Study of a 2-Dimensional Aggregation Model with Non-Linear Adaptations

by

Eryn Frawley

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Science
in the
Faculty of Science
Modelling and Computational Science

University of Ontario Institute of Technology

August 2018

Abstract

Study of a 2-Dimensional Aggregation Model with Non-Linear Adaptations

Eryn Frawley
Faculty of *Science* (*Modelling and Computational Science*)
University of Ontario Institute of Technology
2018

In this paper we will explore the impacts on collective animal motion of adding non-linearity to the animals' turning rate in a 2-dimensional model of aggregation. The model in question was proposed by Fetecau [6] and implements a dependence on an individual's relation to neighbours' position and the direction in which they are moving. By utilizing a turning rate formula similar to one proposed by Eftimie [4], we can retain many of the dependences of the original 2-dimensional model while ensuring that the turning rate is both bounded and non-linear. The group formations resulting from this introduction of non-linearity are explored.

Author's Declaration

I declare that this work was carried out in accordance with the regulations of the *University of Ontario Institute of Technology*. The work is original except where indicated by special reference in the text and no part of this document has been submitted for any other degree. Any views expressed in the dissertation are those of the author and in no way represent those of the *University of Ontario Institute of Technology*. This document has not been presented to any other University for examination either in Canada or overseas.

Eryn Frawley

Date: August 9, 2018

Dedication

I would like first and foremost to dedicate this paper to my supervisors, Dr. Lennaert van Veen and Dr. Pietro-Luciano Buono. Your support through the last two years and our entire academic relationship means the world to me. Thank you for selflessly giving of your time and energy to help make this thesis all it could be.

I wish to extend a special thanks to the Ontario Graduate Scholarship whose funding supported me greatly in my first year of research.

I would like to thank the University of Ontario Institute of Technology for the last six years of my academic career. The opportunities and experiences that I have received from this institution have changed my life and I am extremely grateful. The teachers, colleagues, and friends I have met in your halls have given me more joy and comfort than I can possibly put into words.

Finally I would like to dedicate this thesis to my friends and family across Canada who have stood by me through every up and down, especially my parents, Beth and Paul, and my brother, Matthew. Without your encouragement, I could never have come as far as I have, in not only this academic pursuit, but in every success I have had. Know that the greatest parts of me come from you.

Contents

List of Figures

CHAPTER 1

Introduction

Most animals in nature move in groups for safety. These groups offer the promise of survival, as they help animals mate, find food sources, and avoid predators. Being able to predict animal movement and other types of aggregation can aid with preventative action. For instance, if we can predict the collective movement of locusts [12], plans can be made in accordance with the destruction they cause.

Animal aggregation has been studied extensively using many different methods. There are so many papers written on the subject, rather than a literature review, we have chosen to include surveys on the topic to give a brief overview of the different ways collective motion has been explored.

The simplest form of collective motion study comes from individual based models [9]. The models in this survey study the collective movement of individual cells, each of which experience attraction, repulsion, or a combination of the two which dictates their movement.

The same method of individual based collective motion can be applied to animals [13]. It is stressed in the basic models found in this survey that an animal

will move away from a neighbour in order to avoid over crowding, will move towards a neighbour in order to form a group, or will assimilate directionally with his neighbours in order to align. The survey suggests that all movement of the animal is dependent on the animal's desire to adhere to any or all of these steering behaviours.

The literature discussing individual based collective motion is extensive and we suggest further exploration of these survey papers, but there is much diversity in aggregation, and the movement of animal groups can also be modelled using partial differential equations. These equations are not looking at individuals, but rather the density of individuals located at a position on the domain. Again, the literature is expansive, and we suggest [3] for reader interest. These partial differential equation models depend on similar steering behaviours, simplifying animals' movements to "towards a neighbour", "away from a neighbour", or "in the same direction as a neighbour".

One way to model these behaviours and the resulting interactions between different members of the group and is through radii [4].

Assume that the circles in Figure 1.1 are around a particular density. Based on which concentric circle a neighbour falls in, the animals will be attracted, be repelled, or align with the neighbour. However, there is very rarely only one neighbour in a group, so the individuals located at the centre of the circles will receive information from all neighbours, which is weighed mathematically and instructs the individual on how to move.

The 1-dimensional partial differential equation model studied by Eftimie et al. [4] simplifies an animal's movements to either "left" or "right" on a 1-dimensional line.

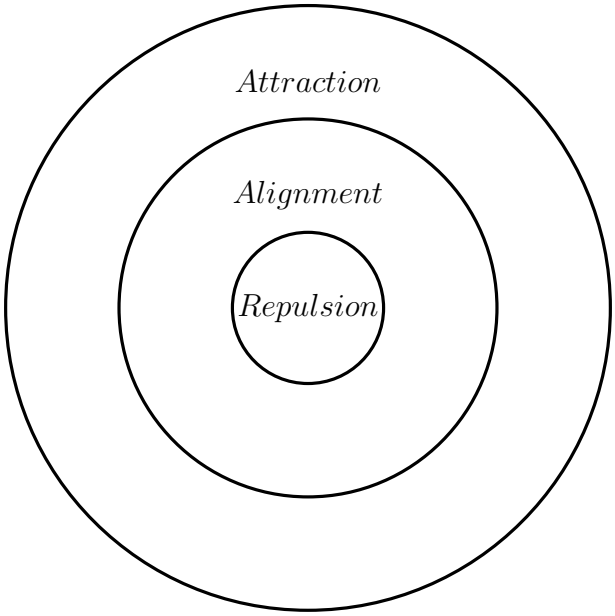


Figure 1.1: The concentric circles indicate the regions at which an individual will be attracted, repelled, or inclined to align with a neighbour.

The equations are

$$\begin{aligned} \partial_t u^+(x,t) + \partial_x(\gamma u^+(x,t)) &= -\lambda^+ u^+(x,t) + \lambda^- u^-(x,t), \\ \partial_t u^-(x,t) - \partial_x(\gamma u^-(x,t)) &= \lambda^+ u^+(x,t) - \lambda^- u^-(x,t), \\ u^\pm(x,0) &= u_0^\pm(x), x \in \mathbb{R} \end{aligned} \tag{1.1}$$

where $u^+(x,t)$ and $u^-(x,t)$ are the density of animals moving right and left at position x and time t respectively, γ is the constant speed at which animals move and λ^+ and λ^- are the turning rate of the individual, either right turning left or left turning right. In this model, $\lambda^\pm$ is a non-linear function, modelled by

$$\lambda^\pm(y^\pm) = \lambda_1 + \lambda_2 f(y^\pm[u^+, u^-]), \tag{1.2}$$

where f is bounded and increasing to represent the saturation rate of information. In her study, Eftimie used some function of the hyperbolic tangent function.

$y^{\pm}[u^+, u^-]$ contains all the information about neighbours' positions and directions

$$y^{\pm}[u^+, u^-] = y_r^{\pm}[u^+, u^-] - y_a^{\pm}[u^+, u^-] + y_{al}^{\pm}[u^+, u^-]. \tag{1.3}$$

y_r, y_a, and y_{al} represent the terms that influence an individual's turn based on repulsion, attraction, and alignment respectively, determined by the information that an individual receives from the right or left (p_r and p_l respectively) and the kernels K_j, for $j = a, r, al$, which determine how much influence each piece of information has. Expanded, these terms are

$$\begin{aligned} y_{r,a}^{+}[u^+, u^-] &= q_{r,a} \int_0^{\infty} K_{r,a}(s)(p_r u(x+s) - p_l u(x-s))\,\mathrm{d}s, \\ y_{r,a}^{-}[u^+, u^-] &= q_{r,a} \int_0^{\infty} K_{r,a}(s)(p_l u(x-s) - p_r u(x+s))\,\mathrm{d}s, \end{aligned} \tag{1.4}$$

and

$$\begin{aligned} y_{al}^{+}[u^+, u^-] &= q_{al} \int_0^{\infty} K_{al}(s)(p_r u^-(x+s) - p_l u^+(x-s))\,\mathrm{d}s, \\ y_{al}^{-}[u^+, u^-] &= q_{al} \int_0^{\infty} K_{al}(s)(p_l u^+(x-s) - p_r u^-(x+s))\,\mathrm{d}s. \end{aligned} \tag{1.5}$$

In these expressions, x is the individual's position and s is the distance between the individual and the neighbour. The critical values of s for repulsion, alignment, and attraction come from the "zones" in Figure 1.1. Each s_j is the radius of the circle corresponding to the region j. For instance, if s is greater than s_r, but less than s_{al}, s is in the alignment zone.

While Eftimie's paper and those discussed in the aforementioned surveys provide useful information on a variety of patterns that develop within collective motion, some types of aggregation can not effectively be modelled in 1-dimension. For instance, if we were to study most animals moving in a group from an aerial view, the model of their motion would be in 2-dimensions.

In this thesis, we will be exploring a 2-dimensional model of animal aggrega-

tion, and adapting it in order to implement some of the same steps explored as in Eftimie's model. We base our analysis on Fetecau's 2-dimensional aggregation model [6], which does not lend itself to non-linearity in λ, an addition important to the demonstration of saturation. Because of this, we will adapt the model in order to implement this non-linearity. These adaptations could, however, lead to variations between our results and Fetecau's, as the previous paper exploring the un-adapted model puts a great deal of importance on the part of the model that we will be changing.

We will then attempt to solve the newly adapted equation with a combination of exponential time differencing and time-stepping, resulting in visual simulations of the solution results.

CHAPTER 2

2-Dimensional Aggregation Model

2.1 Fetecau's Model

Fetecau extended the concepts proposed in 1-dimensional aggregation to 2-dimensions [6]. The model defined a density's spatial position by $X = (x, y)$ and extended the direction of motion of the density from left or right on a 1-dimensional line to an angle ϕ' which can range from $-\pi$ to π radians. Consequently, each neighbour to the individual will also have a 2-dimensional position ($S = (s_x, s_y)$) and direction (θ). The equation appearing in Fetecau's paper is

$$\partial_t u + \gamma e_\phi \cdot \nabla_X u = -\lambda(X, \phi)u + \int_{-\pi}^{\pi} T(X, \phi', \phi)u(X, \phi', t)\,\mathrm{d}\phi' \tag{2.1.1}$$

where u is the density of individuals at a specific position and moving in a specific direction and γ is the constant speed that animals in the system move at. The system operates with periodic boundary conditions in both space and angular direction. The most interesting variables in the system, however, are λ and T. $\lambda(X, \phi)$ is the

rate at which an individual located at position X and moving in direction ϕ is able to change directions. $T(X, \phi', \phi)$ is the rate at which an individual located at position X moving in direction ϕ' can change directions from ϕ' to ϕ.

Both T and λ contain three functions, each which provide the information with respect to attraction, alignment, or repulsion respectively. That is,

$$\begin{aligned} \lambda(X, \phi) &= \lambda_{al}(X, \phi) + \lambda_a(X, \phi) + \lambda_r(X, \phi), \\ T(X, \phi, \phi') &= T_a(X, \phi, \phi') + T_r(X, \phi, \phi') + T_{al}(X, \phi, \phi'), \end{aligned} \tag{2.1.2}$$

where

$$\begin{aligned} \lambda_{al}(X, \phi) &= q_{al} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_{al}^d(X - S) K_{al}^o(\theta; \phi) u(S, \theta, t) \, \mathrm{d}\theta \, \mathrm{d}S, \\ \lambda_j(X, \phi) &= q_j \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_j^d(X - S) K_j^o(S; X, \phi) u(S, \theta, t) \, \mathrm{d}\theta \, \mathrm{d}S, \end{aligned} \tag{2.1.3}$$

where $j \in \{a, r\}$, and

$$\begin{aligned} T_{al}(X, \phi, \phi') &= q_{al} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_{al}^d(X - S) K_{al}^o(\theta; \phi) w_{al}(\phi - \phi', \phi' - \theta) u(S, \theta, t) \, \mathrm{d}\theta \, \mathrm{d}S \\ T_j(X, \phi, \phi') &= q_j \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_j^d(X - S) K_j^o(S; X, \phi) w_j(\phi - \phi', \phi' - \psi) u(S, \theta, t) \, \mathrm{d}\theta \, \mathrm{d}S. \end{aligned} \tag{2.1.4}$$

The functions K_j^d and K_j^o represent the distance and orientation kernels respectively. The variable θ is the direction that a neighbour is moving in, similar to the individual's ϕ. The variable ψ appears in the kernels for attraction and repulsion. Rather than depending on the direction a neighbour is moving in, ψ depends on the neighbour's position in relation to the individual in question. Assume that the neighbour's position is $S = (s_x, s_y)$. In order to obtain ψ, we take the vector $S - X$ to be (u, v). The angle that this vector makes with the x-axis gives ψ as

$$\cos \psi = \frac{u}{\sqrt{u^2 + v^2}}, \quad \sin \psi = \frac{v}{\sqrt{u^2 + v^2}}.$$

What differentiates the double integral in the function T from the function λ is the presence of w. This function is an approximation of a Gaussian distribution and depends on two variables. The first is the difference between the direction an individual wants to turn into (namely, ϕ') and the direction it is currently moving in. The second is the difference between ϕ' and the direction the neighbour is moving in, θ, or the angular variable ψ. As is, the model cannot use a non-linear function for λ similar to the method used by Eftimie, because for the turning rate T_j, the density is multiplied by the probability function w_j as well as the two kernels within the double integrals.

As T dictates the rate at which an individual can turn from one direction into another specified direction, we derive λ, the rate at which an individual can turn from one direction into an unspecified direction, by integrating T over all possible direction angles, so that

$$\lambda(X, \phi) = \int_{-\pi}^{\pi} T(X, \phi, \phi') \, \mathrm{d}\phi'. \tag{2.1.5}$$

2.2 Adaptations to Implement Non-Linearity

If possible, we want to include a non-linear λ in the system for two reasons. First, adding non-linearity complicates the model; therefore, it has the potential for more interesting and complex solutions. Second, we want to demonstrate saturation in the model. Eventually, the number of neighbours influencing an individual to turn in a certain direction will become large and any addition to that number will trivially affect the individual's decision to move. In the 1-dimensional study of aggregation, Eftime [4] used the hyperbolic tangent function to demonstrate this saturation.

In an attempt to adapt Fetecau's model to use non-linear functions, we first

change the dependence of the probability function w. If we eliminate the second variable in w and have the probability function depend solely on the difference between the direction an individual wants to turn into and the direction it is currently moving in, then it will no longer have any dependence on θ or S. Therefore, in this approximation, w can be removed from the double integral of T with respect to the alignment, attraction, and repulsion case. Then,

$$\begin{aligned} T_{al}(X,\phi,\phi') &= q_{al} w_{al}(\phi-\phi') \int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K_{al}^{d}(X-S) K_{al}^{o}(\theta;\phi) u(S,\theta,t)\,\mathrm{d}\theta\,\mathrm{d}S \\ &= w_{al}(\phi-\phi')\lambda_{al}(X,\phi), \\ T_{a,r}(X,\phi,\phi') &= q_{a,r} w_{a,r}(\phi-\phi') \int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K_{a,r}^{d}(X-S) K_{a,r}^{o}(S;X,\phi) u(S,\theta,t)\,\mathrm{d}\theta\,\mathrm{d}S \\ &= w_{a,r}(\phi-\phi')\lambda_{a,r}(X,\phi). \end{aligned} \tag{2.2.1}$$

Considering this new adaptation, if we now examine (2.1.5), Fetecau's relationship simplifies to

$$\lambda(X,\phi) = \int_{-\pi}^{\pi} T(X,\phi,\phi')\,\mathrm{d}\phi' = \int_{-\pi}^{\pi} \lambda(X,\phi) w(\phi-\phi')\,\mathrm{d}\phi'. \tag{2.2.2}$$

λ can be removed from the integral on the right hand side of (2.2.2) as it has no dependence on ϕ' and the resulting integral $\int_{-\pi}^{\pi} w_j(\phi-\phi')d\phi' = 1$, as w is a probability function. Therefore, the equality holds and we can introduce non-linear functions into λ.

Using a method similar to Eftimie [4], we set

$$\lambda(X,\phi) = g_1 + g_2 f(\Gamma - 2), \tag{2.2.3}$$

where g_1 and g_2 are constants that weight the turning rate function. The constant g_1 is the baseline turning rate and the constant g_2 is the bias within the turning rate,

multiplied by the non-linear function of the neighbours' influence. Subtracting 2 from Γ insures that for $\Gamma = 0$ the turning rate is small. f can be any non-linear function that satisfies the systems requirements for saturation.For instance

$$f(\Gamma) = \frac{1}{2} + \frac{1}{2}\tanh(\Gamma - 2), \tag{2.2.4}$$

where Γ contains all information about the individuals neighbours

$$\Gamma = \sum_{j=a,r,al} q_j \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_j^d K_j^o u(S, \theta, t)\, \mathrm{d}\theta\, \mathrm{d}S. \tag{2.2.5}$$

Γ integrates over all possible neighbours, so that the information is weighted accurately regarding the turning rate of the animal. Consequently, Fetecau's system (2.1.1) becomes

$$\begin{aligned} \partial_t u + \gamma e_\phi \cdot \nabla_X u = &-\left(g_1 + \frac{1}{2}g_2 + \frac{1}{2}g_2 \tanh(\Gamma - 2)\right) u \\ &+ \int_{-\pi}^{\pi} w(\phi - \phi') \left[g_1 + \frac{1}{2}g_2 + \frac{1}{2}g_2 \tanh(\Gamma - 2)\right] u(X, \phi', t)\, \mathrm{d}\phi'. \end{aligned} \tag{2.2.6}$$

2.3 Symmetry of the System

One of the benefits of this system is that it is symmetry invariant with respect to translation, rotation, and reflection. The presence of symmetry indicates that the animal aggregation only depends on the difference between position and direction of the moving animals and the location in space where aggregation is occurring is irrelevant. In order to accept the symmetries in the system, we can prove them for each of the three types of symmetry. The collection of these symmetries generates the Euclidean group E(2).

