

Thesis Title

University of Pisa

Niccolò Picchiarelli

February 6, 2025

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 [9]. Later, Ornstein and Uhlenbeck [30] 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” [8], 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.

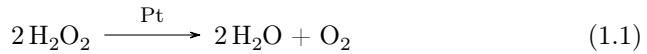
The scaling laws of physical quantities, especially regarding friction and hydrodynamics, makes it challenging to construct a micrometer scale robot which is just a miniaturized version of a macroscopic one. Moving in a fluid for the size and speed of few $\mu\text{m}/\text{s}$ we are interested in-meaning, 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 [21]. 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. When talking about micro-robots, 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”; 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

articolo?

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 [4].

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 [31], 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 [15, 24].

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.[31]

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, [6, 2] are essential papers for the study of collective motion, for 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 et al., Ballerini 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.

non capisco questa frase

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

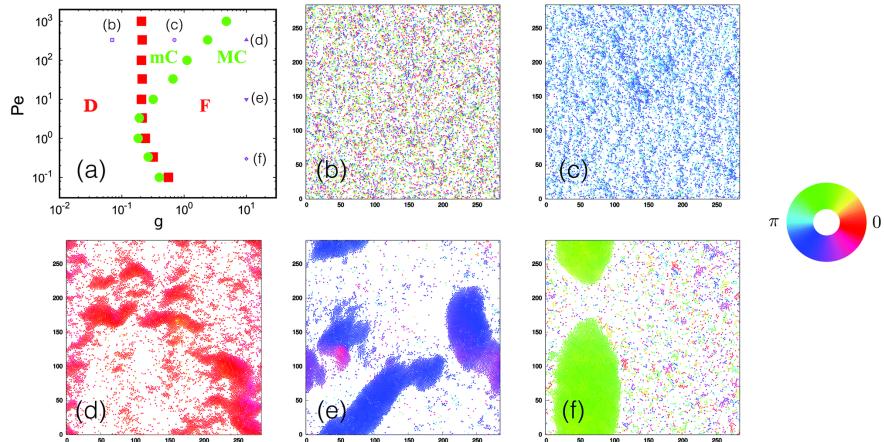


Figure 1.1: [14]

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 [5], 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 order parameter, even when the computation is restricted to clusters, thus the

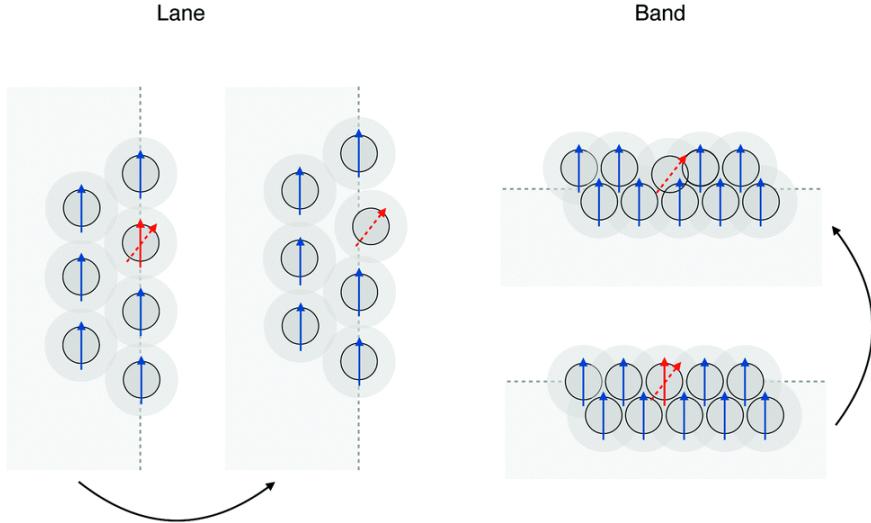


Figure 1.2: [14]

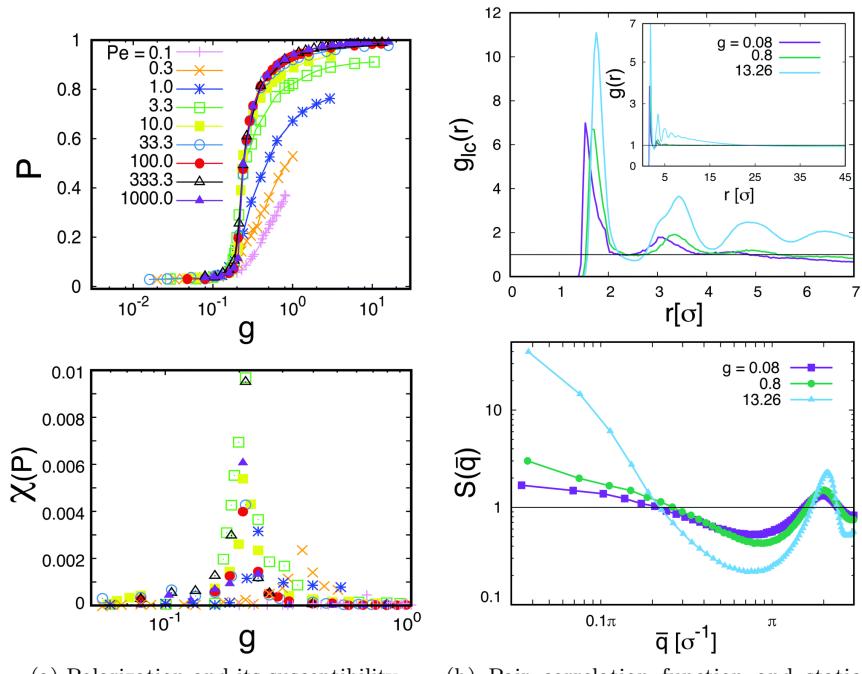


Figure 1.3: [14]

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.

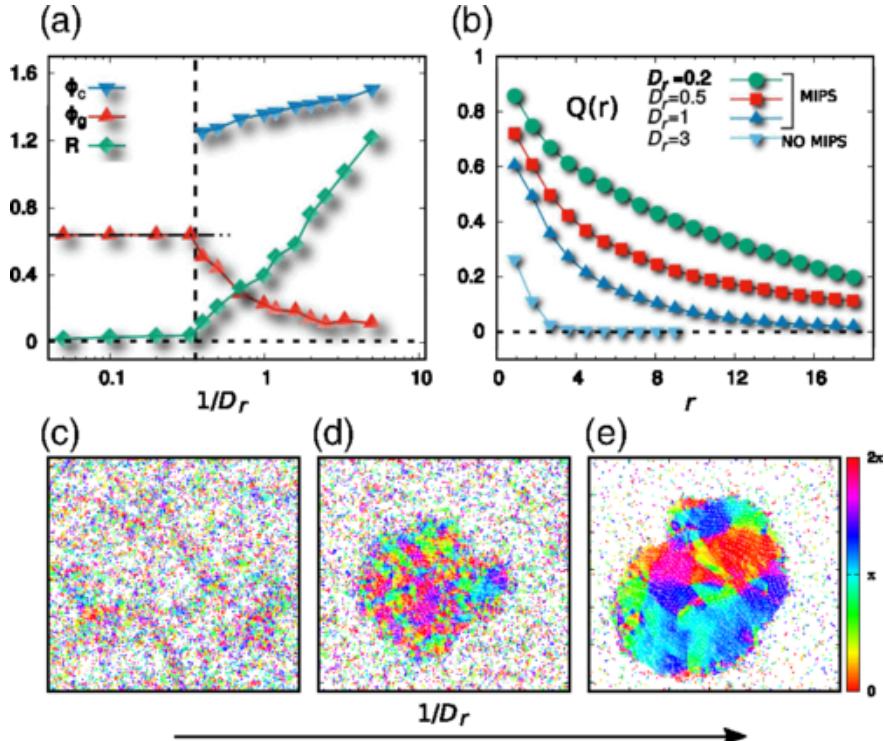


Figure 1.4: [5]

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. [17] 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.

