

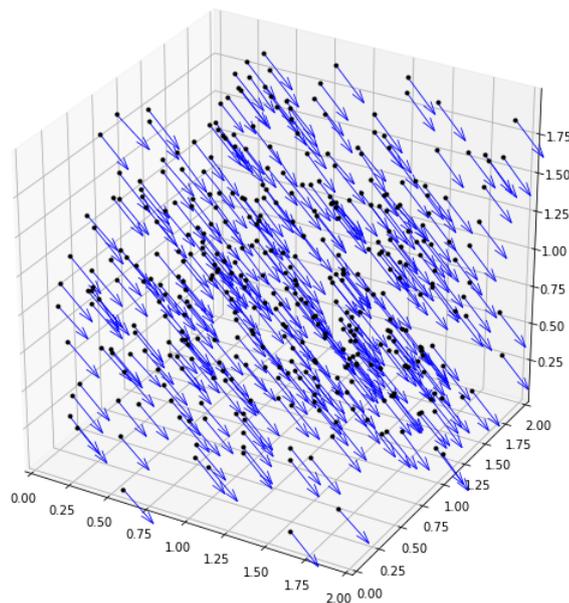
Master 1 Calcul Haute Performance, Simulation  
2018/2019

Travail Encadré de Recherche

**Systèmes de particules en interaction et  
équation d'agrégation**

Présenté par :

Alexandre Rotolo & Maydine Ghestin



Encadrant : Thomas Rey

# Table des matières

<b>1</b>	<b>Modèles particuliers</b>	<b>4</b>
1.1	Modèle de Champ Moyen . . . . .	4
1.1.1	Implémentation . . . . .	5
1.1.2	Propriétés . . . . .	6
1.1.3	Résultats . . . . .	10
1.2	Modèle de Cucker-Smale . . . . .	12
1.2.1	Implémentation . . . . .	13
1.2.2	Propriétés . . . . .	14
1.2.3	Résultats . . . . .	18
1.3	Modèles mixtes . . . . .	22
1.3.1	Implémentation . . . . .	23
1.3.2	Résultats . . . . .	28
<b>2</b>	<b>Modèle de densité : Équation d'agrégation</b>	<b>32</b>
2.1	Résultat théorique . . . . .	32
2.2	Représentation par histogrammes . . . . .	33
2.3	Comparaison et conclusions . . . . .	34
2.3.1	Comparaison . . . . .	34
2.3.2	Conclusion . . . . .	40

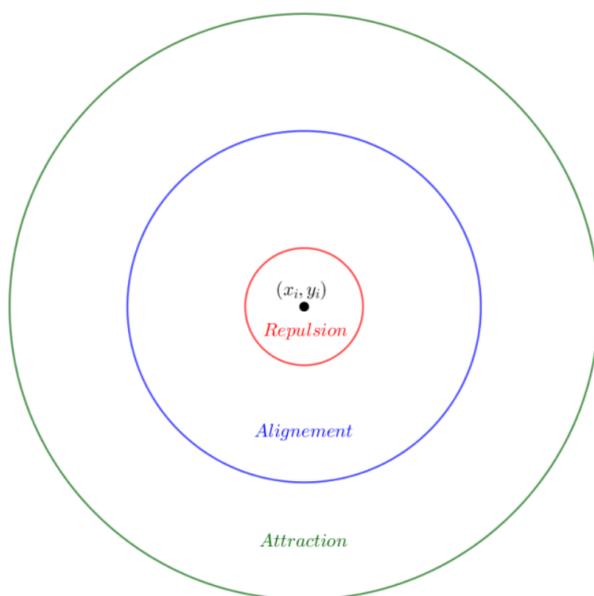
# Introduction

Dans de nombreux domaines tels que la biologie, les mathématiques, la physique ou l'ingénierie, des travaux sont réalisés afin de modéliser et prévoir des interactions entre individus vues comme résultantes de comportements individuels. Il est possible d'illustrer très simplement de tels phénomènes : nuées d'oiseaux, bancs de poissons, micro-organismes, robots interagissant à courte distance.

Le but de ces travaux est, en faisant évoluer un système au niveau microscopique, i.e. à l'échelle de chaque particule, d'obtenir un phénomène visible à l'échelle macroscopique : l'émergence d'un comportement collectif. Ce type de comportement, en apparence difficile à modéliser, est donc sujet à de nombreuses interprétations, donnant lieu à différentes approches et modèles descriptifs. Les paramètres pouvant intervenir dans ces modèles comme la prise en compte du milieu ou la présence d'obstacles à éviter permettent de modéliser nombre de comportements, cela étant expliqué plus en profondeur dans [3].

L'exemple classique d'un modèle descriptif est construit de façon relativement simple, il décrit les interactions selon trois zones :

- Si deux individus sont trop proches, ils préféreront avoir un espace personnel -> Répulsion.
- Si un individu est trop éloigné du groupe, il tiendra à s'en rapprocher -> Attraction.
- Chaque individu cherche à imiter ses congénères -> Orientation/Alignement.



Les recherches mathématiques à ce sujet sont tout à fait d'actualité et continuent de progresser, le champ des possibles étant très étendu : groupes de véhicules autonomes, modélisation et prédiction pour l'industrie de la pêche, pour la sécurité...

En fonction du phénomène dont nous chercherons à étudier le comportement, nous utiliserons différentes approches, à l'aide notamment de différents potentiels d'interaction proposés par [1], [2], [4] ou encore [6]. Nous effectuerons cette étude dans la première partie de ce TER, afin de mettre en évidence l'émergence de comportements collectifs à partir de considérations particulières.

Dans la deuxième partie de notre travail, nous serons amenés à implémenter l'équation de transport non locale suivante, appelée équation d'agrégation :

$$\frac{\partial \rho}{\partial t} - \nabla_x \cdot (\rho(\nabla W * \rho)) = 0,$$

afin d'illustrer des résultats de convergence des modèles particulières de notre première partie vers la solution de cette équation lorsque le nombre de particules étudiées devient très grand ([2],[3],[6]).

# Chapitre 1

## Modèles particuliers

Dans la suite, nous décrirons mathématiquement chaque particule étudiée de la manière suivante : Un individu  $i \in 1, \dots, N$  est décrit comme une particule massique, située au temps  $t \geq 0$  à la position  $(x_i, y_i)$  et se déplaçant à la vitesse  $(Vx_i, Vy_i)$ .

### 1.1 Modèle de Champ Moyen

Dans ce premier modèle, nous considérons les particules interagissant deux à deux grâce à un potentiel d'interaction  $W : \mathbb{R}^2 \mapsto \mathbb{R}$  et leurs couples position/vitesse sont soumis aux lois de Newton :

$$(MF) \quad \begin{cases} x'_i(t) = v_i(t), \\ v'_i(t) = \frac{1}{N} \sum_{j=1}^N \nabla W(x_i(t) - x_j(t)), \end{cases}$$

où nous considérons l'abus de notation suivant :  $x_i(t) := \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix}$  et  $v_i(t) := \begin{pmatrix} Vx_i(t) \\ Vy_i(t) \end{pmatrix}$ .

Nous admettrons que le potentiel d'interaction  $W$  peut être décrit comme

$$W(x) = k(\sqrt{x^2 + y^2}) = k(r),$$

où  $k \in \mathcal{C}^1(\mathbb{R}_+, \mathbb{R})$ ,  $k(0) = 0$ .

Nous exprimons ainsi le fait que  $W$  soit une fonction radiale (symétrique en 1D), ce qui, d'un point de vue de la modélisation, signifie simplement que les particules n'ont pas de direction d'interaction privilégiée. La condition  $k(0) = 0$  traduit quant à elle le fait qu'une particule ne peut avoir d'interaction avec elle-même.

### 1.1.1 Implémentation

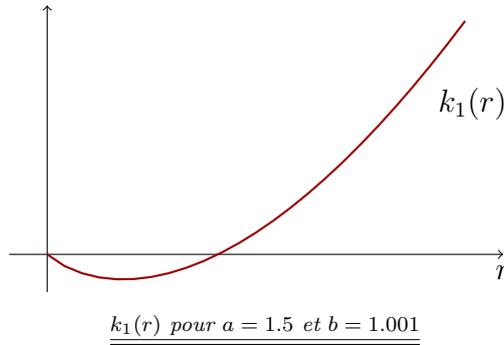
On utilise la méthode des différences finies, via le schéma d'Euler explicite, afin d'obtenir le système à implémenter :

$$\begin{cases} \frac{x_i(t + \Delta t) - x_i(t)}{\Delta t} = v_i(t), \\ \frac{v_i(t + \Delta t) - v_i(t)}{\Delta t} = \frac{1}{N} \sum_{j=1}^N \nabla W(x_i(t) - x_j(t)). \end{cases}$$

$$\Rightarrow \begin{cases} x_i(t + \Delta t) = \Delta t v_i(t) + x_i(t), \\ v_i(t + \Delta t) = \Delta t \frac{1}{N} \sum_{j=1}^N \nabla W(x_i(t) - x_j(t)) + v_i(t). \end{cases}$$

Nous considérerons pour l'implémentation le potentiel suivant, appelé potentiel de loi de puissance, proposé dans [4] :

$$k_1(r) = \frac{r^a}{a} - \frac{r^b}{b}, \quad a > b > 0.$$



Ce potentiel, si nous nous référons au schéma d'interaction à trois zones, ne prend en compte que deux de ces zones :

- La zone de répulsion, où la puissance de la répulsion est caractérisée par le paramètre  $b$ .
- La zone d'attraction, où la puissance de celle-ci est caractérisée par le paramètre  $a$ .

Ce qui explique la condition  $a > b > 0$  : Si la répulsion est plus puissante que l'attraction, il y aurait directement éparpillement des particules.

## 1.1.2 Propriétés

Pour simplifier les notations, nous omettrons souvent la dépendance en  $t$  des particules, ainsi  $x_i \equiv x_i(t)$  et  $v_i \equiv v_i(t)$ .

Propriété 1 :

Le schéma d'Euler explicite pour la résolution de (MF) est consistant d'ordre 1 et stable, donc il converge.

*Démonstration.*

Commençons par montrer la consistance du schéma :

Soit  $u$  dérivable, un développement de Taylor à l'ordre 1 nous donne alors :

$$u(t + \Delta t) = u(t) + \Delta t \times u'(t) + \mathcal{O}(\Delta t)$$

Ainsi :

$$\frac{u(t + \Delta t) - u(t)}{\Delta t} = u'(t) + \mathcal{O}(\Delta t)$$

Montrons maintenant la stabilité du schéma :

On peut écrire :

$$u(t + \Delta t) = u(t) + \Delta t f(u(t)),$$

$$\text{où } u(t) = \begin{pmatrix} x_i \\ v_i \end{pmatrix} \in \mathbb{R}^{2N} \text{ et } (f(u(t)))_i = \begin{cases} v_i, & \forall 1 \leq i \leq N, \\ \frac{1}{N} \sum_{j=1}^N \nabla W(x_i - x_j), & \forall (N+1) \leq i \leq 2N. \end{cases}$$

Alors pour que le schéma soit stable, il suffit que la fonction  $f$  soit lipschitzienne en  $u$ , i.e.  $\exists \alpha \geq 0$  tel que :

$$\forall t \geq 0, \quad \forall u_1, u_2 \in \mathbb{R}^{2N}, \quad \|f(u_1) - f(u_2)\| \leq \alpha \|u_1 - u_2\|.$$

Ainsi, sous l'hypothèse que  $W$  soit au moins  $\mathcal{C}^2$ , alors  $\nabla W$  est au moins de classe  $\mathcal{C}^1$  et le caractère lipschitzien de  $f$  nous est donné par le Théorème des Accroissements Finis, il en découle ainsi la stabilité du schéma.

Pour conclure, le schéma étant consistant et stable, alors il est convergent. □

Propriété 2 :

Soit  $(x_i, v_i)_{i=1, \dots, N}$  solution de (MF), alors la vitesse moyenne des particules est conservée, i.e.

$$\frac{1}{N} \sum_{i=1}^N v_i(t) = \frac{1}{N} \sum_{i=1}^N v_i(0), \quad \forall t \geq 0.$$

*Démonstration.*

$$\begin{aligned} \frac{d}{dt} \sum_{i=1}^N v_i &= \sum_{i=1}^N \frac{d}{dt} v_i \\ &= \sum_{i=1}^N \frac{1}{N} \sum_{j=1}^N \nabla W(x_i - x_j) \\ &= \frac{1}{N} \sum_{1 \leq i, j \leq N} \nabla W(x_i - x_j). \end{aligned}$$

Or en reconnaissant  $r = \sqrt{x^2 + y^2}$ , on a :

$$\nabla W(x) = \begin{pmatrix} \frac{x}{\sqrt{x^2 + y^2}} \times k'(\sqrt{x^2 + y^2}) \\ \frac{y}{\sqrt{x^2 + y^2}} \times k'(\sqrt{x^2 + y^2}) \end{pmatrix} = x \times \frac{k'(r)}{r}.$$

Ainsi,  $\nabla W(x_i - x_j) = -\nabla W(x_j - x_i), \forall 1 \leq i, j \leq N$ .

Donc, en sommant sur tous les  $1 \leq i, j \leq N$ , chaque  $\nabla W(x_i - x_j)$  est annulé par  $\nabla W(x_i - x_j)$ .

