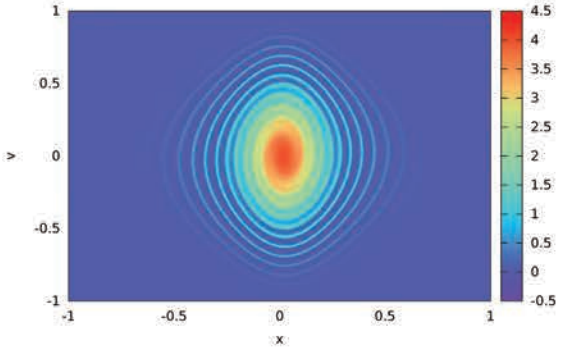
(d) maillage à $t = 5$ uta.



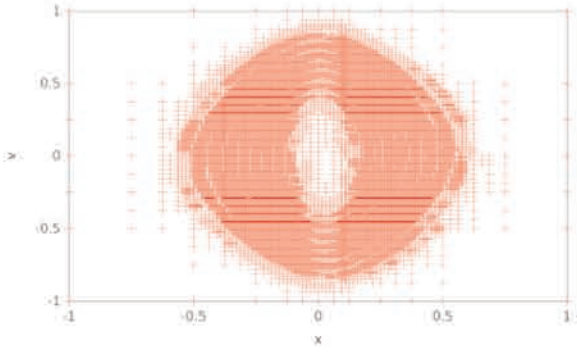
(e) Distribution à $t = 15$ uta.



(f) maillage à $t = 15$ uta.

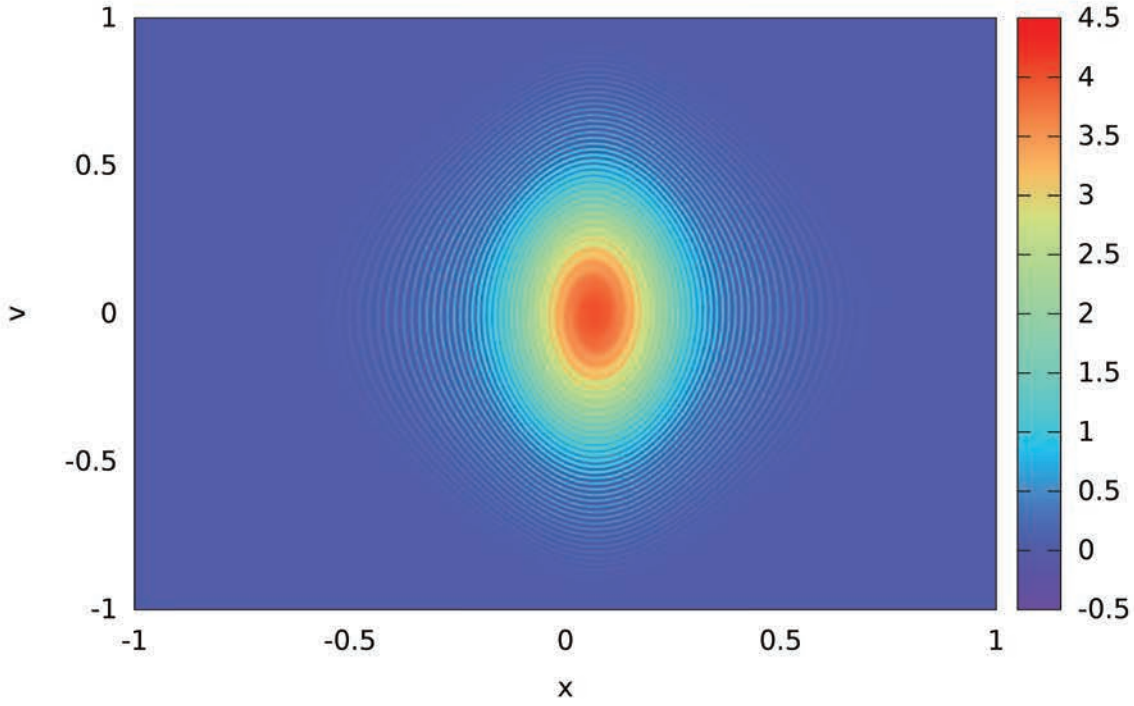


(g) Distribution à $t = 30$ uta.

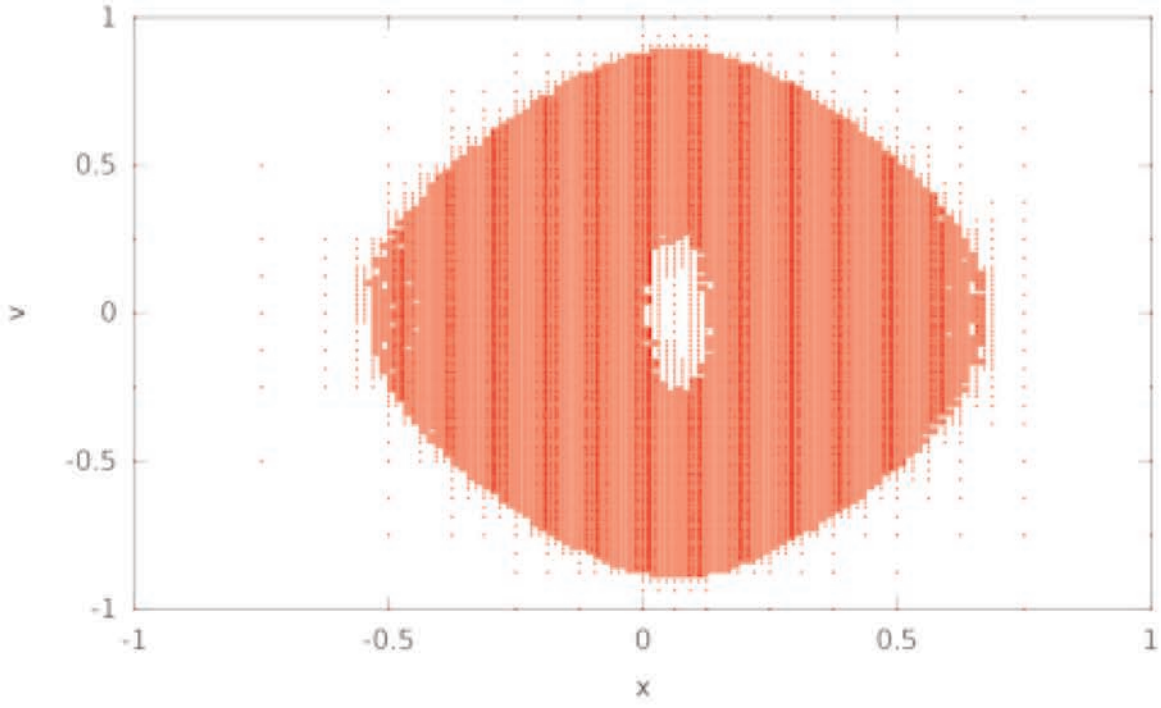


(h) maillage à $t = 30$ uta.

FIGURE 6.33 – Distribution et maillage pour le cas test astrophysique de distribution initiale Gaussienne par la méthode CGD.

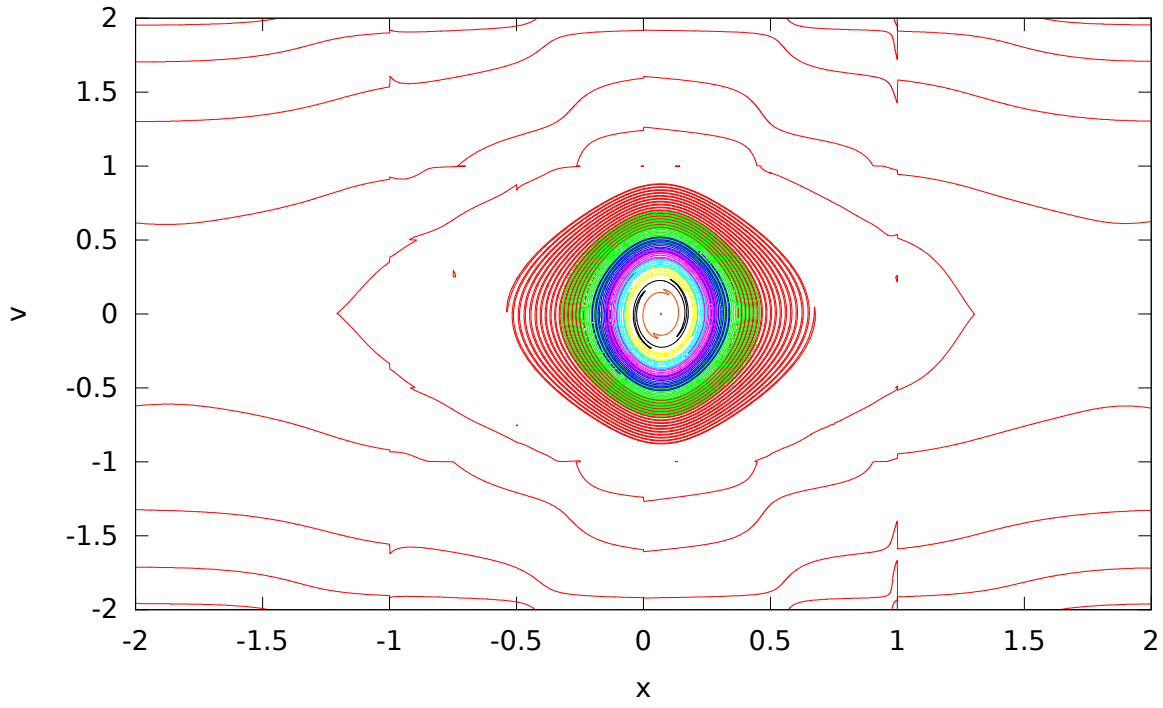


(i) Distribution à $t = 100$ uta.

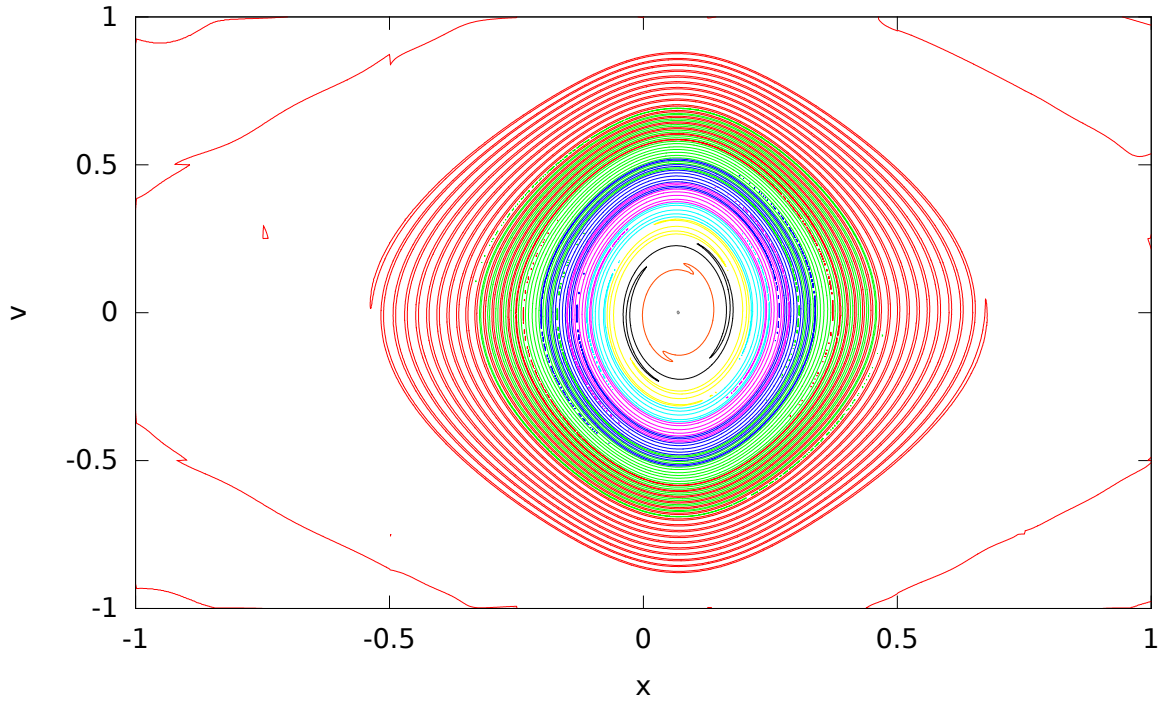


(j) maillage à $t = 100$ uta.

FIGURE 6.33 – Distribution et maillage pour le cas test astrophysique de distribution initiale Gaussienne par la méthode CGD.

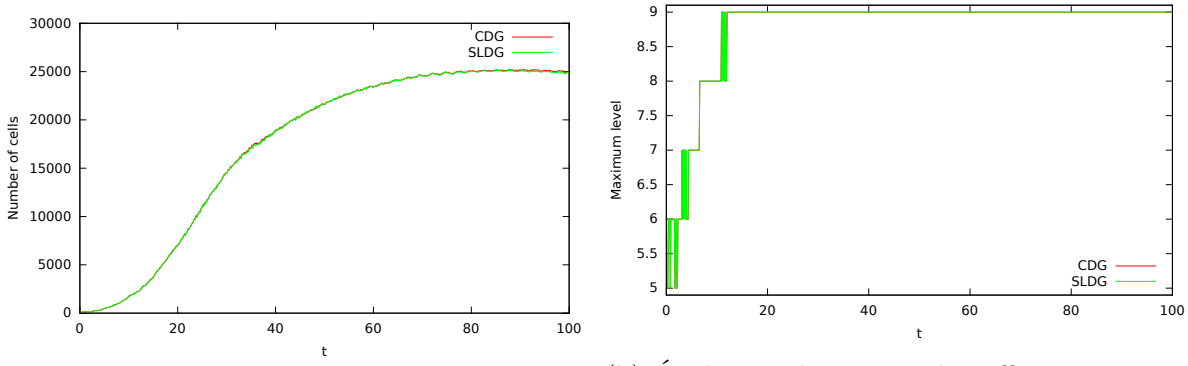


(a) ContourS sur $[-2\ ;\ 2]^2$.



(b) ContourS sur $[-1\ ;\ 1]^2$.

FIGURE 6.34 – Distribution à $t = 100$ uta pour le cas test astrophysique de distribution initiale gaussienne par la méthode CGD.



(a) Évolution du nombre de cellules.

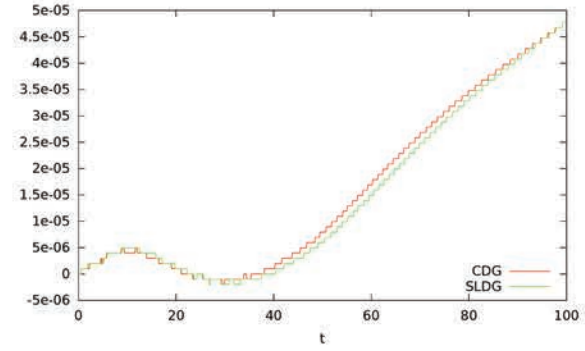
(b) Évolution du niveau de raffinement maximum utilisé.

FIGURE 6.35 – Évolution du maillage pour le cas test astrophysique de distribution initiale gaussienne.

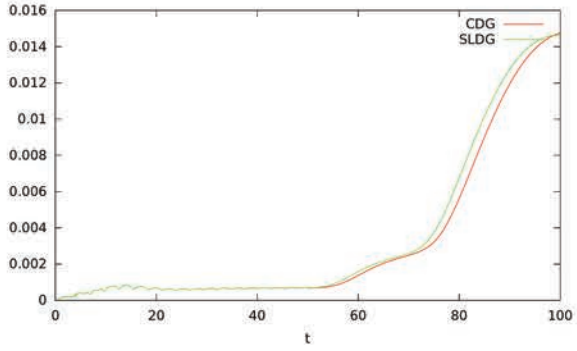
s'allongent et se resserrent. Au-delà de 80 uta, le maillage est limité à son niveau de raffinement le plus fin et ne peut plus suivre la formations de structures toujours plus fines.

Les variations de la masse numérique, des normes L^1 et L^2, de la quantité de mouvement, du minimum et du maximum de la distribution sont illustrées sur la figure 6.36. On remarque que la masse numérique varie par marches, ce qui est typique d'une variation due à l'erreur machine et amplifiée par un facteur donné. On peut néanmoins considérer que la masse numérique est conservée. On constate une fois encore la corrélation entre l'augmentation de la norme L^1 et la chute du minimum de la fonction de distribution dans des valeurs négatives. La dissipation et la baisse de la norme L^2 s'accélèrent simultanément. On observe la même augmentation initiale de la quantité de mouvement puis sa stagnation. Le maximum de la distribution varie régulièrement avec une variation relative de l'ordre de 10^{-3}.

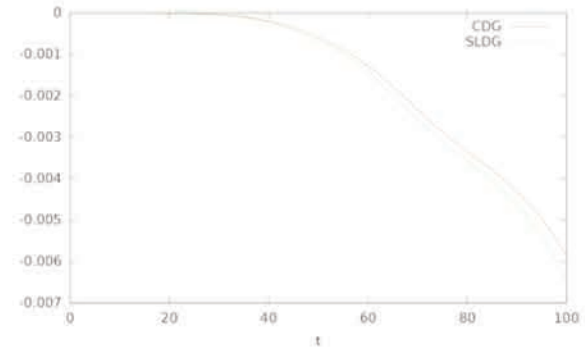
De façon générale, les cas test présentés présentent de très bonnes propriétés de conservation de la masse et des normes. L'utilisation d'un seuil de raffinement ϵ_0 associé à une forte capacité de raffinement ont montré, notamment à travers la section 6.5.3, la capacité de l'algorithme à résoudre des filaments avec très peu de diffusion.



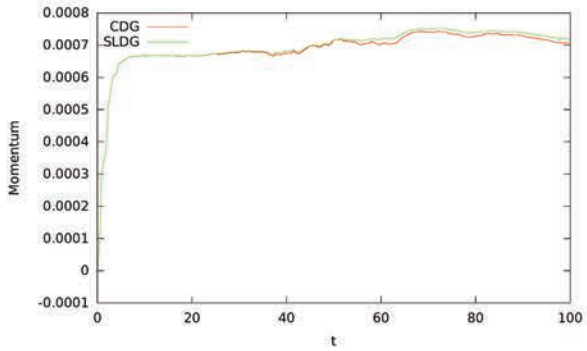
(a) Variation relative de la masse numérique.



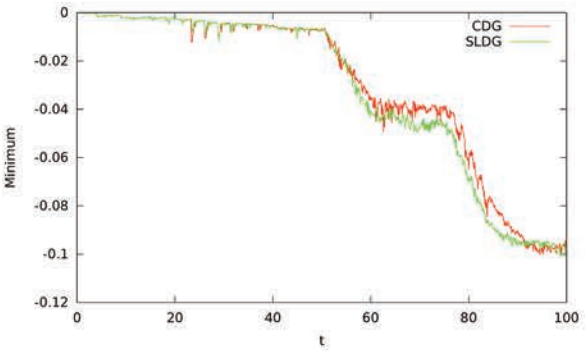
(b) Variation relative de la norme L^1.



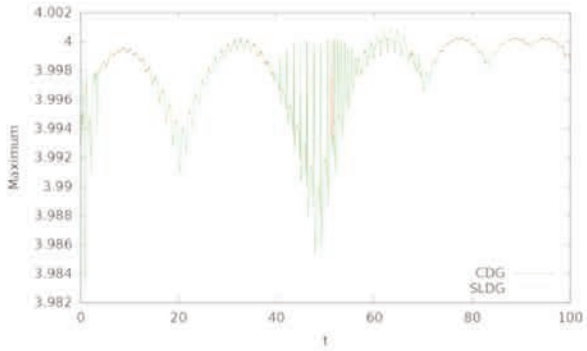
(c) Variation relative de la norme L^2.



(d) Variation absolue de la quantité de mouvement.



(e) Variation absolue du minimum.



(f) Variation absolue du maximum.

FIGURE 6.36 – Évolution des principales quantités de référence pour le cas test astrophysique de distribution initiale gaussienne.

Chapitre 7

Passage en dimension 4

Ce chapitre porte sur le passage d'un problème en dimension 2 à un problème en dimension 4. Il regroupe la présentation des équations dans un espace des phases à deux dimensions d'espace et deux dimensions de vitesse, l'implémentation des méthodes, la validation du code et quelques résultats.

7.1 Les équations de Vlasov-Poisson en dimension 4

Comme dans la section 2.1, on part des équations générales normalisées (2.9), (2.2) et (2.3) que l'on ramène dans un espace des phases à deux dimensions en espace et deux dimensions en vitesse. L'équation de Vlasov devient alors

$$\begin{aligned}\partial_t f(x,y,v_x,v_y,t) + v_x\partial_x f(x,y,v_x,v_y,t) + v_y\partial_y f(x,y,v_x,v_y,t) \\ + E_x(x,y,t)\partial_{v_x} f(x,y,v_x,v_y,t) + E_y(x,y,t)\partial_{v_y} f(x,y,v_x,v_y,t) = 0, \\ (x,y,v_x,v_y,t) \in \Omega_x \times \Omega_y \times \mathbb{R}^2 \times \mathbb{R}_+. \quad (7.1)\end{aligned}$$

L'équation de Poisson pour la physique des plasmas s'écrit

$$-\partial_x^2\Phi(x,y,t) - \partial_y^2\Phi(x,y,t) = \rho(x,y,t) - 1, \quad (7.2a)$$

$$E_x(x,y,t) = -\partial_x\Phi(x,y,t), \quad (7.2b)$$

$$E_y(x,y,t) = -\partial_y\Phi(x,y,t), \quad (7.2c)$$

où $\rho(x,y,t)$ est la densité obtenue en intégrant en vitesse la fonction de distribution

$$\rho(x,y,t) = \iint_{\mathbb{R}^2} f(x,y,v_x,v_y,t)\,dv_x dv_y. \quad (7.3)$$

Le système est fermé par des conditions aux limites de Dirichlet homogènes sur le potentiel Φ, c'est-à-dire $\Phi|_{\partial(\Omega_x\times\Omega_y)} = 0$. Lors de la résolution numérique, on remplace le domaine infini en vitesse par un domaine fini avec des conditions de Dirichlet homogènes aux bords. Les propriétés de conservation (conservation des extremums, des normes et de la masse, de la quantité de mouvement et de l'énergie totale) sont inchangées, comme énoncé à la section 2.1.

7.2 Formulation caractéristiques-Galerkin discontinue de l'équation de Vlasov

On choisit de ne résoudre ce problème qu'avec la méthode caractéristiques-Galerkin discontinue, toujours associée au splitting de Strang pour la résolution en temps. Les différentes étapes sont alors

$$\partial_t f + v_x \partial_x f = 0 \qquad \text{sur } [t^n \ ; \ t^n + 0.25\Delta t], \tag{7.4a}$$
$$\partial_t f + v_y \partial_y f = 0 \qquad \text{sur } [t^n \ ; \ t^n + 0.5\Delta t], \tag{7.4b}$$
$$\partial_t f + v_x \partial_x f = 0 \qquad \text{sur } [t^n + 0.25\Delta t \ ; \ t^n + 0.5\Delta t], \tag{7.4c}$$
$$\partial_t f + E_x \partial_{v_x} f = 0 \qquad \text{sur } [t^n \ ; \ t^n + 0.5\Delta t], \tag{7.4d}$$
$$\partial_t f + E_y \partial_{v_y} f = 0 \qquad \text{sur } [t^n \ ; \ t^n + \Delta t], \tag{7.4e}$$
$$\partial_t f + E_x \partial_{v_x} f = 0 \qquad \text{sur } [t^n + 0.5\Delta t \ ; \ t^n \Delta t], \tag{7.4f}$$
$$\partial_t f + v_x \partial_x f = 0 \qquad \text{sur } [t^n + 0.5\Delta t \ ; \ t^n + 0.75\Delta t], \tag{7.4g}$$
$$\partial_t f + v_y \partial_y f = 0 \qquad \text{sur } [t^n + 0.5\Delta t \ ; \ t^n + \Delta t], \tag{7.4h}$$
$$\partial_t f + v_x \partial_x f = 0 \qquad \text{sur } [t^n + 0.75\Delta t \ ; \ t^n + \Delta t]. \tag{7.4i}$$

Dans le prolongement de ce qui a été présenté dans les sections 2.4.1 et 2.4.2, les formules (2.65) et (2.66) sont étendues au problème en dimension 4. On obtient donc, dans la direction x

$$\tilde{f}^{n+1}_{\boldsymbol{k},T_{i,j,k,l}} = \sum_{l_g} w_{l_g} \sum_{k_g} w_{k_g} \sum_{j_g} w_{j_g} \sum_{\lambda} \sum_{i_g} w_{i_g} f^n(x^*_{i_g,k_g,\lambda}, y_{j_g}, v_{x,k_g}, v_{y,l_g}) \\ P_{\boldsymbol{k}}(x_{i_g,k_g,\lambda}, y_{j_g}, v_{x,k_g}, v_{y,l_g}) \frac{\Gamma(T^*_{i,j,k,l,\lambda,k_g})}{\Delta x_{i,j,k,l}}. \tag{7.5}$$

L'indice λ est ici utilisé pour marquer les cellules traversées. De la même façon, on obtient dans la direction y

$$\tilde{f}^{n+1}_{\boldsymbol{k},T_{i,j,k,l}} = \sum_{k_g} w_{k_g} \sum_{i_g} w_{i_g} \sum_{l_g} w_{l_g} \sum_{\lambda} \sum_{j_g} w_{j_g} f^n(x_{i_g}, y^*_{j_g,l_g,\lambda}, v_{x,k_g}, v_{y,l_g}) \\ P_{\boldsymbol{k}}(x_{i_g}, y_{j_g,l_g,\lambda}, v_{x,k_g}, v_{y,l_g}) \frac{\Gamma(T^*_{i,j,k,l,\lambda,l_g})}{\Delta y_{i,j,k,l}}, \tag{7.6}$$

dans la direction v_x

$$\tilde{f}^{n+1}_{\boldsymbol{k},T_{i,j,k,l}} = \sum_{l_g} w_{l_g} \sum_{j_g} w_{j_g} \sum_{i_g} w_{i_g} \sum_{\lambda} \sum_{k_g} w_{k_g} f^n(x_{i_g}, y_{j_g}, v^*_{x,k_g,i_g,j_g,\lambda}, v_{y,l_g}) \\ P_{\boldsymbol{k}}(x_{i_g}, y_{j_g}, v_{x,k_g,i_g,j_g,\lambda}, v_{y,l_g}) \frac{\Gamma(T^*_{i,j,k,l,\lambda,i_g,j_g})}{\Delta v_{x,i,j,k,l}}, \tag{7.7}$$

et dans la direction v_y

$$\tilde{f}^{n+1}_{\boldsymbol{k},T_{i,j,k,l}} = \sum_{k_g} w_{k_g} \sum_{i_g} w_{i_g} \sum_{j_g} w_{j_g} \sum_{\lambda} \sum_{l_g} w_{l_g} f^n(x_{i_g}, y_{j_g}, v_{x,k_g}, v^*_{y,l_g,i_g,j_g,\lambda}) \\ P_{\boldsymbol{k}}(x_{i_g}, y_{j_g}, v_{x,k_g}, v_{y,l_g,jg,\lambda}) \frac{\Gamma(T^*_{i,j,k,l,\lambda,i_g,j_g})}{\Delta x_{i,j,k,l}}. \quad (7.8)$$

On remarque que dans les directions x et y, la longueur $\Gamma(T^*_{i,j,k,l,\lambda,.})$ ne dépend que de la vitesse dans la direction associée. En revanche, dans les directions v_x et v_y, la longueur $\Gamma(T^*_{i,j,k,l,\lambda,.,.})$ dépend des deux coordonnées d'espace car le champ électrique dépend de x et de y.

Contrairement à ce qui a été fait dans le cas 1d×1v, les polynômes considérés ici ne sont pas de degré inférieur ou égal à n par direction, mais de degré total inférieur ou égal à n.

Dans cette section on considère l'espace de polynômes

$$\mathfrak{P}^K = \left\{ \sum_{\boldsymbol{k}} p_{k_1} \dots p_{k_d}, \quad \boldsymbol{k} \text{ tels que } \sum_{i=0}^{4} k_i \leq K \right\}.$$

C'est donc le degré total des polynômes qui est majoré par K et non le degré par direction. L'ordre d'erreur en espace n'est a priori que peu affecté mais le nombre de degrés de liberté par cellule est restreint à

$$\frac{(K+4)!}{4!K!}$$

contre $(K+1)^4$ si on avait utilisé l'espace $\mathbb{P}^K$ comme en dimension deux. Les fonctions d'échelles sont toujours construites à partir de l'espace $\mathbb{P}^K$, les multi-ondelettes sont également construites à partir de l'espace $\mathbb{P}^K$ et de la généralisation de (3.33) en dimension quatre, mais on considère simplement que pour tous les $\boldsymbol{k}$ tels que $|\boldsymbol{k}| = k_1+k_2+k_3+k_4 > K$, alors les coefficients d'échelle $s^m_{\boldsymbol{k}}$ donnent des informations d'un ordre supérieur à l'ordre considéré. Ces coefficients ne sont donc jamais calculés et considérés comme nuls lors du raffinement. Afin d'obtenir une représentation adaptative correcte et un seuillage correct, il est par contre nécessaire de considérer l'ensemble des coefficients de multi-ondelettes $d^m_{\boldsymbol{k}}$, avec $\left\|\boldsymbol{k}\right\|_\infty \leq K$.

Le maillage utilisé est un maillage en dimension quatre. Chaque cellule est donc un volume en dimension quatre et peut être raffinée en seize cellules filles.

7.3 Résultats numériques

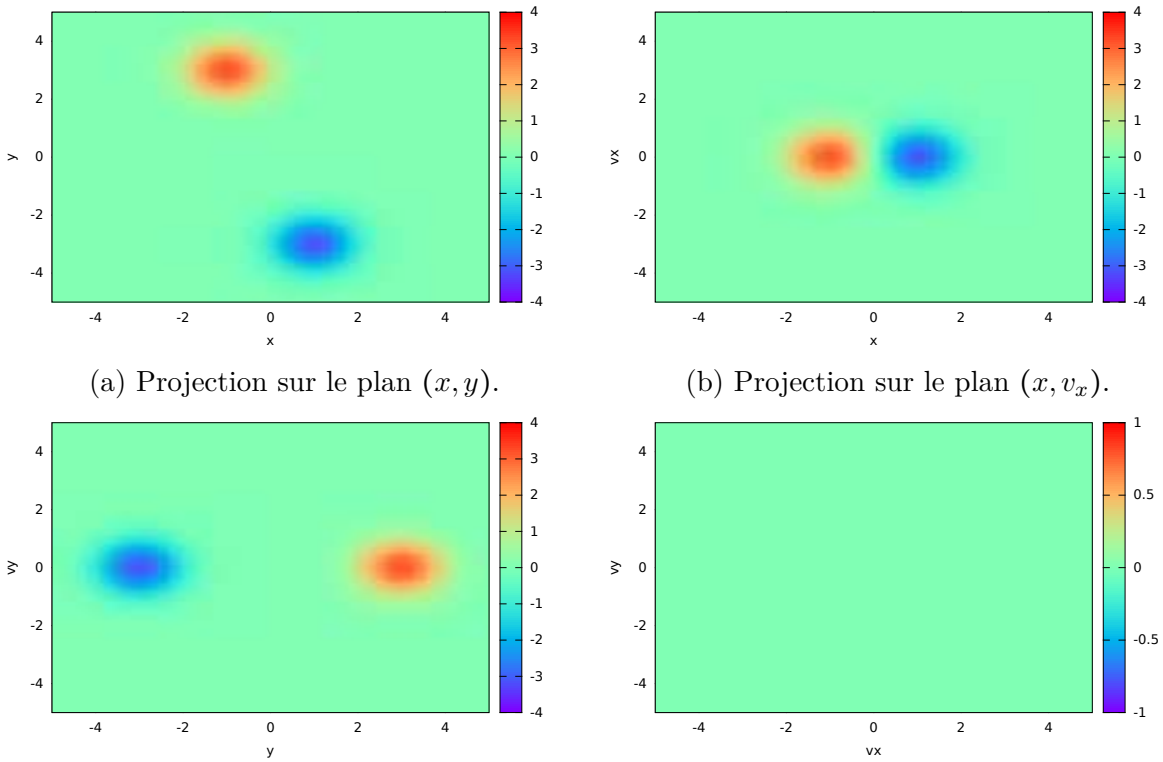
7.3.1 Rotation

On résout ici l'équation

$$\partial_t f + v_x \partial_x f - v_y \partial_y f - x \partial_{v_x} f + y \partial_{v_x} f = 0. \quad (7.9)$$

Il s'agit de la combinaison d'une rotation dans le plan (x, v_x) avec une vitesse angulaire de +1 avec une rotation dans le plan (y, v_y) avec une vitesse angulaire de −1. La condition initiale est

$$f_0(x, y, v_x, v_y) = \exp\left(-(x+1)^2 - (y-3)^2 - v_x^2 - v_y^2\right) - \exp\left(-(x-1)^2 - (y+3)^2 - v_x^2 - v_y^2\right). \quad (7.10)$$



(a) Projection sur le plan (x, y).

(b) Projection sur le plan (x, v_x).

(c) Projection sur le plan (y, v_y).

(d) Projection sur le plan (v_x, v_y).

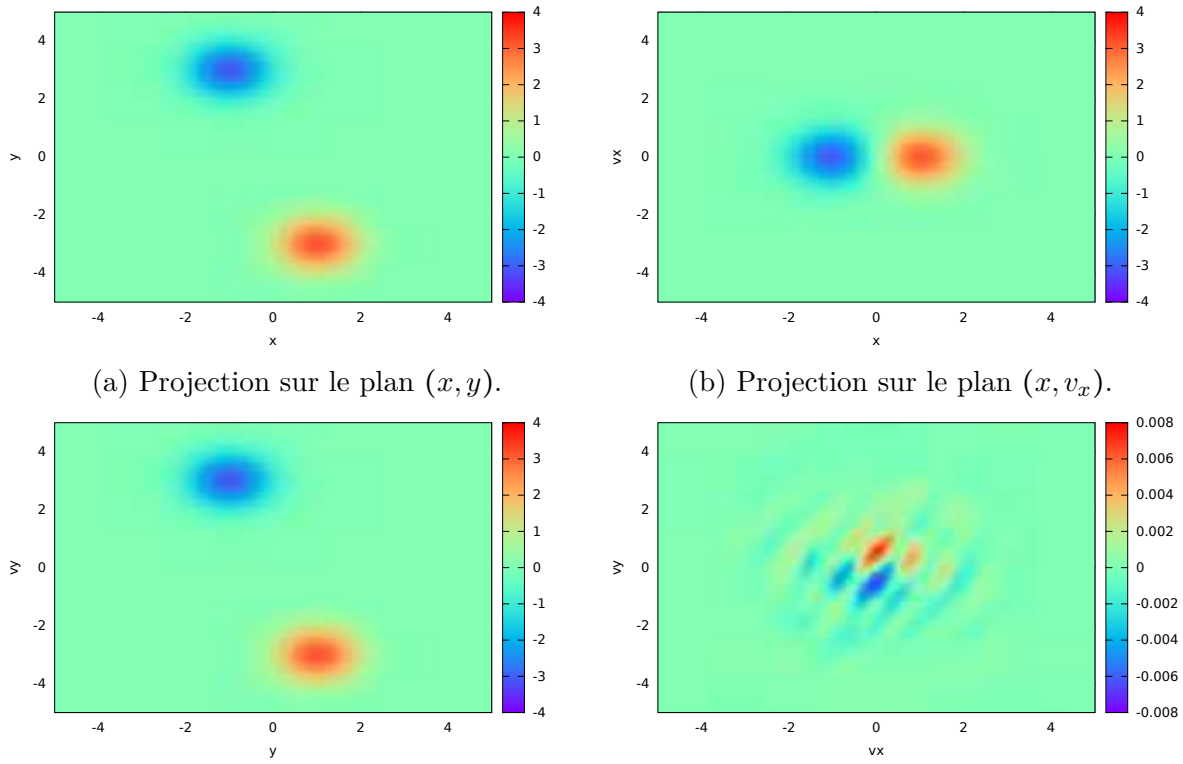
FIGURE 7.1 – Projections initiales.

