



## RAPPORT DE STAGE

*Deuxième année Centrale Marseille*

2 octobre 2012

Frédéric GAUTARD

**Intitulé du stage :** Etude d'un code simple Vlasov-Poisson.

**Période de stage :** Du 14 juin 2012 au 31 août 2012

**Nom du laboratoire d'accueil :**

CMI (Centre de Mathématiques et d'Informatiques)

Marseille



**Maitre de stage :** Maxime HAURAY, Maître de Conférences de l'université d'Aix-Marseille.

## Résumé

Ce stage de deuxième année que j'ai réalisé dans le laboratoire de recherche du Centre de Mathématiques et d'Informatiques (CMI) de Marseille, du 14 juin au 31 août, sous la direction de M. HAURAY, s'est porté sur l'étude d'un code simple de l'équation de Vlasov-Poisson permettant la modélisation des plasmas. Il avait aussi pour but de me faire connaître le monde de la recherche en mathématiques.

Mon travail s'est donc porté, tout d'abord, sur l'étude des différentes propriétés de cette équation aux dérivées partielles, celle-ci étant utiles pour la suite de notre travail.

Puis, nous avons travaillé sur l'étude et l'amélioration d'un programme *SCILAB*, réalisé par des élèves de master, pour la modélisation des plasmas, par une méthode numérique dite de saute-mouton. Et ainsi nous avons observé des phénomènes particuliers qui ont attirés notre attention, car étant plutôt inattendu. En effet, nous nous sommes concentrés notamment sur des "trous" qui serait dû au puits de potentiel ou encore sur le phénomène d'amortissement Landau.

# Remerciement

Mes premiers remerciements vont tout naturellement à M. Maxime HAURAY, mon tuteur de stage, qui m'a accueilli dans le laboratoire de recherche du CMI et qui m'a accompagné pendant toute cette période. Je le remercie pour sa grande disponibilité et son intérêt pour mon travail.

Je souhaite également remercier les doctorants Thomas, Flore et Nicklas, pour leur accueil chaleureux dans leur bureau ainsi que pour l'aide qu'ils ont pu m'apporter durant ce stage.

Enfin, je voudrais remercier tout le personnel du laboratoire qui a facilité mon arrivée en me donnant libre accès aux installations.

# Mots Clés

Les différents mots clés pouvant décrire mon stage sont les suivants :

- Équation aux dérivée partielle
- Vlasov-Poisson
- Plasma
- Modélisation
- Champs de force
- Puits de potentiel
- Amortissement Landau

# Table des matières

<b>Introduction</b>	<b>3</b>
<b>1 Introduction aux équations de Vlasov</b>	<b>5</b>
1.1 Description de l'équation de Vlasov . . . . .	5
1.2 La Force $F$ . . . . .	5
1.3 D'où vient cette équation ? . . . . .	6
1.4 Propriété de l'équation . . . . .	7
1.4.1 Invariance galiléenne . . . . .	7
1.4.2 Réversibilité . . . . .	8
1.4.3 Conservation de volume . . . . .	9
1.4.4 Conservation de la masse, de l'énergie et du moment . . . . .	10
<b>2 Le code</b>	<b>12</b>
2.1 Méthode numérique de saute-mouton . . . . .	12
2.2 Le programme . . . . .	13
<b>3 Résultat numérique et interprétation</b>	<b>15</b>
3.1 Transport libre . . . . .	15
3.2 Évolution du système . . . . .	15
3.3 Phénomène de "trou" . . . . .	15
3.4 Étude du champ de force . . . . .	17
3.5 Influence du pas de temps sur la force . . . . .	18
<b>Conclusion</b>	<b>19</b>
<b>Bibliographie</b>	<b>21</b>
<b>Annexe</b>	<b>22</b>

# Introduction

Les plasma décrivent un état de la matière constitué de particules chargées (ions et électrons) et composent 99% de l'univers. L'équation de Vlasov-Poisson a été la première à être utilisée pour la description de ces plasmas et elle reste très étudiée en physique des plasmas. Notre étude portera sur l'analyse de la programmation et de certains phénomènes liés à cette équation. Pour cette modélisation, nous nous limiterons à l'étude, certes moins réaliste mais plus accessible, de la dimension 1.

Après avoir décrit cette équation, nous étudierons ces différentes propriétés utiles lors de la résolution.

Dans un second temps, nous présenterons la méthode numérique, et ses caractéristiques, utilisée dans le code.

Enfin, nous exposerons les résultats obtenus avec ce code puis nous chercherons à interpréter les phénomènes observés

# 1 Introduction aux équations de Vlasov

## 1.1 Description de l'équation de Vlasov

Lors de ce stage, nous nous sommes concentrés sur le modèle cinétique de **l'équation de Vlasov** (1.1) dans le cas d'un milieu sans champs magnétique. Les particules étant définies par leur position mais aussi par leur vitesse, il est donc possible, dans ce type de milieu, de trouver plusieurs particules au même endroit mais avec des vitesses différentes. On doit donc aussi prendre en compte la vitesse comme variable de ce système, en plus de la position et du temps.

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

$$\partial_t f + v \cdot \nabla_x f + F \cdot \nabla_v f = 0, \quad (1.1)$$

avec comme condition initiale :  $f(0, x, v) = f_0(x, v)$

Cette équation dépend des trois paramètres classiques d'une équation cinétique :

- **Le temps**  $t$ , défini sur  $\mathbb{R}^+$ .
- **La position**  $x$ , qui peut être définie sur  $\mathbb{R}$ ,  $\mathbb{R}^d$ , un domaine périodique ou bien sur un ouvert borné de  $\mathbb{R}^d$ . Étant donné la complexité du problème en dimension  $n$  et du fait qu'il est nécessaire de connaître les conditions aux limites de notre problème, nous nous sommes placés dans le cas d'un domaine périodique. De plus il est très courant de trouver des plasmas de géométrie torique, donc périodique en position, dans le domaine du nucléaire notamment.
- **La vitesse**  $v$ , défini sur  $\mathbb{R}$  ou  $\mathbb{R}^d$  comme il n'y a pas de raison qu'elles soient bornées. Mais lors de notre étude nous nous limiterons au domaine  $[-2, 2]$

Dans ce modèle, intervient aussi **la densité de particules**  $f(t, x, v)$ , dépendant du temps  $t$ , de la position  $x$  et de la vitesse  $v$ . C'est à dire que  $f(t, x, v) dx dv$  représente le nombre de particules contenues dans le petit élément de volume  $dx dv$ . On suppose que  $f$  reste positive au cours du temps et est d'intégrale égale à 1.

L'étude des ondes dans le plasma est le phénomène le plus important couvert par l'équation de Vlasov. En effet les plasmas sont des milieux considérés comme peu dense car ils contiennent de l'ordre de  $10^8$  à  $10^{15}$  particules, ce qui est petit en comparaison avec le nombre d'Avogadro ( $6,02 \cdot 10^{23} \cdot mol^{-1}$ ). Il est donc possible de trouver plusieurs particules presque au même endroit mais avec des vitesses différentes. De ce fait, le phénomène de collision entre les particules peut être négligé. C'est à dire, que contrairement au cas des fluides, lorsque l'on parle de la vitesse au point  $x$ , on parle que de celle d'une particule, et non de l'ensemble des molécules au voisinage de  $x$  qui possède un mouvement 'collectif' dû aux collisions entre les particules.

Concentrons nous maintenant sur la force  $F$ , qui est un élément fondamental de cette équation et pour l'évolution des particules.

## 1.2 La Force $F$

La force  $F$  peut être de nature très diverse et peut être dû a plusieurs phénomènes différents. Mais ici, nous nous sommes intéressés au cas de la force donnée par l'interaction entre les particules. Dans le cas des plasmas, on obtient **l'équation de Poisson** :

$$-\Delta_x \phi = \rho - \rho_e \quad \text{avec} \quad \rho(t, x) = \int f(t, x, v) dv \quad (1.2)$$

où  $\phi$  est le potentiel électrique et  $\rho$  la densité en position, c'est à dire le nombre de particules contenues dans un intervalle de position de longueur  $dx$ , et cela quelque soit leur vitesse. De plus, le terme  $\rho_e$  est une constante qui décrit le fait qu'il existe des ions et des électrons qui sont plus rapides, car étant de l'ordre de  $10^4$  fois plus légère. Ces particules atteignent donc rapidement leur équilibre et, ainsi, ils sont toujours considérés dans cet état.

De plus, la force  $F$  dérive du potentiel. On a alors l'équation suivante :

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

Pour plus de facilité, nous nous sommes penchés sur le cas de la dimension 1. Le modèle de Vlasov donné par l'équation (1.1) s'écrit alors :

$$\partial_t f + v \partial_x f + F \partial_v f = 0 \quad (1.3)$$

L'équation de Poisson (1.2) devient alors :

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

On a ainsi  $F$  défini à une constante  $C$  près, or, pour avoir un système bien défini, il est nécessaire de connaître une unique valeur. Dans le cas périodique sur l'intervalle  $[-1, 1]$ , on a :

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

c'est à dire que la force  $F$  est à moyenne nulle. Ainsi, on peut la déterminer de manière unique. Avec la méthode suivante, on a, par exemple :

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

d'où, en intégrant de  $-1$  à  $1$

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

et donc, on obtient

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

Nous allons maintenant nous intéresser à la façon dont cette équation est obtenue.