D'où :

$$\frac{1}{N} \sum_{1 \leq i, j \leq N} \nabla W(x_i - x_j) = \frac{d}{dt} \sum_{i=1}^N v_i = 0 \iff \sum_{i=1}^N v_i = K, \quad K \in \mathbb{R}.$$

En particulier, on a donc :

$$\sum_{i=1}^N v_i(t) = \sum_{i=1}^N v_i(0), \quad \forall t \geq 0.$$

□

## Codes Python :

### Modèle 2D :

---

```
# Potentiel d'interaction
def k(r,a,b) :
    return pow(r,a)/a - pow(r,b)/b

def kprime(r,a,b) :
    return pow(r,a-1) - pow(r,b-1)

#Passage du temps t au temps t + dt :
def Mean_Field2D(x,y,vx,vy,a,b,dt) :

    # Xi - Xj
    xdifff = np.tile(x,(N,1)) - np.tile(x,(N,1)).T
    ydifff = np.tile(y,(N,1)) - np.tile(y,(N,1)).T
    normeXdifff = np.sqrt( xdifff**2 + ydifff**2 )

    # gradient évalué en Xi - Xj
    gradW = kprime(normeXdifff,a,b)

    # On évite les divisions par zéro
    np.fill_diagonal(normeXdifff,1.)

    # x(t + dt)
    x += dt*vx
    y += dt*vy

    # v(t + dt)
    vx += ((np.sum(xdifff*gradW/normeXdifff, axis=1))/N)*dt
    vy += ((np.sum(ydifff*gradW/normeXdifff, axis=1))/N)*dt

    return(x,y,vx,vy)
```

---

## Modèle 3D :

---

```
# Potentiel d'interaction
def k(r,a,b) :
    return pow(r,a)/a - pow(r,b)/b

def kprime(r,a,b) :
    return pow(r,a-1) - pow(r,b-1)

#Passage du temps t au temps t + dt :
def Mean_Field3D(x,y,z,vx,vy,vz,a,b,dt) :

    # Xi - Xj
    xdifff = np.tile(x,(N,1)) - np.tile(x,(N,1)).T
    ydifff = np.tile(y,(N,1)) - np.tile(y,(N,1)).T
    zdifff = np.tile(z,(N,1)) - np.tile(z,(N,1)).T
    normeXdifff = np.sqrt( xdifff**2 + ydifff**2 + zdifff**2)

    # gradient évalué en Xi - Xj
    gradW = kprime(normeXdifff,a,b)

    # On évite les divisions par zéro
    np.fill_diagonal(normeXdifff,1.)

    # x(t + dt)
    x += dt*vx
    y += dt*vy
    z += dt*vz

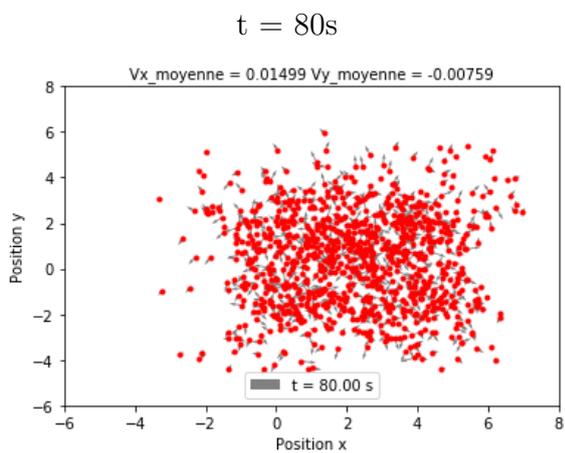
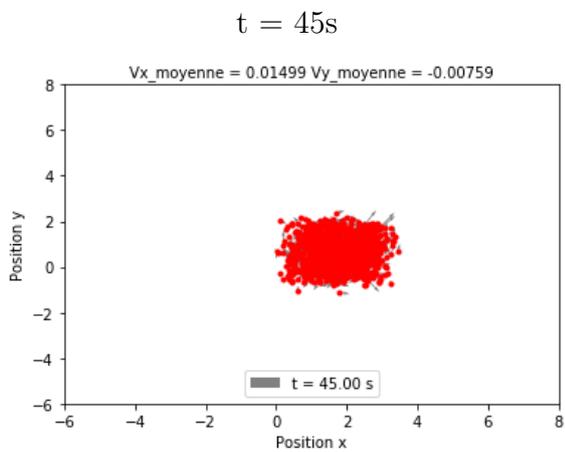
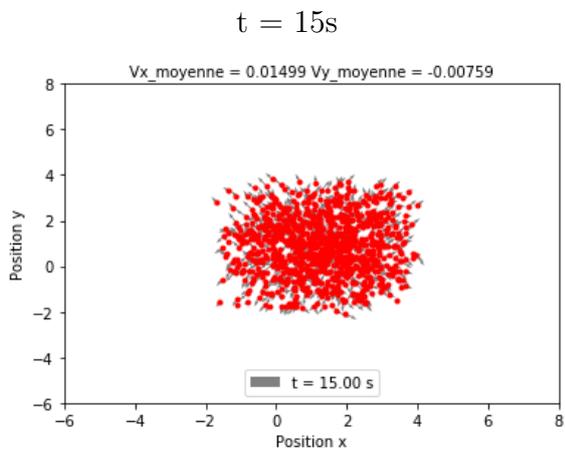
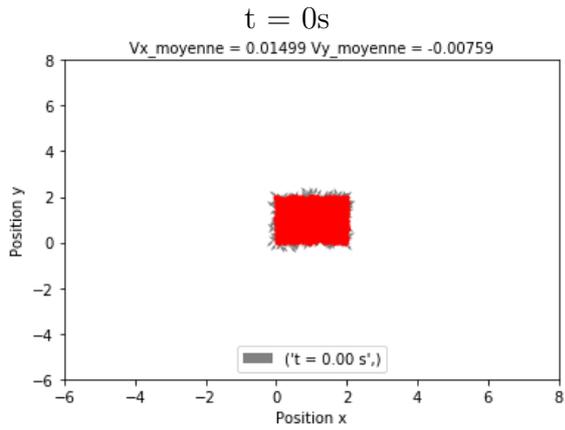
    # v(t + dt)
    vx += ((np.sum(xdifff*gradW/normeXdifff, axis=1))/N)*dt
    vy += ((np.sum(ydifff*gradW/normeXdifff, axis=1))/N)*dt
    vz += ((np.sum(zdifff*gradW/normeXdifff, axis=1))/N)*dt

    return(x,y,z,vx,vy,vz)
```

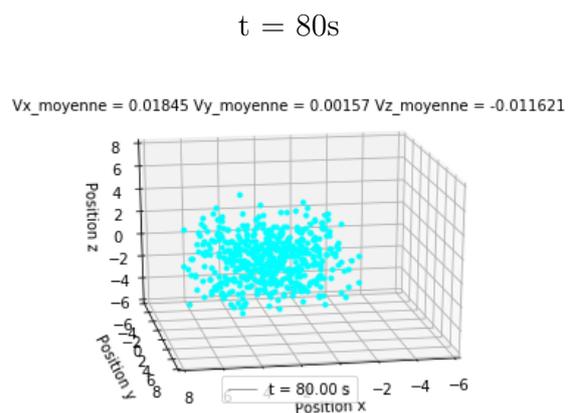
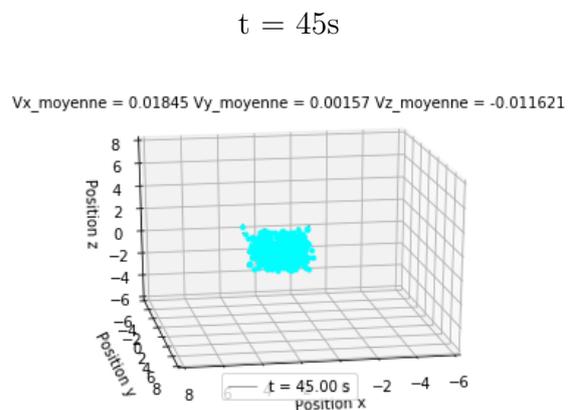
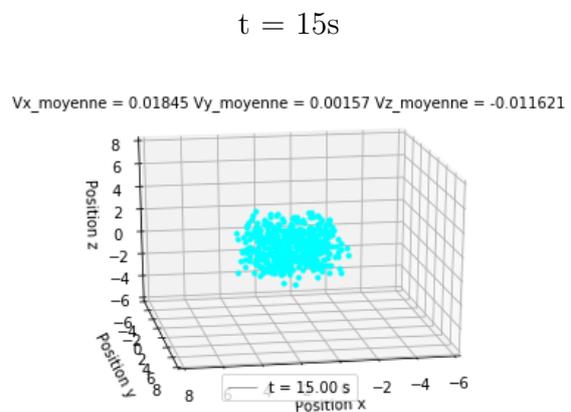
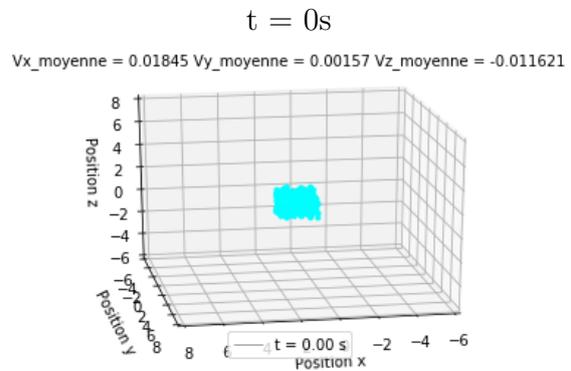
---

### 1.1.3 Résultats

2D



3D



## Commentaires :

En dimension  $d = 2$  ou  $d = 3$ , nous sommes partis d'une disposition au temps  $t = 0$  aléatoire. Nous avons ainsi considérés des particules à position aléatoire dans la boîte  $[0, 2]^d$  et à vitesse aléatoire dans  $[-1, 1]$ . Nous observons ainsi le même comportement, quelque soit la dimension.

Tout d'abord, on vérifie que la condition  $a > b > 0$  est bien nécessaire : si ce n'est pas le cas, les particules ne font que s'éloigner et partent à l'infini. (Ce cas n'est pas représenté car peu intéressant.)

Lorsque la condition est vérifiée, on peut alors observer comme attendu l'émergence d'un comportement collectif : les particules s'éloignent et se rapprochent de façon périodique, l'amplitude de ces mouvements étant contrôlée par le rapport entre les forces d'attraction et de répulsion (i.e entre  $a$  et  $b$ ).

Dans tous les cas, la phase d'éloignement s'amplifie au fur et à mesure du temps, les particules reviennent toujours les unes vers les autres, mais partent de plus en plus loin. Plus le coefficient  $a$  est fort, plus cette croissance dans l'amplitude est lente, elle a cependant bien lieu.

En temps long, on peut observer que le modèle n'est pas stable car les particules qui se sont éloignées le plus loin reviennent vers les autres avec une vitesse de plus en plus grande, cela conduit à l'explosion des vitesses. Le temps avant explosion est d'autant plus court que le coefficient  $a$  est peu élevé.

## 1.2 Modèle de Cucker-Smale

Le modèle proposé par Cucker-Smale ([1],[2],[4]) prend la forme :

$$(CS) \begin{cases} x'_i(t) = v_i(t), \\ v'_i(t) = \frac{\lambda}{N} \sum_{j=1}^N H(x_i(t) - x_j(t))(v_j(t) - v_i(t)). \end{cases}$$

Où nous utilisons toujours l'abus de notation énoncé en (1.1).

Ici, nous considérons  $\lambda$  étant une constante positive, et  $H(x)$  est un potentiel d'interaction que nous pourrions écrire sous la forme :

$$H(x) = g(\sqrt{x^2 + y^2}) = g(r),$$

où  $g \in \mathcal{C}^1(\mathbb{R}_+, \mathbb{R})$ .

Nous voyons ainsi que comme dans (1.1), le potentiel  $H$  est une fonction radiale.

Contrairement au modèle de Champ Moyen, le modèle de Cucker-Smale ne nous permettra pas de faire jouer la répulsion et l'attraction entre particules mais bien l'alignement de celles-ci. Nous verrons plus loin comment mêler ces deux approches afin de prendre en compte le schéma à trois zones, énoncé en introduction, dans son ensemble.

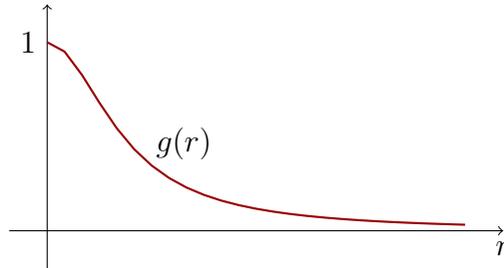
### 1.2.1 Implémentation

On utilise la méthode des différences finies, via le schéma d'Euler explicite, afin d'obtenir le système à implémenter :

$$\begin{cases} x_i(t + \Delta t) = \Delta t v_i(t) + x_i(t), \\ v_i(t + \Delta t) = \Delta t \frac{\lambda}{N} \left( \sum_{j=1}^N H(x_i(t) - x_j(t)) (v_j(t) - v_i(t)) \right) + v_i(t). \end{cases}$$

Pour la suite, nous prendrons une interaction du type :

$$g(r) = \frac{1}{1 + r^2}.$$



Ce potentiel étant proposé dans [1], [3] et [4].

## 1.2.2 Propriétés

Propriété 1 :

Soit  $(x_i, v_i)_{i=1, \dots, N}$  solution de (CS), alors la vitesse moyenne des particules est conservée, i.e.

$$\frac{1}{N} \sum_{i=1}^N v_i(t) = \frac{1}{N} \sum_{i=1}^N v_i(0), \quad \forall t \geq 0.$$

*Démonstration.*

$$\begin{aligned} \frac{d}{dt} \sum_{i=1}^N v_i &= \sum_{i=1}^N \frac{d}{dt} v_i \\ &= \sum_{i=1}^N \frac{\lambda}{N} \sum_{j=1}^N H(x_i - x_j)(v_j - v_i) \\ &= \frac{\lambda}{N} \sum_{1 \leq i, j \leq N} H(x_i - x_j)(v_j - v_i) \end{aligned}$$

Or  $H(x_i - x_j) = g(|x_i - x_j|)$ , on a donc  $H(x_i - x_j) = H(x_j - x_i), \forall 1 \leq i, j \leq N$ .

Ainsi en sommant sur tous les  $1 \leq i, j \leq N$ , chaque  $H(x_i - x_j)(v_j - v_i)$  est annulé par  $H(x_i - x_j)(v_i - v_j)$ .

