Mathematics MSc Dissertation MTHM038, 2019/20

The Dynamics of Swarms (the Vicsek model)

Yuren Sheng, ID: 170325598

Supervisor: Dr. Wolfram Just

A thesis presented for the degree of
Master of Science in *Mathematics*

School of Mathematical Sciences
Queen Mary University of London

Declaration of original work

This declaration is made on September 10, 2020.

Student's Declaration: I Yuren Sheng 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.

Acknowledgements

I want to sincerely thank my supervisor Dr. Wolfram Just. This MSc project was completed under his careful guidance. From the selection of the project, implementation to the final completion, Dr. Just gave me patient guidance and support, and every piece of his advice has benefited me a lot.

I also want to thank my tutor, Prof. Alexander Gnedin. He gave me a lot of guidance on courses selection and many of them are very helpful in completing this project.

Finally, I want to thank my classmate and friend Yuyu Zhu. In the process of the project, his suggestions and encouragement often inspired me.

Abstract

This project is aimed at studying the dynamics of swarms by Vicsek model. It would first reproduce a simple Vicsek model on MATLAB. Then, the model would be modified, adding the functions of limited vision angle and adaptive individual speeds respectively. Finally, The parameters and convergence properties of the model are studied. The influence of the starting speed for particles in the system on the order parameters and which value of certain parameters can make the movement of particles be synchronized fastest were found.

Contents

Chapter 1

Introduction

1.1 Motivation

From cellular migrations to flocks of sheep, from cooperative behavior of molecular motors to flustered people, collective motions of abundant organisms universally exist in living nature (Ginelli, 2016).

Although the perception and intelligence of individuals in many swarms are not very high, the entire group can show complex behaviors, for instance, keeping individuals consistent in their speed of movement and migrating toward the same target (e.g. food or habitat) (Parrish and Hamner, 1997). Some swarms can even form special structures to deal with emergencies such as avoiding obstacles or predators (Inada and Kawachi 2002, Couzin et al. 2003).

The study of dynamics of swarms is a way to understand biological complexity. On the other hand, research can apply this distributed strategy to the control theory of autonomous multi-agent systems (e.g. drone clusters) (Tanner et al., 2003). By designing some control rules, the system can produce the desired behaviors. The research on dynamics of swarms may also be used to explain the reasons why some species are not very intelligent, but through cooperation between others, they can show some very intelligent behaviors (Jadbabaie et al., 2003). Thus, the study of swarms has potential application values in engineering (Olfati-Saber and Murray, 2002).

1.2 Background

There are two balanced and opposite behaviors in a school of fish or a flock of birds in nature (Partridge, 1982): Expect to move closer to the center of adjacent individuals: avoid collisions with adjacent individuals. Birds seem to be able to divide the entire group into three categories, namely: themselves, nearby birds and other individuals. Many researchers have tried to explain how the group animals achieve the same direction of action without unified control.

With the development of computer science, Reynolds (1987) proposed three laws of individual movement based on observations of birds in 1987 i.e. individual birds will fly according to the following rules: 1) Gather: try to get close to each other to avoid isolation; 2) Repel: avoid collisions with surrounding individuals; 3) Synchronize: try to keep the velocity consistent with the surrounding individuals. Reynolds established the earliest calculation model based on these three rules.

In subsequent studies, researchers usually consider the specific actions of individual birds in flight in the model. The most typical example is that Heppner and Grenader in their model proposed an arithmetic differential equation with 15 parameters (Heppner and Grenander, 1990). However, for a cellular migrations system with hundreds of individuals, it is difficult to draw general conclusions from a complex mathematical model.

In 1995, the Hungarian physicist T. Vicsek proposed a simple and effective model for simulating cellular migrations, i.e. Vicsek's model from the perspective of statistical mechanics. In the Vicsek model, each individual involved in cellular migrations can be regarded as an independent self-propelled individual and each individual moves in a two-dimensional plane with a constant speed. The speed and direction at the next moment are related the average velocity of other individuals around it (including itself). After a limited number of steps, all individuals will eventually reach a common velocity, i.e. the movement directions of all individuals tends to be the same (Vicsek et al., 1995).

In the Vicsek model, the status of each individual is equal, but it is different in the real world. A good example is the migration of wild geese. There are often some "leaders" among them, which play a leading role in the movement of the flock. Based on this phenomenon, the Vicsek model with some "leaders" was proposed (Couzin et al., 2005). Moreover, natural enemies have a huge influence on their group behavior in some populations. Therefore, some models with predators are proposed. Gazi and Passino (2002) studied the self-organization behavior of a model formed by self-propelled individuals with interactions and found that in two-dimensional situations, all individuals rotate around a common center;Jadbabaie et al. (2003) studied the colony behavior of bacteria, the migration movement of bacterial colonies and so on.

In addition, Czirók et al. (1997) further linked biological cellular migrations with ferromagnetic dynamics, and applied the average field method for ferromagnetization to biological cellular migrations. Subsequently, many physical methods were introduced to the study of cellular migrations. Simha and Ramaswamy (2002) gave an explanation of fluid mechanics. Olfati-Saber and Murray (2004) discussed the influence of information organization structure. In addition to in-depth exploration of cluster behavior description methods, many scholars began to study the impact of individual or partial behavior on the entire group. Mu et al. (2005) pointed out that in the model with a leader, the larger the cluster size, the smaller the proportion of individuals (e.g. leaders) who need to know the cluster goals. Three years later, Ballerini et al. (2008) pointed out that regardless of the cluster density, when an individual has a fixed number of neighbors, it has the best robustness against obstacles and external attacks.

1.3 Outline of Project

In the first step of this project, The research program reproduced the Vicsek model proposed by Vicsek et al. in 1995 with MATLAB and named it basic VM (i.e. basic vicsek model). Then the simulation results would be compared with the original paper to verify whether the model can work correctly.

In the second step, different starting speed and noise were used to do the simulations and studied the influence of these parameters on the order parameter (i.e. average normalized velocity).

In the third step, the code of the two improved models were built in MATLAB and whether they can run correctly would be verified. Then we investigated whether the improved models have the same parameter properties as basic VM, which was found in the second step.

In the fourth step, a rule was design for judging whether the order parameter has converged and it would be applied into the three different VMs. Then we studied the convergence properties of the models and found out under what conditions the models converge fastest.

Chapter 2

About Vicsek Model (VM)

This chapter describes the details of three different VMs. The basic VM was proposed by (Vicsek et al., 1995) and the other two VMs was improved on the basic VM, to which the limited vision field angle and adaptive speed were added separately.

2.1 Basic Vicsek Model

A basic Vicsek model (VM) describes the movements of N self-propelled particles (to be regarded as mass points) in an $L * L$ map. The initial conditions of the VM meet the following rules: 1) The initial location of each point is randomly distributed in the map; 2) The initial movement direction of each individual is randomly distributed in $(-\pi, \pi]$; 3) The moving distance of each step of the particle is constant (i.e. the moving speed is constant). Let $\mathbf{x}_i(t)$ be the location of the i-th point at time t, and then the update rule follow as

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t)\Delta t \tag{2.1}$$

where Δt donates the time between step t and step $t+1$, $\mathbf{v}_i$ means the velocity of the i-th point at time t. Because the speed is constant for each point, the update of velocity is only related to the direction. For the i-th point, the the angle of direction at time t $\theta_i(t)$ is obtained from the expression

$$\theta_i(t+1) = < \theta_i(t) >_r + \Delta\theta_i \tag{2.2}$$

where $\Delta\theta_i$ is a random number that obeys a normal distribution from the interval $[-\frac{\eta}{2}, \frac{\eta}{2}]$ and it represents the noise in the update. The $< \theta_i(t) >_r$ donates the average directions angle of all points (including the point i) within the radius r and the it satisfies

$$< \theta_i(t) >_r = arctan[\frac{< sin\theta_i(t) >_r}{< cos\theta_i(t) >_r}]. \tag{2.3}$$

For a given size system, there are three free parameters: η, ρ and vel, where ρ donates the density of the system which $\rho = N/L^2$. The vel is the starting speed of each point and it also represents the distance of each point makes from t to $t+1$.

Vicsek et al. (1995) did the simulations according to the above rules obtained some interesting conclusions as shown in Figure 2.1.