On résout le problème sur $[-10\ ;\ 10]^4$. On considère un temps final $T = 2\pi$ et 200 pas de temps. On utilise 6 niveaux de raffinements et un seuil $\epsilon_0 = 10^{-10}$. Des essais préliminaires ont permis de constater la nécessité d'utiliser un seuil aussi petit. On trace dans cette section la projection de la distribution sur différents plans. La projection de $f(x, y, v_x, v_y)$ sur le plan (x, v_x) correspond à l'opération

$$Pf(x, v_x) = \int_{\Omega_y} \int_{\Omega_{v_y}} f(x, y, v_x, v_y)\, dv_y\, dy. \tag{7.11}$$

Même si le domaine de calcul est $[-10\ ;\ 10]^4$, on peut se contenter de tracer les projections sur un domaine $[-5\ ;\ 5]^2$ pour les différents temps représentés. La distribution est uniformément nulle à l'extérieur. La distribution initiale projetée sur les plans (x, y), (x, v_x), (y, v_y) et (v_x, v_y) est présentée sur la figure 7.1. Les figures 7.2 et 7.3 montrent respectivement les projections sur les mêmes plans à $t = \pi$, soit après une demi-période, et à $t = 2\pi$, c'est-à-dire au temps final. On constate notamment avec la projection sur le plan (v_x, v_y) qu'initialement les deux gaussiennes se compensent parfaitement. À $t = \pi$, la superposition des gaussienne n'est pas parfaite et on voit apparaître un écart entre les deux, toujours visible sur le plan (v_x, v_y).

La figure 7.4 trace l'écart entre entre la projection numérique et la projection analytique $Pf_h - Pf$ au temps initial, à $t = \pi$ et à $t = 2\pi$. On observe à $t = \pi$ et à $t = 2\pi$ un écart entre la solution numérique et la solution exacte. Cet écart augmente au cours de



(a) Projection sur le plan (x, y).

(b) Projection sur le plan (x, v_x).

(c) Projection sur le plan (y, v_y).

(d) Projection sur le plan (v_x, v_y).

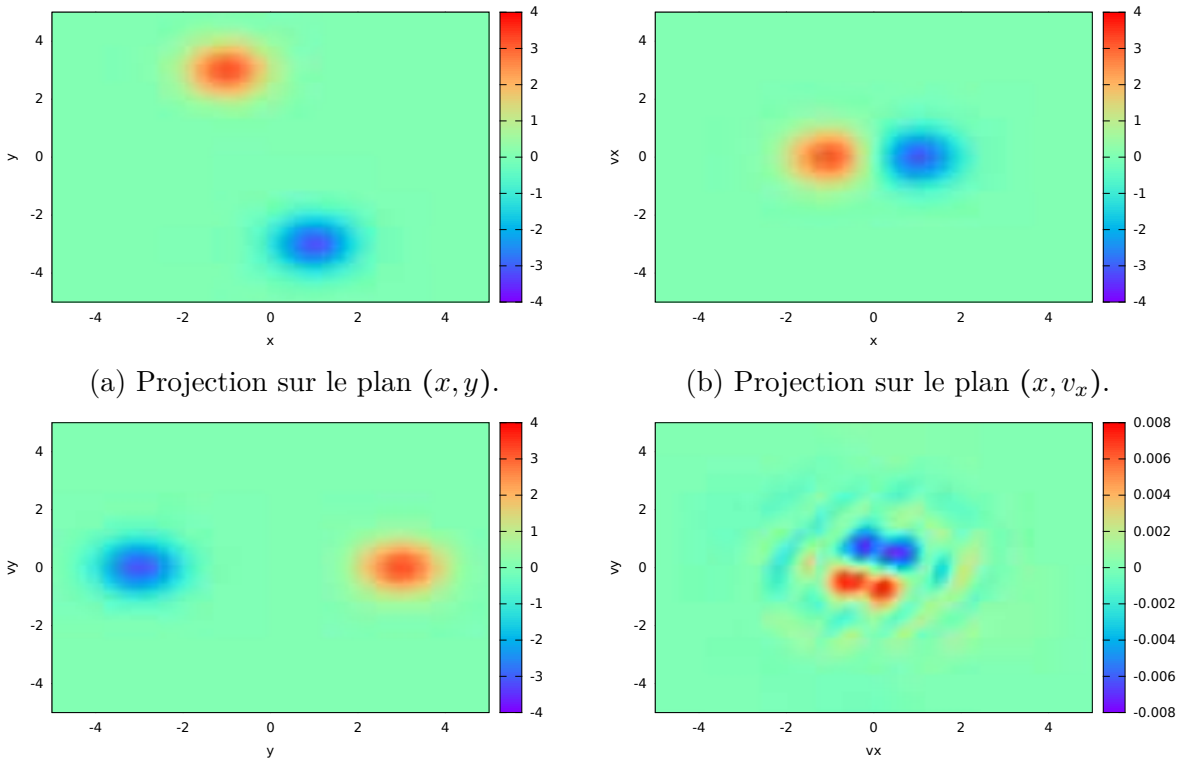
FIGURE 7.2 – Projections à $t = \pi$.

la simulation. Cet écart entre la solution numérique et la solution exacte est aussi visible sur les figures 7.2d et 7.3d.

Les variations de la masse totale, les variations relatives des normes L^1 et L^2 et les variations relatives des minimums et maximums sont tracées sur la figure 7.5. La variation de la masse totale est d'ordre 10^{-14}, ce qui correspond à l'erreur machine légèrement amplifiée. On peut donc considérer qu'elle est conservée. Les normes L^1 et L^2 par contre ne sont pas aussi bien conservées. Pour ces deux normes on constate des phénomènes particuliers à $t = \pi/2$ et à $t = 3\pi/2$. On observe alors des augmentations des deux normes, suives d'une décroissance lente pour la norme L^1 et d'une décroissance rapide pour la norme L^2.

Considérons le cas d'une gaussienne en dimension deux initialement portée par le maillage de la figure 7.6a. Cette gaussienne se déplace le long de la trajectoire verte sur la figure. (Le nom des direction n'a pas d'importance pour cette explication.) On voit qu'initialement, cette gaussienne est portée par une seule cellule de niveau 1. Cette gaussienne arrive finalement à la position représentée sur la figure 7.6b avec le maillage indiqué. Entre les deux, on voit que la trajectoire du sommet de la gaussienne passe par l'angle reliant les quatre cellules de niveau 1.

Dans la rotation en dimension quatre, les temps $t = \pi/2$ et $t = 3\pi/2$ sont les temps où les gaussiennes passent par des angles de cellules de niveau 1. Comme les détails sont portés par plus de cellules en dimension quatre qu'en dimension deux, les variations induites



(a) Projection sur le plan (x, y).

(b) Projection sur le plan (x, v_x).

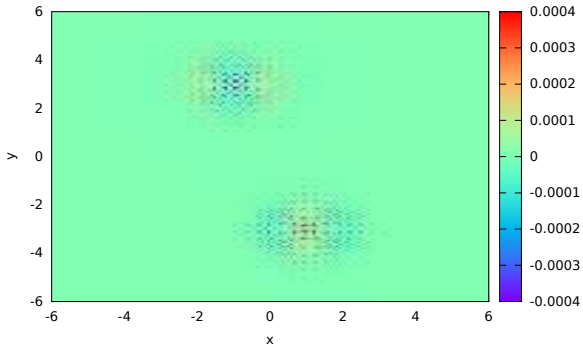
(c) Projection sur le plan (y, v_y).

(d) Projection sur le plan (v_x, v_y).

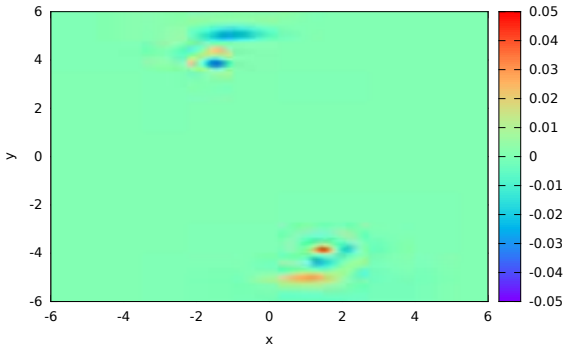
FIGURE 7.3 – Projections à $t = 2\pi$.

par cet événement sont plus importante qu'en dimension deux. Les pics observés sur les normes sont donc des phénomènes dûs au maillage. La variation relative du minimum et du maximum est très faible, moins de 1%. On constate que les deux valeurs ont la même variation relative, ce qui n'est pas surprenant compte tenu du fait que les deux gaussiennes sont symétriquement disposées et ont le même mouvement de rotation.

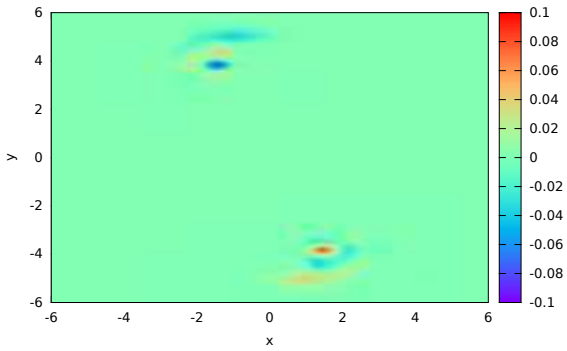
La figure 7.7 indique l'évolution du nombre total de cellules feuilles au cours du temps et l'évolution du nombre de cellules feuilles de niveau de raffinement inférieur ou égal à 5. On constate que l'immense majorité des cellules feuilles est portée par le sixième niveau de raffinement, le plus fin autorisé ici, et que le nombre total de cellules est à peu près constant. Dans l'hypothèse d'un maillage uniformément raffiné sur le sixième niveau, il y aurait eu un total de plus de 16 millions de cellules. La simulation a été réalisée avec environ 500 000 cellules, soit moins de 3% de ce qui aurait été nécessaire sur un maillage uniforme. Ceci illustre l'intérêt d'un maillage adaptatif pour simuler des phénomènes localisés dans des problèmes en dimension élevée.



(a) Écart au temps initial.

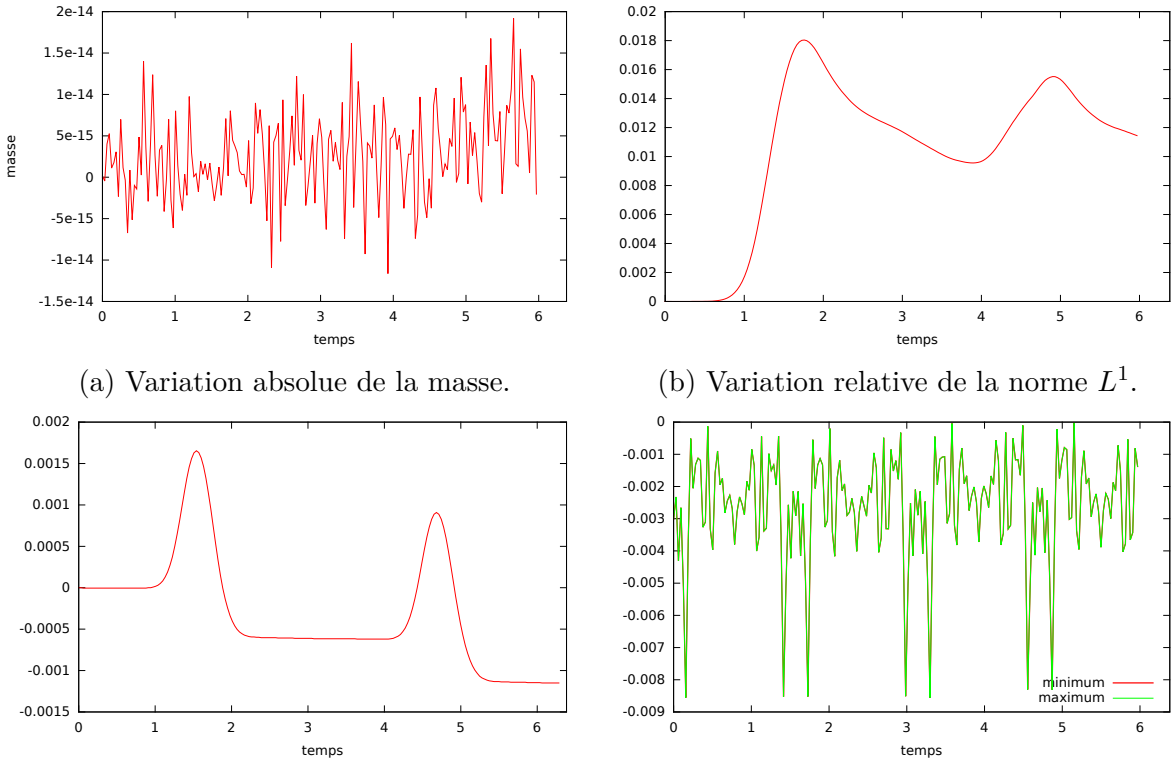


(b) Écart à $t = \pi$.



(c) Écart à $t = 2\pi$.

FIGURE 7.4 – Écart entre la projection sur le plan (x, y) et la solution exacte.



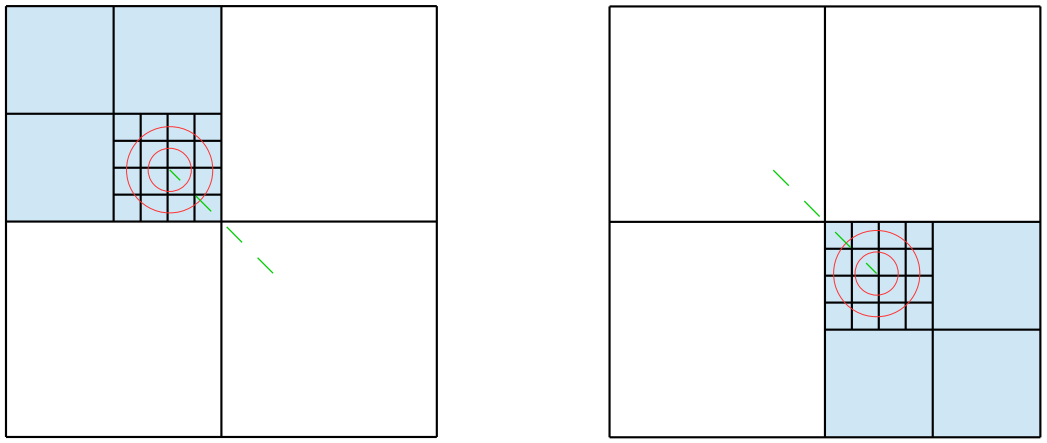
(a) Variation absolue de la masse.

(b) Variation relative de la norme L^1.

(c) Variation relative de la norme L^2.

(d) Variation relative des extremums.

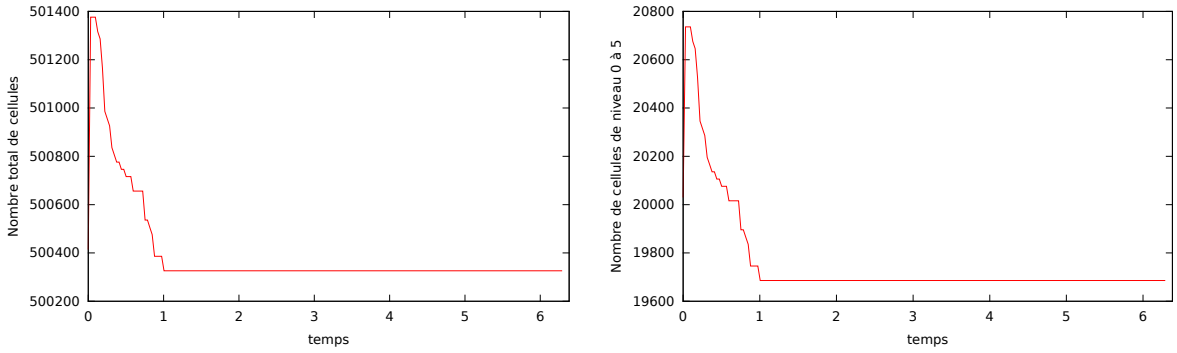
FIGURE 7.5 – Évolution des principales quantités de référence pour la rotation en dimension 4.



(a) Support initial.

(b) Support final.

FIGURE 7.6 – Changement de support de la distribution en passant par l'angle des cellules mères en dimension 2. En noir les contours des cellules, en bleu les cellules mère de niveau 1 porteuses d'information, en rouge des lignes de niveau, en vert la trajectoire du centre les lignes de niveau.



(a) Nombre total de cellules. (b) Nombre de cellules de niveau 0 à 5.

FIGURE 7.7 – Évolution du nombre de cellules.

Chapitre 8

Conclusion

Au cours de ce travail sur la résolution numérique des équations de Vlasov-Poisson pour la physique des plasmas et l'astrophysique, deux méthodes Galerkin discontinues adaptatives ont été mises en œuvre. L'utilisation de données locales sous forme de polynômes, tant pour la formulation Galerkin discontinue que pour la décomposition en multi-ondelettes, assure la compatibilité et la simplicité d'utilisation et d'association des deux aspects.

Il est montré dans [36] que pour du transport à coefficient constant, les méthodes GDSL et CGD sont équivalentes. Dans le cas de Vlasov-Poisson avec splitting, on se ramène à un transport à vitesse uniforme dans chaque direction, il est donc normal que les deux méthodes donnent des résultats semblables. Ces résultats ont également pu être confrontés à ceux de la littérature (pour les cas test de rotation-déformation et d'amortissement Landau) et validés.

Les multi-ondelettes et la méthode de seuillage mise en œuvre pour l'adaptativité ont également montré la capacité du maillage à choisir un niveau de raffinement adapté aux détails de la distribution suivie, tant pour les détails existants que pour la prédiction des détails à venir. Néanmoins, les difficultés rencontrées avec le passage en dimension quatre montrent que le travail dans ce domaine n'est pas terminé. L'analyse du critère de seuil est une tâche compliquée. Des travaux sur ce points sont présentés dans [12], [19] et [37] et mentionnés dans la section 3.6. Le niveau de raffinement optimal dépend de la précision voulue, de la fonction à décrire, et de la complexité de l'algorithme utilisé. Le choix du critère de seuil est finalement fait de façon empirique.

Une autre difficulté rencontrée au cours de cette thèse est l'écart entre l'écriture des méthodes sous forme d'algorithme ou de pseudo-code sur papier et l'obtention d'un code efficace. La première version du code se voulait facile à lire et à entretenir. Le code était alors rédigé en c++. Néanmoins, afin d'optimiser le code, il a été repris dans une version légèrement moins lisible en langage c. Aucun résultat n'a été présenté ici en matière de temps de calcul, de comparaison entre maillage uniforme et maillage non-uniforme et d'efficacité de parallélisation. Des petits test ont été réalisés pour tester l'efficacité de la parallélisation de la boucle de calcul (partie calcul des étapes 2, 4 et 5 de l'algorithme présenté à la section 4.5). La partie calculatoire du code a été parallélisée avec OpenMP. Cependant, une parallélisation efficace et complète du code n'a pas été envisagée dans le cadre de cette thèse parce que ce travail demande souvent de repenser complètement

l'algorithme numérique.

Des travaux ont été amorcés pour essayer de résoudre l'équation de Vlasov avec un couplage par les équations de Maxwell relativiste ou dans le cas de Vlasov-Poisson relativiste. Le passage à un modèle $2d \times 2v$ a aussi été amorcé mais, pour l'instant, se trouve confronté au problème d'adaptativité mentionné plus haut. Le passage à la dimension quatre est pour l'instant mon axe de travail principal. L'optimisation du code et des comparaisons approfondies entre ce code est d'autres codes, comme le code Obiwan qui a servi à l'article [12] pour Vlasov-Maxwell relativiste ou ce qui est fait avec la bibliothèque SeLaLib [3] développée par Inria, sont également des pistes envisagées pour des travaux futurs. Un axe de travail supplémentaire est l'utilisation de schémas en temps d'ordre plus élevé.

Bibliographie

[1] http://www.tomshardware.fr/articles/apollo-kune,1-6297.html.

[2] site internet de l'IPP. http://www.ipp.mpg.de/ippcms/eng/index.html.

[3] site internet de la bibliothèque Selalib. http://selalib.gforge.inria.fr/.

[4] site internet du projet Vlasix. http://http://www.vlasix.org/index.php.

[5] B. Alpert, G. Beylkin, D. Gines, and L. Vozovoi. Adaptive Solution of Partial Differential Equations in Multiwavelet Bases. *Journal of Computational Physics*, 182(1) :149–190, 2002.

[6] B. K. Alpert. A Class of Bases in L^2 for the Sparse Representation of Integral Operator. *SIAM Journal on Mathematical Analysis*, 24(1) :246–262, 1993.

[7] R. Archibald, G. Fann, and W. Shelton. Adaptive discontinuous Galerkin methods in multiwavelets bases. *Applied Numerical Mathematics*, 61(7) :879–890, 2011.

[8] B. Ayuso de Dios, J. A. Carrillo de la Plata, and C.-W. Shu. Discontinuous Galerkin methods for the one-dimensional Vlasov-Poisson system. *Kinetic and Related Models*, 4(4) :955–989, 2011.

[9] B. Ayuso de Dios, J. A. Carrillo de la Plata, and C.-W. Shu. Discontinuous Galerkin methods for the multi-dimensional Vlasov-Poisson problem. *Mathematical Models and Methods in Applied Sciences*, 22(12) :1250042—1–45, 2012.

[10] B. Ayuso de Dios and H. Soheil. High order and energy preserving Discontinuous Galerkin methods for the Vlasov-Poisson system. septembre 2012.

[11] N. Besse. *Étude mathématique et numérique de l'équation de Vlasov non linéaire sur des maillages nan structurés de l'espace des phases*. PhD thesis, Institut de recherche mathématique avancée, 2003.

[12] N. Besse, G. Latu, A. Ghizzo, E. Sonnendrüker, and P. Bertrand. A wavelet-MRA-based adaptive semi-Lagrangian method for the relativistic Vlasov-Maxwell system. *Journal of Computational Physics*, 227(16) :7889–7916, 2008.

[13] O. Bokanowski and G. Simarmata. Semi-Lagrangian discontinuous Galerkin schemes for some first and second-order partial differential equations. *ArXiv e-prints*, Mar. 2015.

[14] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral Methods, Fundamentals in Single Domains*. Springer, 2006.

[15] C. Cheng and G. Knorr. The integration of the vlasov equation in configuration space. *Journal of Computational Physics*, 22(3) :330–351, 1976.

[16] Y. Cheng and I. M. Gamba. Numerical study of one-dimensional Vlasov–Poisson equations for infinite homogeneous stellar systems. *Communications in Nonlinear Science and Numerical Simulation*, 17(5) :2052–2061, 2012. Special Issue : Mathematical Structure of Fluids and PlasmasDedicated to the 60th birthday of Phil Morrison.

[17] Y. Cheng, I. M. Gamba, and P. J. Morrison. Study of conservation and recurrence of Runge-Kutta discontinuous Galerkin scheme for Vlasov-Poisson system. *Journal of Scientific Computing*, 56(2) :319–349, 2013.

[18] P. N. Childs and K. W. Morton. Characteristic Galerkin Methods for Scalar Conservation Laws in One Dimension. *SIAM Journal on Numerical Analysis*, 27(3) :553–594, 1990.

[19] A. Cohen, R. DeVore, P. Petrushev, and H. Xu. Nonlinear Approximation and the Space BV($\mathbb{R}^2$). *American Journal of Mathematics*, 121(3) :587–628, 1999.

[20] S. Colombi and J. Touma. Vlasov-Poisson in 1D : waterbags. *Monthly Notices of the Royal Astronomical Society*, 441(3) :2414–2432, 2014.

[21] N. Crouseilles, M. Mehrenberger, and F. Vecil. Discontinuous Galerkin semi-Lagrangian method for Vlasov-Poisson. *ESAIM : Proc.*, 32 :211–230, 2011.

[22] J. Douglas and T. F. Russel. Numerical Methods for Convection-Dominated Diffusion Problems Based on Combining the Method of Characteristics with Finite Element or Finite Difference Procedures. *SIAM Journal on Numerical Analysis*, 5(19) :871–885, 1982.

[23] E. Frénod, S. A. Hirstoaga, and M. Lutz. Long-time simulation of a highly oscillatory Vlasov equation with an exponential integrator. *Comptes Rendus Mécanique*, 342(10–11) :595–609, 2014. Theoretical and numerical approaches for Vlasov-maxwell equations.

[24] E. Frénod, F. Salvarani, and E. Sonnendrücker. Long Time Simulation of a Beam in a Periodic Focusing Channel via a Two-Scale PIC-Method. *Mathematical Models and Methods in Applied Sciences*, 19(02) :175–197, 2009.

[25] S. Gérald. *Méthode de Galerkin Discontinue et intégrations explicites-implicites en temps basées sur un découplage des degrés de liberté : Applications au système des équations de Navier-Stokes*. PhD thesis, 2013. Thèse de doctorat dirigée par Coquel, Frédéric Mathématiques Université Pierre et Marie Curie - Paris 6 2013.

[26] N. Gerhard, F. Iacono, G. May, S. Müller, and R. Schäfer. A High-Order Discontinuous Galerkin Discretization with Multiwavelet-Based Grid Adaptation for Compressible Flows. *Journal of Scientific Computing*, 62(1) :25–52, 2015.

[27] N. Gerhard and S. Müller. Adaptive multiresolution discontinuous Galerkin Schemes for conservation laws : multi-dimensional case. *Computational and Applied Mathematics*, pages 1–29, Apr. 2014.

[28] W. Guo, R. D. Nair, and J.-M. Qiu. A conservative semi-Lagrangian discontinuous Galerkin scheme on the cubed-sphere. *Monthly Weather Review*, 142(1) :457–475, 2014. doi : 10.1175/MWR-D-13-00048.1.

[29] R. Heath, I. Gamba, P. Morrison, and C. Michler. A discontinuous Galerkin method for the Vlasov–Poisson system. *Journal of Computational Physics*, 231(4) :1140–1174, 2012.

[30] N. Hovhannisyan, S. Müller, and R. Schäfer. Adaptive multiresolution discontinuous Schemes for conservation laws. *Mathematics on Computation*, 83 :113–151, 2014.

[31] É. Madaule. Work placement report : Study and implementation of an energy conserving discontinuous Galerkin method for the Vlasov Poisson equations. Research report, Max-Planck Institute for Plasmaphysik, Aug. 2013.

[32] É. Madaule, M. Restelli, and E. Sonnendrüker. Energy conserving discontinuous Galerkin spectral element method for the Vlasov-Poisson system. *Journal of Computational Physics*, 279 :261–288, 2014.

[33] T. Nakamura and T. Yabe. Cubic Interpolated Propagation Scheme for Solving the Hyper-Dimensional Vlasov-Poisson Equation in Phase Space. *Computer Physics Communications*, 120(2–3) :122–154, 1999.

[34] K. Pfaffelmoser. Global classical solutions of the Vlasov-Poisson system in three dimensions for general initial data. *Journal of Differential Equations*, 95(2) :281–303, 1992.

[35] O. Pironneau. On the transport-Diffusion Algorithm and Its Application to the Navier-Stockes Equations. *Numerishe Mathematik*, 38(3) :309–332, 1982.

[36] J.-M. Qiu and C.-W. Shu. Positivity preserving semi-Lagrangian discontinuous Galerkin formulation : Theoretical analysis and application to the Vlasov–Poisson system. *Journal of Computational Physics*, 230(23) :8386–8409, 2011.

[37] X. M. Y. Ronald A. DeVore. Degree of Adaptive Approximation. *Mathematics of Computation*, 55(192) :625–635, 1990.

[38] J. A. Rossmanith and D. C. Seal. A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov-Poisson equations. *Journal of Computational Physics*, 230(16) :6203–6232, mars 2011.

[39] J. Schaeffer. Global existence of smooth solutions to the Vlasov Poisson system in three dimensions. *Communications in Partial Differential Equations*, 16(8-9) :1313–1335, 1991.

Annexe A

Polynôme de Legendre

Les polynômes de Legendre sont les solutions de l'équation différentielle ordinaire de Legendre

$$\frac{d}{dx}\left[(1-x^2)\frac{d}{dx}P_n(x)\right]+n(n+1)P_n(x)=0. \tag{A.1}$$

La famille des $\{P_k\}_{k=0}^{n}$ forme une base orthogonale de l'espace des polynômes $\mathbb{P}^n([-1\,;1])$. Chaque polynôme P_n est de degré n. Ils peuvent être exprimés par la formule de Rodrigue

$$P_n(x)=\frac{1}{2^n n!}\frac{d^n}{dx^n}\left[(x^2-1)^n\right]. \tag{A.2}$$

Une méthode simple de construire les polynômes est de le faire par la formule de récurrence de Bonnet

$$\begin{aligned}&P_0(x)=1,\\&P_1(x)=x,\\&(n+1)P_{n+1}(x)=(2n+1)xP_n(x)-nP_{n-1}(x),\quad \forall n>1.\end{aligned} \tag{A.3}$$

Les sept premiers polynômes sont

$$\begin{aligned}P_0(x)&=1,\\P_1(x)&=x,\\P_2(x)&=\frac{1}{2}(3x^2-1),\\P_3(x)&=\frac{1}{2}(5x^3-3x),\\P_4(x)&=\frac{1}{8}(35x^4-30x^2+3),\\P_5(x)&=\frac{1}{8}(63x^5-70x^3+15x),\\P_6(x)&=\frac{1}{16}(231x^6-315x^4+105x^2-5).\end{aligned} \tag{A.4}$$

Ces polynôme sont tracés figure A.1.

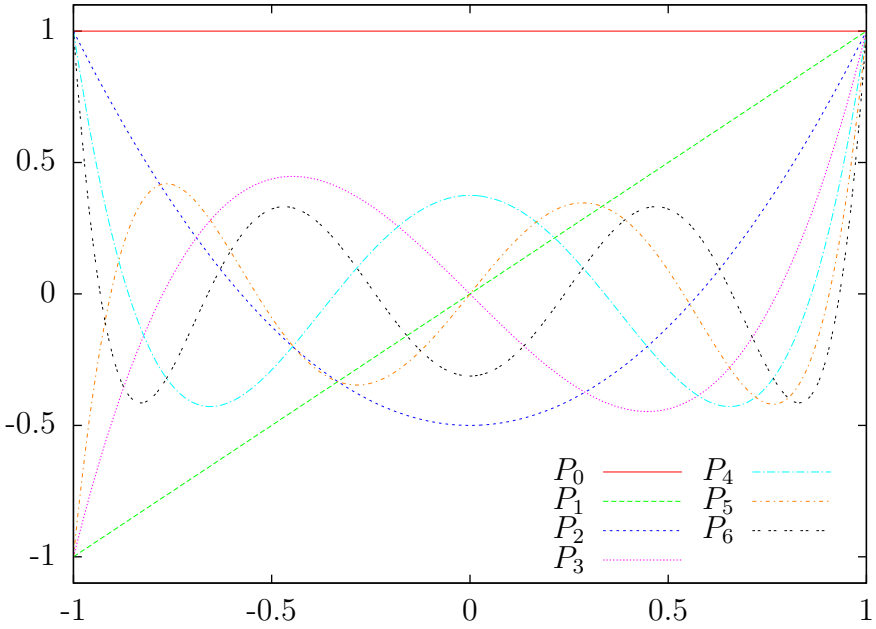


FIGURE A.1 – Polynômes de Legendre de degré 0 à 6

Chaque polynôme $P_n(x)$ est ensuite multiplié par $\sqrt{n + 1/2}$ pour être normalisé à 1 en norme L^2, c'est à dire pour avoir

$$\int_{-1}^{1} P_n(x)P_k(x)\,dx = \delta_{n,k}. \tag{A.5}$$

La famille $\{P_k\}_{k=0}^{n}$ forme alors une base orthonormée de $\mathbb{P}^n([-1\ ;\ 1])$.

Annexe B

Multi-ondelettes : courbes et équations 1D

Cette annexe présente les équations des polynômes par morceaux que sont les multi-ondelettes et les courbes associées pour quelques cas.

Pour des polynômes de degré n il y a toujours $n+1$ multi-ondelettes en une dimension dont l'expression dépend du degré. On ne peut pas se contenter de compléter la base de multi-ondelettes du degré $n-1$. En voici quelques exemples (en simple précision) donnés pour $x \in]0\ ;\ 1]$. Les fonctions ψ_i peuvent être étendues sur $[-1\ ;\ 0[$ par la relation $\psi_i(x) = (-1)^{i+n}\psi_i(-x)$.

