

C.H.S. Hamster

Two Scale Dependent Feedback as a Model for Pattern Formation in Ecology

Master Thesis
22-03-2016

Supervisor:
Dr. V. Rottschäfer



Mathematisch Instituut,
Universiteit Leiden

Abstract

In this thesis we will discuss several models to explain pattern formation in mussel beds. The focus will lie on explaining the observed wavelength selection. We will explain why the Cahn-Hilliard model as derived in [19] does not show wavelength selection. Then we will explain different models based on biological experiments using two scale dependent feedback, i.e. the assumption that mussels cooperate and aggregate at short length scales but compete at longer length scales and derive nonlinear diffusion equations from these models. We will show numerically that the solution of the derived nonlinear diffusion equations are in good agreement with the experimentally observed patterns and that the equations evolve towards stationary solutions with a fixed wavelength.

Contents

Introduction	1
1 The Cahn-Hilliard equation	5
1.1 Derivation of the Cahn-Hilliard equation	5
1.2 Some theory on the Cahn-Hilliard equation	7
1.3 Comparison with the biological data	9
1.4 Validity of the CH-equation	11
2 Modelling two Scale Dependent Feedback	14
2.1 Individual Based Models	15
2.2 Fokker-Planck equation for the IBM by Liu et al.	17
2.3 The fractional diffusion equation	21
2.4 Discretization of the fractional Laplacian	24
2.5 Langevin Equations	31
3 Numerical analysis of the different equations	33
3.1 Simulations of the continuous approximations of Liu's IBM . . .	33
3.2 Simulations of the Lévy models	41
3.3 Existence of patterns	47
3.4 Stability of the stationary states	49
4 Analysis of the new model	51
4.1 Rescaling	51
4.2 Linear stability analysis	52
4.3 Modulation Equations	57
Conclusion & Outlook	63
Acknowledgements	66
A Eyre's Unconditionally Stable Scheme	70
B Code for equation (2.2.20)	72

Introduction

Most people tend to think of mussels as an exquisite type of food, not necessarily as a subject of mathematical research. One could agree with this, if you would assume that a mussel spends its life at the bottom of the sea waiting to be eaten by you. On the contrary, mussels take a lot of effort *nó*t to be eaten. The mussels' survival strategies, which allow them to live a happy life on the sea floor, cause the mussels to organize themselves in spatial inhomogeneous patterns which are shown in figure 2. The question “How do these patterns form” does serve as a good subject for mathematical research.



(a) The blue mussel, *Mytilus edulis*, the subject of our research.

(b) Mussels the way we are used to see them, as food.

Figure 1: Different ways of looking at a mussel.

Pattern formation is very common in nature, ranging from ripples in sand dunes and fairy circles in the Namibian desert, to the stripes of a zebra. Some examples are shown in figure 3. Scientists have worked hard, and still work hard, to explain all patterns we see around us. The modelling of these patterns by physicists, chemists, biologists and others has led to the development of a lot of new mathematics. The equations by Duffing and Lorenz led to the development of chaos theory and Van der Pol's equation to the study of relaxation oscillators [14]. Another class of equations that has been widely studied are reaction-diffusion equations, since Alan Turing discovered a mechanism for pattern formation in these equations [33]. These systems have been proven very useful in explaining a wide variety of patterns, see for an overview [17] or take a look at the beautiful book by Meinhardt [23] which gives a broad overlook of different patterns in seashells.

However, not all patterns found in nature can be explained by this mechanism. Another prominent mechanism for pattern formation is *self organisation*. Animals form flocks and herds to protect themselves from predation and trees



Figure 2: Patterns as observed in the field (left, from [20]) and in the laboratory (right, from [19]). Both figures are approximately a meter wide.

in arid climates group together to keep their water supply safe [28, 29]. Writing down models for self organisation is often not straightforward. First, we should figure out why it is a (dis)advantage to cooperate and then we should describe how the communication between the organisms works, even when we do not have a clear idea how. Despite these difficulties, Liu, Doelman, Rottschäfer et al., wrote down an equation to model the pattern formation in mussel beds [19].

The Cahn-Hilliard equation

Interestingly enough, the equation they found is the Cahn-Hilliard equation, even though this equation originates from solid state physics [3]. The derivation of this equation is based on one important observation: It has experimentally been shown that the movement speed of a mussel depends on the number of mussels in it's neighbourhood. In numerous papers by ecologists De Jager, Van de Koppel and Liu, models have been introduced which describe the movement of individual mussels, so called Individual Based Models. These models include density depended movement and can accurately describe pattern formation in mussel beds, see for example [8, 34] and the references therein or figure 4. Based on this observation, they derived the following version of the Cahn-Hilliard equation:

$$\frac{\partial m}{\partial t} = \nabla [g(m)\nabla m - \kappa\nabla(\Delta m)].$$

Here, m is the rescaled mussel density and $g(m) = v(m)(v(m) + m\frac{\partial v}{\partial m})$ with $v(m)$ the experimental found density dependent speed of the mussels. As the authors showed, there is a good agreement between the numerical simulations of the Cahn-Hilliard equation and the patterns observed in the field, but there is one problem. The Cahn-Hilliard equation describes the early stages of pattern formation very well, but the solutions of the Cahn-Hilliard equation evolve into increasingly larger structures, a process called coarsening. The patterns in mussel beds however, stop at some point with coarsening and freeze into a pattern with a fixed wavelength as shown in figure 2. In mathematical terms, the patterns in the Cahn-Hilliard equation are unstable while the patterns in the mussels are stable.

The original aim of this thesis was to find a way to explain why the coarsening stops in the mussel beds within the framework of the Cahn-Hilliard equation.



(a) Ripples in sand dunes. (b) Fairy circles in the Namibian desert. (c) Pattern on a sea shell.

Figure 3: Different examples of patterns in nature.

That this is not a futile quest is shown in [6], where in a different context an extra potential is added to the equation in which case the patterns become stable. Different options were put forward to explain why the present Cahn-Hilliard model does not accurately describe the coarsening, e.g. missing interactions with food supplies (algae) and the nonzero size of the mussels which would make it hard to make a valid continuous model. In [1] a more detailed discussion of these options is given. Chapter 1 gives an introduction to the Cahn-Hilliard equation and the modelling behind it.

However, studying the Cahn-Hilliard equation made us realize that we should not focus on the equation, but rather on the modelling behind it. We realised that the density dependent movement speed of the mussels should be incorporated in another way into the equations. This realisation led to the rest of this thesis, in which we start over again with the modelling and study the newly derived equations.

Two scale dependent feedback

As mentioned before, the Individual Based Models (IBM's) as developed in [8, 20], do show pattern formation with a fixed wavelength. Therefore, our aim in chapter 2 will be to find a continuous approximation of these models. The key ingredient of the IBM's is the following observation by De Jager [8], to which we will refer as scale depended feedback:

By moving into cooperative aggregations, mussels increase their local density, which decreases wave stress and predation risk. Conversely, competition for algae, which occurs on a larger spatial scale than facilitation, prevents the formation of larger clumps by limiting the number of mussels within a long range.

Next, we need a way to translate this into a mathematical framework. Therefore we introduce the attractiveness A :

$$A(\rho) = D_s \rho_s - D_l \rho_l.$$

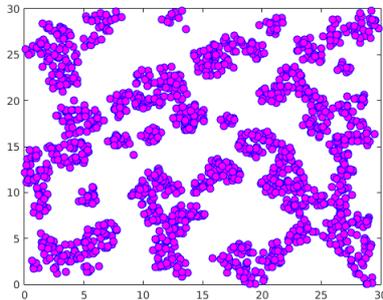


Figure 4: Example of a simulation of the IBM by De Jager after completion, [8]. Each magenta dot represents a single mussel and structures similar to the structures in figure 2 have been formed.

Here, $\rho_s(x)$ and $\rho_l(x)$ stand for the average density of mussels at a short and long length scale around a point x and are computed by

$$\rho_\mu(x) = \frac{1}{\pi r_\mu^2} \int_{B_{r_\mu}(x)} \rho(x') dx',$$

with r_s, r_l the short respectively long length scale.

We see that the attractiveness $A(\rho)$ increases as ρ_s increases, but decreases as ρ_l increases, which is in line with the observation by De Jager as stated above.

The main aim of this thesis is to derive a model that incorporates this scale dependent feedback and hence produces similar patterns as those observed in the field. Chapter 2 is mainly devoted to the study of the different IBM's and the derivation of equations corresponding to these models. In chapter 3 we will simulate the equations derived in chapter 2 and compare these solutions with the experimentally observed patterns. Chapter 4 will be devoted to a more in depth study of one of the derived equations.

The Cahn-Hilliard equation

In this chapter we will discuss the Cahn-Hilliard equation (CH-equation) as a model for pattern formation in mussel beds. First, we derive the equation following [19] for which we must study the ecological background. Next, simulations of the CH-equation are compared with the observed patterns. Afterwards, we will discuss the validity of this model. Note that detailed description of the CH-equation is not included here. For a more extensive treatment of the CH-equation as a model for pattern formation in mussel beds see [1].

1.1 Derivation of the Cahn-Hilliard equation

In this section, we will use the notation as is used in [19]. The first observation we make is that the patterns form in less than 24 hours. Therefore, we assume that we can ignore the growth of the individual mussels and the mortality rate, hence we assume that the number of mussels is conserved. Now we want to derive an equation for the local mussel density $M = M(x, t)$ of mussels in a 2-d bounded space, more specifically, we take as domain $\Omega = [0, L] \times [0, H]$, $x \in \Omega$, and assume Neumann boundary conditions for M . For the modelling, we start with a generic conservation equation:

$$\frac{\partial M}{\partial t} = -\nabla \cdot J. \quad (1.1.1)$$

In this equation J describes the total flux of mussels at a given point. In the derivation of the standard diffusion equation we now would use Fick's law which states

$$J = -D\nabla M. \quad (1.1.2)$$

In words, this equation just states that particles (or mussels) tend to move away from high densities. Mussels in general do not follow this rule. The following scale dependent feedback was observed by De Jager [8]:

By moving into cooperative aggregations, mussels increase their local density, which decreases wave stress and predation risk. Conversely, competition for algae, which occurs on a larger spatial scale than facilitation, prevents the formation of larger clumps by limiting the number of mussels within a long range.