D'où :

$$\frac{\lambda}{N} \sum_{1 \leq i, j \leq N} H(x_i - x_j)(v_j - v_i) = \frac{d}{dt} \sum_{i=1}^N v_i = 0 \iff \sum_{i=1}^N v_i = K, \quad K \in \mathbb{R}.$$

En particulier, on a donc :

$$\sum_{i=1}^N v_i(t) = \sum_{i=1}^N v_i(0), \quad \forall t \geq 0.$$

□

Propriété 2 :

En définissant le centre de masse du système à la position  $x_c$  allant à la vitesse  $v_c$  par :

$$x_c(t) = \frac{1}{N} \sum_{i=1}^N x_i, \quad v_c(t) = \frac{1}{N} \sum_{i=1}^N v_i, \quad \forall t \geq 0.$$

Alors nous pouvons montrer que la vitesse  $v_c$  du centre de masse est constante et que la trajectoire  $x_c$  est une ligne droite, i.e.

$$v_c(t) = v_c(0), \quad x_c(t) = v_c(0)t + x_c(0), \quad \forall t \geq 0.$$

*Démonstration.* En utilisant la conservation de la vitesse moyenne des particules (cf. Propriété 1), on obtient :

$$\begin{aligned} v_c(t) &= \frac{1}{N} \sum_{i=1}^N v_i(t) \\ &= \frac{1}{N} \sum_{i=1}^N v_i(0) \\ &= v_c(0), \quad \forall t \geq 0. \end{aligned}$$

Quant au cas de  $x_c(t)$ , on exploite le fait que dans ce modèle la position s'exprime en fonction de la dérivée de la vitesse i.e,

$$x'_i(t) = v_i(t), \quad \forall t \geq 0.$$

Cela induit que :

$$\begin{aligned} x'_c(t) &= \frac{1}{N} \sum_{i=1}^N x'_i(t) \\ &= \frac{1}{N} \sum_{i=1}^N v_i(t) \\ &= v_c(0). \end{aligned}$$

D'où :

$$x_c(t) = v_c(0)t + K, \quad K \in \mathbb{R}.$$

Et pour  $t = 0$ , on trouve :

$$K = x_c(0).$$

Finalement,

$$x_c(t) = v_c(0)t + x_c(0), \quad \forall t \geq 0.$$

□

## Codes Python :

### Modèle 2D :

---

```
# Potentiel d'interaction :
def g(r):
    return 1/(r**2 + 1)

# Modèle de Cucker Smale en dimension 2 :
def Cucker_Smale2D(x,y,vx,vy,dt,lamda):

    #  $X_i - X_j$ 
    xdifff = np.tile(x,(N,1)) - np.tile(x,(N,1)).T
    ydifff = np.tile(y,(N,1)) - np.tile(y,(N,1)).T
    normeXdifff = np.sqrt( xdifff**2 + ydifff**2 )

    #  $V_j - V_i$ 
    vxdifff = np.tile(vx,(N,1)) - np.tile(vx,(N,1)).T
    vydifff = np.tile(vy,(N,1)) - np.tile(vy,(N,1)).T

    # r évalué en  $X_i - X_j$ 
    H = g(normeXdifff)

    #  $x(t + dt)$ 
    x += dt*vxdifff
    y += dt*vydifff
    # Les particules restent dans la boîte
    x = x%lim
    y = y%lim

    #  $v(x + dt)$ 
    vx += (np.sum(vxdifff*H,axis=1)/N)*dt*lamda
    vy += (np.sum(vydifff*H,axis=1)/N)*dt*lamda

    return(x,y,vx,vy)
```

---

## Modèle 3D :

---

```
# Potentiel d'interaction :
def g(r):
    return 1/(r**2 + 1)

# Modèle de Cucker Smale en dimension 3 :
def Cucker_Smale3D(x,y,z,vx,vy,vz,dt,lamda):

    # Xi - Xj
    xdifff = np.tile(x,(N,1)) - np.tile(x,(N,1)).T
    ydifff = np.tile(y,(N,1)) - np.tile(y,(N,1)).T
    zdifff = np.tile(z,(N,1)) - np.tile(z,(N,1)).T
    normeXdifff = np.sqrt( xdifff**2 + ydifff**2 + zdifff**2)

    # Vj - Vi
    vxdifff = np.tile(vx,(N,1)) - np.tile(vx,(N,1)).T
    vydifff = np.tile(vy,(N,1)) - np.tile(vy,(N,1)).T
    vzdifff = np.tile(vz,(N,1)) - np.tile(vz,(N,1)).T

    # r évalué en Xi - Xj
    H = g(normeXdifff)

    # x(t + dt)
    x += dt*vx
    y += dt*vy
    z += dt*vz
    # Les particules restent dans la boîte
    x = x%lim
    y = y%lim
    z = z%lim

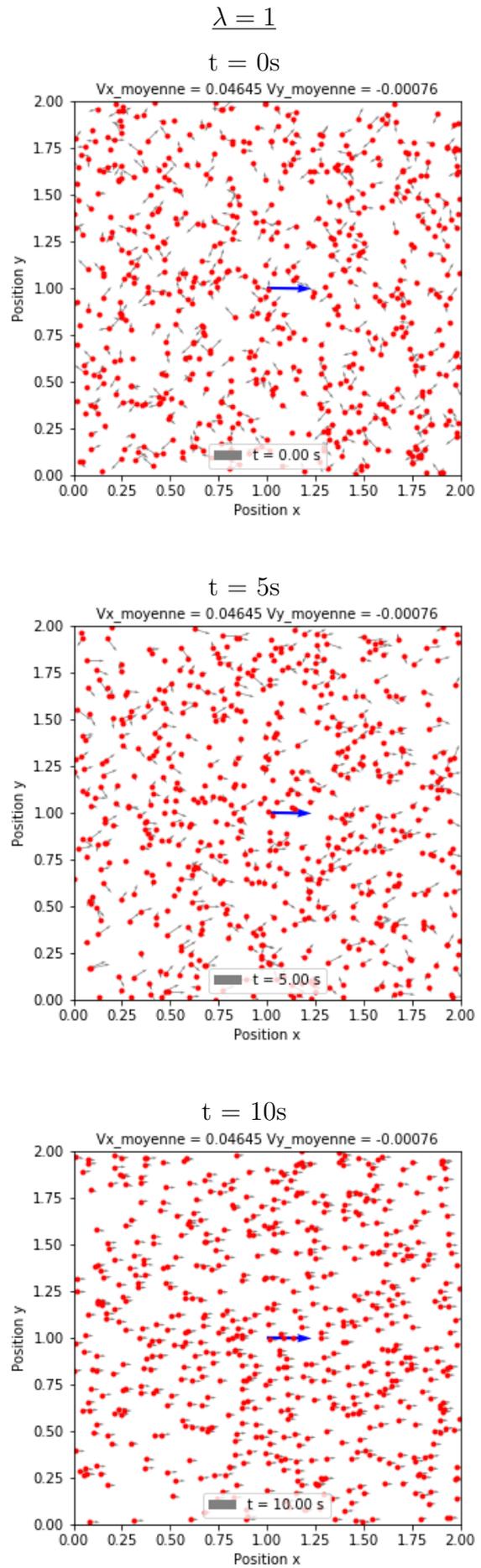
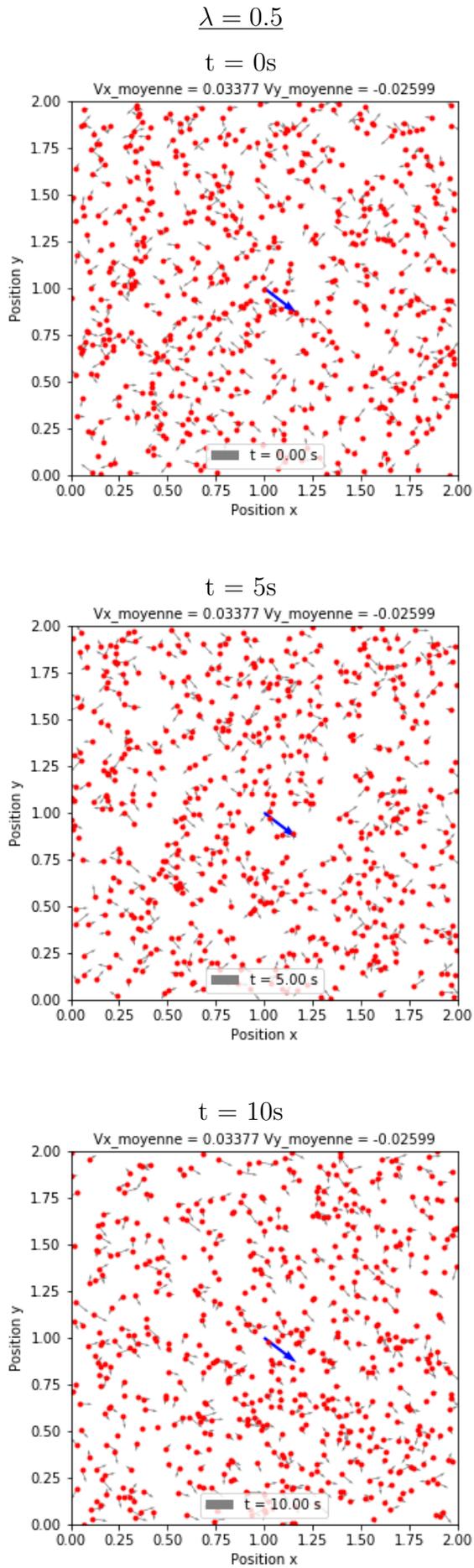
    # v(t + dt)
    vx += (np.sum(vxdifff*H,axis=1)/N)*dt*lamda
    vy += (np.sum(vydifff*H,axis=1)/N)*dt*lamda
    vz += (np.sum(vzdifff*H,axis=1)/N)*dt*lamda

    return(x,y,z,vx,vy,vz)
```

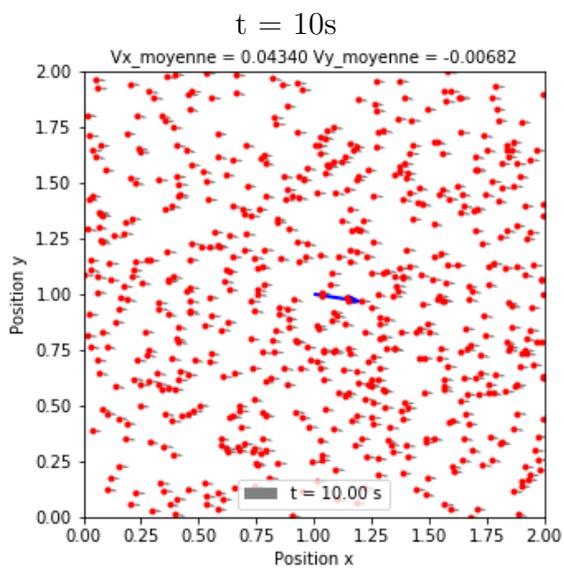
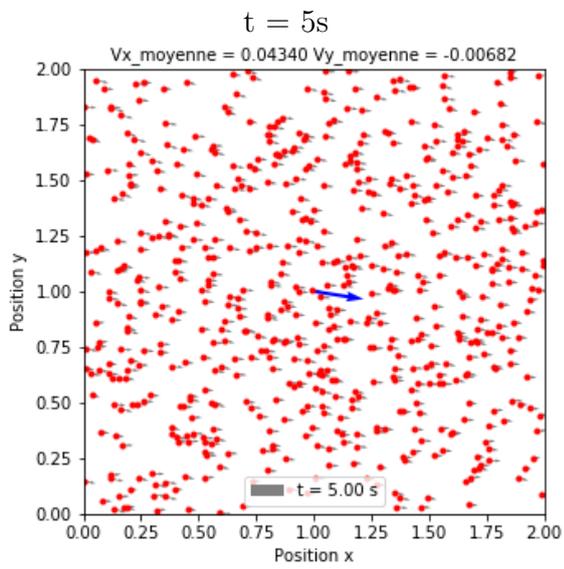
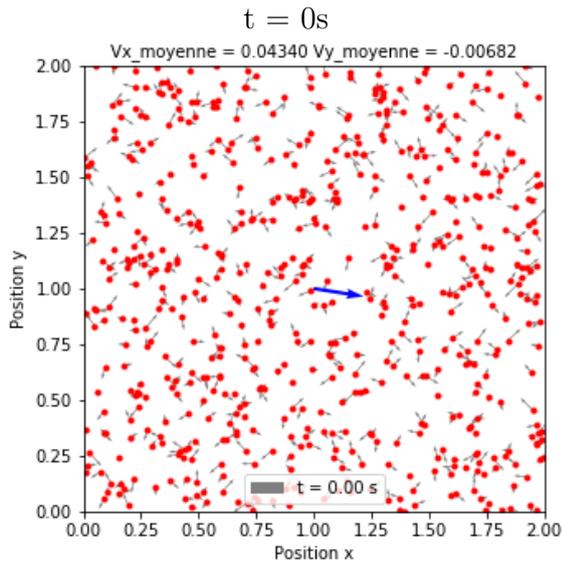
---

