

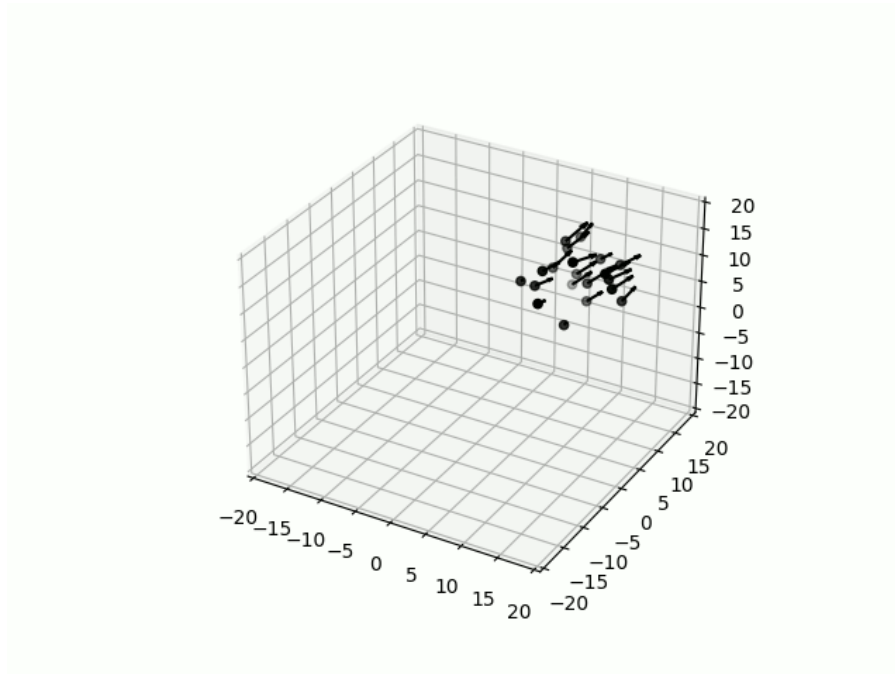
LICENCE 3 DE PHYSIQUE, CHIMIE, ASTROPHYSIQUE, MÉTÉOROLOGIE ET  
ÉNERGIE

---

RAPPORT DU PROJET NUMÉRIQUE

# MODÈLE DE VICSEK

Alexis PEYROUTET, Antoine ROYER



Janvier – Juin 2023

---

# Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Présentation et explications</b>	<b>3</b>
<b>2 Méthode employée</b>	<b>5</b>
2.1 Classes et méthodes . . . . .	5
2.2 Utilisation du code . . . . .	5
<b>3 Premières interprétations physiques</b>	<b>6</b>
3.1 Animations en fichier GIF . . . . .	6
3.2 Mouvements de groupe . . . . .	7
3.3 Autres paramètres en jeu . . . . .	8
3.3.1 Cône de vision . . . . .	8
3.3.2 Paramètre de bruit . . . . .	9
<b>4 Résultats expérimentaux</b>	<b>10</b>
4.1 Résultats historiques de VICSEK . . . . .	10
4.1.1 Paramètre d'alignement en fonction du bruit . . . . .	10
4.1.2 Paramètre d'alignement en fonction de la densité . . . . .	11
4.2 Au-delà du modèle de VICSEK . . . . .	12
4.2.1 Création d'un agent leader . . . . .	12
4.2.2 Mise en place d'une prédation et du paramètre de sensibilité . . . . .	13
4.2.3 Système évolutif . . . . .	14
4.2.4 Ajout d'obstacles . . . . .	14
<b>Conclusion</b>	<b>16</b>

# Introduction

Notre sujet est intitulé « Modèle de VICSEK ». Le but de ce projet numérique est de reproduire de manière numérique le modèle de VICSEK.

Le modèle de VICSEK a été créé par le scientifique Tamás VICSEK. Il s'agit d'un physicien hongrois connu pour ses contributions à la physique statistique, à la biologie et à la dynamique des systèmes. Il est né le 10 mai 1948 ( 74 ans aujourd'hui ) à Budapest. Il est aujourd'hui professeur à l'Université Eötvös LORÁND de Budapest. Ce brillant physicien est d'ailleurs un des membres de l'Académie hongroise des sciences et a reçu de nombreux prix pour ses contributions à la physique, notamment le prix SZÉCHENYI ( en 1999 ) ou encore le prix Lars ONSAGER ( en 2020 ).

Mais Tamás VICSEK est surtout connu pour son travail sur les systèmes auto-organisés ; ces modèles permettent d'étudier les mouvements collectifs. Il a ainsi travaillé sur le comportement d'agents individuels qui interagissent avec d'autres agents aux alentours, ses observations montrent que des motifs de mouvements collectifs émergent d'eux-même.

Le groupe se déplace alors de manière coordonnée sans qu'il n'y ait de leader comme on peut l'observer notamment dans la migration des grues. Nous pouvons citer comme exemples : les bancs de poissons, les regroupements de certains oiseaux, les essaims d'insectes, ou encore le mouvement de foules.