Therefore, we expect that the speed at which mussels move depends on the density of mussels. At low densities, mussels move fast to find other mussels and

at high densities they also move fast to get away from other mussels. Mussels move slowly when their density is at a preferred intermediate density. In [19] it has been shown that the speed $V = V(M)$ of the mussels can be described by a quadratic function of the density, $V = aM^2 + bM + c$, for suitable a, b, c . Now we need to find a new Fickian law to describe the flux in terms of M where we take the density dependent movement speed V into account. Under the assumption that the mussels do a random walk and at each time step take a step of length $V(M)$ after which they reorientate, Schnitzer [30] derived the following expression for the flux:

$$J = -\frac{1}{2\tau} \left[V \left(V + M \frac{\partial V}{\partial M} \right) \right] \nabla M, \quad (1.1.3)$$

where $\tau = \tau(M)$ is the turning rate which can be interpreted as the characteristic time between two turns. The equation above is only valid when $v(M)|\nabla\tau(M)| \ll \tau^2$. This means that the change of τ over a typical distance V between two turns should be small. From now on we assume that τ is constant so this condition is met.

Substituting equation (1.1.3) into equation (1.1.1) results in the following equation:

$$\frac{\partial M}{\partial t} = \nabla \left[\frac{1}{2\tau} V \left(V + M \frac{\partial V}{\partial M} \right) \nabla M \right]. \quad (1.1.4)$$

However, the term $[V (V + M \frac{\partial V}{\partial M})]$ can become negative, in which case the equation becomes notoriously unstable. We can solve this by adding a higher order correction to Fick's original law, namely $\nabla(\kappa\Delta M)$ for $\kappa > 0$ small. Note that the sign is opposite from the original $-D\nabla M$ term. Equation (1.1.4) now becomes

$$\frac{\partial M}{\partial t} = \nabla \left[\frac{1}{2\tau} V \left(V + M \frac{\partial V}{\partial M} \right) \nabla M - \nabla(\kappa\Delta M) \right], \quad (1.1.5)$$

When we introduce the rescaling $m = \sqrt{a/c}M$, $D_0 = \frac{c^2}{2\tau}$, $\kappa_1 = \frac{2\tau\kappa}{c^2}$ and $\beta = b/\sqrt{ac}$ we end up with the following CH-equation:

$$\frac{\partial m}{\partial t} = D_0 \nabla [g(m)\nabla m - \kappa_1 \nabla(\Delta m)]. \quad (1.1.6)$$

Here, $g(m) = v(m) (v(m) + m \frac{dv}{dm})$ with $v(m) = m^2 - \beta m + 1$. This equation is also often written in an integrated form:

$$\frac{\partial m}{\partial t} = D_0 \Delta [G(m) - \kappa_1 \Delta m], \quad (1.1.7)$$

with $G(m)$ the anti-derivative of g .

Notice that at this point, we do have experimental values for a, b and c , but not for τ and κ . However, κ describes the intensity of the higher order correction to the diffusion equation and therefore we expect κ_1 to be significantly smaller than D_0 .

1.2 Some theory on the Cahn-Hilliard equation

Before we compare simulations of the CH-equation, we first give some background on the CH-equation. From a modelling point of view, we need the equation to be conserved, i.e. we want

$$\frac{d}{dt} \int_{\Omega} m dx = 0. \quad (1.2.1)$$

Now when we substitute equation (1.1.6) in the equation above and integrate by parts we find

$$\frac{d}{dt} \int_{\Omega} m dx = \int_{\Omega} \frac{\partial m}{\partial t} dx \quad (1.2.2)$$

$$= \int_{\Omega} D_0 \nabla [g(m) \nabla m - \kappa_1 \nabla(\Delta m)] dx \quad (1.2.3)$$

$$= \int_{\partial\Omega} D_0 [g(m) \nabla m - \kappa_1 \nabla(\Delta m)] \cdot \vec{n} dx. \quad (1.2.4)$$

We assumed Neumann boundary conditions, so by assumption $\nabla m \cdot \vec{n} = 0$ on the boundary. As we can see, just assuming Neumann boundary conditions is not enough for conservation of mass, we also need $\nabla(\Delta m) \cdot \vec{n} = 0$ on the boundary. Therefore, we need to add the assumption $\nabla(\Delta m) \cdot \vec{n} = 0$ on the boundary for conservation of mass.

Now we have determined the conservation of mass, we need to determine when patterns can develop in the CH-equation. A minimal requirement for this is that the spatial homogeneous background needs to become unstable, because in the biological experiments patterns form from a homogeneous background. To determine the stability of the homogeneous background we need to do a linear stability analysis for the CH-equation.

1.2.1 Linear stability analysis for the CH-equation

For the stability analysis we assume that $m = m_0 + \varepsilon e^{\sigma(\vec{k})t + i\vec{k} \cdot \vec{x}}$. \vec{k} is here a 2-dimensional wave vector $\vec{k} = (k_x, k_y)$. When we are precise, we should take $\vec{k} = k_{mn} = (\frac{\pi m}{L}, \frac{\pi n}{H})$, but for simplicity, we shall use the continuous spectrum. When we substitute the given expansion into equation (1.1.6) we see that the $\mathcal{O}(1)$ equation drops out and at $\mathcal{O}(\varepsilon)$ we find, after dividing by $e^{\sigma(\vec{k})t + i\vec{k} \cdot \vec{x}}$, the following dispersion relation:

$$\sigma(\vec{k}) = -|\vec{k}|^2 (g(m_0) + \kappa_1 |\vec{k}|^2). \quad (1.2.5)$$

Note that $g(m_0)$ is explicitly a function of β . From the dispersion relation (1.2.5) we immediately see that all wavenumbers are stable when $g(m_0) > 0$. The moment $g(m_0)$ becomes negative, wavenumbers with $|\vec{k}|^2$ in the interval $[0, -g(m_0)/\kappa_1]$ will be unstable. The function $g(m)$ is given by $g(m) = (m^2 - \beta m + 1)(3m^2 - 2\beta m + 1)$. From this function we can see that $g(m)$ can only become negative when $\beta > \sqrt{3}$. However, $v(m) = m^2 - \beta m + 1$ must remain positive because mussels can not have negative speed so $\beta < 2$. In the case that $\sqrt{3} < \beta < 2$, we find that $g(m_0)$ becomes negative for m_0 in the following

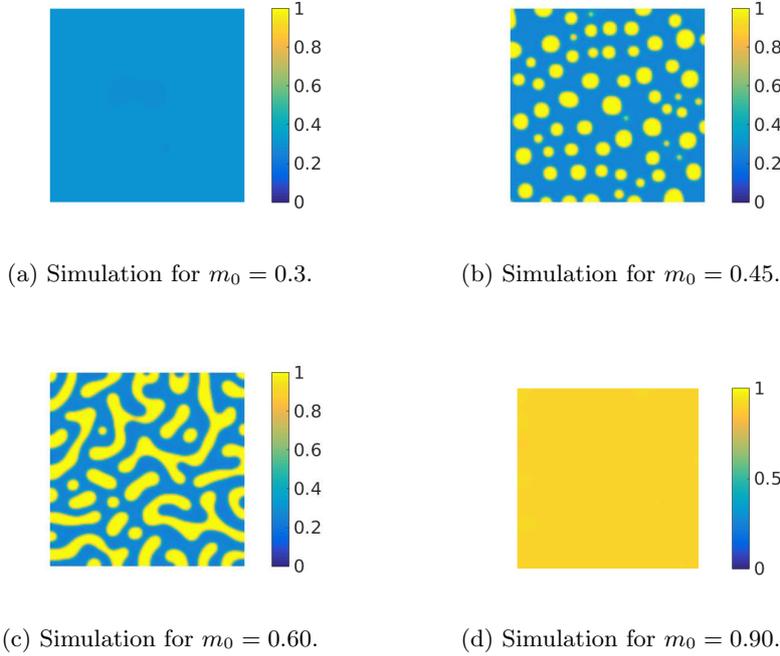


Figure 1.1: Simulations of equation (1.1.6) after 6000 iterations with $dt = 0.1$ for different m_0 . The other parameters were $D_0 = 1$, $\kappa_1 = 0.0025$, $\beta = 1.89$, $L = H = 40$ and $dx = L/128$.

interval:

$$\frac{\beta - \sqrt{\beta^2 - 3}}{3} < m_0 < \frac{\beta + \sqrt{\beta^2 - 3}}{3}. \quad (1.2.6)$$

This means that when the mussel density is very low or high, no patterns are formed. This is in line with our ecological intuition. First, if the number of mussels is very low, it will be difficult for the mussels to find each other and form clusters. On the other hand, when the sea floor is completely filled with mussels, there is no point in forming clusters because it will not improve your chances of survival.

At this point there is no clear biological meaning we can give to κ and hence to κ_1 . However, we do see that when we make κ smaller, more and more small wavenumbers become unstable. Hence, κ determines the spatial scale at which patterns can emerge on a short time scale and we must keep the ratio $-g(m_0)/\kappa_1$ in mind when we start simulating. If the domain size is too small, it is possible that the long unstable waves do not fit in the grid because the smallest discrete wavenumbers k_{01} or k_{10} might have an absolute value that is not in the interval $[0, -g(m_0)/\kappa_1]$. Therefore, to get a good agreement between the stability analysis on \mathbf{R}^2 and the simulations on a finite grid, L must be chosen large or κ_1 small.



Figure 1.2: Pattern formation in mussels under laboratory conditions. Notice the similarities in shape between this figure and figure 1.1c. Figure comes from [19].

