

Collective Behaviors of Interacting Active Brownian Particles: Simulations and Analysis

University of Pisa

Niccolò Picchiarelli

March 3, 2025

Contents

1	Introduction	1
1.1	Microrobotics and CELLOIDS	2
1.2	Objectives	3
1.2.1	Objective 1: Minimal model accounting for aligning and non-aligning interactions	3
1.2.2	Objective 2: Analysis	4
1.2.3	Objective 3: Inference and Deep Learning	4
1.3	State of the Art	4
1.3.1	Interactions and emerging behaviours	5
1.3.2	Experimental analysis of collective behaviors	9
1.3.3	Inference of interaction potentials	13
1.4	Background	18
1.4.1	Passive Brownian Motion	18
1.4.2	ABP simulations	20
1.4.3	Stochastic integration	20
1.4.4	Algorithms	22
2	Interaction Implementation	25
2.1	Simulation Framework	25
2.2	Methods	26
2.3	Boundary Conditions and Hard-Spheres Correction	27
2.3.1	Boundary Conditions	27
2.3.2	Hard-Spheres Correction	28
2.3.3	Modified Periodic Boundary Steric Interactions	29
2.4	Interaction Forces	31
2.4.1	Central Potentials	31
2.4.2	Tested Potentials	31
2.4.3	Interaction Range	32
2.4.4	Aligning Interactions	34
2.5	Qualitative Model Validation	35
2.5.1	Qualitative Comparison between Simulations and Experiments	36
2.6	Conclusions to Chapter 2	42
3	Analysis of Structure and Dynamics	43
3.1	Analysis Functions	43
3.1.1	Pair Correlation Function	43
3.1.2	Cluster Size	44

3.1.3	Local and Global Polarization	44
3.1.4	Local Polarization	45
3.2	Flocking as a Phase Transition	45
3.2.1	Methods	46
3.2.2	Results	48
3.3	Clustering and Local Order	50
3.3.1	Velocity	52
3.3.2	Velocity Distribution	53
3.3.3	Orientation Coupling	53
3.3.4	Angular Velocity	55
3.4	Conclusions to chapter 3	57
4	Discovering Interaction Laws	85
4.1	Methods	85
4.1.1	The Graph Neural Network	86
4.1.2	GNN Implementation	87
4.1.3	Simulations	88
4.2	Training and Testing	89
4.3	Results	89
4.4	Conclusions to Chapter 4	90
5	Conclusions	95

List of Figures

1.1	(a) Phase diagram in the g –Pe plane. (b-f) representing snapshots for different values of parameters. Adapted from [14]	6
1.2	Different types of polarization. Adapted from [14].	7
1.3	Adapted from [14].	7
1.4	(a) Density as a function of $1/D_r$ for gaseous and cluster phase. (b) Velocity alignment order parameter R as a function of $1/D_r$. (c-e) Representative snapshots for different values of D_r . Adapted from [15].	8
1.5	Schematic of nonreciprocal interaction with vision cone. Adapted from [16].	9
1.6	Representative snapshots for the configurations that model in [16] can achieve.	10
1.7	Classification of approach and departure configuration. Adapted from [17].	11
1.8	(a) The demixing process. (b) Steady state of binary colloidal flocks. (c) Radial density profiles of the two species. Adapted from [18]	12
1.9	Experimental radial probability densities. Adapted from [19].	12
1.10	Comparison between relative probability density in (a) experiments and (b) simulations. Adapted from [19].	13
1.11	Potential shapes for different values of parameters [20].	14
1.12	Equivalence between an equilibrium system with attraction and non-equilibrium one without attraction. Adapted from [20].	15
1.13	The GNN learning workflow in the case of Newtonian dynamics. Adapted from [22].	17
1.14	The ActiveNet structure. Adapted from [23].	18
2.1	If in one step two particles superimpose, hard-spheres correction moves them at one diameter distance [8]	28
2.2	Orange particle is sticking out on the left side. Its <i>real</i> distance from blue particle on x-axis is Δx . Its periodic projection (faded orange) has a distance from blue particle $\Delta x_p < 2R$, hence superimposing.	30
2.3	Tested interaction forces for given values of their parameters. For simplicity here, everything is kept non-dimensional.	32
2.4	Cutoff for Lennard Jones force at 1/10 of the minimum. The cutoff distance is $\sim 1.9\sigma$	33

2.5	Red: original points with Z as central point and its interaction box sticking out on the opposite side. Blue: shifted points w.r.t. Z. Green: shifted points with PBC applied.	34
2.6	Red point: position from which interaction distances are calculated and to which force is applied. In this case, interacting position is at $-0.5R$ from particle's center, <i>i.e.</i> on the back side compared to the direction of self propulsion.	35
2.7	In the orange circle, formation and dissolution of a 2-particles cluster.	37
2.8	In the orange circle, formation and dissolution of a 2-particles cluster.	38
2.9	In the orange circle, direct formation and dissolution of 3-particles cluster.	39
2.10	In the orange circle, formation and dissolution of 3-particles cluster starting from a couple.	40
2.11	In the orange circle, formation and dissolution of 3-particles cluster starting from a couple.	41
3.1	"Illustration of DBSCAN cluster analysis (minPts=3). Points around A are core points. Points B and C are not core points, but are density-connected via the cluster of A (and thus belong to this cluster). Point N is Noise, since it is neither a core point nor reachable from a core point". [36].	45
3.2	Behavior of two particles with explicitly aligning interactions, compared with excluded volume steric interaction only. Adapted from [14].	46
3.3	Flocking transition: all particles in the system tend to polarize after few seconds. Images from top left to bottom right correspond to situation after 1 s, 5 s, 10 s, 15 s, 20 s and 25 s	47
3.4	Temporal evolution of polarization for $v = 10 \mu\text{m s}^{-1}$, and several values of the off center coupling. Black dashed line is the number of points we discarded.	48
3.5	Standard deviation of polarization and susceptibility versus blocking size. Black dashed line is the block size of 2500 points that was used for this work. For this plot, $v = 10 \mu\text{m s}^{-1}$, $\alpha = 0.25$. Both a blocking procedure and a JackKnife resampling are used here.	49
3.6	Diagrams for polarization and relative susceptibility for the transition to ordered phase.	51
3.7	Representative simulation snapshots corresponding to values of velocity and α : (a) $v = 10 \mu\text{m s}^{-1} \alpha = 0.5$, (b) $v = 20 \mu\text{m s}^{-1} \alpha = 0.5$, (c) $v = 10 \mu\text{m s}^{-1} \alpha = -0.5$, (d) $v = 20 \mu\text{m s}^{-1} \alpha = -0.5$	52
3.8	Local (blue) and global (green) polarization corresponding to values of velocity and α : (a) $v = 10 \mu\text{m s}^{-1} \alpha = 0.5$, (b) $v = 20 \mu\text{m s}^{-1} \alpha = 0.5$, (c) $v = 10 \mu\text{m s}^{-1} \alpha = -0.5$, (d) $v = 20 \mu\text{m s}^{-1} \alpha = -0.5$	55
3.9	Maximum cluster size and number of clusters corresponding to values of velocity and α : (a) $v = 10 \mu\text{m s}^{-1} \alpha = 0.5$, (b) $v = 20 \mu\text{m s}^{-1} \alpha = 0.5$, (c) $v = 10 \mu\text{m s}^{-1} \alpha = -0.5$, (d) $v = 20 \mu\text{m s}^{-1} \alpha = -0.5$	57

3.10 Radial distribution function at last instant of simulation corresponding to values of velocity and α : (a) $v = 10 \mu\text{m s}^{-1} \alpha = 0.5$, (b) $v = 20 \mu\text{m s}^{-1} \alpha = 0.5$, (c) $v = 10 \mu\text{m s}^{-1} \alpha = -0.5$, (d) $v = 20 \mu\text{m s}^{-1} \alpha = -0.5$	59
3.11 Representative simulation snapshots for a normal velocity distribution with $\mu = 10 \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 1 \mu\text{m s}^{-1}$, (b) $\sigma = 2 \mu\text{m s}^{-1}$, (c) $\sigma = 5 \mu\text{m s}^{-1}$, (d) $\sigma = 10 \mu\text{m s}^{-1}$	60
3.12 Local (blue) and global (green) polarization for a normal velocity distribution with $\mu = 10 \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 1 \mu\text{m s}^{-1}$, (b) $\sigma = 2 \mu\text{m s}^{-1}$, (c) $\sigma = 5 \mu\text{m s}^{-1}$, (d) $\sigma = 10 \mu\text{m s}^{-1}$	62
3.13 Maximum cluster size and number of clusters for a normal velocity distribution with $\mu = 10 \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 1 \mu\text{m s}^{-1}$, (b) $\sigma = 2 \mu\text{m s}^{-1}$, (c) $\sigma = 5 \mu\text{m s}^{-1}$, (d) $\sigma = 10 \mu\text{m s}^{-1}$	64
3.14 Radial distribution function at last instant of simulation for a normal velocity distribution with $\mu = 10 \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 1 \mu\text{m s}^{-1}$, (b) $\sigma = 2 \mu\text{m s}^{-1}$, (c) $\sigma = 5 \mu\text{m s}^{-1}$, (d) $\sigma = 10 \mu\text{m s}^{-1}$	66
3.15 Transient time (time needed to reach 95% of maximum) for global polarization as a function of α . Only positive values of off center coupling (where the system polarizes) are showed here.	66
3.16 Representative simulation snapshots corresponding to values of off center parameter α : (a) $\alpha = -0.75$, (b) $\alpha = -0.5$, (c) $\alpha = -0.25$, (d) $\alpha = 0$, (e) $\alpha = 0.25$, (f) $\alpha = 0.5$, (g) $\alpha = 0.75$	68
3.17 Local (blue) and global (green) polarization for corresponding to values of off center parameter α : (a) $\alpha = -0.75$, (b) $\alpha = -0.5$, (c) $\alpha = -0.25$, (d) $\alpha = 0$, (e) $\alpha = 0.25$, (f) $\alpha = 0.5$, (g) $\alpha = 0.75$	71
3.18 Maximum cluster size and cluster number corresponding to values of off center parameter α : (a) $\alpha = -0.75$, (b) $\alpha = -0.5$, (c) $\alpha = -0.25$, (d) $\alpha = 0$, (e) $\alpha = 0.25$, (f) $\alpha = 0.5$, (g) $\alpha = 0.75$	74
3.19 Radial distribution function at last instant of simulation corresponding to values of off center parameter α : (a) $\alpha = -0.75$, (b) $\alpha = -0.5$, (c) $\alpha = -0.25$, (d) $\alpha = 0$, (e) $\alpha = 0.25$, (f) $\alpha = 0.5$, (g) $\alpha = 0.75$	77
3.20 Representative simulation snapshots for a normal angular velocity distribution with $\mu = 0 \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 0 \mu\text{m s}^{-1}$, (b) $\sigma = 0.5 \mu\text{m s}^{-1}$, (c) $\sigma = 1 \mu\text{m s}^{-1}$, (d) $\sigma = 2 \mu\text{m s}^{-1}$	78
3.21 Local (blue) and global (green) polarization for a normal angular velocity distribution with $\mu = 0 \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 0 \mu\text{m s}^{-1}$, (b) $\sigma = 0.5 \mu\text{m s}^{-1}$, (c) $\sigma = 1 \mu\text{m s}^{-1}$, (d) $\sigma = 2 \mu\text{m s}^{-1}$	80
3.22 Maximum cluster size and cluster number for a normal angular velocity distribution with $\mu = 0 \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 0 \mu\text{m s}^{-1}$, (b) $\sigma = 0.5 \mu\text{m s}^{-1}$, (c) $\sigma = 1 \mu\text{m s}^{-1}$, (d) $\sigma = 2 \mu\text{m s}^{-1}$	82
3.23 Radial distribution function at last instant of simulation for a normal angular velocity distribution with $\mu = 0 \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 0 \mu\text{m s}^{-1}$, (b) $\sigma = 0.5 \mu\text{m s}^{-1}$, (c) $\sigma = 1 \mu\text{m s}^{-1}$, (d) $\sigma = 2 \mu\text{m s}^{-1}$	84
4.1 Schematic of our GNN.	87

4.2	Potential used in GNN training.	88
4.3	Learning curve for elastic potential (left) and Lennard Jones (right). Loss is the absolute difference between predicted and ground truth velocities.	89
4.4	Representative message-force plots of early epochs of training for spring potential (a) and Lennard Jones (b).	91
4.5	Representative message-force plots of last epoch of training for spring potential (a) and Lennard Jones (b).	92

List of Tables

2.1 Structure definition of ABPE in Julia	26
---	----

Chapter 1

Introduction

The phenomenon of (passive) Brownian Motion is a well investigated topic in physics since 1827, when Robert Brown studied pollen grains suspended in water, finding out that they moved in an erratic, extremely irregular way. After experimenting with non-organic suspended particles, he concluded that this random motion must be caused by the fluid. Physically, it is possible to define Brownian motion as the effect of the impacts between the fluid particles and the suspended grain: since fluid's motion can be described by thermodynamics at particle level as a series of random fluctuations, impacts will be random as well, leading to an intrinsic stochasticity in the position of the particle. A notable contribution in the mathematical formalization of Brownian motion, based on random impacts, was made by Einstein and Langevin [1]. Later, Ornstein and Uhlenbeck [2] studied some fundamental properties such as the mean-square displacement of a Brownian particle in 2D. Although relatively simple, this model is able to explain the statistical properties of particles suspended in a medium, but more can be added to it to mimic real world behaviors. There are some kind of *particles*, such as bacteria, algae or other living beings that are able to propel themselves with a deterministic velocity, leading to more complex behaviors than the ones observed with pollen grains in water. One of the models that can be created adding deterministic properties along the stochastic behavior of a Brownian motion is Active Brownian Particles (ABPs).

Active particles are distinguished by their ability to extract energy from the environment and use it to autonomously propel themselves, thus making them a fundamental model for non-equilibrium active matter systems. The motility mechanism can be mechanic, like cilia or flagella used by micro-organisms, or thermo- and chemodynamic, like phoresis of various nature Moran and Posner.

Incorporating self-propulsion within the Brownian framework enables the emergence of numerous novel behaviors, among which self-organization being noteworthy and the subject of this thesis. Across different systems, details about a single particle may differ, especially in propulsion mechanisms, nonetheless it is possible to build minimal statistical physics models that mimic real world dynamics, leading the way to the discovery of new physics as well as methods to analyze real living beings' behaviors and new ideas in material science.

1.1 Microrobotics and CELLOIDS

The first one to bring the concept of robotics at micro scale, especially for medical applications into a scientific context was Richard Feynman in the famous 1959 CalTech conference “There’s Plenty of Room at the Bottom” [4], from which the following quote is taken:

Many of the cells are very tiny, but they are very active [...] Consider the possibility that we too can make a thing very small which does what we want — that we can manufacture an object that maneuvers at that level!

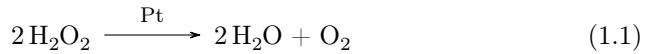
[...] it would be interesting in surgery if you could swallow the surgeon. You put the mechanical surgeon inside the blood vessel and it goes into the heart and "looks" around. [...] It finds out which valve is the faulty one and takes a little knife and slices it out. Other small machines might be permanently incorporated in the body to assist some inadequately functioning organ.

The scaling laws of physical quantities, especially regarding friction and hydrodynamics, makes it challenging to construct a micrometer scale robot which is just a miniaturized version of a macroscopic one. Moving in a fluid for the size and speed of few $\mu\text{m}/\text{s}$ we are interested in-meaning, at low Reynolds number ($\text{Re} \equiv \rho v L / \eta$) requires quite a different approach than what we are used to in the macroscopic world. When inertia is negligible, it is impossible to swim using a *reciprocal motion*, that is, a cyclic motion that follows the same path forward and back [5]. Needless to say that turning a propeller with a miniaturized macroscopic motor won’t be optimal at this scale, since its starting torque would be extremely large. These huge difficulties are encountered if one just wants to swim in a simple fluid. Using a micro robot inside a human body brings some other challenges in the problem, such as swimming in viscoelastic media like hyaluronic acid lattices or shrinking through the small interstices in between living cells. Most of the problems we enumerated have been solved by nature already e.g. immune cells can overcome pretty much all challenges mentioned in this paragraph. When trying to tackle said challenges a convenient approach would be imitating what has been already done in billions of years of evolution, instead of developing new strategies from scratch. [6] When talking about microrobots, we need to be clear about what a microrobot *is*. A *robot* is “a machine that can perform a complicated series of tasks by itself” [7]; a microrobot is defined as a robot with

- μm - mm size
- some mobility properties
- no tether

and its design needs some key ingredients, such as an actuation to make it motile, some kind of control and a power source to be able to perform desired tasks, as well as some sensing capability, to sample and observe surrounding environment. The aim of CELLOIDS, the project inside of which this work has taken place, is to build a micro-scale intelligent robotic system that takes inspiration from the amoeboid propulsion of immune cells. CELLOIDS’ concept for a microrobot is

using a phospholipids GUV (Giant Unilamellar Vesicle), filled with ABPs, as robot's body. GUV's walls are deformable and can be pushed from the inside by ABPs, leading to further deformation and, in the end, motility, mimicking the behavior of living cells that can shrink through a biological tissue. A single ABP moves exploiting a mechanism called *self-phoresis*: phoresis is defined in fluid dynamics as the migration of a particle due to a gradient in a scalar quantity (ϕ) e.g. temperature or concentration. When the gradient is caused by the particle itself *self-phoretic* movement takes place. When studying motion of a particle in a fluid it is common to take as boundary condition a no-slip condition, i.e. the layer of fluid in contact with particle's solid boundary has zero velocity w.r.t. the boundary itself, while in the study of phoresis, a small interfacial layer around particle's surface shows, on macroscopic length scales, an apparent finite slip velocity. Slip velocity is proportional to the local gradient of ϕ and the net migration pf the particle is equal in magnitude and opposite in sign to the area average of the slip velocity on particle surface. When ϕ is the concentration of some solute in the fluid, *diffusiophoresis* takes place; CELLOIDS tries to achieve this using as active particles Janus spheres, with an inert hemisphere and a catalytic one, which turns a fuel into some products generating a gradient in the concentration field in its vicinity; this makes the solvent flow around the particle causing it to move. In particular, the most studied Janus Sphere in this field is a SiO₂ sphere where Pt is deposited using a sputtering machine to coat just one hemisphere; the fuel is H₂O₂ that is decomposed in



1.2 Objectives

In the previously described framework, collective behaviors and emergent properties of active particles populations, both self-induced and caused by the interaction with a rigid or deformable confinement, are of paramount importance for the microrobot to work properly.

The objective of this work is to develop a model, a simulation framework and a suite of analysis tools to study how an explicit interaction potential changes single and collective behaviors of an active particles system, while keeping in mind the experimental and applied point of view.

1.2.1 Objective 1: Minimal model accounting for aligning and non-aligning interactions

Active matter models can be either *dry* or *wet*: in *dry* models the solvent is not explicitly simulated while *wet* simulations solve equations both for colloidal and fluid particles, including all possible hydrodynamics. However, details about the hydrodynamic and electro-chemical fields around active particles are still a matter of investigation. Moreover, even if known, they can be realistically simulated only for individual or few particles. In terms of interactions among particles, two main types are reported in the literature: aligning interaction, involving the direction of particles, and non-aligning, which can be central potentials like a hard sphere excluded volume interaction [8].

This work aims at a minimal *dry* model that takes into account both aligning and non-aligning interactions in the attempt of matching experimental observations. Here, the term *minimal* means not only that a unique interaction potentials is used to couple both positional and orientational degrees of freedom, but also that this simple model could be enough to simulate real world phenomena without making assumptions on the hydrodynamics and chemistry involved.

Given this novel modeling technique and its implementation, the next step of this work is to obtain a qualitative, eyesight agreement between simulations and experiments, making the *in silico* experiment able to capture all the features and dynamics observed *in vitro* scanning the different parameters.

Moreover, with the right parameters, this model is capable of showing rich collective behaviors, making it a feasible alternative to study phase transitions in a statistical physics fashion. A well-known model in this field involving aligning interactions is the Vicsek model [9], which captures a plethora of natural world phenomena. Present work shows how a system with coupled positions and orientations can turn into a continuous Vicsek-like case study.

1.2.2 Objective 2: Analysis

Although qualitative agreement between experiments and simulations plays a significant role, we believe that developing quantitative tools to analyze structure and dynamics of the system we are studying is necessary to understand the ongoing physics. Moreover, with slight adjustments, this tools could easily be adapted to the analysis of experimental data, namely, positions of particles obtained from videos tracking.

Here, we used well-known analysis tools, as well as some novel instruments developed during this thesis work.

1.2.3 Objective 3: Inference and Deep Learning

The last objective of the project is to build a Deep Learning based tool which, when trained starting from a minimal set of simulation data, is able to infer the interaction potential between couples of particles both in simulated and experimentally observed situations.

1.3 State of the Art

Active Matter is a term that refers to systems, both at macroscopic and microscopic scales, which can be described as sets of individual constituents, often called active or self-propelled particles, that have the ability of taking energy from the environment or an internal source to convert it to work. Such systems show peculiar behaviors, both individual and collective, due to their intrinsic far-from-equilibrium physical properties, as well as interactions that may occur between active particles [10], [11].

In the next few sections, we will focus on how researchers have dealt with active matter systems, both from an individual behaviors and collective characteristics standpoint.

1.3.1 Interactions and emerging behaviours

Investigation of interactions and emerging behaviours (e.g. collective motion) is an extremely important topic in the field of active particles, since it not only lead to a better understanding of the physics behind active particle systems, but it helped developing physical and mathematical tools, such as order parameters, which can be used both in simulation and experimental context.

In the following section the Vicsek model will often be cited. This is the right time to give a brief introduction to it with the authors' words:

The only rule of the model is: at each time step a given particle driven with a constant absolute velocity assumes the average direction of motion of the particles in its neighborhood of radius r with some random perturbation added.[9]

Although extremely simple, this model is of paramount importance in the study of active matter since it can be studied in terms of a phase transition and recreates to some extent the behavior of real living systems.

Although not strictly related to the present work, [12], [13] are essential papers for the study of collective motion. In these works, the problem of relating theoretical models and reality is tackled analyzing videos of starling swarms in order to understand characteristics of interactions. Understandably, the system studied by Cavagna, Cimarelli, Giardina, *et al.*, Ballerini, Cabibbo, Candelier, *et al.* is pretty different from the active Brownian particles ensemble investigated in this project, with the fist difference being the dimensionality (a bird swarm, just like a school of fish moves in a 3D environment), but some of the problems reported in that paper are still present in these days active matter community.

As for the case of self-propelled particles, [14] analyzes how an explicit polar aligning interaction can make the system transition to an ordered flocking phase, where almost all particles align their orientation in the same direction, even though an orientational noise is present. The model is built implementing an aligning torque between particles i and j which goes as $K \sin(\theta_i - \theta_j)$, only within a certain distance, and a repulsive potential $\epsilon \left(\frac{g}{r} \right)^{12}$ to take into account the excluded volume.

The parameters of the system are the Péclet number $\text{Pe} = \frac{v_0}{\sigma\gamma}$ which is the ratio between the self propulsion velocity and the product of the characteristic length of the particles and the rotational diffusion coefficient, quantifying the relative importance of advection to diffusion in solute transport, and $g = \frac{K}{4\pi\sigma^2\gamma}$ which quantifies the relative intensity of the orientational coupling strength and the reorienting noise. The order parameter is the mean global polarization

$$P = \frac{1}{N} \left| \sum_{k=1}^N \exp(i\theta_k(t)) \right| \quad (1.2)$$

which is ~ 0 in the disordered phase and > 0 in the flocking phase. The authors build a phase diagram in the $\text{Pe} - g$ plane Figure 1.1, which shows the separation between disorder and flocking as well as intermediate clustering phases, one with a microscopic cluster structure and one where a macroscopic single cluster structure arises.

After noting that the phase transition happens with increasing g for any $\text{Pe} > 1$, authors focus on studying this transition in g , in the spirit of an

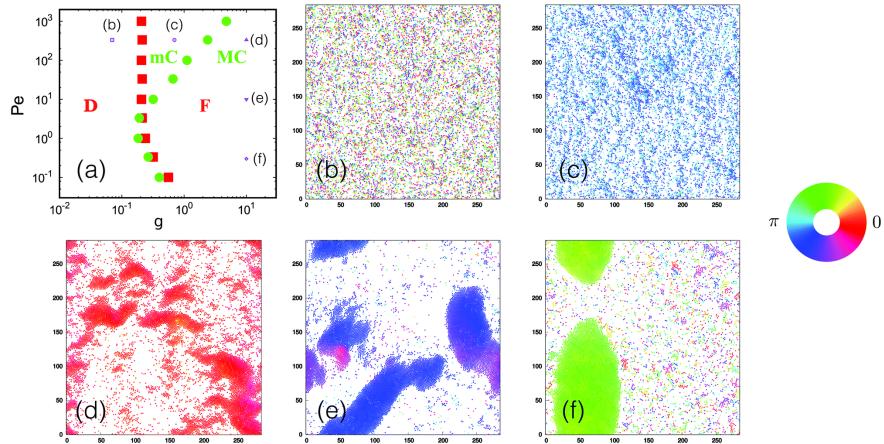


Figure 1.1: (a) Phase diagram in the g – Pe plane. (b-f) representing snapshots for different values of parameters. Adapted from [14]

equilibrium phase transition, with P as the order parameter and its susceptibility $\chi = N(\langle P^2 \rangle - \langle P \rangle^2)$. The behavior of the two quantities is similar to what is expected in a continuous phase transition with the critical coupling $g^c = 0.21 \pm 0.02$ (Figure 1.3(a)).

The authors then focus on clustering phenomena, noting that the cluster size distribution decays exponentially for $g < g^*$ and algebraically for $g > g^*$. This defines the two phases of microscopic and macroscopic clustering. Introducing two new order parameters

$$P_x = \left\langle \left| \frac{1}{N} \sum_{i=1}^N \cos(\theta_i) \right| \right\rangle; \quad P_y = \left\langle \left| \frac{1}{N} \sum_{i=1}^N \sin(\theta_i) \right| \right\rangle, \quad (1.3)$$

it is possible to distinguish between lane-like and band-like behavior in aligned clusters.

The authors use the radial distribution function

$$g(r) = \frac{1}{N} \left\langle \sum_{j \neq i} \sum_i \delta(r - |\mathbf{r}_i - \mathbf{r}_j|) \right\rangle \quad (1.4)$$

to characterize the global structure of the largest cluster. Increasing the coupling g at fixed Péclet number makes the peaks higher and shifts them to larger distances, showing that the system is developing a longer range order.

In [15], authors investigate the alignment of instantaneous velocities in cases where motility-induced phase separation (MIPS) occurs. Most literature focuses on the effect that a central 2-body potential, or an explicit aligning interaction which couples the orientational degrees of freedom of single particles, has on the system, but the interplay between phase separation of particles systems and alignment in their velocity has hardly been studied.

In the aforementioned paper, it is shown how, when active particle systems with a simple repulsive only Weeks-Chandler-Andersen potential phase-separate in a cluster, their velocity tend to form aligned domains, regardless the self propulsion orientation. For this reason, the global polarization is not a good

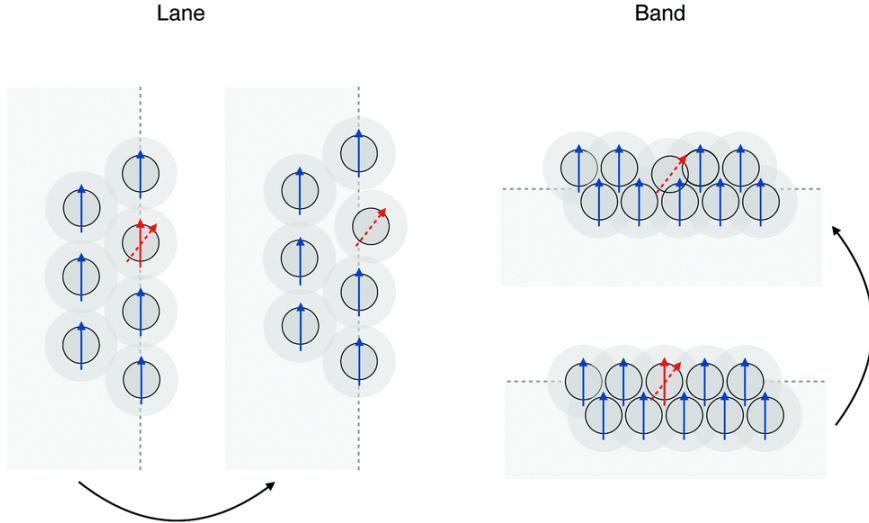
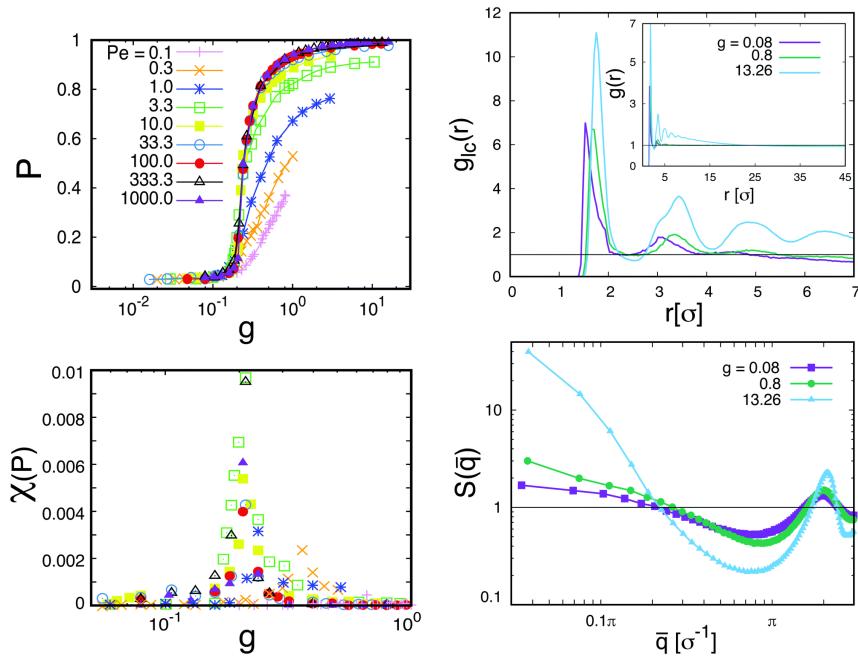


Figure 1.2: Different types of polarization. Adapted from [14].



(a) Phase transition behavior of polarization and its susceptibility

(b) Pair correlation function and static structure factor

Figure 1.3: Adapted from [14].

order parameter, even when the computation is restricted to clusters, thus the authors introduce the spatial correlation function of the velocity orientation $Q_i(r) = 1 - 2 \sum_j \frac{d_{ij}}{\mathcal{N}_k \pi}$, being $d_{ij} = \min[|\theta_i - \theta_j|, 2\pi - |\theta_i - \theta_j|]$ the angular distance between two particles and \mathcal{N}_k the number of particles in a circular shell around the i -th particle, taken with a thickness $\bar{r} = \text{argmax}\{g(r)\}$, and mean radius $k\bar{r}$ with integer k .

It is possible to derive an order parameter from $Q(R)$ integrating it

$$R = \int Q(r) dr \quad (1.5)$$

where the integral is performed on the cluster domain when present. This seems to be a good order parameter, since it makes it possible to distinguish the different phases of the system: varying the reorientation time $1/D_r$, R is discontinuous at the point where the MIPS occurs and the result is consistent with established MIPS order parameters as shown in Figure 1.4.

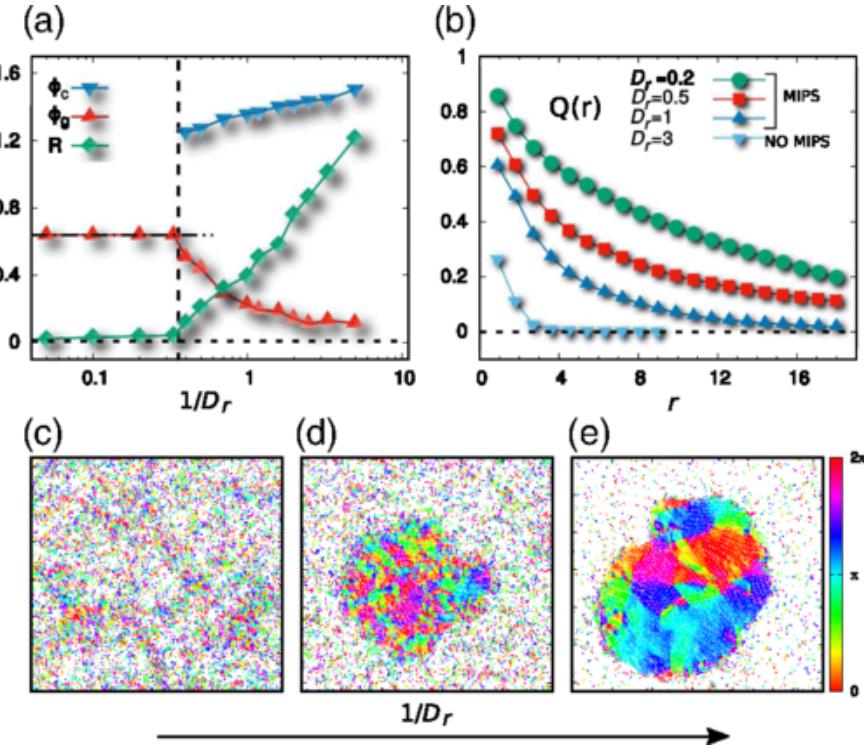


Figure 1.4: (a) Density as a function of $1/D_r$ for gaseous and cluster phase. (b) Velocity alignment order parameter R as a function of $1/D_r$. (c-e) Representative snapshots for different values of D_r . Adapted from [15].

Going on, the article shows analytically how it is possible to rewrite the equation of motion for the velocity: considering the symmetry in the hexagonal lattice which the particles arrange into, such equation involves a term which depends on the difference between particle's velocity and average velocity of the six surrounding it, thus re-obtaining a Vicsek-like model without a specific aligning interaction.

Some of the first papers cited in this section focus on animal behavior; it is believed that to better represent animal collective motion one have to implement some kind of vision in the model. [16] does so, with an aligning interaction similar to the one in [14], with the difference that the total torque on particle i is calculated taking into account only particles in the *vision cone* of particle i , i.e. a circular sector with center of mass of i as center as shown in Figure 1.5, with a given aperture angle and radius. This is useful to see what happens when *nonreciprocal* interactions, i.e. particle i feels the effect of particle j but the vice versa is not true.

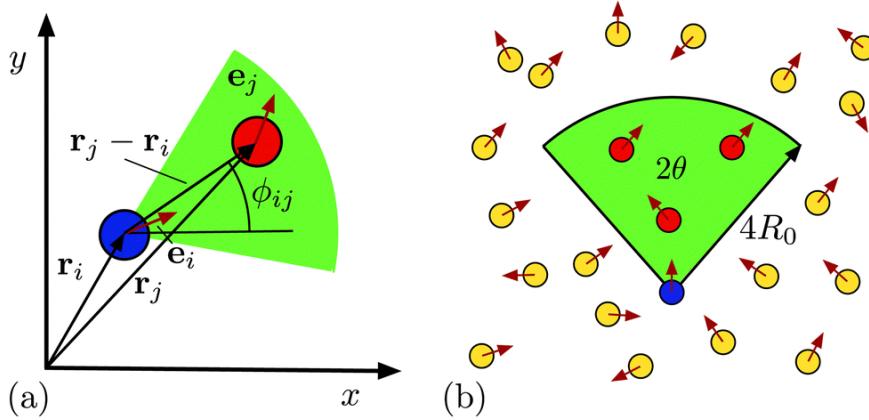


Figure 1.5: Schematic of nonreciprocal interaction with vision cone. Adapted from [16].

The authors use some tools, such as cluster size distribution, Mean Square Displacement (MSD) and velocity correlation function, to study the properties of a collection of *intelligent* active particles, but they also display some snapshot of the particles' simulation, which are particularly interesting to see what *can* happen when an interaction is acting on the system, both at small and large packing fraction.

1.3.2 Experimental analysis of collective behaviors

The interaction between active particles and their emergent collective behaviours have been studied also in experiments. [17] studies what happens when two SiO₂-Pt Janus particles come together in different configurations; authors model a collision analyzing the overlap between concentration fields around the two particles as well as the torque caused by the interaction between solvent flows around each hemisphere of a pair of Janus microswimmers. Although here we are trying to build a dry active matter simulation — means, absence of hydrodynamic and medium behavior is not simulated — papers like this accurately describe the phenomenology of these close encounters, which is essential to build minimal models that mimic real-world behavior. The main fact, supported both by simulations and experimental evidence, to take away is that chemical and hydrodynamic interactions do not cause reorientation for pairs of particles separated by a long distance, and that leading effects are short-range chemical torques.

