Emergence in Flocking Boids with Graph Theory

Christian Lambert

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
February 28, 2019

ABSTRACT

Emergence in Flocking Boids with Graph Theory

Christian Lambert

Department of Physics and Astronomy

Bachelor of Science

Emergent behavior - behavior exhibited by groups that is not seen in individuals - is a critical part of our world and is difficult to model well. We present a dynamic model where a flock of simulated birds (boids) exists in two dimensions. Each boid has a constant speed and a fixed randomly determined number of neighbors, defined as those boids that influence the direction of its motion (consensus). Modifications of the boids' flight following a specific algorithm (frustration) during the simulation results in emergent behavior. The flock of boids is mapped to a directed graph. Changing the boids' neighbors also modifies the graph. Rigorously defined sub-flocks are identified using graph theory. Using a new method of frustration, α turns, we can enhance the emergent behavior exhibited. Analyzing this emergent behavior is done through order parameters that help us understand how ordered the flock or sub-groups of the flock are. Analyzing the mapping of the graph to the flock can expand our understanding of how and when dynamic emergence occurs in this flocking model. This is done by showing how physical the model is in whether the flock splits like we see real flocks of birds doing. Using α turns and the nearest neighbor consensus method we find we have emergent behavior within a specific range of the model parameters.

Keywords: emergence, boid, graph theory, directed graph, dynamic graph, dynamic edges

Acknowledgments

I’d like to thank my advisor, Dr. Manuel Berrondo for helping me learn about and begin to understand emergence and for his guidance and help in making this possible. I’d also like to thank the physics department for the funding they gave me, enabling me to do this research. I would also thank the quantum information research group for their suggestions and feedback in my work. And a special thank you to my wife for supporting me while writing this thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

This chapter explores what emergence is, what boids are, and why we model them. Then the model and its components are discussed. The chapter ends with an explanation of graph theory and the portion of graph theory relevant to this thesis.

1.1 Emergence

Emergence encompasses coherent behavior exhibited by groups that is not seen in individuals. Emergence is the reason rush hour is so much worse than a few drivers on the road. Emergence is why a flock of birds flies very differently than a single bird does. Representing the world accurately requires understanding emergent behavior.

Because emergent behavior is exhibited all around us, understanding it allows us to have better models of the world. Even people exhibit emergent behavior. From driving in traffic to walking around a crowded shopping center, there is emergent behavior all around us. If a mall is not very crowded, then you will probably just use simple paths between your destinations. However, if it is crowded, there will be a sort of stream of people, and you only go against or cross the flow of people if you must. Having more people around influences your and their behavior, causing that behavior to emerge from the interactions within those nearby. Being able to accurately model this sort of behavior improves self-driving car algorithms, drone swarm flights, and building and pathway designs.

Emergence results as a coherent form of chaotic, non-linear behavior, like a damped, driven, non-linear oscillator. This kind of oscillator does not necessarily exhibit explicit chaotic behavior; only in specific scenarios can this chaotic motion be possible. Similarly, with emergence, the capacity to exhibit emergence does not mean emergence always manifests. This makes modeling emergence difficult because one must create a realistic model, and then vary its parameters to determine if the model can express emergent behavior.

Dynamic phases are common when attempting to model emergence in flocking. A dynamic phase in this context appears when the flock is stuck moving in a periodic way. The dynamic phase occurs if the periodicity of the motion is short compared to the run time of the simulation. Dynamic phases constitute a generalization of phases in statistical mechanics to the time-dependent domain.[1] We can define order parameters which measure what phase the system is in and tell us about the behavior of the group.[2] In particular, I use the order parameters defined in these other works, [3-6] and more is discussed in Section 2.6 about order parameters. Emergent behavior is not periodic, or the period is much larger than the time the system is examined. For example, suppose we are simulating a flock in a constrained area, and is exhibiting emergent behavior. Eventually, as the length of the simulation approaches infinity, the state of the flock, the position and direction of the boids, will eventually repeat itself. This repetition does not mean the boids are in a dynamic phase.

1.2 Boids

Boids are virtual birds, a contraction of birdoid.[7] They were created to model emergence in bird flocking. In this paper boids refers to individuals within the models, and birds refer to actual, physical birds. Boids strip away some of the complexity of fully modeling a bird. The only similarity to birds that boids have is how they move, including the rules they follow when flying in a flock.

We model boids to better understand emergent behavior because it is easy to observationally compare those models to real flocks of birds. This is done by calculating order parameters and by visual inspection of the accompanying animation. Order parameters are a measure of how ordered a system is according to a metric. Multiple order parameters can be used to describe a system, and they need not necessarily agree. Often a combination of order parameters is needed with a visual inspection to determine if the model is emergent. Order parameters also tell us if the emergence is transient, meaning they signal emergence for only a portion of the simulation, and a dynamic phase for the rest of the simulation.[3]

A group in Italy has observed flocks of starlings to determine the specific rules that starlings follow while flying in flocks.[8] We aim to translate rules observed in real birds to improve models of emergence. These rules - observed by Attanasi et al.[1] - are the basis for our model. We also use the rules to judge how realistic and physical the model is. When needed we deviate from the rules to get our model to exhibit emergent behavior. While it is technically feasible to have a model that demonstrates emergent properties without following physical rules, we strive to create emergent models that could be used to for physical situations.

1.3 The Model

1.3.1 Consensus

Consensus refers to the way the boids influence one another's motion in an average fashion. We base our rules for consensus[3-6] on the observational study by Attanasi et al.[1] They found that birds follow each other in a very specific way. Birds have flockmates - other birds they follow. The number of a bird's flockmates remains almost constant throughout the flight of the flock. This is true regardless of the physical distance between the birds. The number of flockmates tends to be small, typically in the range of six to seven. In our case boids that are modeled in this way are called "topological" neighbors.

1.3.2 Frustration

Frustration is intended to provide a counter-balance to consensus. It is a similar idea to that of frustration in solid-state physics. For example, favored configurations of spin-states in a lattice could be frustrated by the geometry of the lattice, making it difficult or impossible for the lattice to stay in the lowest energy configuration.[9] Frustration in our model was created to "frustrate" the boids in following the rules defined for consensus. Its overall effect is to adjust the velocities of the boids in a way that is intended to transform the phase-like behavior induced by the consensus rules to the emergent behavior exhibited by the model.[3]

1.3.3 The Original Model

The original model[3, 5] uses an exact interpretation of the rules for consensus and a simple frustration model. The strict interpretation of the rules for consensus is as follows. Each boid is assigned a fixed number of flockmates at the beginning of the simulation, and they do not change at any point in time. The flockmates are effectively randomly assigned at the beginning of the simulation.[5] The exact method of assignment is detailed in Section 2.1 below. The frustration used is a simple U-turn. When a boid is selected to be frustrated, the direction of its velocity is reversed. The rules for selecting a boid to be frustrated are detailed in Section 2.4.

Using consensus without frustration leads to dynamic phases. The possible dynamic phases are linear, and clockwise and counter-clockwise phase-locked rotations.[3, 5] Topological neighbors were used, and they are discussed further in Section 2.1. Briefly, topological neighbors are the most like what is seen in nature[8] where the neighbors are fixed and are not constrained by distance.