$n = 0$:

$\psi_0(x) = 0.707107$

$n = 1$:

$\psi_0(x) = -1.22474 + 2.44949x$

$\psi_1(x) = -1.41421 + 2.12132x$

$n = 2$:

$\psi_0(x) = 0.235702 - 5.65685x + 7.07107x^2$

$\psi_1(x) = 1.83712 - 9.79796x + 9.18559x^2$

$\psi_2(x) = 2.10819 - 7.90569x + 6.32456x^2$

$n = 3$:

$\psi_0(x) = -0.664211 - 2.65684x + 19.9263x^2 - 18.5979x^3$

$\psi_1(x) = -0.617213 + 16.2019x - 46.291x^2 + 32.4037x^3$

$\psi_2(x) = -2.5365 + 24.3504x - 53.2665x^2 + 32.4672x^3$

$\psi_3(x) = -2.76026 + 18.1142x - 33.1231x^2 + 18.1142x^3$

$n = 4$:

$\psi_0(x) = 0.0733236 + 2.19971x + 15.3979x^2 - 61.5918x^3 + 46.1938x^4$

$\psi_1(x) = -0.405554 - 11.6799x + 93.6829x^2 - 181.688x^3 + 102.199x^4$

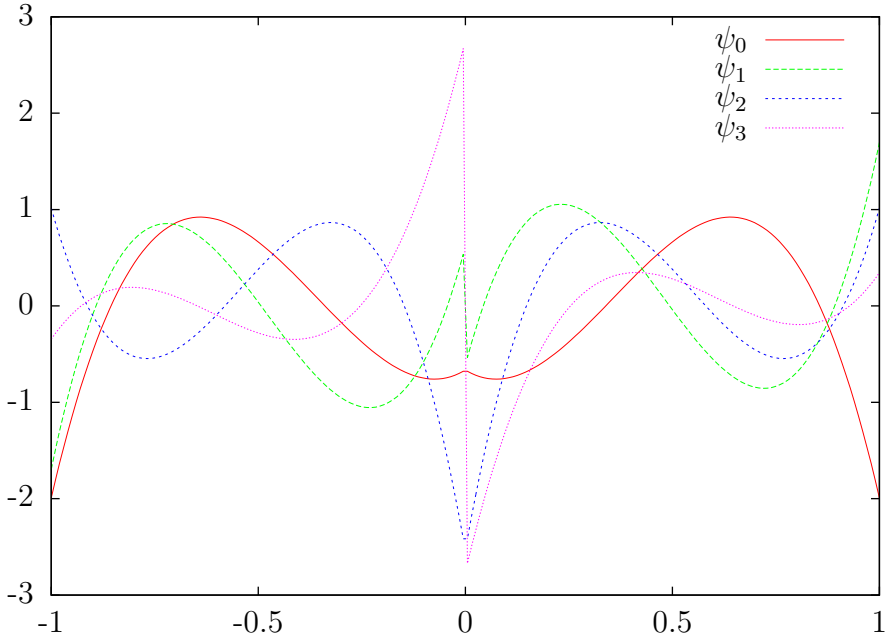


FIGURE B.1 – Multi-ondelettes pour $n = 3$

$\psi_2(x) = 1.07371 - 35.8716x + 171.013x^2 - 266.475x^3 + 131.773x^4$
$\psi_3(x) = 3.25234 - 47.5771x + 175.626x^2 - 237.885x^3 + 107.327x^4$
$\psi_4(x) = 3.36775 - 33.1513x + 101.033x^2 - 121.555x^3 + 50.5163x^4$

$n = 5$:

$\psi_0(x) = -0.83666 + 1.67332x + 3.11116e - 10x^2 + 66.9328x^3 - 175.699x^4 + 110.439x^5$
$\psi_1(x) = 0.258295 - 4.52016x + 101.251x^2 - 433.935x^3 + 632.822x^4 - 298.33x^5$
$\psi_2(x) = -0.0681925 - 30.5502x + 306.866x^2 - 941.446x^3 + 1136.43x^4 - 473.217x^5$
$\psi_3(x) = -1.55503 + 66.7718x - 464.902x^2 + 1188.34x^3 - 1278.08x^4 + 490.628x^5$
$\psi_4(x) = -3.94177 + 80.0867x - 433.595x^2 + 961.041x^3 - 939.455x^4 + 336.364x^5$
$\psi_5(x) = -3.93893 + 53.314x - 236.336x^2 + 462.055x^3 - 413.587x^4 + 138.616x^5$

$n = 6$:

$\psi_0(x) = 0.0213961 - 7.18908x + 16.1754x^2 - 35.9454x^3 + 247.125x^4 - 474.479x^5 + 257.01x^6$
$\psi_1(x) = 0.958514 - 8.76356x + 86.2663x^2 - 613.449x^3 + 1739.7x^4 - 2024.38x^5 + 822.405x^6$
$\psi_2(x) = 0.551042 - 21.5953x + 371.121x^2 - 1986.65x^3 + 4419.56x^4 - 4342.76x^5 + 1562.18x^6$
$\psi_3(x) = 0.297951 - 61.7468x + 786.591x^2 - 3414.24x^3 + 6604.09x^4 - 5873.21x^5 + 1959.92x^6$
$\psi_4(x) = 2.02678 - 109.772x + 1034.11x^2 - 3805.77x^3 + 6600.31x^4 - 5423.35x^5 + 1703.33x^6$
$\psi_5(x) = 4.59346 - 122.174x + 895.725x^2 - 2850.73x^3 + 4478.63x^4 - 3420.88x^5 + 1015.16x^6$
$\psi_6(x) = 4.48197 - 78.8634x + 470.607x^2 - 1314.39x^3 + 1882.43x^4 - 1340.68x^5 + 376.486x^6$

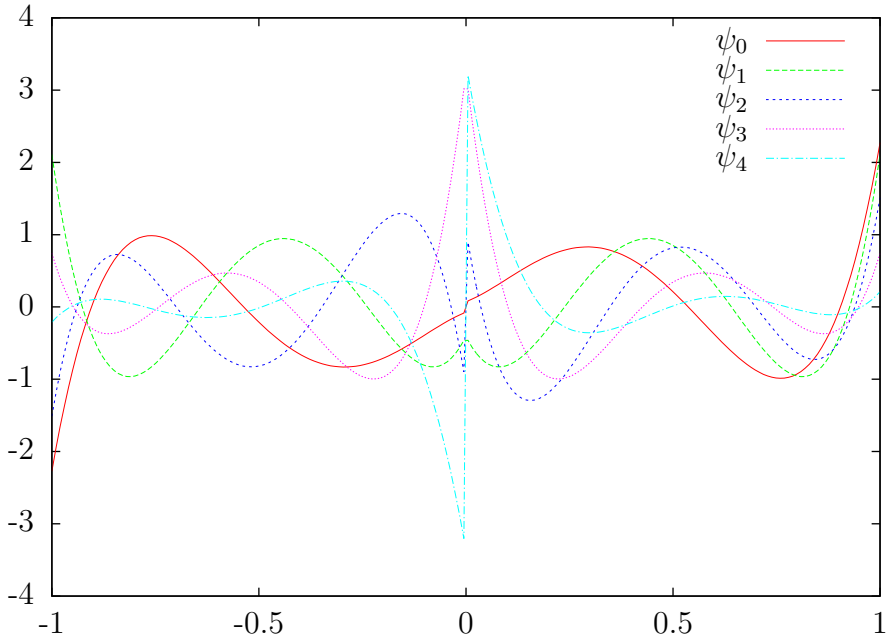


FIGURE B.2 – Multi-ondelettes pour $n = 4$

$$\begin{pmatrix}
0.707107 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.612372 & 0.353553 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.684653 & 0.176777 & 0 & 0 & 0 & 0 \\
0.233854 & 0.405046 & -0.522913 & 0.0883883 & 0 & 0 & 0 \\
0 & 0.153093 & 0.592927 & -0.35078 & 0.0441942 & 0 & 0 \\
-0.146575 & -0.253876 & -0.163876 & 0.581703 & -0.219863 & 0.0220971 & 0 \\
0 & -0.0689981 & -0.267229 & -0.421586 & 0.478033 & -0.132121 & 0.0110485
\end{pmatrix}$$

TABLE B.1 – Matrice des filtres H^0 pour $n = 6$.

Les multi-ondelettes pour $n = 3$ sont tracées sur la figure B.1 et celles pour $n = 4$ sont tracées sur la figure B.2. Ces figures correspondent à la figure 2 de [6].

La table B.1 donne la matrice des filtres H^0 pour $n \leq 6$ (obtenus par l'équation (3.18a)). Si $n < 6$, il ne faut alors considérer que les $n+1$ premières lignes et colonnes de la matrice. La matrice H^1 est obtenue par la relation (3.19a)

$$h^{(1)}_{i,j} = (-1)^{i+j} h^{(0)}_{i,j}, \quad 0 \leq i, j \leq n. \tag{B.1}$$

Les matrices des filtres G^0 pour $n = 0$ à 6 (obtenues par l'équation (3.18c)) sont présentées dans la table B.2. Les matrices G^1 correspondantes sont obtenues par la relation (3.19b)

$$g^{(1)}_{i,j} = (-1)^{i+j+k+1} g^{(0)}_{i,j}, \quad 0 \leq i, j \leq n. \tag{B.2}$$

$$\begin{pmatrix} -0.707107 \end{pmatrix}$$

(a) $n = 0$.

$$\begin{pmatrix} 0 & -0.707107 \\ 0.353553 & 0.612372 \end{pmatrix}$$

(b) $n = 1$.

$$\begin{pmatrix} 0.235702 & 0.408248 & -0.527046 \\ 0 & 0.176777 & 0.684653 \\ -0.263523 & -0.456435 & -0.471405 \end{pmatrix}$$

(c) $n = 2$.

$$\begin{pmatrix} 0 & -0.153393 & -0.594089 & 0.351468 \\ -0.154303 & -0.267261 & -0.172516 & 0.612372 \\ 0 & -0.0878669 & -0.340307 & -0.613572 \\ 0.215645 & 0.373509 & 0.443622 & 0.342327 \end{pmatrix}$$

(d) $n = 3$.

$$\begin{pmatrix} -0.146647 & -0.254 & -0.163956 & 0.581988 & -0.219971 \\ 0 & -0.0702439 & -0.272054 & -0.429198 & 0.486664 \\ 0.122012 & 0.211332 & 0.208908 & -0.0553396 & -0.627492 \\ 0 & 0.0536497 & 0.207784 & 0.439025 & 0.511082 \\ -0.184174 & -0.318999 & -0.395017 & -0.387834 & -0.240554 \end{pmatrix}$$

(e) $n = 4$.

TABLE B.2 – Matrices des filtres G^0 pour $n = 0$ à 6.

$$\begin{pmatrix} 0 & -0.0690066 & -0.267261 & -0.421637 & 0.478091 & -0.132137 \\ -0.107623 & -0.186408 & -0.182895 & 0.0569486 & 0.538114 & -0.356944 \\ 0 & -0.0417814 & -0.161819 & -0.319725 & -0.221974 & 0.566192 \\ -0.105568 & -0.182849 & -0.204677 & -0.0936565 & 0.245278 & 0.587023 \\ 0 & -0.0361235 & -0.139906 & -0.31137 & -0.469258 & -0.402451 \\ 0.161558 & 0.279826 & 0.352652 & 0.376555 & 0.319269 & 0.165851 \end{pmatrix}$$

(f) $n = 5$.

$$\begin{pmatrix} 0.10698 & 0.185295 & 0.181804 & -0.0566087 & -0.534902 & 0.354814 & -0.0771446 \\ 0 & 0.0395285 & 0.153093 & 0.301904 & 0.205396 & -0.529839 & 0.246855 \\ -0.0864762 & -0.149781 & -0.165786 & -0.0656248 & 0.221886 & 0.411325 & -0.468908 \\ 0 & -0.0286704 & -0.11104 & -0.236849 & -0.290857 & -0.0078428 & 0.588295 \\ 0.0950932 & 0.164706 & 0.194921 & 0.146795 & -0.048165 & -0.374917 & -0.511277 \\ 0 & 0.0258315 & 0.100045 & 0.227981 & 0.377817 & 0.44917 & 0.304711 \\ -0.144439 & -0.250175 & -0.318081 & -0.353198 & -0.33762 & -0.252721 & -0.113007 \end{pmatrix}$$

(g) $n = 6$.

TABLE B.2 – Matrices des filtres G^0 pour $n = 0$ à 6.

Annexe C

Code

Ce chapitre reproduit une partie du code en dimension deux. La figure C.1 présente l'organisation globale du code entre les différents fichiers. Les fichiers encadrés en rouges sont les fichiers header appartenant au standards du c. Ceux encadrés en noir sont les fichiers propres au code. Les flèches indiquent que le fichier à la base de la flèche est appelé dans le fichier à la pointe de la flèche. De nombreux commentaires parsèment les headers et servent de documentation. Ce code fonctionne avec de nombreux pointeurs et appel de la fonction `malloc` parce que l'idée initiale était d'avoir un code facilement réutilisable. Certains aspects ont été améliorés dans le code en dimension 4. De nombreuses améliorations apportées au code en dimension 4 auraient pu être incorporées au code en dimension 2 au prix d'un énorme travail de réécriture que je n'avais plus le temps de faire.

Le fichier `pre-proc-datas.h` est un fichier de paramètres à régler par l'utilisateur. Ce fichier contient le choix du cas test et du schéma, le degré des polynômes en une dimension, le nombre de points de Gauss associés et le nombre correspondant de polynômes en deux dimensions. Il contient également le pas de temps, le temps final et le nombre de pas de temps. Ces trois données peuvent être modifiées par le fichier `parameters.c`. La dernière information indiquée est le seuil de raffinement utilisé et le niveau maximal de raffinement. Toutes ces informations sont contenues sous la forme de directive de pré-processeurs.

`utils.h` inclus les headers nécessaire au fonctionnement du code. Les fonctions les plus basiques nécessaires au programme y sont aussi déclarées. Initialement, les fonctions n'étaient pas déclarées en `static inline` et le corps des fonctions se trouvait dans le fichier `utils.c`. Les fonctions ont ensuite été transformées en fonctions `static inline` à des fins d'optimisation. Dans le code en dimension 4, les fonctions de type `static inline` sont remplacées autant que possible par des macros (des directives pré-processeur).

Le module `mesh` définit une structure de pile utilisée notamment pour lister les cellules feuilles, présentée dans l'encart 4, la structure du maillage en deux dimensions, déjà introduit à l'encart 1 et rappelée dans l'encart 5, la structure du maillage à une dimension présentée dans l'encart 6. Ces structures sont accompagnées des fonctions assurant le fonctionnement du code, comme la création et la suppression de cellules, ou le recensement de toutes les cellules feuilles en dimension 2. Dans le cas de la pile, la suppression d'un élément ne supprime pas l'objet pointé dans la pile.

Le fichier `parameters.h` définit la structure contenant le domaine d'étude. Lors de l'initialisation de cette structure, le pas de temps et le temps final peuvent être modifiés

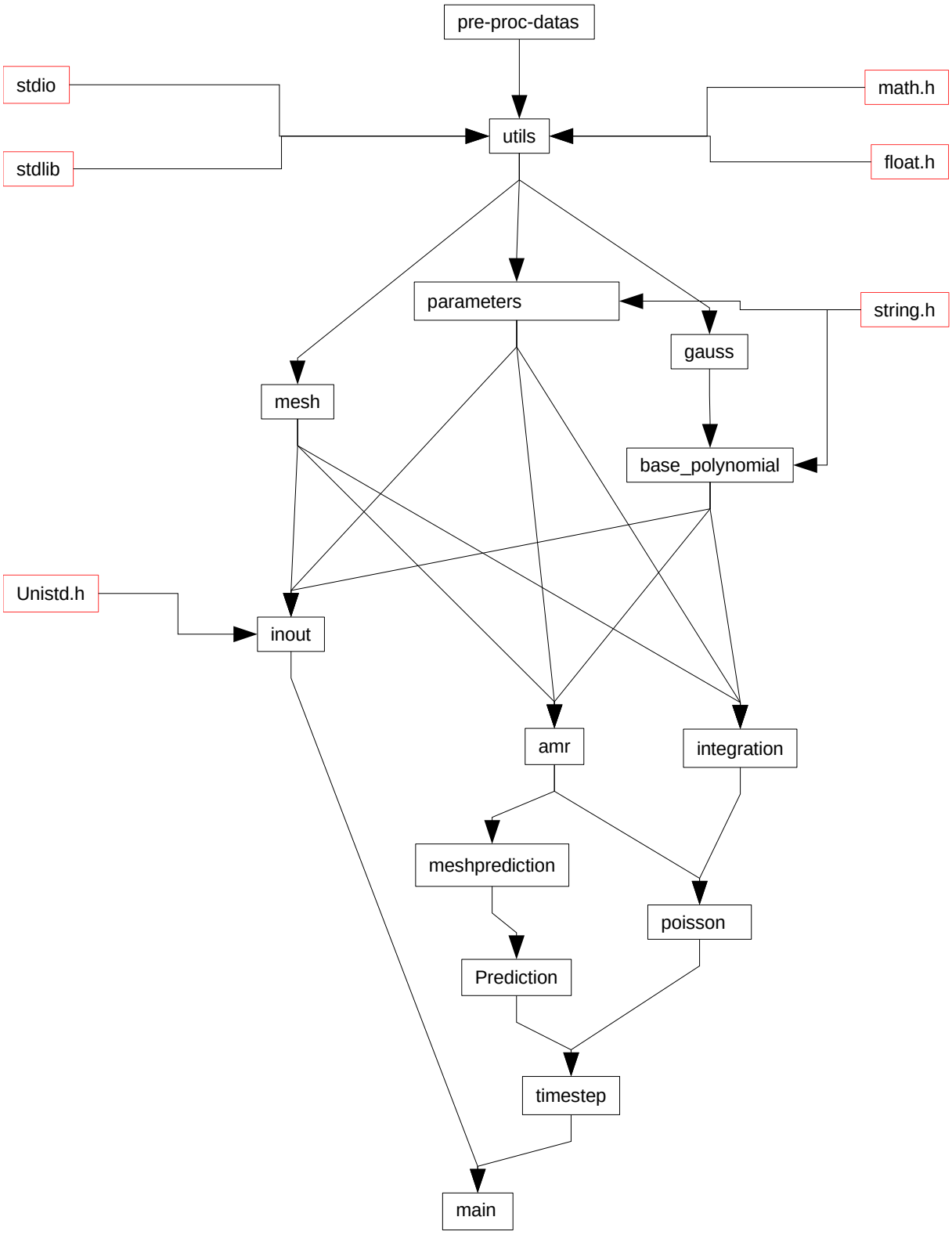


FIGURE C.1 – Organisation du code.

afin de garantir un pas de temps non nul et que le temps final réel soit au moins celui indiqué. Dans la fonction `main`, la structure des paramètres est déclarée sous la forme d'un pointeur constant sur un objet constant (sous la forme `st_Parameters const * const param;`). Une fonction d'initialisation est donc également déclarée dans le fichier `parameters.h`. Ce fichier contient également une fonction permettant de sauvegarder l'ensemble des paramètres utilisés dans un fichier de données à part et appelée automatiquement à l'initialisation, afin d'avoir toutes les informations relative à la simulation accessibles.

Le fichier `gauss.h` définit la structure contenant les points et poids de Gauss sur l'intervalle de référence à une dimension. On définit également une fonction pour générer la structure et une fonction pour l'effacer. Dans le code 2D, les points et poids sont stockés dans un pointeur alloué à la taille souhaitée, dans le code 4D la structure contient des tableaux de taille fixée. Les points et poids de Gauss sont stockés en dur pour un nombre de points allant de 1 à 64 dans le fichiers `gauss.c`. La structure contient également le produit des poids.

La structure des polynômes est décrite dans le fichier `base_polynomial.h`. Chaque polynôme de Legendre est décrit sur la base $\{1, x, \ldots, x^n\}$. Chacun des polynômes est interpolé aux points de Gauss et aux extrémités de l'intervalle. Pour chaque polynôme ont calcul également sa dérivée et sa primitive. La dérivée sert pour la méthode GDSL et est interpolée aux points de Gauss. La primitive sert à l'optimisation du calcul comme expliquée à la section 4.4. Les multi-ondelettes et les filtres $h_{i,j}^{(0)}$, $h_{i,j}^{(1)}$, $g_{i,j}^{(0)}$ et $g_{i,j}^{(1)}$ sont également stockés dans la structure. Dans la structure polynomiale en dimension 4 présentée à l'encart 11, `NB_SPACE_COEFFS` = (`MAX_DEGREE`+1)^2, et `FULL_COEFFS` = (`MAX_DEGREE`+1)^4.

Le fichier `inout.h` et le fichier `inout.c` contiennent respectivement les prototypes et les corps des fonctions de sortie du code. Ce sont ces fonctions qui sauvegardent les valeurs instantanées de la distribution, calculent et sauvegardent les différentes normes et diagnostiques.

Les fichiers `amr.h` et `amr.c` contiennent les fonctions permettant de restreindre la distribution à des niveaux plus grossiers ou d'interpoler vers des niveaux plus fins.

Les fichiers `integration.h` et `integration.c` contiennent les fonctions permettant de faire le calcul des intégrales des équations (2.51) et (2.53), et les étapes 2 à 5 de la méthode CGD.

Le fichier `poisson.c` contient les fonctions résolvant le problème de Poisson pour le cas Vlasov-Poisson standard et pour le cas Vlasov-Poisson pseudo-polaire.

Le module `prediction` calcule les caractéristiques lors de l'étape de prédiction.

Les fichiers `meshprediction.h` et `meshprediction.c` contiennent les prototypes et le corps des fonctions permettant de générer le maillage prédit à partir du calcul des caractéristiques puis de copier le maillage prédit sur la structure portant la distribution. Ces fichiers contiennent également les fonctions pour tester la condition de déraffinement (3.42) et le déraffinement échéant.

Le module `timestep` contient les fonctions de splitting pour les deux schémas GDSL et CGD. Deux fonctions (une pour la méthode CGD et une pour la méthode GDSL) permettent de résoudre le cas test de rotation, deux fonctions résolvent le cas test de rotation déformante, deux fonctions permettent de résoudre les divers cas test de Vlasov-Poisson en physique des plasmas et en astrophysique, et deux fonctions résolvent les cas

test de Vlasov-Poisson pseudo polaire.

Code 2 Aperçu de `pre-proc-datas.h`

```
#define SLDG 1
#define CG 2
// choisi le schéma en temps, GDSL ou CGD
#define TIME_SCHEME CG

#define ROTATION  1 // rotations
#define WLD    2 // weak Landau damping, amortissement Landau linéaire
#define SLD    3 // strong Landau damping, amortissement Landau fort
#define TSI    4 // two string instability, instabilité double faisceau
#define BOT    5 // Bump on Tail
#define VPc    9 // Vlasov-Poisson pseudo-polaire
#define ASTRO  10 // astrophysique
// choix du cas test
#define TEST_CASE __DEFAULT__

/*
Précision du cas test
différents cas tests de rotation et rotation-déformation
choix du cas test de Vlasov-Poisson pseudo-polaire
choix du cas test astrophysique
...
*/
#define  SUB_CASE __DEFAULT__

#define MAX_DEGREE 4 // maximum degree
#define NB_GAUSS_PTS 5 // maximum degree+1
#define NB_POL_2D 25 // (maximum degree+1)^2

#define DT (double)(0.1) // longueur du pas de temps
#define NT 1 // nombre de pas de temps
#define FINAL_TIME 100 // temps final

#define THRESHOLD (double)(1.e6) // seuil de raffinement
#define MAX_LVL (unsigned char)(8) // niveau maximum de raffinement
```

Code 3 Aperçu du fichier `utils.h`

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <float.h>
#include <errno.h>
#include <time.h>
#include <omp.h>
#include "pre-proc-datas.h"

// définition de l'élément de référence
#define REF_EL_MIN (double)(-1)
#define REF_EL_MAX (double)(1)
#define REF_EL_LENGTH (double)(2) // longueur de l'élément de référence
#define REF_EL_VOL (double)(4)
// volume de l'élément de référence en dimension 2

#define PI (double)(3.14159265358979323846264338327950288419716939937510 58)
// depuis Wolfram Alpha
#define TWO_PI (double)(6.28318530717958647692528676655900576839433879875021 16)
// depuis Wolfram Alpha, 2*pi

// code d'erreur
#define ERROR printf("\nERROR in file %s, function %s, line %d.\n",
__FILE__, __func__, __LINE__);
#define QUIT printf("#Exiting...\n"); exit(EXIT_FAILURE);
#define FERROR(...) ERROR printf(__VA_ARGS__); QUIT

// comparateurs, renvoient 1 si vrai, 0 si faux
// a==b avec une tolérance EPSILON
static inline char AreEqual(double const a, double const b);
// a<b avec une tolérance EPSILON
static inline char LessThan(double const a, double const b);
// a<=b avec une tolérance EPSILON
static inline char LessEqual(double const a, double const b);
// a>b avec une tolérance EPSILON
static inline char MoreThan(double const a, double const b);
// a>=b avec une tolérance EPSILON
static inline char MoreEqual(double const a, double const b);

// projecteurs
// projette x depuis [inf ; sup] vers [REF_EL_MIN ; REF_EL_MAX]
static inline double ProjectorToRef(double const x, double const inf,
                                                    double const sup);
//projette x depuis [REF_EL_MIN ; REF_EL_MAX] vers [inf ; sup]
static inline double ProjectorFromRef(double const x, double const inf,
                                                      double const sup);
```

Code 4 Aperçu de la structure de pile dans le fichier `mesh.h`

```
struct Stack
{
  Stack *m_prev, *m_next; // élément précédent et élément suivant de la pile
  void *m_obj; // élément stocké dans la pile
};

// manipulation de la pile ;
// renvoit toujours le premier élément de la pile

// ajoute un objet au début de la pile
Stack* AddAtFirst(void *obj, Stack *stackIn);
// ajoute un objet juste après l'objet donné
Stack* AddAfter(void *obj, Stack *stackIn);
// ajoute un objet à la fin de la pile
Stack* AddAtEnd(void *obj, Stack *stackIn);

// supprime le premier élément de la pile
Stack* RemoveFirst(Stack* stackIn);
// supprime l'élément donné de la pile à laquelle il appartient
Stack* RemoveHere(Stack* stackIn);
// supprime le dernier élément de la pile
Stack* RemoveLast(Stack *stackIn);
```

Code 5 Aperçu de la structure du maillage à deux dimensions dans le fichier `mesh.h`

```
struct st_Mesh2D
{
  // données d'arbre
  unsigned char m_lvl; // niveau de la cellule
  st_Mesh2D *m_parent; // cellule mère
  st_Mesh2D *m_children[4]; // cellules filles
  char m_leaf; // 1 si la cellule est une feuille, 0 sinon

  // données géométriques
  double m_boundaries[2][2]; // bordure de la cellule
  double m_centre[2]; // coordonnées du centre
  double m_length[2]; // longueur de la cellule

  // distribution au temps n et au temps n+1
  double *m_dist, *m_distnp1;
  // coefficients sur la base des multi-ondelettes
  double *m_mwdist;
  // dans le code 4D, les coefficients d'échelle sont stockés sous la forme
  // double m_dist[NB_4D_COEFFS], m_distnp1[NB_4D_COEFFS];
  // et les coefficients des multi-ondelettes ne sont pas stockés
  // puisqu'ils portent des détails considérés négligeables
};

// alloue et initialise une nouvelle cellule
st_Mesh2D* New2DCell(unsigned char const lvl); // level of the new cell

// désalloue la cellule pointée et tout ses enfants,
// mais ne met pas à jour la cellule mère
void DeleteM2D(st_Mesh2D *mycell);

// raffine la cellule pointée en créant les enfants,
// mais ne projette pas la distribution
void Refine2DCell(st_Mesh2D *myCell); // leaf cell to refine (i for input)
// déraffine une cellule avec quatre cellules filles
// la distribution doit être restreinte avant l'appel de cette fonction
void Coarsen2DCell(st_Mesh2D *myCell); // cell to coarsen

// détecte toutes les cellules filles et les renvoie sous forme d'un tableau
// renvoie également le nombre de cellules feuilles
void AllLeaf2D(st_Mesh2D *myCell, // cellule mère
               st_Mesh2D ***out, // tableau dynamique de cellules feuilles
               unsigned long int *length // nombre de cellules feuilles
               );

// renvoie la cellule 2D contenant le point (x1 ; x2)
st_Mesh2D* CellAt2D(double const x1, double const x2,
                    st_Mesh2D const *myCell);
```

Code 6 Aperçu de la structure du maillage à une dimension dans le fichier `mesh.h`

```
struct st_Mesh1D
{
  // données d'arbre
  unsigned char m_lvl; // niveau de la cellule
  st_Mesh2D *m_parent; // cellule mère
  st_Mesh2D *m_children[2]; // cellules filles
  char m_leaf; // 1 si la cellule est une feuille, 0 sinon

  // cellules suivante et précédente dans la direction x
  st_Mesh1D *m_prev, *m_next;

  // données géométriques
  double m_boundaries[2];
  double m_centre, m_length;

  // densité, potentiel et champ électrique
  double *m_rho, *m_phi, *m_e;
};

// alloue une nouvelle cellule avec le niveau indiqué
st_Mesh1D* New1DCell(unsigned char const lvl);

// désalloue la cellule pointée et tout ses enfants,
// mais ne met pas à jour la cellule mère
void DeleteM1D(st_Mesh1D *mycell);

// raffine la cellule pointée en créant les enfants,
// mais ne projette pas la distribution
void Refine1DCell(st_Mesh1D *myCell);
// détecte toutes les cellules filles et les renvoie sous forme d'un tableau
// renvoie également le nombre de cellules feuilles
void Coarsen1DCell(st_Mesh1D *myCell);
```

Code 7 Aperçu du fichiers `parameters.h`

```
struct st_Parameters
{
  // degré maximum des polynômes 1D
  unsigned char m_maxDegree; // n'existe plus dans le code 4D
  double m_bounds[2][2]; // limites du domaine dans chaque direction
  double m_length[2]; // longueur du domaine dans chaque direction
  double m_dt, m_tFinal; // longueur du pas de temps et temps final
  unsigned int m_nt; // nombre de pas de temps
  // fréquence des diagnostiques et des instantanés pour la distribution
  unsigned int m_outShort, m_outLong;
  double m_threshold; // seuil de raffinement
};

// génération de la structure
st_Parameters* GenerateParameters();

// sauvegarde de la structure
void SaveParam(st_Parameters const * const param);
```

Code 8 Aperçu de la fonction `st_Parameters* GenerateParameters()`

```
st_Parameters* GenerateParameters()
{
  st_Parameters *param=malloc(sizeof(st_Parameters));
  if (param==NULL) {
    printf("ERROR:\nunable to allocate param.\n");
    QUIT
  }

  /*
  selection du domaine à l'aide de directive pré-processeur
  puis calcul de la longueur du domaine
  */
#if TEST_CASE==...
  param->m_bounds[0][0]=...
  param->m_bounds[0][1]=...
  param->m_bounds[1][0]=...
  param->m_bounds[1][1]=...
#elif TEST_CASE=...
...
#endif
  param->m_length[0]=param->m_bounds[0][1]-param->m_bounds[0][0];
  param->m_length[1]=param->m_bounds[1][1]-param->m_bounds[1][0];

  // ajustement du pas de temps, du temps final et du nombre de pas de temps
  if (DT==0)
    param->m_dt=FINAL_TIME/D_(NT);
  else
    param->m_dt=DT;
  param->m_nt=NT;
  param->m_tFinal=FINAL_TIME;
  // empêche le mauvais calcul des paramètres d'entrée en
  // utilisant le temps le plus long
  param->m_nt=imax(param->m_nt, ceil(param->m_tFinal/param->m_dt));
  param->m_tFinal=param->m_dt*D_(param->m_nt);

  SaveParam(param);

  return param;
}
```

Code 9 Aperçu du fichier `gauss.h`

```
struct st_GaussLegendre
{
  unsigned char m_n; // nombre de points en dimension 1
  // supprimé dans le code 4D
  double *m_xi, *m_wi; // points et poids
  // remplacé par m_xi[NB_GAUSS_PTS] et m_wi[NB_GAUSS_PTS] dans le code 4D
  double **m_wij; // produit m_wi[i]*m_wi[j]
  // remplacé par m_wij[NB_GAUSS_PTS][NB_GAUSS_PTS] dans le code 4D
};
```

Code 10 Aperçu du fichier `base_polynomial.h en 2D`

```
struct st_Polynomial
{
  // Polynômes
  // double pointeur pour les polynômes 1D
  double **m_basis1D;
  // double pointeurs pour les polynômes 2D
  unsigned char **m_basis2D;

  // pre-calculs
  // dérivation
  unsigned char m_nbPtsDeriv; // nombre de points pour la dérivée
  double *m_ptsDeriv;
  double **m_valDeriv; // m_valDeriv[i][j] = P_i'(x_j)
  double **m_deriv; // m_deriv[i] = P_{i+1}'(x)
  // interpolation
  unsigned char m_nbPtsInterp; // nombre de points pour l'interpolation
  double *m_ptsInterp;
  double **m_valInterp; // m_valInterp[i][j] = P_i(x_j)
  // primitives
  double **m_valPrim; // primitives des polynômes
  double **m_valPrimInterp; // interpolations des primitives

  // Multi-ondelettes,

  // m_basis1D_mw[k][0][i] représente le polynomial k,
  // coefficient sur x^i pour x<0
  // m_basis1D_mw[k][1][i] représente le polynomial k,
  // coefficient sur x^i pour x>0
  double ***m_basis1D_mw;

  // coefficients to change level
  // in 1D
  double **m_h0, **m_h1, **m_g0, **m_g1;
}
```

Code 11 Aperçu du fichier `base_polynomial.h` en 4D

```
struct st_Poly
{
  // coefficients des polynômes 1D
  double m_1Dbasis[MAX_DEGREE+1][MAX_DEGREE+1];
  // définition des polynômes 2D et 4D
  unsigned char m_2Dbasis[NB_SPACE_COEFFS][2];
  unsigned char m_4Dbasis[FULL_COEFFS][4];

  // dérivées sur la base (1, x,...,x^n)
  double m_derivBasis[MAX_DEGREE+1][MAX_DEGREE];
  // dérivées sur la base de Legendre
  double m_derivProj[MAX_DEGREE+1][MAX_DEGREE];
  // primitives sur la base (1, x,...,x^n)
  double m_primBasis[MAX_DEGREE+1][MAX_DEGREE+2];

  // estimation des polynômes aux points de Gauss et aux extrémités
  // de l'intervalle de référence
  double m_eval1d[MAX_DEGREE+1][NB_GAUSS_PTS+2];
  // estimation des dérivées aux mêmes points
  double m_evalDer[MAX_DEGREE+1][NB_GAUSS_PTS+2];
  // estimation des primitives aux mêmes points
  double m_evalPrim[MAX_DEGREE+1][NB_GAUSS_PTS+2];

  // coefficients des multi-ondelettes
  double m_mwb1d[MAX_DEGREE+1][2][MAX_DEGREE+1];
  // composition de l'espace des multi-ondelettes W en 4D
  // 0 pour les fonctions d'échelles et 1 pour les multi-ondelettes
  char m_mwb4d[15][4];
  // pre-calculs pour l'opération de restriction
  double m_restric[16][FULL_COEFFS][FULL_COEFFS][16];
  // pre-calculs pour l'opération de projection
  double m_projec[16][NB_PHASE_COEFFS][NB_PHASE_COEFFS];
  // estimations aux points de gauss sur [-1 ; 0] et sur [0 ; 1]
  double m_mweval[MAX_DEGREE+1][2*NB_GAUSS_PTS]; // polynômes
  double m_eval1dmw[MAX_DEGREE+1][2*NB_GAUSS_PTS]; // multi-ondelettes

