

Étude et programmation d'un modèle particulier pour une Équation de Vlasov-Poisson 1D

Fanny Buffet Delmas, Julia De Biasi, Elie Nakache

2ème semestre 2011

Résumé

L'équation de Vlasov-Poisson, un modèle très étudié en physique des plasmas et en astrophysique. Nous présenterons dans ce rapport notre travail sur l'étude et la programmation d'un modèle particulier pour cette équation. Nous nous limiterons au cas moins réaliste mais plus accessible de la dimension 1.

Après avoir décrit l'équation, nous expliquerons la méthode de résolution dans le cas linéaire et donnerons les principales propriétés de ses solutions.

Dans une deuxième partie, nous présenterons les méthodes numériques utilisées pour l'approximation des solutions de cette équation, en étudiant partiellement l'ordre de convergence des schémas.

Enfin, nous exposerons les résultats numériques obtenus avec notre code, en vérifiant la validité de nos calculs, et en interprétant les résultats les plus intéressants.

Table des matières

1	Introduction aux équations de Vlasov	3
1.1	Description de l'équation de Vlasov	3
1.2	La force F	3
1.3	D'où vient cette équation ?	5
1.4	La méthode des caractéristiques	6
1.4.1	Présentation de la méthode	6
1.4.2	Le flot	7
1.4.3	Résolution	10
1.5	Propriétés qualitatives de l'équation	11
1.5.1	Réversibilité	11
1.5.2	Préservation du volume	12
1.5.3	Conservation de la masse, du moment et de l'énergie	13
2	La méthode d'approximation numérique	15
2.1	Les différentes méthodes numériques	15
2.1.1	Approximation numérique et erreurs	15
2.1.2	Méthode d'Euler explicite	16
2.1.3	Méthode du point milieu	18
2.1.4	Méthode de Saute mouton.	19
2.2	Équations et problème.	19
2.3	Le programme.	20
2.3.1	Première étape : la résolution générale avec la force $F(t, x)$ donnée.	20
2.3.2	Deuxième étape : la force dépend de la position.	21
3	Résultats numériques et interprétations	22
3.1	Vérification du code	22
3.1.1	Utilisation de la réversibilité	22
3.1.2	Comparaison pour des intervalles de temps distincts	26
3.1.3	Conservation du moment et de l'énergie	27
3.2	Cas linéaire	28
3.2.1	La rotation	28
3.2.2	Filamentation	29
3.3	Cas non linéaire	31
3.3.1	Cas répulsif : l'exemple des plasmas	31
3.3.2	Cas attractif : l'exemple des galaxies	32
4	Code	33
4.1	Le cas linéaire	33
4.2	Le cas non linéaire	34
5	Bibliographie	37

1 Introduction aux équations de Vlasov

1.1 Description de l'équation de Vlasov

Dans ce TER, nous étudions un modèle cinétique particulier : **l'équation de Vlasov**. On utilise des modèles cinétiques dans les milieux peu denses lorsque deux ou plusieurs particules peuvent être au même endroit, avec des vitesses différentes. C'est le cas par exemple dans les plasmas, les galaxies en astro-physique où les particules sont alors des étoiles. Il faut tenir compte du fait que la position et la vitesse font alors partie des variables du système.

L'équation de Vlasov dans \mathbb{R}^N s'écrit :

$$\partial_t f + v \cdot \nabla_x f + F \cdot \nabla_v f = 0 \quad (1)$$

avec la donnée initiale : $f(0, x, v) = f_0(x, v)$.

Nous retrouvons les trois variables caractéristiques des problèmes cinétiques : t le temps, x la position et v la vitesse.

- **Le temps** est clairement défini sur \mathbb{R} , ou bien \mathbb{R}^+ .
- **La position** x , peut être définie sur \mathbb{R} , \mathbb{R}^d , un domaine périodique ou bien un ouvert borné de \mathbb{R}^d . Dans le dernier cas, nous sommes confrontés à une difficulté : il faut se donner des conditions aux bords, ce qui est un problème assez délicat. On a donc choisit de se placer sur un domaine périodique. Étant donné que se positionner dans \mathbb{R}^d n'est pas possible, en plus de faciliter les simulations numériques, ce choix se justifie également par le fait que l'on retrouve souvent cette configuration dans le domaine nucléaire, où les plasmas sont dans des géométries toriques, donc dans des domaines qui sont périodiques en position.
- **La vitesse** v , on se place dans \mathbb{R}^d car les vitesses n'ont pas de raisons d'être bornées.

Ce modèle fait aussi intervenir **la densité de particule** $f(t, x, v)$ qui dépend du temps t , de la position x et de la vitesse v . C'est-à-dire que dans un rectangle de côtés dx et dv il y a $f(t, x, v)dx dv$ éléments. On suppose que f reste positive au cours du temps et qu'elle est d'intégrale égale à 1 (on verra un peu plus loin que la masse est conservée).

Avant de nous intéresser à la force F , demandons nous pourquoi un tel modèle est utilisé. Les plasmas et les galaxies sont en fait des milieux considérés comme peu dense, en effet, ils ne contiennent que de l'ordre de 10^8 à 10^{15} particules ce qui est "petit" comparé par exemple au nombre d' Avogadro (environ $6.10^{23} \text{ mol}^{-1}$). Cela implique que plusieurs particules peuvent être presque au même endroit avec des vitesses différentes. Nous sommes bien dans un cas différent de celui des fluides où lorsque par exemple on parle de vitesse de l'eau "en un point x ", on fait référence à la vitesse de la plupart des molécules d'eau "proches de x " car la forte densité des éléments entraîne de nombreuses collisions qui forcent les molécules à avoir des comportements "collectifs".

Il nous reste donc à nous intéresser à la force F qui est un élément fondamental aussi bien pour notre équation que dans les simulations numériques développées dans la suite de notre travail.

1.2 La force F

La force F qui intervient dans notre équation peut être de nature très différente, mais ici, nous ne traiterons que les deux principaux cas.

- Soit F est donnée par un système extérieur, c'est-à-dire qu'elle ne dépend pas de f : le comportement des particules n'a aucune influence sur la force. Dans la suite nous verrons que l'on a simulé cette situation dans le programme informatique en posant $F(x) = -x - \cos(x)$

ou encore $F(x) = -x$. Nous nous plaçons ainsi dans le cas, peu fréquent, où il n'y a pas d'interaction entre les particules. Sous cette hypothèse, l'équation de Vlasov est **linéaire**.

- Le deuxième type de force auquel nous nous intéressons est le cas où **la force est donnée par les particules qui interagissent les unes avec les autres**. Dans l'exemple des plasmas, on obtient ce que l'on appelle **l'équation de Poisson** :

$$-\Delta_x \phi = \rho - \rho_e \quad \text{avec} \quad \rho(t, x) = \int f(t, x, v) dv.$$

où ϕ est le potentiel électrique et ρ la densité en position, c'est-à-dire le nombre de particules contenues dans un intervalle de position de longueur dx , quelle que soit leur vitesse. Le terme ρ_e est une constante qui caractérise le fait que dans un plasmas, il y a des ions et des électrons, qui sont de l'ordre de 10^4 fois plus légers et qui sont donc plus rapides et atteignent rapidement l'équilibre. Ils sont alors considérés comme toujours dans cet état.

On a alors les équations suivantes :

$$F = -\nabla_x \phi(t, x)$$

Remarquons pour la suite de notre exposé que dans le cas gravitationnel, l'équation de Poisson devient $\nabla_x \phi = \rho$, le signe change et il n'y a que des masses positives.

Pour plus de facilités dans notre travail, nous nous sommes penchés sur le cas de la dimension 1, où le modèle de Vlasov s'écrit :

$$\partial_t f + v \partial_x f + F \partial_v f = 0 \tag{2}$$

L'équation de Poisson devient

$$-\partial_{xx}^2 \phi = \rho - \rho_e \iff \partial_x \underbrace{(-\partial_x \phi)}_F = \rho - \rho_e \iff \partial_x F = \rho - \rho_e$$

F est ainsi définie à une constante C près, or il nous faut absolument une unique valeur pour avoir un système bien défini. Dans le cas, périodique, on a :

$$\int_{-1}^1 F dx = \phi(1) - \phi(-1) = 0,$$

c'est-à-dire que la force est aussi de moyenne nulle, ce qui permet alors de la déterminer de manière unique. Par exemple avec la méthode suivante. On a :

$$F(x) = \underbrace{\int_{-1}^x \rho - \rho_e dx}_F + C$$

d'où

$$\int_{-1}^1 F dx = \int_{-1}^1 F^* dx + 2C = 0$$

et donc

$$C = -\frac{1}{2} \int_{-1}^1 F^* dx$$

Nous allons nous intéresser à la façon dont cette équation est obtenue et ensuite voir comment elle peut être étudiée avec l'outil des caractéristiques.

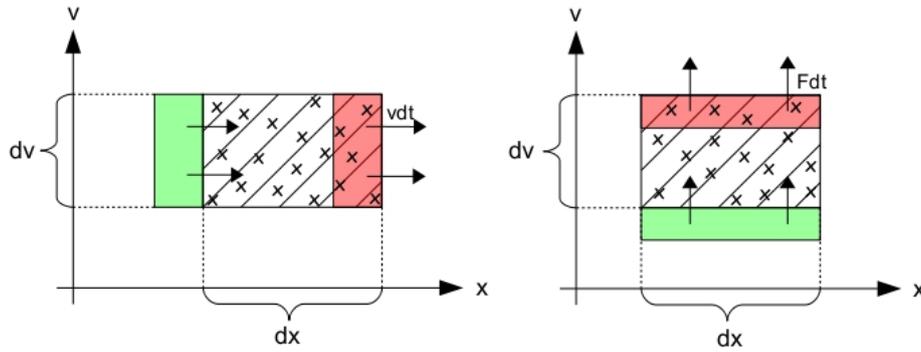
1.3 D'où vient cette équation ?

Plaçons nous dans un système discret de N particules de position X_1, \dots, X_N et de vitesse V_1, \dots, V_N qui interagissent avec un potentiel ϕ , tel que $F = -\nabla_x \phi$. Chaque élément est de masse $1/N$ de sorte que la masse totale du système est fixée à 1. Les lois de la mécanique classique nous amènent aux équations suivantes :

$$\begin{cases} \dot{X}_i = V_i \\ \dot{V}_i = F(t, X_i) \end{cases}$$

Où F est une force extérieure ou une force d'interaction. Généralement, un plasma contient au minimum $N = 10^8$ particules par unité de volume, un calcul avec un tel nombre d'équations n'est pas possible. On fait alors intervenir la densité f des particules, mais il faut justifier que cette fonction est bien définie. Le grand nombre d'éléments présents dans une unité de volume permet d'affirmer que dans chaque rectangle de côtés dx et dv on trouve un certain nombre de particules, et que si l'on déplace un peu le rectangle, il y a encore environ le même nombre de particules. D'où l'existence de f . Pour comprendre d'où provient l'équation, **plaçons nous en dimension 1 dans le rectangle de côté dx et dv** qui à l'instant t contient $f(t, x, v)dx dv$ particules et demandons nous que devient $f(t + dt, x, v)$ lorsque l'on passe de t à $t + dt$.

Dans ce rectangle, on considère que les vitesses des particules valent à peu près la même valeur v et donc les particules se déplacent de vdt pendant un intervalle de temps dt . Nous observons donc qu'il y a des particules qui entrent et d'autres qui sortent du rectangle.



x Particules.

Les particules se trouvant dans cette zone sortent du rectangle.

Les particules se trouvant dans cette zone entrent dans le rectangle.