### 1.3 D'où vient cette équation ?

Plaçons nous dans un système à  $N$  particules de positions  $X_1, X_2, \dots, X_N$  et de vitesse  $V_1, V_2, \dots, V_N$  qui interagissent dans un potentiel  $\phi$ , tel que  $F = -\nabla_x \phi$ . Pour que la masse totale soit égale à 1, on considère que la masse de chaque particule vaut  $1/N$ . Ainsi, grâce aux lois de la mécanique classique, on obtient le système d'équations suivantes :

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

Où  $F$  est la force d'interaction entre les particules. Généralement, les plasmas contiennent au minimum  $N = 10^8$  particules, ce qui rend le calcul impossible, à cause du trop grand nombre d'équations qui en découlent. Or, du fait du nombre important de particules présentes dans une unité de volume, on peut affirmer que dans chaque rectangle de coté  $dx$  et  $dv$ , on en trouve une certaine quantité. Et que si on déplace un peu le rectangle, il y a encore environ le même nombre de particules. Ainsi, on peut donc utiliser la densité  $f$ .

Pour comprendre d'où vient cette équation, plaçons nous en dimension 1 dans le rectangle de coté  $dx$  et  $dv$ , qui à l'instant  $t$  contient  $f(t, x, v) dx dv$  particules et demandons nous ce que devient  $f(t + dt, x, v)$  lorsque l'on passe de  $t$  à  $t + dt$ . De plus, dans ce rectangle, on considère que toutes les particules ont la même valeur de vitesse  $v$  et



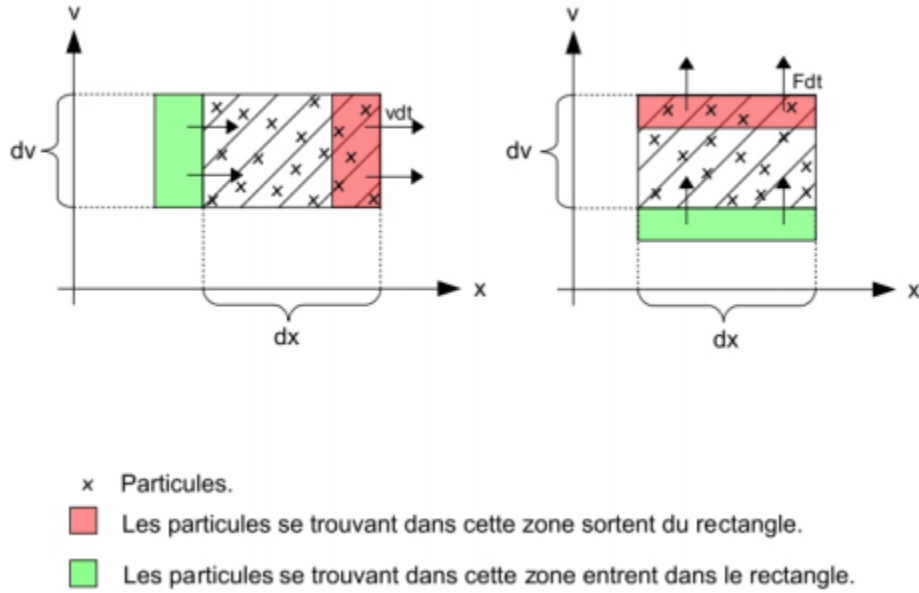


FIGURE 1.1 – Représentation schématique du mouvement des particules

qu'ainsi, pendant un intervalle de  $dt$ , elles se déplacent toute de  $vdt$ . Nous observons donc que certaines particules entrent dans le rectangle et que d'autres en sortent.

Traduisons le mouvement des particules représenté en figure 1.1 avec la densité  $f$

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

D'où :

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

En écrivant que  $f(t + dt, x, v) - f(t, x, v) = \partial_t f(t, x, v)dt$ , cela nous donne :

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

En divisant le tout par  $dx dv dt$ , on obtient l'équation de Vlasov-Poisson :

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

## 1.4 Propriété de l'équation

### 1.4.1 Invariance galiléenne

Le principe de **l'invariance galiléenne**, nous dit que les lois de la physique sont identiques dans tous les référentiels en translations uniforme les uns par rapport aux autres. C'est à dire que si  $f(t, x, v)$  est solution de l'équation de Vlasov (1.3), alors  $f(t, x + x_D + v_D t, v + v_D)$ , avec  $v_D$  vitesse de translation et  $x_D$  le changement d'origine, est encore solution de cette équation.

*Démonstration.* Supposons donc  $f(t, x, v)$  comme une solution de l'équation de Vlasov (1.3). Pour vérifier cette invariance galiléenne, on va effectuer la transformation suivante :

$$T \begin{cases} y = x + x_D + v_D t \\ w = v + v_D \end{cases}$$

Cette transformation, faisant passer de l'élément de volume  $dx dv$  à  $dy dw$ , a un Jacobien égal à 1, ce qui signifie que  $dx dv = dy dw$ . Or lorsque l'on réalise ce changement de variable, le nombre de particules est inchangé dans le petit élément de volume, c'est à dire que l'on a  $f(t, x, v) dx dv = g(t, y, w) dy dw$ . Et ainsi, on retrouve l'égalité suivante :

$$\begin{aligned} g(t, y, w) &= f(t, x, v) \\ \Leftrightarrow (t, x + x_D + v_D t, v + v_D) &= f(t, x, v) \\ \Leftrightarrow g(t, y, w) &= f(t, y - x_D - v_D t, w - v_D) \end{aligned}$$

On peut ainsi exprimer les dérivées partielles de cette fonction  $g(t, y, w)$  à l'aide de celle de  $f$  pour pouvoir l'appliquer à l'équation de Vlasov (1.3). Donc, on a :

$$\begin{cases} \partial_t g(t, y, w) = \partial_t f(t, y - x_D - v_D t, w - v_D) - v_D \partial_x f(t, x, v) \\ \partial_y g(t, y, w) = \partial_x f(t, x, v) \\ \partial_w g(t, y, w) = \partial_v f(t, x, v) \end{cases}$$

Or, en ce qui concerne la densité, on a :

$$\begin{aligned} \rho_g(t, y) &= \int g(t, y, w) dw = \int f(t, x, v) dv = \rho_f(t, x) \\ \Rightarrow \partial_y F_g(t, y) &= \partial_x F_f(t, x) \end{aligned}$$

Or la translation que l'on réalise avec ce changement de variable n'affecte pas les primitives. On obtient donc que :

$$F_g(t, y) = F_f(t, x)$$

Appliquons donc maintenant cette fonction  $g(t, y, w)$  à l'équation de Vlasov :

$$\begin{aligned} \partial_t g(t, y, w) + w \partial_y g(t, y, w) &+ F_g(t, y) \partial_w g(t, y, w) \\ &= \partial_t f(t, x, v) - v_D \partial_x f(t, x, v) (v + v_D) \partial_x f(t, x, v) + F_f(t, x) \partial_v f(t, x, v) \\ &= \partial_t f(t, x, v) + v \partial_x f(t, x, v) + F_f(t, x) \partial_v f(t, x, v) \\ &= 0 \end{aligned}$$

La fonction  $g$  est donc aussi solution de l'équation de Vlasov. Cela prouve donc l'invariance galiléenne de notre équation.  $\square$

### 1.4.2 Réversibilité

Dans cette partie, nous allons voir que notre système est réversible. Pour cela, on inverse l'échelle des temps et la vitesse de nos particules.

On a vu que notre système est régi par les équations suivantes :

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

avec  $f$  solution sur l'intervalle  $[0, T]$ . On définit donc maintenant, la fonction  $g$  que l'on obtient en inversant le temps et les vitesses dans  $f$ , c'est à dire tel que :

$$g(t, x, v) = f(T - t, x, -v) \tag{1.4}$$

Il en résulte ainsi les relations suivantes :

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

De plus, la force  $F$  dépend de la densité de particules et donc du temps. Elle vérifie en effet la relation :

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

Or, si on intègre la fonction  $g$ , donnée en (1.4), par rapport à la vitesse, on obtient :

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

Ainsi,  $f$  et  $g$  ont la même densité et on peut donc définir une nouvelle force  $\tilde{F}(t, x)$  telle que :

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

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

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

car  $f$  est solution de l'équation de Vlasov.