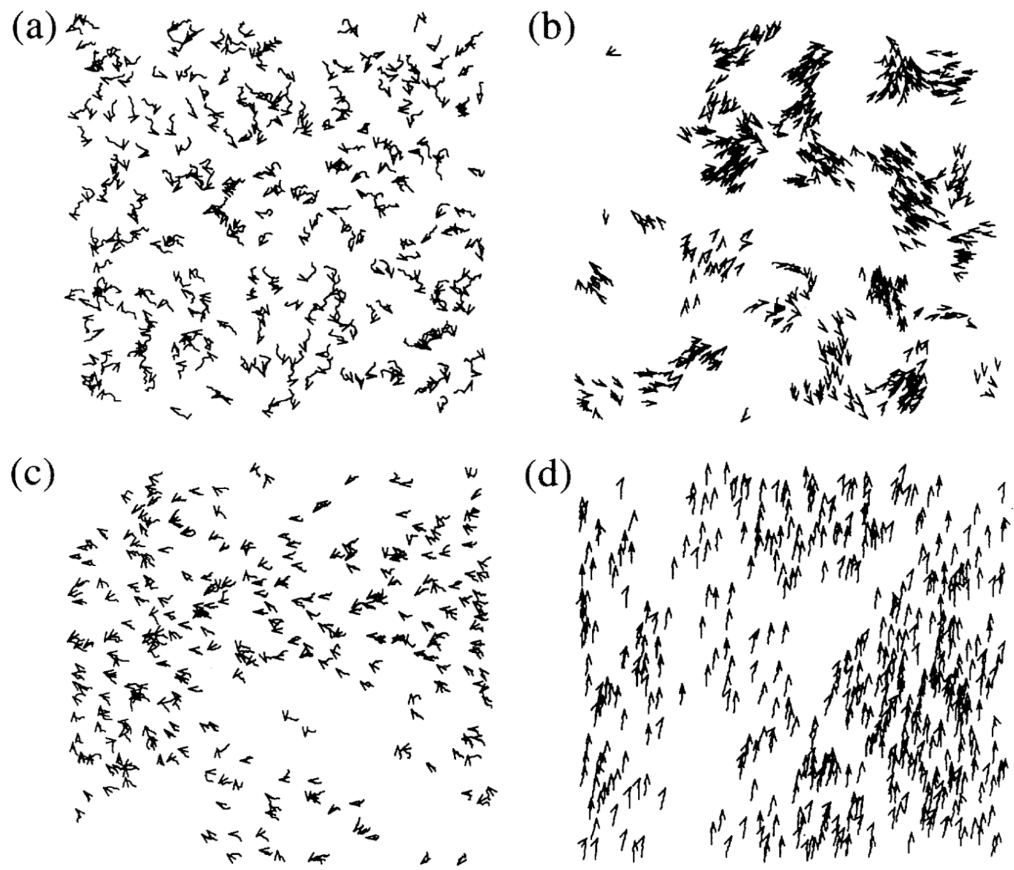

Figure 2.1: In this figure, the velocity of each particles $vel = 0.03$ and the arrows indicate the direction of particles movement. The number of particles is $N = 300$ in each case. (a) $L = 7$, $\eta = 2$ when $t = 0$; (b) $L = 25$, $\eta = 0.1$; (c) $L = 7$, $\eta = 2$; (d) $L = 5$, $\eta = 0.1$.

The simulation result shows that in the condition of low density and noise, the system tends to be stable after a number of steps and form groups in random directions (Figure 2.1(b)); For high density and noise, the particles seems moving randomly (Figure 2.1(c)); In the high density but low noise condition, the motion of particles converge to the same direction (Figure 2.1(d)).

In order to evaluate the order degree of the system, Dr.Vicsek introduced the average normalized velocity as the **order parameter**

$$v_a = \frac{1}{Nv}|\sum_{i=1}^{N}\mathbf{v}_i|. \tag{2.4}$$

If the direction of the particles in the system distributes randomly, the order parameter v_a tends to be zero. If $v_a \to 1$, all particles move in the same direction in the system.

From the simulation results of Figure 2.1, it can be known that the density and noise of the system have the impact on the moving state of the particles. Thus, they studied the effects of noise and density separately on v_a.

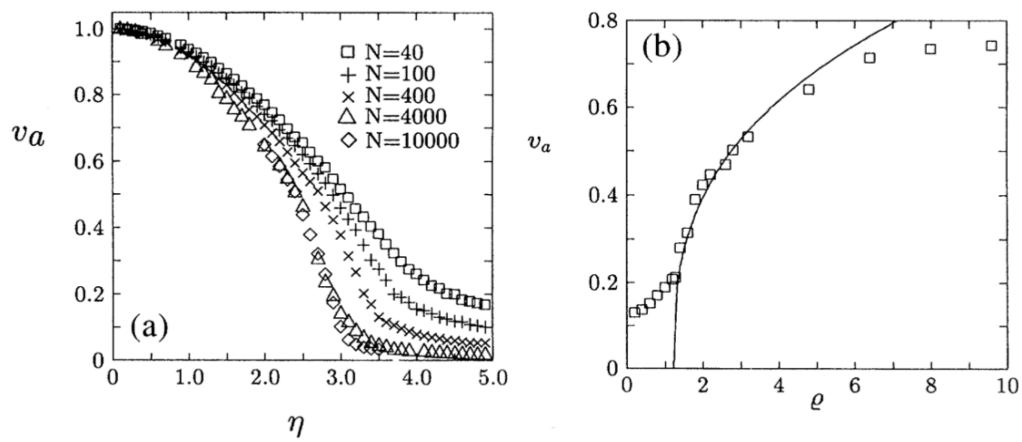

Figure 2.2: (a) The value of v_a versus the noise η in different number of particles with a constant density $\rho = 4$. (b) The $L = 20$ and the noise η is a constant value.

The Figure 2.2(a) shows that as the noise increases, the final value of v_a in the system will decrease. And only in the low noise situation, the particles in the system can reach a state of synchronized movement. The Figure 2.2(b) shows that when the noise is fixed, v_a increases with the increase of density i.e. the movement of the particles in the system can only be synchronized, when the density is greater than a certain value.

According to the simulations, Dr.Vicsek pointed out that when the density is fixed, as the noise increases, there is a critical value about noise η_c that makes $v_a = 0$. Also, when the noise is fixed, as the density increases, there is a critical value of density ρ_c that makes $v_a = 0$. Assuming that the critical values of noise and density when $L \to \infty$ are $\eta_c(\rho)$ and $\rho_c(\eta)$ respectively, then the average normalized velocity can be written as

$$v_a \sim [\eta_c(\rho) - \eta]^\beta, \ v_a \sim [\rho - \rho_c(\eta)]^\delta \tag{2.5}$$

where β and δ are critical exponents (i.e. $\beta = \ln(\eta_c)$ and $\delta = \ln(\rho_c)$). When the map size L is limited, the critical values η_c and ρ_c both depend on L, which are denoted as $\eta_c(L)$ and $\rho_c(L)$. Then they take the logarithm of Eq 2.5 and ploting $\ln v_a$ as a function of $\ln[(\eta_c(L) - \eta)/\eta_c(L)]$ and $\ln[(\rho - \rho_c(\eta))/\rho_c(L)]$ for some fixed value of η and ρ in Figure 2.3.

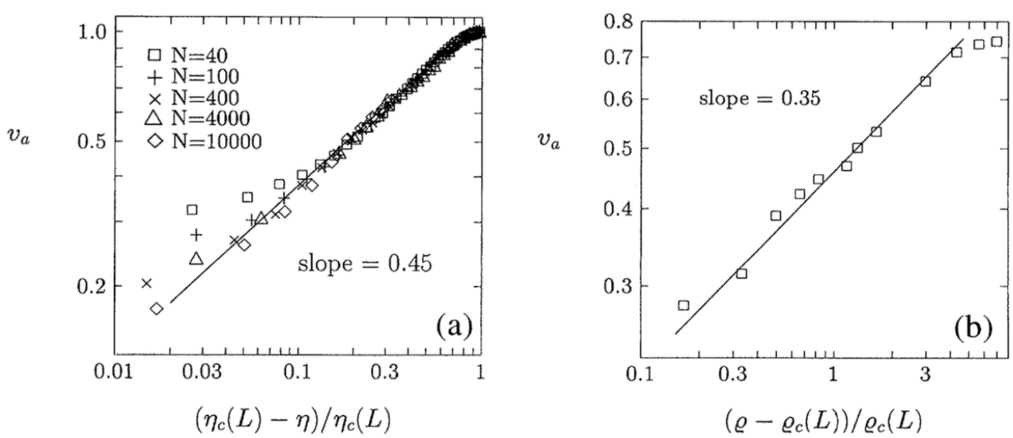

Figure 2.3: (a) is for $\rho = 4$; (b) is for $L = 20$ and $\eta = 2$.