Results of this model exhibit emergence[5] often enough that it is reasonable to conclude that it is an emergent model. To identify emergence we must first determine if the flock is trapped in a dynamic phase by analyzing the order parameters. If it is not in a dynamic phase, then it is most likely exhibiting emergent behavior. Visual analysis of the accompanying animation confirms this. This analysis is qualitative; we look for periodic behavior to determine if the simulation is emergent. One weakness of this model is that it does not always stay emergent. It may exhibit emergent behavior for a certain amount of time, but then get locked into a dynamic phase.[3, 5]

1.4 Graph Theory

A graph is a mathematical construct used to describe relationships between items in a set. They are comprised of nodes and edges.[10] When I refer to a graph, I mean a set of edges and nodes. Plot refers to what is more commonly called a graph. In Figure 1 below, each node is labeled is a circle, and the arrows between nodes are the edges. Nodes are items within the set, and edges are information about the relationship between two nodes. An edge may have a cost associated with traversing it, or going from one node to next, often in the form of a number. This number is termed a weight.[10] Figure 1 does not have any weight labels. An edge may be one-directional, termed a directed edge, or bi-directional, or undirected edge.[10] All the edges used in this thesis are directed. For simplicity I will shorten directed edge to edge from this point on.

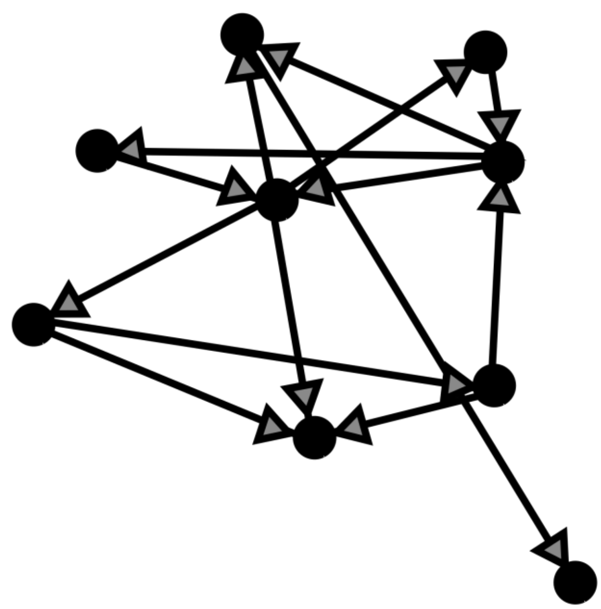

Figure 1 Graph example. Black circles represent nodes, and arrows denote a relationship between nodes. There are directed edges between the nodes. No weights are shown.

Operations can be performed on graphs to give information about the relationships between the graph elements. Operations traverse the edges of a graph following specific algorithms. These operations can give information about the shortest path between two nodes, the cost from going from one node to another, and many other things.[10] We use one algorithm to find self-contained sub-graphs, called strongly connected components. More is said about these in Section 1.5 below, and more can be found in any textbook on graph theory, such as Discrete Mathematics and its Applications.[10]

A flock of boids under our model has a one-to-one mapping to a directed graph. Each boid is represented as a node. The flockmate relationships between the nodes are edges. In the initial model, the edges all have weights of one. There is exactly one edge from a boid to each of its flockmates, and the flockmate does not necessarily have an edge back to the boid. This means the relationship is not reciprocal. For the purposes of consensus, each boid is considered to have an edge of weight one to itself. We do not consider boids beyond one edge away to be flockmates, so the label flockmate is not transitive.

Our graph can be represented as a matrix, the adjacency matrix, where the elements of the matrix are the edges between nodes in the graph. Each element's value is the weight of the edge. This matrix will be defined in detail in Section 2.1 and illustrated in Figure 3. Other methods of storing the graph information are possible, but this is the most compact form and the most convenient for calculations.

1.5 Strongly Connected Components

Strongly connected components are self-contained sub-graphs. A strongly connected component is a set of nodes whose edges allow access to every other node in the component.[10] Strongly connected component is what is meant by self-contained sub-graph. Edges pointing to nodes outside the strongly connected component are not prohibited and are quite common.

Calculating strongly connected components is neither trivial nor intuitive. Knowledge of how this is done and how it is implemented is not necessary for this thesis. Those who wish to know more may consult a textbook on graph theory.[10]

Strongly connected components tell us which parts of the graph have strong relationships. In the context of a flock of boids, a strongly connected component is a set of boids that often behave as a discrete sub-flock. For this thesis, strongly connected component and sub-flock mean the same thing and may be used interchangeably. This also means that if one boid turns, that turn propagates throughout the whole strongly connected component due to consensus. The strength of that influence is highly dependent on the longest path between boids in the strongly connected component. However, we do not analyze the propagation of a turn, as it does not yield any valuable information for studying emergence.

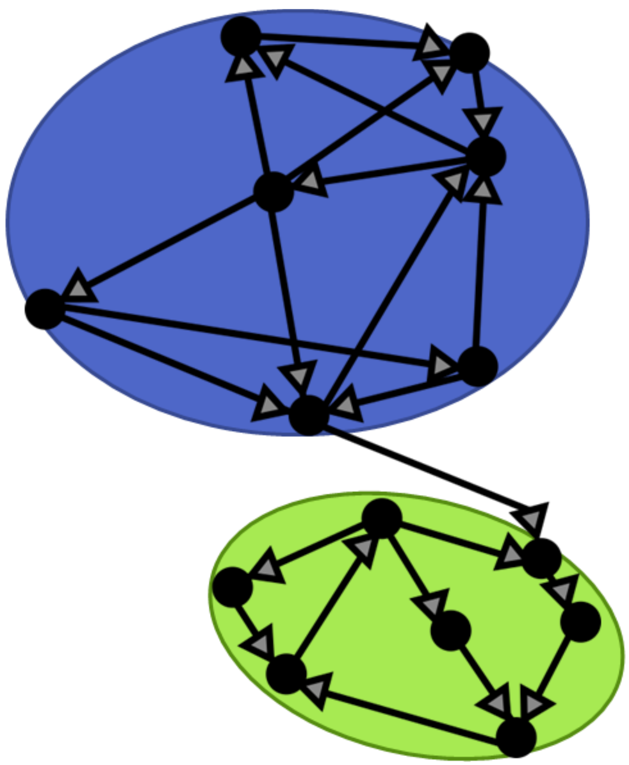

Figure 2 A graph with two strongly connected components identified by the colors surrounding the nodes and edges of each component.

Identifying the strongly connected components of a graph shows us the sub-flocks. In Figure 2, the two groups of nodes highlighted in green and blue are separate strongly connected components. If we pick any node in either group, you can verify that you can reach any other node in that group. Note how you can go from the blue group to the green group, but not back. Because of the unidirectional flow from blue to green the two strongly connected components are still considered separate. If there was even one edge pointing from the green group to the blue they would be one strongly connected component. The edge from blue to green implies that the blue group "follows" the green one in the context of a flock of boids. This means the blue and green groups are both sub-flocks. The implications of the splitting of the flock is discussed further in Section 2.3.