Ainsi, nous obtenons que  $g$  est solution de l'équation de Vlasov-Poisson suivante :

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

### 1.4.3 Conservation de volume

Avec le temps, les petits éléments de volume se déplacent et changent de forme. Et chacun des points de l'espace contenus dans ce petit élément suit une trajectoire qui est celle d'une particule qui se déplace sous l'effet du champ électrique. De plus les équations du mouvement définissent de manière univoque les trajectoires de chaque particule à partir des conditions initiales et donc il est impossible qu'un élément du volume ne dépasse les particules qui sont devant lui. En effet, si tel était le cas, cela impliquerait que deux points différents se confondent et en renversant le temps, on aurait une évolution non univoque. Par conséquent, **le nombre de particules contenues dans un élément de volume est conservé pendant son mouvement**. Ainsi, pour un élément de volume  $dx dv$  et pendant un intervalle de temps  $dt$ , on a :

$$f(t, x, v) dx dv = f(t', x', v') dx' dv'$$

où  $dx' dv'$  est le nouvel élément de volume à l'instant  $t' = t + dt$ , le point  $x, v$  s'étant déplacé au point  $x', v'$  tel que :

$$\begin{aligned} x' &= x + v dt \\ v' &= v + a dt \end{aligned}$$

Le nouvel élément de volume est  $dx' dv' = J dx dv$  où  $J$  est le jacobien de la transformation ci-dessus, c'est à dire :

$$J = \begin{vmatrix} \frac{\partial x'}{\partial x} & \frac{\partial x'}{\partial v} \\ \frac{\partial v'}{\partial x} & \frac{\partial v'}{\partial v} \end{vmatrix}$$

Ainsi, en calculant les éléments de ce déterminant en se limitant au premier ordre en  $dt$  et en prenant compte du fait que  $\partial v / \partial x = 0$  puisque  $x$  et  $v$  sont des variables indépendantes, on obtient :

$$J = 1 + \left( \frac{\partial a}{\partial v} \right) dt = 1 \quad (1.5)$$

Ce résultat  $J = 1$  implique que les éléments de volume  $dx dv$  et  $dx' dv'$  sont égaux et prouve ainsi le fait que l'écoulement dans l'espace des phases est celui d'un milieu incompressible. C'est à dire que l'on peut considérer le volume constant pendant toute la simulation.

### 1.4.4 Conservation de la masse, de l'énergie et du moment

Cette équation à aussi la propriété de conserver la masse. Il nous faut donc prouver que la dérivée par rapport au temps de la masse  $\int f dx dv$  est nulle. Pour cela, partons de l'équation de Vlasov (1.3) en l'écrivant sous forme conservative :

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

En intégrant cette équation selon  $x$  et  $v$ , on obtient :

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

Or l'intégrale par rapport à  $x$  de la dérivée en  $x$  d'une fonction périodique est nulle. Donc, on a  $\int \partial_x(v.f) = 0$  et en utilisant le théorème de Fubini, on obtient aussi que  $\int \partial_v(F.f) = 0$ . Et ainsi on obtient que  $\int \partial_t f dx dv$ . Et donc, cela prouve la conservation de la masse :

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

De plus, on peut montrer aussi la **conservation de l'énergie** au cours de l'évolution. Celle-ci est donnée par :

$$E(t) = \int f(t, x, v) \frac{v^2}{2} dx dv + \frac{1}{2} \int |F|^2 dx \quad (1.6)$$

*Démonstration.* De même, on part de l'équation de Vlasov (1.3) que l'on multiplie par  $\frac{v^2}{2}$  et que l'on intègre selon  $x$  et  $v$  :

$$\underbrace{\int \frac{v^2}{2} \partial_t f dx dv}_A + \underbrace{\int \frac{v^3}{2} \partial_x f dx dv}_B + \underbrace{\int \frac{v^2}{2} F \partial_v f dx dv}_C$$

Ensuite, on va calculer chacun des termes de cette équation séparément

$$A = \partial_t \left( \int \frac{v^2}{2} f dx dv \right)$$

$$B = \int_v \frac{v^2}{2} \int_x \partial_x f dx dv = 0$$

car il s'agit de l'intégrale en  $x$  d'une dérivée par rapport à  $x$  et par périodicité.

$$C = \int_x F(x, t) \int_v \frac{v^2}{2} \partial_v f dv dx$$

Ensuite, en réalisant une intégration par partie où  $\frac{v^2}{2}$  est la fonction à dériver et  $\partial_v f$  à intégrer, on obtient :

$$\begin{aligned} C &= - \int_x F(x, t) \underbrace{\int_v v f dv}_{\text{courant } j(x,t)} dx \\ &= - \int_x F(x, t) j(x, t) dx \end{aligned}$$

Or

$$\begin{aligned} \int_v \partial_t f + v \partial_x f + F \partial_v f dv &= 0 \\ \Leftrightarrow \partial_t \int_v f dv + \partial_x \int_v v f dv &= 0 \\ \Leftrightarrow \partial_t \rho + \partial_x j &= 0 \end{aligned}$$

Donc

$$\begin{aligned} C &= - \int_x \Phi \partial_x j dx \quad \text{par intégration par partie} \\ &= \int_x \Phi \partial_t \rho \\ &= - \int_x F \partial_x \partial_t \rho dx \quad \text{par IPP} \\ &= \int_x F \partial_t F dx \\ &= \int_x \partial_t \frac{|F|^2}{2} dx \end{aligned}$$

On retrouve donc,

$$\partial_t \left( \int f \frac{v^2}{2} dx dv \right) = - \frac{1}{2} \partial_t \int |F|^2 dx$$

□

## 2 Le code

On cherche à approcher  $x(t)$  et  $v(t, x)$  solution du problème suivant, à l'aide d'une méthode numérique :

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

Dans cette partie, nous allons détailler la méthode numérique utilisée dans le programme puis expliciter sa mise en place dans le code, en prenant en compte que dans la suite de notre étude, nous avons considéré que les particules ne se déplacent que selon la direction  $x$ , et que le plasma est périodique sur l'intervalle  $[-1,1]$ . Nous allons aussi expliquer les différentes améliorations apportées à ce programme. De plus, on utilisera les notations suivantes :

- $t\_f$  : le temps final
- $M$  : le nombre d'itérations
- $\Delta t = \frac{t\_f}{M}$  le pas de temps
- $dx$  : le pas d'espace

### 2.1 Méthode numérique de saute-mouton

Pour modéliser le plasma, nous avons utilisé la méthode de Saute-mouton dans le but de simuler les trajectoires de nos points au cours du temps.

La méthode de Saute-mouton consiste à approcher la solution aux instants  $t^n$  du problème de Cauchy mais en réalisant plusieurs étapes pour une même itération. En effet, on calcule tout d'abord la position de la particule au temps  $t + \frac{1}{2}$  et ainsi cela permet d'exprimer sa vitesse au temps  $t + 1$  à l'aide de la force calculée en cette nouvelle position. Ensuite, on retrouve la nouvelle position au temps  $t$  avec la vitesse trouvée à cet instant. Dans le cas de notre schéma, elle s'écrit :

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

L'avantage de cette méthode par rapport à une autre, telle que la méthode d'Euler explicite par exemple, est que celle de Saute-mouton est d'ordre 2. C'est à dire que l'erreur obtenue à chaque opération est plus faible.

Ce code a aussi d'autres propriétés intéressantes pour notre étude.

**Proposition 1.** *Le code de la méthode de saute mouton est réversible*

*Démonstration.* Le code se fait en quatre étapes. Partons, tout d'abord, d'une position  $X^0$  et d'une vitesse  $V^0$  données d'une particule

1.  $X_i^{\frac{1}{2}} = X_i^0 + \frac{1}{2} V_i^0 \Delta t$
2. On calcule ensuite la force  $F_i^{\frac{1}{2}}(X_i^{\frac{1}{2}})$  exercée sur cette particule après un demi pas de temps
3.  $V_i^1 = V_i^0 + F_i^{\frac{1}{2}}(X_i^{\frac{1}{2}}) \Delta t$

$$4. X_i^1 = X_i^{\frac{1}{2}} + \frac{1}{2}V_i^1\Delta t$$

Ensuite, pour vérifier la réversibilité du code, on refait une itération en inversant le temps et les vitesses. C'est à dire que l'on débute le code de la position  $Y_i^0 = X_i^1$  et avec la vitesse  $W_i^0 = -V_i^1$

$$1'. Y_i^{\frac{1}{2}} = Y_i^0 + \frac{1}{2}W_i^0\Delta t = X_i^1 - \frac{1}{2}V_i^1\Delta t = X_i^{\frac{1}{2}}$$

2'. Le calcul de la force est identique car on se trouve dans la même position que précédemment au temps  $t = \frac{1}{2}$ .

$$\text{Donc } G_i^{\frac{1}{2}}(Y_i^{\frac{1}{2}}) = F_i^{\frac{1}{2}}(X_i^{\frac{1}{2}})$$

$$3'. W_i^1 = W_i^0 + G_i^{\frac{1}{2}}(Y_i^{\frac{1}{2}})\Delta t = -V_i^1 + F_i^{\frac{1}{2}}(X_i^{\frac{1}{2}}) = -V_i^0$$

$$4'. Y_i^1 = Y_i^{\frac{1}{2}} + \frac{1}{2}W_i^1\Delta t = X_i^{\frac{1}{2}} - \frac{1}{2}V_i^0\Delta t = X_i^0$$

Après inversion, on retombe bien sur la position initiale de la particule. □

Cela signifie que quelque soit l'état initial de nos particules, il existera toujours un mécanisme pour revenir à notre état de départ. Et cela, même si on part d'un état instable particulier, par exemple dans lequel toutes les particules sont comprises dans un petit espace, et qui atteindra un état désordre stable dans lequel les particules occupent tout l'espace.