2.3.1 Translation

To show the translational symmetry, let τ be the translational operator acting on density u and define it as $(\tau u)(X, \phi, t) = u(X - \tau, \phi, t)$, where $\tau = (\tau_x, \tau_y)$.

To show that equation (2.1.1) is translation invariant, we apply τ to all functions u in the system

$$\partial_t(\tau u) + \gamma e_\phi \cdot \nabla_X(\tau u) = -\lambda(X, \phi)(\tau u) + \int_{-\pi}^{\pi} T(X, \phi', \phi)(\tau u)(X, \phi', t)\,\mathrm{d}\phi'.$$

With respect to the first term, (τu) does not change the variable t, and therefore $\partial_t(\tau u) = \tau \partial_t u$. The derivative of $X - \tau$ with respect to X is the same as the derivative of X with respect to X, and γe_ϕ has no spatial component, so it is τ invariant as well. Therefore, we can write the entire term as $\tau(\gamma e_\phi \cdot \nabla_X u)$. Because both λ and T depend on u, when looking at the last two terms, we must first look at the impact of τ on λ and T. Recall that $\lambda = \lambda_{al} + \lambda_a + \lambda_r$, so $-\lambda(\tau u) = -\lambda_{al}(\tau u) - \lambda_a(\tau u) - \lambda_r(\tau u)$.

With respect to attraction and repulsion

$$\begin{aligned}(\tau\lambda_{a,r})(X, \phi) &= q_{a,r} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K^d_{a,r}(X - S) K^o_{a,r}(S - X; \phi) \tau u(S, \theta, t)\,\mathrm{d}\theta\,\mathrm{d}S \\ &= q_{a,r} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K^d_{a,r}(X - S) K^o_{a,r}(S - x; \phi) u(S - \tau, \theta, t)\,\mathrm{d}\theta\,\mathrm{d}S,\end{aligned}$$

and with respect to alignment

$$\begin{aligned}(\tau\lambda_{al})(X, \phi) &= q_{al} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K^d_{al}(X - S) K^o_{al}(\phi - \theta) \tau u(S, \theta, t)\,\mathrm{d}\theta\,\mathrm{d}S \\ &= q_{al} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K^d_{al}(X - S) K^o_{al}(\phi - \theta) u(S - \tau, \theta, t)\,\mathrm{d}\theta\,\mathrm{d}S.\end{aligned}$$

In all three cases, if we let $Z = S - \tau \Rightarrow S = Z + \tau$, then we get

$$\begin{aligned}(\tau\lambda_{a,r})(X,\phi) &= q_{a,r}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{a,r}(X - Z - \tau)K^o_{a,r}(Z + \tau - X;\phi)u(Z,\theta,t)\,\mathrm{d}Z\,\mathrm{d}S \\ &= q_{a,r}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{a,r}((X-\tau) - Z)K^o_{a,r}(Z - (X-\tau);\phi)u(Z,\theta,t)\,\mathrm{d}Z\,\mathrm{d}S \\ &= \lambda_{a,r}(X - \tau,\phi),\end{aligned}$$

and

$$\begin{aligned}(\tau\lambda_{al})(X,\phi) &= q_{al}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{al}(X - Z - \tau)K^o_{al}(\phi - \theta)u(Z,\theta,t)\,\mathrm{d}Z\,\mathrm{d}S \\ &= q_{al}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{al}((X-\tau) - Z)K^o_{al}(\phi - \theta)u(Z,\theta,t)\,\mathrm{d}Z\,\mathrm{d}S \\ &= \lambda_{al}(X - \tau,\phi).\end{aligned}$$

Note, we can change the integral to integrate over Z instead of S because S, and consequently Z, is on an infinite domain. The entire term for attraction, repulsion, and alignment becomes

$$-\lambda_j(X,\phi)(\tau u) = -\lambda_j(X - \tau,\phi)u(X - \tau,\phi,t) = \tau[-\lambda_j(X,\phi)u]$$

for $j \in \{a, r, al\}$, which means that we can remove τ from the entire summation that makes up the third term.

The function T is also the sum of an attraction, repulsion, and alignment component, where T_{al}, T_a, and T_r are identical to their λ_{al}, λ_a, and λ_r counterparts, save the addition of the probability function $w(\phi' - \phi)$. Because the probability function doesn't depend on space, it will be τ invariant, and so the same proof as for λ's invariance with respect to τ also proves that T is invariant with respect to τ. Therefore, the last term will also be invariant with respect to τ.

Because all terms are individually invariant with respect to τ, the entire equation

is, in fact, invariant with respect to translation.

2.3.2 Rotation

To show the rotational symmetry, let α be the rotational operator acting on density u and define it as $\alpha u(X, \phi, t) = u(R_{-\alpha}X, \phi - \alpha, t)$, where

$$R_{-\alpha}X = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} x^*\cos\alpha + y^*\sin\alpha \\ -x^*\sin\alpha + y^*\cos\alpha \end{pmatrix}.$$

To show that equation (2.1.1) is rotation invariant, we apply α to all functions u in the system

$$\partial_t(\alpha u) + \gamma e_\phi \cdot \nabla_X(\alpha u) = -\lambda(X, \phi)(\alpha u) + \int_{-\pi}^{\pi} T(X, \phi', \phi)(\alpha u)(X, \phi', t)\,\mathrm{d}\phi'.$$

With respect to the first term, (αu) does not change the variable t, and therefore $\partial_t(\alpha u) = \alpha\partial_t u$.

The second term expands to

$$\gamma[x'\cos\phi + y'\sin\phi]R_\alpha\nabla_X u(X, \phi, t),$$

where $x' = \frac{\partial x}{\partial x*} + \frac{\partial x}{\partial y*}$ and $y' = \frac{\partial y}{\partial x*} + \frac{\partial y}{\partial y*}$, or $x' = \cos\alpha + \sin\alpha$ and $y' = -\sin\alpha + \cos\alpha$. Because we use the chain rule on $\nabla_X u(R_{-\alpha}X, \phi, t)$, we are allowed to move R_α to the other side of the gradient, so that

$$\begin{aligned}\gamma[x'\cos\phi + y'\sin\phi]R_\alpha\nabla_X u(X, \phi, t) &= \gamma[(\cos\alpha + \sin\alpha)(\cos\phi) + \cdots \\ &\quad (-\sin\alpha + \cos\alpha)(\sin\phi)]R_\alpha\nabla_X u(X, \phi, t) \\ &= \gamma[(\cos\alpha)(\cos\phi) + (\sin\alpha)(\cos\phi) + \cdots \\ &\quad (-\sin\alpha)(\sin\phi) + (\cos\alpha)(\sin\phi)]R_\alpha\nabla_X u(X, \phi, t).\end{aligned}$$

Expanding this expression using the fact that $\cos\phi\cos\alpha = \cos(\phi+\alpha) + \sin\alpha\sin\phi$, and $\sin\phi\cos\alpha = \sin(\phi+\alpha) - \cos\phi\sin\alpha$; therefore, we obtain

$$\begin{aligned}
&\gamma[\cos(\phi+\alpha) + \sin\alpha\sin\phi + \sin\alpha\cos\phi - \cdots \\
&\qquad \sin\alpha\sin\phi + \sin(\phi+\alpha) - \cos\phi\sin\alpha]R_\alpha\nabla_X u(X,\phi,t) \\
&= \gamma[\cos(\phi+\alpha) + \sin(\phi+\alpha)]R_\alpha\nabla_X u(X,\phi,t).
\end{aligned}$$

Recall that R_α changes ϕ to $\phi+\alpha$; therefore,

$$\gamma[\cos(\phi+\alpha) + \sin(\phi+\alpha)]R_\alpha\nabla_X u(X,\phi,t) = \gamma[R_\alpha\cos\phi + R_\alpha\sin\phi]R_\alpha\nabla_X u(X,\phi,t).$$

The entire term can now be rewritten as

$$\begin{aligned}
&R_\alpha\gamma[\cos\phi + \sin\phi]\nabla_X u(X,\phi,t) \\
&= \alpha[\gamma e_\phi \cdot \nabla_X u(X,\phi,t)].
\end{aligned}$$

Because both λ and T depend on u, when looking at the last two terms, we must first look at the impact of α on λ and T. Recall that $\lambda = \lambda_{al} + \lambda_a + \lambda_r$, so $-\lambda(\alpha u) = -\lambda_{al}(\alpha u) - \lambda_a(\alpha u) - \lambda_r(\alpha u)$.

With respect to attraction and repulsion

$$\begin{aligned}
(\alpha\lambda_{a,r})(X,\phi) &= q_{a,r}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{a,r}(X-S)K^o_{a,r}(S-X;\phi)\alpha u(S,\theta,t)\,\mathrm{d}\theta\,\mathrm{d}S \\
&= q_{a,r}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{a,r}(X-S)K^o_{a,r}(S-X;\phi)u(R_{-\alpha}S,\theta-\alpha,t)\,\mathrm{d}\theta\,\mathrm{d}S
\end{aligned}$$

Let $\sigma = \theta - \alpha$ and let $Y = R_{-\alpha}S$, which means $S = R_\alpha Y$, then the previous expression can be rewritten

$$q_{a,r}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{a,r}(X-R_\alpha Y)K^o_{a,r}(R_\alpha Y - X;\phi)u(Y,\sigma,t)\,\mathrm{d}\sigma\,\mathrm{d}Y. \tag{2.3.7}$$

We can change the integral to integrate over Y instead of S because S, and consequently Y, is on an infinite domain. Similarly, we change the integral to integrate over σ instead of θ because θ, and consequently σ, is a periodic variable. Let $Z = R_{-\alpha}X$ and write $X = R_\alpha Z$, then equation (2.3.7) becomes

$$q_{a,r} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K^d_{a,r}(R_\alpha Z - R_\alpha Y) K^o_{a,r}(R_\alpha Y - R_\alpha Z; \phi) u(Y, \sigma, t) \, \mathrm{d}\sigma \, \mathrm{d}Y. \tag{2.3.8}$$

K^d is a function that only depends on the distance between Z and Y and K^o is a function that only depends on the angle that Z and Y make together. Therefore, K^d is invariant to the transformation, as long as both variables undergo the same one. Similarly, K^o is invariant as long as both variables undergo the same transformation. We use this property of K^d and K^o here. Let $\phi = \eta + \alpha$. Then equation (2.3.8) becomes

$$q_{a,r} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K^d_{al}(R_\alpha Z - R_\alpha Y) K^o_{al}(R_\alpha Y - R_\alpha Z; \eta + \alpha) u(Y, \sigma, t) \, \mathrm{d}\sigma \, \mathrm{d}Y.$$

Now all of the variables of K^d and K^o experience the transformation $R_{-\alpha}$

$$\begin{aligned} q_{a,r} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K^d_{al}(Z - Y) K^o_{al}(Y - Z; \eta) u(Y, \sigma, t) \, \mathrm{d}\sigma \, \mathrm{d}Y &= \lambda_{a,r}(Z, \eta) \\ &= \lambda_{a,r}(R_{-\alpha}X, \phi - \alpha). \end{aligned}$$

With respect to alignment, the only difference from the attraction and repulsion terms is that the orientation kernel doesn't depend on position, but depends on the direction variable ϕ. Therefore, after applying the same assumptions to the spatial

variables we have

$$\begin{aligned}
&q_{al} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_{al}^d(Z-Y) K_{al}^o(\phi - \sigma - \alpha) u(Y, \sigma, t) d\sigma dY \\
&= q_{al} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_{al}^d(Z-Y) K_{al}^o((\phi - \alpha) - \sigma) u(Y, \sigma, t) d\sigma dY \\
&= \lambda_{al}(Z, \phi - \alpha) \\
&= \lambda_{al}(R_{-\alpha} X, \phi - \alpha).
\end{aligned}$$

Therefore, all three terms are rotationally invariant, so the third term is rotationally invariant.

The fourth term contains the function T which contains a u, and therefore will be acted upon by α. In the same fashion as the translational symmetry proof, we notice that each component of T is identical to its λ counterpart, except for the probability function $w(\phi' - \phi)$. While this term does depend on the angle, and will therefore be impacted by rotation, it is important to note that the term $\phi' - \phi$ which is the only term w depends on is equal to $\phi' - \phi + \alpha - \alpha \Rightarrow (\phi' - \alpha) - (\phi - \alpha)$. Therefore, if we include this term and follow the same steps done in the proof for the third term, T under the influence of α will depend on the terms $R_\alpha X$, $\phi' - \alpha$, and $\phi - \alpha$ for all three T components. α can then be removed from the sum that makes up T, and the fourth term is rotationally invariant.

As each term is rotationally invariant, the entire equation is rotationally invariant.

2.3.3 Reflection

To show the reflection symmetry, let κ be the reflection operator acting on density u and define it as $\kappa u(X, \phi, t) = u(\tilde{X}, \tilde{\phi}, t)$, where if $X = (x, y)$, $\tilde{X} = (x, -y)$, and $\tilde{\phi} = -\phi$.

To show that equation (2.1.1) is reflection invariant, we apply κ to all functions

u in the system

$$\partial_t \kappa u + \gamma e_\phi \cdot \nabla_X \kappa u = -\lambda(X, \phi)\kappa u + \int_{-\pi}^{\pi} T(X, \phi', \phi)\kappa u(X, \phi', t)\, \mathrm{d}\phi'$$

As with the other two symmetries, the first term, (κu) does not change the variable t, and therefore $\partial_t(\kappa u) = \kappa \partial_t u$.

As explored when proving rotational symmetry, we can expand the second term to

$$\gamma[x' \cos\phi + y' \sin\phi]\kappa \nabla_X u(X, \phi, t),$$

where $x' = \frac{\partial x}{\partial x}$ and $y' = \frac{\partial y}{\partial y} \Rightarrow x' = 1$ and $y' = -1$. Again, because we use the chain rule on $\nabla_X u(R_{-\alpha}X, \phi, t)$ we are allowed to move κ to the other side of the gradient.

$$\gamma[x' \cos\phi + y' \sin\phi]\kappa \nabla_X u(X, \phi, t) = \gamma[\cos\phi + (-1)(\sin\phi)]\kappa \nabla_X u(X, \phi, t)$$

Using the fact that cosine is even and sine is odd, we can rewrite this as

$$\begin{aligned}
&\gamma[\cos(-\phi) + \sin(-\phi)]\kappa \nabla_X u(X, \phi, t) \\
&= \gamma[\kappa \cos(\phi) + \kappa \sin(\phi)]\kappa \nabla_X u(X, \phi, t) \\
&= \kappa[\gamma e_\phi \cdot \nabla_X u(X, \phi, t)]
\end{aligned}$$

Because both λ and T depend on u, when looking at the last two terms, we must first look at the impact of α on λ and T. Recall that $\lambda = \lambda_{al} + \lambda_a + \lambda_r$, so $-\lambda(\alpha u) = -\lambda_{al}(\alpha u) - \lambda_a(\alpha u) - \lambda_r(\alpha u)$.

With respect to attraction and repulsion

$$\begin{aligned}(\kappa\lambda_{a,r})(X,\phi) &= q_{a,r}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{a,r}(X-S)K^o_{a,r}(S-X;\phi)\kappa u(S,\theta,t)\,\mathrm{d}\theta\,\mathrm{d}S \\ &= q_{a,r}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{a,r}(X-S)K^o_{a,r}(S-X;\phi)u(\tilde{S},\tilde{\theta},t)\,\mathrm{d}\theta\,\mathrm{d}S\end{aligned}$$

We are able to change the integral to integrate over $\tilde{\theta}$ instead of θ because θ, and consequently $\tilde{\theta}$, is a periodic variable. We can also integrate over $\tilde{S}$ instead of S because S, and consequently $\tilde{S}$, is on an infinite domain; therefore, the previous expression can be written

$$q_{a,r}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{a,r}(x-s_x,y-s_y)K^o_{a,r}(s_x-x,s_y-y;\phi)u(\tilde{S},\tilde{\theta},t)\,\mathrm{d}\tilde{\theta}\,\mathrm{d}\tilde{S}. \tag{2.3.9}$$

Notice that

$$K^d_j = \frac{1}{A_j}e^{-(\sqrt{x^2+y^2}-d_j)^2/m_j^2},$$

for $j \in \{a,r,al\}$ where A_j, d_j, and m_j are constants. Because K^d only depends on the absolute value of $x-s_x$ and $y-s_y$, $x-s_x$ and $-y+s_y$ will give the same results. Therefore, we can replace $X-S$ with $\tilde{X}-\tilde{S}$, where $\tilde{X}=(x,-y)$ and $\tilde{S}=(s_x,-s_y)$, then (2.3.9) becomes

$$q_{a,r}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K^d_{a,r}(\tilde{X}-\tilde{S})K^o_{a,r}(s_x-x,s_y-y;\phi)u(\tilde{S},\tilde{\theta},t)\,\mathrm{d}\sigma\,\mathrm{d}Y. \tag{2.3.10}$$

Note

$$K^o_a(\phi-\psi) = \frac{1}{2\pi}(-\cos(\phi-\psi)+1),$$

and

$$K^o_r(\phi-\psi) = \frac{1}{2\pi}(\cos(\phi-\psi)+1).$$

We obtain the variable ψ from the equations

$$\cos\psi = \frac{u}{\sqrt{u^2+v^2}} \quad \text{and} \quad \sin\psi = \frac{v}{\sqrt{u^2+v^2}}, \tag{2.3.11}$$

where $S - X = (s_x - x, s_y - y) = (u, v)$. If we input $\tilde{S} - \tilde{X}$, this would give $(s_x - x, -s_y + y) = (s_x - x, -(s_y - y)) = (u, -v)$. When substituted into equations (2.3.11) we get

$$\cos\eta = \frac{u}{\sqrt{u^2+(-v)^2}} \quad \text{and} \quad \sin\eta = \frac{-v}{\sqrt{u^2+(-v)^2}}.$$

Because sine is an odd function, $\sin(\eta) = -\sin(\psi) = \sin(-\psi)$, and because cosine is an even function, $\cos(\eta) = \cos(-\psi)$ as well. Therefore, $\eta = -\psi$ and changing the dependence of K^o to $\tilde{S} - \tilde{X}$ will result in a dependence on $\phi - \tilde{\psi} = \phi + \psi$. Because both K_a^o and K_r^o are even functions, $-(\phi + \psi)$ will give the same results as $\phi + \psi$. Therefore, we can input $-\tilde{\phi} - \psi$ or $\tilde{\phi} - \psi$, and (2.3.10) becomes

$$q_{a,r} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_{a,r}^d(\tilde{X} - \tilde{S}) K_{a,r}^o(\tilde{S} - \tilde{X}; \tilde{\phi}) u(\tilde{S}, \tilde{\theta}, t)\, \mathrm{d}\tilde{\theta}\, \mathrm{d}\tilde{S} = \lambda_{a,r}(\tilde{X}, \tilde{\phi})$$

If we look at λ with respect to alignment, we would use the same proof for the distance kernel. The orientation kernel, however, depends on θ instead of the spatial variables. If $\tilde{\theta} = -\theta$, then $\theta = -\tilde{\theta}$, and the term becomes

$$q_{al} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_{al}^d(\tilde{X} - \tilde{S}) K_{al}^o(\phi + \tilde{\theta}) u(\tilde{S}, \tilde{\theta}, t)\, \mathrm{d}\sigma\, \mathrm{d}Y. \tag{2.3.12}$$

Note that

$$K_{al}^o(\phi + \tilde{\theta}) = \frac{1}{2\pi}(-\cos(\phi - \tilde{\theta}) + 1).$$

Because K_{al}^o is an even function, $-(\phi - \tilde{\theta})$ will give the same output as $\phi - \tilde{\theta}$. Therefore, we can change the dependence to $-\phi - \tilde{\theta}$ which equals $\tilde{\phi} - \tilde{\theta}$. Then, (2.3.12)

becomes

$$q_{al} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_{al}^d(\tilde{X} - \tilde{S}) K_{al}^o(\tilde{\phi} - \tilde{\theta}) u(\tilde{S}, \tilde{\theta}, t) \, \mathrm{d}\tilde{\theta} \, \mathrm{d}\tilde{S} = \lambda_{al}(\tilde{X}, \tilde{\phi}).$$

Because all three λ functions can have κ removed from them, the entire third term is reflection invariant.