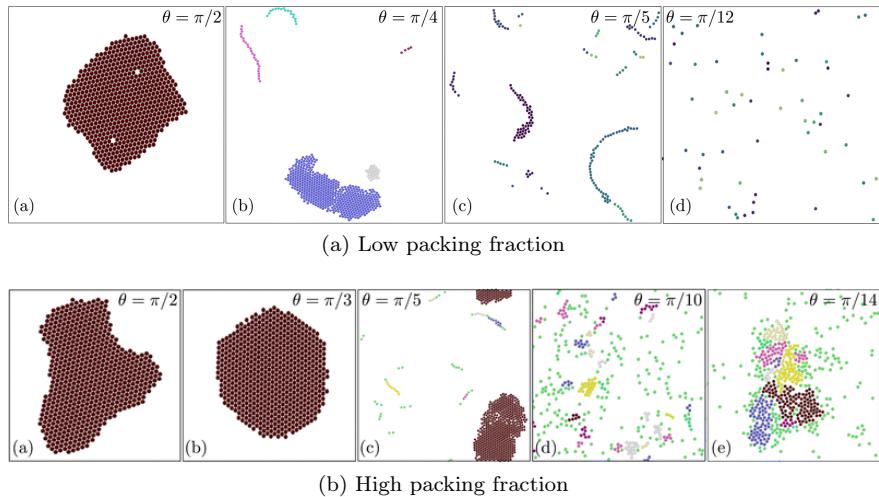


Figure 1.6: Representative snapshots for the configurations that model in [16] can achieve.

Dispersing Janus colloids in a H₂O₂ solution in a quasi-2D environment, the types of configurations particles can scatter in are limited and categorized by the authors in four dynamical states. The classification of both approach and departure states, along with a relative frequency histogram is in 1.7.

It is evident from this article that such particles exert a torque on each other at short range and this interaction, which is a product of the intrinsic asymmetry of a Janus colloid, must be considered in order to capture all the dynamics of these micro-swimmers.

[18] investigate what happens with a binary population of active colloidal particles, that is obtained by mixing particles of 7 μm and 10 μm in diameter. These colloids move due to a different mechanism than the one seen before, whose details will not be discussed here, called Quincke instability; with enough particle density, the solvent mediates an aligning interaction which makes the system undergo a Vicsek-like flocking transition that, within a circular confinement, takes place as a vortex motion. The two populations are distinguished by a difference in diameter and in velocity. Within some minutes, the initially uniform sample demixes spontaneously, heading to a segregated state where the two populations are well separated and their relative densities have a different radial profile, as shown in Figure 1.8.

Ostapenko, Schwarzenbach, Böddeker, *et al.*, in [19] look into the dynamics of biological micro-swimmers in a confinement. Authors tracked the movements of a single *Chlamydomonas reinhardtii* algae cell within a round and elliptical space. At the same time, they performed a simulation of the algae behavior as an ABP; to better model the dynamics, the alga's placeholder was a dumbbell shaped particle made of two attached spheres of different radii. In the simulation, the steric interaction with the wall was modeled as a Weeks-Chandler-Andersen potential and a torque reorienting the particle near the wall was inserted too.

For different circular confinement radii, the radial probability density $P(r)$ was extracted, resulting in a striking agreement between simulations and experiments, as shown in Figure 1.9.

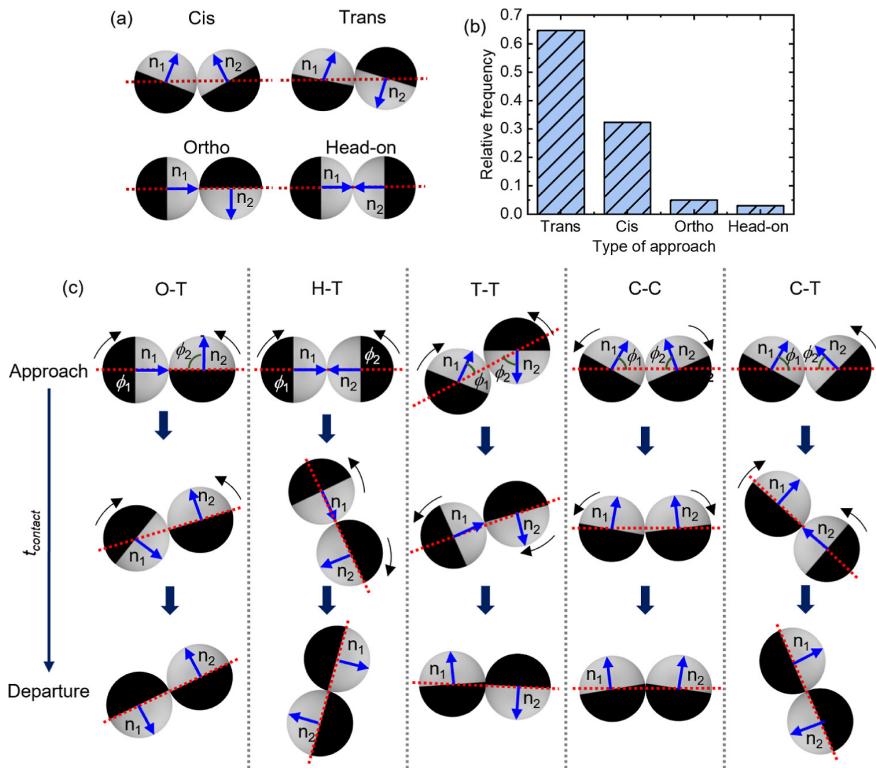


Figure 1.7: Classification of approach and departure configuration. Adapted from [17].

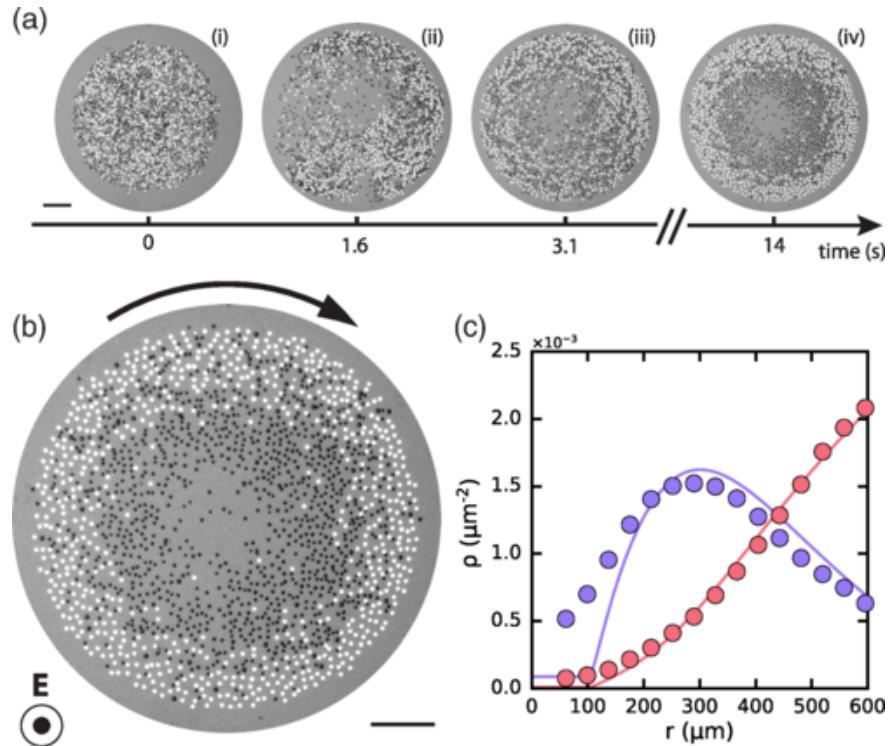


Figure 1.8: (a) The demixing process. (b) Steady state of binary colloidal flocks. (c) Radial density profiles of the two species. Adapted from [18]

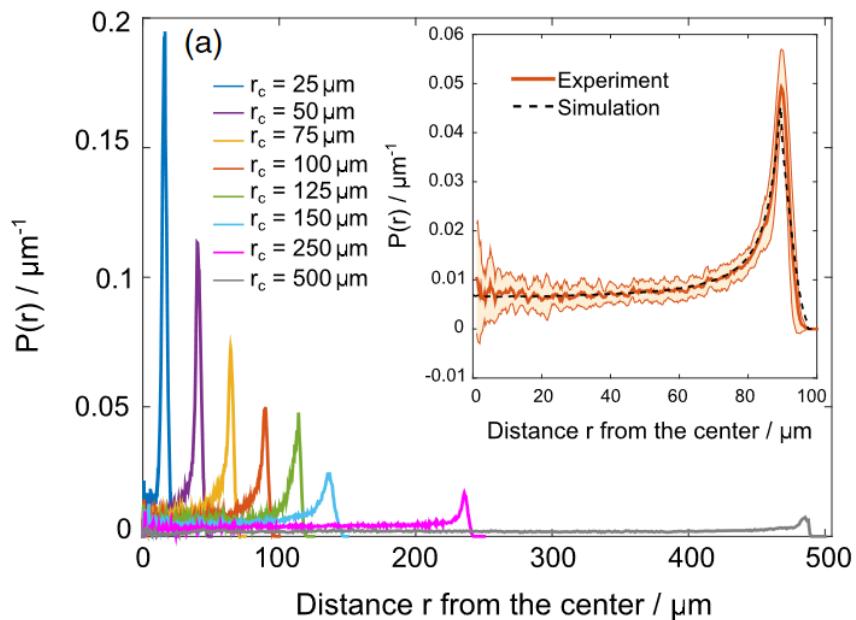


Figure 1.9: Experimental radial probability densities. Adapted from [19].

Next, Ostapenko, Schwarzenbach, Böddeker, *et al.* focus on how the swimming statistics depend on the radius of curvature. In order to do so, keeping the results independent from confinement size, simulations and experiments were performed within an elliptical chamber, with the result that the alga spent more time near the walls with the smaller curvature radius. The result is that near-wall swimming probability increases monotonically with the curvature (or decreases monotonically with the radius), and once again, there is a good agreement between experimental data and simulations.

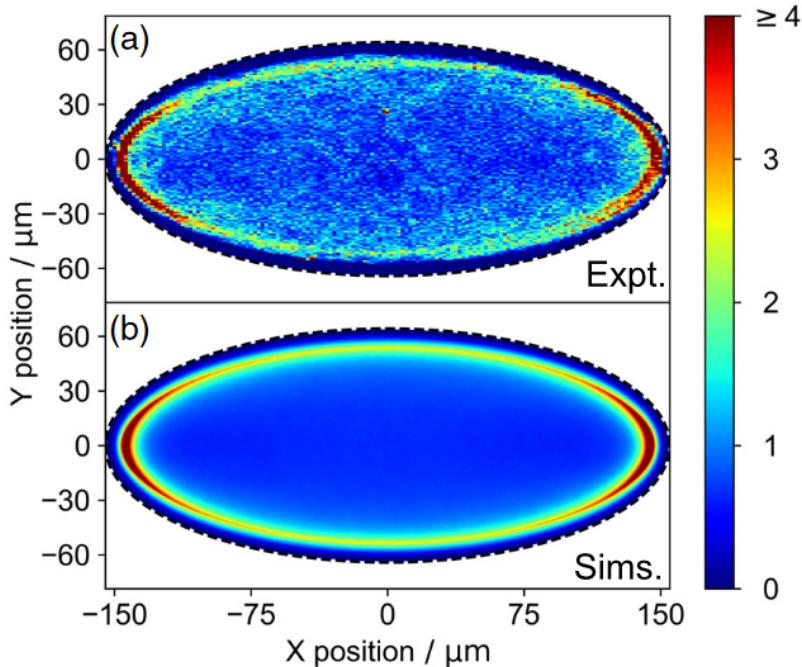


Figure 1.10: Comparison between relative probability density in (a) experiments and (b) simulations. Adapted from [19].

1.3.3 Inference of interaction potentials

Inferring interaction potentials starting from experimental or simulated data is challenging and different techniques could be adopted. Since the real interactions that occur between a pair of colloidal particles are mostly unknown and in principle may involve chemical gradients, hydrodynamics, electromagnetism and all kinds of interplay between these and other mechanisms, it would be extremely useful to have a tool that predicts forces between particles mapping them in terms of some kind of minimal model such as a simple central potential. Here, we will focus only on inference strategies based on Deep Learning (DL), which have shown particularly promising.

In terms of machine learning model, the simplest approach is using some global attribute of the active particle ensemble to predict the potential via a Deep Neural Network. The idea is to concatenate all attributes in a vector and feed it as input to a DNN, that has as target function another vector containing

the values of interaction potentials; the loss function to minimize is the difference (absolute or square) between the predicted potential and the ground-truth one. [20] aims at doing exactly that, exploiting a fact from statistical and matter physics; as authors claim, they use a theorem saying that “for the fluids with only pairwise interaction (quantum or classical), the pair potential $V(r)$ that leads to a specific $g(r)$ is unique” and the question of predicting which $V(r)$ causes a specific structural correlation like $g(r)$ or, equivalently, $S(q)$ “is a well defined one”.

Authors simulated both passive and active Brownian particles to see the difference between equilibrium and non-equilibrium configurations. All the tested potential were of the form $V(r_{ij}) = 4\varepsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^a - \lambda \left(\frac{\sigma}{r_{ij}} \right)^b \right]$ with different values of the parameters. Letting the system reach steady state and then taking 100 snapshots of the pair correlation function to train the network, the authors show pretty good results regarding the accordance between predicted and real potentials in all the possible phases: gas-like, crystal and liquid-like. (1.11)

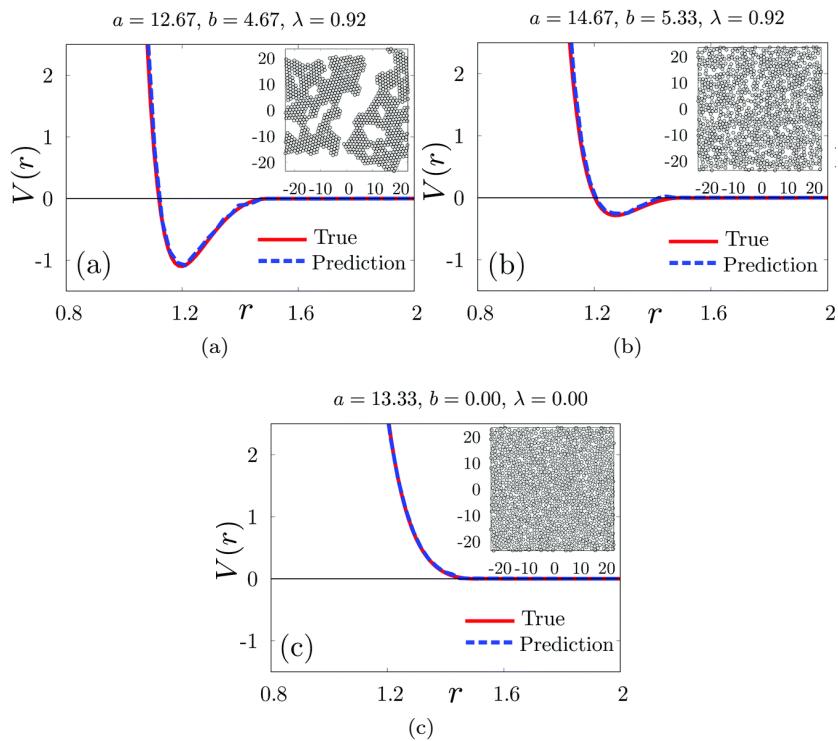


Figure 1.11: Potential shapes for different values of parameters [20].

The non-equilibrium case is a different matter: as soon as self-propulsion is involved, phenomena like MIPS start to occur, pushing the system to cluster. Some literature has introduced effective many-body attractive potential to explain the change in structure caused by particles’ motility, even in cases where just an explicit repulsion is present, and the results presented by Bag and Mandal seem to point in that direction as well. As Figure 1.12(b) shows, pair correlation

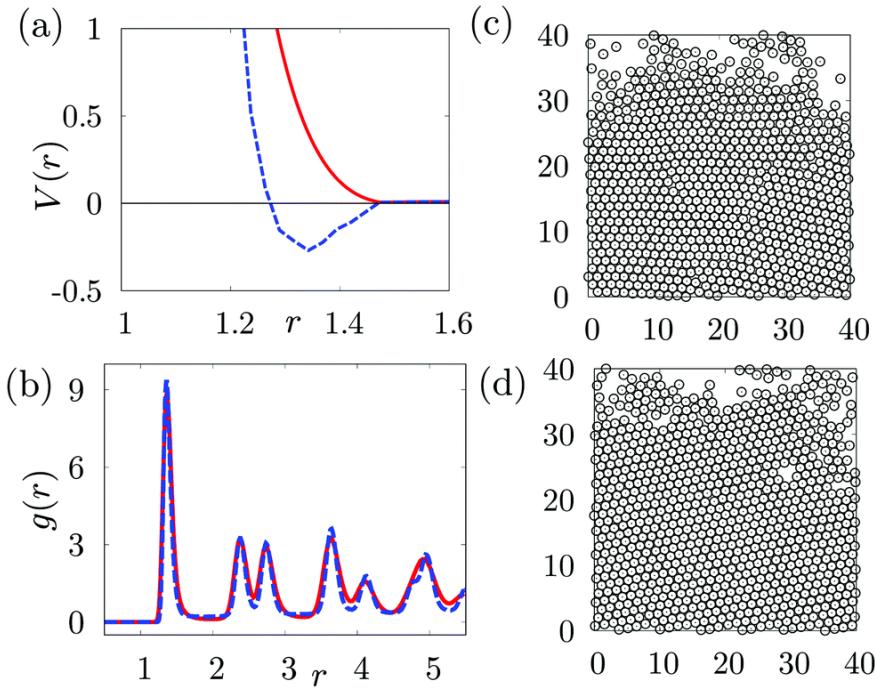


Figure 1.12: Equivalence between an equilibrium system with attraction and non-equilibrium one without attraction. Adapted from [20].

functions in the case of active repulsive particles is similar to that of passive attractive particles, since only structure — i.e. particles’ positions — is involved, tricking the network into predicting an attractive potential. This is a good result in a simulation context since it shows the capability of some methods to predict potentials starting from particle’s positions and, though in out of equilibrium cases it does not work as desired, it sheds light on some theoretical statements. Nonetheless this approach does not fit the objective of the present thesis project, which tries to obtain actual 2-body interactions in active particle systems.

An alternative approach is working directly on positions and velocities of individual particles. It is an accepted fact that the best way to make AI learn something about a problem is having the invariances and symmetries of said problem taken in to account in the model design. The most efficient way to store information about a set of interacting bodies is a graph, where vertices represent single body information like position, velocity, mass or charge and edges contain information about the potential that make pairs of particles communicate. With these facts in mind, it is clear why Graph Neural Networks (GNNs or GNs) are so popular in several fields of physics. The distinction between GNNs and GNs is that the first one is a term that represent every kind on Neural Networks related with graphs, regardless the structure, while the Graph Network framework represents a specific network with a graph structure. An actual GN is actually a container, structured as a graph, which in principle holds three networks or functions, or models: one is the edge function ϕ^e , which predicts edge features, a node function ϕ^v which predicts single nodes features and a global function ϕ^u to predict global features. ϕ^e maps from a pair of nodes and some information

stored on the edge to a message vector which lies in the *latent space*. Then, all the messages from sending nodes into a receiving one are aggregated by means of a permutation-invariant function, namely sum, mean, max etc. The node model ϕ^v takes the aggregated messages along with the node information and predicts some feature. Finally, a global function gathers all the information and predicts a global attribute [21].

In the case of interacting bodies, edge function works as the force law and node function plays the role of the second Newton's law, calculating the resulting force and applying it to obtain particle's acceleration. In [22] such an approach is applied to the case of Newtonian dynamics (along with Hamiltonian dynamics and a Dark Matter simulation, which we will not focus on) of a set of interacting particles. After simulating the dynamics of a set of interacting bodies with several kinds of interaction forces, Cranmer, Sanchez Gonzalez, Battaglia, *et al.* experiment with different message dimensionalities. Message dimensions is a topic that causes doubts when working with GNs and multiple strategies can be explored, the most natural of them is creating a bottleneck using the dimension of the problem, e.g. if particles move in N dimensions than forces are N-D and one may be tempted to think that, since messages should represent forces, this is the best dimensionality to use; the other approach is using a high-dimensional message, e.g. 100D, in training and then selecting the N most meaningful dimensions to plot the force in N-D. Cranmer, Sanchez Gonzalez, Battaglia, *et al.* try both of these strategies as well as a hybrid one: instead of implementing a hard bottleneck, they let the network learn with a 100D message but with regularization terms that encourage the model to learn compact representations of the forces, in accordance with an Occam's razor type of reasoning. Doing this, the message is still 100-dimensional, but now most of its components have no variability, leaving few of them informational. Their results show that, although an explicit bottleneck works well, the best performing strategy is the L_1 regularization. The whole workflow is explained in Figure 1.13.

Since all the operations that happen inside the node function are linear, it is evident that the message components will be some kind of linear combination of the actual forces. After letting the network train, one can search for the linear operation that minimizes the difference between the learned message components and the forces.

An alternative way of extracting the forces is using symbolic regression, which will give some insights about the functional form of the interaction potential. The modeling engine used is *eureqa*, and the best model is selected between several candidates at different complexity levels, choosing a more complicated one only when it is worth it.

The application to real active particles can make use some of these approaches, but it requires some tweaks with respect to the Newtonian dynamics approach. The most important change is in the physics: in a system without inertia, the meaning of acceleration is not clear and there is a linear relation between a force and a velocity. In this framework, Ruiz-Garcia, Barriuso G., Alexander, *et al.* have developed a variant of the graph network by Cranmer, Sanchez Gonzalez, Battaglia, *et al.* that can work with active particles dynamics, called ActiveNet. In [23], the graph network takes as input positions and orientations of a set of simulated active particles and tries to predict their velocity, given the instantaneous velocity $\frac{\vec{x}(t+\Delta t)-\vec{x}(t)}{\Delta t}$ as the ground truth. In this case the role of

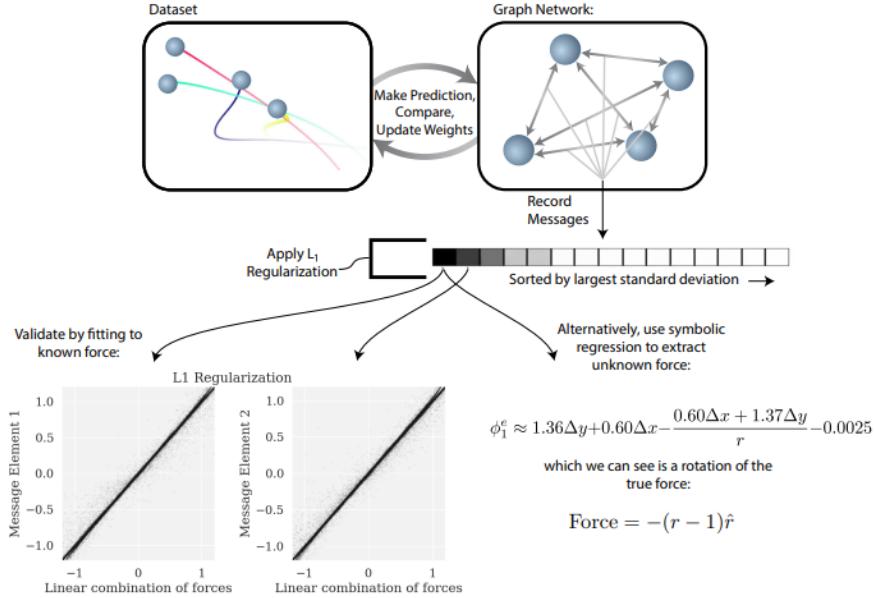


Figure 1.13: The GNN learning workflow in the case of Newtonian dynamics. Adapted from [22].

the node function is not only to predict the velocity starting from the sum of the forces, but also of learning and adding to it the self-propulsion velocity that drives the particles. This is a useful result, since after predicting the interaction force one can take the output of the node function and subtracting the edge function output in order to understand which part of the velocity can be explained through interaction and which is due to self-propulsion. A schematic representation of the ActiveNet GN is in Figure 1.14. In order to train the network, successive iterations are done, increasing the threshold distance to consider two particles as interacting. At the end of the training, the network gets tested and results show that it works well to predict both the self-propulsive force and the interaction potential. Having an internal way of taking into account the active velocity, this method works better than static structure-based ones, like [20], in out of equilibrium particles ensembles where MIPS takes place, learning a repulsive interaction even though the system is clustered. Clearly, ActiveNet fails at short distance, since very few data are presented. Having Brownian motion at his base, the studied system has an internal degree of randomness. The goal of a machine learning task should be to reduce the loss (difference between ground truth and predictions) to make it as small as the noise. This can be useful to estimate the diffusion coefficient and works well even with few examples, as long as the temperature is high enough.

An important result of this paper is the application of the method to experimental data. The network needs some adjustments in order to work since experiments have some hurdles that must be overcome. To not be influenced by stuck particles, the network is fed with the bias that forces depend only on distances and not positions. Moreover, self-propulsion is forced to depend only on the orientation of particles. The experimental observations studied in that

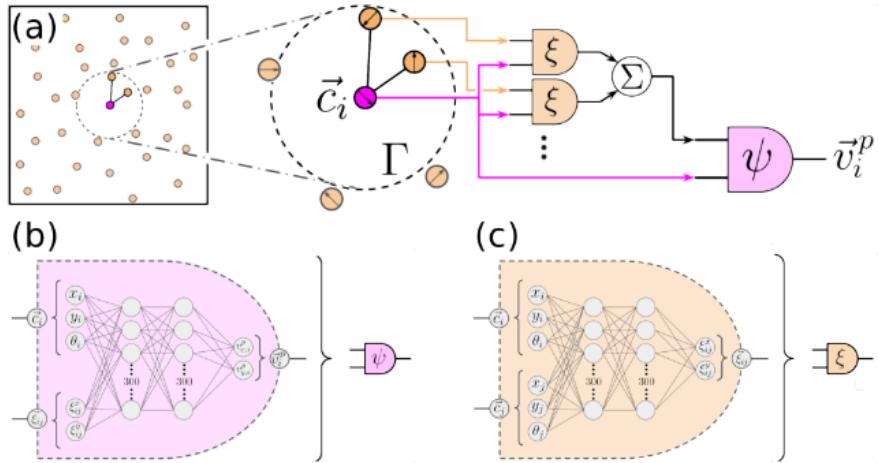


Figure 1.14: The ActiveNet structure. Adapted from [23].

work contain thousands of particles with a clearly detectable in-plane orientation. Results show some similarities with the expectations but not much can be said since the actual interactions are not known exactly. Authors claim that probably symbolic regression is needed to discover the actual form of the interaction.

1.4 Background

1.4.1 Passive Brownian Motion

The first formulation of the theory of Brownian motion was obtained by Albert Einstein in a 1905 paper, where the mean square displacement was given in terms of a linear relation with time, with a diffusion coefficient D_t as $\sqrt{\bar{x}^2} = \sqrt{2D_t t}$. Here, we outline the derivation as it was presented by Paul Langevin some years afterwards, following [1].

It is known from statistical mechanics that a Brownian particle in equilibrium will have as mean kinetic energy

$$\langle \frac{1}{2}mv^2 \rangle = \frac{1}{2}k_B T \quad (1.6)$$

where T is the absolute temperature and k_B is Boltzmann's constant. If we assume the same formula as in macroscopic hydrodynamics, the viscous drag acting on the particle will be of the form $-6\pi\eta a \frac{dx}{dt}$, being a the particle's radius and η the viscosity of the fluid. Due to the random collisions with the fluid molecules, a passive Brownian particle experiences a *fluctuating force* X . The equation of motion for the particle is thus

$$m \frac{d^2x}{dt^2} = -6\pi\eta a \frac{dx}{dt} + X \quad (1.7)$$

which we multiply by x getting

$$\frac{m}{2} \frac{d^2}{dt^2} (x^2) - mv^2 = -3\pi\eta a \frac{d(x^2)}{dt} + Xx \quad (1.8)$$

where we called $v = \frac{dx}{dt}$. Averaging on the ensemble of particles we have

$$\frac{m}{2} \frac{d^2\langle x^2 \rangle}{dt^2} + 3\pi\eta a \frac{d\langle x^2 \rangle}{dt} = k_B T \quad (1.9)$$

where the value for the mean kinetic energy was plugged in and we averaged the product Xx to zero, being X highly irregular (in modern times we just write everything in terms of a 0 mean Gaussian stochastic process, namely a Wiener process). The general solution to the differential equation for $\langle x^2 \rangle$ is

$$\frac{d\langle x^2 \rangle}{dt} = \frac{k_B T}{3\pi\eta a} + C \exp(-6\pi\eta at/m) \quad (1.10)$$

with C an arbitrary constant. It is possible to neglect the exponential, since Langevin estimated its characteristic time about 10^{-8} s, and then integrate to get the equation

$$\langle x^2 \rangle - \langle x_0^2 \rangle = \frac{k_B T}{3\pi\eta a} t \quad (1.11)$$

which corresponds to what Einstein found out if $D_t \equiv k_B T/(6\pi\eta a)$.

Adapting this framework to our case-study, where the particle's orientation is considered, we need to take in to account both random forces and torques so that its motion is purely diffusive, both in position and orientation with the following diffusion coefficients

$$D_t = \frac{k_B T}{\gamma_t} \quad D_r = \frac{k_B T}{\gamma_r} \quad (1.12)$$

being $\gamma_t = 6\pi\eta a$ and $\gamma_r = 8\pi\eta a^3$ the respective drag coefficients, where a is the particle radius and η is the fluid viscosity. The basis to build the model upon is the Langevin equation

$$m\ddot{\mathbf{r}} = -\gamma_t \dot{\mathbf{r}} + \mathbf{F}_{th} \quad (1.13)$$

where \mathbf{F}_{th} is the random force given by the collisions with the fluid molecules.

Given that a typical Brownian particle will have a characteristic body-length in the order of μm and a velocity of $\mu\text{m s}^{-1}$ the system can be studied in low-Reynolds number regime being

$$Re = \frac{\rho v a}{\eta} \sim 10^{-6} \quad (1.14)$$

where ρ is the fluid density, v is the particle speed, a is the particle radius and η is the fluid viscosity and the values of density and viscosity for water were plugged in. As a consequence of this fact, inertial effects can be neglected and it is possible to study the system in the *overdamped* regime, turning the Langevin equation 1.13 into

$$\gamma_t \dot{\mathbf{r}} = \mathbf{F}_{th} \quad (1.15)$$

which can be rewritten as:

$$\dot{\mathbf{r}} = \sqrt{2D_t} d\mathbf{W} \quad (1.16)$$

where $d\mathbf{W}$ is the derivative of a zero-mean, unitary-variance Wiener process. In a homogeneous environment, rotational and translational motions are independent from each other, so that the equations of motion for a passive Brownian particle are

$$\dot{x} = \sqrt{2D_t} dW_x, \quad \dot{y} = \sqrt{2D_t} dW_y, \quad \dot{\theta} = \sqrt{2D_r} dW_\theta \quad (1.17)$$

1.4.2 Numerical Simulations of active particles

Despite the variety in single agent properties and self-propulsion mechanisms, it is possible to identify some key features that are shared between all active particle systems. The most important characteristic of an active, self-propelled particle is that, notwithstanding a symmetric shape (often in literature, and always in this work, spherical), each particle has a preferred axis which lies along the direction of self-propulsion.

With this hypothesis, rotational diffusion has now become relevant since the direction of self-propulsion varies randomly with a characteristic time scale which corresponds to the inverse of the rotational diffusion coefficient $\tau_r = D_r^{-1}$.

A simple and effective model to describe the dynamics of active particles is the Active Brownian Particle (ABP) model, which is a generalization of the Brownian particle model. This consists in adding a constant-magnitude self-propulsion velocity \mathbf{v} term to the equations of Brownian motion:

$$\dot{x} = v \cos \theta + \sqrt{2D_t} dW_x, \quad \dot{y} = v \sin \theta + \sqrt{2D_t} dW_y, \quad \dot{\theta} = \sqrt{2D_r} dW_\theta \quad (1.18)$$

where θ is the particle orientation and v is the magnitude of the self-propulsion velocity.

As a consequence, the resulting finite-differences equations are:

$$\begin{cases} x_{n+1} = x_n + v \cos(\theta) \Delta t + \sqrt{2D_t \Delta t} W_{x,n}, \\ y_{n+1} = y_n + v \sin(\theta) \Delta t + \sqrt{2D_t \Delta t} W_{y,n}, \\ \theta_{n+1} = \theta_n + \omega \Delta t + \sqrt{2D_r \Delta t} W_{\theta,n} \end{cases} \quad (1.19)$$

where the case of a deterministic self-propulsion angular velocity ω is also taken into account.

It is possible to insert external forces and torques in the system, where *external* means they are not due to the self-propulsion. Even though in literature there are examples of uniform external potentials, e.g. electric and magnetic fields, being applied on the ABP ensemble as a whole, in this work the only external force is the interaction between particles, which is applied in the low-Reynolds number regime as:

$$\begin{cases} x_{n+1} = x_n + \left(v \cos(\theta) + \frac{D_t}{k_B T} F_{ext,x} \right) \Delta t + \sqrt{2D_t \Delta t} W_{x,n}, \\ y_{n+1} = y_n + \left(v \sin(\theta) + \frac{D_t}{k_B T} F_{ext,y} \right) \Delta t + \sqrt{2D_t \Delta t} W_{y,n}, \\ \theta_{n+1} = \theta_n + \left(\omega + \frac{D_r}{k_B T} T_{ext} \right) \Delta t + \sqrt{2D_r \Delta t} W_{\theta,n} \end{cases} \quad (1.20)$$

where applying a force or a torque just linearly translates to a linear or angular velocity change, with the respective drag coefficients as proportionality constant.

1.4.3 Stochastic integration

The topic of integrating a stochastic differential equation (SDE) is a fundamental part of present work and an important object of investigation in the physics and mathematics community. The most simple SDE one can write is

$$\frac{dx}{dt} = f(x) + g(x)\xi(t) \quad (1.21)$$

in which $\xi(t)$ is a stochastic process we assumed to be Gaussian with zero mean and no correlation at different times, which reads $\langle \xi(t)\xi(t') \rangle = \delta(t-t')$. If we define $dW = \xi(t) dt$, equation 1.21 is equivalent to

$$dx = f(x) dt + g(x) dW \quad (1.22)$$

where dW is the increment of a Wiener process $W(t)$, defined by its probability

$$P(W(t)) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{W(t)^2}{2t}}. \quad (1.23)$$

Now, to solve such SDE, we need to compute integrals of the form $\int W dW$. We start by defining the mean square limit as

$$\text{m.s. } \lim_{n \rightarrow \infty} X_n = X \iff \lim_{n \rightarrow \infty} \langle (X_n - X)^2 \rangle = 0 \quad (1.24)$$

where the average $\langle \cdot \rangle$ is taken over different realizations of the stochastic process. With this, the stochastic integral is defined in terms of a discrete sum

$$\int_0^t W dW = \text{m.s. } \lim_{n \rightarrow \infty} \sum_{i=1}^n W(t_i^*) [W(t_i) - W(t_{i-1})] \quad (1.25)$$

with $t_{i-1} < t_i^* < t_i$, that is better defined in terms of a constant α as $t_{i-1} + \alpha(t_i - t_{i-1})$. Using the increment independence property of the Wiener process, we can compute the sum getting

$$\sum t_i - 1 + \alpha(t_i - t_{i-1}) = \alpha t \quad (1.26)$$

so that the result of the integral depends on α . In principle, no value in $[0, 1]$ is denied, but in literature only two values are found: $\alpha = 0$, which defines Itô method, and $\alpha = 1/2$, specific for Stratonovich method, which gives the *standard* result that can be obtained through Riemann integration.

With all of this, we are ready to integrate our SDE, formally

$$x(t) - x(0) = \int_0^t f(x(s)) ds + \int_0^t g(x(s)) dW \quad (1.27)$$

considering only the second term in the right-hand side, and Taylor-expanding it we get

$$\int_0^t g(x) dW \approx \int_0^t dW [g(x(0)) + g'(x(0))(x(s) - x(0))]. \quad (1.28)$$

The Itô versus Stratonovich controversy poses a subtle but fundamental problem: since g depends on x which depends on t , to compute the i -th step when we have the $i-1$ -th we need to evaluate g at an instant t_i^* , subsequent to t_i , where the value of x is not known yet, in principle. Only the Itô prescription solves this issue, using the concept of *nonanticipating* functions.