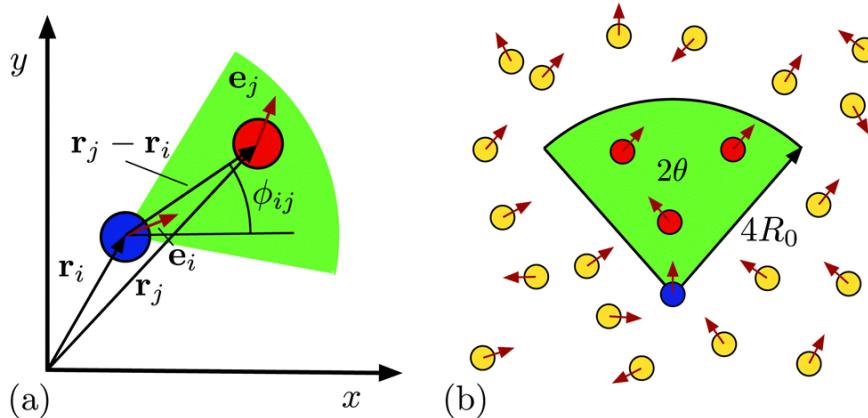


Figure 1.5: [17]

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.

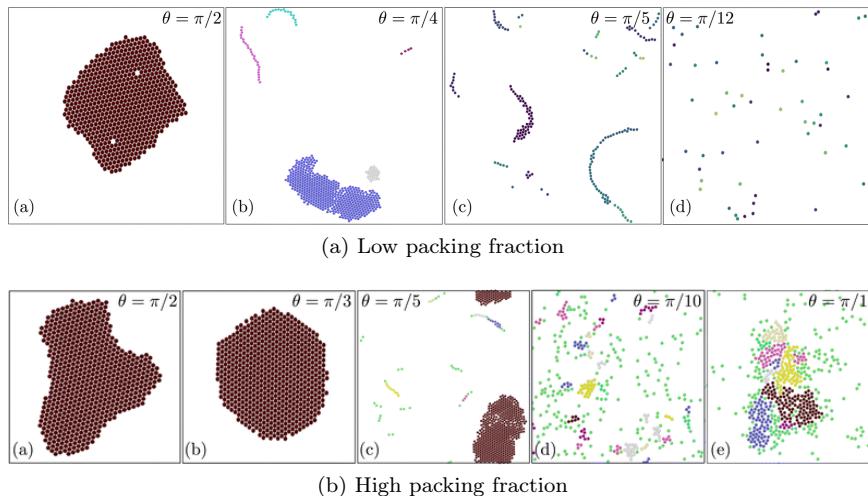


Figure 1.6: [17]

1.3.2 Experimental analysis of collective behaviors

The interaction between active particles and their emergent collective behaviours have been studied also in experiments. [28] studies what happens when two $\text{SiO}_2\text{-Pt}$ 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.

Dispersing Janus colloids in a H_2O_2 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.

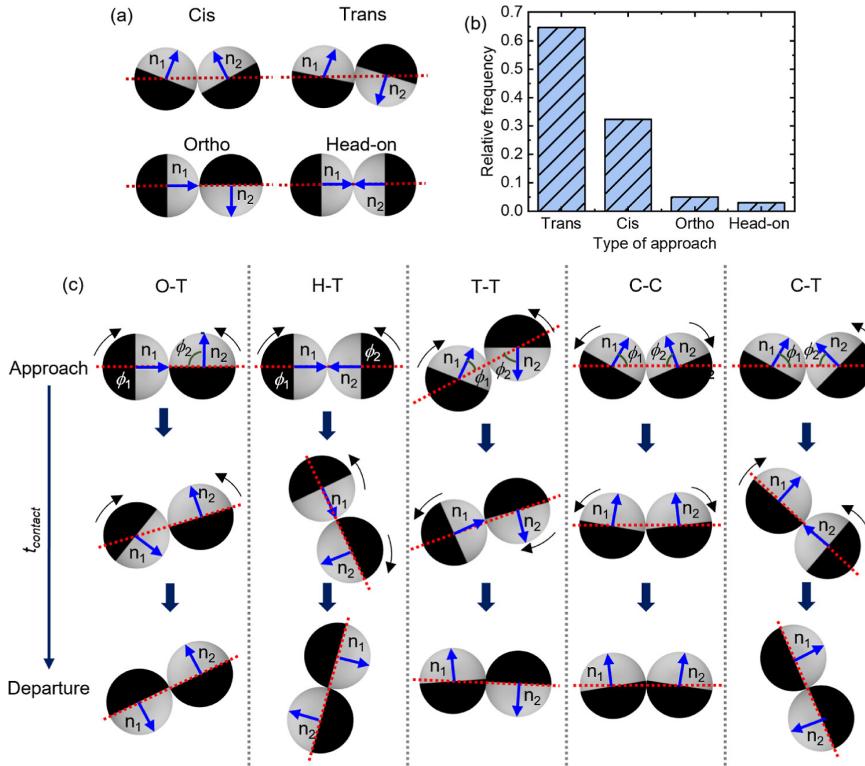


Figure 1.7: [28]

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.

[11] investigate what happens with a binary population of active colloidal particles, that is obtained by mixing particles of $7\text{ }\mu\text{m}$ and $10\text{ }\mu\text{m}$ 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.

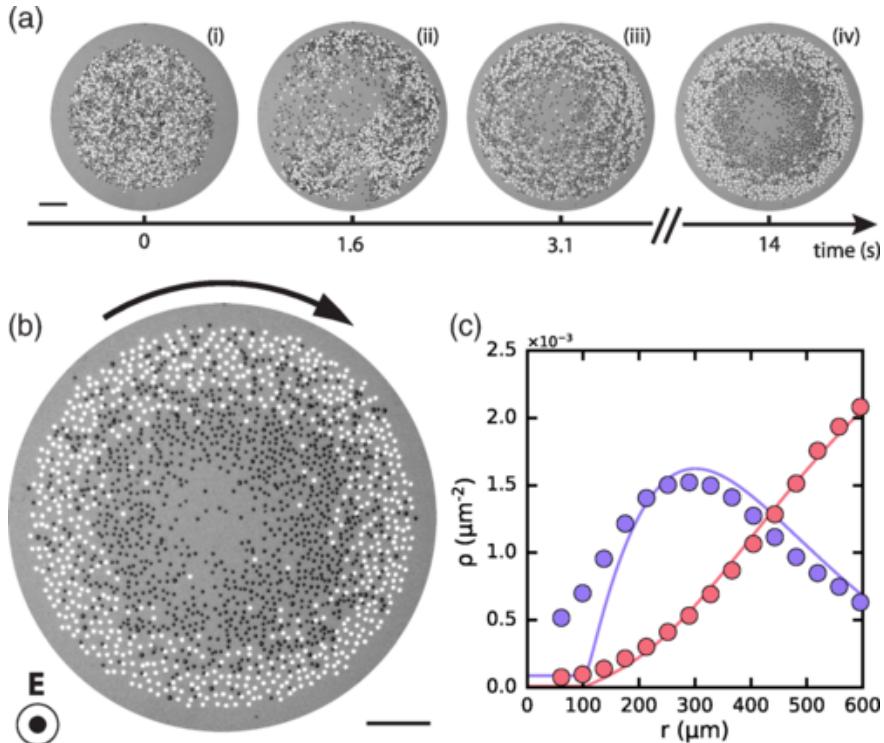


Figure 1.8: [11]

Ostapenko 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.

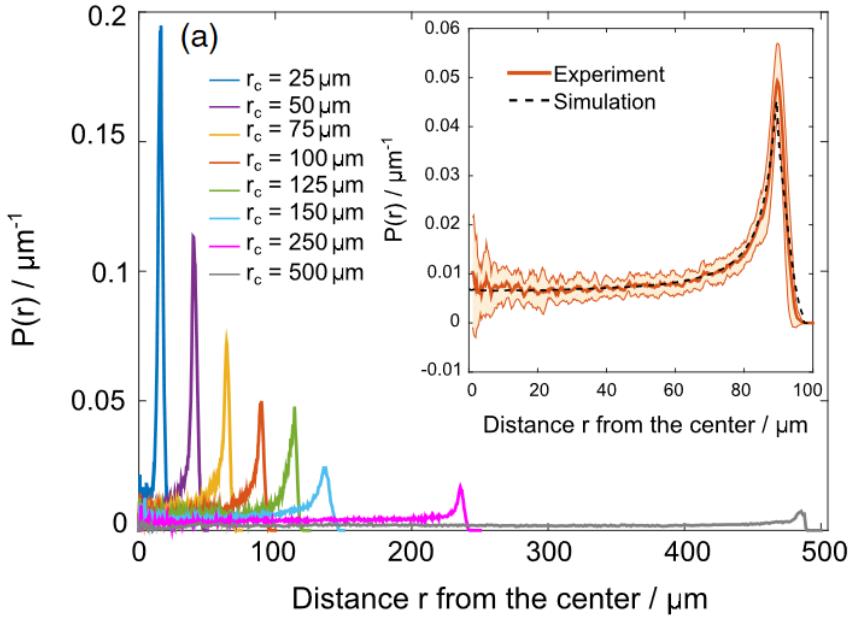


Figure 1.9: [19]

Next, Ostapenko 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.