Le premier modèle de VICSEK voit ainsi le jour en 1995.

# Chapitre 1

## Présentation et explications

Le modèle de VICSEK permet d'étudier un groupe d'agents qui se déplace dans un espace.

Chacun des agents a une vitesse donnée ( en norme et en direction ) et va interagir avec ses voisins. Ce qui concrètement se traduit par des changements concernant la norme et la direction de la vitesse.

Nous nous attendons alors à observer la création d'un mouvement de groupe du aux interactions entre les agents.

Cependant, les agents observables dans la vie réelle ne suivent pas toujours le groupe à la perfection, et il peut arriver qu'un agent s'écarte, de manière aléatoire, des autres. C'est pour cela que VICSEK a introduit une notion de bruit dans son modèle. En effet, à chaque pas de temps, chaque agent va prendre la direction moyenne des agents autour de lui et à cette direction va venir se superposer un bruit qui le fera peut-être dévier dans une autre direction.

En augmentant significativement le bruit, le groupe perd son mouvement collectif et les agents prennent alors des directions aléatoires.

Le modèle de VICSEK s'implémente très simplement puisqu'il se réduit à deux équations :

$$\begin{cases} \Theta_i(t + dt) &= \Theta_{j|r_i-r_j|<r} + \eta_i(t) \\ r_i(t + dt) &= r_i(t) + v_i \Delta t \end{cases}$$

dans lesquelles  $r_i$  la position de chaque individu donnée par un vecteur de position, nous prendrons  $i$  comme indice de l'agent en question et  $t$  le temps. Nous noterons également  $\eta$  le bruit et  $\Theta$  pour l'angle définissant la direction de sa vitesse. Ici,  $\Theta_{j|r_i-r_j|<r}$  nous indiquera la direction moyenne des vitesses des agents dans un cercle de rayon  $r$ . L'indice  $j$  représentera alors l'ensemble des voisins de  $i$  compris dans ce cercle.

Ce qui est intéressant, c'est que nous pouvons, en modifiant certains paramètres du système étudié, observer un mouvement de foule plus fort ou plus faible. Nous pourrions alors jouer sur la surface et les dimensions du plan étudié, le nombre d'agent et donc par conséquent la densité de population et même le bruit.

Le modèle de VICSEK est important pour étudier le comportement de certains animaux en biologie ou encore l'étude des foules. Ce modèle peut même être utile à la construction de bâtiment. En effet, le comportement des foules peut être intéressant dans la conception d'entrées et de sorties d'un espace fermé, notamment dans un moment de panique. La foule va s'éloigner du danger est emprunter les sorties. Il est alors crucial de prévoir le comportement des agents pour placer les sorties de manière à ce que le débit d'agent sortant soit le plus important possible.

Nous pouvons également retrouver le modèle de VICSEK dans la robotique. C'est un précieux outil pour la technologie du monde moderne. Il peut être utilisé dans des programmes informatiques qui gèrent le déplacement de systèmes de robots (comme les drones).

C'est avec tout cela que nous essayerons, à travers ce projet, de reproduire numériquement des mouvements collectifs et ainsi étudier le modèle de VICSEK.

# Chapitre 2

## Méthode employée

### 2.1 Classes et méthodes

Pour ce projet, la programmation orientée objet s’est naturellement imposée. Nous utilisons ainsi deux classes principales appelées **Agent** et **Group** qui fixent respectivement les paramètres de l’agent et du groupe. Assez simplement, la classe **Group** contient une liste d’instances de la classe **Agent** et permet de les représenter dans l’espace et le temps. La classe **Agent** permet d’encapsuler toutes les données de chaque agent : position, vitesse et direction.

Ainsi, nous pouvons retrouver dans chaque classe, plusieurs méthodes qui vont nous aider à mieux définir les groupes et les agents ainsi qu’à les faire évoluer.

### 2.2 Utilisation du code

Nous avons commencé par programmer les classes avec des méthodes basiques qui permettaient d’initialiser, de faire évoluer et de représenter les agents.

Une fois les deux équations implémentées, nous avons créé la fonction `group_generator` qui nous permettait de créer simplement un groupe en prenant des paramètres aléatoires dans des limites fixées pour les agents.

Après avoir créé notre groupe avec `group_generator`, nous le faisons évoluer grâce à la méthode `Group.run` et nous générions une image au format PNG en deux ou trois dimensions avec `Group.compute_figure`.

