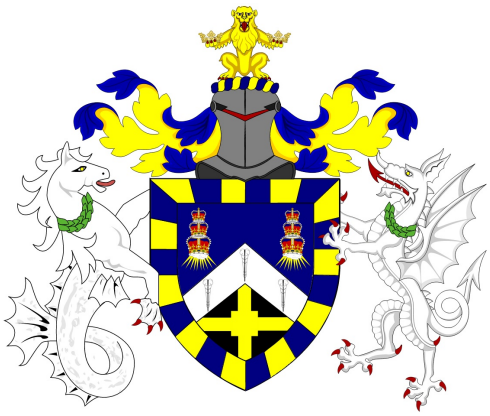
Data Analytics MSc Dissertation MTH797P, 2019/20

Collective motion of self-propelled particles

Overview of the Vicsek Model and some variations

Hadrian Jules Y. Ang, ID 190808763

Supervisor: Dr. Wolfram Just



A thesis presented for the degree of
Master of Science in Data Analytics

School of Mathematical Sciences
Queen Mary University of London

Declaration of original work

This declaration is made on September 11, 2020.

Student's Declaration: I Hadrian Ang hereby declare that the work in this thesis is my original work. I have not copied from any other students' work, work of mine submitted elsewhere, or from any other sources except where due reference or acknowledgement is made explicitly in the text, nor has any part been written for me by another person.

Referenced text has been flagged by:

1. Using italic fonts, **and**
2. using quotation marks "…", **and**
3. explicitly mentioning the source in the text.

Abstract

Collective motion is a phenomenon commonly seen in nature such as in schools of fish and flocks of birds. The Standard Vicsek Model (SVM) is a simple model of collective motion that can result in complex behaviour. In this thesis, we first demonstrate the phase transition exhibited by the SVM, running simulations with the order parameter (normalised average velocity) as a function of noise and density. We study some of the different variations in the implementation of the model from other works throughout the years such as update rules, and noise types. It is then shown that though the update rule has minimal effect on the nature of the phase transition, noise type can affect its order with extrinsic (vectorial and simple extrinsic) noise resulting in first-order transitions and intrinsic (angular) noise resulting in second-order. However, arguments from the literature compiled and expounded on here seem to show that the order of the phase transition in the SVM still cannot be definitively settled as experiments with much larger system sizes might still be required. Afterward, we discuss and recreate results of a Modified Vicsek Model proposed by Costanzo & Hemelrijk [*Journal of Physics D: Applied Physics* 51(13),134004 (2018)], which involves two new parameters, field-of-view and maximum angular velocity. We demonstrate some of the findings from this study, namely that the addition of these two parameters leads to new patterns of motion such as bands, lines, and mills. Particular attention is given to the conditions that are favourable for milling (particles moving in a circular pattern around a common centre). We show that high-density, low-noise, environments combined with a moderate field-of-view and moderate absolute and maximum angular velocities are conducive to milling.

Contents

1 Introduction **5**

1.1 Motivation for this work . . . 7

1.2 Contents of the thesis . . . 7

1.2.1 Chapter 2 . . . 8

1.2.2 Chapter 3 . . . 8

1.2.3 Chapter 4 . . . 8

1.3 Hardware and Software Notes . . . 8

2 The Standard Vicsek Model **9**

2.1 Model Definition . . . 9

2.2 On the Phase Transition . . . 10

2.2.1 Background . . . 12

2.2.2 Order and Control Parameters . . . 13

2.2.3 Simulation Results . . . 14

2.3 Order of the Phase Transition . . . 17

2.3.1 Intrinsic and Extrinsic Noise . . . 18

2.3.2 Velocity Regimes . . . 20

2.3.3 Backward vs. Forward Update . . . 28

3 The Modified Vicsek Model **31**

3.1 Model Definition . . . 32

3.2 Implementation Details . . . 34

3.2.1 Computing the Field-of-View . . . 35

3.2.2 Centre of Mass . . . 36

3.3 Model Results 39

4 Conclusions **46**

A Core Model Implementation **48**

Chapter 1

Introduction

Vicsek & Zafeiris define collective motion as the "*phenomenon occurring in collections of similar, interacting units moving with about the same absolute velocity*" [36]. Studying this phenomenon is of interest due to how often it appears in many different systems throughout nature. This is perhaps most apparent when groups of birds fly together as a flock, when fish swim in schools, or when animals move as a herd.

All these cases fit into the above description of collective motion since (a) each animal in the group is quite similar to each of the other animals, (b) the animals perceive and interact with each other, moving in a way that keeps them from simply colliding, and (c) they move together as a group with similar absolute velocity (otherwise some animals will end up separated from the group). Because these examples from nature are so apparent, some of the early work into the study of collective motion focused on animals such as birds and fish.

In 1982, Aoki published his work on a model that simulated the schooling of fish by programming each individual fish with three interaction rules: attraction, avoidance, and parallel orientation (with rules applying depending on the positions of neighbours relative to each fish) [3]. Despite each fish having limited perception (not knowing every position and heading in the school) and the absence of a leader fish, the simulated fish still moved together as a cohesive school.

In the interest of efficiently generating computer animations, Reynolds intro-

duced a computational model for the flocking of birds [31]. The model shares a few key things with the one proposed by Aoki for fish, most notably the lack of a leader and the concept of agent perception, where simulated birds do not have complete information about every other member of the flock. Instead, they interact with nearby neighbours, devising flight paths that keep them from colliding with each other while maintaining a proper velocity to keep up with the flock. Using this method from Reynolds, the animator no longer needs to directly chart the path of each bird [31]. All that is required is to program the behaviour of a general bird agent in response to its perception of the environment. What emerges then is a pattern of motion that is a good enough facsimile of birds flocking as to look pleasant when used in animated films.

Further studies have found forms of collective motion in other biological systems. Ben-Jacob et al. observed some strains of bacteria (which they referred to as *vortex morphotype*) under a microscope and found that millions of cells would move in a circular pattern around a common centre [8]. They then proposed a simulated model where particles achieved similar patterns of motion with a combination of velocity-alignment interactions and reactions to local concentrations of a secreted chemical signal.

Collective motion can be found not only in biological systems such as those mentioned earlier, but also in chemical and physical systems. For example, Schaller et al. demonstrate the emergence of collective motion in actin filaments propelled by molecular motors on a plane [32]. Through their experiments, they found that the filaments would move coherently in an ordered phase above a critical density $\rho_c \sim 5$ filaments per square micrometre, but below this critical point, filaments would simply perform random walks.

In another study, Ibele, Mallouk, and Sen found collective motion in silver chloride (AgCl) particles in deionised water when they were exposed to ultraviolet light [21]. In this case, there is interaction among particles in the form of ions that they secrete, the concentrations of which influence the motion of other particles, causing a “schooling” type of behaviour [21].

Proposed by Vicsek et al. in their pioneering paper, the Standard Vicsek Model (SVM) is one of the simplest models that allows for collective motion [35]. This

model shares many features with the earlier mentioned models from Reynolds and Aoki: (a) the lack of a leader, (b) the programming of behaviour rather than an explicit path of motion for agents, and (c) the implementation of agent perception and interaction. Despite the simplicity of the model, it can at times exhibit complex behaviour. Most notably, it features a transition from an ordered to a disordered phase as a function of multiple parameters.

Due to its simplicity, the Vicsek Model also serves as a good foundation for extensions that may augment its behaviour and bring about new results. Costanzo & Hemelrijk propose a Modified Vicsek Model (MVM) that allows for new patterns of motion, while only introducing a few new parameters and without adding new interaction rules [15]. This thesis will explore different properties and behaviours of the SVM and MVM under a variety of conditions by presenting and expounding upon numerical results from simulations of the models.

1.1 Motivation for this work

Due to collective motion being commonplace in nature, the Vicsek Model and its variants may often serve as a starting point to better study some real-world systems. Yet, despite the apparent simplicity of the model definition (to be formally defined in later chapters), the Vicsek Model can behave in complex ways. Because of this, quite a number of studies have been conducted on the model with a variety of implementations that have at times given different results [5, 19, 29]. It may be of value then to review the model and its variations, while creating a dialogue between different, sometimes conflicting sources regarding the model.

1.2 Contents of the thesis

The remainder of this thesis will be split into three other chapters, the first focusing on the SVM (as defined in [35]), the second on the MVM (based on [15]) followed by a concluding chapter summarising the work done.

1.2.1 Chapter 2

Chapter 2 will expand on the SVM, first giving a formal definition of the model and the order parameter of the disordered-ordered phase transition that it exhibits. Simulation results are then presented, including the order parameter as a function of other model parameters. Afterward, some variations on the standard model are tackled and their impact on simulation results is shown.

1.2.2 Chapter 3

Chapter 3 zooms into the MVM, first giving a formal definition and highlighting in what way it is different from the SVM. Key implementation details are then briefly explained. Particular focus is then given to the phenomenon of milling, how it is quantified, and the proportion of milling cases as a function of different parameters.

1.2.3 Chapter 4

Chapter 4 will summarise the work done and present conclusions made using the results from experiments.

1.3 Hardware and Software Notes

Simulation results in the proceeding sections were generated with code written mostly in the Julia programming language (and Python for some visuals) with the use of some software packages [10, 11, 12, 20, 26, 27, 28, 30, 34, 37] on standard laptop hardware: an AMD Ryzen 7 3750H CPU and 16GB of RAM. Because of this, there are limitations with regard to the size of the simulations and the length of time the systems can be evolved. When systems with large numbers of agents or particles are involved, it is usually asymptotic behaviour (the number of agents $N \to \infty$ and $t \to \infty$) that is of interest. Even at smaller scales, however, some of the properties of the Vicsek Model can still be studied (though with the awareness that finite size effects may affect results).

Chapter 2

The Standard Vicsek Model

The Standard Vicsek Model has been the topic of numerous studies throughout the years. This has spawned different variations in the implementation and definition of the model. While it is often argued that these changes do not impact the asymptotic results of the simulations, some of them appear to do so (as will be tackled later in this chapter). We focus first on the original model in the paper by Vicsek et al. [35] then tackle these said variants later on.

2.1 Model Definition

The Standard Vicsek Model is a simple model for the motion of self-propelled particles (SPPs) defined in [35]. At time $t = 0$, N particles are randomly placed on a two-dimensional $L \times L$ space with periodic boundary conditions, each with some direction of motion $\theta_i(0)$ selected uniformly at random from the interval $[-\pi, \pi]$. They then move off-lattice with the same absolute velocity v_0. There are three free-parameters to the model: (1) the density of particles $\rho = N/L^2$, (2) the level of noise in the system η, and (3) the absolute velocity v_0 [35].

After each time step of size Δt, particles move based on their current direction of motion. The new position is then found with the equation

$$x_i(t + \Delta t) = x_i(t) + \mathrm{v}_i(t)\Delta t, \tag{2.1}$$

where $\mathrm{v}_i(t)$ is the velocity of particle i at time t given by

$$\mathrm{v}_i(t) = v_0 \begin{bmatrix} \cos(\theta_i(t)) \\ \sin(\theta_i(t)) \end{bmatrix}. \tag{2.2}$$

Each particle i interacts with others by aligning its direction of motion with its neighbours, defined as all particles within some interaction radius r (including the particle i itself). We assume here, as is standard in the literature, that $\Delta t = 1$ and $r = 1$ [4, 19, 29, 35]. Let $\langle\theta(t)\rangle_{r,i}$ be the mean direction of motion of all particles within the radius r around particle i given by the following equation

$$\langle\theta(t)\rangle_{r,i} = \arctan\left(\frac{\langle\sin(\theta(t)\rangle_{r,i}}{\langle\cos(\theta(t)\rangle_{r,i}}\right). \tag{2.3}$$

Given this, the new direction of motion is updated for all particles simultaneously according to

$$\theta_i(t + \Delta t) = \langle\theta(t)\rangle_{r,i} + \Delta\theta_i, \tag{2.4}$$

where $\Delta\theta_i$ is a noise term generated uniformly at random from the interval $[-\eta/2, \eta/2]$ [15, 35]. Figure 2.1 shows particle positions and velocities resulting from Vicsek Model simulations.

2.2 On the Phase Transition

One of the most interesting behaviours of the Vicsek Model is the phase transition from disordered to ordered motion that it exhibits. In the disordered phase, particles move about randomly, while in the ordered phase, almost all particles have just about the same direction of motion [35]. In this section, we first provide some background on phase transitions and briefly go through a few examples. Afterward, we show results on the phase transition exhibited by the Vicsek Model.