According to the slope in Figure 2.3, they obtained the $\beta = 0.45 \pm 0.07$ and $\delta = 0.35 \pm 0.06$.

2.2 VM with Limited Vision Field

The vision field of basic VM is a circle which the particle is the circle center and r is the interacting radius (i.e. the vision field is 360-degree). However, in nature, creatures with a 360-degree vision field are rare. The most creatures have a limited vision field (e.g. vision field of horses is 350 degrees and vision field of sheep is from 320 degrees to 340 degrees) and the vision field is also determined by the intersection area (i.e. stereo vision zone) and the number of eyeballs.

In this project, we designed a simplified limited vision angle mechanism, i.e. ignoring the influence of cross vision and the number of eyeballs on the field of vision, only limiting the perception range of particles.

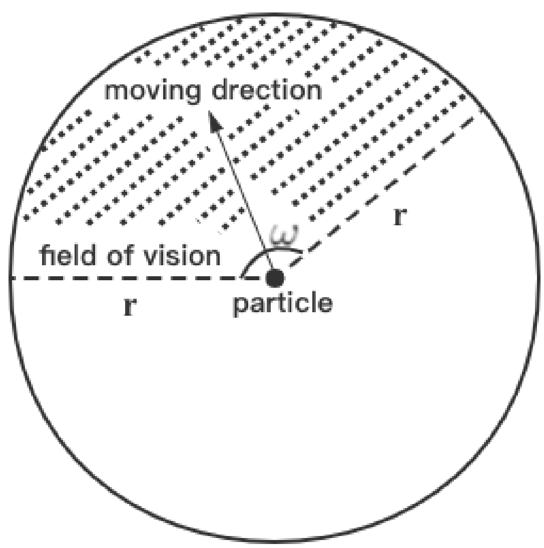

Figure 2.4: This is the interacting circle with radius r. The shaded part is the vision field of particles and the ω is the angle of vision field.

As the Figure 2.4 shown, the arrow indicates the direction of particle movement, the ω indicates the angle of vision field and the part covered by the shadow is the vision field to a particle. In the simulation process, the particle no longer considers the average direction of all other particles in the circle, but only considers the average speed in the shaded part in Figure 2.4 (i.e. The fan-shaped area with ω as the angle and r as the radius in the particle movement direction).

2.3 VM with Adaptive Speed

In basic VM, the speed (i.e. the distance moved in each step) is a fixed value (the value of starting velocity vel). Only the direction changes when the particles are moving. However, in a real particle system such as fish schools or sheep flocks, the moving speed of individuals is different, so giving each particle different moving speeds may shorten the time required for system convergence (the definition of convergence in this project will be explained in chapter 3). In this project, a modified VM with adaptive speeds for each particles proposed by Zhang et al. (2009) would be adopted. This model was designed to shortening the time required for the convergence of order parameter (Zhang et al., 2009).

In the original Vicsek Model, particles adjust their moving direction by other particles within the interacting radius (i.e. take the average direction as the next moving direction of the particle). However, if the movement of surrounding particles is very chaotic, only changing directions without adjust the moving speed will make the result become weird. For instance, assuming that particle A is moving to the $\pi/2$ direction and there is only one particle near it moving to the $-\pi/2$ direction. Then based on the update rule (Eq. 2.3), the moving direction of particle A at the next moment equals to 0, but the result makes no sense. Another disadvantage of the fixed moving speed is that when the particles move chaotically and the moving speed is fast, the communicated information might be wasted because the particle will move to a new position before fully communicate in the old position. When the particle movement is orderly and the moving speed is slow, it will make a lot of useless repeated communication. The particles are expected to have a lower speed when the other particles nearby moving chaotically and a higher speed when the surrounding particles move toward a similar direction. Thus in this model, Zhang et al. (2009) introduced the concept of order parameter ϕ_i

which can be written as

$$\phi_i = \frac{|\sum_{j=1}^{N_i} \mathbf{v}_j^{(i)}|}{\sum_{j=1}^{N_i} |\mathbf{v}_j^{(i)}|}, \tag{2.6}$$

for the i-th particle, N_i is the number of particles within the interacting circle; $\mathbf{v}_j^{(i)}$ is the velocity of j-th particle near the particle i. As the same with v_a, the value of the local order parameter ϕ_i is from 0 to 1 and the larger the value, the better the direction consistency. However, we cannot simply take the value of the local average normalized speed as the particle's speed at next moment because if the system is very chaotic, all particles will tend to stop since the local average speeds will be very small. In this project, an update rule designed by Zhang et al. (2009) would be used

$$v_i(t+1) = vel \times exp\ [\beta(\phi_i(t) - 1)], \tag{2.7}$$

where $v_i(t+1)$ is the speed of the i-th particle at time $t+1$, vel is the value of starting speed and β is the control parameter. If $\beta = 0$, the improved model will become a basic VM (i.e. $v_i = vel$ for all i at each moment); If $\beta > 0$, the range of v_i is $[vel\ e^{-\beta}, vel]$. When $v_a^{(i)} \to 0$ (i.e. the movement of other particles nearby is very chaotic), the speed of particle i at the next moment will be much smaller than the starting speed vel; when $v_a^{(i)} \to 1$ (i.e. the local particle movement direction tends to be the same), the speed of particle i at next moment will be very close to vel. Overall, the speed adaptation process of particles is

Algorithm 1 The speed of particle i at the next time

Input: Starting speed: vel; The speed of all particles at time t: $\mathbf{v}(t)$; The position of all particles at time t: $X(t)$; Interacting radius: r; Control parameter: β;

Output: The speed of particle i at the time $t+1$: $v_i(t+1)$;

1: Based on $X(t)$, calculate the distance between particle i and other particles and return the speed of particles whose distance is less than r as $\mathbf{v}^{(i)}$;

2: Calculate the local order parameter $\phi_i(t)$ based on Eq 2.6;

3: Calculate the $v_i(t+1)$ with vel, β and $\phi_i(t)$ based on Eq 2.7.

4: **return** $v_i(t+1)$;

Chapter 3

Experiment

3.1 Building Models

This section describes how to build the three VMs with MATLAB and verify whether the program can be run correctly.

3.1.1 Basic VM

The code of basic VM in this project is mainly improved on the code in *vicsek_model.m* provided on the Matlab official website: mathworks.com (2020). We have made some optimizations such as vectorization and parallel operations to the original code to improve the operating efficiency and added a function to calculate the average normalized speed. Except this part, all codes in this project are original.

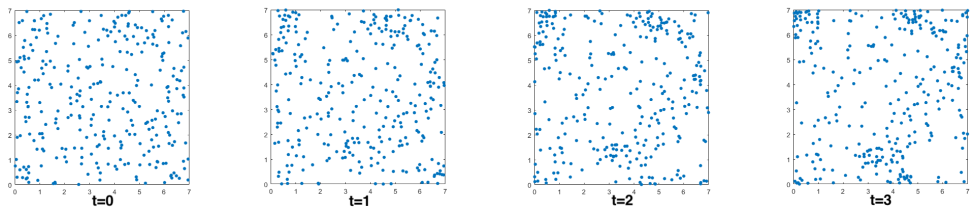

Figure 3.1: The *vicsek_model.m* can simply do the simulation of basic VM and plot the map of particles at each moment. The figure shows the simulation using *vicsek_model.m*, which plot the particles position at the initial moment ($t = 0$) and the first three steps ($t = 1, 2, 3$). The particles number $N = 300$, map size $L = 7$, noise $\eta = 2$, starting speed $vel = 0.3$ and interacting radius $r = 1$.

In the *vicsek_model.m*, the matrix M should be explained because it is very important for judging whether a particle is within the interaction circle of another particle. M is a format distance matrix and the value of M_{ij} is the distance between particle i and j. Because the distance between particles i and j is equivalent to the distance between j and i, M is a symmetric matrix and the diagonal elements are equal to 0. Each column (or row) of M represents the distance between the one particle and the remaining particles. The `Code box 1` shows how to generate the matrix M and update the direction angle θ using MATLAB.

```
%%% Code box 1
%% 'x' is the abscissa of particles (type: array);
%% 'y' is the ordinate of particles (type: array);
%% 'theta' is the direction angle of particles (type: array);
%% 'noise' is the noise parameter (type: double).

D = pdist([x' y'],'euclidean'); % Calculation of distance ...
    in the interacting circle
M = squareform(D); % Matrix representation for the ...
    distance between particles
[l1,l2] = find(0≤M & M<r); % Search for particles within ...
    the interacting circle

% update moving directions theta
for i = 1:N
    list = l1(l2==i);
    ave_theta(i) = ...
        atan2(mean(sin(theta(list))),mean(cos(theta(list))));
end

% update directions
theta = ave_theta + noise'*(rand(1,N) - 0.5);
```

The same parameters were used as in Figure 2.1 to simulate the model, and the result (Figure 3.2) shows that our model has the same properties as the model proposed by Dr. Vicsek in 1995 in the way of particles movement.