### 1.2.3 Résultats

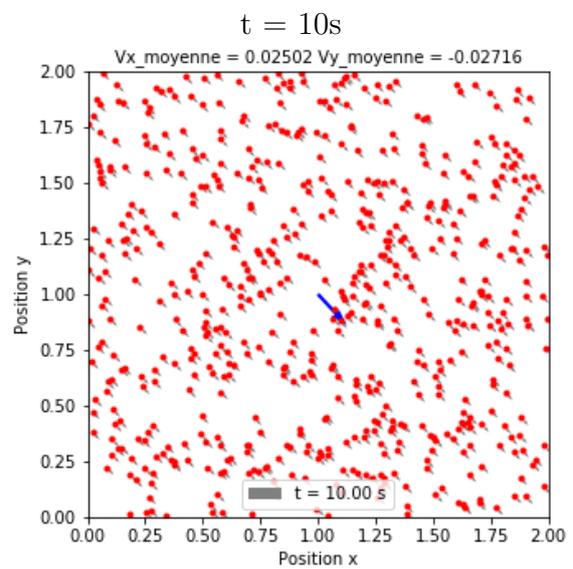
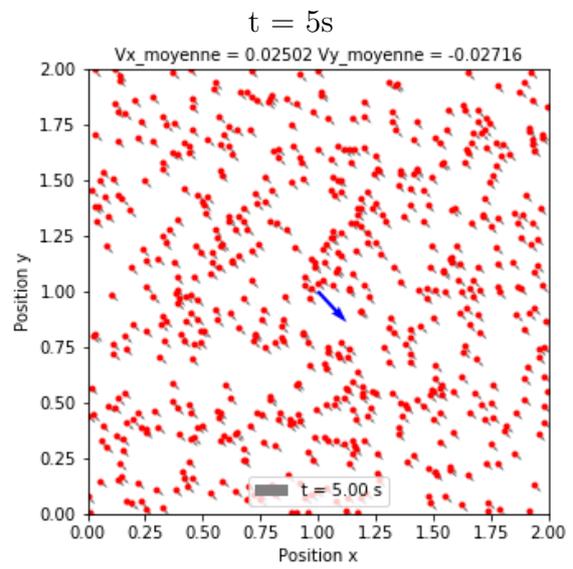
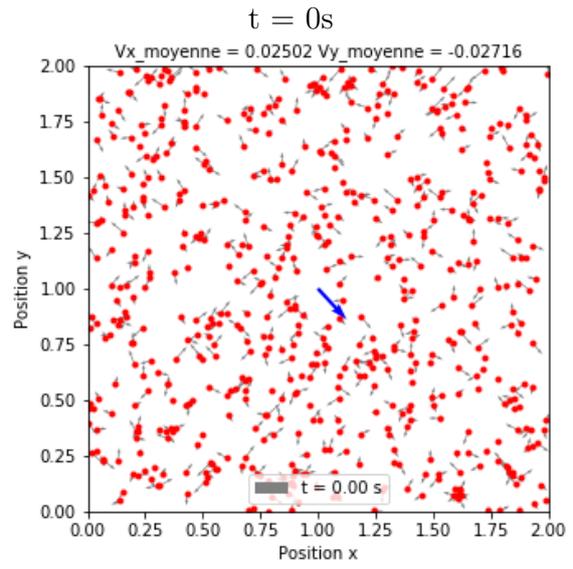
En 2D :



$$\lambda = 2$$



$$\lambda = 3$$

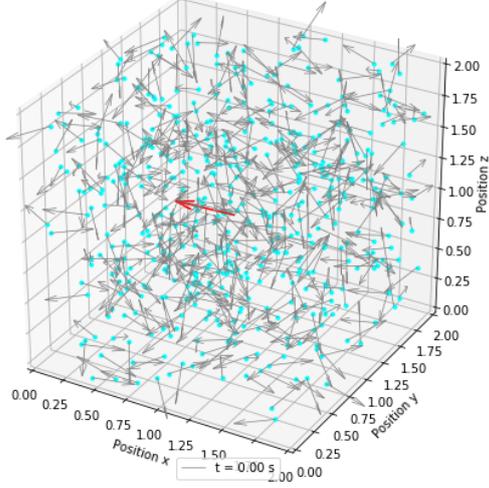


En 3D :

$$\lambda = 1$$

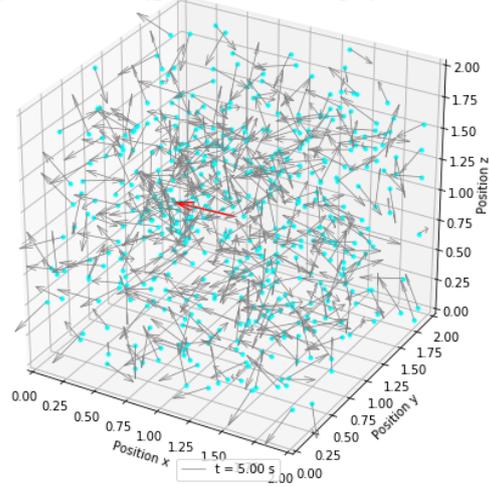
$$t = 0s$$

Vx\_moyenne = -0.02494 Vy\_moyenne = -0.02340 Vz\_moyenne = 0.014015



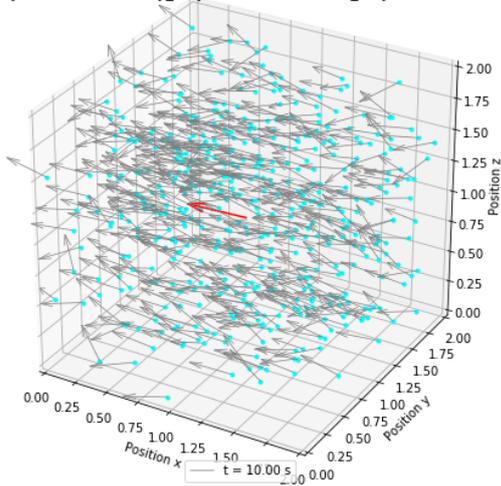
$$t = 5s$$

Vx\_moyenne = -0.02494 Vy\_moyenne = -0.02340 Vz\_moyenne = 0.014015



$$t = 10s$$

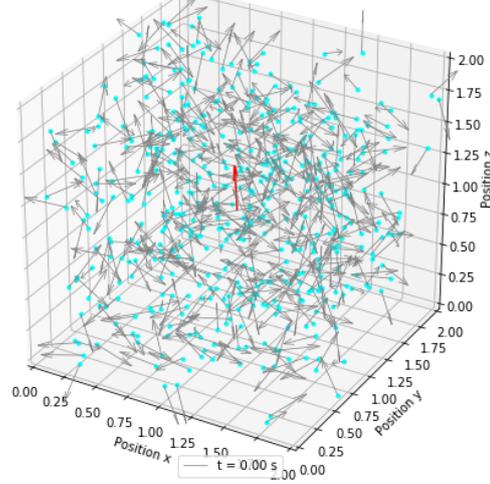
Vx\_moyenne = -0.02494 Vy\_moyenne = -0.02340 Vz\_moyenne = 0.014015



$$\lambda = 2$$

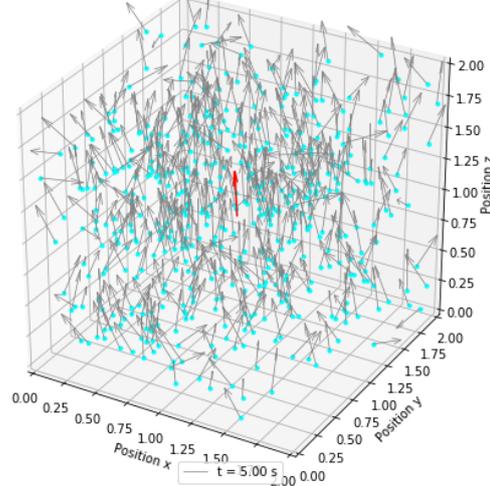
$$t = 0s$$

Vx\_moyenne = 0.00016 Vy\_moyenne = -0.00290 Vz\_moyenne = 0.031428



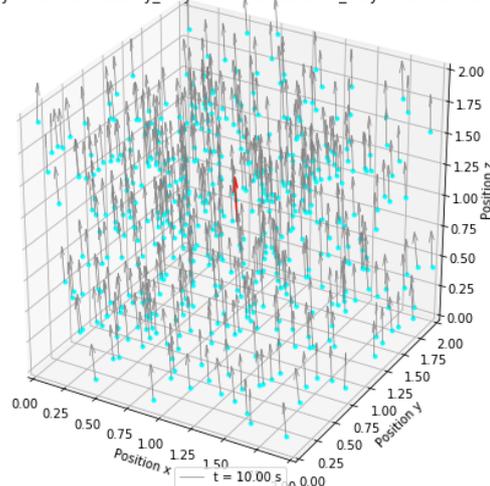
$$t = 5s$$

Vx\_moyenne = 0.00016 Vy\_moyenne = -0.00290 Vz\_moyenne = 0.031428



$$t = 10s$$

Vx\_moyenne = 0.00016 Vy\_moyenne = -0.00290 Vz\_moyenne = 0.031428



## Commentaires :

En dimension  $d = 2$  ou  $d = 3$ , pour chaque  $\lambda$  étudié, nous sommes partis d'une disposition au temps  $t = 0$  aléatoire. Nous avons ainsi considérés des particules à position aléatoire dans la boîte  $[0, 2]^d$  et à vitesse aléatoire dans  $[-1, 1]$ .

Nous pouvons observer que  $\forall \lambda > 0$ , la vitesse moyenne des particules est constante et leur trajectoire converge vers un déplacement en ligne droite, uniquement fonction des conditions initiales simulées, comme démontré dans la section (1.2.2 Propriétés). La valeur du paramètre  $\lambda$  influe uniquement sur la vitesse de convergence, plus  $\lambda$  est grand, plus la convergence est rapide.

### 1.3 Modèles mixtes

Dans cette section, nous considérerons deux nouvelles approches :

- La première, issue des modèles de Champ Moyen et de Cucker-Smale :

$$(CS + MF) \begin{cases} x'_i(t) = v_i(t), \\ v'_i(t) = \frac{\lambda}{N} \sum_{j=1}^N H(x_i(t) - x_j(t))(v_j(t) - v_i(t)) + \frac{1}{N} \sum_{j=1}^N \nabla W(x_i(t) - x_j(t)). \end{cases}$$

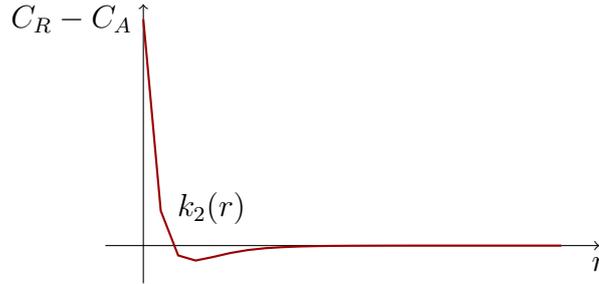
- Et la seconde, issue du modèle de Champ Moyen et d'un terme d'auto-propulsion/friction :

$$(SP + MF) \begin{cases} x'_i(t) = v_i(t), \\ v'_i(t) = S(|v_i(t)|)v_i(t) + \frac{1}{N} \sum_{j=1}^N \nabla W(x_i(t) - x_j(t)). \end{cases}$$

### 1.3.1 Implémentation

Pour l'implémentation de ces deux modèles, nous considérerons le même potentiel  $H$  que dans la section (1.2), pour le potentiel  $W$ , nous expérimenterons le même potentiel que dans la section (1.1) ainsi que le potentiel de Morse, proposé dans [4] et [6] :

$$k_2(r) = -C_A e^{-r/l_A} + C_R e^{-r/l_R},$$



$k_2(r)$  pour  $C_A = 2$ ,  $C_R = 5$ ,  $l_A = 0.2$  et  $l_R = 0.1$

où  $C_A$  et  $C_R$  sont les forces d'attraction et de répulsion et  $l_A$  et  $l_R$  leur distance d'action respective. L'intérêt de ce potentiel est d'observer, contrairement au potentiel introduit en (1.1), des interactions qui n'agissent qu'entre particules plus ou moins proches.

Enfin nous utiliserons le terme d'auto-propulsion/friction étudié dans [4] :

$$S(|v|) = \alpha - \beta|v|^2, \quad \alpha, \beta > 0.$$

## Codes Python :

CS + MF (2D) :

---

```
def k1prime(r,a,b) :
    return pow(r,a-1) - pow(r,b-1)

def k2prime(r) :
    Cr, Ca, lr, la = 5., 2., .1, .2
    return Ca/la*np.exp(-r/la) - Cr/lr*np.exp(-r/lr)

def g(r):
    return 1/(r**2 + 1)

#Passage du temps t au temps t + dt :
def CS_MF2D(x,y,vx,vy,a,b,dt) :

    # Xi - Xj
    xdifff = np.tile(x,(N,1)) - np.tile(x,(N,1)).T
    ydifff = np.tile(y,(N,1)) - np.tile(y,(N,1)).T
    normeXdifff = np.sqrt( xdifff**2 + ydifff**2 )

    # Vj - Vi
    vxdifff = np.tile(vx,(N,1)) - np.tile(vx,(N,1)).T
    vydifff = np.tile(vy,(N,1)) - np.tile(vy,(N,1)).T

    # gradient évalué en Xi - Xj
    gradW = k1prime(normeXdifff,a,b)
    gradW̄ = k2prime(normeXdifff)

    # r évalué en Xi - Xj
    H = g(normeXdifff)

    # On évite les divisions par zéro
    np.fill_diagonal(normeXdifff,1.)

    # x(t + dt)
    x += dt*vx
    y += dt*vy

    # v(t + dt)
    vx += ((np.sum(xdifff*gradW/normeXdifff, axis=1)+np.sum(vxdifff*H,axis=1))/N)*dt
    vy += ((np.sum(ydifff*gradW/normeXdifff, axis=1)+np.sum(vydifff*H,axis=1))/N)*dt

    return(x,y,vx,vy)
```

---

## CS + MF (3D) :

---

```
def k1prime(r,a,b) :
    return pow(r,a-1) - pow(r,b-1)

def k2prime(r) :
    Cr, Ca, lr, la = 5., 2., .1, .2
    return Ca/la*np.exp(-r/la) - Cr/lr*np.exp(-r/lr)

def g(r):
    return 1/(r**2 + 1)

#Passage du temps t au temps t + dt :
def CS_MF3D(x,y,z,vx,vy,vz,a,b,dt) :

    # Xi - Xj
    xdifff = np.tile(x,(N,1)) - np.tile(x,(N,1)).T
    ydifff = np.tile(y,(N,1)) - np.tile(y,(N,1)).T
    zdifff = np.tile(z,(N,1)) - np.tile(z,(N,1)).T
    normeXdifff = np.sqrt( xdifff**2 + ydifff**2 + zdifff**2)

    # Vj - Vi
    vxdifff = np.tile(vx,(N,1)) - np.tile(vx,(N,1)).T
    vydifff = np.tile(vy,(N,1)) - np.tile(vy,(N,1)).T
    vzdifff = np.tile(vz,(N,1)) - np.tile(vz,(N,1)).T

    # gradient évalué en Xi - Xj
    gradW = k1prime(normeXdifff,a,b)
    gradW = k2prime(normeXdifff)

    # r évalué en Xi - Xj
    H = g(normeXdifff)

    # On évite les divisions par zéro
    np.fill_diagonal(normeXdifff,1.)

    # x(t + dt)
    x += dt*vx
    y += dt*vy
    z += dt*vz

    # v(t + dt)
    vx += ((np.sum(xdifff*gradW/normeXdifff, axis=1)+np.sum(vxdifff*H,axis=1))/N)*dt
    vy += ((np.sum(ydifff*gradW/normeXdifff, axis=1)+np.sum(vydifff*H,axis=1))/N)*dt
    vz += ((np.sum(zdifff*gradW/normeXdifff, axis=1)+np.sum(vzdifff*H,axis=1))/N)*dt

    return(x,y,z,vx,vy,vz)
```

---

SP + MF (2D) :