As with the first two symmetry proofs, we call upon the fact that each T component is equivalent to its λ counterpart except for the probability function. Therefore, we only must show that $w(\phi' - \phi)$ is invariant under the symmetry. $w(\phi' - \phi)$ is an undetermined probability function, but it always depends on the absolute difference between ϕ' and ϕ. Therefore, an input of $-(\phi' - \phi)$ would give the same results as $\phi' - \phi$. We can replace $\phi' - \phi$ with $-\phi' + \phi$ which is equivalent to $\tilde{\phi}' - \tilde{\phi}$. Using this substitution, the function T becomes a function of $\tilde{X}$, $\tilde{\phi}'$, and $\tilde{\phi}$.

As each term is invariant with respect to reflection, so too is the entire equation.

CHAPTER 3

Solving the Equation with Time-Stepping

Having adapted the model to include non-linearity, we want to solve equation (2.2.6) numerically. Matthews and Cox developed a method to solve ODEs depending on both linear and non-linear terms [2]. The method implements both time differencing and time stepping to solve equations and is called ETD2RK, short hand for Second Order Exponential Time Differencing Method with Runge-Kutta Time Stepping. The utilization of the Runge-Kutta method is beneficial as it gives small errors and doesn't rely on past evaluations of the non-linear terms.

3.1 ETD2RK Method

In order to implement ETD2RK to solve (2.2.6) numerically, we first want to derive an ordinary differential equation from the existing partial differential equation. To obtain this ODE, it is ideal to transform the equation into Fourier space. Once

transformed, we should have an equation of the form

$$\partial_t u = Lu + F(u, t), \tag{3.1.1}$$

where L is a linear operator acting on u and F is a function producing the non-linear terms in the system.

ETD2RK involves discretizing time, calculating an exact answer for the linear terms in the system at each time and then estimating the non-linear terms at a future time using this solution. For each segment of time, t_n, we denote $u(t_n) = u_n$ and $F(u_n, t_n) = F_n$. When solving an equation of the form (3.1.1) we first calculate

$$a_n = e^{Lh} u_n + M_1 F_n$$

where

$$M_1 = L^{-1}(e^{Lh} - I).$$

To calculate the update step, we substitute these values into

$$u_{n+1} = a_n + M_2(F(a_n, t_n + h) - F_n)/h$$

where

$$M_2 = L^{-2}[e^{Lh} - (I + Lh)].$$

This method yields a truncation error of only $-h\ddot{F}/12$, and is fairly easy to implement. Because of this, we want to transform equation (2.2.6) to an ordinary differential equation.

3.2 Semi-Discrete Fourier Transformation

With the non-linear terms removed from (2.2.6), our linear system is as follows

$$\partial_t u + \gamma e_\phi \cdot \nabla_X u = -Gu + G\int_{-\pi}^{\pi} w(\phi - \phi')u(X, \phi', t)\,\mathrm{d}\phi', \tag{3.2.2}$$

where $G = g_1 + \frac{1}{2}g_2$ is a constant.

In order to find an exact solution to the linear system, we must first simplify it as much as possible. Due to the convolution in the last term on the right hand side, it is ideal to transform the system to Fourier space by taking the semi-discrete Fourier transform. To do this, we must first define a basis. Because our system depends on two spatial variables and one angular variable, we will need to take a 3-dimensional Fourier transformation. Therefore, our basis will depend on all three variables

$$\Phi_{k,l,m} = \Phi_k \Phi_l \Phi_m = e^{ikx} e^{ily} e^{im\phi} = e^{ikx+ily+im\phi}. \tag{3.2.3}$$

Applying this basis, we obtain the semi-discrete Fourier transformation and its inverse

$$\hat{u}_{k,l,m}(t) = \frac{KLM}{8\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\pi}^{\pi} u(x, y, \phi, t)\Phi_{-k,-l,-m}(x, y, \phi)\,\mathrm{d}\phi\,\mathrm{d}x\,\mathrm{d}y, \tag{3.2.4}$$

$$u(x, y, \phi, t) = \frac{1}{KLM}\sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}}\sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}}\sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}} \hat{u}_{k,l,m}(t)\Phi_{k,l,m}(x, y, \phi), \tag{3.2.5}$$

where K, L, and M are the number of discretized x, y, and ϕ points respectively. The constant in front of (3.2.4) is applied to ensure that our semi-discrete Fourier transformation yields the same results as MATLAB's fully discrete Fourier transformation. We can now replace all functions u in equation (3.2.2) with (3.2.5), in order to transform our partial differential equation into Fourier space. We will look

at each term in (3.2.2) separately in order to obtain the coefficients.

The first term is the time derivative of u. This derivative has no dependence on space or direction, so it can be moved into the triple summation, where it is applied only to $\hat{u}_{k,l,m}(t)$. $\Phi_{k,l,m}$ does not depend on time, and is therefore regarded as constant. The term then becomes

$$\frac{1}{KLM} \sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}} \sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}} \sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}} \partial_t \hat{u}_{k,l,m}(t) \Phi_{k,l,m}(x, y, \phi).$$

The second term is the spatial derivative $\gamma e_\phi \cdot \nabla_X u$. Note that $\widehat{(u_x)}_{k,l,m} = ik\hat{u}_{k,l,m}$ and $\widehat{(u_y)}_{k,l,m} = il\hat{u}_{k,l,m}$, because Φ contains the only spatial variables in the inverse Fourier transform. Because of this, the term $\widehat{(\nabla_X \cdot u)}$ inside the triple summation would be equal to $(ik\hat{u}_{k,l,m}, il\hat{u}_{k,l,m})$.

The term e_ϕ ensures that individuals move in a direction across the space at speed γ, as it provides the x and y direction for the individual derived from the angle ϕ [6]. Note that e_ϕ can be rewritten from $(\cos\phi, \sin\phi)$ to $(\frac{e^{i\phi}+e^{-i\phi}}{2}, \frac{e^{i\phi}-e^{-i\phi}}{2i})$ using Euler's formula. This is a more favourable term, as the exponential of ϕ and $-\phi$ terms are of the same form as our Fourier basis for ϕ. Therefore, when the dot product is taken between these two vectors, the second term becomes

$$\frac{1}{KLM} \sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}} \sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}} \sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}} \gamma \left[\frac{(e^{i\phi}+e^{-i\phi})}{2} ik\hat{u}_{k,l,m}(t) + \frac{(e^{i\phi}-e^{-i\phi})}{2i} il\hat{u}_{k,l,m}(t) \right] \Phi_{k,l,m}(x, y, \phi).$$

We can simplify this term by cancelling the purely imaginary terms from the second term in the brackets and removing the common factor of $\frac{\hat{u}_{k,l,m}(t)}{2}$ from the brackets giving

$$\frac{1}{KLM} \sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}} \sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}} \sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}} \frac{\gamma}{2} [(e^{i\phi}+e^{-i\phi})ik + (e^{i\phi}-e^{-i\phi})l] \hat{u}_{k,l,m}(t) \Phi_{k,l,m}(x, y, \phi)$$

We can simplify even further by substituting in $\Phi_{k,l,m} = e^{ikx} e^{ily} e^{im\phi}$. The Φ_m component can be combined with the other exponential terms found inside the

brackets, and the term becomes

$$\frac{1}{KLM}\sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}}\sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}}\sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}}\frac{\gamma}{2}e^{ikx+ily}[(ik+l)e^{i(m+1)\phi}+(ik-l)e^{i(m-1)\phi}]\hat{u}_{k,l,m}.(t),$$

which is the resulting Fourier transform of the second term.

The first term on the right hand side of equation (3.2.2) is simply the function u multiplied by a constant. The constant $-G$ is not dependent on space or direction, so, similar to the temporal derivative, the constant can be moved into the triple summation and rewritten as

$$\frac{1}{KLM}\sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}}\sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}}\sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}}-G\hat{u}_{k,l,m}(t)\Phi_{k,l,m}(x,y,\phi).$$

The final term on the right hand side of (3.2.2) was the motivation for transforming the equation into Fourier space. This term is a convolution, so when the Fourier transformation of the term is taken, it should become a product rather than an integral. When we substitute in the semi-discrete inverse Fourier transformation for $u(X,\phi',t)$, we get

$$\int_{-\pi}^{\pi}w(\phi-\phi')\frac{1}{KLM}\sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}}\sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}}\sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}}\hat{u}_{k,l,m}(t)\Phi_{k,l,m}(x,y,\phi')\,\mathrm{d}\phi'.$$

Except for Φ_m and $w(\phi-\phi')$, which depend on ϕ', all terms can be factored from the integral. It is important to note that the second term on the right hand side of (3.2.2) depends on ϕ' instead of ϕ because the u term that was originally in the integral depends on ϕ' rather than ϕ. The entire term then becomes

$$\frac{1}{KLM}\sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}}\sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}}\sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}}\hat{u}_{k,l,m}(t)e^{ikx}e^{ily}\int_{-\pi}^{\pi}w(\phi-\phi')e^{im\phi'}\,\mathrm{d}\phi'.$$

If we allow $s = \phi - \phi'$, and integrate over s, we get

$$\frac{1}{KLM} \sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}} \sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}} \sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}} \hat{u}_{k,l,m}(t) e^{ikx} e^{ily} \int_{-\pi}^{\pi} w(s) e^{im(\phi - s)} ds.$$

We can remove $e^{im\phi}$ from the integral and the remaining integral is equal to the transform of w multiplied by the constant $\frac{2\pi}{n}$, where n is the size of the discretized w. Therefore the entire final term becomes

$$\frac{1}{KLM} \sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}} \sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}} \sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}} \frac{2\pi}{n} \widehat{w}(\phi - \phi') \hat{u}_{k,l,m}(t) \Phi_{k,l,m}(x, y, \phi),$$

which is a product as expected. Note that the $\Phi_{k,l,m}(x, y, \phi)$ term is once again complete because of the left over $e^{im\phi}$ term.

3.3 Formation of the Ordinary Differential Equation

We can now substitute these four coefficients into equation (3.2.2)

$$\begin{aligned}
&\frac{1}{KLM} \sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}} \sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}} \sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}} \partial_t \hat{u}_{k,l,m}(t) \Phi_{k,l,m}(x, y, \phi) + \cdots \\
&\frac{1}{KLM} \sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}} \sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}} \sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}} \frac{\gamma}{2} e^{ikx+ily} [(ik + l) e^{i(m+1)\phi} + (ik - l) e^{i(m-1)\phi}] \hat{u}_{k,l,m}(t) \\
&= \frac{-G}{KLM} \sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}} \sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}} \sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}} \hat{u}_{k,l,m}(t) \Phi_{k,l,m}(x, y, \phi) + \cdots \\
&\frac{G}{KLM} \sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}} \sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}} \sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}} \frac{2\pi}{n} \widehat{w(\phi - \phi')} \hat{u}_{k,l,m}(t) \Phi_{k,l,m}(x, y, \phi).
\end{aligned} \tag{3.3.6}$$

The triple summation and the term $\frac{1}{KLM}$ are common amongst all terms, and therefore can be moved to the front of the equation . Furthermore, $\Phi_k(x)$ and $\Phi_l(y)$ are

both common in all terms and can be extracted

$$\frac{1}{KLM}\sum_{k=-\frac{K}{2}+1}^{\frac{K}{2}}\sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}}\sum_{m=-\frac{M}{2}+1}^{\frac{M}{2}}\Bigg[\Bigg(\partial_t\hat{u}_{k,l,m}(t)\Phi_m(\phi)+\frac{\gamma}{2}[(ik+l)e^{i(m+1)\phi}+\cdots \\ (ik-l)e^{i(m-1)\phi}]\hat{u}_{k,l,m}(t)+G\hat{u}_{k,l,m}(t)\Phi_m(\phi)-\cdots \\ G\frac{2\pi}{n}\widehat{w}(\phi-\phi')\hat{u}_{k,l,m}(t)\Phi_m(\phi)\Bigg)\Phi_k(x)\Phi_l(y)\Bigg]=0. \tag{3.3.7}$$

$\Phi_m(\phi)$ can't be removed from the brackets because it does not appear in the second term. However, $e^{i(m+1)\phi}$ and $e^{i(m-1)\phi}$ are equal to $\Phi_{m+1}(\phi)$ and $\Phi_{m-1}(\phi)$ respectively.

To have this equality hold, we need only for the summand to be equal to 0. Therefore, the equation we need to solve is

$$\begin{aligned}\partial_t\hat{u}_{k,l,m}(t)\Phi_m(\phi)=&-\frac{\gamma}{2}[(ik+l)\Phi_{m+1}(\phi)+(ik-l)\Phi_{m-1}(\phi)]\hat{u}_{k,l,m}(t)\\&-G\hat{u}_{k,l,m}(t)\Phi_m(\phi)+G\frac{2\pi}{n}\widehat{w}(\phi-\phi')\hat{u}_{k,l,m}(t)\Phi_m(\phi).\end{aligned} \tag{3.3.8}$$

If we look at (3.3.8) exclusively with respect to ϕ, we get the equation

$$\partial_t\hat{u}_m(t)\Phi_m(\phi)=-\frac{\gamma}{2}[(ik+l)\Phi_{m+1}(\phi)+(ik-l)\Phi_{m-1}(\phi)]\hat{u}_m(t)+\mathcal{F}_j\hat{u}_m(t)\Phi_m(\phi), \tag{3.3.9}$$

where $\mathcal{F}_j$ is equal to $-G+G\frac{2\pi}{n}\widehat{w}_j(\phi-\phi')$. It is important to note, however, that because the equation contained Φ_{m-1} and Φ_{m+1}, eventually the equation will call a Φ that is out of bounds. That is, if $m=-\frac{M}{2}+1$, Φ_{m-1} will equal $\Phi_{-\frac{M}{2}}$, which is not in bounds. Similarly, if $m=\frac{M}{2}$, Φ_{m+1} will equal $\Phi_{\frac{M}{2}+1}$, which is also not in bounds. Because Φ depends on ϕ, the periodic angle indicating the direction the individual is turning into, then these periodic boundary conditions can be applied

to the linear operator

$$\partial_t \begin{bmatrix} \hat{u}_{-\frac{M}{2}+1} \\ \hat{u}_{-\frac{M}{2}+2} \\ \vdots \\ \hat{u}_{\frac{M}{2}} \end{bmatrix} = \begin{bmatrix} \mathcal{F}_{-\frac{M}{2}+1} & \frac{\gamma}{2}(ik-l) & \dots & 0 & \frac{\gamma}{2}(ik+l) \\ \frac{\gamma}{2}(ik+l) & \mathcal{F}_{-\frac{M}{2}+2} & \frac{\gamma}{2}(ik-l) & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & \frac{\gamma}{2}(ik+l) & \mathcal{F}_{\frac{M}{2}-1} & \frac{\gamma}{2}(ik-l) \\ \frac{\gamma}{2}(ik-l) & 0 & \dots & \frac{\gamma}{2}(ik+l) & \mathcal{F}_{\frac{M}{2}} \end{bmatrix} \begin{bmatrix} \hat{u}_{-\frac{M}{2}+1} \\ \hat{u}_{-\frac{M}{2}+2} \\ \vdots \\ \hat{u}_{\frac{M}{2}} \end{bmatrix}. \tag{3.3.10}$$

CHAPTER 4

Solving the Linear Terms Analytically

4.1 Splitting Method

ETD2RK relies on the linear terms in the system having an exact solution. The equation given in (3.3.10) is a simple ODE and has an exact solution. The solution to the system is $u(X, \phi, t) = e^{tL}u_0(X, \phi)$, with initial condition u_0 and linear operator L. If the linear operator L can be diagonalized, then the exponential of the matrix is equal to the diagonal matrix E, where all entries $E_{j,j}$ on the diagonal are equal to $e^{L_{j,j}}$.

