BEYOND PHASE TRANSITIONS: AN ALGORITHMIC APPROACH TO FLOCKING BEHAVIOR

Garett Brown

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Manuel Berrondo, Advisor

Department of Physics and Astronomy

Brigham Young University

April 2017

ABSTRACT

BEYOND PHASE TRANSITIONS: AN ALGORITHMIC APPROACH TO FLOCKING BEHAVIOR

Garett Brown
Department of Physics and Astronomy, BYU
Bachelor of Science

The complexity and pattern found in animal aggregations, such as starling murmurations, reveals emergent phenomena which arise from the simple, individual interactions of its members. Simulated in a two-dimensional algorithmic model, self-driven particles (boids) group together and display emergent flocking characteristics. The model is based on the ideas of consensus and frustration, where consensus is a nonlinear topological averaging that drives the boids toward one of three unique phases, and frustration is a perturbation that pushes the boids beyond these simple phases and toward disordered behavior. The nonlinearity merged with the perturbation produces characteristics which go beyond the dynamic interplay of global and local phase transitions. The emergent results are interpreted in terms of global and local order parameters, and correlation functions. The results also strongly agree with observational data and empirical analysis.

Keywords: Emergence, Self-Organization, Collective motion, Flocking Behavior, Boids, Algorithm, Mathematica, Matlab, Code, Phase, Phase Transition, Simulation, Chaos

ACKNOWLEDGMENTS

I would like to thank my wife, Jessica, who is always first in my life. She has been a strength and a support to me and her courage is inspiring. I would also like to thank Dr. Berrondo for the opportunity to work with him and for the mentorship that he provided to me. I'm grateful for his help and patience and I'm grateful to him for often correcting my mistakes. I would also like to thank John Ellsworth for giving me hands on experience in experimental physics and then being there for me when I decided that wasn't the path that I wanted to take. I would also like to thank Dr. Stephens for helping me lead BYU's Astronomical Society and for submitting so many letters of recommendation on my behalf. I would also like to thank Dr. Canizares of the University of Cambridge for the opportunity to do original research with her in gravitational waves. Additionally, I would like to thank Dr. Transtrum for being there when I had questions and for helping me find my path into theoretical physics. I would also like to thank Dr. Rees who, along with Dr. Transtrum, agreed to be on my committee and who has also imparted much wisdom regarding good pedagogical practices. I would also like to thank Dr. Colton, Dr. Gus Hart, Dr. Turley, and Dr. Davies for their excellent pedagogical practices, guidance, and extra time they have spent with me, asking me how I was doing, and imparting of their wisdom. And finally, where would I be without my good friend, Kolten Barfuss. Together we have worked hard to succeed in physics and mathematics, seeking to learn beyond our assignments and always sharing with each other the new things we learn in our research. In addition to the twenty-four classes that we took together, he has become my best friend.

Thank you everyone.

Contents

List of Figures

Chapter 1

Emergence

Have you ever wondered why some birds flock? Or even how a group of birds in a flock manage to move with such unity or precision [1]? The usual approach that would lead to an understanding would be to simplify the problem by breaking it down into more easily understood pieces, and then reassemble those pieces to recreate the whole. While certainly a useful method, and one which has provided answers to a variety of other physical phenomena, reductionism has its limits.

An alternative to reductionism is based on the concept of emergence [2]. Emergence considers a multi-element system which, as a whole, exhibits a set of behaviors, and the elements within the system also exhibit a set of behaviors different from the whole. With this in mind, the accepted definition of emergence centers around the notion that emergence is a phenomenon or behavior that arises from a system from the simpler, smaller interactions of the entities of that system which do not exhibit such behavior themselves. While this is a very broad and useful definition, we briefly present varying contexts for the definition and elaborate further upon the subject.

It is asserted by some that an ideal gas is an emergent phenomenon [3]. Consider the basic derivation of the ideal gas law, beginning with point-masses that behave as billiard balls in an enclosed system, possessing no inherent potential energy and assuming that the collisions and interactions with the walls and other particles to be instantaneous and elastic. Yet, with these assumptions, one can derive a relationship between pressure, volume, temperature, and the number of particles, without ever solving Newton's second law. The debate goes that since the individual particles do not possess these macroscopic qualities (pressure, volume, and temperature), then these macroscopic qualities are emergent properties of the system. However, because the ideal gas law can be written in a closed form ($PV = nRT$), we (my advisor and I) do not consider it to be emergent. The derivation of the ideal gas law is not much more than the sum of its parts and its simplicity impedes it from accounting for phase transitions. Thus, we argue that complexity and nonlinearity are required for emergence.

Does emergence require a chaotic system? Here it seems to make the most sense to turn our attention to meteorology and the ever-unpredictable weather [4]. The complexity and irregularity of the atmosphere causes us to attach to it phrases such as erratic and random. However, it seems that our ignorance of the precise position and momentum of each atmospheric particle is the true culprit in our inability to predict weather. While that might be, it seems more likely that there exists some threshold of complexity that, when crossed, reveals the true limits of a reductionist approach to a deterministic system. As those from the Midwest know, during the spring time, when the cold fronts of the North collide with the warm fronts of the South, tornados emerge. When the basic ingredients are mixed together, it seems possible, and likely, that this kind of severe weather will appear out of thin air, but when the number of issued tornado warnings are compared to the number of actual tornados the evidence suggests that chaotic ingredients *alone* do not guarantee emergent results.

Does this need for more than a combination of ingredients mean then that there is something deeper, something non-trivial, or something completely indiscernible that is inherent in complex systems that produce emergent phenomena? We believe that there is. It seems to us that if emergent phenomena were able to be defined by a single metric (like $PV = nRT$), then there would be no inherent 'mystery' that inspires us to call something emergent.

Biology seems to possess the best examples of emergent phenomena [5], which is why we have focused our attention there. The definition of ‘life’ is extremely difficult to define unanimously [6]; however, it is agreed that life is emergent. Even though there is consensus on what the building blocks of the universe are, the complex combination of elementary particles somehow produces organisms which can metabolize and maintain a homeostasis state. However, with so many biological emergent properties to explore, we focuse our study on animal aggregation, particularly ‘flocking’ behavior (see Fig. 1.1). Whether that flocking is the result of a group of birds, a group of fish, or a collection of insects, in this paper we consider all types of animal aggregations to be some form of ‘flocking.’

Our attempt to create emergent behavior has been to use the standard, simplified, reductionist approach, by creating rules that regulate a system of ‘particles’ which interact with each other to produce a wholly different set of emergent properties. We do this by combining the system together in a nonlinear way, which results in phases and phase transitions (Chapter 2.2). We move beyond these phases and phase transitions by introducing more complexity that breaks the symmetry of the system (Chapter 2.3). We then discuss the different tools used to probe and understand the resulting emergent phenomenon (Chapter 3). We discuss the empirical results of our model and its robustness (Chapter 4). Finally, we discuss the realism of our model and its agreement to observational data (Chapter 5). Our approach to understanding emergence is to stand with one foot in reductionism, one foot in emergence, and try to bridge the gap by using the best tools from each method, and as we continue to probe more deeply, it seems that as the complexity of a system increases, the more room there is for emergent phenomena.

[1] CC BY-SA 2.0 can be found at https://creativecommons.org/licenses/by-sa/2.0/deed.en

WikiCommons can be found at https://commons.wikimedia.org/wiki/File:Starling_murmuration.jpg

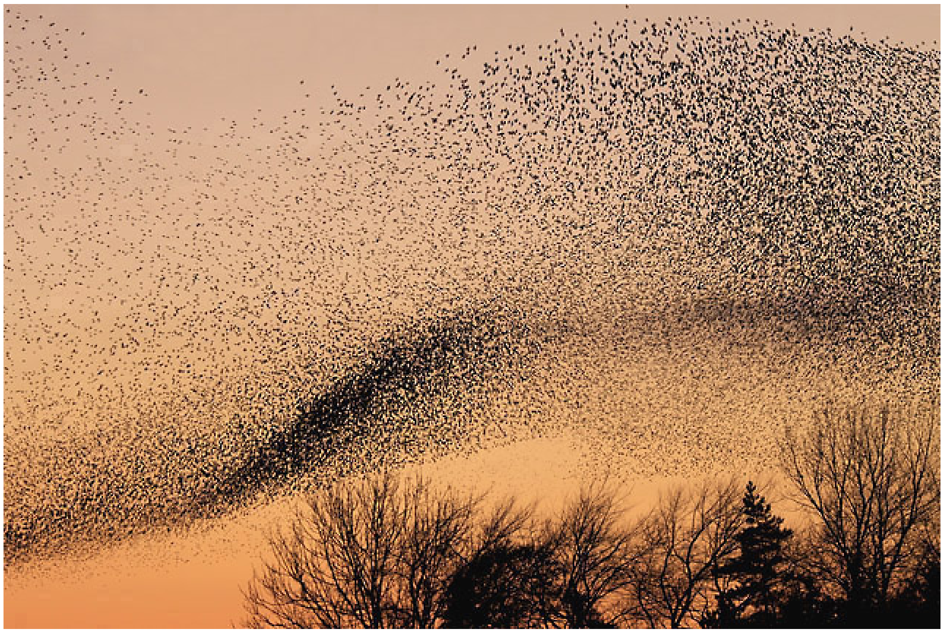


Figure 1.1 A murmuration of starlings at Gretna
Photo taken by Walter Baxter, at Gretna Green, Scotland on 7 November 2011. (© Copyright Walter Baxter under Creative Commons CC BY-SA 2.0) Image retrieved from WikiCommons.[1]No changes were made.
This image shows the natural flocking behavior of a group of starlings, a phenomenon which we attempt to simulate with our model.

Chapter 2

Our Model

As we approached the task of unmasking the mechanics behind flocking behavior, we began by considering what the previous models were, along with their utility. After a review of these models we created our own model based on the principles of consensus and frustration. Using these two elements, we were able to imitate emergent flocking phenomena.

2.1 Flocking Models

There are essentially three different approaches to modeling aggregation (flocking): the Eulerian method (fluid dynamics using partial differential equations), the Lagrangian method (equations of motion and detailed velocities for each particle), and the rule method (using discrete, individual behavior rules) [1]. While nature provides a plethora of varying animal aggregations, in this paper we often, though not exclusively, consider the aggregate flocking behavior of birds.

Bird flocking behavior was first modeled by C.W. Reynolds [7], where he also coined the term *boid* which is a shortened form of 'bird-oid' or a simulated bird. He created an algorithm based on three simple rules: separation, alignment, and cohesion. Separation is defined as steering the boids to avoid crowding local flockmates; alignment is steering the boids toward the average heading of local flockmates; and cohesion is steering the boids to move toward the average position (center of mass) of local flockmates. Reynolds also allowed for each boid to have a varying speed, but restricted that speed within the constraint of a maximum and minimum speed. However, he found that during simulations, because his model was based on a spatially nearest neighbors interaction, the boids would spontaneously group together and would tend toward a uniformly 'polarized' direction with a shared constant speed. In this paper, we refer to this uniform 'polarization' as a phase.

After Reynolds, one of the most significant contributors to modeling animal aggregations has been T. Vicsek. His novel approach to flocking formation is analogous to the ferromagnetic interactions and equilibrium spin domain alignment of the Ising model, except rather than aligning spins, Vicsek's model aligns the direction of motion (velocity) of boids [8]. For each time step in the simulation, each boid aligns its direction with the average direction of the nearest boids, with some random perturbation added, while maintaining a constant absolute velocity. The perturbations added to the velocity averaging is analogous to temperature, however, Vicsek admits that this analogy does break down because animal aggregations are quite different from magnetic domains [9].

Building off of Vicsek's model, F. Ginelli and H. Chaté developed a 'metric-free,' topological definition of neighbors[1] based on the observations of the StarFlag project [10]. Like Vicsek, the velocities of the boids are averaged with its neighbors (including a small noise factor) and then normalized. However, rather than defining neighbors based on a metric threshold of some finite length, Ginelli and Chaté define neighbors as those forming the first shell around boid j during the Voronoi tessellation construction[2] of all the boid positions at time t. Thus, their description of

[1] The terms 'metric free' and 'topological' refer to an interaction that is independent of spatial (or physical) distance.

[2] To visualize a Voronoi tessellation construction consider how plant cells fit together. The cells walls form rigid lines that share a boundary with the cells around it. Likewise, the boids in this algorithm define nearest neighbors with

topological refers to the notion that boids interactions are metric-free, or independent of distance (this is because there is no bound to the Voronoi shell size). Different from Reynolds and Vicsek, this results in a cohesive group with no subgroup splitting.

Another model type presented by H. Levine and W.-J. Rappel is based on a discrete Lagrangian method that considers both an attractive and repulsive force through a limited spatial range of interactions [11]. With this technique, Levine and Rappel were able to obtain a 'vortex state,' which is commonly found in schools of fish. Even so, while their model had the capacity to spontaneously create vortices, under certain parameter values these rotational phases would only appear if each boid was initialized with the same initial angular momentum.

Our model begins by randomly distributing the positions and (normalized) velocities of the boids and is two-dimensional for simplicity. Thus, rather than considering the flocking behavior that our model simulates to be literal, we consider the animations as a projection from three-dimensional space into two-dimensional space. By considering this projection, the algorithm is significantly simplified through the elimination of the $O(n^2)$ complexity needed to calculate the possibility of collision between any two boids. Therefore, any overlapping of boids that occurs in our simulations is considered to be the projection of their movement into a plane.

Our model differs from all the other previously proposed models particularly because other models use a reciprocal, nearest-neighbor interaction while our model uses a novel, democratic, non-reciprocal, topological, nearest-neighbor interaction. We call this topological interaction 'consensus' and from this interaction we obtain three unique phases: aligned, clockwise, and counter-clockwise. However, in order to obtain truly emergent flocking behavior, we add a frustration into the system. The interplay between these forces of consensus and frustration is akin to the constant biological battle to maintain a homeostasis state that is found in nature [5].

those in which they share a Voronoi cell wall. For a visualization of this consider Wikipedia's Voronoi diagram.

2.2 Consensus

The idea of consensus is that there is a ‘force’ that drives each individual boid to behave like all the other boids. In our case, consensus is created by averaging the velocities of a flockmate group, normalizing that average, and then assigning that to be the new velocity of the first boid of that group. This concept of consensus is very similar to Vicsek’s model, except instead of using spatial nearest neighbors like Vicsek, we use topological nearest neighbors. In our model, topological nearest neighbors are defined by randomly assigning each boid to a fixed index value in a list at the beginning of the simulation. Boids adjacent in the list form groups of topological nearest neighbors called flockmate groups. By dividing the flock into flockmate groups with boids that are topologically adjacent to each other, we can define consensus as the averaging of flockmate velocities through

$$\mathbf{v}_i'(t+1) = \frac{1}{n}\sum_{j=i}^{n+i} \mathbf{v}_j(t) \tag{2.1}$$

and,

$$\mathbf{v}_i(t+1) = \frac{\mathbf{v}_i'(t+1)}{\left\|\mathbf{v}_i'(t+1)\right\|} , \tag{2.2}$$

with the standard Euclidean norm (l^2-norm), where $\mathbf{v}_i(t)$ is the velocity of the i^{th} boid at time step t, and n is the number of boids in a flockmate group (excluding the first boid in the group which is the i^{th} boid).

This averaging is done for each boid starting from $i = 1$ to $i = N$ (the size of the flock) and the the first and last boids in the flock are considered topologically adjacent so that boid $b_{N+1} = b_1$. Most importantly, the nonlinearity found in maintaining a constant speed (through normalizing the velocity average) is essential to the emergent properties of the model. If the model were linear, then the result would literally be the sum of its parts, which would eliminate the possibility of emergent characteristics before we even got started.

This consensus interaction between boids is democratic (each boid possesses the same influential power), but it is not reciprocal. Consider a flock of N boids, $\{b_1, b_2, ..., b_N\}$, where b_i represents the i^{th} boid. Then consider dividing the flock into groups of flockmates of size n, say $n = 2$. This would partition the set into sets of three containing

$$\{\{b_1, b_2, b_3\}, \{b_2, b_3, b_4\}, ..., \{b_{N-1}, b_N, b_1\}, \{b_N, b_1, b_2\}\}.$$

Thus, each boid is contained in multiple flockmate groups, but each boid appears as the first entry of only one group. This can best be seen in Fig. 2.1 and Fig. 2.2.

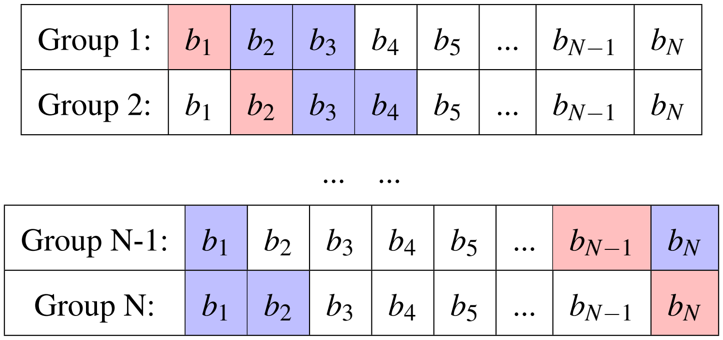
Fig. 2.1 shows how a flock of N boids partitioned into flockmate groups of $n = 2$ would be listed. It also describes how the nearest neighbors used in consensus are defined. Fig. 2.2 relates how nearest neighbor interactions based on spatial positions would be reciprocal by contrasting it with a topological interaction, using a flock of $N = 7$ boids and flockmate groups of $n = 2$. It shows how spatial nearest neighbors would have boid 1 averaging with boids 3 and 7 and boid 7 averaging with boids 1 and 5 [as seen in Fig. 2.2(a)], while topological nearest neighbors has boid 1 averaging with boids 2 and 3 and boid 7 averaging with boids 1 and 2 [as seen in Fig. 2.2(b)]. This makes the topological averaging non-reciprocal because boid 7 averages with boid 1, but boid 1 does not average with boid 7. In other words, this topological averaging has boid b_i averaging with boids $\{b_{i+1}, b_{i+2}, ..., b_{i+n}\}$, where the subscript index on the boid is calculated (mod N) to create a cyclical array.

Once the topological averaging is done and the newly averaged and normalized velocity is assigned to be the new velocity of the first boid in the flockmate group, the position of that boid is updated through Eq. (2.3), where $\mathbf{r}_i(t)$ is the position of the i^{th} boid at time t.

$$\mathbf{r}_i(t+1) = \mathbf{r}_i(t) + \mathbf{v}_i(t+1) \tag{2.3}$$

This completes the mathematical definition of consensus, providing the algorithmic foundation needed to drive a system of boids to behave uniformly.

With this definition of consensus we have enough framework to begin evaluating the time



Group 1:	b_1	b_2	b_3	b_4	b_5	...	b_{N-1}	b_N
Group 2:	b_1	b_2	b_3	b_4	b_5	...	b_{N-1}	b_N

... ...

Group N-1:	b_1	b_2	b_3	b_4	b_5	...	b_{N-1}	b_N
Group N:	b_1	b_2	b_3	b_4	b_5	...	b_{N-1}	b_N

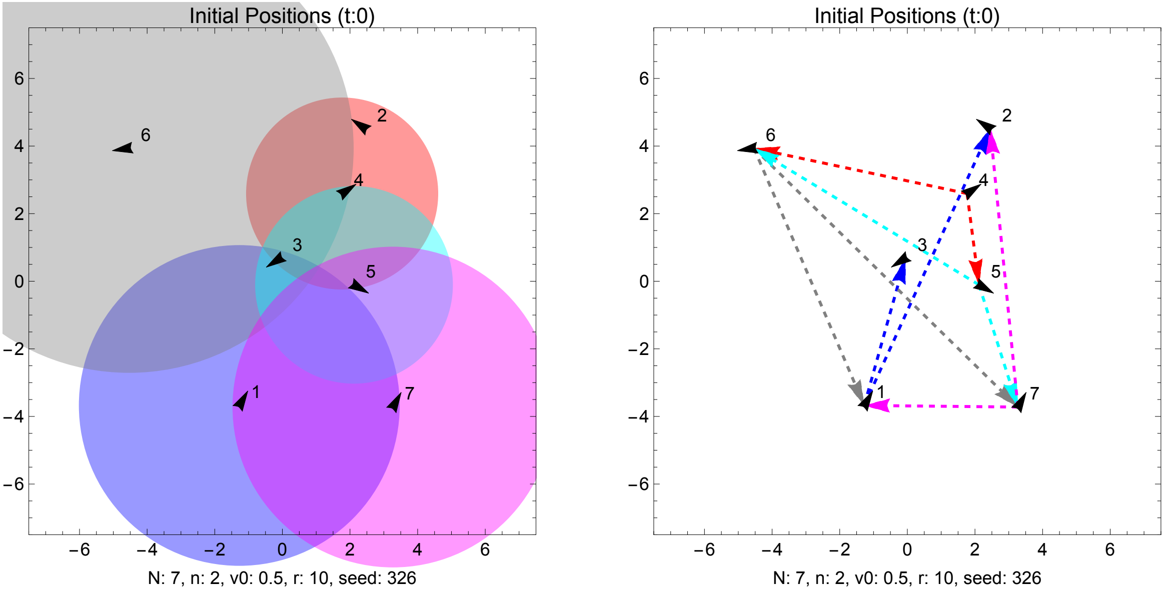
Figure 2.1 Nearest neighbors are defined topologically by initially assigning them to a fixed index value in a list. Thus, there is no correlation between these topological neighbors and the spatial distance between them. The first boid in the flockmate group (shown in red and defined as b_i) with its flockmate group (shown in blue and including the first boid in the group and defined as $\{b_{i+1}, b_{i+2}, ..., b_{i+n}\}$), where the subscript index on the boid is calculated (mod N) to create a cyclical array. This definition of nearest neighbors creates a boid-boid interaction that is democratic (each boid holds equal weight) but not reciprocal (b_i averages with b_{i+1}, but b_{i+1} does not average with b_i).

evolution of the system. However, with only a finite space to view the flock, we implement periodic boundary conditions (PBC) to simulate an unbounded space. With only this rule of consensus determining the motion of the boids, and without any other conditions, we obtain three distinct phases — aligned, clockwise, and counter-clockwise.