Cela nous a permis d’avoir des premières représentations de nos groupes d’agents :

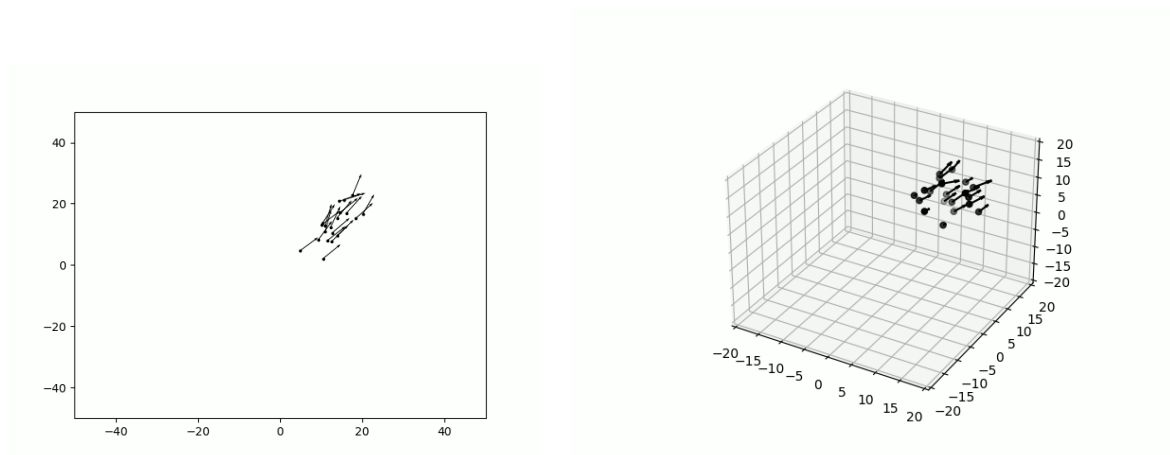


FIGURE 2.1 – Premières images obtenues en 2D et 3D

# Chapitre 3

## Premières interprétations physiques

### 3.1 Animations en fichier GIF

Pour pouvoir observer un mouvement, il est plus utile de regarder une vidéo que des images. Nous avons alors créé une méthode : `Group.compute_animation` qui utilise la classe `Artist_animation` du module `matplotlib`. Nous arrivions alors à générer des fichiers GIF avec le nombre de frames souhaité.

Après avoir généré plusieurs animations avec des groupes de tailles différentes, nous avons pu déjà tirer quelques conclusions. En effet, nous observions des mouvements de groupes. Les agents qui avaient des positions de départ aléatoires, sont influencés les uns les autres selon la distance avec leurs voisins.

On observe d'ailleurs des mouvements collectifs plus importants quand la densité d'agent est plus forte. En revanche, quand les agents sont moins nombreux dans un même espace, nous observons davantage de formations de petits groupes ou des agents solitaires. Nous pouvons régler ce paramètre de densité avec `length` qui correspond à la longueur de l'espace considéré.

Cela s'explique par le fait que les agents ne se voient plus.

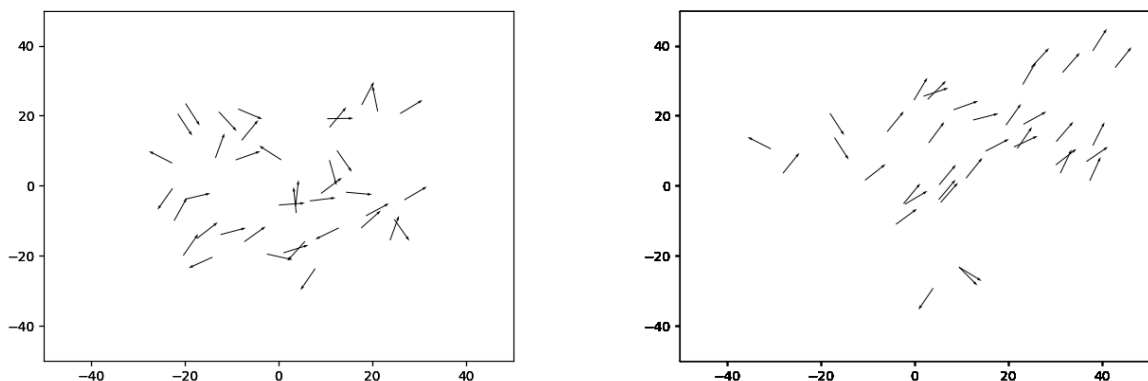


FIGURE 3.1 – Début et fin d'une animation 2D

Nous avons très vite privilégié la représentation 2D pour la suite du projet. En effet, celle-ci permet d'observer plus facilement le comportement des agents, et les différents groupe créés.

## 3.2 Mouvements de groupe

Pour mieux observer les mouvements de groupe, nous avons décidé de mettre une couleur à nos agents selon leur direction. Cela permet de mieux visualiser les différents groupes et de s'affranchir des flèches qui étaient devenue gênantes pour bien discerner le mouvement collectif à haute densité.

Nous avons gardé cette représentation de l'angle des agents pour tout le reste du projet.