  // filtres
  double m_h[2][NB_GAUSS_PTS][NB_GAUSS_PTS], m_g[2][NB_GAUSS_PTS][NB_GAUSS_PTS];
};
```

Annexe D

Méthode de Galerkin discontinue par élément spectraux

Ce chapitre porte sur le travail réalisé dans le cadre de mon stage de deuxième année de master et de troisième année d'école d'ingénieur sous la tutelle de messieurs Sonnendrücker Eric et Restelli Marco au Max-Planck-Institut für Plasmaphysik (IPP) à Garching (à côté de Munich) [2]. L'objet du stage était de développer une méthode Galerkin discontinue spectrale par élément pour la résolution du problème de Vlasov-Poisson. Durant mon stage, nous avons développé une méthode en dimensions deux dans l'espace des phases. Cette réalisation a fait l'objet de mon rapport de stage [31]. Ce code a ensuite été étendu au problème en dimension quatre. Les résultats finaux ont été publiés dans le journal of Computational Physics et est cité ci-dessous. Cet article est également cité dans la thèse avec l'indice bibliographique [32].

Journal of Computational Physics 279 (2014) 261–288

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Energy conserving discontinuous Galerkin spectral element method for the Vlasov–Poisson system

Éric Madaule [a], Marco Restelli [b,*], Eric Sonnendrücker [b,c]

[a] Institut Jean Lamour, Université de Lorraine, CNRS, UMR 7198, France
[b] Max-Planck-Institut für Plasmaphysik, Boltzmannstr. 2, 85748 Garching, Germany
[c] TU Munich, Mathematics Center, Boltzmannstr. 3, 85747 Garching, Germany

ARTICLE INFO

Article history:
Received 9 May 2014
Received in revised form 11 August 2014
Accepted 15 September 2014
Available online 22 September 2014

Keywords:
Vlasov–Poisson system
Discontinuous Galerkin method
Spectral element method

ABSTRACT

We propose a new, energy conserving, spectral element, discontinuous Galerkin method for the approximation of the Vlasov–Poisson system in arbitrary dimension, using Cartesian grids. The method is derived from the one proposed in [4], with two modifications: energy conservation is obtained by a suitable projection operator acting on the solution of the Poisson problem, rather than by solving multiple Poisson problems, and all the integrals appearing in the finite element formulation are approximated with Gauss–Lobatto quadrature, thereby yielding a spectral element formulation. The resulting method has the following properties: exact energy conservation (up to errors introduced by the time discretization), stability (thanks to the use of upwind numerical fluxes), high order accuracy and high locality. For the time discretization, we consider both Runge–Kutta methods and exponential integrators, and show results for 1D and 2D cases (2D and 4D in phase space, respectively).

1. Introduction

The Vlasov–Poisson system describes the behaviour of a collisionless plasma subject to electrostatic effects in terms of the corresponding distribution function in phase space, thus representing one of the basic models in plasma physics. Due to the high dimensionality of the problem, efficient numerical techniques for the computation of approximate solutions are of paramount importance. Moreover, such techniques should ideally reproduce the main features of the system, namely: the absence of dissipation, the stability of the solution and the conservation of physical invariants such as the total number of particles, the total energy and the total momentum. In the literature, various approaches have been proposed, corresponding to different trade-offs concerning the fulfillment of these requirements. Semi-Lagrangian schemes emphasize computational efficiency, obtained using large time steps [13,19,27,29,48,52]. Conservative finite difference formulations, constructed in terms of the Hamiltonian formulation of the problem, result in conservation equations for the physical invariants of the discrete system which parallel those of the continuous one [36]. Discontinuous Galerkin methods, finally, have been considered because of their locality, which allows optimal scaling in massively parallel computations, as well as their accuracy and flexibility in terms of the computational grid [3–5,9,10,12,29,32,44].

In this paper, we consider a discontinuous Galerkin (DG) spectral element method for the Vlasov–Poisson system with the introduction of a local reconstruction of the electric field which ensures exact discrete energy conservation up to errors

* Corresponding author.
E-mail addresses: eric.madaule@univ-lorraine.fr (É. Madaule), marco.restelli@ipp.mpg.de (M. Restelli), eric.sonnendruecker@ipp.mpg.de (E. Sonnendrücker).

http://dx.doi.org/10.1016/j.jcp.2014.09.010
0021-9991/

introduced by the time discretization. This choice is motivated by the following considerations. The DG method provides the highest locality of the numerical discretization, representing an optimal choice for massively parallel computations, it combines the use of upwind numerical fluxes with high order accuracy, so that the numerical solution tends to be stable but not excessively dissipative, and it permits the use of nonuniform and even nonconforming grids [7,40]. Moreover, the DG discretization ensures discrete conservation of the total number of particles in a very natural way. The spectral element version of the DG method is obtained by using a quadrature formula for the computation of the integrals appearing in the finite element formulation and by collocating the discrete degrees of freedom at the quadrature nodes [6,35] and, for high polynomial orders, results in a more efficient method compared to the standard DG one. The spectral element DG method has been used extensively for fluid dynamics computations where, as is the case for the Vlasov–Poisson system, the flow regime is strongly advection dominated but yet the solution does not develop shocks (for instance, see [46] for the case of atmospheric flows). The use of a special reconstruction of the electric field computed in the solution of the Poisson problem is introduced to compensate the spurious energy sources resulting from the upwind numerical flux employed in the DG method, as we discuss in details in the sequel of the paper; such a reconstruction is local (i.e. defined at the element level), in order to preserve the locality of the formulation, and consistent with the accuracy of the scheme, so that it does not degrade the overall convergence rate.

The resulting formulation can be regarded as a modification of the method proposed and analyzed in [3–5], the main modifications consisting in the facts that a) the method is of spectral element type and b) energy conservation is achieved using a local reconstruction of the electric field, instead of solving multiple Poisson problems. We notice that these two aspects are independent from each other, and in particular the energy conserving reconstruction of the electric field is also applicable in the framework of a standard DG method. We also observe that an alternative approach to obtain energy conserving formulations for the Vlasov–Poisson system can be found in [9] (these ideas are then extended in [10] to the Maxwell system), where the starting point for the numerical discretization is provided by Ampère's law and where energy conserving schemes are presented also for the completely discretized problem in space and time. The counterpart to this is that, by discretizing Ampère's law, charge conservation is not preserved for the numerical solution and most of the time marching schemes discussed in [9] are implicit; in this respect, the pros and the cons of the method of [9] are dual compared with those of the scheme discussed in the present paper.

After discretizing the Vlasov–Poisson system in space with the spectral element DG formulation, we obtain an Ordinary Differential Equation (ODE) which is then integrated in time following the classical method of lines. Typically, this is done with explicit Runge–Kutta (RK) methods, along the lines of the RKDG method introduced in the nowadays classical series of papers [16,18,21,22,24]. This leads to an accurate and stable formulation, which computational cost however tends to be adversely affected by a somewhat stringent stability condition on the time-step, especially for high order polynomial spaces. A possible alternative to this strategy is represented by the use of exponential time integrators [37,53], and we present some preliminary comparisons between RK and exponential time marching schemes in the present paper. An alternative approach, which we postpone for further investigation, is the use of semi-Lagrangian DG methods [44,45].

The remaining of the paper is organized as follows. In Section 2, the Vlasov–Poisson system is introduced. The DG numerical discretization is discussed in Section 3, which mostly follows the ideas of [3,4]. Section 4 is devoted to the illustration of the local reconstruction strategy for the electric field. Section 5 then addresses all the aspects related with the spectral element formulation. In Section 6 we summarize the time marching schemes considered for the time integration of the resulting ODE system. The resulting numerical scheme is validated numerically in Section 7, and finally some conclusions are drawn in Section 8.

2. The Vlasov–Poisson system

In d spatial dimensions, the Vlasov–Poisson system is

$$\begin{aligned} &\partial_t f + \boldsymbol{v}\cdot\nabla_{\boldsymbol{x}} f - \boldsymbol{E}\cdot\nabla_{\boldsymbol{v}} f = 0 \\ &\nabla_{\boldsymbol{x}}\cdot\boldsymbol{E} = 1-\rho, \end{aligned} \tag{2.1}$$

where $f = f(t,\boldsymbol{x},\boldsymbol{v})$ is the particle distribution function, $\boldsymbol{E} = \nabla_{\boldsymbol{x}}\Phi$ is the electric field, $\Phi = \Phi(t,\boldsymbol{x})$ being the electrostatic potential, and $\rho = \rho(t,\boldsymbol{x})$ is the particle density. Eq. (2.1), supplemented with an initial condition $f(0,\boldsymbol{x},\boldsymbol{v}) = f_0(\boldsymbol{x},\boldsymbol{v})$, must be solved for times $t\in(0,T]$ and phase space coordinates $(\boldsymbol{x},\boldsymbol{v})\in\Omega_{\boldsymbol{x}}\times\Omega_{\boldsymbol{v}} = \Omega$. Here we assume that $\Omega_{\boldsymbol{x}} = (0,L_{\boldsymbol{x}})^d$ and $\Omega_{\boldsymbol{v}} = (-\frac{L_{\boldsymbol{v}}}{2},\frac{L_{\boldsymbol{v}}}{2})^d$ and enforce periodic boundary conditions on both $\partial\Omega_{\boldsymbol{x}}$ and $\partial\Omega_{\boldsymbol{v}}$; vector components in $\Omega_{\boldsymbol{x}}$ and $\Omega_{\boldsymbol{v}}$ are denoted by a Greek superscript, such as x^α or v^β, for $1\le\alpha,\beta\le d$. From a physical viewpoint, the periodic boundary condition represents a natural assumption for $\Omega_{\boldsymbol{x}}$, while the proper choice for the velocity space should be considering $\Omega_{\boldsymbol{v}} = \mathbb{R}^d$. However, for numerical computations, the velocity space must be truncated, which is acceptable provided that $L_{\boldsymbol{v}}$ is sufficiently large, and in this case the choice of the boundary condition on $\partial\Omega_{\boldsymbol{v}}$ is not relevant. Assuming periodic conditions then simplifies both the presentation and the numerical assessment of the conservation properties of the method. Concerning the particle density ρ, it is related to the distribution function by

$$\rho(t,\boldsymbol{x}) = \int_{\Omega_{\boldsymbol{v}}} f(t,\boldsymbol{x},\boldsymbol{v})\,\mathrm{d}\boldsymbol{v}. \tag{2.2}$$

In terms of the electric potential, $(2.1)_2$ can be written as

$$\Delta_{\boldsymbol{x}}\Phi = 1 - \rho, \tag{2.3}$$

which, together with the periodic boundary conditions on $\partial\Omega_{\boldsymbol{x}}$, requires the compatibility condition

$$\int_{\Omega_{\boldsymbol{x}}} \rho(t,\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \iint_{\Omega} f(t,\boldsymbol{x},\boldsymbol{v})\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = L_{\boldsymbol{x}}^{d}. \tag{2.4}$$

Since, as it will be discussed shortly, (2.1) implies that the integrals appearing in (2.4) are constant in time, it suffices to assume that (2.4) is satisfied for the initial condition f_0. Moreover, to provide a unique definition of the electrostatic potential, we require

$$\int_{\Omega_{\boldsymbol{x}}} \Phi(t,\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 0, \tag{2.5}$$

thereby completing the definition of the continuous problem.

The Vlasov–Poisson system is characterized by a number of conservation properties, which we summarize here since one of the focuses of this article is investigating to what extent they are preserved by the numerical discretization. Let us first consider the weak form of the Vlasov equation, obtained multiplying $(2.1)_1$ by a test function $u = u(\boldsymbol{x},\boldsymbol{v})$ and integrating over the phase space,

$$\frac{d}{dt}\iint_{\Omega} f u\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} - \iint_{\Omega} \boldsymbol{v} f\cdot\nabla_{\boldsymbol{x}} u\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \iint_{\Omega} \boldsymbol{E} f\cdot\nabla_{\boldsymbol{v}} u\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = 0. \tag{2.6}$$

Taking now $u = 1$ we obtain

$$\iint_{\Omega} f\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = const, \tag{2.7}$$

representing the physical constraint that no particles are created or destroyed. As anticipated, this also shows that the integrals appearing in (2.4) are constant in time.

Taking $u = v^{\alpha}$, for $\alpha = 1,\dots,d$, and using $(2.1)_2$, we arrive at

$$\iint_{\Omega} \boldsymbol{v} f\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = const, \tag{2.8}$$

representing the fact that the total momentum of the system, which is isolated from external forces, is conserved.

Taking $u = \frac{v^2}{2} + \Phi$, and using again $(2.1)_2$, we arrive at

$$\iint_{\Omega} \frac{v^2}{2} f\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \frac{1}{2}\int_{\Omega_{\boldsymbol{x}}} E^2\,\mathrm{d}\boldsymbol{x} = const, \tag{2.9}$$

representing the fact that the total energy of the system is conserved.

Two additional properties follow from the fact that, by virtue of $(2.1)_1$, the particle distribution function is constant along the trajectories of the vector field $(\boldsymbol{v},\boldsymbol{E})$ in the phase space. It follows that

$$\iint_{\Omega} |f|^p\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = const, \quad 1 \le p \le \infty \tag{2.10}$$

and

$$\min_{\Omega} f_0 \le f(t,\boldsymbol{x},\boldsymbol{v}) \le \max_{\Omega} f_0 \quad \forall t > 0,\ \forall(\boldsymbol{x},\boldsymbol{v}) \in \Omega. \tag{2.11}$$

Eq. (2.10) corresponds to the conservation of the system entropy, while (2.11) guarantees that the distribution function is nonnegative, provided that such is the initial condition.

Various results concerning the well-posedness of the Vlasov–Poisson system, as well as the regularity of the solution, are summarized in [3,4].

3. Discontinuous Galerkin discretization of the Vlasov–Poisson system

Relevant references concerning the introduction and the analysis of the DG method are in [16,18,21,22,24,34,47] for hyperbolic problems and [2,8,11,23] for elliptic problems. Concerning the specific application to the Vlasov–Poisson system, the DG method has then been studied in [3–5,12,32]; in particular, error estimates of order $k+1$, where k is the polynomial degree of the local finite element spaces, are shown in [3,4] for the particle distribution function and the electric field in the L^2 norm. The presentation of the method given in this section closely follows [4].

3.1. Notation

Let us first introduce two tessellations $\mathcal{T}_{\boldsymbol{x}}$ and $\mathcal{T}_{\boldsymbol{v}}$ of $\Omega_{\boldsymbol{x}}$ and $\Omega_{\boldsymbol{v}}$, respectively, composed of rectangular elements $K_{\boldsymbol{x}}$ and $K_{\boldsymbol{v}}$; the element boundaries will be denoted by $\partial K_{\boldsymbol{x}}$ and $\partial K_{\boldsymbol{v}}$. The tessellation $\mathcal{T}$ for the phase space domain Ω is then composed of all the elements $K = K_{\boldsymbol{x}} \times K_{\boldsymbol{v}}$ resulting from the Cartesian product of two arbitrary elements $K_{\boldsymbol{x}} \in \mathcal{T}_{\boldsymbol{x}}$ and $K_{\boldsymbol{v}} \in \mathcal{T}_{\boldsymbol{v}}$, with boundary

$$\partial K = \partial_{\boldsymbol{x}} K \cup \partial_{\boldsymbol{v}} K = (\partial K_{\boldsymbol{x}} \times K_{\boldsymbol{v}}) \cup (K_{\boldsymbol{x}} \times \partial K_{\boldsymbol{v}}). \tag{3.1}$$

The outward pointing normal vector on the boundary of any element is denoted by $\boldsymbol{n}$. The edges of the tessellations are indicated with $e_{\boldsymbol{x}}$, $e_{\boldsymbol{v}}$ and e for $\mathcal{T}_{\boldsymbol{x}}$, $\mathcal{T}_{\boldsymbol{v}}$ and $\mathcal{T}$, respectively, and their collections are $\mathcal{E}_{\boldsymbol{x}}$, $\mathcal{E}_{\boldsymbol{v}}$ and $\mathcal{E}$. For edges $e \in \mathcal{E}$ we also distinguish between $e_{\boldsymbol{v},\boldsymbol{x}} = e_{\boldsymbol{x}} \times K_{\boldsymbol{v}}$ and $e_{\boldsymbol{x},\boldsymbol{v}} = K_{\boldsymbol{x}} \times e_{\boldsymbol{v}}$. We assume that the tessellations are conforming and, although this might not be strictly required, that the edges are parallel to the coordinate axes, so that, for instance, $x^\alpha = const$ on $e_{\boldsymbol{x}} \in \mathcal{E}_{\boldsymbol{x}}$ for some $1 \le \alpha \le d$. In view of definition (4.7), we also assume that the hyperplane $\boldsymbol{v} = 0$ is entirely covered by sides $e_{\boldsymbol{x},\boldsymbol{v}} \in \mathcal{E}$, i.e. no element K lies "across" this hyperplane, so that the sign of each velocity component does not change within K. It is also convenient to introduce a notation for a stripe in phase space, local in space and global in velocity; we thus let

$$S_{K_{\boldsymbol{x}}} = K_{\boldsymbol{x}} \times \Omega_{\boldsymbol{v}}, \qquad \partial S_{K_{\boldsymbol{x}}} = \partial K_{\boldsymbol{x}} \times \Omega_{\boldsymbol{v}}.$$

The standard notation for averages and jumps is then introduced as follows (see for instance [2]): considering a generic function χ_h, piecewise continuous on $\mathcal{T}_{\boldsymbol{x}}$, for a given element $K_{\boldsymbol{x}}$, edge $e_{\boldsymbol{x}} \subset \partial K_{\boldsymbol{x}}$ and point $\boldsymbol{x} \in e_{\boldsymbol{x}}$, we define

$$\chi_h\big(\boldsymbol{x}^{int(K_{\boldsymbol{x}})}\big) = \lim_{\substack{\boldsymbol{y} \to \boldsymbol{x} \\ \boldsymbol{y} \in K'_{\boldsymbol{x}}}} \chi_h(\boldsymbol{y}).$$

This definition can be extended to a vector valued function $\boldsymbol{r}_h$ by applying it componentwise. Then, for $e_{\boldsymbol{x}} = \partial K_{\boldsymbol{x}} \cap \partial K'_{\boldsymbol{x}}$, $\boldsymbol{x} \in e_{\boldsymbol{x}}$, let

$$\begin{aligned}
\{\chi_h\}(\boldsymbol{x}) &= \frac{1}{2}\chi_h\big(\boldsymbol{x}^{int(K_{\boldsymbol{x}})}\big) + \frac{1}{2}\chi_h\big(\boldsymbol{x}^{int(K'_{\boldsymbol{x}})}\big) \\
[\![\chi_h]\!](\boldsymbol{x}) &= \boldsymbol{n}\chi_h\big(\boldsymbol{x}^{int(K_{\boldsymbol{x}})}\big) + \boldsymbol{n}'\chi_h\big(\boldsymbol{x}^{int(K'_{\boldsymbol{x}})}\big) \\
\{\boldsymbol{r}_h\}(\boldsymbol{x}) &= \frac{1}{2}\boldsymbol{r}_h\big(\boldsymbol{x}^{int(K_{\boldsymbol{x}})}\big) + \frac{1}{2}\boldsymbol{r}_h\big(\boldsymbol{x}^{int(K'_{\boldsymbol{x}})}\big) \\
[\![\boldsymbol{r}_h]\!](\boldsymbol{x}) &= \boldsymbol{n} \cdot \boldsymbol{r}_h\big(\boldsymbol{x}^{int(K_{\boldsymbol{x}})}\big) + \boldsymbol{n}' \cdot \boldsymbol{r}_h\big(\boldsymbol{x}^{int(K'_{\boldsymbol{x}})}\big),
\end{aligned}$$

where $\boldsymbol{n}$ and $\boldsymbol{n}'$ are the outward pointing normals at $\boldsymbol{x}$ on $\partial K_{\boldsymbol{x}}$ and $\partial K'_{\boldsymbol{x}}$, respectively, with $\boldsymbol{n}' = -\boldsymbol{n}$. Considering the product of a scalar and a vector valued function, the following identities are easily verified:

$$\{\chi_h \boldsymbol{r}_h\} \cdot \boldsymbol{n} = \{\chi_h\}\{\boldsymbol{r}_h\} \cdot \boldsymbol{n} + \frac{1}{4}[\![\boldsymbol{r}_h]\!][\![\chi_h]\!] \cdot \boldsymbol{n} \tag{3.2}$$

and

$$[\![\chi_h \boldsymbol{r}_h]\!] = \{\chi_h\}[\![\boldsymbol{r}_h]\!] + [\![\chi_h]\!] \cdot \{\boldsymbol{r}_h\}, \tag{3.3}$$

which are often used in the analysis of DG formulations. These definitions can be extended to elements in $\mathcal{T}_{\boldsymbol{v}}$ and $\mathcal{T}$.

Having defined the tessellation of the computational domain, the next element required to define the spectral element DG formulation is a basis for the local polynomial space where the discrete solution is sought, as well as some quadrature formulae. We denote by $\{\hat{\xi}_i^{1\mathrm{D},k}\}_{i=0}^k$ the Legendre–Gauss–Lobatto (LGL) points on the interval $[-1\,,1]$, defined as the roots of the polynomial $(1-\xi^2)\hat{\psi}'_k(\xi)$, where $\hat{\psi}_k$ is the k-th Legendre polynomial on $[-1\,,1]$ (see [14,15]), and let $\{\hat{\varphi}_i^{1\mathrm{D},k}\}_{i=0}^k$ be the Lagrangian basis of the polynomial space $\mathbb{P}_k([-1\,,1])$ associated with such points. Next, we introduce the $(k+1)^d$ multivariate polynomials

$$\hat{\varphi}_{\boldsymbol{i}}(\boldsymbol{\xi}) = \prod_{\alpha=1}^{d} \hat{\varphi}_{i^\alpha}^{1\mathrm{D},k}\big(\xi^\alpha\big),$$

where $\boldsymbol{\xi} = (\xi^1, \ldots, \xi^d)$, with $\xi^\alpha \in [-1\,,1]$, and $\boldsymbol{i}$ is the multi-index $i^1, \ldots, i^d$, with $0 \le i^\alpha \le k$, as well as the $(k+2)(k+1)^{d-1}$ multivariate polynomials

$$\hat{\varphi}_{\boldsymbol{i}_+^\alpha}(\boldsymbol{\xi}) = \hat{\varphi}_{i^\alpha}^{1\mathrm{D},k+1}\big(\xi^\alpha\big) \prod_{\substack{\beta=1 \\ \beta \neq \alpha}}^{d} \hat{\varphi}_{i^\beta}^{1\mathrm{D},k}\big(\xi^\beta\big),$$

where $\boldsymbol{i}_+^\alpha$ is the multi-index $i^1,\dots,i^d$, with $0 \le i^\beta \le k$ if $\beta \neq \alpha$ and $0 \le i^\alpha \le k+1$. The polynomials $\hat{\varphi}_{\boldsymbol{i}}$ and $\hat{\varphi}_{\boldsymbol{i}_+^\alpha}$ clearly represent the Lagrangian bases associated with the nodes $\boldsymbol{\xi}_{\boldsymbol{i}} = (\hat{\xi}_{i^1}^{1\mathrm{D},k},\dots,\hat{\xi}_{i^d}^{1\mathrm{D},k})$ and $\boldsymbol{\xi}_{\boldsymbol{i}_+^\alpha} = (\hat{\xi}_{i^1}^{1\mathrm{D},k},\dots,\hat{\xi}_{i^\alpha}^{1\mathrm{D},k+1},\dots,\hat{\xi}_{i^d}^{1\mathrm{D},k})$, respectively. Now, for a generic element $K_{\boldsymbol{x}}$, let $\mathcal{F}_{K_{\boldsymbol{x}}}$ be a transformation mapping $[-1,1]^d$ onto $\overline{K}_{\boldsymbol{x}}$, define

$$\varphi_{K_{\boldsymbol{x}},\boldsymbol{i}}(\boldsymbol{x}) = \begin{cases} (\hat{\varphi}_{\boldsymbol{i}} \circ \mathcal{F}_{K_{\boldsymbol{x}}}^{-1})(\boldsymbol{x}), & \boldsymbol{x} \in \overline{K}_{\boldsymbol{x}} \\ 0, & \boldsymbol{x} \notin \overline{K}_{\boldsymbol{x}} \end{cases}$$

and denote by $\mathbb{Q}_k(K_{\boldsymbol{x}})$ the space generated by the functions $\varphi_{K_{\boldsymbol{x}},\boldsymbol{i}}(\boldsymbol{x})$ for all the possible values of the multi-index. Similarly, define

$$\varphi_{K_{\boldsymbol{x}},\boldsymbol{i}_+^\alpha}(\boldsymbol{x}) = \begin{cases} (\hat{\varphi}_{\boldsymbol{i}_+^\alpha} \circ \mathcal{F}_{K_{\boldsymbol{x}}}^{-1})(\boldsymbol{x}), & \boldsymbol{x} \in \overline{K}_{\boldsymbol{x}} \\ 0, & \boldsymbol{x} \notin \overline{K}_{\boldsymbol{x}} \end{cases}$$

and denote by $\mathbb{Q}_{k_+^\alpha}(K_{\boldsymbol{x}})$ the space generated by these functions for all the possible values of the multi-index. The following finite element spaces can now be introduced:

$$V_h^k(\mathcal{T}_{\boldsymbol{x}}) = \left\{\chi_h \in L^2(\Omega_{\boldsymbol{x}}) : \chi_h|_{K_{\boldsymbol{x}}} \in \mathbb{Q}_k(K_{\boldsymbol{x}})\right\},$$
$$\boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}}) = \left\{\boldsymbol{r}_h \in \left(L^2(\Omega_{\boldsymbol{x}})\right)^d : r_h^\alpha|_{K_{\boldsymbol{x}}} \in \mathbb{Q}_k(K_{\boldsymbol{x}}),\ 1 \le \alpha \le d\right\},$$

and

$$\boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}}) = \left\{\boldsymbol{r}_h \in \left(L^2(\Omega_{\boldsymbol{x}})\right)^d : r_h^\alpha|_{K_{\boldsymbol{x}}} \in \mathbb{Q}_{k_+^\alpha}(K_{\boldsymbol{x}}),\ 1 \le \alpha \le d\right\},$$

where the first two spaces are the standard piecewise polynomial spaces, while the third one is the Raviart–Thomas space introduced in [49]. The same construction of course can be repeated for $\mathcal{T}_{\boldsymbol{v}}$ and $\mathcal{T}$. In this paper, we always assume that $k \ge 2$; this is not a real limitation, since spectral element formulations typically use high-order polynomials anyway.

The LGL points define a quadrature rule on $[-1,1]$ with weights

$$\hat{w}_i^{1\mathrm{D},k} = \frac{2}{k(k+1)\hat{\psi}_k(\hat{\xi}_i^{1\mathrm{D},k})}.$$

A tensor product, multidimensional quadrature rule can then be obtained introducing the weights

$$\hat{w}_{\boldsymbol{i}} = \prod_{\alpha=1}^{d} \hat{w}_{i^\alpha}^{1\mathrm{D},k}$$

and approximating integrals as

$$\int_{K_{\boldsymbol{x}}} \chi_h(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \approx \int_{K_{\boldsymbol{x}},w} \chi_h(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \frac{|K_{\boldsymbol{x}}|}{2^d}\sum_{\boldsymbol{i}} \hat{w}_{\boldsymbol{i}}\chi_{\boldsymbol{i}},$$

where the summation is extended over all the possible values of the multi-index, $|K_{\boldsymbol{x}}|$ is the volume of the element and $\chi_{\boldsymbol{i}} = \chi_h(\mathcal{F}_{K_{\boldsymbol{x}}}(\boldsymbol{\xi}_{\boldsymbol{i}}))$. Notice that, when $\chi_h \in V_h^k(\mathcal{T}_{\boldsymbol{x}})$, the values $\chi_{\boldsymbol{i}}$ are the coefficients with respect to the Lagrangian basis $\varphi_{K_{\boldsymbol{x}},\boldsymbol{i}}$. A similar construction allows to approximate integrals on $e_{\boldsymbol{x}} \subset \partial K_{\boldsymbol{x}}$. Clearly, this quadrature formula can be applied also for the integration of components of vector valued functions belonging to $\boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$, while for functions in $\boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$ it is convenient to introduce the quadrature formula