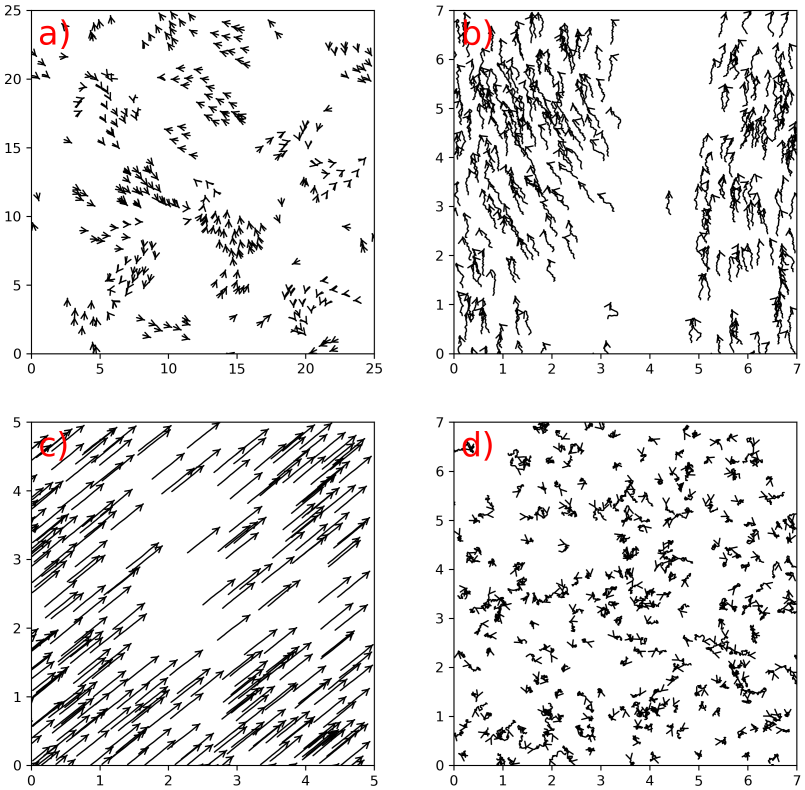


Figure 2.1: Positions of particles on a 2D plane with arrows indicating the direction of motion at $t = 100$ and the tails representing positions in the last 20 time steps. To make the images cleaner, tails were cut off if particles passed through the edges of the frame. Values of the parameters used here are similar to those in [35]. In all cases, $N = 300$ and $v_0 = 0.03$. (a) If the density of particles is relatively small and the noise is low, particles form small flocks, each moving in its own direction ($L = 25$, $\eta = 0.1$). (b) For moderately dense cases with moderate noise, ($L = 7$, $\eta = 2.0$), the particles still move toward the same general direction, but in a rather turbulent manner as shown by the way the tails are shaped. (c) In high-density, low-noise cases ($L = 5$, $\eta = 0.1$), the Vicsek Model exhibits an ordered state where all particles have almost the same direction of motion. (d) For cases with very high noise ($L = 7$, $\eta = 5.0$), particles move in random directions (resulting in velocities that often cancel each other out).

2.2.1 Background

Systems of interacting units may at times undergo a transition into another phase with markedly different properties as a result of some other external parameters [36]. These phase transitions may be seen throughout Physics, such as in matter changing from a liquid to a gaseous phase. When some water moves from a liquid to a gaseous phase, the system is still made up of the same water molecules, but taken as a whole, it now behaves in a very different way.

In general, these transitions involve an order parameter, a normalised value representing the degree of order in the system that moves from a non-zero value (indicating an ordered state) to a zero value (indicating a disordered state) as a function of a control parameter [9, 24, 36]. This transition then occurs when the control parameter passes some critical value (sometimes called the critical point). The transition may also be described as first-order (the order parameter is a discontinuous function of the control parameter) or second-order (the order parameter is a continuous function of the control parameter, but the derivative with respect to the control parameter is discontinuous) [33, 36].

Phase transitions do not happen exclusively in thermodynamic systems. Without going too deep into the mathematics, we draw two examples from Network Science based on lecture notes from Bianconi to illustrate this concept [9].

Example 1: Node Percolation An uncorrelated network is a network where the probability an edge selected uniformly at random is connected to some node j is given by $q_j = k_j/(\langle k \rangle N)$, where k_j is the degree of j, $\langle k \rangle$ is the mean node degree in the network, and N is the number of nodes [9].

Suppose then that G is an uncorrelated network and each of the N nodes has a probability p of being damaged (the node and all its edges being removed from the network). This particular system exhibits a second-order phase transition with p as the control parameter, and an order parameter S, the fraction of nodes that are part of the giant component (a connected component containing a finite fraction of nodes in the limit $N \to \infty$) [9].

There is some critical value of the control parameter that becomes the point of the phase transition. If $p > p_c$, too many nodes are damaged and there is no

giant component, resulting in a network made up of numerous small components, but if $p < p_c$ there exists a giant component. Given this, the network then has a very different topology and thus very different properties depending on how much it is damaged (based on the value of p).

Example 2: Epidemic Outbreaks The SIR (Susceptible - Infected - Removed) Model is a simple model of the epidemic spreading process. Suppose there are N individuals that can be in one of three states. Susceptible individuals become Infected by a given disease at some rate β and Infected individuals become Removed at some rate μ [9]. Once an individual is Removed, they can no longer be infected again.

This system exhibits a phase transition with control parameter $\lambda = \beta/\mu$, sometimes called the spreading rate of a disease, and order parameter ρ, the fraction of individuals that are not Susceptible (have become Infected or Removed) [9]. If λ is less than some critical value λ_c then only an infinitesimally small number of individuals are infected in the large population limit ($N \to \infty$). On the other hand, if $\lambda > \lambda_c$ then there is an epidemic breakout and the virus infects a finite fraction of individuals in the same limit. Given this, the state of the individuals in the system can be vastly different depending on which phase it is in.

2.2.2 Order and Control Parameters

As mentioned earlier, the Vicsek Model also exhibits a phase transition. Based on the original model from [35], the order parameter of this transition is the normalised average velocity of the particles given as

$$v_a = \frac{1}{Nv_0}\left\|\sum_{i=0}^{N} \mathrm{v}_i\right\|. \tag{2.5}$$

In the disordered phase, the particles will be moving in random directions, thus the value of v_a will be approximately zero. When the system is in the ordered phase, the value of v_a will be close to one since almost all particles will be moving in about the same direction.

Depending on the system size, the time required for the order parameter to converge may vary. Based on Figure 2.2, for fixed densities and noise, the amount

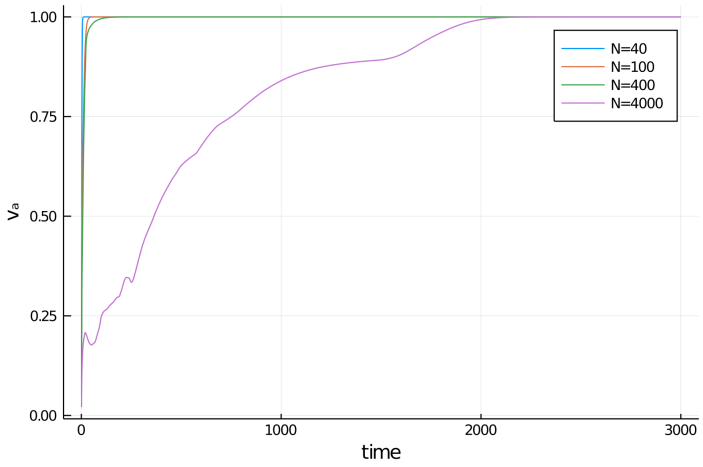


Figure 2.2: Value of the order parameter over time for $\rho = 4$, $v_0 = 0.03$, and $\eta = 0$. Note that since there is no noise in this system, it is expected that almost all particles will have about the same direction of motion, resulting in a normalised average velocity of one.

of time required for the order parameter to converge increases. This may be due to the increased number of interactions between particles required for a uniform direction of motion to be transmitted across the entire system. For the following results to be presented, we evolve the systems for as long as possible depending on their size, but due to the earlier mentioned hardware limitations and often slow convergence, there may still be small fluctuations in the results.

2.2.3 Simulation Results

The phase transition may more easily be viewed in terms of one of two different control parameters at a time, the density of particles ρ or the noise level in the system η (fixing the other one in the process). First, we view the order parameter as a function of η, relaxing the system until $t = 2 \times 10^4$ for each value of η in the interval $[0, 6]$, in increments of 0.1. The order parameter is computed at each time

step and then the average over time is taken to provide more stable results. This experiment is repeated for different values of N and results are displayed in Figure 2.3.

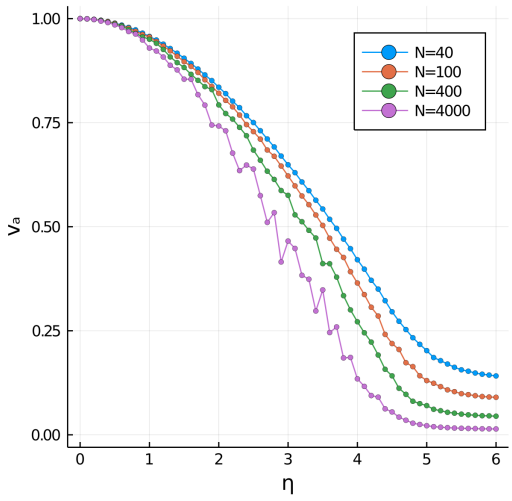


Figure 2.3: The order parameter (v_a) by noise (η) for a fixed density $\rho = 4$ and various values of N with $v_0 = 0.03$ (with similar parameters to experiments in [35]). Note that for smaller systems, the value of v_a does not truly become zero, but by increasing system sizes, one can see a trend and infer the asymptotic behaviour of the system as $N \to \infty$. The order parameter goes from a non-zero value to a zero (or close to zero) value as a function of the control parameter, exhibiting a phase transition in the process.

Another perspective would be to view the transition in terms of density by fixing the noise level η and observing v_a as a function of ρ. This was simulated by having a constant value for L with the number of agents N steadily increasing after every 8000 time steps. The time average of the order parameter for each specific value of ρ is then taken to allow for more stable results displayed in Figure 2.4.

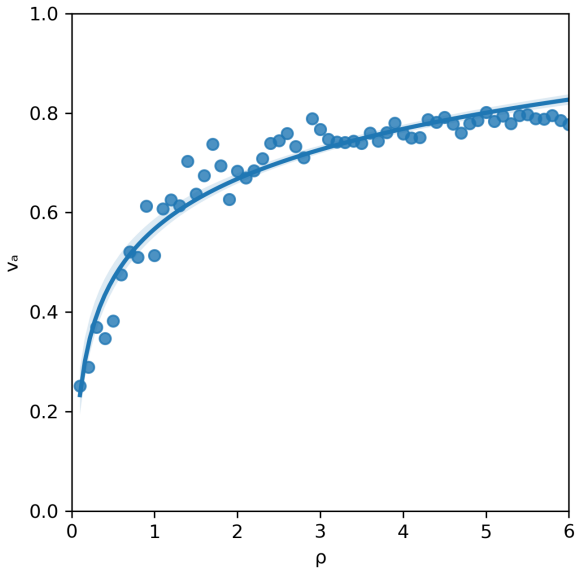


Figure 2.4: The order parameter (v_a) as a function of density (ρ) fitted with a logarithmic curve with $L = 20$ and $\eta = 2.0$ (with similar parameters to experiments in [35]). In low-density cases, particles may be very far apart and have no chance to interact. Due to the lack of opportunities to align themselves, particles may be travelling in different, random directions. As density increases, so do the opportunities for interaction. This allows the information on a uniform direction of motion to be transmitted throughout the system.

2.3 Order of the Phase Transition

While the existence of the phase transition in the Vicsek Model is generally agreed upon, there have been contentions in the past with regard to the order of this transition. Based on the results from the previous section, the phase transition appears to be second-order (evidenced by the continuous nature of the curves in Figure 2.3), which is in agreement with the original study done by Vicsek et al. [35]. However, a study by Grégoire & Chaté claimed that the transition was instead first-order in nature, arguing that initial results by the group of Vicsek were caused by finite size effects as the simulations were not run for large enough system sizes to capture the asymptotic behaviour of the system [19].

Part of the difficulty surrounding this issue is the number of parameters of the Vicsek Model and variations in implementations throughout the years. This section will go through arguments on both sides: that the transition is first-order (mainly by the group of Chaté, Grégoire, Ginelli, and Raynaud across various studies) [14, 18, 19] and that the transition is second-order (supported by different authors throughout the years) [2, 5, 29, 35].

Different studies have introduced variations to the original implementation of the Vicsek Model from [35]. Some of these changes are simple and will not affect results. For example, some studies use an alternate definition of the noise η, having $\Delta\theta \in [-\eta\pi, \eta\pi]$ instead of $\Delta\theta \in [-\eta/2, \eta/2]$ as in the original model [1, 4, 29]. This change can be seen as a simple rescaling of the noise variable and it is not too difficult to convert between the two definitions.