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.

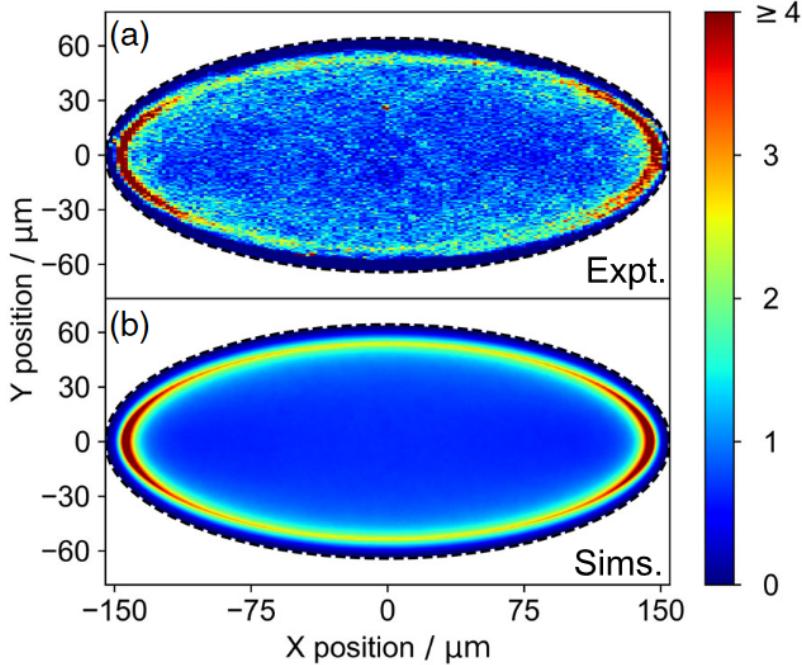


Figure 1.10: [19]

[1] 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 $V(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^a - \lambda \left(\frac{\sigma}{r_{ij}} \right)^b \right]$ 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)

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 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

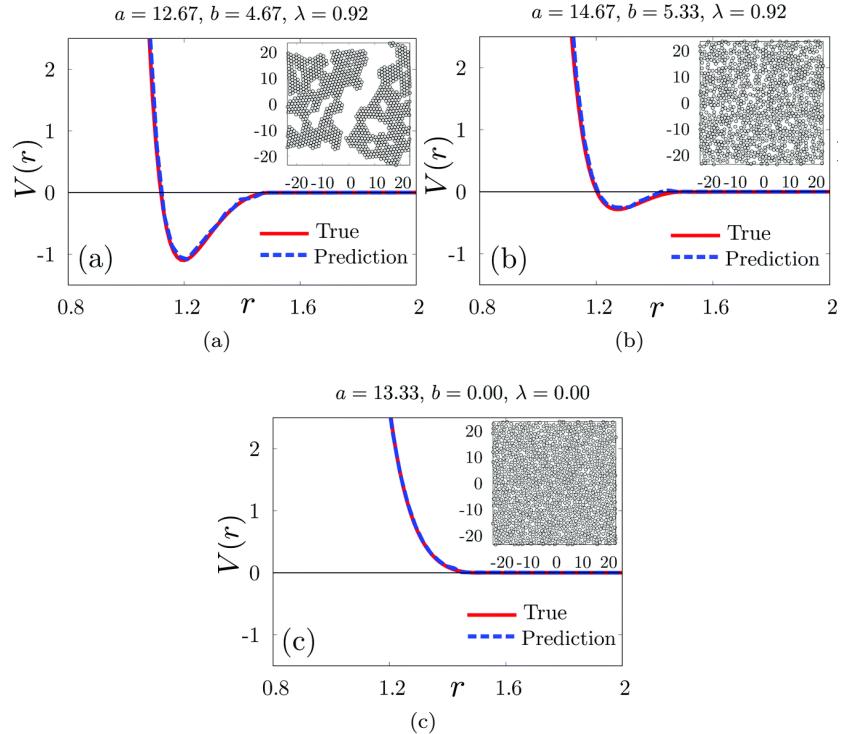


Figure 1.11: Potential shapes for different values of parameters [1].

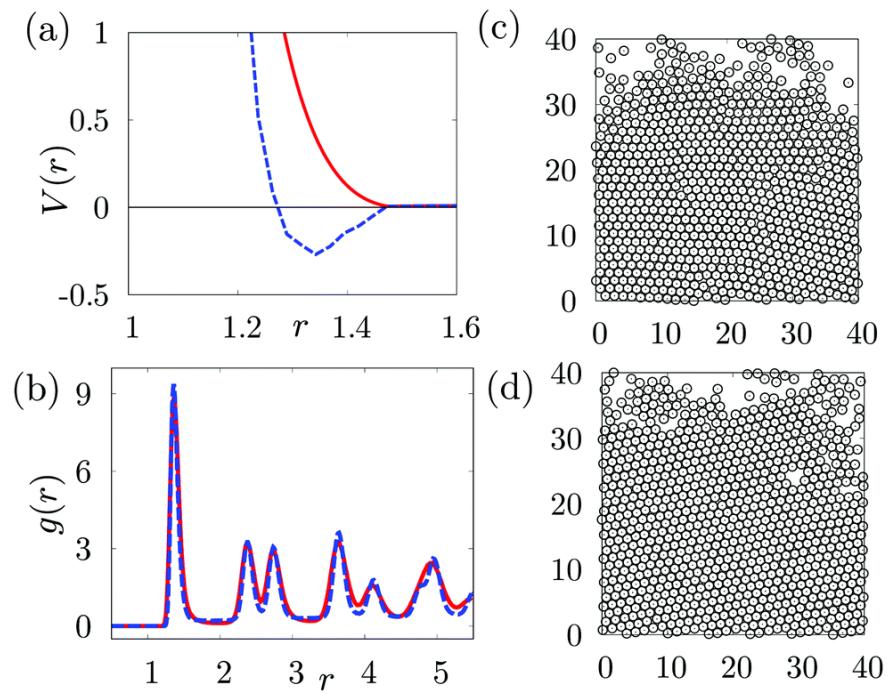


Figure 1.12: [1]

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 [3].

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 [7] 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 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 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

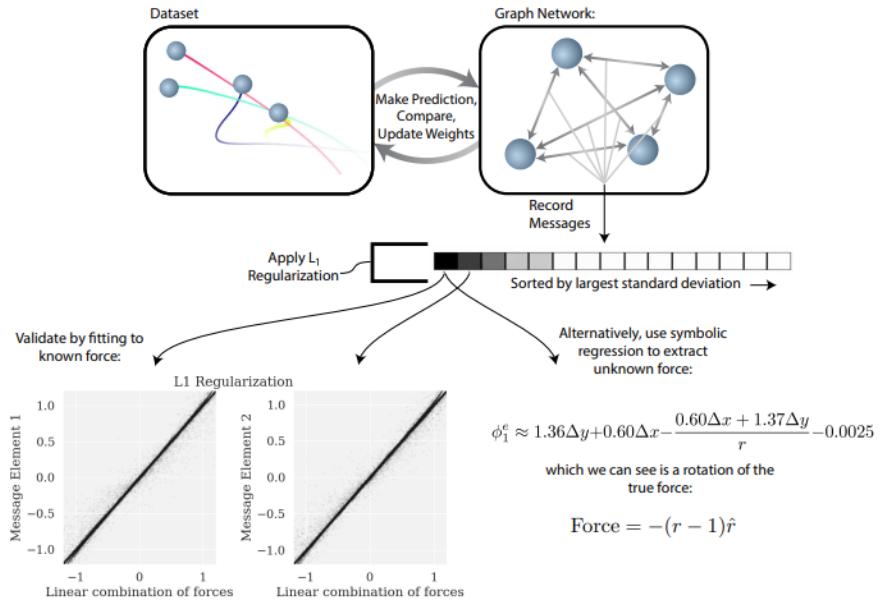


Figure 1.13: [7]

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 et al. have developed a variant of the graph network by Cranmer et al. that can work with active particles dynamics, called ActiveNet. In [26], 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 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