## 2.2 Le programme

Pour commencer, on associe aux positions initiales, le matrice  $Y0$  de taille  $2 \times N$ , où  $N$  est le nombre de particules. La première ligne de cette matrice correspond aux positions  $X_i$  tandis que la deuxième contient les vitesses associées  $V_i$ . On a fait en sorte de prendre des conditions initiales connues particulières. Pour cela, nous avons créé deux fonctions nommées **position** et **position\_gaussien** qui placent respectivement les particules de manière uniforme (à l'aide de la fonction *rand*) dans un carré de taille  $[0,1] \times [-1,1]$  et selon une gaussienne centrée en 0.

En outre, d'une manière générale, dans un plasma, la force d'interaction  $F(t,x)$  d'une particule  $X_i$  dépend de la densité de particules autour de celle-ci.

C'est pourquoi, nous avons réutilisé une fonction, nommée **densité**, permettant de calculer le nombre de particules présentes dans une petite bande de largeur  $dx$ . Cette fonction nous renvoie un vecteur de taille  $2/dx$  dont la  $i$ -ème composante donne le nombre de particules comprises dans l'intervalle  $[-1 + (i-1)dx; -1 + idx]$ .

Ainsi, à l'aide de cette nouvelle fonction, on a pu écrire une fonction **force** qui calcule la force  $F_i$  en chaque point de coordonnée  $-1 + idx$ . Rappelons que la force d'interaction est liée par la densité selon l'expression suivante :

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

on a pris pour la suite de nos calcul  $\rho_e = \frac{1}{2}$ . On obtient finalement que :

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

Comme cela a été précisé auparavant, l'espace est périodisé en position. Il nous a donc fallu utiliser une fonction **miseenplace** pour conserver toutes les particules dans l'intervalle  $[-1,1]$ . Il est en effet plus facile d'étudier leur mouvement sur un espace réduit plutôt que sur l'ensemble de l'axe  $Ox$ .

Pour appliquer notre méthode de résolution, nous avons amélioré la fonction **position\_inter** qui au temps  $t + dt$  applique le schéma de saute-mouton à la matrice  $Y$  de taille  $2 \times N$  correspondant au vecteur position et vitesse des  $N$  particules à l'instant  $t$ . Ce programme calcule les positions et les vitesses de chaque particules et à chaque intervalle de temps et nous pouvons ainsi étudier l'évolution du système au cours du temps. Dans cette fonction nous avons intégré des commandes permettant de calculer le champs de force moyen ou la densité moyenne à chaque pas de temps intermédiaire (la densité et la force sont calculées aux demi pas de temps  $t + \frac{1}{2}$ ).

Par la suite nous avons aussi écrit une fonction appelée **position\_inv** qui inverse les vitesses et le temps. Elle nous a été utile pour observer si les erreurs sont importantes lorsque l'on applique la réversibilité à notre système.

Pour la suite de l'étude de notre plasma, nous avons créé deux nouvelles fonctions à ce code :

- la fonction **aller-retour** qui a pour but d'étudier la réversibilité du système en changeant les pas de temps ou d'espace entre l'aller et le retour et en faisant varier aussi le temps final de notre simulation. On peut ainsi comparer l'évolution du champs de force moyen

- la fonction **comparaison\_force** pour regarder l'influence du pas de temps sur la précision de notre système. En effet on fait évoluer le système initial par deux simulations dont une qui a un pas de temps quatre fois plus petit que l'autre.



# 3 Résultat numérique et interprétation

## 3.1 Transport libre

Dans le cas du transport libre, c'est à dire, sans prendre en compte la force d'interaction, on observe un phénomène d'allongement sur tout l'intervalle des positions, les vitesses restant entre -1 et 1. Mais au bout d'un certains temps, on assiste à une homogénéisation. En effet les particules sont réparties de manière uniforme dans tout l'intervalle d'étude. La figure 3.1 représentant l'évolution de la densité des particules dans le cas du transport libre, montre bien les trois phases de la simulation.

C'est un modèle qui est, quand même, très peu réaliste, car on néglige les interactions entre les particules alors que ce phénomène est une des principales raisons des mouvements de particules et donc de l'évolution du plasma.

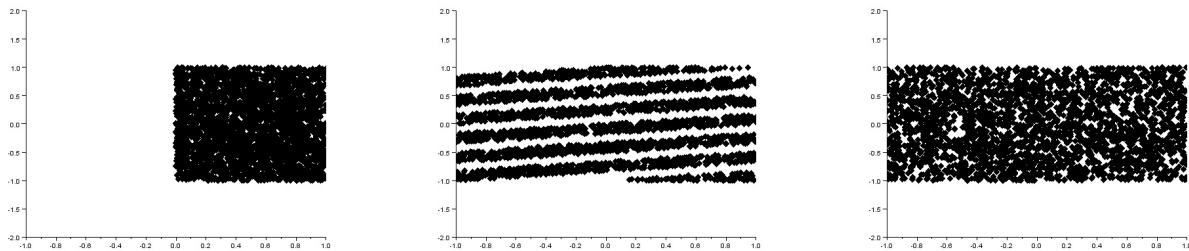


FIGURE 3.1 – Transport libre à  $t=0$ ,  $t=3$  et  $t=10$

## 3.2 Évolution du système

Lors de la simulation de l'évolution des particules dans un plasma, on observe une filamentation, c'est à dire un allongement dans une direction, et cela sans pour autant modifier le volume. Ce phénomène est dû à la force répulsive engendrée par l'interaction entre les particules et par le fait que les particules ne se déplacent pas toute à la même vitesse. Au bout d'un petit moment, le plasma semble s'homogénéiser, c'est à dire que les particules semblent être répartie de façon uniforme dans l'espace. Mais ensuite, on observe un phénomène plutôt inattendu et qui n'existe pas dans le cas du transport libre. En effet, des “trous” sans particules apparaissent au milieu du plasma. Cependant, lorsque l'on renverse le système à l'aide de la fonction *position\_inv*, on remarque que les particules reviennent à leur état initial. Ce résultat s'explique par le fait que la trajectoire d'une particule est unique et particulière. C'est à dire que même si on obtient une configuration homogène en  $x$  qui ne dépend que de la vitesse, il est toujours possible de retrouver cet état particulier en  $x$ .

Les graphiques 3.2 et 3.3 qui précèdent, représentent les différentes phases observées lors de l'évolution d'un plasma contenant 10000 particules, jusqu'au temps final  $t_f = 20$  dans les cas où on a pris un pas de temps  $dt = 0.001$  et un pas d'espace  $dx = 0.01$ . On retrouve la position  $x$  en abscisse et la vitesse  $v$  en ordonnée.

## 3.3 Phénomène de “trou”

Comme on a pu l'apercevoir avec la simulation précédente, on voit apparaitre, à partir d'un certain moment, des trous dans le graphique de la densité. On a tout d'abord cherché à savoir si ce phénomène n'était pas dû aux erreurs

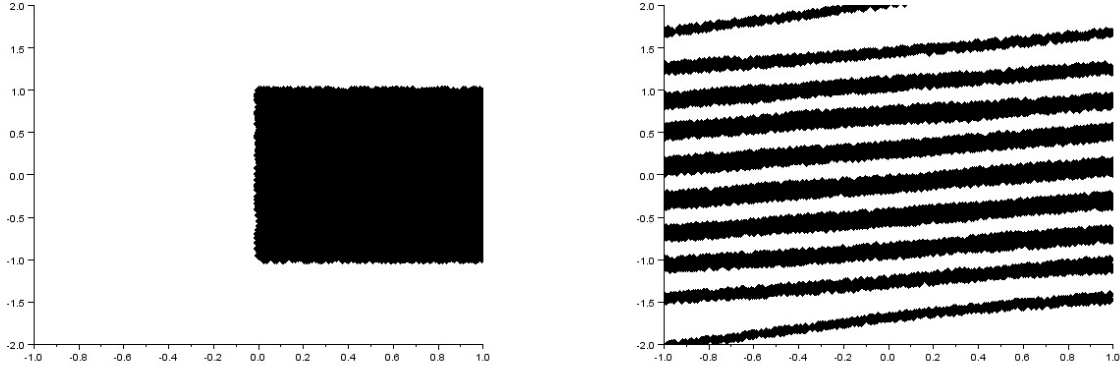


FIGURE 3.2 – Densité à  $t=0$  et à  $t=5$

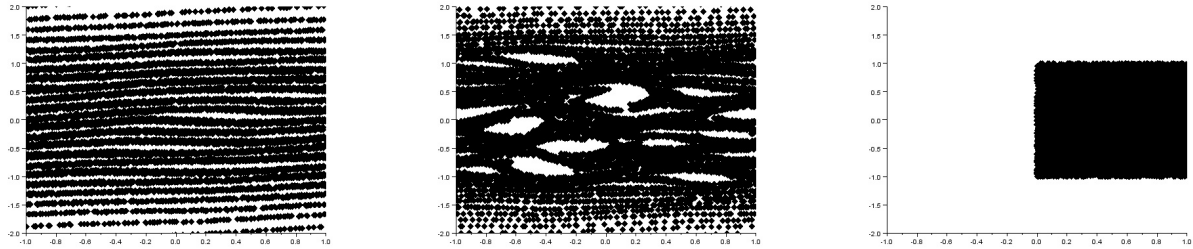


FIGURE 3.3 – Densité à  $t=10$ , à  $t=20$  et après retour

et aux incertitudes de notre simulation. C'est pourquoi nous avons essayé de faire plusieurs simulations différentes, en changeant les pas de temps et d'espace par exemple, mais aussi en augmentant le nombre de particules ou en partant d'une distribution initiale gaussienne.