1.6 Prior Work at BYU

There are two theses my work builds off. Wes Kruger[3] examined complex behavior and consensus with a simple frustration term. The frustration he used was a U-turn with a hard boundary. He also introduced the appropriate order parameters. Garett Brown[5] examined consensus with additional methods of frustration. He used U-turns, specular, and a frustration he termed free boids, all with hard and soft boundaries. Both theses worked with topological neighbors.

1.7 Overview

I now give an overview of the following chapters and sections. In Chapter 2 my methods for modeling emergence are discussed. This includes the two consensus models described in Section 2.1, how the consensus changes the graph, and what those changes mean in Sec. 2.2 and 2.3, respectively. Chapter 2 concludes with a depiction of the frustration methods I use in Sections 2.4 and 2.5.

Chapter 3 contains the discussion of results and the conclusions I draw. Section 3.1 discusses the results, and Section 3.2 contains the discussion of the results of consensus and frustration. The conclusions are presented in Section 3.3. Then suggestions for future work are given in Section 3.4.

Chapter 2 Methodology

Our model primarily consists of consensus and frustration. We use two models of consensus, and two of frustration. Graph theory is used with consensus to analyze the structure of the flock, and order parameters are used to analyze the flock's motion and whether it is exhibiting emergent behavior. Each of these is now discussed.

2.1 Consensus

The two models of consensus we use are based on observed behaviors of flocks of birds. [8] Both involve the same basic method of adjusting a boid's velocity, v. This is accomplished by giving the boid the normalized average of its flockmates velocities and its own velocity.[5] The net effect of this is to change each boid's direction. This happens in every step of the simulation for every boid. The calculations for consensus can be seen in Equations (2)-(4). Equation (1) shows the consensus calculations for a flock of n boids each with m neighbors in matrix form with adjacency matrix M. The adjacency matrix is a representation of the flockmate relationships within the flock. Equation (2)[3, 5] shows this calculation in summation form. The consensus calculations are performed every step of the simulation.

$$\begin{bmatrix} v'_{1\,\mathrm{x\,new}} & v'_{1\,\mathrm{y\,new}} \\ v'_{2\,\mathrm{x\,new}} & v'_{2\,\mathrm{y\,new}} \\ \vdots & \vdots \\ v'_{n\,\mathrm{x\,new}} & v'_{n\,\mathrm{x\,new}} \end{bmatrix} = M \begin{bmatrix} v_{1\,x} & v_{1\,y} \\ v_{2\,x} & v_{2\,y} \\ \vdots & \vdots \\ v_{n\,x} & v_{n\,y} \end{bmatrix} \frac{1}{m} \tag{1}$$

$$v'_n(t+1) = \frac{1}{m}\sum_{j=1}^{m} v_j(t) \tag{2}$$

Each velocity is normalized to v_0,[1] the initial velocity magnitude, which for this thesis is 0.15 (arbitrary units) as seen in Equation (3). The constant velocity magnitude is a convenient way to allow for emergence, as it introduces a non-linearity, which is critical for emergent behavior.[3]

$$v_n(t+1) = \frac{v'_n(t+1)}{\|v'_n(t+1)\|} \tag{3}$$

Then the position is updated according Equation (4).[3, 4]

$$r_n(t+1) = r_n(t) + v_n(t+1) \tag{4}$$

Topological neighbors are best understood by looking at the adjacency matrix *M*. Going across a row represents the connection from that boid to its flockmates. For example, in Figure 3 below, if we go across the row labeled B_1, that boid is flockmates with boids B_2-B_5. It is also connected to itself represented in *M* by ones along the diagonal.

	B_1	B_2	B_3	B_4	B_5	⋯	B_{n-3}	B_{n-2}	B_{n-1}	B_n
B_1	1	1	1	1	1	0	⋯	0	0	0
B_2	0	1	1	1	1	1	⋯	⋯	0	0
B_3	0	0	1	1	1	1	⋯	⋯	⋯	0
⋮	⋮	⋱	⋱	⋱	⋱	⋱	⋱	⋱	⋱	⋮
B_{n-1}	1	1	1	0	⋯⋯		0	0	1	1
B_n	1	1	1	1	0	⋯	0	0	0	1

Figure 3 Sample *n* x *n* adjacency matrix for topological neighbors with *n* boids and four flockmates for each boid. A one represents a connection between the boid represented by the row and the boid represented by the column, and a zero represents no connection. In addition, each boid points to itself.

Nearest neighbors are named so because each boid's flockmates are a subset of their physical nearest neighbors. The algorithm we use for determining a boid's flockmates is as follows. First, we find the nearest m neighbors of boid a by calculating the Euclidean distance between them then randomly select n of them to be boid a's flockmates. These n boids remain boid a's flockmates for l steps, where l is a number greater than or equal to one called the recalculation interval. After l steps, the flockmates are recalculated. This repeats for as long as the simulation runs.[11]

Both models of consensus can be represented as an adjacency matrix. Figure 3 shows what an adjacency matrix for a topologically connected flock looks like. An adjacency matrix for nearest neighbors would also have ones along the diagonal, but the other ones representing the boid's flockmates would be found distributed along in the rest of the row.

The adjacency matrix is also able to represent the graph that the flock maps to.

2.2 Manipulating the graph

By manipulating the edges, or flockmate relationships, between boids and how they are assigned, we can influence the behavior of the consensus. The weights of the edges are used in the averaging for consensus. In Figure 3, each edge between the topological neighbors would have a weight corresponding to the one or zero in the matrix, where each row represents the connections a boid has. If an edge has a weight of zero, it is considered to not be present for calculating strongly connected components. This makes sense when considering how the averaging for consensus works. If the weight of a boid's contribution is zero it is effectively not a flockmate.

Dynamic edges can be used instead of a static edge weight to simulate changes within the graph. This is effectively assigning a function to be the weight of an edge. In Figure 4 below, I have plotted a sample of the functions I used as dynamic weights. I used a sine wave, a normalized summation of sine waves of varying frequency, square waves, both with an even period and with an uneven period. Each of these varied with time. Values tended to be positive, but occasionally negative values were allowed. Negative values create a repulsion between boids, while positive values create an attraction. The edge weights are substituted in for the values of the adjacency matrix, and then the consensus calculation turns into a weighted average after also dividing the new velocities by the sum of the weights in each row. This, in effect, turns *M* from a static matrix into a matrix of functions. Other functions used included position-based weight values and time and position dependent weights. These functions would give *M* time and/or position dependence corresponding to the dependence of the function used.