In its current form, the linear operator matrix

$$A = \begin{bmatrix} \mathcal{F}_{-\frac{M}{2}+1} & \frac{\gamma}{2}(ik-l) & \dots & 0 & \frac{\gamma}{2}(ik+l) \\ \frac{\gamma}{2}(ik+l) & \mathcal{F}_{-\frac{M}{2}+2} & \frac{\gamma}{2}(ik-l) & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & \frac{\gamma}{2}(ik+l) & \mathcal{F}_{\frac{M}{2}-1} & \frac{\gamma}{2}(ik-l) \\ \frac{\gamma}{2}(ik-l) & 0 & \dots & \frac{\gamma}{2}(ik+l) & \mathcal{F}_{\frac{M}{2}} \end{bmatrix},$$

is not easily diagonalizable. In order to derive a formula for diagonalizing the matrix A, we call upon the splitting method. Proposed by MacNamara and Strang [8], the splitting method allows us to approximate the exponential of a matrix if we can split the matrix into two matrices that each can be diagonalized. The idea is that

$$e^{h(A+B)} \approx e^{hA}e^{hB}$$

is first order accurate. However,

$$e^{h(A+B)} \approx e^{hB}e^{hA}$$

is equally accurate. Therefore, if we take the average of these two approximations, the symmetry adds accuracy, and the approximation

$$e^{h(A+B)} \approx \frac{e^{hA}e^{hB} + e^{hB}e^{hA}}{2}$$

is actually second order accurate.

In order to implement this method, we must be able to split the matrix A into the sum of two matrices, both of which can be easily diagonalized.

If we take all of the of the terms on the diagonal of A and put them in a diagonal matrix, we end up with

$$A_1 = \begin{bmatrix} \mathcal{F}_{-\frac{M}{2}+1} & 0 & \dots & 0 & 0 \\ 0 & \mathcal{F}_{-\frac{M}{2}+2} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \mathcal{F}_{\frac{M}{2}-1} & 0 \\ 0 & 0 & \dots & 0 & \mathcal{F}_{\frac{M}{2}} \end{bmatrix}.$$

This is already a diagonal matrix, so we can easily find e^{A_1}. The remaining matrix is

$$A_2 = \begin{bmatrix} 0 & \frac{\gamma}{2}(ik-l) & \dots & 0 & \frac{\gamma}{2}(ik+l) \\ \frac{\gamma}{2}(ik+l) & 0 & \frac{\gamma}{2}(ik-l) & \dots & 0 \\ \vdots & \ddots & & \ddots & \vdots \\ 0 & \dots & \frac{\gamma}{2}(ik+l) & 0 & \frac{\gamma}{2}(ik-l) \\ \frac{\gamma}{2}(ik-l) & 0 & \dots & \frac{\gamma}{2}(ik+l) & 0 \end{bmatrix}$$

which is a circulant matrix.

4.2 Circulant Matrices

A circulant matrix is an $N \times N$ matrix of the form

$$B = \begin{bmatrix} b_0 & b_{N-1} & \dots & b_2 & b_1 \\ b_1 & b_0 & b_{N-1} & \dots & b_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ b_{N-2} & \dots & b_1 & b_0 & b_{N-1} \\ b_{N-1} & b_{N-2} & \dots & b_1 & b_0 \end{bmatrix}$$

where every column's entries are the previous column's entries shifted down one.

An important property of circulant matrices is their ability to be diagonized by the discrete Fourier transform [1]. That is, $F^{-1}BF = diag(Fb)$, where b is equal to the first column of circulant matrix B and F is the $N \times N$ Fourier matrix. Therefore, the eigenvectors for all circulant matrices are the same and will be of the form

$$v_\xi = (1, e^{i(\xi-1)\frac{2\pi}{N}}, e^{2i(\xi-1)\frac{2\pi}{N}}, \cdots, e^{(N-1)i(\xi-1)\frac{2\pi}{N}})^{\mathrm{T}},$$

where $e^{i\xi\frac{2\pi}{n}}$ is the n^{th} root of unity. The eigenvalues α will be of the form

$$\alpha_\xi = b_0 + b_{N-1}e^{i(\xi-1)\frac{2\pi}{N}} + b_{N-2}e^{2i(\xi-1)\frac{2\pi}{N}} + \cdots + b_1 e^{(N-1)i(\xi-1)\frac{2\pi}{N}}.$$

In the case of A_2, only b_{N-1} and b_1 are non-zero. Therefore, our eigenvalues are of the form

$$\begin{aligned} \alpha_\xi &= b_{N-1}e^{i(\xi-1)\frac{2\pi}{N}} + b_1 e^{(N-1)i(\xi-1)\frac{2\pi}{N}} \\ &= b_{N-1}e^{i(\xi-1)\frac{2\pi}{N}} + b_1 e^{(-1)i(\xi-1)\frac{2\pi}{N}} e^{i(\xi-1)2\pi} \end{aligned} \tag{4.2.1}$$

If we substitute in Euler's formula $e^{ix} = \cos(x) + i\sin(x)$ into (4.2.1), we get

$$\begin{aligned} \alpha_\xi = b_{N-1}&\left[\cos\left((\xi-1)\frac{2\pi}{N}\right) + i\sin\left((\xi-1)\frac{2\pi}{N}\right)\right] \\ + b_1&\left[\cos((\xi-1)2\pi) + i\sin((\xi-1)2\pi)\right]\left[\cos\left(-(\xi-1)\frac{2\pi}{N}\right) + i\sin\left(-(\xi-1)\frac{2\pi}{N}\right)\right] \end{aligned} \tag{4.2.2}$$

Because ξ will be a whole number from 1 to N, $\sin((\xi-1)2\pi)$ is equal to 0 and $\cos((\xi-1)2\pi)$ is equal to 1. Finally, by implementing the properties of even and odd functions, we know that $\cos(-x) = \cos(x)$ and $\sin(-x) = \sin(x)$. Therefore

(4.2.2) simplifies to

$$\begin{aligned}\alpha_\xi = b_{N-1}&\left[\cos\left((\xi-1)\frac{2\pi}{N}\right)+i\sin\left((\xi-1)\frac{2\pi}{N}\right)\right]+\cdots\\ &b_1\left[\cos\left((\xi-1)\frac{2\pi}{N}\right)-i\sin\left((\xi-1)\frac{2\pi}{N}\right)\right]\\ =[b_{N-1}+b_1]&\cos\left((\xi-1)\frac{2\pi}{N}\right)+i[b_{N-1}-b_1]\sin\left((\xi-1)\frac{2\pi}{N}\right)\end{aligned} \tag{4.2.3}$$

As previously stated, A_2 is a circulant matrix with $b_{N-1}=\frac{\gamma}{2}(ik-l)$, $b_1=\frac{\gamma}{2}(ik+l)$, and $b_n=0$ for all other n. Therefore, the eigenvectors of A_2 will be equal to v_ξ shown above with $N=M$, and the eigenvalues will be

$$\begin{aligned}\alpha_\xi = &\left[\frac{\gamma}{2}(ik-l)+\frac{\gamma}{2}(ik+l)\right]\cos\left((\xi-1)\frac{2\pi}{M}\right)+\cdots\\ &i\left[\frac{\gamma}{2}(ik-l)-\frac{\gamma}{2}(ik+l)\right]\sin\left((\xi-1)\frac{2\pi}{M}\right)\\ =[\gamma ik]&\cos\left((\xi-1)\frac{2\pi}{M}\right)-i[\gamma l]\sin\left((\xi-1)\frac{2\pi}{M}\right).\end{aligned} \tag{4.2.4}$$

Using the discrete Fourier transform, we now have a method for diagonalizing A_2 exactly for all wave numbers k and l.

4.3 Approximate Solution to the Linear Terms

Because both A_1 and A_2 can be diagonalized, and $A_1+A_2=A$, we can apply the splitting method to obtain a second-order approximation to (3.3.10). To compute e^{hA_1}, we multiply the entire matrix A_1 by step size h. This is still a diagonal matrix. The exponential of a diagonal matrix will just be a diagonal matrix with e raised to the diagonal of the original matrix A_1. To compute e^{hA_2}, we multiply A_2 by the step size h, and then multiply the resulting matrix on the left by the Fourier matrix

and on the right by the inverse Fourier matrix. This will result in a diagonal matrix, which we can take the exponential of the same way we did for A_1. Before we can use this exponential, however, we must remember to reverse the Fourier transformation by multiplying the new matrix by the inverse Fourier matrix on the left and by the Fourier matrix on the right.

The final approximation for the exponential is

$$e^{h(A_1+A_2)} \approx \frac{e^{hA_1}F^{-1}(e^{F(hA_2)F^{-1}})F + F^{-1}(e^{F(hA_2)F^{-1}})Fe^{hA_1}}{2} \tag{4.3.5}$$

making the approximate solution to the linear terms

$$\hat{u} \approx \frac{e^{hA_1}F^{-1}(e^{F(hA_2)F^{-1}})F + F^{-1}(e^{F(hA_2)F^{-1}})Fe^{hA_1}}{2}\hat{u_0}. \tag{4.3.6}$$

CHAPTER 5

Coding the Time-Stepping in MATLAB

In order to compute numerical approximations to (2.2.6) using ETD2RK, it is ideal to use some kind of software to aid in the computations. We use MATLAB for all computations here, which has a built in fast Fourier transform, useful given the number of times we implement the Fourier transform.

5.1 Dimensional Analysis

While running simulations of the system, we can obtain time scales from the dimensional analysis found in Table 5.1

Table 5.1: The dimensions of each of the terms found in (2.1.1)

Dimensional Analysis of the System	
Term	Dimension
u	N/m^{2}radians
g_1	1/s
g_2	1/s

Continuation of Table 5.1	
Term	Dimension
γ	m/s
w	"probability"/radians
m_j	m
d_j	m
A_j	m^2
K_j^d	$1/\text{m}^2$
K_j^o	radians
q_j	m^2/N radians

Because the term Γ must be dimensionless, we are able to determine the dimensions of all parameters in the system. For the interested reader, the time scales acquired from Table 5.1 give an idea of how quickly we will see behaviour in the system based on the magnitude set for each parameter.

5.2 Coding a Solution to the Linear Terms

At the beginning of the code, we must initialize the density u for all space and directions in the domain. Once we have an initial condition for u we can take the 3-dimensional Fourier transform of the density. For every time step in the code, we loop through all wave numbers k and l for the two spatial variables. With these values set, we can extract the decoupled $1 \times m$ vector pertaining to these two wave numbers that depends only on angular direction.

We compute the approximate solution to the linear terms using (4.3.6). We first compute the matrix A_1 using the Fourier coefficients of an approximate Gaussian distribution and the constants g_1 and g_2. For each wave number k and l, we are able

to compute the matrix A_2. We use the formula (4.3.5) to compute the exponential of $A_1 + A_2$ and multiply this by u_n to obtain u_{n+1} for times t_n and t_{n+1} respectively. The vector u_{n+1} corresponds with the same wave numbers k and l and is stored in a 3-dimensional matrix, in the position associated with the current k and l values.

Once we have computed u_{n+1} for all values of k and l, we can re-establish the $n \times n \times m$ matrix in Fourier space, and then take the inverse Fourier transform to get our new density.

5.3 Testing the Code Analytically

Once the time stepping method has been implemented into MATLAB, it is important to check the accuracy of the code. If we look solely at the solution to the linear parts of the system, it is possible to solve some cases analytically. Because of this, we first remove the non-linearity from the system and choose two test cases for the linear terms that could be checked analytically.

These tests were not chosen for realism, but for the sake of simplifying the equation to a familiar form.

5.3.1 An Initial Condition with no Spatial Dependence

If we assume that the density u has no spatial dependence, then the second term on the left hand side of (3.2.2), regarding the spatial derivative of u, goes to zero. This leaves

$$\partial_t u = -Gu + G\int_{-\pi}^{\pi} w(\phi - \phi')u(X, \phi', t)d\phi'.$$

We transform the remaining terms to Fourier space to simplify the convolution in the last term to a product

$$\partial_t \hat{u_j} = (G\hat{w}_j(\phi - \phi') - G)\hat{u}_j(X, \phi', t).$$

This is an ordinary differential equation, where $(Gw_j\widehat{(\phi-\phi')}-G)$ can be represented as a diagonal matrix. There is no need to use the splitting method. The solution to this system is $\widehat{u_j}=e^{tA_j}\hat{u}_{0j}$, where $A_j=G\hat{w}_j-G$. Therefore, we can expect that in Fourier space, the first coefficient will stay constant and the remaining terms will decrease exponentially.

Coding the Probability Function w

We require that the condition $u_n=e^{-G+Gw_n}u_0$ is met which would require w_0 to be equal to 1. w is a descritized approximation of the Gaussian distribution. In order for the initial condition to hold, the first component of $\frac{2\pi}{m}\hat{w}$ in our code must be equal to 1, where m is the size of w. This is only true when m is large and the discretized grid for the angle is dense.

Consider the graph given in Figure 5.1 which illustrates the log of error of the Gaussian approximation as n increases.

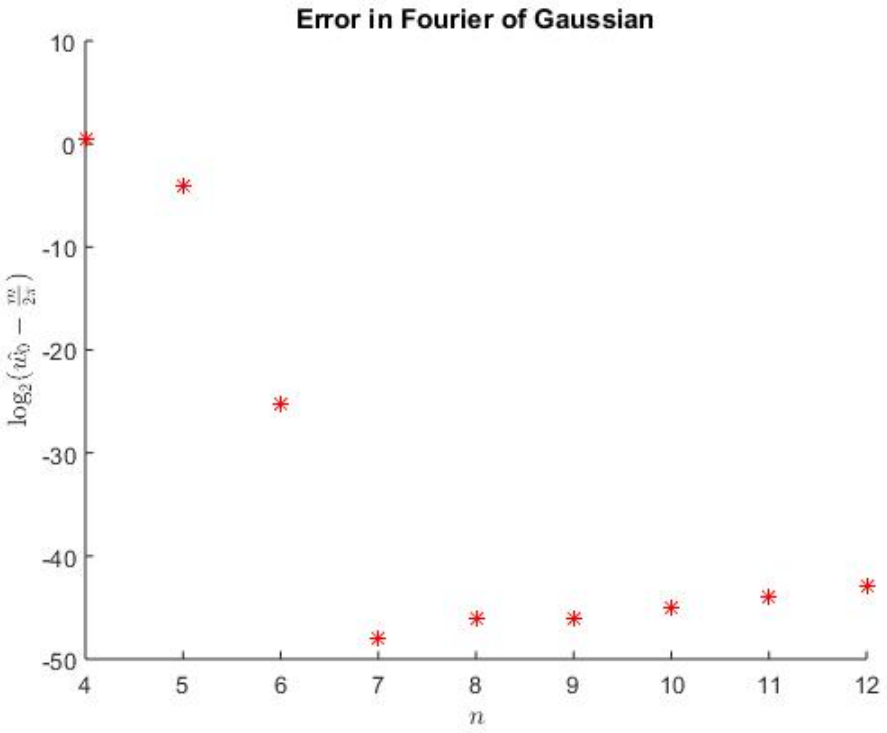


Figure 5.1: The semi-log plot of error when computing the Gaussian approximation for the probability function w. The number of points in our discretization of w is equal to $m = 2^n$.

To test our results with no spatial dependence in density, we set the initial condition on density to be $u_0 = 1 + \cos\phi$. The initial condition is chosen for three reasons. First, it is a non-linear function. Second, it depends only on the direction of the animal, and not on the position. Finally, the initial condition must be periodic to satisfy the boundary conditions, making sinusoidal functions a good choice.

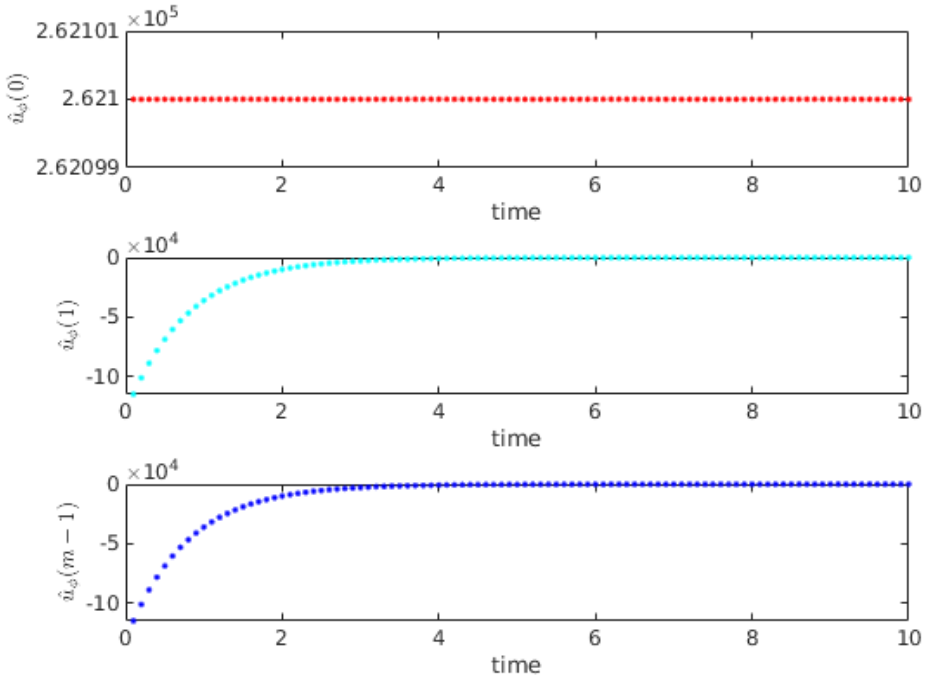


Figure 5.2: The progression of the 0, 1, and $m-1$ Fourier coefficients of density u, which has a total of 64 coefficients overall. $u_0 = 1 + \cos\phi$, $\gamma = 0.1$, $g_1 = 0.2$ and $g_2 = 0.9$. The total run time is 8 with a step size of 0.1.