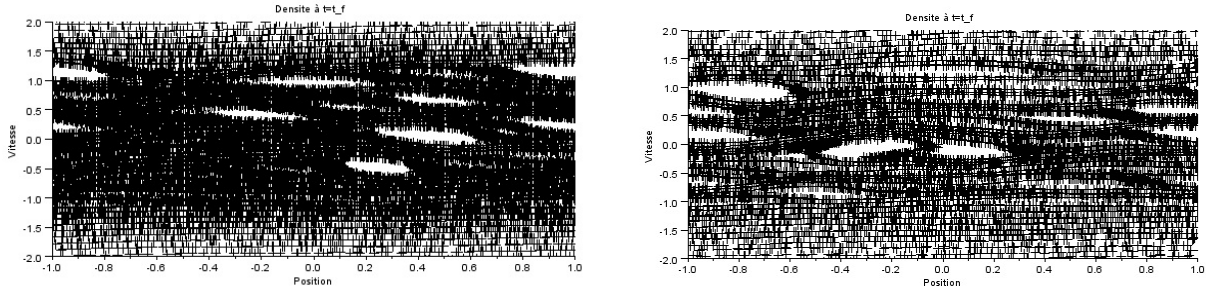


FIGURE 3.4 – densité à  $t=20$  pour 20000 particules et pour une distribution gaussienne

On voit que ce phénomène persiste toujours même s'il est peut être un peu moins important.

Ces trous sont souvent expliqués par le fait qu'il existe des particules piégées dans un puits de potentiel. En effet l'énergie d'un ion est égale à  $E_+ = \frac{1}{2}m_+v^2 + e\phi$  où l'on retrouve la vitesse  $v$ , la masse d'un ion  $m_+$ ,  $e$  la charge et  $\phi$  le potentiel électrique. Si on considère un potentiel de la forme montrée en figure 3.5, les ions sont restreints aux zones telles que  $E_+ > e\phi$ . Ainsi, un ion avec une énergie  $E_+ = e\phi$  est piégé dans la zone C ou F que l'on voit figure 3.5 et ne peut pas passer d'une région à une autre. Les trois points  $x_1$ ,  $x_2$  et  $x_3$  sont des tournants, c'est en ces points que la vitesse des particules s'inversent.

On observe les mêmes phénomènes pour les électrons, mais dans les régions sous la courbe, car l'énergie d'un électron est de  $E_- = \frac{1}{2}m_+v^2 + e\phi$ , qui est donc inférieur à  $e\phi$ .

En fait les particules se "baladent" dans ces régions, c'est à dire que leur vitesse varie dans ces puits de potentiel.

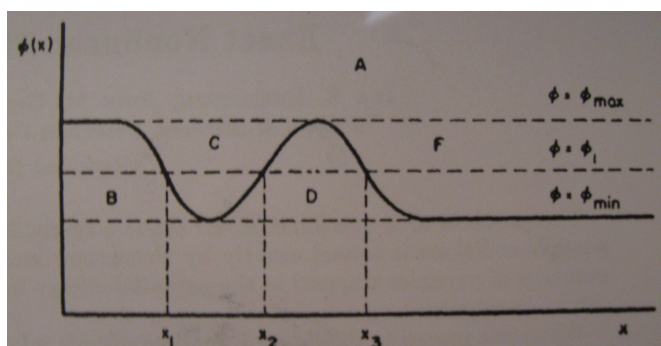


FIGURE 3.5 – Énergie potentielle illustrant les particules piégées (source : Physical Review)

Il y aurait donc des particules de vitesse nulle que l'on retrouve au deux bouts des trous et, autour, d'autres qui accélèrent ou décélèrent. On aurait ainsi un phénomène qui ferait avancer les trous dans le plasma pas rotation des particules autour de celui-ci. On observe effectivement un peu cela entre deux pas de temps.

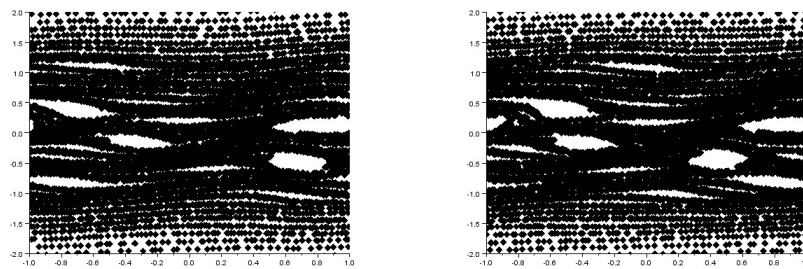


FIGURE 3.6 – Densité à  $t=18$  et à  $t=18.5$

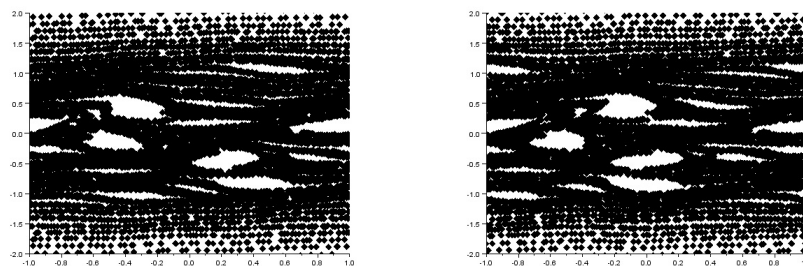


FIGURE 3.7 – Densité à  $t=19$  et à  $t=19.5$

Dans les figures 3.6 et 3.7, on peut observer l'évolution d'un trou de particule au voisinage de  $v = 0.5$ . Il se déplacent de  $x = 0.8$  à  $x = 0$ .

### 3.4 Étude du champ de force

Par la suite, nous nous sommes concentrés sur l'étude de l'évolution du champ de force moyen dans notre système. Nous l'avons donc calculer à chaque intervalle de temps en norme infini dans notre fonction *position\_inter*. On voit apparaître un phénomène d'amortissement du champs de force, celui-ci étant connu sous le nom **amortissement**

**de Landau.** Il est un peu surprenant de retrouver cet amortissement non collisionnel ici étant donné que l'équation de Vlasov-Poisson est réversible alors que l'amortissement d'une onde est un phénomène irréversible auquel on s'attend en présence de collisions

Sur la figure 3.8, on a tracé l'évolution du champs de force moyen dans le cas d'une simulation pour 10000 particules, avec un pas de temps  $dt = 0.0005$  et pas d'espace  $dx = 0.005$ , et jusqu'au temps  $t_f = 10$

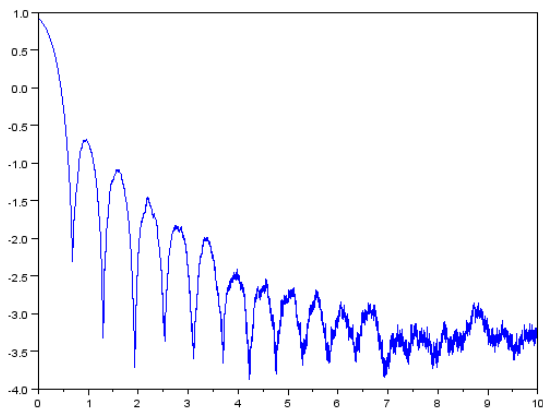


FIGURE 3.8 – Évolution du champs de force en échelle logarithmique

Ce phénomène d'amortissement de Landau peut s'expliquer dans notre cas à l'aide de la densité. En effet, lors de la filamentation, il y a des instants où les deux bouts de ce "filament" sont à la même position  $x$  ce qui fait qu'il y a, à peu près, le même nombre de particules dans tous les intervalles et donc la force  $F$  est faible. Alors que dans le cas où les particules sont moins bien réparties, c'est à dire, quand la position des deux bouts ne correspondent pas, il y a des pics au niveau du champs de force. C'est pour cela que l'on peut remarquer qu'il y a autant de pic à un instant  $t$  dans l'amortissement Landau qu'il y a de fil de particules dans la densité à ce même instant  $t$ .