Sur les dessins ci dessus, nous avons représenté schématiquement les mouvements des particules. Écrivons cela avec f :

$$f(t + dt, x, v) = \underbrace{-f(t, x + dx, v)vdt}_{\text{sorties}} + \underbrace{f(t, x, v)vdt}_{\text{entrées}} - \underbrace{f(t, x, v + dv)F(x)dt}_{\text{sorties}} + \underbrace{f(t, x, v)F(x)dt}_{\text{entrées}}$$

Ce qui donne :

$$[f(t + dt, x, v) - f(t, x, v)]dx dv = [f(t, x, v) - f(t, x + dx, v)]vdt + [f(t, x, v) - f(t, x, v + dv)]F(x)dt$$

Et donc en écrivant $f(t + dt, x, v) - f(t, x, v) \simeq f(t, x, v)dt$
On obtient :

$$\partial_t f(t, x, v) dt dx dv = -v \partial_x f(t, x, v) dt dx dv - F(x) \partial_v f(t, x, v) dt dx dv$$

D'où en divisant par $dt dx dv$, l'équation de Vlasov-Poisson :

$$\partial_t f + v \cdot \partial_x f + F \cdot \partial_v f = 0$$

1.4 La méthode des caractéristiques

1.4.1 Présentation de la méthode

Nous allons maintenant présenter un outil standard d'analyse des équations aux dérivées partielles, la méthode des caractéristiques. Plaçons nous tout d'abord dans le cas général suivant pour ensuite l'appliquer à Vlasov linéaire qui est un cas particulier de ce type d'équation. Considérons le problème :

$$\partial_t u + a(t, x) \partial_x u = 0, \quad 0 \leq t \leq T, \quad x \in \mathbb{R}^n \quad (3)$$

Ce type d'équation est appelé **équation de transport linéaire** et dans cette partie on supposera que $a :]0, T[\times \mathbb{R}^N \rightarrow \mathbb{R}^N$ est suffisamment régulière, par exemple $a \in C^1([0, T] \times \mathbb{R}^N)$.

Définition 1 (Courbe caractéristique)

On appelle *courbe caractéristique associée à la fonction a de (2)* une solution $s \rightarrow x(s)$ de classe C^1 définie sur un intervalle de \mathbb{R} à valeurs dans \mathbb{R}^N et telle que :

$$\frac{dx}{ds}(s) = a(s, x(s))$$

La trajectoire est alors l'ensemble $\{x(t), t \in \mathbb{R}\}$.

Dans la suite, nous allons faire référence au théorème de Cauchy Lipschitz, qui permet de montrer l'existence et l'unicité d'une solution pour les équations différentielles ordinaires du type :

$$Y'(t) = a(t, y) \quad (4)$$

Nous rappelons son énoncé :

Théorème 1

Soit $a : I \times O \rightarrow \mathbb{R}^N$ un champ de vecteurs localement Lipschitzien en espace, O un ouvert de \mathbb{R}^N . Soit t^0 un temps dans I et x^0 une position dans O . Alors il existe une unique solution maximale définie sur un intervalle $J \subset I$ à l'équation (3) vérifiant la condition $x(t^0) = x^0$.

Précisons ce que l'on entend par un **champ a localement Lipschitzien** : pour tout compact K de \mathbb{R}^N , il existe une constante L_K telle que :

$$\forall x, y \in K, |a(x) - a(y)| \leq L_K |x - y|.$$

1.4.2 Le flot

Théorème 2 (Définition du flot)

Soit $a \in C^2([0, T] \times \mathbb{R}^N)$ différentiable en x avec $\partial_x a \in C([0, T] \times \mathbb{R}^N)$ et

$$\forall t \in [0, T], \quad x \in \mathbb{R}^N, \quad |a(t, x)| \leq k(1 + |x|)$$

alors pour tout $t \in [0, T]$ et $x \in \mathbb{R}^N$, il existe une unique caractéristique définie sur $[0, T]$ et telle que $x(t) = x$. C'est le flot, on le note $x(t, s, x)$ et on a :

$$x \in C^1([0, T]_s \times [0, T]_t \times \mathbb{R}_x^N).$$

Cela définit ainsi le flot associé à l'ODE, et cela traduit le fait que si une particule située en x au temps t_0 suit le flot alors elle sera en $x(t, t_0, x_0)$ à l'instant t . Le flot est un élément important de notre travail puisqu'il permet de résoudre la même EDO pour des conditions initiales différentes et c'est ce que nous faisons avec la méthode numérique.

Pour démontrer ce théorème nous allons avoir besoin d'un lemme très souvent utilisé dans ce domaine des mathématiques, le lemme de Gronwall.

Lemme 1 (Gronwall)

Soit $a, b \in \mathbb{R}, C \in \mathbb{R}$ et z, ϕ deux fonctions continues sur $[a, b]$ à valeurs réelles. On suppose que

- ϕ est positive.
- L'inégalité suivante est vérifiée :

$$z(t) \leq C + \int_a^t \phi(s)z(s)ds, \quad \forall a \leq t < b.$$

Alors, on a l'estimation :

$$z(t) \leq C \exp\left(\int_a^t \phi(s)dt\right), \quad \forall a \leq t < b.$$

Preuve du théorème 2

Nous avons le problème de Cauchy suivant :

$$\begin{cases} \frac{d}{dt}x(t, s, x_0) = a(t, x(t, s, x_0)) \\ x(s, s, x_0) = x_0 \end{cases}$$

L'hypothèse sur $\partial_x a$ permet d'appliquer l'inégalité des accroissements finis à $a(t, x)$ qui nous donne, pour tout compact K de \mathbb{R}^N , l'existence d'une constante réelle positive L_K qui vérifie pour tout $c \in]x, y[$ l'inégalité $|\partial_x a(t, c)| \leq L_K$ et telle que pour tout $(x, y) \in K$ on ait :

$$\left| \frac{a(t, x) - a(t, y)}{x - y} \right| \leq L_K$$

Ceci prouve que a est localement Lipschitzienne par rapport à la variable x . Donc on peut appliquer le théorème de Cauchy Lipschitz : il existe une unique solution maximale définie sur un intervalle $J \subset [0, T]$.

Montrons à l'aide du lemme suivant que la solution est en fait définie sur $[0, T]$ tout entier. Soit $x(t)$ une solution définie sur J et soit $b \in J$, on a alors :

$$\begin{aligned} x(t) &= x(b) + \int_b^t x'(s)ds \\ &= x(b) + \int_b^t a(s, x(s))ds \end{aligned}$$

On obtient donc les inégalités suivantes :

$$\begin{aligned}
\|x(t)\| &\leq \|x(b)\| + \int_b^t \|a(s, x(s))\| ds \\
&\leq \|x(b)\| + k \int_b^t (1 + \|x(s)\|) ds \\
&\leq \underbrace{\|x(b)\| + k(T-b)}_{=C} + k \int_b^t \|x(s)\| ds
\end{aligned}$$

En appliquant le lemme de Gronwall on a alors l'estimation :

$$\|x(t)\| \leq C e^{k(t-b)}$$

ce qui montre que la solution ne peut pas exploser et donc qu'elle est définie sur $[0, T]$ tout entier.

Le caractère C^1 est un peu plus complexe à obtenir. On sait que le flot vérifie l'équation :

$$\frac{d}{dt}x(t, s, x) = a(t, x(t, s, x))$$

Pour simplifier, nous supposons que $s = 0$ car les calculs dans le cas général sont les mêmes, et donc pour plus de lisibilité, on effacera la dépendance en s des fonctions.

Supposons que l'on puisse différencier le flot en x , on a alors :

$$\frac{d}{dt} \left(\frac{d}{dx} x(t, x) \right) = \frac{d}{dx} a(t, x(t, x)) \cdot \frac{d}{dx} x(t, x)$$

On obtient alors une nouvelle équation différentielle qui est linéaire.

Nous sommes dans un cas lipschitzien et le théorème de Cauchy Lipschitz nous donne l'existence d'une unique solution que l'on appelle $Y(t, x)$ qui vérifie :

$$\begin{cases} \frac{d}{dt} Y(t, x) = \frac{d}{dx} a(t, x(t, x)) \cdot Y(t, x) \\ Y(0, x) = \frac{d}{dx} x(0, 0, x) = 1 \end{cases}$$

On va montrer qu'elle est égale à la différentielle du flot et que, de ce fait, il est bien C^1 . Pour $h \in \mathbb{R}$, on pose $g(t) = x(t, x+h) - x(t, x) - Y(t, x)h$. On a alors :

$$\begin{aligned}
\frac{d}{dt}(g(t)) &= a(t, x(t, x+h)) - a(t, x(t, x)) - \frac{d}{dx} a(t, x(t, x)) Y(t, x) h \\
&= \frac{d}{dx} a(t, \zeta) (x(t, x+h) - x(t, x)) - \frac{d}{dx} a(t, x(t, x)) Y(t, x) h \quad \zeta \in]x(t, x), x(t, x+h)[\\
&= \frac{d}{dx} a(t, \zeta) (g(t) + Y(t, x)h) - \frac{d}{dx} a(t, x(t, x)) Y(t, x) h \\
&= \frac{d}{dx} a(t, \zeta) g(t) + \frac{d}{dx} a(t, \zeta) Y(t, x) h - \frac{d}{dx} a(t, x(t, x)) Y(t, x) h \\
&= \frac{d}{dx} a(t, \zeta) g(t) + \left(\frac{d}{dx} a(t, \zeta) - \frac{d}{dx} a(t, x(t, x)) \right) (Y(t, x) h) \\
&= \frac{d}{dx} a(t, \zeta) g(t) + (Y(t, x) h) \left(\frac{d^2}{dx^2} a(t, \gamma) \right) (\zeta - x(t, x)) \quad \gamma \in]x(t, x), \zeta[
\end{aligned}$$

On a utilisé le théorème des accroissements finis aux lignes 2 et 6.

La fonction a étant supposée très régulière, on a : $|\frac{d}{dx}a(t, \zeta)| \leq C$. De plus Y étant C^0 et évoluant dans un compact, on a également : $|Y(t, x)(\frac{d^2}{dx^2}a(t, \gamma))| \leq D$.

En intégrant l'équation précédente, et en majorant comme précédemment, il en découle :

$$\begin{aligned} |g(t)| &\leq g(0) + C \int_0^t |g(u)| du + hD \int_0^t |\zeta - x(u, x)| du \\ &\leq C \int_0^t |g(u)| du + hD \int_0^t |\zeta - x(u, x)| du \end{aligned}$$

Car $g(0) = x(0, x+h) - x(0, x) - Y(0, x)h = x+h - x - 1.h = 0$

$$\begin{aligned} |g(t)| &\leq C \int_0^t |g(u)| du + hD \int_0^t |x(u, x+h) - x(u, x)| du \\ &\leq C \int_0^t |g(u)| du + hD \int_0^t |g(u) + Y(u, x).h| du \\ &\leq C \int_0^t |g(u)| du + hD \int_0^t |g(u)| du + h^2D \int_0^t |Y(u, x)| du \\ &\leq D\|Y\|_\infty Th^2 + (C+D) \int_0^t |g(u)| du \quad \text{on a choisi } h \text{ plus petit que } 1 \end{aligned}$$

Par le lemme de Gronwall, on arrive donc au résultat :

$$|g(t)| \leq D\|Y\|_\infty Th^2 \exp((C+D)T)$$

Donc $x(t, s, x)$ est bien différentiable par rapport à x et sa différentielle est $Y(t, s, x)$, donc x est C^1 par rapport à x .

Le caractère C^1 par rapport à t est quant à lui évident par l'équation même du flot :

$$\frac{d}{dt}x(t, s, x) = a(t, x(t, s, x))$$

a étant une fonction C^0 , x est C^1 par rapport à t .

On admet le caractère C^1 par rapport à s , et le plus souvent, nous supposons $s = 0$.

Le flot possède les propriétés suivantes bien utiles :

Proposition 1

Sous les hypothèses du théorème (2) on a

- Propriété de groupe : $\forall r, s, t \in [0, T]$ et $\forall x \in \mathbb{R}^N$

$$x(t, s, x(s, r, x)) = x(t, r, x) \tag{5}$$

- Pour tout $s, t \in [0, T]$, l'application $x \rightarrow x(s, t, x)$ est un C^1 difféomorphisme de \mathbb{R}^N , d'inverse $x(t, s, \cdot)$.

Preuve

– Il suffit de remarquer que les deux fonctions $x(t, s, x(s, r, x))$ et $x(t, r, x)$ vérifient le même problème de Cauchy :

$$\begin{cases} \frac{d}{dt}x(t, s, x(s, r, x)) = a(t, x(t, s, x(s, r, x))) \\ x(s, s, x(s, r, x)) = x(s, r, x) \end{cases}$$

pour la première et :

$$\begin{cases} \frac{d}{dt}X(t, r, x) = a(t, X(t, r, x)) \\ X(s, r, x) = X(s, r, x) \end{cases}$$

pour la seconde. Donc par unicité de la solution d'un problème de Cauchy, on déduit l'égalité.

- Posons $r = t$ on a alors $X(t, s, X(s, t, x)) = X(t, t, x) = x$ et donc $x \rightarrow X(s, t, x)$ est inversible d'inverse $x \rightarrow X(t, s, \cdot)$.