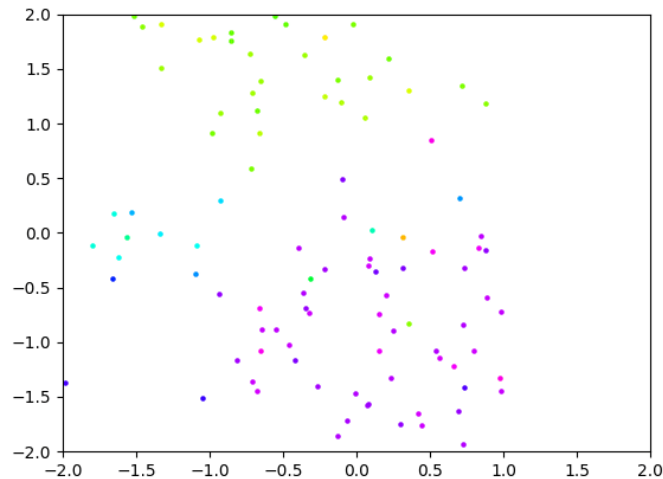


FIGURE 3.2 – Image colorée pour la visualisation de groupe

Sur cette nouvelle figure 3.2, nous voyons bien les différents groupes créés. On distingue encore mieux les mouvements lors de l'animation.

Nous avons d'ailleurs pu observer une organisation intéressante des agents sur certaines animations. En effet, les agents se regroupent premièrement en plusieurs petits groupes ( sur l'image de gauche de la figure 3.4, on discerne en effet trois groupe principaux en violet, vert et cyan ). Enfin, les petits groupes s'unissent pour former un seul et même grand groupe.

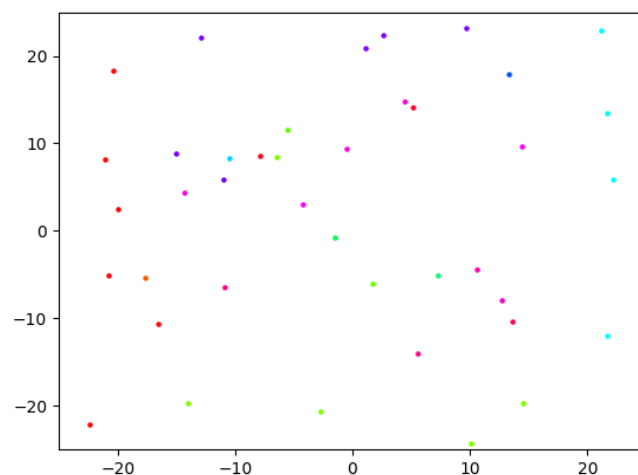


FIGURE 3.3 – Image de départ



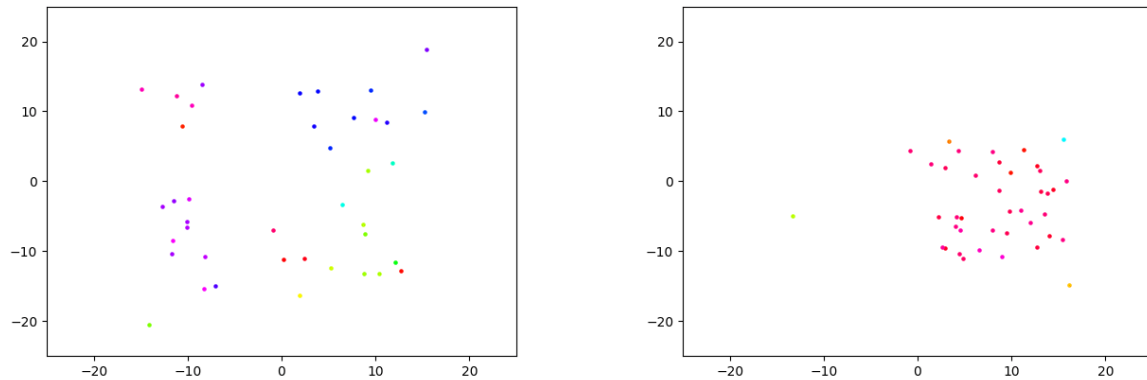


FIGURE 3.4 – Formations de petits groupe puis d’un seul et même groupe

Ceci montre encore un fois très bien l’influence entre les agents.

### 3.3 Autres paramètres en jeu

#### 3.3.1 Cône de vision

Afin de mieux visualiser ce que peut voir un agent en particulier, nous avons alors décidé de rajouter un cône de vision. Pour ce faire, nous importons le module `import matplotlib.patches` qui nous permettra de tracer ces cônes en 2 dimensions.

Nous serons ainsi capable de mieux voir comment un agent réagit selon ce qu’il voit.

Ainsi, un agent un considéré comme voisin s’il est dans le cône de vision de l’agent testé. En refaisant le même test que précédemment, nous observons que les agents restent alignés moins longtemps.

En effet, les agents étant plus sensible à l’orientation pour voir les autres, si le bruit augmente, les agents vont avoir des déviations angulaires plus importante et peuvent donc perdre de vue les autres agents plus facilement ce qui fait chuter le paramètre d’alignement plus rapidement.