1.2.2 Coarsening and stationary solutions

Although the CH-equation is frequently used for explaining patterns as a result of phase separation, the CH-equation often only describes this behaviour at a short time scale. Numerical simulations show that the patterns evolve into patterns with an increasingly larger wavelength. This process is called coarsening, see e.g. figures 1.3 and 1.5. The speed of the coarsening is given by the Lifshitz-Slyozov law, which states that the growth in time of the dominant wavelength $\ell(t)$ of the solutions is bounded by $t^{1/3}$ [18]. Now the interesting question is where the coarsening stops, or in other words, what do the stable stationary solutions of the CH-equation look like? Unfortunately, the coarsening of the CH-equation just goes on and on. Hence, the stationary states of the CH-equation show no patterns, but the stationary states are just solutions with one single interface between the high and low phase [4].

1.3 Comparison with the biological data

Now that we have derived an equation, we need to compare the solutions of the CH-equation with actual patterns observed in nature. Before we can start simulating, we need an efficient way to simulate the CH-equation.

1.3.1 Numerical simulations of the CH-equation

For the numerical integration we used Eyre's unconditionally stable scheme [35]. The code for the original Cahn-Hilliard equation, i.e. with $G(m) = m^3 - m$ can be found at Eyre's homepage.¹ The fact that this scheme is unconditionally stable gives us the opportunity to use increasingly larger time steps to see the behaviour of the CH-equation at long time scales. As is noted in [35], the domain walls (the steep interfaces between the two different phases) tend to move with a speed $v \sim t^{-2/3}$. This means that when we use a fixed time step dt we eventually simulate an almost stationary solution with a very high accuracy. This however blocks us from converging to the stationary solution. With Eyre's scheme we can increase the time step during the simulations and

¹<http://www.math.utah.edu/~eyre/computing/matlab-intro/ch.txt>

therefore speed up the simulation. But even then, there are no guarantees that the solution will converge to the stationary solutions within a reasonable amount of computation time. If one is really interested in the stationary solutions, there are several options such as adding noise [15] or using spectral weighting [6]. However, we are interested in the comparison of the CH-equation with the experimental results and the comparison breaks down long before the stationary state is reached and therefore there is no need to simulate up to the moment a single interface has formed. For a more detailed explanation of the code, see appendix A.

1.3.2 Results of the simulations

First, we need to estimate values for D_0, β, m_0 and κ_1 . We notice that the value of D_0 is not important for the qualitative behaviour of the equation so we can set D_0 equal to 1. To keep the rescaling consistent, we now find $\kappa_1 = \kappa$. This means that we can choose κ freely. Furthermore, we know that $\beta = b/\sqrt{ac}$ and with the experimental values for a, b and c from [19] we find $\beta = 1.89$. For this value of β , equation (1.2.6) becomes

$$0.38 < m_0 < 0.88. \quad (1.3.1)$$

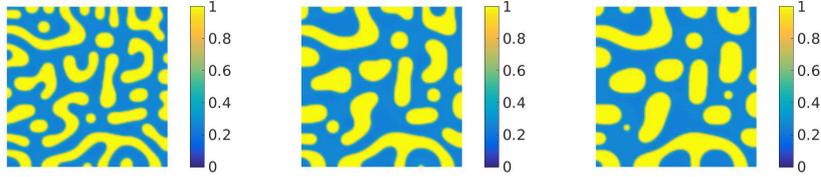
Now we can compare the numerical simulations of the CH-equation with the observations from the field. In figure 1.1 simulations of the CH-equation for different values of m_0 are shown. Indeed, when $m_0 < 0.38$ or $m_0 > 0.88$ we see no patterns. For $0.38 < m_0 < 0.88$ we see phase separation and patterns. However, what is important to keep in mind is that the simulations shown in figure 1.1 are just intermediate stages, at least for $m_0 = 0.45$ and $m_0 = 0.6$. These solutions will evolve into similar looking structures with a larger wavelength which is called coarsening, see section 1.2.2. For $m_0 = 0.6$, this is shown in figure 1.3. As we can see, from $t = 1000$ to $t = 2000$ the labyrinth structures in the solution have become noticeably “thicker”, but from $t = 2000$ to $t = 3000$ the difference is less obviously. This illustrates the fact that the coarsening and converging to a stationary state slows down considerably.

To make extra clear that the CH-equation governs phase separation, we plot an intersection of a solution, see figure 1.4. This figure shows clearly that the solution exists of two phases, a low phase around $\rho = 0.25$ and a high phase around $\rho = 1.1$, connected by steep interfaces.

1.3.3 Comparison with the biological data

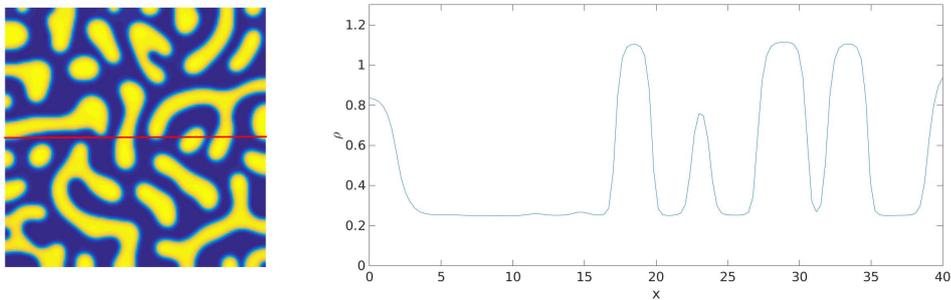
Figure 1.2 shows the patterns in mussel beds under laboratory conditions. The mussels were spread out evenly and the picture shows the resulting pattern after 24 hours [19]. For videos of the experiment and more on the experimental set up, do visit Van de Koppel’s homepage². As is pointed out in [19], the comparison between the solutions of the CH-equation and the experimental results is remarkably good. For low values of m_0 , we see separate patches of mussels in the field and we see dots in the solution of the CH-equation as in figure 1.1a. For higher values of m_0 , we see labyrinth like structures in the field as well as in the simulations. However, the most convincing comparison

²www.johanvandekoppel.nl



(a) Solution at $t = 1000$. (b) Solution at $t = 2000$. (c) Solution at $t = 3000$.

Figure 1.3: Time evolution of the Cahn-Hilliard equation (1.1.6) with $m_0 = 0.6$. Figure (a) shows the solution at time $t = 1000$ when patterns have formed, figures (b) and (c) show the evolution of the solution into thicker structures, i.e. coarsening. $D_0 = 1$, $\kappa_1 = 0.0025$, $\beta = 1.89$, $L = H = 40$ and $dx = L/128$.



(a) Solution at $t = 10^3$. (b) Intersection over the red line in left figure.

Figure 1.4: Simulation of equation (1.1.6) with $m_0 = 0.6$. The other parameters are kept the same as in figure 1.1.

between the CH-equation and the laboratory experiments is the fact that both show coarsening in the same way. In figure 1.5 we see that the coarsening in the CH-equation follow the Lifshitz-Slyozov law $\ell(t) \sim t^{1/3}$, but also the pattern formation in the experimental set-up shows coarsening following the Lifshitz-Slyozov in the early phases of the evolution.

1.4 Validity of the CH-equation

The first discrepancy between the simulations and the experimental results can be found when we compare figure 1.2 with figure 1.4. In the experimental setup, the density of mussels between the patches of mussels is zero, while this clearly is not the case in the simulations. We do not consider this as an actual problem, but apparently the authors of [19] did, because the simulations shown are not the actual solutions of equation (1.1.6). Without mentioning it, they shifted down the solutions in such a way that the low phase is zero in order to get a better agreement with the experimental results. The second more troubling difference between the evolution of the mussel patterns in the experiments and the simulated patterns in the CH-equation, is the fact that the CH-equation goes on and on with coarsening whereas the mussels do not.

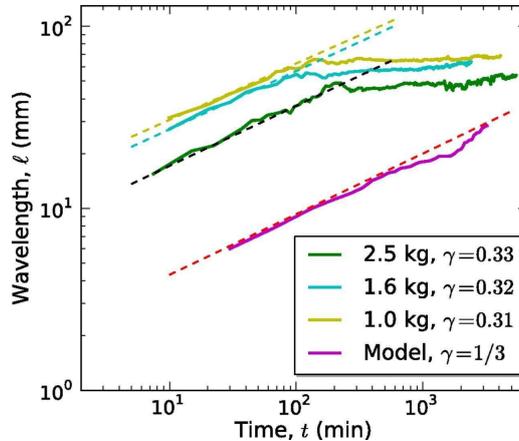
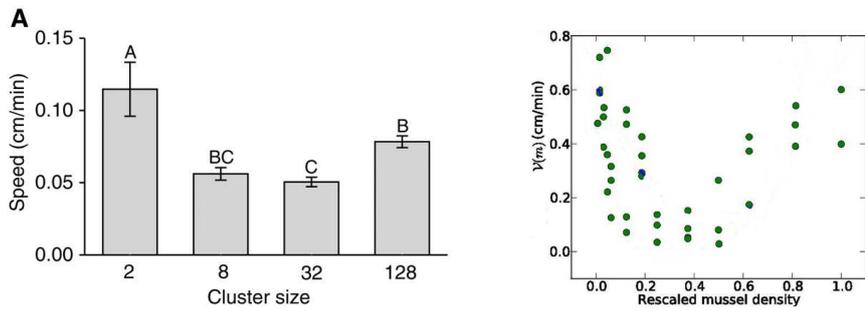


Figure 1.5: Coarsening in the experimental set-up (top 3 curves) and in the CH-equation (1.1.6), figure from [19]. The top 3 curves show the coarsening of the dominant wavelength ℓ versus time for 3 different experimental set-ups. The dashed lines are fits to the data of the form $\ell(t) \sim t^\gamma$ with the gamma given in the figure. The bottom curve shows the coarsening in the CH-equation. Note that the coarsening in the mussels stops after approximately 2 hours, while the coarsening in the CH-equation continues.

How can we explain this difference? This question lies at the very beginning of this thesis. Bastiaansen proposed a few solutions in his thesis [1]. One idea is to extend the CH-equation to a predator-prey model, where mussels are the predator and algae the prey, but no parameter regime could be found in which the patterns become stable. Another explanation could be that the nonzero size of the mussels is the limiting factor. At long time scales, the coarsening is very slow and the interaction between two clumps is very small. However, in the real mussel beds, the interaction between two clumps of mussels can not be smaller than one mussel which limits the coarsening behaviour. On the other hand, videos from the experiments show that mussels keep jumping from one clump to another, even after the patterns have formed so different clumps certainly do “communicate”. All in all no satisfactory solution was found to our taste.