1.4.3 Résolution

On peut à présent résoudre le problème de Cauchy associé à (3) en montrant l'existence et l'unicité d'une solution.

Théorème 3

Sous les hypothèses du théorème (2) on suppose de plus que $u_0 \in C^1(\mathbb{R}^N)$. Alors il existe une unique solution $u \in C^1([0, T] \times \mathbb{R}^N)$ à l'équation (3) avec pour donnée initiale $u(0, x) = u_0$ et u est donnée par :

$$u(t, x) = u_0(X(0, t, x)). \quad (6)$$

u est constante le long des courbes caractéristiques.

Preuve

Si $u \in C^1$ est solution de l'équation (3), alors pour tout (t_0, x_0) on a :

$$\begin{aligned} \frac{d}{ds}(u(s, X(s, t_0, x_0))) &= \partial_t u(s, X(s, t_0, x_0)) + \partial_x u(s, X(s, t_0, x_0)) \partial_s X(s, t_0, x_0) \\ &= \partial_t u(s, X(s, t_0, x_0)) + \partial_x u(s, X(s, t_0, x_0)) a(s, X(s, t_0, x_0)) \\ &= 0 \end{aligned}$$

La solution est donc constante le long des caractéristiques. On en déduit que :

$$u(s, X(s, t_0, x_0)) = u_0(X(0, t_0, x_0)).$$

Si on pose $s = t_0$ on obtient alors :

$$u(t_0, x_0) = u_0(X(0, t_0, x_0))$$

On vient d'effectuer la remontée des caractéristiques.

La théorie que nous venons de présenter s'applique à l'équation de Vlasov et on obtient le résultat suivant :

Théorème 4

Supposons $F \in C[0, T] \times \mathbb{R}^d$ différentiable, telle que $\partial_x F \in (C[0, T] \times \mathbb{R}^d)$ et

$$\forall t \in [0, T], \quad x \in \mathbb{R}^N, \quad |F(t, x)| \leq k(1 + |x|)$$

alors pour tout $f_0 \in C^1(\mathbb{R}^{2d})$, il existe une unique solution $f \in C^1([0, T] \times \mathbb{R}^{2d})$ à l'équation (1) avec pour donnée initiale $f(0, \cdot) = f_0$, et f est donnée par :

$$f(t, x, v) = f_0(X(0, t, x, v), V(0, t, x, v))$$

où $(X, V) \in C^1$ est la caractéristique passant par (x, v) au temps t .

Preuve

C'est une conséquence des théorèmes (2) et (3) que l'on applique avec $a(t, x, v) = (v, F(t, x))$.

Si l'on note $y = (x, v)$ on a $a(t, x, v) = a(t, y)$ et donc

$$f(t, x, v) = f_0(x(0, t, x, v), v(0, t, x, v))$$

ou encore

$$f(t, y) = f_0(Y(0, t, y))$$

Où x est la position des particules et v leur vitesse. Dans cette égalité, on connaît f_0 il nous reste donc à déterminer $x(0, t, x, v)$ et $v(0, t, x, v)$.

1.5 Propriétés qualitatives de l'équation

1.5.1 Réversibilité

Nous allons dans cette partie démontrer la propriété de réversibilité de notre système. Cette propriété nous sera utile pour la suite, notamment afin de vérifier notre code.

On a vu que notre système est régi par l'équation suivante :

$$\begin{cases} \partial_t f + v \cdot \partial_x f + F \cdot \partial_v f = 0 \\ f(0, x, v) = f_0(x, v) \end{cases}$$

où f est solution de l'équation sur l'intervalle $[0, T]$. On peut alors définir la fonction $g(t, x, v)$, qui correspond à la fonction $f(t, x, v)$ en inversant le temps et les vitesses :

$$g(t, x, v) = f(T - t, x, -v)$$

Il en découle les relations suivantes :

$$\begin{cases} \partial_t g(t, x, v) = -\partial_t f(T - t, x, -v) \\ \partial_x g(t, x, v) = \partial_x f(T - t, x, -v) \\ \partial_v g(t, x, v) = -\partial_v f(T - t, x, -v) \end{cases}$$

Enfin, on peut définir une nouvelle force $\tilde{F}(t, x)$ telle que :

$$\tilde{F}(t, x) = F(T - t, x)$$

En effet, notre étude distingue deux cas concernant la force. Dans le premier, le cas linéaire, la force ne dépend pas du temps ; c'est une force extérieure au système et donc stationnaire. Nous avons donc bien : $\tilde{F}(t, x) = F(T - t, x) = F(t, x) = F(x)$.

Dans le second cas, la force F dépend de la densité de particules et donc du temps. Nous avons vu qu'elle vérifiait :

$$\partial_x F(t, x) = \rho(t, x) - \rho_e.$$

Si on intègre la fonction $g(t, x, v)$ par rapport à la vitesse, on obtient :

$$\rho_g(t, x) = \rho_f(T - t, x)$$

Ainsi f et g ont la même densité, donc on peut bien définir cette force $\tilde{F}(t, x)$.

En conséquence, si on applique $g(t, x, v)$ dans l'équation de Vlasov, on obtient :

$$\begin{aligned} \partial_t g(t, x, v) + v \cdot \partial_x g(t, x, v) + \tilde{F}(t, x) \cdot \partial_v g(t, x, v) &= -\partial_t f(T - t, x, -v) - v \cdot \partial_x f(T - t, x, -v) \\ &\quad + F(T - t, x) \cdot [-\partial_v f(T - t, x, -v)] \\ &= -[\partial_t f(T - t, x, -v) + v \cdot \partial_x f(T - t, x, -v) \\ &\quad + F(T - t, x) \cdot \partial_v f(T - t, x, -v)] \\ &= 0 \end{aligned}$$

car f est solution de l'équation de Vlasov-Poisson.

Nous obtenons finalement que g est solution de l'équation de Vlasov-Poisson suivante :

$$\begin{cases} \partial_t g(t, x, v) + v \cdot \partial_x g(t, x, v) + \tilde{F}(t, x) \cdot \partial_v g(t, x, v) = 0 \\ g(0, x, v) = f(T, x, -v) \end{cases}$$

1.5.2 Préservation du volume

Nous allons ici calculer le jacobien de la fonction $X(t, s, x)$, que l'on notera $JX(t, s, x)$ et qui correspond au déterminant de la différentielle du vecteur $X(t, s, x)$: $JX(t, s, x) = \det(DX(t, s, x))$. Cette quantité est très utile car l'application X multiplie les volumes par le jacobien :

$$\int f(X(t, s, x)) |JX(t, s, x)| dx = \int f(y) dy$$

avec le changement de variable : $y = X(t, s, x)$, $dy = |JX(t, s, x)| dx$.

Nous avons montré au théorème 2 que X est C^1 , et que sa dérivée vérifie :

$$\frac{d}{dt} DX(t, s, x) = Da(s, X(t, s, x)) DX(t, s, x).$$

Donc, lorsque $t \rightarrow s$:

$$DX(t, s, x) = Id + (t - s) Da(s, x) + o(t - s)$$

On utilise alors la propriété ci-dessous.

Lemme 2 *La différentielle du déterminant en l'identité est la trace.*

Lorsque $H \rightarrow 0$ on a :

$$\det(I + H) = 1 + \text{tr}H + o(\|H\|)$$

On obtient donc ici puisque $DX(s, s, x) = Id$ (avec $t \rightarrow s$) :

$$\det(DX(t, s, x)) = \det(Id + (t - s) Da(s, x) + o(t - s)) = 1 + (t - s) \text{tr}(Da(s, x)) + o(t - s)$$

Nous avons donc :

$$\frac{d}{dt} \det(DX(t, s, x))|_{t=s} = \text{tr}(Da(s, x)).$$

De plus, si $a(x) = (a_1, a_2, \dots, a_n)$ alors $Da(x) = (\frac{\partial a_i}{\partial x_j})_{1 \leq i, j \leq n}$, et donc :

$$\text{tr}(Da(x)) = \sum_{i=1}^n \frac{\partial a_i}{\partial x_i} = \text{div}(a(x))$$

Finalement, pour $t = s$ on obtient :

$$\frac{d}{dt} JX(s, s, x) = \text{tr}(Da(x)) = \text{div}(a(x))$$

Pour calculer le jacobien en temps quelconque, nous utilisons la relation suivante :

$$X(t, s, x) = X(t, r, X(r, s, x)), \quad \forall r \in [s, t]$$

Il en découle que :

$$\begin{cases} DX(t, s, x) = DX(t, r, X(r, s, x))DX(r, s, x) \\ JX(t, s, x) = JX(t, r, X(r, s, x))JX(r, s, x) \end{cases}$$

Pour r fixé, on dérive par rapport à t :

$$\frac{d}{dt} JX(t, s, x)|_{t=r} = \text{div}(a(X(r, s, x))) \cdot JX(r, s, x)$$

On obtient ainsi une équation différentielle linéaire, dont la résolution nous amène à :

$$JX(r, s, x) = \exp\left(\int_0^t \text{div}(a(X(u, s, x))) du\right) \quad \forall r \in [s, t]$$

En particulier pour l'équation de Vlasov-Poisson on a : $\text{div}(a) = 0$. En effet :

$$a(t, x, v) = (v, F(t, x))$$

Et donc :

$$\text{div}_{x,v} a(t, x, v) = \partial_x v + \partial_v F(t, x) = 0.$$

Ainsi pour l'équation de Vlasov, nous obtenons que $JX = 1$, et donc **le volume est conservé**.

1.5.3 Conservation de la masse, du moment et de l'énergie

Une importante propriété de notre équation est la **conservation de la masse**. Nous allons ici donner une idée de la démonstration. Nous allons montrer que la dérivée par rapport au temps de la masse $\int f dx dv$ est nulle. Rappelons l'équation de Vlasov :

$$\partial_t f + v \cdot \partial_x f + F \cdot \partial_v f = 0$$

On peut l'écrire sous la forme conservative :

$$\partial_t f + \partial_x(v \cdot f) + \partial_v(F \cdot f) = 0$$

En intégrant en x et v , on obtient :

$$\int \partial_t f dx dv + \int \partial_x(v \cdot f) dx dv + \int \partial_v(F \cdot f) dx dv = 0$$

Or $\int \partial_x(v \cdot f) dx dv = 0$ car on intègre en x une dérivée en x et par périodicité on obtient le résultat. De même, $\int \partial_v(F \cdot f) dx dv = 0$ en utilisant le théorème de Fubini et les conditions aux limites. Par suite $\int \partial_t f dx dv = 0$. Et donc :

$$\partial_t \left(\int f dx dv \right) = 0$$

De plus, on peut vérifier que notre système **conserve le moment**. Dans le cas discret, le moment s'écrit :

$$M_N(t) = \frac{1}{N} \sum_{i=1}^N V_i(t)$$

En dérivant par rapport à t cette égalité nous obtenons :

$$\frac{d}{dt} \left(\frac{1}{N} \sum_{i=1}^N V_i(t) \right) = \frac{1}{N} \sum_{i=1}^N F_i(t)$$

Or, par le principe de l'action et de la réaction dans un système fermé, la somme $\sum_{i=1}^N F_i(t)$ est nulle. Donc le moment est conservé. Il est également possible de le montrer dans le cas continu. Le moment s'écrit alors :

$$M_f(t) = \int f(t, x, v)v \, dx dv$$

Enfin, on peut aussi montrer que notre équation **conserve l'énergie** donnée par :

$$E_f(t) = \int f(t, x, v) \frac{v^2}{2} \, dx dv + \frac{1}{2} \int |F|^2 \, dx$$

2 La méthode d'approximation numérique

On cherche à approcher $x(t)$ et $v(t, x)$ à l'aide d'une méthode numérique. Dans cette partie, nous allons commencer par détailler ce qu'est une méthode numérique, puis expliciter l'équation différentielle qui nous intéresse et la mise en place de notre programme.

2.1 Les différentes méthodes numériques

Nous devons maintenant résoudre le problème suivant :

$$\begin{cases} \dot{X} = V \\ \dot{V} = F(X) \end{cases}$$

Ce système n'a pas toujours de solutions analytiques, on utilisera donc des méthodes d'approximation numérique. Dans la suite, nous allons étudier différentes méthodes numériques dans le cas général qui s'appliquent bien à notre problème.