2.2.1 Phases

Phases arise from short-range interactions that establish long-range order. Our model is unique with its possession of three distinct phases (aligned, clockwise, and counter-clockwise motion) that emerge without the implementation of any boundary conditions (frustration). This is because of the topological averaging method that we employ, connecting all the boids in a democratic, cyclical, non-reciprocal way.

This interaction (topological consensus) allows for changes in motion to ‘ripple’ through the flock. Without any damping or dispersion (frustration), if enough of the boids all begin to move in a particular way, that motion propagates through the rest of the flock, returning to reinforce that



(a) Spatial Nearest Neighbors **(b)** Topological Nearest Neighbors

Figure 2.2 Initialized with random positions and velocities, these figures show spatial nearest neighbors (a) as well as topological nearest neighbors (b). Topological closeness is created by numbering (listing) all of the boids and assigning each a set of flockmates, see Fig. 2.1. The arrangement of the spatial positions is the same in both parts of the figure; however, a spatial nearest neighbors interaction (a) is a reciprocal interaction, while a topological nearest neighbors interaction (b) is non-reciprocal.

same behavior over and over again. Thus, if the flock tends to a certain direction, eventually the entire flock travels in that direction. Additionally, if the flock begins to travel in a circular motion, that motion spreads through the rest of the flock returning to strengthen the original behavior.

Variations of the initial conditions such as the initial random positions and velocities of the boids, as well as the number of flockmates compared to the size of the flock determine which phase the flock transitions to. The phase transitions only exist as the flock moves from the random initial conditions to the organized behavior that consensus brings. However, due to the limited number of interactions that join the flock together, through the consensus of flockmates, the phase transitions that are present in our model are not dramatic and sharp. Rather, there is often a meta-stable state which causes a slow transition from random motion to completely uniform, see Chapter 3 and Fig. B.3 in Appendix B.

The cyclic, non-reciprocal, democratic arrangement of the interactions between the boids allows for circular phases. When in a circular phase, all of the boids in the flock are phase locked.[3] Other models which rely on a spatial nearest neighbors interaction achieve only an aligned phase because individual deviance from the group is not reinforced. The results in Figures 2.3, 2.4, and 2.5 show flocks that appear to be displaying true emergent, flock-like behavior; however, please note how flock-like the behavior of each phase is.[4] While flocks will travel in straight lines, or circle around each other, these collective motions are only a part of the dynamic behavior that we are seeking to imitate.

[3] Phase locked means all of the boids are moving in their own circles at the same rate and the radii and periods of each boid are all identical. Thus, the relative position of each boid in their circle compared to another boid in its circle only differs by a phase, and this phase remains constant throughout time. Hence, their relative phases are locked, or unchanging.

[4] Animations of these three phase are hosted on YouTube. Videos demonstrating these three unique global phases can be found at: Aligned, Clockwise, and Counter-Clockwise. See Appendix B for a reference to the entire channel. Figures 2.3, 2.4, and 2.5 show one frame from each video.

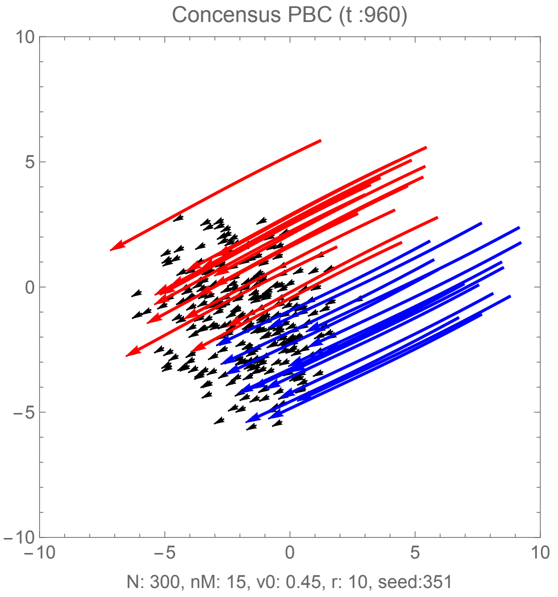


Figure 2.3 A still frame taken from a simulation of our flocking model implementing consensus with no boundary conditions or frustration. Here the flock members all spontaneously travel in exactly the same direction. The red and blue lines highlight two different flockmate groups and their trajectories.

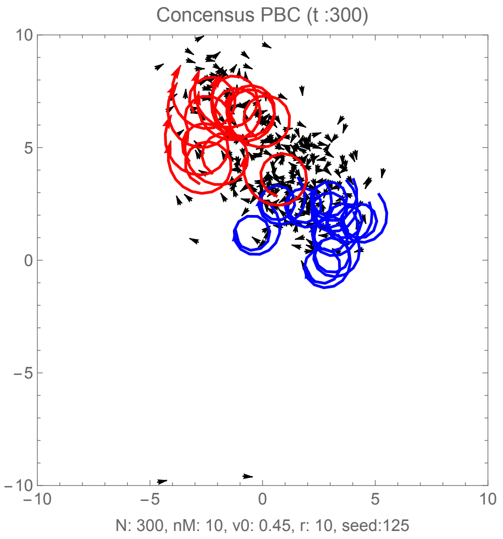


Figure 2.4 A still frame taken from a simulation of our flocking model implementing consensus with no boundary conditions or frustration. Here the flock members all spontaneously circle around clockwise. All of the boids in the flock are phase locked. The red and blue lines highlight two different flockmate groups and their trajectories.

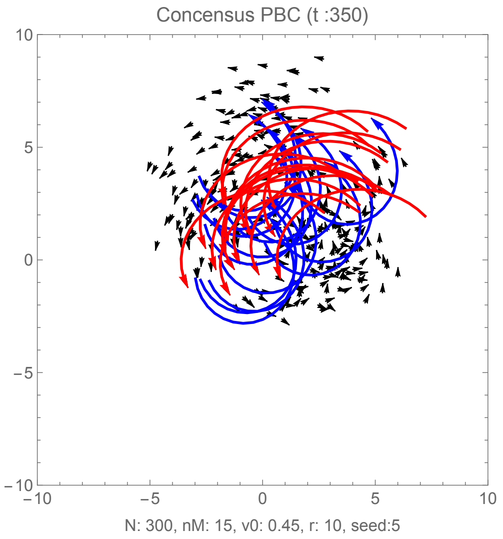


Figure 2.5 A still frame taken from a simulation of our flocking model implementing consensus with no boundary conditions or frustration. Here the flock members all spontaneously circle around counter-clockwise. All of the boids in the flock are phase locked. The red and blue lines highlight two different flockmate groups and their trajectories.

2.3 Frustration

To achieve a more dynamic behavior in our model, we employ the concept of frustration. This essentially breaks the symmetry of the group and damps out excessive consensus. Thus, this idea of frustration is essentially an antagonistic ‘force’ that opposes the consensus. We implement frustration in three different ways, each with increasing complexity but greater physical realism. We call these frustrations u-turns, specular reflections, and free boids.

2.3.1 U-Turn Reflection

Since our initial approach to the model was to place more focus on simplicity rather than accuracy, we began with a very basic frustration method that is implemented as individual boids making abrupt u-turns when they reached some certain boundary imposed upon them. As a natural transition from our periodic boundary conditions used to establish the model and underlying consensus, we simply remove the periodicity and implement a rigid, finite space, very much like a glass circular box. While mid-flight u-turns are not something common (or even physical) amongst birds, it does have some truth in schools of fish or other collective, swarm-like behaviors (like insects). However, in spite of the non-physical movement of each individual u-turn, collectively, the the motion of the group reveals a very complex and rich behavior that is beyond the simple phases that result from pure consensus.

Frustration to the consensus as a spatial u-turn after a certain distance away from the origin is based in the simplicity of the implementation. The implementation of the frustration as a u-turn is trivially calculated by reversing the velocity vector for each boid whose position is at some distance away from the origin. However, distance from the origin is not the only factor in determining when to implement a boundary condition; therefore, we also make a check to see whether or not the velocity of each boid was directed away from the origin. The distance check is made by

$$\mathbf{r}_i(t) \cdot \mathbf{r}_i(t) < L^2, \tag{2.4}$$

where $\mathbf{r}_i(t)$ is the position of the i^{th} boid at time t and L is the radius (length) of the space confining the flock. The use of a dot product to determine the distance of a boid from the origin rather than the standard Euclidean norm of the position vector increases the execution speed by avoiding the square root operation. Then, in a similar way, the directional check is made by

$$\mathbf{v}_i(t) \cdot \mathbf{r}_i(t) > 0, \tag{2.5}$$

where $\mathbf{v}_i(t)$ is the velocity of the i^{th} boid at time t. Then, if both of these conditions are true, the u-turn frustration is implemented by Eq. (2.6), and is shown in Fig. 2.6 (a),

$$\mathbf{v}_i(t) = -\mathbf{v}_i(t)\ . \tag{2.6}$$

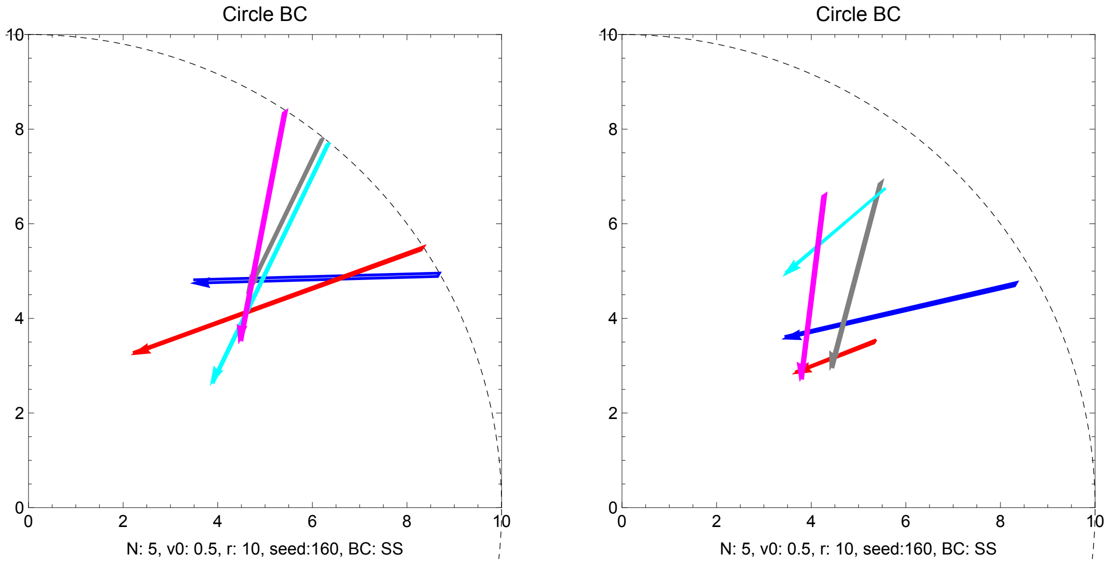
While this frustration method proves to be useful, the resulting animations come across as flies on the wall rather than birds in flight.[5] Additionally, there are often no hard and fast boundaries that a flock of birds would interact with and that if a flock were going to change direction the birds would turn more spontaneously and at different points in space. Thus, we create what we call a soft-basin boundary as seen in Fig. 2.6 (b).

To apply this soft boundary, or soft basin, to what we already have, we simply alter the distance from the origin condition, Eq. (2.4). Rather than checking if each boid were already at least beyond the radius of confinement, we compare a normalized distance, raised to some power p, with a random real number R chosen between zero and one. Thus, Eq. (2.4) becomes

$$\left(\frac{\mathbf{r}_i(t) \cdot \mathbf{r}_i(t)}{L^2}\right)^p > R\ . \tag{2.7}$$

A visualization of this soft basin boundary is shown in Fig. 2.7. By making this alteration we maintain the certainty that each boid is redirected by the time it reaches the previously rigid edge.

[5] A striking visualization of this can be found online as well as in Fig. 2.10 (a).



(a) Rigid U-Turn Implementation

(b) Soft U-Turn Implementation

Figure 2.6 Examples of the u-turn frustration with both rigid and soft boundary conditions. In spite of how unphysical a u-turn maneuver would be for a bird, this particular implementation of frustration is surprisingly robust. By adding a soft boundary the flock looks less like it is trapped in a glass box, and more like it is moving in an open, physical space.

Furthermore, through this random real number R, this soft basin possesses spontaneous turns and a propensity to chaos — bringing us closer to emergent flocking behavior.

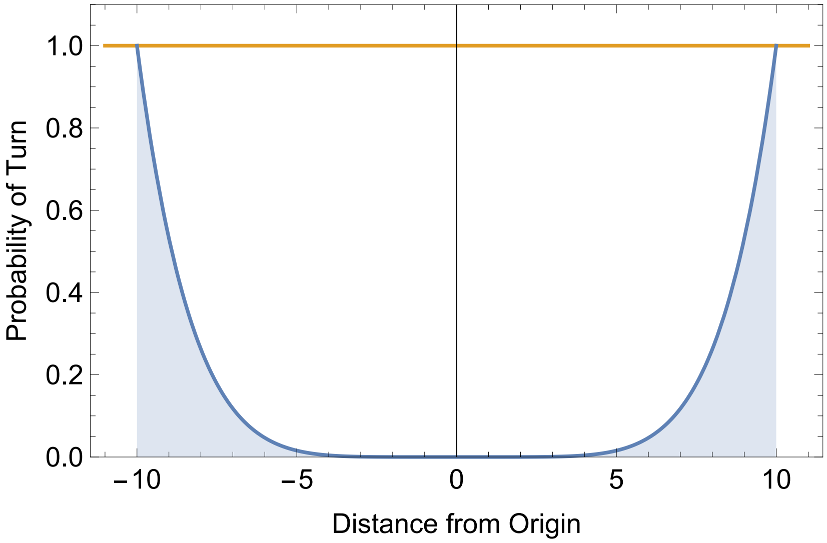


Figure 2.7 A representation of our implementation of a soft basin. The farther and farther the boids travel from the origin, the more likely it is that the frustration condition will be applied, see Eq. (2.7). The example shown here is for a spatial radius $L = 10$ and a power exponent of $p = 3$.

2.3.2 Specular Reflection

Since our implementation of the frustration as a u-turn is unphysical in many ways, we use a slightly more complex boundary condition approaching something more physical. Still wanting to keep the model as simple as possible, we move to applying specular reflections at the boundary. Therefore, in an almost identical way to implementing the rigid u-turn frustration, our implementation of the rigid specular reflection frustration is dominated by Eq. (2.4) and Eq. (2.5).

In spite of the differences between a u-turn and specular turn frustration, calculating a reflection that obeys the law of reflection requires a vector perpendicular to the boundary for each boid. We can obtain a sufficient expression for a vector normal to the boundary, $\mathbf{n}_i(t)$, if we normalize a boid's position vector $\mathbf{r}_i(t)$. Thus, we define the normal vector for each boid as

$$\mathbf{n}_i(t) = \frac{\mathbf{r}_i(t)}{\|\mathbf{r}_i(t)\|}\,. \tag{2.8}$$

From this we can change the direction of velocity by first finding the component of the boid's velocity which is parallel to $\mathbf{n}_i(t)$, called $\mathbf{v}_i^{\|}(t)$, and then subtracting twice that component off the original velocity. This results in Eq. (2.9) and Eq. (2.10) and is depicted in Fig. 2.8 (a).

$$\mathbf{v}_i^{\|}(t) = (\mathbf{v}_i(t) \cdot \mathbf{n}_i(t))\, \mathbf{n}_i(t) \tag{2.9}$$

$$\mathbf{v}_i(t) = \mathbf{v}_i(t) - 2\, \mathbf{v}_i^{\|}(t) \tag{2.10}$$

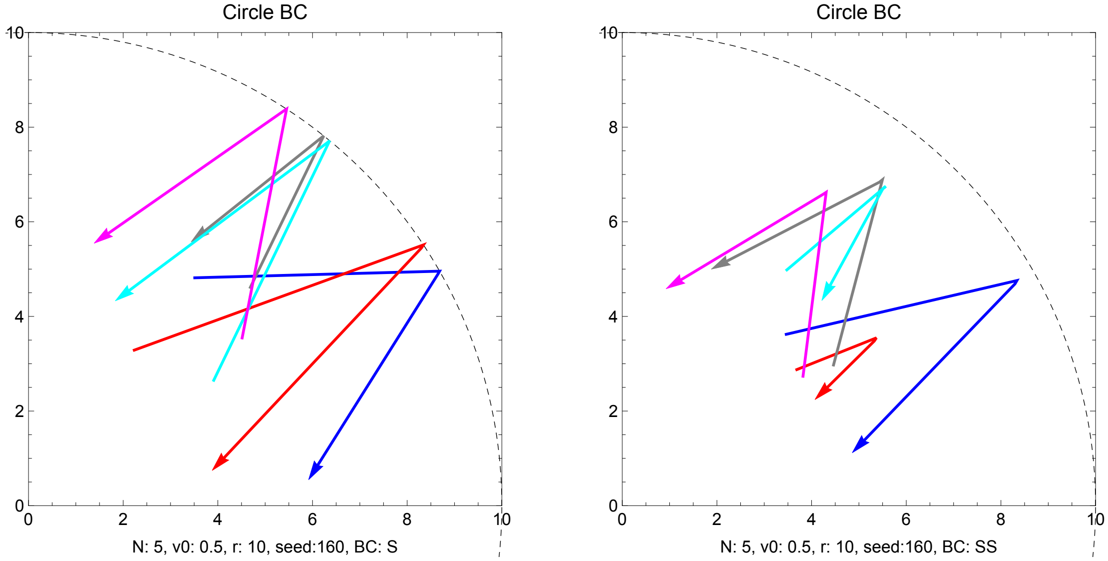
Once again, rigid physical barriers are not often encountered by flocks in nature, so we replace the condition found in Eq. (2.4) with Eq. (2.7) to create a soft specular boundary. A demonstration of the soft specular reflection implementation can be seen in Fig. 2.8 (b).

The alterations made from soft u-turns to soft specular reflections, while leading to more physical realism, do create a less robust model for emergent flocking behavior. Since the specular reflections do not frustrate the consensus as strongly as u-turns do, under certain conditions, the flock eventually enters into a global pattern – often conforming to the circular boundary that confines it.[6]

2.3.3 Free Boids

Our final implementation of frustration is certainly much more complicated than a simple u-turn or specular reflection at a boundary, but it possesses the physical realism we see in nature. We move to completely eliminate the concept of a boundary condition, which is why we called this implementation 'free boids'. Additionally, instead of imposing the frustration separately from the

[6] This conforming behavior can be seen online in Bubble to Ring.



(a) Rigid Specular Reflection

(b) Soft Specular Reflection

Figure 2.8 Examples of the specular reflection frustration with both rigid and soft boundary conditions. As expected, the implementation of specular reflections obey the law of reflection.

consensus averaging, we implement it as a perturbation during the consensus. This perturbation acts as a noise in the system. This also enables us to change the strength of the noise, allowing us to more delicately find a strength of frustration which results in a more robust emergence. Furthermore, based on observations [12], we set the number of flockmates to be, on average seven, ranging randomly between five and nine.

In order to implement the free boids frustration, we divide all the possible orientations of a boid into the four different cases that could be encountered. These four cases are then divided into two classes: when a boid's trajectory is away from the origin and when its trajectory is toward the origin. These two classes allow us to efficiently determine how to apply the frustration. To visualize this implementation refer to Fig. 2.9.