---

```
def kprime(r,a,b) :
    return pow(r,a-1) - pow(r,b-1)

def S(v,alpha,beta) :
    return alpha - beta*v**2

#Passage du temps t au temps t + dt :
def SP_MF2D(x,y,vx,vy,a,b,dt) :

    # Xi - Xj
    xdifff = np.tile(x,(N,1)) - np.tile(x,(N,1)).T
    ydifff = np.tile(y,(N,1)) - np.tile(y,(N,1)).T
    normeXdifff = np.sqrt( xdifff**2 + ydifff**2 )

    # gradient évalué en Xi - Xj
    gradW = kprime(normeXdifff,a,b)

    # On évite les divisions par zéro
    np.fill_diagonal(normeXdifff,1.)

    # x(t + dt)
    x += dt*vx
    y += dt*vy

    # v(t + dt)
    vx += (S(vx,1.,0.05)*vx + np.sum(xdifff*gradW/normeXdifff, axis=1)/N)*dt
    vy += (S(vy,1.,0.05)*vy + np.sum(ydifff*gradW/normeXdifff, axis=1)/N)*dt

    return(x,y,vx,vy)
```

---

SP + MF (3D) :

---

```
def kprime(r,a,b) :
    return pow(r,a-1) - pow(r,b-1)

def S(v,alpha,beta) :
    return alpha - beta*v**2

#Passage du temps t au temps t + dt :
def SP_MF3D(x,y,z,vx,vy,vz,a,b,dt) :

    # Xi - Xj
    xdifff = np.tile(x,(N,1)) - np.tile(x,(N,1)).T
    ydifff = np.tile(y,(N,1)) - np.tile(y,(N,1)).T
    zdifff = np.tile(z,(N,1)) - np.tile(z,(N,1)).T
    normeXdifff = np.sqrt( xdifff**2 + ydifff**2 + zdifff**2)

    # gradient évalué en Xi - Xj
    gradW = kprime(normeXdifff,a,b)

    # On évite les divisions par zéro
    np.fill_diagonal(normeXdifff,1.)

    # x(t + dt)
    x += dt*vx
    y += dt*vy
    z += dt*vz

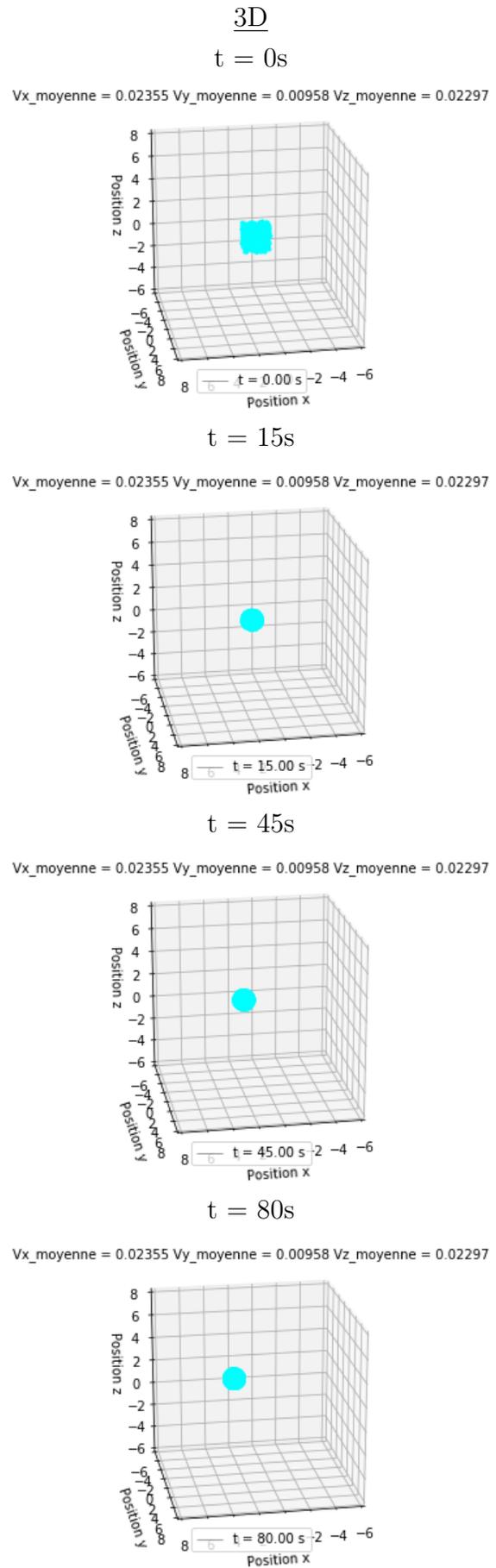
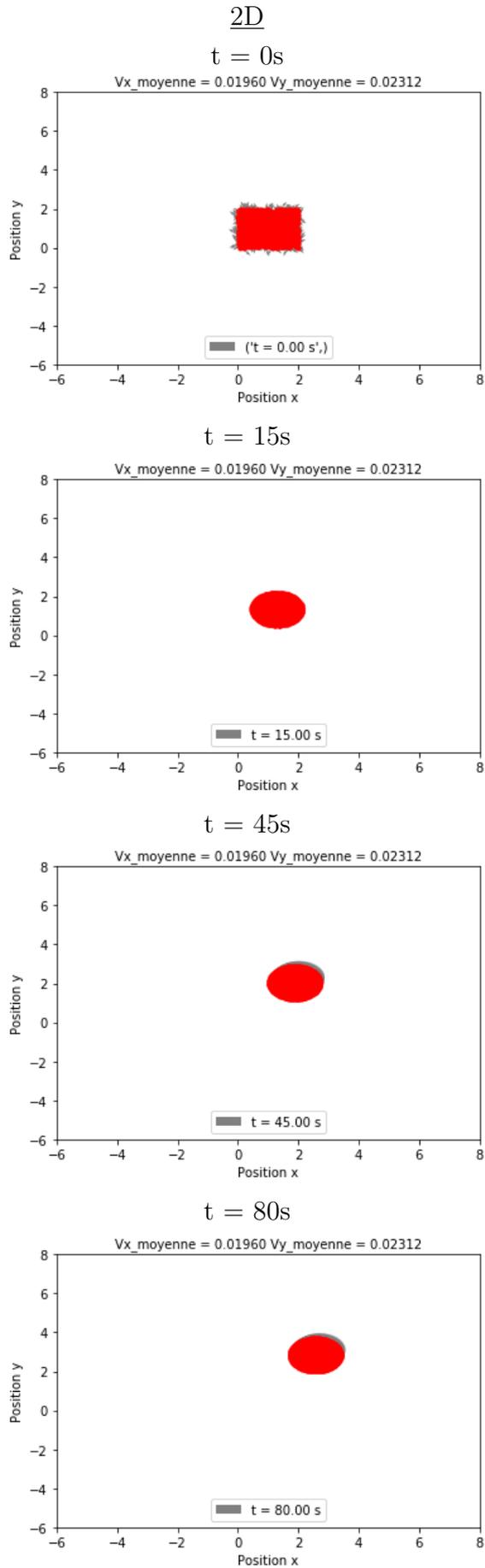
    # v(t + dt)
    vx += (S(vx,1.,0.05)*vx + np.sum(xdifff*gradW/normeXdifff, axis=1)/N)*dt
    vy += (S(vy,1.,0.05)*vy + np.sum(ydifff*gradW/normeXdifff, axis=1)/N)*dt
    vz += (S(vz,1.,0.05)*vz + np.sum(zdifff*gradW/normeXdifff, axis=1)/N)*dt

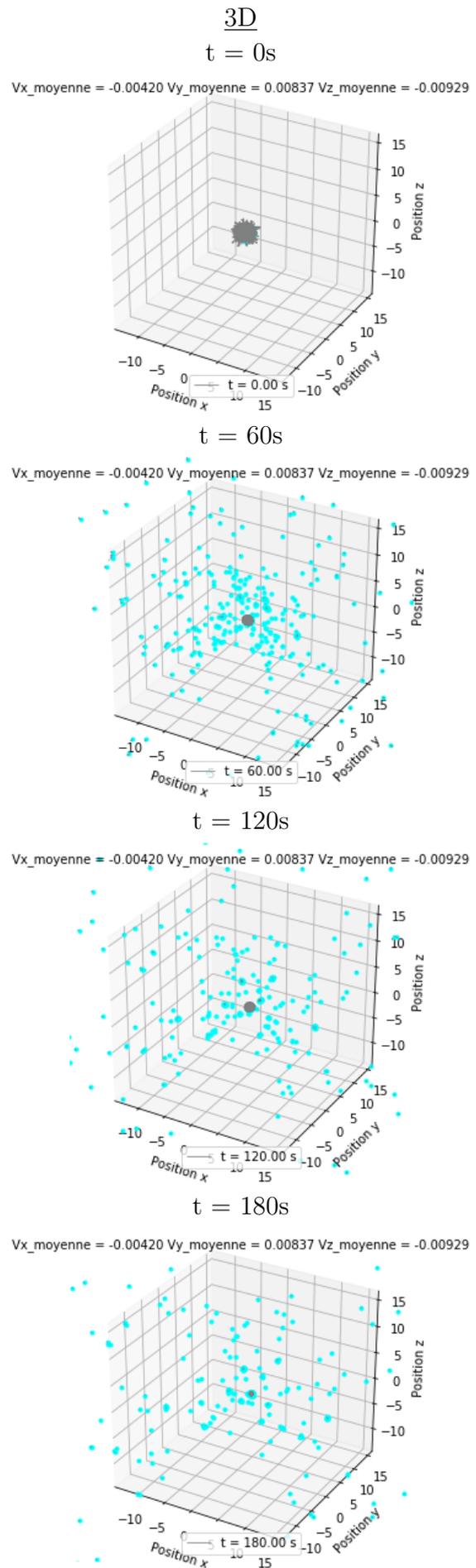
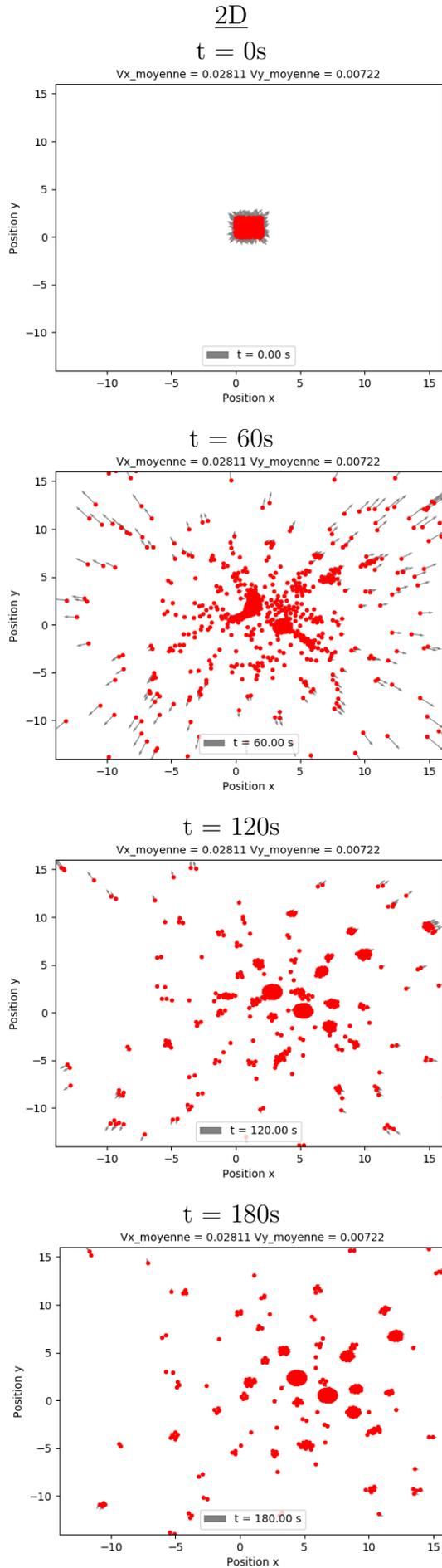
    return(x,y,z,vx,vy,vz)
```

---

### 1.3.2 Résultats

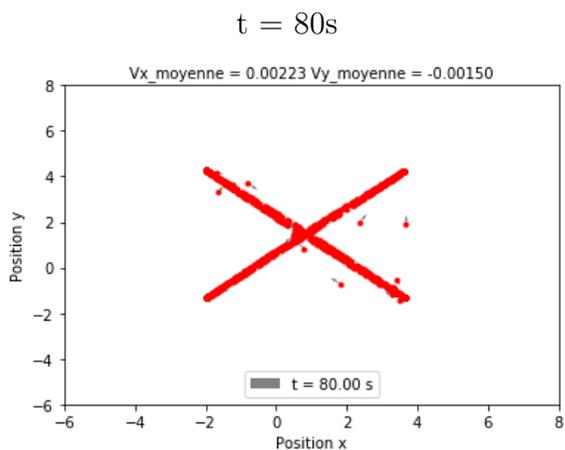
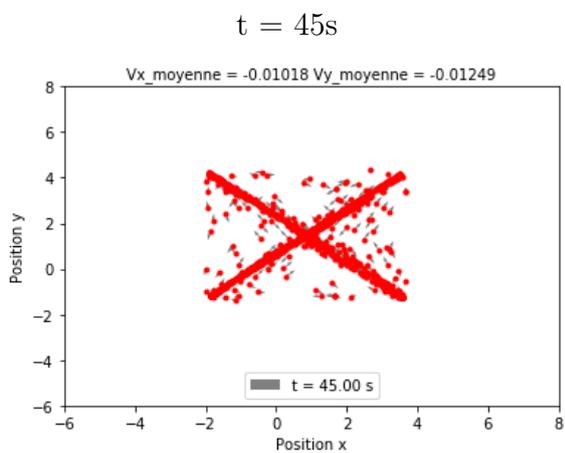
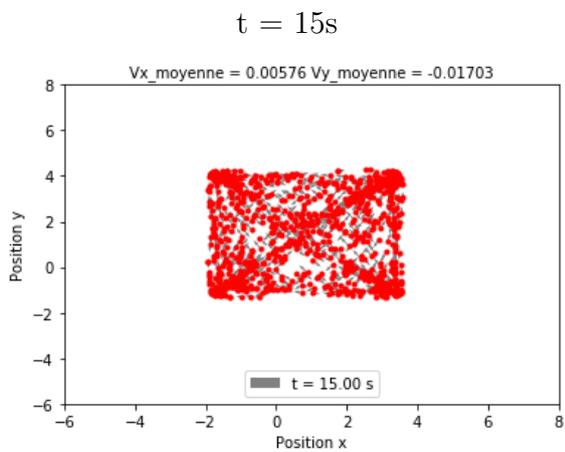
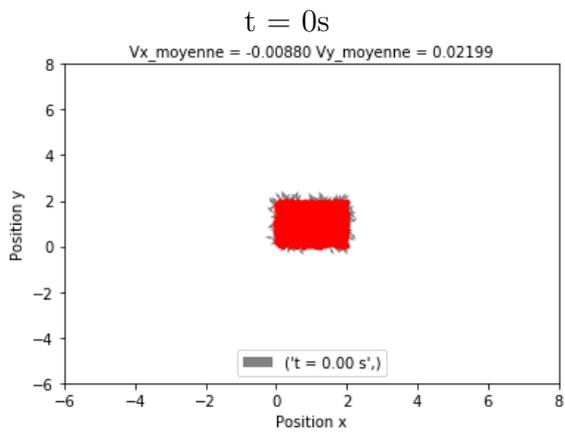
CS+MF(1) :