With all of this said, it is possible to state that the type of calculus used, i.e. the value of α , is actually a parameter of the model and there is literature showing that the physical quantities obtained through simulations are different if one chooses a method over another [24].

Let us apply this theoretical notions to the equations for ABPs. One must be very careful when dealing with discretization step in stochastic integration, especially when dealing with nonlinear functions of noise terms, for it is difficult to know *a priori* which are the correct expansions at any order. Only accounting for θ and x , we have

$$\begin{cases} dx = f_x(x) dt + v_0 \cos(\theta) dt + \sqrt{2D_t} dW_x \\ d\theta = \sqrt{2D_r} dW_\theta \end{cases} \quad (1.29)$$

where in f we included all possible interactions a particle can undergo. For now we set that part to 0 since it does not raise integration problems and consider only the terms which contain stochastic parts. We integrate to get

$$\begin{cases} x(h) - x(0) = \int_0^h v_0 \cos(\theta(s)) ds + \int_0^h \sqrt{2D_t} dW \\ \theta(s) - \theta(0) = \sqrt{2D_r s} \sim \mathcal{O}(\sqrt{s}) \end{cases} \quad (1.30)$$

being $\int_0^s dW = \sqrt{s} Y_1$ with $Y_1 \sim \mathcal{N}(0, 1)$. We plug what obtained for θ in the first order Taylor expansion of cosine to get ($\theta_x \equiv \theta(x)$ for short)

$$x(h) - x(0) = \int_0^h v_0 [\cos(\theta_0) - \sin(\theta_0)(\theta_s - \theta_0)] ds + \int_0^h \sqrt{2D_t} dW \quad (1.31)$$

$$= \sqrt{2D_t h} Z_1 + h v_0 \cos \theta_0 - \sin \theta_0 Y_1 \sqrt{h} \int_0^h ds \quad (1.32)$$

with $Z_1 \sim \mathcal{N}(0, 1)$, and we notice that the last term is $\mathcal{O}(h^{3/2})$, leaving us with the first order unchanged. Having the right first order is fundamental because that is the order that enters in finite difference equations. We can thus conclude that finite difference equations in the form we wrote them in expressions 1.19-1.20 are the correct ones for our problem.

1.4.4 Algorithms

A variety of algorithms to integrate SDEs have been reported in the literature. Here we only give the details of what has been used in present thesis.

The simplest method to integrate differential equation is the Euler scheme, which was first applied to ODEs in XVIII century. It is a first order method and it is the basis to all higher order schemes. As we did before, we take the first order expansion in the integration step h to integrate equation 1.22

$$x(h) - x(0) = \int_0^h (f_0 + g_0 \xi(t)) dt = h f_0 + g_0 \int_0^h \xi(t) dt \quad (1.33)$$

which in general is not correct: in stochastic integration, being $\int_0^h dW \sim \mathcal{O}(\sqrt{h})$, some unexpected $\mathcal{O}(h)$ terms, which are combination of lower orders, tend to appear. In particular, it is straightforward to show that the correct first order in h is

$$x(h) - x(0) = g_0 Z_1(h) + f_0 h + \frac{1}{2} g'_0 g_0 Z_1(h)^2 \quad (1.34)$$

being $Z_1 \sim \mathcal{N}(0, \sqrt{h})$ [25].

Anyway, for cases like ours where noise is purely additive (i.e. $g(x) = \sqrt{2D}$), the last term is 0 and the simple-minded first order is correct. Basically, this algorithm means that at any step one should propagate with the first order of the deterministic part and then add a randomly generated Gaussian number with 0 mean and the right variance. A pseudo code for Euler algorithm applied to the case of interacting ABPs is in Algorithm 1.

Algorithm 1 The Euler algorithm

```

1: for n in timesteps do
2:   for i in particles in ensemble do
3:      $\vec{F}_{i,n} = \sum_j^{N_p} \vec{F}_{ij,n}$ 
4:      $w_{i,n} \sim \mathcal{N}(0, \sqrt{2D_t \Delta t})$ 
5:      $z_{i,n} \sim \mathcal{N}(0, \sqrt{2D_r \Delta t})$ 
6:      $\vec{r}_{i,n+1} \leftarrow \vec{r}_n + w_{i,n} + v(\cos \theta_{i,n}, \sin \theta_{i,n}) \Delta t + \vec{F}_{i,n} \Delta t / \gamma_t$ 
7:      $\theta_{i,n+1} \leftarrow \theta_{i,n} + z_{i,n} + \omega_{i,n} \Delta t + T_{i,n} \Delta t / \gamma_r$ 

```

The first higher order correction to Euler is the Heun scheme. This algorithm involves the calculation of an intermediate step x_{int} which helps in dealing with the nonlinearity of the deterministic function f . It works as follows

$$\begin{aligned} x_{\text{int}} &= x(0) + \sqrt{2D} Z_1(h) + f_0 h \\ x(h) &= x(0) + \sqrt{2D} Z_1(h) + \frac{h}{2} (f_0 + f(x_{\text{int}})). \end{aligned} \quad (1.35)$$

Algorithm 2 The Heun algorithm

```

1: for n in timesteps do
2:   for i in particles in ensemble do
3:      $w_{i,n} \sim \mathcal{N}(0, \sqrt{2D_t \Delta t})$ 
4:      $z_{i,n} \sim \mathcal{N}(0, \sqrt{2D_r \Delta t})$ 
5:      $\vec{F}_{i,n} = \sum_j^{N_p} \vec{F}_{ij}(\vec{r}_{i,j}(\vec{r}_{i,n}, \vec{r}_{j,n}))$ 
6:      $\vec{r}_{i,\hat{n}} \leftarrow \vec{r}_n + w_{i,n} + v(\cos \theta_{i,n}, \sin \theta_{i,n}) \Delta t + \vec{F}_{i,n} \Delta t / \gamma_t$ 
7:      $\theta_{i,\hat{n}} \leftarrow \theta_{i,n} + z_{i,n} + \omega_{i,n} \Delta t + T_{i,n} \Delta t / \gamma_r$ 
8:      $\vec{F}_{i,\hat{n}} = \sum_j^{N_p} \vec{F}_{ij}(\vec{r}_{i,j}(\vec{r}_{i,\hat{n}}, \vec{r}_{j,\hat{n}}))$ 
9:      $\delta \vec{r} \leftarrow \frac{\Delta t}{2} \left[ v(\cos \theta_{i,n}, \sin \theta_{i,n}) + \vec{F}_{i,n} / \gamma_t + v(\cos \theta_{i,\hat{n}}, \sin \theta_{i,\hat{n}}) + \vec{F}_{i,\hat{n}} / \gamma_t \right]$ 
10:     $\vec{r}_{i,n+1} \leftarrow \vec{r}_n + w_{i,n} + \delta \vec{r}$ 
11:     $\theta_{i,n+1} \leftarrow \theta_{i,n} + z_{i,n} + \frac{\Delta t}{2} [\omega_{i,n} \Delta t + T_{i,n} / \gamma_r + \omega_{i,\hat{n}} \Delta t + T_{i,\hat{n}} / \gamma_r]$ 

```

Now one can ask which is the "best" algorithm. The answer lies in two distinct topics: deterministic accuracy and stochastic behavior. Regarding deterministic accuracy, one can analyze which is the order of the numerical error made in the integration, obtaining that, being a first order algorithm, Euler scheme is accurate up to $\mathcal{O}(h)$, while Heun is $\mathcal{O}(h^2)$, making the latter preferable especially when working with highly nonlinear and steep potentials.

In order to analyze stochastic behavior, if one is interested in large time behavior, it is possible to study the equilibrium distribution $P(x)$. In particular, a system described by $\dot{x} = -V'(x) + \sqrt{2D}\xi(t)$, the equilibrium distribution

should be $P(x, \infty)_{\text{true}} = N \exp(-V(x)/D)$, while a simulated distribution will be $P(x, \infty)_{\text{true}} = N' \exp(-(V(x) + hS(h, x))/D)$, with an error S . It is possible to show [25] that for Euler $S(h, x) = (V')^2/4 - DV''/2$ while Heun has $S(h, x) = \mathcal{O}(h)$. For Heun error does not depend on the value of potential and its derivatives but only on integration step, making it possible to get more accurate estimates of the equilibrium distribution.

Analyses suggest that Heun scheme is among the most convenient algorithms, since higher order schemes do not do better, but they are much more expensive to implement on a computer.

Nonetheless, with respect to Euler, Heun is already more computationally expensive: if we restrict to the specific case of sets of interacting particles, the pseudo code in Algorithm 2 shows how this algorithm has to compute all the forces twice to perform a single step. However, in most cases the Heun scheme allows for large increases in simulation time steps compared to the Euler one, compensating for its higher single-step computational time and even making it possible to simulate larger times.

Chapter 2

Implementation of interactions in simulations

All the code work in this thesis is built on top of an existing code written and used in Microscale Robotics Lab, which can be found in repository [26]. Existing code performed simulations on 2D active Brownian particles moving in open or closed boundary featuring confinement. Original code used an Euler integrator to perform dry simulations of Active Brownian Particles ensembles with the only interaction being the steric hard-spheres correction. Moreover, it could simulate variously shaped hard boundaries as well as periodic and open boundary conditions.

2.1 Simulation Framework

In order to simulate most physical systems, some fixed steps are needed and our case study makes no exception. The main nodes of our simulation framework are:

1. Creation and preparation of all needed files
2. Initialization of particle ensemble inside the simulation box
3. Updating loop: a `for` loop on time steps, which includes:
 - (a) Force computation at time n , starting from positions at time n
 - (b) Generation of the needed random numbers for noise
 - (c) Integrator step: one step of the integration algorithm to get the ensemble at time $n + 1$
 - (d) Application of boundary conditions and hard-spheres correction
 - (e) File writing of positions and orientations of all particles in ensemble
 - (f) Computation and file writing of time-dependent global properties, e.g. polarization, pair correlation function etc.
4. Files closing
5. Plotting of necessary quantities, saving of animation for dynamics

The main file that gets created at point 1. is the *history* file, the one referred to in (d), to store positions and orientations of all particles. It gets prepared as a comma spaced values (CSV) files, where the column names are written in advance. CSV formatted files are more understandable and universally readable, which is useful when exchanging data between programming languages and manual checks are needed. At every step — or every n_{ds} steps when a down-sampling is used — we append a line to the history file, corresponding to the new simulation instant. In the original simulation code the full history was kept in a vector and then written to the file but in this new version we just update a single instant of the simulation and then write it to the file to improve memory efficiency.

In point 2. particles position are randomly initialized, with a uniform distribution in the simulation space. We need to make sure that particles are not superimposed to avoid nonphysical effects, thus we utilize the hard-spheres correction (see section 2.3.2) right after the generation of particles' positions. At this point it is possible to compute forces and torques acting on every particle, which are needed to compute positions at the next step.

From the computational point of view, our simulation framework is based upon object oriented programming: particle ensemble is initialized as an instance of the ABPE class, which holds all the variables that define the state of the system at a certain time 2.1. Ensemble parameters are numbers and all single

Variable	Type	Description
Np	Int64	Number of particles
L	Float64	Size of observation space (μm)
R	Float64	Particle radius (μm)
T	Float64	Temperature (K)
v	Vector{Float64}	Self-Propulsion Velocity ($\mu\text{m s}^{-1}$)
ω	Vector{Float64}	Self-Propulsion Angular velocity (rad s^{-1})
D _T	Float64	Translational diffusion coefficient ($\mu\text{m}^2 \text{s}^{-1}$)
D _R	Float64	Rotational diffusion coefficient ($\text{rad}^2 \text{s}^{-1}$)
x	Vector{Float64}	Particle X position (μm)
y	Vector{Float64}	Particle Y position (μm)
θ	Vector{Float64}	Particle orientation (rad)

Table 2.1: Structure definition of ABPE in Julia

particle variables, such as positions, are vectors of Np elements.

In point 3. positions and orientations get updated with both the deterministic and stochastic parts of the dynamics. This operation can be done either by updating a single instance of ABPE class or by appending to a pre-allocated vector one instance for every time step. The first method is more memory efficient, and it is the one to prefer if no memory is needed, i.e. only step n counts to determine what happens at step $n + 1$. A minimal example of this workflow, with Euler scheme as an integrator, is in algorithm 3.

2.2 Methods

All the simulation and analysis code for this thesis was written in [Julia](#) [27], except for the machine learning and inference part, where the Graph Network definition

Algorithm 3 The simulation algorithm

```

1: for  $i$  in  $[1, N_p]$  do
2:    $\vec{r}_i \leftarrow \vec{r}_{i,0}$   $\triangleright$  position  $\vec{r}_{i,0}$  is randomly initialized in the simulation space
3:    $\theta_i \leftarrow \theta_{i,0}$   $\triangleright \theta_{i,0} \sim \text{Uniform}([0, 2\pi])$ 
4:    $v \leftarrow v_0$   $\triangleright$  Can be initialized as a constant or a random variable
5: for  $n$  in timesteps do
6:   for  $i$  in  $[1, N_p]$  do
7:      $\vec{F}_{i,n} = \sum_{j \neq i}^{N_p} \vec{F}_{ij,n}$ 
8:      $w_{i,n} \sim \mathcal{N}(0, \sqrt{2D_t \Delta t})$ 
9:      $z_{i,n} \sim \mathcal{N}(0, \sqrt{2D_r \Delta t})$ 
10:     $\vec{r}_{i,n+1} \leftarrow \vec{r}_{i,n} + w_{i,n} + v(\cos \theta_{i,n}, \sin \theta_{i,n}) \Delta t + \vec{F}_{i,n} \Delta t D_t / k_B T$ 
11:     $\theta_{i,n+1} \leftarrow \theta_{i,n} + z_{i,n} + \omega_{i,n} \Delta t + M_{i,n} \Delta t D_r / k_B T$ 

```

and training makes use of Python libraries like PyTorch [28] and PyTorch-geometric [29], along with standard scientific computing tools like NumPy [30] and Pandas [31].

The Heun algorithm was chosen as the standard integration scheme for all the simulations, and integration steps vary with the different potentials and are indicated case by case. All the simulations were performed with real units, so that particles have a radius of 2 μm and the value for solvent viscosity is kept fixed at 1×10^{-3} Pas. If not indicated, standard temperature for simulations is held fixed at 300 K.

All simulations were performed in a two-dimensional square environment, where particles are disks, featuring periodic boundary conditions and a non-superposition condition (hard-spheres correction). Given that positions are simulated in open periodic boundary, all interactions are calculated periodically too, taking the shortest between the in-box and the periodic distance.

2.3 Boundary Conditions and Hard-Spheres Correction

2.3.1 Boundary Conditions

Original code featured open, periodic or hard boundary conditions. We will not go through all of them, but briefly, open boundary conditions means letting particles free to move out the simulation box, which in this case becomes just an *observation space*, while a hard wall involves computing boundary's gradient in any point and using it as a locally straight wall condition.

Periodic boundary conditions for a square simulation area are straightforward to make work if one places the origin in the center of the square. Then, if L is square's side, limits for particles' motion will be $\pm L/2$ in both axes. The actual limit for them to be reflected is $L/2 + R$, meaning a particle must be entirely sticking out before being moved to the other side. These physical boundary conditions are implemented according to algorithm 4.

Algorithm 4 Periodic Boundary Conditions

```

1: for all positions  $(x, y)_i$  do
2:   if  $|x_i| > L/2 + R$  then            $\triangleright R$  is the particles' radius,  $L$  box side
3:      $x_i \leftarrow x_i - \text{sign}(x_i)L$ 
4:   if  $|y_i| > L/2 + R$  then            $\triangleright R$  is the particles' radius,  $L$  box side
5:      $y_i \leftarrow y_i - \text{sign}(y_i)L$ 

```

2.3.2 Hard-Spheres Correction

Original simulation code featured just steric hard-spheres correction as interactions among particles, which are enough to study clustering, MIPS and boundary accumulation. The hard-spheres correction follows algorithm 5 according to [8].

Algorithm 5 The hard-spheres correction algorithm

```

1: for all couples of particles  $\{i, j\}$  do
2:    $d_{i,j} \leftarrow d(\mathbf{r}_i, \mathbf{r}_j)$             $\triangleright d(\cdot, \cdot)$  is the Euclidean distance
3:    $\mathbf{n}_{i,j} = (\mathbf{r}_i - \mathbf{r}_j) / d_{i,j}$ 
4:   if  $d_{i,j} < 2R$  then            $\triangleright R$  is the particles' radius
5:      $\mathbf{r}_i \leftarrow \mathbf{r}_i - \mathbf{n}_{i,j} d_{i,j} / 2$ 
6:      $\mathbf{r}_j \leftarrow \mathbf{r}_j - \mathbf{n}_{j,i} d_{i,j} / 2$ 

```

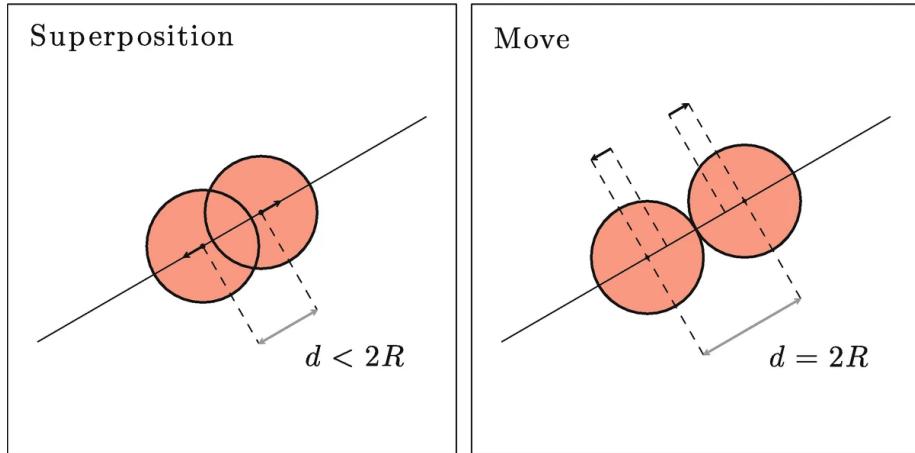


Figure 2.1: If in one step two particles superimpose, hard-spheres correction moves them at one diameter distance [8]

This operation involves calculating the distance matrix of a set of N_p particles, i.e. computing N_p^2 distances. Being a $\mathcal{O}(n^2)$ algorithm, this kind of correction can become computationally demanding, especially for high density and velocity systems, where collisions and clustering are more likely to happen. Moreover, correction showed in algorithm 5 is for only one step and it does not guarantee zero superposition afterwards. It becomes important to let the algorithm have a small tolerance, like 1/1000 of particle radius and to cycle the correction until the number of superposition becomes zero.

In the work cited in section 1.3, Martín-Gómez, Levis, Díaz-Guilera, *et al.* use a steep repulsive potential $\sim r^{-12}$ to simulate the excluded volume. In principle, since it does not require cycling, this approach seems faster, especially if one has to compute interactions. During the development of this thesis some tests were run and, with all integration algorithms, even though computing the excluded volume effect is actually faster than applying a hard-spheres correction, the steep potentials involved require a much smaller time step (1 or 2 orders of magnitude), making the simulation slower.

Separating superimposing particles in a way to keep them in contact may seem a strong assumption, since it is simulating a perfectly inelastic collision. This choice arise due to the finite contact time and micron sized physics of the active particles under study. In macroscopic Newtonian dynamics scenario (balls moving on a pool table), it is relatively easy to state collisions' elastic or inelastic nature. Furthermore, in the Microscale Robotics Lab and in literature [17], Janus active particles collisions are observed to have a finite contact time $\sim s$. This long collision time encouraged us to implement inelastic collision in simulations.

2.3.3 Modified Periodic Boundary Steric Interactions

Since hard-spheres corrections were implemented to study ABPs in confinement rather than open periodic boundary, this interaction could not take periodicity into account leading to possible superpositions at boundaries.

When boundary conditions are periodic, not taking them into account in steric interactions may cause some problems, on top of being not formally correct. With the hard-spheres correction as it was described in section 2.3.2, particles are separated when their distance is less than their diameter. With periodic boundary conditions, a particles is reflected *e.g.*, on the left hand side only when its center is out of the right hand side by more than one particle radius (R). This may leave a *frame* of size R around the boundary where inter-particle periodic distance can be less then $2R$.

When combined with potentials that feature a *soft* non-superposition condition, like a diverging positive force at a $2R$ distance, this can lead to pairs of particles sticking out of the border on opposite sides and superimposing a lot before one of them gets reflected on the other side, but when this happens, the force between them diverges, resulting in nonphysical situations like particles moving far outside the box.

To address this issue, we propose a modified version of hard-spheres corrections where distances are not computed in a Euclidean fashion, but x and y components are separately checked. For a box of size L , we compute $\Delta x = \min\{|x_1 - x_2|, |x_1 - x_2| - L\}$, and the same for y , which is the right periodic distance between any given pair of particles. If $\Delta x < 2R$, particles are guaranteed to superimpose when periodic boundary condition is applied. To eliminate this superposition, we implemented the hard-spheres correction as in section 2.3.2 to superimposing pairs at the boundary. This method allows us to identify superpositions at the boundaries, thus giving the possibility to form clusters starting from a *nucleation site* on the border.

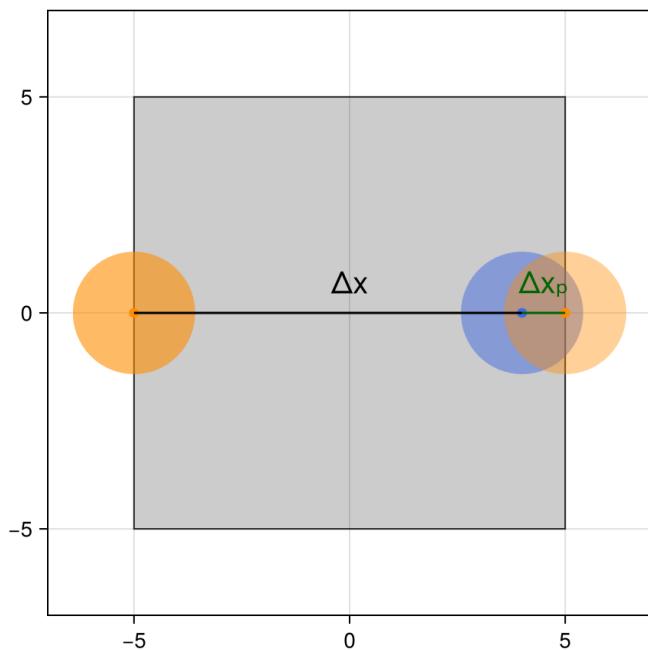


Figure 2.2: Orange particle is sticking out on the left side. Its *real* distance from blue particle on x-axis is Δx . Its periodic projection (faded orange) has a distance from blue particle $\Delta x_p < 2R$, hence superimposing.

2.4 Interaction Forces

2.4.1 Central Potentials

Spring, Coulomb, Lennard Jones, Yukawa, Weeks-Chandler-Anderson are just few examples of the myriad of central potentials that are used to model interactions in physics. The central potential is the basis of this thesis too, and having a fast implementation is crucial to simulate interactions in dense systems.

The first step to apply a potential is to compute a distance matrix between the coordinates of all particles. Then a function $F(d)$, only of inter-particle distance d , is applied to all the entries of this matrix to get the magnitude of the force between each pair of particles. In 2D, radial direction for the force can be computed from distance matrices both for x and y coordinates. If $d_{i,j}^x = x_i - x_j$ and $d_{i,j}^y = y_i - y_j$, then Euclidean distance is $d_{i,j} = \sqrt{(d_{i,j}^x)^2 + (d_{i,j}^y)^2}$, and if $\varphi_{i,j}$ is the radial direction from particle i to particle j , then

$$\begin{cases} \cos \varphi_{i,j} = d_{i,j}^x / d_{i,j} \\ \sin \varphi_{i,j} = d_{i,j}^y / d_{i,j} \end{cases} \quad (2.1)$$

splitting the force in its two components. Now the computed force can be used to calculate the deterministic step in the coordinate as in algorithm 3: given that we are in low Reynolds number regime, *velocity* is proportional to force .

2.4.2 Tested Potentials

We give here the expressions and parameters explanation for the potentials tested in this thesis. In the following, where not differently stated, units in constants are chosen to get a force in μN when distances are given in μm .

Spring potential

We used this potential to test a strong long-range interaction between particles, with some control over the equilibrium position. Spring force has two parameters, elastic constant k_S and rest length r_0 , so that

$$V(r) = \frac{1}{2}k_S(r - r_0)^2 \rightarrow F(r) = -k_S(r - r_0). \quad (2.2)$$

Coulomb potential

What we will often call Coulomb force is actually a r^{-2} order potential with a constant k_C that does not involve charges and it is present only to adjust the strength of interaction, that writes

$$V(r) = -\frac{k_C}{r} \rightarrow F(r) = \frac{k_C}{r^2} \quad (2.3)$$

and any sign variation, as well as the difference between repulsive and attractive, is absorbed in k_C . Being an attractive or repulsive-only potential, it can be used to study the effect of a potential in a relatively simple situation, especially when dealing with aligning interactions. Its divergence is not so steep thus letting us utilize larger integration steps.

Lennard Jones potential

The last interaction potential is Lennard Jones (LJ)

$$V(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \rightarrow F(r) = 24\epsilon \frac{1}{r} \left[2 \left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (2.4)$$

where σ is a location parameter that in our simulations will be fixed at the value of particle's diameter $2R$, and ϵ adjusts the strength of interaction. This potential has an absolute minimum for $r = 2^{1/6}\sigma$. LJ is a standard in molecular dynamics and disordered systems, both its original version and its variations are often used in ABP research, since this potential encloses both short range repulsion to avoid superpositions and medium range attraction to give structure to the system. The short range divergence is pretty steep and its strong non-linearity requires very short integration times, better if coupled with advanced integration algorithms.

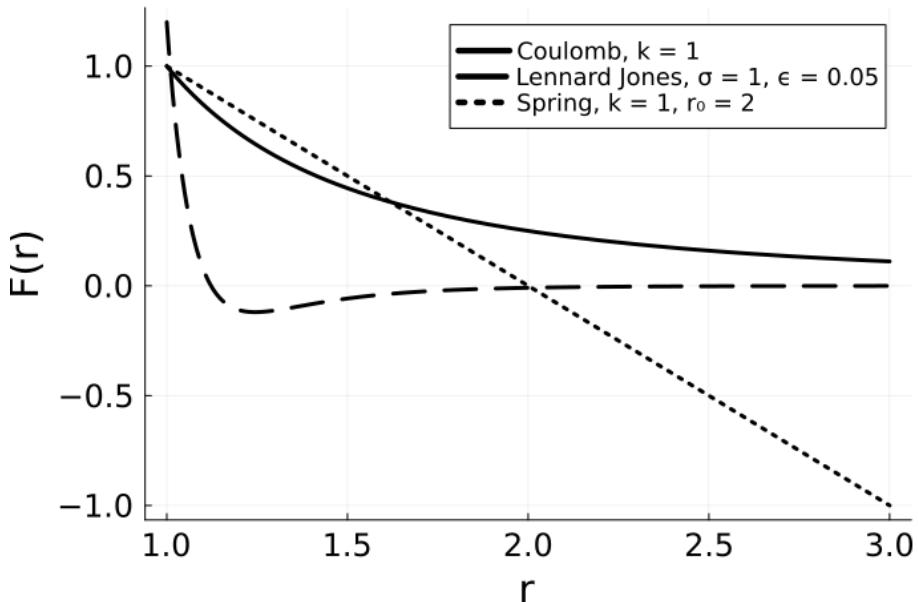


Figure 2.3: Tested interaction forces for given values of their parameters. For simplicity here, everything is kept non-dimensional.

2.4.3 Interaction Range

In some situations is useful to have a way to restrict the force computation only to particles closer than a threshold distance.

One could do this to imitate a topological interaction at short range, where a particle only interacts with its nearest neighbors. In the case of circular particles (or spherical in 3D), setting the range to $3R$ is enough to get this effect, since two particles in contact will have their centers at $2R$ and no other particle can be at less than $3R$ in the same directions. This way, central particle will interact with 6 other particles at most (this is the number of nearest neighbors for hexagonal

close packing order 2D) giving the effect of a nearest neighbor interaction in the case of contact. Clearly, this is not a *true* topological interaction, where two nearest neighbors interact regardless of their distance, but it can give similar effects if applied in high packing fraction cases, where particles come in contact.

Moreover, when dealing with fast decreasing potentials, the magnitude of the force between two distant particles can be effectively negligible and even lower than the numerical precision. If this is the case, it is obviously pointless to look into pairs of particles which are more than a certain distance apart. One can compute this distance for any given potential, provided some reference magnitude. An example is given in Figure 2.4. This is of paramount importance

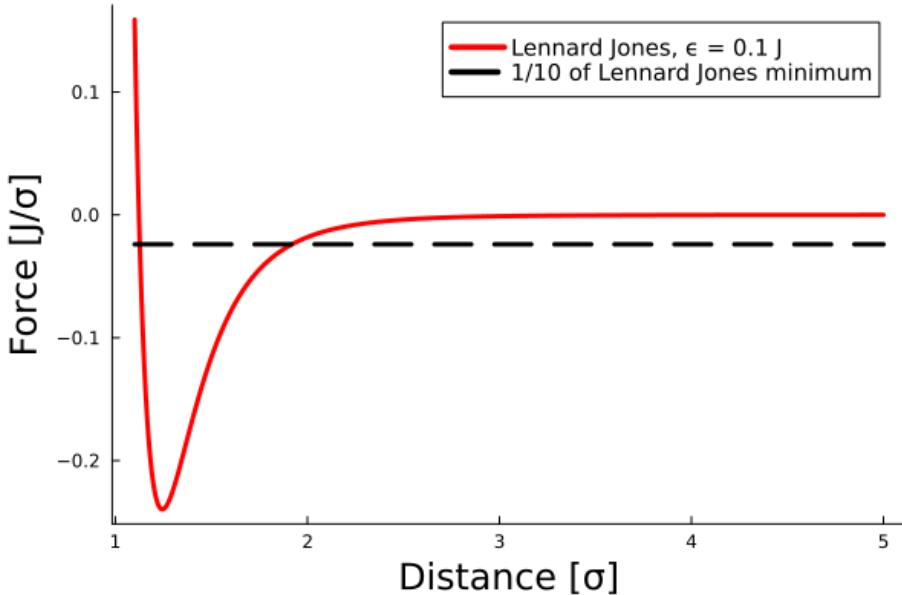


Figure 2.4: Cutoff for Lennard Jones force at 1/10 of the minimum. The cutoff distance is $\sim 1.9\sigma$

for speed. While a complete calculation of all the possible forces between n particles is an $\mathcal{O}(n^2)$ algorithm (the actual number of possible interactions is $n(n-1)$), the number of particles in a box of given area is constant in n if density remains fixed, so, scaling up the number of particles in the same conditions will be less demanding in terms of computational power.

Inserting a range in interactions makes straightforward to apply them in a periodic fashion. Though interactions were not featured in original code, functions for boundary conditions were implemented in a fast vectorized way, letting us reuse the existing function to reflect particles in periodic boundary conditions (pseudo code in algorithm 4) in the interactions part. The idea is: take one particle and make it the *central* one, then shift coordinates of the other particles so that the *central* is in $(0, 0)$, now just apply periodic boundary conditions to the shifted coordinates. This process should guarantee that the *central* particle interacts only once with any of the others, following a simple rule: when choosing between the true particle and the periodically shifted one the interacting particle is the closest one. An explanation of this process can be

found in Figure 2.5.

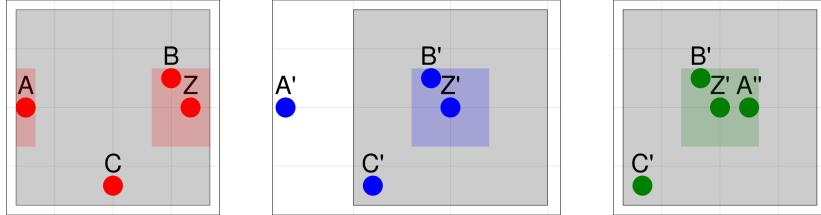


Figure 2.5: Red: original points with Z as central point and its interaction box sticking out on the opposite side. Blue: shifted points w.r.t. Z. Green: shifted points with PBC applied.

2.4.4 Aligning Interactions

There are several examples in literature of aligning interactions applied to ABPs systems [8], [14]. Most of them involve applying a torque which explicitly depends on the difference between the orientation angle of the two interacting particles; some of them have a constant torque which is applied to particles closer than a certain distance. To the best of our knowledge, nobody have implemented interactions in a way that unifies the central force interactions and the aligning interactions.

The model used in this thesis aims to keep a minimal amount of parameters while coupling positions and orientations of interacting particles, actually using only one more parameter than a force-only model. When applying a central potential interaction to a set of particles, the distances r between particles are calculated with respect to the particles' centers and the computed interaction force $F^i(r)$ is applied to the particles centers as $[F_x^i \ F_y^i]$ (2D simulations). Here every particle is identified by two positions: its center of mass, which lies in its geometric center, and an interaction center, which is translated with respect to the center of mass of an amount αR in the direction of self-propulsion, so that the off center position of each particle writes:

$$\vec{x}_{oc} = \vec{x}_c + \alpha R \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix} \quad (2.5)$$

where θ is the self-propulsion direction, R is the particle's radius, and $-1 < \alpha < 1$. Compared to the self-propulsion direction, the off-center position is on the front half of the particle when $\alpha > 0$ and on the back half when $\alpha < 0$. Distances, and hence forces, are thus computed between the off-center positions. Applying forces off center results also in an effective torque applied to the particle, leading to both translational (repulsive/attractive) and rotational (aligning) effects, coupling the two degrees of freedom.

The idea is to take into account the underlying particle's asymmetry not only in terms of self-propulsion but also in terms of interactions. It is a fact that the two faces of a Janus particle have different physical and chemical properties, which make them interact differently depending on the angle between their orientations and the line connecting their centers [17]. This can be due to a plethora of effects like chemical gradients, electrical forces, hydrodynamics

or fabrication defects that would be hard to model in details and effectively impossible to simulate for even a few particles. This model takes only the asymmetry and uses it along with standard potentials, to create a minimal dry framework that mimics such a complex wet dynamics. In doing so, we also show that this model displays some interesting statistical mechanics properties.

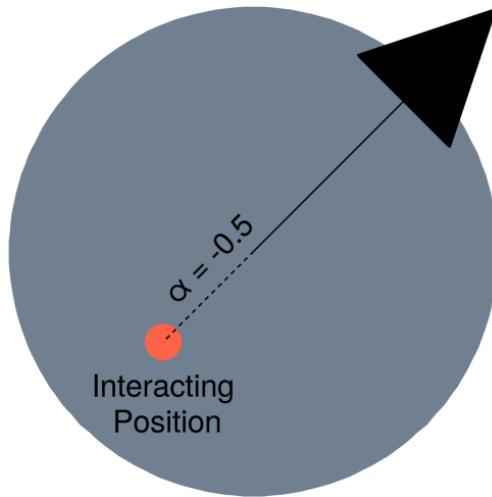


Figure 2.6: Red point: position from which interaction distances are calculated and to which force is applied. In this case, interacting position is at $-0.5R$ from particle's center, *i.e.* on the back side compared to the direction of self propulsion.

2.5 Qualitative Model Validation

Since the real interaction forces and torques between self-diffusio-phoretic particles are still a matter of deep investigation, it is not straightforward to check if a model is accurate or complete. Here, we look into a qualitative comparison between simulated and experimental behaviors, focusing on interactions between pairs or small groups of particles.

It is important to note that often in experiments clusters of particles are not caused by self-assembly, but are there as a by-product of the fabrication process, during which particles may remain attached. Nonetheless, studying single particles' interaction with clusters gives useful insights on contact behaviors. Our simulation framework does not have the possibility of physically attached particles, which might need a high attractive potential to mimic experimentally observed clumps. We chose Lennard Jones for its short range attraction, keeping $\sigma = 2R$ with R particle's radius, with negative off-center parameter α .

Furthermore, due to differences and irregularities in the fabrication process, not all particles have the same self-propulsion velocity. In simulations we reproduced these inhomogeneous velocities generating them from a distribution. Although most videos do not contain enough particles to build a sample velocity distribution, it is noticeable how very fast particles are rare if compared with slow particles. To catch this asymmetry, we generated simulated particles' velocities from an exponential distribution, allowing us to maximize the number of collisions. Also, particles orientation is difficult to measure in our experiments but some particles rotate around their axis, thus having both linear and rotational self-propulsion, like Chiral Brownian Particles [8], [32]. Since we do not observe a preferred direction in self-propulsive rotation, particles' angular velocities are generated from a distribution which is symmetric around 0; moreover, few particles present large angular velocities, so we generated them in simulations using a Normal distribution, to have control on its broadness.