In order to determine which of the four cases each boid falls into during a given time step, we first define a relative quadrant system determined by the boid's position vector and a vector tangential to that position vector. The position vector $\mathbf{r}_i(t)$ is easily accessible and the perpendicular, tangential vector is calculated by

$$\mathbf{T}_i(t) = \begin{pmatrix} r_{iy}(t) \\ -r_{ix}(t) \end{pmatrix} . \tag{2.11}$$

With these two vectors, $\mathbf{r}_i(t)$ and $\mathbf{T}_i(t)$, a relative quadrant system can be defined for each boid at each time step and, with the quadrants defined, a check is made to find which relative quadrant the velocity is in. Once again, this involves the condition found in Eq. (2.5), but now also includes a similar check which we define as

$$\sigma = \mathrm{sgn}\left(\mathbf{v}_\mathrm{i}(\mathrm{t}) \cdot \mathbf{T}_\mathrm{i}(\mathrm{t})\right) . \tag{2.12}$$

Here, $\mathrm{sgn}(x)$ is the sign function which extracts the sign of the real number x. This is more formally

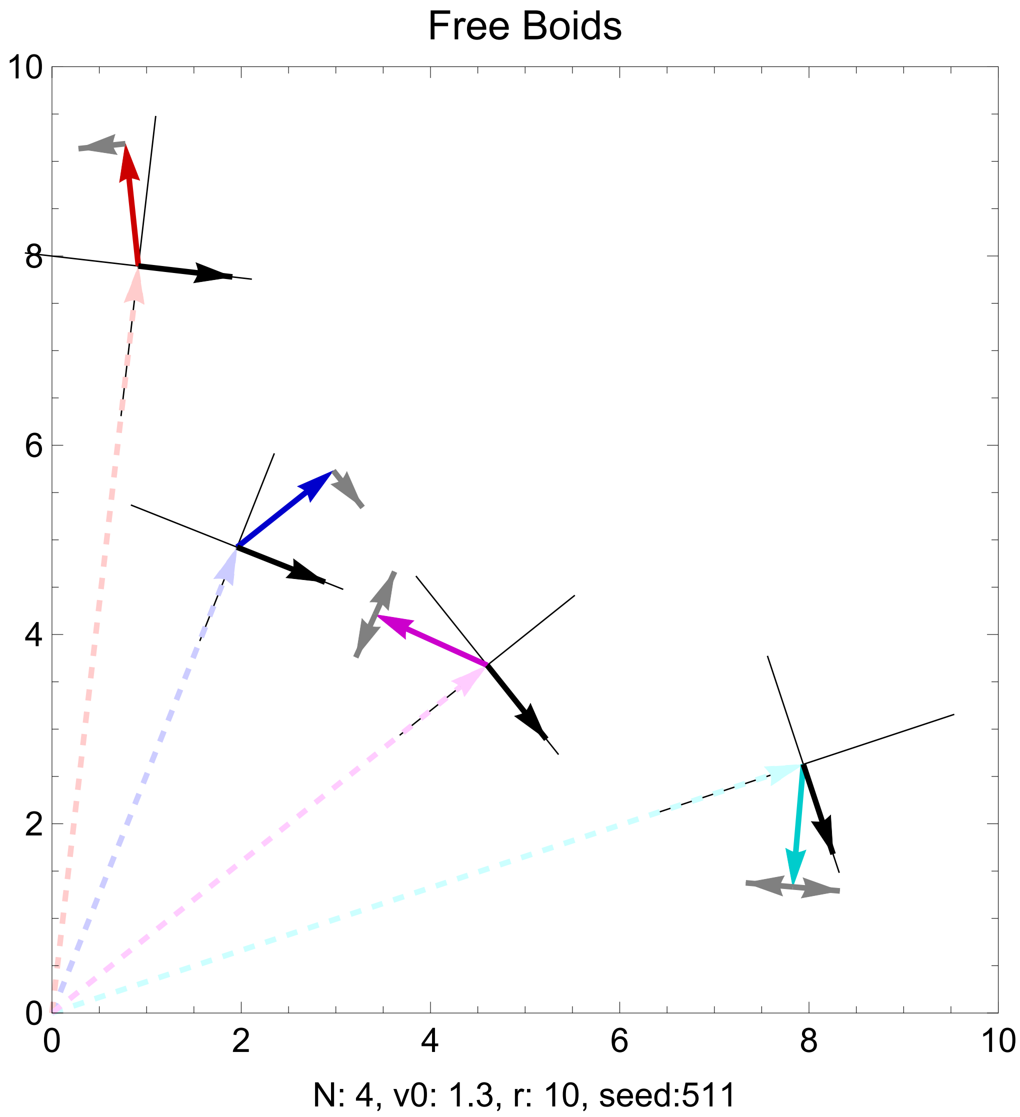


Figure 2.9 The four different cases that are encountered during the implementation of the free boids frustration. First, a relative quadrant system is determined by the position vector (shown as the dashed lightly colored arrows) and a vector tangential to the position vector (shown as the black arrows). Second, a check is made to find which relative quadrant the velocity (shown as the solid colored arrows) is in. Third, a perturbation is added to the velocity (shown as the grey arrows). Thus, if a boid's trajectory is away from the origin then it is perturbed back toward the origin, but if it is not going away then a random perturbation is added.

defined as

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases} \cdot \tag{2.13}$$

Using both of these conditions, the velocity can be reduced to one of the four quadrants. However, unlike the Eq. (2.5) condition, which is simply used to determine whether or not a frustration should be applied, σ is used in actually calculating the frustration, which is why the sign is needed but not the value.

The perturbation δ for each of the cases is calculated by considering the distance the boid is away from the origin d [Eq. (2.14)], a normally distributed random number η, and the sign of the direction of the velocity σ. Additionally, the overall strength of the perturbations can be uniformly increased or decreased by a multiplicative factor, ε. A simple change to Eq. (2.7) gives us the distance from the origin, which defines

$$d = \left(\frac{\mathbf{r}_i(t) \cdot \mathbf{r}_i(t)}{L^2} \right)^p . \tag{2.14}$$

After determining which quadrant the velocity is in, the free boid frustration is implemented by adding a perturbation to the velocity. If a boid's trajectory is away from the origin, then it is perturbed back toward the origin, but if it is not going away then a simple random perturbation is added. This perturbation is represented by δ:

$$\delta = \begin{cases} d\,\varepsilon\,\sigma\,|\eta| & \text{if } \mathbf{v}_\mathrm{i}(\mathrm{t}) \cdot \mathbf{r}_\mathrm{i}(\mathrm{t}) > 0 \\ d\,\varepsilon\,\eta & \text{if } \mathbf{v}_\mathrm{i}(\mathrm{t}) \cdot \mathbf{r}_\mathrm{i}(\mathrm{t}) < 0 \end{cases} . \tag{2.15}$$

However, because this is a perturbation to consensus, it is also necessary to alter Eq. (2.2) by adding an additional, scaled tangential component. Thus, we first redefine Eq. (2.1) to be

$$\bar{\mathbf{v}}_i'(t) = \frac{1}{n} \sum_{j=i}^{n+i} \mathbf{v}_j(t) . \tag{2.16}$$

Then a tangential velocity vector can be found just like Eq. (2.11) using

$$\bar{\mathbf{v}}_i^{\perp\prime}(t) = \begin{pmatrix} \bar{v}'_{iy}(t) \\ -\bar{v}'_{ix}(t) \end{pmatrix} . \tag{2.17}$$

And finally, a new definition for consensus is calculated by combining Eq. (2.16) and Eq. (2.17) into $\mathbf{v}'_i(t)$, then normalizing $\mathbf{v}'_i(t)$ to acheive Eq. (2.19).

$$\mathbf{v}'_i(t) = \begin{cases} \bar{\mathbf{v}}'_i(t) + \delta\,\bar{\mathbf{v}}_i^{\perp\prime}(t) & \text{if } \mathbf{v}_\mathrm{i}(\mathrm{t})\cdot\mathbf{r}_\mathrm{i}(\mathrm{t}) > 0 \\ \bar{\mathbf{v}}'_i(t) + \delta\,\bar{\mathbf{v}}_i^{\perp\prime}(t) & \text{if } \mathbf{v}_\mathrm{i}(\mathrm{t})\cdot\mathbf{r}_\mathrm{i}(\mathrm{t}) < 0 \end{cases} \tag{2.18}$$

$$\mathbf{v}_i(t+1) = \frac{\mathbf{v}'_i(t)}{\left\|\mathbf{v}'_i(t)\right\|} \tag{2.19}$$

This last result is the combination of determining which direction each boid is moving in and then perturbing that motion to confine the boid to a certain spatial area without enforcing either a rigid or soft boundary. In addition to removing the concept of a boundary, this free boid frustration is also extremely physical. This is due to the small changes in velocity during each time step (as seen in Fig. 2.13).

2.3.4 Beyond Phase Transitions

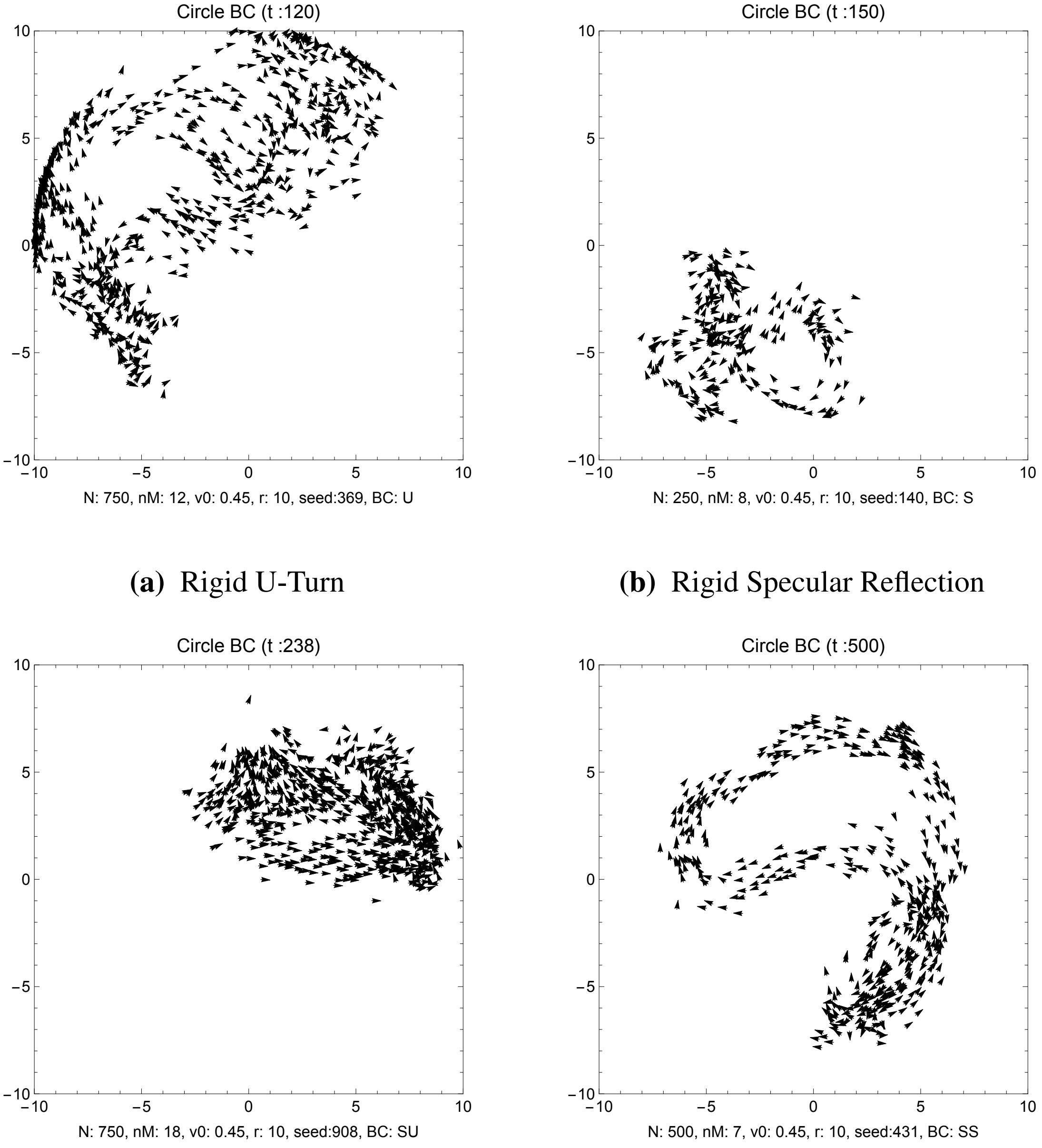
The dynamic interplay of phases and phase transitions is the heart of emergence in our model. Phases and phase transitions alone do not produce emergent flocking behavior. The behavior resulting from a single phase transition does not possess the dynamics that are present in the long term behavior of flocks. Thus, the continuous interplay between the three unique phases of our model create time dependent variations in flocking behavior. Without these shifts between phases, flocking phenomena would not emerge from the simple interactions of the boids.

The competing forces of consensus and frustration (regardless of how we implement the frustration) produce a system whose behavior is perhaps totally unexplainable by the simple rules and boid-boid interactions. During the course of a single simulation, a flock may stay tightly bound the entire time, or it may spread out to fill the entire space the entire time, or it may transition back and forth spontaneously between being tightly bound and filling the space.

It would be absurd to believe that still frames or screenshots could adequately capture the dynamic evolution of the animated simulations of our model,[7] even so, the figures at the end of this chapter contain just a small sampling of the flocking variety that our model produces. In Fig. 2.10, samples of the rigid and soft u-turn and specular frustrations are provided. These are followed by Fig. 2.11 which gives a sampling of the free boid frustration. The following two sets of figures, Fig. 2.12 and Fig. 2.13, show successive screenshots of particular simulations while highlighting the behavior of particular flockmate groups for easier analysis.

It is most important to note Fig. 2.13 and its depiction of how emergent behavior is the dynamic interplay of the three unique phases that arise from pure consensus. Compared to Fig. 2.3, Fig. 2.4, and Fig. 2.5, it becomes clear that both the shape and evolution of the frustrated flock are complex and active. Thus, phase transitions are the essential building blocks of emergent flocking behavior, however, they are insufficient in and of themselves to produce a truly emergent flocking phenomenon.

[7] As such, we have a YouTube channel devoted to animations produced by our model. See Appendix B for more information.



(a) Rigid U-Turn

(b) Rigid Specular Reflection

(c) Soft U-Turn

(d) Soft Specular Reflection

Figure 2.10 Each of these figures are samples of the different types of frustrations that we use in the implementation of our model. In the rigid examples, there is a clear and distinct hard boundary that causes the boids to appear like flies on a wall. In contrast, the soft boundaries show more physical behavior for a flock of birds. Additionally, this is a demonstration of a variety of flocking formations that can be achieved using our model.

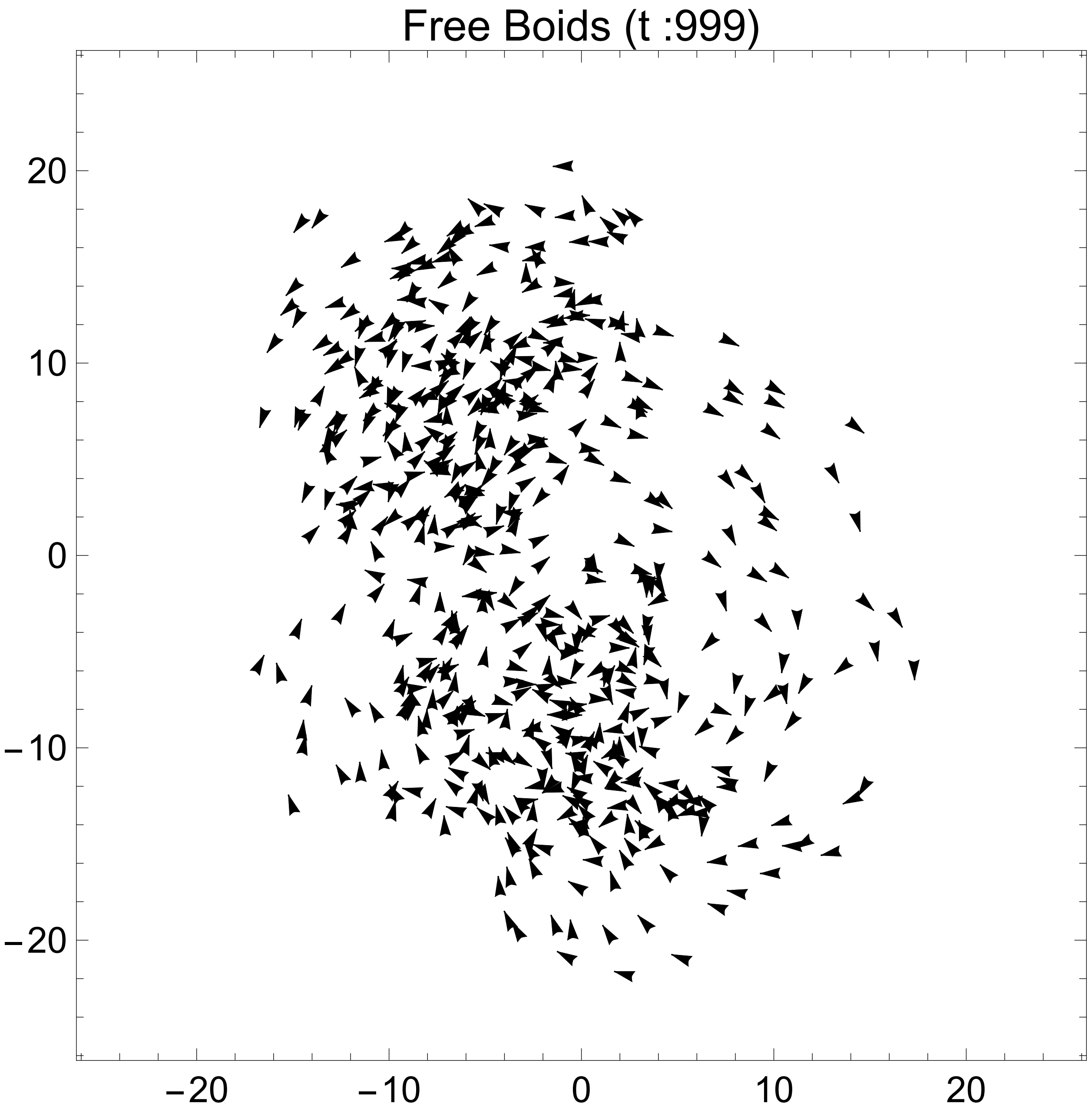


Figure 2.11 A snapshot in time demonstrating the emergent behavior that arises in the algorithmic evolution of our model using our most physical implementation of frustration, free boids.

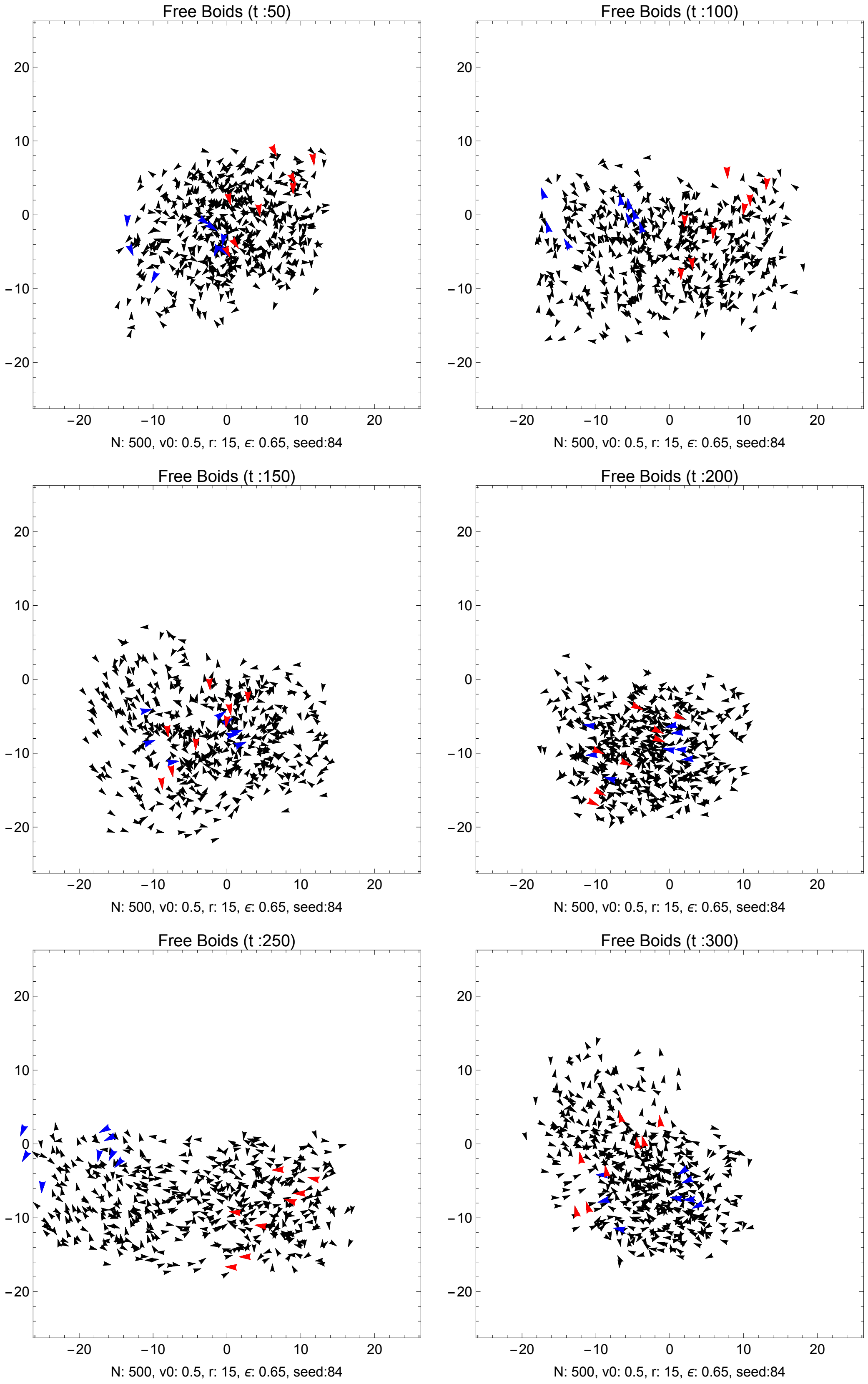


Figure 2.12 Depicted here are stills of a simulation created with our model. The red and blue boids are flockmate groups colored to highlight how the flock might appear during the animation. The sequence is viewed left to right, top to bottom.