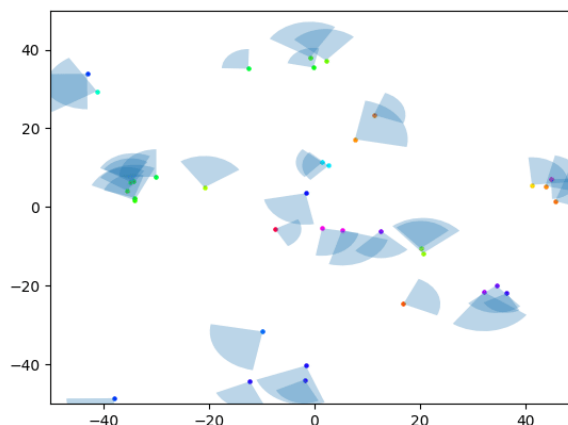


FIGURE 3.5 – Image de l’animation générée avec cône de vision

Nous avons cependant préféré retirer la représentation visuelle de ce cône dans la suite du projet. En effet, l'animation générée devenait trop chargée et riche en informations. Il était alors difficile de bien observer le comportement des agents.

### 3.3.2 Paramètre de bruit

Le bruit correspond à l'influence des voisins sur un agent du groupe. Plus le bruit est fort, plus celui-ci aura tendance à ne pas s'occuper de ses voisins et prendre sa propre direction. À l'inverse, avec un bruit qui tend vers zéro, l'agent aura tendance à imiter ses voisins et prendre une direction similaire aux agents autour de lui.

Nous avons déjà pu observer l'impact du bruit sur les populations. Nous voyons alors que ce paramètre est capital pour l'observation de mouvements collectifs. Si celui-ci est trop fort, les agents feront route seuls et ne s'occuperont pas du mouvement des voisins. En revanche, les mouvements collectifs seront davantage présents avec un bruit qui tend vers zéro.

Par exemple, en regardant ces images :

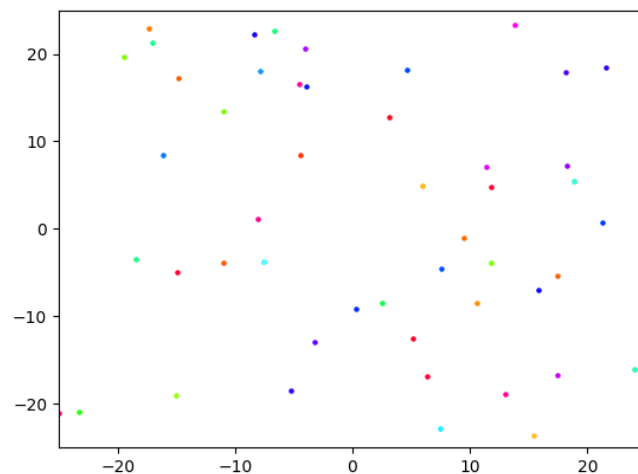


FIGURE 3.6 – Visualisation de l'impact du bruit ( bruit fort dans ce cas )

Nous voyons bien ici que les agents ont des directions plutôt différentes. Nous avons un bruit qui est fort dans ce cas, et nous remarquons que les agents ont tendance à prendre des directions sans s'occuper de leurs voisins. Il est alors évident que le bruit est un paramètre essentiel dans le modèle de VICSEK.

# Chapitre 4

## Résultats expérimentaux

### 4.1 Résultats historiques de VICSEK

#### 4.1.1 Paramètre d'alignement en fonction du bruit

Nous avons commencé par chercher à reproduire les résultats obtenus par VICSEK en reprenant les mêmes paramètres.

Chaque agent n'est ainsi influencé que par ses plus proches voisins et évolue dans un espace torique. En laissant la densité constante et en faisant varier le bruit pour voir son influence sur le paramètre d'alignement, nous observons alors un profil de transition de phase.

Sur la figure suivante chaque points est la moyenne sur cinq mesures de 200 pas et 50 agents.

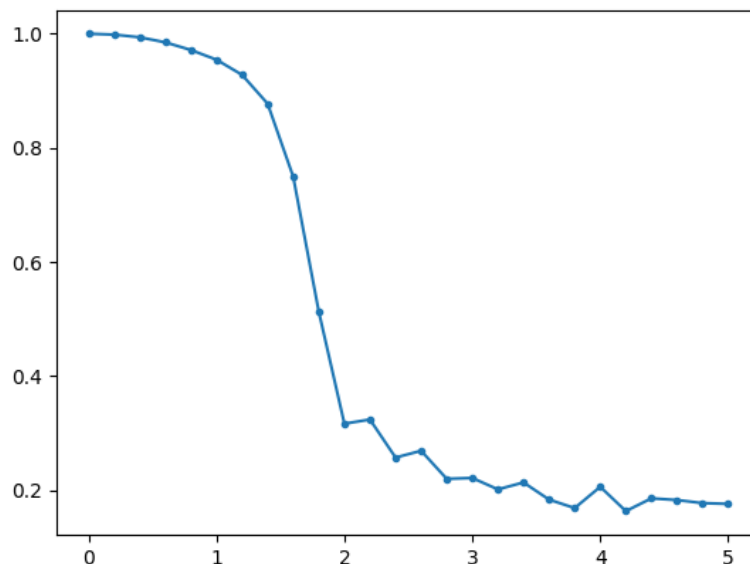


FIGURE 4.1 – Paramètre d'alignement en fonction du bruit

**Mettre aussi une figure avec différents nombre d'agent et LEGENDER LES AXES QUI N'EN ONT PAS** Pour un bruit très faible, les agents sont alignés avec un paramètre d'alignement de 1 ce qui est cohérent puisqu'ils vont s'aligner sans aucune part d'aléatoire. Pour un bruit élevé, ce qui correspond en fait à une probabilité d'avoir une déviation angulaire importante par rapport à la direction moyenne, les agents sont très peu alignés

pour un même nombre d'itération.