2.1.1 Approximation numérique et erreurs

Dans cette partie, on va se pencher sur des équations de la forme

$$Y' = F(t, Y).$$

Définition 2 (Vocabulaire)

Dans la suite, nous prendrons les notations suivantes :

- T : le temps final,
- M : le nombre d'itérations totales,
- $\Delta t = \frac{T}{M}$: le pas de temps,
- $y(t^n)$: la solution exacte au temps $t^n = n\Delta t$,
- y^n : la solution approchée au temps t^n .

Pour évaluer l'efficacité d'une méthode numérique, on définit l'erreur de consistance, l'erreur et l'ordre du schéma :

Définition 3 (Erreur de consistance)

L'erreur de consistance R^n correspond à la différence entre la solution réelle du problème et celle calculée par le schéma pour une itération donnée. On peut aussi dire qu'elle représente l'erreur commise à chaque itération. Si on note S le schéma, on a alors $y^{n+1} = S(y^n)$ et

$$R^n = y(t^{n+1}) - S(y(t^n)).$$

Définition 4 (Erreur)

L'erreur d'un schéma notée e^n correspond à la différence entre la valeur de la solution réelle du problème et celle calculée par le schéma pour une itération donnée :

$$e^n = y(t^n) - y^n$$

Définition 5 (Ordre d'un schéma)

Un schéma est dit d'ordre i lorsque l'on a l'estimation de l'erreur suivante :

$$e^n \leq C\Delta t^i \quad \text{pour } 0 \leq n \leq M.$$

Une des méthodes numériques que nous utilisons dans notre travail est la méthode d'Euler explicite. Nous allons donc tout d'abord nous intéresser aux propriétés de ce schéma.

2.1.2 Méthode d'Euler explicite**Définition 6 (Euler explicite)**

La méthode d'Euler explicite consiste à approcher la solution aux instants t^n du problème de Cauchy à l'aide d'une relation de récurrence donnée par :

$$y^{n+1} = y^n + \Delta t F(t^n, y^n), \quad \forall 0 \leq n \leq M-1$$

Nous pouvons calculer l'erreur de consistance de ce schéma :

$$R^n = y(t^{n+1}) - y(t^n) - \Delta t F(t^n, y(t^n)), \quad \forall 0 \leq n \leq M-1$$

et l'erreur $e^n = y(t^n) - y^n$ ce qui nous amène par un calcul simple à la relation :

$$e^{n+1} = e^n + \Delta t (F(t^n, y(t^n)) - F(t^n, y^n)) + R^n$$

Théorème 5

La méthode d'Euler explicite est d'ordre 1 et on a $e^n \leq \Delta t C e^{LT}$

Preuve

D'après les formules de développement limité de Taylor appliquées à y au point t^n on a :

$$y(t^{n+1}) = y(t^n + \Delta t) = y(t^n) + y'(t^n)\Delta t + \frac{y''(u^n)\Delta t^2}{2} \quad t^n \leq u^n < t^{n+1}$$

On applique cela à la définition de R^n et il en découle :

$$\begin{aligned} R^n &= y'(t^n)\Delta t + \frac{y''(u^n)\Delta t^2}{2} - \Delta t F(t^n, y^n) \\ &= \frac{y''(u^n)\Delta t^2}{2} \end{aligned}$$

Sachant que y est solution exacte du problème de Cauchy on a $y'(t^n) - F(y(t^n), y^n) = 0$. Enfin, y étant supposée régulière, $\sup |y''| < \infty$, on obtient :

$$R^n \leq C\Delta t^2$$

Avec $C = \frac{\|y''\|_\infty}{2}$.

Revenons maintenant à l'équation :

$$e^{n+1} = e^n + \Delta t (F(t^n, y(t^n)) - F(t^n, y^n)) + R^n$$

F étant supposée L -lipschitzienne, on a :

$$\begin{aligned} \|F(t^n, y(t^n)) - F(t^n, y^n)\| &\leq L\|y(t^n) - y^n\| \\ &\leq L e^n \end{aligned}$$

Par suite, en utilisant cette inégalité, ainsi que l'inégalité triangulaire, on obtient :

$$\|e^{n+1}\| < (1 + L\Delta t)\|e^n\| + R^n$$

Et donc par l'inéquation précédente on a :

$$\begin{aligned} \|e^1\| &\leq (1 + L\Delta t)\|e^0\| + R^0 \\ \|e^2\| &\leq (1 + L\Delta t)\|e^1\| + R^1 \\ &\dots \\ \|e^{n+1}\| &\leq (1 + L\Delta t)\|e^n\| + R^n \end{aligned}$$

En sommant et simplifiant, on arrive à :

$$\|e^{n+1}\| \leq \|e^0\| + L\Delta t \sum_{k=0}^n \|e^k\| + \sum_{k=0}^n \|R^k\|$$

Or,

$$\begin{aligned} \sum_{k=0}^n \|R^k\| &\leq (n+1)C\Delta t^2 \\ &\leq MC\Delta t^2 \\ &\leq \frac{T}{\Delta t}C\Delta t^2 \\ &\leq TC\Delta t \end{aligned}$$

D'autre part, en utilisant la fait que $e^0 = y^0 - y(0) = 0$, on obtient :

$$\|e^{n+1}\| \leq L\Delta t \sum_{k=0}^n \|e^k\| + TC\Delta t$$

Il nous suffit maintenant d'utiliser le lemme de Gronwall discret afin d'avoir :

$$\begin{aligned} \|e^n\| &\leq TC\Delta te^{L\Delta tn} \\ &\leq TC\Delta te^{L\Delta tM} \\ &\leq TC\Delta te^{LT} \\ &\leq C'\Delta t \end{aligned}$$

La méthode d'Euler explicite est donc bien d'ordre 1.

Lemme 3 (Gronwall discret)

Soit $(z^n)_n$ une suite de nombres positifs avec $z^0 = 0$ et $C, L > 0$. On suppose que

$$\forall n \geq 0, \quad z^{n+1} \leq C + L\Delta t \sum_{k=0}^n z^k,$$

alors on a

$$\forall n \geq 0, z^n \leq Ce^{L\Delta tn}.$$

Dans le cas de notre travail, nous ne sommes pas en dimension 1. En effet, il y a deux inconnues qui sont X et V , qui dépendent l'une de l'autre. Le schéma d'Euler explicite peut se transposer à la dimension 2 et on obtient le système d'équations suivant :

$$\begin{cases} X^{n+1} = X^n + V^n \Delta t \\ V^{n+1} = V^n + F(X^n) \Delta t \end{cases}$$

Les calculs des erreurs sont similaires à ceux fait précédemment, mais font appel au calcul différentiel en dimension supérieure à 1. Il faut donc les adapter en utilisant les différentielles au lieu des dérivées, la hessienne au lieu des dérivées secondes...

Pour le travail que nous souhaitons effectuer, nous avons estimé que la précision d'une méthode d'ordre 1 n'était pas suffisante. Nous allons donc utiliser la méthode du point milieu qui est un peu plus complexe que la méthode d'Euler explicite.

2.1.3 Méthode du point milieu

Définition 7 (Méthode du point milieu)

La méthode du point milieu consiste à approcher la solution aux instants t^n du problème de Cauchy à l'aide d'une relation de récurrence donnée par :

$$y^{n+1} = y^n + \Delta t F \left(t^n, y^n + \frac{\Delta t}{2} F(t^n, y^n) \right), \quad \forall 0 \leq n \leq M-1$$

Théorème 6

La méthode du point milieu est d'ordre 2.

Preuve

Pour une meilleure lisibilité des calculs, on ne notera pas la dépendance de F en t^n . On a l'erreur de consistance suivante :

$$\begin{aligned} R^n &= y(t^{n+1}) - y(t^n) - \Delta t F \left(y(t^n) + \frac{\Delta t}{2} F(y(t^n)) \right) \\ &= y(t^{n+1}) - y(t^n) - \Delta t F \left(y(t^n) + \frac{\Delta t}{2} y'(t^n) \right) \\ &= y(t^{n+1}) - y(t^n) - \Delta t F \left(y(t^n + \frac{\Delta t}{2}) + O(\Delta t^2) \right) \quad \text{par Taylor d'ordre 1 sur } y \\ &= y(t^{n+1}) - y(t^n) - \Delta t F(y(t^n + \frac{\Delta t}{2})) + \Delta t F'(y(t^n + O(\Delta t^2))) \quad \text{par Taylor d'ordre 0 sur } F \\ &= y(t^{n+1}) - y(t^n) - \Delta t F(y(t^n + \frac{\Delta t}{2})) + O(\Delta t^3) \\ &= y(t^{n+1}) - y(t^n) - \Delta t y'(t^n + \frac{\Delta t}{2}) + O(\Delta t^3) \end{aligned}$$

Nous allons maintenant faire deux développements de Taylor de y au point $t^n + \frac{\Delta t}{2}$

$$\begin{aligned} y(t^{n+1}) &= y(t^n + \frac{\Delta t}{2} + \frac{\Delta t}{2}) = y(t^n + \frac{\Delta t}{2}) + y'(t^n + \frac{\Delta t}{2}) \frac{\Delta t}{2} + y''(t^n + \frac{\Delta t}{2}) \frac{\Delta t^2}{4} + O(\Delta t^3) \\ y(t^n) &= y(t^n + \frac{\Delta t}{2} - \frac{\Delta t}{2}) = y(t^n + \frac{\Delta t}{2}) - y'(t^n + \frac{\Delta t}{2}) \frac{\Delta t}{2} + y''(t^n + \frac{\Delta t}{2}) \frac{\Delta t^2}{4} + O(\Delta t^3) \end{aligned}$$

En remplaçant dans le calcul de R^n et en simplifiant, on obtient :

$$R^n = O(\Delta t^3) \leq C \Delta t^3$$

Par un calcul semblable à celui effectué dans le cas d'Euler explicite, on arrive à :

$$e^n \leq C' \Delta t^2$$

La méthode du point milieu est donc bien d'ordre 2.

2.1.4 Méthode de Saute mouton.

La méthode de Saute-mouton combine deux méthodes du point milieu. Dans le cas de notre schéma elle s'écrit comme suit :

$$\begin{cases} X^{n+\frac{1}{2}} = X^n + V^n \frac{1}{2} \Delta t \\ V^{n+1} = V^n + F(X^{n+\frac{1}{2}}) \Delta t \\ X^{n+1} = X^{n+\frac{1}{2}} + V^{n+1} \frac{1}{2} \Delta t \end{cases}$$

Étant d'ordre 2 comme la méthode du point milieu, c'est de celle ci dont nous allons nous servir dans notre programme car elle est bien plus efficace que la méthode d'Euler explicite.

2.2 Équations et problème.

On considère un système discret de N particules, qui peuvent être, comme nous l'avons vu des plasmas, ou des galaxies. Tout au long de notre travail, et pour simplifier nos calculs, nous allons considérer que les particules ne se déplacent que dans une seule direction.

On considère qu'une particule présente une masse de valeur $1/N$. Chaque particule agit sur les particules voisines. Si on considère les particules i et j de notre système, la force exercée par la particule j sur la particule i s'écrit $F(X_i - X_j)$. Dans la pratique, cette force correspond au champ électrique, ou bien à la force gravitationnelle.

Tout d'abord, la relation générale de la cinétique, qui dit que la dérivée de la position $x(t)$ par rapport au temps est égale à la vitesse $v(t)$ et d'autre part, le principe fondamental de la dynamique appliqué à ce système de particules nous permettent d'écrire le système d'équations suivant :

$$\begin{cases} \dot{X}_i = V_i \\ \frac{1}{N} \dot{V}_i = \sum_{j \neq i} F(X_i - X_j) \end{cases}$$

Or, la force gravitationnelle s'écrit :

$$F_g(X_i - X_j) = -\frac{G}{N^2} \cdot \frac{X_i - X_j}{(X_i - X_j)^3}$$

Et, la force coulombienne est donnée par :

$$F_c(X_i - X_j) = \frac{1}{4} \cdot \frac{q_i \cdot q_j}{(X_i - X_j)^2}$$

en notant q_i la charge de la particule i .

D'une manière générale, on peut considérer que la charge d'une particule est proportionnelle à sa masse. Donc on peut réécrire le système d'équation en remplaçant la force $F(X_i - X_j)$ par la force gravitationnelle ou coulombienne. Dans les deux cas, on obtiendra :

$$\begin{cases} \dot{X}_i = V_i \\ \dot{V}_i = \frac{1}{N} \sum_{j \neq i} \tilde{F}(X_i - X_j) \end{cases}$$

Avec $\tilde{F}(X_i - X_j) = N^2 F_g$ ou $N^2 F_c$.