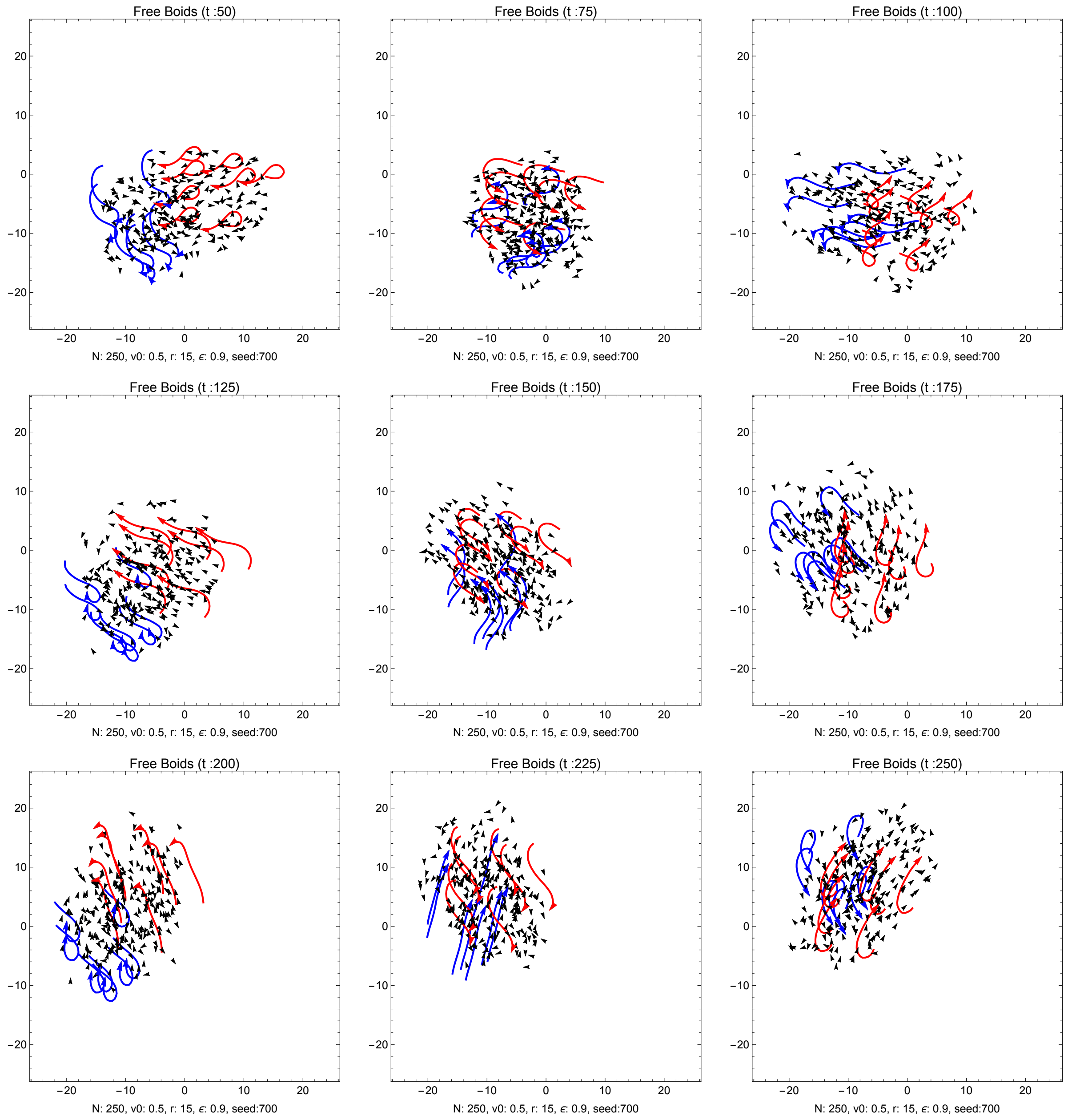


Figure 2.13 Comparing these still images to the images describing the three distinct phase transitions (Fig. 2.3, Fig. 2.4, and Fig. 2.5), it is clear that the motion of the boids (traced out as the red and blue trails) possess a combination of all three phases and that the dynamic interplay between each of the phases produces the emergent behavior that resembles a flock of birds. The sequence is viewed left to right, top to bottom.

Chapter 3

Classification

Since emergence has not been defined by a single metric (or perhaps cannot be), we present two order parameters that we used to empirically decide whether or not a particular animation is emergent. Each of these order parameters assist in recognizing phases and phase transitions, while also highlighting patterns or abrupt changes that occur during a simulation. A directional correlation function and a measure of the average distance between flockmates is also presented.

3.1 Order Parameters

3.1.1 Aligned

The first phase (aligned) can be defined by a directional alignment of the flock. This phase is common in all flocking models regardless of implementation and is therefore, significant to us as well. The aligned parameter $\langle v(t) \rangle$, defined in Eq. (3.1), is the absolute value of a simple normalized average of the velocities of the entire flock at a particular time t. N is the number of boids in the flock, $\mathbf{v}_i(t)$ is the velocity of the i^{th} boid at time t, and v_0 is the constant speed of each boid.

$$\langle v(t) \rangle = \frac{1}{N v_0} \left| \sum_{i=1}^{N} \mathbf{v}_i(t) \right| \tag{3.1}$$

Similarly, we define a group alignment order parameter that provides a way for us to measure how aligned a particular flockmate group is. This allows us to glean insight into the connections between different flockmate group alignments and provides more detailed information regarding a flock's emergence. The group alignment order parameter we define, Eq. (3.2), is nearly exactly the same as Eq. (3.1), except rather than summing up to N, the total number of boids in the flock, we only sum from $j = i$ to $n + i$ over j, where n is the total number of flockmates in the i^{th} flockmate group.

$$\langle v_i(t) \rangle = \frac{1}{n v_0} \left| \sum_{j=i}^{n+i} \mathbf{v}_j(t) \right| \tag{3.2}$$

Fig. 3.1 shows the alignment of the entire flock (in black) and the alignment of two particular flockmate groups (in red and blue). During this typical simulation the entire flock does not achieve complete alignment because if it did, it would indicate that the entire flock had transitioned into a flock-wide aligned phase. For more information and figures regarding global phases and phase transitions see Chapter 2.2.1 and Fig. B.3 in Appendix B respectively. Furthermore, the two flockmate groups that were monitored show very consistent group alignment throughout the simulation. This is what is expected considering that consensus is averaging their velocities at every time step.

Using the flock alignment parameter and the group alignment parameter, $\langle v(t) \rangle = 1$ implies that the flock is completely aligned, while $\langle v(t) \rangle = 0$ implies that the flock is either rotating or moving randomly (see Fig. 3.1). Thus, in order to distinguish between a rotating flock and a randomly moving flock, we create a rotational order parameter.

3.1.2 Rotational

Unlike other rule based boid models, ours possesses three distinct phases. Two phases unique to our model are the clockwise and counter-clockwise rotational phases. In order to measure when the

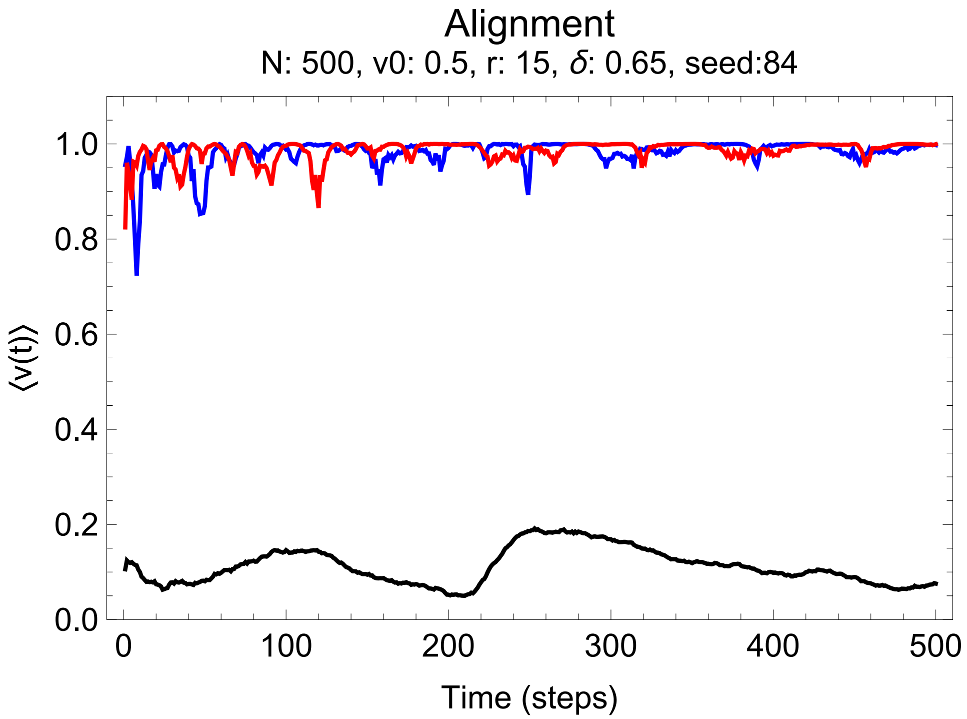


Figure 3.1 The alignment of a particular flock as it evolves in time. The black curve represents the alignment of the entire flock, Eq. (3.1), while the red and blue curves represent the alignment of two particular groups of flockmates Eq. (3.2).

flock or a particular flockmate group is rotating, we define an order parameter that essentially measures the curvature of a particular boid over a given time interval. This rotational order parameter is defined as

$$\langle L(t)\rangle = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\tau}\sum_{t=1}^{\tau}\frac{\mathbf{v}_i(t)\wedge\mathbf{v}_i(t+1)}{v_0^2}\,, \tag{3.3}$$

where τ is a given number of time steps needed to calculate the curvature and where the wedge product is essentially a two-dimensional cross product defined as:

$$\mathbf{v}_i(t)\wedge\mathbf{v}_i(t+1) \equiv v_{ix}(t)v_{iy}(t+1) - v_{ix}(t+1)v_{iy}(t) \tag{3.4}$$

for each boid. Essentially, the more perpendicular the velocity at time t is from the velocity at time $t+1$, the greater the value of the wedge product for each boid. Additionally, the sign of the wedge product indicates whether the boid is turning clockwise or counter-clockwise (see Fig. 3.2). If $\langle L(t)\rangle < 0$, then the flock is rotating clockwise, if $\langle L(t)\rangle > 0$, then the flock is rotating counter-clockwise, and if $\langle L(t)\rangle = 0$, the flock is either entirely aligned, or completely random. Moreover, just like we can monitor the alignment of an individual flockmate group, we can also monitor the rotation of an individual group with

$$\langle L_i(t)\rangle = \frac{1}{n}\sum_{j=i}^{n+i}\frac{1}{\tau}\sum_{t=1}^{\tau}\frac{\mathbf{v}_j(t)\wedge\mathbf{v}_j(t+1)}{v_0^2}\ . \tag{3.5}$$

Fig. 3.2 shows the rotation of the entire flock (in black) and the rotation of two particular flockmate groups (in red and blue). During this typical simulation, the entire flock tends toward a particular value because there exists a general rotation throughout the flock. However, note how the two flockmate groups that were monitored alternate between clockwise and counter-clockwise rotations throughout the simulation. This is what is expected during emergent flocking behavior, as parts of the flock alternate between one of the three different phases. If the flockmate groups began converging on the same rotational value it would be an indication of a global rotational phase transition. For more information and figures regarding global phase transitions see Chapter 2.2.1 and Fig. B.4 in Appendix B respectively.

3.2 Directional Correlation Function

Observations of starling murmurations, specifically European starlings (*Sturnus vulgaris*), made by Andrea Cavagna and his team in Italy, reveal a correlation between the change of direction of the birds on the edges of the flock followed by changes in the overall direction of the flock [13]. Prior to the flock changing direction, individuals deviate from the group, causing a ripple effect throughout the group.

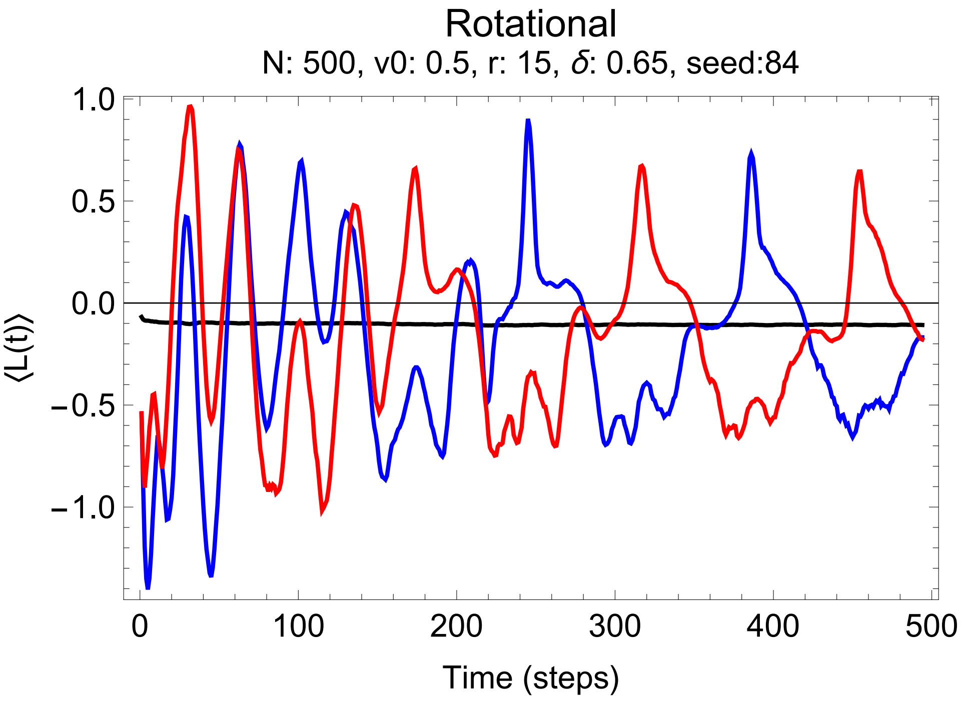


Figure 3.2 The rotation of a particular flock as it evolves in time. The black curve represents the rotation of the entire flock (in a general clockwise direction), Eq. (3.3), while the red and blue curves represent the rotation of two particular groups of flockmates, Eq. (3.5).

The observations and statistical analysis done by Cavagna and his group share numerous similarities with the flocking model that we present. As such, we too define a directional (velocity) correlation function as

$$C_{i,n}(t) = \frac{\mathbf{v}_i(t)}{v_i(t)} \cdot \frac{\mathbf{V}_n(t)}{V_n(t)} \,, \tag{3.6}$$

where $\mathbf{V}_n(t) = \sum_{j=i}^{2n+i} \mathbf{v}_j(t)$ which is the average velocity of a sub-group of the flock originating with the i^{th} boid. Eq. (3.6) allows us to understand how variations in direction of an individual boid correlate with a large portion of its topological peers. As mentioned in Chapter 2.2, updates to each boid in the flock are done sequentially from boid $i = 1$ to $i = N$. Thus, this correlation function, Eq. (3.6), will show that if one boid makes a sharp turn, then that motion will propagate down the topological chain of flockmates, eventually returning to the original boid.

As is shown in Fig. 3.3, the individual deviation of a particular boid from the group propagates through the flock. The red boid is topologically as far away from the blue boid as possible but the motion and movement of the red boid is imitated by the blue boid. Of course, it is unclear whether or not it was some other boid that first moved in a unique direction, but the propagation of that motion is seen in the repeating shapes that alternate between red and blue as the time steps increase. However, it is important to note that as the motion propagates through the flock, it is slowly ‘damped’ out by the frustration. It is also significant that the boid is often never completely aligned with the large portion of its topological peers. If it were, then we would expect to see a global aligned phase from Eq. (3.1). Once again, the red boid is *not* being correlated with the blue boid, rather Fig. 3.3 overlays the correlation of the red boid [as calculated with Eq. (3.6)] and the correlation of the blue boid [as calculated with Eq. (3.6)] in the same plot.

3.3 Average Distance Between Flockmates

By calculating the average spatial distance between a given boid and its flockmates, we are able to determine how the dynamics of the flock changes over time. We expect to see the boids that follow each other to congregate more closely together over time because the frustration will turn the individual boids more sharply at different times than others, resulting in a general coming together. However, if the average distance between the flockmates becomes very small, then it is most likely that the frustration is too strong and is overpowering the consensus. In order to calculate this separation correlation $\langle R_i(t) \rangle$, we simply compute the distance from the i^{th} boid to all of its flockmates and then take the average of those distances.

$$\langle R_i(t) \rangle = \frac{1}{n} \sum_{j=i}^{n+i} \left\| \mathbf{r}_i(t) - \mathbf{r}_j(t) \right\| \tag{3.7}$$

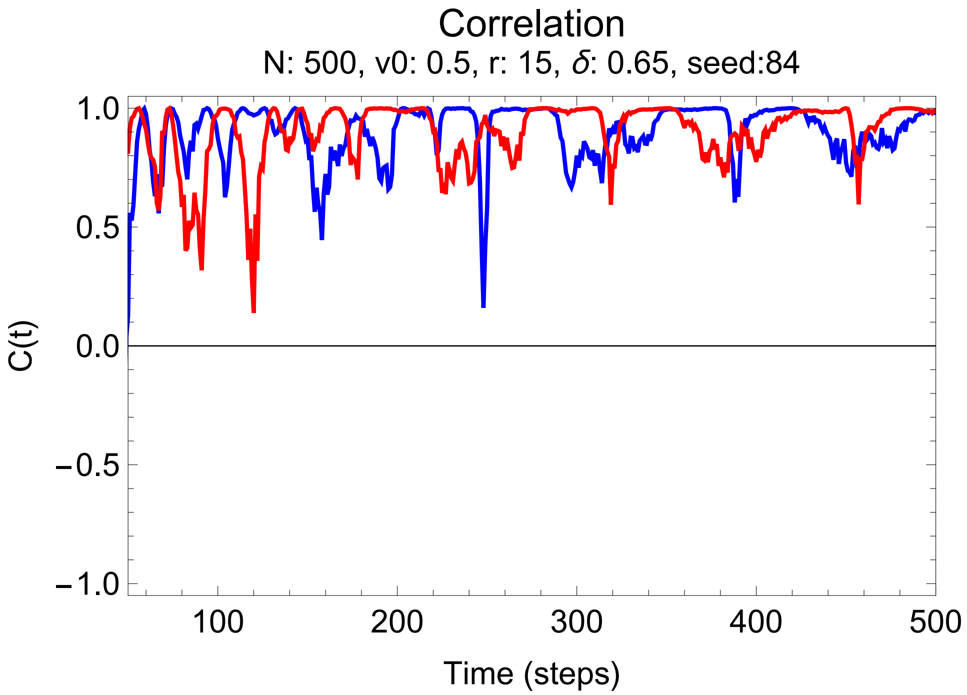
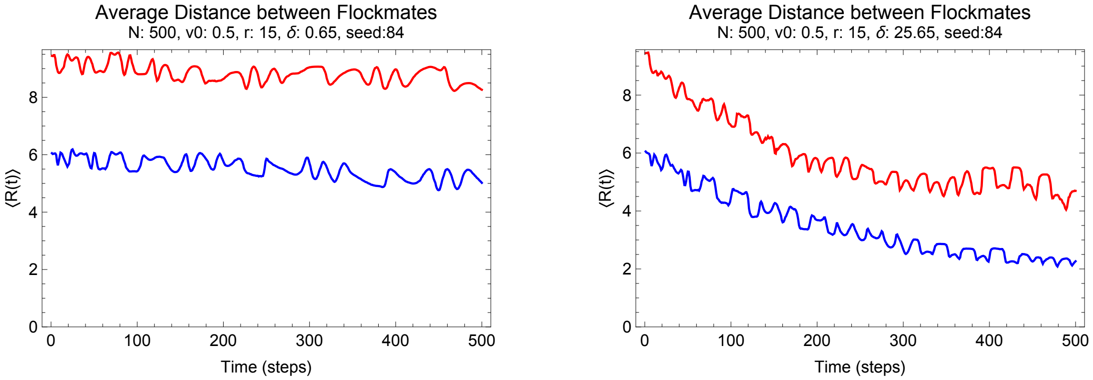


Figure 3.3 The directional correlation of two boids (one shown in red, the other in blue) with their successive topological neighbors (respectively). The individual deviation of these two boids from their groups propagates through the flock. The red boid is topologically as far away from the blue boid as possible but the motion and movement of the red boid is imitated by the blue boid. It is important to note that as the motion propagates through the flock, it is slowly 'damped' out by the frustration.