Figure 5.2 shows three graphs with the progression of three different coefficients of $\hat{u}$ over time. The 0^{th} coefficient stays constant at 2.621×10^5. The other two coefficients (which are in fact complex conjugates of one another) tend towards 0. Figure 5.3, two semi-logarithmic graphs of the progression of the first and the last coefficients over time, shows that these same two coefficients are actually decreasing exponentially. This is, indeed, agrees with our expected results.

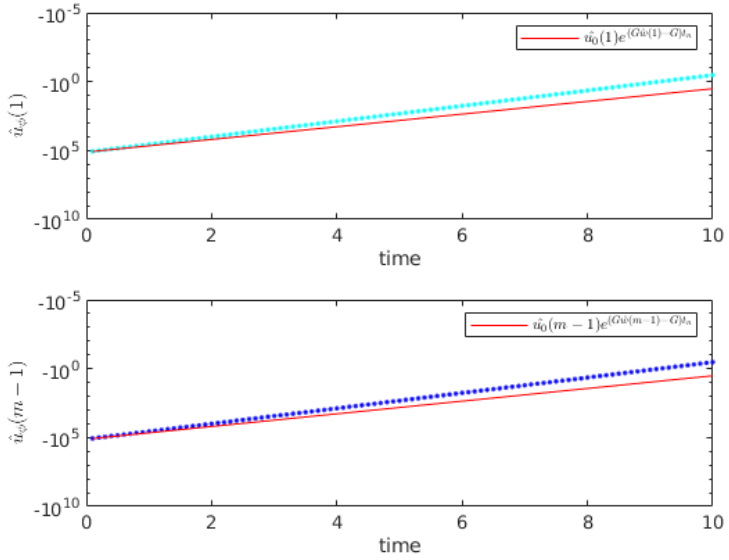


Figure 5.3: The same case as Figure 5.2 for coefficients 1 and $m-1$, on a semi-logarithmic plot.

5.3.2 Eliminating the Probability Function

If we assume that the probability function w in the system is negligible and set it to 0, then the last term on the right hand side of the equation goes to zero. This leaves

$$\partial_t u + \gamma e_\phi \cdot \nabla_X u = -Gu.$$

Because there is no convolution in this system, it is unnecessary to evaluate it in Fourier space. This is the inhomogeneous transport equation and can be solved using characteristics.

If we let $\frac{\partial t}{\partial r} = 1$, $\frac{\partial x}{\partial r} = \gamma \cos\phi$, $\frac{\partial y}{\partial r} = \gamma \sin\phi$, and $t(0;s) = 0$, $x(0;s) = x_0$, $y(0;s) = y_0$, then $t = r$, $x = x_0 + \gamma r \cos\phi$, and $y = y_0 + \gamma r \sin\phi$. Along this characteristic base curve

$$\frac{du}{dr} = \frac{dt}{dr}\frac{\partial u}{\partial t} + \frac{dx}{dr}\frac{\partial u}{\partial x} + \frac{dy}{dr}\frac{\partial u}{\partial y} = -Gu, \quad u(0;s) = F(s)$$

has the solution

$$u(r;s) = F(s)e^{-rG}.$$

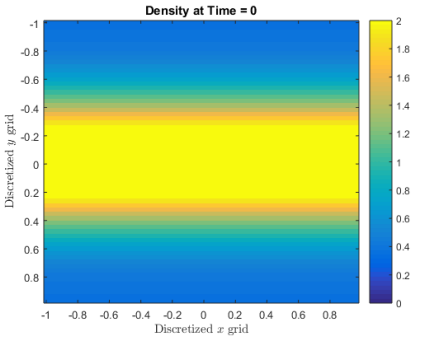
Eliminating r and s, one has

$$u = F(x - \gamma\cos(\phi)t, y - \gamma\sin(\phi)t)e^{-tG}.$$

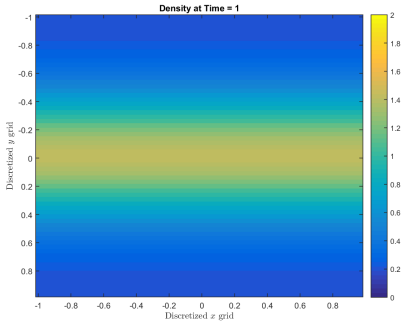
In order to test that the code gives this solution, we first test an initial condition that only depends on y. For this test, we used the initial condition $F(x, y) = \exp(\cos(\pi y))$ which is periodic in y and satisfies the boundary conditions. As the solution will depend not only on y, but on ϕ, the best way to check the solution is to set the angle to a constant.

In Figure 5.4, we see the solution for u with $\phi = -\pi$. When $\phi = -\pi$, the solution simplifies to $u(y, t) = F(y - \gamma\sin(-\pi)t)e^{-tG} = F(y)e^{-tG}$. This would imply that the distribution of the density will stay the same, but the density over the entire domain will decrease exponentially. In Figure 5.4 the density distribution is constant over x, and periodic over y. The distribution stays the same as time goes to infinity, but the total number of animals is decreasing exponentially.

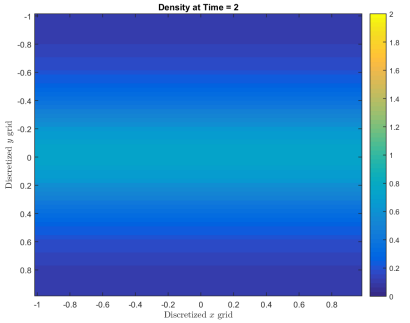
Because x is constant in this system, we can set $x = 1$ and look at how the solution changes in y over time. In Figure 5.5 we can see that the solution is not shifting as time increases, but is approaching zero.



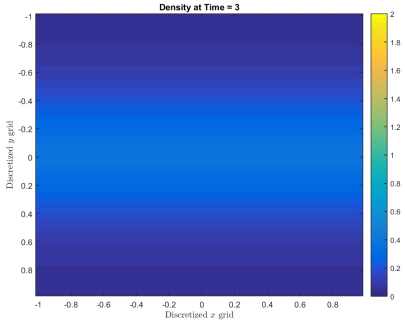
(a) Initial density $\exp(\cos(\pi y))$.



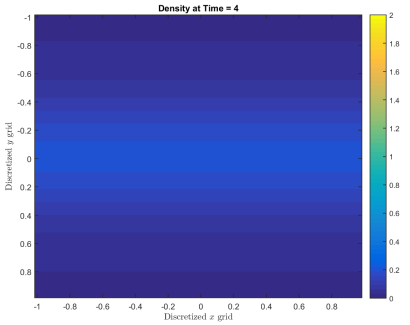
(b) Density after 10 time steps.



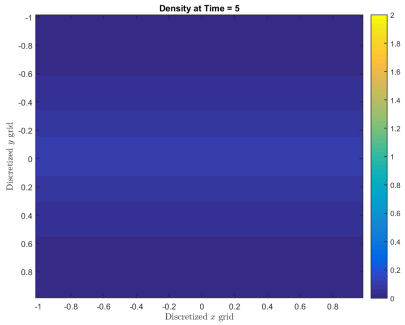
(c) Density after 20 time steps.



(d) Density after 30 time steps.



(e) Density after 40 time steps.



(f) Density after 50 time steps.

Figure 5.4: The density distribution colourmaps at six different times for $\phi = -\pi$. $\gamma = 0.1$, $g_1 = 0.2$, and $g_2 = 0.9$.

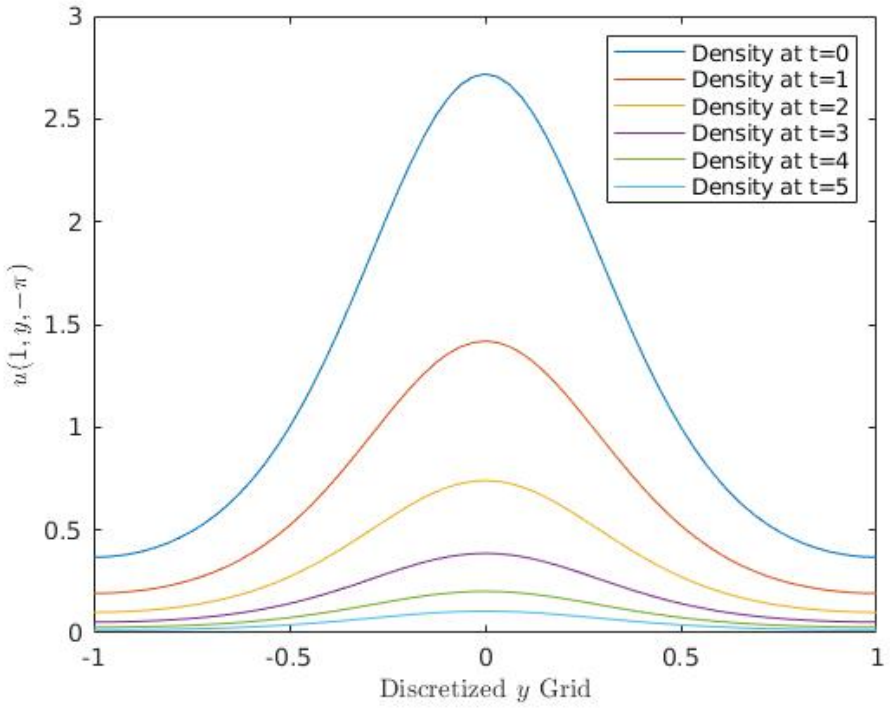
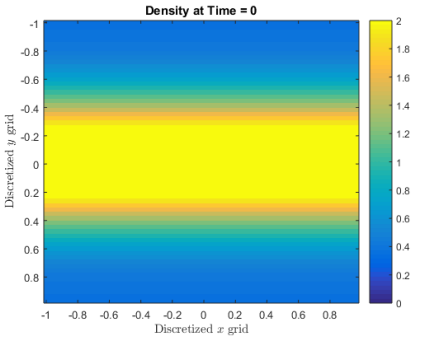
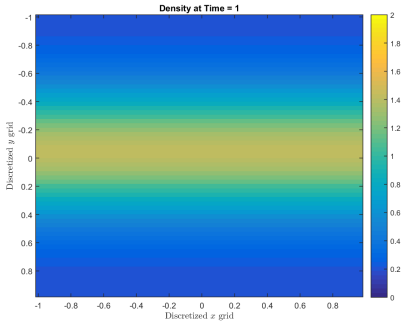


Figure 5.5: The density distribution at six different time steps for $\phi = -\pi$ and $x = 1$. $\gamma = 0.1$, $g_1 = 0.2$, and $g_2 = 0.9$.

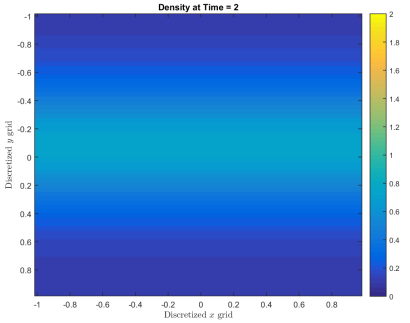
If we set ϕ to a value other than $-\pi$, π or 0, the shifting of the distribution becomes apparent, such as in Figure 5.6 where $\phi = -\frac{\pi}{2}$. As $\sin(-\frac{\pi}{2}) = 1$, the maximum of sine, this value of ϕ should actually yield the largest shift. Also, the shift should be in the upwards direction on our graph as it is a negative shift in the y direction. Again, in order to see the shift more clearly, we can fix x and see how the solution changes over time with respect to y in Figure 5.7.



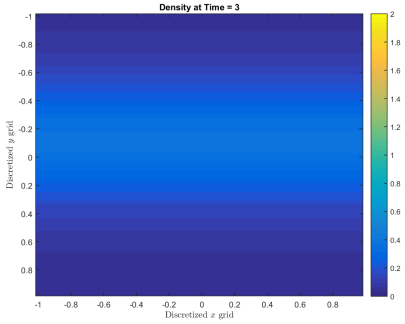
(a) Initial density $\exp(\cos(\pi y))$.



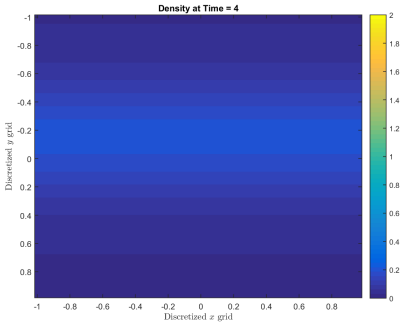
(b) Density after 10 time steps.



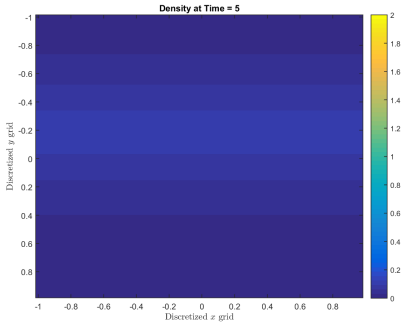
(c) Density after 20 time steps.



(d) Density after 30 time steps.



(e) Density after 40 time steps.



(f) Density after 50 time steps.

Figure 5.6: The density distribution colourmaps at six different times for $\phi = \frac{-\pi}{2}$. $\gamma = 0.1$, $g_1 = 0.2$, and $g_2 = 0.9$.

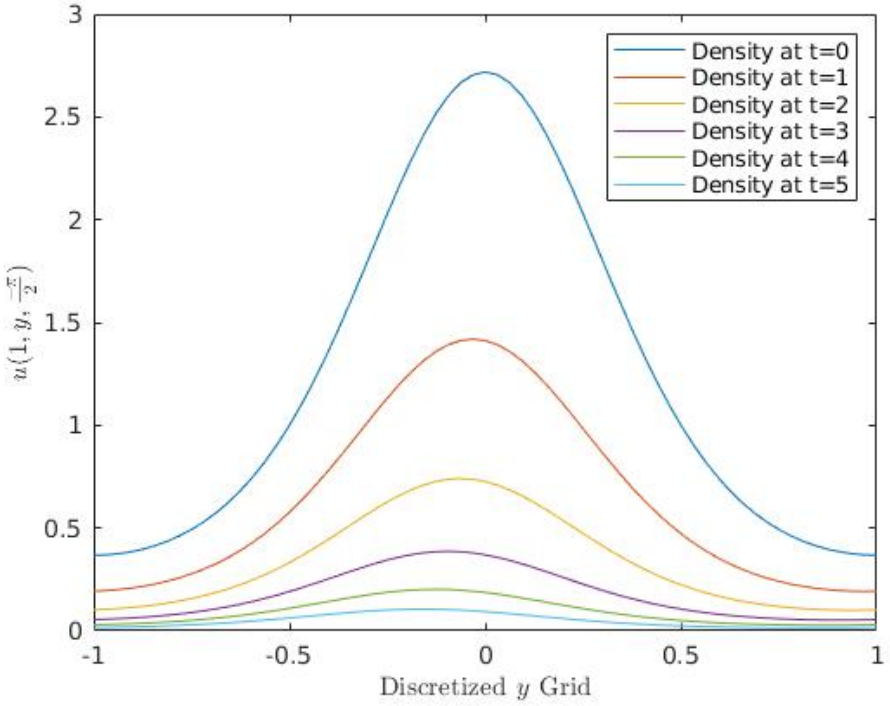
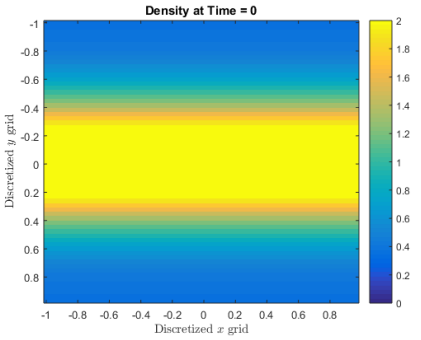
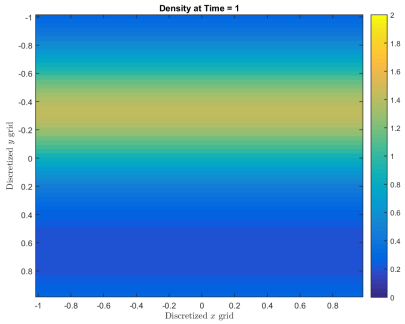


Figure 5.7: The density distribution colourmaps at six different times for $\phi = \frac{-\pi}{2}$ and $x = 1$. $\gamma = 0.1$, $g_1 = 0.2$, and $g_2 = 0.9$.

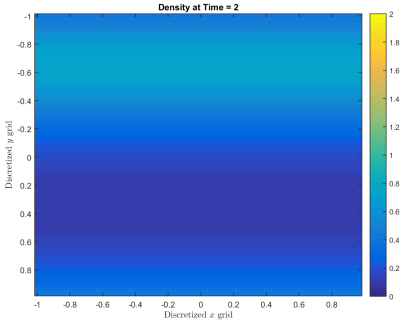
While the shift in Figures 5.6 and 5.7 is present, it is difficult to see due to the step size and the speed. In order to see the shift more clearly, we can set the value of γ to be 1. In Figures 5.8 and 5.9.



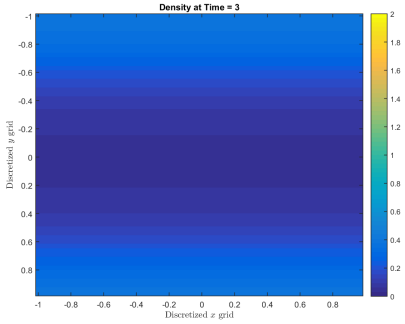
(a) Initial density $\exp(\cos(\pi y))$.