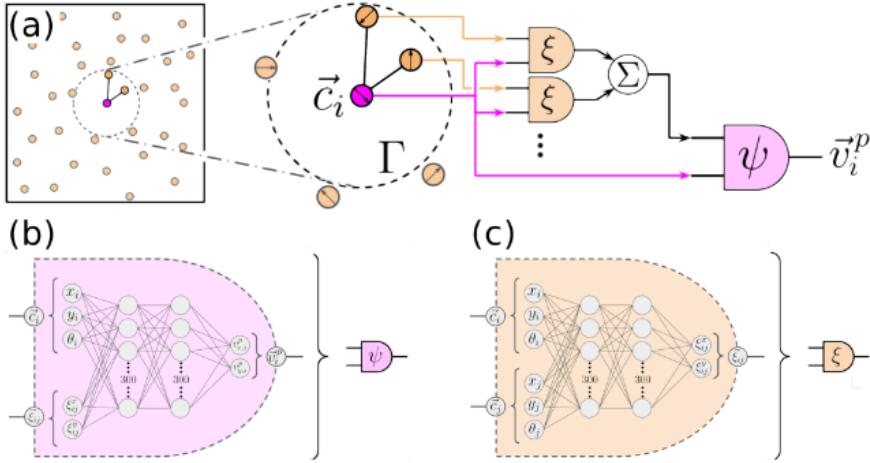


Figure 1.14: [26]

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 [1], 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 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 [9].

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:

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

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 [13].

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 $Z_1 \sim \mathcal{N}(0, \sqrt{h})$ [12].

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

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

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.

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:      $\delta\vec{r} \leftarrow \frac{\Delta t}{2} [v(\cos \theta_{i,n}, \sin \theta_{i,n}) + \vec{F}_{i,n} / \gamma_t + v(\cos \theta_{i,\hat{n}}, \sin \theta_{i,\hat{n}}) + \vec{F}_{i,\hat{n}} / \gamma_t]$ 
10:     $\vec{r}_{i,n+1} \leftarrow \vec{r}_n + w_{i,n} + \delta\vec{r}$ 
11:     $\theta_{i,n+1} \leftarrow \theta_{i,n} + z_{i,n} + \frac{\Delta t}{2} [\omega_{i,n} \Delta t + T_{i,n} / \gamma_r + \omega_{i,\hat{n}} \Delta t + T_{i,\hat{n}} / \gamma_r]$ 

```

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 [12] 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.

quindi? spiega anche per qualcuno che non è né te né Mannella :)

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 [27]. Existing code performed simulations on 2D active Brownian particles moving in open or closed boundary featuring confinement.

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 sphere correction
 - (e) File writing of positions and orientations of all particles in ensemble
 - (f) Computation and file writing of 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.

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. 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. Global variables are numbers and all single particle

| 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

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 as an integrator, is in algorithm 3.

2.2 Methods

All the simulation and analysis code for this thesis was written in **julia** [29], except for the machine learning and inference part, where the Graph Network definition and training makes use of Python libraries like PyTorch [23] and PyTorch-geometric [22], along with standard scientific computing tools like NumPy [18] and Pandas [20].

Heun was chosen as the standard integration scheme for all the simulations, 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 $2 \mu\text{m}$ and the value for solvent viscosity is kept fixed at $1 \times 10^{-3} \text{ Pas}$. If not indicated, standard temperature for simulations is held fixed at 300 K.

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:     $\vec{r}_{i,n+1} \leftarrow \vec{r}_n + w_{i,n} + v(\cos \theta_{i,n}, \sin \theta_{i,n}) \Delta t + \vec{F}_{i,n} \Delta t D_t / k_B T$ 
11:     $\theta_{i,n+1} \leftarrow \theta_{i,n} + z_{i,n} + \omega_{i,n} \Delta t + M_{i,n} \Delta t D_r / k_B T$ 

```

All simulations were performed in a 2 dimensional square environment, where particles are disks, featuring periodic boundary conditions and a non-superposition condition (hard sphere 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 Interaction Forces

2.3.1 All to All 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 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 and it can simply be added to the deterministic step in the coordinate as in algorithm 3, given that we are in low Reynolds number regime.

2.3.2 Tested Potentials

We give here the expressions and parameters explanation for the potentials tested in this thesis. Spring force has two parameters, elastic constant k and rest length

r_0 , so that

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

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

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

and any sign variation, as well as the difference between repulsive and attractive, is absorbed in k . The last interaction potential is Lennard Jones (LJ)

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

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 $r = 2^{1/6}\sigma$.

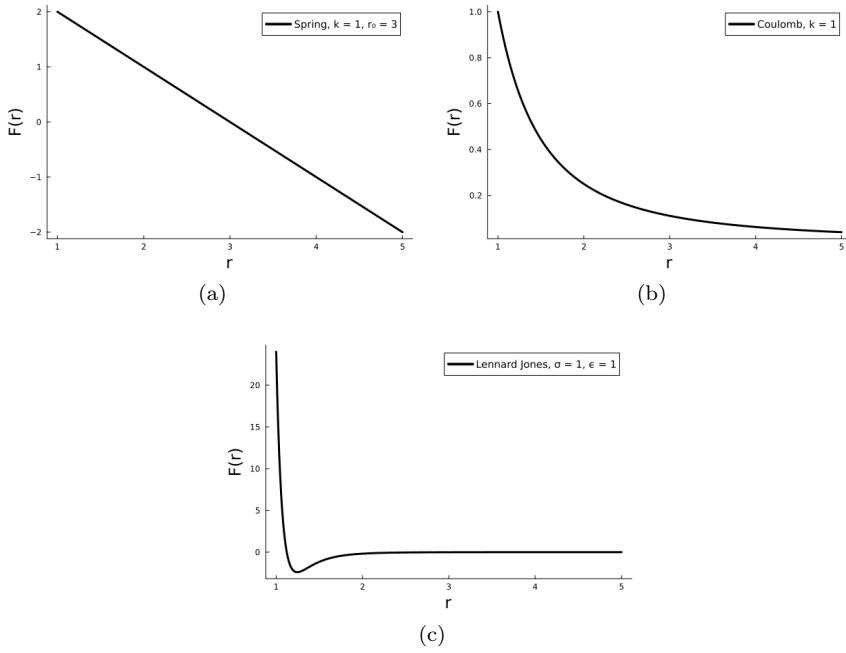


Figure 2.1: Tested interaction forces for given values of their parameters.

2.3.3 Interactions 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 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, it is possible for the magnitude of the force between two particles to be less than the machine epsilon. 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 2.2. This is of paramount importance for speed. While a complete

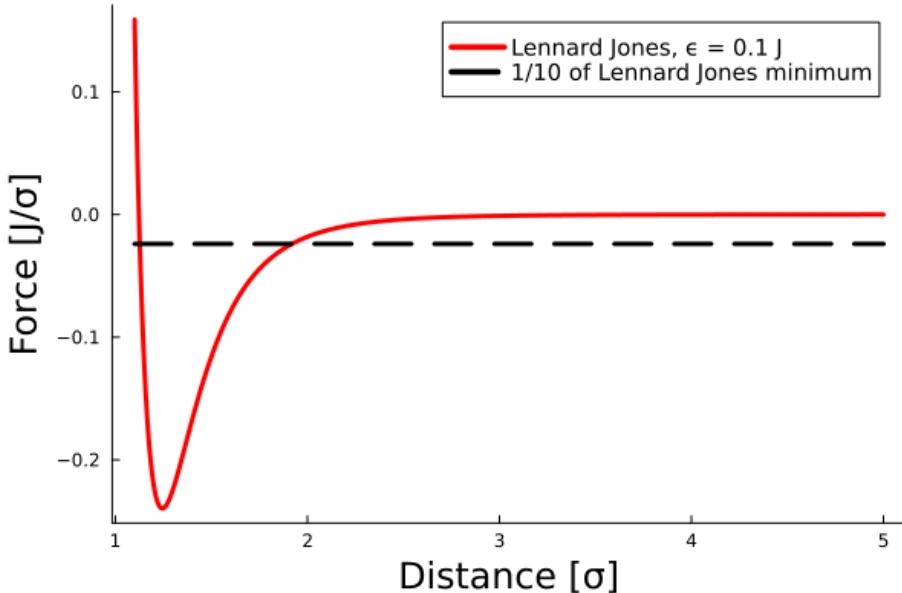


Figure 2.2: Cutoff for Lennard Jones force at 1/10 of the minimum. The cutoff distance is $\sim 1.9\sigma$

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. Original code had a function to reflect particles in periodic boundary conditions, which worked in a very fast vectorized way (pseudo code in algorithm 4). 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 2.3.