Working on this thesis, we realised that something else might be the problem. That is, what is the meaning of M , the local mussel density? To examine the problem, compare figure 1.6a with figure 1.6b. Figure 1.6b shows the data to which the function $V(M)$ is fitted and figure 1.6a shows where the data comes from. What we see is that in figure 1.6a, the speed depends on the number of mussels in a cluster and larger clusters take up a larger physical space, whereas in the second figure the speed depends on the density. This raises the following question: how is the local mussel density M defined? The clump with 128 mussels has been rescaled to density 1. This means that in the process of going from cluster size to rescaled density, one just divides by 128. However, a clump of 128 mussels takes up a larger space than a small clump. Dividing by 128 disregards this fact, but more important, it is also in contradiction with the scale dependent feedback we introduced at the beginning of this chapter.

Therefore, we argue that one cannot relate the speed of the mussel to the



(a) Average speed of a mussel versus cluster size, figure from [34]. (b) Rescaled mussel speed versus rescaled mussel density as is used in the derivation of the CH-equation. A cluster of 128 mussels corresponds to a density of 1, figure from [19].

Figure 1.6: Two different interpretations of the density dependent speed.

density at a single point, but one should relate the speed to the density at larger length scales.

The derivation of the CH-equation crucially depends on the fact that the speed of the mussels depends on the density at a single point. Hence, if we want to incorporate the scale dependent feedback into our model in the hope that this will result in solutions with a fixed wavelength, we should start the modelling all over again, and that is exactly what we will do in the next section.

Modelling two Scale Dependent Feedback

Modelling should always start with the following question: With what kind of description do I want to end? Are we interested in the mussel movement per individual, or do we want an equation that describes the (probability) density of the mussels? And if we want an equation, what kind of equation, a PDE or an integral equation?

Different levels of descriptions have different advantages and disadvantages. It is quite straightforward to write down equations for, say, N individual mussels. One should just identify how a mussel moves with respect to its neighbours and the nonzero size of the mussels can easily be accounted for. Although writing down these equations and translating them into computer codes is not extremely difficult, the mathematical study of a system of N coupled nonlinear equations can be very hard, especially because the systems will probably be stochastic. The best way to deal with the inherent stochastic nature of the individual animal movement is to write down mesoscopic equations. In that approach we write down equations for the probability density of the whole population. This means that we do not follow individual trajectories, but are interested in the average of all individual trajectories for large populations. These equations mostly tend to come in the form of integral equations. At the macroscopic level we end up with the more familiar partial differential equations (PDE's). Although PDE's are better understood than integral equations and therefore easier to study, a lot of information is lost in going from the mesoscopic level to the macroscopic level. This is best explained by the standard diffusion equation. The diffusion equation is a well understood PDE and is a macroscopic approximation of individual particle movement at the microscopic level. However, in the diffusion equation information travels at infinite speed, which would mean that particles are allowed to take arbitrarily large jumps in arbitrarily small time steps. Another problem with the macroscopic formulation is that the step from a mesoscopic equation to a PDE often fails in the case of long range dispersal. We will come to this problem when we discuss Lévy walks.

In this chapter, we will first discuss some Individual Based Models (IBM) as developed in several papers by the group of Johan van de Koppel at the NIOZ Royal Netherlands Institute for Sea Research, e.g. [20, 34, 8]. Then we will write down mesoscopic equations using the knowledge from the IBMs and try to derive PDEs. The success of this approach depends on the underlying random walk in the IBM. In section 2.2 we derive a nonlinear diffusion equation based on the IBM from [20], but we will also show why this derivation process

does not work for the IBM from [8]. In sections 2.3 and 2.4 we develop the tools necessary to model the IBM from [8] and derive a nonlinear fractional diffusion equation.

2.1 Individual Based Models

In chapter 1 we stated the principle of *scale dependent feedback*:

By moving into cooperative aggregations, mussels increase their local density, which decreases wave stress and predation risk. Conversely, competition for algae, which occurs on a larger spatial scale than facilitation, prevents the formation of larger clumps by limiting the number of mussels within a long range.

Consequently, we assume that a mussel can determine the attractiveness of its position at every moment and adjust its movement to this. Therefore, we have come up with a way to describe the attractiveness of a point and for simplicity we start with a discrete model in one dimension. Hence, we take \mathbf{Z} to be the grid and N_k the number of mussels at point k . We furthermore assume that two points are separated by a distance ℓ . For every point k , we define the attractiveness $A_k(t)$ which is given by the scale dependent feedback. We assume that there is positive feedback on the short length scale R_s and negative feedback on the large length scale R_l :

$$A_k(t) = \frac{D_s}{2R_s + 1} \sum_{\substack{i \in \mathbf{Z} \\ |i-k| \leq R_s}} N_i(t) - \frac{D_l}{2R_l + 1} \sum_{\substack{i \in \mathbf{Z} \\ |i-k| \leq R_l}} N_i(t). \quad (2.1.1)$$

Note that $N^\mu = \frac{D_\mu}{2R_\mu + 1} \sum_{|i-k| \leq R_\mu} N_i(t)$ is the average of $N(t)$ over the interval $[k - R_\mu, k + R_\mu]$ for $\mu = s, l$. D_l and D_s are positive constants. In words this equation states that the attractiveness of a point increases as the short scale density N^s increases and decreases as the long scale density N^l increases. Of course, we can also formulate the attractiveness in two dimensions. In that case, we take as grid \mathbf{Z}^2 and average over a circle instead of an interval:

$$A_{mn}(t) = \frac{D_s}{\pi R_s^2} \sum_{\substack{(i,j) \in \mathbf{Z}^2 \\ |(i,j)-(m,n)| \leq R_s}} N_{ij}(t) - \frac{D_l}{\pi R_l^2} \sum_{\substack{(i,j) \in \mathbf{Z}^2 \\ |(i,j)-(m,n)| \leq R_l}} N_{ij}(t). \quad (2.1.2)$$

Note here that πR_μ^2 is an approximation for the number of grid points in a circle with radius R_μ .

With this definition at hand we can formulate movement rules for the individual mussels.

When we study the code from the IBM by De Jager et al. [8] (we will refer to this model as De Jager's IBM or just De Jager), we find the following rules for the Random Walk the mussels execute. At every timestep, a mussel takes a step with the following chance:

```
1 pmove = (0.6 - 1.3.*ssd + 1.1.*lsd) > rand(1,np);
```

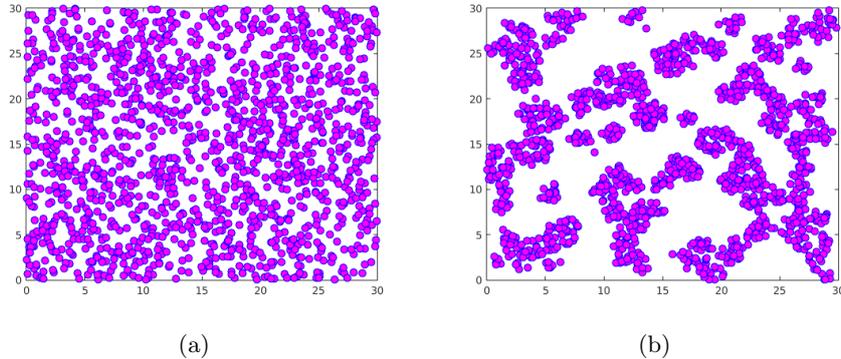


Figure 2.1: Simulations of the IBM from [8]. (a) The initial condition. (b) The simulation after convergence. $D_s = 1.3$, $D_l = 1.1$, $R_s = 1.1$, $R_l = 7.5$. A simulation has converged when the average of N^s/N^l is larger than 1.5.

where `ssd` and `lsd` stand for the short respectively long scale density. With our definition of A , this means that a mussel at position (m, n) takes a step if $0.6 - A > X$ for X a random variable uniformly distributed on $[0, 1]$. Then, if a mussel takes a step, it takes a step where the angle is uniformly distributed on $[0, 2\pi]$ and the step length is distributed with a Pareto distribution with parameters independent of the density. This is coded as:

```
L = xmin./(1 - rand(1,np)).^(1/(mu-1));
```

where `xmin` is the minimum jump distance and μ the exponent of the Pareto distribution.

These two core rules prove to be able to generate patterns that are very alike to the observed patterns, see figure 2.1. However, we should notice here that the distribution of the steps, i.e. angle and step length, is properly normalized in $[0, 2\pi] \times \mathbf{R}^+$, but not in \mathbf{R}^2 . It is also good to note that in this model only the chance of taking a step depends on A , but the step length is independent of A .

In the IBM as described by Liu et al. in [20] (we will refer to this model as Liu's IBM or just Liu) the dependence on A is different. They state the following:

Statistical analysis of experimental movement trails revealed that the distances covered by the mussels during 1 min could be approximated by an exponential distribution [34, 7], where the frequency h of occurrence decreased with movement distance r ; $h(r, \beta) = \frac{1}{\beta} e^{-r/\beta}$. Here, the scaling parameter β is a function of the densities of mussels in the neighbourhood. Statistical analysis of the relationship between movement speed and mussel density revealed that this scale parameter β is negatively affected by density at a scale of 1.5 cm, but positively affected by density at a scale of 6.0 cm. The scale-dependent feedback can be expressed as $\beta = \frac{1}{p_0 + p_1 L_1 - p_2 L_2}$, where L_1 and L_2 represent the mussel densities at the two different scales, and

the coefficients p_0 , p_1 and p_2 represent an intercept (movement speed at zero mussel density), and the coefficients of density dependence, respectively, which were all obtained from the regression analysis.