2.5.1 Qualitative Comparison between Simulations and Experiments

Both the simulations we will refer to have been run in a $150 \mu\text{m} \times 150 \mu\text{m}$ box, with an exponential distribution in velocity characterized by its scale (mean) $\beta = 12/\log(50) \approx 3.07 \mu\text{m s}^{-1}$. This parameter is found as an empiric guess in order to have match experimental fractions of slow and fast particles. With the same line of thinking, we chose to generate angular velocities from a normal distribution centered around zero with $\sigma = 0.01 \text{ rad s}^{-1}$. Lennard Jones strength parameter is different for the two simulations and we will discuss it later.

As showed by Figure 2.7, the typical two body interaction in our experiments works as follows: two particle approach each other, they come in contact, then they undergo some interaction-driven rotation and they separate. This tendency to slide and rotate on each other is pretty evident and it pushed us to implement aligning interactions in simulations.

Figure 2.8 shows how this behavior is captured in a simulation with Lennard Jones strength $\epsilon = 0.1 \text{ pJ}$ and off center parameter $\alpha = -0.5$. As the interaction takes place, one of the two particles gets rotated to the right, and its motion continues in that direction. It is evident how the sliding contact behavior takes place as the result of the interplay between particles' self-propulsion, hard-spheres correction and attractive interaction potential.

Both in experiments and simulations, we also observe some other dynamics: one is the approach of the two particles that then keep rotating around each other, the other is the medium range (no contact) deflection of particles' trajectories.

Regarding triplets we have two different behaviors, the less frequent one is the direct formation (Figure 2.9) of a three particles' cluster while in the most frequent one (Figure 2.10) two particles get together and then a third particle joins the cluster.

In simulations the direct triplet formation is pretty rare and we almost never observed it. With a Lennard Jones strength $\epsilon = 0.3 \text{ pJ}$ and an off center parameter $\alpha = -0.7$, the observed triplet forms starting from a two particle cluster, then rotates on itself and breaks apart (Figure 2.11). Here we observe that the sliding behavior of particles upon each other is kept also in higher order clusters.

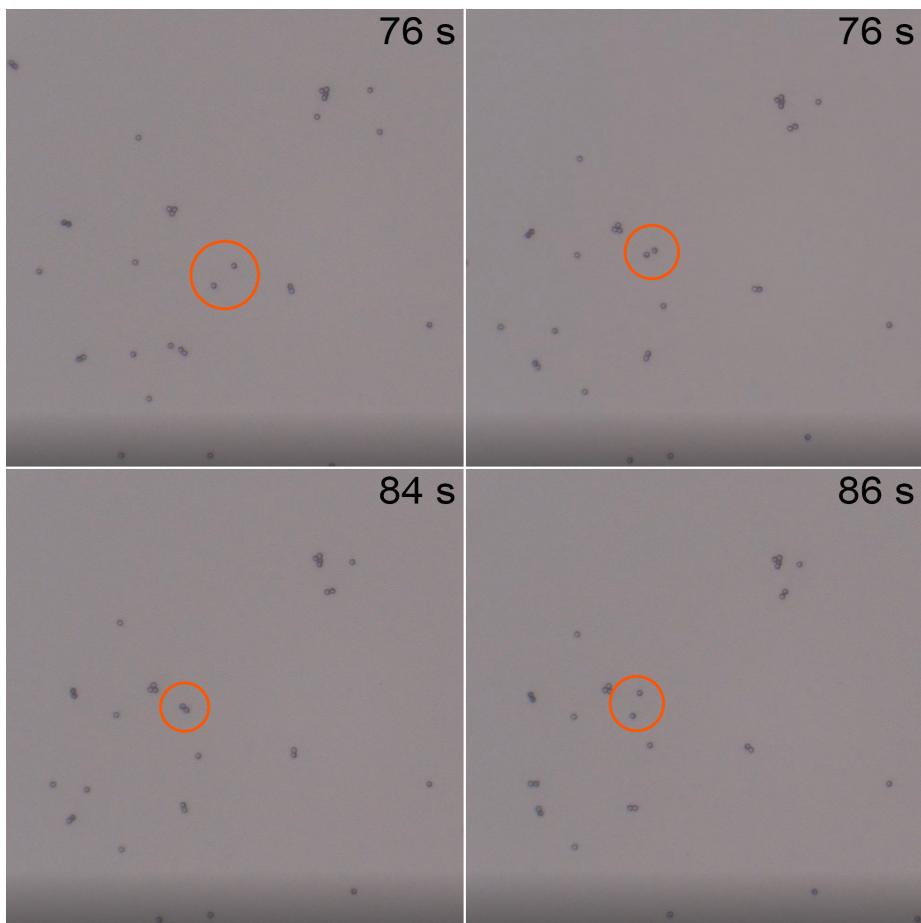


Figure 2.7: In the orange circle, formation and dissolution of a 2-particles cluster.

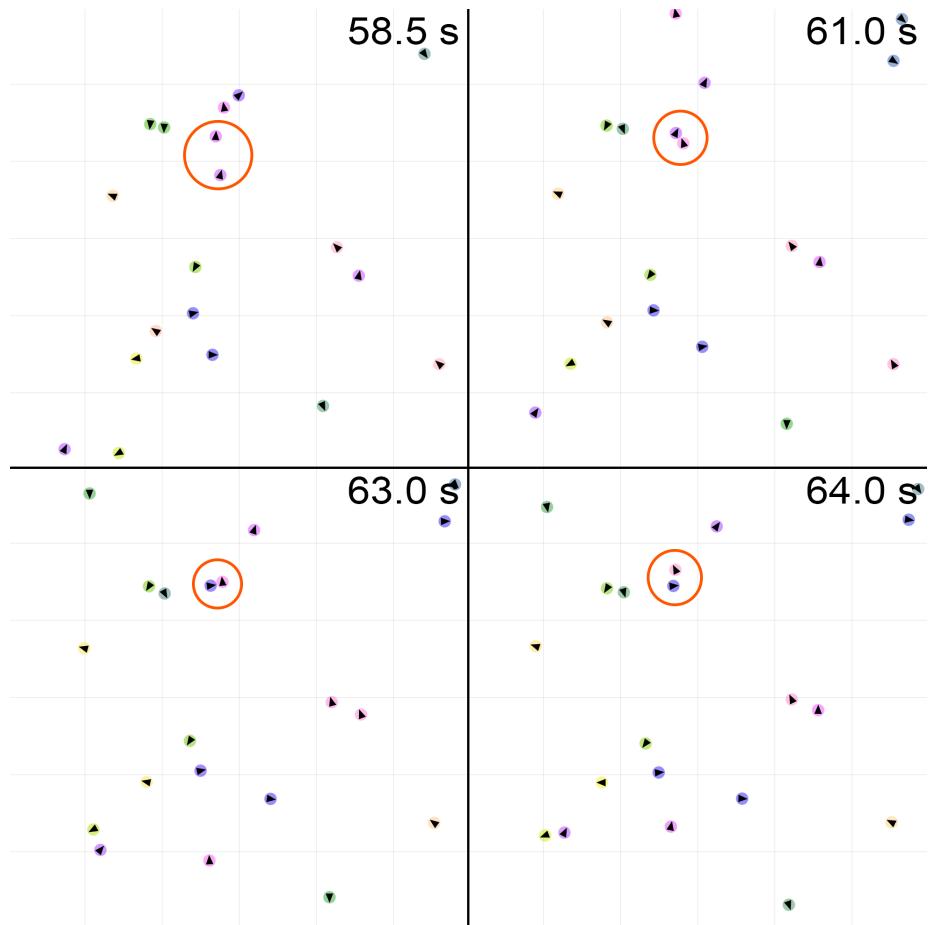


Figure 2.8: In the orange circle, formation and dissolution of a 2-particles cluster.

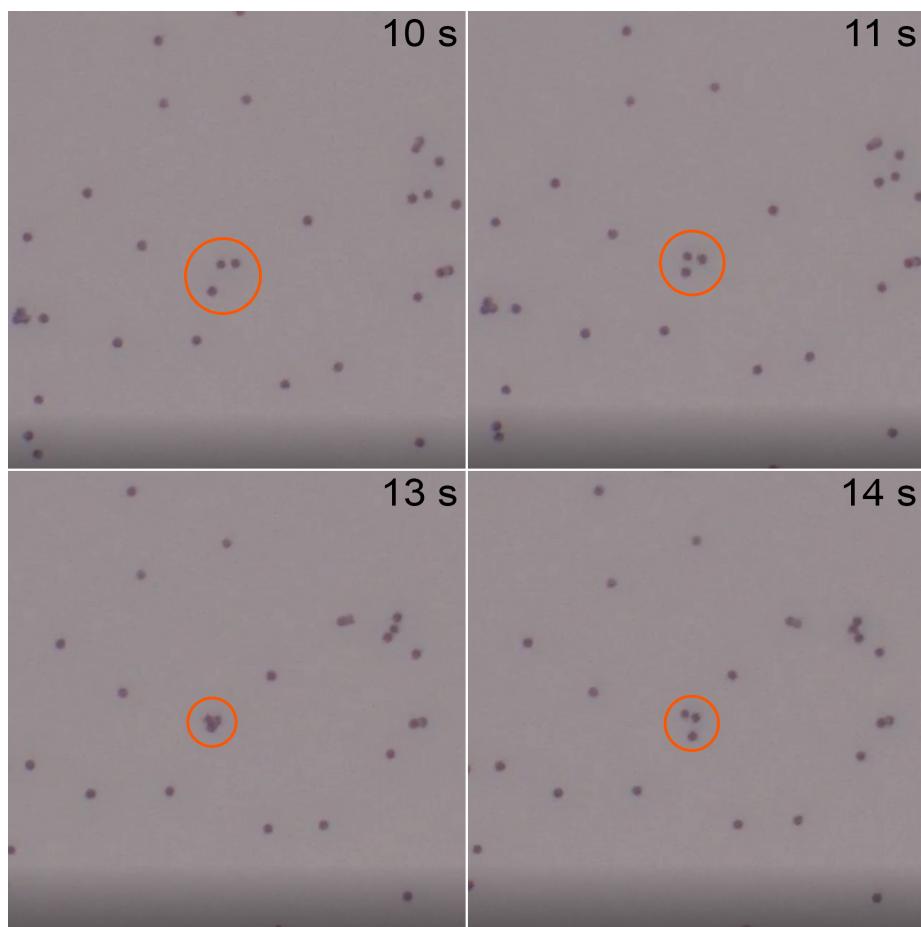


Figure 2.9: In the orange circle, direct formation and dissolution of 3-particles cluster.

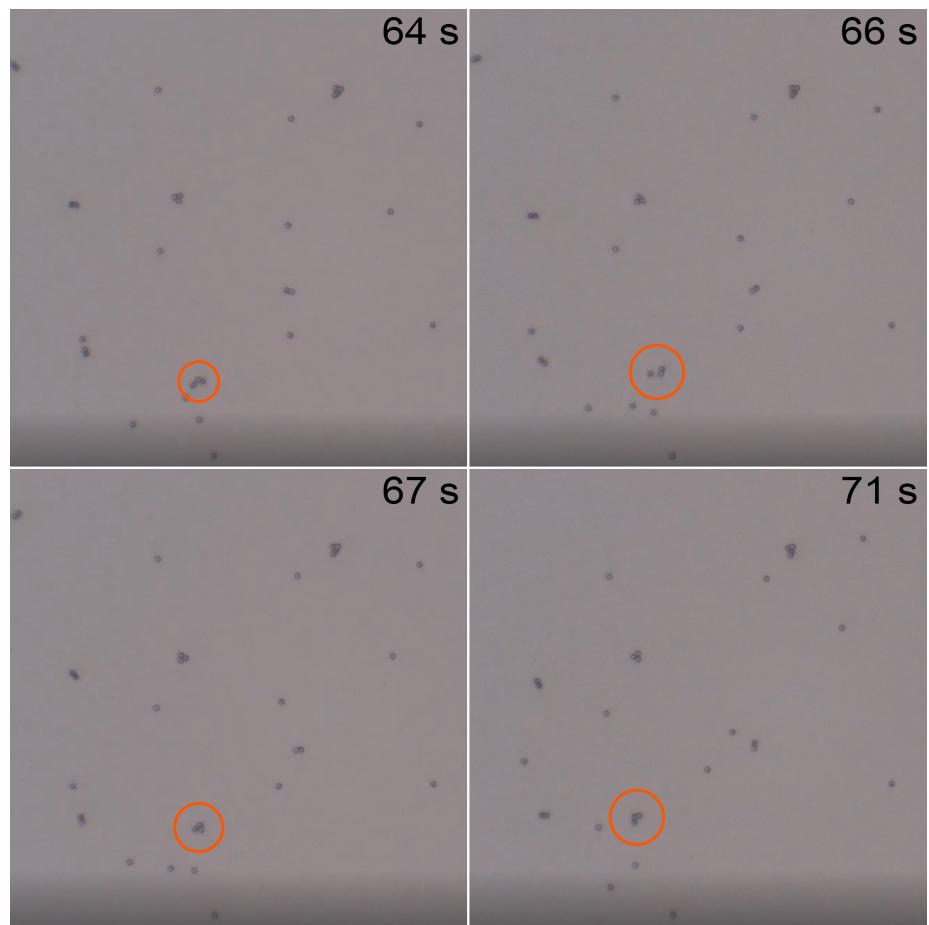


Figure 2.10: In the orange circle, formation and dissolution of 3-particles cluster starting from a couple.

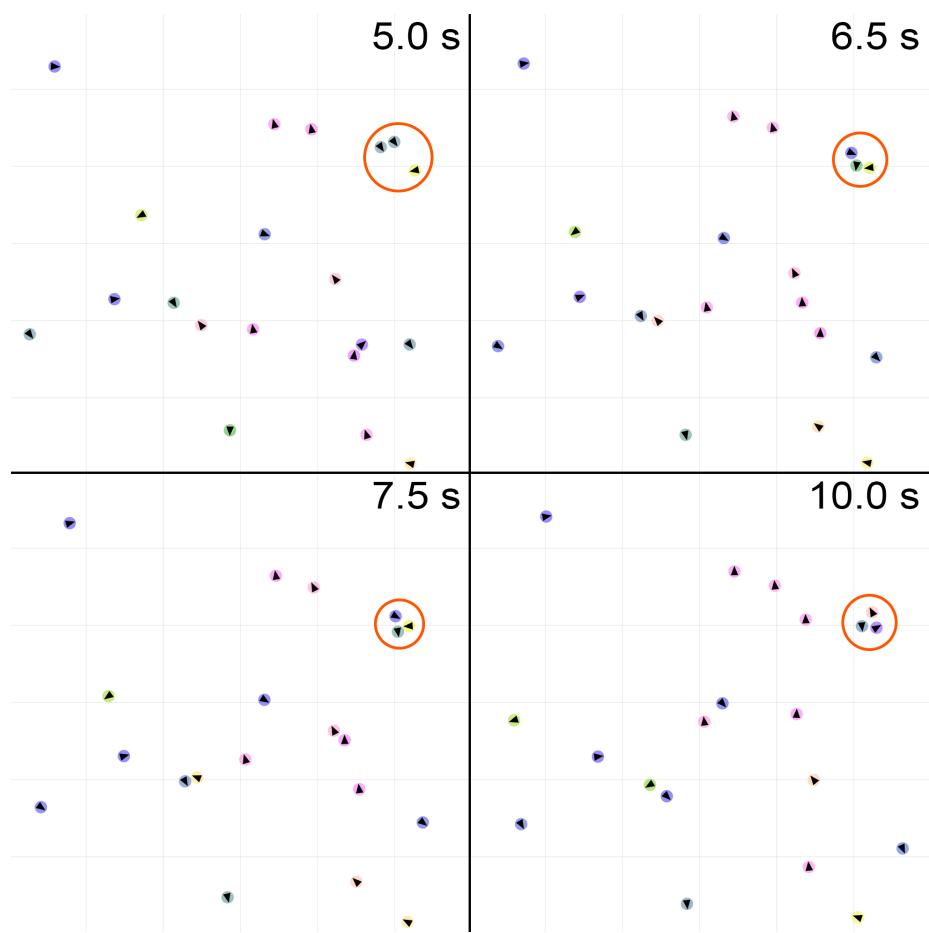


Figure 2.11: In the orange circle, formation and dissolution of 3-particles cluster starting from a couple.

We note that in the simulated situations, with $\epsilon = 0.1 \text{ pJ}$, $\alpha = -0.5$, with the observed packing fraction, no macroscopic clustering nor flocking is observed, as experiments show, while in the other case, with larger strength and off-center parameter the system tends to cluster after a while. These stronger parameters are only used in order to get noticeable clusters faster, while we are aware that they do not reflect the experimental behavior at steady state.

2.6 Conclusions to Chapter 2

We developed a simulation framework for systems of ABPs that features both steric and long-range interactions. To this, we applied a model for aligning interactions that tries to unify forces and torques, aiming to a minimal, yet descriptive representation of self-phoretic Janus particles.

We adapted our software to perform simulations with periodic boundary conditions, integrating steric and long-range interactions with such prescription.

From the optimization standpoint, although our code works well with small numbers of particles and packing fraction, some work is needed in order to scale it up for larger systems. One could implement a $\mathcal{O}(n \log(n))$ algorithm [33] for force calculation, but, as explained in section 2.4.3, limiting the range of interactions is a strong optimization on its own. This optimization would only make sense for potentials that go to zero with distance, while a hierarchical algorithm like the aforementioned one would be useful for longer range interaction forces, where an all-to-all calculation must be performed.

We qualitatively validated our simulations on experimental data, trying to match two- and three-bodies interactions with encouraging results. At this point we are missing some tools to analyze quantitatively our simulated system, investigating how features like off-center parameter, velocity, velocity distribution and angular self-propulsion change collective behaviors of ABPs ensembles. The development and utilization of such tools will be illustrated in the following chapters.

Chapter 3

Analysis of Structure and Dynamics

This chapter focuses on the development of tools to study ensembles of Active Brownian Particles with aligning interactions, and how these tools can be used to study collective behaviors of ABPS systems.

3.1 Analysis Functions

Our investigation in structure and dynamics is based upon a small yet descriptive set of tools that help unveiling how particles spatially organize and how they move.

3.1.1 Pair Correlation Function

Pair correlation function, *radial distribution function*, or simply $g(r)$ describes particles' density distribution with respect to a reference point in the system. Therefore, $g(r)$ is a function only of the separation $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ between particles' pairs. It quantifies how particle in an ensemble position one with respect to each other, giving information about the *structure* of the system. Here we will use it to get insights about the onset of long range order in the system. This function plays a key role in physics of matter since it is possible to measure it in radiation-scattering experiments [34] and it is often used in cases where, differently from our problem, particles' positions are not directly accessible.

It is possible to derive definitions for $g(r)$ starting from first principles, like phase-space distribution functions, as it is done in [34]. Here we will use the Dirac's δ -based expression, which is most reliable measure when particles positions are well known,

$$\left\langle \frac{1}{N} \sum_{i=1}^N \sum_{j=1}' \delta(\mathbf{r} - \mathbf{r}_j + \mathbf{r}_i) \right\rangle = \rho g(r) \quad (3.1)$$

where the prime on the summation sign means not counting terms with $i = j$. It is possible to show that the left hand side approaches the value of the overall number density ρ at large distance.

In order to compute $g(r)$ in our simulations, we first need to take distances between all pairs of particles accounting for periodic boundary conditions. Then, to simulate the behavior of a sum of deltas, we divide simulation space in bins by radius, where the area of each bin is computed as the intersection between a circular crown and a square, to respect the geometry of our simulation box without introducing biases. After counting the number of particles in each bin, we normalize by the average density to make $g(r) \rightarrow 1$ at long distance.

3.1.2 Cluster Size

As discussed in section 1.3, ensembles of ABPs tend to cluster when an attractive interaction is present or conditions (velocity, packing fraction) are right for MIPS to occur. In most cases, particles form one big cluster while some of them remain in a gaseous phase outside of said cluster. The way system form a big cluster is often through the formation of small clusters throughout the simulation box. These small clusters are dynamic in the size and stable enough to observe for long time periods.

It is possible to extract some information about small clusters from the $g(r)$, however, it is more suited for global structure formation study than for the local cluster formation. In order to get some insight about partial clustering, we developed the cluster size analysis.

Our method was built using DBSCAN [35], a density-based clustering algorithm, which was originally created for data-science but, with the right parameters, can become useful for physics. DBSCAN starts from a point and adds to the cluster associated with that point all the others which are at a distance of less than a threshold. A point having more than M_p points in its vicinity is said *core point* or *seed*. A point in the vicinity of a *core point* is a *border point*. All the others are *noise points*. Using 1 as M_p (i.e. a cluster must be formed with at least 2 points) and particle radius or interactions range as threshold distance, this algorithm's results correspond to the common sense conception of what a cluster is. In our case, we used $5\text{ }\mu\text{m}$ as threshold distance, enough to join contacting particles with $0.5R$ tolerance, but restricting to one layer of particles in contact, in order to consider only continuous groups as clusters. The advantage of adapting an existing algorithm is that it has some very fast pre-built implementations which are probably more efficient than one could write from scratch.

3.1.3 Local and Global Polarization

Whenever the agents of a system have a directed motion, as is the case for ABPs, it is worthwhile to analyze the order in the orientation degree of freedom of agents.

For global polarization, we use the following definition [15]

$$P = \frac{1}{N} \left| \sum_{k=1}^N e^{i\theta_k(t)} \right| \quad (3.2)$$

where θ_k is the orientation of k -th particle. This parameter is 1 when all particles are aligned and 0 when they are pointing in random directions.

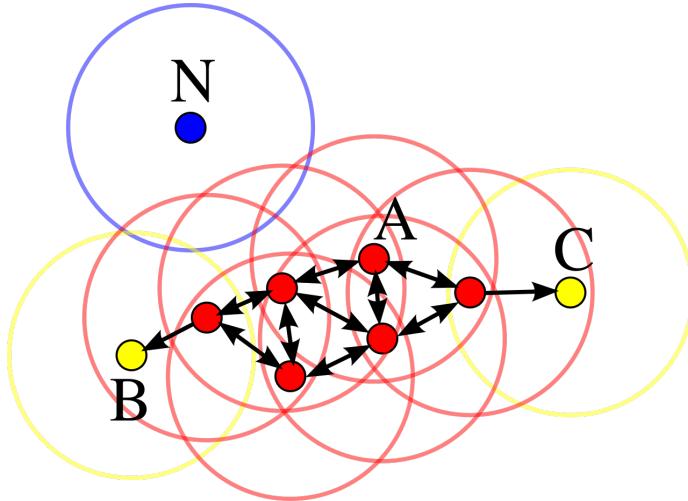


Figure 3.1: “Illustration of DBSCAN cluster analysis (minPts=3). Points around A are core points. Points B and C are not core points, but are density-connected via the cluster of A (and thus belong to this cluster). Point N is Noise, since it is neither a core point nor reachable from a core point”. [36].

In the phase transition section we will use the polarization susceptibility $\chi(P) = N_P(\langle P^2 \rangle - \langle P \rangle^2)$. For phase transition like the one we analyze, from disordered to flocking phase, the susceptibility of an order parameter has a peak in the proximity of the critical value for the transition, in this case, the orientational coupling parameter.

3.1.4 Local Polarization

As explained in previous section, local properties could provide crucial information about the emergence and properties of global self-organization in the system. When aligning interactions are present and packing fraction is small or we are in a transient, small clusters of particles tend to align their directions, which in terms of global polarization order parameter would correspond to a disordered state, i.e. $P = 0$. We can exploit the fact that clustering algorithms assign points to clusters and then measure how polarized the single groups are, meaning we compute a local polarization order parameter. Averaging over clusters, this parameter \bar{P} will be close to 1 even if P is kept low by the different orientations of clusters, making it a good parameter to study local properties and short range interactions.

3.2 Flocking as a Phase Transition

As mentioned in section 1.3, Martín-Gómez, Levis, Díaz-Guilera, *et al.*, as well as Negi, Winkler, and Gompper [14], [16] showed how inserting an explicitly aligning interaction in the simulation can make the whole system polarize, meaning all particles move in the same direction. This phenomenon is called *flocking*.

In Figure 3.2, an example of how the alignment process works. The interaction

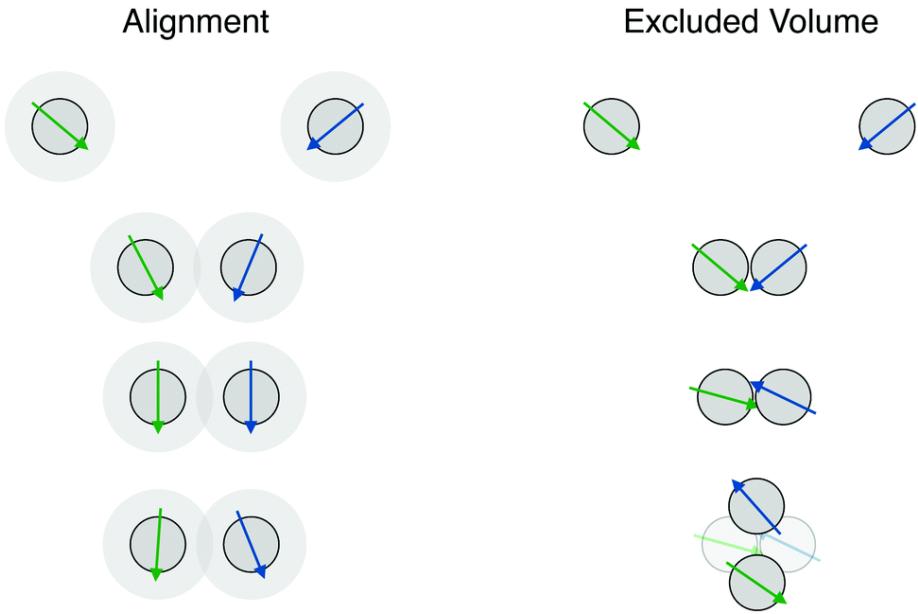


Figure 3.2: Behavior of two particles with explicitly aligning interactions, compared with excluded volume steric interaction only. Adapted from [14].

used by Martín-Gómez, Levis, Díaz-Guilera, *et al.* is an explicitly aligning torque where $T \sim \sin(\theta_i - \theta_j)$, with particles orientation are *explicitly* included in the expression and $|T|$ has a minimum where $\theta_i = \theta_j$ [14].

Differently from their case, here we show how a repulsive off-centered interaction, although with limited range, can lead an ABPs system to reach a flocking state. The selected interaction force is $F(r) = kr^{-2}$, akin to a Coulomb or gravitational force, where k is a constant. When both the off-center magnitude α and the constant k have a positive value, front sides of interacting particles repel each other, meaning that two particle swimming towards each other will tend to align their orientations. Our alignment thus comes automatically from applying a force to a non centered position. In Figure 3.3 an example of how flocking transition happens in our simulations with $k = 10$, with an interaction range of $40\text{ }\mu\text{m}$.

3.2.1 Methods

We studied flocking as a II order phase transition, using the global polarization P as order parameter and varying α , to change the magnitude of orientations coupling. We have $k = 1.0$ and the range of interaction is $5\text{ }\mu\text{m}$, which, being $2.5R$, with R particle radius, is enough to make particles interact only with one layer of particles around them, simulating a nearest-neighbor interaction. Though interactions range is so short, a long-range order establishes for large enough α , making the whole ensemble polarize.

Integration step is $5 \times 10^{-2}\text{ s}$ and simulation is 2×10^5 steps long, for a resulting simulation time of 10^4 s , enough to let the system thermalize and reach a steady state as shown in Figure 3.4. Every 100 steps, global polarization is computed

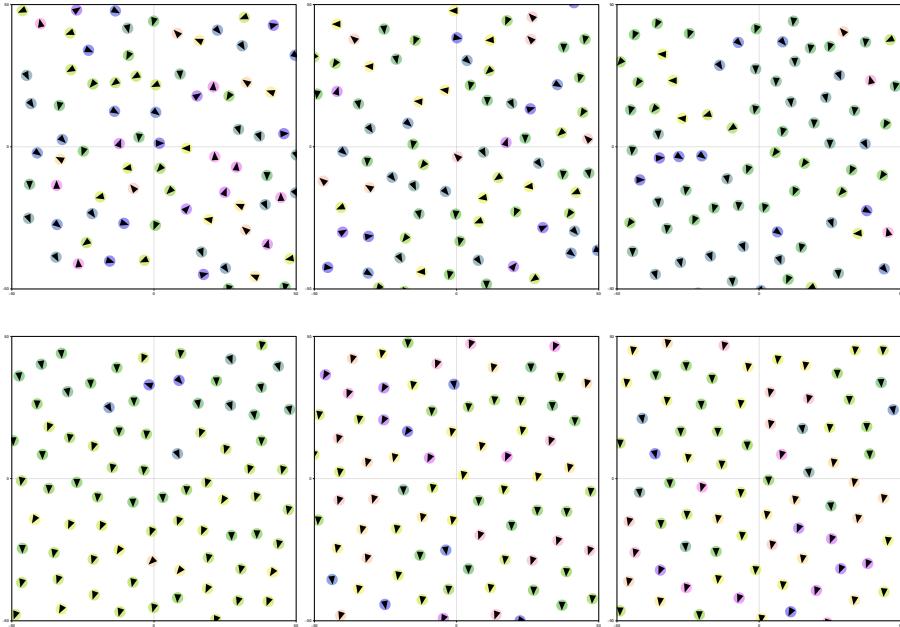


Figure 3.3: Flocking transition: all particles in the system tend to polarize after few seconds. Images from top left to bottom right correspond to situation after 1 s, 5 s, 10 s, 15 s, 20 s and 25 s

and appended to a vector that, from now on, will be called the *polarization history* vector. For each parameter set, we performed one simulation only: in this aspect, our analysis differs from the one done by Martín-Gómez, Levis, Díaz-Guilera, *et al.*, where means and standard deviations are computed over independent realizations. We believe that, with fixed simulation time, making one longer simulation is more convenient than more shorter ones, since the system has to thermalize only once, letting us waste less data. When everything is done in one simulation, some resampling technique is required to compute an estimate of the error as will be explained shortly. We proceeded to analyze data as follows: after discarding the first 2×10^4 steps (1000 s) as thermalization, the polarization history vector is treated using both a blocking and a JackKnife resampling.

Blocking means grouping all the history in subsets of subsequent points and replacing each of them with their average. This process is executed in order to compute an estimate of the standard deviation for observables of interest, calculating it as if each block was independent from the others. Choosing the block size is a matter of discussion: a large size will result in blocks with smaller correlation, but the small number of the obtained bins will make oscillations (error on the error) larger. In theory, the value of the standard deviation will saturate to its *real* value increasing the size, but in reality one has to make some trade off with the oscillation size.

Blocking only will not solve the fact that all data comes from the same sample. In order to have good estimators for means and variances of relevant observables

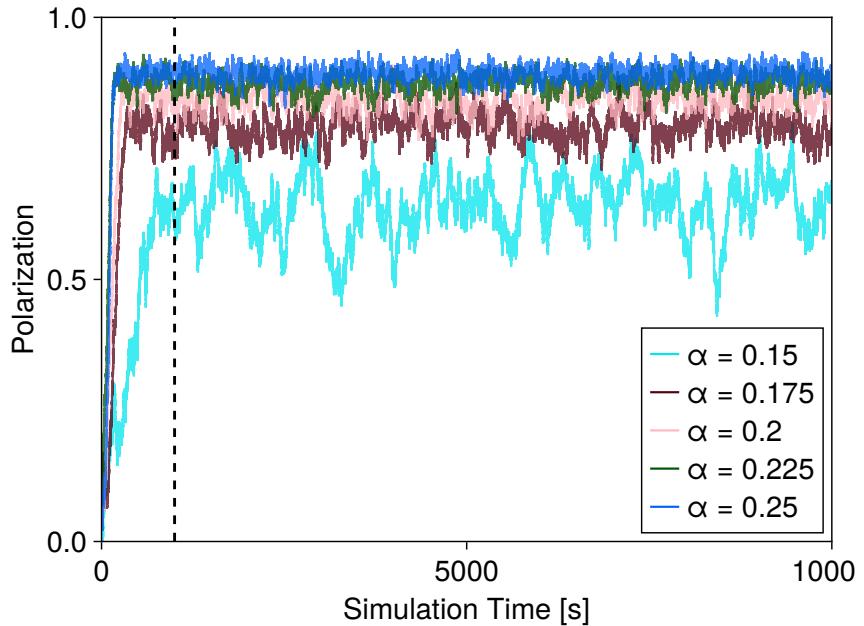


Figure 3.4: Temporal evolution of polarization for $v = 10 \mu\text{m s}^{-1}$, and several values of the off center coupling. Black dashed line is the number of points we discarded.

we need a resampling technique, such as the well-known Bootstrap or, as we used, JackKnife, where we replace each point with the mean of all the other points in the history. Resulting sample standard deviations for polarization and susceptibility are plotted in Figure 3.5 as a function of block size.

For increasing block size, standard deviation of relevant observables approaches its limit value, while oscillations around this value increase as the number of blocks becomes smaller. We used 2500 as a block size, to saturate the standard deviation while containing the amplitude of oscillations, as is explained in 3.5.

Using both the blocking and JackKnife resampling, we compute standard deviations taking care of auto-correlations in data.

3.2.2 Results

For every value of α , polarization is taken as the mean of the polarization history vector, with thermalization cut off, while we computed susceptibility using both a blocking and a JackKnife resampling. As a result, Figure 3.6 shows the expected behavior of a II order phase transition: as polarization P jumps with a discontinuity from 0 to 1, its susceptibility has a peak when α approaches critical value and then returns back to zero (see, as a reference, Figure 1.3). Here we notice how the susceptibility peak shifts up and to the right for decreasing velocity.

Though the lowest-energy configuration is the one with all particles aligned due to the repulsion that take place between the fronts of interacting particles, this model does not present an explicitly aligning torque like the one of the

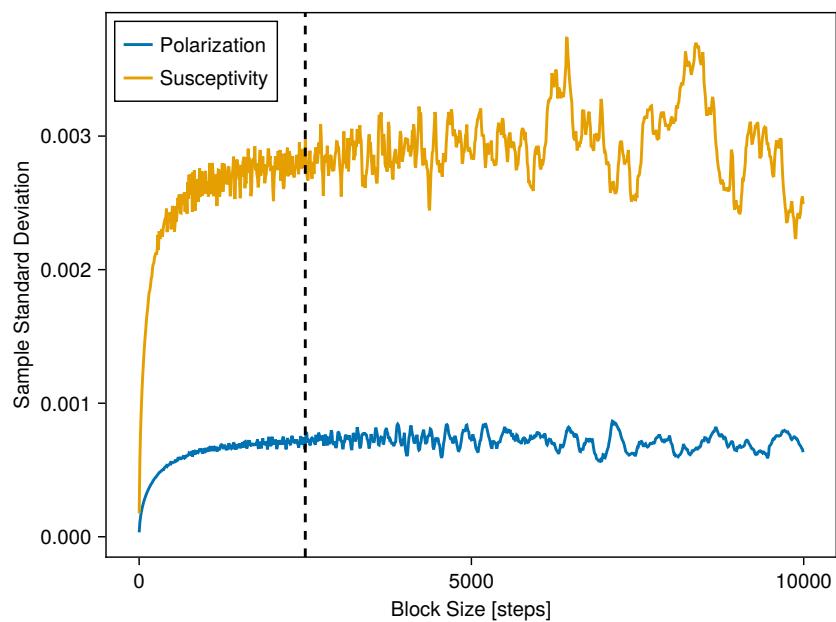


Figure 3.5: Standard deviation of polarization and susceptibility versus blocking size. Black dashed line is the block size of 2500 points that was used for this work. For this plot, $v = 10 \mu\text{m s}^{-1}$, $\alpha = 0.25$. Both a blocking procedure and a JackKnife resampling are used here.

form $T \sim \sin(\theta_i - \theta_j)$ used by Martín-Gómez, Levis, Díaz-Guilera, *et al.* Still, a flocking phase transition takes place with an evolution akin to the one showed in [14] as graphs in Figure 3.6 clearly show. Here, the off-center parameter α plays the role of the coupling g in reference paper. In cited work, Péclet number was changed to see its effect, while here we used velocity as an equivalent parameter, with the only disadvantage of not being non-dimensional.