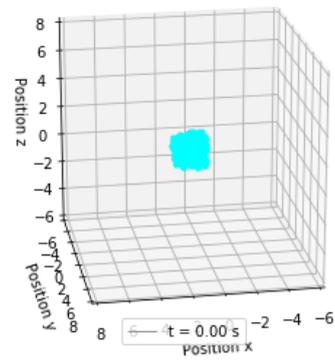
SP+MF :

2D

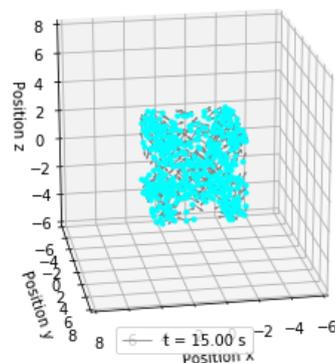


3D

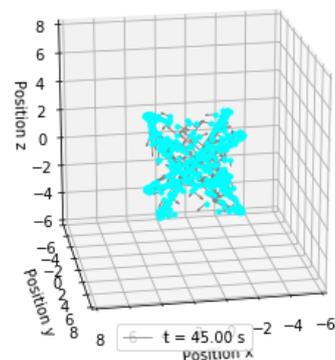
t = 0s  
 $Vx_{moyenne} = -0.00840$   $Vy_{moyenne} = 0.02069$   $Vz_{moyenne} = -0.028475$



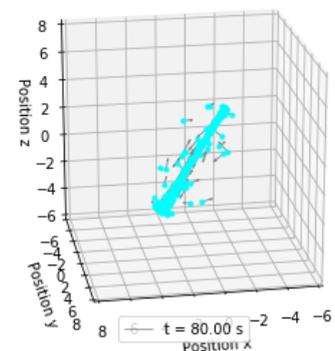
t = 15s  
 $Vx_{moyenne} = 0.00964$   $Vy_{moyenne} = 0.00886$   $Vz_{moyenne} = 0.013012$



t = 45s  
 $Vx_{moyenne} = 0.02132$   $Vy_{moyenne} = -0.01459$   $Vz_{moyenne} = 0.012326$



t = 80s  
 $Vx_{moyenne} = -0.00524$   $Vy_{moyenne} = 0.02124$   $Vz_{moyenne} = -0.0021413$



## Commentaires :

Pour le modèle CS + MF, Nous avons utilisé les potentiels  $k_1$  et  $k_2$ , on peut alors remarquer que les constantes  $a$  et  $b$  jouent le même rôle pour  $k_1$  que les constantes  $C_A$  et  $C_R$  pour  $k_2$ , plus la constante d'attraction est supérieure à la constante de répulsion, plus les particules vont s'agglomérer rapidement, puis suivre une trajectoire uniforme en gardant une structure stable.

C'est dans la prise en compte de la distance entre les particules que les deux potentiels se différencient,  $k_1$  représente une interaction "longue portée" qui fait que toutes les particules s'attirent.  $k_2$  en revanche, permet de définir la distance de prise en compte de ces interactions, plus  $l_A$  est grand, plus les particules vont s'attirer à grande portée, plus  $l_R$  est grand, plus les particules vont se repousser à grande portée. Donc, si  $l_A$  est beaucoup plus grand que  $l_R$ , toutes les particules vont s'attirer et on retrouvera un comportement similaire à  $k_1$ . En revanche, si les paramètres  $l_A$  et  $l_R$  sont bien choisis, nous observons la formation de groupes de particules indépendants.

Concernant le modèle SP + MF, on observe l'émergence d'un comportement collectif que ce soit en dimension  $d = 2$  ou  $d = 3$ . Le motif vers lequel convergent les particules dépend des potentiels utilisés, et il est possible d'obtenir de nombreux motifs.

Pour conclure sur les modèles que nous avons présenté, nous avons montré qu'à partir de considérations au niveau de chaque particule prise individuellement, il nous est possible d'obtenir un comportement collectif cohérent. La forme de ces comportements dépend des potentiels que nous utilisons et il est ainsi possible d'obtenir un mouvement général de n'importe quelle forme. De tels exemples étant présentés dans [5]. De plus, nous n'avons utilisé dans nos modèles que des cas de particules évoluant dans un espace sans obstacles. Il est possible de modéliser des particules se déplaçant en cherchant à éviter un obstacle fixe ou mouvant (ce dernier cas pouvant représenter par exemple une situation prédateur/proie), ce cas de figure étant approfondi dans [3], et simulé dans [7].

# Chapitre 2

## Modèle de densité : Équation d'agrégation

Dans cette partie, nous considérons l'équation de transport non locale, appelée équation d'agrégation, suivante :

$$(\star) \quad \frac{\partial \rho}{\partial t} - \nabla_x \cdot (\rho(\nabla W * \rho)) = 0.$$

Nous faisons le lien entre  $(\star)$  et notre problématique en considérant  $\rho$  comme la densité de particules dans une unité de surface (resp. volume si  $d = 3$ ).

### 2.1 Résultat théorique

Considérons l'approche par modèle de Champ Moyen décrite dans (1.1), alors la distribution des particules dans l'espace obtenue tend vers la solution  $\rho$  de  $(\star)$  lorsque le nombre de particules étudiées devient très grand, i.e :

Soit  $\rho(t)$  solution de  $(\star)$  et soit  $\mu_N(t) = \frac{1}{N} \sum_{i=1}^N \delta_{x_i(t)}$  la distribution empirique associée aux positions  $x_i(t)$  obtenues par l'approche particulaire. Alors :

$$\text{Si } \mu_N(0) \xrightarrow{N \rightarrow +\infty} \rho(0), \text{ alors : } \mu_N(t) \xrightarrow{N \rightarrow +\infty} \rho(t), \quad \forall t > 0.$$

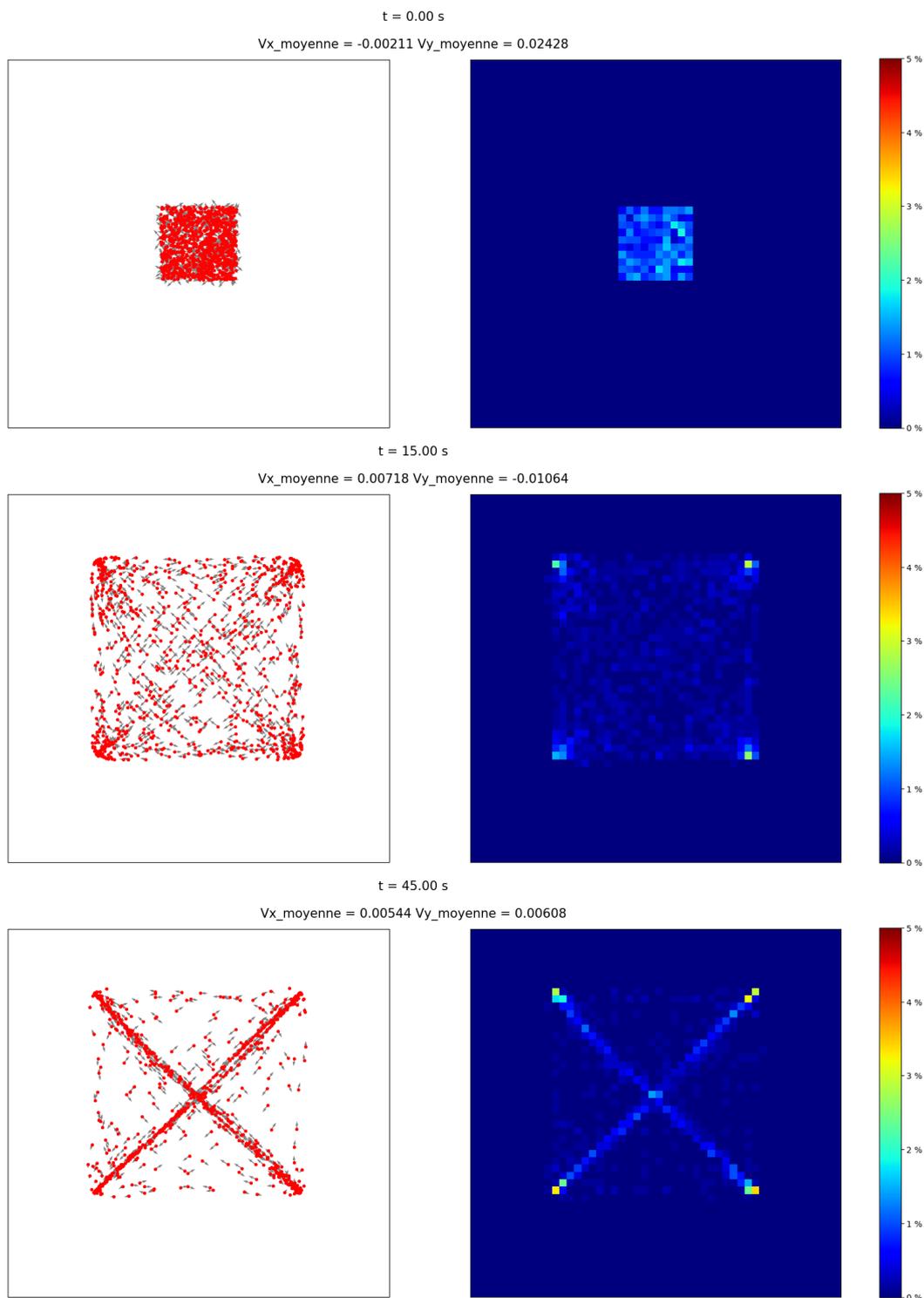
La démonstration de cette convergence, extrêmement ardue, est disponible dans [2] et [6].

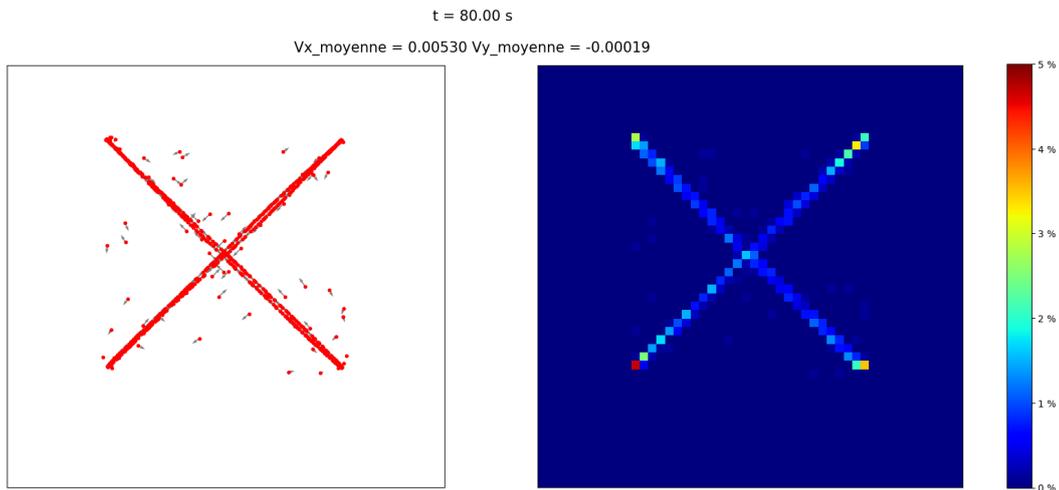
**N.B. :** Il est possible d'obtenir un résultat similaire pour les autres approches étudiées dans la partie 1, il faudrait cependant considérer la solution  $\rho$  d'équations sensiblement différentes de  $(\star)$  et la difficulté de résolution, déjà importante pour le cas que nous étudions, s'en voit encore augmentée.

## 2.2 Représentation par histogrammes

Afin de mettre en évidence le lien évoqué précédemment, nous représentons l'espace dans lequel nous faisons évoluer les particules sous la forme d'histogrammes représentant la densité observée dans de petits domaines de l'espace.

En procédant de la sorte, nous obtenons, par exemple, le résultat suivant pour le modèle (SP + MF) pour 1000 particules :





Notre but est alors d'implémenter l'évolution en temps de la solution  $\rho$  de l'équation ( $\star$ ), et d'obtenir un comportement de cette solution similaire au comportement observé sur l'histogramme obtenu par (MF).

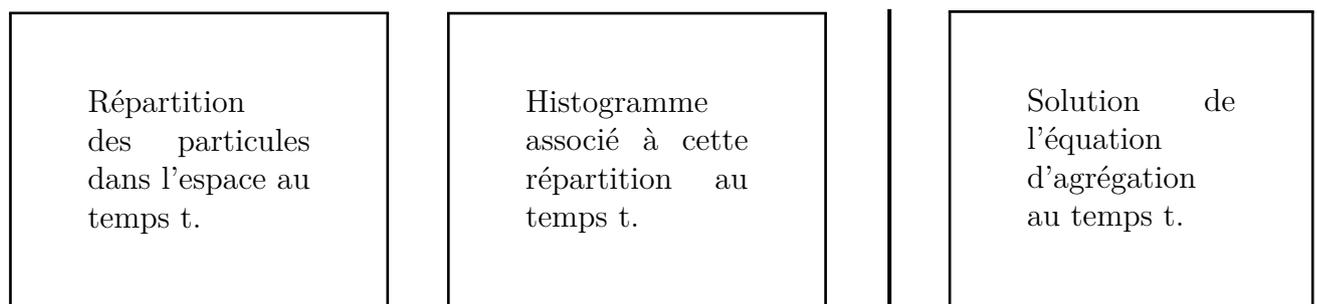
## 2.3 Comparaison et conclusions

### 2.3.1 Comparaison

Pour les raisons évoquées précédemment, nous nous proposons d'étudier l'histogramme obtenu par la résolution de (MF) pour des N de plus en plus grand, puis de comparer les résultats obtenus avec le résultat de l'équation d'agrégation.