Although both models do generate the labyrinth structures we are looking for, there is from a mathematical point of view a problem. De Jager compares $0.6 - 1.3 \cdot \text{ssd} + 1.1 \cdot \text{lsd}$ with a uniform distribution on $[0, 1]$ while the range of $0.6 - 1.3 \cdot \text{ssd} + 1.1 \cdot \text{lsd}$ far exceeds $[0, 1]$. This problem becomes even bigger in the IBM by Liu, where the denominator of β becomes negative and hence passes through zero while this parameter should be positive in an exponential distribution. Liu solved this problem by replacing $p_0 + p_1 L_1 - p_2 L_2$ with $\max\{0.001, p_0 + p_1 L_1 - p_2 L_2\}$. These problems set aside, both models show that it is possible to form patterns solely based on the fact that the random walk the mussels execute depends on A . As we will show in the next section, deriving a continuous equation for the IBM by Liu is straightforward, but unfortunately the same is not true for the IBM by De Jager.

2.2 Fokker-Planck equation for the IBM by Liu et al.

Following Grindrod [13], we propose the following mesoscopic or master equation for the density of mussels $\rho(x, t)$:

$$\rho(x, t) = \int_{\mathbf{R}^2} \Phi(x, t, y, s) \rho(y, s) dy. \quad (2.2.1)$$

Here, $\Phi(x, t, y, s)$ is the probability that a mussel (or particle in general) is at position x on time t , given that it started in y at time s . In ecology, Φ is also called the dispersal kernel as it contains all the information on how the particles (or animals) disperse. By this definition of Φ we have that $\Phi(x, s, y, s) = \delta(x - y)$. Notice here that we made the approximation that the mussels move continuously in time instead of making separate jumps. Now given the information from the IBM by Liu in the previous section, we write down an exponentially decaying kernel that is radially symmetric. Hence, we propose the following expression for $\Phi(x, t, y, s)$ which is normalized to 1 for every time $(t - s)$:

$$\Phi(x, t, y, s) = \frac{\lambda^2}{2\pi(t - s)} e^{-\frac{\lambda|x-y|}{\sqrt{t-s}}}. \quad (2.2.2)$$

Here, we should take $\lambda = 1/\beta(y, s)$ following Liu. As pointed out before, this is not well defined because β is not strictly positive and therefore we propose to take

$$\lambda(y, s) = \frac{1}{2} + \frac{1}{2} \tanh(A(y, s)) =: g(A), \quad (2.2.3)$$

with $A(x, s)$ the continuum approximation of equation (2.1.2):

$$A(x, t) = \frac{D_s}{\pi r_s^2} \int_{B_{r_s}(x)} u(x', t) dx' - \frac{D_l}{\pi r_l^2} \int_{B_{r_l}(x)} u(x', t) dx'. \quad (2.2.4)$$

Here, and in the rest of this thesis, we use r_s and r_l to denote the real short and long length scale, whereas R_s and R_l stand for the short and long length scale in terms of grid points. The reason we took $g(x) = \frac{1}{2} + \frac{1}{2} \tanh(x)$ is because it maps $-\infty$ to zero and is monotonic in order to be sure that λ remains positive. Every other function with this property might do as well. From this point we can write down the Fokker-Planck equation corresponding to equation (2.2.1), but Φ is explicitly dependent on $\rho(y, s)$, not just on y and s , so we should dig through the derivation of the Fokker-Planck equation to see if it still holds. Furthermore, the derivation will show that the existence of the second moment of Φ is crucial to the derivation process. Hence, it is instructive to give the full derivation.

Therefore, we again start with equation (2.2.1) for a general Φ , but we introduce the variable $\mathbf{z} = \mathbf{x} - \mathbf{y}$ and use bold letters to explicitly stress the fact that the variables are in \mathbf{R}^2 :

$$\rho(x, t) = \int_{\mathbf{R}^2} \rho(\mathbf{x} - \mathbf{z}, s) \Phi(\rho(\mathbf{x} - \mathbf{z}, s), \mathbf{z}, t, s) d\mathbf{z}. \quad (2.2.5)$$

We explicitly wrote down that $\Phi = \Phi(\rho(\mathbf{y}, s), \mathbf{z}, t, s)$. Now we approximate $\rho(\mathbf{x} - \mathbf{z}, s)$ and $\Phi(\rho(\mathbf{x} - \mathbf{z}, s), \mathbf{z})$ for $|z| \ll 1$:

$$\rho(\mathbf{x} - \mathbf{z}, s) = \rho(\mathbf{x}, s) - D\rho(\mathbf{x}) \cdot \mathbf{z} + \frac{1}{2} \mathbf{z}^T D^2 \rho(\mathbf{x}) \mathbf{z} + \mathcal{O}(\mathbf{z}^3), \quad (2.2.6)$$

where $D\rho$ is the gradient of ρ with respect to \mathbf{x} and $D^2\rho$ the Hessian of ρ with respect to \mathbf{x} .

$$\begin{aligned} \Phi(\rho(\mathbf{x} - \mathbf{z}, t), \mathbf{z}) &= \Phi(\rho(\mathbf{x}, t), \mathbf{z}) - D(\Phi(\rho(\mathbf{x}, t), \mathbf{z})) \cdot \mathbf{z} \\ &+ \frac{1}{2} \mathbf{z}^T D^2(\Phi(\rho(\mathbf{x}, t), \mathbf{z})) \mathbf{z} + \mathcal{O}(\mathbf{z}^3). \end{aligned} \quad (2.2.7)$$

Before we substitute both expansions back into (2.2.5), we need some assumptions on $\Phi(\rho, \mathbf{z})$. We assume that $\Phi(\rho, \mathbf{z})$ is a radially symmetric probability density with mean 0 and that the second moments exists, i.e.:

$$\int_{\mathbf{R}^2} \Phi(\rho(\mathbf{x}, t), \mathbf{z}) d\mathbf{z} = 1, \quad (2.2.8)$$

$$\int_{\mathbf{R}^2} z_1 z_2 \Phi(\rho(\mathbf{x}, t), \mathbf{z}) dz_1 dz_2 = 0, \quad (2.2.9)$$

$$\int_{\mathbf{R}^2} z_1^2 \Phi(\rho(\mathbf{x}, t), \mathbf{z}) dz_1 dz_2 = \int_{\mathbf{R}^2} z_2^2 \Phi(\rho(\mathbf{x}, t), \mathbf{z}) dz_1 dz_2 = m_2(\rho), \quad (2.2.10)$$

where $\mathbf{z} = (z_1, z_2)$ and m_2 is the second moment of Φ which explicitly depends on ρ . Now we can substitute expansions (2.2.6) and (2.2.7) into equation (2.2.5) and collect terms of the same order in \mathbf{z} . On the right hand side, we find at $\mathcal{O}(1)$:

$$\int_{\mathbf{R}^2} \rho(\mathbf{x}, s) \Phi(\rho(\mathbf{x}, s), \mathbf{z}) d\mathbf{z}, \quad (2.2.11)$$

which, by assumption (2.2.8), just equals $\rho(\mathbf{x}, s)$. At $\mathcal{O}(\mathbf{z})$ the right hand side of 2.2.5 becomes

$$- \int_{\mathbf{R}^2} \rho D(\Phi(\rho(\mathbf{x}, s), \mathbf{z})) \cdot \mathbf{z} + D\rho(\mathbf{x}) \cdot \mathbf{z} \Phi(\rho, \mathbf{z}) d\mathbf{z}. \quad (2.2.12)$$

We apply the chain rule to find $D(\Phi(\rho(x), \mathbf{z})) = \frac{\partial}{\partial \rho} \Phi(\rho, \mathbf{z}) D\rho$, then we pull $\rho, D\rho$ and the derivative with respect to ρ outside the integral so equation 2.2.12 becomes

$$-\rho D\rho \frac{\partial}{\partial \rho} \cdot \int_{\mathbf{R}^2} \Phi(\rho, \mathbf{z}) \mathbf{z} d\mathbf{z} - D\rho \cdot \int_{\mathbf{R}^2} \mathbf{z} \Phi(\rho, \mathbf{z}) d\mathbf{z}. \quad (2.2.13)$$

Both integrals are 0 by (2.2.9). Hence, the whole $\mathcal{O}(\mathbf{z})$ term disappears. Next comes the $\mathcal{O}(\mathbf{z}^2)$ term:

$$\int_{\mathbf{R}^2} \rho \frac{1}{2} \mathbf{z}^T D^2(\Phi(\rho(\mathbf{x}), \mathbf{z})) \mathbf{z} + \Phi(\rho, \mathbf{z}) \frac{1}{2} \mathbf{z}^T D^2 \rho(\mathbf{x}) \mathbf{z} + D\rho(\mathbf{x}) \cdot \mathbf{z} D(\Phi(\rho(\mathbf{x}), \mathbf{z})) \cdot \mathbf{z} d\mathbf{z}. \quad (2.2.14)$$

We can use assumption (2.2.9) to get rid of all the terms containing $z_1 z_2$. Then we are left with

$$\begin{aligned} & \frac{\rho}{2} \int_{\mathbf{R}^2} (z_1^2 \partial_{x_1}^2 \Phi(\rho, \mathbf{z}) + z_2^2 \partial_{x_2}^2 \Phi(\rho, \mathbf{z})) d\mathbf{z} + \frac{1}{2} (\Delta \rho) m_2(\rho) + \\ & \int_{\mathbf{R}^2} (z_1^2 \partial_{x_1} \rho \partial_{x_1} \Phi(\rho, \mathbf{z}) + z_2^2 \partial_{x_2} \rho \partial_{x_2} \Phi(\rho, \mathbf{z})) d\mathbf{z}. \end{aligned} \quad (2.2.15)$$

We apply the chain rule twice to $D^2(\Phi(\rho, \mathbf{z}))$ and pull again the derivatives with respect to ρ outside the integral. Putting everything together, we find that equation (2.2.5) up to $\mathcal{O}(z^2)$ becomes

$$\begin{aligned} \rho(\mathbf{x}, t) &= \rho(\mathbf{x}, s) + \frac{\rho}{2} ((\partial_{x_1} \rho)^2 + (\partial_{x_2} \rho)^2) \partial_\rho^2 m_2(\rho) + (\Delta \rho) \partial_\rho m_2(\rho) \\ &+ \frac{1}{2} (\Delta \rho) m_2(\rho) + ((\partial_{x_1} \rho)^2 + (\partial_{x_2} \rho)^2) \partial_\rho m_2(\rho) \\ &= \rho(\mathbf{x}, s) + \frac{1}{2} \Delta(m_2(\rho) \rho). \end{aligned} \quad (2.2.16)$$