De plus on a voulu étudier l'influence du temps final sur la réversibilité de l'équation, notamment pour voir si ce phénomène de particules piégées influencées sur cette propriété caractéristique de l'équation. On a pour cela réalisé une simulation à l'aide de notre fonction *aller\_retour* avec un pas de temps retour différent du premier, et de sorte qu'il existe pas d'instant  $t$  commun aux deux simulations. Cela permet d'étudier plus la réversibilité du système plutôt que celle du programme. Les caractéristiques de la modélisation ici étudié sont  $dt1 = 0.037$ ,  $dt2 = 0.01$ , des pas d'espace égaux  $dx = dx2 = 0.01$  et cela lorsque l'on fait évoluer un plasma contenant 10000 particules.

On peut déjà remarquer qu'il y a plusieurs phases dans l'évolution du champs de force. En effet à partir du moment où il y a homogénéisation, l'amortissement Landau disparaît et ensuite, au moment de l'apparition des trous, il y a une augmentation de ce champs de force et on n'observe plus grand chose de particulier. On peut ensuite voir sur la figure 3.9 que malgré la présence de quelques trous, la réversibilité se fait plutôt bien a part pour certaines particules et que même l'amortissement Landau se réalise lors de la réversibilité. Mais dès lors que l'on augmente le temps de la simulation, comme sur la figure 3.10, le désordre est trop important pour réussir à revenir à notre rectangle initiale de particules.

### 3.5 Influence du pas de temps sur la force

Par la suite, nous avons comparer pour deux simulations ayant les mêmes conditions initiales, mais des pas de temps différents, l'évolution de l'erreur entre ces deux systèmes, et cela à l'aide de nos fonctions *position\_inter\_double* et *comparaison\_force*. Nous avons choisi de prendre une simulation dont le pas de temps est quatre fois plus faible que pour l'autre. Dans la suite nous allons noter  $F1$  le vecteur force de la simulation avec le pas de temps  $dt$  et  $F2$  celui pour laquelle on a choisit un intervalle de temps  $dt/4$ .

Ces programmes nous ont donc permis de relever à chaque intervalle de temps les forces  $F1$  et  $F2$  et ainsi on a pu tracer l'évolution de l'erreur moyenne entre les deux forces, le champs de force de  $F2$  (la simulation 2 étant la

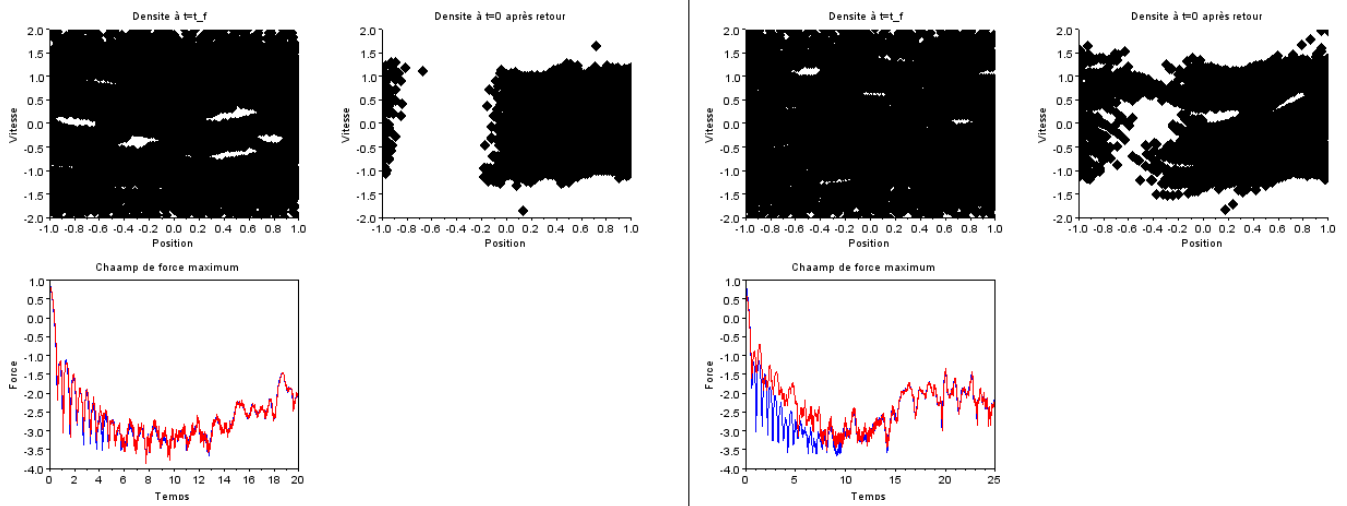


FIGURE 3.9 – Densité et champs de force à  $t=20$  et  $t=25$

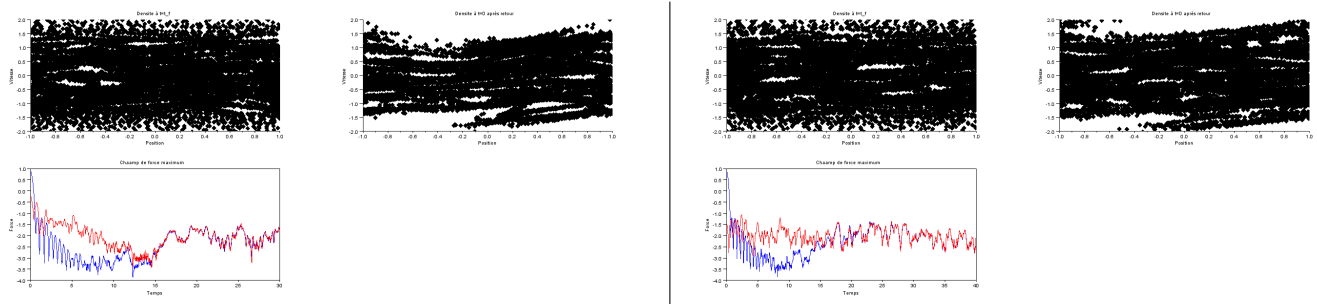


FIGURE 3.10 – Densité et champs de force à  $t=30$  et  $t=40$

plus précise), la variation de l'erreur absolue et ainsi que l'évolution du gradient de cette force  $F_2$ , i.e. la densité, au cours du temps.

Ainsi pour une distribution initiale uniforme ou gaussienne de 10000 particules, on obtient les graphiques 3.11 et 3.12.

Tout d'abord, on observe toujours, dans les deux cas, le phénomène d'amortissement de Landau. Ensuite, on peut voir que l'erreur entre les deux courbes est plutôt faible, surtout dans les premiers instants de la simulation, car étant de l'ordre des  $2.10^{-4}$  ou  $5.10^{-4}$ . Il faut préciser que le petit saut à  $t=1$  sur l'erreur est dû au fait que, pour avoir plus de précision sur les premiers instants, on a encore divisé le pas de temps par 4 dans les deux simulations. Mais, lorsque l'on regarde l'erreur relative, elle possède des pics qui peuvent monter jusqu'à 10-15%, une erreur plutôt élevée dû à l'amortissement de Landau de la force  $F_2$ . On peut aussi remarquer que bien que l'écart moyen entre les deux forces augmentent au cours du temps, notamment à partir de l'apparition des trous, l'erreur relative reste autour des 5%. L'erreur relative étant plutôt faible au moment des trous, on peut en déduire que le manque de précision du pas de temps n'est pas à l'origine de ce phénomène. Cela ne viendrait donc pas du programme mais serait bien réel.

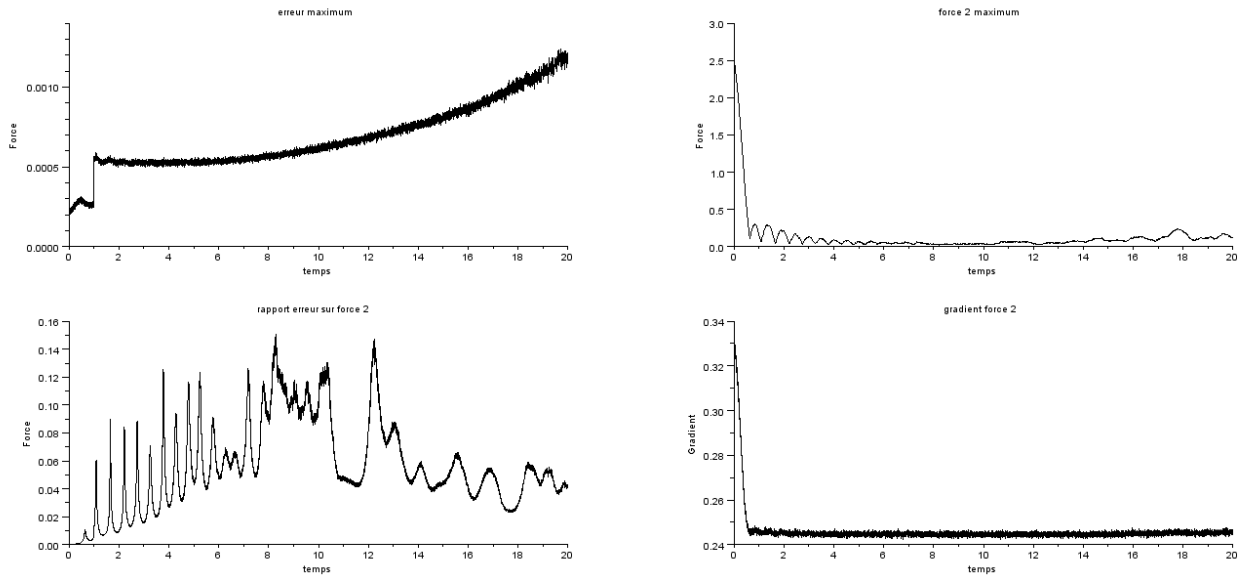


FIGURE 3.11 – Graphique sur les Forces dans le cas uniforme avec  $dt=0.001$ ,  $dx=0.001$  et  $t\_f=20$