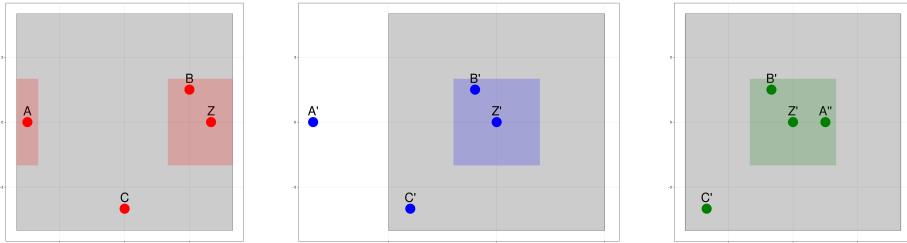


Figure 2.3: Red: original points with Z as central point and its interaction box sticking out on the opposite side. Blue: shifted points w.r.t. Z. Green: shifted points with PBC applied.

2.3.4 Aligning Interactions

There are several examples in literature of aligning interactions applied to ABPs systems [14, 4]. 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 tried before to implement an interaction in the system which unifies the central force and the torque.

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. If one was to apply a central force to a set of particles, the way would be computing a distance matrix between the positions of particles' centers, calculating the absolute value of the force with a function $F(r)$, and applying it to particle positions, splitting the two components (we are simulating in 2D). 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:

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

where θ is the self-propulsion direction, R is the particle's radius, and $-1 < \alpha < 1$. This way distances, and hence forces, are computed starting from off center positions; applying forces to such positions results both in a translation effect and in a torque applied to the particle, coupling the two degrees of freedom.

The idea is to take into account the underlying particle's asymmetry in simulations. 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 [28]. This can be due to a plethora of effects like chemical gradients, electrical forces, hydrodynamics or fabrication defects, but this model takes only the asymmetry and uses it along with standard potentials, to create a minimal dry framework

that mimics wet dynamics. In doing so, we also show that this model displays some interesting statistical mechanics properties.

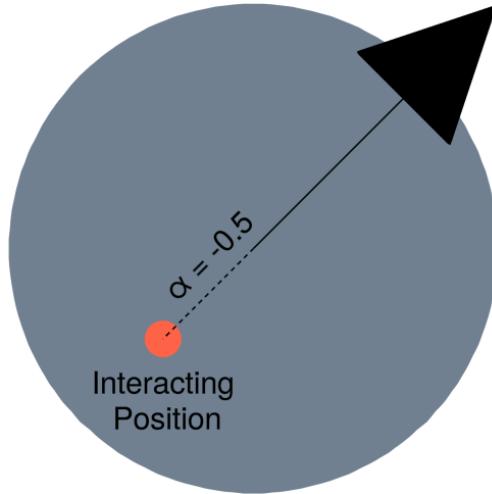


Figure 2.4: Red point: position from which interaction distances are calculated and to which force is applied. In this case, interacting position is at $0.5R$ from particle's center, in the direction of self propulsion, to the back.

2.4 Boundary Conditions and Hard Sphere Correction

2.4.1 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 just doing nothing, 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:      $y_i \leftarrow y_i - \text{sign}(y_i)L$ 

```

2.4.2 Hard Sphere Correction

Interactions among particles were steric hard sphere correction, which are enough to study clustering, MIPS and boundary accumulation. The hard sphere correction follows algorithm 5 according to [4].

Algorithm 5 The hard sphere 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:      $\mathbf{r}_j \leftarrow \mathbf{r}_j - \mathbf{n}_{j,i} d_{i,j} / 2$ 

```

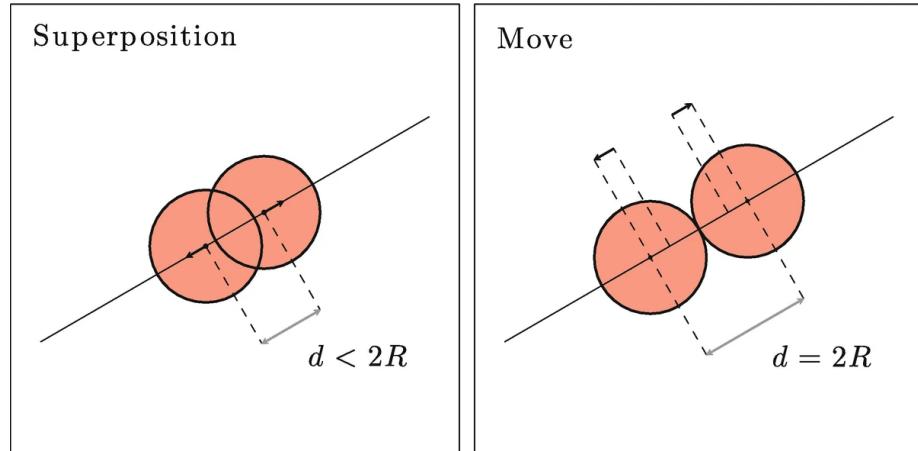


Figure 2.5: If in one step two particles superimpose, hard sphere correction moves them at one diameter distance [4]

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 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 sphere correction, the steep potentials involved require a much smaller time step (1 or 2 orders of magnitude) hence, 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 relatively easy to state collisions' elastic or inelastic nature. Furthermore, in Microscale Robotics Lab and in Literature [28] Janus active particles collisions are observed to have a finite contact time $\sim s$. This long collision time encouraged us to implement inelastic collision, where adhesion forces keep particles together after the impact.

2.4.3 Modified Periodic Boundary Steric Interactions

Since hard sphere 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 sphere correction as it was described in section 2.4.2, 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 sphere 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 sphere correction as in section 2.4.2 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.

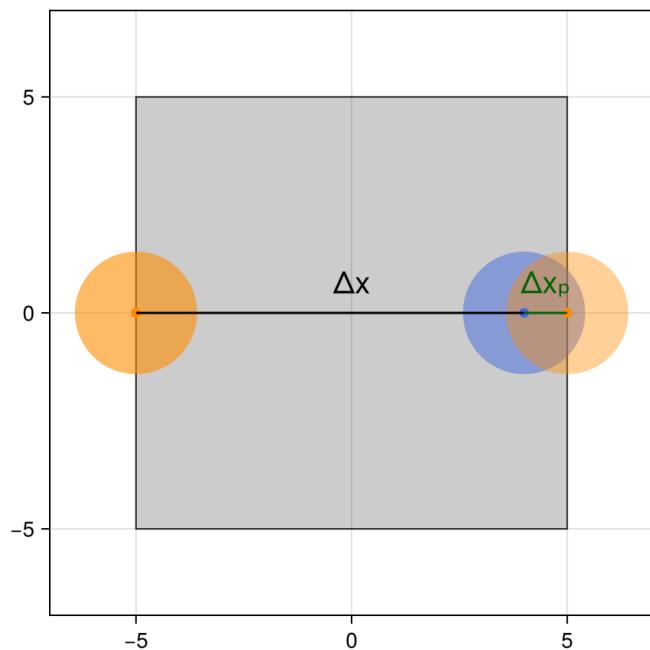


Figure 2.6: Orange particle is sticking out on the left side. Its *real* distance from blue particle on x-axis is Δx . Its periodic projection (faded orange) has a distance from blue particle $\Delta x_p < 2R$, hence superimposing.

2.5 Qualitative comparison between experiments and simulations

Chapter 3

Analysis of Structure and Dynamics

Simulations of ABPs ensembles offer an interesting possibility: they make it possible to investigate the physical systems without the limitations that an experimental setup would have. In simulations it is possible to work in a case with the same number of agents every time, keeping the physical conditions fixed, without all non-idealities, having computational power and time as the only limitation. This is crucial to study statistical properties of a system that displays fascinating behaviors.

3.1 Methods

Our investigation in structure and dynamics is based upon a small, but hopefully complete, set of tools that help unveiling how particles place themselves with respect to one another and how do they move [insert citation about 1994 Nobel, "where atoms are, what atoms do"](#).