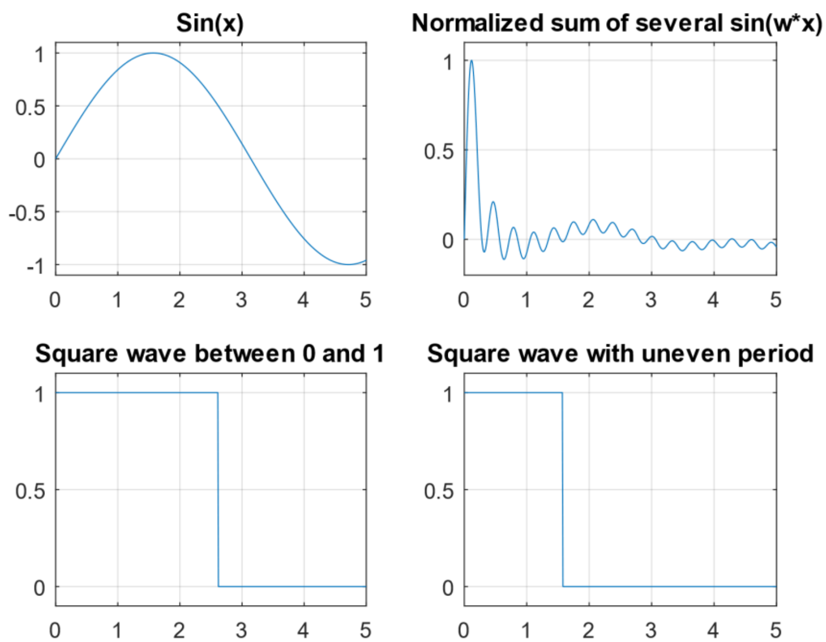

Figure 4 Sample plots of functions used for dynamic edges.

The effectiveness of dynamic edges is discussed in Section 3.1. Allowing edges to vary between zero and non-zero values changes the structure of the graph with the convention that a zero-weighted edge is not currently traversable. This changes the strongly connected components of the graph by effectively removing zero-weighted edges from the graph. Dynamically weighted edges were only used with static neighbors, meaning the neighbors were never recalculated as they are for nearest neighbors. The flockmates were not necessarily topological neighbors, but they were fixed throughout the simulation. The goal of these dynamic edge weights was to simulate the nearest neighbor method without needing to do the nearest neighbor calculations.

2.3 Graphs

Strongly connected components are self-contained sub-graphs, and in the context of flocking boids, tell us what the structure of the flock looks like. This structure is determined by flockmate relationships. There are two types of strongly connected components we observe in flocks of boids: large and trivial. We define a large strongly connected component as one with at least $n + 1$ nodes, or boids as there is a one-to-one mapping between boids and nodes, where n is the number of flockmates used. The reason for this definition is that $n + 1$ boids can exclusively be flockmates with each other. Any fewer and at least one of them would have at least one flockmate external to the group. Therefore $n + 1$ is the smallest possible independent sub-flock. Typically, the strongly connected components have many more boids than $n + 1$. A trivial strongly connected component is a strongly connected component with fewer than $n + 1$ boids. The most common trivial component is with just one boid. In this case no boids have the single boid as a flockmate, so this boid is influenced by, but does not influence, the rest of the flock. When there are two or more non-trivial strongly connected components, the flock has split as discussed in Section 1.5.

There are two ways for the flock to meaningfully split. The two meaningful splits, along with an example of a trivial strongly connected component, are shown in Figure 5 below. For simplicity, not all nodes or edges are shown, and only two strongly connected components are detailed. On the left we find an example of a partial split, because the blue group has a connection to the green group, as shown by the edge highlighted in red. When a change in direction happens in any boid in a strongly connected component, that change propagates throughout the component. It also propagates to any other boids or groups with edges pointing to a member of the component. In this example, any changes in the green group propagate to the blue group. However, any changes in the blue group do not propagate the other way. If the changes in the blue group did propagate, there would be one strongly connected component, not two. In the other example, the two groups are not connected at all. This means the flock has a total split, where the strongly connected components act completely independently of each other. When I refer to splits in this paper I mean multiple large strongly connected components.

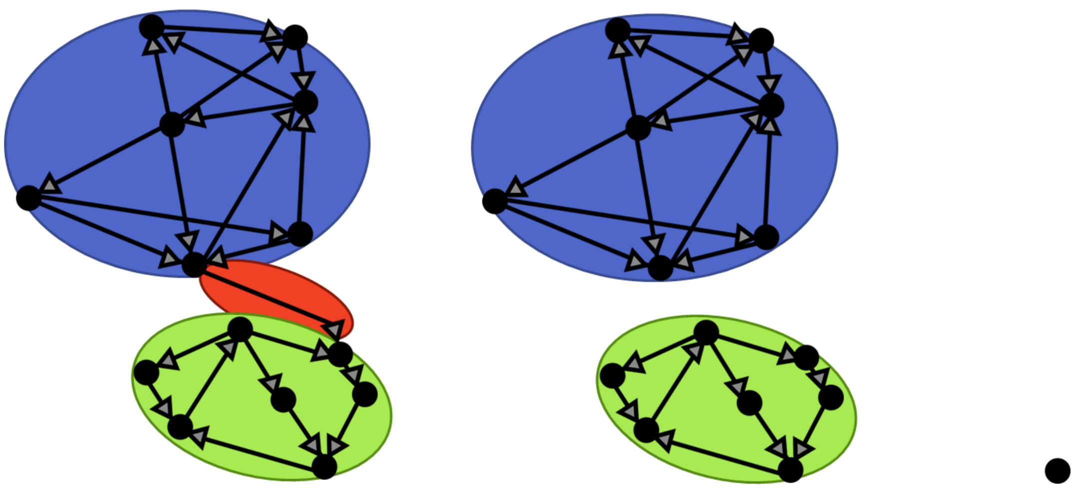

Figure 5 (left) A graph with two large strongly connected components, with the blue component "following" the green one because of the red edge. (center) Two large strongly connected components that are independent. (right) A trivial strongly connected component.

2.4 Frustration

Frustration provides an additional influence that allows the model to exhibit more realistic and emergent behavior. Going back to the non-linear oscillator analogy, frustration corresponds to the damping force. It, along with the normalization of the boids' velocity, provides for the non-linearities necessary for emergence to be possible.[3]

We use a boundary for the flock's motion to induce frustration. We construct a basin that restricts the flock's motion using the equation

$$\left(\frac{r_i}{R}\right)^f > C, 0 \leq C \leq 1, \tag{5}$$

where r_i is the magnitude of boid i's position, R is the radius of the basin, f is a predetermined number, and C is a random number between zero and one. The basin is the space in which the boids are allowed to move. If $C = 1$, then the basin has a steep wall where the boids will only turn when they are at the edge of the basin. When C is between zero and one the basin becomes soft, and the boids have a probability of turning at any point in the basin. I only used a soft basin model. Larger values for f corresponds with stronger frustration and steeper sides of the basin. Figure 6 and Figure 7 show the shapes of the basin for different values of f. Figure 7 shows the radial cross section of the distributions shown in Figure 6.