Some modifications to the model are larger. In their study, Czirók, Stanley, and Vicsek implement a Vicsek-like Model where instead of particles interacting with neighbours within an interaction radius r, the $L \times L$ space is divided into a lattice of $r \times r$ tiles and particles interact with those in the 9 neighbouring tiles (including its own) [16]. Aldana et al. implement a Vicsek-like Model where instead of particles moving in a 2D space, there are instead N nodes, each representing a 2D vector indicating a velocity that may change through time [1]. Each node then interacts with K other random nodes in the network with the same alignment rules as in the Vicsek Model. The authors argue here that this networked model is able to simulate the correlations generated by particle motion and interaction in the

Vicsek Model since the K neighbours of each node are randomly chosen [1].

Since these modifications are not as recurrent in the literature though, we do not tackle them further or recreate them in the simulations. Instead, the remainder of this section will focus on some of the more common variations in the Vicsek Model: namely the noise type, velocity regime, and update rule.

2.3.1 Intrinsic and Extrinsic Noise

Recall that in Equation 2.4, noise is added to the computed mean angle of neighbouring particles within the interaction radius with the equation reiterated below

$$\theta_i(t + \Delta t) = \langle \theta(t) \rangle_{r,i} + \Delta\theta_i.$$

This form of noise would later be called "*angular noise*" in studies from Chaté et al. and Baglietto & Albano [5, 14] and at times it has been labelled "*intrinsic noise*" since the uncertainty lies with the required for the particle to update its heading [1, 36].

In their work, Grégoire & Chaté introduced a new form of noise called "*vectorial noise*" different from the one found in the original Vicsek Model [19]. They argued that noise could be found in particle to particle interactions (caused by errors in perception, for example) rather than in a particle's ability to reorient itself to a perfectly measured mean heading. In the vectorial noise scenario, an average of velocities of neighbours within the interaction radius r is first computed, and then it is perturbed by a random vector scaled according to the noise level $\eta \in [0, 1]$ before the angle of the new heading is computed [14, 36]. The equation for new directions of motion is then given as

$$\theta_i(t + \Delta t) = \text{Angle}\left(\sum_{j \in S_i} u_j(t) + \eta \mathcal{N}_i \xi_i \right) \tag{2.6}$$

where S_i are the particles within the interaction radius of i (including i itself), $u_j(t)$ is a unit vector in the direction of $\theta_j(t)$, $\mathcal{N}_i$ is the number of particles in S_i, ξ_i is a random unit vector, and $\text{Angle}(\vec{v}) = \arctan(\vec{v}_y, \vec{v}_x)$. Note that this is quite a large change to the original model since the impact of the perturbation

introduced by vectorial noise in local areas where there is disorder (thus having a small average velocity) is greater than in areas with strong local alignment (thus having a large average velocity) [5, 19, 36].

Vectorial noise has at times been called "*extrinsic noise*" since extrinsic factors such as the environment may affect the interaction or communication between the particles (thus causing the noise) [2, 36]. We introduce here a kind of extrinsic noise we call *simple extrinsic noise* (SEN), that is more of a direct translation of noise in the original Vicsek Model to an extrinsic noise scenario based on the idea from Grégoire and Chaté on noise being in interactions between particles [19]. Applying SEN, the new direction of motion for particles is given as

$$\theta_i(t + \Delta t) = \langle \theta(t) + \Delta\theta \rangle_{r,i}. \tag{2.7}$$

This means noise is added not to the final computed mean angle, but to every neighbouring direction of motion used to compute it. This may also be written as

$$\langle \theta(t) + \Delta\theta \rangle_{r,i} = \arctan\left(\frac{\langle \sin(\theta(t) + \Delta\theta) \rangle_{r,i}}{\langle \cos(\theta(t) + \Delta\theta) \rangle_{r,i}} \right), \tag{2.8}$$

where $\Delta\theta$ is some number selected uniformly at random from the interval $[-\eta/2, \eta/2]$ (just like the angular noise case stated earlier). While the following discussion will focus mostly on vectorial and angular noise since these have been well studied in the past, we present the results of this noise type for comparison later on.

Aldana et al. argue that there is no reason to expect similar results from the two different types of noise since uncertainty in intrinsic and extrinsic noise lie in different places [1]. While studies would maintain the second-order nature of the phase transition in the angular noise case [1, 4, 29], it was also found that vectorial noise results in a first-order transition [2, 5].

Work from the group of Chaté, Grégoire, Ginelli, and Raynaud claimed, however, that the noise type should not actually change the order of the phase transition and that both cases of vectorial and angular noise result in discontinuous, first-order transitions [13, 14, 18, 19]. According to Chaté et al., simulations that showed a continuous transition for angular noise were run on system sizes that were too small, arguing that there is a minimum system size L^*, which they called

the *crossover size*, beyond which the transition becomes discontinuous [14]. Thus in order to capture the true asymptotic behaviour of the system the model should be run for $L > L^*$ (though this is difficult to compute and is a function of multiple parameters) [14].

Later results would contest this claim, showing that though vectorial noise results in a first-order transition, there may be numerical artifacts with the implementation of simulations causing the appearance of a discontinuous transition when angular noise is applied (to be discussed more later) [5, 29].

We then proceed to present the results of our simulations to compare and contrast the different forms of noise. One quantity often used to study the order of a phase transition is the Binder Cumulant, G given as

$$G = 1 - \frac{\langle \varphi^4 \rangle}{3\langle \varphi^2 \rangle^2}, \tag{2.9}$$

where φ here is the order parameter of the phase transition (making use of v_a for the Vicsek Model case) [25, 29]. This value is sensitive to fluctuations in the associated order parameter and a definite minimum as a function of the control parameter is usually seen as characteristic of a first-order transition [19, 29]. Expressing the order parameter v_a as a function of noise η (for the noise types tackled here) and computing the corresponding Binder Cumulant yields the results in Figure 2.5 (vectorial noise), Figure 2.6 (simple extrinsic noise), and Figure 2.7 (angular noise).

Based on our simulations, both kinds of extrinsic noise discussed (vectorial and simple extrinsic noise) seem to result in a first-order transition, while angular noise results in a second-order one. These conclusions on the order of the transition in the vectorial and angular noise scenarios agree with the results found by Nagy, Daruka, and Vicsek [29], and Baglietto & Albano [5], but runs counter to the conclusions from Chaté et al [14].

2.3.2 Velocity Regimes

The absolute velocity v_0 may be interpreted as a parameter that controls the frequency of interactions between particles [6]. In cases where the value of v_0

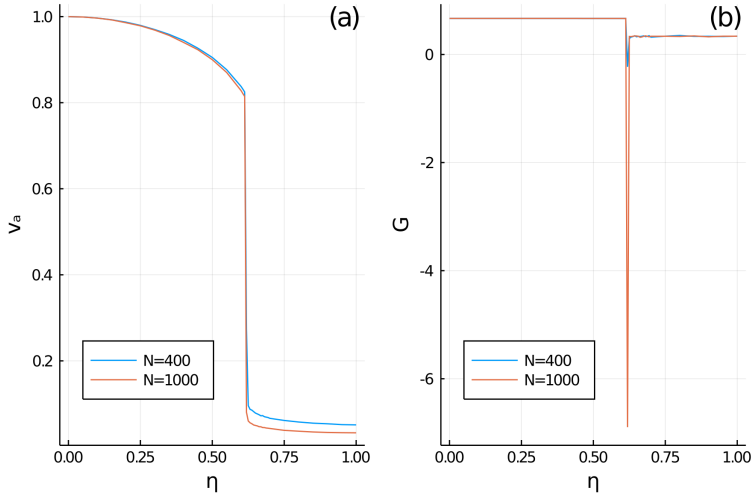


Figure 2.5: (a) The order parameter (v_a) as a function of noise (η) in the vectorial noise scenario ($\rho = 4$, $v = 0.5$). (b) The corresponding Binder Cumulant G as a function of noise η. The transition here appears to be a clear first-order transition characterised by the definite minimum in the Binder Cumulant and the order parameter changing discontinuously with η.

becomes large relative to the value of the interaction radius r, particles may pass by each other without actually interacting. Given the default settings of $r = 1$ and $\Delta t = 1$, the large-velocity regime is generally given as $v_0 >= 0.3$, with the small-velocity regime being $v_0 < 0.3$ [35].

Simulations show that within the small-velocity regime, the actual value of v_0 does not matter much as the results (summarised in Figure 2.8) are very similar across different input values. This agrees with the original conclusion by Vicsek et al. [35].

As mentioned previously, Chaté et al. argued that even in cases when angular noise is applied, there is a first-order phase transition, and this asymptotic behaviour of the system is only visible if $L > L^*$ as finite size effects would affect the results otherwise [14]. As evidence of a discontinuous transition, they show the probability density function (PDF) of the order parameter as a two-peaked distribution. On the other hand, Nagy et al. claim it is instead a second-order transition, arguing that the PDF of v_a only has a single peak [29]. Results of

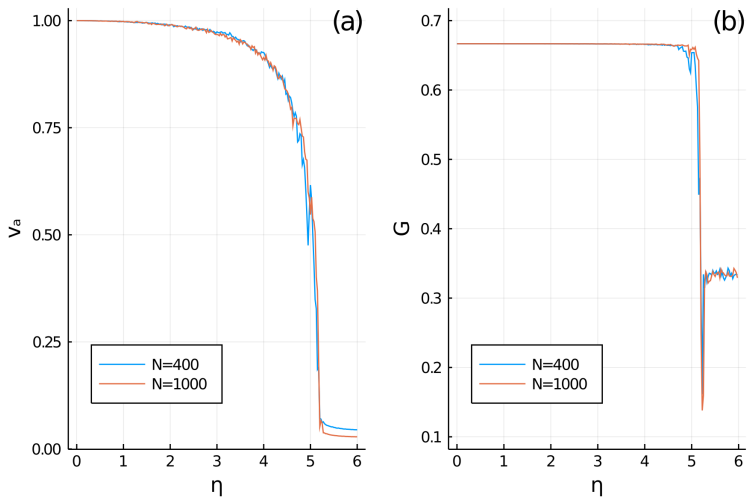


Figure 2.6: (a) The order parameter (v_a) as a function of noise (η) in the SEN scenario ($\rho = 4$, $v = 0.03$). Note that despite making use of the same parameters as Figure 2.7, results here look very different due to a discontinuous transition. (b) The corresponding Binder Cumulant G as a function of noise η. There is a clear minimum in the computed Binder Cumulant, which is usually associated with first-order transitions.

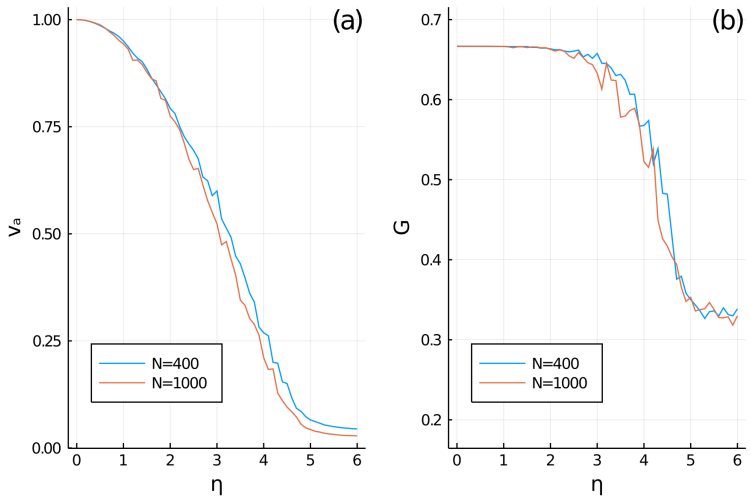


Figure 2.7: (a) The order parameter (v_a) as a function of noise (η) in the angular noise scenario ($\rho = 4$, $v = 0.03$, the same parameters as the simple extrinsic noise simulation with results in Figure 2.6). (b) The corresponding Binder Cumulant G as a function of noise η. Based on the results from these two plots, the phase transition appears to be second-order, as indicated by the smooth curve through the critical point in (a) and the lack of a definite minimum in (b).

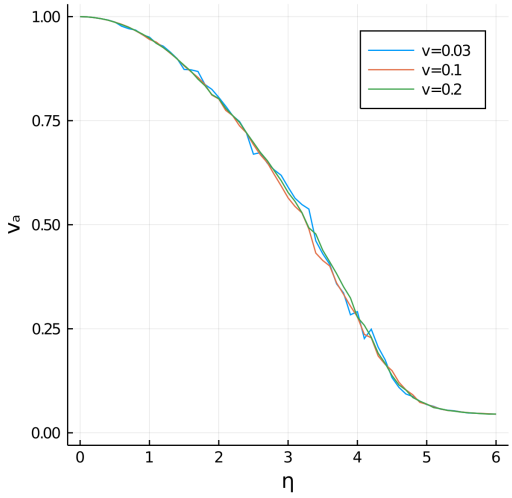


Figure 2.8: The value of the order parameter (v_a) as a function of noise (η) for various values of v_0 in the small-velocity regime ($L = 10$, $N = 400$). Different values of v_0 within this regime seem to make very little difference in the overall result.

our simulation shown in Figure 2.9 agree with this single-peaked result, showing evidence that the transition seems to be second-order in nature. It is difficult to verify, however, if $L = 256$ (the system size used in our simulation) is greater than the earlier described L^* since this quantity is hard to compute given that it is a function of multiple parameters and seems to diverge for cases of angular noise (especially in the small-velocity regime) [14].