$$\int_{K_{\boldsymbol{x}}} r_h^\alpha(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \approx \int_{K_{\boldsymbol{x}},w_+^\alpha} r_h^\alpha(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \frac{|K_{\boldsymbol{x}}|}{2^d}\sum_{\boldsymbol{i}_+^\alpha} \hat{w}_{\boldsymbol{i}_+^\alpha} r_{\boldsymbol{i}_+^\alpha}^\alpha,$$

with

$$\hat{w}_{\boldsymbol{i}_+^\alpha} = \hat{w}_{i^\alpha}^{1\mathrm{D},k+1} \prod_{\substack{\beta=1 \\ \beta\neq\alpha}}^{d} \hat{w}_{i^\beta}^{1\mathrm{D},k}$$

and $r_{\boldsymbol{i}_+^\alpha}^\alpha = r_h^\alpha(\mathcal{F}_{K_{\boldsymbol{x}}}(\boldsymbol{\xi}_{\boldsymbol{i}_+^\alpha}))$. Notice that in both cases $\boldsymbol{r}_h \in \boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$ and $\boldsymbol{r}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$ the normal flux $\boldsymbol{n}\cdot\boldsymbol{r}_h|_{e_{\boldsymbol{x}}}$ is a polynomial of degree k.

3.2. DG discretization of the Vlasov equation

The DG discretization of the Vlasov equation is obtained multiplying $(2.1)_1$ by a test function $u_h(\boldsymbol{x}, \boldsymbol{v})$, integrating over $K \in \mathcal{T}$, replacing the exact solution with the corresponding numeric approximation and integrating by parts. Taking $u_h \in V_h^k(\mathcal{T})$ and $f_h(t,\cdot,\cdot) \in V_h^k(\mathcal{T})$ $\forall t \in [0,T]$ we obtain

$$\begin{aligned}
&\iint_K \partial_t f_h u_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} - \iint_K \boldsymbol{v} f_h \cdot \nabla_{\boldsymbol{x}} u_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int_{\partial_{\boldsymbol{x}} K} \boldsymbol{n} \cdot \widehat{\boldsymbol{v} f} u_h \,\mathrm{d}\sigma \,\mathrm{d}\boldsymbol{v} \\
&\quad + \iint_K \tilde{\boldsymbol{E}}_h f_h \cdot \nabla_{\boldsymbol{v}} u_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} - \int_{\partial_{\boldsymbol{v}} K} \boldsymbol{n} \cdot \widehat{\boldsymbol{E} f} u_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{\tau} = 0,
\end{aligned} \tag{3.4}$$

where the hat symbol denotes the so-called numerical fluxes, to be defined shortly, and $\tilde{\boldsymbol{E}}_h$ is an approximation of the electric field which precise form will be discussed in Section 4 and which will play an important role in the construction of an energy conserving discretization. According to (3.1), $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ denote the integration variables on $\partial K_{\boldsymbol{x}}$ and $\partial K_{\boldsymbol{v}}$, respectively, where $K = K_{\boldsymbol{x}} \times K_{\boldsymbol{v}}$.

Concerning the numerical fluxes, we consider here two alternatives: centered fluxes and upwind fluxes. In the space coordinate, these are defined by

$$\widehat{\boldsymbol{v} f}^{centered} = \{\boldsymbol{v} f_h\}, \qquad \widehat{\boldsymbol{v} f}^{upwind} = \{\boldsymbol{v} f_h\} + \frac{1}{2}|\boldsymbol{v}\cdot\boldsymbol{n}|[\![f_h]\!], \tag{3.5}$$

while in the velocity coordinate these are

$$\widehat{\boldsymbol{E} f}^{centered} = \{\boldsymbol{E} f_h\}, \qquad \widehat{\boldsymbol{E} f}^{upwind} = \{\boldsymbol{E} f_h\} - \frac{1}{2}|\boldsymbol{E}\cdot\boldsymbol{n}|[\![f_h]\!]. \tag{3.6}$$

Whatever definition is used, it is important to realize that these numerical fluxes are single valued quantities on $\partial_{\boldsymbol{x}} K$ and $\partial_{\boldsymbol{v}} K$, respectively. Notice that centered and upwind fluxes can be regarded as special cases of a weighted upwind flux

$$\widehat{\boldsymbol{v} f}^{\omega} = \{\boldsymbol{v} f_h\} + \frac{\omega}{2}|\boldsymbol{v}\cdot\boldsymbol{n}|[\![f_h]\!], \qquad \widehat{\boldsymbol{E} f}^{\omega} = \{\boldsymbol{E} f_h\} - \frac{\omega}{2}|\boldsymbol{E}\cdot\boldsymbol{n}|[\![f_h]\!] \tag{3.7}$$

for an upwinding parameter $\omega \in [0,1]$.

Notice that, taking $u_h = 1|_K$, the characteristic function of the element K, we obtain the local mass balance

$$\frac{d}{dt}\iint_K f_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = -\int_{\partial_{\boldsymbol{x}} K} \boldsymbol{n}\cdot\widehat{\boldsymbol{v} f}\,\mathrm{d}\sigma\,\mathrm{d}\boldsymbol{v} + \int_{\partial_{\boldsymbol{v}} K} \boldsymbol{n}\cdot\widehat{\boldsymbol{E} f}\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{\tau}, \tag{3.8}$$

and, since the numerical fluxes are single valued on ∂K, the change in the number of particles in K is exactly balanced by the changes in the neighbouring elements, so that the total number of particles is constant. In other terms, the DG method is by its nature a *flux form* method.

3.3. DG discretization of the Poisson equation

To obtain the DG discretization of the Poisson equation, we first rewrite $(2.1)_2$ in mixed form

$$\begin{aligned}
&\nabla_{\boldsymbol{x}}\cdot\boldsymbol{E} = 1-\rho \\
&\boldsymbol{E} - \nabla_{\boldsymbol{x}}\Phi = 0.
\end{aligned} \tag{3.9}$$

Then, multiplying $(3.9)_1$ and $(3.9)_2$ by $\boldsymbol{z}_h(\boldsymbol{x})$ and $p_h(\boldsymbol{x})$, respectively, integrating over $K_{\boldsymbol{x}} \in \mathcal{T}_{\boldsymbol{x}}$, replacing the exact solution with the corresponding numeric approximation and integrating by parts we obtain

$$\begin{aligned}
&\int_{K_{\boldsymbol{x}}} \boldsymbol{E}_h\cdot\boldsymbol{z}_h\,\mathrm{d}\boldsymbol{x} - \int_{K_{\boldsymbol{x}}} \nabla_{\boldsymbol{x}}\Phi_h\cdot\boldsymbol{z}_h\,\mathrm{d}\boldsymbol{x} - \int_{\partial K_{\boldsymbol{x}}} (\widehat{\Phi}-\Phi_h)\boldsymbol{n}\cdot\boldsymbol{z}_h\,\mathrm{d}\sigma = 0 \\
&\int_{K_{\boldsymbol{x}}} \boldsymbol{E}_h\cdot\nabla_{\boldsymbol{x}} p_h\,\mathrm{d}\boldsymbol{x} - \int_{\partial K_{\boldsymbol{x}}} \boldsymbol{n}\cdot\widehat{\boldsymbol{E}}\,p_h\,\mathrm{d}\sigma = -\int_{K_{\boldsymbol{x}}} (1-\rho_h)\,p_h\,\mathrm{d}\boldsymbol{x},
\end{aligned} \tag{3.10}$$

with

$$\rho_h(t,\boldsymbol{x}) = \int_{\Omega_{\boldsymbol{v}}} f_h(t,\boldsymbol{x},\boldsymbol{v})\,\mathrm{d}\boldsymbol{v}. \tag{3.11}$$

The choice of the finite element spaces for the primal variable Φ_h and the dual variable $\boldsymbol{E}_h$ plays an important role in the definition of the characteristics of the resulting formulation. We consider here $\Phi_h, p_h \in V_h^k(\mathcal{T}_{\boldsymbol{x}}) \cap L_0^2(\Omega_{\boldsymbol{x}})$, where $L_0^2(\Omega_{\boldsymbol{x}})$ is the space of square integrable functions on $\Omega_{\boldsymbol{x}}$ with vanishing integral, while for the dual variable we consider two possibilities: $\boldsymbol{E}_h, \boldsymbol{z}_h \in \boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$ and $\boldsymbol{E}_h, \boldsymbol{z}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$.

The numerical fluxes are defined as

$$\widehat{\boldsymbol{E}} = \{\boldsymbol{E}_h\} - c_{11}[\![\Phi_h]\!], \qquad \widehat{\Phi} = \{\Phi_h\}, \tag{3.12}$$

where $c_{11} \geq 0$. Notice that, compared with the general Local Discontinuous Galerkin (LDG) method analyzed in [8] and considered in [3–5] we restrict ourself to the case $\boldsymbol{c}_{12} = 0$, $c_{22} = 0$.

Notice also that, in (3.10), the differential operator $\nabla_{\boldsymbol{x}}$ acts on the unknown Φ_h in the first equation and on the test function p_h in the second equation, which corresponds to what is sometimes referred to as "strong form" and "weak form" respectively, see for instance [30]. The two forms are of course equivalent, but differences can arise substituting the exact integration appearing in (3.10) with numerical integration. The chosen form is motivated by the fact that it will ensure energy conservation even in the case of approximate integration.

The choice of the finite element space for the dual variable, together with the condition $c_{11} \geq 0$, deserves some comments. The case $\boldsymbol{E}_h \in \boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$ is the standard LDG method analyzed in [2,8], and requires $c_{11} > 0$. The case $\boldsymbol{E}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$ is introduced here because this space will allow us to define an energy conserving formulation in Section 4. In this second case, we assume $c_{11} \geq 0$, which requires a minor modification of the stability proof of the scheme, namely Proposition 2.1 of [8]. Thus, we state the following

Lemma 3.1. *Let $\boldsymbol{E}_h, \boldsymbol{z}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$ and $\Phi_h, p_h \in V_h^k(\mathcal{T}_{\boldsymbol{x}}) \cap L_0^2(\Omega_{\boldsymbol{x}})$ and let $c_{11} \geq 0$. Then (3.10) admits a unique solution.*

Proof. Substituting the numerical fluxes (3.12) in (3.10), summing for all $K_{\boldsymbol{x}} \in \mathcal{T}_{\boldsymbol{x}}$ and changing sign in $(3.10)_2$ we have

$$\int_{\Omega_{\boldsymbol{x}}} \boldsymbol{E}_h \cdot \boldsymbol{z}_h \,\mathrm{d}\boldsymbol{x} - \int_{\Omega_{\boldsymbol{x}}} \nabla_{\boldsymbol{x}} \Phi_h \cdot \boldsymbol{z}_h \,\mathrm{d}\boldsymbol{x} + \sum_{e_{\boldsymbol{x}} \in \mathcal{E}_{\boldsymbol{x}}} \int_{e_{\boldsymbol{x}}} [\![\Phi_h]\!] \cdot \{\boldsymbol{z}_h\} \mathrm{d}\boldsymbol{\sigma} = 0$$

$$-\int_{\Omega_{\boldsymbol{x}}} \boldsymbol{E}_h \cdot \nabla_{\boldsymbol{x}} p_h \,\mathrm{d}\boldsymbol{x} + \sum_{e_{\boldsymbol{x}} \in \mathcal{E}_{\boldsymbol{x}}} \int_{e_{\boldsymbol{x}}} \left(\{\boldsymbol{E}_h\} - c_{11}[\![\Phi_h]\!]\right) \cdot [\![p_h]\!] \mathrm{d}\boldsymbol{\sigma} = \int_{\Omega_{\boldsymbol{x}}} (1 - \rho_h)\, p_h \,\mathrm{d}\boldsymbol{x}.$$

Due to the linearity and finite dimensionality of the problem, we need to show that $\rho_h = 1$ implies $\boldsymbol{E}_h = 0$, $\Phi_h = 0$. Indeed, taking $\boldsymbol{z}_h = \boldsymbol{E}_h$ and $p_h = \Phi_h$ and subtracting the resulting equations yields

$$\int_{\Omega_{\boldsymbol{x}}} E_h^2 \,\mathrm{d}\boldsymbol{x} + \sum_{e_{\boldsymbol{x}} \in \mathcal{E}_{\boldsymbol{x}}} \int_{e_{\boldsymbol{x}}} c_{11}[\![\Phi_h]\!]^2 \mathrm{d}\boldsymbol{\sigma} = 0,$$

from which we can deduce $\boldsymbol{E}_h = 0$ and

$$-\int_{\Omega_{\boldsymbol{x}}} \nabla_{\boldsymbol{x}} \Phi_h \cdot \boldsymbol{z}_h \,\mathrm{d}\boldsymbol{x} + \sum_{e_{\boldsymbol{x}} \in \mathcal{E}_{\boldsymbol{x}}} \int_{e_{\boldsymbol{x}}} [\![\Phi_h]\!] \cdot \{\boldsymbol{z}_h\} \mathrm{d}\boldsymbol{\sigma} = 0.$$

As discussed in Lemma 6 of [49], we can choose $\boldsymbol{z}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}}) \cap H(\mathrm{div}, \Omega_{\boldsymbol{x}})$ specifying the following degrees of freedom:

- the values of $\boldsymbol{z}_h \cdot \boldsymbol{n} = \{\boldsymbol{z}_h\} \cdot \boldsymbol{n}$ at $k+1$ distinct points on each $e_{\boldsymbol{x}}$
- the moments

$$\int_{K_{\boldsymbol{x}}} z_h^\alpha \pi^\alpha \,\mathrm{d}\boldsymbol{x}$$

on each element $K_{\boldsymbol{x}}$, with

$$\pi^\alpha(\boldsymbol{x}) = \prod_{\beta=1}^{d} \left(x^\beta\right)^{k_{\alpha,\beta}}, \qquad \begin{cases} 0 \leq k_{\alpha,\beta} \leq k-1, & \alpha = \beta \\ 0 \leq k_{\alpha,\beta} \leq k, & \alpha \neq \beta. \end{cases}$$

This implies that, for any $\Phi_h \in V_h^k(\mathcal{T}_{\boldsymbol{x}})$, we can choose $\boldsymbol{z}_h$ so that

$$\int_{\Omega_{\boldsymbol{x}}} \nabla_{\boldsymbol{x}} \Phi_h \cdot \boldsymbol{z}_h \,\mathrm{d}\boldsymbol{x} = \int_{\Omega_{\boldsymbol{x}}} (\nabla_{\boldsymbol{x}} \Phi_h)^2 \,\mathrm{d}\boldsymbol{x}, \qquad \int_{e_{\boldsymbol{x}}} [\![\Phi_h]\!] \cdot \{\boldsymbol{z}_h\} \mathrm{d}\boldsymbol{\sigma} = \int_{e_{\boldsymbol{x}}} [\![\Phi_h]\!]^2 \mathrm{d}\boldsymbol{\sigma},$$

which shows $\Phi_h = const$. Since $\Phi_h \in L_0^2(\Omega_{\boldsymbol{x}})$, we conclude $\Phi_h = 0$. □

Concerning the accuracy of the method obtained with $\boldsymbol{E}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$, we observe that, for $c_{11} > 0$, the error estimates provided in [8], Theorem 2.2, namely convergence of order k and $k+1$ for $\|\boldsymbol{E} - \boldsymbol{E}_h\|_{L^2}$ and $\|\Phi - \Phi_h\|_{L^2}$, respectively, remain valid, since they are based on the *inclusion property* (2.14) of such reference, while for the case $c_{11} = 0$ we provide in Section 7.1 some numerical results showing comparable accuracy with the $c_{11} > 0$ case.

An interesting observation can be made comparing the method given by (3.10) with $\boldsymbol{E}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$, $c_{11} = 0$, and the classical Raviart–Thomas formulation [49], especially considering the hybridizable formulation analyzed in [1]. In fact, the two methods employ the same finite element spaces, but different definitions of the numerical fluxes. In the hybridizable Raviart–Thomas method, the numerical flux for the primary variable is treated as an additional unknown, setting $\hat{\Phi} = \lambda$, which is then determined by the condition that $\boldsymbol{E}_h \cdot \boldsymbol{n}$ shall be single valued on each edge (i.e. $\boldsymbol{E}_h \in H(\mathrm{div}, \Omega_{\boldsymbol{x}})$), while in our case $\hat{\Phi}$ is a combination of the two local values $\Phi_h|_{\partial K_{\boldsymbol{x}}}$ and $\Phi_h|_{\partial K'_{\boldsymbol{x}}}$, in the spirit of the DG method, and in general we have $\boldsymbol{E}_h \notin H(\mathrm{div}, \Omega_{\boldsymbol{x}})$. We refer to [11] for a detailed analysis of the relations between DG schemes and hybridizable schemes.

4. An energy conserving formulation

In this section, we first analyze to which extent the energy balance (2.9) is reproduced by the discrete method, and then we show how it is possible to obtain an exactly energy conserving spatial discretization by taking advantage of the freedom in the choice of the finite element spaces for $\boldsymbol{E}_h$ and Φ_h and in the definition of $\tilde{\boldsymbol{E}}_h$.

The total energy appearing in (2.9) is the sum of two contributions: the kinetic energy and the electrostatic energy, and the conversion between these two energies takes place because of the work done by the electrostatic force. To obtain a local in space, kinetic energy balance for the discrete problem, let us take $u_h = v^2/2$ in (3.4) (this is always possible since we assume that $u_h \in V_h^k(\mathcal{T})$ with $k \geq 2$) and add the resulting equations for $K \in S_{K_{\boldsymbol{x}}}$, for a generic element $K_{\boldsymbol{x}}$. Since the chosen test function is independent from $\boldsymbol{x}$, we have $\nabla_{\boldsymbol{x}} u_h = 0$, and since it is continuous, the integrals over $\partial_{\boldsymbol{v}} K$ cancel. The resulting equation is

$$\frac{d}{dt} \iint_{S_{K_{\boldsymbol{x}}}} f_h \frac{v^2}{2} \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = - \int_{\partial S_{K_{\boldsymbol{x}}}} \boldsymbol{n} \cdot \widehat{\boldsymbol{v} f} \frac{v^2}{2} \,\mathrm{d}\sigma \,\mathrm{d}\boldsymbol{v} - \iint_{S_{K_{\boldsymbol{x}}}} f_h \tilde{\boldsymbol{E}}_h \cdot \boldsymbol{v} \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v}, \tag{4.1}$$

representing the fact that the kinetic energy contained in $K_{\boldsymbol{x}}$ varies because of the fluxes of kinetic energy on $\partial K_{\boldsymbol{x}}$ and because of the work done by the electrostatic force. To obtain the corresponding electrostatic energy balance, one needs to consider $u_h = \Phi_h$ in (3.4), which is indeed possible given our choice of finite element spaces. Then using the fact that Φ_h does not depend on $\boldsymbol{v}$, and thus it is also continuous on $\partial_{\boldsymbol{v}} K$, we obtain

$$\iint_{S_{K_{\boldsymbol{x}}}} \partial_t f_h \Phi_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = \iint_{S_{K_{\boldsymbol{x}}}} f_h \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \Phi_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} - \int_{\partial S_{K_{\boldsymbol{x}}}} \boldsymbol{n} \cdot \widehat{\boldsymbol{v} f} \Phi_h \,\mathrm{d}\sigma \,\mathrm{d}\boldsymbol{v}. \tag{4.2}$$

The first term can be expressed in terms of the time derivative of the electrostatic energy considering the Poisson equation. In particular, taking the time derivative of $(3.10)_2$, setting $p_h = \Phi_h$ and using (3.11), we have

$$\iint_{S_{K_{\boldsymbol{x}}}} \partial_t f_h \Phi_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = \int_{K_{\boldsymbol{x}}} \partial_t \rho_h \Phi_h \,\mathrm{d}\boldsymbol{x} = \int_{K_{\boldsymbol{x}}} \partial_t \boldsymbol{E}_h \cdot \nabla_{\boldsymbol{x}} \Phi_h \,\mathrm{d}\boldsymbol{x} - \int_{\partial K_{\boldsymbol{x}}} \boldsymbol{n} \cdot \partial_t \widehat{\boldsymbol{E}} \, \Phi_h \,\mathrm{d}\sigma.$$

The first term in the right-hand side can be computed taking $\boldsymbol{z}_h = \partial_t \boldsymbol{E}_h$ in $(3.10)_1$, so that

$$\iint_{S_{K_{\boldsymbol{x}}}} \partial_t f_h \Phi_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = \frac{d}{dt} \int_{K_{\boldsymbol{x}}} \frac{E_h^2}{2} \,\mathrm{d}\boldsymbol{x} - \int_{\partial K_{\boldsymbol{x}}} \boldsymbol{n} \cdot \partial_t \widehat{\boldsymbol{E}} \, \Phi_h \,\mathrm{d}\sigma - \int_{\partial K_{\boldsymbol{x}}} \boldsymbol{n} \cdot \partial_t \boldsymbol{E} (\widehat{\Phi} - \Phi_h) \,\mathrm{d}\sigma.$$

Substituting in (4.2) yields the local electrostatic energy balance

$$\frac{d}{dt} \int_{K_{\boldsymbol{x}}} \frac{E_h^2}{2} \,\mathrm{d}\boldsymbol{x} = \iint_{S_{K_{\boldsymbol{x}}}} f_h \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} \Phi_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int_{\partial K_{\boldsymbol{x}}} \boldsymbol{n} \cdot \left(\partial_t \boldsymbol{E}_h \widehat{\Phi} - \partial_t \boldsymbol{E}_h \Phi_h + \partial_t \widehat{\boldsymbol{E}} \, \Phi_h - \int_{\Omega_{\boldsymbol{v}}} \widehat{\boldsymbol{v} f} \Phi_h \,\mathrm{d}\boldsymbol{v} \right) \mathrm{d}\sigma. \tag{4.3}$$

The local energy balance can thus be obtained adding (4.1) and (4.3), so that

$$\begin{aligned} \frac{d}{dt} \left[\iint_{S_{K_{\boldsymbol{x}}}} f_h \frac{v^2}{2} \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int_{K_{\boldsymbol{x}}} \frac{E_h^2}{2} \,\mathrm{d}\boldsymbol{x} \right] &= \iint_{S_{K_{\boldsymbol{x}}}} f_h \boldsymbol{v} \cdot (\nabla_{\boldsymbol{x}} \Phi_h - \tilde{\boldsymbol{E}}_h) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} - \int_{\partial S_{K_{\boldsymbol{x}}}} \boldsymbol{n} \cdot \widehat{\boldsymbol{v} f} \frac{v^2}{2} \,\mathrm{d}\sigma \,\mathrm{d}\boldsymbol{v} \\ &\quad + \int_{\partial K_{\boldsymbol{x}}} \boldsymbol{n} \cdot \left(\partial_t \boldsymbol{E}_h \widehat{\Phi} - \partial_t \boldsymbol{E}_h \Phi_h + \partial_t \widehat{\boldsymbol{E}} \, \Phi_h - \int_{\Omega_{\boldsymbol{v}}} \widehat{\boldsymbol{v} f} \Phi_h \,\mathrm{d}\boldsymbol{v} \right) \mathrm{d}\sigma. \end{aligned} \tag{4.4}$$

Eq. (4.4) shows that, in general, spurious energy sources and sinks can be introduced in the discrete energy balance for two reasons: the fact that $\tilde{\boldsymbol{E}}_h \neq \nabla_{\boldsymbol{x}} \Phi_h$ in the first term of the right-hand side, so that the contributions associated with the work of the electrostatic force do not balance exactly, and the fact that the boundary fluxes of electrostatic energy, i.e. the third term in the right-hand side, are not single valued on $\partial K_{\boldsymbol{x}}$. Notice that the boundary fluxes of kinetic energy, i.e. the second term in the right-hand side of (4.4), are by construction a single valued quantity on $\partial K_{\boldsymbol{x}}$, so no spurious terms in the energy balance appear because of these fluxes. We now consider various methods ensuring an exact energy balance.

4.1. Centered discretization of the Vlasov equation

The simplest case is the case of a centered flux in the Vlasov equation.

Theorem 4.1. *Consider the discrete problem (3.4), (3.10) and (3.11) with $\boldsymbol{E}_h \in \boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$ or $\boldsymbol{E}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$, assume that*

$$\widehat{\boldsymbol{v} f} = \{\boldsymbol{v} f_h\} \quad \text{on } \partial_{\boldsymbol{x}} K$$

and let

$$\tilde{\boldsymbol{E}}_h = \boldsymbol{E}_h. \tag{4.5}$$

Then the following total energy is conserved:

$$\mathcal{E} = \iint_{\Omega} f_h \frac{v^2}{2} \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int_{\Omega_{\boldsymbol{x}}} \frac{E_h^2}{2} \,\mathrm{d}\boldsymbol{x} + \sum_{e_{\boldsymbol{x}} \in \mathcal{E}_{\boldsymbol{x}}} \frac{1}{2} \int_{e_{\boldsymbol{x}}} c_{11} [\![\Phi_h]\!]^2 \mathrm{d}\sigma. \tag{4.6}$$

Proof. Consider the energy balance (4.4). For a fixed value $\boldsymbol{v}$, the function $\boldsymbol{v} f_h(\boldsymbol{x}, \boldsymbol{v})$ belongs to $\boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$ (and thus also to $\boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$), so that we can take $\boldsymbol{z}_h = \boldsymbol{v} f_h$ in $(3.10)_1$ and obtain, using (4.5),

$$\begin{aligned}
\iint_{S_{K_{\boldsymbol{x}}}} f_h \boldsymbol{v} \cdot (\nabla_{\boldsymbol{x}} \Phi_h - \tilde{\boldsymbol{E}}_h) \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} &= \int_{\Omega_{\boldsymbol{v}}} \left(\int_{K_{\boldsymbol{x}}} \boldsymbol{v} f_h \cdot (\nabla_{\boldsymbol{x}} \Phi_h - \boldsymbol{E}_h) \mathrm{d}\boldsymbol{x} \right) \mathrm{d}\boldsymbol{v} \\
&= -\int_{\Omega_{\boldsymbol{v}}} \left(\int_{\partial K_{\boldsymbol{x}}} \boldsymbol{v} f_h \cdot \boldsymbol{n} (\widehat{\Phi} - \Phi_h) \mathrm{d}\sigma \right) \mathrm{d}\boldsymbol{v} \\
&= \int_{\partial K_{\boldsymbol{x}}} \boldsymbol{n} \cdot \left(\int_{\Omega_{\boldsymbol{v}}} \boldsymbol{v} f_h (\Phi_h - \widehat{\Phi}) \mathrm{d}\boldsymbol{v} \right) \mathrm{d}\sigma.
\end{aligned}$$

This suggests the definition of

$$\boldsymbol{q}^{\mathcal{E}} = (\widehat{\Phi} - \Phi_h) \left(\partial_t \boldsymbol{E}_h - \int_{\Omega_{\boldsymbol{v}}} \boldsymbol{v} f_h \,\mathrm{d}\boldsymbol{v} \right) + \Phi_h \left(\partial_t \widehat{\boldsymbol{E}} - \int_{\Omega_{\boldsymbol{v}}} \widehat{\boldsymbol{v} f} \,\mathrm{d}\boldsymbol{v} \right) - \int_{\Omega_{\boldsymbol{v}}} \widehat{\boldsymbol{v} f} \frac{v^2}{2} \,\mathrm{d}\boldsymbol{v}$$

so that (4.4) can be rewritten as

$$\frac{d}{dt} \left[\iint_{S_{K_{\boldsymbol{x}}}} f_h \frac{v^2}{2} \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int_{K_{\boldsymbol{x}}} \frac{E_h^2}{2} \,\mathrm{d}\boldsymbol{x} \right] = \int_{\partial K_{\boldsymbol{x}}} \boldsymbol{n} \cdot \boldsymbol{q}^{\mathcal{E}} \,\mathrm{d}\sigma$$

and summing over all the elements $K_{\boldsymbol{x}} \in \mathcal{T}_{\boldsymbol{x}}$

$$\frac{d}{dt} \left[\iint_{\Omega} f_h \frac{v^2}{2} \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int_{\Omega_{\boldsymbol{x}}} \frac{E_h^2}{2} \,\mathrm{d}\boldsymbol{x} \right] = \sum_{e_{\boldsymbol{x}} \in \mathcal{E}_{\boldsymbol{x}}} \int_{e_{\boldsymbol{x}}} [\![\boldsymbol{q}^{\mathcal{E}}]\!] \,\mathrm{d}\sigma.$$

The thesis follows observing that the jump of the energy flux vanishes, up to a term which can be included in the definition of the total energy. In fact, using (3.3) and the expressions of the numerical fluxes, we have

$$\begin{aligned}
[\![\boldsymbol{q}^{\mathcal{E}}]\!] &= \left(\widehat{\Phi} - \{\Phi_h\}\right) \left([\![\partial_t \boldsymbol{E}_h]\!] - \int_{\Omega_{\boldsymbol{v}}} [\![\boldsymbol{v} f_h]\!] \,\mathrm{d}\boldsymbol{v} \right) + [\![\Phi_h]\!] \cdot \left(\partial_t \widehat{\boldsymbol{E}} - \{\partial_t \boldsymbol{E}_h\} - \int_{\Omega_{\boldsymbol{v}}} \left(\widehat{\boldsymbol{v} f} - \{\boldsymbol{v} f_h\} \right) \mathrm{d}\boldsymbol{v} \right) \\
&= -c_{11} \frac{\partial}{\partial t} \frac{[\![\Phi_h]\!]^2}{2}. \qquad \square
\end{aligned}$$

4.2. Upwind discretization of the Vlasov equation

The definition of $\tilde{\boldsymbol{E}}_h$ is less trivial in the case of an upwind numerical flux in the Vlasov equation. In this case, it is useful to introduce, for a generic element $K = K_{\boldsymbol{x}} \times K_{\boldsymbol{v}}$, the tensor

$$\mathcal{H}_K = \begin{bmatrix} \eta_K^1 & & \\ & \ddots & \\ & & \eta_K^d \end{bmatrix} \tag{4.7}$$

where

$$\eta_K^\alpha = \eta_K^\alpha(\boldsymbol{x}, \boldsymbol{v}) = \begin{cases} \hat{\eta}^+(\hat{\xi}^\alpha(\boldsymbol{x})), & v^\alpha \geq 0 \\ \hat{\eta}^-(\hat{\xi}^\alpha(\boldsymbol{x})), & v^\alpha < 0, \end{cases}$$

with $\hat{\boldsymbol{\xi}}(\boldsymbol{x}) = \mathcal{F}_{K_{\boldsymbol{x}}}^{-1}(\boldsymbol{x})$, and, for $\xi \in [-1\,,\,1]$,

$$\hat{\eta}^+(\xi) = 1 + \xi, \qquad \hat{\eta}^-(\xi) = 1 - \xi.$$

For two elements $K = K_{\boldsymbol{x}} \times K_{\boldsymbol{v}}$, $K' = K'_{\boldsymbol{x}} \times K_{\boldsymbol{v}}$, with $e_{\boldsymbol{x}} = \partial K_{\boldsymbol{x}} \cap \partial K'_{\boldsymbol{x}}$ and $e = e_{\boldsymbol{x}} \times K_{\boldsymbol{v}} = \partial K \cap \partial K'$, definition (4.7) ensures that a) for fixed $\boldsymbol{v}$, $\mathcal{H}_K \boldsymbol{v}$ is a vector in $\boldsymbol{RT}_h^0(\mathcal{T}_{\boldsymbol{x}})$, i.e. the α-th component is an affine function of x^α and the remaining ones are constant; b) since $\boldsymbol{v}$ is a continuous function on $\mathcal{T}$,

$$\big(\{\mathcal{H}_K \boldsymbol{v}\}\big)^\alpha\big|_e = \big\{\eta_K^\alpha\big\} v^\alpha\big|_e = v^\alpha\big|_e \quad \Longrightarrow \quad \{\mathcal{H}_K \boldsymbol{v}\}|_e = \boldsymbol{v}|_e; \tag{4.8}$$

and c) assuming that on $e_{\boldsymbol{x}}$ the normal vector is $\boldsymbol{e}_\alpha$, the unit vector directed along the α-th coordinate axis, one has

$$[\![\mathcal{H}_K \boldsymbol{v}]\!]|_e = 2\big(\eta_K^\alpha - \eta_{K'}^\alpha\big) v^\alpha\big|_e = 2|\boldsymbol{v} \cdot \boldsymbol{n}|\,|_e. \tag{4.9}$$

With these definitions, we can now prove the following theorem, representing the counterpart of Theorem 4.1 in the upwind case.

Theorem 4.2. *Consider the discrete problem (3.4), (3.10) and (3.11) with $\boldsymbol{E}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$, assume*

$$\widehat{\boldsymbol{v} f} = \{\boldsymbol{v} f_h\} + \frac{1}{2}|\boldsymbol{v} \cdot \boldsymbol{n}|[\![f_h]\!] \quad \text{on } \partial_{\boldsymbol{x}} K$$

and let

$$\tilde{\boldsymbol{E}}_h|_K = \mathcal{H}_K \boldsymbol{E}_h|_K + (\mathcal{I} - \mathcal{H}_K)\nabla_{\boldsymbol{x}} \Phi_h|_K, \tag{4.10}$$

where $\mathcal{I}$ is the identity. Notice that this choice results in $\tilde{\boldsymbol{E}}_h = \tilde{\boldsymbol{E}}_h(t, \boldsymbol{x}, \boldsymbol{v})$, due to the $\boldsymbol{v}$ dependence of $\mathcal{H}_K$. Then the following total energy is conserved:

$$\mathcal{E} = \iint\limits_{\Omega} f_h \frac{v^2}{2}\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int\limits_{\Omega_{\boldsymbol{x}}} \frac{E_h^2}{2}\,\mathrm{d}\boldsymbol{x} + \sum_{e_{\boldsymbol{x}} \in \mathcal{E}_{\boldsymbol{x}}} \frac{1}{2} \int\limits_{e_{\boldsymbol{x}}} c_{11} [\![\Phi_h]\!]^2 \mathrm{d}\sigma\,. \tag{4.11}$$

Proof. The proof proceeds as in Theorem 4.1, with some minor modification. Consider the energy balance (4.4) and substitute

$$\nabla_{\boldsymbol{x}} \Phi_h - \tilde{\boldsymbol{E}}_h = \mathcal{H}_K(\nabla_{\boldsymbol{x}} \Phi_h - \boldsymbol{E}_h),$$

according to (4.10). For a fixed value $\boldsymbol{v}$, the function $\mathcal{H}_K(\boldsymbol{x}, \boldsymbol{v})\,\boldsymbol{v} f_h(\boldsymbol{x}, \boldsymbol{v})$ belongs to $\boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$, so that we can take $\boldsymbol{z}_h = \mathcal{H}_K \boldsymbol{v} f_h$ in $(3.10)_1$ and obtain

$$\begin{aligned} \iint\limits_{S_{K_{\boldsymbol{x}}}} f_h \boldsymbol{v} \cdot (\nabla_{\boldsymbol{x}} \Phi_h - \tilde{\boldsymbol{E}}_h)\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} &= \int\limits_{\Omega_{\boldsymbol{v}}} \left(\int\limits_{K_{\boldsymbol{x}}} \mathcal{H}_K \boldsymbol{v} f_h \cdot (\nabla_{\boldsymbol{x}} \Phi_h - \boldsymbol{E}_h)\mathrm{d}\boldsymbol{x} \right) \mathrm{d}\boldsymbol{v} \\ &= -\int\limits_{\Omega_{\boldsymbol{v}}} \left(\int\limits_{\partial K_{\boldsymbol{x}}} \mathcal{H}_K \boldsymbol{v} f_h \cdot \boldsymbol{n}(\widehat{\Phi} - \Phi_h)\mathrm{d}\sigma \right) \mathrm{d}\boldsymbol{v} \\ &= \int\limits_{\partial K_{\boldsymbol{x}}} \boldsymbol{n} \cdot \left(\int\limits_{\Omega_{\boldsymbol{v}}} \mathcal{H}_K \boldsymbol{v} f_h (\Phi_h - \widehat{\Phi})\mathrm{d}\boldsymbol{v} \right) \mathrm{d}\sigma\,. \end{aligned}$$

Now define

$$\boldsymbol{q}^{\mathcal{E}} = (\widehat{\Phi} - \Phi_h)\left(\partial_t \boldsymbol{E}_h - \int_{\Omega_{\boldsymbol{v}}} \mathcal{H}_K \boldsymbol{v} f_h \,\mathrm{d}\boldsymbol{v}\right) + \Phi_h\left(\partial_t \widehat{\boldsymbol{E}} - \int_{\Omega_{\boldsymbol{v}}} \widehat{\boldsymbol{v} f} \,\mathrm{d}\boldsymbol{v}\right) - \int_{\Omega_{\boldsymbol{v}}} \widehat{\boldsymbol{v} f}\,\frac{v^2}{2}\,\mathrm{d}\boldsymbol{v}$$

so that (4.4) can be rewritten as

$$\frac{d}{dt}\left[\iint_{S_{K_{\boldsymbol{x}}}} f_h \frac{v^2}{2}\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int_{K_{\boldsymbol{x}}} \frac{E_h^2}{2}\,\mathrm{d}\boldsymbol{x}\right] = \int_{\partial K_{\boldsymbol{x}}} \boldsymbol{n}\cdot\boldsymbol{q}^{\mathcal{E}}\,\mathrm{d}\boldsymbol{\sigma}.$$

The thesis follows from the fact that $[\![\boldsymbol{q}^{\mathcal{E}}]\!]$ vanishes up to a term which can be included in the definition of the total energy. In fact we have

$$\begin{aligned}[\![\boldsymbol{q}^{\mathcal{E}}]\!] &= \left(\widehat{\Phi} - \{\Phi_h\}\right)\left([\![\partial_t \boldsymbol{E}_h]\!] - \int_{\Omega_{\boldsymbol{v}}} [\![\mathcal{H}_K \boldsymbol{v} f_h]\!]\,\mathrm{d}\boldsymbol{v}\right) + [\![\Phi_h]\!]\cdot\left(\partial_t \widehat{\boldsymbol{E}} - \{\partial_t \boldsymbol{E}_h\} - \int_{\Omega_{\boldsymbol{v}}} \left(\widehat{\boldsymbol{v} f} - \{\mathcal{H}_K \boldsymbol{v} f_h\}\right)\mathrm{d}\boldsymbol{v}\right)\\ &= -c_{11}\frac{\partial}{\partial t}\frac{[\![\Phi_h]\!]^2}{2} - [\![\Phi_h]\!]\cdot\int_{\Omega_{\boldsymbol{v}}} \left(\widehat{\boldsymbol{v} f} - \{\mathcal{H}_K \boldsymbol{v} f_h\}\right)\mathrm{d}\boldsymbol{v}.\end{aligned}$$

To see that the last term vanishes, observe that

$$[\![\Phi_h]\!] = \left([\![\Phi_h]\!]\cdot\boldsymbol{n}\right)\boldsymbol{n},$$

so that

$$[\![\Phi_h]\!]\cdot\int_{\Omega_{\boldsymbol{v}}} \left(\widehat{\boldsymbol{v} f} - \{\mathcal{H}_K \boldsymbol{v} f_h\}\right)\mathrm{d}\boldsymbol{v} = [\![\Phi_h]\!]\cdot\boldsymbol{n}\int_{\Omega_{\boldsymbol{v}}} \boldsymbol{n}\cdot\left(\widehat{\boldsymbol{v} f} - \{\mathcal{H}_K \boldsymbol{v} f_h\}\right)\mathrm{d}\boldsymbol{v}$$

and use (3.2), (4.8) and (4.9) to obtain

$$\boldsymbol{n}\cdot\{\mathcal{H}_K \boldsymbol{v} f_h\} = \boldsymbol{n}\cdot\{\mathcal{H}_K \boldsymbol{v}\}\{f_h\} + \boldsymbol{n}\cdot\frac{1}{4}[\![\mathcal{H}_K \boldsymbol{v}]\!][\![f_h]\!] = \boldsymbol{n}\cdot\{\boldsymbol{v} f_h\} + \boldsymbol{n}\cdot\frac{1}{2}|\boldsymbol{v}\cdot\boldsymbol{n}|[\![f_h]\!]. \qquad \square$$

We conclude this section with various remarks.

First of all, notice that only the upwind in the "space direction", i.e. on sides $e = \partial K_{\boldsymbol{x}} \times K_{\boldsymbol{v}}$, affects the energy balance and the choice of the electric field $\tilde{\boldsymbol{E}}_h$, while the upwinding in the "velocity directions", i.e. on sides $e = K_{\boldsymbol{x}} \times \partial K_{\boldsymbol{v}}$, has no effect on the energy conservation.

As a second point, it is clear from the proof of Theorem 4.2 that the important step in the closure of the local energy balance is

$$\iint_{S_{K_{\boldsymbol{x}}}} f_h \boldsymbol{v}\cdot(\nabla_{\boldsymbol{x}}\Phi_h - \tilde{\boldsymbol{E}}_h)\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} = \int_{\partial K_{\boldsymbol{x}}}\left(\int_{\Omega_{\boldsymbol{v}}} \mathcal{H}_K \boldsymbol{v} f_h \mathrm{d}\boldsymbol{v}\right)\cdot\boldsymbol{n}(\Phi_h - \widehat{\Phi})\mathrm{d}\boldsymbol{\sigma}, \tag{4.12}$$

and that the choice (4.10) is not necessarily the only possibility. Also, notice that only the boundary values $\mathcal{H}_K|_{\partial K_{\boldsymbol{x}}\times K_{\boldsymbol{v}}}$ appear in (4.12), so the precise definition of $\mathcal{H}_K$ on the element interior is not important.

The third observation concerns the apparently unnatural character of (4.10), in that the electric field depends also on the velocity coordinate. Nevertheless, the fact that such a dependence is necessary in order to close the energy balance can be understood observing the structure of (4.12). In fact, assuming that $\tilde{\boldsymbol{E}}_h$ is independent from $\boldsymbol{v}$ leads to an identity where the left-hand-side is proportional to $\int_{\Omega_{\boldsymbol{v}}} f_h \boldsymbol{v}\,\mathrm{d}\boldsymbol{v}$, while the right-hand-side is proportional to $\int_{\Omega_{\boldsymbol{v}}} f_h \mathcal{H}_K \boldsymbol{v}\,\mathrm{d}\boldsymbol{v}$. Since these two integrals are independent from each other, and they may even vanish independently, it is hard to imagine being able to satisfy such a condition. The idea of using an electric field which depends explicitly on the velocity coordinate was first introduced in [3], albeit in a different form from the one considered here.

The fourth and last remark concerns the L^2-norm stability of the resulting scheme. Taking $u_h = f_h$ in (3.4) yields

$$\frac{1}{2}\frac{d}{dt}\iint_K f_h^2\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int_{\partial_{\boldsymbol{x}} K}\boldsymbol{n}\cdot\left(\widehat{\boldsymbol{v} f} - \frac{1}{2}\boldsymbol{v} f_h\right)f_h\,\mathrm{d}\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{v} - \int_{\partial_{\boldsymbol{v}} K}\boldsymbol{n}\cdot\left(\widehat{\boldsymbol{E} f} - \frac{1}{2}\tilde{\boldsymbol{E}}_h f_h\right)f_h\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{\tau} = 0$$

and, after summing over all elements K,

$$\frac{1}{2}\frac{d}{dt}\iint_{\Omega} f_h^2\,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \sum_{e_{\boldsymbol{v},\boldsymbol{x}}, e_{\boldsymbol{x},\boldsymbol{v}}\in\mathcal{E}}\left[\int_{e_{\boldsymbol{v},\boldsymbol{x}}}\left(\widehat{\boldsymbol{v} f}\cdot[\![f_h]\!] - \frac{1}{2}[\![\boldsymbol{v} f_h^2]\!]\right)\mathrm{d}\boldsymbol{\sigma}\,\mathrm{d}\boldsymbol{v} - \int_{e_{\boldsymbol{x},\boldsymbol{v}}}\left(\widehat{\boldsymbol{E} f}\cdot[\![f_h]\!] - \frac{1}{2}[\![\tilde{\boldsymbol{E}}_h f_h^2]\!]\right)\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{\tau}\right] = 0.$$

Concerning the first term, substituting the weighted upwind flux (3.7) we have

$$\widehat{\boldsymbol{v}f}\cdot[\![f_h]\!]-\frac{1}{2}[\![\boldsymbol{v}f_h^2]\!]=\frac{1}{2}\omega|\boldsymbol{v}\cdot\boldsymbol{n}|[\![f_h]\!]^2\geq 0.$$

Concerning the second term, using (3.2) and (3.3) we have

$$-\widehat{\tilde{\boldsymbol{E}}f}\cdot[\![f_h]\!]+\frac{1}{2}[\![\tilde{\boldsymbol{E}}_h f_h^2]\!]=\frac{1}{2}\omega|\tilde{\boldsymbol{E}}_h\cdot\boldsymbol{n}|[\![f_h]\!]^2+\frac{1}{2}[\![\tilde{\boldsymbol{E}}_h]\!]\left(\{f_h\}^2-\frac{1}{4}[\![f_h]\!]^2\right).$$

Now, if $\tilde{\boldsymbol{E}}_h$ is independent from $\boldsymbol{v}$, its jump on $e_{\boldsymbol{x},\boldsymbol{v}}$ vanishes and $\|f_h\|_{L^2(\Omega)}$ turns out to be a nonincreasing quantity. However, when a dependence of $\tilde{\boldsymbol{E}}_h$ is introduced, as in the energy conserving method (4.10), the sign of the last term is unknown, and nothing can be said about the time evolution of $\|f_h\|_{L^2(\Omega)}$. Since, as discussed above, a $\boldsymbol{v}$ dependence for $\tilde{\boldsymbol{E}}_h$ seems to be required in order to ensure energy conservation with upwind fluxes, this result seems to indicate a contradiction between discrete energy conservation and nonincreasing $\|f_h\|_{L^2(\Omega)}$ for the upwind method. Concerning the practical impact of this remark, we anticipate that all our numerical computations employing (4.10) and upwind fluxes show the desired, nonincreasing behaviour of $\|f_h\|_{L^2(\Omega)}$, suggesting that, for these specific cases, the terms proportional to ω dominate the term proportional to $[\![\tilde{\boldsymbol{E}}_h]\!]$.

5. Spectral element DG discretization of the Vlasov–Poisson system

To obtain the spectral element formulation corresponding to (3.4), (3.10) and (3.11), one simply makes the formal substitution

$$\int_{\cdot}\ \mapsto\ \int_{\cdot,w}$$

introducing the quadrature formulae defined in Section 3.1. The resulting scheme can be analyzed within the framework of the *generalized Galerkin* methods, which is discussed in [15], Chapter 5. We provide now the details for the energy conserving formulation addressed in Theorem 4.2, since all the other cases can be easily deduced from this one.

We consider the Poisson problem (3.10) first. Introducing the numerical quadrature formulae yields the problem: find $(\boldsymbol{E}_h,\Phi_h)\in\boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})\times V_h^k(\mathcal{T}_{\boldsymbol{x}})$ such that, for any $(\boldsymbol{z}_h,p_h)\in\boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})\times V_h^k(\mathcal{T}_{\boldsymbol{x}})$ and any $K_{\boldsymbol{x}}\in\mathcal{T}_{\boldsymbol{x}}$,

$$\begin{aligned}
&\int_{K_{\boldsymbol{x}},\boldsymbol{w}_+}\boldsymbol{E}_h\cdot\boldsymbol{z}_h\,\mathrm{d}\boldsymbol{x}-\int_{K_{\boldsymbol{x}},\boldsymbol{w}}\nabla_{\boldsymbol{x}}\Phi_h\cdot\boldsymbol{z}_h\,\mathrm{d}\boldsymbol{x}-\int_{\partial K_{\boldsymbol{x}},w}(\widehat{\Phi}-\Phi_h)\boldsymbol{n}\cdot\boldsymbol{z}_h\,\mathrm{d}\sigma=0\\
&\int_{K_{\boldsymbol{x}},\boldsymbol{w}}\boldsymbol{E}_h\cdot\nabla_{\boldsymbol{x}}p_h\,\mathrm{d}\boldsymbol{x}-\int_{\partial K_{\boldsymbol{x}},w}\boldsymbol{n}\cdot\widehat{\boldsymbol{E}}\,p_h\,\mathrm{d}\sigma=-\int_{K_{\boldsymbol{x}},w}(1-\rho_h)\,p_h\,\mathrm{d}\boldsymbol{x},
\end{aligned}\tag{5.1}$$

where, for two vector valued functions $\boldsymbol{r}_h$, $\boldsymbol{s}_h$, we use the notation

$$\int_{K_{\boldsymbol{x}},\boldsymbol{w}_+}\boldsymbol{r}_h\cdot\boldsymbol{s}_h\,\mathrm{d}\boldsymbol{x}=\sum_{\alpha=1}^{d}\int_{K_{\boldsymbol{x}},w_+^\alpha}r_h^\alpha s_h^\alpha\,\mathrm{d}\boldsymbol{x}$$

and

$$\int_{K_{\boldsymbol{x}},\boldsymbol{w}}\boldsymbol{r}_h\cdot\boldsymbol{s}_h\,\mathrm{d}\boldsymbol{x}=\sum_{\alpha=1}^{d}\int_{K_{\boldsymbol{x}},w}r_h^\alpha s_h^\alpha\,\mathrm{d}\boldsymbol{x}.$$

Notice that the second term in $(5.1)_1$ and the first one in $(5.1)_2$ require the interpolation of $\boldsymbol{z}_h$ and $\boldsymbol{E}_h$ on points which are different from the nodes of the Lagrangian basis of the corresponding space $\boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$, while for all the remaining terms the quadrature nodes coincide with the nodes of the Lagrangian basis. The required interpolation, anyway, turns out to be one-dimensional, so that it does not result in a significant overhead.

We then consider the definition of $\tilde{\boldsymbol{E}}_h$, Eq. (4.10). The following definition allows to recover Theorem 4.2 even in the spectral element case: let $\tilde{\boldsymbol{E}}_h\in\boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$ be defined by

$$\int_{K_{\boldsymbol{x}},\boldsymbol{w}}\tilde{\boldsymbol{E}}_h\cdot\boldsymbol{z}_h\,\mathrm{d}\boldsymbol{x}=\int_{K_{\boldsymbol{x}},\boldsymbol{w}_+}\boldsymbol{E}_h\cdot\mathcal{H}_K\boldsymbol{z}_h\,\mathrm{d}\boldsymbol{x}+\int_{K_{\boldsymbol{x}},\boldsymbol{w}}\nabla_{\boldsymbol{x}}\Phi_h\cdot(\mathcal{I}-\mathcal{H}_K)\boldsymbol{z}_h\,\mathrm{d}\boldsymbol{x}\tag{5.2}$$

for any $\boldsymbol{z}_h\in\boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$ and any $K_{\boldsymbol{x}}\in\mathcal{T}_{\boldsymbol{x}}$.

Finally, concerning the Vlasov equation (3.4), we have: find $f_h\in V_h^k(\mathcal{T}_{\boldsymbol{x}})$ such that, for any $u_h\in V_h^k(\mathcal{T}_{\boldsymbol{x}})$ and any $K\in\mathcal{T}$,

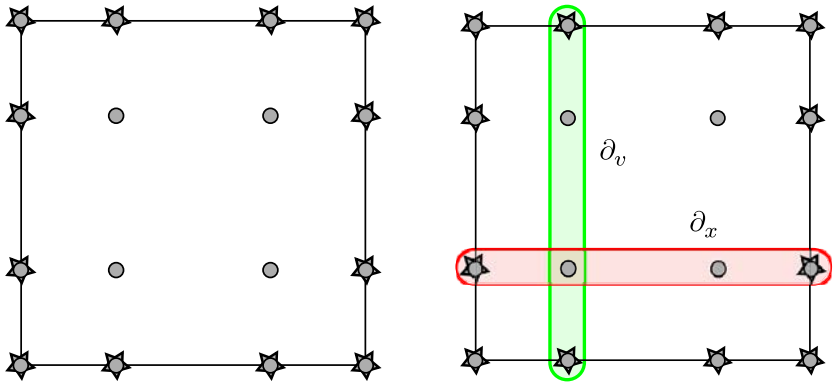


Fig. 5.1. LGL points for a two-dimensional element, $k = 3$ (left) and corresponding stencils for the computation of the x and v partial derivatives (right). (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

$$\iint_{K,w} \partial_t f_h u_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} - \iint_{K,\boldsymbol{w}} \boldsymbol{v} f_h \cdot \nabla_{\boldsymbol{x}} u_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int_{\partial_{\boldsymbol{x}} K,w} \boldsymbol{n} \cdot \widehat{\boldsymbol{v} f} u_h \,\mathrm{d}\sigma \,\mathrm{d}\boldsymbol{v}$$
$$+ \iint_{K,\boldsymbol{w}} \tilde{\boldsymbol{E}}_h f_h \cdot \nabla_{\boldsymbol{v}} u_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} - \int_{\partial_{\boldsymbol{v}} K,w} \boldsymbol{n} \cdot \widehat{\boldsymbol{E} f} u_h \,\mathrm{d}\boldsymbol{x}\mathrm{d}\tau = 0. \tag{5.3}$$

Notice that for all the integrals appearing in (5.3) the quadrature nodes coincide with the nodes of the Lagrangian basis, so that no interpolation is required.

As anticipated, it is easy to verify that all the steps of the proof of Theorem 4.2 can be repeated for the system (5.1), (5.2) and (5.3), so that the following energy is conserved:

$$\mathcal{E}_w = \iint_{\Omega,w} f_h \frac{v^2}{2} \,\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{v} + \int_{\Omega_{\boldsymbol{x}},\boldsymbol{w}_+} \frac{E_h^2}{2} \,\mathrm{d}\boldsymbol{x} + \sum_{e_{\boldsymbol{x}} \in \mathcal{E}_{\boldsymbol{x}}} \frac{1}{2} \int_{e_{\boldsymbol{x}},w} c_{11} [\![\Phi_h]\!]^2 \mathrm{d}\sigma. \tag{5.4}$$

The main advantage of the spectral element formulation compared to the standard method, using exact integration, is its higher efficiency, resulting from the collocation of the quadrature nodes and the degrees of freedom of the Lagrangian basis. This can be understood observing that: a) the mass matrices are diagonal, b) the computation of the directional derivatives at a given quadrature point involves only a limited subset of the degrees of freedom of each element, and c) the computation of the numerical fluxes and the corresponding boundary integrals involves only few degrees of freedom per each element.

We now briefly illustrate points b) and c) in the case $d = 1$. In Fig. 5.1, left, the LGL points for $k = 3$ and for an element $K \in \mathcal{T}$ in a two-dimensional phase space are shown with "bullets". According to the definition of the Lagrangian basis and of the quadrature rule given in Section 3.1, these points serve both as degrees of freedom and quadrature nodes. In the same figure, the LGL points defined on each side of the element are also displayed with "stars". Since the LGL points always include the boundaries of the quadrature integrals, the "stars" are superimposed to some of the "bullets". To evaluate the volume integrals appearing in the DG-SE formulation, we need to sample the various fields at the internal quadrature points, together with the corresponding derivatives. Sampling the fields at an LGL point is straightforward, since it amounts to accessing the collocated degree of freedom. Sampling the derivatives can be done in terms of few degrees of freedom: for instance, considering Fig. 5.1, right, to compute $\partial_{\boldsymbol{x}} f_h$ at the bottom left, internal node, only the four degrees of freedom highlighted in red are required. Similarly, the computation of $\partial_{\boldsymbol{v}} f_h$ at the same point requires the four degrees of freedom highlighted in green. To evaluate the boundary integrals, the numerical fluxes are required at the boundary quadrature points, marked with a star, which requires sampling the various fields at these points. Thanks to the fact that the "stars" are superimposed to the "bullets", this operation reduces again to access a limited number of local degrees of freedom.

6. Time discretization

Eq. (5.3) is an ordinary differential equation which can be written as

$$\mathbf{f}' = \mathbf{R}(\mathbf{f}), \tag{6.1}$$

where $\mathbf{f}$ is the array of the nodal values of the distribution function f_h and $\mathbf{R}$ denotes the right-hand-side; (6.1) can be discretized in time using a time integrator method, following the classical approach of the method of lines. The standard choice is represented by Runge–Kutta type methods, and we consider here the classical RK4 method. Hence, introducing a time step Δt and time levels $t^n = n\,\Delta t$, for $n = 0, \ldots, T/\Delta t$, to advance the numerical solution from $\mathbf{f}^n = \mathbf{f}(t^n)$ to $\mathbf{f}^{n+1}$ we have to solve

$$\mathbf{K}_1 = \mathbf{R}(\mathbf{f}^n), \qquad \mathbf{K}_2 = \mathbf{R}\left(\mathbf{f}^n + \frac{\Delta t}{2}\mathbf{K}_1\right)$$
$$\mathbf{K}_3 = \mathbf{R}\left(\mathbf{f}^n + \frac{\Delta t}{2}\mathbf{K}_2\right), \qquad \mathbf{K}_4 = \mathbf{R}(\mathbf{f}^n + \Delta t\,\mathbf{K}_3)$$
$$\mathbf{f}^{n+1} = \mathbf{f}^n + \frac{\Delta t}{6}(\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4). \tag{6.2}$$

At each stage of (6.2), the computation of $\mathbf{R}(\mathbf{f})$ requires the solution of the Poisson problem (5.1).

Notice that the energy balances (4.6) and (4.11) are derived using the product rule for the exact time derivative, which does not hold for the time integrator (6.2). Hence, an error is introduced in the energy balance proportional to the time discretization error. This error is quantified numerically in Section 7. This error could be avoided using a centered scheme as discussed in [36], but this would lead to an implicit formulation.

The time integrator (6.2), albeit representing and adequate choice for our problem, could be less than optimal concerning the computational efficiency. In fact, in order to compute stable numerical simulations, the time-step must be typically smaller than what would be desirable in terms of a trade-off between accuracy and efficiency. There are at least three possibility to tackle this issue: semi-Lagrangian methods [44,52], implicit time integrators, and exponential time integrators. Each of these methods has merits and drawbacks; here we would like to explore the use of exponential time integrators as an alternative to RK ones.

The idea of integrating exactly the linear part of an ODE, thus obtaining an exponential integrator, in order to improve both the accuracy and the stability of the scheme, has been first considered in [37]. A number of methods have then been proposed and analyzed, for both scalar equations and systems. Here, we follow the presentation of [53], where the interested reader can also find a more comprehensive bibliography. The first step in the construction of an exponential time integrator is rewriting a generic time-step of (6.1) as

$$\mathbf{f}' = \mathcal{J}^n_{\mathbf{R}}(\mathbf{f} - \mathbf{f}^n) + \mathcal{O}\big((\mathbf{f} - \mathbf{f}^n)^2\big), \quad t \in (t^n, t^{n+1})$$
$$\mathbf{f}(t^n) = \mathbf{f}^n \tag{6.3}$$

where $\mathcal{J}^n_{\mathbf{R}}$ is the Jacobian of the right-hand-side evaluated at $\mathbf{f}^n$. Discarding the higher order term, (6.3) reduces to a linear equation, which can be integrated exactly obtaining

$$\mathbf{f}^{n+1} = \mathbf{f}^n + \Delta t\,\phi\big(\Delta t\,\mathcal{J}^n_{\mathbf{R}}\big)\mathbf{R}(\mathbf{f}^n) \tag{6.4}$$

where the function ϕ is defined as

$$\phi(z) = \frac{\mathrm{e}^z - 1}{z}$$

for a scalar argument $z \in \mathbb{C}$ and is extended for matrix arguments by power series (see also [31]). Method (6.4) is the simplest exponential integrator that can be constructed; thanks to the presence of the Jacobian matrix, it is a second order method, which is referred to as Exponential Rosenbrock of second order (ERB2). Before this method can be used to solve large systems of differential equations, however, two nontrivial problems must still be addressed, namely the computation of the Jacobian matrix and the computation of $\phi(\Delta t\,\mathcal{J}^n_{\mathbf{R}})$. Many approaches have been discussed to compute the exponential of a generic matrix $\mathcal{A}$, or equivalently the quantity $\phi(\mathcal{A})$, and a review can be found in [42]; in this work, we consider the following two alternatives: Krylov space iterations (see [33,50]) and interpolation at real Leja points (see [20,25]). Both these techniques are iterative methods, and each iteration requires the computation of a matrix-vector product of the form $\mathcal{A}\mathbf{v}$, for suitable vectors $\mathbf{v}$. This has some important consequences.

First of all, the fact that only the action of $\mathcal{A}$ is required, and not the explicit knowledge of the matrix itself, provides a convenient solution for the problem of computing the Jacobian matrix. In fact, since we are interested in the case $\mathcal{A} = \Delta t\mathcal{J}^n_{\mathbf{R}}$, we can consider the approximation

$$\mathcal{J}^n_{\mathbf{R}}\mathbf{v} \approx \frac{1}{\varepsilon_{\text{diff}}}\big(\mathbf{R}(\mathbf{f}^n + \varepsilon_{\text{diff}}\mathbf{v}) - \mathbf{R}(\mathbf{f}^n)\big), \tag{6.5}$$

where $\varepsilon_{\text{diff}}$ is a small parameter. Notice that a centered finite difference could be also considered, although at the price of doubling the evaluations of $\mathbf{R}(\mathbf{f})$; this solution has been considered in [28].

A second consequence is that a stopping criterion must be provided: a large number of iterations results in an accurate approximation of $\phi(\Delta t\,\mathcal{J}^n_{\mathbf{R}})$, but also leads to a higher computational cost. For both methods, an adaptive stopping criterion is available, based on an estimate of the error in the approximation of $\phi(\Delta t\,\mathcal{J}^n_{\mathbf{R}})$. For the Krylov space iterations, we follow [53], Section 2.2, and stop the Krylov iterations when

$$\big\|\phi\big(\Delta t\,\mathcal{J}^n_{\mathbf{R}}\big) - \widetilde{\phi\big(\Delta t\,\mathcal{J}^n_{\mathbf{R}}\big)}\big\|_2 \le \varepsilon_{\text{Kry}},$$

where a tilde denotes the numerical approximation and ε_{Kry} is a prescribed tolerance, while for the Leja point interpolation method new terms are introduced in the Newton polynomial interpolation as far as they are larger than $\varepsilon_{\text{Lej}}\|\widetilde{\phi(\Delta t\,\mathcal{J}^n_{\mathbf{R}})}\|_2$.

Concerning the computational cost, by virtue of (6.5), each iteration requires one additional evaluation of the right-hand-side, and thus a computational cost analogous to a stage of (6.2). A reasonable comparison of the computational cost of each time step of RK4 and ERB2 can thus be made in terms of the efficiency index

$$\eta_{\text{eff}} = \frac{4}{\Delta t^{\text{RK4}}} \frac{\Delta t^{\text{ERB2}}}{n_{\text{rhs}}}, \tag{6.6}$$

where n_{rhs} is the number of evaluations of the right-hand-side $\mathbf{R}(\mathbf{f})$ in the time step, namely 1 plus the number of evaluations of $\mathcal{J}_{\mathbf{R}}^n \mathbf{v}$ in (6.5), for either the Krylov space iterations or the interpolation at the Leja points.

The last consequence of the iterative nature of the computation of $\phi(\Delta t\, \mathcal{J}_{\mathbf{R}}^n)$ is that the convergence rate of the methods depends on the spectral properties of the matrix $\mathcal{J}_{\mathbf{R}}^n$. Some details concerning this aspect are collected in Appendix A.

7. Numerical validation

In this section we address the numerical validation of the spectral element DG formulation. The various test cases are selected to assess the performance of the method concerning the following aspects: convergence rates, overall ability to represent relevant physical phenomena, robustness and discrete conservation properties. References for the considered test cases can be found in [12,17,19,32,41,43,51]. For Sections 7.1, 7.2 and 7.3, the emphasis is on the spatial discretization, while for the time integration we consider the RK4 scheme with appropriately small time-step. The effect of the time discretization is considered in Section 7.4, where the RK4 scheme is compared with the exponential time integrators discussed in Section 6.

For most of the test cases, the electrostatic energy is an important diagnostic. To ensure consistency with the numerical discretization, for $\boldsymbol{E}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$ we compute the electrostatic energy as

$$\mathcal{E}_w^{\text{el}} = \int_{\Omega_{\boldsymbol{x}}, \boldsymbol{w}_+} \frac{E_h^2}{2}\, \mathrm{d}\boldsymbol{x} + \sum_{e_{\boldsymbol{x}} \in \mathcal{E}_{\boldsymbol{x}}} \frac{1}{2} \int_{e_{\boldsymbol{x}}, w} c_{11} [\![\Phi_h]\!]^2, \tag{7.1}$$

alongside with the kinetic energy

$$\mathcal{E}_w^{\text{kin}} = \iint_{\Omega, w} f_h \frac{v^2}{2}\, \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{v} \tag{7.2}$$