When we take $\rho(\mathbf{x}, s)$ to the other side, divide by $t - s$ and take the limit of $t - s$ to zero, we end up with the familiar Fokker-Planck equation:

$$\frac{\partial \rho}{\partial t} = \frac{1}{2} \Delta(K_2(\rho) \rho), \quad (2.2.17)$$

where $K_2 = \lim_{t-s \rightarrow 0} \frac{m_2(\rho(\mathbf{y}, t-s))}{t-s}$. We can recognize this coefficient as a Kramer coefficient from the Fokker-Planck equation. Next, we must check whether or not the dispersal kernel from equation (2.2.2) fulfils our assumptions. $\Phi(\mathbf{z})$ is constructed to be rotationally symmetric and normalized so assumptions (2.2.8) and (2.2.9) hold. For assumption (2.2.10) to hold, the second moment of Φ must exist, which can be shown by computing $m_2(\rho)$:

$$\begin{aligned} m_2(\rho) &= \int_{\mathbf{R}^2} x_1^2 \frac{\lambda(\mathbf{y}, s)^2}{2\pi(t-s)} e^{-\frac{\lambda(\mathbf{y}, s)|\mathbf{x}-\mathbf{y}|}{\sqrt{t-s}}} d\mathbf{x} \\ &= \int_0^{2\pi} \int_0^\infty (r \cos(\theta))^2 \frac{\lambda(\mathbf{y}, s)^2}{2\pi(t-s)} e^{-\frac{\lambda(\mathbf{y}, s)r}{\sqrt{t-s}}} r dr d\theta \\ &= \frac{3(t-s)}{\lambda(\mathbf{y}, s)^2}. \end{aligned} \quad (2.2.18)$$

Hence, we find

$$K_2 = \lim_{t-s \rightarrow 0} \frac{3(t-s)}{(t-s)\lambda^2} = \frac{3}{\lambda^2}, \quad (2.2.19)$$

where λ is defined as in equation (2.2.3).

Finally, the derivation resulted in a nonlinear diffusion equation as an approximation for the IBM by Liu:

$$\frac{\partial \rho}{\partial t} = \frac{3}{2} \Delta \left(\frac{\rho}{g(A)^2} \right). \quad (2.2.20)$$

For simulations of this equation, see section 3.1.

2.2.1 Fokker-Planck equation for the IBM by De Jager

De Jager puts forward the hypothesis that a Lévy walk with parameter $\alpha = 1$ (we will explain this later) is the optimal search strategy. If this is true, we would like to see this in our continuous model as well. For that reason, we need to replace the exponential in the previous section by a heavy tailed probability density. At this point there are two problems. First, a pure Lévy walk P_α is only defined via it's Fourier transform:

$$\hat{P}_\mu = e^{-a|k|^\alpha}. \quad (2.2.21)$$

Unfortunately, there is no direct expression for the inverse Fourier transform of \hat{P} , but we do find for $x \rightarrow \infty$ that $P_\mu \sim x^{-(\mu+1)}$ which is the heavy tail we need. The second problem is that, even if we had an exact expression for the probability density, the method described above can not be used because the second moment of P_α does not exist. Before we continue with the next section and give a formal derivation of the continuous equation corresponding to a Lévy walk, lets first write down a master equation for the IBM as described by De Jager. She explicitly states that only the waiting times depend on A , not the jump length. We can model this following [25]. This time, the dispersal kernel $\Phi(z)$ is *independent* of t . Next, we need to introduce the jump rate $\lambda(\rho)$. This means that the probability that a mussel takes a step in the small time interval $[t, t + dt]$ is $\lambda(\rho(x, t))dt$.

Now we can write down the following balance equation:

$$\rho(x, t + dt) = dt \int_{\mathbf{R}^2} \lambda(\rho(x - z))\rho(x - z, t)\Phi(z)dz + (1 - dt\lambda(\rho(x, t)))\rho(x, t) + \mathcal{O}(dt^2). \quad (2.2.22)$$

Subtracting $\rho(x, t)$ from both sides, dividing by dt and taking the limit of $t \rightarrow 0$ results in the following mesoscopic equation:

$$\frac{\partial \rho}{\partial t} = \int_{\mathbf{R}^2} \lambda(\rho(x - z))\rho(x - z, t)\Phi(z)dz - \lambda(\rho(x, t))\rho(x, t). \quad (2.2.23)$$

Now we would like to use the same strategy as before: expand $\rho(x - z)$ and $\lambda(\rho(x - z))$ around $|x - z| \ll 1$ and compute all the integrals using the assumptions on Φ . However, since the second moment is infinite, we can not

use this strategy. Now, before we go to the next section and try to solve this problem, we assume for the moment that Φ does have a second moment m_2 . In that case we end up with the following equation [25]:

$$\frac{\partial \rho}{\partial t} = \frac{m_2}{2} \Delta(\lambda(\rho)\rho), \quad (2.2.24)$$

which is in shape similar to equation (2.2.20). Now, we have not said anything on $\lambda(\rho)$ at this moment. For example we could take $\lambda = 1/g(A)^2$ in which case the equation above would become identically to equation (2.2.20) or we could take the most simple jump rate $\lambda = 1 - g(A)$, but probably a large class of monotonic decreasing functions would work. Some more details on which choices for λ work or do not work can be found in section 3.3.

2.3 The fractional diffusion equation

In the previous section, we ran into trouble when we wanted to model the mussel movement by using a dispersal kernel without a second moment. What we need to learn from the derivation process in the previous section, is that as long as the second moment of a distribution exists, the corresponding macroscopic equation will be the diffusion equation, which only depends on the dispersion kernel via its second moment. We could have known this beforehand by invoking the Central Limit Theorem, which tells us that all random walks of which second moment exists, converge to a Gaussian process. The Central Limit Theorem heavily depends on the existence of the second moment. Therefore, we need a generalisation of the Central Limit Theorem [11]:

Generalized Central Limit Theorem. *The sum of n random variables on \mathbf{R} with power law distributions, i.e. distributions with a tail that decays as $\sim |x|^{-\alpha-1}$ with $0 < \alpha < 2$, converges to an α -stable distribution which is a distribution whose characteristic function ϕ is given by $\phi(t) = e^{-|ct|^\alpha}$.*

Now the definition of a Lévy distribution is exactly the distribution of which the characteristic function is given by $\phi(t) = e^{-|ct|^\alpha}$ for $0 < \alpha < 2$. Hence, we can expect that when we want to find a continuum approximation to De Jager's IBM, the PDE will have the same operator as the one that appears in the continuum approximation of a pure Lévy walk.

To derive the diffusion equation that corresponds to the Lévy distribution, we start, following [24], with a balance equation. We will write down a balance equation for a continuous time random walk (CTRW). Let $\phi(t)$ denote the probability that no step has occurred until time t and let $\Phi(z)$ denote the dispersal kernel, i.e. the distribution of the step sizes which we assume to be symmetric (a nonsymmetric kernel would induce an advection term in which we are not interested at the moment). Note that at this point we explicitly assume that $\phi(t)$ and $\Phi(z)$ do not depend on the density $\rho(x, t)$. When we denote with $\varphi(t)$ the probability density of taking a step, then by definition we have $\phi(t) = 1 - \int_0^t \varphi(t') dt'$. For the density $\rho(x, t)$ we now find the following balance equation:

$$\rho(x, t) = \underbrace{\rho(x, 0)\phi(t)}_{\text{Stay}} + \underbrace{\int_0^t j(x, t-u)\phi(u) du}_{\text{Arrival}}. \quad (2.3.1)$$

This equation states that the number of mussels (or density) at point x and time t equals the number of mussels that didn't leave plus the integral over u of all mussels that arrive at time $t - u$ but do not leave in the remaining time u . Therefore, $j(x, t)$ is not the density of mussels that are at x on time t , but the density of mussels that arrive exactly at x on time t . To get a closed equation, we also need an expression for $j(x, t)$ in terms of ρ :

$$j(x, t) = \int_{\mathbf{R}} \rho(x - z, 0) \Phi(z) \varphi(t) dz + \int_0^t \int_{\mathbf{R}} j(x - z, t - u) \Phi(z) \varphi(u) dz du. \quad (2.3.2)$$

The first integral describes the mussels that wait a time t at point $x - z$ before they make their first jump. The second integral describes the mussels that arrived at $x - z$ at time $t - u$, wait a time u and then arrive at x at exactly time t . Finding a closed equation for ρ by substituting equation (2.3.2) into (2.3.1) is difficult, but when we take the Fourier transform with respect to x and the Laplace transform with respect to t of both balance equations, solving for ρ has become much easier. When we recognize the integrals as convolutions, we can apply the convolution theorem for transforms and find

$$\begin{aligned} \rho(k, s) &= \rho(k, 0) \phi(s) + j(k, s) \phi(s), \\ j(k, s) &= \rho(k, 0) \Phi(k) \varphi(s) + j(k, s) \Phi(k) \varphi(s). \end{aligned} \quad (2.3.3)$$

Notice that we just write $\rho(k, s)$ when we mean $\mathcal{L}(\mathcal{F}(\rho))$, the variables should make clear what is meant. Now we can solve for $\rho(k, s)$:

$$\begin{aligned} \rho(k, s) &= \rho(k, 0) \phi(s) \frac{1}{1 - \Phi(k) \varphi(s)} \\ &= \rho(k, 0) \frac{1 - \varphi(s)}{s} \frac{1}{1 - \Phi(k) \varphi(s)}. \end{aligned} \quad (2.3.4)$$