Work by the group of Chaté et al. then describe the presence of density waves (high-density bands of particles) in the ordered phase of the Vicsek Model [14, 19]. Nagy et al. argue that these may be due to inherent numerical artifacts in the high-velocity regime related to the shape of the frame with periodic boundary conditions [29]. They show that these waves have the tendency to run parallel or slightly diagonal to the frame of the space. To prove this, they simulated the Vicsek Model in hexagonal cells and found the waves travelling almost parallel to the sides of the hexagons. Because of this possibility of artifacts, the Vicsek Model becomes more difficult to interpret in the large-velocity regime [29].

We run the Vicsek Model 100 times and generate a histogram of the average

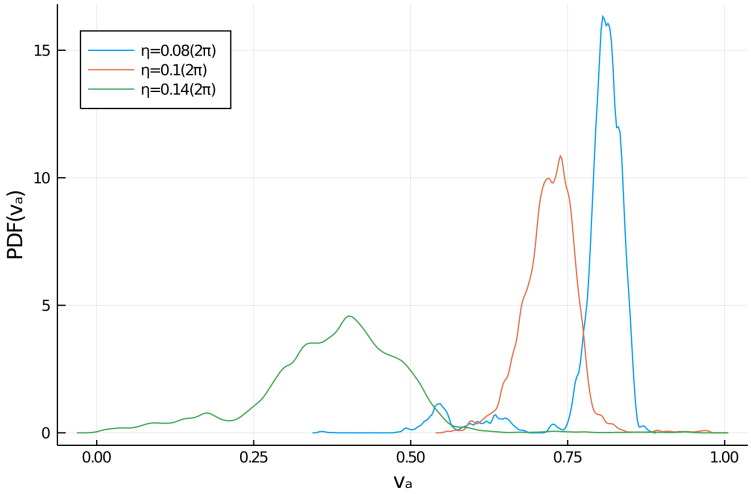


Figure 2.9: PDF of the order parameter v_a in the small-velocity regime with $v_0 = 0.25$ ($L = 256, \rho = 1/8$). Results displayed here were gathered from $\sim 10^5$ time steps after an initial relaxation period of 10^4 time steps.

velocity of the particles after 1.5×10^4 time steps (displayed on Figure 2.10).

Note that even for only a relatively small number of simulations, there are peaks around multiples of $\pi/2$ due to the high-density bands of particles moving almost parallel to the square shaped frame of the simulation space. Figure 2.11 shows some examples of these waves (computed for larger v_0). This conclusion on the presence of artifacts related to the boundary conditions is reiterated also in the work by Baglietto & Albano though with perhaps a different underlying cause (more on this later) [5].

We then express the order parameter (v_a) as a function of the noise (η) and compute the corresponding Binder Cumulant in the small and large-velocity regimes in order to observe the phase transition. The results of this are summarised in Figure 2.12.

Based on all this, our results seem to agree with those from Nagy et al. that while the transition in the case of angular noise is continuous in the small-velocity regime, there may be a discontinuous transition in the large-velocity regime [29]. Proving this is a result of numerical artifacts is outside the scope of this study,

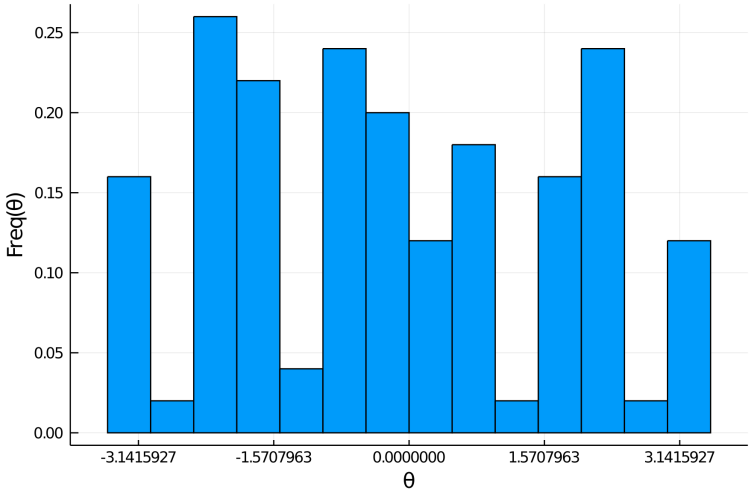


Figure 2.10: Normalised histogram of mean directions of motion of particles in the large-velocity regime across 100 runs of the SVM ($\eta = 0.191(2\pi)$, $L = 256$, $\rho = 1/8$, $v_0 = 0.5$).

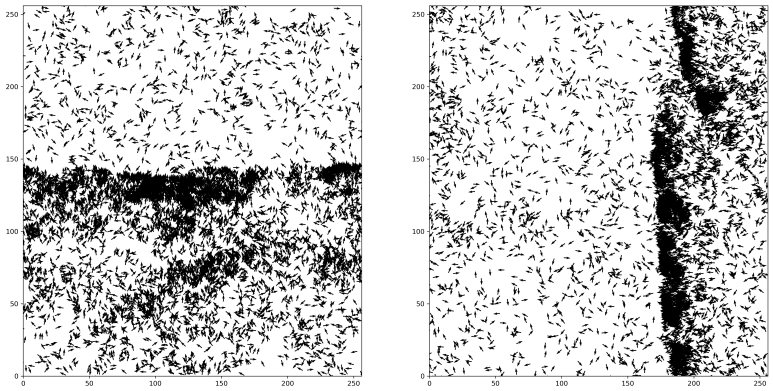


Figure 2.11: Examples of high-density waves of particles moving almost parallel to boundaries of the frame in the large-velocity regime ($\eta = 0.191(2\pi)$, $L = 256$, $\rho = 1/8$, $v = 3.0$).

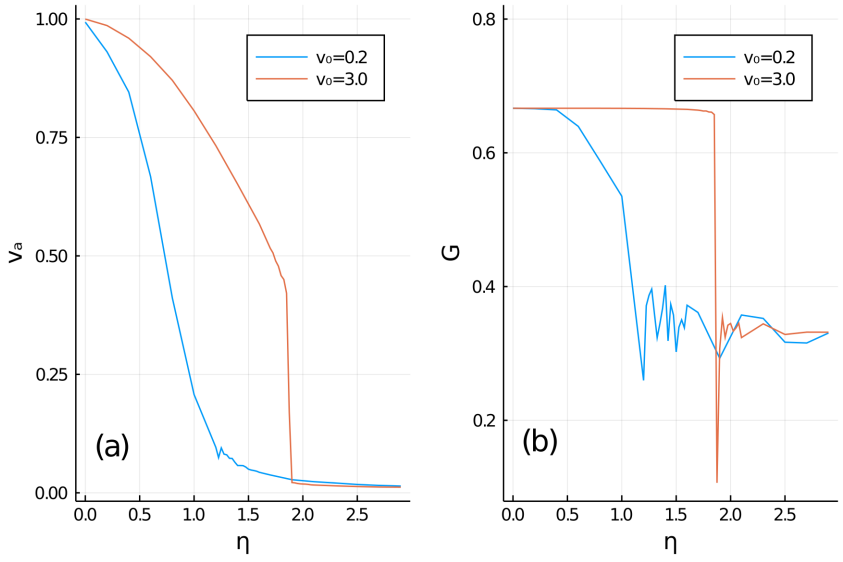


Figure 2.12: (a) The order parameter (v_a) expressed as a function of the noise (η) for two different values of v_0 ($L = 256$, $\rho = 1/8$). While the curve appears continuous for $v_0 = 0.2$, it seems to be discontinuous in the case of $v_0 = 3.0$. (b) The corresponding Binder Cumulant (G) as a function of the noise level (η). There indeed seems to be a definite minimum in the large-velocity regime, characteristic of a first-order transition. In the small-velocity regime, the transition appears to be second-order based on the smoother curve in (a) (though fluctuations in G that may be caused by slow convergence make the Binder Cumulant plot less interpretable in this regard).

but evidence presented in the earlier mentioned study from Nagy et al. seem to indicate this to be the case [29].

Chaté et al. argue, however, that even in the small-velocity regime (where results should be more interpretable), the phase transition should be first-order [14]. They show simulations with $L = 1024$, $\rho = 1/2$, and $v_0 \approx 0.05$ resulting in a discontinuous transition (as L^* seems to be much greater in the small-velocity regime than it is in the large one) [14]. There is still the possibility though that there may be numerical artifacts in this result, as discussed in the next subsection.

2.3.3 Backward vs. Forward Update

Recall that in the original Vicsek Model [35], the new directions of motion $\theta(t+\Delta t)$ are first computed then the positions updated according to Equation 2.1, which we reiterate below

$$x_i(t + \Delta t) = x_i(t) + \mathrm{v}_i(t)\Delta t.$$

Note that even if $\theta(t+\Delta t)$ is computed first the new positions are computed using $\mathrm{v}_i(t)$. In the literature, the term *backward update* (or *backward difference*) is used to describe this update rule [5, 19].

Some studies such as those from the group of Chaté, Grégoire, Ginelli, and Raynaud [14, 18, 19] made use of a slightly modified update rule later called the *forward update* (or *forward difference*), which was also called such and tested in [5]. While the new directions are still computed first, unlike the backward update rule, they are immediately used to update the particle positions. The new positions are then computed as

$$x_i(t + \Delta t) = x_i(t) + \mathrm{v}_i(t + \Delta t)\Delta t. \tag{2.10}$$

It is argued that this update rule should not affect asymptotic results ($N \to \infty$, $t \to \infty$) [14]. However, Baglietto & Albano show that in high-density, large-velocity cases ($\rho = 2.0$, $v_0 \approx 0.5$), the use of the forward update rule results in a first-order transition, while the phase transition remains second-order when applying the backward update rule [5]. They claim that this may be the result of a

numerical artifact related to the boundaries of the space. As mentioned earlier, a similar kind of numerical artifact was found by Nagy et al. despite making use of the backward update rule [29]. In this regard though, there is some disagreement from the studies since Nagy et al. show a first-order transition with the backward update rule and large-velocity regime (which they claim results from a numerical artifact), while Baglietto & Albano show it to be second-order, only finding a first-order transition when applying the forward update rule [5, 29]. This could be a result of the interaction of the other SVM parameters though as Nagy et al. ran the Vicsek Model for a much larger absolute velocity ($v_0 = 10$ vs. $v_0 = 0.5$ in the study by Baglietto & Albano) and for different particle densities [5, 29].

More importantly though, Baglietto & Albano demonstrated that this was an artifact related to the periodic boundary conditions (claimed to be associated with the forward update rule) [5]. As mentioned earlier, their experiments with the forward update rule in high-density, large-velocity cases initially resulted in a first-order transition, but by randomly rotating the angle of the frame of the simulation space at each time step, the transition became second-order once again. This is an important result since it shows that the appearance of a discontinuous transition may have stemmed from a numerical artifact [5]. Whether this artifact also exists in the small-velocity regime was not clearly demonstrated, however.

In the small-velocity regime where results are easier to interpret, Baglietto & Albano claim the forward and backward update rules generally do not change the order of the phase transition [5]. Our simulation results (shown in Figure 2.13) agree with this claim since for the more interpretable small-velocity regime, the update rule does not appear to be changing the order of the transition. Instead, a change in update rule shifts the critical point of the transition (in this case, a leftward shift).

It is important to note though that our results here are with smaller system sizes ($L = 16$). It is likely that given the parameters used here (being in the small-velocity regime) that the value of L^* is much larger and exceeds the capabilities of the hardware being used. While Baglietto & Albano run the SVM for the low-density, small-velocity regime ($v = 0.1$, $\rho = 0.75$, $N = 32768$, $L \approx 209$), showing it to have a second-order transition, they do not do so for the very large system size

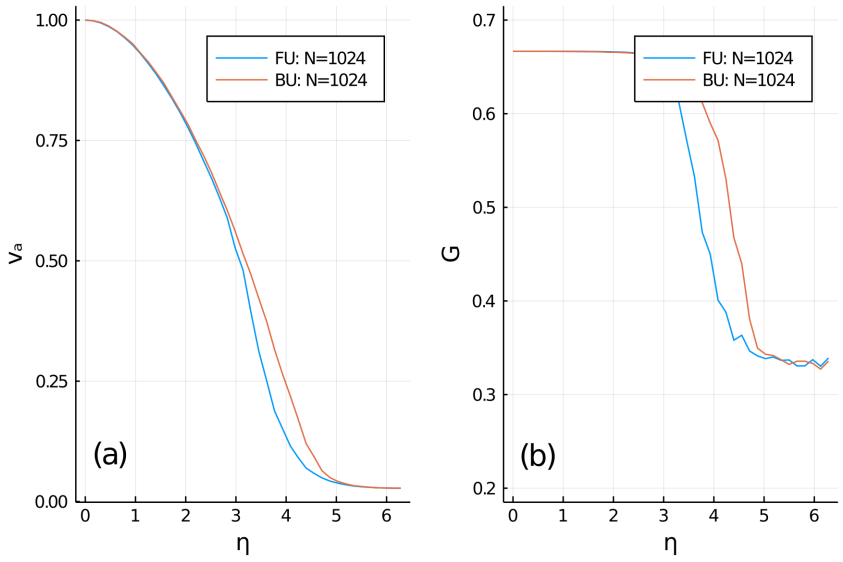


Figure 2.13: (a) The order parameter (v_a) as a function of noise (η) for both the forward (FU) and backward (BU) update rules ($\rho = 4$, $v_0 = 0.25$). (b) The corresponding Binder Cumulant (G) as a function of noise (η). Based on these results, both the forward and backward update rules lead to second-order phase transitions as there is no definite minimum in G for either case.