Pour résoudre ces équations, nous avons conçu un programme à l'aide du logiciel Scilab. Nous allons maintenant détailler la mise en place de notre code.

2.3 Le programme.

2.3.1 Première étape : la résolution générale avec la force $F(t, x)$ donnée.

Pour commencer, nous avons considéré le cas où la force $F(t, x)$ est connue, on la note $\alpha(x)$. Nous avons alors simplement utilisé le solveur *Ode* de Scilab pour résoudre notre équation différentielle.

On associe aux positions initiales la matrice $Y0$, qui a deux lignes et N colonnes, avec N le nombre de particules. La première ligne correspond aux positions X_i et la deuxième correspond aux vitesses V_i de chaque particule i . On a fait en sorte que le programme place uniformément les valeurs des X_i et V_i dans l'intervalle $[0,1]$ à l'aide de la fonction *rand* de scilab. Ainsi, si on trace le vecteur position initiales $Y0$ avec la fonction *plot2d*, toutes les particules se trouvent dans un carré de taille $[0,1] \times [0,1]$ en position et en vitesse.

Nous avons fait une fonction, appelée ***positionfinale***, qui au vecteur $Y0$ initial associait le vecteur Y au temps final, noté t_f , qui était calculé avec le solveur *Ode* du logiciel. L'application de cette fonction nous donnait alors un nuage de point des particules au temps initial, et un autre au temps final.

Cependant, cette approche ne nous permettait pas de voir les positions évoluer au cours du temps. Nous avons alors cherché à résoudre notre équation par une autre méthode, qui fait intervenir les différentes positions intermédiaires. La première solution que l'on ait choisit était d'utiliser la méthode numérique d'Euler, que l'on a intitulée ***Euler*** dans notre programme.

C'est une méthode de résolution numérique explicite qui est, comme on l'a vu dans la partie précédente, d'ordre 1. Elle s'écrit :

$$\begin{cases} X_i = X_{i-1} + dt.V_{i-1} \\ V_i = V_{i-1} - dt.\alpha(X_{i-1}) \end{cases}$$

En complément, on a utilisé une autre méthode numérique, que l'on a appelée ***sautemouton***, qui d'ordre 2. Elle s'écrit :

$$\begin{cases} X_{i-\frac{1}{2}} = X_{i-1} + \frac{1}{2}.dt.V_{i-1} \\ V_i = V_{i-1} - dt.\alpha(X_{i-\frac{1}{2}}) \\ X_i = X_{i-\frac{1}{2}} + \frac{1}{2}.V_i.dt \end{cases}$$

Pour appliquer ces méthodes de résolution, nous avons modifié notre fonction *positionfinale* en une fonction appelée ***position_inter***. Au temps t , elle applique la fonction *sautemouton* (ou bien, au choix *Euler*) au vecteur Y de taille $2 \times N$ correspondant au vecteur position et vitesse de toutes les particules calculé à l'instant précédent $t - dt$. Notre programme calcule alors les positions de chaque particule, à tous les intervalles de temps. Avec la fonction *plot2d*, que l'on applique pour toutes les solutions, donc à tous les intervalles de temps, nous observons l'évolution du système au cours du temps. De plus, nous avons pu comparer la convergence des deux méthodes numériques utilisées.

Le fait de modifier la fonction α nous a permis d'observer les différentes évolutions que pouvait prendre le système de particules.

2.3.2 Deuxième étape : la force dépend de la position.

D'une manière générale, on a vu que la force $F(t, x)$ d'une particule X_i dépend de la densité de particules autour de celle-ci.

Ainsi, nous avons créé une fonction, appelée *densite*, qui calcule le nombre de particules situées dans les bandes de largeur dx . Elle renvoie un vecteur de taille $2/dx$ dont la i -ème composante donne le nombre de particules dans l'intervalle $[-1 + (i - 1)dx; -1 + idx]$.

Enfin nous avons pu écrire la fonction *force*. Notre but était donc de calculer la force F_i en chaque point de coordonnée $-1 + idx$. Comme nous l'avons présenté dans la première partie, la force vérifie l'équation :

$$\partial_x F(t, x) = \rho(t, x) - \rho_e$$

Pour nos calculs, nous avons prit $\rho_e = 1/2$. Finalement on obtient :

$$\frac{F_{i+1} - F_i}{dx} = \rho - \frac{1}{2} \iff F_{i+1} = F_i + dx \cdot (\rho - \frac{1}{2})$$

D'autre part, nous avons ajouté une fonction, appelée *miseenplace* qui périodise en position le vecteur Y . En effet cela nous permet de conserver toutes les particules dans l'intervalle $[-1; 1]$, afin de pouvoir observer leur mouvement sur un espace réduit (de longueur 2) plutôt que sur tout l'axe Ox .

Pour ce cas là, nous avons repris la fonction *position_inter* en l'adaptant. Elle calcule la force F_i et calcule les nouvelles vitesses et positions, selon le schéma de sautemouton. De même, on affiche les positions et vitesses à tous les intervalles de temps.

Par la suite, nous avons écrit une fonction appelée *réversibilité*. Elle permet de vérifier que le programme fonctionne correctement. En effet, cette fonction lance *position-inter* pour une durée t_f , puis elle inverse les vitesses, et relance la fonction *position-inter* pendant la même durée t_f . Si on obtient le même carré après application de cette fonction que le carré initial, alors cela signifie que le programme fonctionne correctement. Cette fonction illustre parfaitement le phénomène de réversibilité de l'équation que l'on a vu en première partie.

Enfin, le coût total de chacun des deux programmes est de l'ordre de $M.N$, avec N le nombre de particules et $M = \frac{t_f}{dt}$

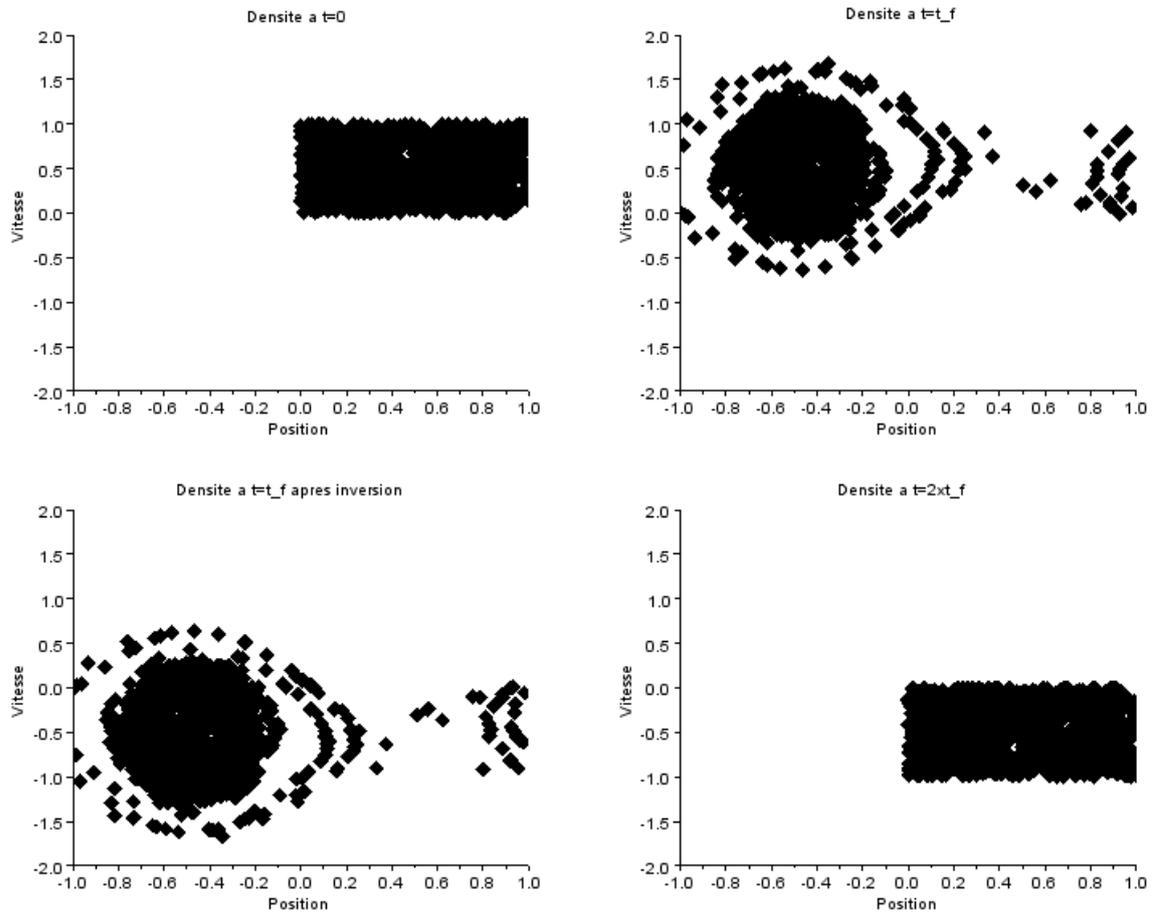
3 Résultats numériques et interprétations

3.1 Vérification du code

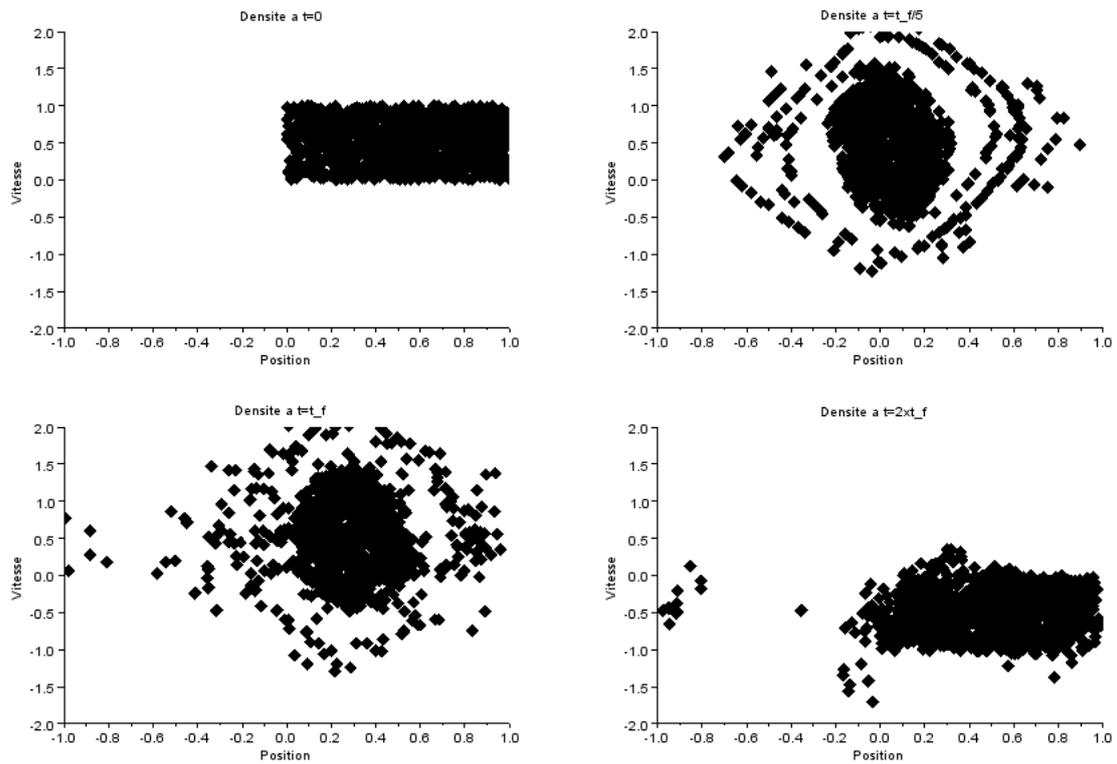
3.1.1 Utilisation de la réversibilité

Nous avons vu dans la première partie, que le volume occupé par les particules était conservé. Dans un premier temps, cette propriété peut être vérifiée visuellement avec notre code. Nous avons programmé la fonction *réversibilité* qui permet de valider le calcul dans le cas où F est une force extérieure et ne dépend pas du temps et pour des paramètres bien choisis. On observe alors, à une symétrie d'axe Ox près, que les particules parcourent le chemin inverse pour se placer dans le rectangle de départ. Cela montre que les erreurs faites sur la trajectoire restent suffisamment petites car celles de l'aller et du retour s'additionnent et que malgré cela, la simulation reste satisfaisante.