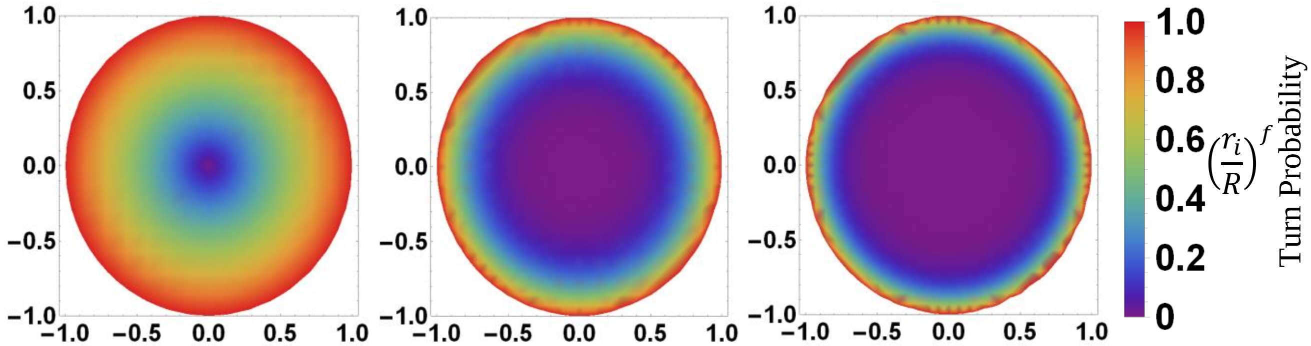

Figure 6 Turn probability examples for different values of *f* in Eq. (5). The colors are the probability of turning, and the axes are scaled to be a fraction of the basin size in arbitrary units. From left to right, *f* is 1, 5, and 10.

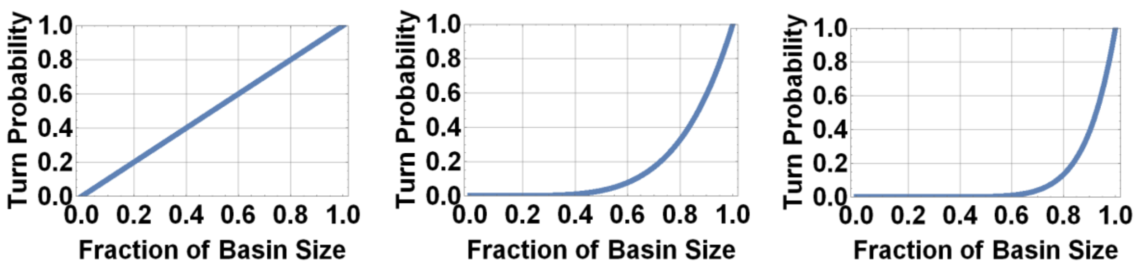

Figure 7 Radial cross section of maps in Figure 6. The horizontal axis is the radial distance as a fraction of the basin size, and the vertical axis is the probability of turning due to frustration. From left to right, *f* is 1, 5, and 10.

A soft basin looks less like the flock is in a bowl and more like natural turning. This probability distribution determines the chosen frustration method (described in the next section), takes effect for the current step of the simulation.

2.5 Frustration methods

Frustration provides for turns in the boids' motion, which can be classified as either smooth turns or sharp turns. A sharp turn takes very few steps to occur, often taking only one step. Smooth turns take many steps to complete. There are many ways to do sharp or smooth turns, but only two of them are discussed here. For more frustration methods, see Garrett Brown's Honors Thesis.[3]

Sharp turns are best seen with the U-turn frustration method. A U-turn is exactly what it sounds like, the boid simply makes an instantaneous U-turn. This is done by reversing the direction of the boid's velocity. This is not entirely physical for real birds, but groups of other animals, like fish, do exhibit such behavior.[3]

Alpha turns are an attempt to make the boids' motion smoother since that is more realistic. The equation for α turns is as follows:

$$\theta_{new} = \phi - \pi + \alpha * \theta_{current} \qquad (6)$$

With θ_{new} being the new direction of the velocity, $\theta_{current}$ being the current velocity direction, ϕ being the angle of the current position of the boid in polar coordinates. The parameter α can be any number. More about α is detailed in Section 3.2.2.

Consensus and frustration help us create an emergent model, and graph theory helps us analyze how physical our model is in one respect, but none of this helps us determine if a model is truly emergent. Analyzing how ordered the flock is after consensus and frustration are performed is done with order parameters and knowing how ordered the flock is compared to the ordering of sub-groups within the flock helps us determine emergence.

2.6 Order Parameters

Order parameters describe how ordered a system is and can tell us how emergent a system is when properly applied. A value close to zero usually indicates a disordered system, and values close to one or minus one indicate a highly ordered system. In

Table 1, the first column indicates the linear order parameter for the whole flock over an interval of eight steps, and the next two columns indicate it for two random sub-groups within the flock.

Note how even though the flock is somewhat disordered, the order parameter is near zero, until the end of the data, group 1 retains order in this slice of the data. Group 2 is disordered until the last couple points. When the flock is disordered and the small groups are ordered, we can conclude that the flock is most likely exhibiting emergence. When the flock and the groups are both highly ordered the flock is likely in a dynamic phase.

Table 1 A sample of sequential alignment order parameter data. The groups are small subsets of the flock. The significant figures are determined by how much Python stores in doing the calculations.

Flock	Group 1	Group 2
0.064268352	0.269784	0.096788
0.087652665	0.219704	0.058267
0.114945515	0.406996	0.049333
0.153834359	0.271618	0.075667
0.227779384	0.354989	0.06029
0.300221863	0.316706	0.162881
0.353839776	0.429595	0.280895
0.429918444	0.507824	0.393018

We use two order parameters, one for alignment and one for rotation. The alignment order parameter tells us how aligned the velocities of a group of boids are. The rotational order parameter tells us how tight the rotations of group of boids are. For more information on these order parameters and how they are calculated, see Garett Brown's Honors Thesis.[3]

To summarize this chapter, our model consists of two items, consensus and frustration. Consensus determines how the flock interacts, and frustration tries to disrupt consensus from ordering the boids too much. Graph theory is used to determine if and how the flock splits, and order parameters help us determine if the flock is emergent or not.

Chapter 3 Results and Conclusion

Understanding emergent behavior is difficult, but with the analysis techniques we have developed we can improve our understanding of emergence. First is a discussion of our results. Then we discuss our models of consensus and frustration. Following that is our conclusions, and then suggestions for future work.

3.1 Results

Figure 8 is a series of screenshots of an animation. The different colors represent different strongly connected components. Included with this thesis are actual animations. For Figure 8, l = 15, m = 6, and f = 3. There are 500 boids. α turns are used, and α = 0.7. Note how the flock varies in the number of strongly connected components over time, including some trivial strongly connected components. What the colors are, or which strongly connected component they are assigned to, do not matter. The splitting is due to the nearest neighbor recalculations.

By changing the flockmates of the boids we change the underlying graph. The graph for topological neighbors does not change at all. For nearest neighbors the graph is rebuilt every time the neighbors are recalculated but is static between recalculations. Both neighbor selection methods assume static edge weights. When the edges are dynamic the weights change, and with the convention that zero weight edges are considered to not exist, this can cause the structure of the graph to change when a weight becomes zero. Weights going to zero change how frequently the flock will split.

Calculating the strongly connected components tell us when the flock has meaningful splits. A number of splits equal to $n/(m+1)$, rounded down, are possible, where n is the number of boids in the flock and m is the number of flockmates a boid has. Typically, we do not see more than two or three splits, and the most frequent number of splits is one, with two strongly connected components.