3.3 Clustering and Local Order

Lennard Jones (LJ) is a standard potential in interacting ABPs analysis and it has a chance of representing well real world dynamics (see section 2.5), thus making it important to study its effects on ABPs collective dynamics. To quantify this effect in various situations, we investigated both global and local polarization, as well as clustering, to better understand the effects of the different model parameters on both positional and orientational degrees of freedom. As discussed in section 2.5, in experiments not all particles move with the same self-propulsion velocity because of imperfections in fabrication processes. For the same reason, some particles show a nonzero angular velocity that makes their dynamics more well-explained by a Chiral Brownian motion [8]. Thus, to capture these experimental observations, we added aforementioned features in simulations, and then studied how such features can change collective behaviors of an ensemble of self-propelled particles.

From now, unless otherwise stated, the interaction used is Lennard Jones with parameters $\sigma = 2R$, and strength parameter $\epsilon = 0.1 \text{ pJ}$. The value for σ is kept fixed to respect the natural units of the system, while the strength parameter is enough to make the system cluster with the right parameters while keeping the integration step not too low. Interaction range here is $40 \mu\text{m}$, or 10σ ; we chose this value not for physical reasons but for speed sake: because it is the cutoff distance after which LJ force magnitude becomes smaller than $10^{-6}|F_{min}|$, where F_{min} is the minimum value of the force, computing forces smaller than that will be completely irrelevant for our purpose and just make simulations heavy. Simulations involved 250 particles with $R = 2 \mu\text{m}$ in a square of $175 \mu\text{m}$ side, for a resulting packing fraction of ~ 0.1 . Simulation time is 1000 s but shorter times are showed in graphs as needed. Integration step can vary with parameters, to ensure the best trade-off between stability in the simulations and execution speed, but for most simulations $5 \times 10^{-3} \text{ s}$ was chosen as integration interval and when different values are used they are declared in sections.

For every parameters set, we simulated five randomly generated starting conditions. When distributions are involved in the simulation, e.g. for linear and angular velocity, probability distribution functions are resampled at every repetition. For cluster size and number, as well as polarization figures, lines in graphs are averages of those five simulations, with ribbons around them represent standard deviations around the mean. Radial distribution functions are plotted for the final instant of just one of the simulations. Transient time for polarization is measured as the time it takes for this order parameter to reach 95 % of its maximum.

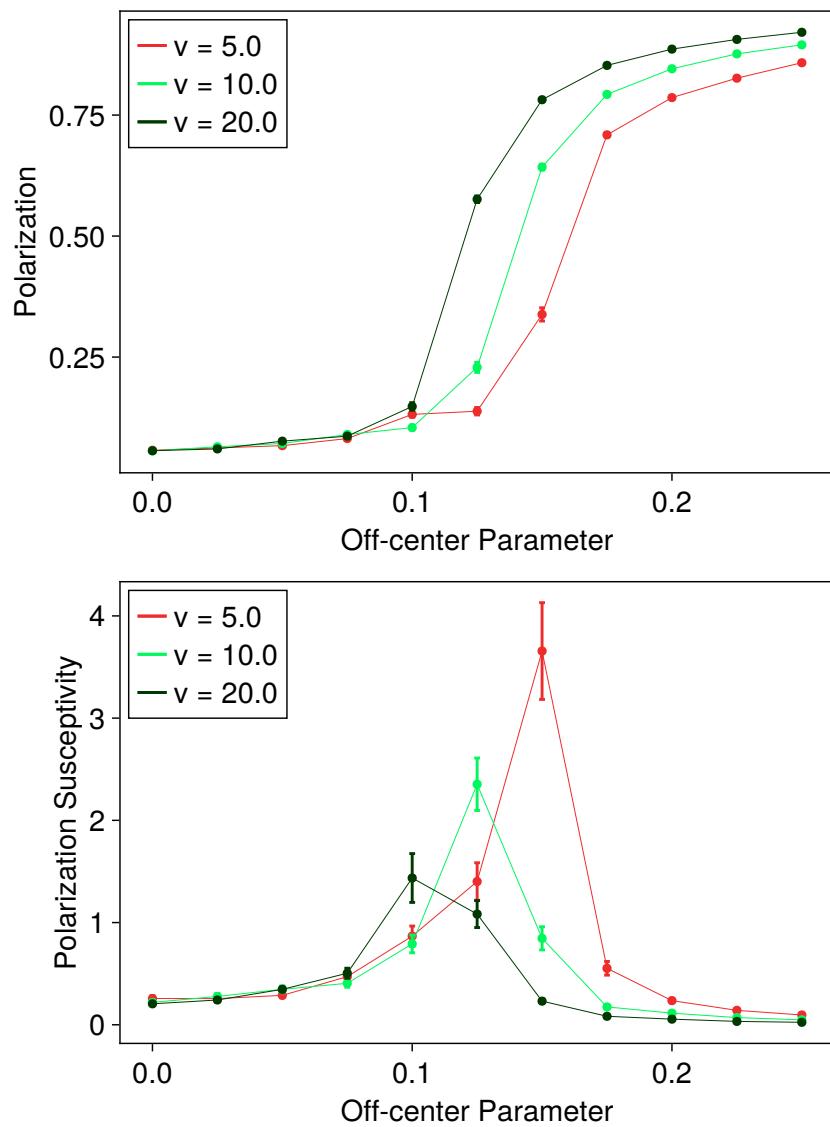


Figure 3.6: Diagrams for polarization and relative susceptibility for the transition to ordered phase.

3.3.1 Velocity

In order to study the effect of velocity, we first computed all the quantities of interest in the case of passive particles with an aligning LJ interaction, and then added a self-propulsion velocity. This active velocity results in non-isotropic system, giving a preferred, giving a preferred direction to particles, so we also investigated how velocity interacts with the sign of the directional coupling parameter α , repeating the analysis for positive and negative values.

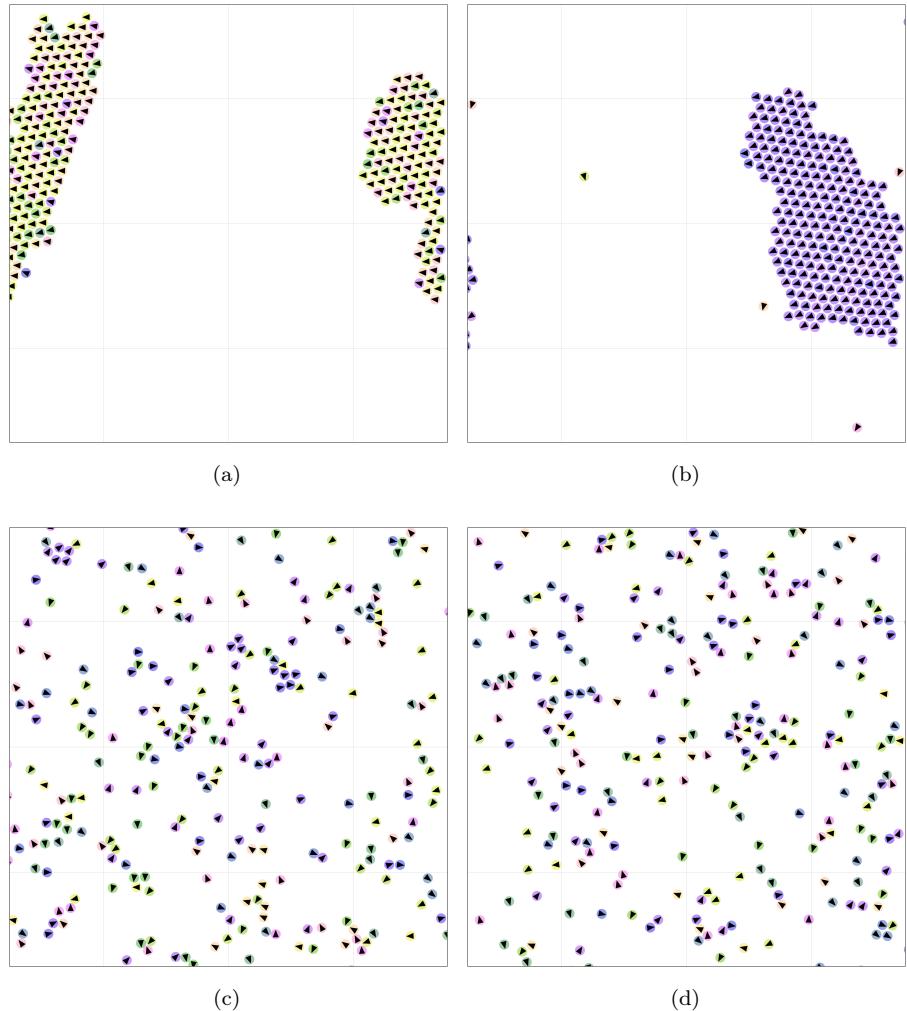


Figure 3.7: Representative simulation snapshots corresponding to values of velocity and α : (a) $v = 10 \mu\text{m s}^{-1} \alpha = 0.5$, (b) $v = 20 \mu\text{m s}^{-1} \alpha = 0.5$, (c) $v = 10 \mu\text{m s}^{-1} \alpha = -0.5$, (d) $v = 20 \mu\text{m s}^{-1} \alpha = -0.5$

In Figure 3.8 it is noticeable how an ensemble of passive particles with an aligning interaction has a baseline local polarization, while the global is almost zero. Adding self-propulsion has the consequence of making the system polarize, with a transient duration that depends on the velocity: faster particles tend

to align faster. A noteworthy characteristic is that particles with interacting position on the back do not tend to align globally, and local polarization does not rise over the baseline value.

This can also be noticed in Figures 3.7 and 3.9, where, when the off center position is on the back, velocity tends to hinder clustering, leading to a more sparse final situation. We anticipate the following explanation for this: for same velocity particles, head to tail collision is unlikely because interacting positions of these particles will always be at a distance $> 2R$, where interactions are relatively weak. Radial distribution functions in Figure 3.10 show how particles which swim with their interacting position on the back tend to develop less long range order, regardless their velocity.

3.3.2 Velocity Distribution

A simple way to add experiments inspired complexity into the model is to initialize particles with different self propulsion velocity. This way collisions are more likely and in a greater variety of directions' configurations. In this sections, we used the standard potential with an off center parameter $\alpha = 0.5$.

To implement different velocities we used normal distribution, keeping the mean constant at $10 \mu\text{m s}^{-1}$ while changing the standard deviation in $1 \mu\text{m s}^{-1}$, $2 \mu\text{m s}^{-1}$, $5 \mu\text{m s}^{-1}$ and $10 \mu\text{m s}^{-1}$. To avoid having particles that swim backwards with respect to their self propulsion direction, the distribution was truncated at zero.

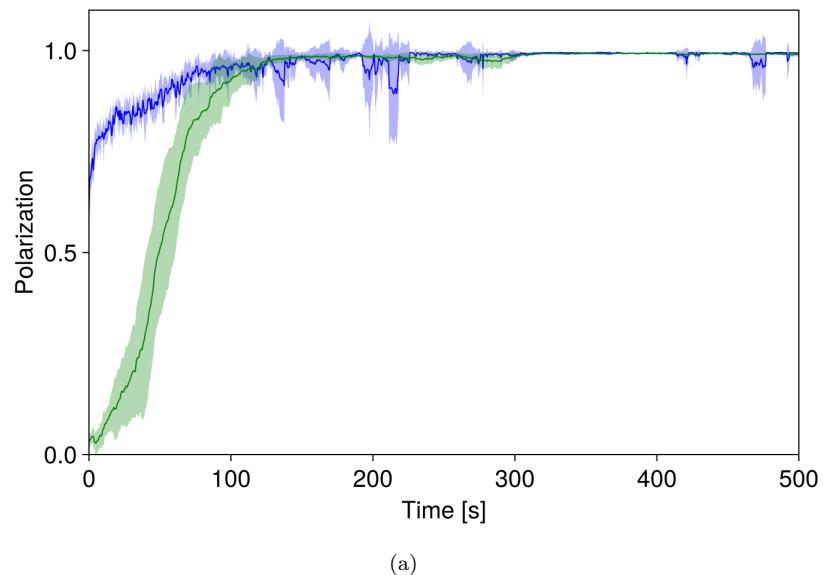
Considerations about collisions could lead to the expectation that, with a velocity distribution, particles will cluster more easily in a MIPS fashion. In fact, both cluster size and $g(r)$ (Figures 3.13 and 3.14) show how adding different velocities to the system tends to hinder packing, leading in the end to a more sparse system, compared to a constant velocity case.

From the polarization standpoint, increasing the broadness of the velocity distribution tends to slightly reduce the amount of total alignment in the system, while local polarization remains higher. A remarkable observation is that although in our study we showed that alignment and clustering are often related, here a drastic downfall of clustering does not reflect in a big difference in polarization, resulting in a sparse system with a high alignment. This is immediately noticeable, in addition to the graphs in Figures 3.12 and 3.13, in the snapshot in panel (d) of Figure 3.11. In line with previous section, faster velocities make transient times shorter. These results are relevant, since alignment in packed ABPs systems is a well known fact [15], while we observed that faster particles tend to align slower ones in a sparse configuration as well, when little or no clustering takes place.

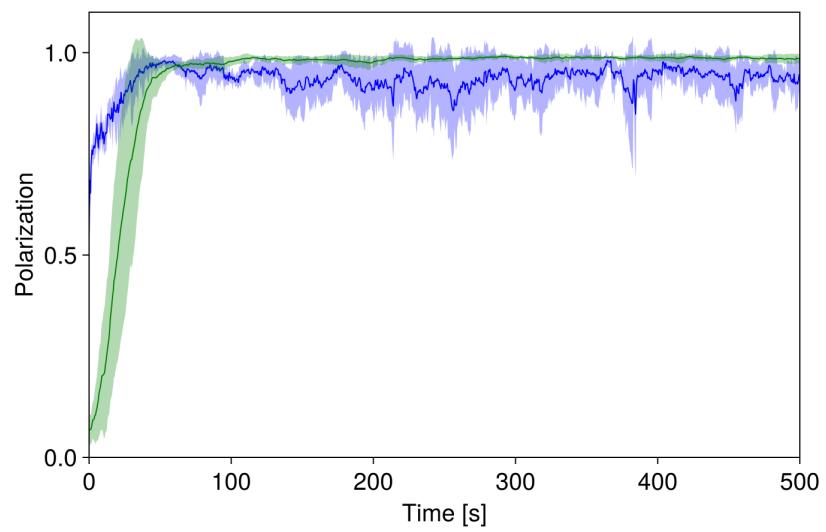
3.3.3 Orientation Coupling

To understand the role of the orientational coupling parameter α , we performed simulations with velocity $20 \mu\text{m s}^{-1}$, varying α between -0.75 and 0.75 in 0.25 increments. Here, a high positive coupling value with positive sign means making the interacting positions of two particles closer when they are moving towards each other in a head to head fashion. For this reason, when $\alpha = 0.75$, integration step is decreased to $2.5 \times 10^{-4} \text{ s}$ to avoid instabilities due to the LJ nonlinearity.

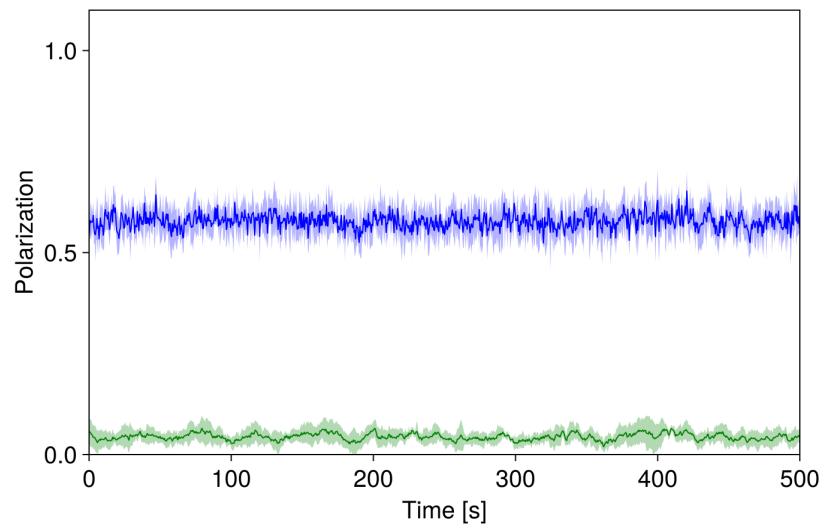
Results relative to zero coupling show how clustering and polarization are related in this case: not only the absence of an alignment interaction hinders



(a)

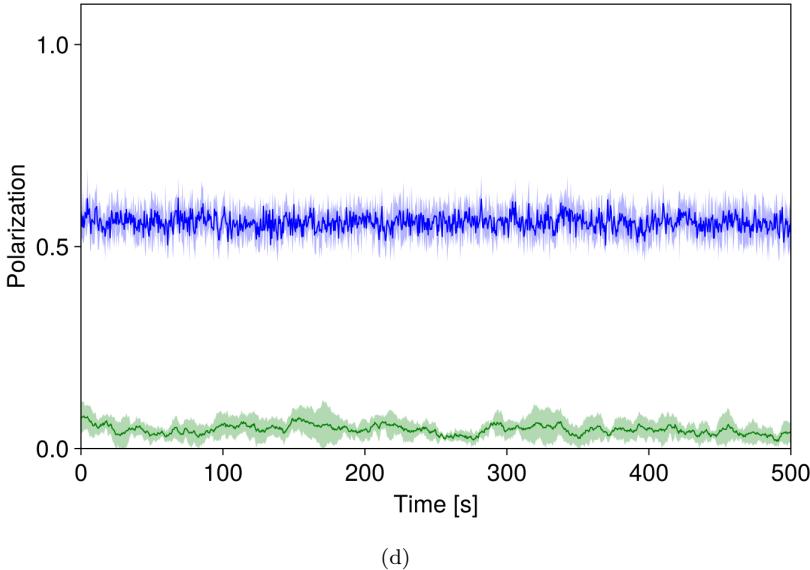


(b)



(c)

Figure 3.8



(d)

Figure 3.8: Local (blue) and global (green) polarization corresponding to values of velocity and α : (a) $v = 10 \mu\text{m s}^{-1} \alpha = 0.5$, (b) $v = 20 \mu\text{m s}^{-1} \alpha = 0.5$, (c) $v = 10 \mu\text{m s}^{-1} \alpha = -0.5$, (d) $v = 20 \mu\text{m s}^{-1} \alpha = -0.5$.

system's polarization as expected, but at this level, large scale clustering does not realize although an attractive force is inserted in simulations.

With $\alpha = 0.25$ system tends to polarize, both locally and globally, but still it does not cluster as a whole. Some peaks at short distance start to rise in the $g(r)$ (Figure 3.19), showing the onset of short range order, though they are not in the typical positions of a hexagonal close packed lattice.

At $\alpha = 0.5$ polarization is almost total and all particles in the system tend to be part of the same cluster. Radial distribution function shows the onset of long range order, with high and narrow peaks, both at short and long distance.

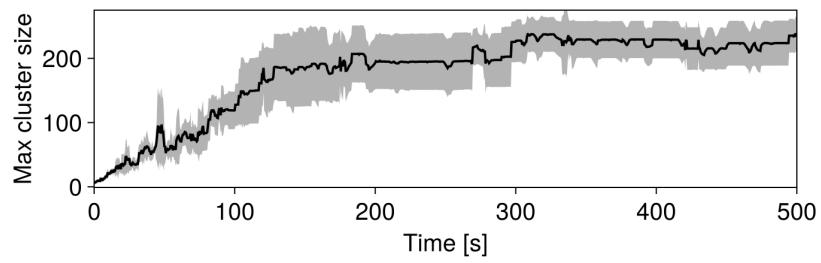
Polarization is total at $\alpha = 0.75$, and once steady state is reached, both local and global alignment present very small oscillations. Ensemble is fully clustered and flocking at this point.

Analyzing the transient time, it is possible to notice how increasing the coupling parameter between orientation angles tends to reduce the transient time, making it faster for the system to align (Figure 3.15).

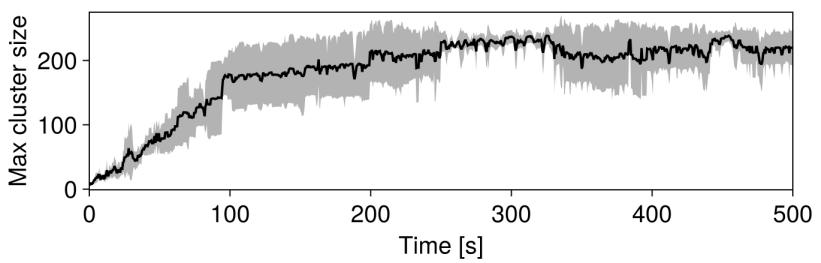
All of the three simulations with negative α show no alignment and no clustering, in accordance with what stated in section 3.3.1.

3.3.4 Angular Velocity

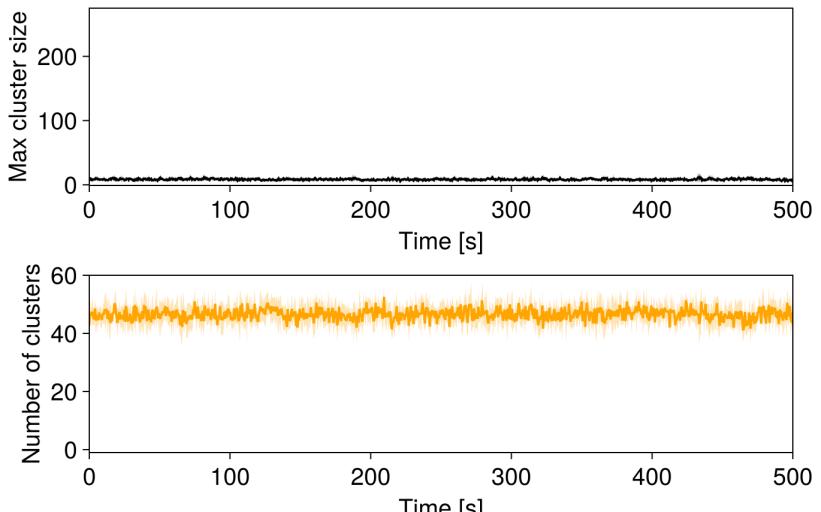
Simulations relative to this section are performed with a Lennard Jones strength parameter of $\epsilon = 0.25 \text{ pJ}$, off centered with $\alpha = 0.5$. The influence of rotational self propulsion was studied generating particles' angular velocity from a normal distribution. To avoid inserting a bias in the system the mean is kept fixed at 0 rad s^{-1} , while changing the standard deviation in 0.5 rad s^{-1} , 1 rad s^{-1} and 2 rad s^{-1} . This simulations tend to be less stable than ones without angular



(a)

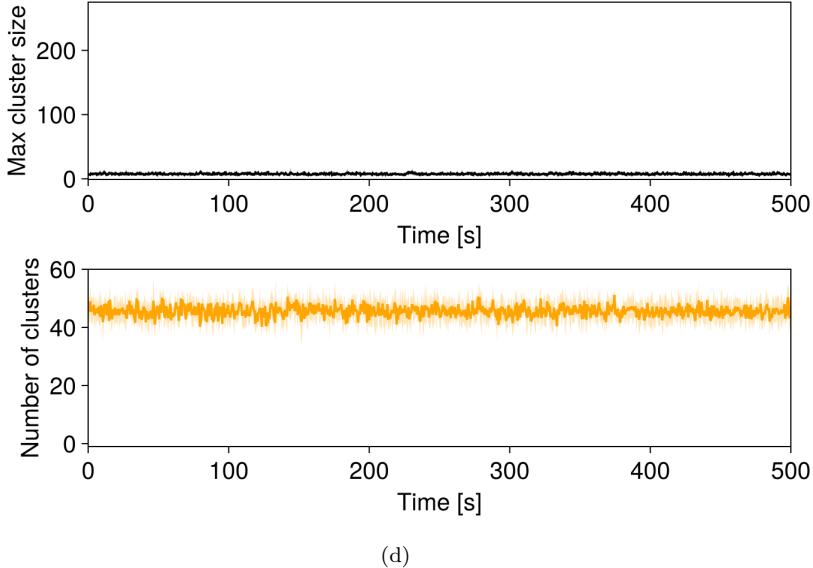


(b)



(c)

Figure 3.9



(d)

Figure 3.9: Maximum cluster size and number of clusters corresponding to values of velocity and α : (a) $v = 10 \mu\text{m s}^{-1} \alpha = 0.5$, (b) $v = 20 \mu\text{m s}^{-1} \alpha = 0.5$, (c) $v = 10 \mu\text{m s}^{-1} \alpha = -0.5$, (d) $v = 20 \mu\text{m s}^{-1} \alpha = -0.5$.

velocity, thus we needed to decrease the integration step to 2.5×10^{-3} s.

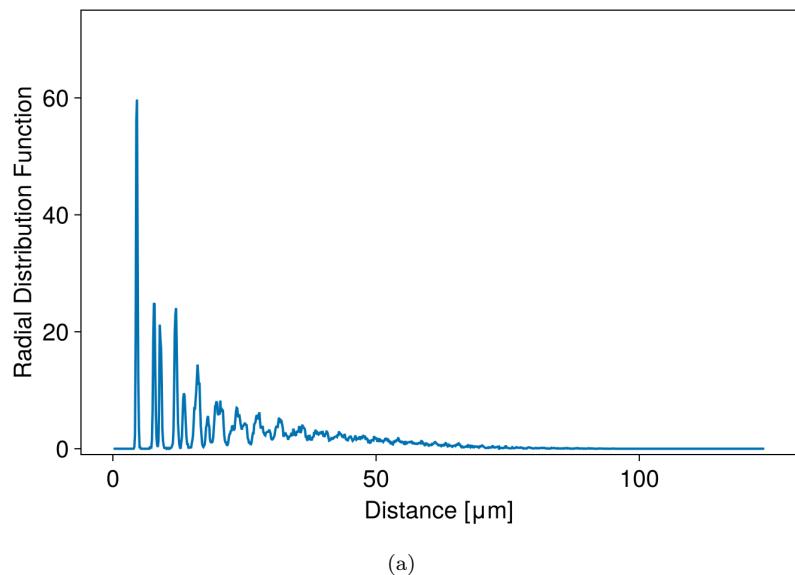
With a 0.5 rad s^{-1} standard deviation, global polarization is strongly hindered, with large oscillations, though still showing an increasing trend. Still, local polarization remains above baseline. Cluster size presents an increasing trend as well, and probably a longer simulation is needed to reach steady state. Some peaks are still present in radial distribution function, a sign that some order is preserved in the system, at least at short and medium range.

A standard deviation of 1 rad s^{-1} destroys global orientational order almost completely, while local polarization is just above the initial value. Situation plot in panel (b) of Figure 3.20 shows a system with relatively small oriented clusters; as a consequence, $g(r)$ peaks are lower than before and in a much smaller number. Analyzing clusters' dynamics in simulation videos, we can state that the aligning interaction, along with a nonzero angular velocity, results in small polarized clusters where particles rotate in phase around their axis.

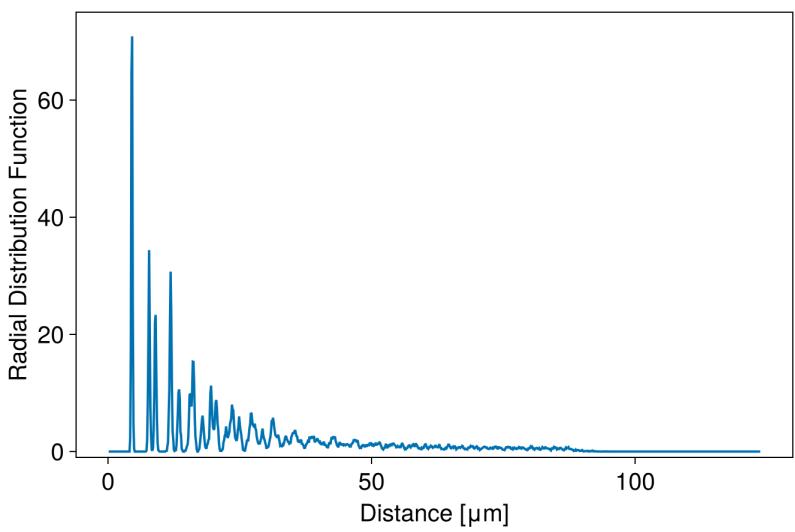
Further broadening the angular velocity distribution makes it impossible for the ensemble to develop long range order, as $g(r)$ shows. Polarization is completely hindered as well, both locally and globally, as graphs and situation plots show (Figures 3.21 and 3.20).

3.4 Conclusions to chapter 3

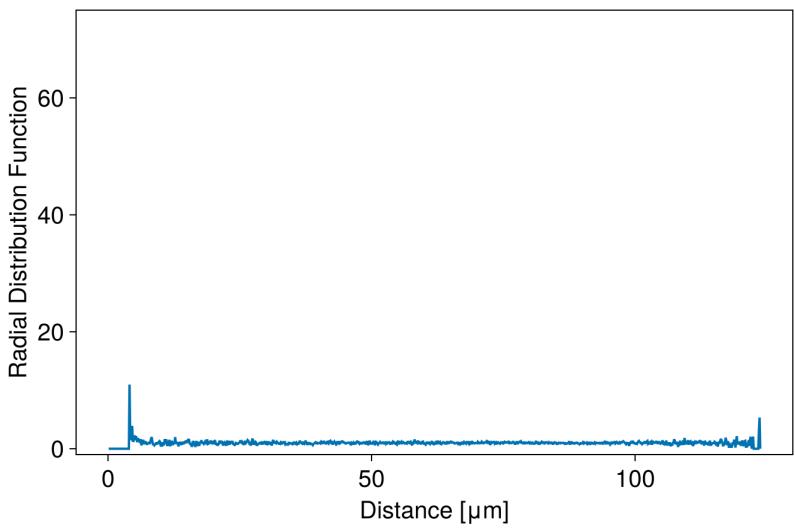
Along this chapter we studied how our model can be used in a theoretical investigation of both phase transitions and collective behaviors. We showed that, without an explicitly aligning torque, our system undergoes a flocking transition, similar to what happens with original or continuous Vicsek models



(a)



(b)



(c)

Figure 3.10

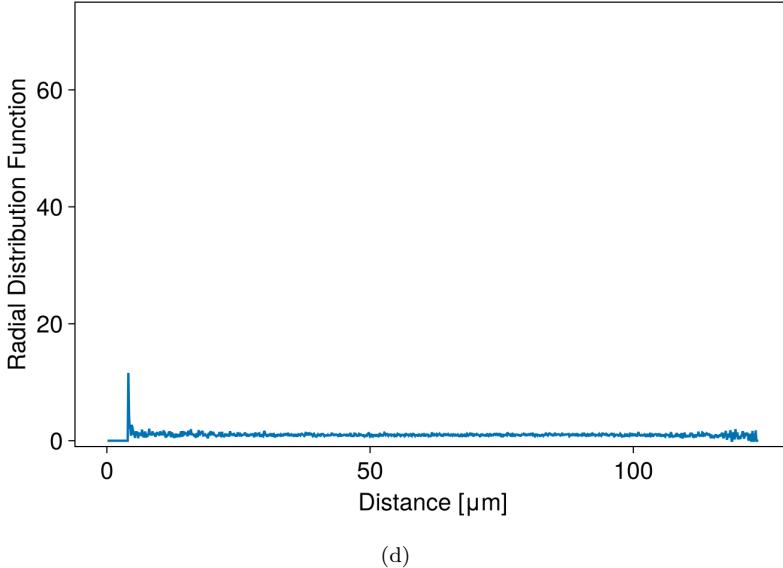


Figure 3.10: Radial distribution function at last instant of simulation corresponding to values of velocity and α : (a) $v = 10 \mu\text{m s}^{-1} \alpha = 0.5$, (b) $v = 20 \mu\text{m s}^{-1} \alpha = 0.5$, (c) $v = 10 \mu\text{m s}^{-1} \alpha = -0.5$, (d) $v = 20 \mu\text{m s}^{-1} \alpha = -0.5$.

that have been built for that purpose. Such a phenomenon can be efficiently studied as a continuous phase transition, where the order parameter shows the expected behavior. The similarities between this model and the one studied in [14] suggests that these aligning particles systems could be part of the same universality class, but more investigation is needed in those regards.

Regarding the effect of simulation features on collective behaviors we showed that, in general, adding more complexity with some angular or linear velocity distribution tends to disrupt long range order in positions, while the presence of larger velocities (both in the case of one velocity and distributions) has the tendency to align the system, with faster particles spreading information across the board. In general, positional and orientational order are related when the same velocity is applied to all particles and a clustered system is a flocking system in most cases, while we showed how different velocities can still result in an aligned, although not clustered, system.

All the macroscopic behaviors, such as clustering and flocking, showed here are not observed during experiments in the host lab. This could be due to weak or short-range interactions, as well as more complex effect involving the solvent.