($v = 0.05$, $\rho = 1/2$, $N = 524288$, $L = 1024$) that Chaté et al. use to demonstrate a first-order transition in the small-velocity regime using the forward update rule [5, 14].

Whether the results in the small-velocity regime from the group of Chaté are affected by some form of artifact due to the update rule is not very clear yet. Overall, it is still difficult to definitively state the order of the phase transition in the SVM without further experiments with large system sizes and tests for numerical artifacts similar to the earlier mentioned experiments by Baglietto & Albano [5].

Chapter 3

The Modified Vicsek Model

The Vicsek Model serves as a good foundation for different modifications that allow us to better study the collective motion of SPPs. In the previous chapter, some variants in the implementation of the SVM were tackled. In all these cases, however, the key interaction between particles remains to be the alignment of headings to neighbours within a circular area of radius r.

Some more impactful modifications to the Vicsek Model involve changes to these interaction rules. One possibility is the introduction of completely new rules such as repulsion (particles pushing neighbours away) and attraction (particles drawing neighbours closer) [22].

There are, however, variants of the Vicsek Model that maintain alignment as the only interaction rule, but implement it differently from the SVM. One such model is proposed by Costanzo & Hemelrijk, which we will refer to as the Modified Vicsek Model (MVM) [15]. Despite the simple nature of the changes made in this model, only introducing two new parameters while maintaining alignment as the sole interaction rule, it can lead to interesting patterns of motion, namely bands, lines, mills, fronts, and flocks [15]. For this thesis, we tackle only the first three of these patterns (bands, lines, and mills) as their occurrence is quite visibly distinct.

In this chapter, we first introduce the modified model, discuss some of the nuances of the implementation, and then study the behaviour of the model as a function of different parameters, giving particular importance to the conditions

that result in mill formation.

3.1 Model Definition

The MVM from Costanzo & Hemelrijk largely retains the definition of the SVM stated in Section 2.1 except for two main changes [15].

1. Field-of-view (ϕ)

   In the SVM, particles align their directions of motion with other particles within a circular area of radius r. In the MVM, a parameter $\phi \in [0, 2\pi]$ controls how much of the circular area around each particle is part of its perception. This means that if $\phi < 2\pi$, a particle is blind to a certain area behind it. Particles in this blind area are not included in alignment computations, even if they are within the interaction radius r. Figure 3.1 shows some examples of the resulting interaction areas for various values of ϕ.

2. Maximum Angular Velocity (ω)

   Recall that in the SVM, $\theta_i(t + \Delta t)$ is computed as the mean direction of motion of particles within the interaction radius of some particle i. Given this, particles can instantaneously change headings as there is no limit on the difference between $\theta_i(t)$ and $\theta_i(t + \Delta t)$. The only role that $\theta_i(t)$ plays is that it is part of the computation for the mean as all particles are deemed to be within their own interaction areas.

   The MVM introduces a maximum angular velocity $\omega \in [0 \text{ rad}/\Delta t, \pi \text{ rad}/\Delta t]$. This limits how much the directions of motion of particles can change within a single time step.

Putting these two changes together, the equation for $\theta_i(t + \Delta t)$ (Equation 2.4

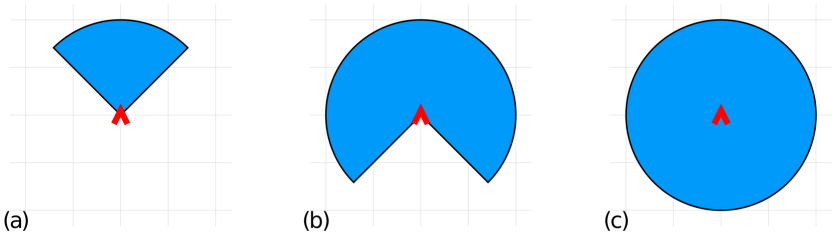


Figure 3.1: All cases in the figure show a particle (in red) with a heading of $\pi/2$ (as shown by the direction of the arrow). The blue filled circles denote the interaction area or perception of the given particle with (a) $\phi = \pi/2$, (b) $\phi = 3\pi/2$, (c) $\phi = 2\pi$. Note that if $\phi = 2\pi$, then the interaction area becomes the same as that in the SVM.

in the SVM) is replaced by

$$\theta_i(t+\Delta t) = \begin{cases} \langle \theta_j(t) \rangle_{r,\phi} + \Delta\theta_i & \text{for } |\Delta\Theta_i| < \omega\Delta t \\ \theta_i(t) + \omega\Delta t + \Delta\theta_i & \text{for } \Delta\Theta_i \geq \omega\Delta t \\ \theta_i(t) - \omega\Delta t + \Delta\theta_i & \text{for } \Delta\Theta_i \leq -\omega\Delta t \end{cases} \tag{3.1}$$

where $\langle \theta_j(t) \rangle_{r,\phi}$ is the average direction of motion of agents within the interaction range r and within the field-of-view ϕ [15]. The variable $\Delta\Theta_i$ is the angular difference or signed minimum magnitude rotation (whether clockwise or counterclockwise) from $\theta_i(t)$ to $\langle \theta_j(t) \rangle_{r,\phi}$. For example, if $\theta_i(t) = \pi/2$ and $\langle \theta_j(t) \rangle_{r,\phi} = \pi/4$ then $\Delta\Theta_i = -\pi/4$ since a $\pi/4$ rotation clockwise (as opposed to a $7\pi/4$ rotation counterclockwise) is the smallest magnitude rotation to get from $\theta_i(t)$ to $\langle \theta_j(t) \rangle_{r,\phi}$. The definition of the noise value remains the same as the original SVM from [35] with $\Delta\theta_i$ as a number selected uniformly at random from the interval $[-\eta/2, \eta/2]$.

The main parameters of the model are then re-expressed as ratios between these quantities and the original parameters in the SVM (with additional factors to make the values dimensionless), namely the absolute velocity to maximum angular velocity ratio ($v_0/(\omega r)$) and the noise to maximum angular velocity ratio ($\eta\Delta t/\omega$)

[15]. With these modifications to the model, new patterns of motion may be observed such as bands, lines, and mills (displayed in Figure 3.2).

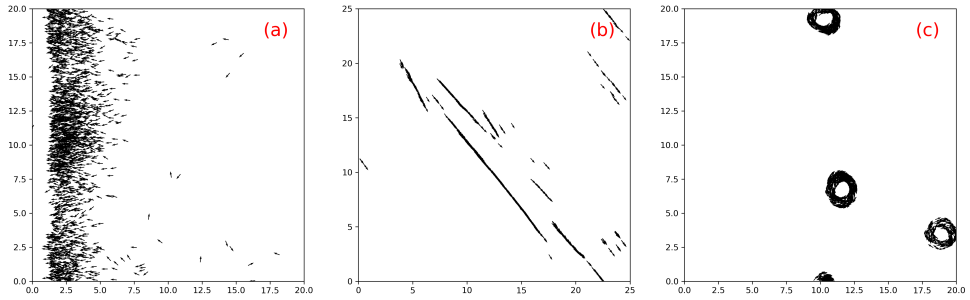


Figure 3.2: Some of the patterns of motion displayed by the MVM using similar parameter values to those in [15] (though with different values for L and N). In all these examples, $\omega = 10^\circ/\Delta t \approx 0.1745$ rad/Δt. In order to vary the ratios then, the other parameter involved is varied (η in $\eta\Delta t/\omega$ and v_0 in $v_0/(\omega r)$). (a) Example of bands, high-density collections of particles moving next to each other in about the same direction, usually spanning the length of one of the sides of the simulation space ($L = 20$, $N = 2000$, $\phi = \pi$, $\eta\Delta t/\omega = 4.5$, $v_0/(\omega r) = 2.0$) [15]. Occasionally more than one band forms and between the bands are scattered particles moving in random directions. (b) Example of particles forming lines, where particles travel one in front of the other ($L = 25$, $N = 2000$, $\phi \approx 0.1745$, $\eta\Delta t/\omega = 0$, $v_0/(\omega r) = 1.03$). (c) Particles forming mills: groups of particles moving in a circular pattern around a common centre ($L = 20$, $N = 1800$, $\phi = \pi$, $\eta\Delta t/\omega = 0$, $v_0/(\omega r) = 0.86$) [15].

3.2 Implementation Details

These changes to the SVM introduce a few additional complexities to the implementation of the model. We tackle two specific details in this section: (a) computing agents in the field-of-view given periodic boundary conditions, and (b) computing the centre of mass of clusters of particles (required to study the model in certain ways discussed later on).

3.2.1 Computing the Field-of-View

The introduction of a limited field-of-view makes the MVM more computationally involved than the SVM. Two main things have to be dealt with in this regard: (i) how particles within the field-of-view are included (or those in the blind area excluded), and (ii) how this computation can be done taking into account the periodic boundary conditions.

We first tackle the computation of which particles are within the field-of-view given some value of the parameter ϕ. Suppose we are interested in computing if some particle j is within the perception of a particle i given that $x_i = (a_i, b_i)$ and $x_j = (a_j, b_j)$. Let α be the angle formed by x_j, x_i and a horizontal line passing through x_i, parallel or equal (in the case that $b_i = 0$) to the x-axis. The size of this angle is given by

$$\alpha = \arctan\left(\frac{b_j - b_i}{a_j - a_i}\right). \tag{3.2}$$

Note that we apply here the two-argument arctan function, sometimes referred to as atan2 in some programming languages.

Let $\Delta\Theta_{i,\alpha}$ be the angular difference from θ_i to α (as described previously in Section 3.1). Assuming j is within the interaction radius r around i, (as there is no need to check the field-of-view otherwise), then j is within the perception of i if $\left|\Delta\Theta_{i,\alpha}\right| \leq \phi/2$. An example of these computations is shown in Figure 3.3 below.

An issue arises though when applying this process in cases such as the MVM where there are periodic boundary conditions since the computed value of α does not take this into account. One way of getting around this is to tile a 3×3 space using the $L \times L$ square that the model is simulated on. Doing this then allows for the proper computation of neighbouring particles within the given field-of-view (computing neighbours only for particles in the central tile). An example of such a tiling is shown in Figure 3.4.

Making use of this tiling and the earlier mentioned computational process, Figure 3.5 shows a specific particle and the resulting neighbours within its field-of-view.

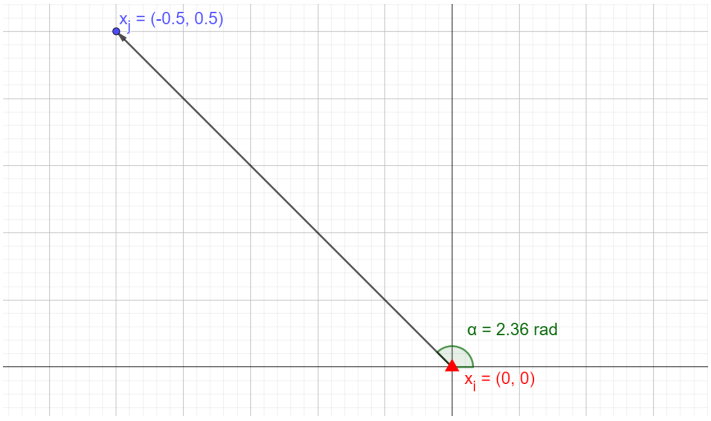


Figure 3.3: Illustration of the process of checking if some particle is within the field-of-view of another particle created using the GeoGebra software [23]. Suppose there are two particles i and j with $x_i = (0, 0)$, $\theta_i = \pi/2$ (as shown by the red, upward facing arrowhead), and $x_j = (-0.5, 0.5)$ (represented by a blue point). We then have $\alpha = \arctan\left(\frac{0.5-0}{-0.5-0}\right) = 3\pi/4$. Suppose then that $\phi = \pi/2$. In this case, $\left|\Delta\Theta_{i,\alpha}\right| = \pi/4 \leq \phi/2$. Because of this, j is within the perception of i and assuming j is also within the interaction radius r around i, then it would be included in the computation of $\theta_i(t + \Delta t)$.