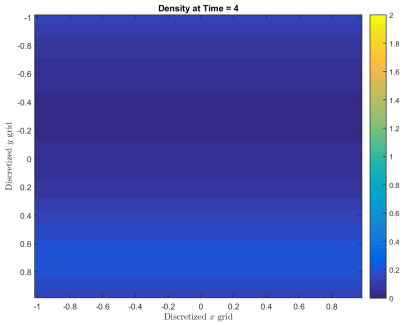
(b) Density after 10 time steps.



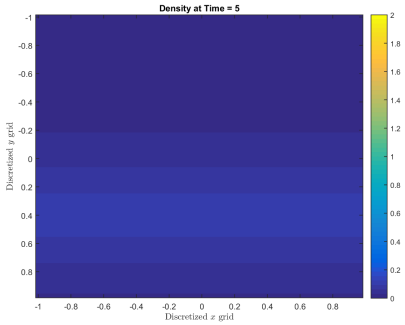
(c) Density after 20 time steps.



(d) Density after 30 time steps.



(e) Density after 40 time steps.



(f) Density after 50 time steps.

Figure 5.8: The density distribution colourmaps at six different times for $\phi = \frac{-\pi}{2}$. $\gamma = 1$, $g_1 = 0.2$, and $g_2 = 0.9$.

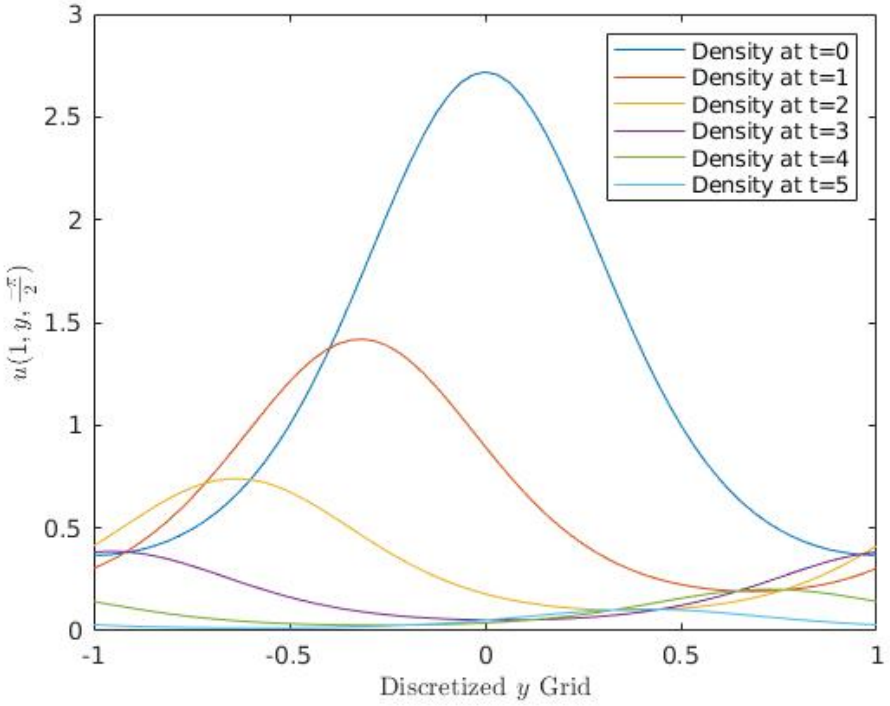
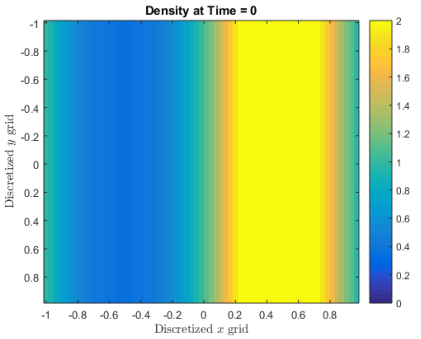
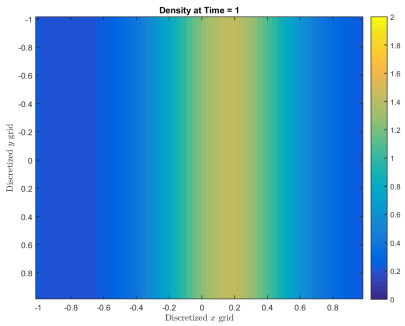


Figure 5.9: The density distribution colourmaps at six different times for $\phi = \frac{-\pi}{2}$ and $x = 1$. $\gamma = 1$, $g_1 = 0.2$, and $g_2 = 0.9$.

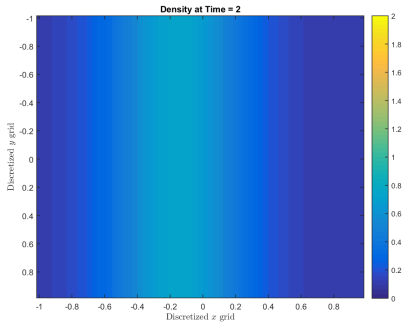
The shift is far more noticeable in Figures 5.8 and 5.9 with every time step. The shift is multiplied by speed γ and by the step size h, and as γ increases, the shift is no longer multiplied by 0.01, but by 0.1. Similar results can be seen when the initial condition depends only on the spatial varaiable x. If we use an initial condition with x, say $\exp(\sin(\pi x))$, and leave γ set to 1, we will see a shift when we set ϕ to $-\pi$ as seen in Figures 5.10 and 5.11.



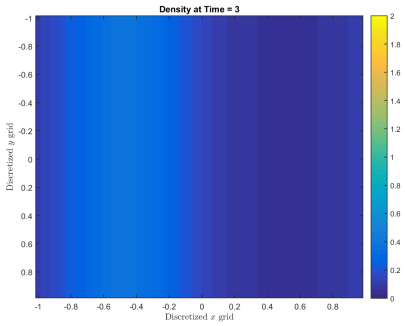
(a) Initial density $\exp(\sin(\pi x))$



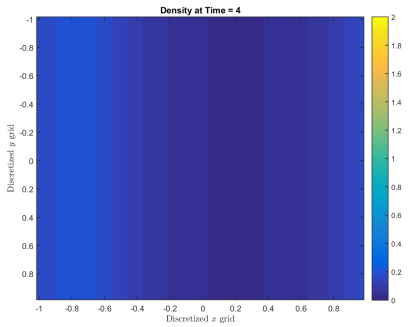
(b) Density after 10 time steps.



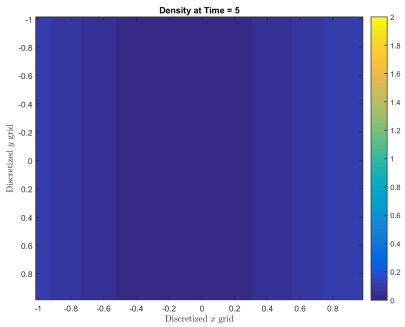
(c) Density after 20 time steps.



(d) Density after 30 time steps.



(e) Density after 40 time steps.



(f) Density after 50 time steps.

Figure 5.10: The density distribution colourmaps at six different times for $\phi = -\pi$. $\gamma = 1$, $g_1 = 0.2$, and $g_2 = 0.9$.

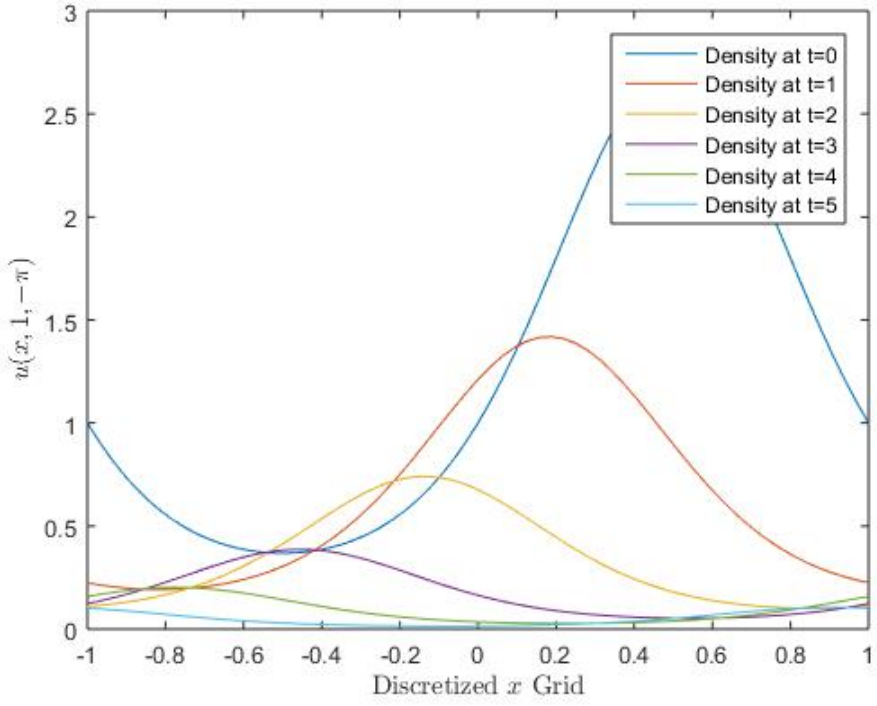


Figure 5.11: The density distribution colourmaps at six different times for $\phi = -\pi$ and $y = 1$. $\gamma = 1$, $g_1 = 0.2$, and $g_2 = 0.9$.

This is because the solution to this system is

$$u = F(x - \gamma\cos(\phi)t, y - \gamma\sin(\phi)t)e^{-tG} = F(x - \gamma\cos(\phi)t)e^{-tG},$$

which will have the largest shift when ϕ is $-\pi$, 0 or π.

5.4 Implementation of the Non-Linear Terms

The remaining terms that need to be computed in order to use ETD2RK are

$$-\frac{1}{2}g_2\tanh(\Gamma - 2)u + \int_{-\pi}^{\pi} w(\phi - \phi')\frac{1}{2}g_2\tanh(\Gamma - 2)u(X, \phi', t)d\phi',$$

where

$$\Gamma = \sum_{j=a,r,al} q_j \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_j^d K_j^o u(S,\theta,t) d\theta dS.$$

In order to compute these non-linear terms in our code, we must first compute Γ. As Γ is just the sum of three convolutions, it makes sense to compute each convolution in Fourier space using our Fourier transform (3.2.4), then using our inverse Fourier transform (3.2.5) to obtain Γ in real space.

As the hyperbolic tangent of Γ and u will both be functions of X and ϕ (or ϕ' in the case of the second non-linear term), it makes sense to multiply these two functions in real space. Then the first term is a constant multiplied by this number and the second term is then a convolution between this product and w. We ultimately want both terms in Fourier space to update the solution, as our linear term updates in Fourier space.

5.4.1 Convolution of Kernels used to Obtain Γ

Γ is the sum of three convolution terms. With respect to attraction and repulsion, the terms in the sum are

$$q_{a,r} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_{a,r}^d (X - S) K_{a,r}^o (S; X, \phi) u(S,\theta,t) d\theta dS.$$

Recall that K_j^o relies on the absolute value of the difference between S and X, so for simplicity's sake, both K_j^o and K_j^d depend on $X - S$. If we multiply these two kernels in real space, we will end up with an integral of the form

$$q_{a,r} \int_{\mathbb{R}^2} \int_{-\pi}^{\pi} K_{a,r}(X - S, \phi) u(S,\theta,t) d\theta dS.$$

If we substitute u for (3.2.5), we get

$$q_{a,r}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi} K_{a,r}(X-S,\phi)\frac{1}{KLM}\sum_{k_1}\sum_{l_1}\sum_{m_1}\hat{u}_{k_1,l_1,m_1}(t)\Phi_{k_1,l_1,m_1}(s_x,s_y,\theta)\,\mathrm{d}\theta\,\mathrm{d}S$$
$$=\sum_{k_1}\sum_{l_1}\sum_{m_1} q_{a,r}\hat{u}_{k_1,l_1,m_1}(t)\int_{\mathbb{R}^2} K_{a,r}(X-S,\phi)\frac{1}{KLM}e^{ik_1s_x+il_1s_y}\int_{-\pi}^{\pi} e^{im_1\theta}\,\mathrm{d}\theta\,\mathrm{d}S. \tag{5.4.1.1}$$

The term $\int_{-\pi}^{\pi} e^{im_1\theta}\,\mathrm{d}\theta = \delta_{m_1,0}$ is a Kronecker delta distribution, so $m_1 = 0$ is the only non-trivial case. Also, if we allow $Z = X - S$ the term

$$\int_{\mathbb{R}^2} K_{a,r}(X-S,\phi)e^{ik_1s_x+il_1s_y}dS$$

becomes

$$\int_{\mathbb{R}^2} K_{a,r}(Z,\phi)e^{ik_1(x-z_x)+il_1(y-z_y)}dZ.$$

We can remove $e^{ik_1x+il_1y}$ from the double integral and the remaining term is equal to the Fourier transform of the kernel multiplied by the constant $\frac{4}{n^2}$. (5.4.1.1) becomes

$$\frac{1}{KLM}\sum_{k_1}\sum_{l_1}\frac{4}{n^2}q_{a,r}\hat{u}_{k_1,l_1,0}(t)\widehat{K_{a,r}}.$$

As the term is equal to the inverse Fourier transform of the product of the Fourier transforms, both the attraction and repulsion terms are convolutions in space.

The alignment term in the sum is

$$q_{al}\int_{\mathbb{R}}\int_{-\pi}^{\pi} K_{al}^{d}(X-S)K_{al}^{o}(\phi'-\theta)u(S,\theta,t)d\theta dS.$$

If we substitute u for (3.2.5), we get

$$q_{al}\int_{\mathbb{R}^2}\int_{-\pi}^{\pi}K_{al}^{d}(X-S)K_{al}^{o}(\phi'-\theta)\frac{1}{KLM}\sum_{k_1}\sum_{l_1}\sum_{m_1}\hat{u}_{k_1,l_1,m_1}(t)\Phi_{k_1,l_1,m_1}(s_x,s_y,\theta)d\theta dS$$
$$=\frac{1}{KLM}\sum_{k_1}\sum_{l_1}\sum_{m_1}q_{al}\hat{u}_{k_1,l_1,m_1}(t)\int_{\mathbb{R}^2}K_{al}^{d}(X-S)e^{ik_1s_x+il_1s_y}dS\int_{-\pi}^{\pi}K_{al}^{o}(\phi'-\theta)e^{im_1\theta}d\theta. \tag{5.4.1.2}$$

If we allow $Z=X-S$ the term

$$\int_{\mathbb{R}^2}K_{al}^{d}(X-S)e^{ik_1s_x+il_1s_y}dS$$

becomes

$$\int_{\mathbb{R}^2}K_{al}^{d}(Z)e^{ik_1(x-z_x)+il_1(y-z_y)}dZ.$$

We can remove $e^{ik_1x+il_1y}$ from the double integral and the remaining term is equal to the Fourier transform of the kernel multiplied by the constant $\frac{4}{n^2}$.

Similarly, if we allow $\eta=\phi'-\theta$ the term

$$\int_{-\pi}^{\pi}K_{al}^{o}(\phi'-\theta)e^{im_1\theta}d\theta$$

becomes

$$\int_{-\pi}^{\pi}K_{al}^{o}(\eta)e^{im_1(\phi'-\eta}d\eta.$$

We can remove $e^{im_1(\phi')}$ from the integral and the remaining term is equal to the Fourier transform of the kernel multiplied by the constant $\frac{2\pi}{m}$. (5.4.1.2) becomes

$$\frac{1}{KLM}\sum_{k_1}\sum_{l_1}\sum_{m_1}\frac{2\pi}{m}q_{al}\hat{u}_{k_1,l_1,m_1}(t)\frac{4}{n^2}\widehat{K^d}_{al}\widehat{K^o}_{al}. \tag{5.4.1.3}$$

As the term is equal to the inverse Fourier transform of the product of the Fourier transforms, the alignment term is a convolution in space and angle. If we substitute

these terms into Γ, we get Γ to be an $n \times n \times m$ array in real space.

5.4.2 Utilization of Γ and the Non-Linear Terms in the Code

To calculate the first of the two non-linear terms, we need only take the hyperbolic tangent of $\Gamma - 2$ and then multiply it by the density u and the constant $g_2/2$. As we want all non-linear terms in Fourier space in order to update the solution, we would take the 3-dimensional Fourier transform of the product.

To calculate the second of the two non-linear functions, first we compute the hyperbolic tangent of $\Gamma - 2$ and multiply the result by the density. If we call this product P, notice that the resulting term has the form

$$\frac{1}{2} g_2 \int_{-\pi}^{\pi} w(\phi - \phi') P(X, \phi', t) d\phi'$$

which is a convolution with respect to angle. Therefore, if we take the Fourier transform of P, we can multiply it by the Fourier transform of $\frac{2\pi}{m} w$.

With both terms in Fourier space, we can add them together to get the complete non-linear component of our equation. This non-linear term is computed twice in the code. Once using the u that is initialized at the beginning of the time step for ETD2RK, and then again using a_n as the density. The difference between these two non-linear terms is then taken and used to compute each new density at the end of each time step.

5.5 Testing the Non-Linear Terms

It is much more difficult to test a solution including the non-linear terms in the system analytically than it was to test the solution to the linear terms. That being said, it is possible to use some intuition to see if the terms are operating as expected.

When coding the convolution components of Γ, all three are computed separately. That is, the attraction, repulsion, and alignment terms are all nested separately within the code. Because of this, we are able to systematically remove certain components and test the effects of each term separately.

The density shown at any point (x, y) is an approximation of the density integrated over all angles at (x, y). That is,

$$u(x, y) = \sum_{j=1}^{m} \frac{2\pi}{m} u(x, y, \phi(j)),$$

where ϕ is discretized from $-\pi$ to π with m points.