Nous avons ici exactement les mêmes résultats que ceux obtenus par VICSEK et son équipe.

Nous en avons d'ailleurs profité pour mesurer l'effet du cône de vision sur le paramètre d'alignement, nous avons tracé ce graphe avec 50 mesures par points sur la figure 4.2.

Nous pouvons ainsi constater que les agents ont tendance à prendre la même direction plus rapidement avec un cône de vision. C'est donc un paramètre très important dans le modèle.

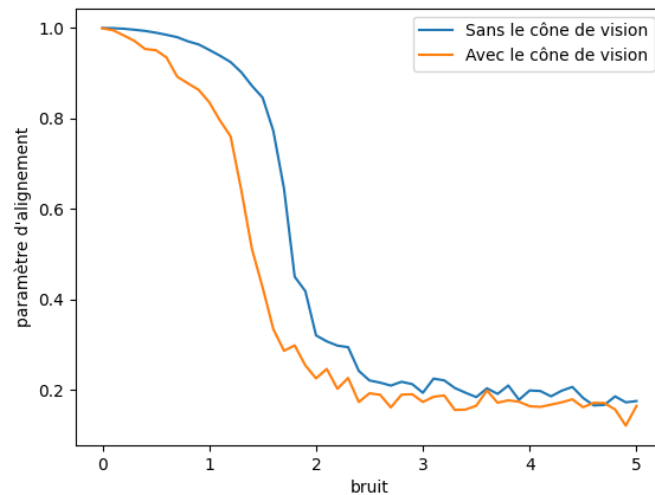


FIGURE 4.2 – Effet du cône de vision sur le paramètre d'alignement

#### 4.1.2 Paramètre d'alignement en fonction de la densité

VICSEK avait aussi mesuré le paramètre d'alignement en fonction de la densité d'agent dans l'espace ( avec un bruit constant ). En traçant le paramètre d'alignement en fonction de la densité, nous obtenons pas exactement la même figure que VICSEK même si cela s'en rapproche.

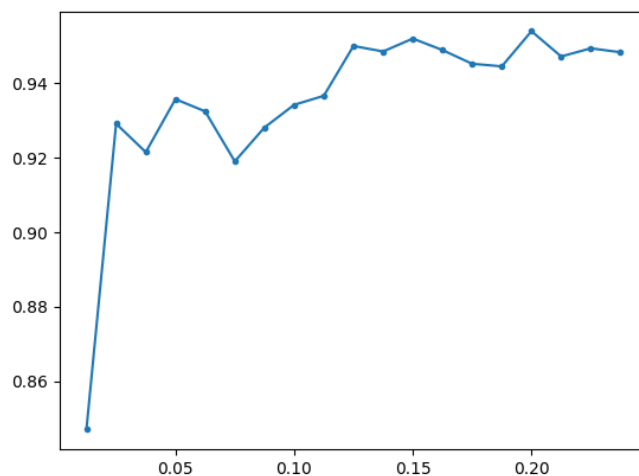


FIGURE 4.3 – Paramètre d'alignement en fonction de la densité d'agent ( bruit=1 )

Nous pouvons tout de même en déduire que plus le nombre d'agent est élevé dans un même espace ( et donc la densité aussi ), plus les agents s'aligneront sur la même direction rapidement. Nous pouvons même rajouter qu'à partir d'une valeur de densité d'environ de 0.13 agents par unité d'espace.

## 4.2 Au-delà du modèle de VICSEK

### 4.2.1 Création d'un agent leader

Nous avons maintenant voulu nous écarté un peu du modèle de VICSEK, en étudiant un nouveau type de mouvement collectif. Nous avons alors créé un nouveau type d'agent, dit leader, qui influence davantage les agents normaux. Nous pouvons définir à quel point celui-ci à de l'influence ( nous pouvons choisir une influence équivalente à  $n$  agents ).

Nous nous attendions à observer des groupes en forme de pyramide. C'est à dire une organisation hiérarchique des agents pour créer un mouvement de groupe. Nous pouvons par exemple observer ce comportement chez certains animaux, notamment lors de la migration des grues.

Nous avons choisi de représenter l'agent leader en noir.