By inspecting animations for different α values, we find that alpha turns allow for emergent behavior within a specific range of parameters. The parameter α can take many shapes. Initially we used a uniform distribution between zero and one. By further experimentation with uniform, Gaussian, and triangular distributions we discovered that only the mean value of α changed the behavior of the flock, as long as the maximum value was equal to or less than one and the minimum value was equal to or greater than minus one. Following this discovery, we replaced the distribution with a fixed value, and found that the flock behaved the exact same way. We believe the reason for this is that the frustration is triggered so often that the law of large numbers applies, and the α turns effectively use the mean value regardless of the distribution used. This effect allowed us to fix α to a desired value. The range of values, and their bearing on the flock's behavior, is discussed in section 3.2.2.

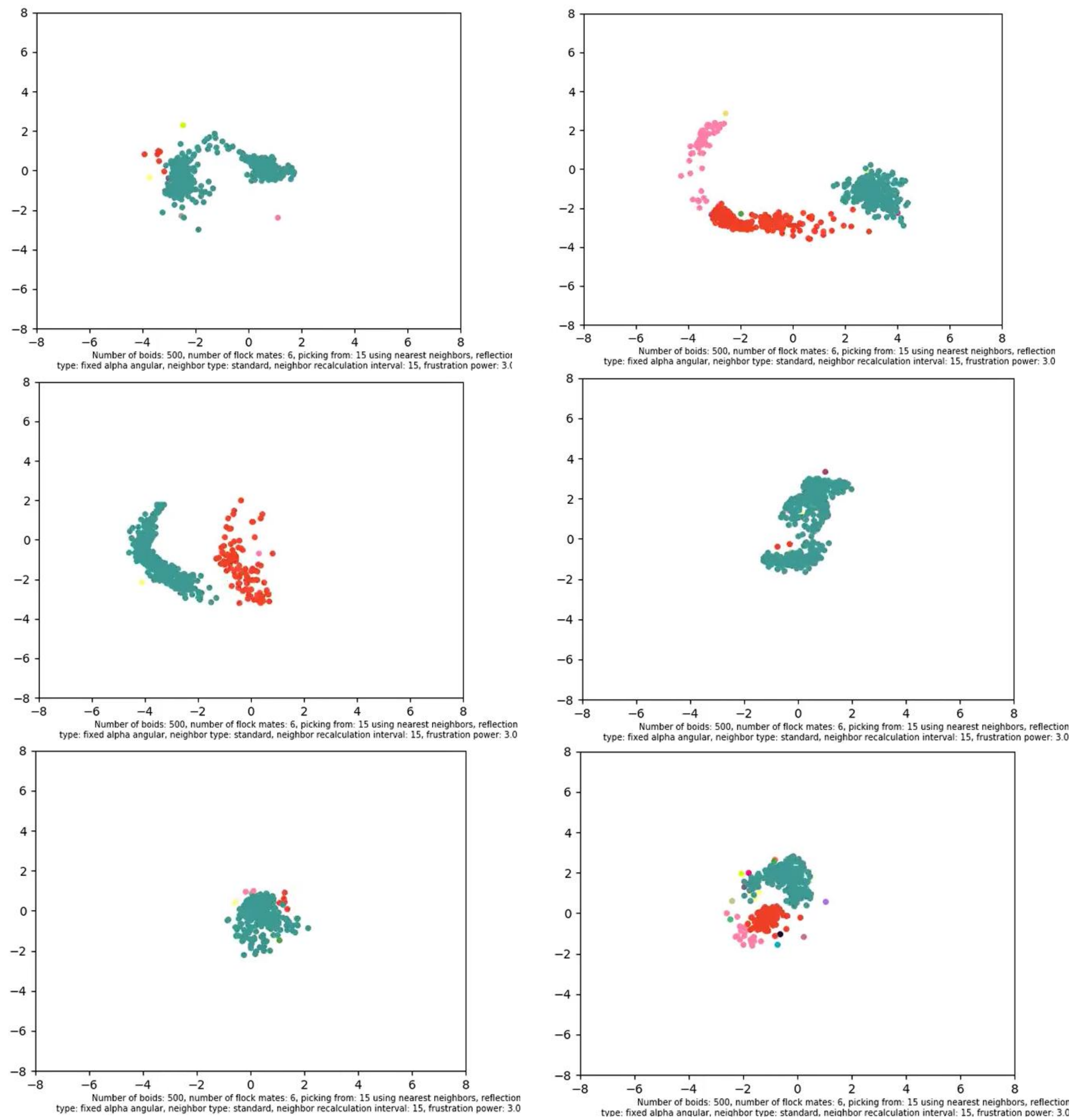

Figure 8 Snapshots of an animation. Each boid is represented by a dot. The different colors indicate different strongly connected components. Boids with their own color are in trivial strongly connected components. Text at the bottom of the snapshots details the parameters used as well as the frustration method. The corresponding parameters are: there are 500 boids, each had six flockmates chosen from 15 of their nearest neighbors. α turns are used with a frustration power of $f = 3$, with $\alpha = 0.7$. The recalculation interval is 15 steps.

Using both the linear and rotational order parameters, we can determine if a flock of boids is exhibiting emergent behavior. The clearest example of emergence is when the flock appears disordered, with order parameters at or near zero, but sub-flocks appear ordered, with order parameters at or near one in the linear and plus or minus one in the rotational case. This lets us define emergence in terms of the order parameters. If the global order parameters show disorder; the parameter is near zero, but the local show order. he parameter is near plus or minus one, then we have emergence. This arises because of the interactions of consensus and frustration. Frustration breaks consensus; it stops global consensus from working, but if we see consensus surviving on local levels, then we know there is emergence.

The alignment order parameter tells us how much a group of boids is going in the same direction. Figure 9 shows the alignment order parameter for three recalculation intervals. In the first interval, with the orange and blue lines, we see that the second strongly connected component is not ordered, but the first is. In the other two intervals we see that the strongly connected components are becoming more ordered.

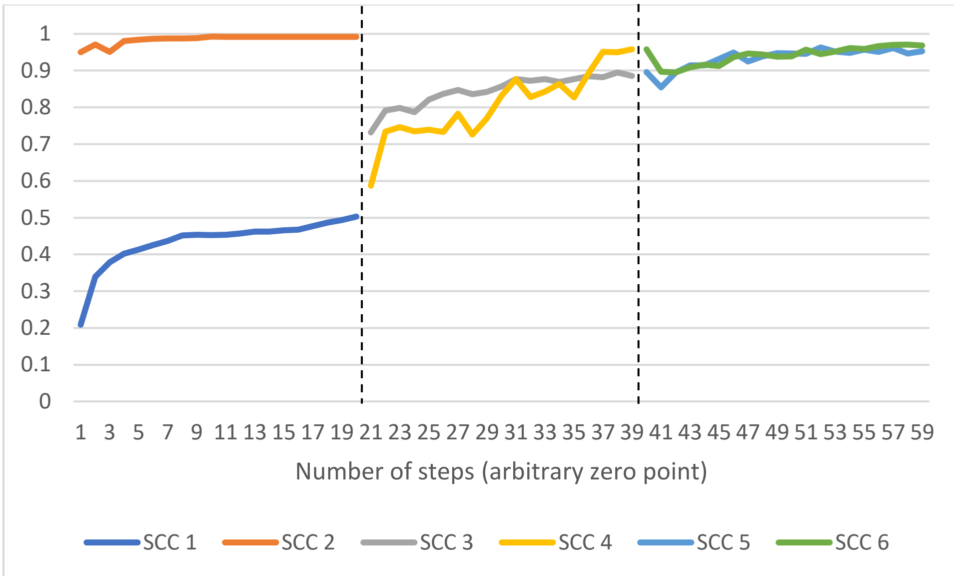

Figure 9 Alignment order parameter plot. Three sequential nearest neighbor selections shown. For each separation, caused by nearest neighbor recalculations, there are two strongly connected components (SCC), shown as different colors. The separating lines are inserted to clarify that the strongly connected components are not necessarily the same from one nearest neighbor recalculation to the next.