We can conclude that the suite of tools we developed offer a fast and efficient way to study at least some of the collective behaviors featured in a systems like the one we are interested in. In principle, adapting these tools to analyze experimental videos of tracked particles will not be hard due to their implementation and will possibly lead to a better study of the quantitative correspondence between simulated and experimental active particles, to deepen what was started with the qualitative agreement in 2.5. Nonetheless, in the experiments like the ones held in Microscale Robotics Lab, it is hard to work with high packing fraction,

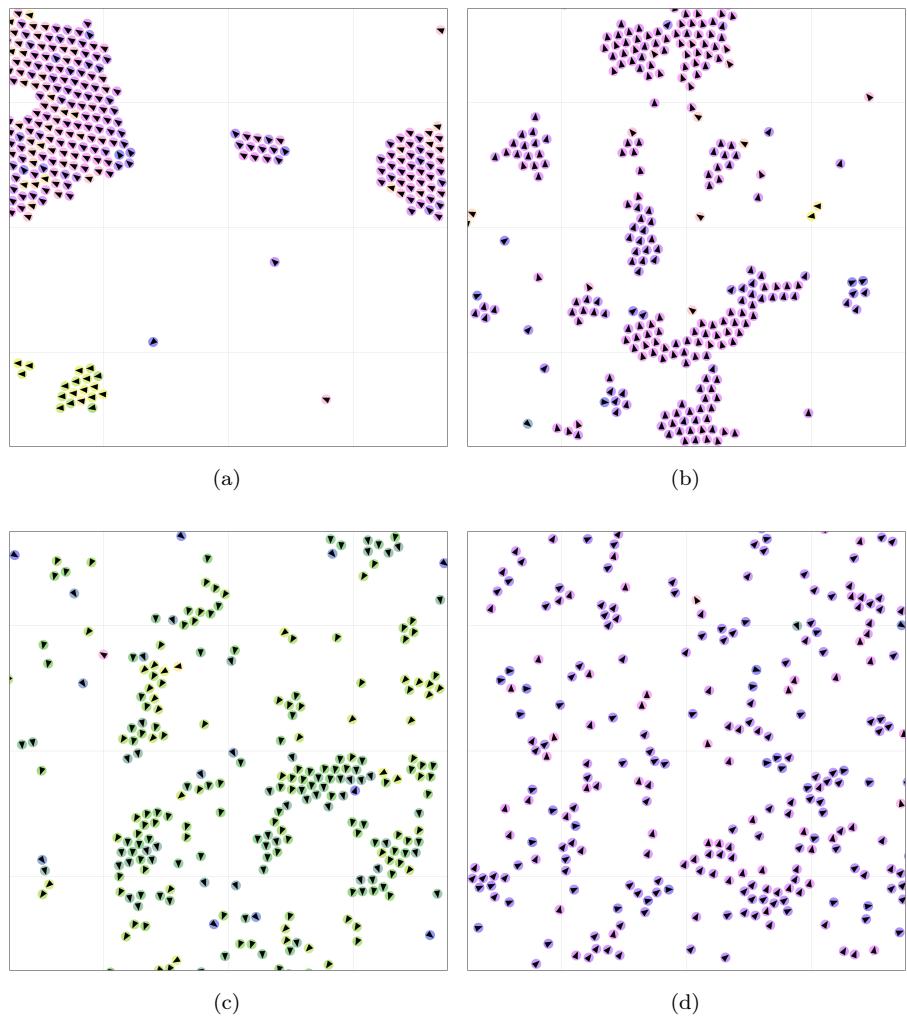


Figure 3.11: Representative simulation snapshots for a normal velocity distribution with $\mu = 10 \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 1 \mu\text{m s}^{-1}$, (b) $\sigma = 2 \mu\text{m s}^{-1}$, (c) $\sigma = 5 \mu\text{m s}^{-1}$, (d) $\sigma = 10 \mu\text{m s}^{-1}$

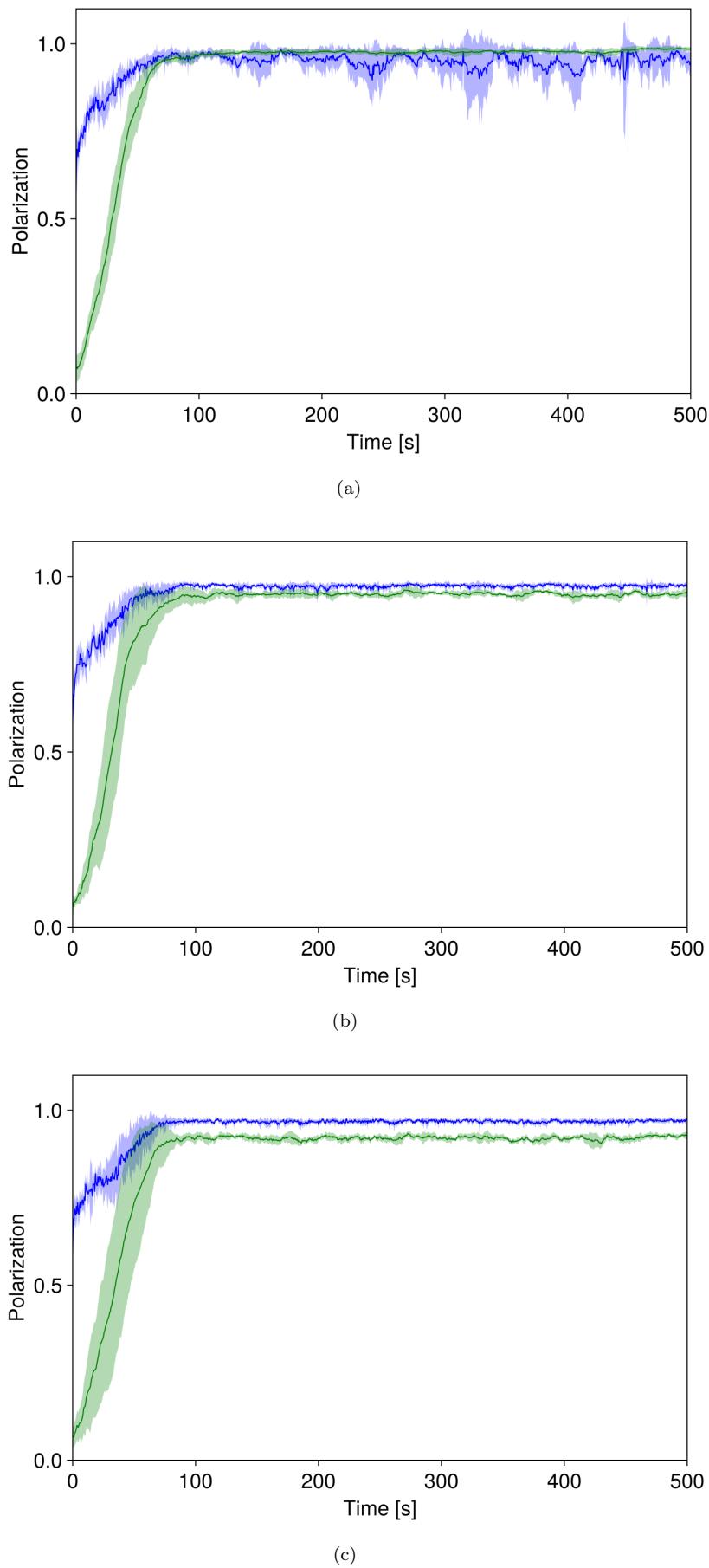


Figure 3.12

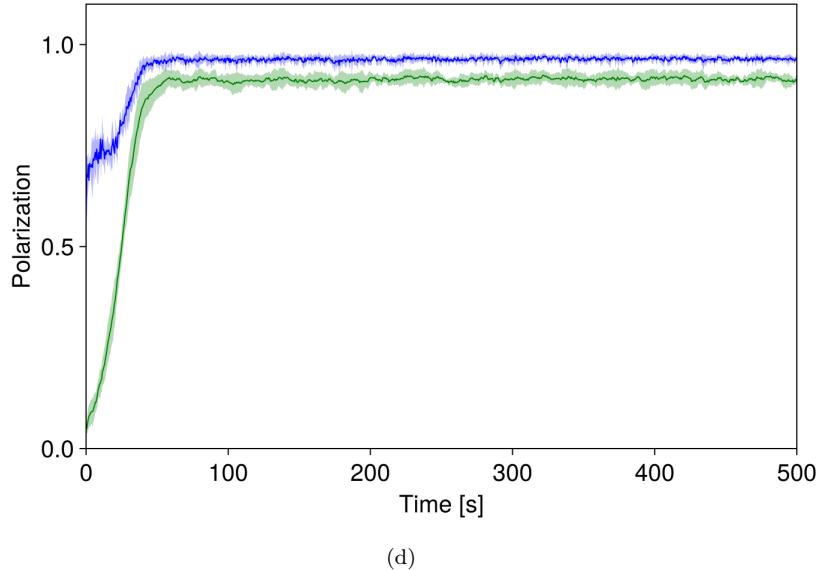


Figure 3.12: Local (blue) and global (green) polarization for a normal velocity distribution with $\mu = 10 \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 1 \mu\text{m s}^{-1}$, (b) $\sigma = 2 \mu\text{m s}^{-1}$, (c) $\sigma = 5 \mu\text{m s}^{-1}$, (d) $\sigma = 10 \mu\text{m s}^{-1}$

since the $2 \text{H}_2\text{O}_2 \xrightarrow{\text{Pt}} 2 \text{H}_2\text{O} + \text{O}_2$ reaction that propels particles tend to produce O_2 bubbles that disturb observation: higher packing fraction means more bubbles, making the system difficult to observe. While this thesis gets written, experimentation with Cu particles is ongoing, and the observations suggest that said particles produce few or no bubbles, making them a more suitable case study for high packing fraction situations. As snapshots in 2.5 show, measuring the orientation of observed particles is not a straightforward task, since the boundary between the two hemispheres is not always clearly observable. This will make harder to apply the polarization measurements to experimental videos. The fact that observed system show only local clustering and no flocking make us predict that the most useful tools among the ones developed in this work will be the local-focused ones, especially the cluster size measurement, which is the immediately applicable one for experimental situations.

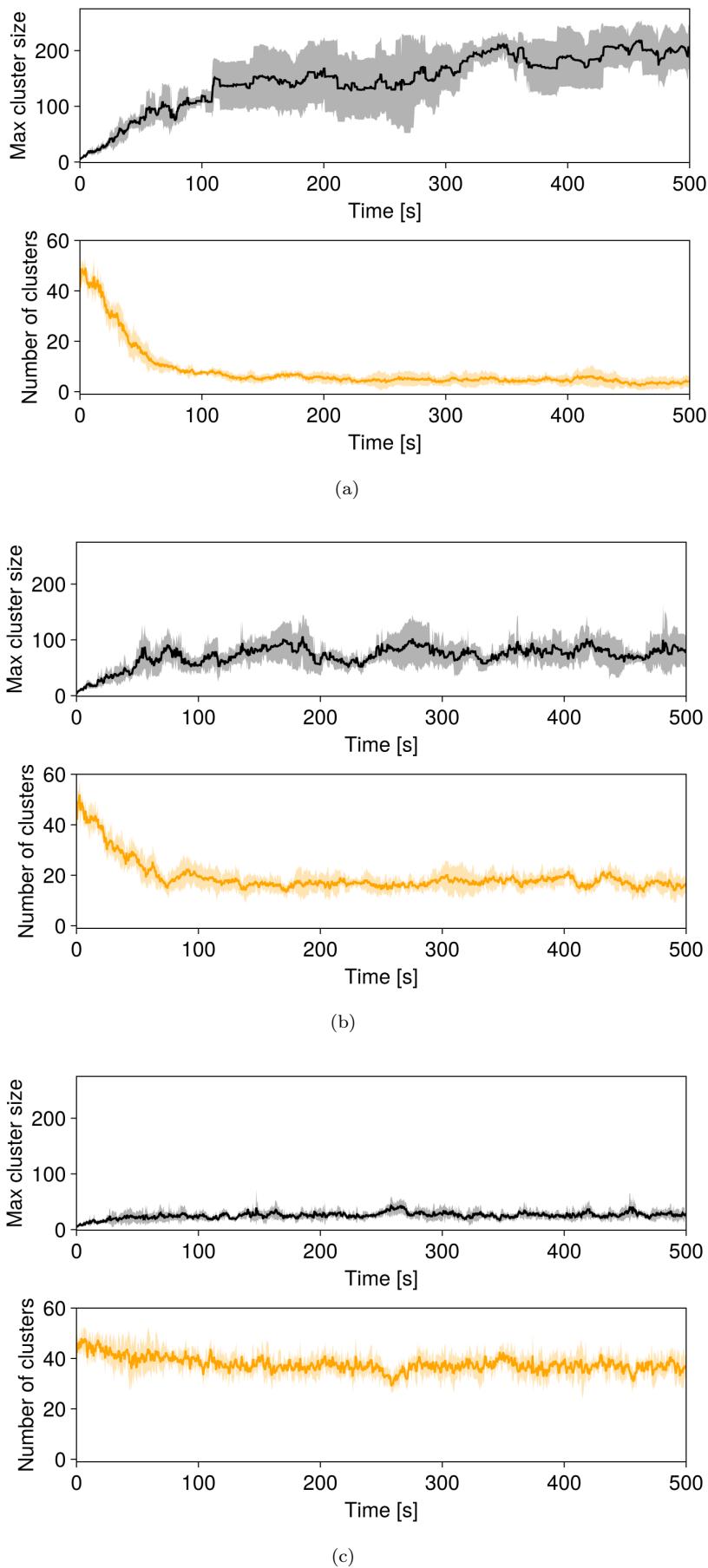
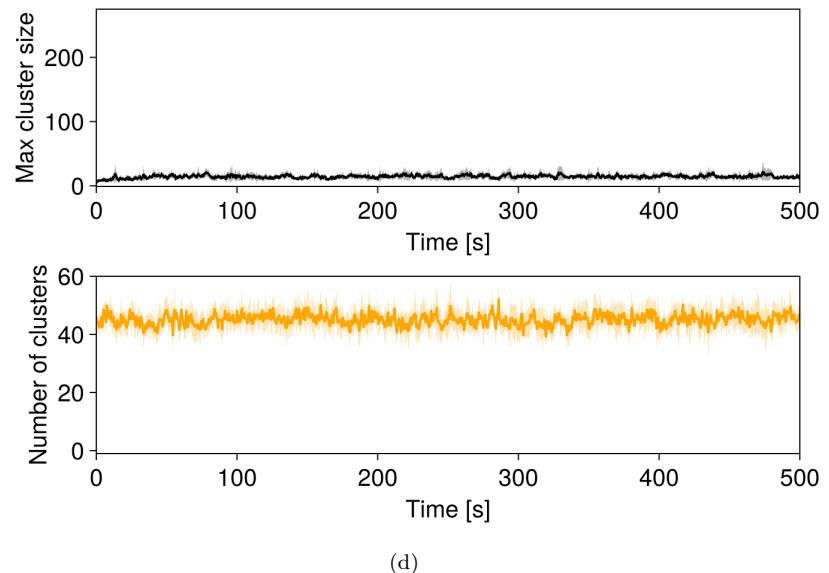


Figure 3.13



(d)

Figure 3.13: Maximum cluster size and number of clusters for a normal velocity distribution with $\mu = 10 \text{ } \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 1 \text{ } \mu\text{m s}^{-1}$, (b) $\sigma = 2 \text{ } \mu\text{m s}^{-1}$, (c) $\sigma = 5 \text{ } \mu\text{m s}^{-1}$, (d) $\sigma = 10 \text{ } \mu\text{m s}^{-1}$

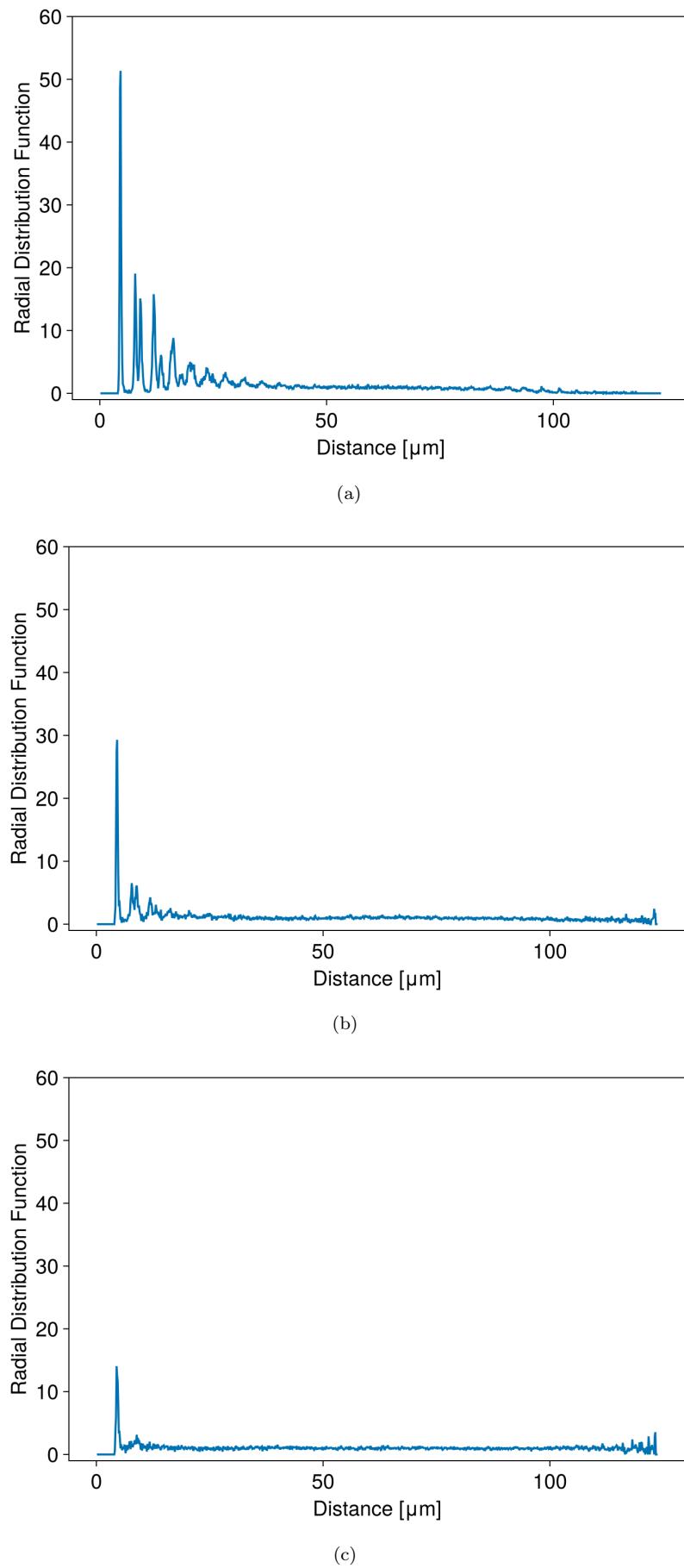
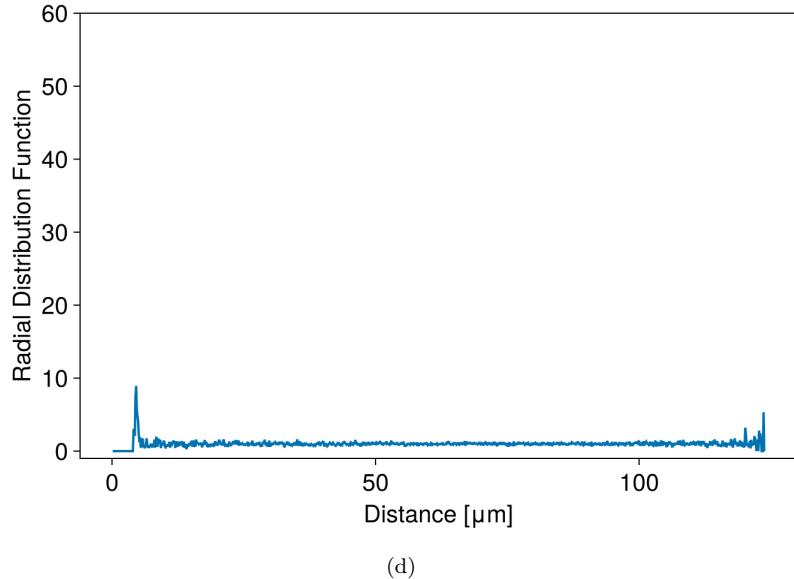


Figure 3.14



(d)

Figure 3.14: Radial distribution function at last instant of simulation for a normal velocity distribution with $\mu = 10 \text{ } \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 1 \text{ } \mu\text{m s}^{-1}$, (b) $\sigma = 2 \text{ } \mu\text{m s}^{-1}$, (c) $\sigma = 5 \text{ } \mu\text{m s}^{-1}$, (d) $\sigma = 10 \text{ } \mu\text{m s}^{-1}$.

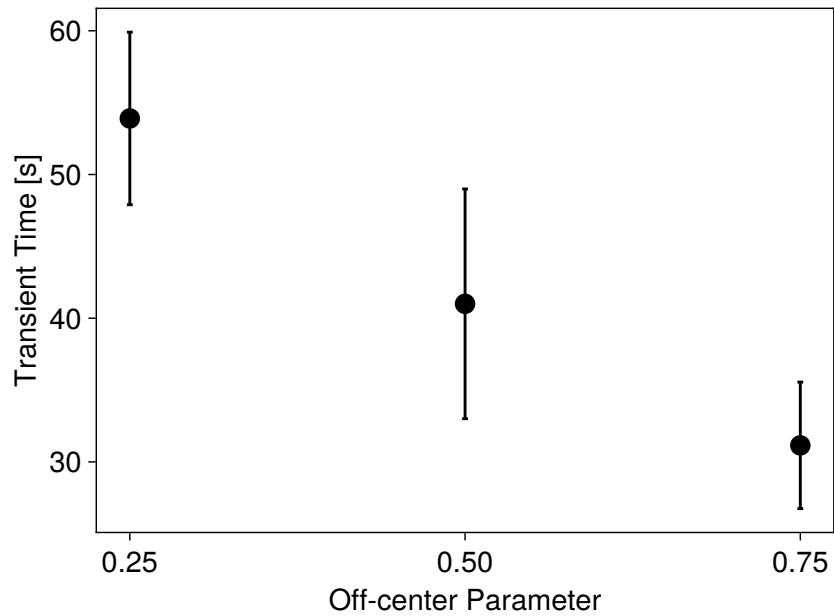


Figure 3.15: Transient time (time needed to reach 95% of maximum) for global polarization as a function of α . Only positive values of off center coupling (where the system polarizes) are showed here.

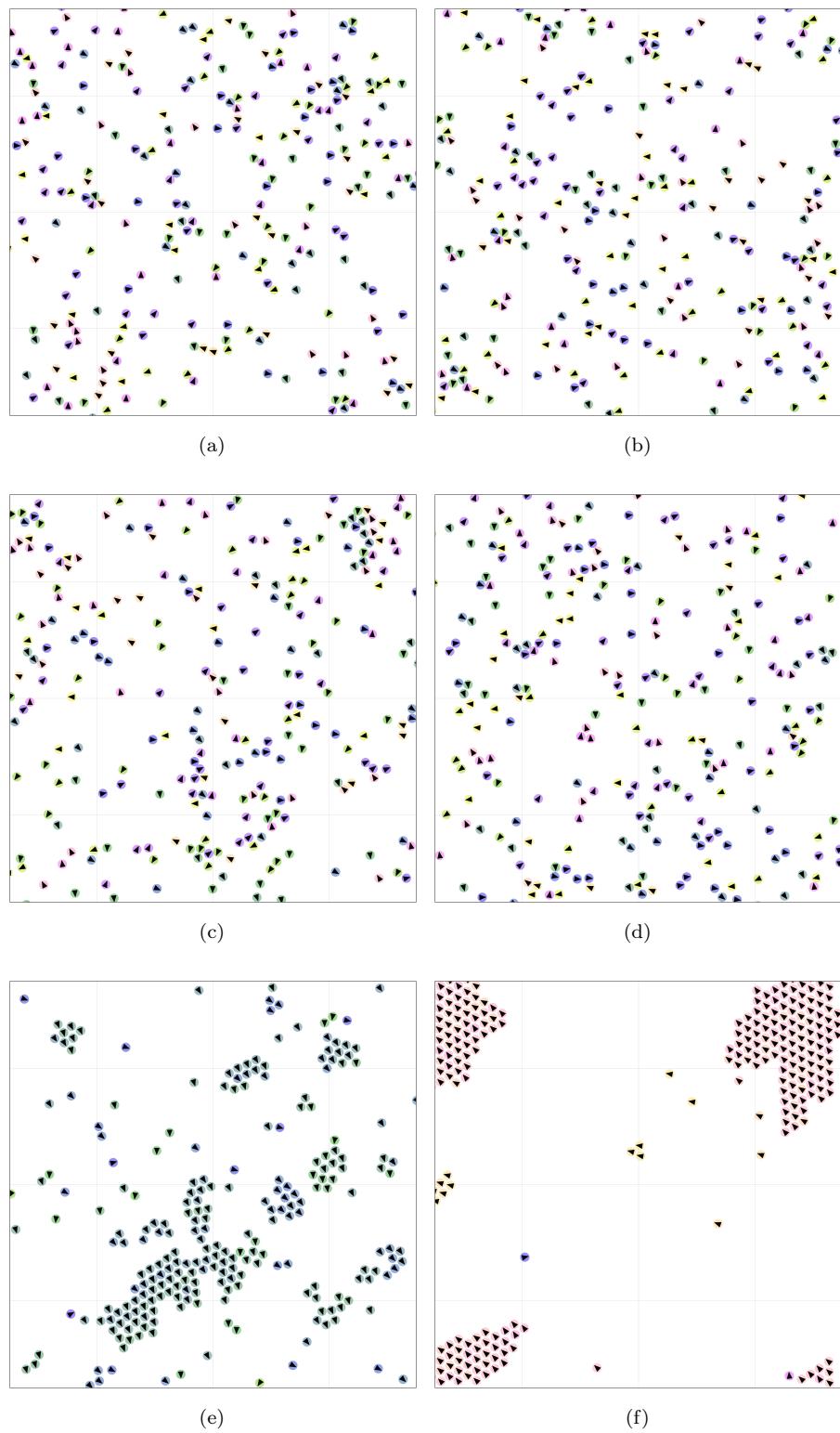


Figure 3.16

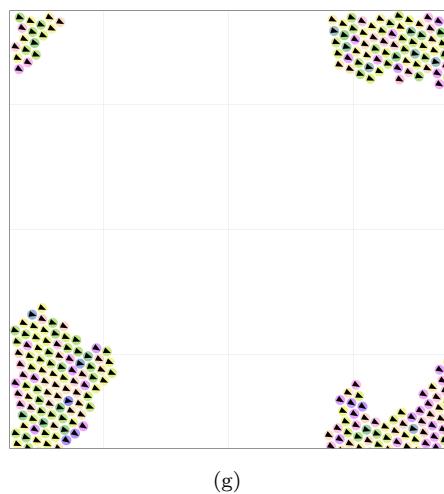


Figure 3.16: Representative simulation snapshots corresponding to values of off center parameter α : (a) $\alpha = -0.75$, (b) $\alpha = -0.5$, (c) $\alpha = -0.25$, (d) $\alpha = 0$, (e) $\alpha = 0.25$, (f) $\alpha = 0.5$, (g) $\alpha = 0.75$

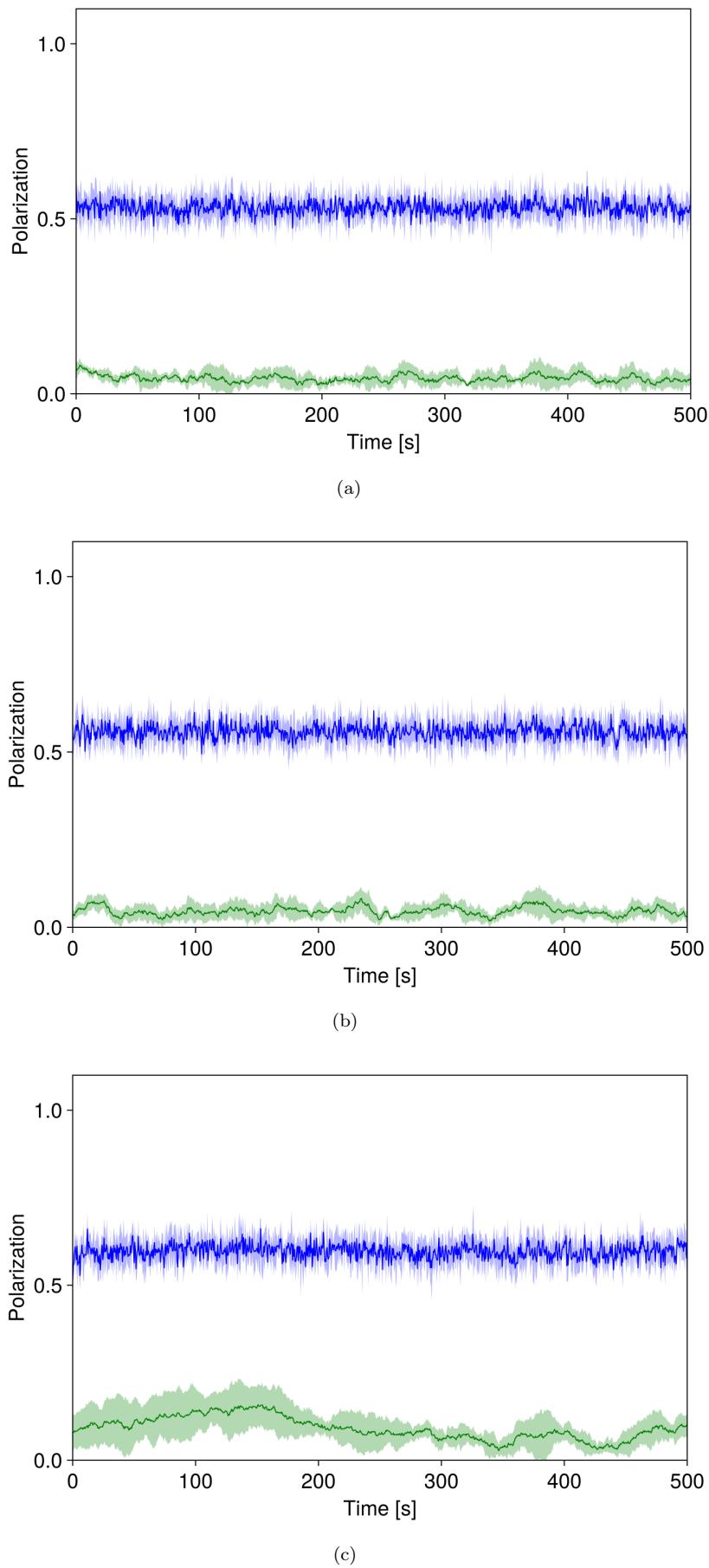


Figure 3.17

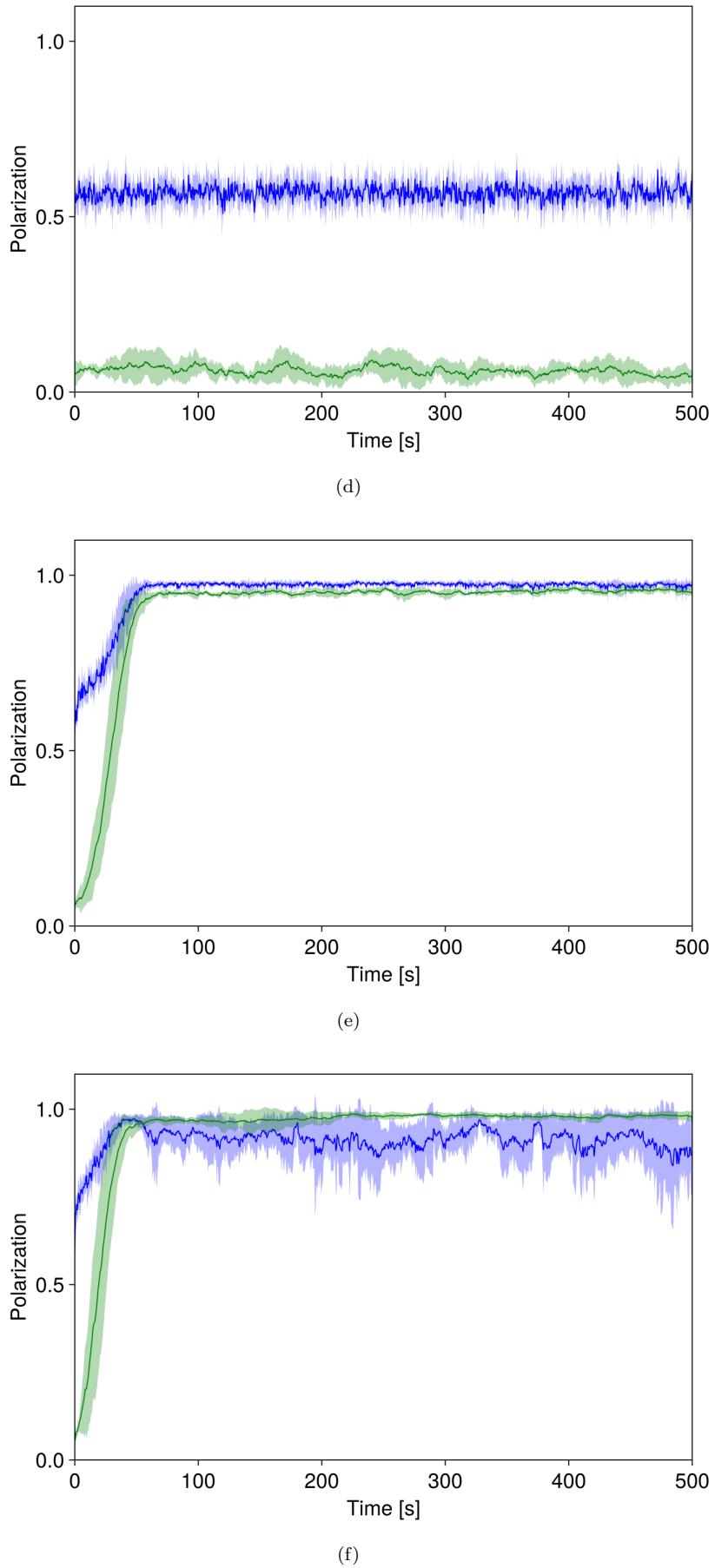


Figure 3.17

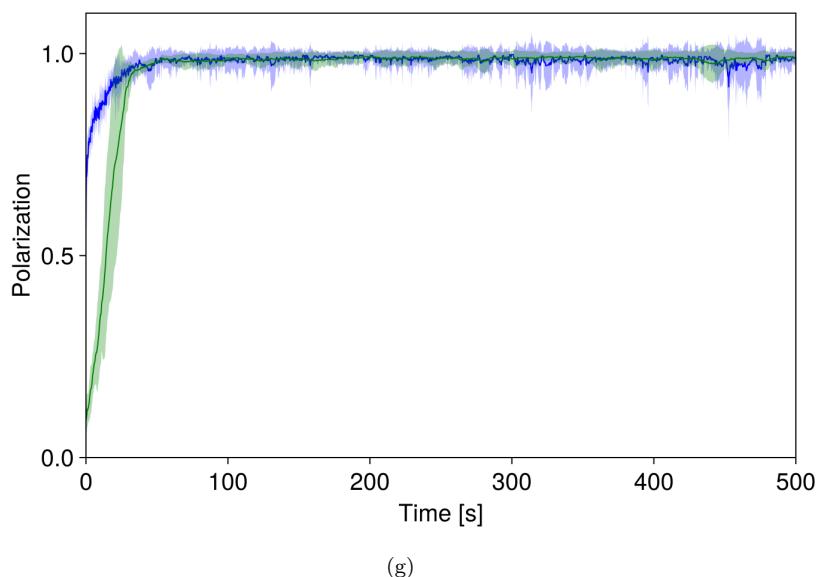
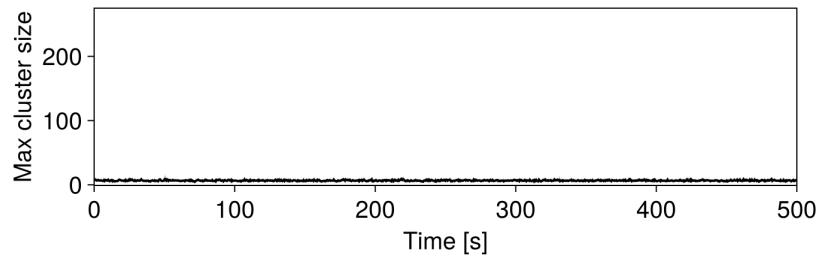
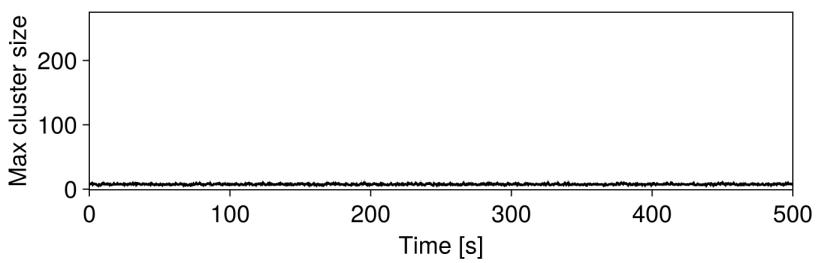


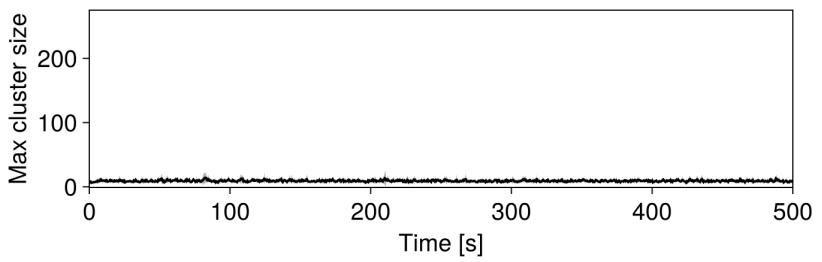
Figure 3.17: Local (blue) and global (green) polarization for corresponding to values of off center parameter α : (a) $\alpha = -0.75$, (b) $\alpha = -0.5$, (c) $\alpha = -0.25$, (d) $\alpha = 0$, (e) $\alpha = 0.25$, (f) $\alpha = 0.5$, (g) $\alpha = 0.75$



(a)



(b)



(c)