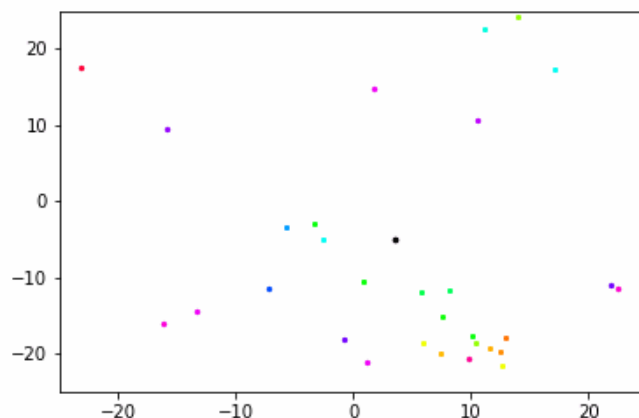


FIGURE 4.4 – Effet observée avec un agent leader

En effet, nous avons observé plusieurs fois une organisation en « triangle ou en arc de cercle », comme une sorte de hiérarchie. L'agent leader produit un mouvement de groupe organisé différemment de ce qu'on a pu observer dans le modèle de VICSEK.

Ainsi, cela montre que le mouvement collectif peut être amené de différentes manières et qu'il n'y a pas de méthode unique pour un déplacement d'un ensemble d'agent.

Nous avons d'ailleurs pu observer quelque chose d'intéressant. En effet, sur une animation, nous pouvons voir un groupe d'agent qui suit l'agent leader pendant un moment. Mais, ce groupe croise un autre groupe d'agent. Les agents qui suivait l'agent leader se sont mis subitement à suivre le nouveau groupe.

Ainsi, nous en déduisons que le groupe a eu plus d'influence que l'agent leader à ce moment là.

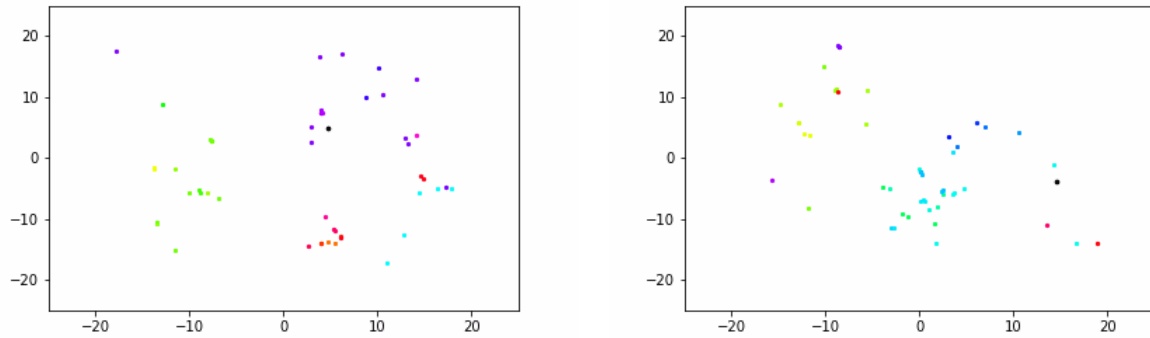


FIGURE 4.5 – Situation intéressante avec un agent leader

### 4.2.2 Mise en place d’une prédation et du paramètre de sensibilité

Nous avons trouvé intéressant de rajouter des agents dit répulsifs, qui joue le rôle de prédateurs. Ces nouveaux agents sont plus rapides que leurs proies. Ainsi, nous avons pu voir comment s’organisent les agents face à la menace.

Nous avons choisit de paramétrer la réaction des agents quand cet agent répulsif rentre dans leur cône de vision. L’agent prendra immédiatement le vecteur de sens opposé à celui vers l’agent répulsif. Nous avons choisit de représenter les agents répulsifs en noir.

Avec cette expérience, nous constatons que le mouvement de groupe est toujours présents avec un seul agent répulsif. Cet effet collectif est même souvent renforcé. Les agents fuient dans la même direction, tout en se servant de leur influence les uns sur les autres.

Cependant, lorsqu’il y a plusieurs agents répulsifs, le mouvement collectif n’est plus observé ou très peu. Ces prédateurs ( qui prennent des directions différentes ) sèment la panique au sein du groupe qui perd visiblement toute cohésion. Chaque agent développe alors un instinct de survie qui le pousse à fuir. La panique et la fuite prennent visiblement le pas sur la cohésion de groupe ( cf. figure 4.6 ).

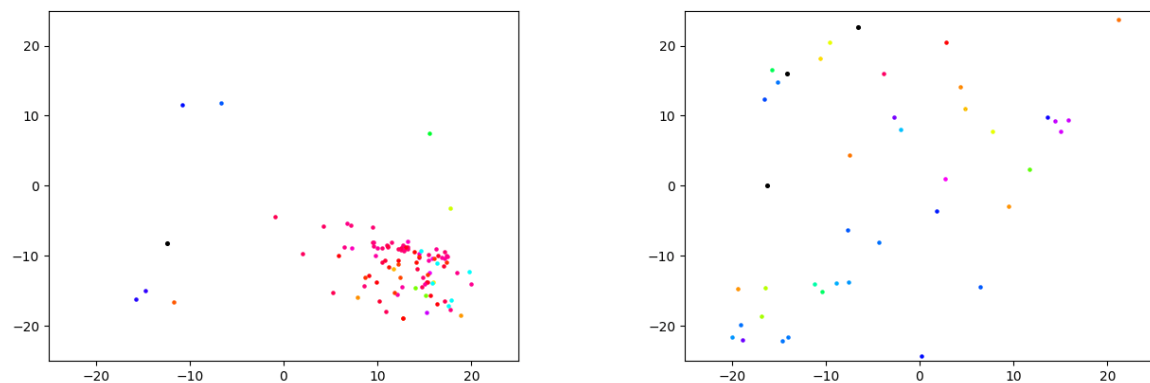


FIGURE 4.6 – Effet d’un ( à gauche ) ou plusieurs agents répulsifs ( à droite ) sur le groupe