Here $\mathbf{r}_i(t)$ is the position of the i^{th} boid at time t, n is the number of flockmates, and the Euclidean norm (l^2-norm) is used.

The comparison made in Fig. 3.4 shows the difference between a normal free boid simulation that results in emergent flocking behavior [Fig. 3.4(a)] and a simulation where the free boid frustration is extremely strong [Fig. 3.4(b)]. The stronger the frustration, the closer together the boids in a flockmate group become, while a normal frustration reveals only a small tendency to concentrate together. Thus, the density of the flock is determined by the strength of the frustration; however, too much frustration restricts the unique behaviors of the flock and results in a concentrated, random motion.



(a) Normal Frustration, $\delta \approx 1$

(b) Strong Frustration, $\delta \approx 25$

Figure 3.4 The average distance between flockmates as defined by Eq. (3.7). As it is shown, the stronger the frustration the closer together the boids in a flockmate group become. A typical simulation will see some drifting together, (a), but nothing like a simulation with a strong frustration, (b). The vertical axis is measured in unitless spatial distance. The red line represents the average distance between flockmates for one particular boid, while the blue line represents the average distance between flockmates for another particular boid. There is no overlap in flockmates between the red and blue boids.

Chapter 4

Results

Discrete parameter spaces are created for both the u-turn and specular frustrations based on the quantitative classifications described in the previous chapter. The parameter space is defined by relating the size of the flock to the number of flockmates as well as the radius of the simulation space to the constant velocity of each boid. Emergence only appears under certain conditions where neither consensus nor frustration overpower each other.

4.1 U-Turn Frustration

The implementation and analysis of the u-turn frustration was developed and executed by M. Berrondo and W. Kruger [14]. With pure consensus, the resulting phases that arise are interesting in their own right, but the simple extension of adding a u-turn frustration provides the versatility that pure consensus lacks. By removing the periodic boundaries and enclosing the flock with a reflective boundary, the symmetry of the phases are broken, allowing for more interesting characteristics while still maintaining its deterministic feature. However, as mentioned before, the rigid u-turns provide extremely unphysical flight paths for avians; thus, a soft-basin of attraction was added to increase the realism.

The parameter space of emergence for u-turn frustration was explored by M. Berrondo and M. Sandoval (Fig. 4.1) [15]. It was determined by the total number of boids in the flock, N; the number of flockmates in each sub-group, n; the radius of the circle that comprises the positional space in which the boids can move, L; and finally, the speed of each boid, v_0. The parameter space was sampled with $N = \{50, 250, 500, 1000\}$, $n = \{5, 10, 20\}$, $v_0 = \{0.1, 0.15, 0.2, 0.3\}$, and $L = \{10, 40, 100\}$.

Using these sets, we plot a discrete parameter space of $\{n/N, L/v_0\}$ as seen in Fig. 4.1. The use of a greyscale gradient underlay is not to suggest a continuous parameter space, but rather to aid in the readability of the data. Each point on the plot corresponds to a percentage of the flock which is flockmates (n/N) plotted against the effective positional space (L/v_0). It is an effective positional space because a multiplicative scalar factor to the velocity is equivalent to its inverse on the radius.

The emergent parameter space with a u-turn frustration is laid out in Fig. 4.1. The black circles represent the combination of parameters that resulted in a flock demonstrating emergent behavior, with the white squares representing the parameters where a flock did not show emergence. However, since Fig. 4.1 is a representation of a four-dimensional parameter space $\{N, n, L, v_0\}$ in two-dimensions $\{n/N, L/v_0\}$, there is inevitable loss of information as it is projected from four-space to two-space. Thus, the grey diamonds represent the places where two different combinations of four-space parameters (specifically one emergent flock, or black circle, and one non-emergent flock, or white square) are mapped to the same combination of two-space parameters, n/N and L/v_0. In these degenerate cases, we plotted the underlying gradient as emergent. While the representation of the parameter space as ratios of n/N and L/v_0 removed some of the clarity in the data, it more intuitively represents the parameters that result in emergent characteristics.

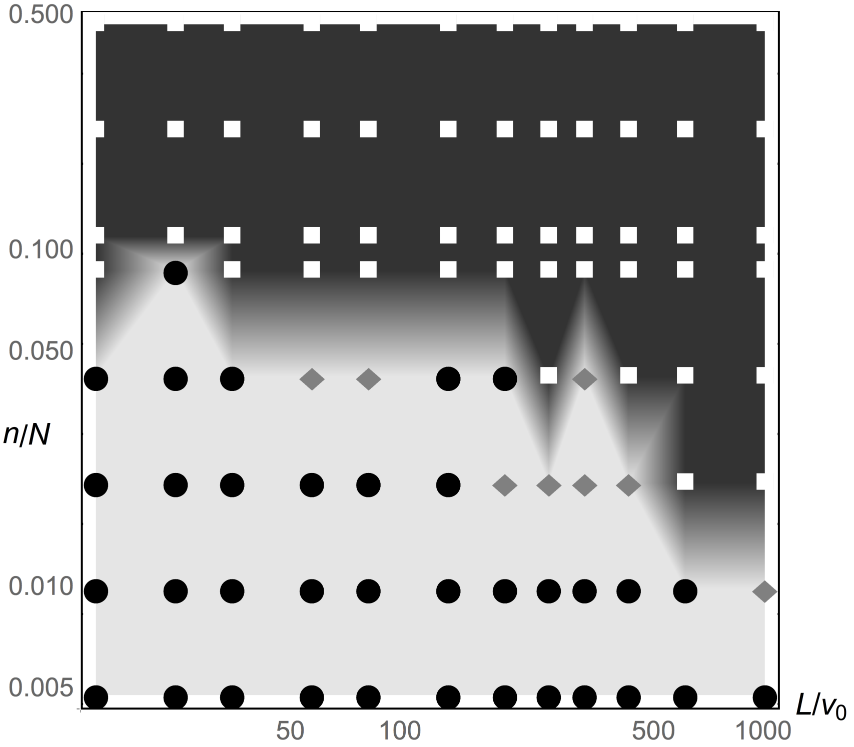


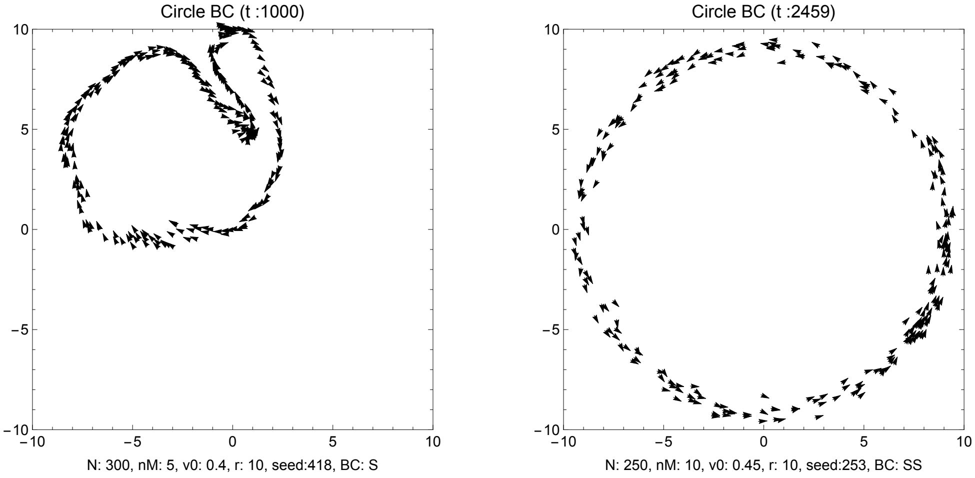
Figure 4.1 Parameter space for u-turn emergence with a soft-basin circular boundary (LogLog scale). The ratio of the flockmates to the flock size (n/N) is plotted against the effective positional space (L/v_0). Simulations ran for 1000 time steps. The black circles represent the combination of parameters which resulted in a flock that demonstrated emergent behavior during the entire simulation. The white squares represent the parameters that resulted in a flock that did not show emergence for any of the simulation. And the grey diamonds represent the places where the combination of n/N and L/v_0 resulted in either an emergent or non-emergent, flock due to the degeneracy in the parameter space. In these cases, we plotted the underlying gradient as emergent. These results originated in a paper published by M. Berrondo and M. Sandoval [15].

4.2 Specular Frustration

Having established that u-turns are an effective rule that produce emergence (given the right set of parameters), we also found an effective emergent parameter space for specular reflections. Our previous attempts with only a rigid boundary suggest that specular reflections are not a strong enough implementation of the frustration to result in consistent emergence. However, we found that given a soft boundary and the right set of parameters, specular reflections produce stable, emergent flocking. The conditions for emergence with specular reflections are more sensitive to the initial conditions than u-turn frustration; however, emergence is consistently strongest when the random initial positions and directions are concentrated to a centered unit square rather than spread uniformly throughout the field. We do note that a concentrated flock is topologically equivalent to a uniform distribution in the absence of frustration, and that both are characterized by identical order parameters. Thus, concentrating the initial positions of the flock is not a serious deviation from our previous approach. However, in the presence of a soft frustration basin, these conditions are different enough that we were able to achieve consistent and dynamic emergence using specular reflections as frustration.

We found that the emergence is not consistent with the simpler, rigid boundary conditions, but that it is consistent with the soft basin boundaries. The soft basin is implemented by a probability function which establishes a 100% chance of reflection at a given radius (L) away from the origin [see Eq. (2.7) and Fig. 2.7]. This creates the possibility for any individual boid to undergo a specular reflection at any given time step, essentially creating an unpredictable evolution in the flocking behavior. Nevertheless, unpredictability is not lack of determinism. By selecting a specific random seed, we are able to identically reproduce the results of a given set of parameters. Still, this unpredictable nature does increase the number of boid-boundary interactions that the flock experiences over the course of the simulation. The probabilistic element of the soft basin reduces the rigid influence that a firm boundary imposes on the flock, further reducing the conforming effect of the boundary on the flock (see Fig. 4.2). Thus, the soft boundary allows for more dynamic emergence.

To further reduce the confining effects of the artificially imposed boundaries, we concentrate



(a) Rigid Specular Reflections **(b)** Soft Specular Reflections

Figure 4.2 Snapshots of animations showing how the flock conforms to the artificially imposed boundaries during simulations with specular reflection as a frustration.[1]

the flock's initial positions. By doing this, the motion of the individual boids is less rigidly influenced by the frustration. This enables the expression of dynamic qualities, rather than the conformed motion of the shape of the basin. This reduced conformity occurs in spite of the fact that the boids interact more frequently with a soft basin. We found that if the flock were confined by a rigid boundary condition, the topological averaging, combined with the weaker, specular reflections, results in the flock conforming to the perimeter of the boundary (as seen in Fig. 4.2). Even so, with the soft boundary we find that under half of the conditions (parameter space) tested, given sufficient time, the flock would circle around the perimeter in a ring-like pattern even for flocks which showed dynamic emergence for nearly the entire simulation.[2]

The explored parameter space for the specular reflection frustration is determined by $N =$

[1] This conforming behavior can be seen online in Bubble to Ring.

[2] We ran all of the simulations for 10 000 time steps. While half of the flocks exhibited emergent behavior for the first 1000 time steps as shown in Fig. 4.3, over 90% of the flocks would conform to the boundary as seen in Fig. 4.2 (b).

$\{50, 250, 500, 1000\}$, $n = \{5, 10, 20\}$, $v_0 = \{0.1, 0.25, 0.45, 0.65\}$, and $L = \{10, 40, 100\}$ as seen in Fig. 4.3. Where N, n, v_0, L, the greyscale underlay, black circles, white squares, grey diamonds represent the same characteristics described previously.

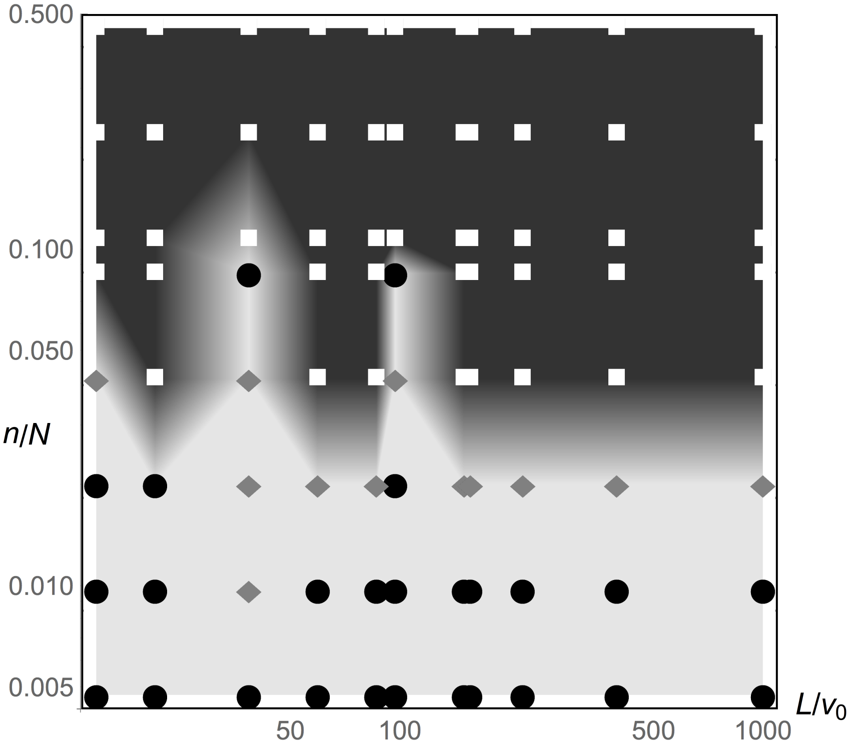


Figure 4.3 Parameter space for specular emergence with a soft-basin circular boundary (LogLog scale). The ratio of the flockmates to the flock size (n/N) is plotted against the effective positional space (L/v_0). Simulations ran for 1000 time steps. The black circles represent the combination of parameters which resulted in a flock that demonstrated emergent behavior during the entire simulation; the white squares represent the parameters that resulted in a flock that did not show emergence for any of the simulation; and the grey diamonds represent the places where the combination of n/N and L/v_0 resulted in either an emergent or non-emergent flock due to the degeneracy in the parameter space. In these cases we plotted the underlying gradient as emergent.

The observations that we can make from Fig. 4.3 are useful in helping to characterize emergence. Note how n/N is the strongest factor contributing to emergence, even more than L/v_0. This indicates that, as we found in the u-turn case (shown in Fig. 4.1), when consensus is too strong relative to the amount of frustration there is no emergence. The strong consensus is manifest as a large n relative to N. As shown in Fig. 4.3, if the groups of topological neighbors are larger than 6% or 8% of the total size of the flock, then there is no emergence. Additionally, the likelihood of the flock demonstrating emergence decreases as the effective positional space increases (the combination of a small velocity and a large radius increases the effective positional space).

The comparison between the different implementations of frustration as specular reflections and u-turn reflection in a soft-basin environment show that emergence from u-turn reflections are more resilient than emergence from specular reflections. This difference comes from in the differing strengths of the two frustrations. However, the relationship between the number of topological neighbors needed for emergence to develop is consistent in both the u-turn and specular reflections.

Finally, the parameter space of the free boids implementation is not included because the free boids implementation does not have variations in the number of flockmates n, resulting in a three-dimensional parameter space $\{N, L, v_0\}$. Additionally, the style of frustration is not a small variation of either the u-turn or the specular turn frustrations. Thus, the free boids frustration is different enough that it is not included for comparison.

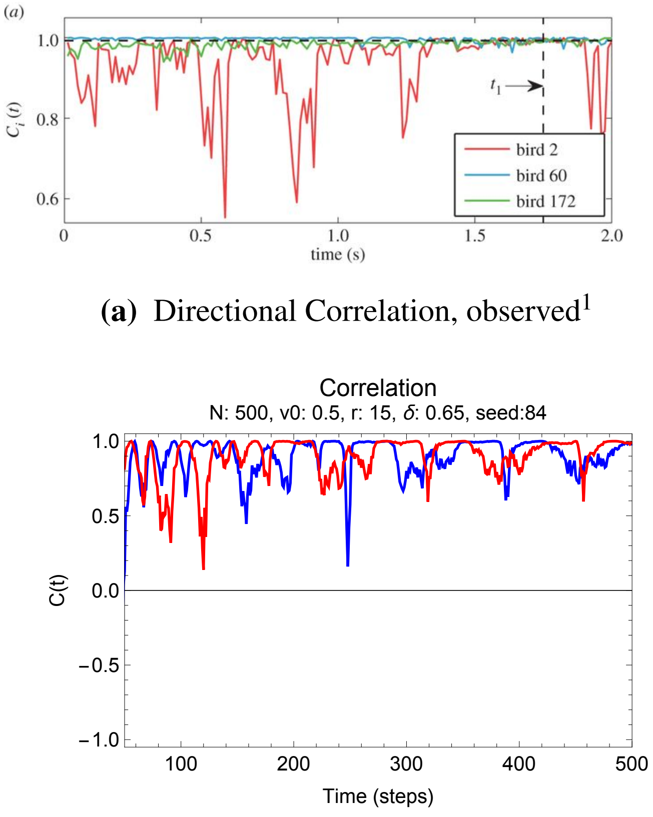
Chapter 5

Discussion

Models possess utility that allow for the prediction of results. However, simply because a model predicts results does not inherently mean the model is ‘correct.’ We have shown in this paper that our novel approach to modeling emergent flocking has proven to be robust and versatile. Even so, after eliminating the unphysical u-turn and specular reflections as frustrations that opposes consensus, our use of topological nearest neighbor interactions rather than spatially dependent nearest neighbor interactions seems counterintuitive. Surprisingly, comparisons of topological neighbors and turning radii between our model and the observations of European starlings (*Sturnus vulgaris*) made by A. Cavagna and his group in Rome, Italy [13], reveal our model to have remarkable accuracy and physical realism.

Topological interactions throughout members of a flock are rooted in observations [16]. Observations found that there are an average of six to seven topological neighbors that each bird interacted with [12], and that the number of interacting neighbors is independent of flock density [17]. Such observations are promising to our approach to understanding emergence. Additionally, observations show that flocks which use topological interactions to maintain cohesion are more robust to external perturbations and more likely to stay together [18].

Correlations between the velocity of an individual boid and its flockmate group also share strong similarities to the observational data made by Cavagna and his group (see Fig. 5.1) [13]. In Ref. [13], it is shown that when an individual bird begins to deviate strongly from the flock, a collective turn is initiated. Recreating the correlation function used in Ref. [13] [see Eq. (3.6) defined previously], we too have found this behavior in our model. When an individual boid deviates strongly from its neighbors, that 'signal' propagates through the flock, continuously instigating the



(a) Directional Correlation, observed[1]

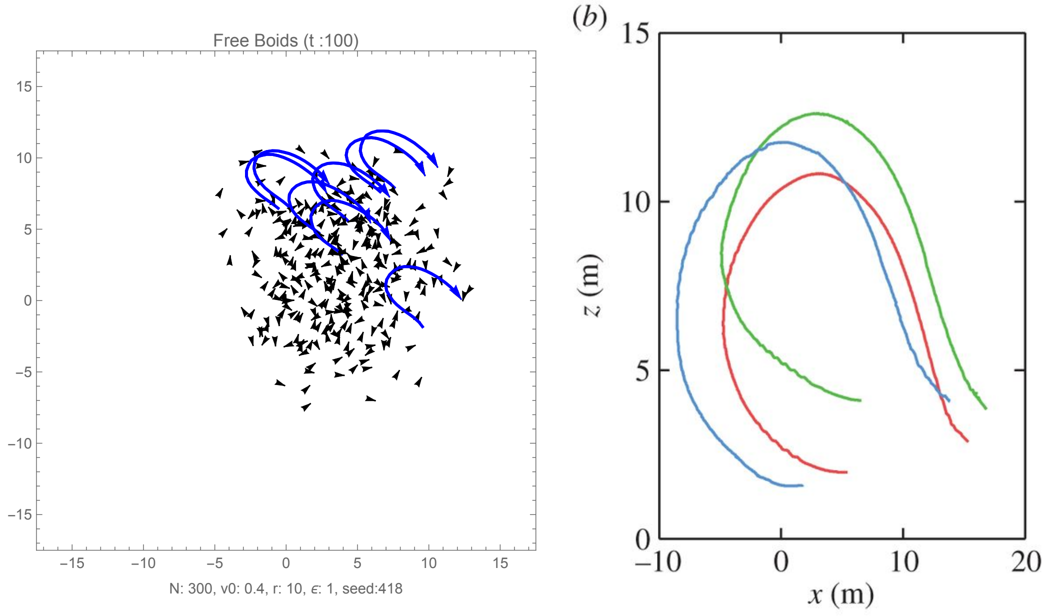
(b) Directional Correlation, model

Figure 5.1 The directional correlation [Eq. (3.6)] of two individual boids against a subgroup of the flock during a simulation done with our model (b), which shows that individual boids will at times deviate strongly from a large portion of the flock, matching observations (a) (originally published in Ref. [13]).