3.2.2 Centre of Mass

In order to study the behaviour of the MVM across numerous runs, the average absolute value of normalised angular momentum (m_a) has to be computed. We tackle this specific quantity in the next section, but part of the process of computing it requires computing the centre of mass or centroid of clusters of particles.

Since all the particles are physically identical, a centre of mass may naively be computed as the mean of the particle coordinate positions. However, because we are dealing with periodic boundary conditions, computing the centre of mass for each cluster becomes more complicated as particles on opposite edges of the frame are actually close to each other. Bai & Breen detail a method for computing this centre of mass in [7], which we go through briefly here.

This method entails first projecting the given 2D points (particle coordinate positions) onto two different 3D "tubes" the first created by wrapping the space to allow the two horizontal axes to meet and the second by wrapping the space the

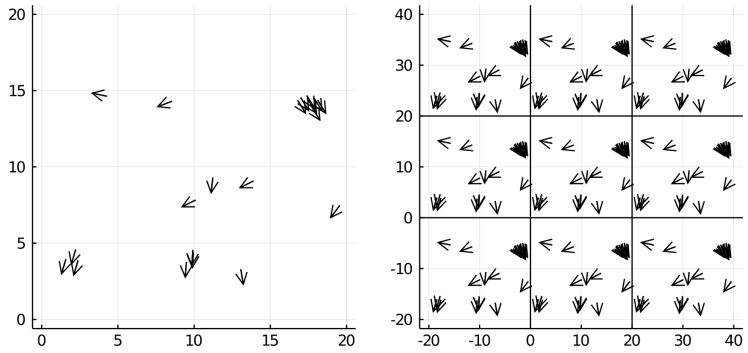


Figure 3.4: On the left is an example of some particles on a 20×20 space. The plot on the right shows a 3×3 tiling using such a space. With this, the correct value of $\theta_i(t + \Delta t)$ which takes into account the periodic boundary conditions may then be computed for the particles in the central tile.

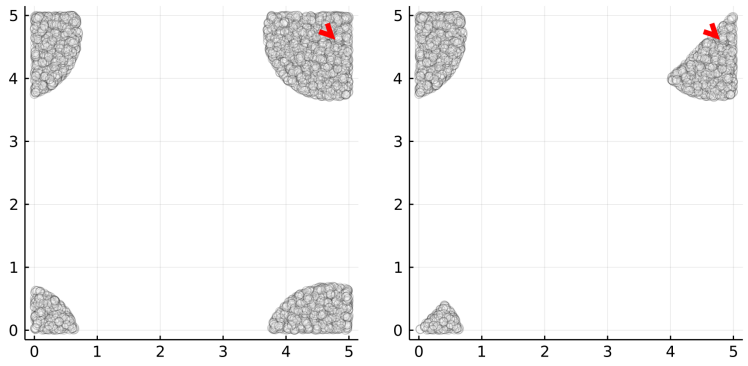


Figure 3.5: On the left, a particle i (red arrow) is surrounded by numerous other particles (the mass of white circles). The figure on the right shows only those particles within the field-of-view of i, assuming $\phi = \pi/2$.

other way to allow the two vertical axes to meet [7]. After projecting the particles onto each 3D surface, the 3D centre of mass for each projection is computed. These 3D centres of mass are then projected back to the surface of their respective tubes, which are then used to compute the true 2D centre of mass that accounts for the periodic boundary conditions.

The proceeding equations in this subsection are from the work of Bai & Breen with some slight changes to fit the context of the Vicsek Model [7]. Given that the Vicsek Model is run on an $L \times L$ square space and each particle has position (i, j), the first projection onto tube T_i made by connecting the vertical axes together ($i = 0$ and $i = L$) is given by

$$\begin{aligned} x = r_i cos(\theta_i), \qquad & y = j, \qquad z = r_i sin(\theta_i), \\ r_i = \frac{L}{2\pi}, \qquad & \theta_i = \frac{i}{L} 2\pi. \end{aligned} \tag{3.3}$$

The second projection on the tube T_j found by connecting the horizontal axes ($j = 0$ and $j = L$) is given by a similar set of equations.

$$\begin{aligned} x = i, \qquad & y = r_j cos(\theta_j), \qquad z = r_j sin(\theta_j), \\ r_j = \frac{L}{2\pi}, \qquad & \theta_j = \frac{j}{L} 2\pi. \end{aligned} \tag{3.4}$$

Once the projections for each agent position denoted X_{T_i} and X_{T_j} have been found, computing the two 3D centres of mass becomes straightforward and is given by

$$\bar{X} = \frac{1}{N} \sum_{k=1}^{N} X_k, \tag{3.5}$$

where $\bar{X}_{T_i}$ is the centre of mass in the T_i projection (setting $X = X_{T_i}$) and $\bar{X}_{T_j}$ is the centre of mass in the T_j projection (setting $X = X_{T_j}$). Suppose $X_{T_i} = (x_i, y_i, z_i)$ and $X_{T_j} = (x_j, y_j, z_j)$, then the true 2D centre of mass $(\bar{i}, \bar{j})$ is then

given by

$$\begin{aligned} \theta_i &= \arctan(-z_i, -x_i) + \pi, & \bar{i} &= \frac{L}{2\pi}\theta_i, \\ \theta_j &= \arctan(-z_j, -y_j) + \pi, & \bar{j} &= \frac{L}{2\pi}\theta_j. \end{aligned} \tag{3.6}$$

An example of the result of this computation may be seen in Figure 3.6 alongside the incorrect result if the naive method were used instead.

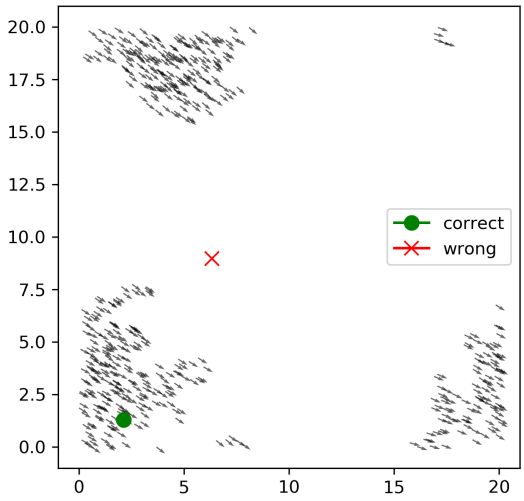


Figure 3.6: The true centre of mass of the given particles computed using the described method from [7] marked with a green circle. A red cross marks an incorrect centre of mass computed using the naive method not taking into account periodic boundary conditions.

3.3 Model Results

This section will go through results of simulations of the MVM with various parameters and the different patterns of motion that may emerge. Just like the original study by Costanzo & Hemelrijk, particular attention is given to the phenomenon

of milling and the conditions that allow for it [15].

While the patterns of motion shown in Figure 3.2 (bands, lines, and mills) can be easily verified through visual inspection of the agent positions and directions of motion, in order to properly study results from the MVM, these patterns have to first be quantified [15]. Two values can be used to achieve this. The first value is the *average normalised velocity* (v_a), which is the order parameter discussed for the phase transitions in the SVM and may be computed with Equation 2.5 (reiterated below) [15, 35].

$$v_a = \frac{1}{Nv_0}\left\|\sum_{i=0}^{N} \mathrm{v}_i\right\|$$

The second value used is the *average absolute value of the normalised angular momentum* (m_a), but to compute this value, particles must first be assigned clusters where the maximum distance between any two particles in a cluster is $d_c = 0.5$ (the value of which was chosen to keep mills in the same cluster) [15]. This clustering can be done by first forming a graph $G = (V, E)$ where $V = \{1, ..., N\}$ (each particle is a node) and there is then an edge between any two nodes within d_c distance of each other (taking into account periodic boundary conditions). A simple graph traversal may then be used to find connected components in G, each of which may be interpreted as a cluster.

Once the particles have been clustered, m_a may be computed with the equation

$$m_a = \frac{1}{N}\sum_{i=1}^{N} \frac{|r_{cm,i} \times u_i|}{|r_{cm,i}|}, \tag{3.7}$$

where $|r_{cm,i}| = |r_i - r_{cm}|$ is the distance between some particle i and the centre of mass of its cluster (not taking into account the periodic boundary conditions), and the product in the numerator is defined with the operation $|a \times b| = |a_1 b_2 - a_2 b_1|$ [15]. Once both v_a and m_a are computed, the different patterns of motion may then be viewed in the context of these two quantities. The following Figures 3.7, 3.8, and 3.9 show the results of this.

Zooming into milling then, one can see from Figure 3.9 that there is generally a small value for v_a and a large value for m_a. From this, a heuristic may be formed

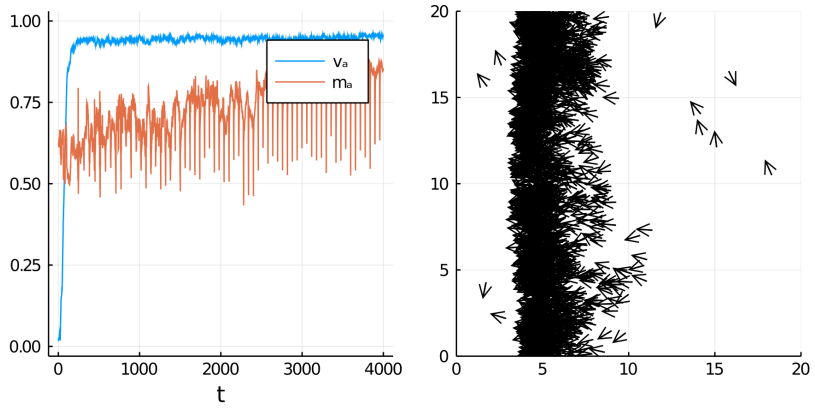


Figure 3.7: On the right is a plot of the particles at time $t = 4000$ forming high-density bands, while the figure on the left shows the values of v_a and m_a over time. One might infer from this that banding is characterised by a large mean v_a and moderately large mean m_a.

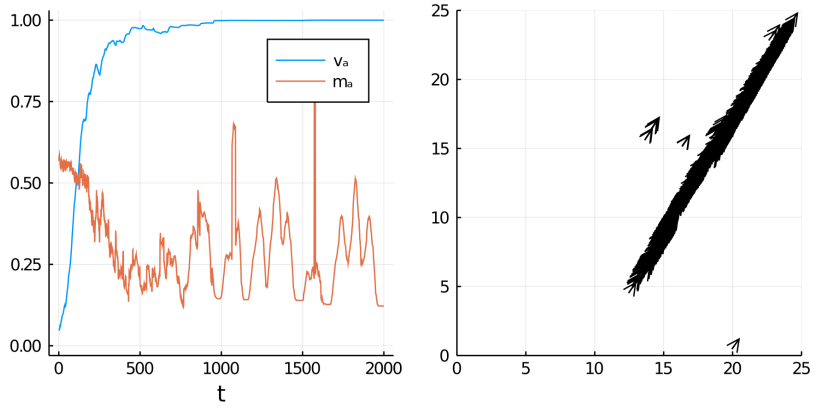


Figure 3.8: On the right is a plot of the particles at time $t = 2000$ forming lines, while the plot on the left shows the values of v_a and m_a over time. One can infer from this that when particles form lines, the value of v_a is typically large, but the value of m_a is smaller than that found in the banding cases.

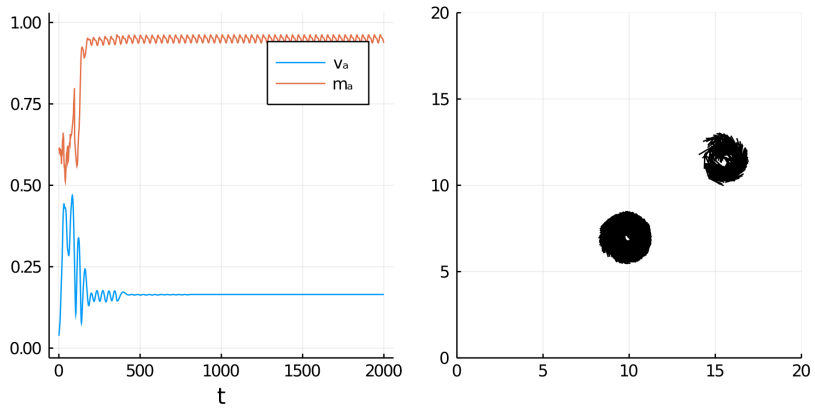


Figure 3.9: The plot on the right shows particles at time $t = 2000$ forming circular mills. On the left, are the value of v_a and m_a over time. One can infer that similar to bands, mills also have a large mean value for m_a, however, the value for v_a is generally much smaller (as particles in a mill may cancel out the velocities of each other).

on when this pattern of motion occurs based on the values of v_a and m_a. Costanzo & Hemelrijk set the thresholds $m_a > 0.75$ and $v_a < 0.5$ as the conditions for milling [15]. We apply the same thresholds, making make use of the mean values over time of m_a and v_a.