This equation is also known as the Montrol-Weiss equation. This equation is extremely useful for deriving diffusion equations. The first thing we should realize is that $\varphi(s)$ is the moment generating function of $\varphi(t)$, provided it exists, and that $\Phi(k)$ is the characteristic function of $\Phi(x)$. Now we want to take the step from the mesoscopic level to the macroscopic level. This means that we are interested in longer length scales and time scales, which correspond to small s and k . We assume $\varphi(t)$ has a first moment $\langle t \rangle$ and $\Phi(x)$ has a second moment $\langle x^2 \rangle$ (remember that $\langle x \rangle = 0$ because we assumed $\Phi(x)$ to be symmetric). Then, by definition of the moment generating function and the characteristic function we have

$$\Phi(k) = 1 - \frac{1}{2} \langle x^2 \rangle k^2 + \mathcal{O}(k^4), \quad (2.3.5)$$

$$\varphi(s) = 1 - \langle t \rangle s + \mathcal{O}(s^2). \quad (2.3.6)$$

Before we substitute this back into equation (2.3.4) we first split the equation in a spatial and temporal part:

$$\frac{1 - \varphi(s)}{s \varphi(s)} (s \rho(k, s) - \rho(k, 0)) = (\Phi(k) - 1) \rho(k, s). \quad (2.3.7)$$

Now we can substitute both expansions into the equation above and then find up to order $\mathcal{O}(s, k^2)$

$$\langle t \rangle (s\rho(k, s) - \rho(k, 0)) = \frac{1}{2} \langle x^2 \rangle k^2 \rho(k, s). \quad (2.3.8)$$

Finally, we invert with Fourier-Laplace and recover the familiar diffusion equation:

$$\rho_t = \frac{\langle x^2 \rangle}{2\langle t \rangle} \Delta \rho. \quad (2.3.9)$$

Notice here that the only way φ and Φ influence the equation is via the diffusion constant $D = \frac{\langle x^2 \rangle}{2\langle t \rangle}$ and the fact that their first respectively second moment exists. Fortunately, we can derive a lot more from the Montrol-Weiss equation. As we have learned from the generalized central limit theorem, the characteristic function of all heavy tailed distributions will converge towards the characteristic function of the Lévy walk, $e^{-a|k|^\alpha}$. The second derivative of this function in $k = 0$, i.e. $\langle x^2 \rangle$ does not exist just as we expected, but we can still expand the characteristic function as $1 - a|k|^\alpha + \dots$. In this case, equation (2.3.7) becomes up to order $\mathcal{O}(s, k^\alpha)$

$$\langle t \rangle (s\rho(k, s) - \rho(k, 0)) = -ak^\alpha \rho(k, s). \quad (2.3.10)$$

We can define the fractional Laplacian $-(\Delta)^s$ as the operator that has the symbol $-|k|^{2s}$, which means

$$-(\widehat{(\Delta)^s} f)(x) = -|k|^{2s} \hat{f}(k). \quad (2.3.11)$$

For clarity, we will write $-(\Delta)^s = \Delta^s$. By this definition, it is now clear that when we invert equation (2.3.10) with Fourier-Laplace, we find the space-fractional diffusion equation:

$$\rho_t = \frac{a}{\langle t \rangle} \Delta^s \rho, \quad (2.3.12)$$

with $s = \alpha/2$. Now this equation might look like a “normal” PDE, but by the definition of Δ^s it really is an integrodifferential equation.

We will not exploit the other possibilities of the Montrol-Weiss equation here, but note that we can easily derive all kinds of other fractional diffusion equations. When we assume for example that φ has such a heavy tail that $\langle t \rangle$ does not exist, we still might be able to expand $\varphi(s)$ as $\varphi(s) = 1 - \langle t \rangle^\beta$ for $\beta < 1$, which would result in a time-fractional diffusion equation.

2.3.1 Application of the Montrol-Weiss equation for modelling the IBM’s

Now suppose that we want to use the formalism described above to the IBM by De Jager, then we would run into immediate trouble. The function $\varphi(t)$ would become a function of not just t , but also of $\rho(x, t_0)$. Therefore, when we apply the Fourier transform to $\rho(x, 0)\phi(t, \rho(x, 0))$ we unfortunately do not get $\rho(k, 0)\phi(t, \rho(k, 0))$. To our knowledge, there is no direct expression for the Fourier transform of $\rho(x, 0)\phi(t, \rho(x, 0))$ in terms of $\rho(k, 0)$ and $\phi(t, \rho(k, 0))$. Hence

we cannot derive a Montrol-Weiss equation and we are stuck. However, there is one lesson we must learn from the derivation of the fractional diffusion equation above. If we want to find a continuum approximation for a random walk without second moment, we should at least expect to find a fractional Laplacian.

2.4 Discretization of the fractional Laplacian

In this section we will derive a nonlinear fractional diffusion equation following Chaturapruek et al. [5], but first we invest some time in the study of the fractional Laplacian. The aim is to give a connection between the fractional Laplacian and a heavy tailed random walk, i.e. a Lévy walk.

We want to find a continuum approximation to a random walk (RW) that is both discrete in space and time. Therefore, let us define a spatial-temporal grid $\{x_j, t_n\} | j \in \mathbf{Z}, n \in \mathbf{N}\}$ with $x_j = jh$ and $t_n = n\tau$ for $\tau, h > 0$. We define p_k as the probability that a particle takes a jump of length k in a time step τ . Hence, all p_k are non-negative and sum up to 1. Then, the probability $y_j(t_n)$ that a particle is at position j at time t_n , must obey the following balance equation or transition law:

$$y_j(t_{n+1}) = \sum_{k=-\infty}^{\infty} p_k y_{j-k}(t_n), \quad y_j(0) = \delta_{j0}. \quad (2.4.1)$$

Now, how can we connect the discrete equation with the continuous equation below?

$$\rho_t = D^\alpha \rho, \quad 0 < \alpha \leq 2. \quad (2.4.2)$$

There are many different definitions of the fractional derivative D^α , but at this point it is useful to take the Grünwald-Letnikov derivative:

$$D_{\pm}^\alpha y_j = \lim_{h \rightarrow 0} \frac{1}{h^\alpha} \sum_{k=0}^{\infty} (-1)^k \binom{\alpha}{k} y_{j \pm 1 \mp k}. \quad (2.4.3)$$

Notice that when we take α to be 1 or 2, we recover the standard definition of the first and second derivative. D_{\pm}^α are called the direct and reverse Grünwald-Letnikov derivatives. Together they form the Riesz fractional derivative [26]:

$$D^\alpha = -\frac{D_+ + D_-}{2 \cos(\alpha\pi/2)}. \quad (2.4.4)$$

With this definition at hand, finding a finite difference scheme for equation (2.4.2) has become obvious:

$$\frac{y_j(t_{n+1}) - y_j(t_n)}{\tau} = \frac{-1}{h^\alpha} \frac{1}{2 \cos(\alpha\pi/2)} \left(\sum_{k=0}^{\infty} (-1)^k \binom{\alpha}{k} y_{j+1-k} + \sum_{k=0}^{\infty} (-1)^k \binom{\alpha}{k} y_{j-1+k} \right). \quad (2.4.5)$$

When we introduce the scaling $\mu = \tau/h^\alpha$, we can compare equation (2.4.1) with the equation above and find the following values for the p_k :

$$\begin{cases} p_0 &= 1 - \frac{\mu^\alpha}{|\cos(\alpha\pi/2)|}, \\ p_{\pm 1} &= \frac{\mu}{2|\cos(\alpha\pi/2)|} \left[\binom{\alpha}{k} + 1 \right], \\ p_{\pm k} &= (-1)^{k+1} \frac{\mu}{2|\cos(\alpha\pi/2)|} \binom{\alpha}{k+1}, \end{cases} \quad (2.4.6)$$

corresponding to the following difference scheme:

$$\frac{y_j(t_{n+1}) - y_j(t_n)}{\tau} = \frac{1}{\mu h^\alpha} \left((p_0 - 1)y_j(t_n) + \sum_{k \neq j} p_k y_{j-k}(t_n) \right). \quad (2.4.7)$$

In [12] it is shown that the RW defined above indeed does converge to the space-fractional diffusion equation. What can we learn from this result? We cannot claim that the RW executed by the mussels must satisfy equation (2.4.6). First, we have learned that the discretization of the fractional Laplacian at a point is dependent on the whole grid, not on a few neighbour grid points. Just as in a Lévy walk the mussels can jump to all other grid points in stead of a few neighbouring as in the classic random walk. Second, we do know that for large k the binomial $\binom{\alpha}{k+1} \sim |k|^{-(\alpha+1)}$, which is the heavy tail we are looking for. Therefore, we propose to take the most intuitive step probabilities to model a Lévy walk:

$$p_0 = 1 - 2\lambda\zeta(\alpha + 1), \quad p_k = \lambda|k|^{-(\alpha+1)} \quad \text{for } k \neq 0, \quad (2.4.8)$$

with $\zeta(\alpha + 1)$ the Riemann zeta function. In [12] it is shown that a RW with probabilities as defined above also converges to the space-fractional diffusion equation. However, this time we do not have the scaling $\tau = \mu h^\alpha$, but

$$\tau = \begin{cases} \frac{\lambda\pi}{\Gamma(\alpha+1)\sin(\alpha\pi/2)} h^\alpha & \text{if } 0 < \alpha < 2, \\ \lambda h^2 |\log(h)| & \text{if } \alpha = 2. \end{cases} \quad (2.4.9)$$

With this scaling and these step probabilities, the RHS of equation (2.4.5) becomes

$$\frac{y_j(t_{n+1}) - y_j(t_n)}{\tau} = \frac{-\Gamma(\alpha + 1) \sin(\alpha\pi/2)}{\lambda\pi h^\alpha} \left(-2\lambda\zeta(\alpha + 1)y_j(t_n) + \sum_{k \neq j} \lambda|j - k|^{-(\alpha+1)} y_k(t_n) \right). \quad (2.4.10)$$

By definition of the zeta function we can write $-2\lambda\zeta(\alpha + 1)$ as a summation and find

$$\begin{aligned} \frac{y_j(t_{n+1}) - y_j(t_n)}{\tau} &= \frac{-\Gamma(\alpha + 1) \sin(\alpha\pi/2)}{\lambda\pi h^\alpha} \left(\sum_{k \neq j} \frac{\lambda y_j(t_n) - \lambda y_k(t_n)}{|j - k|^{\alpha+1}} \right) \\ &= \frac{-\Gamma(\alpha + 1) \sin(\alpha\pi/2)}{\lambda\pi h^\alpha} h^\alpha \left(\sum_{k \neq j} \frac{\lambda y_j(t_n) - \lambda y_k(t_n)}{|x_j - x_k|^{\alpha+1}} h \right). \end{aligned} \quad (2.4.11)$$

Now we can take the limit of $\tau, h \rightarrow 0$ on both sides and find

$$\begin{aligned} \frac{\partial y(x, t)}{\partial t} &= \frac{-\Gamma(\alpha + 1) \sin(\alpha\pi/2)}{\pi} \int_{\mathbf{R}} \frac{y(x, t) - y(x', t)}{|x - x'|^{\alpha+1}} dx' \\ &= \Delta^{\alpha/2} y(x, t). \end{aligned} \quad (2.4.12)$$

with $\Delta^{\alpha/2}$ the fractional Laplacian as defined in [5]. Note that for the definition of the fractional Laplacian above, it does not matter when we replace in the

integral $y(x, t) - y(x', t)$ with $h(y(x, t)) - h(y(x', t))$ for any (integrable) function h , we will just find $\Delta^{\alpha/2} h(y(x, t))$. Therefore, our objective for the rest of this section is to write down a RW for the mussels and rewrite it as a sum as in equation (2.4.11). Following this procedure we should be able to derive a nonlinear fractional diffusion equation.