The rotational order parameter tells us how sharply a group of boids turns, and in what direction. Clockwise is positive one, counterclockwise is negative one, and not rotating at a specific rate. The rotational order parameter is tuned to a specific turning rate, one rotation in the length of the interval used, based on how many steps are used in calculating it.[3] Figure 10 shows the rotational order parameter for two recalculation intervals, taking in 20 steps. When calculating this order parameter for strongly connected components I did it for the whole recalculation interval to capture the length of the life of the strongly connected component. In general, the rotation order parameter can be between one and minus one. The turns in this flock are so gradual that the rotational order parameter is always small. Therefore, according to the rotational order parameter, the boids in these strongly connected components are not rotating in the same direction and very sharply.

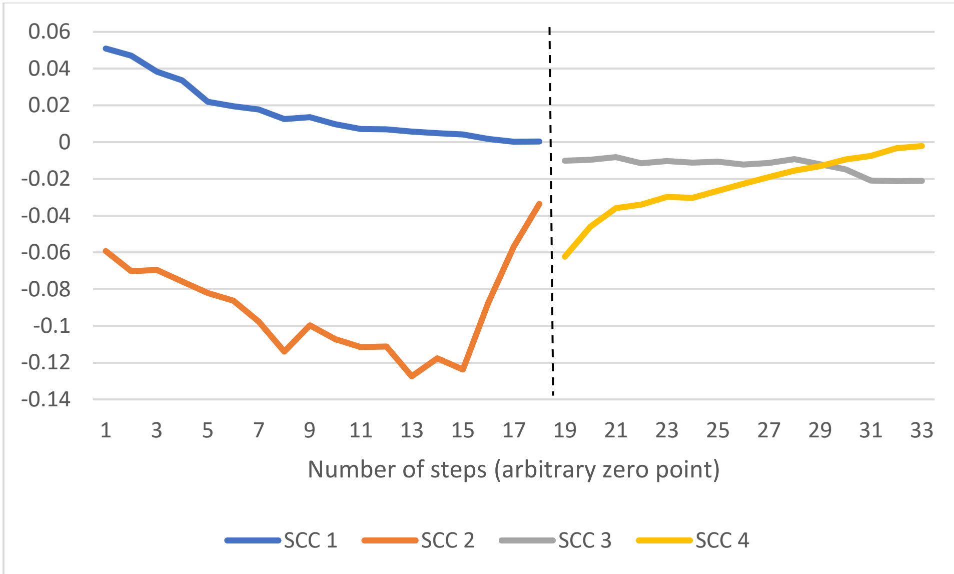

Figure 10 Rotational order parameter plot. Two sequential nearest neighbor selections shown. The separation is made to remind the reader that the strongly connected components are not the same after a nearest neighbor recalculation.

There are a few parameters in our model, but not all of them have a strong bearing on the presence of emergent behavior. By systematically varying parameters and analyzing the accompanying animation and data we have determined which parameters have a meaningful impact on the emergence of the model. Our parameters are as follows but were only tested with nearest neighbors.

The number of boids in the flock does not impact emergence if there are enough for a flock, on the order of hundreds of boids. The lowest number tested was 300 boids. An upper limit of ~2000 boids is a good rule of thumb for a couple reasons. One is the speed of the simulation decreases due to how the nearest neighbor calculations work. We calculate the distance between every boid, so doubling the number of boids quadruples the number of nearest neighbor calculations necessary. Another reason to limit the number of boids is due to how large the graph gets. The way I implemented strongly connected component algorithm does not allow for excessively large and connected graphs due to limitations in the Python programming language.

The number of flockmates did influence whether the flock exhibited emergent behavior. The number of flockmates did not have much of an impact if it was significantly below 10% of the flock's size. If the number of flockmates approached 10% there were marginal effects on the flock's behavior, but not enough to prevent emergent behavior. When the number of flockmates reached 11-12% of the flock size there were clear and significant changes to the flock's overall behavior. It would clump up and get locked in a dynamic phase.

Another parameter I examined was the recalculation interval, l, for finding the nearest neighbors. This was done by determining the incidence of splitting, the percentage of times the flock splits given the chance to do so. The flock has a chance to split when the flockmates are recalculated. If l was short, less than five, there were fewer incidences of splitting. This also increased the time it took for the simulation to run. The incidence of splitting, as seen in Figure 11, peaks about $l = 5$. If l is beyond 5, then the incidence of splitting decreases, roughly by 9%, or 0.09 in Figure 11. for each increase in l by 10. The vertical axis in Figure 11 is the ratio of how many times the flock split over the times the flock did split.

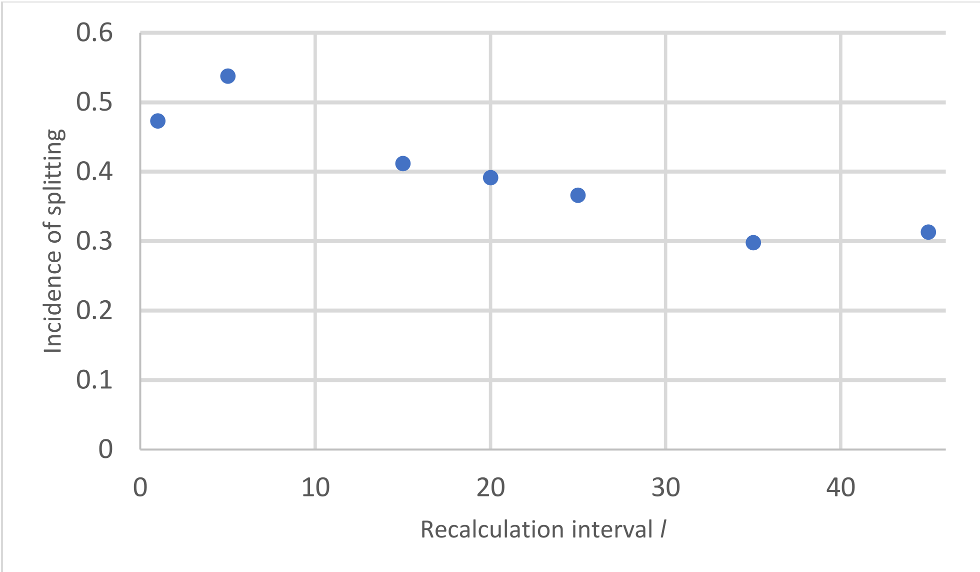

Figure 11 Incidence of splitting versus the recalculation interval. An incidence of one would indicate the flock split every chance it could, and zero would mean it never split.