Nous avons ensuite décidé de rajouter un nouveau paramètre que nous appelons **fear** dans notre code. Il s’agit de la sensibilité des agents face aux prédateurs, c’est à dire à la peur des agents pour les agents répulsifs.

Ainsi, les agents prennent plus facilement la fuite face aux prédateurs.

### 4.2.3 Système évolutif

En ajoutant des agents répulsifs, nous avons eu l'idée de créer un système de « mort » où les agents touchés par un agent répulsifs sont retirés de la liste des agents et placés dans un autre attribut du groupe. Cela permet notamment de comparer les caractéristiques des agents morts avec celles des survivants et de voir quels paramètres permettent de survivre. En mettant des valeurs limites pour le bruit et la sensibilité aux agents répulsifs et en moyennant les résultats sur 10 mesures, nous obtenons les résultats suivants :

bruit	sensibilité	pourcentage de survivants
1	0	30
0	1	79
1	1	86
0	0	39

Nous voyons bien que statistiquement, les agents ont plus de chance de survivre avec un bruit et une sensibilité à 1. En effet, dans ce cas là, les agents avec une grande sensibilité cherche à fuir à tout prix le prédateur. De plus, le bruit fort fait que les agents ne cherche pas à imiter leurs voisins, ce qui ne perturbe pas la fuite qu'il avait entrepris.

Cependant, en enlevant le bruit, nous voyons tout de même que le pourcentage de survivant reste très élevé. La sensibilité est un élément clé pour la survie des agents.

Au contraire, les agents ont peu de chance de survie avec un bruit fort et un sensibilité nulle. En effet, les agents n'ont pas peur de leurs prédateurs, ce qui devient très dangereux. De plus, le bruit élevé, fait qu'il n'y a aucun mouvement de groupe.

Cependant, en retirant le bruit, nous voyons le que pourcentage de survivant est à peine plus haut. Nous revenons à la même conclusion : la sensibilité est un élément clé pour la survie des agents.

### 4.2.4 Ajout d'obstacles

De plus, nous avons voulu voir comment réagissent les agents face à des obstacles. Nous voulions voir le mouvement de groupe est conservé.

Cette approche permet de se placer dans des conditions encore plus réalistes. Par exemple, un banc de poissons peut rencontrer des rochers lors de son déplacement.

Nous avons alors crée des murs, quifont office d'agents répulsifs immobiles. A leur vue, les agents font marche arrière pour éviter l'obstacle.

Nous voyons bien que les agents rebrousse un peu chemin. En réalité, nous pourrions plutôt nous attendre à ce que le groupe contourne l'obstacle. Il y a tout de même quelques agents qui ont visiblement contourné l'obstacle, mais la majorité du groupe fait demi-tour.

Cependant, il faut admettre que l'ajout d'obstacle est assez difficile à coder.

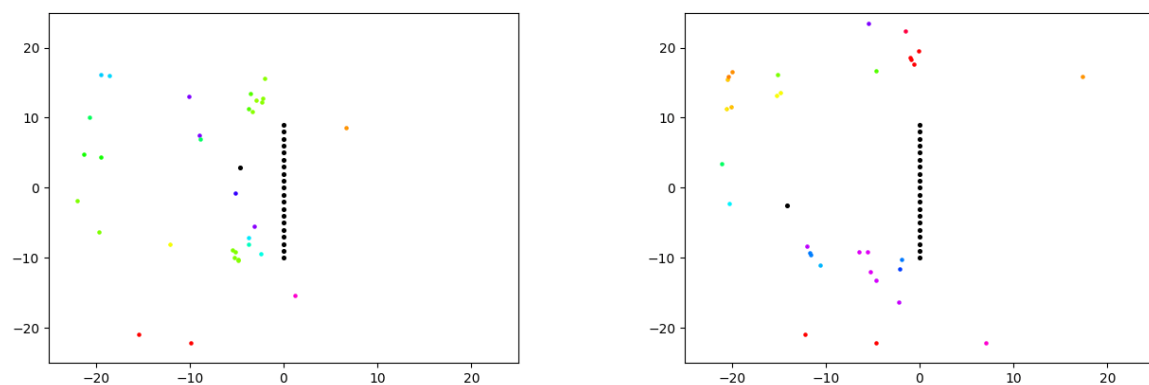


FIGURE 4.7 – Effet d’un obstacle sur le mouvement des agents ( arrivée des agents vers le mur à gauche, directions prises par les agents après avoir vu le mur à droite )



# Conclusion