Voici l'exemple d'une simulation réalisée dans le cas attractif avec pour paramètres 1000 particules, $t_f = 10$, $dt = 0.1$ et $dx = 0.01$.

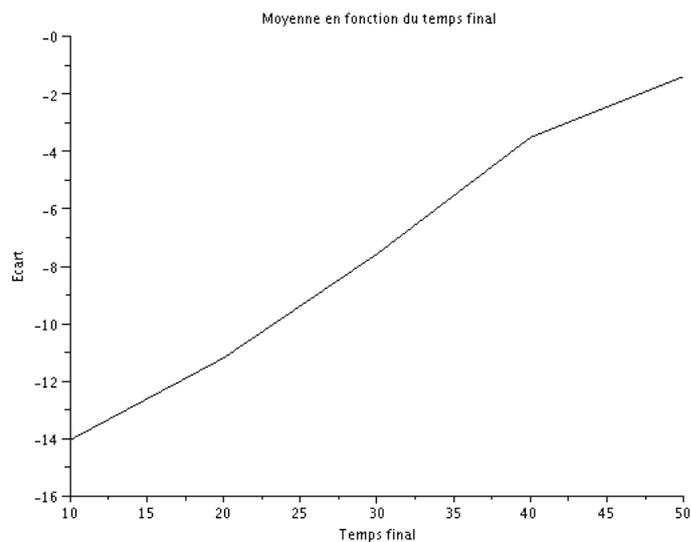


Cette fonction permet aussi de mettre en évidence les limites de notre programme. On observe par exemple que pour un temps final trop élevé la simulation est inexploitable puisque les erreurs deviennent trop importantes et les vitesses et positions des particules obtenues ne correspondent plus à ce que l'on doit obtenir. Nous avons ici fait une simulation dans le même cas attractif avec un temps final $t_f = 35$. On peut en effet observer que le rectangle obtenu à la fin n'est plus aussi net.



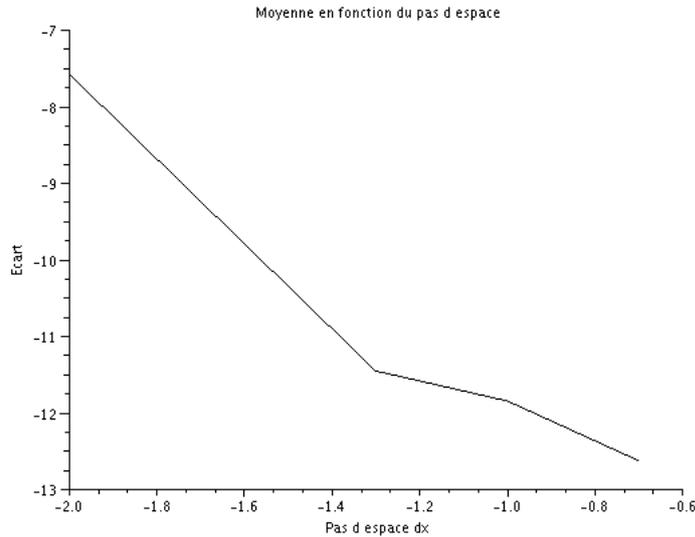
L'amélioration de la fonction *reversibilite* nous permet d'obtenir une **estimation quantitative** de l'erreur. En effet, elle calcule la distance moyenne entre les positions initiales et celles obtenues après aller-retour.

Dans les quatre graphiques suivants, nous allons observer la relation entre les variations des paramètres t_f , dx et dt et les distances moyennes des positions.



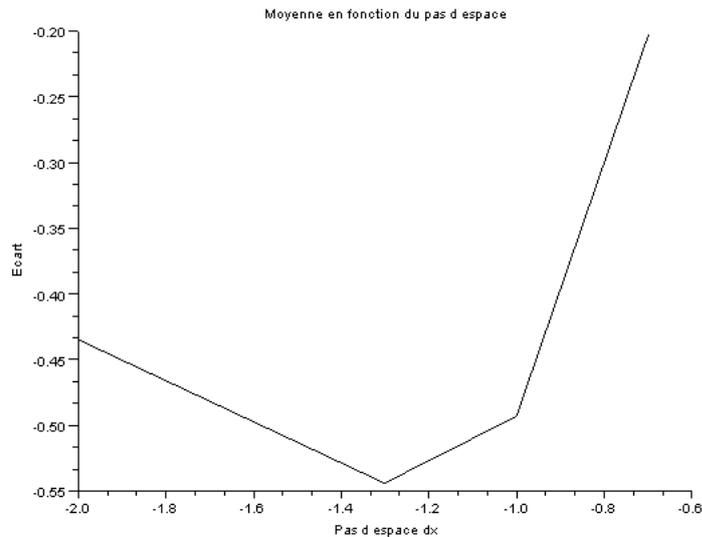
Ce graphique a été obtenu pour des valeurs de t_f variant de 10 à 50 unités de temps, dans le cas d'une force attractive, pour 1000 particules et $dt = 0.05$ et $dx = 0.01$. De plus, pour mieux visualiser les résultats, nous avons utilisé l'échelle logarithmique (en base 10) pour l'axe des ordonnées.

La représentation graphique obtenue est quasi-linéaire. On retrouve donc le résultat obtenu dans la partie théorique (théorème 5) qui lie l'erreur à l'exponentielle du temps final. De plus, on constate que pour un temps final de 40 unités de temps, l'erreur reste très petite. Cela veut dire que notre code préserve bien la propriété de réversibilité.

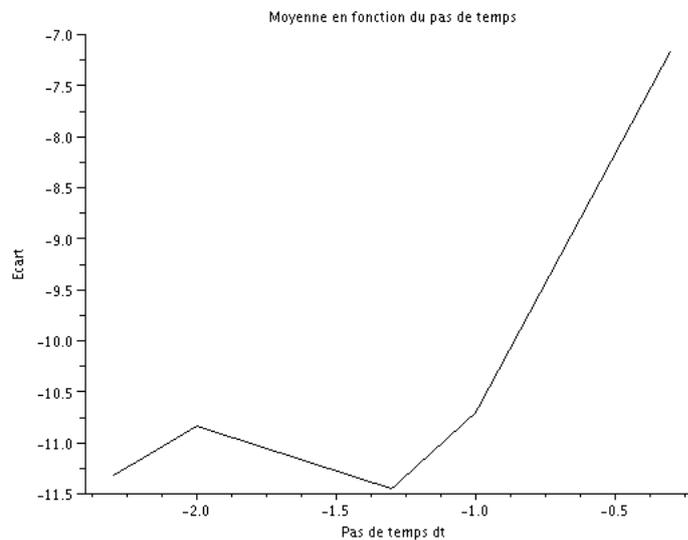


Le deuxième graphique montre les variations de l'écart moyen en fonction du pas d'espace dx . Les paramètres utilisés sont : une force attractive, 1000 particules, $dt = 0.05$ et un temps final de 30 ; pour dx prenant les valeurs 0.2, 0.1, 0.05 et 0.01. Cette fois-ci, nous avons utilisé l'échelle logarithmique pour les deux axes.

Sur ce graphique, on constate que les erreurs sont petites, comprises entre 10^{-7} et 10^{-12} . On remarque aussi que l'écart augmente quand le pas d'espace diminue. En effet, lorsque dx est grand, la dynamique est triviale, c'est-à-dire que la densité, et donc la force, ne varient pas beaucoup, et on fait peu d'erreurs. Cependant, cette situation n'est pas raisonnable pour notre approximation. En revanche, quand dx diminue, la dynamique devient plus proche du système physique mais les erreurs réalisées sont plus grandes. Il faut donc choisir le dx en fonction du nombre de particules N , de telle façon que le rapport entre N et le nombre d'intervalles soit de l'ordre de 10.



Dans le graphique ci-dessus, nous avons comparé l'écart entre les positions finales obtenues pour $dt = 0.01$ et $dt/2$ en faisant varier dx . On s'est placé dans les conditions d'une force attractive avec 1000 particules et un temps final de 30 ; dx prenant les valeurs 0.2, 0.1, 0.05 et 0.01. Ces résultats confirment ce que l'on a annoncé précédemment : il y a un choix de dx optimal, il ne doit être ni trop grand ni trop petit.



Ici, nous avons mesuré l'écart en fonction du pas de temps, en échelle logarithmique. Les données utilisées sont : t_f de 30, 1000 particules, et $dx = 0.05$, pour une force attractive ; et $dt=0.5, 0.1, 0.05, 0.01, 0.005$. On constate qu'il est inutile de trop diminuer le pas de temps, car l'erreur est de l'ordre de 10^{-10} à partir de $dt = 0.1$.

Ces graphiques nous permettent enfin de vérifier que **le code préserve la réversibilité de l'équation**, qui est une propriété importante de l'équation. En effet, notre point de départ est un rectangle aux bords bien dessinés et à la fin nous obtenons une figure qui est presque aussi nette pour des paramètres bien choisis.

3.1.2 Comparaison pour des intervalles de temps distincts

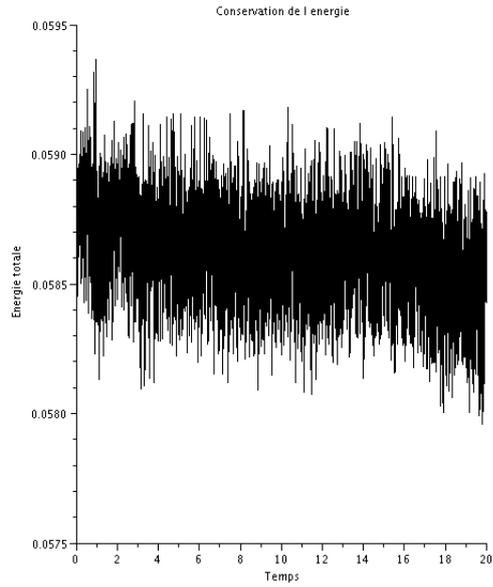
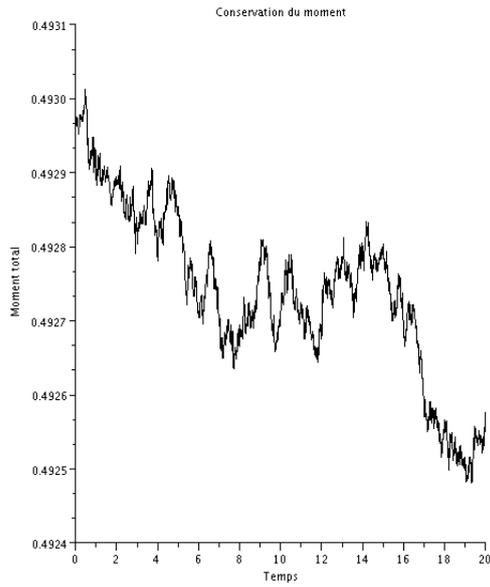
Dans cette partie, nous utilisons une fonction qui nous donne l'écart moyen entre les positions finales obtenues pour dt et $dt/2$. Ces graphiques mettent en évidence que la convergence de notre méthode ne dépend pas réellement de la valeur de dt .



Ce graphique a été obtenu pour 1000 particules, $dx = 0.01$, et $t_f = 5$, dans le cas d'une force attractive. Nous constatons que pour dt petit, l'écart entre les deux solutions obtenues devient très faible, et donc que la valeur de dt n'influe plus sur la convergence de la simulation.