We then run similar experiments to those from Costanzo & Hemelrijk to verify their results on the proportion of runs that result in mills as a function of different parameters [15]. For each of these experiments, the simulation was run 40 times (unless otherwise specified) for each value of the independent variable. The dependent variable, p_{mill} (the proportion of the runs that resulted in milling) was then computed for the different values of the independent variable. To get more stable results, the first 1500 time steps were not included for the computation of the time averages of m_a and v_a as the first few time steps usually involve large fluctuations.

First, we run this experiment with noise to maximum angular velocity ratio ($\eta\Delta t/\omega$) as the independent variable. Based on the results from Figure 3.10, the value of p_{mill} is generally large (close to one) for smaller values of $\eta\Delta t/\omega$, but

there is a sudden drop to a zero value where milling becomes very unlikely (in this case for $\eta\Delta t/\omega \approx 4$).

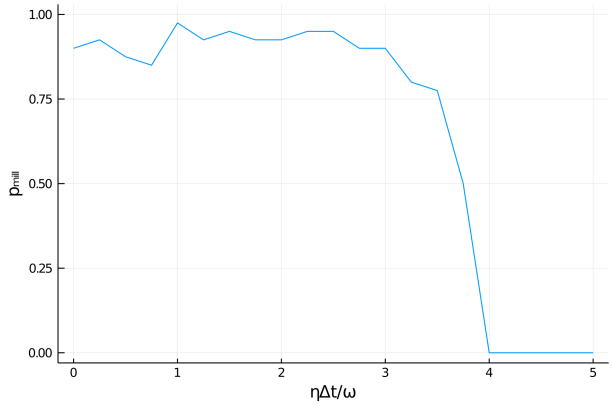


Figure 3.10: The proportion of 40 cases that resulted in milling as a function of the noise to maximum angular velocity ratio $\eta\Delta t/\omega$ for parameter values similar to [15] ($L = 20$, $\rho = 2.5$, $\phi = \pi$, $v_0/(\omega r) = 1.03$). Note that for all runs, the maximum angular velocity was fixed with $\omega = 10^\circ \approx 0.1745$ rad and only the value of η was varied.

We then run an experiment this time with p_{mill} as a function of the density of particles ρ. The results from Figure 3.11 show that just like the order parameter of the SVM, the proportion of milling cases increases quite rapidly as the density of particles increases. At $\rho \approx 4$, the value of p_{mill} gets close to one, meaning there is almost always milling past a certain density (for the given values of the other parameters).

Being a new parameter introduced by the MVM, it is also of interest to view p_{mill} as a function of the field-of-view parameter ϕ. The result of this simulation is displayed in Figure 3.12. Based on this, one can infer that mills only form for moderate values of ϕ within the approximate range $\pi/2 < \phi < 3\pi/2$. Values outside this rough estimate cause a sudden drop in p_{mill} to an approximately zero value.

Lastly, we view the mill proportion as a function of the speed of each particle (v_0) by fixing the ratio $v_0/(\omega r) = 0.9$ and varying both v_0 and ω (similar to the experiment in [15]). Results in Figure 3.13 show that for the fixed ratio $v_0/(\omega r)$,

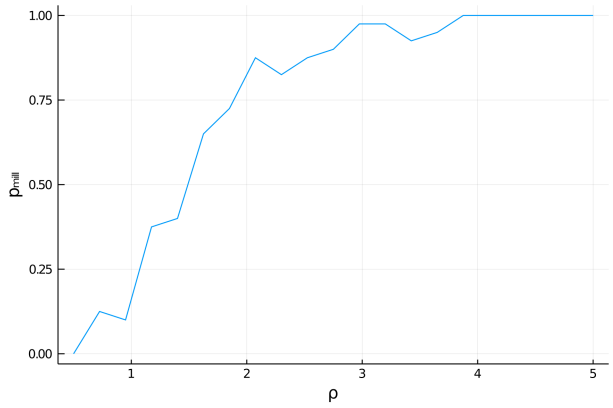


Figure 3.11: The proportion of milling cases over 40 runs (p_{mill}) as a function of particle density ρ for parameter values similar to [15] ($L = 20$, $\phi = \pi$, $\eta\Delta/\omega = 0.5$, $v_0/(\omega r) = 1.03$, $\omega = 10^\circ \approx 0.1745$ rad).

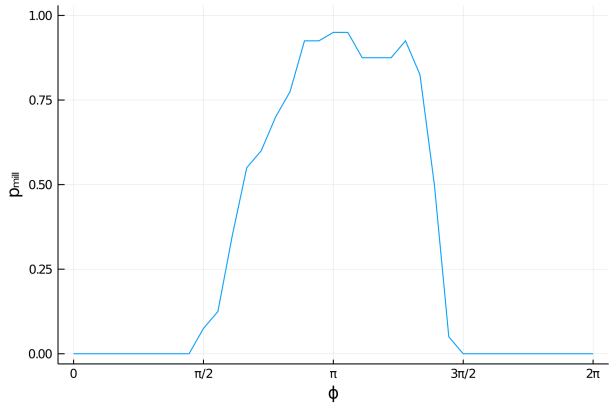


Figure 3.12: The proportion of milling cases over 40 runs (p_{mill}) as a function of the field-of-view ϕ for parameter values similar to those from [15] ($L = 20$, $\rho = 2.5$, $\eta\Delta/\omega = 0.5$, $v_0/(\omega r) = 1.03$, $\omega = 10^\circ \approx 0.1745$ rad).

p_{mill} has value close to one until it starts decreasing with $v_0 > 0.3$. At around $v_0 \approx 0.5$, the mill proportion is already 0. This means that even for a given value of $v_0/(\omega r)$, moderate values of v_0 and ω are still more favourable for mill formation.

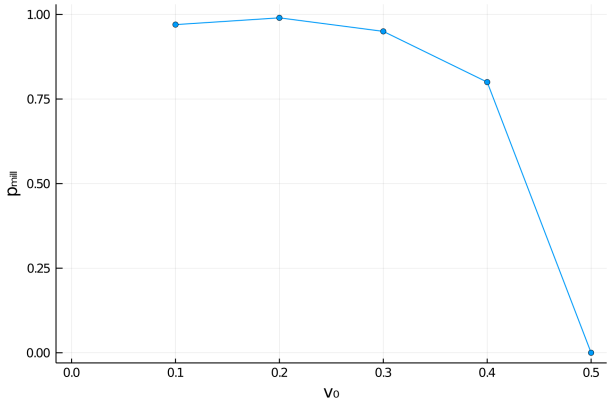


Figure 3.13: The proportion of milling cases over 100 runs (p_{mill}) as a function of v_0 with fixed ratio $v_0/(\omega r) = 0.9$ for parameter values from [15] ($L = 20$, $\rho = 2.5$, $\phi = \pi$, $\eta\Delta t/\omega = 0$).

From these experiments, one can infer the conditions favourable to the formation of mills. Similar to an ordered state in the SVM, milling is most prevalent in a low-noise, high-density environment (assuming some fixed value for ω). However, this only occurs for moderate values of ϕ, as based on trials, too small or too large a value of ϕ reduces the proportion of cases that result in milling rather steeply. Given some fixed value of the ratio $v_0/(\omega r)$, milling occurs only for smaller to moderate values of v_0 and ω, as allowing particles to move or rotate too quickly is not conducive for mill formation. Results of the experiments carried out here and the resulting inferences on the conditions for milling agree quite closely with those from Costanzo & Hemelrijk [15].

Chapter 4

Conclusions

In this work, we first explored the Standard Vicsek Model as originally proposed by Vicsek et al. in [35], and then demonstrated some of its behaviours, most notably a phase transition from a disordered phase to an ordered phase as a result of changing the level of noise or density of particles in the system.

There has been some debate in the literature regarding the order of the phase transition in the context of different variations in the implementation of the model. Here we ran simulations testing some of these variations: namely changes in the noise type, the velocity regime, and the position update rule. First, we ran the SVM with intrinsic noise (angular noise) and extrinsic noise (vectorial noise and simple extrinsic noise). Simulations showed that while angular noise results in a second-order phase transition, the two kinds of extrinsic noise tested here (SEN and vectorial noise) appear to result in first-order transitions. Simulation results show that the phase transition in the large-velocity regime appears to be first-order as opposed to the second-order one in the small-velocity regime. Aside from this, runs of the Vicsek Model in the large-velocity regime exhibit high-density waves of particles in the ordered phase that seem to be influenced by the shape of the frame of the simulation space, which Nagy, et al. argue stems from numerical artifacts [29]. Based on our results, the update rule does not seem to have too significant an effect on the results of the model, but it does cause a shift in the critical point. However, work from Baglietto & Albano claim that the use of the forward update

rule introduces numerical artifacts into the simulation [5].

Our results here agree with studies from Baglietto & Albano [5], Nagy et al. [29], and Aldana et al. [2], but run contrary to those from the group of Chaté et al. [13, 14, 19]. However, simulations performed here were for relatively small system sizes. Overall, there is still some contention with regard to the order of the phase transition in the SVM as the value of L^*, beyond which the system supposedly shows its asymptotic behaviour with a discontinuous transition can grow prohibitively large [14]. Further experiments would be required to see if factors such as the update rule introduce numerical artifacts in this regard.

Afterward, we then tackled a Modified Vicsek Model as proposed by Costanzo & Hemelrijk in [15], which introduced two new parameters, the maximum angular velocity ω and the particle field-of-view ϕ. These additions allow for the emergence of new patterns of motion of which bands, lines, and mills were tackled. These patterns were then quantified and the conditions for milling were fleshed out through experiments. Results showed mills were more likely to form in systems with smaller values for the noise to maximum angular velocity ratio ($\eta\Delta t/\omega$), higher particle densities (ρ), moderate values for the field-of-view (ϕ), and given some fixed value for $v_0/(\omega r)$, moderate values for absolute and maximum angular velocities. Overall, these results reaffirm conclusions from the original study [15].