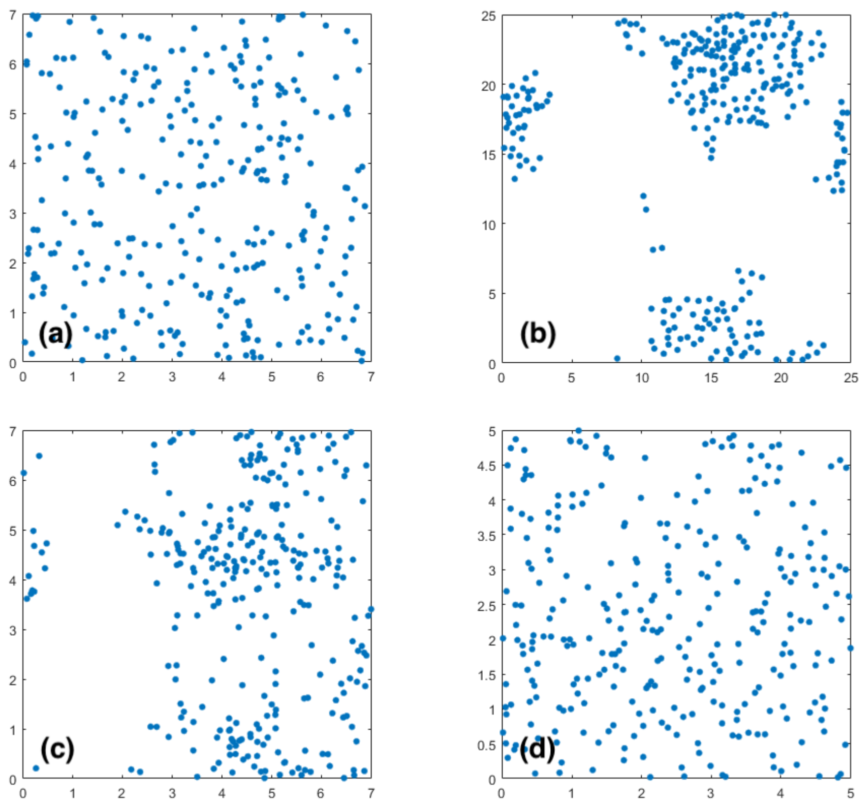

Figure 3.2: Observing the distribution of particles, we can see that the simulation results of (a) and (b) are similar to Figure 2.1; For (c), the average normalized velocity $v_a = 0.698$ means that the particles moving randomly as the same as Figure 2.1(c); For (d), $v_a = 0.999$ shows that all particles move in the same direction like Figure 2.1(d).

The research program rewrote the code of the basic VM into a function (shown in Appendix) which return the last value of v_a in the system. The function was run 100 times using the '`parfor`' statement for each result and took the arithmetic mean as the final value of v_a. In addition, the number of running times should be as large as possible because if the running times is not big enough, the data will be too discrete to catch the features (e.g. as the different between Figure 3.3(a) and 3.3(b)).

Using the same parameter in Figure 2.2(a), we achieved a very similar result when $N = 40, 100, 400$ (shown in Figure 3.3). However, in view of the limitations of hardware equipment, we cannot reproduce the simulation when $N = 4000$ and $N = 10000$ because the simulation time was too long to accept. But in this project, simulation experiments were not done with more than 400 particles and the good performance of the code when $N \leq 400$ is enough to show that it has implemented the model proposed by Vicsek et al. in 1995 very well.

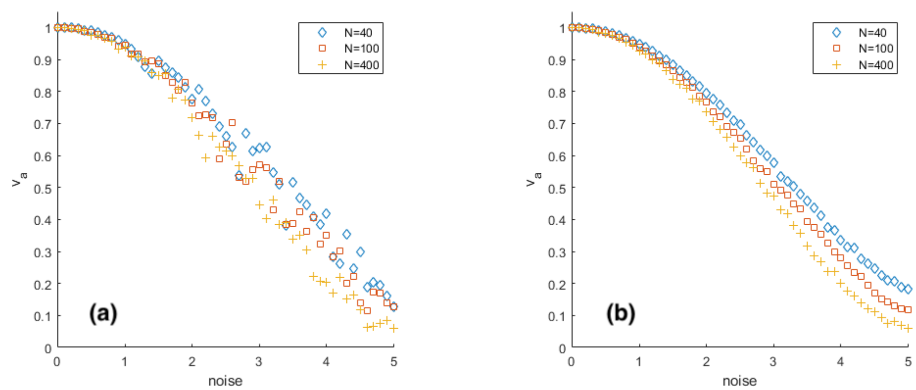

Figure 3.3: As the same with Figure 2.2(a), the starting velocity $vel = 0.03$, interacting radius $r = 1$ and the density of system $\rho = 4$ ($\rho = N/L^2$). For (a), each point was simulated 10 times and took the average value; For (b), each point was simulated 100 times and took the average value.

3.1.2 VM with Limited Vision Field

In the `Code box 1`, a distance matrix M was generated. Each element in M represents the distance between two particles (e.g. M_{ij} represents the distance between particle i and particle j). All the particles within the interaction circles can be located by finding the qualified particles in M (i.e. searching the value of element within $(0, r)$ and return their index). In this part, M was produced by the same size matrix A pointwisely to zero the elements corresponding to the particles outside the vision field angle ω controlled by the vision coefficient F with $\omega = 2\pi F$

($F \in [0, 1]$). This new matrix $A \circ M$ would then replace the original M and be used to locate the eligible particles. Since the program will search for the values greater than 0 and less than r in the matrix, the particles outside the vision field will be excluded.

This is the algorithm to calculate the value of A_{ij} when $i \neq j$; If $i = j$, $A_{ij} = 0$.

Algorithm 2 Determine whether particle j is in the vision field of particle i.

Input: The abscissa of particles i and j: $\mathbf{x}$; The ordinate of particles i and j: $\mathbf{y}$; Moving direction of particle i: θ_i; Vision coefficient: F.

Output: Whether particle j is in the vision field of particle i: A_{ij};

1: Use particle i as the origin O to center the coordinates;
2: Use the new coordinates of particle j to calculate the angle α between Oj and the x-axis;
3: Let $\Theta = |\theta_i - \alpha|$, if $\Theta \geq F$, $A_{ij} = 1$; if $\Theta < F$, $A_{ij} = 0$.
4: **return** A_{ij};

The test result of the model is shown in Figure 3.4. When the vision field of the particles is 360 degrees, the simulation result is the same as the basic VM; when the field of view of the particles is 0 degrees (i.e. can only see objects directly in front), the system is always in a disordered state.

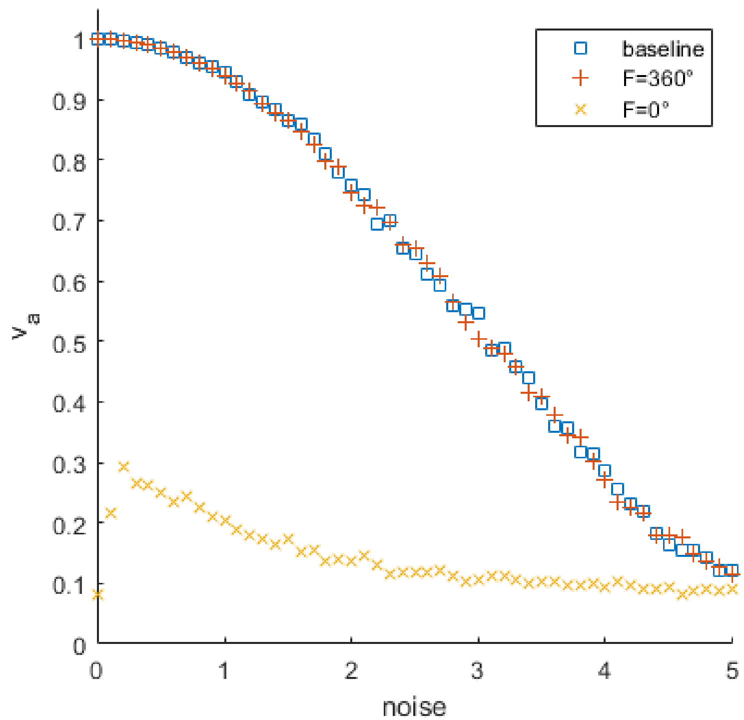

Figure 3.4: In the simulations, $N = 100$, $L = 5$, $r = 1$. The v_a points took the average of 100 independent simulations. The 'baseline' in the legend is simulated by the basic VM using the same parameters.