Figure 3.18

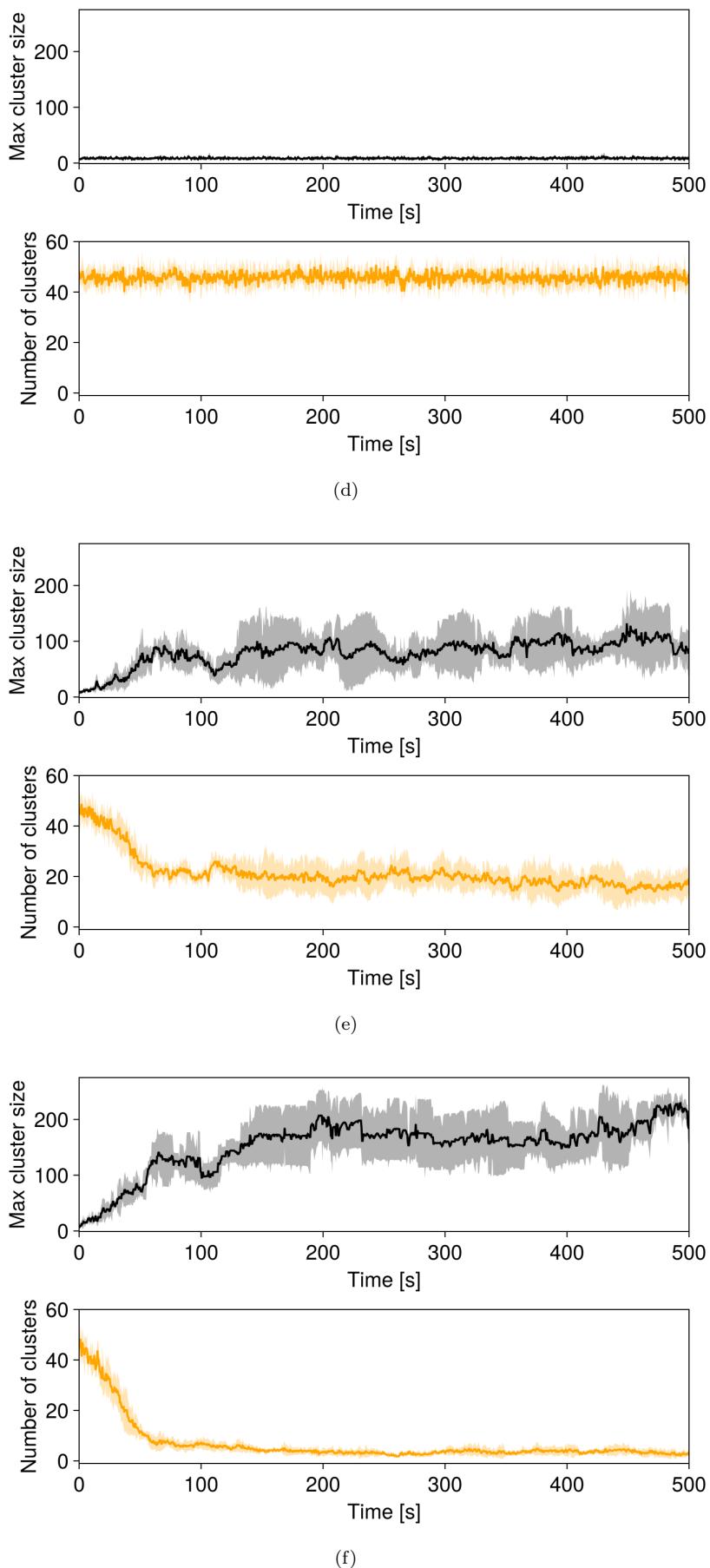
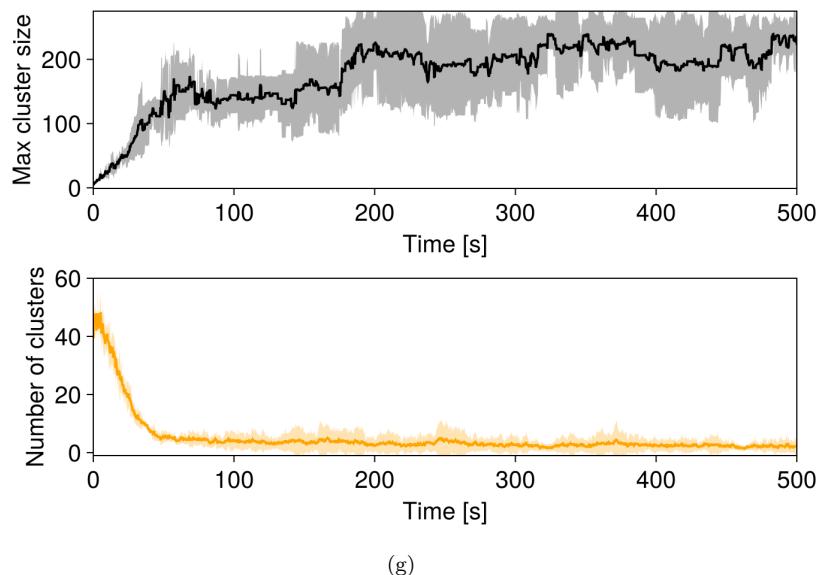


Figure 3.18



(g)

Figure 3.18: Maximum cluster size and cluster number corresponding to values of off center parameter α : (a) $\alpha = -0.75$, (b) $\alpha = -0.5$, (c) $\alpha = -0.25$, (d) $\alpha = 0$, (e) $\alpha = 0.25$, (f) $\alpha = 0.5$, (g) $\alpha = 0.75$

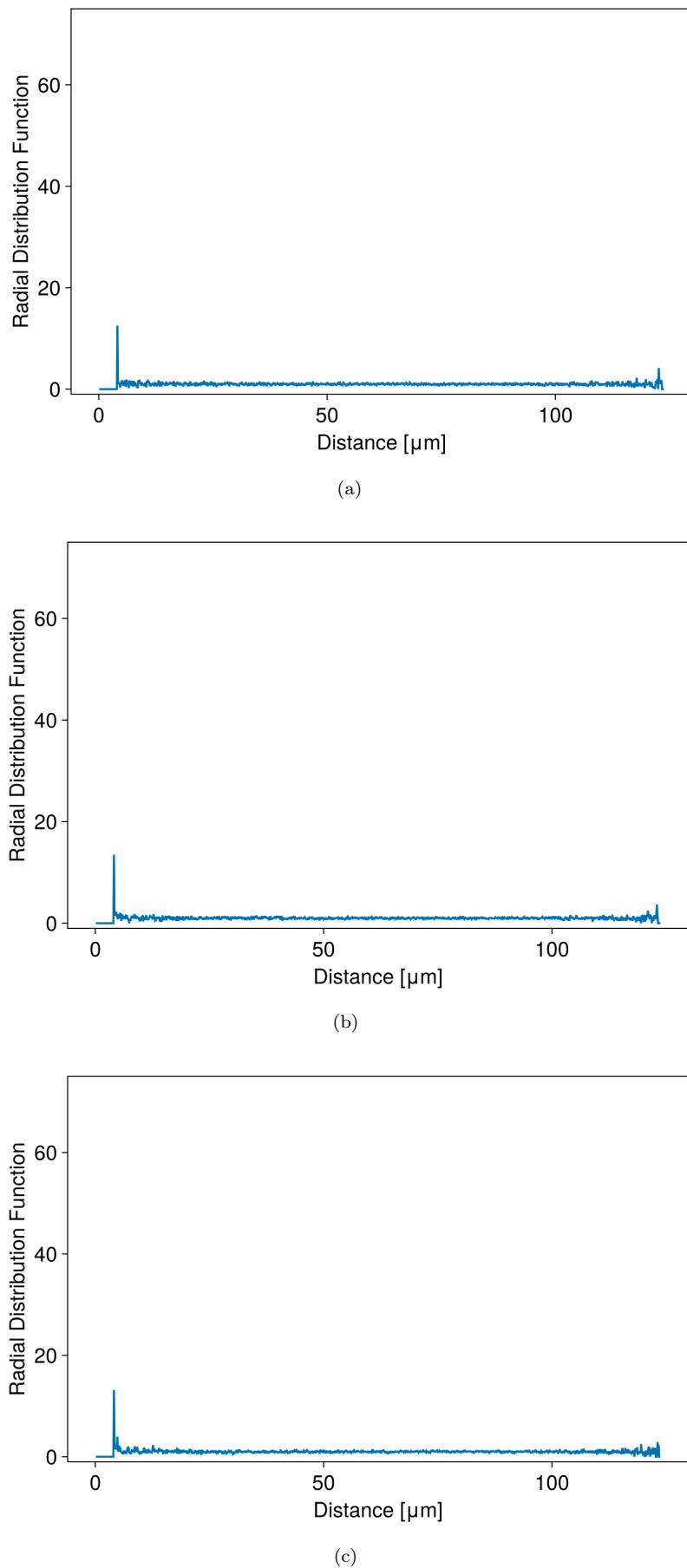
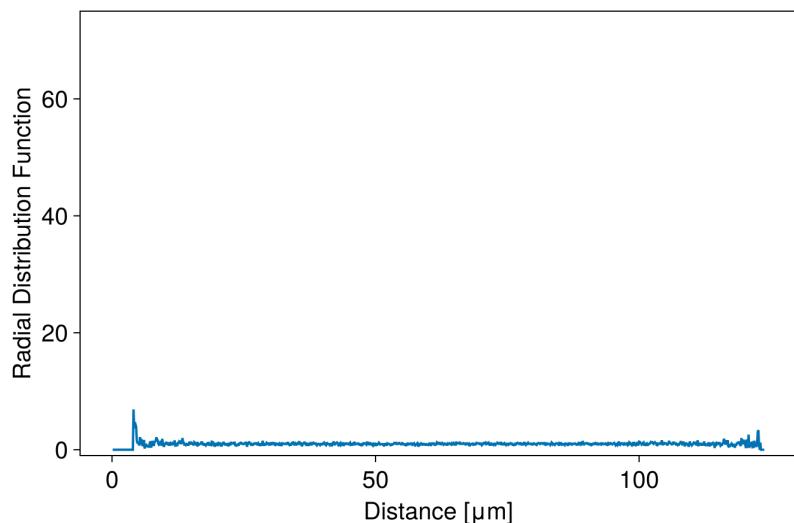
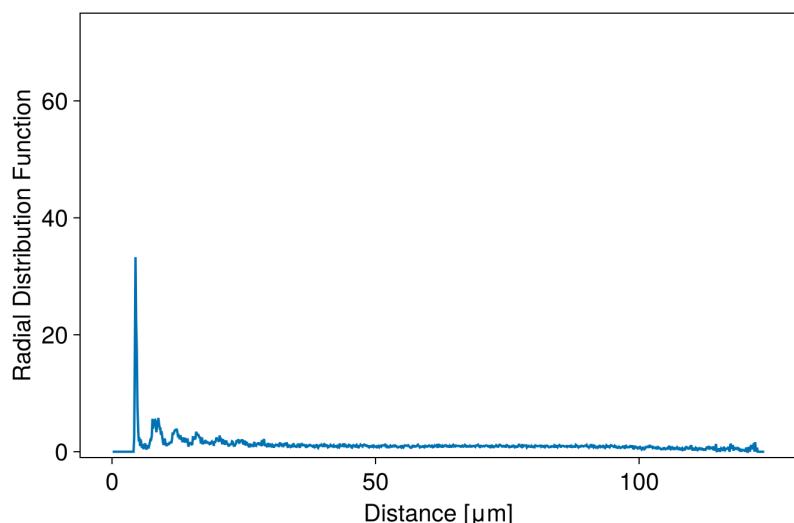


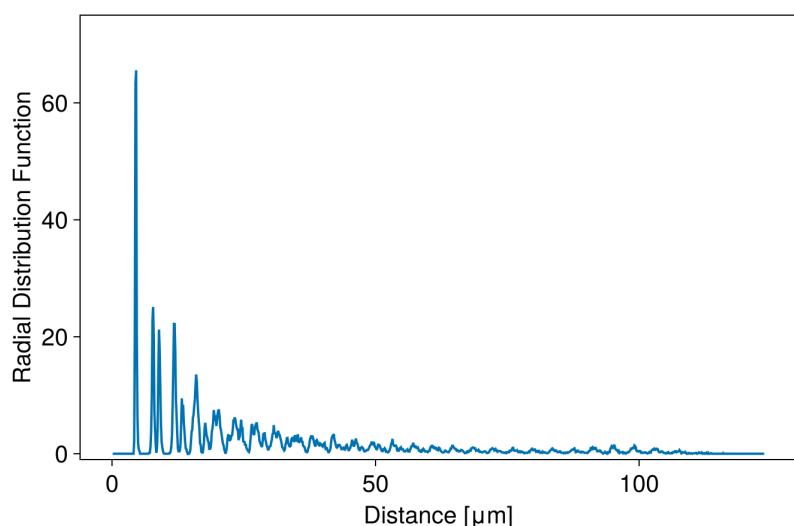
Figure 3.19



(d)



(e)



(f)

Figure 3.19

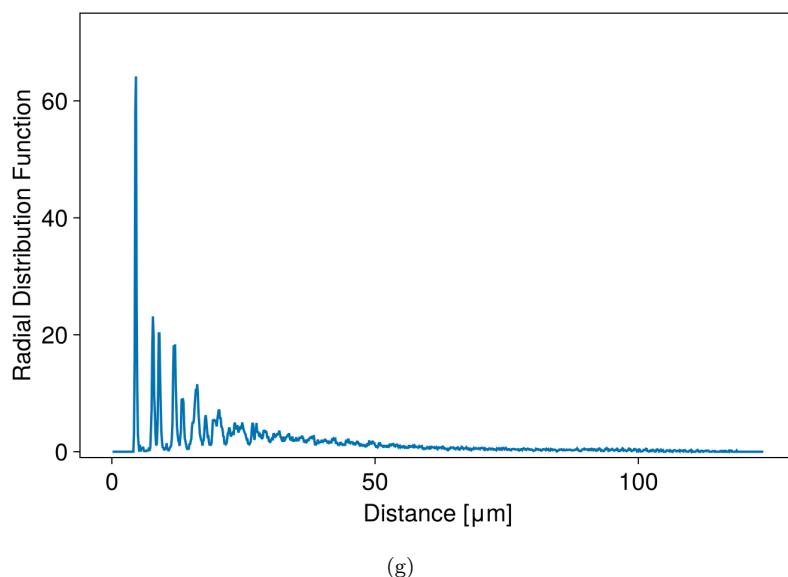


Figure 3.19: Radial distribution function at last instant of simulation corresponding to values of off center parameter α : (a) $\alpha = -0.75$, (b) $\alpha = -0.5$, (c) $\alpha = -0.25$, (d) $\alpha = 0$, (e) $\alpha = 0.25$, (f) $\alpha = 0.5$, (g) $\alpha = 0.75$

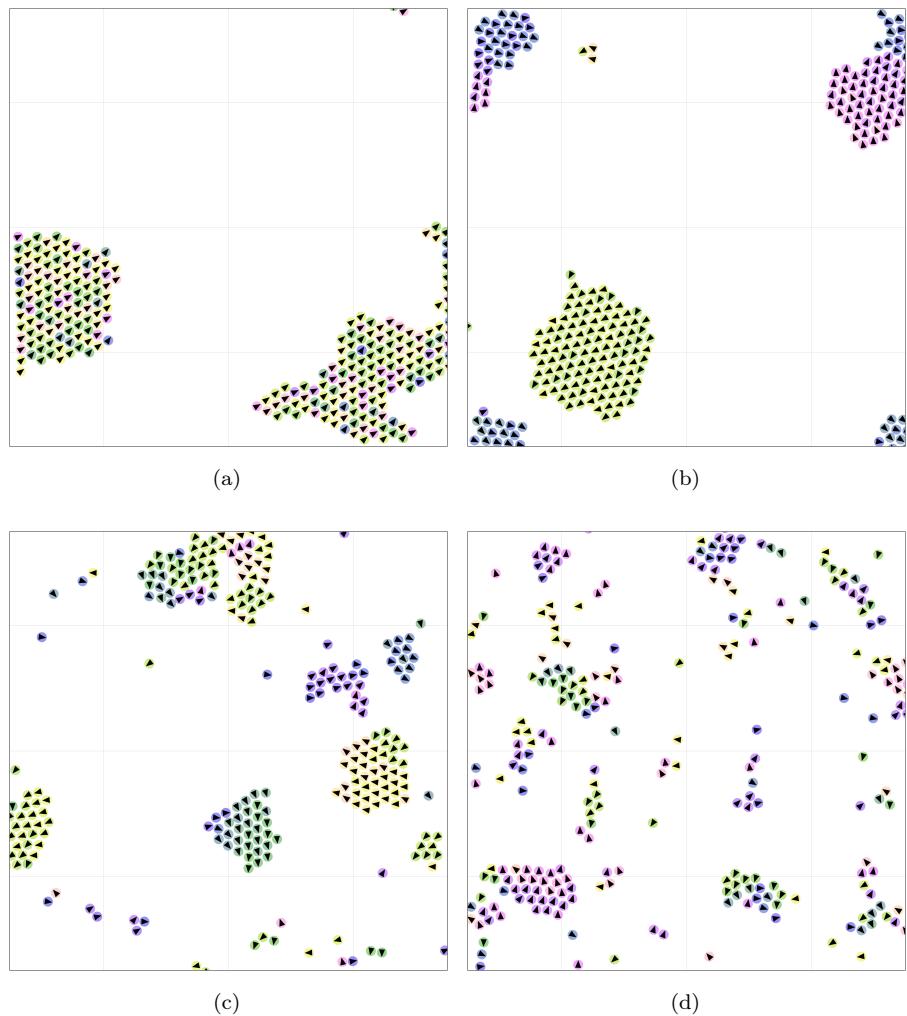


Figure 3.20: Representative simulation snapshots for a normal angular velocity distribution with $\mu = 0 \text{ } \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 0 \text{ } \mu\text{m s}^{-1}$, (b) $\sigma = 0.5 \text{ } \mu\text{m s}^{-1}$, (c) $\sigma = 1 \text{ } \mu\text{m s}^{-1}$, (d) $\sigma = 2 \text{ } \mu\text{m s}^{-1}$

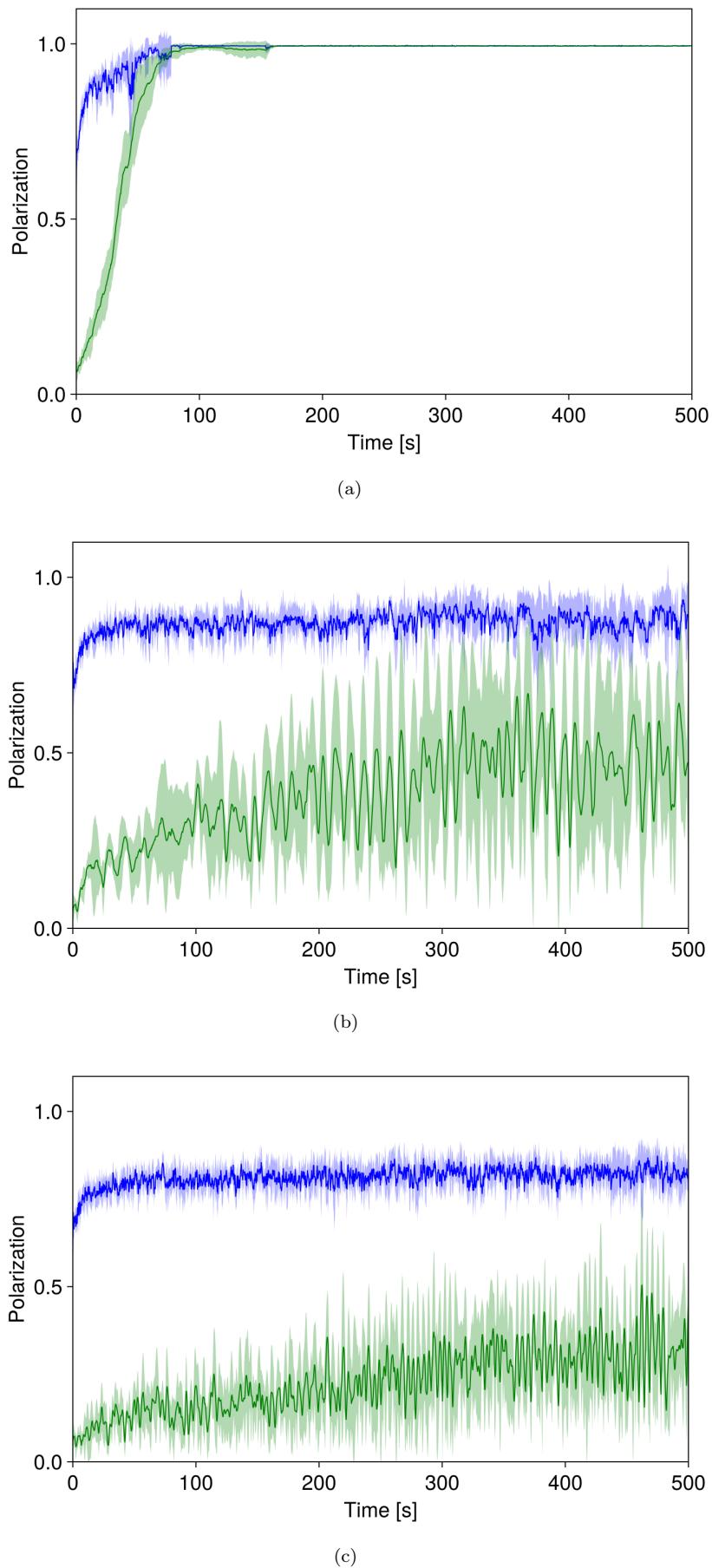


Figure 3.21

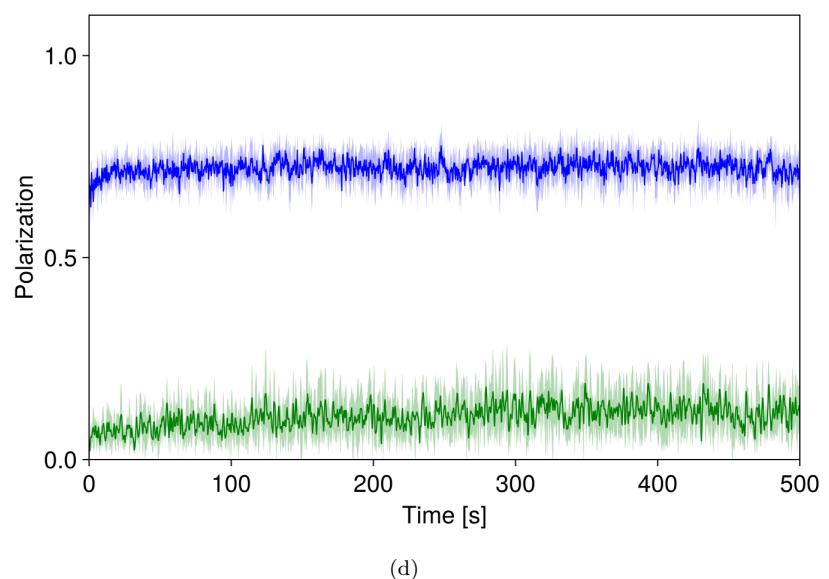


Figure 3.21: Local (blue) and global (green) polarization for a normal angular velocity distribution with $\mu = 0 \text{ } \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 0 \text{ } \mu\text{m s}^{-1}$, (b) $\sigma = 0.5 \text{ } \mu\text{m s}^{-1}$, (c) $\sigma = 1 \text{ } \mu\text{m s}^{-1}$, (d) $\sigma = 2 \text{ } \mu\text{m s}^{-1}$

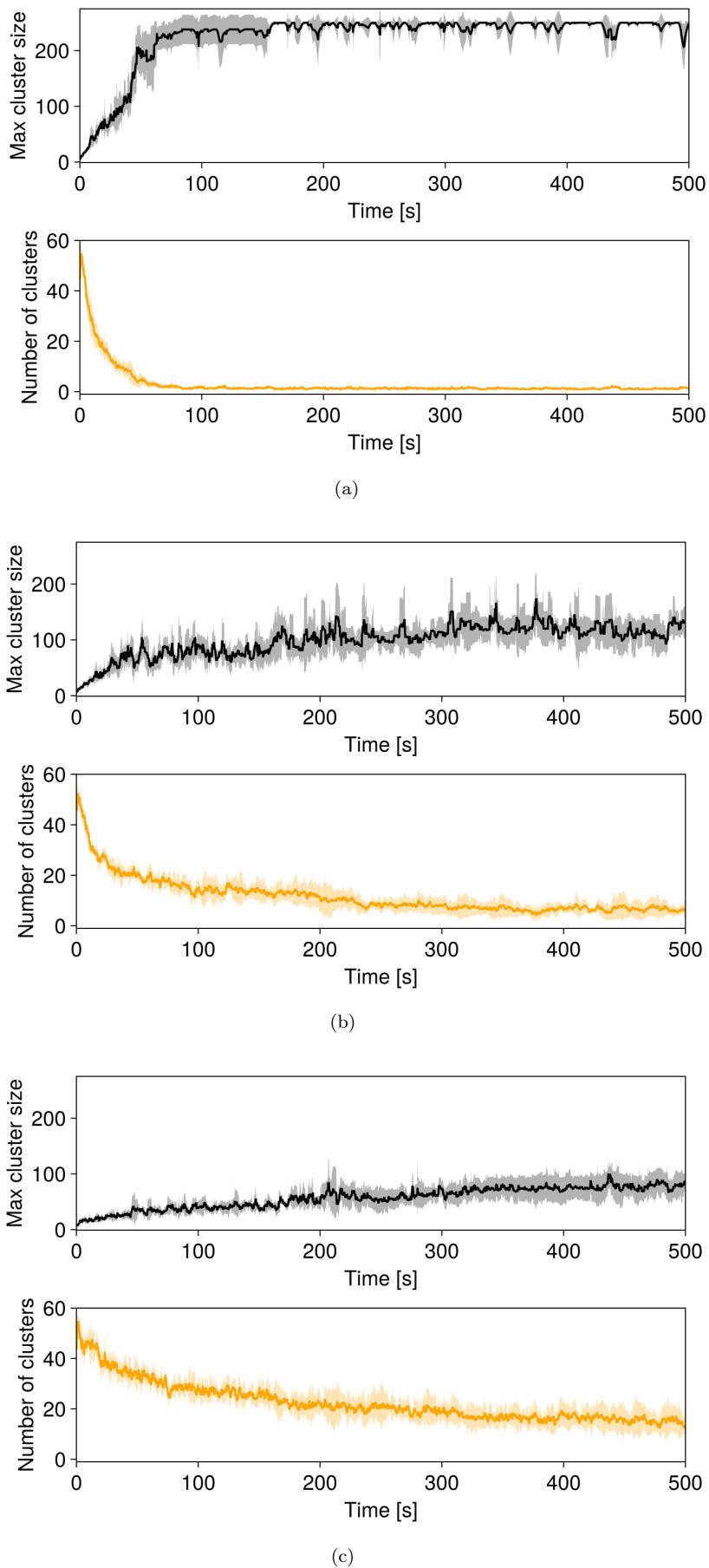


Figure 3.22

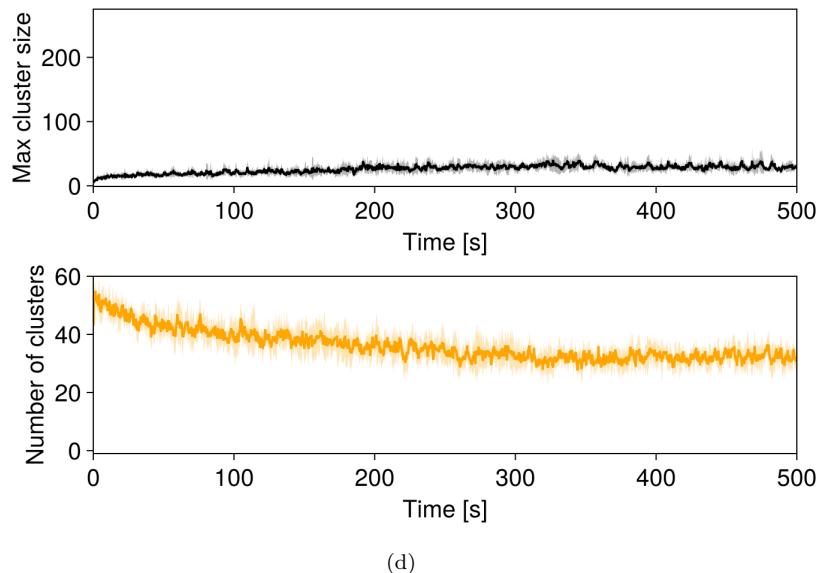
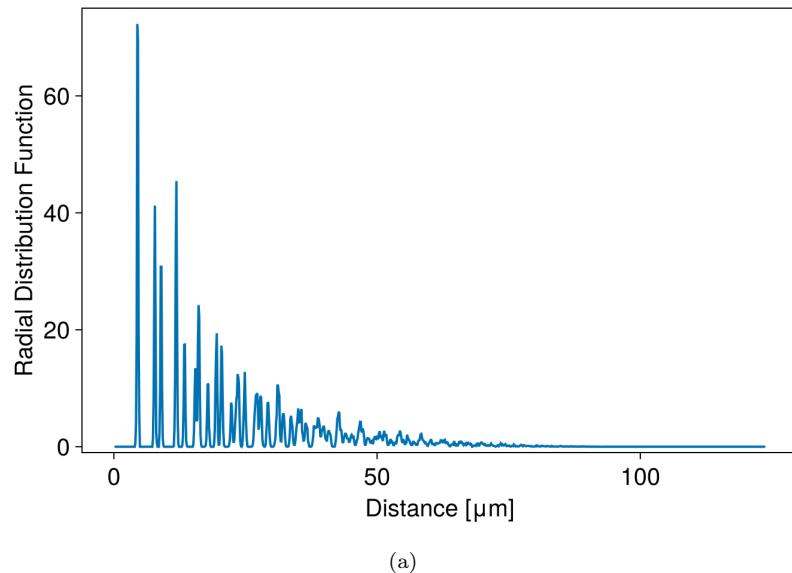
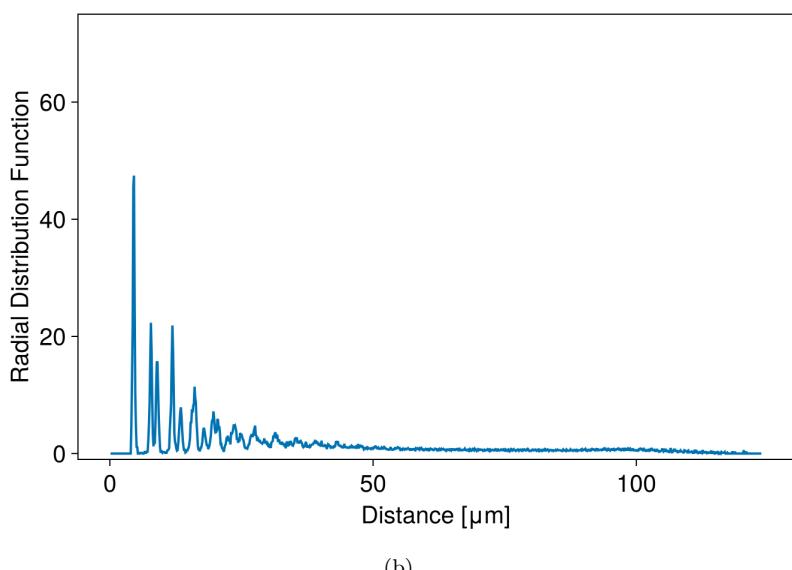


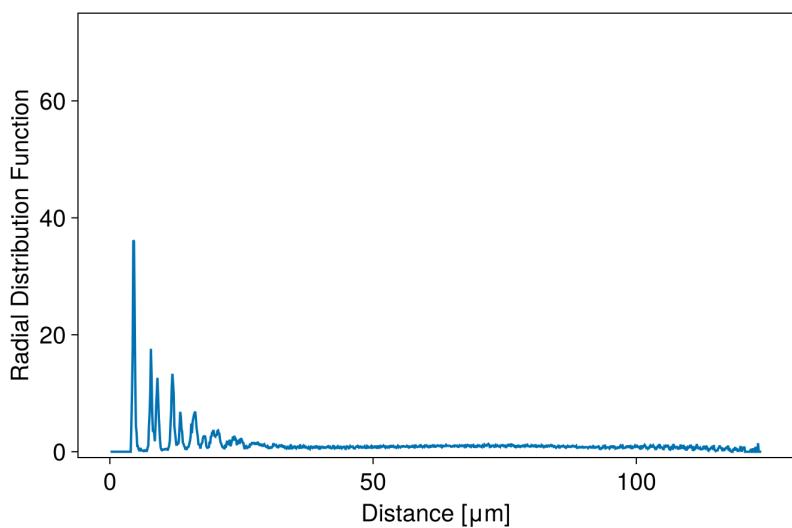
Figure 3.22: Maximum cluster size and cluster number for a normal angular velocity distribution with $\mu = 0 \text{ } \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 0 \text{ } \mu\text{m s}^{-1}$, (b) $\sigma = 0.5 \text{ } \mu\text{m s}^{-1}$, (c) $\sigma = 1 \text{ } \mu\text{m s}^{-1}$, (d) $\sigma = 2 \text{ } \mu\text{m s}^{-1}$



(a)

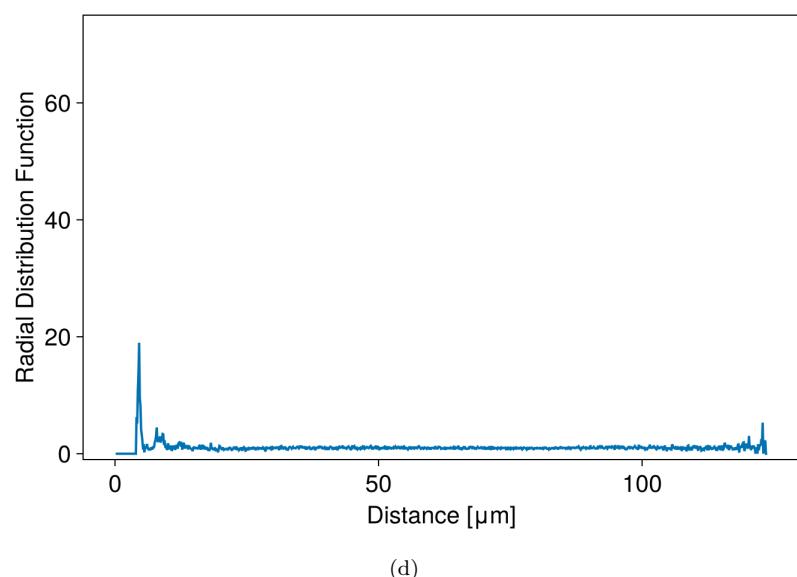


(b)



(c)

Figure 3.23



(d)

Figure 3.23: Radial distribution function at last instant of simulation for a normal angular velocity distribution with $\mu = 0 \text{ } \mu\text{m s}^{-1}$ and standard deviation: (a) $\sigma = 0 \text{ } \mu\text{m s}^{-1}$, (b) $\sigma = 0.5 \text{ } \mu\text{m s}^{-1}$, (c) $\sigma = 1 \text{ } \mu\text{m s}^{-1}$, (d) $\sigma = 2 \text{ } \mu\text{m s}^{-1}$

Chapter 4

Discovering Interaction Laws

Our goal is to infer the one-body (self-propulsion) and the two-body forces (interactions) that take place in ensembles of active particles using a machine learning tool. Such systems present a high degree of complexity and a full understanding of the actual interactions has yet to be reached; in this situation, machine learning could be the way of inferring systems' dynamic laws from the observation of simulated and experimental behaviors, with nothing more than some simple assumptions as an underlying model.

As explained in section 1.3, one could use as input some global attribute of the system, such as the radial distribution function, but, as showed by Bag and Mandal, this approach does not work well in out-of-equilibrium cases where an active velocity is present, since self-propulsion may make the system cluster in a MIPS (Motility Induced Phase Separation) fashion and thus the structure may reflect that of an equilibrium system where an attractive interaction is applied.

4.1 Methods

The tool we propose is based on a Graph Neural Network (GNN) that takes as input the positions and orientations of a set of active particles and tries to predict the resulting velocities. Here, we tried to replicate the approach used by Ruiz-Garcia, Barriuso G., Alexander, *et al.*, where a vector of positions and orientations of a set of active particles is given as input to a Graph Neural Network which tries to predict resulting velocities. Both in our work and in the reference paper the objective is to train a network on simulated data and then use it to predict interaction potential in experiments. The main difference is in the experimental setup used: Ruiz-Garcia, Barriuso G., Alexander, *et al.* used snapshots taken from electrophoretic Janus particles, where some control is held on the system via an external electric field. This is not the case in the host lab, where autonomous self-diffusiophoretic Janus particles are used. Simulation framework is also different: in the reference work particles are simulated using a modified version of LAAMPS [37] molecular dynamics simulator, while our simulation code is made in-house as explained in 2.

In principle, one could also use a Dense Neural Network (DNN) to learn a high-dimensional nonlinear function that maps a $3N_p$ vector (in 2D, with N_p number of particles) of all particles' positions and orientations into a $2N_p$

vector of velocities, but a Network like this would struggle to adapt to cases with different N_p , especially if the number of particles is changing, as happens in experimental settings.

A GNN is structured as a combination of (at most) three DNNs on a graph structure, to predict respectively node (single particle), edge (two particles interactions) and global features. This way, one could initialize a graph structure at each time step, with nodes storing positions and orientations of particles and undirected, unweighted edges that represent which pairs of particles are considered as interacting. In principle, such a structure could be initialized with varying number of nodes (particles), since this number does not affect the GNN compatibility, as each edge and node is itself used as input to the network.

4.1.1 The Graph Neural Network

Our GNN is structured as explained in section 1.3, namely with a node and a message function. Both of them have 4 layers, with 300 hidden nodes; each hidden layer has a ReLU (Rectified Linear Unit) activation function, while the output layer of each Network is a simple linear layer without any activation. As for Figure 1.14, the input layer of the message function is $2n_f$ dimensional, where n_f is the number of single particle features, namely x , y , and θ , to have information about positions and orientations of each pair of interacting agents. Messages are aggregated using sum as an aggregation function ('add' in PyG jargon) to respect the physics of the problem. The node function's input dimension is $n_f + n_m$ where n_m is the output dimension of the message function; this is done because the node function has the purpose of taking the aggregated message from all the senders (particle inside a threshold distance Γ) along with the single features of the receiver and trying to predict the receiver's velocity. In Newtonian dynamics, an acceleration is the result of a force, while in low-Reynolds number regime there is a linear relation between force and velocity. In this regime, it is convenient to use velocity as a target function to infer force's characteristics. Overdamped regime is one of the basic assumptions made to build this tool, together with the fact that we neglect many-body interactions, focusing only on one- and two-body forces. A schematic of our GNN's structure is in Figure 4.1.