The arrows that appear on this graph are found using the Kuramoto order parameters [10]. They represent the average direction of individuals located at the position (x, y) on the graph. To obtain the arrows, first we take the weighted average of all complex numbers $e^{i\phi(j)}$ for every descritized ϕ_j from $-\pi$ to π. Each $e^{i\phi(j)}$ is weighted with the density of individuals at the position (x, y) in question, moving in direction $\phi(j)$

$$z_{(x,y)} = \frac{1}{m} \sum_{j=1}^{m} u(x, y, \phi(j)) e^{i\phi(j)}.$$

This equation results in a complex number z. We take the argument of z to be the angle that the animals are travelling in on average. We take $|z|$ to be the length of the arrow.

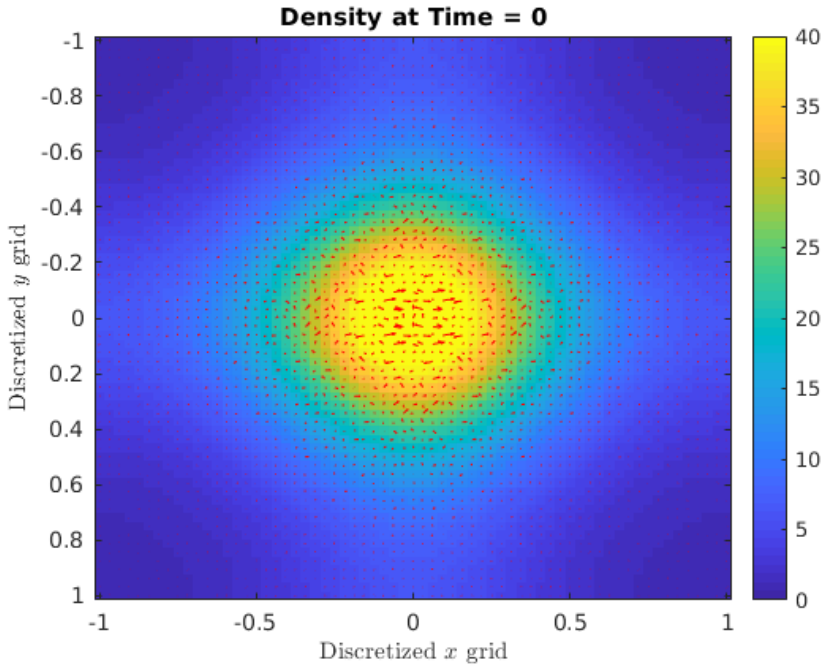
5.5.1 Allow only the Alignment Term to have an Effect

If we eliminate the attraction and repulsion terms by setting $q_a = q_r = 0$ and q_{al} to be non-zero ($q_{al} = 2$), Γ only depends on the alignment term and we expect that the results should show groups of animals moving uniformly, based on the overwhelming force of alignment.

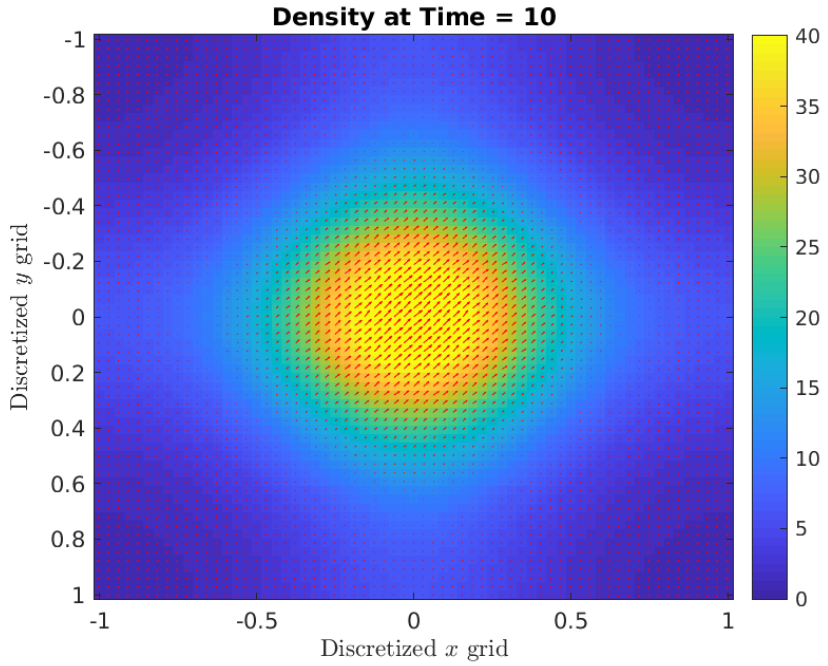
Figure 5.12 shows a group of animals with the initial distribution $u_0 = \exp(\cos(\pi x) +$

$cos(\pi y))$. In order to emphasize the influence of the non-linear terms, we set the speed of animals, γ, to zero. The parameters g_1 and g_2 are set to 0.2 and 0.9 respectively. It is important to have a large g_2, because it weighs the bias from the non-linear terms.

Notice that all animals appear to align over time. Unsurprisingly, the spatial distribution of animals does not appear to change over time, as the speed the animals are moving at is set to zero.



(a) Initial density $\exp(\cos(\pi x) + cos(\pi y))$.



(b) Density after 100 time steps.

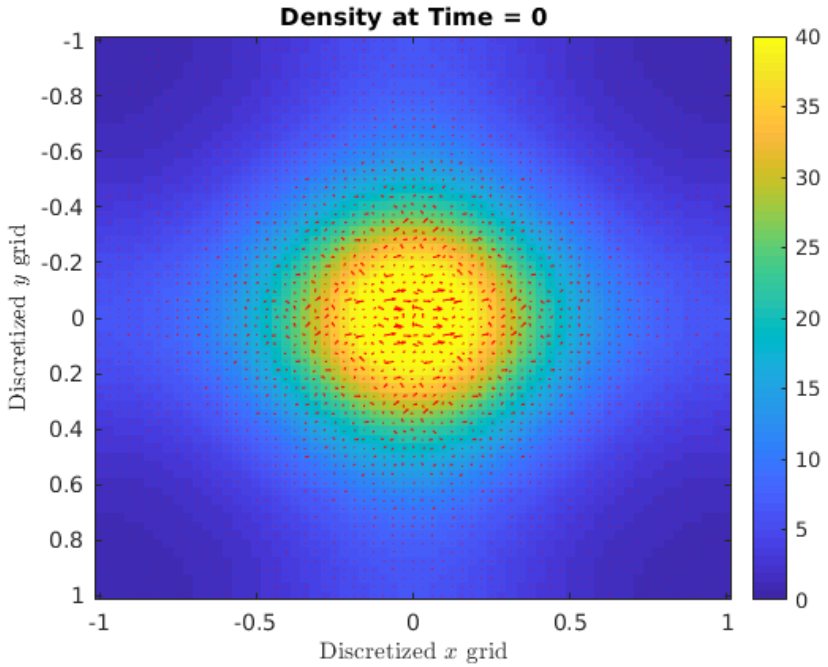
Figure 5.12: The colourmap of density at two different times. $\gamma = 0$, $g_1 = 0.2$, $g_2 = 0.9$, $h = 0.1$, and $q_{al} = 2$.

5.5.2 Allow only the Attraction Term to have an Effect

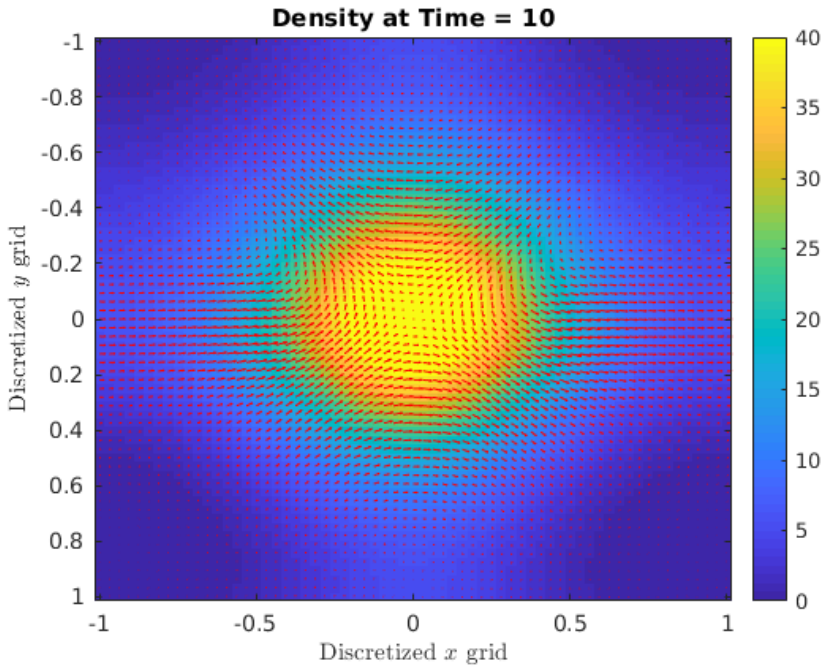
If we exclude the alignment and repulsion terms by setting $q_{al} = q_r = 0$ and q_a to be non-zero ($q_a = 1$), Γ only contains the attraction term and we expect that the resulting patterns should show animals moving towards densely populated areas on the graph, based on the overwhelming force of attraction.

Figure 5.13 shows the average density of animals with the initial distribution $u_0 = \exp(\cos(\pi x) + cos(\pi y))$. In similar fashion to our alignment test, we set $g_1 = 0.2$ and $g_2 = 0.9$, but unlike the previous test, we cannot set the speed of animals to zero. If animals are attracted to one another, they will move towards one another. In order to see this behaviour, the animals must be able to move.

Over time, the density of animals becomes more concentrated around the initial highest point of density.



(a) Initial density $\exp(\cos(\pi x) + cos(\pi y))$.



(b) Density after 100 time steps.

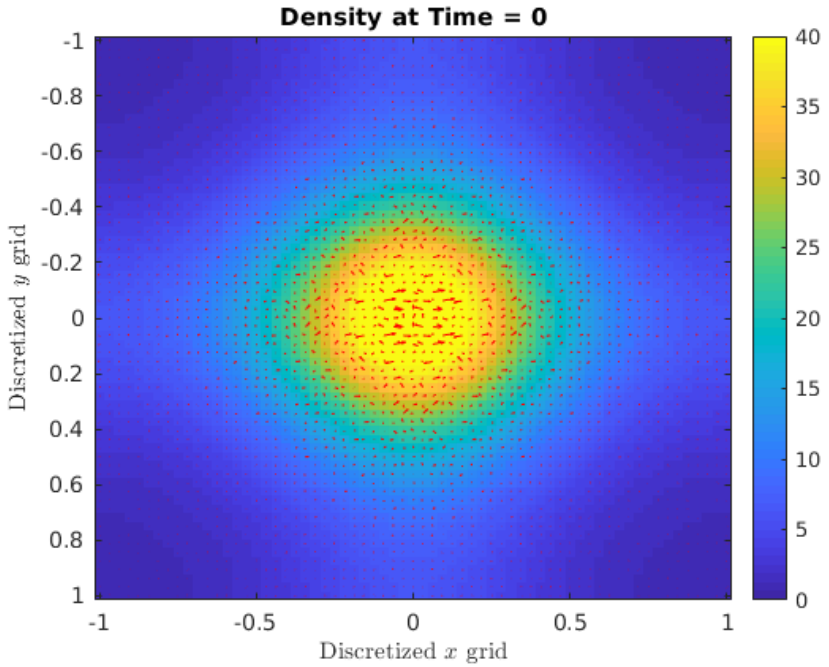
Figure 5.13: The colourmap of density at two different times. $\gamma = 0.1$, $g_1 = 0.2$, $g_2 = 0.9$, $h = 0.1$, and $q_a = 1$.

5.5.3 Allow only the Repulsion Term to have an Effect

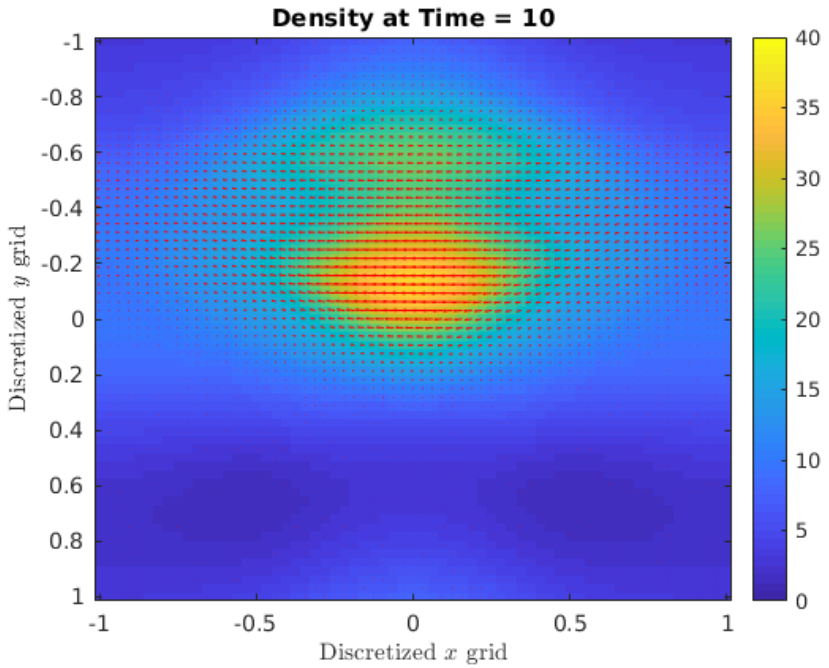
If we exclude the alignment and attraction terms by setting $q_{al} = q_a = 0$ and q_r to be non-zero ($q_r = 0.5$), Γ only contains the repulsion term and we expect that the resulting patterns should show animals moving away from densely populated areas on the graph, based on the overwhelming force of repulsion.

Figure 5.14 shows the average density of animals with the initial distribution $u_0 = \exp(\cos(\pi x) + cos(\pi y))$. As in our previous two tests, $g_1 = 0.2$ and $g_2 = 0.9$ and as in the attraction test, speed must be non-zero, in order to demonstrate any behaviour of repulsion.

Over time, the the highest point of density disperses as the animals a repelled from one another.



(a) Initial density $\exp(\cos(\pi x) + cos(\pi y))$.



(b) Density after 100 time steps.

Figure 5.14: The colourmap of density at two different times. $\gamma = 0.1$, $g_1 = 0.2$, $g_2 = 0.9$, $h = 0.1$, and $q_r = 0.5$.

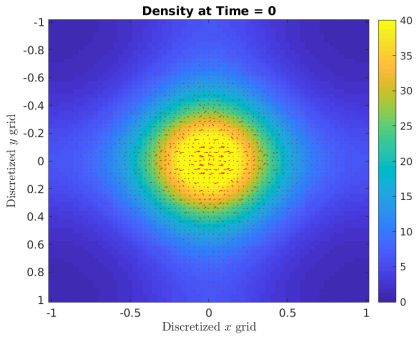
5.6 Results

In order to compute a solution to the system using as much information and accuracy as possible, we have the ability to include any combination of the attraction, repulsion, and alignment terms as are applicable to the particular subject in question. Figure 5.15 includes the contribution of all three components operating on a randomly distributed group of animals.

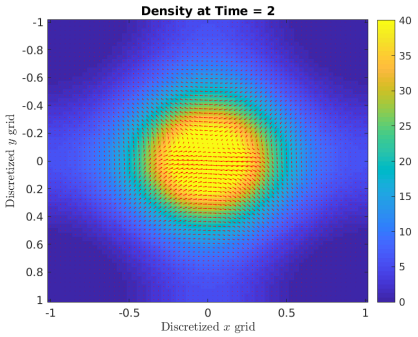
Figure 5.15 uses the same initial condition as the three non-linear test cases, $u_0 = \exp(\cos(\pi x) + cos(\pi y))$, with $g_1 = 0.2$, $g_2 = 0.9$, $\gamma = 0.1$, and step-size $h = 0.1$. To test the non-linear components, we set $q_{al} = 2$, $q_a = 1$, and $q_r = 0.5$.

At different times throughout the simulation, all non-linear components appear to take effect on the distribution of animals. At times, the animals appear to align, especially at Time = 2. The attraction and repulsion influences appear to balance one another to some extent, as there is no apparent overwhelming migration to or away from the initial high density areas.

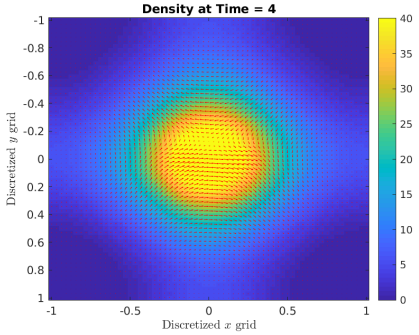
Simulations can be run with different initial conditions, so long as they are periodic in all three variables. Also, the strength parameters q_a, q_r and q_{al} can be altered to increase or decrease the impact of the attraction, repulsion, and alignment terms respectively.



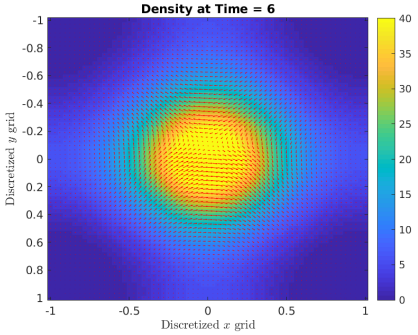
(a) Initial density $\exp(\cos(\pi x) + cos(\pi y))$.



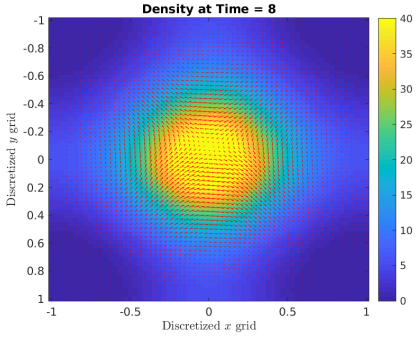
(b) Density after 20 time steps.