3.1.3 VM with Adaptive Speed

Because the method of judging the interaction relationship between particles is the same with basic VM, the study needs to add the part of the calculation of local order parameter ϕ inside `Code box 1`. Then the speed update rule was added in the bottom of `Code box 2`.

```
%%% Code box 2
%% 'x' is the abscissa of particles (type: array);
%% 'y' is the ordinate of particles (type: array);
%% 'theta' is the direction angle of particles (type: array);
%% 'noise' is the noise parameter (type: double).
%% 'v' is the current speeds for particles (type: array).
D = pdist([x' y'],'euclidean'); % Calculation of distance ...
    in the interacting circle
M = squareform(D); % Matrix representation for the ...
    distance between particles
[l1,l2] = find(0≤M & M<r); % Search for particles within ...
    the interacting circle

% update moving directions theta and local order parameter phi
for i = 1:N
    list = l1(l2==i);
    ave_theta(i) = ...
        atan2(mean(sin(theta(list))),mean(cos(theta(list))));
    xx = sum(v(list).*cos(theta(list)));
    yy = sum(v(list).*sin(theta(list)));
    phi(i) = ((xx.^2) + (yy.^2)).^0.5 ./ sum(v(list));
end

% update directions
theta = ave_theta + noise'*(rand(1,N) - 0.5);

% update speeds
v = vel*exp(beta.*(phi - 1));
```

The research exported the speed array **v** to judge whether the speeds of the particles would change. The result shows that the particle speed is always the same when $\beta = 0$ (i.e. the model becomes a basic VM), when $\beta \geq 1$, each element in the array **v** changes significantly, which proves the program can run correctly.

3.2 Study on Order Parameter

This section describes the effect of different starting speeds on order parameters of different VMs and tried to verity a conclusion declared by Vicsek et al. (1995) that "in a wide range of the velocities ($0.003 < vel < 0.3$), the actual value of vel does not affect the results".

For the basic VM, the result cannot perfectly verify the conclusion in Vicsek et al. (1995) about the range of starting speed. In our simulation, the range of starting speed can be much more wider. First, extremely small speed cannot change the curve shape of v_a (E.g. in Figure 3.5(a). The vel reaches the smallest number 0.0001 that can be calculated in MATLAB but the curve shape does not change). When the number of starting speed is bigger than the side length L of the map, the curve shape starts to change. A good instance is that when $N = 100$, $L = vel = 3.5$, the value of vel can still maintain a smooth decline. But when $L < v$, the curve tends to be bell-shaped (shown in Figure 3.5(b)). This phenomenon also occurs in different N number model (e.g. Figure 3.5(c)) and different L model (e.g. Figure 3.5(d)).

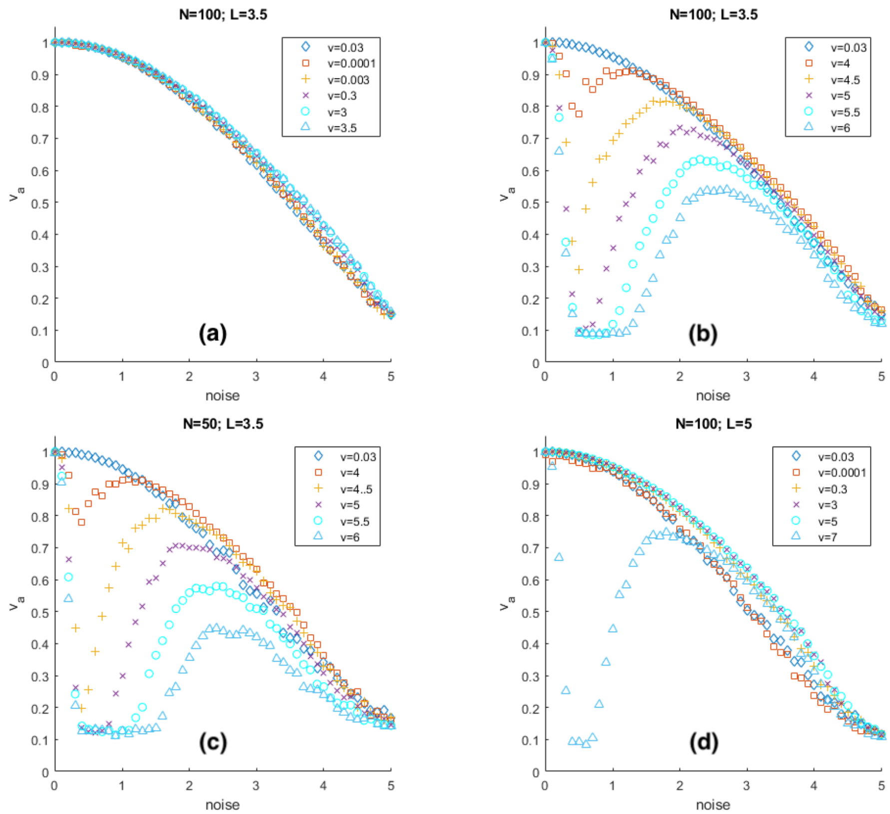

Figure 3.5: For (a) and (b), $N = 100$ and $L = 3.5$; For (c), $N = 50$ and $L = 3.5$; For (d), $N = 100$ and $L = 5$. Each point takes the average value of 100 independent simulations.

The correlation test on the data also supports the above conclusion. Firstly, Lilliefors test confirm that all the data obey normal distribution. Then take the simulated data when $vel = 0.03$ as the baseline (this is because the most used starting speed in the paper proposed by Vicsek et al. (1995) was 0.03, so we regard 0.03 as the most "typical" starting speed for the vicsek model) and did the analysis of variance between baseline and data with other speeds (we used different speeds and map length data to do the test, but the density are the same in each data to eliminate the influence of irrelevant variables). The P-value for each test is shown in Table below.

L\vel	*0.0001*	*0.003*	*0.3*	*3*	*5*	*7*	*9*
3.5	0.95	0.95	0.63	0.55	**4.1e-04**	**2.6e-16**	**3e-19**
5	0.87	0.89	0.51	0.38	0.36	**0.01**	**1.1e-11**
7	0.66	0.82	0.5	0.32	0.32	0.31	**0.01**

The table shows that when the values of starting speed is outside the range $(0.003, 0.3)$ but less than L, there is no significant difference with the baseline (because $P > 0.05$). When $vel > L$, the P-value is smaller than 0.05. It means that the data is significantly different from the baseline.

For the improved models, the baseline was the same to analyze the influence of different parameters on order parameters. The VM of limited vision shows that if one narrows the field of vision, the phase transition shifts and may ultimately even disappear. It also means that the smaller the particle's vision angle, the greater the influence of noise on the order of system. Figure 3.6(a) shows the result simulated by vision angle ω varies from 0 degree to 300 degree (F is the vision coefficient that $\omega = 2\pi F$). The result of VM with adaptive speed is almost the same with the baseline when $0 < \beta \leq 5$. When $\beta > 5$, the order parameters are significantly higher than the baseline in the high noise area ($noise > 3$) (Figure 3.6(b)).

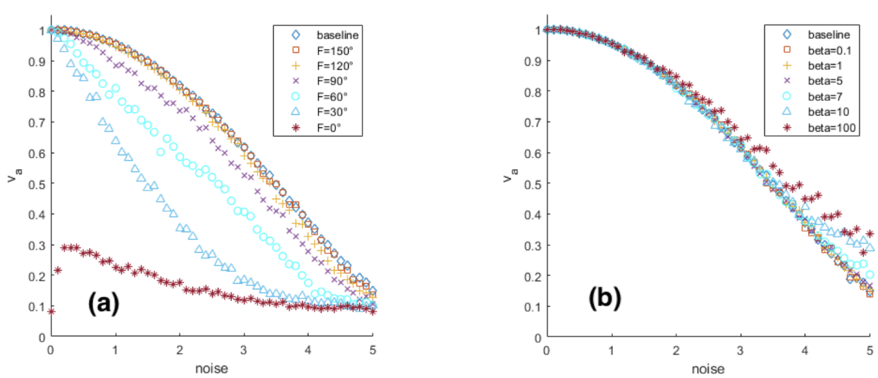

Figure 3.6: For both images, $N = 100$, $L = 3.5$, $r = 1$, $vel = 0.03$ and each v_a took the average of 100 independent simulations.