and the total energy defined by (5.4). When $\boldsymbol{E}_h \in \boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$, the same formula is used with the substitution $\int_{\Omega_{\boldsymbol{x}}, \boldsymbol{w}_+} \mapsto \int_{\Omega_{\boldsymbol{x}}, \boldsymbol{w}}$. For ease of comparison with other references, we often plot the square root $\sqrt{\mathcal{E}_w^{\text{el}}}$.

Given that in all the formulations considered here we take $f_h \in V_h^k(\mathcal{T})$, $\Phi_h \in V_h^k(\mathcal{T}_{\boldsymbol{x}})$ and the elliptic numerical fluxes (3.12), we then distinguish among the following formulations:

- centered method (CM): with $\boldsymbol{E}_h \in \boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$, $\tilde{\boldsymbol{E}}_h = \boldsymbol{E}_h$, according to (4.5), and centered numerical fluxes in (3.5) and (3.6);
- space centered, velocity upwind method (SCVU): with $\boldsymbol{E}_h \in \boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$, $\tilde{\boldsymbol{E}}_h = \boldsymbol{E}_h$ and centered and upwind numerical fluxes in (3.5) and (3.6), respectively;
- upwind method (UM): with $\boldsymbol{E}_h \in \boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$, $\tilde{\boldsymbol{E}}_h = \boldsymbol{E}_h$ and upwind numerical fluxes in (3.5) and (3.6);
- upwind, energy conserving method (UECM): with $\boldsymbol{E}_h \in \boldsymbol{RT}_h^k(\mathcal{T}_{\boldsymbol{x}})$, $\tilde{\boldsymbol{E}}_h$ defined by (4.10) and upwind numerical fluxes in (3.5) and (3.6).

7.1. Convergence rate of the electric field

We consider in this section the accuracy of the reconstruction of the electric field $\tilde{\boldsymbol{E}}_h$ for different choices of the finite element spaces and the stabilization coefficient. For this purpose, we solve (3.9) in two dimensions with

$$\rho(x^1, x^2) = \frac{1}{2}\pi^2 \cos\left(\frac{\pi}{2}x^1\right) \cos\left(\frac{\pi}{2}x^2\right) + 1 \tag{7.3}$$

and $\Omega_{\boldsymbol{x}} = (-2, 2)^2$, with solution

$$\Phi_{ex} = \cos\left(\frac{\pi}{2}x^1\right) \cos\left(\frac{\pi}{2}x^2\right).$$

We consider the L^2 error for the following definitions of the reconstructed electric field: $\boldsymbol{E}_{h,\text{LDG}}$ is the solution of (3.10) with $\boldsymbol{E}_h \in \boldsymbol{W}_h^k(\mathcal{T}_{\boldsymbol{x}})$ and $c_{11} = h^{-1}(k+1)^2$, $\tilde{\boldsymbol{E}}_{h,h^{-1}}$ is the field obtained with the reconstruction (4.10) and $c_{11} = h^{-1}(k+1)^2$ and $\tilde{\boldsymbol{E}}_{h,0}$ is the field obtained again with the reconstruction (4.10) but now setting $c_{11} = 0$ (for these last two reconstructions we consider the values corresponding to $v^\alpha \geq 0$ in (4.7), however similar results are obtained for different combinations of v^α).

Table 7.1
L^2 error norms for the three considered reconstructions of the electric field for problem (7.3) for polynomial degree $k = 4$ and $k = 5$, with the corresponding numerical convergence rates. The results are obtained considering uniform grids with $16n^{-2}$ elements for increasing n.

			$k = 4$			
$\log_2 n$	$\|\nabla_{\boldsymbol{x}}\Phi_{ex} - \boldsymbol{E}_{h,\mathrm{LDG}}\|_{L^2}$		$\|\nabla_{\boldsymbol{x}}\Phi_{ex} - \tilde{\boldsymbol{E}}_{h,h^{-1}}\|_{L^2}$		$\|\nabla_{\boldsymbol{x}}\Phi_{ex} - \tilde{\boldsymbol{E}}_{h,0}\|_{L^2}$	
1	$1.6 \cdot 10^{-1}$	–	$1.6 \cdot 10^{-1}$	–	$1.6 \cdot 10^{-1}$	–
2	$7.2 \cdot 10^{-3}$	4.485	$7.4 \cdot 10^{-3}$	4.443	$7.9 \cdot 10^{-3}$	4.356
3	$4.4 \cdot 10^{-4}$	4.042	$4.6 \cdot 10^{-4}$	4.026	$4.6 \cdot 10^{-4}$	4.094
4	$2.7 \cdot 10^{-5}$	4.013	$2.8 \cdot 10^{-5}$	4.008	$2.8 \cdot 10^{-5}$	4.029
			$k = 5$			
$\log_2 n$	$\|\nabla_{\boldsymbol{x}}\Phi_{ex} - \boldsymbol{E}_{h,\mathrm{LDG}}\|_{L^2}$		$\|\nabla_{\boldsymbol{x}}\Phi_{ex} - \tilde{\boldsymbol{E}}_{h,h^{-1}}\|_{L^2}$		$\|\nabla_{\boldsymbol{x}}\Phi_{ex} - \tilde{\boldsymbol{E}}_{h,0}\|_{L^2}$	
1	$6.0 \cdot 10^{-3}$	–	$6.0 \cdot 10^{-3}$	–	$6.0 \cdot 10^{-3}$	–
2	$5.8 \cdot 10^{-4}$	3.358	$5.9 \cdot 10^{-4}$	3.336	$6.0 \cdot 10^{-4}$	3.331
3	$1.9 \cdot 10^{-5}$	4.925	$1.9 \cdot 10^{-5}$	4.940	$1.9 \cdot 10^{-5}$	4.936
4	$6.1 \cdot 10^{-7}$	4.978	$6.1 \cdot 10^{-7}$	4.983	$6.1 \cdot 10^{-7}$	4.989

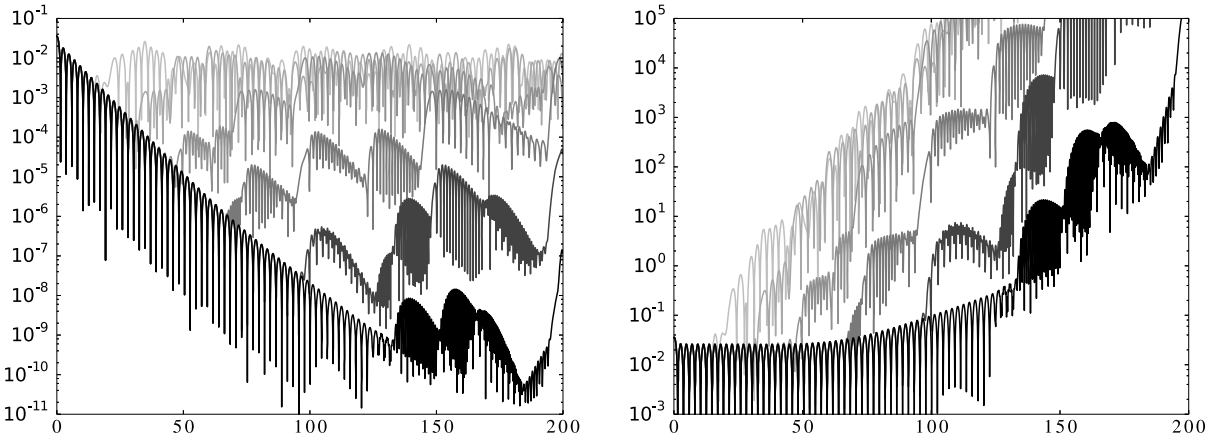


Fig. 7.1. Linear Landau damping, time evolution of the electrostatic energy for $k = 6$ and grid resolution, from light gray to black: 5×10 elements in space and velocity, respectively, 10×20 elements, 20×40 elements, 40×80 elements, 80×160 elements and 160×320 elements. The left plot shows $\sqrt{\mathcal{E}_w^{\mathrm{el}}}$, while the right plot shows $\exp(0.1534\,t)\sqrt{\mathcal{E}_w^{\mathrm{el}}}$.

The results are shown in Table 7.1 for the two cases $k = 4$ and $k = 5$; it can be seen that all the reconstructions display the optimal convergence rate, and that the values of the errors are comparable. Other results, not included here, suggest that the fact that the asymptotic convergence rate is approached from above for even k and from below for odd k is a general feature.

7.2. One-dimensional test cases

The first test case considered here is the standard Landau damping problem, using the setup described in [12,19,32,41], which we briefly recall for the sake of completeness. The computational domain is $\Omega_{\boldsymbol{x}} = (0, 4\pi)$ and $\Omega_{\boldsymbol{v}} = (-10, 10)$ and the initial condition is

$$f_0(x, v) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\frac{v^2}{2}} \big(1 + A \sin(\kappa x)\big), \tag{7.4}$$

with $\kappa = 0.5$. The parameter A can be chosen in order to obtain either a linear or a nonlinear problem: we consider here $A = 0.01$, resulting in a linear regime, and $A = 0.5$, corresponding to a nonlinear one.

Concerning the linear regime, the time evolution of $\sqrt{\mathcal{E}_w^{\mathrm{el}}}$ is represented in Fig. 7.1 for the UECM with $c_{11} = 0$ and polynomial degree $k = 6$, for various phase space resolutions. The time-step is $\Delta t = 6.25 \cdot 10^{-4}$ for all the computations, so that the error results essentially from the space discretization. For this problem, the dominant mode of the linear solution is characterized by the damping constant $\gamma = -0.1534$ and the angular frequency $\omega = 1.416$. It can be observed that all the computations are in good agreement with the linear solution until a discrete recurrence appears. To simplify the comparison with the linear solution, we plot in Fig. 7.1, right, the quantity $\exp(0.1534\,t)\sqrt{\mathcal{E}_w^{\mathrm{el}}}$, which should exhibit oscillations with constant amplitude. Here, it can be observed that the computed electric energy in fact does show constant amplitude oscillations up to $t \approx 50$, provided a sufficient resolution is used, while for larger times the amplitude increases (i.e., the numerical solution decays at a lower rate compared with the linear solution). This effect, which can be observed also in [41],

is independent from the resolution and is due to nonlinear effects; additional computations with $A = 0.001$, not reported here, are in agreement with the linear solution for longer times. The LGL points for $k = 6$ are

$$\left\{\hat{\xi}_i^{1D,6},\ i = 0, \ldots, 6\right\} = \{-1.00, -0.83, -0.47, 0.00, 0.47, 0.83, 1.00\}.$$

These points, together with the associated weights, define a local grid within each element with nonuniform spacing Δv, which is responsible for the various recurrence times observed in the electric energy, according to the analysis of [17]. A more detailed analysis is presented in [12]; here, we mention that our results are qualitatively similar to those of this last reference as well as [5]. From our results, it seems that the nonuniform distribution of the LGL points anticipates the recursion: a uniform grid with the same number of degrees of freedom as the spectral element, DG formulation with $k = 6$ would have $\Delta v \approx \frac{1}{7}|K_{\boldsymbol{v}}|$, resulting in a recurrence time $\tau^{\text{rec}} = \frac{2\pi}{\kappa\,\Delta v}$, while our results suggest that the electric energy reaches a level comparable to the initial condition after a time which is consistent with a spacing $\Delta v \approx \frac{1}{4}|K_{\boldsymbol{v}}|$. Since the clustering of the LGL points toward the element boundaries becomes more pronounced for increasing polynomial degree k, this could provide an upper limit for k for practical computations.

Concerning the nonlinear Landau damping problem, we consider the time evolution of the electrostatic energy, the effect of the numerical dissipation and the positivity of the distribution function for the UECM with $c_{11} = 0$ and different combinations of polynomial degree and grid resolution. More specifically, we consider the low resolution cases: $k = 4$ and 56×112 elements in space and velocity, $k = 5$ and 47×94 elements, $k = 6$ and 40×80 elements, $k = 7$ and 35×70 elements; high resolution cases are then obtained, for the same polynomial degrees, by doubling the number of elements in space and velocity. The number of degrees of freedom is approximately the same among the low resolution cases as well as among the high resolution ones. For all the computations, the time-step is $\Delta t = 6.25 \cdot 10^{-4}$. The evolution of the electrostatic energy for the first four modes, defined as (see [32])

$$\mathcal{E}_{w,n}^{\text{el}} = \frac{\kappa}{2\pi}\left(E_{s,n}^2 + E_{c,n}^2\right)$$

with

$$E_{s,n} = \int_{\Omega_{\boldsymbol{x}},\boldsymbol{w}_+} E_h \sin(n\kappa x)\,\mathrm{d}x, \qquad E_{c,n} = \int_{\Omega_{\boldsymbol{x}},\boldsymbol{w}_+} E_h \cos(n\kappa x)\,\mathrm{d}x,$$

is displayed in Fig. 7.2. Since the results for the high resolution cases are almost identical, suggesting that this is a grid converged solution, for such cases we only include plots for $k = 4$ and $k = 7$, while all the low resolution cases are shown. It can be observed that all the considered combinations of polynomial degrees and space resolutions result in an accurate representation of the first two modes, while the third and the fourth modes are better represented when using fewer elements with higher polynomial degree, especially concerning the build-up of electrostatic energy for $t \in [20\,, 40]$. To assess the role of the numerical dissipation, we plot in Fig. 7.3 the L^2 norm of the distribution function. Here, it can be observed that the behaviour of the scheme is always dissipative, as anticipated in Section 4.2; the numerical dissipation is larger for lower polynomial degrees. The L^1 norm of the distribution function is reported in Fig. 7.4. Since the number of particles is conserved up to machine precision by construction, changes in $\|f\|_{L^1}$ are due to violations of the positivity constraint $f(t, x) \geq 0$ implied by (2.11). Here, one can notice that negative values are produced roughly at the same rate for different values of k, and they are then dissipated at a rate which is inversely proportional to the polynomial degree; clearly, this is consistent with the numerical dissipation pattern resulting from Fig. 7.3. For all the considered simulations, the total energy shows deviations within a range $\mathcal{O}(10^{-12})$: this is expected, since the total energy is conserved up to errors introduced by the time discretization, and the time-step is small.

The second test case considered in this section is the so-called "bump-on-tail" instability. This problem has been considered in [51] and some additional reference results are available in [19,41,43]. The computational domain is $\Omega_{\boldsymbol{x}} = (0\,, 20\pi)$ and $\Omega_{\boldsymbol{v}} = (-9\,, 9)$ and the initial condition is

$$f_0(x, v) = \frac{1}{10\sqrt{2\pi}}\left(1 + A\cos(\kappa x)\right)\left(9\mathrm{e}^{-\frac{v^2}{2}} + \frac{1}{v_{\text{th}}}\mathrm{e}^{-\frac{1}{2}\left(\frac{v - v_d}{v_{\text{th}}}\right)^2}\right) \tag{7.5}$$

with $A = 0.04$, $\kappa = 0.3$, $v_d = 4.5$ and $v_{\text{th}} = 0.5$. $\mathcal{T}_{\boldsymbol{x}}$ is composed of 18 uniform elements, while for $\mathcal{T}_{\boldsymbol{v}}$ we use 12 nonuniform elements with boundaries $[-9, -5.5, -3, -1, 0, 1, 2, 3, 4, 5, 6, 7.5, 9]$. Notice that it is important that the number of elements in $\mathcal{T}_{\boldsymbol{x}}$ is a multiple of three to represent correctly the periodicity of the solution. The polynomial order is $k = 6$. Fig. 7.5 shows the distribution function at time $t = 50$ using the four discretizations: CM, SCVU, UM, UECM. For the first three methods we set $c_{11} = h^{-1}(k + 1)^2$, with $h = \frac{1}{2}(|K_{\boldsymbol{x}}| + |K_{\boldsymbol{v}}|)$, while for the energy conserving upwind method we set $c_{11} = 0$. The time-step is $\Delta t = 3.75 \cdot 10^{-3}$, so that the time discretization error is negligible. Clearly, the use of an upwind flux in the x direction is required in order to obtain a solution free from spurious oscillations. The solution obtained with the centered method becomes unstable for larger times, while the solution computed with upwinding in the sole v direction is stable and, for $t \geq 100$, the oscillations are mild and a qualitative agreement with the solutions obtained with the two fully upwind methods is observed. The solutions obtained with the two upwind methods are very similar during the whole time evolution. Fig. 7.6 displays the L^2 and L^1 norms of the distribution function for the three stable methods. These plots confirm that the two solutions obtained with the fully upwind methods are very similar, and show that, despite the fact

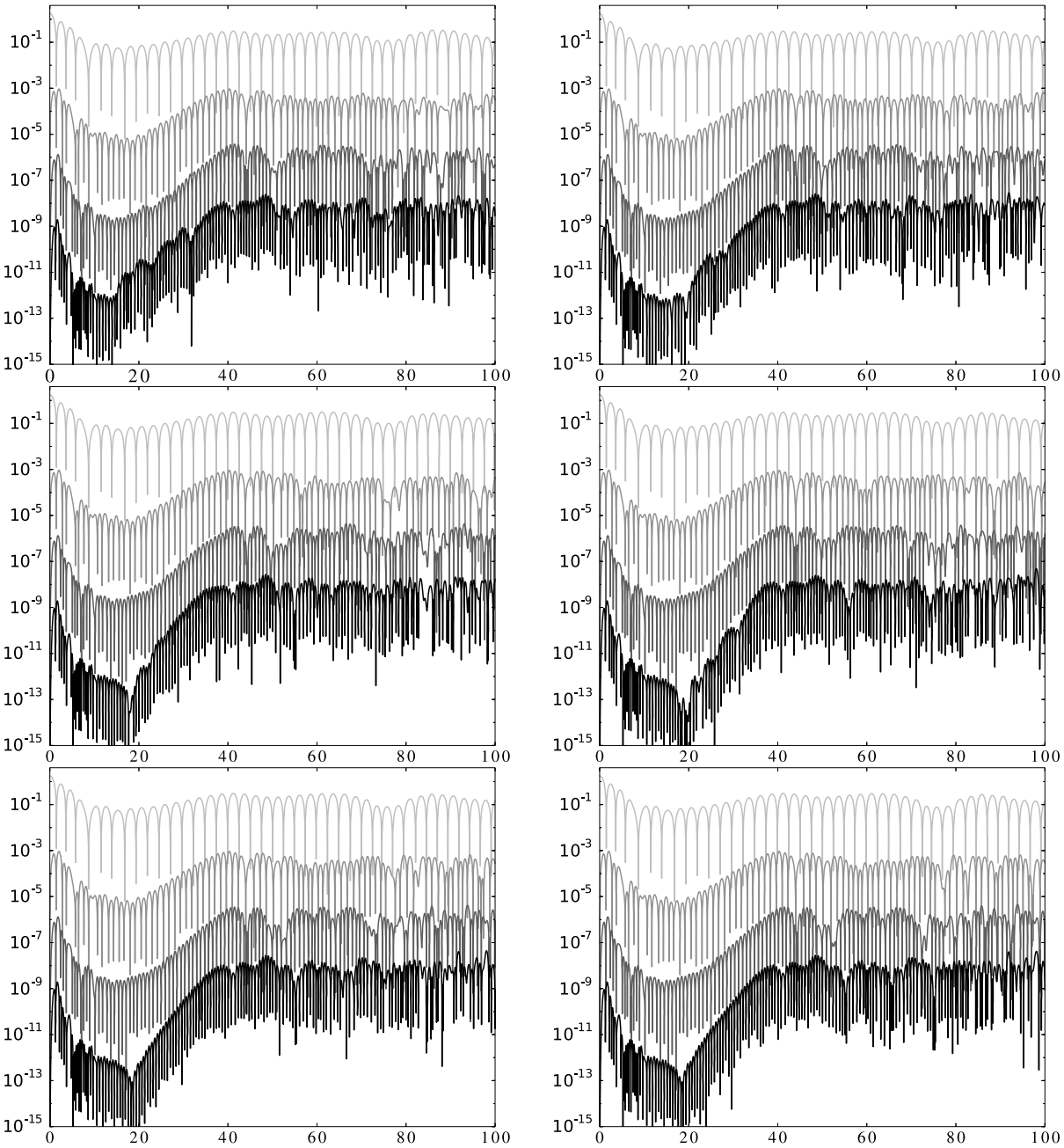


Fig. 7.2. Nonlinear Landau damping, time evolution of the electrostatic energy for the first four modes, n increases from light gray to black. To separate the curves of the four modes, for each mode we plot $100^{-n+1}\sqrt{\mathcal{E}^{\mathrm{el}}_{w,n}}$. Results for the low resolution cases $k=4$ and 56×112 elements (top, left), $k=5$ and 47×94 elements (top, right), $k=6$ and 40×80 elements (center, left), $k=7$ and 35×70 (center, right), and for two high resolution cases $k=4$ and 112×224 elements (bottom, left) and $k=7$ and 70×140 elements (bottom, right).

that the L^2 norm of the solution is dissipated roughly at the same rate, the violation of the positivity constraint $f(t,x)\geq 0$ is significantly larger when the upwind method is used only in the velocity space. Concerning the total energy conservation, we plot the time evolution of the total energy in Fig. 7.7 for different time-steps, the largest of which is the maximum time-step resulting in a stable computation. It can be seen that, while for the upwind method the deviation of the total energy is independent from Δt (except for the very beginning of the computation) and is thus dominated by the space discretization error, for the energy conserving, upwind method the energy is exactly conserved in the limit $\Delta t\to 0$. Finally, concerning the momentum conservation, we mention that both upwind methods show variation of the total momentum within 0.1% of the initial condition, and these variations are almost independent from the time-step.

The third test case is the two-stream instability, for which we refer to [19,32]. The initial condition is

$$f_0(x,v)=\frac{1}{2\sqrt{2\pi}\,v_{\mathrm{th}}}\left(1+A\cos(\kappa x)\right)\left(\mathrm{e}^{-\frac{1}{2}(\frac{v-v_d}{v_{\mathrm{th}}})^2}+\mathrm{e}^{-\frac{1}{2}(\frac{v+v_d}{v_{\mathrm{th}}})^2}\right) \tag{7.6}$$

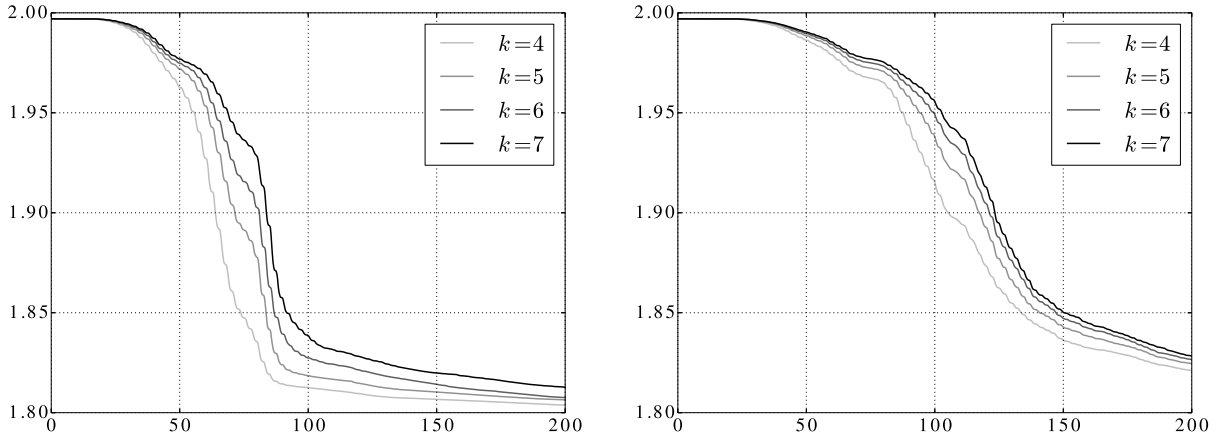


Fig. 7.3. Nonlinear Landau damping, time evolution of the L^2 norm of the distribution function for the low resolution runs (left) and the high resolution ones (right).

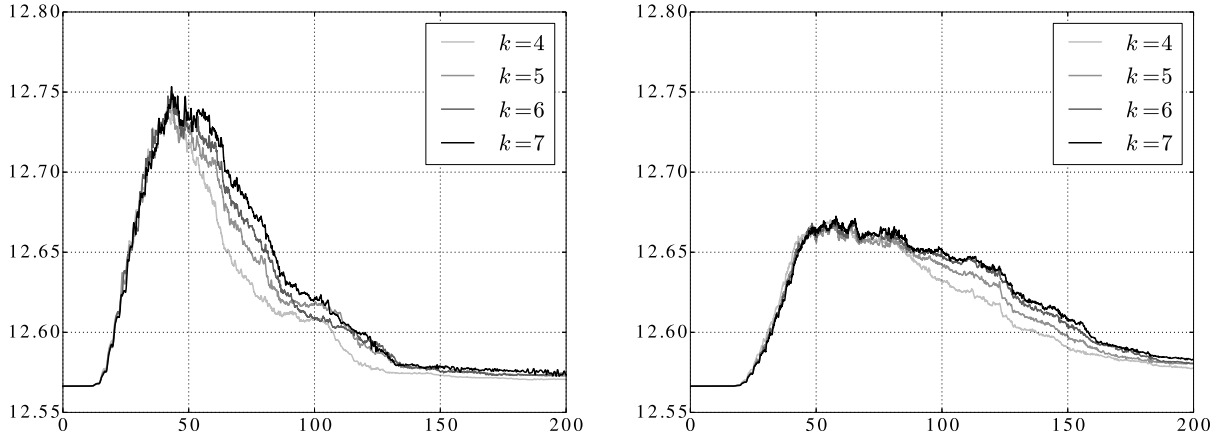


Fig. 7.4. Nonlinear Landau damping, time evolution of the L^1 norm of the distribution function for the low resolution runs (left) and the high resolution ones (right).

with $v_{\text{th}} = 0.3$, $A = 0.05$, $\kappa = 1$, $v_d = 0.99$. The computational domain is $\Omega_{\boldsymbol{x}} = (0, 26\pi)$, $\Omega_{\boldsymbol{v}} = (-8, 8)$, and we use a mesh composed of 52 elements in x and 120 elements in v; the polynomial order is $k = 5$ and the time-step $\Delta t = 5 \cdot 10^{-3}$. There are many aspects of this test case which can be considered, and a thorough discussion can be found in [32]. Here, we are mostly interested in using this test as an example of a problem exhibiting very sharp gradients in the distribution function, thus representing a challenging problem for the stability of the numerical method. For this reason, the resolution is rather coarse: each wave period in x is resolved with four elements and the relevant dynamics in v is resolved with ≈ 15 elements. During the initial phase of the computation, the instability develops and thirteen electron hole-like structures are formed for $20 \leq t \leq 40$, which are then stable until $t \approx 110$. At this point, because of the truncation errors introduced by the numerical computation, the periodicity, as well as the symmetry, of the solution are destroyed and these structures start merging, resulting into fewer, larger structures. This is the process that leads to the sharpest gradients in the distribution function. At the final time $t = 1000$ only two electron hole-like structure are still present. The whole process can be seen in Fig. 7.8 for the UECM computation (with $c_{11} = 0$); similar results, not included here, are obtain with the UM (using $c_{11} = h^{-1}(k+1)^2$ as in the previous test case). Indeed, the solutions computed with UECM and UM are very similar, as it is also confirmed by Fig. 7.9, where the L^2 and L^1 norms of the distribution function are displayed. Notice the "staircase" pattern in the L^2 norm of the solution: rapid changes in this norm correspond to strong numerical dissipation, which takes place during the formation of the electron hole-like structures for $t \approx 20$ and during the merging of such structures occurring, for UECM, at times 110, 350 and 570. The fact that the secondary merging happens at different times for the UM compared to the UECM should not be of concern, since this process is triggered by the truncation errors. The main differences between the UM and the UECM can be found in the total energy balance and in the presence of numerical overshoots and undershoots of the distribution function; these quantities are shown in Fig. 7.10. During the formation of the electron hole-like structures at $t \approx 20$ the UM shows a significant change in the total energy, and both methods display over- and undershoots of the same entity. For larger times, during the merging of these structures, the UM shows a second, more pronounced change in the total energy and both methods exhibit severe over- and undershoots, with the UECM being characterized by overshoots twice as large as the UM. The larger overshoots of the UECM method are due to few, isolated

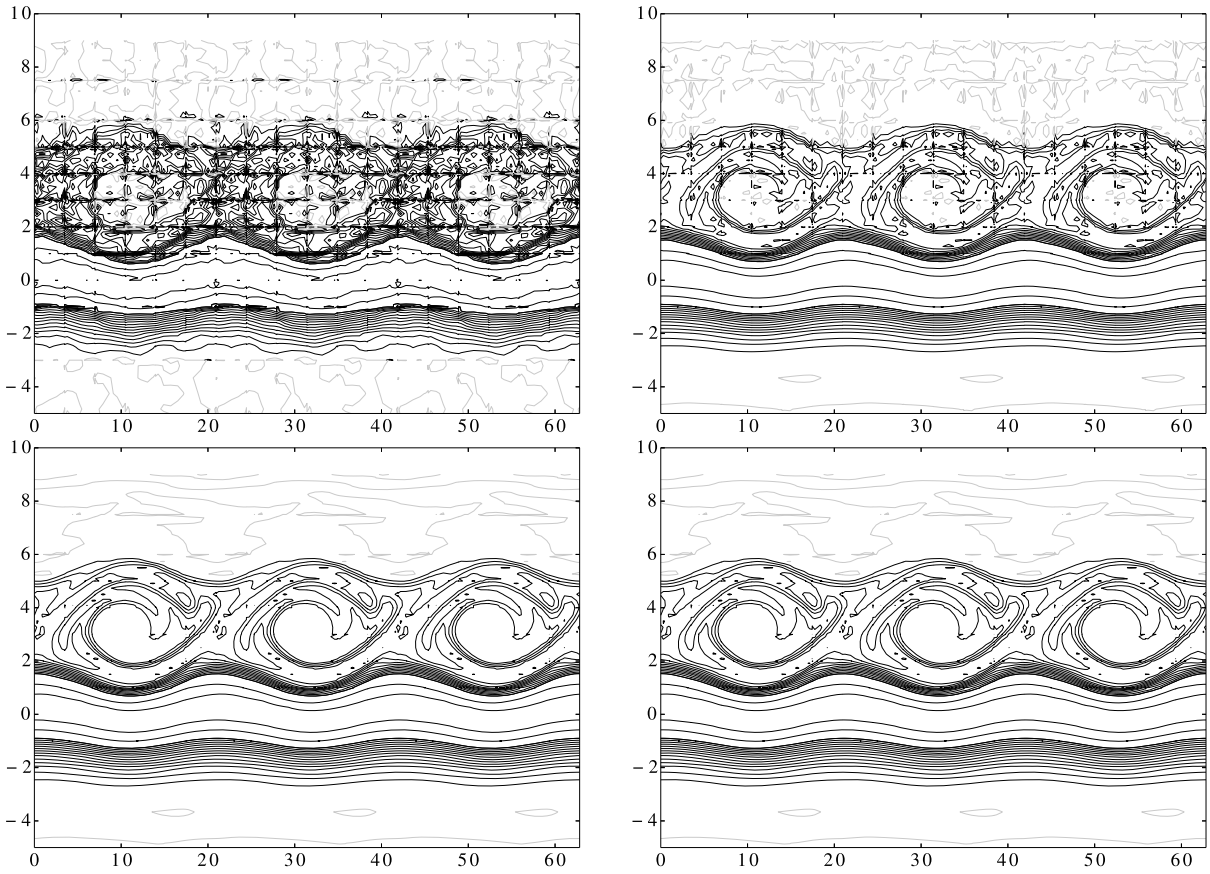


Fig. 7.5. Bump-on-tail test case, contour plot of the distribution function at $t = 50$, zero contour in gray. Results for the CM (top, left), SCVU (top, right), UM (bottom, left), and UECM (bottom, right).

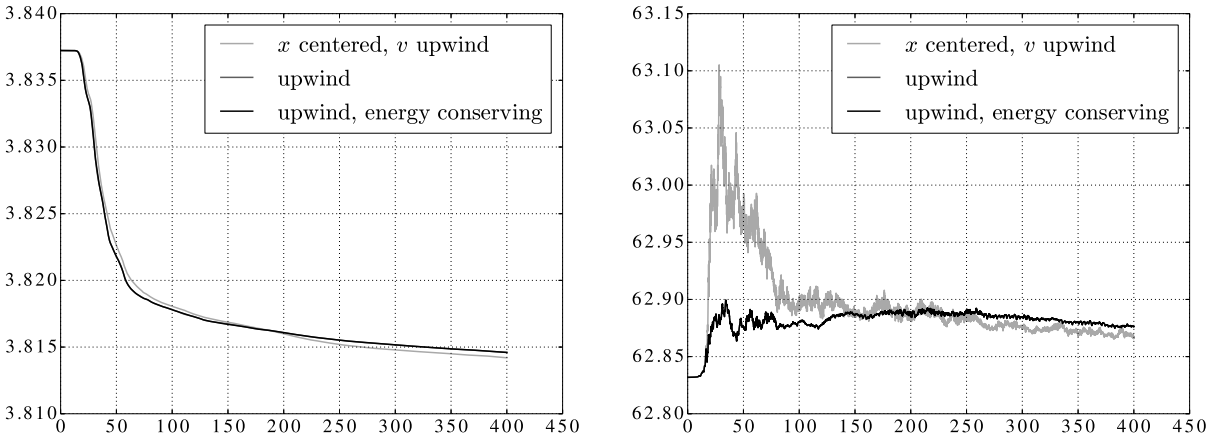


Fig. 7.6. Bump-on-tail test case, time evolution of the L^2 (left) and L^1 (right) norms of the distribution function. The two curves of the fully upwind methods are almost superimposed.

points located on the $v = 0$ axis at positions where $\tilde{\boldsymbol{E}}_h$ has large jumps going from $v < 0$ to $v > 0$; for this reason, one could regard these larger overshoots as the "price" to be paid for the introduction of an energy conserving reconstruction of the electric field. We mention that it might be possible to cure this effect with the introduction of some smoothing mechanism for elements close to the $v = 0$ axis, or by introducing a smooth transition in the definition of the upwind scheme exploiting the upwinding parameter ω in (3.7), but we do not explore this topic any further in this work. Finally, we mention that the total momentum remains with $\mathcal{O}(10^{-12})$ for $t \leq 110$; after this point the symmetry of the solution is lost and fluctuations in the total momentum appear which are of the order $\mathcal{O}(10^{-3})$ for both methods.

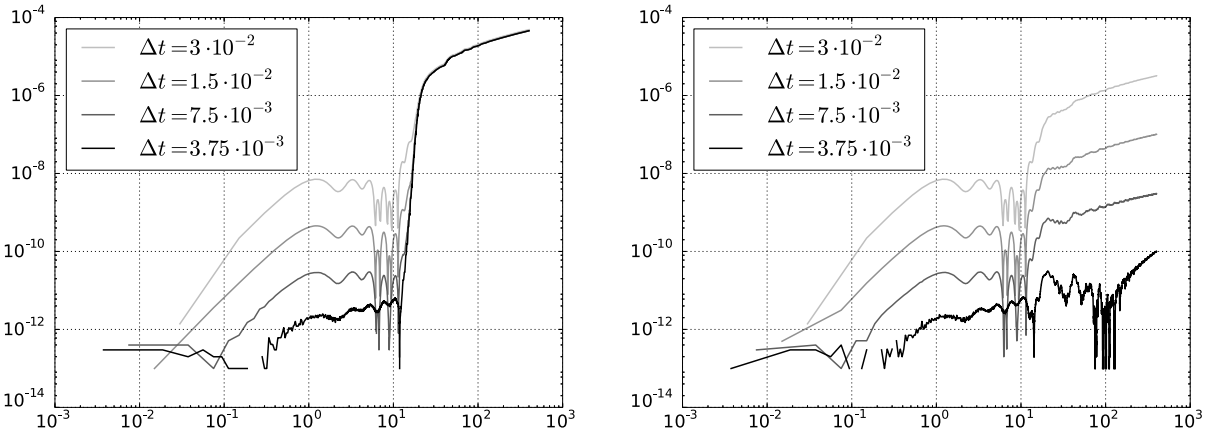


Fig. 7.7. Total energy deviation $|\mathcal{E}_w(t) - \mathcal{E}_w(0)|$ for the bump-on-tail test case for different time-steps; a logarithmic scale is used for the time axis to highlight both the initial transient and the long term behaviour. Results for the UM (left) and the UECM (right).

7.3. Two-dimensional test cases

In this section, we consider a two dimensional version of the two-stream instability problem. The initial condition is

$$f_0(x^1, x^2, v^1, v^2) = \frac{1}{4\pi\, v^1_{\text{th}}\, v^2_{\text{th}}} \left(1 - A^1 \sin(\kappa^1 x^1) - A^2 \sin(\kappa^2 x^2)\right) \times \left(\mathrm{e}^{-\frac{1}{2}\left[\left(\frac{v^1 - v_d}{v^1_{\text{th}}}\right)^2 + \left(\frac{v^2}{v^2_{\text{th}}}\right)^2\right]} + \mathrm{e}^{-\frac{1}{2}\left[\left(\frac{v^1 + v_d}{v^1_{\text{th}}}\right)^2 + \left(\frac{v^2}{v^2_{\text{th}}}\right)^2\right]}\right), \tag{7.7}$$

with $v^1_{\text{th}} = 1.0$, $v^2_{\text{th}} = 0.5$, $A^1 = 0.05$, $A^2 = 0.25$, $\kappa^1 = \kappa^2 = 0.2$, and $v_d = 2.4$. The perturbation in x^1 direction is small since its purpose is triggering the instability of the two streams, which are directed along x^1. In the x^2 direction the velocity has a Gaussian distribution, so we apply a relatively large perturbation to obtain a Landau damping behaviour. The chosen combination $\kappa^2, v^2_{\text{th}}$ produces a very small damping, in order to ensure that the problem maintains its two-dimensional character. The computational domain is $\Omega_{\boldsymbol{x}} = (0, 10\pi)^2$, $\Omega_{\boldsymbol{v}} = (-10, 10)^2$, and the mesh is composed of 5×5 uniform elements in $\Omega_{\boldsymbol{x}}$ and 64×32 uniform elements in $\Omega_{\boldsymbol{v}}$ with $k = 5$. The computations are performed with the UECM setting $c_{11} = h^{-1}(k+1)^2$. The time evolution of the electrostatic energy and of the L^2 norm of the distribution function are plotted in Fig. 7.11, where it can be seen that all the quantities show the expected behaviour. Fig. 7.12 shows two phase-space sections for $x^2 = 0$, $v^2 = 0$ of the distribution function for time levels $t = 12$ and $t = 100$. Here, it can be seen that the growth of the instability in the two streams has lead to a pattern similar to Fig. 7.8. For this computation, energy is conserved within relative fluctuations of order $\mathcal{O}(10^{-10})$, while momentum exhibits fluctuations of order $\mathcal{O}(10^{-2})$.

7.4. Accuracy and efficiency of the time discretization

In this section, we compare the RK4 and ERB2 time integration methods discussed in Section 6. To this end, we consider the "bump-on-tail" instability test case with the same setup described in Section 7.2. A reference solution is obtained using the RK4 method with $\Delta t = 3.75 \cdot 10^{-3}$, which is eight times smaller than the maximum stable time-step for RK4, namely $\Delta t^{RK4} = 3.0 \cdot 10^{-2}$. We now solve this problem for $t \in (0, 61.44)$ with the ERB2 method using both Krylov iterations and Leja point interpolation for the computation of the matrix exponential, for different time-steps and different tolerances ε_{Kry} and ε_{Lej}, and compute the L^2 error norm at $t = 61.44$ and the efficiency index η_{eff}, defined in (6.6), resulting from the average number of iterations per time step. As discussed in Section 6, such an index quantifies the ratio of the computational costs of the RK integrator and of the exponential one. The interval for the interpolation with the Leja points is chosen as $[-76\Delta t, 0]$, based on an estimate of the eigenvalues of $\mathcal{J}^n_{\mathbf{R}}$. The results are collected in Table 7.2 for the Krylov iteration method and in Table 7.3 for the Leja point interpolation one. For comparisons, the RK4 method with time-step Δt^{RK4} yields an error equal to $9.5 \cdot 10^{-5}$. Clearly, for $\Delta t^{\text{ERB2}} \leq \Delta t^{\text{RK4}}$, ERB2 is not competitive with RK4, since it tends to be both less accurate and less efficient. However, for $\Delta t^{\text{ERB2}} > \Delta t^{\text{RK4}}$, ERB2 can be up to twice more efficient than RK while the resulting errors are still of the order $\mathcal{O}(10^{-2})$, thereby allowing a trade-off between accuracy and efficiency which is not possible with RK4. Comparing now the two options for the computation of the exponential matrix, despite the fact that for each combination of accuracy and time-step one option can perform better than the other, no systematic pattern emerges, so that none of them seems to be clearly superior for the considered problem. On the one hand, for small time-steps the L^2 error in the solution is dominated for both methods by the error in the computation of $\phi(\Delta t\, \mathcal{J}^n_{\mathbf{R}})$, so that small values of ε_{Kry} and ε_{Lej} result in higher accuracy, and for the smallest considered tolerance the second order convergence rate of the ERB2 scheme can be observed. For larger time-steps, on the other hand, the limiting factor is the accuracy of the ERB2 scheme and no significant difference is observed reducing ε_{Kry} or ε_{Lej}. The largest considered time step $\Delta t = 0.48$ results in rather inaccurate solutions which, albeit being stable, are of little practical interest.

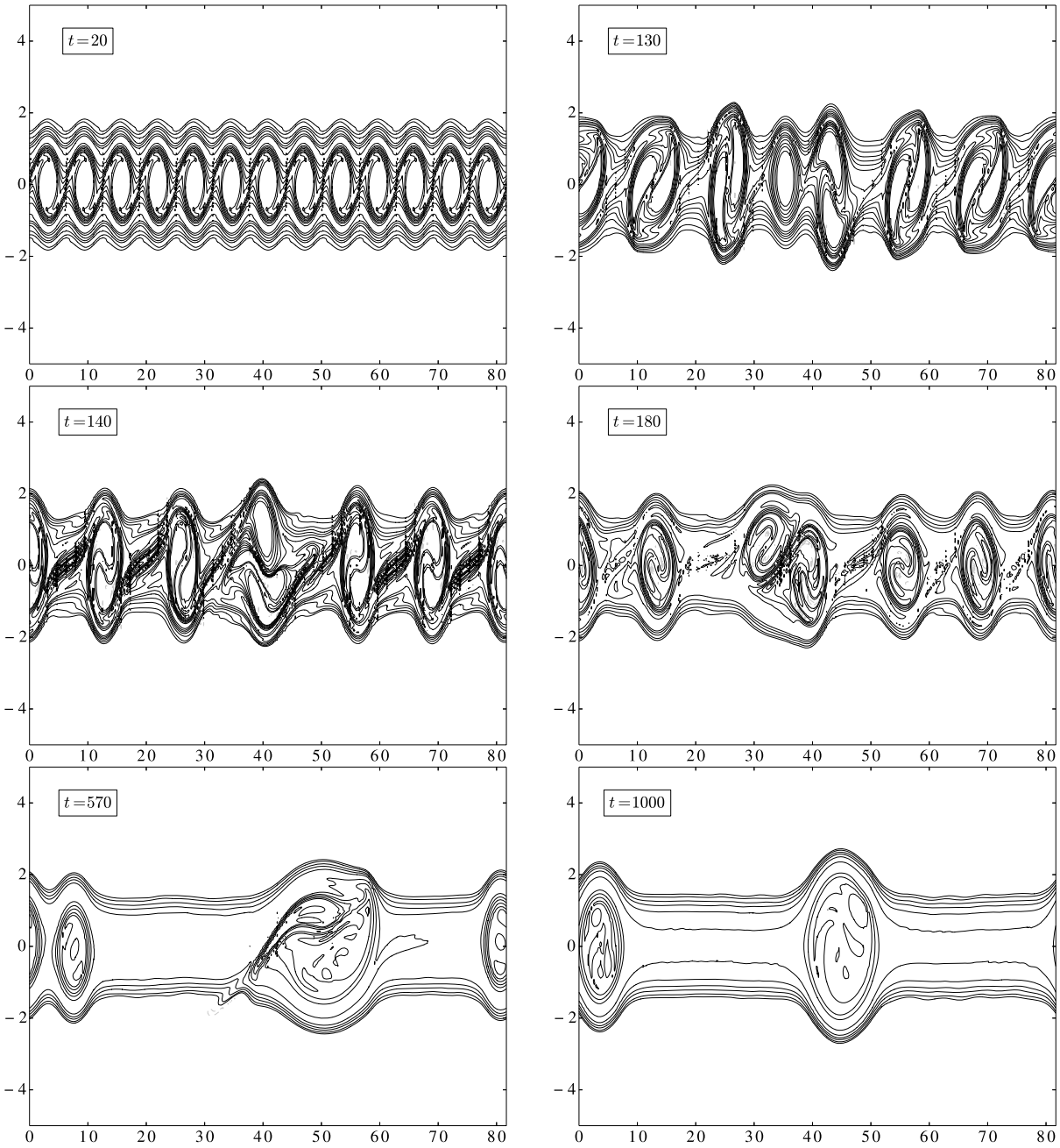


Fig. 7.8. Two-stream instability test case: contour plots of the distribution function obtained with the UECM at different time levels.

Table 7.2
L^2 error norms and efficiency index (6.6) for the ERB2 method with Krylov iterations, for varying accuracy in the computation of the exponential matrix; the problem considered here is the "bump-on-tail" instability, using the upwind, energy conserving method with the same setup of Section 7.2.

Δt^{ERB2}	$\varepsilon_{\text{Kry}} = 10^{-9}$		$\varepsilon_{\text{Kry}} = 10^{-6}$		$\varepsilon_{\text{Kry}} = 10^{-3}$	
	$\|f_{\text{ref}} - f_h\|_{L^2}$	η_{eff}	$\|f_{\text{ref}} - f_h\|_{L^2}$	η_{eff}	$\|f_{\text{ref}} - f_h\|_{L^2}$	η_{eff}
$7.5 \cdot 10^{-3}$	$4.6 \cdot 10^{-5}$	0.17	$4.2 \cdot 10^{-4}$	0.26	$6.3 \cdot 10^{-1}$	0.39