3.1.3 Conservation du moment et de l'énergie

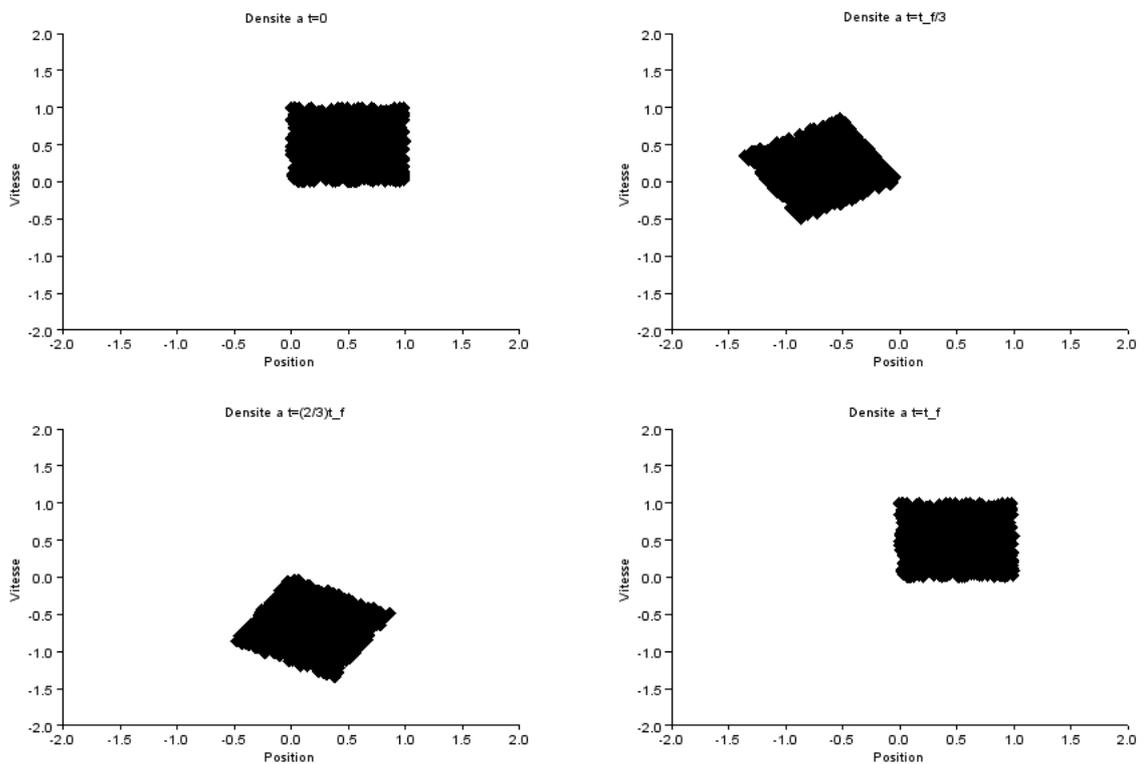
Nous avons vu dans la partie théorique que le moment et l'énergie sont conservés. On peut en effet le vérifier sur ces graphiques, où l'on a représenté les variations du moment et de l'énergie en fonction du temps. Ils ont été obtenus pour une force répulsive, un temps final de 10 unités de temps, $dt = 0.001$ et $dx = 0.01$. Pour le moment, les variations restent très petites, de l'ordre de 0,1 %, alors que pour l'énergie, elles sont de l'ordre de 0,3 % (rapportées à l'énergie cinétique qui vaut ici 0,33).



3.2 Cas linéaire

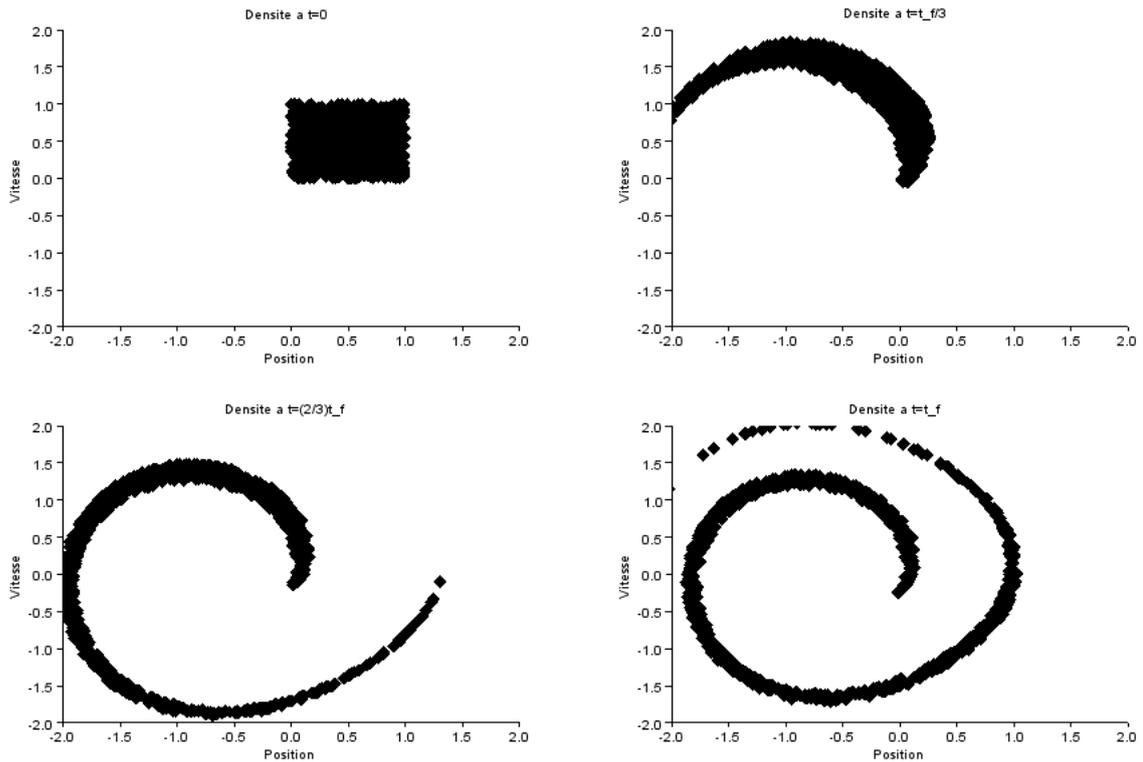
3.2.1 La rotation

Nous nous sommes penchés en détail sur le cas de l'oscillateur harmonique, c'est-à-dire lorsque la force F est donnée par $F(x) = -x$. On observe alors que le carré fait une rotation mais conserve sa forme car chaque particule tourne avec la même vitesse angulaire. De plus, on peut constater que la **vitesse numérique** est proche de ce que l'on doit obtenir puisque en 10π le carré fait 5 tours. La simulation suivante a été réalisée avec 1000 particules, $t_f = 10\pi$ et $dt = 0.01$.



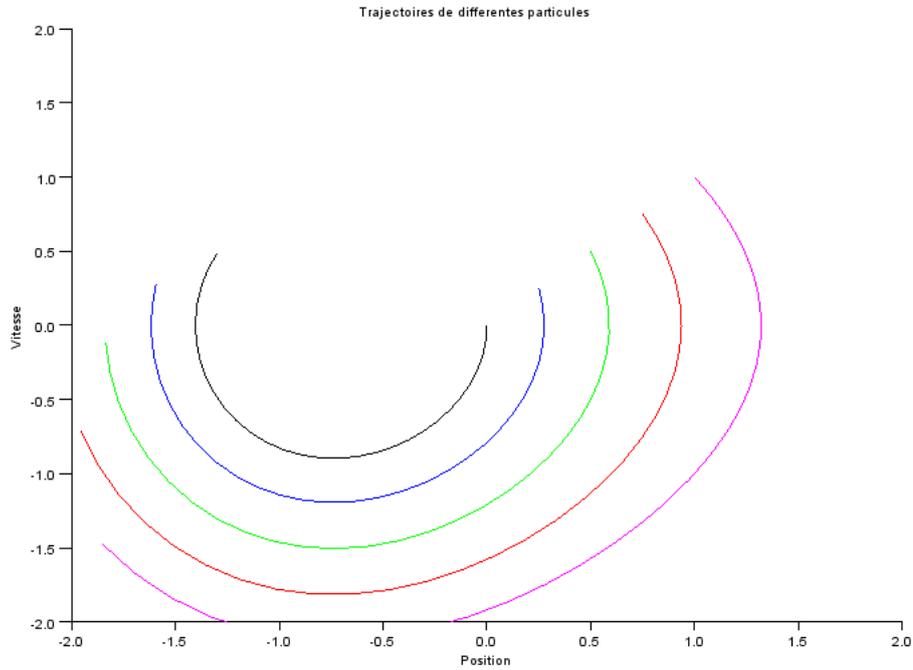
3.2.2 Filamentation

Plaçons nous dans le cas linéaire et lorsque la force exercée sur les particules est une force extérieure. Nous choisissons pour cet exemple la force $f(x) = -x - \cos x$ et pour paramètres 1000 particules, $t_f = 60$ et $dt = 0.01$. Pour réaliser la simulation suivante, nous avons utilisé la partie du programme non périodique en espace, de façon à mieux voir apparaître le phénomène de filamentation.



D'après la partie théorique, le volume est préservé, mais on observe un allongement dans une direction. C'est le phénomène de **filamentation**. Les particules, qui sont au début positionnées dans le rectangle de position et vitesse, se déplacent en s'allongeant pour finalement former un filament. Ce phénomène s'explique par le fait que les particules ne se déplacent pas toutes à la même vitesse.

On peut visualiser sur le graphique suivant les écarts de vitesse. Nous avons représenté les trajectoires de 5 particules en fonction du temps. Les positions initiales sont : $(0,0)$ pour la trajectoire noire, $(0.25,0.25)$ pour la bleue, $(0.5,0.5)$ pour la verte, $(0.75,0.75)$ pour la rouge et $(1,1)$ pour la trajectoire rose.



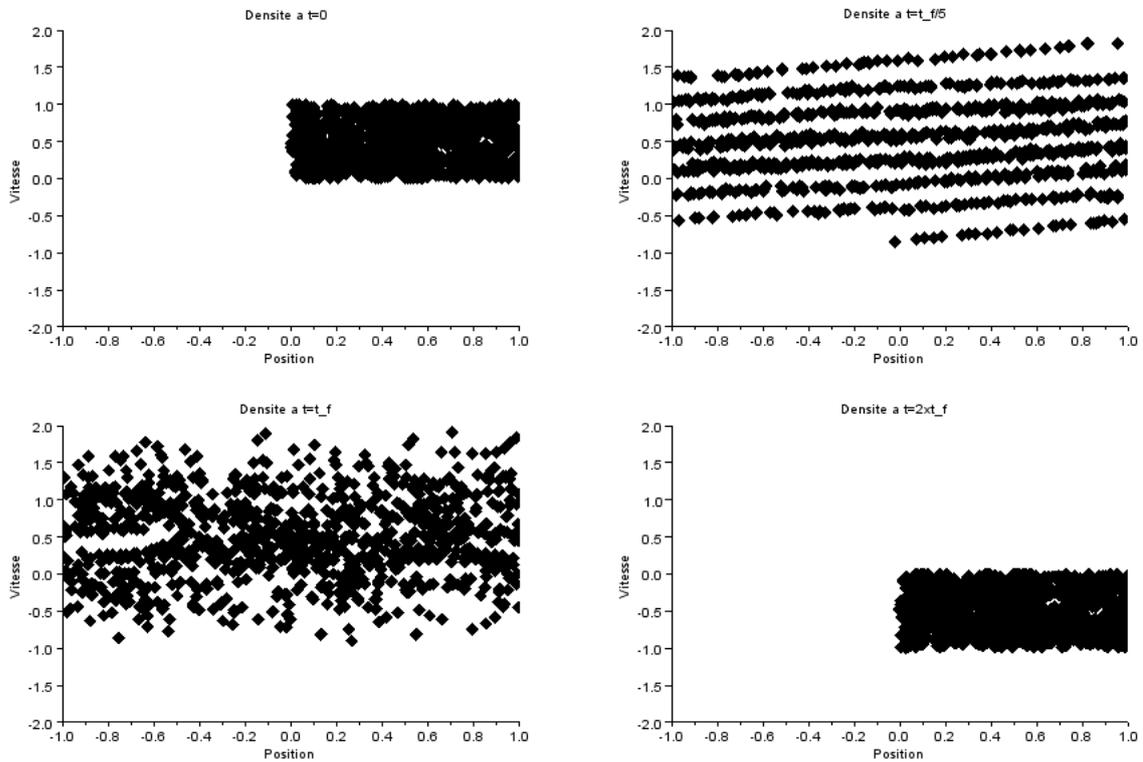
C'est dans le cas où la filamentation apparaît que l'on peut voir le principal **intérêt de notre code et de la méthode particulaire**. En effet, si pour réaliser ce programme on s'était appuyé sur un maillage fixé, on n'aurait pas pu voir les détails en dessous de l'échelle du maillage. C'est la méthode particulaire qui nous permet de voir se dessiner des lignes.