It is found that when the value of starting speed becomes extreme, the shape of the order parameter points would undergone the same change as the basic VM, which means when the value of map side length is greater than the value of starting speed, the order parameter points will tend to become a h-shape (Some samples are shown in Figure 3.7).

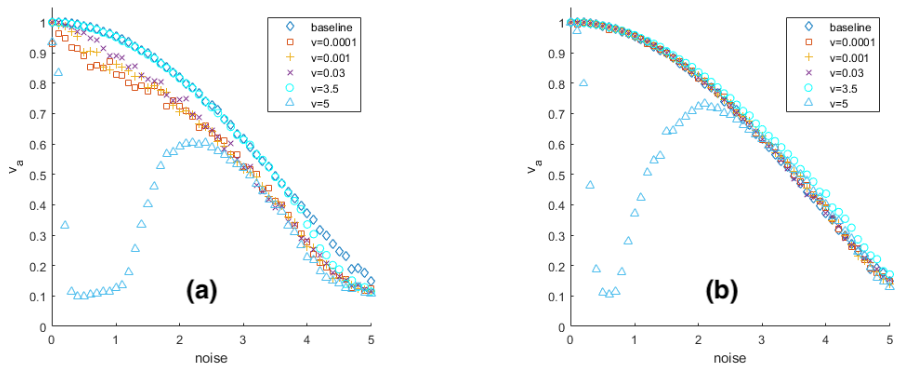

Figure 3.7: $N = 100$, $L = 3.5$, $r = 1$ and each v_a took the average of 100 independent simulations. For (a), the vision coefficient is 0.5 (i.e. the range of 180 degrees in front). It can be seen that when $vel \leq 3.5$, the order parameter points increase slightly with the increasing speed but not obviously. When $vel > 3.5$, the shape of order parameter points begin to become an h-shape. For (b), $\beta = 1$. The results are the same with the basic VM.

3.3 Study on Convergence Time

In this section, the definition of the convergence time T would be given firstly and how to obtain the T value in this project will be described. Then the effect of different parameters on the convergence time would be presented.

The "convergence" here means the value of the order parameter v_a converge to a specific value after a period of time and the convergence time T represents the number of steps used for the order parameter convergence. When the $noise = 0$ in the system, the order parameter v_a will always converge to a value very close to 1 (0.999 ± 0.0005 in 10000 runs). When the noise is small, the order parameter value will fluctuate in a small range. As the noise increases, the fluctuation of order parameter value will become larger and cannot be considered as convergent (Figure 3.8(a)). As we did not find out what degree of fluctuation of the order parameter can be defined as convergence, and the evolution of models will be very complicated when there is noise, so this project only considers the situation without noise. In this project, the first number of steps reaching 0.999 as T was set. Also, T can be regarded as the time when the particles' moving direction is synchronized in a system without noise.

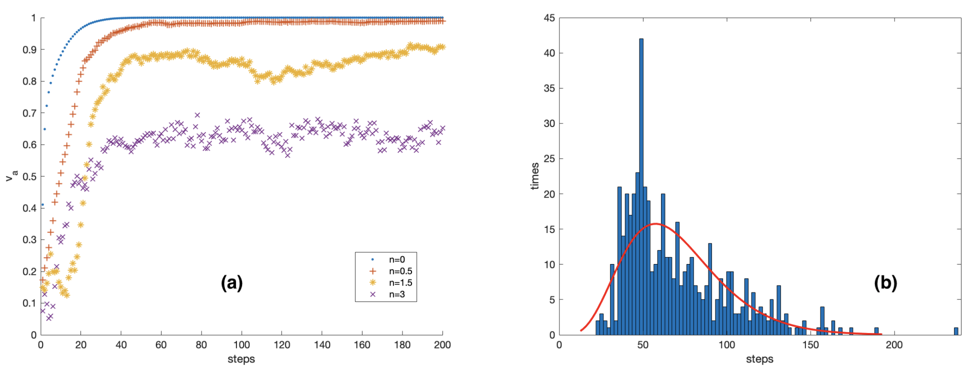

Figure 3.8: (a) shows the value of order parameter from first step to the 200-th step with different noise. The histogram with 100 bins in (b) shows the distribution of T value after 500 independent simulations under the same parameters without noise. The red curve is a fit of the distribution using the Gamma density function. Both Figures used $N = 100$, $L = 5$ to simulate.

According to a large number of independent simulations, the skewness of the T value distribution under the same parameters is large (often greater than 1) and they often have some very large outliers. These outliers are accidental phenomena in multiple simulations but it will have a great impact on the arithmetic mean (A good example is that in Figure 3.8(b), the arithmetic mean of the data is 70.052 and the median is 61. According to the histogram, we can see that most of values are concentrated around 40 to 60, so the median can better reflect the data). Therefore, we used the median of 500 repeated simulation results as the T value points in this project.

In the experiment, the influence of the interacting radius r and the particles' number N on the convergence time was firstly analysed. For the selection of parameters, the randomness of the system will become very large when there are too few particles, which will cause the results of independent repeated simulation to vary greatly; When there are too many particles in the system, the simulation process will be so slow that it cannot be completed in an acceptable time. Thus, the number of particle N from 50 to 200 was decided to be used. Within this range, the randomness and efficiency of the simulation are acceptable. For the interacting radius r, we chose $[0.5, 1]$ as the experimental range. When $r < 0.5$, because the information received by each particle is too little, many particles appear to behave like random walks, which greatly increases the randomness; when $r > 1$, the area ratio of the interaction circle to the map is too large , Which makes the particles obtain too much information. It will make the model lose locality and reduce simulation efficiency.

For the basic VM, according to the simulations, we found that When r increases, the convergence time T will shorten; when the number of particles increases (in the case of a fixed map area L), the convergence time will also shorten (Figure 3.9 shows a example with $L = 5$).

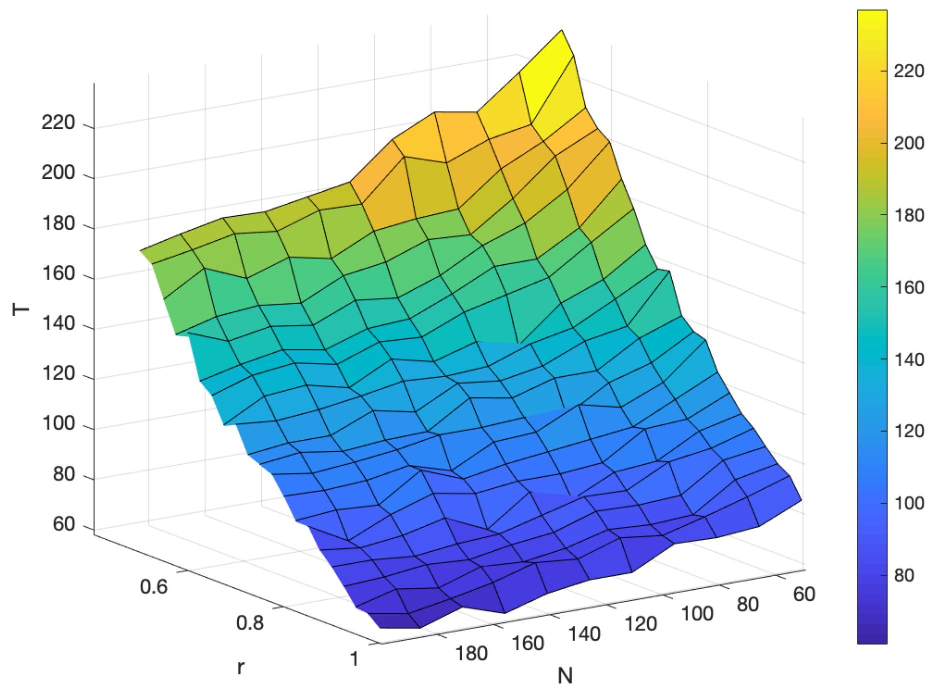

Figure 3.9: In this simulation, $L = 5$ and $v = 0.03$. The maximum value of T is 239, and the corresponding parameter $(N, r) = (50, 0.5)$; the minimum value of T is 62, and the corresponding parameter $(N, r) = (200, 1)$.