For each value of *l* in Figure 11 the incidence of splitting varies by 15-20%. I usually had *l* at 10 or 15, as that gave a good balance between speed and incidence of splitting.

3.2 Discussion

3.2.1 Consensus

The topological neighbors model represents a more realistic model for consensus. Our model of topological neighbors is physical.[1**Error! Bookmark not defined.**] However, it is not enough for emergence, as topological neighbors tend towards a dynamic phase, although there may be brief times when they appear emergent.[3] Our results make sense when you consider our consensus algorithm. If all that occurs is an averaging of directions, the flock eventually ends up going in the same direction. The nearest neighbors model is not as physical. Thus, pure consensus simulations were not done with nearest neighbors.

3.2.2 Frustration

Frustration is necessary for emergence. With nearest neighbors, depending on the frustration method, we can see emergence. Alpha turns, with the proper parameter value, did exhibit emergence. More about the alpha parameter is discussed next.

Comparing α turns and U-turns cause a boid to turn tells us that a more realistic turning model also allows for emergent behavior. A sharp turn occurs in one step. A smooth turn, in contrast, occurs over multiple steps. Determining the smoothness of a frustration method is mostly visual. U-turns are inherently sharp and are not very physical in birds. Other groupings of animals, like fish or insects, may exhibit this behavior however. α turns tend to be smoother, leading to a more physical looking animation.

The alpha parameter allows for emergence when it is in a specific range of values: α = 0.35-0.75. This range was determined empirically by varying alpha and observing how that affected the emergent behavior of the boids. Alpha determines the strength of the turn. For convenience, Equation (6) is replicated below with θ being the velocity direction and ϕ being the polar angle of the boid's position.

$$\theta_{new} = \phi - \pi + \alpha * \theta_{current}$$

The general behavior is to take the radial direction of the boid, subtract π, and add the current direction of the velocity multiplied by some scalar for the new velocity direction. The larger α is, the more the boid tends toward its current direction of motion. If α is between 0.35 and 0.75, it pulls the boid toward the center. If it is too low then the boids do not exhibit any emergent behavior. If it is too high, then the boids end up going in roughly the same direction and slowly spiral out of the basin. This happens because the frustration is not strong enough to pull the boids back in. The boids then keep on a course that slowly spirals out.

3.3 Conclusion

Looking at the alpha turn frustration method and its parameter range, we can derive general rules for a frustration method to follow. It should vary with some sort of parameter: for our model that parameter is C from Equation (5), and it also needs to vary based on some parameter of the object you are modelling: in our case it is r_i, the boid's radial distance from the origin. C is a random number from a uniform distribution from zero to one. Our parameter r_i varies between zero and the basin size, typically six, but the basin size is fairly arbitrary.

A boid must be able to reverse its direction within a short number of iterations. Without this feature, a boid tends toward a dynamic phase, and is not be able to break out of that phase. A boid getting trapped in a dynamic phase is due to the averaging effects of consensus in our model. More generally, if an object, like a boid, cannot deviate from its current trajectory sufficiently, it is unable to avoid a dynamic phase.

The frustration method must not be too strong, but it must have a meaningful impact on a boid's motion. In Equation (5), the exponent f determines how strong the frustration is. A large value for f means it is very strong, a small value means it is weak. Because C varies randomly, the point at which the frustration calculations are used varies with each check. The range we use for C is fine, but you could scale it or the basin size and achieve the same effect.

Graphs give us another way to analyze and manipulate flocks of boids. Strongly connected components tell us if and how the flock is split up. Knowing that allows us to better gauge if the model is physical or not, since real flocks of birds split.

Dynamic edges do not perform well. At best they almost approximate the nearest neighbor method. Whether the dynamic edges were functions of time, functions of space, or were simply turned on or off, they consistently performed worse than the static edges. It may be possible to contrive a set of dynamic edges that allow for emergence. However, it is not the most physical way to model bird interactions.

3.4 Suggestions for future work

Other models of consensus and frustration should be developed. Alternatives to or other interpretations of the rules found in Attanasi et al.[1] could be created. Other, more physical frustration methods could also be explored as in this other senior thesis.[3]

While I only used strongly connected components to analyze the flock, there are many other algorithms in graph theory.[10] For example, finding the longest, or shortest, path between any two boids could provide information about how long, or how quickly, it takes a change to propagate through the graph at any step in the simulation.

This is a field where machine learning could prove a useful tool. Machine learning could be used to identify emergent models. This could be done with supervised learning. Training could be done on various aspects of the data and analytics generated by our models. [12-14]

Machine learning could also be used to generate novel models. This could be done with unsupervised learning or by determining rules with which an emergent model must obey and training a machine learning model to generate emergent models. Markov chains could be a component of the underlying network.[12-14] There are many avenues to explore with emergence and machine learning, especially with models that lend themselves to easy numerical analysis.

Appendix

Code

Here is the link to the GitHub repository where this code is stored: https://github.com/Sotaur/Boids. The whole code is also included in the zip file that this thesis can be found in.

Index

Bibliography

[1] A. Czirok and T. Vicsek. Collective behavior of interacting self-propelled particles. Physica A 281 (2000) 17-19.

[2] R. Bowley and M. Sanchez. Introduction to Statistical Mechanics. Clarendon Press, Oxford, 2nd Ed., 2000. Sec. 12.3.

[3] W. Krueger. Holistic Solution Methods for Complex Multi-Particle Systems. 2010. Brigham Young University Senior Thesis.

[4] G. Brown. Beyond Phase Transitions: An Algorithmic Approach to Flocking Behavior. 2017. Brigham Young University Physics Department Senior Thesis.

[5] M. Berrondo and M. Sandoval. Defining Emergence: Learning from Flock Behavior. Complexity 00 2015 DOI 10.1002/cplx.21711

[6] M. Berrondo & W. Krueger, "The Role of Consensus and Frustration in the Emergence of Flocking Behavior", Journal UASAL, Centennial Edition, 85, 219-234 (2009)

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

[8] Attanasi A et al. 2015 Emergence of collective changes in travel direction of starling flocks from individual birds' fluctuations. J. R. Soc. Interface 12:20150319. http://dx.doi.org/10.1098/rsif.2015.0319.

[9] K. Binder & A.P. Young, "Spin glasses: Experimental Facts, Theoretical Concepts, and Open Questions", Rev. Mod. Phys. 58, 801–976 (1986)

[10] K. Rosen. Discrete Mathematics and its Applications with Combinatorics and Graph Theory. McGraw Hill Higher Education, Global edition of 7th revised edition, July 1, 2012. Sec 8.

[11] M. Berrondo, G. Brown & M. Sandoval, "An Algorithmic Approach to Flocking Behavior: Reaching beyond Global Phases", to be published

[12] C. Bishop. Pattern Recognition and Machine Learning. Springer, 2013.

[13] I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016. http://www.deeplearningbook.org/. There are other online eBooks freely available on this topic.

[14] I recommend these free online courses: https://www.kadenze.com/courses/machine-learning-for-musicians-and-artists-v and https://www.kadenze.com/courses/creative-applications-of-deep-learning-with-tensorflow-iv. Other online courses should also be suitable.