3.3 Cas non linéaire

3.3.1 Cas répulsif : l'exemple des plasmas

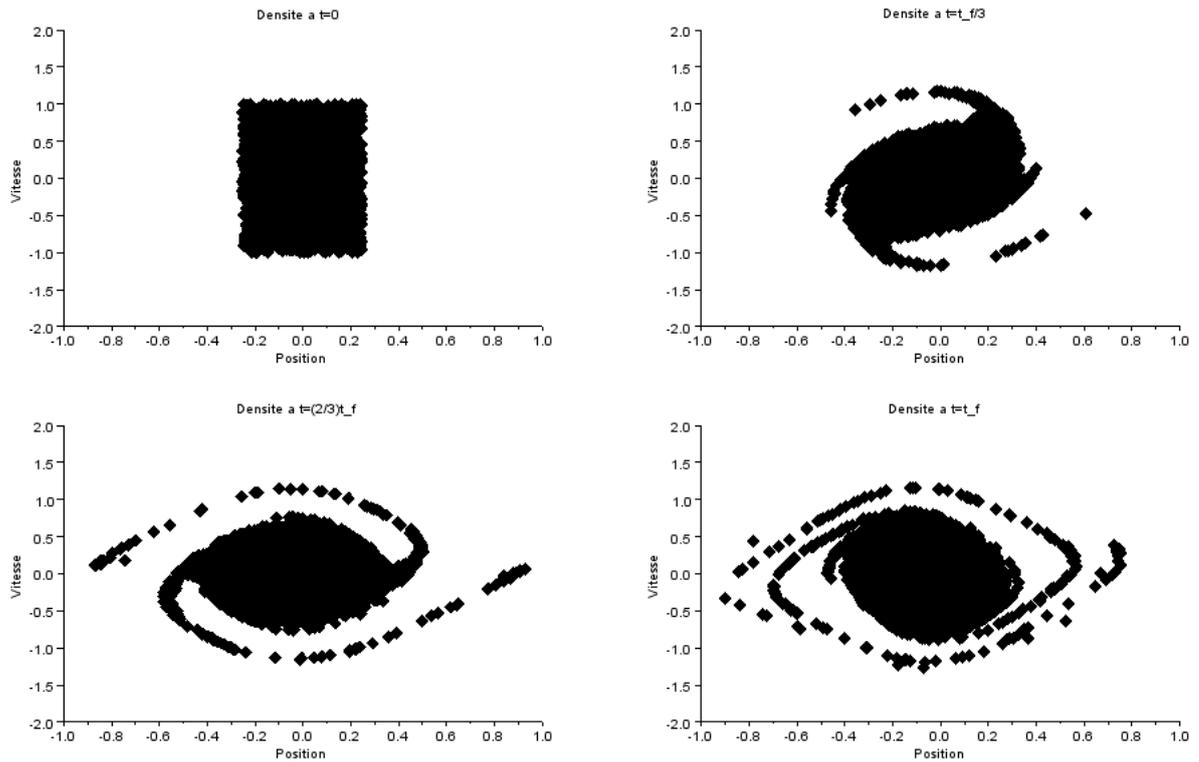
Pour la simulation des plasmas, nous nous plaçons dans le cas d'une force répulsive. Nous avons choisi comme paramètres 1000 particules, $t_f = 30$, $dt = 0.1$ et $dx = 0.01$.

Après avoir vu une filamentation, nous pouvons observer ici que le plasma semble s'homogénéiser, c'est-à-dire que les particules sont presque équiréparties en x (la distribution des particules ne dépend presque plus de x). Cependant, en utilisant la fonction *réversibilité* on s'aperçoit qu'elles reviennent à leur position initiale. Ce résultat s'explique par le fait que les particules ont en quelque sorte gardé en "mémoire" leur position initiale. Si nous étions au temps final dans une configuration véritablement homogène en x , c'est-à-dire avec une distribution ne dépendant que de $v : f(v)$, nous serions à une configuration d'équilibre. Lorsque l'on inverse les vitesses on ne devrait pas revenir au carré mais rester proche de cette configuration d'équilibre.



3.3.2 Cas attractif : l'exemple des galaxies

Dans cette partie, nous avons simulé l'évolution des étoiles dans une galaxie. Nous avons donc considéré une force attractive, 1000 étoiles, un temps final de 7 unités de temps, $dt = 0.01$ et $dx = 0.01$.



Nous voyons apparaître deux filaments très fins, difficiles à obtenir avec une simulation numérique. Ce phénomène s'explique comme dans le cas linéaire par des différences de vitesses de rotation entre les particules. Celles du centre tournent lentement autour de l'origine. Celles de la périphérie (et surtout des coins) sont assez rapidement "expulsées" du centre, et acquièrent ensuite une vitesse de rotation assez élevée.

Tous ces résultats ont été obtenus avec le `code` donné dans la partie suivante.

4 Code

4.1 Le cas linéaire

```
// Cas d'une force exterieure connue.

// La fonction position permet de repartir uniformement les N particules
// dans le carre [0,1]x[0,1] en position et en vitesse. Elle est utilisee
// pour les simulations dans le cas repulsif.

function Y0=position(N)
    Y0=rand(2,N,'uniform');
endfunction

// Cette fonction definit la force exterieure qui s'exerce sur le systeme.

function ydot=f2(x)
    ydot=-x-cos(x);
endfunction

// Methode d'Euler explicite utilisee pour le debut de nos simulations.

function Y=euler(y0)
    Y=zeros(2,1);
    Y(1,1)=y0(1,1) + y0(2,1)*dt;
    Y(2,1)=y0(2,1)+f2(y0(1,1))*dt;
    if Y(1,1)>1 then
        Y(1,1) = Y(1,1) - 2;           // On periodise en position.
    elseif Y(1,1)< -1 then
        Y(1,1) = Y(1,1)+ 2;
    end
endfunction

// Methode de saute-mouton, utilisee dans le cas lineaire.

function Y=sautemouton(y0)
    Y=zeros(2,1);
    x_int=y0(1,1)+0.5*dt*y0(2,1); // x_int correspond au x_1/2.
    Y(2,1)=y0(2,1)+f2(x_int)*dt; // Nouvelle vitesse.
    Y(1,1)=x_int+0.5*Y(2,1)*dt;  // Nouvelle position.
    if Y(1,1)>1 then
        Y(1,1) = Y(1,1) - 2;           // On periodise en position.
    elseif Y(1,1)< -1 then
        Y(1,1) = Y(1,1)+ 2;
    end
endfunction

// La commande Range nous permet d'avoir des graphiques a echelle constante.

Range=[-1,-2,1,2];
```

```

// La fonction position_inter est la fonction cle de notre code. En faisant
// appel aux fonctions precedentes elle permet de visualiser l'avancee du
// systeme.

function Y=position_inter(Y0,t_f,dt)
    M=floor(t_f/dt); // nombre d'etapes
    N=size(Y0,'c');
    Y=zeros(2,N);
    plot2d(Y0(1,:),Y0(2,:),-4,rect=Range); // Affichage des
                                           // positions/vitesses initiales.

    for j=1:M
        for i=1:N
            Y(:,i)=sautemouton(Y0(:,i)); // On peut remplacer sautemouton par euler.
        end
        clf
        plot2d(Y(1,:),Y(2,:),-4,rect=Range); // Affichage des
                                           // positions/vitesses au temps t=j*dt

        Y0=Y;
    end
endfunction

```

4.2 Le cas non linéaire

// La fonction position permet de repartir uniformement les N particules // dans le carre [0,1]x[0,1] en position et en vitesse. Elle est utilisee // pour les simulations dans le cas repulsif.

```

function Y0=position(N)
    Y0=rand(2,N,'uniform');
endfunction

```

// La fonction position2 permet de repartir uniformement les N particules // dans le rectangle [-0.25,0.25]x[-1,1] en position et en vitesse. Elle est // utilisee pour les simulations dans le cas attractif.

```

function Y0=position2(N)
    Y0=rand(2,N,'uniform') - .5 * ones(2,N);
    A=[.5,0;0,2]
    Y0 = A * Y0
endfunction

```

// Cette fonction sert a periodiser dans l'intervalle [-1,1] les vecteurs en // position.

```

function Y=miseenplace(Y0)
    Y=Y0;
    N=size(Y0,'c');
    for i=1:N
        if Y(1,i)>= 1 then
            Y(1,i) = Y(1,i) - 2;
        elseif Y(1,i)< -1 then
            Y(1,i) = Y(1,i)+ 2;
        end
    end
endfunction

```

```

        end
    end
endfunction

```

```

// Cette fonction nous permet de connaitre le nombre de particules dans
// les differentes bandes de largeur dx.

```

```

function Rho = densite(Y,dx)
    Rho = zeros(1,2/dx);
    N=size(Y,'c');
    for i=1:N;
        j= floor((Y(1,i)+1)/dx) + 1;
        Rho(j)=Rho(j)+1;
    end
    Rho= Rho/(N*dx);
endfunction

```

```

// La fonction Force calcule la force exercee sur les particules situees dans
// les differents bandes de largeur dx.

```

```

function Force = force(Rho,dx)
    d=2/dx;
    Force = zeros(1,d+1);
    c=0;
    for i=2:d
        Force(i)=Force(i-1) + dx*(Rho(i-1)-.5);
        c=c+Force(i);
    end
    Force=Force - c/d;
    Force= - 10*Force;
endfunction

```

```

// La commande Range nous permet d'avoir des graphiques a echelle constante.

```

```

Range=[-1,-2,1,2];

```

```

// La fonction position_inter est la fonction cle de notre code. En faisant
// appel aux fonctions precedentes elle permet de visualiser l'avancee du
// systeme. Pour cela elle utilise la methode numerique de saute-mouton.

```

```

function Y=position_inter(Y0,t_f,dt,dx)
    M=floor(t_f/dt); // nombre d'etapes.
    N=size(Y0,'c');
    Y=zeros(2,N);
    scf(1),clf(1);plot2d(Y0(1,:),Y0(2,:),-4,rect=Range); // Affichage des
// positions/vitesses initiales.
    for j=1:M
        Y0(1,:)=Y0(1,:)+ 0.5*dt*Y0(2,:); // calcul du x_1/2 de saute-mouton.
        Y0=miseenplace(Y0);
        Rho= densite(Y0,dx); // Calcul de la densite et de la force.
        F=force(Rho,dx);
        for i=1:N

```

```

        p= floor((Y0(1,i)+1)/dx) + 1;
        q= ((F(p+1)-F(p))/dx)*(Y0(1,i)-(-1+(p-1)*dx)) + F(p);
        Y0(2,i)=Y0(2,i)+q*dt ;    // Calcul de la nouvelle vitesse.
    end
    Y0(1,:)=Y0(1,:)+ 0.5*dt*Y0(2,:); //Calcul de la nouvelle position
    Y0=miseenplace(Y0);
    scf(1); clf(1)
    plot2d(Y0(1,:),Y0(2,:),-4,rect=Range); // Affichage des
                                        // positions/vitesses au temps t=j*dt
    end
    Y=Y0;
endfunction

```

// Cette fonction inverse les vitesses en conservant les positions. Elle est
// utilisee pour verifier la propriete de reversibilite du systeme.

```

function Y1 = position_inv(Y0)
    N=size(Y0,'c');
    for i=1:N
        Y0(2,i)=-Y0(2,i);
    end
    Y1=Y0;
endfunction

```

// Cette fonction illustre le phenomene de reversibilite.

```

function []=reversibilite(Y0,t_f,dt,dx)
    Y1=Y0;
    Y=position_inter(Y0,t_f,dt,dx);
    clf
    Y0= position_inv(Y);
    Y=position_inter(Y0,t_f,dt,dx);
    Y2=position_inv(Y);
endfunction

```

5 Bibliographie

- X. GOURDON, *Les maths en tête, ANALYSE, seconde édition*, Ellipses
- J.P DEMAILLY *Analyse numérique et équations différentielles*
- H.QUEFFÈLEC, C.ZUILY *Analyse pour l'agrégation*, Dunod
- M. HAURAY, *Equation de Liouville, limites en grand nombre de particules*, disponible à cette adresse : <http://www.latp.univ-mrs.fr/~hauray/Publications/these.pdf>
- F. HÉRAU, *Notes de cours M2, Solutions faibles pour l'équation de Vlasov-Poisson*, que l'on peut consulter à l'adresse : <http://www.math.sciences.univ-nantes.fr/~herau/Docs/Her09-course-M2cinetique-v11.pdf>
- D. HAN-KWAN *Approximation gyrocentrique des plasmas*, disponible à l'adresse suivante : <http://www.fimfa.ens.fr/fimfa/IMG/File/memoires/2008/han-kwan.pdf>