3.1.1 Pair Correlation Function

Pair correlation function, *radial distribution function*, or simply $g(r)$ describes the variation of density with respect to distance to a reference in a system; it is a function only of the separation $r_{12} = |\mathbf{r}_1 - \mathbf{r}_2|$ between any pair of particles. It quantifies how particle in an ensemble position one with respect to each other, giving information about the *structure* of the system. This function plays a key role in physics of matter since it is possible to measure it in radiation-scattering experiments 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 [10]. Here we will use the Dirac's δ -based expression, since it is the most useful in cases where particle's positions are known,

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

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 (counting twice the same couple) accounting for periodic boundary conditions as any particle became the center of the system, similarly to the periodic interaction method in section 2.3.3. 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. Now we just normalize by the number of particles and the density to make $g(r) \rightarrow 1$ at long distance.

3.1.2 Polarization

Whenever the agents of a system have a preferred axis, as is the case for ABPs, it becomes worthwhile to analyze the order in the orientation degree of freedom of agents. This is useful especially when particles have non symmetric (in some axis) shape. Our particles are assumed to be perfectly symmetrical circles (spheres in 3D), but still, the presence of a well defined swimming direction introduces a relevant anisotropy in the system.

For global polarization, we use the following definition

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

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

3.1.3 Cluster Size Distribution

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, system clusters as a whole, leaving at most some particles in a gaseous phase outside of the cluster. Observing the evolution of such systems, nonetheless, it is possible to observe long periods of time during which ABPs are only partially clustered, moving in small groups across the simulation boards.

It is possible to extract some information about partial clusters from the $g(r)$, but, being originally created to study systems a whole, it is not the optimal tool. In order to get some insight about partial clustering, which often happens as a transient state before global clustering, we developed the cluster size distribution.

Our method was built using DBSCAN, 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. 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. Getting cluster sizes as the result of a DBSCAN passage, it becomes easy to compute the sample cluster size distribution function, that is, plotting sizes on a histogram.

3.1.4 Local Polarization

As explained in previous section, local properties are useful to observe, in particular in transients. 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, so that measuring the global polarization P would get in the same result of a disordered system, close to 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 Coulomb Flocking as a Phase Transition

As mentioned in section 1.3, Martín-Gómez et al., as well as Negi, Winkler, and Gompper 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*.

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. Here we stress the fact that when all particles in ensemble have the same velocity and clustering is not present, a head-to-head collision is by far the most frequent, and for sure a head-to-tail collision is almost impossible, since two particles with the same directions will chase each other without ever touching.

In Figure 3.1, an example of how the alignment process works. Differently from our case, the interaction in mentioned work is an explicitly aligning torque where $T \sim \sin(\theta_i - \theta_j)$, where particles orientation are *explicitly* included in the expression and $|T|$ has a clear minimum where $\theta_i = \theta_j$. Our alignment comes automatically from applying a force to a non centered position.

In Figure 3.2, an example of how flocking transition happens with $k = 10$, with an interaction range of 40 μm .

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. The range of interaction is 5 μm , 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

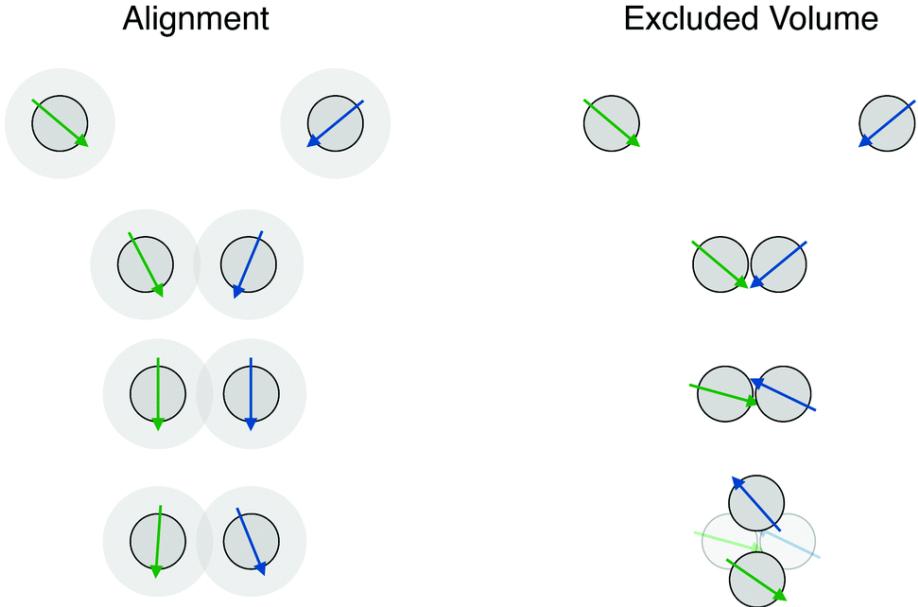


Figure 3.1: Behavior of two particles with aligning interactions, compared with excluded volume steric interaction only Martín-Gómez et al.

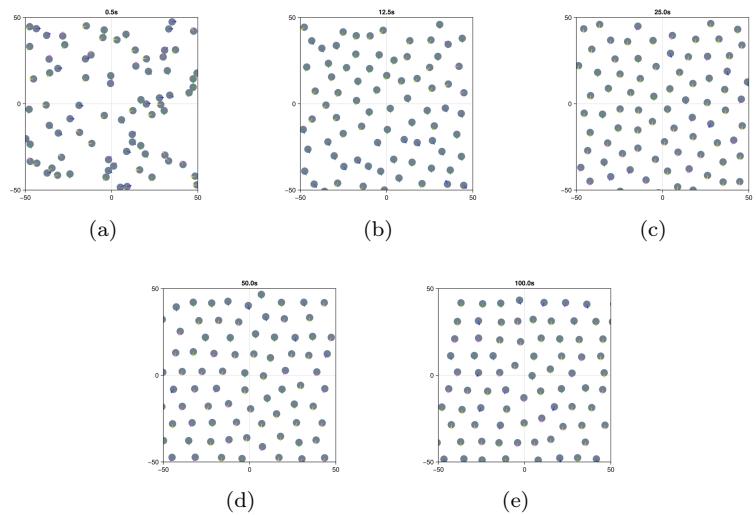


Figure 3.2: **NOT DEFINITIVE** Flocking transition: all particles in the system tend to polarize after few seconds.

range is so short, a long-range order establishes for large enough α , making the whole ensemble polarize.

As a result, Figure 3.3 shows the expected behavior of a II order phase transition: as the order parameter P gets with a discontinuity from 0 to 1, its susceptibility diverges to a peak when α approaches critical value and then returns back close to zero (see, as a reference, Figure 1.3).

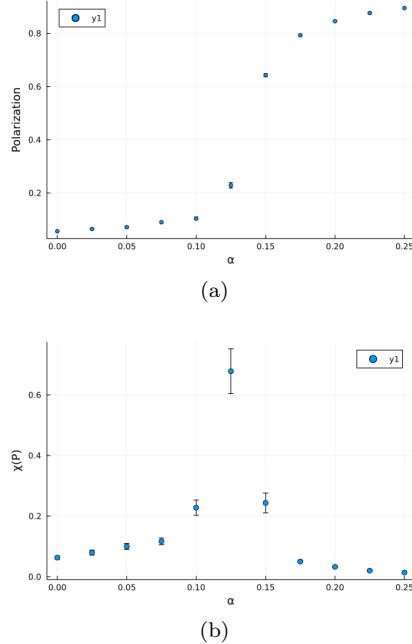


Figure 3.3: **NOT DEFINITIVE** Diagrams for polarization and relative susceptibility for the transition to ordered phase.

3.3 Aligning Lennard Jones: Clustering and Local Order

Lennard Jones (LJ) is the potential which has the best chance of representing well real world dynamics (see section 2.5), so it is worthwhile to study it in detail, to see how it affects individual and collective behaviors of ABPs ensembles. In order to do that, 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.

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. An active speed makes the system non-isotropic, 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 of it.

The interaction used is LJ with parameters $\sigma = 2R$, with R particle's radius, and strength parameter $\epsilon\sigma = 0.1J$. Magnitude of α is kept fixed at 0.5, changing the sign. Simulations involved 250 particles in a square of 175 μm side, for a resulting packing fraction of ~ 0.1 . Simulation time is 1000 s.