$1.5 \cdot 10^{-2}$	$1.8 \cdot 10^{-4}$	0.30	$1.9 \cdot 10^{-4}$	0.42	$1.2 \cdot 10^{-1}$	0.70
$3.0 \cdot 10^{-2}$	$7.4 \cdot 10^{-4}$	0.47	$7.5 \cdot 10^{-4}$	0.71	$2.4 \cdot 10^{-1}$	1.24
$6.0 \cdot 10^{-2}$	$3.0 \cdot 10^{-3}$	0.69	$3.0 \cdot 10^{-3}$	1.00	$6.1 \cdot 10^{-2}$	1.53
$1.2 \cdot 10^{-1}$	$1.3 \cdot 10^{-2}$	0.91	$1.3 \cdot 10^{-2}$	1.25	$3.1 \cdot 10^{-2}$	1.76
$2.4 \cdot 10^{-1}$	$7.9 \cdot 10^{-2}$	1.14	$7.9 \cdot 10^{-2}$	1.46	$8.5 \cdot 10^{-2}$	1.89
$4.8 \cdot 10^{-1}$	$8.8 \cdot 10^{-1}$	1.35	$8.8 \cdot 10^{-1}$	1.62	$8.9 \cdot 10^{-1}$	1.99

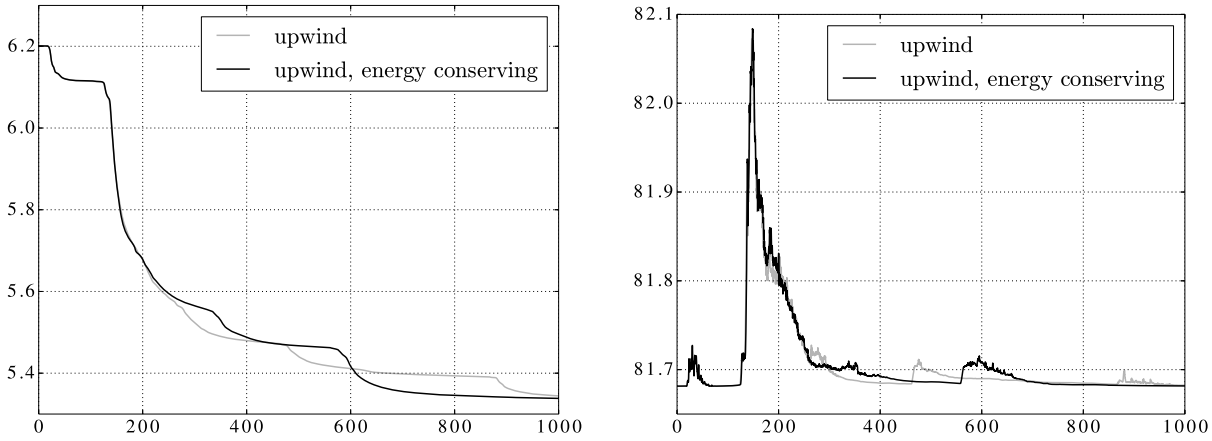


Fig. 7.9. Two-stream instability test case: time evolution of the L^2 norm (left) and L^1 norm (right) of the distribution function computed with the UM and the UECM.

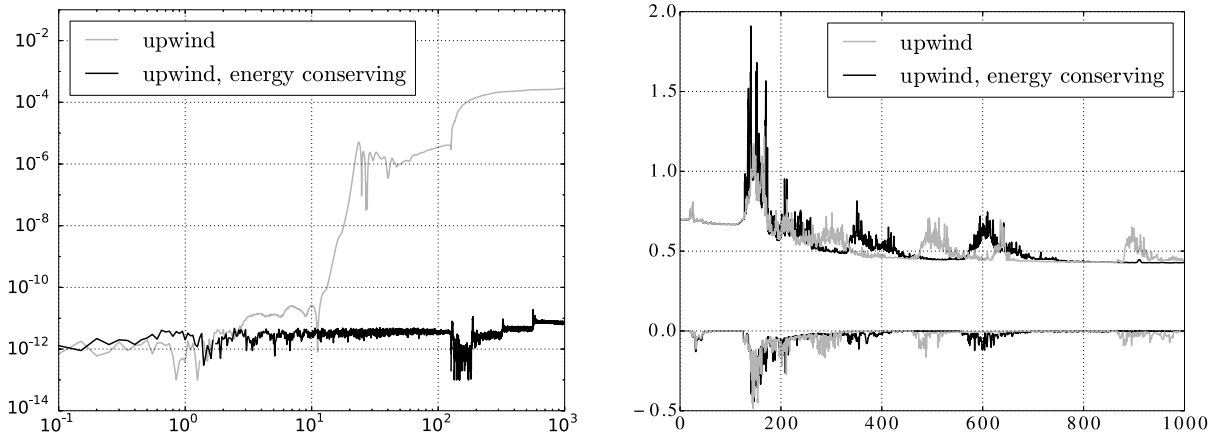


Fig. 7.10. Two-stream instability test case: total energy deviation $|\mathcal{E}_w(t) - \mathcal{E}_w(0)|$ (left, a logarithmic scale is used also for the time axis) and time evolution of $\max(f_h)$ and $\min(f_h)$ (right) for the UP and UECM.

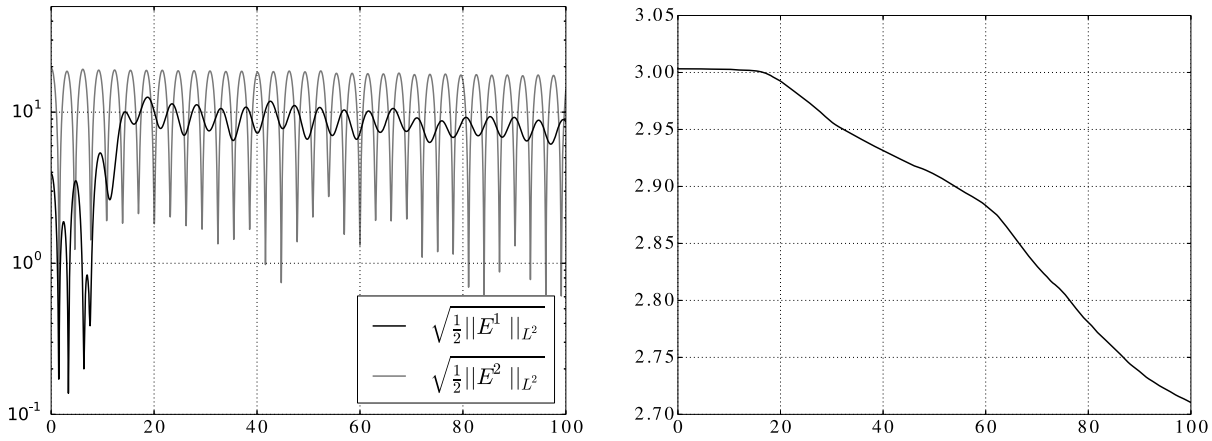


Fig. 7.11. Two-dimensional two-stream instability: time evolution of the electrostatic energy, separating the contributions of the electric field in the x^1 and x^2 directions (left) and time evolution of $\|f_h\|_{L^2}$ (right).

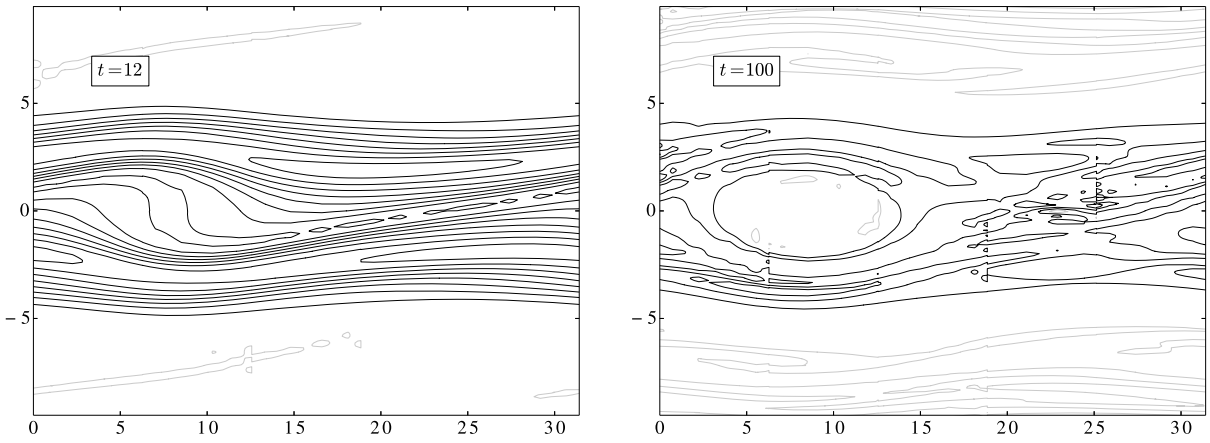


Fig. 7.12. Two-dimensional two-stream instability: contour plot of the distribution function on the plane $x^2 = 0$, $v^2 = 0$ for time levels $t = 12$ and $t = 100$. The zero contour line is shown in gray.

Table 7.3
L^2 error norms and efficiency index (6.6) for the ERB2 method with Leja point interpolation, for varying accuracy in the computation of the exponential matrix; the problem considered here is the "bump-on-tail" instability, using the upwind, energy conserving method with the same setup of Section 7.2.

Δt^{ERB2}	$\varepsilon_{\text{Lej}} = 10^{-3}$		$\varepsilon_{\text{Lej}} = 10^{-2}$		$\varepsilon_{\text{Lej}} = 10^{-1}$	
	$\|f_{\text{ref}} - f_h\|_{L^2}$	η_{eff}	$\|f_{\text{ref}} - f_h\|_{L^2}$	η_{eff}	$\|f_{\text{ref}} - f_h\|_{L^2}$	η_{eff}
$7.5 \cdot 10^{-3}$	$4.6 \cdot 10^{-5}$	0.15	$4.6 \cdot 10^{-5}$	0.17	$5.3 \cdot 10^{-5}$	0.20
$1.5 \cdot 10^{-2}$	$1.8 \cdot 10^{-4}$	0.26	$1.9 \cdot 10^{-4}$	0.32	$5.9 \cdot 10^{-4}$	0.40
$3.0 \cdot 10^{-2}$	$7.4 \cdot 10^{-4}$	0.49	$7.7 \cdot 10^{-4}$	0.58	$1.2 \cdot 10^{-2}$	0.80
$6.0 \cdot 10^{-2}$	$3.0 \cdot 10^{-3}$	0.81	$3.5 \cdot 10^{-3}$	1.02	$2.3 \cdot 10^{-1}$	1.60
$1.2 \cdot 10^{-1}$	$1.3 \cdot 10^{-2}$	1.24	$1.2 \cdot 10^{-2}$	1.64	$9.5 \cdot 10^{-2}$	1.95
$2.4 \cdot 10^{-1}$	$7.9 \cdot 10^{-2}$	1.57	$7.9 \cdot 10^{-2}$	1.58	$1.0 \cdot 10^{-1}$	1.60
$4.8 \cdot 10^{-1}$	$8.8 \cdot 10^{-1}$	1.44	$8.8 \cdot 10^{-1}$	1.45	1.2	1.47

8. Conclusion and future developments

In this work, a spectral element DG formulation for the Vlasov–Poisson equation has been discussed, with particular attention to its discrete conservation properties. The scheme conserves naturally the number of particle, and a suitable reconstruction of the electric field has been discussed resulting in exact energy conservation (up to errors introduced by the time discretization). The accuracy and stability of the resulting method have been investigated using some classical test cases.

Concerning the time discretization of the problem, two methods have been considered: a classical RK scheme and a second order exponential time integrator. This latter method can be competitive in terms of computational efficiency when a moderate accuracy in the distribution function can be accepted.

The most natural continuation of this work concerns the use of higher order exponential integrators, which could allow large time steps without sacrificing the accuracy of the numerical solution.

Other possibilities that we plan to explore are the use of limiters for the control of the spurious oscillations, for instance along the lines of [54], and the use of adaptive mesh refinement.

Acknowledgements

Many ideas addressed in this work are the result of various discussions with Blanca Ayuso de Dios and Soheil Hajian.

We would also like to thank Luca Bonaventura, Daniel Y. Le Roux, Omar Maj, Michael Kraus and Philip J. Morrison for many interesting comments and suggestions.

The first author (Éric Madaule) worked at this paper during his Master Degree at the ENSEIRB-Matmeca engineering school in Bordeaux.

Appendix A. Dispersion analysis of the DG method

The target of the dispersion analysis is to assess to which extent the linear behaviour of the discretized system resembles that of the continuous one. While doing this, one also obtains some information about the spectral properties of the Jacobian matrix $\mathcal{J}_{\mathbf{R}}^n$ in (6.3), which significantly affect the stability of the time discretization for both RK4 and ERB4 and, for ERB2, also the convergence of the iterative computation of $\phi(\Delta t\, \mathcal{J}_{\mathbf{R}}^n)$, as discussed for instance in [25,33,50]. The dispersion analysis for DG schemes presents some additional difficulties compared to the standard one for finite difference

methods due to the fact that, even on a uniform grid, not all the degrees of freedom are equivalent; this is the motivation for including the present section.

Rather than analyzing the complete Vlasov–Poisson problem, we focus here on the one-dimensional advection equation, which, albeit being much simpler, still represents an important feature of the model. Moreover, we only consider linear elements; much more general results can be found in [38,39].

Let us thus consider the problem

$$\partial_t u + a\partial_x u = 0 \tag{A.1}$$

for constant $a > 0$. Introducing a uniform grid in space with elements of size h and discretizing (A.1) with piecewise linear polynomials and upwind numerical fluxes, one arrives at the ODE system

$$\begin{aligned} &\frac{h}{3}\dot{u}_i^L + \frac{h}{6}\dot{u}_i^R + \frac{a}{2}\left(u_i^L + u_i^R\right) - a u_{i-1}^R = 0 \\ &\frac{h}{6}\dot{u}_i^L + \frac{h}{3}\dot{u}_i^R - \frac{a}{2}\left(u_i^L + u_i^R\right) + a u_i^R = 0, \end{aligned} \tag{A.2}$$

where u_i^L and u_i^R are the degrees of freedom collocated at the leftmost and rightmost point of the i-th element, respectively. Such an equation can be written in matrix notation as

$$\mathcal{M}\dot{\mathbf{u}} + \mathcal{A}\mathbf{u} = 0. \tag{A.3}$$

In the dispersion analysis, we look for solutions of (A.3) of the form $\mathbf{u}(t) = \hat{\mathbf{u}}\mathrm{e}^{I\omega t}$, where $\hat{\mathbf{u}}$ is an arbitrary but fixed vector, $\omega \in \mathbb{C}$ and $I = \sqrt{-1}$. Notice that, once such a solution has been found, it satisfies

$$I\omega\mathcal{M}\hat{\mathbf{u}} + \mathcal{A}\hat{\mathbf{u}} = 0$$

and, observing that (A.3) can be recast in form (6.3) upon multiplying by $\mathcal{M}^{-1}$ and setting $\mathcal{J}_{\mathbf{R}} = -\mathcal{M}^{-1}\mathcal{A}$, we see that $\lambda = I\omega$ is an eigenvalue of $\mathcal{J}_{\mathbf{R}}$ and $\hat{\mathbf{u}}$ is the corresponding eigenvector; this provides the link with the study of the spectral properties of $\mathcal{J}_{\mathbf{R}}$. As anticipated, the difficulty in performing the dispersion analysis of (A.3) is the presence of *two* kinds of degrees of freedom, the "L" values and the "R" values, which are not equivalent because they solve two different equations, namely $(A.2)_1$ and $(A.2)_2$. We proceed by making the following ansatz:

$$\begin{bmatrix} u_i^L(t) \\ u_i^R(t) \end{bmatrix} = \begin{bmatrix} \hat{u}^L \\ \hat{u}^R \end{bmatrix} \mathrm{e}^{I(\omega t - \kappa h i)}, \tag{A.4}$$

where $\hat{u}^L$ and $\hat{u}^R$ are two scalar constants which do not depend on t or i. Substituting (A.4) into (A.3) we arrive at the condition

$$\begin{bmatrix} 2I\tilde{\omega} + 3 & I\tilde{\omega} + 3(1 - 2\Psi) \\ I\tilde{\omega} - 3 & 2I\tilde{\omega} + 3 \end{bmatrix} \begin{bmatrix} \hat{u}^L \\ \hat{u}^R \end{bmatrix} = 0, \tag{A.5}$$

where $\tilde{\omega} = \frac{h\omega}{a}$ and $\Psi = \mathrm{e}^{I\kappa h}$. Eq. (A.5) is a 2×2 generalized eigenvalue problem, providing the dispersion relation of the discrete system. To complete the dispersion analysis, the modes of the discrete system must be compared with those of the continuous one (A.1), which are easily seen to be

$$\mathrm{e}^{I(\omega t - \kappa x)}, \quad \omega = a\kappa. \tag{A.6}$$

The difficulty now arises because, while (A.6) provides one frequency ω for each wave number κ, Eq. (A.5) admits two solutions for each κ. To deal with this aspect, we need to take into account that the space variation, and thus the wave number, of a discrete mode depends on both κ *and* the two values $\hat{u}^L, \hat{u}^R$. This can be done as follows. For a generic value $\kappa \in [-\frac{\pi}{h}, \frac{\pi}{h}]$, let us indicate by $\omega_{\kappa,j}$ the two solutions of (A.5) and by $\hat{u}^L_{\kappa,j}, \hat{u}^R_{\kappa,j}$ the components of the corresponding eigenvectors, for $j = 1, 2$. We can introduce $\varrho_{\kappa,j} > 0$ and $\vartheta_{\kappa,j} \in [-\pi, \pi)$ such that

$$\hat{u}^R_{\kappa,j} = \varrho_{\kappa,j}\mathrm{e}^{-I\vartheta_{\kappa,j}}\hat{u}^L_{\kappa,j},$$

so that from (A.4) the corresponding mode is (up to a multiplicative constant)

$$u_i^L(t;\kappa,j) = \mathrm{e}^{I(\omega_{\kappa,j}t - \kappa h i)}, \qquad u_i^R(t;\kappa,j) = \varrho_{\kappa,j}\mathrm{e}^{I(\omega_{\kappa,j}t - \kappa h i - \vartheta_{\kappa,j})}. \tag{A.7}$$

The "L" component of (A.7) is consistent with the sampling at $x_i = hi$ of the continuous mode

$$\mathrm{e}^{I(\omega_{\kappa,j}t - (\kappa + \frac{2\pi}{h}p)x)}, \tag{A.8}$$

for an arbitrary integer p. We now choose p in order to maximize the agreement with the "R" component of (A.7), thereby completely identifying the continuous mode which should be compared with the discrete one. To this end, we define p according to the following condition: minimize $|\kappa + \frac{2\pi}{h}p|$ among the values p such that

$$|\kappa h + 2\pi p| \geq |\vartheta_{\kappa,j}|, \qquad \mathrm{sign}(\kappa h + 2\pi p) = \mathrm{sign}(\vartheta_{\kappa,j}). \tag{A.9}$$

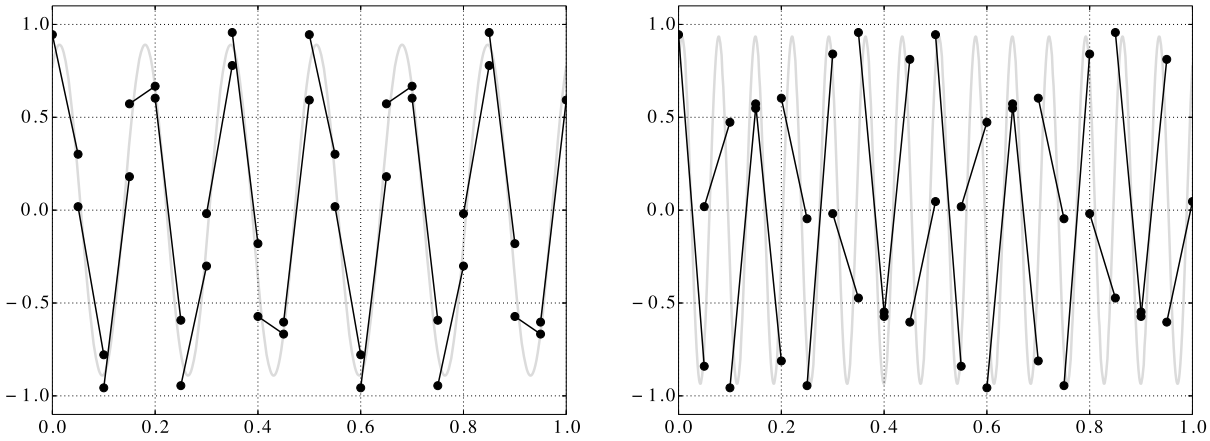


Fig. A.1. DG discretization of the advection equation with linear elements, plots of the two modes associated with $\kappa h = 1.885$; discrete modes are represented in black while the corresponding continuous ones in gray. Results for $\tilde{\omega}_1 = 1.930 + 0.142I$ (left) and $\tilde{\omega}_2 = -3.832 + 3.240I$ (right).

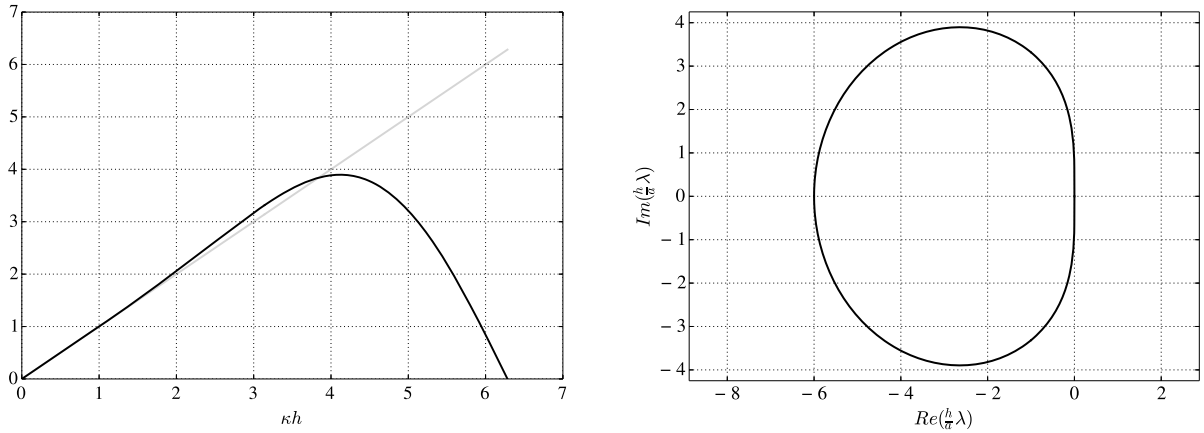


Fig. A.2. DG discretization of the advection equation with linear elements: dispersion relation $Re(\omega(\kappa))$, left, and values $\frac{h}{a}\lambda$ for λ eigenvalue of $\mathcal{J}_{\mathbf{R}} = -\mathcal{M}^{-1}\mathcal{A}$ as resulting from (A.3), right.

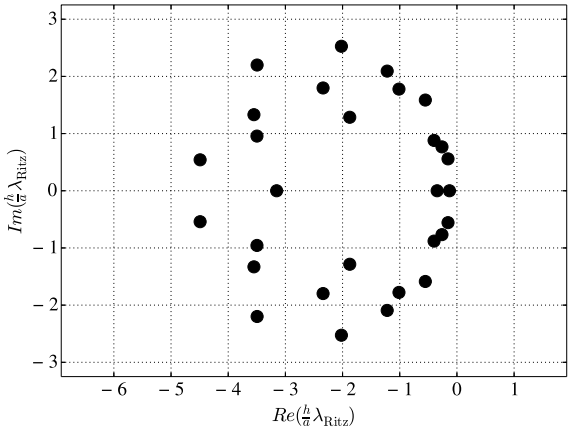


Fig. A.3. Ritz values for $\mathcal{J}^n_{\mathbf{R}}$ for the "bump-on-tail" test case. The values are scaled by $\frac{h}{a}$ with $h = \frac{20\pi}{18\cdot 6}$, $a = 9$, representing an equivalent element size and the maximum velocity, to make them comparable with Fig. A.2, left.

An example of this procedure is shown in Fig. A.1. Here, we fix $\kappa h = 1.885$ in (A.5), obtaining $\tilde{\omega}_1 = 1.930 + 0.142I$, $\tilde{\omega}_2 = -3.832 + 3.240I$ and $\vartheta_1 = 1.508$, $\vartheta_2 = -2.548$; according to (A.9), these values yield $p = 0$ and $p = -1$, respectively, and the resulting discrete and continuous modes are shown. The dispersion relation $Re(\omega(\kappa))$ is shown in Fig. A.2, which is similar to the dispersion equation obtained for standard second order finite differences formulations; the same figure also shows the complete set of eigenvalues λ of $\mathcal{J}_{\mathbf{R}} = -\mathcal{M}^{-1}\mathcal{A}$ according to (A.3).

To provide an indication concerning the relevance of this analysis for the complete Vlasov–Poisson system, we plot in Fig. A.3 the Ritz values of $\mathcal{J}^n_{\mathbf{R}}$ for the "bump-on-tail" instability problem with the same setup used in Section 7.2. As explained in [53], these values are easily obtained as a byproduct of the computation of $\phi(\Delta t\, \mathcal{J}^n_{\mathbf{R}})$ and are approximations

of the eigenvalues of $\mathcal{J}^n_{\mathbf{R}}$. Notice the similarity in the pattern with Fig. A.2, right; similar results can be also observed in [26], Fig. 3.

References

[1] Douglas N. Arnold, Franco Brezzi, Mixed and nonconforming finite-element methods — implementation, postprocessing and error-estimates, RAIRO. Modél. Math. Anal. Numér. 19 (1) (1985) 7–32.
[2] Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, L. Donatella Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (5) (January 2002) 1749–1779.
[3] Blanca Ayuso de Dios, José A. Carrillo, Chi-Wang Shu, Discontinuous Galerkin methods for the one-dimensional Vlasov–Poisson system, Kinet. Relat. Models 4 (4) (December 2011) 955–989.
[4] Blanca Ayuso de Dios, José A. Carrillo, Chi-Wang Shu, Discontinuous Galerkin methods for the multi-dimensional Vlasov–Poisson problem, Math. Models Methods Appl. Sci. 22 (12) (June 2012) 1250042.
[5] Blanca Ayuso de Dios, Hajian Soheil, High order and energy preserving discontinuous Galerkin methods for the Vlasov–Poisson system, Technical report, arXiv:1209.4025, 2012.
[6] John P. Boyd, Chebyshev and Fourier Spectral Methods, Courier Dover Publications, 2001.
[7] Sébastien Blaise, Amik St-Cyr, A dynamic *hp*-adaptive discontinuous Galerkin method for shallow-water flows on the sphere with application to a global tsunami simulation, Mon. Weather Rev. 140 (3) (September 2011) 978–996.
[8] Paul Castillo, Bernardo Cockburn, Ilaria Perugia, Dominik Schötzau, An a priori error analysis of the local discontinuous Galerkin method for elliptic problems, SIAM J. Numer. Anal. 38 (5) (January 2001) 1676–1706.
[9] Yingda Cheng, Andrew J. Christlieb, Xinghui Zhong, Energy-conserving discontinuous Galerkin methods for the Vlasov–Ampère system, J. Comput. Phys. 256 (January 2014) 630–655.
[10] Yingda Cheng, Andrew J. Christlieb, Xinghui Zhong, Energy-conserving discontinuous Galerkin methods for the Vlasov–Maxwell system, J. Comput. Phys. 279 (2014) 145–173, http://dx.doi.org/10.1016/j.jcp.2014.08.041.
[11] Bernardo Cockburn, Jayadeep Gopalakrishnan, Raytcho Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal. 47 (2) (2009) 1319–1365.
[12] Yingda D. Cheng, Irene M. Gamba, Philip J. Morrison, Study of conservation and recurrence of Runge–Kutta discontinuous Galerkin schemes for Vlasov–Poisson systems, J. Sci. Comput. 56 (2) (August 2013) 319–349.
[13] Andrew Christlieb, Wei Guo, Maureen Morton, Jing-Mei Qiu, A high order time splitting method based on integral deferred correction for semi-Lagrangian Vlasov simulations, J. Comput. Phys. 267 (June 2014) 7–27.
[14] Claudio Canuto, Yousuff M. Hussaini, Alfio Quarteroni, Thomas Zang, Spectral Methods. Fundamentals in Single Domains, Springer-Verlag, Berlin, 2006.
[15] Claudio Canuto, Yousuff M. Hussaini, Alfio Quarteroni, Thomas Zang, Spectral Methods. Evolution to Complex Geometries and Applications to Fluid Dynamics, Springer-Verlag, Berlin, 2007.
[16] Bernardo Cockburn, Suchung Hou, Chi-Wang Shu, The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV. The multidimensional case, Math. Comput. 54 (1990) 545–581.
[17] C.Z. Cheng, Georg Knorr, The integration of the Vlasov equation in configuration space, J. Comput. Phys. 22 (3) (November 1976) 330–351.
[18] Bernardo Cockburn, San-Yih Lin, Chi-Wang Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84 (1) (September 1989) 90–113.
[19] Nicolas Crouseilles, Michel Mehrenberger, Eric Sonnendrücker, Conservative semi-Lagrangian schemes for Vlasov equations, J. Comput. Phys. 229 (6) (March 2010) 1927–1953.
[20] Marco Caliari, Alexander Ostermann, Implementation of exponential Rosenbrock-type integrators, Appl. Numer. Math. 59 (3–4) (March 2009) 568–581.
[21] Bernardo Cockburn, Chi-Wang Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput. 52 (1989) 411–435.
[22] Bernardo Cockburn, Chi-Wang Shu, The Runge–Kutta local projection P1-discontinuous-Galerkin finite element method for scalar conservation laws, RAIRO. Modél. Math. Anal. Numér. 25 (3) (1991) 337–361.
[23] Bernardo Cockburn, Chi-Wang Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM J. Numer. Anal. 35 (6) (November 1998) 2440–2463.
[24] Bernardo Cockburn, Chi-Wang Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (2) (April 1998) 199–224.
[25] Marco Caliari, Marco Vianello, Luca Bergamaschi, Interpolating discrete advection–diffusion propagators at Leja sequences, J. Comput. Appl. Math. 172 (1) (November 2004) 79–99.
[26] Dannert Tilman, Frank Jenko, Vlasov simulation of kinetic shear Alfvén waves, Comput. Phys. Commun. 163 (2) (2004) 67–78.
[27] Francis Filbet, Eric Sonnendrücker, Pierre Bertrand, Conservative numerical schemes for the Vlasov equation, J. Comput. Phys. 172 (1) (September 2001) 166–187.
[28] Ferran Garcia, Luca Bonaventura, Marta Net, Juan Sánchez, Exponential versus IMEX high-order time integrators for thermal convection in rotating spherical shells, J. Comput. Phys. 264 (May 2014) 41–54.
[29] Wei Guo, Jing-Mei Qiu, Hybrid semi-Lagrangian finite element-finite difference methods for the Vlasov equation, J. Comput. Phys. 234 (February 2013) 108–132.
[30] Francis X. Giraldo, Marco Restelli, A study of spectral element and discontinuous Galerkin methods for the Navier–Stokes equations in nonhydrostatic mesoscale atmospheric modeling: equation sets and test cases, J. Comput. Phys. 227 (8) (April 2008) 3849–3877.
[31] Gene H. Golub, Charles F. van Loan, Matrix Computations, JHU Press, 2012.
[32] R.E. Heath, Irene M. Gamba, Philip J. Morrison, C. Michler, A discontinuous Galerkin method for the Vlasov–Poisson system, J. Comput. Phys. 231 (4) (February 2012) 1140–1174.
[33] Marlis Hochbruck, Christian Lubich, Hubert Selhofer, Exponential integrators for large systems of differential equations, SIAM J. Sci. Comput. 19 (5) (1998) 1552–1574.
[34] Claes Johnson, Juhani Pitkäranta, An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation, Math. Comput. 46 (173) (January 1986) 1–26.
[35] David A. Kopriva, Implementing Spectral Methods for Partial Differential Equations, Springer, 2009.
[36] Michael Kraus, Variational integrators in plasma physics, Technical report, Max-Planck-Institut für Plasmaphysik, 2013.
[37] J. Douglas Lawson, Generalized Runge–Kutta processes for stable systems with large Lipschitz constants, SIAM J. Numer. Anal. 4 (3) (1967) 72–380.
[38] Daniel Y. Le Roux, G.F. Carey, Stability/dispersion analysis of the discontinuous Galerkin linearized shallow-water system, Int. J. Numer. Methods Fluids 48 (3) (2005) 325–347.
[39] Daniel Y. Le Roux, Fourier analyses of high order continuous and discontinuous Galerkin methods, Imperial College, 2013.

[40] Andreas Müller, Jörn Behrens, Francis X. Giraldo, Volkmar Wirth, Comparison between adaptive and uniform discontinuous Galerkin simulations in dry 2D bubble experiments, J. Comput. Phys. 235 (February 2013) 371–393.
[41] Michel Mehrenberger, Christophe Steiner, Luca Marradi, Nicolas Crouseilles, Eric Sonnendrücker, Bedros Afeyan, Vlasov on GPU (VOG project), ESAIM Proc. 43 (December 2013) 37–58.
[42] Cleve Moler, Charles van Loan, Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later, SIAM Rev. 45 (1) (2006) 3–49.
[43] Takashi Nakamura, Takashi Yabe, Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov–Poisson equation in phase space, Comput. Phys. Commun. 120 (2–3) (August 1999) 122–154.
[44] Jing-Mei Qiu, Chi-Wang Shu, Positivity preserving semi-Lagrangian discontinuous Galerkin formulation: theoretical analysis and application to the Vlasov–Poisson system, J. Comput. Phys. 230 (23) (September 2011) 8386–8409.
[45] Marco Restelli, Luca Bonaventura, Riccardo Sacco, A semi-Lagrangian discontinuous Galerkin method for scalar advection by incompressible flows, J. Comput. Phys. 216 (1) (July 2006) 195–215.
[46] Marco Restelli, Francis X. Giraldo, A conservative discontinuous Galerkin semi-implicit formulation for the Navier–Stokes equations in nonhydrostatic mesoscale modeling, SIAM J. Sci. Comput. 31 (3) (January 2009) 2231–2257.
[47] Wm.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, October 1973.
[48] Thomas Respaud, Eric Sonnendrücker, Analysis of a new class of forward semi-Lagrangian schemes for the 1D Vlasov–Poisson equations, Numer. Math. 118 (2) (2011) 329–366.
[49] Pierre-Arnaud Raviart, J.M. Thomas, A mixed finite element method FPR 2-nd order elliptic problems, in: Mathematical Aspects of Finite Element Methods, in: Lecture Notes in Mathematics Series, vol. 666, 1977.
[50] Yousef Saad, Analysis of some Krylov subspace approximations to the matrix exponential operator, SIAM J. Numer. Anal. 29 (1) (1991) 209–228.
[51] Magdi M. Shoucri, Nonlinear evolution of the bump-on-tail instability, Phys. Fluids (1958–1988) 22 (10) (1979) 2038–2039.
[52] Eric Sonnendrücker, Jean Roche, Pierre Bertrand, Alain Ghizzo, The semi-Lagrangian method for the numerical resolution of the Vlasov equation, J. Comput. Phys. 149 (2) (1999) 201–220.
[53] Jan C. Schulze, Peter J. Schmid, Jörn L. Sesterhenn, Exponential time integration using Krylov subspaces, Int. J. Numer. Methods Fluids 60 (6) (2009) 591–609.
[54] Xiangxiong Zhang, Chi-Wang Shu, On maximum-principle-satisfying high order schemes for scalar conservation laws, J. Comput. Phys. 229 (9) (May 2010) 3091–3120.

Résumé Le système d'équations de Vlasov-Poisson est un système très connu (mais pas totalement compris) de la physique des plasmas, un enjeu majeur des futures simulations de plasmas à grande échelle et l'objet de nombreuses expérimentations numériques. Ce système est également utilisé en astrophysique pour décrire des systèmes auto-gravitants sans collision. Le but est de développer des schémas numériques utilisant une discrétisation par la méthode Galerkin discontinue combinée avec une résolution en temps semi-Lagrangienne et un maillage adaptatif basé sur l'utilisation des multi-ondelettes. La formulation Galerkin discontinue autorise des schémas d'ordres élevés avec des données locales. Cette formulation a récemment fait l'objet de nombreuses publications par Ayuso de Dios, Carrillo de la Plata et Shu, Rossmanith et Seal, etc. Leurs travaux sont présentés avec des schémas eulériens, mais Quo, Nair et Qiu, Qiu et Shu et Bokanowski et Simarta ont publié des résultats avec une résolution en temps semi-lagrangienne. On utilise les multi-ondelettes pour l'adaptativité (et plus précisément pour la décomposition multi-échelle de la fonction de distribution). Les multi-ondelettes ont été largement étudiées par Alpert *et al.* pendant les années 1990 et au début des années 2000. Des travaux combinant la résolution multi-échelle avec les méthodes Galerkin discontinues ont fait l'objet de publications par Müller et ses collègues en 2014 pour les lois de conservation hyperboliques dans le contexte des éléments finis. Besse, Latu, Ghizzo, Sonnendrücker et Bertrand ont présenté les avantages d'un maillage adaptatif dans le contexte de Vlasov-Poisson relativiste en utilisant des ondelettes à support large. La combinaison de la méthode Galerkin discontinue avec l'utilisation des multi-ondelettes ne requière en revanche qu'un support compact. Ceci favorise la parallélisation, bien qu'actuellement, les méthodes semi-lagrangiennes restent un obstacle à la mise en œuvre de schémas efficaces lors de l'utilisation de mémoire distribuée. Le but ici est d'associer les trois aspects afin d'obtenir un schéma capable de suivre les détails de la distribution avec une implémentation indépendante de l'ordre spatial, et prenant avantage de la formulation semi-lagrangienne pour ignorer la condition de CFL. Bien que la majorité de la thèse soit présentée dans un espace des phases `1d×1v`, nous avons obtenus quelques résultats dans l'espace des phases `2d×2v`.

Mots-clés: méthodes numériques, Vlasov-Poisson, schémas adaptatifs

Abstract Many numerical experiments are performed on the Vlasov-Poisson problem since it is a well known (but not fully understood) system from plasma physics and a major issue for future simulation of large scale plasmas. These equations are also used in astrophysics to describe self gravitating collisionless system. Our goal is to develop adaptive numerical schemes using discontinuous Galerkin discretisation combined with semi-Lagrangian description whose mesh refinement based on multi-wavelets. The discontinuous Galerkin formulation enables high-order accuracy with local data for computation. It has recently been widely studied by Ayuso de Dios, Carrillo de la Plata et Shu, Rossmanith et Seal, etc. Their work is done with Eulerian description of the time resolution but Guo, Nair and Qiu or Qiu and Shu or Bokanowski and Simarta performed semi-Lagrangian time resolution. We use multi-wavelets framework for the adaptive part (more precisely, for the multi-scale decomposition of the distribution function of particles). Those have been heavily studied by Alpert *et al.* during the nineties and the two thousands. Some works merging multi-scale resolution and discontinuous Galerkin methods have been described by Müller and his colleagues in 2014 for non-linear hyperbolic conservation laws in the finite volume framework. In the framework of relativistic Vlasov equation, Besse, Latu, Ghizzo, Sonnendrücker and Bertrand presented the advantage of using adaptive meshes. While they used wavelet decomposition, which requires large data stencil, multi-wavelet decomposition coupled to discontinuous Galerkin discretisation only requires local stencil. This favours the parallelisation but, at the moment, semi-Lagrangian remains an obstacle to highly efficient distributed memory parallelisation. The goal here is to combine the three aspects in order to produce a scheme able to follow details of the distribution function, with an implementation independent of spatial reconstruction order, and taking advantage of the semi-Lagrangian formulation to ignore CFL condition. Although most of our work is done in a `1d×1v` phase space, we were able to obtain a few results in a `2d×2v` phase space.

Keywords: numérical méthods, Vlasov-Poisson, adaptive schemes