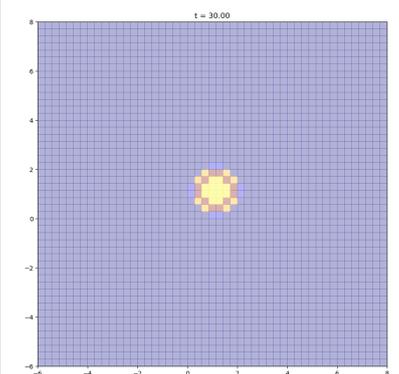
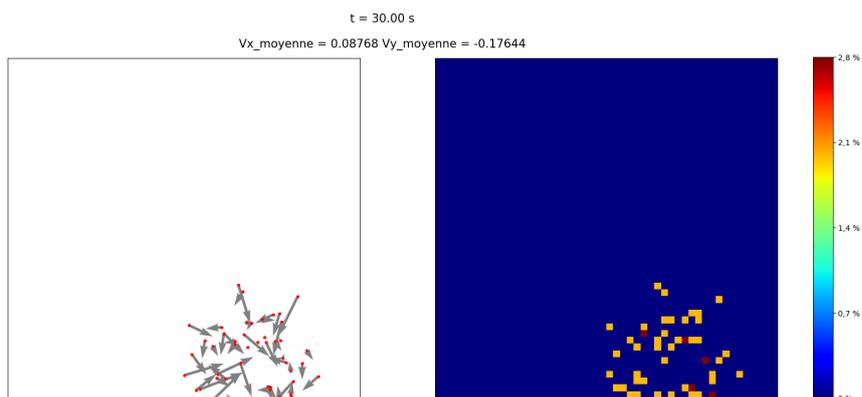
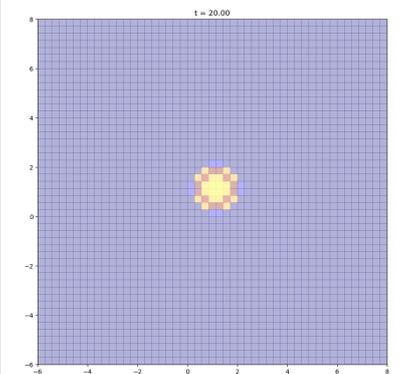
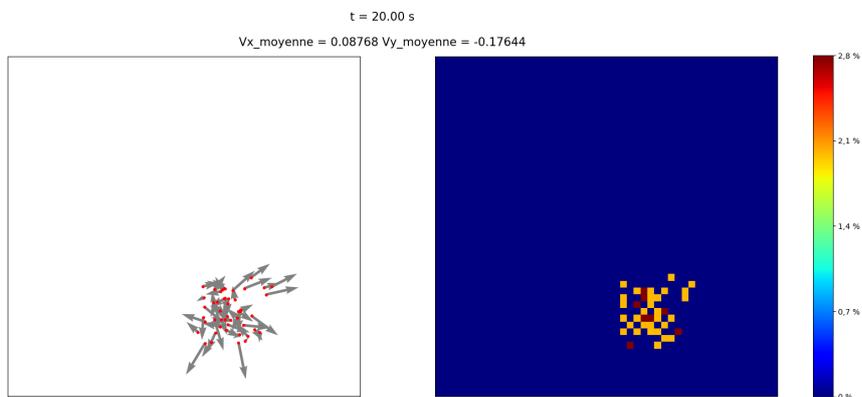
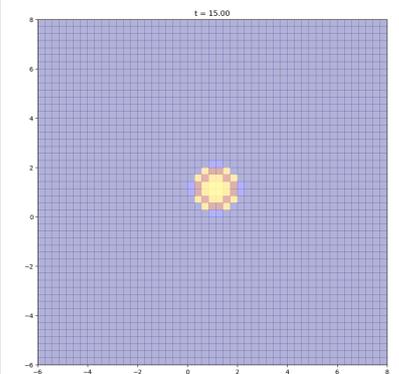
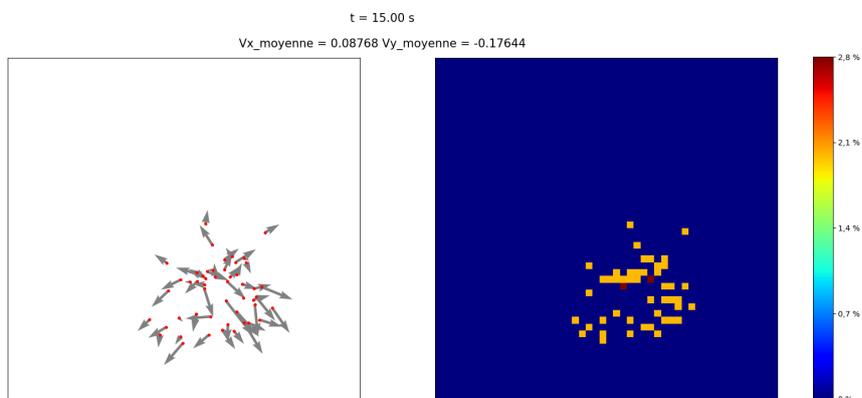
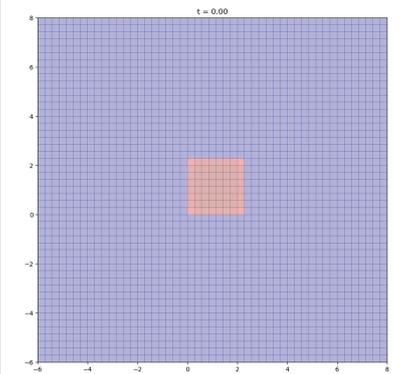
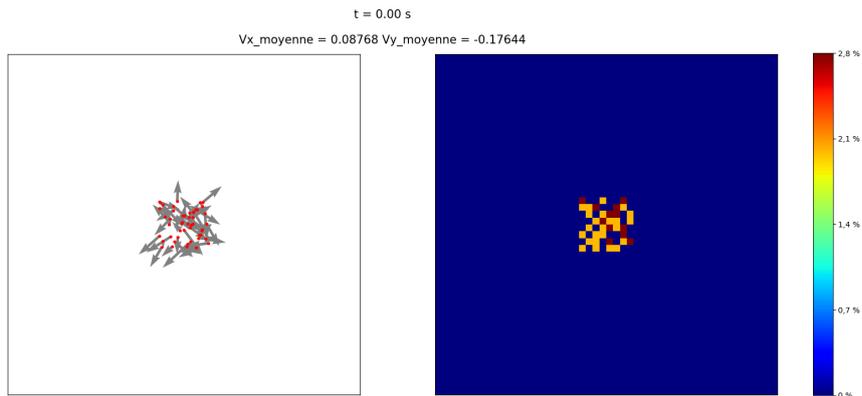
Afin d'effectuer cette comparaison, nous présentons les résultats comme suit :

Pour chaque N étudié :