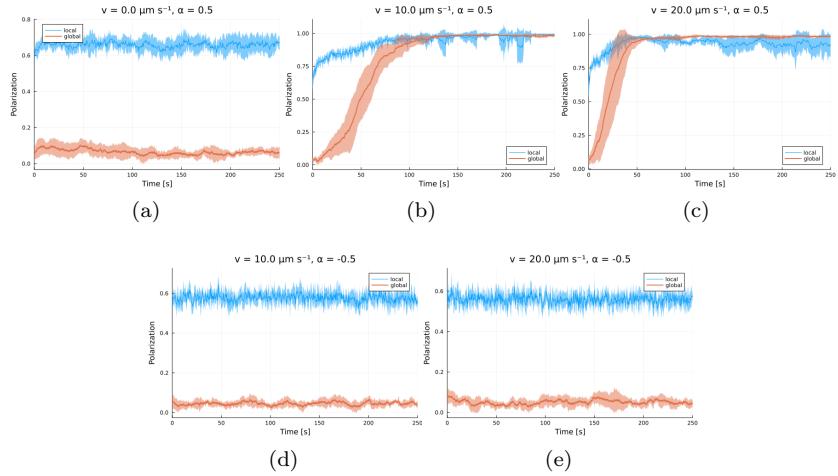


Figure 3.4: **NOT DEFINITIVE** Average of 5 simulations, strip around line is stddeviation

In Figure 3.4 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 length 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 at all.

This can also be noticed in Figures 3.5 and 3.6, where, when the off center position is on the back, velocity tends to hinder clustering, leading to a more sparse final situation. This can be due to the aforementioned fact about velocity: all particles have the same velocity so a head-to-tail collision is unlikely, as a consequence the off cen