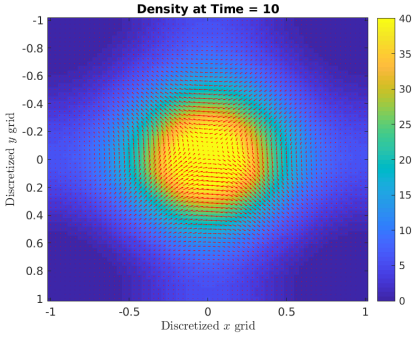
(c) Density after 40 time steps.



(d) Density after 60 time steps.



(e) Density after 80 time steps.



(f) Density after 100 time steps.

Figure 5.15: The colourmap of density at two different times. $\gamma = 0.1$, $g_1 = 0.2$, $g_2 = 0.9$, $h = 0.1$, $q_{al} = 2$, $q_a = 1$, and $q_r = 0.5$.

CHAPTER 6

Conclusions

Implementing non-linearity into this system is important, but does not come without a cost. In Fetecau's paper, there was a great deal of importance placed upon the dependence $\theta - \phi'$ in the probability function w. Fetecau defines $w(\phi' - \phi, \phi' - \theta)$ as $g_\alpha(\phi' - \phi - v(\phi' - \theta))$, where $v(\phi' - \theta)$ is some function that has its own strength coefficient κ. This constant dictates the amount that the neighbour's direction influences the individual's probability of turning.

This dependence indicates that the closer a neighbour's direction is to the possible new direction of an individual, the less likely he is to change directions. If an animal is moving in a direction different from the neighbour's by $180°$, that is when he is most likely to readjust his direction to be more similar to that of the neighbour's.

In Fetecau's simulations, he uses the variable κ frequently to increase the effects of alignment, attraction, and repulsion based on the neighbour's angular direction. As our system effectively has κ set to 0, we cannot control the strength of these forces as easily.

This does not to imply, however, the removal of the dependence of $\phi' - \theta$ from w has eliminated the dependence of the neighbour's direction from the system altogether. Indeed, θ appears in all three convolutions within Γ, and appears in the orientation kernel for alignment.

The addition of bounded non-linearity to λ has further influenced the results. It makes practical sense to bound λ, because there is only so much information that an individual can take in from neighbours.

It should be noted, removing this dependence from w does not stop the system from forming patterns. Though not shown in this paper, tests with random initial conditions form distinct patterns. Depending on how strong the forces are, animals often seem to take on saddle like formations around areas of high density and low density, and move in vortices in areas of neither high nor low density.

In order for our system to be applicable to some type of animal with the changes we made to the probability function, we must look at animals that either can't see the direction that their neighbours are moving in, or are indifferent to the information. For instance, the phylum Cnidaria contains jellyfish, an animal that has no head, front or back. While the aggregation of jellyfish depends greatly on exterior factors [7] and may not have a high correlation with this model, Cnidaria, with their seeming lack of ability to sense neighbours' directions, appear to be prime candidates for the adjustments made to this model.

Similarly, the model could be used to explain the motion of cancer cells. It is known that cancer and other cells appear to aggregate to form clusters [5, 9, 11], and it may be worth exploring if this model produces realistic simulations of cell grouping patterns. As cells have neither a front nor a back, it is possible that the direction cells are moving in may be less perceptible to neighbouring cells, meaning the changes made here to w could still predict cell migration patterns.

We believe, while the model may have lost some relevance with the adaptations

to w, it is still a very good predictor of animal motion and animal group formation.

Further steps can be taken with this study. With the addition of the non-linear terms, more analysis can be run to test the accuracy of the results. For instance, in order to test the alignment term, we should test the correlation length of the individuals in the domain. In order to test the attraction and repulsion terms, we can examine the cluster radii in order to explore how the animals move to or from areas of high density.

The only non-linear approximation of λ used in this paper depends on a hyperbolic tangent, but can be replaced with any bounded and increasing function. It would be interesting to see the patterns that form and the impact of the non-linear terms if the upper bound of λ was larger.

Also, a deeper look into the types of animals and cells that rely less on the direction their neighbours are moving in than the mutual distance between animals would be beneficial in distinguishing the subjects this adapted system could model.

Bibliography

[1] M. Chen. On the solution of circulant linear systems. *Society for Industrial and Applied Mathematics Journal*, 24:668–683, 1987.

[2] S. M. Cox and P. C. Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176:430–455, 2002.

[3] R. Eftimie. Hyperbolic and kinetic models for self-organized biological aggregations and modement: A brief review. *Mathematical Biology*, 65:35–75, 2012.

[4] R. Eftimie, G. de Vries, M. A. Lewis, and F. Lutscher. Modeling group formation and activity patterns in self-organizing collectives of individuals. *Bulletin of Mathematical Biology*, 69:1537–1565, 2007.

[5] T. Elsdale and J. Bard. Collagen substrata for studies on cell behavior. *Journal of Cell Biology*, 54:626–637, 1972.

[6] R. C. Fetecau. Collective behavior of biological aggregation in two dimensions: A nonlocal kinetic model. *Mathematical Models and Methods in Applied Sciences*, 21:1539–1569, 2011.

[7] S. Fossette, A. C. Gleiss, J. Chalumeau, T. Bastian, C. D. Armstrong, S. Vandenabeele, M. Karpytchev, and G. C. Hays. Current-oriented swimming by jellyfish and its role in bloom maintenance. *Current Biology*, 25:342–347, 2015.

[8] S. MacNamara and G. Strang. Operator splitting. In R. Glowinski, S. Osher, and W. Yin, editors, *Splitting Method in Communications, Imaging, Science, and Engineering*, chapter 3, pages 95–114. Springer International Publishing Switzerand, 2016.

[9] E. Méhes and T. Vicsek. Collective motion of cells: From experiments to models. *Integrative Biology*, 6:831–854, 2014.

[10] D. Q. Nykamp. *The Idea of Synchrony of Phase Oscillators*, June 2018. https://mathinsight.org/synchrony_phase_oscillators_idea.

[11] A. Puliafito, A. De Simone, G. Seano, P. A. Gagliardi, L. Di Blasio, F. Chianale, A. Gamba, L. Primo, and A. Celani. Three-dimensional chemotaxis-driven aggregation of tumor cells. *Scientific Reports*, 5, 2015.

[12] S. J. Simpson, A. R. McCaffery, and B. F. Hägele. A behavioural analysis of phase change in the desert locust. *Biological Reviews*, 74:461–480, 1999.

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

Appendices

APPENDIX A

Pseudo Code for ETD2RK

Algorithm 1 ETD2RK

procedure COMPUTES UPDATE STEP FOR DENSITY (Input: γ, u_0, $\widehat{u_0}$, h, g_1, g_2, k (index for x), l (index for y),$\hat{w}$)
  for n=0:$\frac{T}{h}$ **do**
    Compute the NLT_n using Algorithm 6 and the initial condition u_0
    if n =0 **then**
      $\widehat{u_{n+1}} \rightarrow \widehat{u_n}$
      $t_n + h \rightarrow t_n$
    else
      $\widehat{u_0} \rightarrow \widehat{u_n}$
      $0 \rightarrow t_n$
    for k=-n/2+1:n/2 **do**
      for l=-n/2+1:n/2 **do**
        $NLT(k, l, :) \rightarrow NLT_\phi$
        $\widehat{u_0(k, l, :)} \rightarrow \widehat{u_{\phi 0}}$
        size$(\widehat{u_{\phi 0}}) \rightarrow M$
        compute the linear operator with Algorithm 3
        if k=0 and l=0 **then**
          exclude first row and column of L
          compute e^L using Algorithm 4
          compute M_1 using Algorithm 5
          compute LUP decomposition of L to get L_1, U_1, P_1
          if k=0 and l=0 **then**
            $a_0(k, l, :) = \widehat{u_{\phi 0}}$
            for n=1:$\frac{T}{h}$ **do**
              $b = Le^L\widehat{u_{\phi 0}} + M_1\widehat{NLT_n}$
              $d = P_1 b$
              $y = L_1 \backslash d$
              $a_n(k, l, :) = U_1 \backslash y$
          else
            $b(= Le^L\widehat{u_{\phi 0}} + M_1\widehat{NLT_n}$
            $d = P_1 b$
            $y = L_1 \backslash d$
            $a_n(k, l, :) = U_1 \backslash y$

Algorithm 2 Continuation of ETD2RK

procedure (STILL WITHIN LOOP OVER n)COMPUTES UPDATE STEP FOR DENSITY (Input: γ, u_0, $\widehat{u_0}$, h, g_1, g_2, k (index for x), l (index for y),$\hat{w}$)
    take the inverse Fourier transform of a_n
    Compute NLT_{n+1} using Algorithm 6 and ifftna_n
    for k=-n/2+1:n/2 **do**
        for l=-n/2+1:n/2 **do**
            $NLTn+1(k,l,:) \rightarrow NLT\phi_{n+1}$
            $\widehat{a_n(k,l,:)} \rightarrow \widehat{a_{n\phi}}$
            compute the linear operator with Algorithm 3
            if k=0 and l=0 **then**
                exclude first row and column of L
                compute e^L using Algorithm 4
                compute M_2 using Algorithm 6
                compute LUP decomposition of L^2 to get L_2, U_2, P_2
                if k=0 and l=0 **then**
                $u_new(k,l,:) = \widehat{a_{n0}}$
                for n=1:$\frac{T}{h}$ **do**
                $b = L^2 e^L \widehat{a_{n\phi}} M_2(NLT\phi_{n+1} - NLT_\phi)/h$
                $d = P_2 b$
                $y = L_2 \backslash d$
                $\widehat{u_{n+1}}(k,l,:) = U_2 \backslash y$
                else
                $b = L^2 e^L \widehat{a_{n\phi}} M_2(NLT\phi_{n+1} - NLT_\phi)/h$
                $d = P_2 b$
                $y = L_2 \backslash d$
                $\widehat{u_{n+1}}(k,l,:) = U_2 \backslash y$
            Take the inverse Fourier transform of u_{n+1}

Algorithm 3 L

procedure COMPUTE THE LINEAR OPERATOR(Input: γ, g_1, g_2, k, l,m,$\hat{w}$)
    $(g_1 + \frac{1}{2}g_2) \rightarrow G$
    zeros(m,1)$\rightarrow F$
    for i=1:m **do**
        $-G + G\widehat{w(i)} \rightarrow F(i)$
    diag$F \rightarrow P$
    zeros$(1, M) \rightarrow C$
    $F \rightarrow C(1)$
    $\frac{\gamma}{2}(ik - l) \rightarrow C(2)$
    $\frac{\gamma}{2}(ik + l) \rightarrow C(M)$
    turn vector C into a circulant matrix K
    $L = P + K$

Algorithm 4 e^L

1: **procedure** COMPUTE EXPONENTIAL OF THE LINEAR OPERATOR(Input: P, K, h)
2: $h * P \rightarrow newP$
3: $e^{newP)} \rightarrow A$
4: size(K)$\rightarrow i$
5: Fourier matrix of size i to F
6: $h * K \rightarrow newK$
7: $F * newK * \frac{1}{i}F^{con} \rightarrow diag$
8: $e^{diag} \rightarrow B$
9: $\frac{1}{i}F^{con} * B * F \rightarrow D$
10: $(A * D + D * A)/2 \rightarrow e^L$

Algorithm 5 M_1

1: **procedure** (Input: L, e^L, m)
2: $M_1 = e^L - I_m$

Algorithm 6 M_2

1: **procedure** (Input: L, e^L, h, m)
2: $M_2 = e^L - (I_m + hL)$

APPENDIX B

Pseudo Code for Nonlinear Terms

Algorithm 7 Nonlinear

procedure COMPUTES NLT IN FOURIER SPACE (Input: u, q_{al}, q_a, q_r, attraction_kernel from Algorithm 7, repulsion_kernel from ALgorithm 8, alignment_kernel_d from Algorithm 9, alignment_kernel_o from Algorithm 10, n, m, g_2, $\hat{w}$)
    zeros(n,n,m)$\rightarrow$ alignment
    zeros(n,n,m)$\rightarrow$ attraction
    zeros(n,n,m)$\rightarrow$ repulsion
    zeros(n,n,m)$\rightarrow \Gamma$
    Take the 3-d Fourier transform of u for $\hat{u}$
    for j=1:m **do**
        attraction_kernel. $* \widehat{u(:,:,1)} \rightarrow$ mult1
        ifft2 mult1 $\rightarrow$ convo1
        convo1$\rightarrow$ attraction(:,:,j)
        repulsion_kernel. $* \widehat{u(:,:,1)} \rightarrow$ mult2
        ifft2 mult2 $\rightarrow$ convo1
        convo1$\rightarrow$ repulsion(:,:,j)
    alignment_kernel_d.*alignment_kernel_o.*$\hat{u} \rightarrow$ mult3
    ifftn mult3 $\rightarrow$ alignment
    q_a*attraction+q_r*repulsion+$q_a l$*alignment $\rightarrow \Gamma$
    $-0.5 * g_2 * tanh(\Gamma - 2\text{ones}). * u \rightarrow$ FirstTerm
    fftn(FirstTerm) $\rightarrow$ FirstTerm_hat

    $tanh(\Gamma - 2\text{ones}) \rightarrow$ hyptan
    hyptan.* u $\rightarrow$ product
    fftn(product) $\rightarrow$ product_hat zeros(n,n,m) $\rightarrow$ SecondTerm_hat
    for k=1:n **do**
        for l=1:n **do**
            for j=1:m **do**
                $0.5 * g_2 * \hat{w}(j)$*product_hat(k,l,j) $\rightarrow$ SecondTerm_hat(k,l,j)

    FirstTerm_hat + SecondTerm_hat $\rightarrow$ NLT

Algorithm 8 kernel_a

procedure COMPUTES THE KERNEL FOR ATTRACTION IN FOURIER SPACE (Input: m_a, d_a, n,m)
    $\pi * m_a(m_a * \exp(-d_a^2/m_a^2) + \sqrt{\pi} * d_a + \sqrt{\pi} * d_a * erf(d_a/m_a)) \rightarrow$ A
    [-1,1) with n points$\rightarrow$ s_x
    [-1,1) with n points$\rightarrow$ s_y
    $[-\pi, \pi)$ with n points$\rightarrow \phi$
    compute K_d using Algorithm 10
    zeros(n,n)$\rightarrow$ K_o
    zeros(n,n)$\rightarrow$ K
    for j=1:m **do**
        for k=1:n **do**
            for l=1:n **do**
                if $s_x = 0$ and $s_y = 0$ **then** $0 \rightarrow \psi$
                else $\arccos(u/\sqrt{s_x^2 + s_y^2}) \rightarrow \psi$
                $(1/2\pi) * (-\cos(\phi(j) - \psi) + 1) \rightarrow$ K_o(k,l)

                K_d(k,l)*K_o(k,l) $\rightarrow$ K(k,l)
        K_hat(:,:,j)=$(\frac{2}{n})^2$*fft2(K)

Algorithm 9 kernel_r

procedure COMPUTES THE KERNEL FOR REPULSION IN FOURIER SPACE (Input: m_r, d_r, n,m)
    $\pi * m_r(m_r * \exp(-d_r^2/m_r^2) + \sqrt{\pi} * d_r + \sqrt{\pi} * d_r * erf(d_r/m_r)) \rightarrow$ A
    [-1,1) with n points$\rightarrow$ s_x
    [-1,1) with n points$\rightarrow$ s_y
    $[-\pi, \pi)$ with n points$\rightarrow \phi$
    compute K_d using Algorithm 10
    zeros(n,n)$\rightarrow$ K_o
    zeros(n,n)$\rightarrow$ K
    for j=1:m **do**
        for k=1:n **do**
            for l=1:n **do**
                if $s_x = 0$ and $s_y = 0$ **then** $0 \rightarrow \psi$
                else $\arccos(u/\sqrt{s_x^2 + s_y^2}) \rightarrow \psi$
                $(1/2\pi) * (\cos(\phi(j) - \psi) + 1) \rightarrow$ K_o(k,l)

                K_d(k,l)*K_o(k,l) $\rightarrow$ K(k,l)
        K_hat(:,:,j)=$(\frac{2}{n})^2$*fft2(K)

Algorithm 10 alignment_kernel_d

procedure COMPUTES THE DISTANCE KERNEL FOR ALIGNMENT IN FOURIER SPACE. IS ALSO USED TO COMPUTE THE DISTANCE KERNEL FOR ATTRACTION AND REPULSION, WITH THEIR RESPECTIVE m AND d. (Input: $m_a l$, $d_a l$, n)
    $\pi * m_a l(m_a l * \exp(-d_a l^2/m_a l^2) + \sqrt{\pi} * d_a l + \sqrt{\pi} * d_a l * erf(d_a l/m_a l)) \rightarrow$ A
    [-1,1) with n points$\rightarrow$ s_x
    [-1,1) with n points$\rightarrow$ s_y
    zeros(n,n)$\rightarrow$ K
    for k=1:n **do**
        for l=1:n **do**
            $\sqrt{s_x^2 + s_y^2} \rightarrow$ length
            $1/A * \exp(-(length - d)^2/m^2) \rightarrow$ K_d(k,l)
    K_hat=$(\frac{2}{n})^2$*fft2(K)

Algorithm 11 alignment_kernel_o

procedure COMPUTES THE ORIENTATION KERNEL FOR ALIGNMENT IN FOURIER SPACE (Input: m)
    $[-\pi, \pi)$ with m points$\rightarrow \theta$
    zeros(1,m)$\rightarrow$ K
    for j=1:m **do**
        $(1/2\pi) * (-\cos(\theta(j)) + 1) \rightarrow$ K(j)
    K_hat=$\frac{2*\pi}{m}$)*fft(K)