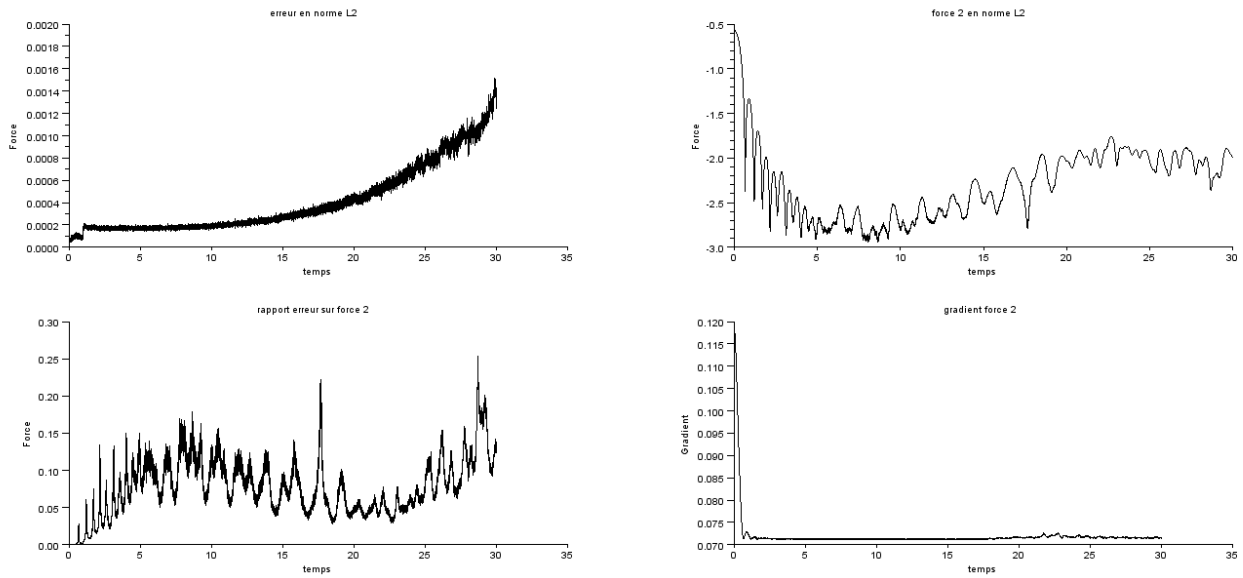


FIGURE 3.12 – Graphique sur les Forces dans le cas gaussien avec  $dt=0.001$ ,  $dx=0.01$  et  $t\_f=30$

# Conclusion

Ce stage au Centre de Mathématiques et d'Informatiques m'a permis de connaître le monde particulier de la recherche. En effet, après la découverte de l'entreprise en première année, il me paraissait intéressant de pouvoir voir aussi l'activité en laboratoire de recherche, et notamment dans le domaine des mathématiques. Il m'a aussi permis de voir comment les recherches sur des Équations aux Dérivées Partielles se déroulaient au CMI.

Au cours de ce stage, je me suis donc concentré sur l'étude et la modélisation des plasmas, dans le but d'y observer et de comprendre des phénomènes un peu inattendu. Je me suis donc notamment concentré sur l'apparition de ces "trous" certainement dû à des puits de potentiel ou encore sur le phénomène d'amortissement de Landau.

# Bibliographie

- J.L. DELCROIX et A. BERS, *Physique des plasmas 2*.
- IRA B. BERNSTEIN, JOHN M. GREENE et MARTIN D. KRUSKAL, Physical Review, *Exact Nonlinear Plasma Oscillations*, Volume 108, number 3.
- M. HAURAY, *cours sur les EDO et les equations de Transport*.



# Annexe

Voici le code que l'on a utilisé lors de ce stage :

```
Range=[-1,-2,1,2]; // [xmin,ymin,xmax,ymax] les coord // du point en bas a gauche et en haut a droite du
```

```
function Y0=position(N) // position initiale
    Y0=rand(2,N,'uniform');
    Y0(2,:)=2*Y0(2,)-1
endfunction
```

```
function Y0=position_gaussien(N) // position initiale gaussienne
    Y0=grand(2,N,"nor",0,0.2);
    Y0(2,:) = 2*Y0(2,:)
    Y0=miseenplace(Y0)
endfunction
```

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

```
function Y=miseenplace(Y0) // Fonction pour periodiser en position.
    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
```

```
function Rho = densite(Y,dx) // fonction pour calculer la densité
    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

```

```

function Force = force(Rho,dx)    // fonction pour calculer la force
    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

```

```

function G=grille(Y,dx)    // quadrille le plan (x,v)
    G=zeros(4/dx,2/dx);
    N=size(Y,'c');
    for i=1:N;
        if Y(2,i) >= -2 & Y(2,i) <= 2 then
            l= floor((Y(1,i)+1)/dx) + 1;
            k= floor((Y(2,i)+2)/dx) + 1;
            G(k,l)=G(k,l)+1;
        end
    end
endfunction

```

```

function [Y,Rho,F,Champ,Densite,dtm]=position_inter(Y0,t_f,dt,dx)
    M=floor(t_f/dt); // nombre d'etapes
    dtm=t_f/M // pas de temps exacte
    N=size(Y0,'c');
    Y=zeros(2,N);
    Champ_moy=[];
    Champ_max=[];
    Densite_moy=[];
    Densite_max=[];
    F=zeros(1,2/dx+1);
    Rho=F;
//    plot2d(Y0(1,:),Y0(2,:),-4,rect=Range); // nuage de points a t=0
    DX=-1:dx:1
    DT=0:dtm:t_f
    DTinter=dtm/2:dtm:(M-1/2)*dtm // pas de temps intermédiaire (à chaque demi pas de temps)
    for j=1:M
        Y0(1,:)=Y0(1,)+ 0.5*dtm*Y0(2,:); // calcul du x_intermediaire
        Y0=miseenplace(Y0); // on periodise
        Rho= densite(Y0,dx); // calcul de la densité au x_intermediaire
        F=force(Rho,dx); // Force au x_intermediaire
        Champ_moy=[Champ_moy,N^(-.5)*norm(F)] // champ de force moyen en norme L2
        Champ_max=[Champ_max,max(abs(F))] // champ de force en norme infini
        Densite_moy=[Densite_moy,norm(Rho-1/2)]; // moyenne de la densité en norme L2
        Densite_max=[Densite_max,max(abs(Rho-1/2))]; // moyenne de la densité en norme infini
        for i=1:N
            // Calcul de la nouvelle vitesse
            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*dtm ;
    end
    Y0(1,:)=Y0(1,:)+ 0.5*dtm*Y0(2,:); //Calcul de la nouvelle position
    Y0=miseenplace(Y0);
end
Champ=zeros(2,M)
Champ(1,:)=Champ_moy;
Champ(2,:)=Champ_max;
Densite=zeros(2,M)
Densite(1,:)=Densite_moy;
Densite(2,:)=Densite_max;
//      scf(3),clf(3)
//      plot(DTinter,log(Champ_max))
    Y=Y0;
endfunction

```

```

////////////////////////////////////
//
//
//      ALLER RETOUR
//
//
////////////////////////////////////

```

```

function []=aller_retour(Y0,t_f,dt,dt2,dx,dx2)
    [Y1,Rho1,F1,Champ1,Densite1,dtm]=position_inter(Y0,t_f,dt,dx);
    Y2=position_inv(Y1);
    [Y3,Rho3,F3,Champ3,Densite3,dtm2]=position_inter(Y2,t_f,dt2,dx2); // simulation inverse
    N=size(Y0,'c')
    M=floor(t_f/dtm);
    M2=floor(t_f/dtm2);
    DTinter=dtm/2:dtm:(M-1/2)*dtm;
    DTinter2=dtm2/2:dtm2:(M2-1/2)*dtm2;

    save('data aller retour',Y0,Y1,Y3,Rho1,Rho3,F1,F3,Champ1,Champ3,Densite1,Densite3,M,M2,DTinter,DTinter2);
    clf(1),Graphe=scf(1)
    cmap=graycolormap(32)
    set(Graphe,"color_map",cmap) // changement de la colormap de la fenetre
    subplot(221)
    //plot2d(Y0(1,:),Y0(2,:),-1,rect=Range)
    G0=grille(Y0,0.02);
    G01=(G0./max(G0)).*32; // normalisation pour les 32 couleurs de la colormap
    [rg,cg]=size(G01);
    for i=1:rg
        G02(i,:)=G01(rg- i+1,:);
    end
    Matplot(G02) // commande pour tracer un graphe en couleur en fonction du nombre de particules contenues
    xtitle('Densite à t=0','Position','Vitesse')
    subplot(222)