If ξ is the edge function and ϕ is the node function (see Figure 1.14 as a reference), we have that the predicted velocity \vec{v}_i^p takes the form

$$\vec{v}_i^p \equiv \tilde{\psi} \left(\vec{c}_i, \sum_{d_{ij} < \Gamma} \vec{\xi}(\vec{c}_i, \vec{c}_j) \right), \quad (4.1)$$

being \vec{c}_k the vector containing positions and orientation of k -th particle. The aim of our network is to minimize the absolute difference between predicted and ground-truth instantaneous velocity, which is our loss function

$$\mathcal{L} = \sum_i |\vec{v}_i - \vec{v}_i^p|. \quad (4.2)$$

Here it is important to note that the history is not relevant: the network just takes one instant at a time and predicts instantaneous velocities starting from positions and orientations, without knowing what happens before or after.

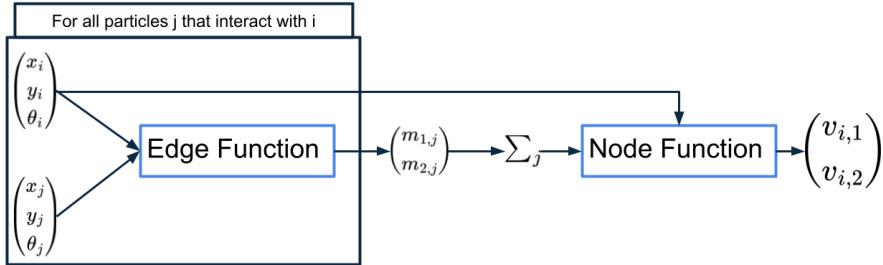


Figure 4.1: Schematic of our GNN.

Ground-truth instantaneous velocities are computed dividing the difference of consecutive positions by the integration time interval, $\frac{\vec{x}(t+\Delta t)-\vec{x}(t)}{\Delta t}$, that is the "right" instantaneous velocity that brings the particle from position $\vec{x}(t)$ at time t to $\vec{x}(t+\Delta t)$. Since our simulations are performed in periodic boundary conditions, from one frame to the other it can happen that a particle jumps by more than L (side of simulation box), leading to non realistic instantaneous velocities. After the computation, all velocities get checked and $L/\Delta t$ is subtracted to the ones corresponding to periodic jumps, in order to have physical velocities as ground truth.

The problem of message dimension n_m is tackled in section 1.3. Here, we chose a message dimension $n_m = 2$, as reported in the reference paper [23]. Some trials with $n_m = 100$ and L1 regularization were made, in the spirit of [22], but in the message extraction it was not possible to limit the maximum standard deviation to two message components, as, ordering them by variance, the most important components changed from epoch to epoch, not leading to a stable reduction in message size.

4.1.2 GNN Implementation

The Graph Network is defined as an object of class `MessagePassing`, which is a base class in PyTorch Geometric. Once this object is defined the two DNNs for message and node functions must be added into it and the base methods of `MessagePassing` will take care of propagating the information in the graph network.

This is imported as a model in the training code, that, using the PyTorch package, has all the tools to train the network. Data preparation consists of taking the snapshots of positions and orientations, in our case we used 750 equally spaced snapshots for training and 250 for testing, and transferring them to a PyTorch Tensor object. In order to store information in a graph structure, we used the `DataLoader` primitive of PyTorch Geometric, that takes the node

features (positions and orientations) along with the `edge_index` matrix, which represents the adjacency matrix of the graph (i.e. which are the linked nodes). We used a batch size of 64 graphs for training.

Training consisted of 100 epochs (see below) and we used `Adam` optimizer as a standard choice. Learning rate was dynamically changed with a `OneCycleLR` scheduler with default settings, being 1×10^{-3} the initial value.

4.1.3 Simulations

To understand the role of the potential's functional form in prediction results, we trained and tested the network on two different simulations, one done using a spring potential with $k_s = 0.1 \text{ N m}^{-1}$ and $x_0 = 12 \mu\text{m}$ and the other using a LJ potential with $\sigma = 4 \mu\text{m}$ and $\epsilon = 0.01 \text{ pJ}$ (Figure ??).

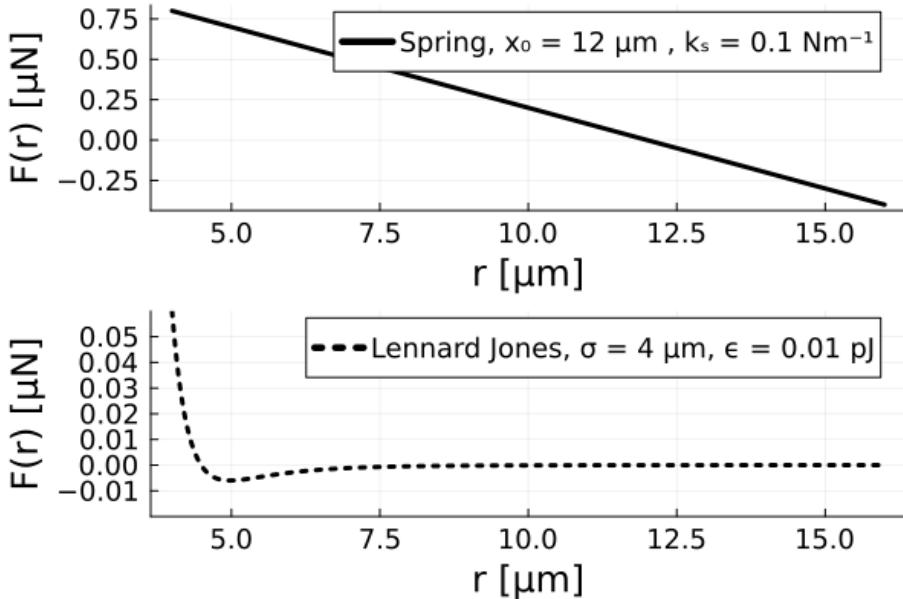


Figure 4.2: Potential used in GNN training.

These potential have the difference in range: without limiting our interaction threshold radius, elastic force not only has effect at long distance but its absolute value increases, while LJ is a short range potential. In simulations, interaction range is not limited in order to have the maximum amount of interacting particles as possible. We expect the network to perform worse on LJ since it has less relevant interactions to learn from.

These two potentials required different integration steps: 10^{-3} s for LJ and $5 \times 10^{-2} \text{ s}$ for spring potential. Both simulations were ran with a particle velocity of $15 \mu\text{m s}^{-1}$, in a $100 \mu\text{m} \times 100 \mu\text{m}$ box with 100 particles of $2 \mu\text{m}$ in radius. Interacting position corresponds to particles' center, since at this level we are interested in just inferring forces.

4.2 Training and Testing

We let the two systems evolve for a total simulated time of 1000 s, then instantaneous velocities were computed and finally a sample of 1000 equally spaced snapshots was taken from each simulation. We divided these snapshots in 750 for training and 250 for testing. The network was trained and tested on each simulation's data separately to have preliminary results. We tried the same procedure explained in [23], where two particles are considered as linked in the graph if their distance is less than some threshold, and then increasing the threshold in subsequent training loops. In our case, after learning with the first threshold distance, loss does not decrease and test results do not improve, so what follows refers to a threshold distance of 20 μm . This threshold was chosen in order to have both repulsion and attraction for the two potentials. In reference paper, only distances up to three particle diameters are plotted in the learned force plots, since no additional information can be extracted from the fast-decreasing potentials Ruiz-Garcia, Barriuso G., Alexander, *et al.* used. Lennard Jones is a fast-decreasing potential so it is possible that after this threshold the interaction force is small enough not to have effect on the learning process. Although elastic force magnitude gets larger with increasing distance, after some time particles will all be oscillating in the vicinity of the equilibrium position x_0 , leading to a lack of data for larger distances.

After each epoch, the network is used to predict velocities on the test set, which contain data with the same potential used in training, saving the message function outputs. Then, a minimization process is used to find the best linear transformation that maps message components into the known forces. This is done to see how results change with the epochs.

4.3 Results

As showed in Figure 4.3, in both cases loss decreases in the 100 epochs of training, meaning that the network is in fact learning.

The absolute value of the loss in this case only means that spring potential leads to faster interacting particles and only the decrease is important in our analysis.

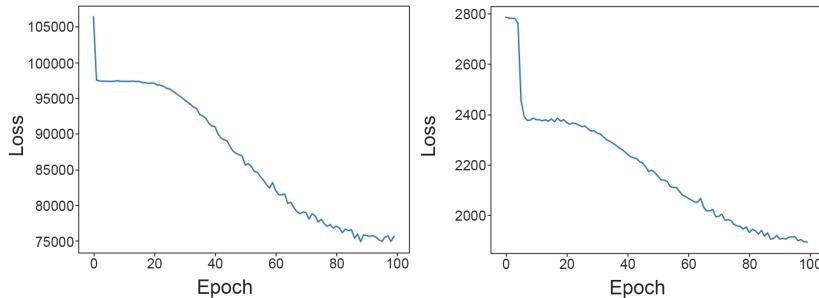


Figure 4.3: Learning curve for elastic potential (left) and Lennard Jones (right). Loss is the absolute difference between predicted and ground truth velocities.

Since the output of the message function should represent interaction forces, in order to check the prediction quality we need to compare messages to known forces. We work in cases where both the distances between particles and the functional form of the force is known, thus we can compute the ground truth. Here we specify that, since the output of the message function is taken as input by the (linear) node function to predict velocity, it will correspond to the actual force only up to a linear transformation. A way to compare messages to forces is to compute the ground truth, extract messages from the network and finding the best linear transformation to map one into the other, finding the right parameters that minimize the square difference between the two. Regarding the message-force plot, a working network with the right linear transformation should show points on the $x = y$ line, being the message components in perfect correspondence with a rotation of the force components as 1.13 shows. In Figure 4.4 we reported the qualitatively best results, since the loss scoring in this case does not reflect a prediction quality.

As the animation in [22] shows, as the network learns, the message-force graphs should show a better agreement between what is learned by the GNN and ground-truth. In our case, as it is shown in fig 4.5, in the case of spring potential, the message-force plot has all the points on a vertical line, showing no correspondence between the learned results and ground-truth forces. This could be caused by a tendency to overfit on the presented data, since loss is decreasing while test result are worse.

Anyway, it is possible to notice a difference between the two potentials in Figure 4.4: in the case of Lennard Jones (Panel (b)), few points are outside the vertical line and they position in an horizontal cloud around it, while for the elastic potential (Panel (a)) most points are scattered around an horizontal cloud and a density increase can be observed in diagonal lines.

4.4 Conclusions to Chapter 4

Here we presented the preliminary version and results of a machine learning tool that predicts forces between interacting active particles, in the spirit of the ActiveNet system developed by Ruiz-Garcia, Barriuso G., Alexander, *et al.*

As shown in section 4.3, more work is needed in order to make this tool work as expected. In particular, training for longer with more simulation snapshots is certainly needed in order to get the right accuracy. In reference paper, the problem of data amount is tackled comparing errors in cases with different simulation temperature. It is clear how with higher temperature noise tends to increase and more data is needed in order to have accurate estimates of the self-propulsion and interaction forces. In general, we can say that, due to computational time limitations, the amount of particles in our simulations (100) is pretty low if compared with what Ruiz-Garcia, Barriuso G., Alexander, *et al.* have used (2500). Knowing that the number of edges in an undirected graph is quadratic with the number of nodes, the number of total interactions to learn from is extremely low in our case if compared with the reference paper. While ActiveNet authors used a molecular dynamics simulator, made to simulate a high number of particles, our simulation framework was built to simulate conditions akin to the ones we see in experiments, thus it is optimized for a low packing fraction and low number of particles setting. Moreover, our aim was to train

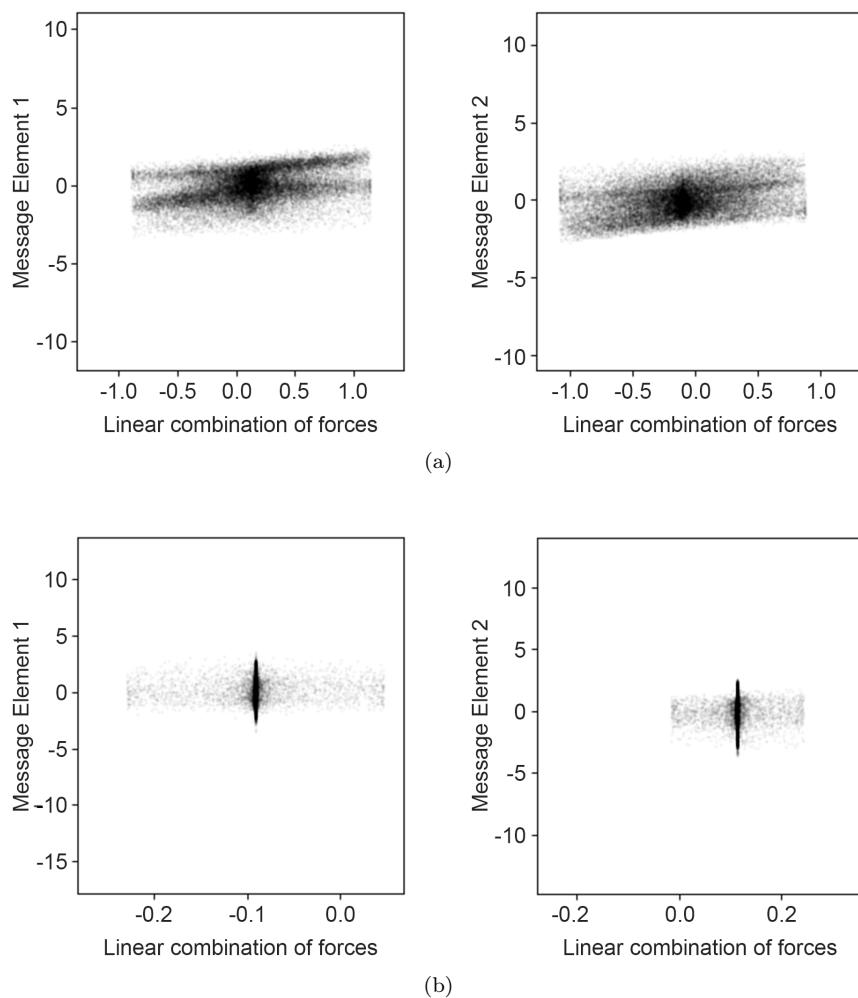


Figure 4.4: Representative message-force plots of early epochs of training for spring potential (a) and Lennard Jones (b).

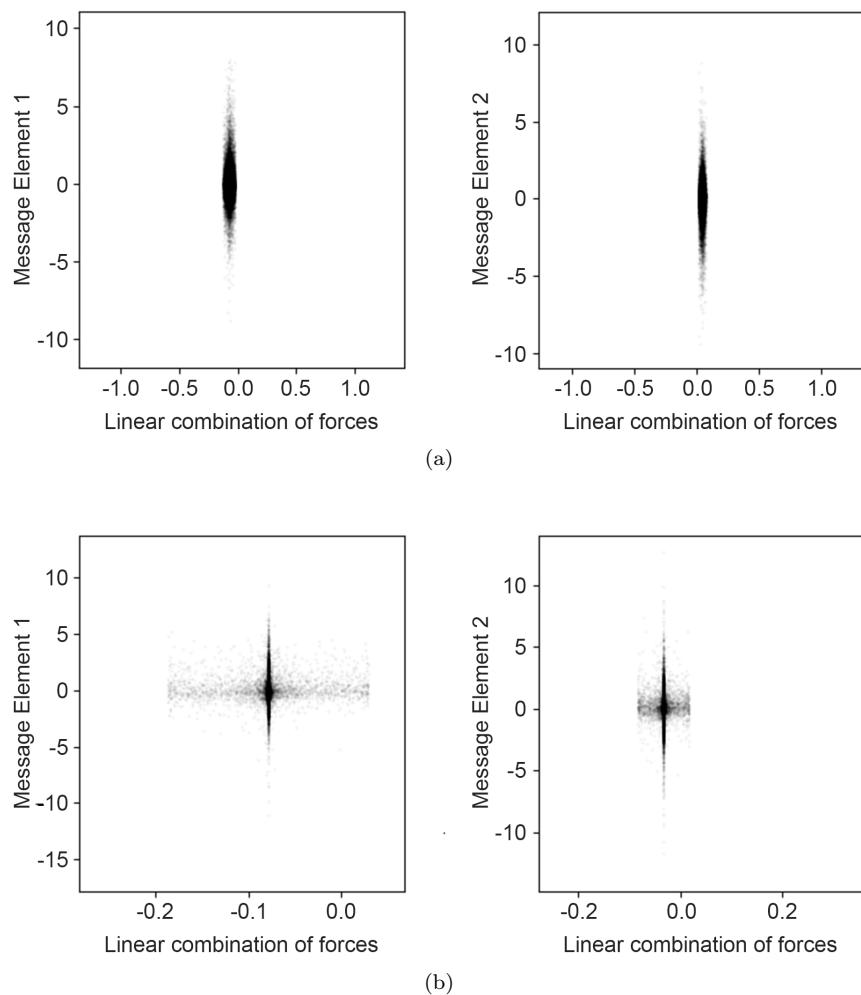


Figure 4.5: Representative message-force plots of last epoch of training for spring potential (a) and Lennard Jones (b).

the network in such conditions to make it more adaptable to the analysis of experimental data. In the end this caused a big difference in the amount of data used for training.

Retraining process with increasing threshold distance must be perfected, in order to get at the same level of the reference paper. For all that was said in previous sections, it might be a good choice to start training with lower threshold distances, since 20 μm already contains most of the significant interactions for the selected potentials.

After making the training process work as desired, the next step is to retrain the network with snapshots taken from simulations with different potentials, in order to increase its generalization capability.

Moreover, testing with experimental data will need some more adjustments. In an experimental setting, defects inherited from the fabrication process like stuck particles and particle clumps can make it harder for a machine learning tool to work properly. In the case of experimental data collected in the Microscale Robotics Lab, it is hard to measure particles' orientation, since the two hemispheres of silica Janus particles are not clearly distinguishable, though a difference in color is noticeable. Moreover, these particles are free to rotate in three dimensions and nothing prevents them from pointing one hemisphere to the microscope objective, not showing the separation line. One workaround to this issue could be taking as particle's orientation the instantaneous angle that its trajectory forms, computing it between two consecutive instants. This would be putting all the information about interactions inside self-propulsion, nonetheless if the two time scales are sufficiently different, one might be able to separate the two, leading to an accurate estimate of particles' orientation. This approach needs some testing in simulations, when orientations are known, to understand its limitations and advantages.

We can state that these results, though certainly not satisfying, can be a starting point to further developments of this tool, getting it to work as expected in the case of active particles like those under investigation in the host laboratory. With the potentials inferred from experiments, we would be able to model particles in a more accurate and complete way, to simulate them and use simulated data back in a simulation-driven inference fashion.

Chapter 5

Conclusions

In this thesis work we investigated how modeling and analysis tools can expand our knowledge regarding active particles systems. To do so, we built on top of a simulation framework able to manage most of the features needed in a *dry* ABPs computer experiment, such as steric interactions and interplay between particles and confinement, adding two-body aligning interactions. The novelty in this work is modeling technique: instead of applying an explicit torque, we applied a force to an off-center position that results (in the overdamped regime) both in a linear velocity applied *to* the center of mass and in an angular velocity *with respect to* the center of mass, thus automatically applying an aligning interaction torque.

To the best of our knowledge, this approach was not tried before in ABPs systems simulations, although a similar idea could be found in [19], where both a force and a torque are applied in a microswimmer interacting with a confinement. That work though analyzes single particle behaviors, and the only interaction is with a wall.

All the described features were added to our simulation framework making sure for them to integrate seamlessly with periodic boundary conditions, in order to have particles interacting across the simulation space, making the movement-accessible space interaction-accessible as well. Hard-spheres correction was made periodic too, resulting in the elimination of the artifact that kept particles from accumulating across simulation box boundaries. Inserting a threshold range in interactions allowed to accelerate the simulation, resulting in reduced execution times, and additional aid in simulating topological interactions and studying the effect that interaction range have on the system. In future developments, the presence of a threshold range could be useful to insert screening effects that particles may exert on each other, as well as solvent-driven ones, that limit the range of interactions in a physical experiment.

Regarding the similarity with experimental behaviors, our simulations perform well for two-particles interactions, showing that an aligning interaction is taking place as claimed by [17], while three bodies dynamics such as rotation of a triplet occur with a faster time scale in simulations, although qualitatively similar. An observation of experimental videos from the host laboratory suggests a stronger, but shorter range interaction is needed to get experimental clustering without the whole system self-organizing in a single large cluster.

We showed that a system with a repulsive interaction and an interacting

position off-centered to the front of the swimming particles presents a phase transition to flocking. As stated in section 3.4 this would make this kind of system a variation of the continuous Vicsek model, making it worthwhile to study its transition from the point of view of universality classes, which is a matter of discussion for the well-known Vicsek model as well [38]. As results in chapter 3 show, in the explored cases an interacting position on the back, coupled with an attractive potential does not let the system flock. This is in good agreement with experimental observations where flocking was never observed, encouraging us to trust our model, since it qualitatively resembles the fact that the interacting part of a Janus particle is on its catalytic side and Pt-silica Janus particles are known to swim with platinum hemisphere on the back. A short-range attraction is not able to align two particles that swim towards each other, as observed through this thesis. As literature [39] shows, Cu-silica Janus particles (also being studied in the host laboratory) swim towards their copper hemisphere and they show a longer interaction range compared to Pt-silica ones, as well as a stronger repulsion. This could make them similar to the repulsive particles we simulated (section 3.2) and, if aligning interactions take place between them, they could be a good case study for flocking.

From the point of view of the host project, CELLOIDS, flocking could be an interesting behavior since the capacity to control the direction of a whole set of particles is crucial to steer a particles-filled microrobot to its objective. Still, making an ensemble of active particles polarize without the presence of an external field is challenging and a deeper investigation, both *in vitro* and *in silico* will be needed to reach such a goal. Furthermore, the presence of faster particles (high Péclet number) makes flocking transition faster, spreading the information about orientation faster across the system. This fact would be useful in an experimental setting, provided studied without periodic boundary conditions and with a confinement.

Moreover, we showed how an angular velocity is able to disrupt global polarization in a system. Nonetheless, we observed the system keep some local polarization in oriented domains in which each particle rotates on its axis. An interesting development of this observations would be labeling each particle with the angular velocity it was generated with. With linear velocity distributions it is pretty evident how slower particles tend to cluster, leaving faster particles in a free state, and it would be remarkable if chiral particles would undergo some spontaneous demixing or segregation process, where particles with similar angular velocity tend to aggregate.

A preliminary version of an inference tool was developed, but more work needs to be done in order to make it work as needed, both in data preparation and network training. Once working on a single potential, this tool should be trained to generalize on any potential and tested with simulations before applying it to experimental data. It should also be able to work in situations where some collective behaviors take place to shed more light on this phenomena from an experimental standpoint. ActiveNet [23] authors stated that their tool has the capacity to generalize to aligning interactions, and active angular velocities: that would be the next step for us, since, among our analysis tools, the GNN is the only one which has not been tested with active and two-body torques.

As a general objective, the work presented here could be implemented in cases where a confinement is present, with several shapes, to make it useful in microrobots designing and testing. Experimenting with different types of

interactions, aligning and not aligning, between particles and boundary will be crucial to apply this knowledge to real cases. A further advancement would be exploring the effect of soft confinements, which will need their own modeling and simulation framework, in order to simulate the encapsulation of Janus particles inside Giant Unilamellar Vesicles and realize active particles-based cell-like microrobots.

In a nutshell, the main contributions of this work are a novel type of modeling, that unifies aligning and not aligning interactions between ABPs, the introduction of linear and angular velocity distributions that allow to better mimic experimental behaviors and to unveil new effects in a theoretical setting. Along with this, we adapted well-known tools in order to get a better understanding of how collective behaviors act on structure and dynamic of the system, concluding that the two are often related. For a system with self-propulsion, global and local polarizations represent the interplay between static and dynamic properties, and we studied both their steady state behavior and their time evolution, getting a better understanding of the interaction between movement and structure of the system. We tested the limits of a machine learning approach to Brownian dynamics, trying to replicate ActiveNet in a down-scaled version, obtaining small or no success.

Bibliography

- [1] C. W. Gardiner, *Handbook of stochastic methods for physics, chemistry and the natural sciences* (Springer Series in Synergetics), Third. Berlin: Springer-Verlag, 2004, vol. 13, pp. xviii+415, ISBN: 3-540-20882-8.
- [2] G. E. Uhlenbeck and L. S. Ornstein, “On the theory of the brownian motion,” *Physical Review*, vol. 36, no. 5, pp. 823–841, Sep. 1, 1930, Publisher: American Physical Society. DOI: 10.1103/PhysRev.36.823. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRev.36.823>.
- [3] J. L. Moran and J. D. Posner, “Phoretic self-propulsion,” *Annual Review of Fluid Mechanics*, vol. 49, pp. 511–540, Volume 49, 2017 Jan. 3, 2017, Publisher: Annual Reviews, ISSN: 0066-4189, 1545-4479. DOI: 10.1146/annurev-fluid-122414-034456. [Online]. Available: <https://www.annualreviews.org/content/journals/10.1146/annurev-fluid-122414-034456> (visited on 01/21/2025).
- [4] R. P. Feynman, “There’s plenty of room at the bottom,” 1959. [Online]. Available: <https://caltech.library.caltech.edu/1976/>.
- [5] E. M. Purcell, “Life at low reynolds number,” *American Journal of Physics*, vol. 45, no. 1, pp. 3–11, Jan. 1, 1977, ISSN: 0002-9505. DOI: 10.1119/1.10903. [Online]. Available: <https://doi.org/10.1119/1.10903>.
- [6] S. Palagi and P. Fischer, “Bioinspired microrobots,” *Nature Reviews Materials*, vol. 3, no. 6, pp. 113–124, Jun. 2018, Publisher: Nature Publishing Group, ISSN: 2058-8437. DOI: 10.1038/s41578-018-0016-9. [Online]. Available: <https://www.nature.com/articles/s41578-018-0016-9>.
- [7] “Robot noun - definition, pictures, pronunciation and usage notes | oxford advanced learner’s dictionary at OxfordLearnersDictionaries.com.” (), [Online]. Available: <https://www.oxfordlearnersdictionaries.com/definition/english/robot>.
- [8] A. Callegari and G. Volpe, “Numerical simulations of active brownian particles,” in *Flowing Matter*, F. Toschi and M. Sega, Eds., Cham: Springer International Publishing, 2019, pp. 211–238, ISBN: 978-3-030-23370-9. DOI: 10.1007/978-3-030-23370-9_7. [Online]. Available: https://doi.org/10.1007/978-3-030-23370-9_7.
- [9] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, “Novel type of phase transition in a system of self-driven particles,” *Physical Review Letters*, vol. 75, no. 6, pp. 1226–1229, Aug. 7, 1995, ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.75.1226. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.75.1226>.

- [10] G. I. Menon, "Active matter," in *Rheology of Complex Fluids*, J. M. Krishnan, A. P. Deshpande, and P. B. S. Kumar, Eds., New York, NY: Springer, 2010, pp. 193–218, ISBN: 978-1-4419-6494-6. DOI: 10.1007/978-1-4419-6494-6_9. [Online]. Available: https://doi.org/10.1007/978-1-4419-6494-6_9.
- [11] S. Ramaswamy, "Active matter," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2017, no. 5, p. 054002, May 2017, Publisher: IOP Publishing and SISSA, ISSN: 1742-5468. DOI: 10.1088/1742-5468/aa6bc5. [Online]. Available: <https://dx.doi.org/10.1088/1742-5468/aa6bc5>.
- [12] A. Cavagna, A. Cimarelli, I. Giardina, *et al.*, "From empirical data to inter-individual interactions: Unveiling the rules of collective animal behavior," *Mathematical Models and Methods in Applied Sciences*, vol. 20, pp. 1491–1510, supp01 Sep. 2010, Publisher: World Scientific Publishing Co., ISSN: 0218-2025. DOI: 10.1142/S0218202510004660. [Online]. Available: <https://www.worldscientific.com/doi/abs/10.1142/S0218202510004660>.
- [13] M. Ballerini, N. Cabibbo, R. Candelier, *et al.*, "Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1232–1237, Jan. 29, 2008, Publisher: Proceedings of the National Academy of Sciences. DOI: 10.1073/pnas.0711437105. [Online]. Available: <https://www.pnas.org/doi/full/10.1073/pnas.0711437105>.
- [14] A. Martín-Gómez, D. Levis, A. Díaz-Guilera, and I. Pagonabarraga, "Collective motion of active brownian particles with polar alignment," *Soft Matter*, vol. 14, no. 14, pp. 2610–2618, Apr. 4, 2018, Publisher: The Royal Society of Chemistry, ISSN: 1744-6848. DOI: 10.1039/C8SM00020D. [Online]. Available: <https://pubs.rsc.org/en/content/articlelanding/2018/sm/c8sm00020d>.
- [15] L. Caprini, U. Marini Bettolo Marconi, and A. Puglisi, "Spontaneous velocity alignment in motility-induced phase separation," *Physical Review Letters*, vol. 124, no. 7, p. 078001, Feb. 19, 2020, Publisher: American Physical Society. DOI: 10.1103/PhysRevLett.124.078001. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.124.078001>.
- [16] R. S. Negi, R. G. Winkler, and G. Gompper, "Emergent collective behavior of active brownian particles with visual perception," *Soft Matter*, vol. 18, no. 33, pp. 6167–6178, Aug. 24, 2022, Publisher: The Royal Society of Chemistry, ISSN: 1744-6848. DOI: 10.1039/D2SM00736C. [Online]. Available: <https://pubs.rsc.org/en/content/articlelanding/2022/sm/d2sm00736c>.
- [17] K. Singh, H. Raman, S. Tripathi, H. Sharma, A. Choudhary, and R. Mangal, "Pair interactions of self-propelled SiO₂-pt janus colloids: Chemically mediated encounters," *Langmuir*, vol. 40, no. 14, pp. 7328–7343, Apr. 9, 2024, Publisher: American Chemical Society, ISSN: 0743-7463. DOI: 10.1021/acs.langmuir.3c03415. [Online]. Available: <https://doi.org/10.1021/acs.langmuir.3c03415>.

- [18] S. Maity and A. Morin, "Spontaneous demixing of binary colloidal flocks," *Physical Review Letters*, vol. 131, no. 17, p. 178304, Oct. 25, 2023, Publisher: American Physical Society. DOI: 10.1103/PhysRevLett.131.178304. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.131.178304>.
- [19] T. Ostapenko, F. J. Schwarzenahl, T. J. Böddeker, *et al.*, "Curvature-guided motility of microalgae in geometric confinement," *Physical Review Letters*, vol. 120, no. 6, p. 068002, Feb. 7, 2018, Publisher: American Physical Society. DOI: 10.1103/PhysRevLett.120.068002. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.120.068002>.
- [20] S. Bag and R. Mandal, "Interaction from structure using machine learning: In and out of equilibrium," *Soft Matter*, vol. 17, no. 36, pp. 8322–8330, Sep. 22, 2021, Publisher: The Royal Society of Chemistry, ISSN: 1744-6848. DOI: 10.1039/D1SM00358E. [Online]. Available: <https://pubs.rsc.org/en/content/articlelanding/2021/sm/d1sm00358e>.
- [21] P. W. Battaglia, J. B. Hamrick, V. Bapst, *et al.*, *Relational inductive biases, deep learning, and graph networks*, Oct. 17, 2018. DOI: 10.48550/arXiv.1806.01261. arXiv: 1806.01261[cs]. [Online]. Available: <http://arxiv.org/abs/1806.01261>.
- [22] M. Cranmer, A. Sanchez Gonzalez, P. Battaglia, *et al.*, "Discovering symbolic models from deep learning with inductive biases," in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 17429–17442. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/hash/c9f2f917078bd2db12f23c3b413d9cba-Abstract.html.
- [23] M. Ruiz-Garcia, C. M. Barriuso G., L. C. Alexander, D. G. A. L. Aarts, L. M. Ghiringhelli, and C. Valeriani, "Discovering dynamic laws from observations: The case of self-propelled, interacting colloids," *Physical Review E*, vol. 109, no. 6, p. 064611, Jun. 26, 2024, Publisher: American Physical Society. DOI: 10.1103/PhysRevE.109.064611. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevE.109.064611>.
- [24] R. Mannella and P. V. E. McCLINTOCK, "Itô versus stratonovich: 30 years later," *Fluctuation and Noise Letters*, vol. 11, no. 1, p. 1240010, Mar. 2012, Publisher: World Scientific Publishing Co., ISSN: 0219-4775. DOI: 10.1142/S021947751240010X. [Online]. Available: <https://www.worldscientific.com/doi/abs/10.1142/S021947751240010X>.
- [25] R. MANNELLA, "INTEGRATION OF STOCHASTIC DIFFERENTIAL EQUATIONS ON a COMPUTER," *International Journal of Modern Physics C*, Nov. 21, 2011, Publisher: World Scientific Publishing Company. DOI: 10.1142/S0129183102004042. [Online]. Available: <https://www.worldscientific.com/worldscinet/ijmpc>.
- [26] J. Sharma, L. Corti, and S. Palagi, *Simulations active brownian particles v1.0*, version v1.0, Jul. 28, 2023. DOI: 10.5281/zenodo.8192894. [Online]. Available: <https://zenodo.org/records/8192894>.
- [27] "The julia programming language." (), [Online]. Available: <https://julialang.org/> (visited on 01/16/2025).

- [28] “PyTorch,” PyTorch. (), [Online]. Available: <https://pytorch.org/>.
- [29] “PyG documentation — pytorch_geometric documentation.” (), [Online]. Available: <https://pytorch-geometric.readthedocs.io/en/latest/index.html>.
- [30] “NumPy.” (), [Online]. Available: <https://numpy.org/>.
- [31] “Pandas - python data analysis library.” (), [Online]. Available: <https://pandas.pydata.org/>.
- [32] C. Bechinger, R. Di Leonardo, H. Löwen, C. Reichhardt, G. Volpe, and G. Volpe, “Active particles in complex and crowded environments,” *Reviews of Modern Physics*, vol. 88, no. 4, p. 045006, Nov. 23, 2016, ISSN: 0034-6861, 1539-0756. DOI: 10.1103/RevModPhys.88.045006. [Online]. Available: <https://link.aps.org/doi/10.1103/RevModPhys.88.045006>.
- [33] J. Barnes and P. Hut, “A hierarchical $O(n \log n)$ force-calculation algorithm,” *Nature*, vol. 324, no. 6096, pp. 446–449, Dec. 1986, Publisher: Nature Publishing Group, ISSN: 1476-4687. DOI: 10.1038/324446a0. [Online]. Available: <https://www.nature.com/articles/324446a0>.
- [34] J. P. Hansen and I. McDonald, *Theory of Simple Liquids*. London: Academic, 1990.
- [35] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD’96, Portland, Oregon: AAAI Press, Aug. 2, 1996, pp. 226–231.
- [36] Chire, *Deutsch: Illustration von clusteranalyse mit de:DBSCAN (minPts=3). punkte bei a sind kernpunkte, und bilden einen cluster. die punkte b und c sind keine kernpunkte, sind aber über die objekte bei a dichte-verbunden und daher teil dieses clusters. punkt n ist weder ein kernpunkt noch im kern eines anderen punktes, daher wird er als rauschen klassifiziert.english: Illustration of en:DBSCAN cluster analysis (minPts=3). points around a are core points. points b and c are not core points, but are density-connected via the cluster of a (and thus belong to this cluster). point n is noise, since it is neither a core point nor reachable from a core point.* Oct. 20, 2011. [Online]. Available: <https://commons.wikimedia.org/wiki/File:DBSCAN-Illustration.svg>.
- [37] A. P. Thompson, H. M. Aktulga, R. Berger, et al., *LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales*, Publication Title: Computer Physics Communications Volume: 271 original-date: 2013-08-09T17:45:11Z, Feb. 2022. DOI: 10.1016/j.cpc.2021.108171. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010465521002836>.
- [38] P. Jentsch and C. F. Lee, “New universality class describes vicsek’s flocking phase in physical dimensions,” *Physical Review Letters*, vol. 133, no. 12, p. 128301, Sep. 17, 2024, Publisher: American Physical Society. DOI: 10.1103/PhysRevLett.133.128301. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.133.128301>.

- [39] P. Sharan, A. Daddi-Moussa-Ider, J. Agudo-Canalejo, R. Golestanian, and J. Simmchen, “Pair interaction between two catalytically active colloids,” *Small*, vol. 19, no. 36, p. 2300817, 2023, ISSN: 1613-6829. DOI: 10.1002/smll.202300817. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/smll.202300817>.