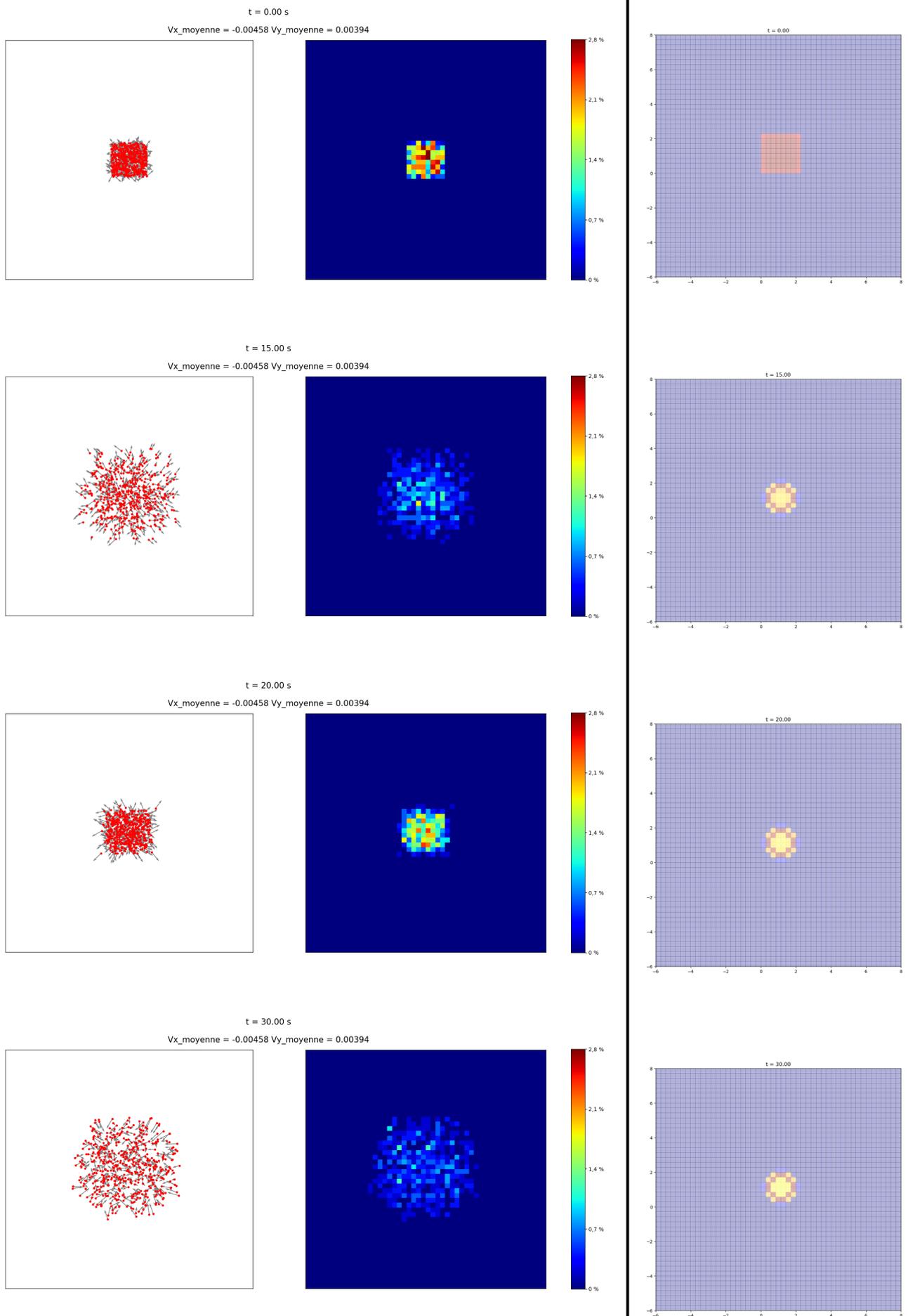
# N = 50

# Agrégation



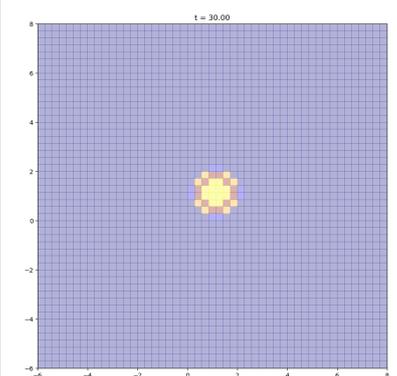
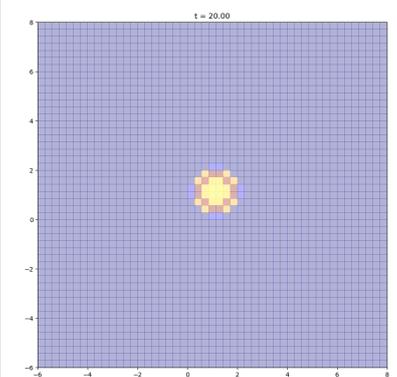
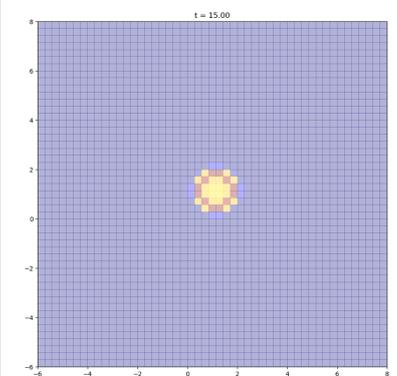
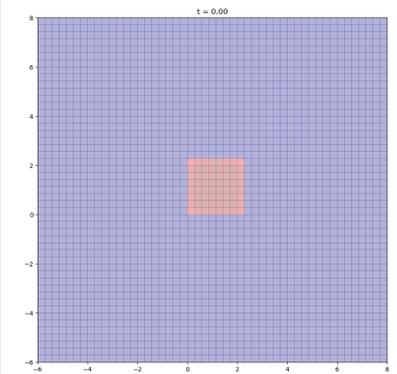
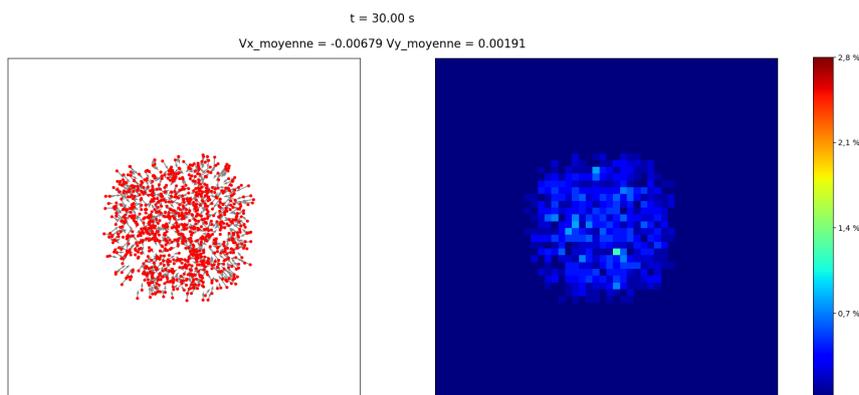
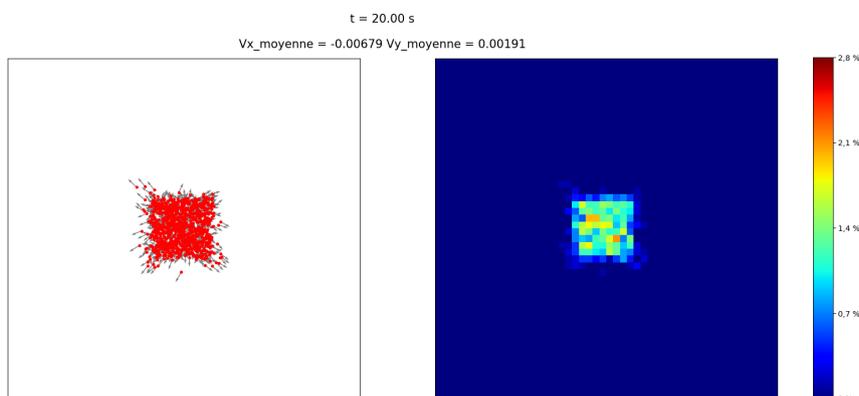
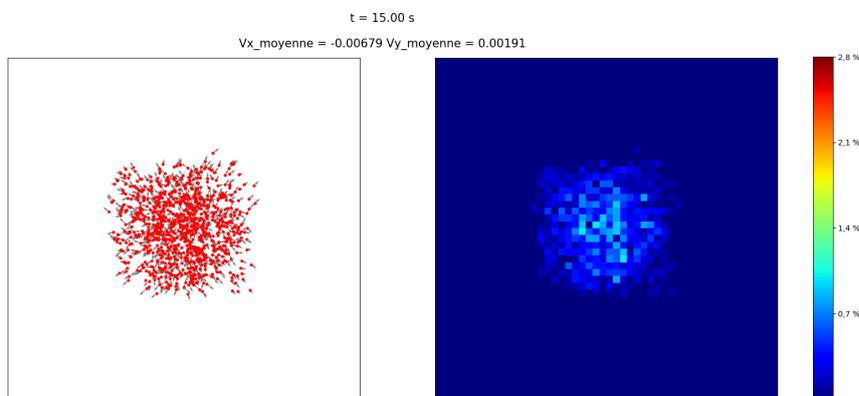
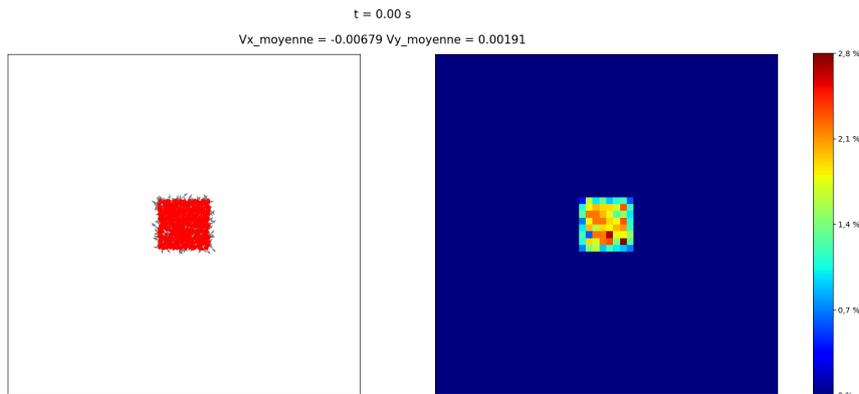
N = 500

Agrégation



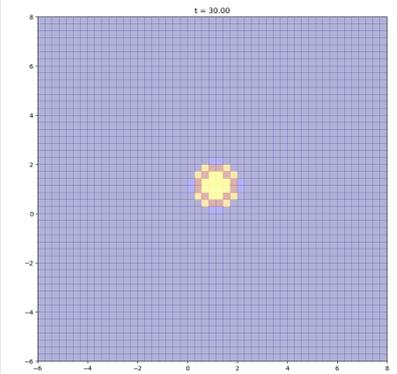
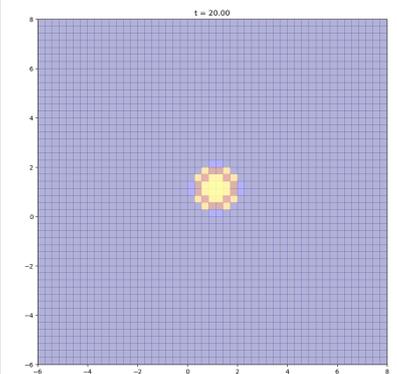
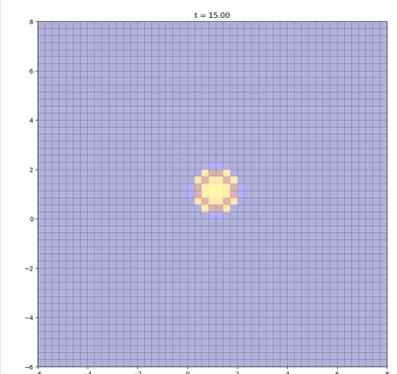
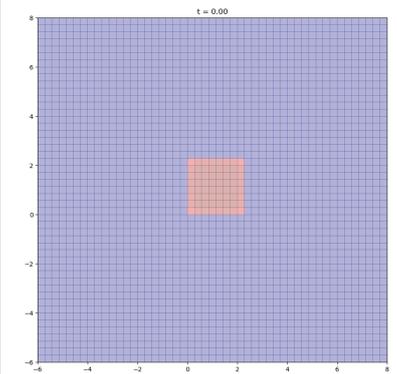
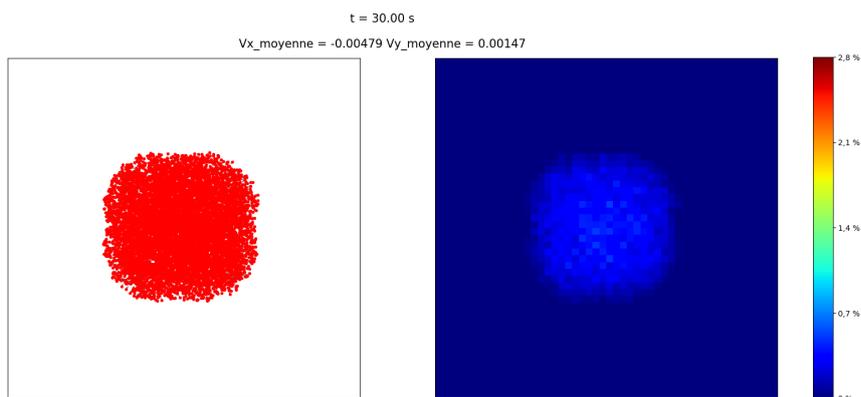
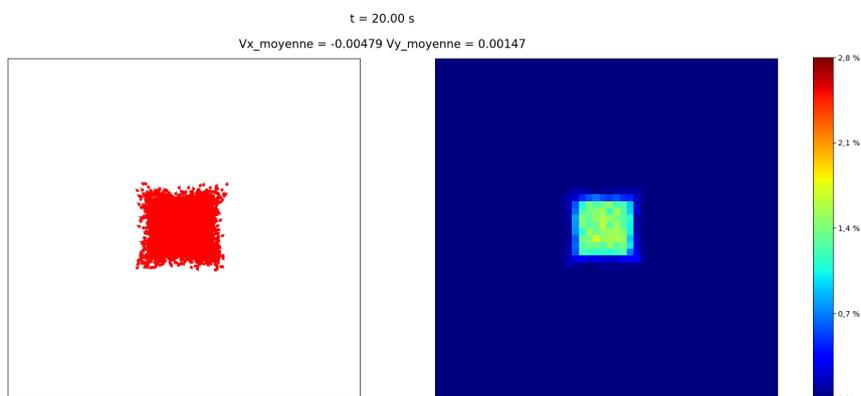
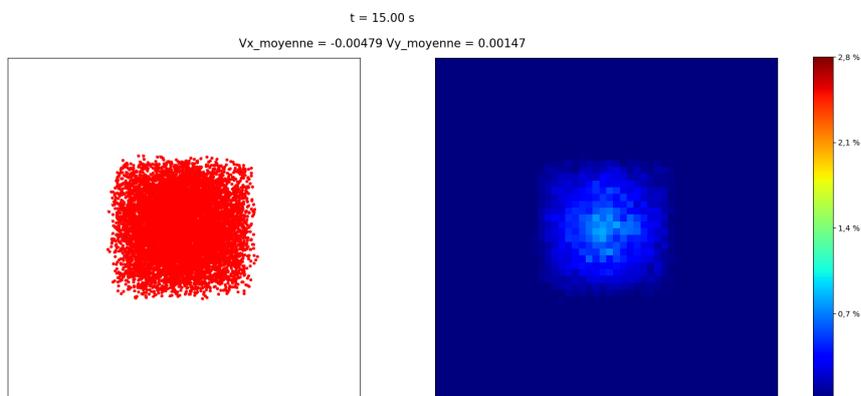
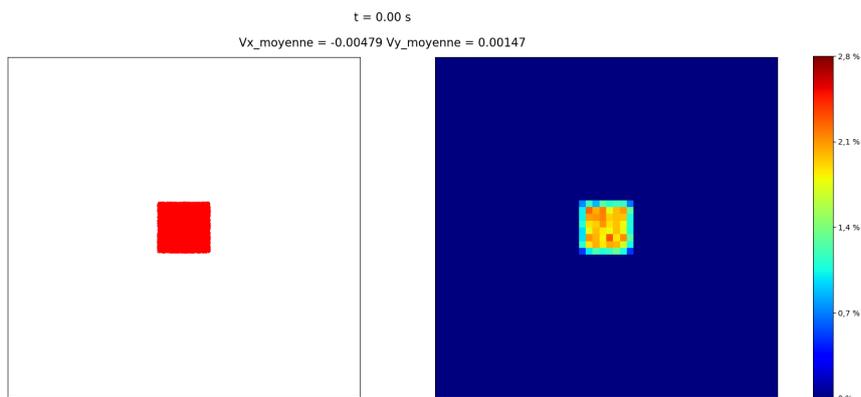
N = 1000

Agrégation



$N = 10000$

Agrégation



## Commentaires :

Malheureusement, nous ne sommes pas parvenus à obtenir les résultats escomptés avec le programme résolvant l'équation d'agrégation pour le modèle de Champ Moyen. En effet, le comportement obtenu est uniquement attractif tandis que la théorie de ce modèle inclut également un phénomène de répulsion lorsque les particules sont très concentrées.

Cependant, nous savons que la convergence du modèle particulaire vers la solution de l'équation d'agrégation est vérifiée. Nous admettons donc que la solution que nous aurions obtenu aurait du être proche de l'histogramme obtenu pour le nombre de particules  $N = 10000$ .

Cette conjecture étant faite, nous voyons que lorsque nous prenons  $N = 50$ , l'histogramme obtenu n'est pas du tout proche de ce que nous attendons, ce qui est cohérent puisque 50 particules est un nombre beaucoup trop petit pour espérer une quelconque convergence. En faisant passer  $N$  à 500 puis 1000, nous observons l'émergence d'un comportement qui se dessine de plus en plus précisément au fur et à mesure de l'augmentation du nombre de particules.

Il nous resterait dès lors à obtenir le résultat espéré pour l'équation d'agrégation afin de savoir si notre hypothèse est vérifiée ou si un plus grand nombre de particules serait nécessaire.

## 2.3.2 Conclusion

Au cours de ce Travail Encadré de Recherche, nous nous sommes intéressés à la simulation de comportements collectifs, observables à l'échelle macroscopique, résultant de considérations particulières, c'est à dire à l'échelle microscopique. Cette optique a été étudiée dans notre première partie au cours de laquelle nous avons pu tester différents modèles menant à l'émergence de comportements distincts.

Nous avons ensuite comparé cette façon de faire avec une approche macroscopique : la résolution de l'équation d'agrégation, tout en cherchant les conditions sous lesquelles les résultats seraient similaires. Une fois ce rapprochement établi, nous pouvons nous rendre compte que chacune de ces deux approches présente ses avantages et ses inconvénients.

D'une part, l'approche particulière est relativement simple à implémenter, mais elle est cependant d'une complexité quadratique. Cela étant compensé par une certaine aisance à paralléliser le processus et ainsi à ne pas obtenir un temps de calcul trop long. En effet nous avons observé que lors du lancement d'une simulation, tous les cœurs de nos ordinateurs étaient sollicités.

D'autre part, la résolution de l'équation d'agrégation est beaucoup plus compliquée à implémenter. La complexité est fortement réduite, de l'ordre de  $N \log(N)$ , cependant il s'avère être encore plus ardu de les paralléliser. Ainsi le temps de calcul est aussi relativement élevé. Lors du lancement de chaque simulation, un seul cœur était exploité.

Nous pourrions donc encore approfondir cette étude avec les enseignements que nous apporterons notre deuxième année de Master. Par exemple par l'apprentissage des volumes finis pour la résolution de l'équation d'agrégation, ou encore la parallélisation des calculs.

*Nous tenons à remercier particulièrement notre enseignant référent, Monsieur Thomas Rey, pour son implication et les nombreuses aides qu'il nous a apportées ainsi que pour sa réactivité.*

# Références

- [1] F. Cucker et S. Smale, *On the mathematics of emergence*, Japan. J. Math. 2 (2007), 197–227.
- [2] S.-Y. Ha et E. Tadmor, *From particle to kinetic and hydrodynamic descriptions of flocking*, Kin. Rel. Mod. 1(3) (2008) 415-435.
- [3] C. W. Reynolds, *Flocks, Herds, and Schools : A Distributed Behavioral Model*, Computer Graphics, 21(4) (1987), 25–34.
- [4] G. Albi, D. Balagué, J.-A. Carrillo et J. Von Brecht, *Stability analysis of flock and mill rings for second order models in swarming*, SIAM J. Appl. Math. 74(3) (2014) 794-818.
- [5] T. Kolokolnikov, Y. Huang et M. Pavloski, *Singular patterns for an aggregation model with a confining potential*, (2012).
- [6] J.-A. Carrillo, Y. Choi et M. Hauray, *The derivation of swarming models : mean-field limit and wasserstein distances*, (2013).
- [7] M. Di Francesco et S. Fagioli, *A nonlocal swarm model for predators-prey interactions*, Math. Mod. Met. App. Sci. 26(2) (2016) 319-355.