[1] Alessandro Attanasi et al. J. R. Soc. Interface 2015;12:20150319 © 2015 The Author(s) Published by the Royal Society. All rights reserved.

same type of turn. This motion can be seen in Fig. 5.1. See how the general shapes of the red and blue curves seem to alternate imitating each other as time progresses. This is what is meant by information propagating through the flock. However, please note how the exact shape changes over time and is not imitated identically. If the shape were to be imitated identically, then there would be no noise (frustration) in the system. Since there is frustration, the behavior is emergent and not phase locked. Fig. 5.1 (a) was originally published in Ref. [13].

Observations have also shown that when a cohesive flock turns, each bird turns at the same radius, regardless of their position in the flock [13]. Due to the mechanism of topological averaging used in our model, our simulations similarly show flocks turning with the same radius [see Fig. 5.2(a)]. Additionally, this signifies that flocks do not behave as rigid or solid body assemblies,



(a) Equal-radius turning, model

(b) Equal-radius turning, observed[2]

Figure 5.2 A snapshot of a simulation done with our model, (a), which shows that the when the flock turns the members of the flock all turn with the same radius, matching observations, (b) (originally published in Ref. [13]).

[2] Alessandro Attanasi et al. J. R. Soc. Interface 2015;12:20150319 © 2015 The Author(s) Published by the Royal

further implying that individuals tend to maintain a constant speed when turning (our model also implements all velocities with a constant magnitude). Furthermore, the general topology of the flock from our model remains the same throughout the turns, matching the observations that have been made, i.e. if there are no ‘holes’ in the flock, then after the flock has turned no additional holes have been created. See in Fig. 5.2 (a) the curves traced out by the turning boids all have the same radii of curvature. This matches the results found by Cavagna in Fig. 5.2 (b). Fig. 5.2 (b) was originally published in Ref. [13].

The characterization of the emergent flocking behavior has been challenging and difficult to achieve. Even with the order parameters and correlation functions that we have defined, determining whether or not a particular simulation produces emergent behavior (rather than random or globally repetitive motion) proved challenging. However, the parameter spaces we have been able to map out (see Fig. 4.1 and Fig. 4.3) show that there is a direct connection between the strength of consensus, the strength of frustration, and emergent flocking. Too much of either consensus or frustration will either result in a locked phase or random motion. Even so, given enough time (time steps), the sensitivity of the initial conditions and parameters used almost certainly result in a loss of cohesion and a decrease in emergent behavior.

The emergent flocking behavior that arises from our model of consensus and frustration is robust and versatile due to the topological interactions that underly its basic assumptions. This emergent behavior is coherent and robust to changes in the initial conditions and parameters. The great uniqueness that results in small changes to the initial conditions suggest that the tools of chaos theory may be useful in its analysis. This work, however, has yet to be done. A study of strange attractors may show correlations to phases and phase transitions and may provide insight to predicting behavior.

Society. All rights reserved.

Bibliography

[1] J. K. Parrish and L. Edelstein-Keshet, “Complexity, pattern, and evolutionary trade-offs in animal aggregation,” Science **284,** 99–101 (1999).

[2] P. W. Anderson, “More Is Different,” Science **177,** 393–396 (1972).

[3] M. R. Forster and A. Kryukov, “The Emergence of the Macroworld: A Study of Intertheory Relations in Classical and Quantum Mechanics,” Philosophy of Science **70,** 1039–1051 (2003).

[4] A. A. Tsonis and J. B. Elsner, “Chaos, Strange Attractors, and Weather,” Bulletin of the American Meteorological Society **70,** 14–23 (1989).

[5] P. L. Luisi, *The Emergence of Life : From Chemical Origins to Synthetic Biology* (Cambridge University Press, Cambridge, GB, 2006).

[6] C. Cleland, “Can we define ‘life’?,”, 2010.

[7] C. W. Reynolds, “Flocks, Herds and Schools: A Distributed Behavioral Model,” Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques **21,** 25 –34 (1987).

[8] T. Vicsek, A.CzirŮok, E. Ben-Jacob, I. Cohen, and O. Shochet, “Novel type of phase transition in a system of self-driven particles,” Phys. Rev. Lett. **75,** 1226–1229 (1995).

[9] A. Czirok and T. Vicsek, “Collective behavior of interacting self-propelled particles,” Physica A: Statistical Mechanics and its Applications **281,** 17 – 29 (2000).

[10] F. Ginelli and H. Chaté, “Relevance of Metric-Free Interactions in Flocking Phenomena,” Phys. Rev. Lett. **105,** 168103 (2010).

[11] H. Levine, W.-J. Rappel, and I. Cohen, “Self-organization in systems of self-propelled particles,” Phys. Rev. E **63,** 017101 (2000).

[12] M. Ballerini *et al.*, “Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study,” Proc. Natl. Acad. Sci. **105,** 1232–1235 (2008).

[13] A. Attanasi, A. Cavagna, L. Del Castello, I. Giardina, A. Jelic, S. Melillo, L. Parisi, O. Pohl, E. Shen, and M. Viale, “Emergence of collective changes in travel direction of starling flocks from individual birds’ fluctuations,” Journal of The Royal Society Interface 12 (2015).

[14] M. Berrondo and W. Krueger, “The role of consensus and frustration in the emergence of flocking behavior,” UASAL Journal **85,** 219–232 (2008).

[15] M. Berrondo and M. Sandoval, “Defining emergence: Learning from flock behavior,” Complexity **21,** 69–78 (2015).

[16] A. Cavagna, A. Cimarelli, I. Giardina, G. Parisi, R. Santagati, F. Stefanini, and M. Viale, “Scale-free correlations in starling flocks,” Proceedings of the National Academy of Sciences **107,** 11865–11870 (2010).

[17] W. Bialek, A. Cavagna, I. Giardina, T. Mora, E. Silvestrib, M. Viale, and A. M. Walczak, “Statistical mechanics for natural flocks of birds,” Proc. Natl. Acad. Sci. **109,** 1–6 (2012).

[18] M. Camperi, A. Cavagna, I. Giardina, G. Parisi, and E. Silvestri, “Spatially balanced topological interaction grants optimal cohesion in flocking models,” Interface Focus **2,** 715–725 (2012).

Appendix A

Translating from MATLAB to *Mathematica*

The decision to translate the code from MATLAB to *Mathematica* was more of a stylistic choice, as well as a way for me to really understand the work that I was building off of. My familiarity with MATLAB and my proficiency with *Mathematica* was an excellent intersection to expand on the work we were doing.

I have included the significant portions of the code that we used to simulate the flocking behavior and calculate the order parameters and correlation functions. Unfortunately the readability of code can be difficult so I have inserted various comments to assist the reader, but I would recommend a comparing the coded versions of the calculations with those found in Chapter 2 and Chapter 3.

Default *Mathematica* functions are defined with CamelCase and are colored black. Functions defined by me are displayed in CamelCase blue. Constants and variables defined within those functions are colored green. Comments begin with, (*, and end with, *), and are colored turquoise. Other text is generally lowercase and black.

Boid Flocking

```
Get["/BoidFlock.m"]
```

Constants

```
totaln = 300;
flockmates = 7;
v0  =  0.4;
radius  =  10;
maxtime = 100;
seed = 418;
frustration = "SU";
noise = 1;
gp1  =  1;
gp2  =  totaln / 2;
displaygroups = True;
```

Frustrations

Circle

```
CircleWalls[totaln, flockmates, v0, radius,
 maxtime, seed, frustration, gp1, gp2, displaygroups]
```

Free Boids

```
FreeBoids[totaln, v0, radius, maxtime, seed, noise, gp1, gp2, displaygroups]
```

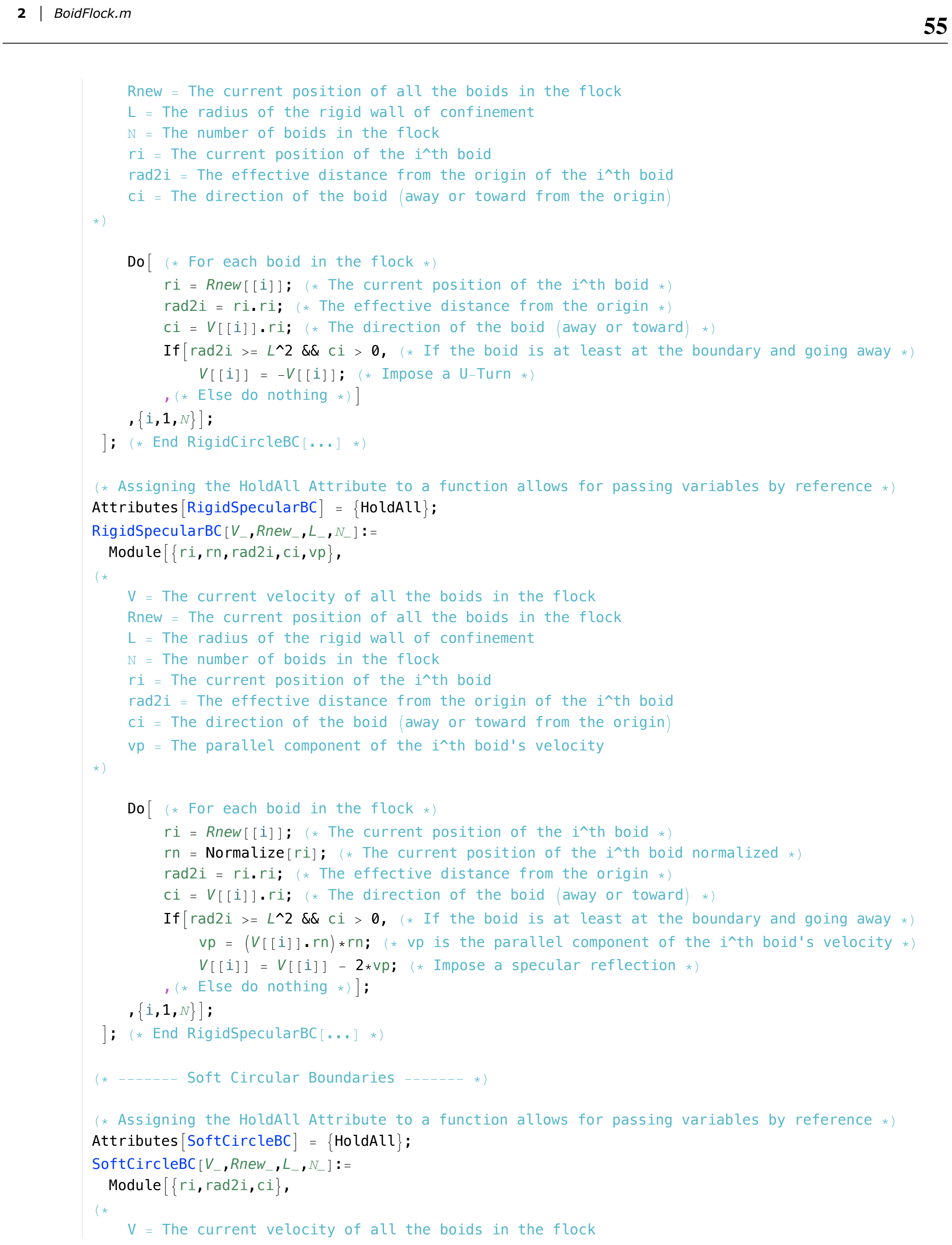
```
BeginPackage["BoidFlock`"]

CircleWalls::usage = "returns an animation of boids confined to a rigid or soft circle";
FreeBoids::usage = "returns an animation of free boids and returns the flock's global and
                        local order parameters";

Begin["`Private`"]
```

Consensus

```
(* This function averages the velocities of the any given boid with its flockmates *)
VAver[V_,v0_,nMates_,N_]:=
  Module[{Vlong,s,c,norm,ind},
(*
    V = Velocities of all the boids
    v0 = The constant velocities
    nMates = The number of flockmates
    N = The total number of boids
    Vlong = Length[Vlong] == Length{V] + nMates
            Vlong extends the array to make it 'appear' cyclical
    s = the average of the y component of the velocities
    c = the average of the x component of the velocities
    norm = the norm of the new velocity
    ind = the index being used to access all of the flockmates of a particular boid
*)

    (* Calculate the 'circular' array by combining two lists and flattening them appropriately *)
    Vlong = Flatten[{V,V[[1;;nMates]]},1];
    (* Pre-allocating s, c, and norm *)
    s = ConstantArray[0,N];
    c = ConstantArray[0,N];
    norm = ConstantArray[0,N];

    (* Calculates the new x and y components of the velocites for each boid *)
    Do[
        ind = (j;;nMates+j);
        s[[j]] = Mean[Vlong[[ind,2]]];
        c[[j]] = Mean[Vlong[[ind,1]]];
        norm[[j]] = Sqrt[s[[j]]^2+c[[j]]^2];
    ,{j,1,N}];
    (* Returns the combined x and y components of the velocities and scales them accordingly *)
    Transpose[v0*{c/norm,s/norm}]
  ]; (* End VAver[...] *)
```

Frustration (Rigid and Soft Boundary Conditions)

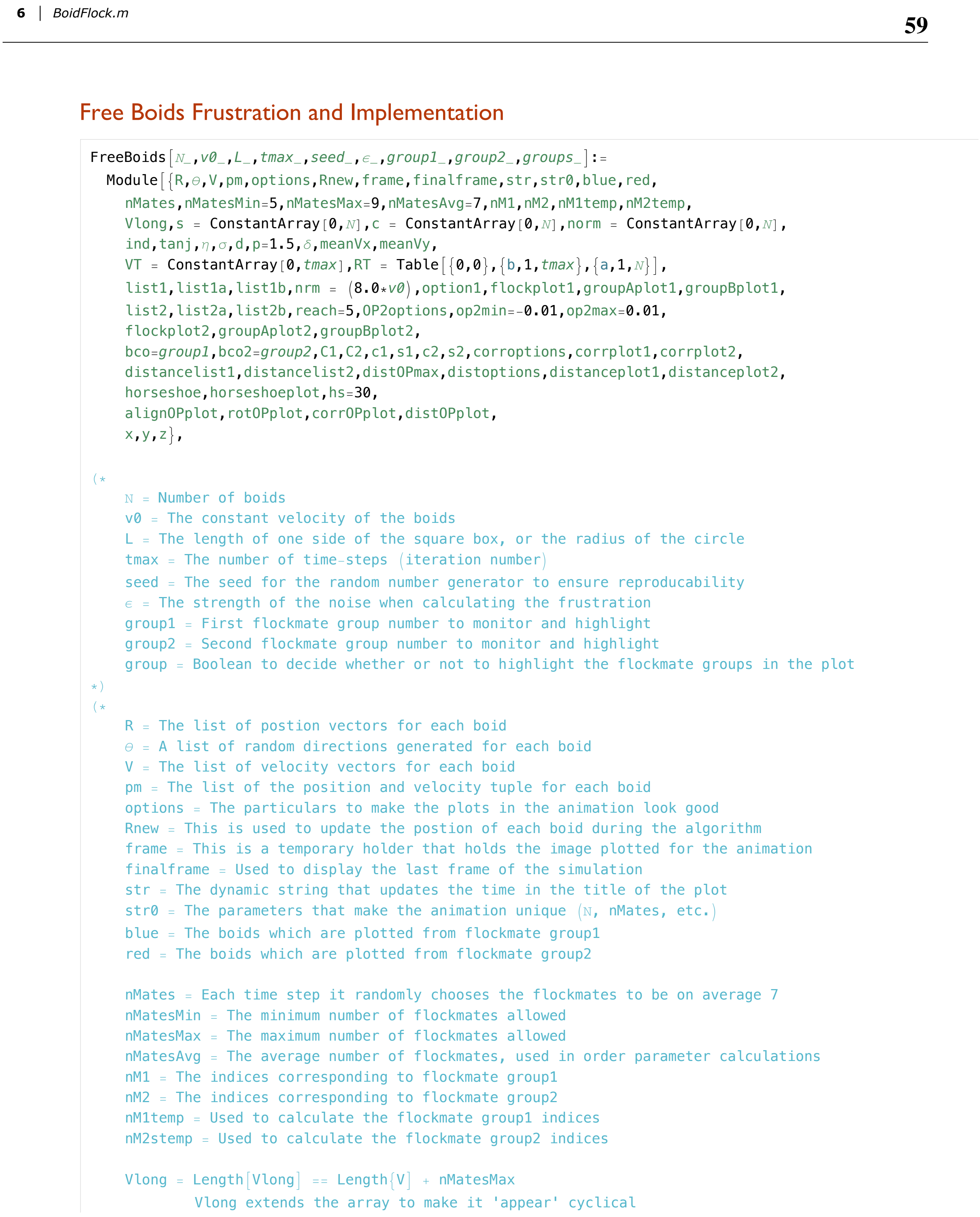
```
(* Assigning the HoldAll Attribute to a function allows for passing variables by reference *)
Attributes[RigidCircleBC] = {HoldAll};
RigidCircleBC[V_,Rnew_,L_,N_]:=
  Module[{ri,rad2i,ci},
(*
    V = The current velocity of all the boids in the flock
```



```
    Rnew = The current position of all the boids in the flock
    L = The radius of the rigid wall of confinement
    N = The number of boids in the flock
    ri = The current position of the i^th boid
    rad2i = The effective distance from the origin of the i^th boid
    ci = The direction of the boid (away or toward from the origin)
*)

    Do[ (* For each boid in the flock *)
        ri = Rnew[[i]]; (* The current position of the i^th boid *)
        rad2i = ri.ri; (* The effective distance from the origin *)
        ci = V[[i]].ri; (* The direction of the boid (away or toward) *)
        If[rad2i >= L^2 && ci > 0, (* If the boid is at least at the boundary and going away *)
            V[[i]] = -V[[i]]; (* Impose a U-Turn *)
        ,(* Else do nothing *)]
    ,{i,1,N}];
 ]; (* End RigidCircleBC[...] *)

(* Assigning the HoldAll Attribute to a function allows for passing variables by reference *)
Attributes[RigidSpecularBC] = {HoldAll};
RigidSpecularBC[V_,Rnew_,L_,N_]:=
  Module[{ri,rn,rad2i,ci,vp},
(*
    V = The current velocity of all the boids in the flock
    Rnew = The current position of all the boids in the flock
    L = The radius of the rigid wall of confinement
    N = The number of boids in the flock
    ri = The current position of the i^th boid
    rad2i = The effective distance from the origin of the i^th boid
    ci = The direction of the boid (away or toward from the origin)
    vp = The parallel component of the i^th boid's velocity
*)

    Do[ (* For each boid in the flock *)
        ri = Rnew[[i]]; (* The current position of the i^th boid *)
        rn = Normalize[ri]; (* The current position of the i^th boid normalized *)
        rad2i = ri.ri; (* The effective distance from the origin *)
        ci = V[[i]].ri; (* The direction of the boid (away or toward) *)
        If[rad2i >= L^2 && ci > 0, (* If the boid is at least at the boundary and going away *)
            vp = (V[[i]].rn)*rn; (* vp is the parallel component of the i^th boid's velocity *)
            V[[i]] = V[[i]] - 2*vp; (* Impose a specular reflection *)
        ,(* Else do nothing *)];
    ,{i,1,N}];
 ]; (* End RigidSpecularBC[...] *)

(* ------- Soft Circular Boundaries ------- *)

(* Assigning the HoldAll Attribute to a function allows for passing variables by reference *)
Attributes[SoftCircleBC] = {HoldAll};
SoftCircleBC[V_,Rnew_,L_,N_]:=
  Module[{ri,rad2i,ci},
(*
    V = The current velocity of all the boids in the flock
    Rnew = The current position of all the boids in the flock
    L = The radius of the rigid wall of confinement
    N = The number of boids in the flock
    ri = The current position of the i^th boid
    rad2i = The effective distance from the origin of the i^th boid
    ci = The direction of the boid (away or toward from the origin)
*)

    Do[ (* For each boid in the flock *)
        ri = Rnew[[i]]; (* The current position of the i^th boid *)
        rad2i = (ri.ri)/(L^2); (* The effective distance from the origin *)
        ci = V[[i]].ri; (* The direction of the boid (away or toward) *)
        (* If the boid is going away and consider the soft edge *)
        If[rad2i^3 >= (RandomReal[]) && ci > 0,
            V[[i]] = -V[[i]]; (* Impose a U-Turn *)
        ,(* Else do nothing *)];
    ,{i,1,N}];
 ]; (* End SoftCircleBC[...] *)

(* Assigning the HoldAll Attribute to a function allows for passing variables by reference *)
Attributes[SoftSpecularBC] = {HoldAll};
SoftSpecularBC[V_,Rnew_,L_,N_]:=
  Module[{ri,rn,rad2i,ci,vp},
(*
    V = The current velocity of all the boids in the flock
    Rnew = The current position of all the boids in the flock
    L = The radius of the rigid wall of confinement
    N = The number of boids in the flock
    ri = The current position of the i^th boid
    rad2i = The effective distance from the origin of the i^th boid
    ci = The direction of the boid (away or toward from the origin)
    vp = The parallel component of the i^th boid's velocity
*)

    Do[ (* For each boid in the flock *)
        ri = Rnew[[i]]; (* The current position of the i^th boid *)
        rn = Normalize[ri]; (* The current position of the i^th boid normalized *)
        rad2i = (ri.ri)/(L^2); (* The effective distance from the origin *)
        ci = V[[i]].ri; (* The direction of the boid (away or toward) *)
        (* If the boid is going away and consider the soft edge *)
        If[rad2i^3 >= (RandomReal[]) && ci > 0,
            vp = (V[[i]].rn)*rn; (* vp is the parallel component of the i^th boid's velocity *)
            V[[i]] = V[[i]] - 2*vp; (* Impose a specular reflection *)
        ,(* Else do nothing *)];
    ,{i,1,N}];
 ]; (* End SoftSpecularBC[...] *)
```