3.3.2 Velocity Distribution

3.3.3 Angular Velocity

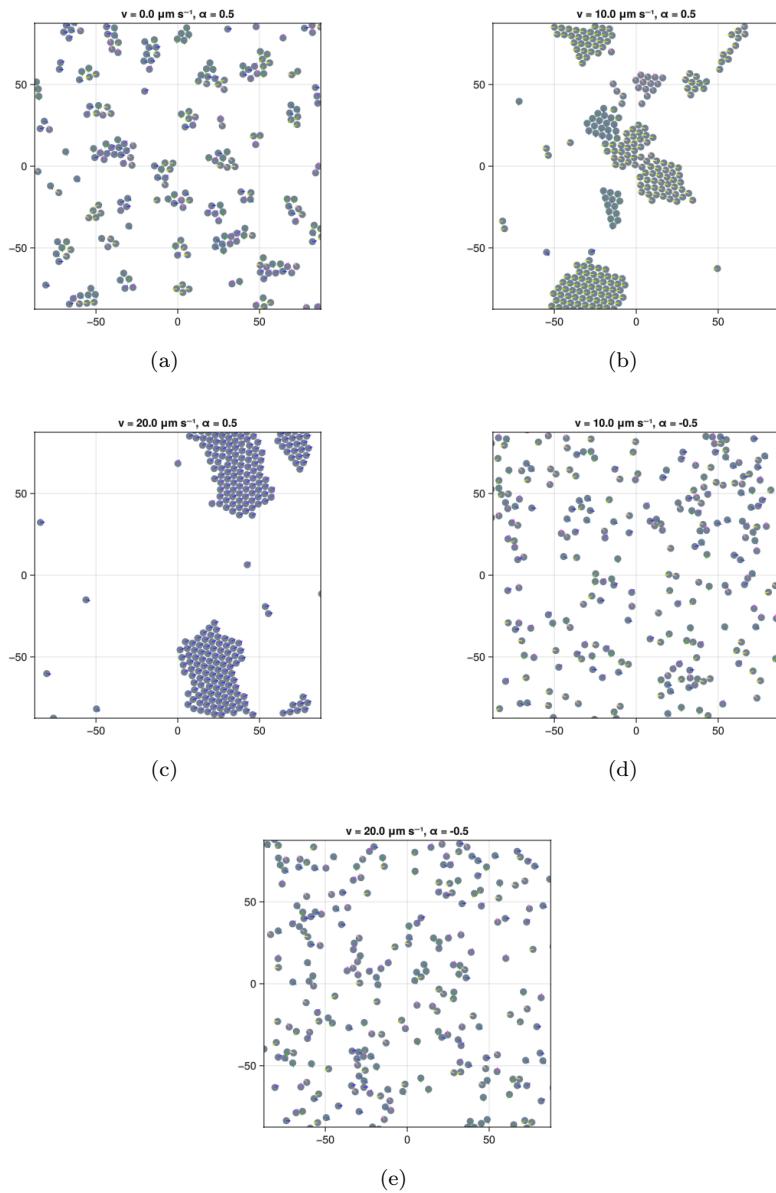


Figure 3.5: NOT DEFINITIVE

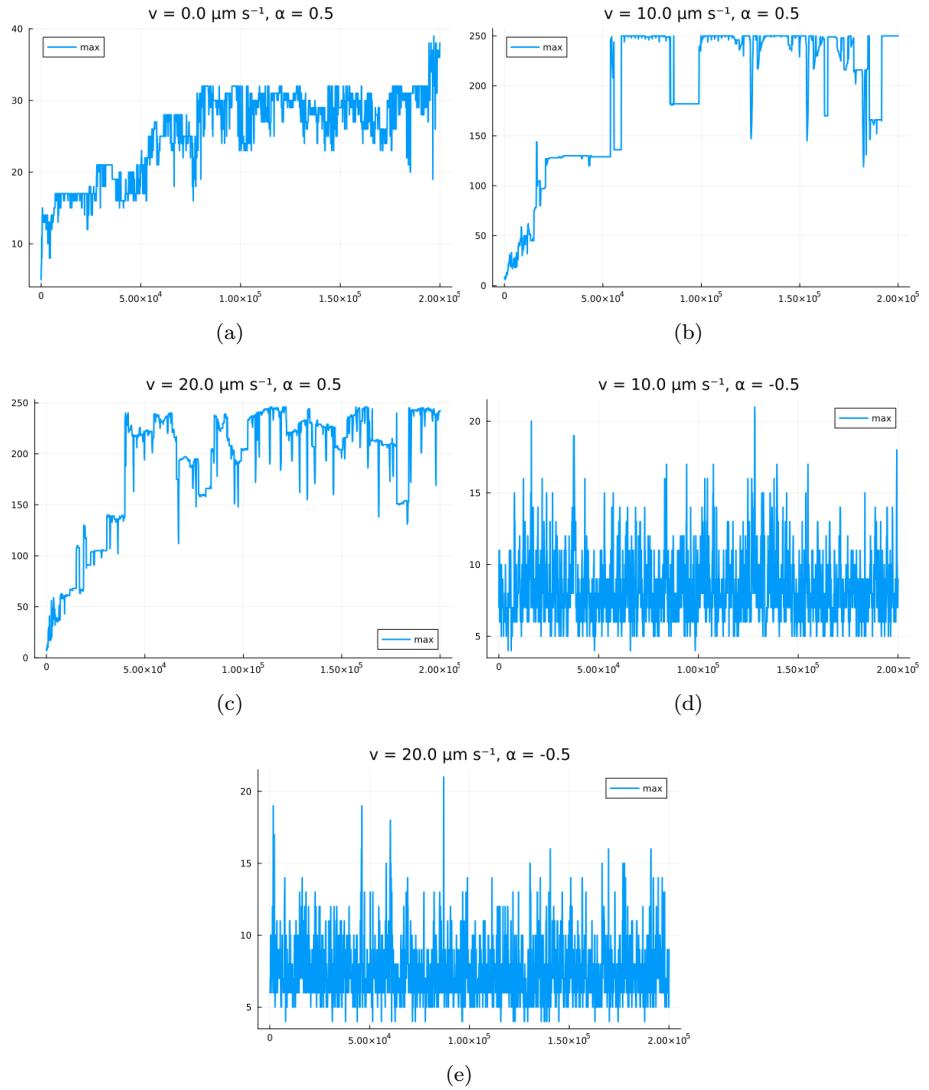


Figure 3.6: NOT DEFINITIVE

Bibliography

- [1] Saientan Bag and Rituparno Mandal. “Interaction from structure using machine learning: in and out of equilibrium”. In: *Soft Matter* 17.36 (Sept. 22, 2021). Publisher: The Royal Society of Chemistry, pp. 8322–8330. ISSN: 1744-6848. DOI: 10.1039/D1SM00358E. URL: <https://pubs.rsc.org/en/content/articlelanding/2021/sm/d1sm00358e>.
- [2] M. Ballerini et al. “Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study”. In: *Proceedings of the National Academy of Sciences* 105.4 (Jan. 29, 2008). Publisher: Proceedings of the National Academy of Sciences, pp. 1232–1237. DOI: 10.1073/pnas.0711437105. URL: <https://www.pnas.org/doi/full/10.1073/pnas.0711437105>.
- [3] Peter W. Battaglia et al. *Relational inductive biases, deep learning, and graph networks*. Oct. 17, 2018. DOI: 10.48550/arXiv.1806.01261. arXiv: 1806.01261[cs]. URL: <http://arxiv.org/abs/1806.01261>.
- [4] Agnese Callegari and Giovanni Volpe. “Numerical Simulations of Active Brownian Particles”. In: *Flowing Matter*. Ed. by Federico Toschi and Marcello Suga. Cham: Springer International Publishing, 2019, pp. 211–238. ISBN: 978-3-030-23370-9. DOI: 10.1007/978-3-030-23370-9_7. URL: https://doi.org/10.1007/978-3-030-23370-9_7.
- [5] L. Caprini, U. Marini Bettolo Marconi, and A. Puglisi. “Spontaneous Velocity Alignment in Motility-Induced Phase Separation”. In: *Physical Review Letters* 124.7 (Feb. 19, 2020). Publisher: American Physical Society, p. 078001. DOI: 10.1103/PhysRevLett.124.078001. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.124.078001>.
- [6] Andrea Cavagna et al. “From empirical data to inter-individual interactions: unveiling the rules of collective animal behavior”. In: *Mathematical Models and Methods in Applied Sciences* 20 (supp01 Sept. 2010). Publisher: World Scientific Publishing Co., pp. 1491–1510. ISSN: 0218-2025. DOI: 10.1142/S0218202510004660. URL: <https://www.worldscientific.com/doi/abs/10.1142/S0218202510004660>.
- [7] Miles Cranmer 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. URL: https://proceedings.neurips.cc/paper_files/paper/2020/hash/c9f2f917078bd2db12f23c3b413d9cba-Abstract.html.
- [8] Richard P. Feynman. “There’s Plenty of Room at the Bottom”. In: 1959. URL: <https://caltech.library.caltech.edu/1976/>.

- [9] C. W. Gardiner. *Handbook of stochastic methods for physics, chemistry and the natural sciences*. Third. Vol. 13. Springer Series in Synergetics. Berlin: Springer-Verlag, 2004, pp. xviii+415. ISBN: 3-540-20882-8.
- [10] J. P. Hansen and I. McDonald. *Theory of Simple Liquids*. London: Academic, 1990.
- [11] Samadarshi Maity and Alexandre Morin. “Spontaneous Demixing of Binary Colloidal Flocks”. In: *Physical Review Letters* 131.17 (Oct. 25, 2023). Publisher: American Physical Society, p. 178304. DOI: 10.1103/PhysRevLett.131.178304. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.131.178304>.
- [12] RICCARDO MANNELLA. “INTEGRATION OF STOCHASTIC DIFFERENTIAL EQUATIONS ON A COMPUTER”. In: *International Journal of Modern Physics C* (Nov. 21, 2011). Publisher: World Scientific Publishing Company. DOI: 10.1142/S0129183102004042. URL: <https://www.worldscientific.com/worldscinet/ijmpc>.
- [13] Riccardo Mannella and Peter V. E. McClintock. “Itô versus stratonovich: 30 years later”. In: *Fluctuation and Noise Letters* 11.1 (Mar. 2012). Publisher: World Scientific Publishing Co., p. 1240010. ISSN: 0219-4775. DOI: 10.1142/S021947751240010X. URL: <https://www.worldscientific.com/doi/abs/10.1142/S021947751240010X>.
- [14] Aitor Martín-Gómez et al. “Collective motion of active Brownian particles with polar alignment”. In: *Soft Matter* 14.14 (Apr. 4, 2018). Publisher: The Royal Society of Chemistry, pp. 2610–2618. ISSN: 1744-6848. DOI: 10.1039/C8SM00020D. URL: <https://pubs.rsc.org/en/content/articlelanding/2018/sm/c8sm00020d>.
- [15] Gautam I. Menon. “Active Matter”. In: *Rheology of Complex Fluids*. Ed. by J. Murali Krishnan, Abhijit P. Deshpande, and P. B. Sunil Kumar. New York, NY: Springer, 2010, pp. 193–218. ISBN: 978-1-4419-6494-6. DOI: 10.1007/978-1-4419-6494-6_9. URL: https://doi.org/10.1007/978-1-4419-6494-6_9.
- [16] Jeffrey L. Moran and Jonathan D. Posner. “Phoretic Self-Propulsion”. In: *Annual Review of Fluid Mechanics* 49 (Volume 49, 2017 Jan. 3, 2017). Publisher: Annual Reviews, pp. 511–540. ISSN: 0066-4189, 1545-4479. DOI: 10.1146/annurev-fluid-122414-034456. URL: <https://www.annualreviews.org/content/journals/10.1146/annurev-fluid-122414-034456> (visited on 01/21/2025).
- [17] Rajendra Singh Negi, Roland G. Winkler, and Gerhard Gompper. “Emergent collective behavior of active Brownian particles with visual perception”. In: *Soft Matter* 18.33 (Aug. 24, 2022). Publisher: The Royal Society of Chemistry, pp. 6167–6178. ISSN: 1744-6848. DOI: 10.1039/D2SM00736C. URL: <https://pubs.rsc.org/en/content/articlelanding/2022/sm/d2sm00736c>.
- [18] NumPy. URL: <https://numpy.org/>.

- [19] Tanya Ostapenko et al. “Curvature-Guided Motility of Microalgae in Geometric Confinement”. In: *Physical Review Letters* 120.6 (Feb. 7, 2018). Publisher: American Physical Society, p. 068002. DOI: 10.1103/PhysRevLett.120.068002. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.120.068002>.
- [20] *pandas - Python Data Analysis Library*. URL: <https://pandas.pydata.org/>.
- [21] E. M. Purcell. “Life at low Reynolds number”. In: *American Journal of Physics* 45.1 (Jan. 1, 1977), pp. 3–11. ISSN: 0002-9505. DOI: 10.1119/1.10903. URL: <https://doi.org/10.1119/1.10903>.
- [22] *PyG Documentation — pytorch_geometric documentation*. URL: <https://pytorch-geometric.readthedocs.io/en/latest/index.html>.
- [23] *PyTorch*. PyTorch. URL: <https://pytorch.org/>.
- [24] Sriram Ramaswamy. “Active matter”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2017.5 (May 2017). Publisher: IOP Publishing and SISSA, p. 054002. ISSN: 1742-5468. DOI: 10.1088/1742-5468/aa6bc5. URL: <https://dx.doi.org/10.1088/1742-5468/aa6bc5>.
- [25] *robot noun - Definition, pictures, pronunciation and usage notes / Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com*. URL: <https://www.oxfordlearnersdictionaries.com/definition/english/robot>.
- [26] Miguel Ruiz-Garcia et al. “Discovering dynamic laws from observations: The case of self-propelled, interacting colloids”. In: *Physical Review E* 109.6 (June 26, 2024). Publisher: American Physical Society, p. 064611. DOI: 10.1103/PhysRevE.109.064611. URL: <https://link.aps.org/doi/10.1103/PhysRevE.109.064611>.
- [27] Jyoti Sharma, Lapo Corti, and Stefano Palagi. *Simulations Active Brownian Particles v1.0*. Version v1.0. July 28, 2023. DOI: 10.5281/zenodo.8192894. URL: <https://zenodo.org/records/8192894>.
- [28] Karnika Singh et al. “Pair Interactions of Self-Propelled SiO₂–Pt Janus Colloids: Chemically Mediated Encounters”. In: *Langmuir* 40.14 (Apr. 9, 2024). Publisher: American Chemical Society, pp. 7328–7343. ISSN: 0743-7463. DOI: 10.1021/acs.langmuir.3c03415. URL: <https://doi.org/10.1021/acs.langmuir.3c03415>.
- [29] *The Julia Programming Language*. URL: <https://julialang.org/> (visited on 01/16/2025).
- [30] G. E. Uhlenbeck and L. S. Ornstein. “On the Theory of the Brownian Motion”. In: *Physical Review* 36.5 (Sept. 1, 1930). Publisher: American Physical Society, pp. 823–841. DOI: 10.1103/PhysRev.36.823. URL: <https://link.aps.org/doi/10.1103/PhysRev.36.823>.
- [31] Tamás Vicsek et al. “Novel Type of Phase Transition in a System of Self-Driven Particles”. In: *Physical Review Letters* 75.6 (Aug. 7, 1995), pp. 1226–1229. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.75.1226. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.75.1226>.