By performing linear regression on the data in Figure 3.9, we found that when the value of N is fixed, T and $1/r^2$ show a similar proportional relationship with a y-intercept (Figure 3.10(a)) and the slope of the fitted line will decrease as N increases. When we fixed the r value, T and the $1/N$ also show a positive relationship. When the interacting radius is smaller, the slope of fitted is greater (Figure 3.10(b)).

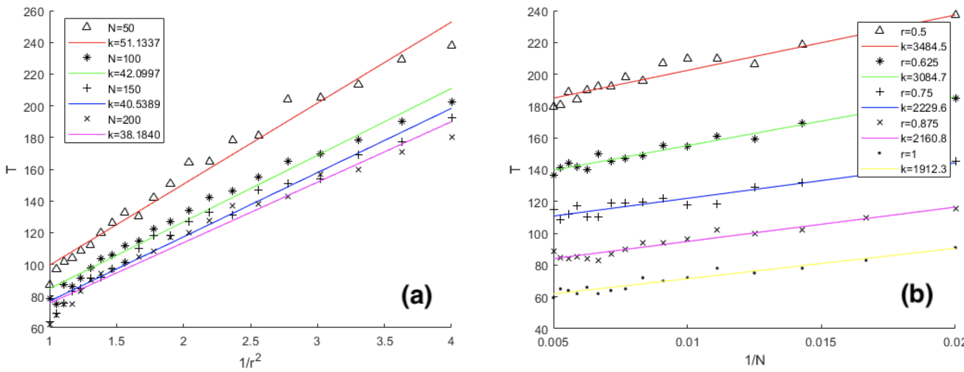

Figure 3.10: The 'k' in the legend shows the slope for each fitted line. The relationship between r and T can be written as $T = \frac{k_1}{r^2} + b_1 + \varepsilon$ where the k_1 and b_1 are coefficients related to N. Also, the relationship between N and T is $T = \frac{k_2}{N} + b_2 + \varepsilon$ where the k_2 and b_2 are coefficients related to r. ε is the error between the fitted line and the real data.

In two improved models, the experimental results show that after fixing the control parameters respectively (F and β), the relationship between r, N and T is the same as the basic VM (i.e. as r and N increase, the convergence time T will shorten). Thus, we fixed the particles number N, interacting radius r and side length L, only change the control parameter in order to find out the value of a set of control parameters with the highest convergence efficiency. In the next experiments, we choose $N = 100$, $L = 5$ and $r = 1$ for the simulations because these parameters can get reasonable data without consuming too much time.

For the VM with limited vision field, the simulation result shows that When $F \in [0.66, 0.76]$ (i.e. the vision field angle ω from 237.6° to 273.6°), the system has the highest convergence efficiency. The best vision angle of this model is 244.8°($F = 0.68$). At this angle, the movement direction of particles can be synchronized in the shortest time ($T = 48$) and is 30% faster than basic VM (the basic VM needs to evolve at least 71 steps under the same parameters to achieve synchronization).

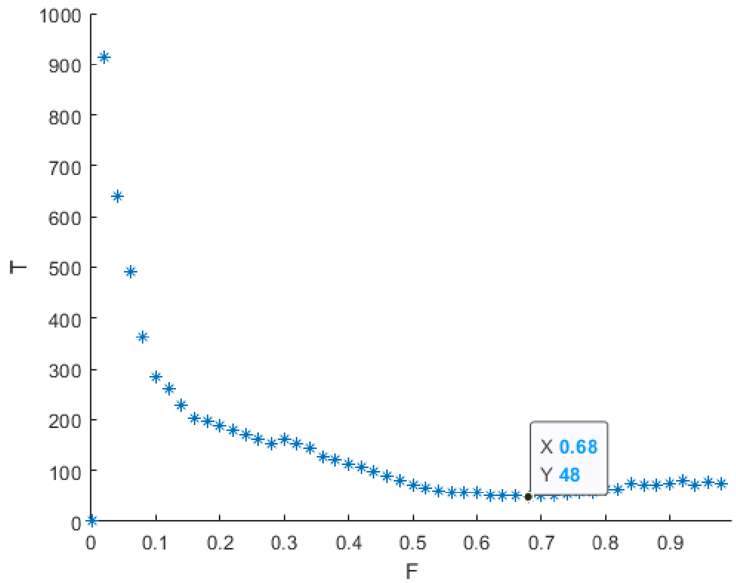

Figure 3.11: The picture shows the convergence time of the system at different vision angles. It is important to note that is the system can finally achieve synchronization at any angle except in 0°, but when the vision angle is very narrow, the system needs to run several hundred steps to make the order parameter converge. When the vision angle is greater than 180°, the convergence efficiency of limited vision model gradually approaches and surpasses the basic VM. At 0°, the order parameter cannot converge because the particles are almost blind. The display in $F = 0$ in the figure is due to the algorithm bug of the program.

For the VM with adaptive speed, it was found that when β changes, the T value changes but there is no evident law and the adaptive speed model is also no evident advantage in convergence efficiency compared with basic VM.

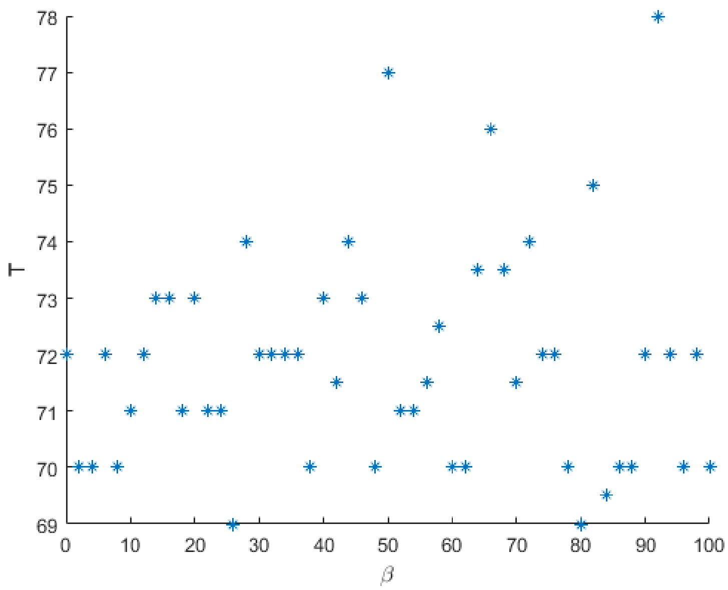

Figure 3.12: When $\beta = 26$ and 80, the T value reaches the smallest with 69. Compare with the fastest convergence time in basic VM ($T = 71$, the improvement of adaptive speed model is not obvious.

Chapter 4

Conclusion and Future Works

4.1 Conclusion

In this project, the demo code provided in mathworks.comwas first meliorated. The function of calculating order parameters was added, and operating efficiency was improved by pre-setting parameter sizes and vectorized operations. Then, two modified models were proposed. The VM with vision field has added the vision field angle ω controlled by the vision coefficient F to imitate the real vision field of animals in nature. In the VM with adaptive speed, each particle in the system has different moving speeds and they will change their movement strategy according to the neighbors (i.e. Adjust the movement speed according to the local order parameter ϕ of the nearby particles in the interacting circle. The updated speeds will not be greater than the starting speed vel preset by the system).

Through the study of order parameters, three main conclusions are proposed. (1)The starting speed vel can be selected in a wider range. Vicsek et al. (1995) pointed out that when the vel is in the range of $(0.003, 0.3)$, the actual value of vel does not affect the results. However, in our simulation, when vel is out of the range (e.g. $vel = 0.0001$ and $vel = 3$ in the figure 3.5(a)), the order parameter points did not changed significantly. (2)When the value of vel is larger than the map length L, the value of order parameter changes dramatically. When the $vel > L$, the order parameter will drop drastically as the noise increases, then raise, and then drop again. The whole order parameter points present an h-shape and the larger the value of L, the more obvious the h-shape. (3)The above two conclusions can be applied to the modified models. Through a number of simulations, we found that the VM with limited vision field and adaptive speed shows the same properties described above.