```

```

// plot2d(Y1(1,:),Y1(2,:),-1,rect=Range)
G1=grille(Y1,0.02);
G11=(G1./max(G1)).*32;
[rg,cg]=size(G11);
for i=1:rg
    G12(i,:)=G11(rg- i+1,:);
end
Matplot(G12)
xlabel('Densite à t=t_f','Position','Vitesse')
subplot(223)
//plot2d(Y3(1,:),Y3(2,:),-1,rect=Range)
G3=grille(Y3,0.02)
G31=(G3./max(G3)).*32;
[rg,cg]=size(G31);
for i=1:rg
    G32(i,:)=G31(rg- i+1,:);
end
Matplot(G32)
xlabel('Densite à t=0 après retour','Position','Vitesse')
subplot(224)
plot(DTinter,log10(Champ_moy1))
plot(t_f-DTinter2,log10(Champ_moy3),'r')
xlabel('Champ de force moyen','Temps','Force')
f=gcf();
f.figure_size=[768,768]
nom_image='AR_Matplot_N'+string(N)+'_Tf='+string(t_f)+'_dt1='+string(dt)+'_dt2='+string(dt2)+'_dx='+s
xs2png(gcf(),nom_image); //sauvegarde de l'image sous format .png

clf(2),scf(2)
subplot(221)
plot2d(Y0(1,:),Y0(2,:),-1,rect=Range) // nuage de point a t=0
xlabel('Densite à t=0','Position','Vitesse')
subplot(222)
plot2d(Y1(1,:),Y1(2,:),-1,rect=Range) // nuage de point au temps final
xlabel('Densite à t=t_f','Position','Vitesse')
subplot(223)
plot2d(Y3(1,:),Y3(2,:),-1,rect=Range)
xlabel('Densite à t=0 après retour','Position','Vitesse') // // nuage de point a t=0 après inversion
subplot(224)
plot(DTinter,log10(Champ_moy1)) // évolution du champ de force moyen a l'aller
plot(t_f-DTinter2,log10(Champ_moy3),'r') // évolution du champ de force moyen au retour
xlabel('Champ de force moyen','Temps','Force')

f=gcf();
f.figure_size=[768,768]
// f.pixmap = "on" // set pixmap status to on
nom_image2='AR_Croix_N'+string(N)+'_Tf='+string(t_f)+'_dt1='+string(dt)+'_dt2='+string(dt2)+'_dx='+s
xs2png(gcf(),nom_image2);
endfunction

////////////////////////////////////
//

```

```

//
// COMPARAISON FORCES
//
//
////////////////////////////////////

function [Y1,Y2,e,Champ_moy1,Champ_moy2,Champ_max1,Champ_max2,Densite_moy1,Densite_moy2,Densite_max1,Densite_max2] =
    e=[];
    r=[];
    grad2=[]
    M=floor(t_f/dt); // nombre d'etapes
    dtm=t_f/M
    N=size(Y1,'c');
    Champ_moy1=[];
    Champ_max1=[];
    Champ_moy2=[];
    Champ_max2=[];
    Densite_moy1=[];
    Densite_max1=[];
    Densite_moy2=[];
    Densite_max2=[];
    // scf(1),clf(1);plot2d(Y0(1,:),Y0(2,:),-4,rect=Range); // nuage de points a t=0
    DX=-1:dx:1
    DT=0:dtm:t_f
    DTinter=dtm/2:dtm:(M-1/2)*dtm
    for j=1:M
        Y1(1,:)=Y1(1,:)+ 0.5*dtm*Y1(2,:); // calcul du x_intermediaire avec un pas de temps de dt
        Y1=miseenplace(Y1); // on periodise
        Rho1= densite(Y1,dx);
        F1=force(Rho1,dx);
        Champ_moy1=[Champ_moy1,N^(-.5)*norm(F1)]
        Champ_max1=[Champ_max1,max(abs(F1))];
        Densite_moy1=[Densite_moy1,norm(Rho1-1/2)]
        Densite_max1=[Densite_max1,max(abs(Rho1-1/2))]
        for i=1:N // Calcul de la nouvelle vitesse
            p= floor((Y1(1,i)+1)/dx) + 1;
            q= ((F1(p+1)-F1(p))/dx)*(Y1(1,i)-(-1+(p-1)*dx)) + F1(p);
            Y1(2,i)=Y1(2,i)+q*dtm ;
        end
        Y1(1,:)=Y1(1,:)+ 0.5*dtm*Y1(2,:); //Calcul de la nouvelle position
        Y1=miseenplace(Y1);
        for k=1:4 // meme evolution mais avec un pas de temps 4 fois plus faible
            dtm2=dtm/4
            Y2(1,:)=Y2(1,:)+ 0.5*dtm2*Y2(2,:); // calcul du x_intermediaire
            Y2=miseenplace(Y2); // on periodise
            Rho2= densite(Y2,dx);
            F2=force(Rho2,dx);
            Champ_moy2=[Champ_moy2,N^(-.5)*norm(F2)]
            Champ_max2=[Champ_max2,max(abs(F2))];
            Densite_moy2=[Densite_moy2,norm(Rho2-1/2)]
            Densite_max2=[Densite_max2,max(abs(Rho2-1/2))]
            for i=1:N // Calcul de la nouvelle vitesse
                p= floor((Y2(1,i)+1)/dx) + 1;
            end
        end
    end
end

```

```

        q= ((F2(p+1)-F2(p))/dx)*(Y2(1,i)-(-1+(p-1)*dx)) + F2(p);
        Y2(2,i)=Y2(2,i)+q*dtm2 ;
    end
    Y2(1,:)=Y2(1,:)+ 0.5*dtm2*Y2(2,:);
    Y2=miseenplace(Y2);
end
e=[e,N^(-.5)*norm(F1-F2)]; // evolution de l'erreur moyenne (en norme L2)
r=[r,norm(F1-F2)/norm(F2)]; // evolution de l'erreur relative (en norme L2)
grad2=[grad2,N^(-.5)*norm(Rho2)]; // evolution de la moyenne du gradient (en norme L2)
end
endfunction

```

```

function []=comparaison_force(Y0,t_f,dt,dx)
    e=[];
    r=[];
    grad2=[];
    Y1=Y0;
    Y2=Y0;
    N=size(Y0,'c')
    [Y1,Y2,e_i,Champ_moy1_i,Champ_moy2_i,Champ_max1_i,Champ_max2_i,Densite_moy1_i,Densite_moy2_i,Densite_moy1_f,Densite_moy2_f,Densite_max1_f,Densite_max2_f]
    M2=floor(16/dt);
    DTinter2=dt/32:dt/16:(M2-1/2)*dt/16;
    [Y1,Y2,e_f,Champ_moy1_f,Champ_moy2_f,Champ_max1_f,Champ_max2_f,Densite_moy1_f,Densite_moy2_f,Densite_max1_f,Densite_max2_f]
    t_e=[dt/4:dt/4:1,1+dt:dt:t_f]
    M2=floor(4*(t_f-1)/dt);
    DTinter2=[DTinter2,1+dt/8:dt/4:1+(M2-1/2)*dt/4];

    e=[e_i,e_f];
    r=[r_i,r_f];
    Champ_moy1=[Champ_moy1_i,Champ_moy1_f];
    Champ_moy2=[Champ_moy2_i,Champ_moy2_f];
    Champ_max1=[Champ_max1_i,Champ_max1_f];
    Champ_max2=[Champ_max2_i,Champ_max2_f];
    grad2=[grad2_i,grad2_f];

    save('data comparaison force',Y1,Y2,t_f,dt,dx,t_e,DTinter2,e,r,Champ_moy1,Champ_moy2,Champ_max1,Champ_max2);
    clf(1),scf(1)
    subplot(221)
    plot2d(t_e,e) // graphe de l'evolution de l'erreur
    xtitle('erreur en norme L2','temps','Force')
    subplot(222)
    plot2d(DTinter2,log10(Champ_moy2)) // graphe de l'evolution du champs de force de la simulation avec 1
    xtitle('force 2 en norme L2','temps','Force')
    subplot(223)
    plot2d(t_e,r) // graphe de l'evolution de l'erreur relative
    xtitle('rapport erreur sur force 2','temps','Force')
    subplot(224)
    plot2d(t_e,grad2) // graphe de l'evolution du gradient
    xtitle('gradient force 2','temps','Gradient')

    nom_image='Force_'+string(N)+'_Tf='+string(t_f)+'_dt='+string(dt)+'_dx='+string(dx)+'.png';
    xs2png(gcf(),nom_image); // sauvegarde de l'image au format .png
endfunction

```