U-Turn and Specular Implementation

```
CircleWalls[N_,nMates_,v0_,L_,tmax_,seed_,BC_,group1_,group2_,groups_]:=
  Module[{R,θ,V,pm,options,Rnew,frame,str,str0,
    blue,red,nM1,nM2,nM1temp,nM2temp},
(*
    N = Number of boids
    nMates = Number of flockmates
    v0 = The constant velocity of the boids
    L = The length of one side of the square box, or the radius of the circle
    tmax = The number of time-steps (iteration number)
    seed = The seed for the random number generator to ensure reproducability
    BC = The type of boundary condition (U-turn or Specular) (Soft or Rigid)
    group1 = First flockmate group number to monitor and highlight
    group2 = Second flockmate group number to monitor and highlight
    group = Boolean to decide whether or not to highlight the flockmate groups in the plot
*)
(*
    R = The list of postion vectors for each boid
    θ = A list of random directions generated for each boid
    V = The list of velocity vectors for each boid
    pm = The list of the position and velocity tuple for each boid
    options = The particulars to make the plots in the animation look good
    Rnew = This is used to update the postion of each boid during the algorithm
    frame = This is a temporary holder that holds the image plotted for the animation
    str = The dynamic string that updates the time in the title of the plot
    str0 = The parameters that make the animation unique (N, nMates, etc.)

    blue = The boids which are plotted from flockmate group1
    red = The boids which are plotted from flockmate group2
    nM1 = The indices corresponding to flockmate group1
    nM2 = The indices corresponding to flockmate group2
    nM1temp = Used to calculate the flockmate group1 indices
    nM2stemp = Used to calculate the flockmate group2 indices
*)


    (* Sets the string to display which BC is being plotted *)
    str = "Circle BC";
    (* Sets the string to display which parameters are being used *)
    str0 = StringJoin["N: ",ToString[N],", nM: ",ToString[nMates],⋱
        ", v0: ",ToString[v0],", r: ",ToString[L],", seed:",ToString[seed],⋱
         ", BC: ",ToString[BC]];

    (* Sets the random seed *)
    If[seed==0,,SeedRandom[seed]];

    (* This  initialises the random starting postions of the boids *)
    R=RandomReal[2L/2,{N,2}] - L/2;

    (* Determines the random starting velocities of the boids *)
    θ = 2π*(RandomReal[1,N]-0.5);
    V = Transpose[{v0*Cos[θ],v0 Sin[θ]}];

    (* Creates the position-momentum array (mostly for plotting) *)
    pm = Transpose[{R,R+V}];

    (* This guarantees the flockmate group numbers are valid *)
    If[group1 >= N-nMates, nM1temp = Round[N-nMates];, nM1temp = Round[group1];];
    If[group2 >= N-nMates, nM2temp = Round[N-nMates];, nM2temp = Round[group2];];
    nM1 = (nM1temp;;(nM1temp+nMates));
    nM2 = (nM2temp;;(nM2temp+nMates));

    (* Runs the algorithm (Monitor is what produces the animation) *)
    Monitor[
        Do[
            (* Update the time-step in the plot *)
            str = StringJoin[{"Circle BC (t :",ToString[t],")"}];
            (* Determines the rest of the options in the plot *)
            options = {PlotLabel->str,FrameLabel->str0,PlotRange->{{-L,L},{-L,L}},Frame->True,
                ImageMargins->25,BaseStyle-> {FontSize->12}};

            (* Assigns colors to the two chosen flockmate groups *)
            blue = pm[[nM1]];
            red = pm[[nM2]];

            (* Plots the boids (without highlighting the flockmate groups) *)
            If[!groups,frame = Graphics[{Arrowheads[0.02],Arrow[pm]},options];,""];
            (* Plots the boids with the two highlighted flockmate groups *)
            If[groups,frame = Graphics[{{Arrowheads[0.02],Arrow[pm]},{Blue,Arrowheads[0.03],
                Arrow[blue]},s{Red,Arrowheads[0.03],Arrow[red]}},options];,""];

            (* Gets the current position of the boids *)
            Rnew = R+V;

            (* Determines the new velocites through flockmate averaging *)
            V = VAver[V,v0,nMates,N];

            (* Implements the appropriate boundary condition based on the user's input *)
            (* All values passed into the functions are passed by reference *)
            Which[
                BC == "SU", SoftCircleBC[V,Rnew,L,N];,
                BC == "U", RigidCircleBC[V,Rnew,L,N];,
                BC == "SS", SoftSpecularBC[V,Rnew,L,N];,
                BC == "S", RigidSpecularBC[V,Rnew,L,N];
            ];

            (* Updates the new positions *)
            R = Rnew;

            (* Recombines the new position and momentum for plotting purposes *)
            pm = Transpose[{R,R+V}];

            (* Pauses to allow for comfortable viewing/refresh rates in the animation *)
            Pause[0.05];

            (* Repeats for tmax time steps *)
        ,{t,1,tmax}];

        (* Prints out the final time step *)
        Graphics[{Arrowheads[0.02],Arrow[pm]},options]
    ,frame]
]; (* End CircleWalls[...] *)
```



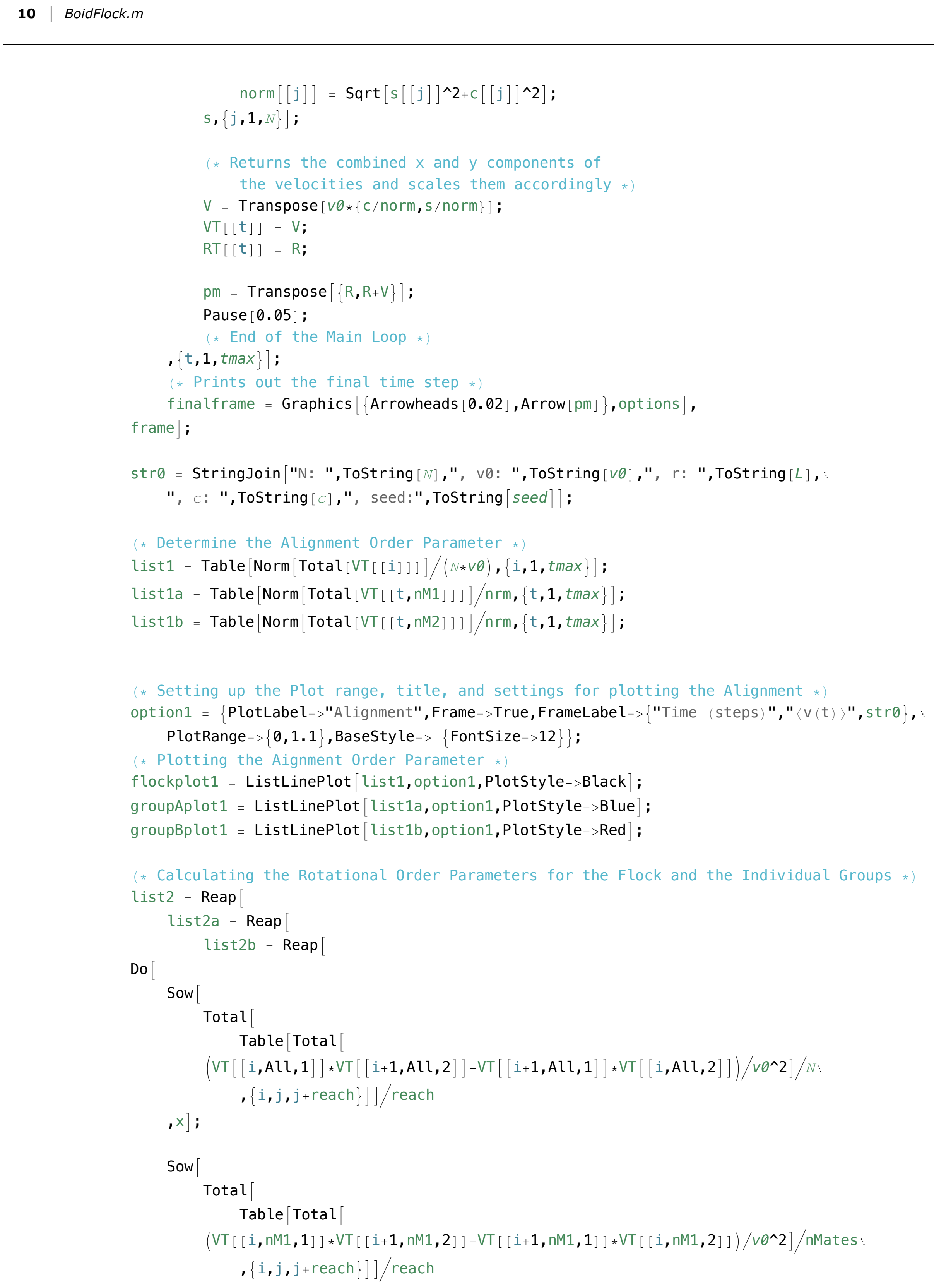
Free Boids Frustration and Implementation

```
FreeBoids[𝒩_,v0_,L_,tmax_,seed_,ϵ_,group1_,group2_,groups_]:=
  Module[{R,θ,V,pm,options,Rnew,frame,finalframe,str,str0,blue,red,
    nMates,nMatesMin=5,nMatesMax=9,nMatesAvg=7,nM1,nM2,nM1temp,nM2temp,
    Vlong,s = ConstantArray[0,𝒩],c = ConstantArray[0,𝒩],norm = ConstantArray[0,𝒩],
    ind,tanj,η,σ,d,p=1.5,δ,meanVx,meanVy,
    VT = ConstantArray[0,tmax],RT = Table[{0,0},{b,1,tmax},{a,1,𝒩}],
    list1,list1a,list1b,nrm = (8.0*v0),option1,flockplot1,groupAplot1,groupBplot1,
    list2,list2a,list2b,reach=5,OP2options,op2min=-0.01,op2max=0.01,
    flockplot2,groupAplot2,groupBplot2,
    bco=group1,bco2=group2,C1,C2,c1,s1,c2,s2,corroptions,corrplot1,corrplot2,
    distancelist1,distancelist2,distOPmax,distoptions,distanceplot1,distanceplot2,
    horseshoe,horseshoeplot,hs=30,
    alignOPplot,rotOPplot,corrOPplot,distOPplot,
    x,y,z},

(*
    𝒩 = Number of boids
    v0 = The constant velocity of the boids
    L = The length of one side of the square box, or the radius of the circle
    tmax = The number of time-steps (iteration number)
    seed = The seed for the random number generator to ensure reproducability
    ϵ = The strength of the noise when calculating the frustration
    group1 = First flockmate group number to monitor and highlight
    group2 = Second flockmate group number to monitor and highlight
    group = Boolean to decide whether or not to highlight the flockmate groups in the plot
*)
(*
    R = The list of postion vectors for each boid
    θ = A list of random directions generated for each boid
    V = The list of velocity vectors for each boid
    pm = The list of the position and velocity tuple for each boid
    options = The particulars to make the plots in the animation look good
    Rnew = This is used to update the postion of each boid during the algorithm
    frame = This is a temporary holder that holds the image plotted for the animation
    finalframe = Used to display the last frame of the simulation
    str = The dynamic string that updates the time in the title of the plot
    str0 = The parameters that make the animation unique (𝒩, nMates, etc.)
    blue = The boids which are plotted from flockmate group1
    red = The boids which are plotted from flockmate group2

    nMates = Each time step it randomly chooses the flockmates to be on average 7
    nMatesMin = The minimum number of flockmates allowed
    nMatesMax = The maximum number of flockmates allowed
    nMatesAvg = The average number of flockmates, used in order parameter calculations
    nM1 = The indices corresponding to flockmate group1
    nM2 = The indices corresponding to flockmate group2
    nM1temp = Used to calculate the flockmate group1 indices
    nM2stemp = Used to calculate the flockmate group2 indices

    Vlong = Length[Vlong] == Length{V] + nMatesMax
            Vlong extends the array to make it 'appear' cyclical
```

```
s = The average of the y component of the velocities
c = The average of the x component of the velocities
norm = The norm of the new velocity

ind = The index being used to access all of the flockmates of a particular boid
tanj = The tangential positional vector for the j^th boid
η = A normally distributed random number
σ = The Sign of the direction of the velocity
d = The effective edge of the simulation area (not strictly enforced)
p = The power used to calculate d, changes the strength of the soft basin
δ = The perturbation of the velocity (the frustration)
meanVx = The average of the x component of the boid's flockmates
meanVy = The average of the y component of the boid's flockmates

VT = The velocity of every boid at every timestep, updated each timestep
RT = The position of every boid at every timestep, updated each timestep

list1 = Used for plotting the aligned order parameter after calculating it
list1a = Used for plotting the aligned order parameter for flockmate group1
            after calculating it
list1b = Used for plotting the aligned order parameter for flockmate group2
            after calculating it
nrm = Defines a consistent number of flockmates to consider for the norm based
            on the average
option1 = Defines the plot options for the alignment order parameter
flockplot1 = Plots the entire flock's alignment order parameter
groupAplot1 = Plots the flockmate group1's alignment order parameter
groupBplot1 = Plots the flockmate group2's alignment order parameter

list2 = Used for plotting the rotational order parameter after calculating it
list2a = Used for plotting the rotational order parameter for flockmate group1
            after calculating it
list2b = Used for plotting the rotational order parameter for flockmate group2
            after calculating it
reach = Defines how far back in time to consider the curvature of the flock or
            flockmate group
OP2options = Defines the plot options for the rotational order parameter
op2min = Used to guarantee that there isn't too much white space in the rotational plot
op2max = Used to guarantee that there isn't too much white space in the rotational plot

flockplot2 = Plots the entire flock's rotational order parameter
groupAplot2 = Plots the flockmate group1's rotational order parameter
groupBplot2 = Plots the flockmate group2's rotational order parameter

bco = The boid being considered for the directional correlation function, first
        boid in group1
bco2 = The boid being considered for the directional correlation function, first
        boid in group2
C1 = The correlation of boid bco for each time step
C2 = The correlation of boid bco2 for each time step
c1 = The normalized velocity of boid bco at time t
s1 = The normalized average of multiple, topologically consecutive boids from
        boid bco at time t
c2 = The normalized velocity of boid bco2 at time t
s2 = The normalized average of multiple, topologically consecutive boids from
```

```
            boid bco2 at time t
    corroptions = The options used in the directional correlation plot
    corrplot1 = The plot of the directional correlation for boid bco
    corrplot2 = The plot of the directional correlation for boid bco2

    distancelist1 = The values for the average distance between flockmates for
                        flockmate group1
    distancelist2 = The values for the average distance between flockmates for
                        flockmate group2
    distOPmax = Used to reduce the amount of white space in the plot
    distoptions = The options used in the average distance plot
    distanceplot1 = The plot of the average distance between flockmates for
                        flockmate group1
    distanceplot2 = The plot of the average distance between flockmates for
                        flockmate group1

    horseshoe = Lines making a trail of the positions of the boids in both flockmate
        group1 and flockmate group2
    horseshoeplot = The superposition of the horseshoe lines on the last frame of
                        the animation
    hs = The number of timesteps used to plot the 'horseshoe tail'

    alignOPplot = The superposition of all of the alignment plots
                    (flockplot1, groupAplot1, groupBplot1)
    rotOPplot = The superposition of all of the rotational plots
                    (flockplot2, groupAplot2, groupBplot2)
    corrOPplot = The superposition of the two directional correlation plots
                    (corrplot1, corrplot2)
    distOPplot = the superposition of the two average distance plots
                    (distanceplot1, distanceplot2)

    x,y,z = An indicator used to signifiy which Sow goes with which Reap when
                calculating order parameters
*)

    (* Sets the string to display which BC is being plotted *)
    str = "Free Boids";
    (* Sets the string to display which parameters are being used *)
    str0 = StringJoin["N: ",ToString[N],", v0: ",ToString[v0],", r: ",ToString[L],
        ", ϵ: ",ToString[ϵ],", seed:",ToString[seed]];

    (* Initialise the seed *)
    SeedRandom[seed];

    (* Initialise the random positions *)
    R=RandomReal[2L/2,{N,2}] - L/2;

    (* Initialise the random velocity vectors *)
    θ = 2π*(RandomReal[1,N]-0.5);
    V = Transpose[{v0 Cos[θ],v0 Sin[θ]}];

    (* Creates the position-momentum array (mostly for plotting) *)
    pm = Transpose[{R,R+V}];
```

```
(* Determine the list-reach for the two groups *)
If[group1 >= N-nMatesAvg, nM1temp = Round[N-nMatesAvg];, nM1temp = Round[group1];];
If[group2 >= N-nMatesAvg, nM2temp = Round[N-nMatesAvg];, nM2temp = Round[group2];];
nM1 = (nM1temp;;(nM1temp+nMatesAvg));
nM2 = (nM2temp;;(nM2temp+nMatesAvg));

(* Running the Main Loop *)
(* Evaluated in time steps (t) in the loop *)
(* Runs the algorithm (Monitor is what produces the animation) *)
Monitor[
    Do[
        (* Determine the options in the plot *)
        str = StringJoin[{"Free Boids (t :",ToString[t],")"}];
        options = {PlotLabel->str,FrameLabel->str0,PlotRange->1.75*{{-L,L},{-L,L}},
            Frame->True,AspectRatio->1,BaseStyle-> {FontSize->12}};

        (* Extracts the highlighted flockmate group's data for plotting *)
        blue = pm[[nM1]];
        red = pm[[nM2]];

        (* Plots the boids (without highlighting the flockmate groups) *)
        If[!groups,frame = Graphics[{Arrowheads[0.02],Arrow[pm]},options];,""];
        (* Plots the boids with the two highlighted flockmate groups *)
        If[groups,frame = Graphics[{{Arrowheads[0.02],Arrow[pm]},{Blue,Arrowheads[0.03],
            Arrow[blue]},{Red,Arrowheads[0.03],Arrow[red]}},options];,""];

        Rnew = R+V; (* Update the position and velocity *)
        R = Rnew; (* Update the new position *)

        (* Calculate the 'cyclical' array by combining
             two lists and 'flattening' them appropriately *)
        Vlong = Flatten[{V,V[[1;;nMatesMax]]},1];

        (* Calculates the new x and y components of the velocites for each boid *)
        Do[
            nMates = RandomInteger[{5,nMatesMax}];

            tanj = {R[[j,2]], -R[[j,1]]};

            η = RandomVariate[NormalDistribution[0,0.01]];
            If[R[[j]].V[[j]] > 0,
                σ = Sign[tanj.V[[j]]]*Abs[η];,
                σ = η;
            ];
            d = Abs[(R[[j]].R[[j]]/L^2)^p];
            δ = ϵ*σ*d;

            ind = (j;;nMates+j);
            meanVx = Mean[Vlong[[ind,1]]];
            meanVy = Mean[Vlong[[ind,2]]];
            c[[j]] = meanVx + δ*meanVy;
            s[[j]] = meanVy - δ*meanVx;
```



```
            norm[[j]] = Sqrt[s[[j]]^2+c[[j]]^2];
        s,{j,1,N}];

        (* Returns the combined x and y components of
            the velocities and scales them accordingly *)
        V = Transpose[v0*{c/norm,s/norm}];
        VT[[t]] = V;
        RT[[t]] = R;

        pm = Transpose[{R,R+V}];
        Pause[0.05];
        (* End of the Main Loop *)
    ,{t,1,tmax}];
    (* Prints out the final time step *)
    finalframe = Graphics[{Arrowheads[0.02],Arrow[pm]},options],
frame];

str0 = StringJoin["N: ",ToString[N],", v0: ",ToString[v0],", r: ",ToString[L],⋱
    ", ϵ: ",ToString[ϵ],", seed:",ToString[seed]];

(* Determine the Alignment Order Parameter *)
list1 = Table[Norm[Total[VT[[i]]]]/(N*v0),{i,1,tmax}];
list1a = Table[Norm[Total[VT[[t,nM1]]]]/nrm,{t,1,tmax}];
list1b = Table[Norm[Total[VT[[t,nM2]]]]/nrm,{t,1,tmax}];


(* Setting up the Plot range, title, and settings for plotting the Alignment *)
option1 = {PlotLabel->"Alignment",Frame->True,FrameLabel->{"Time (steps)","〈v(t)〉",str0},⋱
    PlotRange->{0,1.1},BaseStyle-> {FontSize->12}};
(* Plotting the Aignment Order Parameter *)
flockplot1 = ListLinePlot[list1,option1,PlotStyle->Black];
groupAplot1 = ListLinePlot[list1a,option1,PlotStyle->Blue];
groupBplot1 = ListLinePlot[list1b,option1,PlotStyle->Red];

(* Calculating the Rotational Order Parameters for the Flock and the Individual Groups *)
list2 = Reap[
    list2a = Reap[
        list2b = Reap[
Do[
    Sow[
        Total[
            Table[Total[
        (VT[[i,All,1]]*VT[[i+1,All,2]]-VT[[i+1,All,1]]*VT[[i,All,2]])/v0^2]/N⋱
            ,{i,j,j+reach}]]/reach
    ,x];

    Sow[
        Total[
            Table[Total[
        (VT[[i,nM1,1]]*VT[[i+1,nM1,2]]-VT[[i+1,nM1,1]]*VT[[i,nM1,2]])/v0^2]/nMates⋱
            ,{i,j,j+reach}]]/reach
    ,y];

    Sow[
        Total[
            Table[Total[
        (VT[[i,nM2,1]]*VT[[i+1,nM2,2]]-VT[[i+1,nM2,1]]*VT[[i,nM2,2]])/v0^2]/nMates
            ,{i,j,j+reach}]]/reach
    ,z];

,{j,1,tmax-reach-1}]
                     ,z][[2,1]]
                 ,y][[2,1]]
            ,x][[2,1]];

(* Setting up the Plot Range of the Rotational Plot *)
op2max = Max[list2,list2a,list2b,op2max]+0.05;
op2min = Min[list2,list2a,list2b,op2min]-0.05;
OP2options = {PlotLabel->"Rotational",Frame->True,FrameLabel->{"Time (steps)","⟨L(t)⟩",str0},
    PlotRange->{op2min,op2max},BaseStyle-> {FontSize->12}};
(* Plotting the Rotational Order Parameter *)
flockplot2 = ListLinePlot[list2,OP2options,PlotStyle->Black];
groupAplot2 = ListLinePlot[list2a,OP2options,PlotStyle->Blue];
groupBplot2 = ListLinePlot[list2b,OP2options,PlotStyle->Red];

(* Correlation Functions *)
C1 = Reap[
  C2 = Reap[
    Do[
        s1 = Normalize[
    {Total[VT[[t,bco;;(bco + 2*nMatesAvg),1]]],Total[VT[[t,bco;;(bco + 2*nMatesAvg),2]]]}
            ];
        c1 = Normalize[VT[[t,bco]]];
        Sow[s1.c1,x];
        s2 = Normalize[
    {Total[VT[[t,bco2;;(bco2+2*nMatesAvg),1]]],Total[VT[[t,bco2;;(bco2+2*nMatesAvg),2]]]}
            ];
        c2 = Normalize[VT[[t,bco2]]];
        Sow[s2.c2,y];
    ,{t,1,tmax}]
          ,y][[2,1]]
        ,x][[2,1]];

corroptions = {PlotLabel->"Correlation",Frame->True,FrameLabel->{"Time (steps)","C(t)",str0},
    PlotRange->{{50,tmax},{-1.05,1.05}},BaseStyle-> {FontSize->12}};
corrplot1 = ListLinePlot[C1,corroptions,PlotStyle->Blue];
corrplot2 = ListLinePlot[C2,corroptions,PlotStyle->Red];

(* Distance between Flockmates *)
distancelist1 = Reap[
    distancelist2 = Reap[
        Do[
                 Sow[Mean[Table[Norm[ (RT[[t,nM1]][[1]]-RT[[t,nM1]][[i]]) ],{i,2,nMatesAvg}]],x];
                 Sow[Mean[Table[Norm[ (RT[[t,nM2]][[1]]-RT[[t,nM2]][[i]]) ],{i,2,nMatesAvg}]],y];
            ,{t,1,tmax}]
                        ,y][[2,1]]
                    ,x][[2,1]];

    distOPmax = Max[distancelist1,distancelist2]+0.1;
    distoptions = {PlotLabel->"Average Distance between Flockmates",Frame->True,
        FrameLabel->{"Time (steps)","⟨R(t)⟩",str0},PlotRange->{0,distOPmax},
        BaseStyle-> {FontSize->12}};
    distanceplot1 = ListLinePlot[distancelist1,distoptions,PlotStyle->Blue];
    distanceplot2 = ListLinePlot[distancelist2,distoptions,PlotStyle->Red];

    (* Horseshoe Plot *)
    horseshoe = Graphics[{{Blue,Thick,Line[Transpose[RT][[nM1,tmax-hs;;tmax]]]},
        {Red,Thick,Line[Transpose[RT][[nM2,tmax-hs;;tmax]]]}}];
    horseshoeplot = Show[{frame,horseshoe}];

    (* Creating a nice graphic to display all of the information *)
    alignOPplot = Show[{flockplot1,groupAplot1,groupBplot1}];
    rotOPplot = Show[{flockplot2,groupAplot2,groupBplot2}];
    corrOPplot = Show[{corrplot1,corrplot2}];
    distOPplot = Show[{distanceplot1,distanceplot2}];
    CellPrint[{ExpressionCell[Style[finalframe,Large],"Output"],
        ExpressionCell[Style[alignOPplot,Large],"Output"],
        ExpressionCell[Style[rotOPplot,Large],"Output"],
        ExpressionCell[Style[corrOPplot,Large],"Output"],
        ExpressionCell[Style[distOPplot,Large],"Output"],
        ExpressionCell[Style[horseshoeplot,Large],"Output"]}]
]; (* End FreeBoids[...] *)

End[]
EndPackage[]
```