By the study of convergence time, we have three discoveries. (1)Both density ρ ($\rho = N/L^2$) and interacting radius r have negative correlations with T value. The larger the value of ρ and r, the faster the convergence time. This conclusion is applicable to the all the three VMs. (2)For the basic VM, the relationship between r and T can be written as $T = \frac{k_1}{r^2} + b_1 + \varepsilon$ where the k_1 and b_1 are coefficients related to N. Also, the relationship between N and T is $T = \frac{k_2}{N} + b_2 + \varepsilon$ where the k_2 and b_2 are coefficients related to r. (3)The optimal vision field of VM with limited vision is 244.8°. In this condition, the model has the highest convergence efficiency and is 30% faster than basic VM.

4.2 Future Works

In this project, the influence of starting speed on order parameter was only studied. The relationship between other basic parameters such r and L is a future goal of work. Also, for the VM with limited vision field, more different mechanisms of vision field (e.g. some birds can only see objects on both sides) could be introduced to simulate the movement of specific animal swarms. The performance of VM with adaptive speed in convergence is not as good as expected, and there seems to be no intuitive connection between the control parameter β and the convergence time T. The convergence efficiency of this model is not much higher than that of basic VM. Thus, the in-depth study and improvement of this model is also a work direction. Moreover, the study on convergence in this project is all carried out in a noise-free state, because noise will increase the randomness of the system, making the distribution of order parameter points very scattered, and it is difficult to judge whether the convergence is happened. In the future work, the further study can try to measure the degree of dispersion of order parameter points to obtain a value (e.g. standard deviation). If the value has been less than a threshold for a period of time, it can be determined that the system has reached convergence and this method could be used to study the system in a low-noise environment. Moveover, this project only studies the Vicsek model on the two-dimensional level. However, many collective motions in nature occur on the three-dimensional level (e.g. the movement of a school of fish). Therefore, it will be a very interesting to extend the model to 3D and study the properties in 3D vicsek model.

Appendix A

The Code of Simulation

```
function [v_a,T] = simulation(TMAX,N,L,r,noise,vel,es,F,beta)
%This function is used to simulate different Vicsek models
% Input:  TMAX  - Maximum number of steps in system evolution
%         N     - The number of particles in the system
%         L     - The side length of the map
%         r     - The interacting radius
%         noise - Noise coefficient of the system
%         vel   - The starting speed
%         es    - This is a Logical value, 1: the function
%                 will stop when the order parameter
%                 convergence, 0: the function will not stop
%                 when the order parameter convergence
%         F     - A Coefficient of vision field angle
%                 (omega=2*pi*F) in the VM with limited vision
%                 field; When simulating other VMs, this
%                 parameter should be set to []
%         beta  - A control parameter of the VM with adaptive
%                 speed; When simulating other VMs, this
%                 parameter should be set to []
%
% Output: v_a   - The normalized average velocity i.e. the
```

```
%                  order parameter in the model.
%
%          T     - The number of iterations of the model when
%                  v_a converges

%% Boundary condition
PERIODIC=1; % 1: periodic boundary condition, 0: unlimited

%% Initial condition
dt=1;
x = L*rand(1,N);
y = L*rand(1,N);
v = vel*ones(1,N);
theta = 2*pi*(rand(1,N)-0.5); % randomly direction
Phi = zeros(1,TMAX);
% Precision control
prs_size = 100;
prs_num = 0.01;

%% Simulation process
for time = 1:TMAX
    tmp_x = zeros(1,N);
    tmp_y = zeros(1,N);
    ave_theta = zeros(1,N);
    phi = zeros(1,N);
    D = pdist([x' y'],'euclidean'); % Calculation of ...
        average angle in the interacting circle

    % Periodic boundary
    if PERIODIC==1
        tmp_x(x<r) = L + x(x<r);
        tmp_x(x>L-r) = x(x>L-r)-L;
        tmp_x(r≤x & x≤L-r) = x(r≤x & x≤L-r);
        tmp_y(y<r) = L + y(y<r);
        tmp_y(y>L-r) = y(y>L-r)-L;
        tmp_y(r≤y & y≤L-r) = y(r≤y & y≤L-r);
        tmp_D = pdist([tmp_x' tmp_y'],'euclidean');
```

```
        D = min([D; tmp_D]);
    end

    M = squareform(D); % Matrix representation for the ...
        distance between particles
    if ¬isempty(F), M = M.*limitView(F,theta,x,y,N); end  ...
        % delete the points outside the vision angle omega

    % calculation of average directions
    [l1,l2] = find(0≤M & M<r);
    for i = 1:N
        list = l1(l2==i);
        ave_theta(i) = ...
            atan2(mean(sin(theta(list))),mean(cos(theta(list))));
        if ¬isempty(beta)
            xx = sum(v(list).*cos(theta(list)));
            yy = sum(v(list).*sin(theta(list)));
            phi(i) = ((xx.^2) + (yy.^2)).^0.5 ./ sum(v(list));
        end
    end

    %% Update
    x = x + v.*cos(theta).*dt;
    y = y + v.*sin(theta).*dt;

    if PERIODIC==1
        x(x<0) = L + x(x<0);
        x(L<x) = x(L<x) - L;
        y(y<0) = L + y(y<0);
        y(L<y) = y(L<y) - L;
    end

    % update directions
    theta = ave_theta + noise'.*(rand(1,N) - 0.5);

    % update speeds
    if ¬isempty(beta), v = vel.*exp(beta.*(phi - 1)); end
```

```
    %% Compute the normalized average velocity
    vel_x = sum(v.*cos(theta));
    vel_y = sum(v.*sin(theta));
    Phi(time) = ((vel_x.^2) + (vel_y^2)).^0.5 ./ sum(v);

    if time≥prs_size
        conv_Phi = Phi(time-prs_size+1:time);
        if (max(conv_Phi)-min(conv_Phi)<prs_num) && ...
            (es==1), break, end
    end
end

if es==1
    v_a = mean(conv_Phi);
    conv_ctr = abs(normalize(conv_Phi,'center'));
    I = find(conv_ctr==min(conv_ctr), 1 );
    T = time-prs_size+I;
else
    v_a = Phi(end);
    T=NaN;
end
```

Bibliography

T. H. (2020). Vicsek model simulation. Website. https://www.mathworks.com/matlabcentral/fileexchange/64208-vicsek-model-simulation Retrieved August 17, 2020.

M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, et al. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the national academy of sciences*, 105(4):1232–1237, 2008.

I. D. Couzin, J. Krause, et al. Self-organization and collective behavior in vertebrates. *Advances in the Study of Behavior*, 32(1):10–1016, 2003.

I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision-making in animal groups on the move. *Nature*, 433(7025):513–516, 2005.

A. Czirók, H. E. Stanley, and T. Vicsek. Spontaneously ordered motion of self-propelled particles. *Journal of Physics A: Mathematical and General*, 30(5):1375, 1997.

V. Gazi and K. M. Passino. Stability analysis of social foraging swarms: combined effects of attractant/repellent profiles. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 3, pages 2848–2853. IEEE, 2002.

F. Ginelli. The physics of the vicsek model. *The European Physical Journal Special Topics*, 225(11):2099–2117, 2016.

F. Heppner and U. Grenander. A stochastic nonlinear model for coordinated bird flocks. *The ubiquity of chaos*, 233:238, 1990.

Y. Inada and K. Kawachi. Order and flexibility in the motion of fish schools. *Journal of theoretical Biology*, 214(3):371–387, 2002.

A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6):988–1001, 2003.

S. Mu, T. Chu, and L. Wang. Coordinated collective motion in a motile particle group with a leader. *Physica A: Statistical Mechanics and its Applications*, 351 (2-4):211–226, 2005.

R. Olfati-Saber and R. M. Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC world congress*, volume 15, pages 242–248. Barcelona, Spain, 2002.

R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533, 2004.

J. K. Parrish and W. M. Hamner. *Animal groups in three dimensions: how species aggregate*. Cambridge University Press, 1997.

B. L. Partridge. The structure and function of fish schools. *Scientific american*, 246(6):114–123, 1982.

C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.

R. A. Simha and S. Ramaswamy. Hydrodynamic fluctuations and instabilities in ordered suspensions of self-propelled particles. *Physical review letters*, 89(5): 058101, 2002.

H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents part i: dynamic topology. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 2, pages 2016–2021. IEEE, 2003.

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.

J. Zhang, Y. Zhao, B. Tian, L. Peng, H.-T. Zhang, B.-H. Wang, and T. Zhou. Accelerating consensus of self-driven swarm via adaptive speed. *Physica A: Statistical Mechanics and its Applications*, 388(7):1237–1242, 2009.