Appendix A

Core Model Implementation

Included here are the core functions necessary to run the SVM (with its variants) and the MVM. For the sake of brevity, not all of the functions are included (some have been excluded such as some of the utility functions to compute the order parameter, compute the centre of mass, or normalise angles to a certain range, etc.). Though what is essential (and less straightforward such as particle clustering) has, for the most part, been included here.

```
using Distances, NearestNeighbors, Distributions, LinearAlgebra, DataStructures
"""
Agents are then moved `v` units in this direction taking into account periodic
boundary conditions with size `L`.

# Arguments
- `agent_pos`: Positions of agents
- `agent_dirs`: Direction of motion of each agent in radians
- `v`: Distance travelled by each agent per unit of time
- `L`: Size of the plane the agents are moving on
# Return
- `agent_pos`: Positions of agents after one step of motion
- `moves`: Vector of motion (velocity) of each agent
"""
```

```
function move_agents(agent_pos, agent_dirs, v, L)
    agent_pos = copy(agent_pos)
    moves = [cos.(agent_dirs) sin.(agent_dirs)].*v
    #Add velocities to the position of each agent
    agent_pos += moves
    #Periodic boundary conditions: add L to negative values and get the modulo
    #of all positions by L
    agent_pos[agent_pos.<0].+=L
    agent_pos.%=L
    return agent_pos,moves
end;

"""
The next three functions are similar in that they all compute θ(t+Δt), but apply
different forms of noise. They all share the same parameters (listed here). The
function "normalise_angle" that these functions call is a utility function (not
included here) that simply normalises an angle to the range (-π,π].

# Arguments
- `agent_pos`: Positions of agents
- `agent_dirs`: Direction of motion of each agent in radians
- `L`: Size of the plane the agents are moving on
- `r`: Range of interaction
- `η`: Noise level (interpretation varying based on noise type as discussed in
the thesis body)
# Return
Returns an array containing the new direction of motion for each agent
"""
function compute_directions(agent_pos, agent_dirs, L, r, η)
    new_dirs=[]
    #Make use of a BallTree (from NearestNeighbors.jl)
    btree = BallTree(transpose(agent_pos), PeriodicEuclidean([L,L]))
```

```
    for i=1:(size(agent_pos,1))
        row = agent_pos[i,:]
        #Get neighbours and compute average velocity
        neighbours = inrange(btree, row, r, true)
        neighbour_dirs = agent_dirs[neighbours,:];
        new_dir = atan(mean(sin.(neighbour_dirs)),mean(cos.(neighbour_dirs)))
        push!(new_dirs,new_dir)
    end
    #Add angular noise and return resulting angle normalised to range (-π,π]
    if η!=0
        return normalise_angle.(new_dirs + rand(Uniform((-η/2), (η/2)),
        size(new_dirs)))
    else
        return normalise_angle.(new_dirs)
    end
end;

#Computes θ(t+Δt) applying simple extrinsic noise as defined in the thesis.
function compute_directions_simple_extrinsic(agent_pos, agent_dirs, L, r, η)
    new_dirs=[]
    #Make use of a BallTree (from NearestNeighbors.jl)
    btree = BallTree(transpose(agent_pos), PeriodicEuclidean([L,L]))
    for i=1:(size(agent_pos,1))
        row = agent_pos[i,:]
        neighbours = inrange(btree, row, r, true)
        neighbour_dirs = agent_dirs[neighbours,:];
        #Add a simple extrinsic noise if the noise level is non-zero
        if η != 0
            neighbour_dirs += rand(Uniform((-η/2), (η/2)), size(neighbour_dirs))
        end
        new_dir = atan(mean(sin.(neighbour_dirs)),mean(cos.(neighbour_dirs)))
        push!(new_dirs,new_dir)
```

```
    end
    #Return new directions (with noise already applied) in range (-π, π]
    return normalise_angle.(new_dirs)
end;

#Computes θ(t+Δt) applying vectorial noise as discussed in the thesis.
function compute_directions_vectorial(agent_pos, agent_dirs, L, r, η)
    new_dirs=[]
    #Make use of a BallTree (from NearestNeighbors.jl)
    btree = BallTree(transpose(agent_pos), PeriodicEuclidean([L,L]))
    for i=1:(size(agent_pos,1))
        row = agent_pos[i,:]
        #Get neighbours as other agents within r units away
        neighbours = inrange(btree, row, r, true)
        neighbour_dirs = agent_dirs[neighbours];
        #Generate unit vector in the direction of neighbour directions of motion
        unit_vecs = hcat([cos.(neighbour_dirs), sin.(neighbour_dirs)]...);
        new_vec = sum(unit_vecs, dims=1)[1,:]
        #If the noise level is not 0, then add the perturbation
        if η != 0
            #Generate and apply noise vector
            noise_dir = rand(Uniform(-π, π))
            noise_vec = [cos.(noise_dir), sin.(noise_dir)] .*
            (η * size(neighbour_dirs)[1]);
            new_vec += noise_vec
        end
        #Compute the angle of the resulting vector as the new direction
        new_dir = atan(new_vec[2], new_vec[1])
        push!(new_dirs,new_dir)
    end
    #Return new directions (with noise already applied) in range (-π, π]
    return normalise_angle.(new_dirs)
```

```
end;

"""
Computes which neighbours are within the field-of-view (in terms only of angle
and not distance, as it assumes that input has already been filtered).

# Arguments
- `neighbour_pos`: Matrix containing positions of neighbours
- `curr_facing`: Direction of motion in radians of the selected agent
- `curr_pos`: Position of the selected agent
- `fov`: Angle (in radians) for the field-of-view of the selected agent
# Return
Returns an array of indices of the neighbour_pos input that are within the
selected agent's field-of-view
"""
function in_fov(neighbour_pos, curr_facing, curr_pos, fov)
    #Get the angle between the current agent and neighbours
    angles_between = [atan(reverse(row)...) for row in
    eachrow(neighbour_pos .- transpose(curr_pos))]
    #Minimum rotation from current facing to the angle of neighbours
    diff = min_rotation.(fill(curr_facing, length(angles_between)),
    angles_between)
    #Include self as within the interaction radius
    in_eps = (x,y) -> all(abs.(x-y) .< (10^-7))
    find_self = in_eps.(Ref(curr_pos), eachrow(neighbour_pos))
    #Get neighbours within the field-of-view (including self)
    in_fov_ind = (abs.(diff) .<= fov/2) .| find_self
    return in_fov_ind
end;

"""
Function below computes new directions of motion (similar to the previous
```

```
functions above), but taking into account two new parameters below (as
discussed in the thesis). This is used in runs of the MVM.
- `fov`: Field-of-view of agents
- `ω`: Maximum angular velocity
"""
function compute_directions_fov(agent_pos, agent_dirs, L, r, η, fov, ω)
    new_dirs=[]
    agent_pos = copy(agent_pos)
    agent_dirs = copy(agent_dirs)
    #Create a 3 x 3 tiled space by shifting around the original L x L square
    pos_builder = []
    dir_builder = []
    offset = [-L 0 L]
    for i in offset
        for j in offset
            curr_pos = [agent_pos[:,1].+i agent_pos[:,2].+j]
            push!(pos_builder, curr_pos)
            push!(dir_builder, copy(agent_dirs))
        end
    end
    pos_builder = vcat(pos_builder...)
    dir_builder = vcat(dir_builder...)
    #Create a BallTree (from NearestNeighbors.jl) to compute neighbours quickly
    btree = BallTree(transpose(pos_builder))
    for i=1:(size(agent_pos,1))
        row = agent_pos[i,:]
        old_dir = agent_dirs[i]
        neighbours = inrange(btree, row, r, true)
        #Compute for neighbours and get only those in the fov
        neighbour_pos = pos_builder[neighbours,:];
        if length(neighbours)>1
            in_fov_ind = in_fov(neighbour_pos, old_dir, row, fov)
```

```
            neighbours = neighbours[in_fov_ind]
        end
        neighbour_dirs = dir_builder[neighbours,:];

        #Compute the new direction and apply the limit defined by ω
        new_dir = atan(mean(sin.(neighbour_dirs)),mean(cos.(neighbour_dirs)))
        min_rot = min_rotation(old_dir, new_dir)
        if min_rot >= ω
            new_dir = old_dir + ω
        elseif min_rot <= -ω
            new_dir = old_dir - ω
        end
        push!(new_dirs,new_dir)
    end
    #Add noise and return computed angle in the range (-π, π]
    if η!=0
        return normalise_angle.(new_dirs + rand(Uniform((-η/2), (η/2)),
        size(agent_dirs)))
    else
        return normalise_angle.(new_dirs)
    end
end

"""
Clusters agents using a graph traversal as discussed in the thesis (necessary
to compute the value of mₐ).

# Arguments
`agent_pos`: Positions of agents
`L`: Size of the plane the agents are moving on
`dc`: Distance between agents in the cluster (maximum edge length in graph)
# Return
```

```
Returns an array containing the assigned cluster ID for each corresponding agent
"""
function cluster_agents(agent_pos, L, dc)
    N = size(agent_pos)[1]
    clusters = collect(1:N);
    vis = fill(false, N);
    btree = BallTree(transpose(agent_pos), PeriodicEuclidean([L,L]))
    #Start a BFS from each agent that hasn't been visited
    for i=1:N
        #If Node i is not visited, start a BFS from it
        if !vis[i]
            queue = Queue{Int}();
            enqueue!(queue, i)
            vis[i] = true
            #While there are nodes in the queue
            while !isempty(queue)
                curr = dequeue!(queue)
                row = agent_pos[curr,:]
                #Get unvisited neighbours and set them to the same cluster
                neighbours = inrange(btree, row, dc, true)
                neighbours = neighbours[.!vis[neighbours]]
                clusters[neighbours] .= i
                vis[neighbours] .= true
                #Put neighbours in the queue to continue the graph traversal
                for ind in neighbours
                    enqueue!(queue,ind)
                end
            end
        end
    end
    return clusters
end;
```

Bibliography

[1] M. Aldana, V. Dossetti, C. Huepe, V. M. Kenkre, and H. Larralde. Phase transitions in systems of self-propelled agents and related network models. *Physical Review Letters*, 98:095702, 2007.

[2] M. Aldana, H. Larralde, and B. Vázquez. On the emergence of collective order in swarming systems: a recent debate. *International Journal of Modern Physics B*, 23(18):3661–3685, 2009.

[3] I. Aoki. A simulation study on the schooling mechanism in fish. *Bulletin of the Japanese Society of Scientific Fisheries*, 48(8):1081–1088, 1982.

[4] G. Baglietto and E. V. Albano. Finite-size scaling analysis and dynamic study of the critical behavior of a model for the collective displacement of self-driven individuals. *Physical Review E*, 78:021125, 2008.

[5] G. Baglietto and E. V. Albano. Nature of the order-disorder transition in the vicsek model for the collective motion of self-propelled particles. *Physical Review E*, 80:050103, 2009.

[6] G. Baglietto, E. V. Albano, and J. Candia. Criticality and the onset of ordering in the standard vicsek model. *Interface Focus*, 2:708–714, 2012.

[7] L. Bai and D. Breen. Calculating center of mass in an unbounded 2d environment. *Journal of Graphics Tools*, 13:53–60, 2008.

[8] E. Ben-Jacob, I. Cohen, A. Czirók, T. Vicsek, and D. L. Gutnik. Chemomodulation of cellular movement, collective formation of vortices by swarming bacteria, and colonial development. *Physica A: Statistical Mechanics and its Applications*, 238:181–197, 1997.

[9] G. Bianconi. Mth751-u: processes on networks, lecture notes, 2020.

[10] T. Breloff et al. Juliaplots/plots.jl, version v1.4.3, 2020. URL: https://github.com/JuliaPlots/Plots.jl.

[11] T. Breloff et al. Juliaplots/statsplots.jl, version v0.14.6, 2020. URL: https://github.com/JuliaPlots/StatsPlots.jl.

[12] K. Carlsson et al. Kristofferc/nearestneighbors.jl: v0.4.6, version v0.4.6, June 2020. DOI: 10.5281/zenodo.3885465. URL: https://doi.org/10.5281/zenodo.3885465.

[13] H. Chaté, F. Ginelli, and G. Grégoire. Comment on "phase transitions in systems of self-propelled agents and related network models". *Physical Review Letters*, 99:229601, 2007.

[14] H. Chaté, F. Ginelli, G. Grégoire, and F. Raynaud. Collective motion of self-propelled particles interacting without cohesion. *Physical Review E*, 77:046113, 2008.

[15] A. Costanzo and C. K. Hemelrijk. Spontaneous emergence of milling (vortex state) in a vicsek-like model. *Journal of Physics D: Applied Physics*, 51(13):134004, 2018.

[16] A. Czirók, H. E. Stanley, and T. Vicsek. Spontaneously ordered motion of self-propelled particles. *Journal of Physics A: Math. Gen.*, 30:1375–1385, 1997.

[17] F. Dunn and I. Parberry. *3D Math Primer for Graphics and Game Development*. CRC Press LLC, 2nd edition, 2011.

[18] F. Ginelli. The physics of the vicsek model. *The European Physical Journal Special Topics*, (225):2099–2117, 2016.

[19] G. Grégoire and H. Chaté. Onset of collective and cohesive motion. *Physical Review Letters*, 92(2):025702, 2004.

[20] J. D. Hunter. Matplotlib: a 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[21] M. Ibele, T. E. Mallouk, and A. Sen. Schooling behavior of light-powered autonomous micromotors in water. *Angewandte Chemie International Edition*, 48:3308–3312, 2009.

[22] T. Iliass, D. Cambui, and L. Youssef. Non-universality of critical exponents of a self-propelled particles system. *Chinese Journal of Physics*, 54:108–114, 2016.

[23] I. G. Institute. Geogebra 6.0.598.0, 2020. URL: www.geogebra.org.

[24] IUPAC. *Compendium of Chemical Terminology*. A. D. McNaught and A. Wilkinson, editors. Blackwell Scientific Publications, Oxford, 2nd edition, 1997.

[25] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, 2000, pages 68–131.

[26] D. Lin et al. Juliacollections/datastructures.jl, version v0.17.19, 2020. URL: https://github.com/JuliaCollections/DataStructures.jl.

[27] D. Lin et al. Juliastats/distances.jl, version v0.8.2, 2020. URL: https://github.com/JuliaStats/Distances.jl.

[28] D. Lin et al. Juliastats/distributions.jl: v0.23.11, version v0.23.11, Sept. 2020. DOI: 10.5281/zenodo.4011714. URL: https://doi.org/10.5281/zenodo.4011714.

[29] M. Nagy, I. Daruka, and T. Vicsek. New aspects of the continuous phase transition in the scalar noise model (snm) of collective motion. *Physica A: Statistical Mechanics and its Applications*, 373:445–454, 2006.

[30] J. Reback et al. Pandas-dev/pandas: pandas 1.0.1, version v1.0.1, Feb. 2020. DOI: 10.5281/zenodo.3644238. URL: https://doi.org/10.5281/zenodo.3644238.

[31] C. W. Reynolds. Flocks, herds, and schools: a distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.

[32] V. Schaller, C. Weber, C. Semmrich, E. Frey, and A. R. Bausch. Polar patterns of driven filaments. *Nature Letters*, 467(2):73–77, 2010.

[33] B. Simons. Phase transitions and collective phenomena, 1997. URL: http://www.tcm.phy.cam.ac.uk/~bds10/phase.html.

[34] S. Van Der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22, 2011.

[35] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226, 1995.

[36] T. Vicsek and A. Zafeiris. Collective motion. *Physics Reports*, 517:71–140, 2012.

[37] M. Waskom et al. Mwaskom/seaborn: v0.10.0 (january 2020), version v0.10.0, Jan. 2020. DOI: 10.5281/zenodo.3629446. URL: https://doi.org/10.5281/zenodo.3629446.