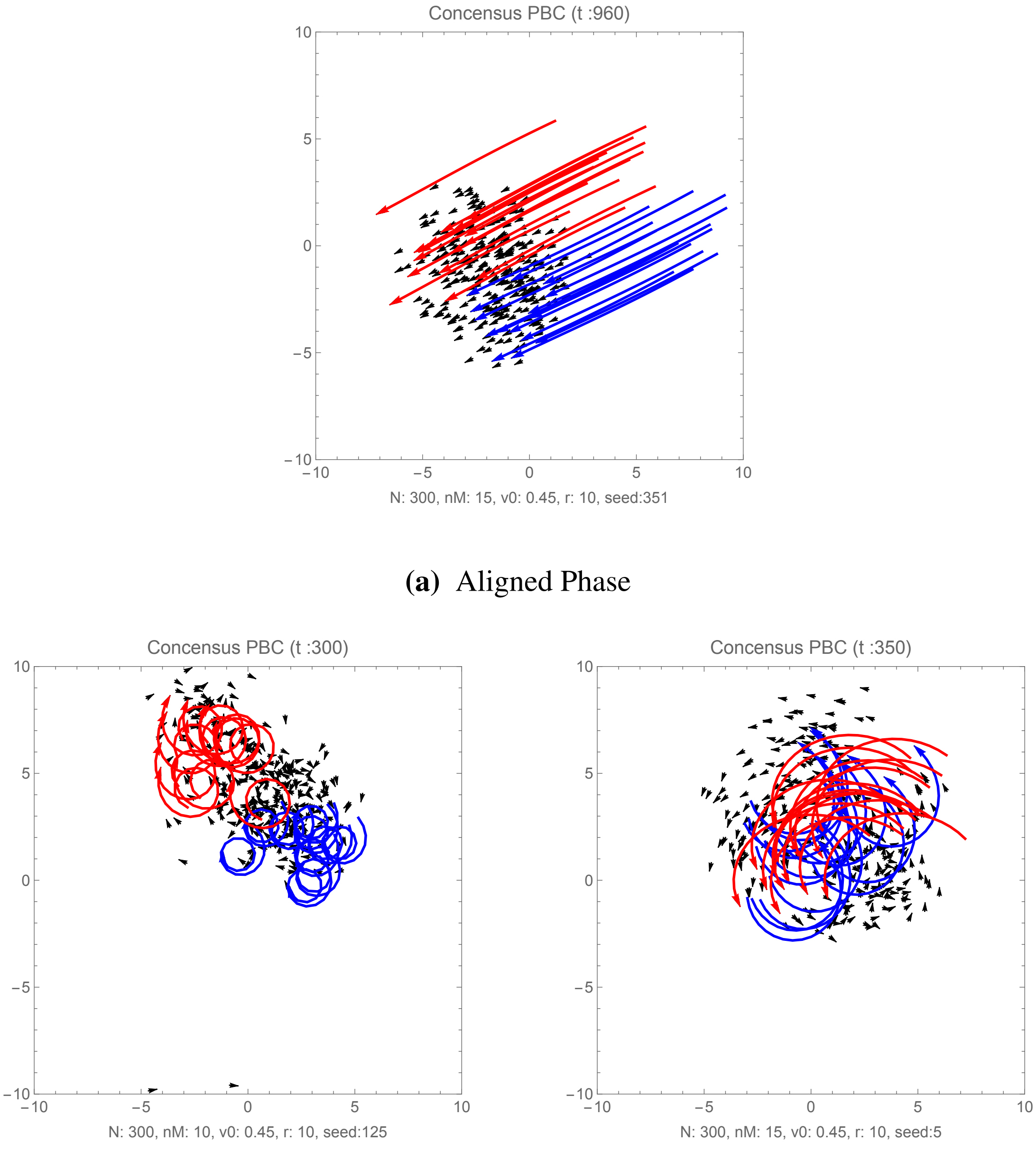
Appendix B

Additional Figures and Animations

In addition to a number of figures included in this appendix, there are also a number of videos on YouTube that we have posted that better capture the flocking behavior. The videos can be found by clicking this link: Boids, or by going to this url:

https://www.youtube.com/channel/UCQlRJQYvymrpZMwW9stdj0w

or by searching for ‘boids’ in the YouTube search bar, filtering for ‘channels’, and then selecting the channel **Boids**.



(a) Aligned Phase

(b) CW Phase

(c) CCW Phase

Figure B.1 Here is a side-by-side comparison of **Figures 2.3**, **2.4**, and **2.5**

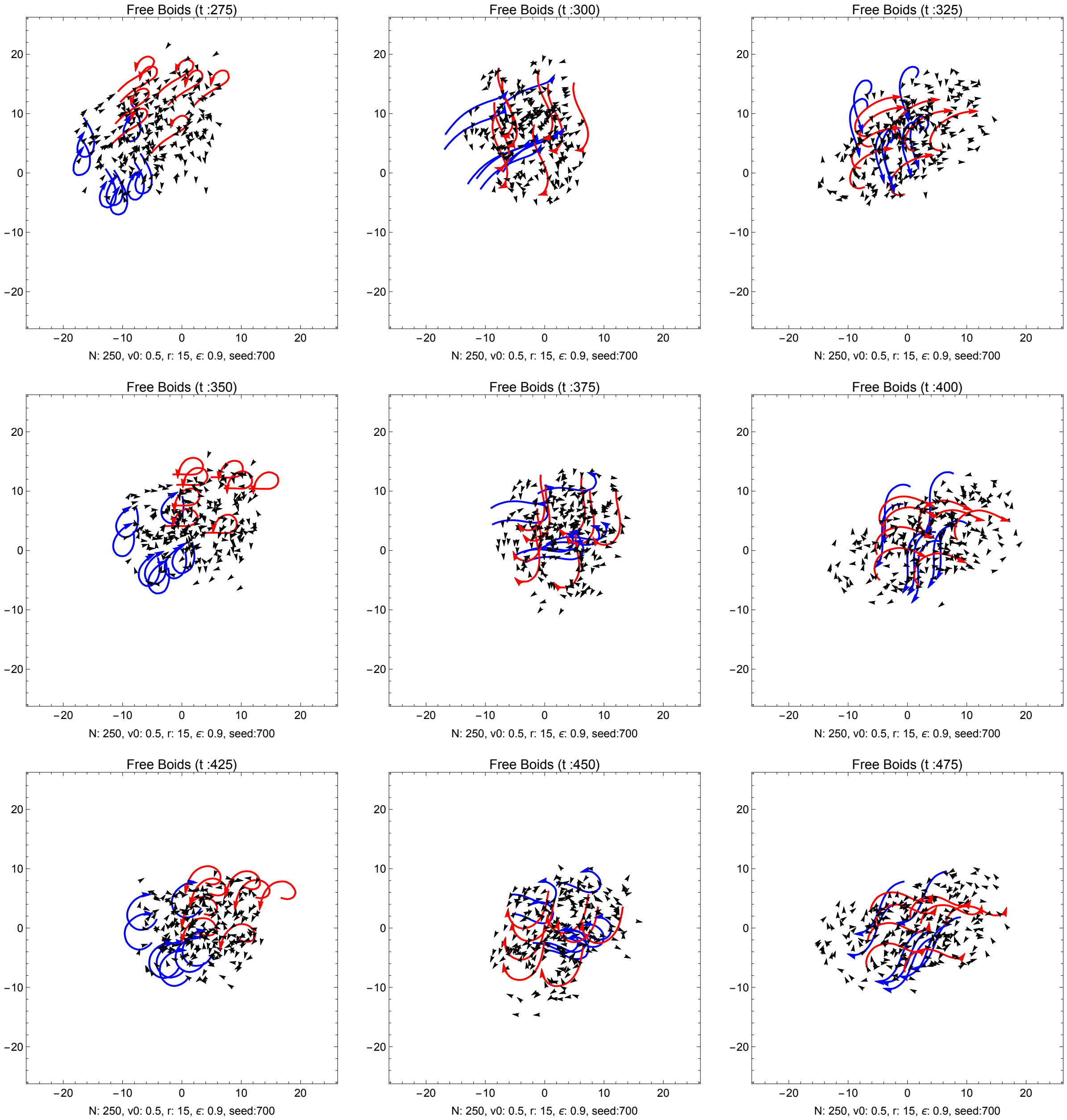
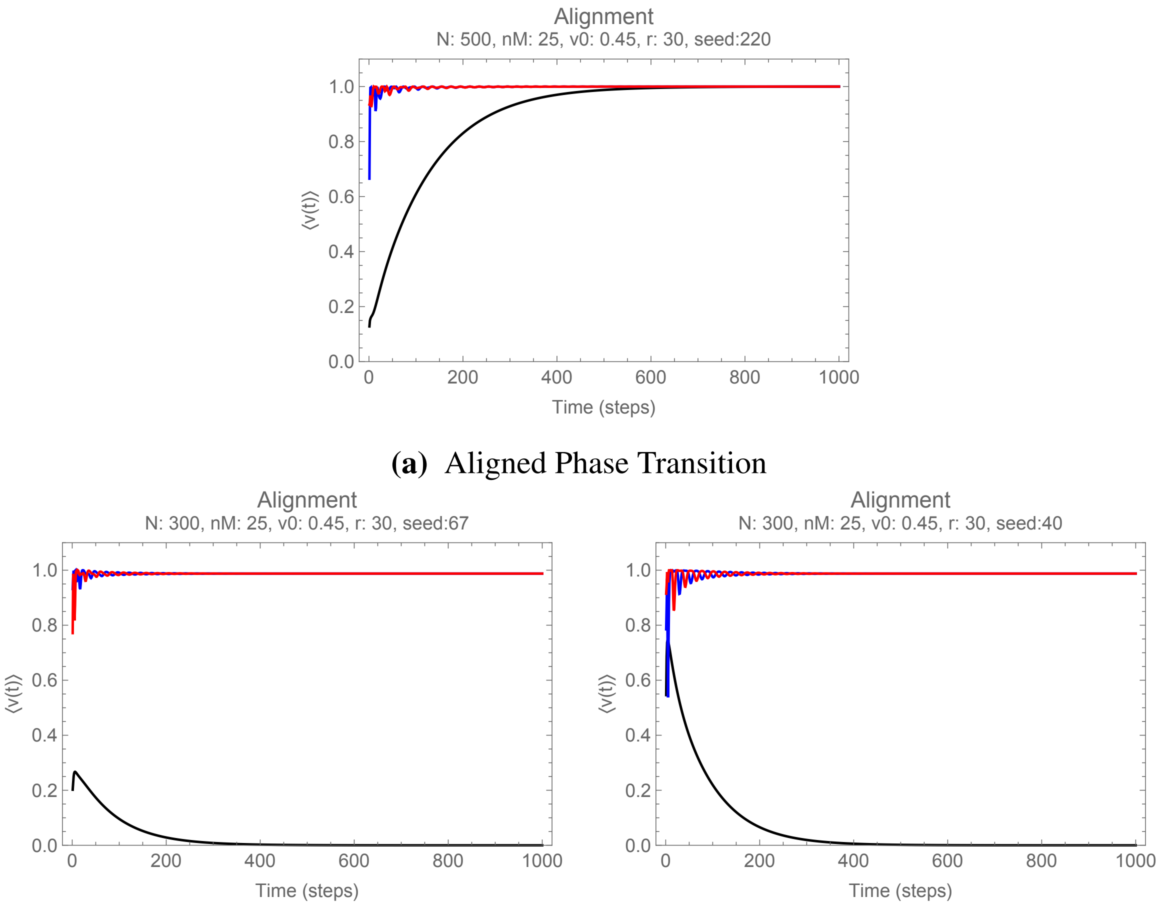


Figure B.2 These figures are a continuation of the snapshots that are also found in Fig. 2.13.

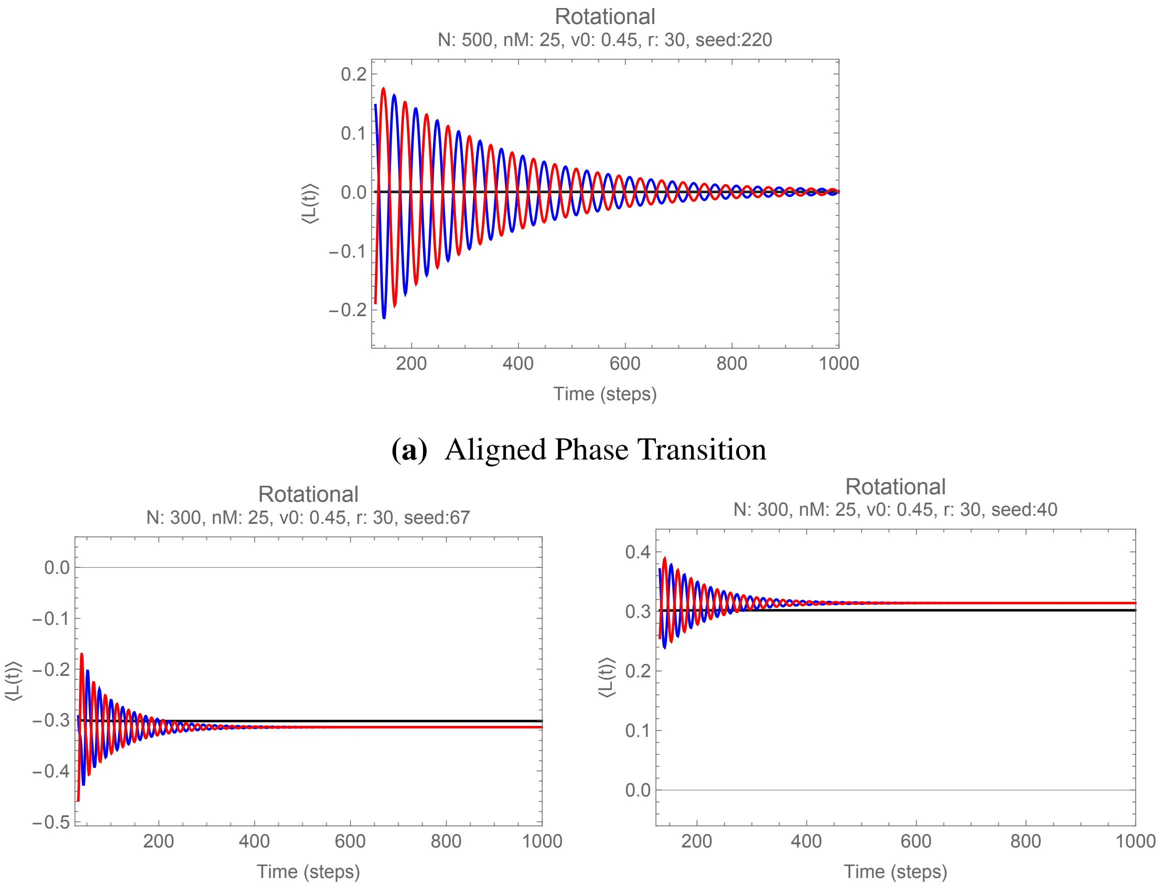


(a) Aligned Phase Transition

(b) Clockwise Phase Transition

(c) Counter-Clockwise Transition

Figure B.3 Changes in the alignment order parameter during flocking simulations. The black curve is the alignment of the entire flock, $\langle v(t) \rangle$, as calculated by Eq. (3.1) and the red and blue curves are the alignments of two separate flockmate groups, $\langle v_i(t) \rangle$, as calculated by Eq. (3.2). As can be seen, it's essentially impossible to distinguish between a clockwise and counter-clockwise phase transition. Additionally, the phase transitions in this model are not dramatic and sharp due to the transient nature discussed in Chapter 2.2.1



(a) Aligned Phase Transition

(b) Clockwise Phase Transition

(c) Counter-Clockwise Transition

Figure B.4 Changes in the rotational order parameter during flocking simulations. The black curve is the rotation of the entire flock, $\langle L(t) \rangle$, as calculated by Eq. (3.3) and the red and blue curves are the rotation of two separate flockmate groups, $\langle L_i(t) \rangle$, as calculated by Eq. (3.5). A clockwise rotation is negative, a counter-clockwise rotation is positive, and an aligned phase is zero.

Index