Before we continue, we must notice the following problem: definition (2.3.11) and (2.4.12) are equivalent when $s < 1$, [32]. When we take the limit of $s \rightarrow 1$ in the definition via the Fourier transform, we recover the standard Laplacian. However, in the integral definition we find that the constant $C_{1,2s}$ tends to zero because $\Gamma(-s)$ blows up for $\sigma \rightarrow 1$. This shouldn't come as a surprise, because in the derivation of the fractional diffusion equation we explicitly used that $s < 1$. When we go back to the scaling in equation (2.4.9) we see that we should have used $\tau \sim \ell^2 |\log(\ell)|$ instead of $\tau \sim \ell^2$. Hence, our derivation in this case is not valid. To correct this, we should either prove that the discrete step probabilities with the correct scaling indeed converge to the standard Laplacian or we must rely on the generalized central limit theorem to know that the discrete random walk for $s = 1$ converges to a Gaussian process.

2.4.1 Discrete approach in 1-d

We start our modelling on the same spatial-temporal grid as defined above: $\{x_j, t_n\} | j \in \mathbf{Z}, n \in \mathbf{N}\}$ with time step τ and grid spacing ℓ . Next, we write down the same equation for the attractiveness of a point x at time t as equation (2.1.1) where we denote with $N_i(t_n)$ the number of mussels at position x_j :

$$A_k(t_n) = \frac{D_s}{2R_s + 1} \sum_{|i-k| \leq R_s} N_i(t_n) - \frac{D_l}{2R_l + 1} \sum_{|i-k| \leq R_l} N_i(t_n). \quad (2.4.13)$$

For modelling purpose, we want $A(t)$ to be positive and therefore we introduce the same function g as in section 2.2,

$$g(A) = \frac{1}{2} + \frac{1}{2} \tanh(A), \quad (2.4.14)$$

to bound A between 0 and 1.

Rules for movement

Following Chaturapruek [5], the easiest model we can take is a model where the step probabilities are independent of $A_k(t_n)$, hence we assume p_{kk} to be constant, $p_{kk} = 1 - \lambda$ for $0 < \lambda < 1$ and $p_{ik} = \frac{\lambda}{z|i-k|^\mu}$ with $z = 2\zeta(\mu)$. μ is defined as $\mu = \alpha + 1$, where α is the Lévy parameter and 1 the dimension of the space. In this case we find as difference scheme (see equation (2.4.7)):

$$\frac{N_k(t + \tau) - N_k(t)}{\tau} = \frac{\lambda}{\ell^{\mu-1}} \sum_{i \in \mathbf{Z}, i \neq k} N_i(t) \frac{1}{z|i-k|^\mu} - \frac{\lambda}{\ell^{\mu-1}} N_k(t) \quad (2.4.15)$$

$$= \frac{\lambda}{\ell^{\mu-1}} \sum_{i \in \mathbf{Z}, i \neq k} \frac{N_i(t) - N_k(t)}{z|i-k|^\mu}, \quad (2.4.16)$$

so when we take the limit of $\ell, \tau \rightarrow 0$ over $\ell^{\mu-1}/\tau = D$ constant, we find

$$\frac{\partial N}{\partial t} = \lambda D \int_{\mathbf{R}} \frac{N(x) - N(y)}{|x - y|^\mu} dy, \quad (2.4.17)$$

which is up to some constants equal to equation (2.4.12), the fractional diffusion equation. This should not come as a surprise because we used the same step probabilities. The computation above served as an example for the following, more involved, case. Now we assume that when a mussel jumps away from a certain point x_k , it can jump to all other places, where the probability that it jumps to a certain point x_i decays as $1/|i - k|^\mu$ but grows as x_i becomes more attractive, hence we propose the following weights w_{ki} :

$$w_{ki}(t_n) = \frac{g(A_i(t_n))}{|i - k|^{\alpha+1}}, \quad \text{for } 0 < \alpha \leq 2 \quad (2.4.18)$$

and therefore the probabilities p_{ki} become

$$p_{ki}(t_n) = (1 - p_{kk}) \frac{w_{ki}}{\sum_{j \neq k} w_{kj}}. \quad (2.4.19)$$

With these probabilities defined, we can write down rules for the evolution of N , i.e. we write down balance equation (2.4.1)

$$N_k(t_{n+1}) = \sum_{i \in \mathbf{Z}, i \neq k} N_i(t_n) p_{ik} + p_{kk} N_k(t_n). \quad (2.4.20)$$

Numerical results for this equation can be found in section 3.2. A short look ahead to this section shows us that the discrete equation above does generate patterns. Therefore, we will continue in the next section with finding a continuum approximation for this random walk.

2.4.2 Continuum Limit

Now we would like to make the step from discrete space and time to continuous space and time, so we assume that $\ell, dt \ll 1$. We will start with the continuum limit of equation (2.1.1). When we introduce the length scale $r_\mu = \ell R_\mu$, $\mu = s, l$, and identify ℓ with Δx , then

$$\begin{aligned} \lim_{\ell \rightarrow 0} \frac{1}{2R_\mu + 1} \sum_{|i-k| \leq R_\mu} N_i(t) &= \lim_{\ell \rightarrow 0} \frac{1}{\frac{2r_\mu}{\ell} + 1} \sum_{|i-k| \leq R_\mu} N_i(t) \frac{\Delta x}{\ell} \\ &= \frac{1}{2r_\mu} \int_{x-r_\mu}^{x+r_\mu} N(x', t) dx', \end{aligned} \quad (2.4.21)$$

which we will write as a convolution:

$$A = D_s K_s * N - D_l K_l * N \quad (2.4.22)$$

where

$$K_\mu(x) = \frac{I_{[-r_\mu, r_\mu]}}{2r_\mu}. \quad (2.4.23)$$

The equation for N is more difficult to derive. First we define

$$z := 2 \sum_{k=1}^{\infty} \frac{1}{k^\mu} = 2\zeta(\mu), \quad (2.4.24)$$

and

$$\mathcal{L}(f_k) := \sum_{i \in \mathbf{Z}, i \neq k} \frac{f_i - f_k}{|i - k|^\mu \ell^\mu}. \quad (2.4.25)$$

We recognize this sum as the sum that will lead us to the fractional Laplacian, see equations (2.4.11)-(2.4.12). With this summation at hand we can write

$$\begin{aligned} \sum_{i \in \mathbf{Z}, i \neq k} w_{ik} &= \ell^\mu \sum_{k \in \mathbf{Z}, i \neq k} \frac{g(A_k) - g(A_i)}{|i - k|^\mu \ell^\mu} + \sum_{i \in \mathbf{Z}, i \neq k} \frac{g(A_i)}{|i - k|^\mu} \\ &= \ell^\mu \mathcal{L}(g(A_i)) + z g(A_i), \end{aligned} \quad (2.4.26)$$

so up to $\mathcal{O}(\ell^\mu)$ we find

$$p_{ik} = \frac{g(A_k)}{|i - k|^\mu} \frac{1}{\ell^\mu \mathcal{L}(g(A_i)) + z g(A_i)} \quad (2.4.27)$$

$$\approx \frac{g(A_k)}{|i - k|^\mu} \left(\frac{1}{z g(A_i)} - \frac{\mathcal{L}(g(A_i)) \ell^\mu}{z^2 g(A_i)^2} \right). \quad (2.4.28)$$

Following equation (2.4.10) we can write down the difference equation corresponding to equation (2.4.20)

$$\begin{aligned} \frac{N_k(t + \tau) - N_k(t)}{dt} &= \frac{1}{\ell^{\mu-1}} \sum_{i \in \mathbf{Z}, i \neq k} N_i(t) p_{ik} (1 - p_{ii}) + (p_{kk} - 1) N_k(t) \\ &\approx \frac{1}{\ell^{\mu-1}} \sum_{i \in \mathbf{Z}, i \neq k} N_i(t) \frac{g(A_k)}{|i - k|^\mu} \left(\frac{1}{z g(A_i)} - \frac{\mathcal{L}(g(A_i)) \ell^\mu}{z^2 g(A_i)^2} \right) \lambda - \lambda N_k(t_n) \\ &\approx \frac{\lambda}{\ell^{\mu-1}} \sum_{i \in \mathbf{Z}, i \neq k} N_i(t) \frac{g(A_k)}{|i - k|^\mu} \left(\frac{1}{z g(A_i)} - \frac{\mathcal{L}(g(A_i)) \ell^\mu}{z^2 g(A_i)^2} \right) - \lambda N_k(t_n). \end{aligned} \quad (2.4.29)$$

By definition of z we have

$$N_k = \sum_{i \in \mathbf{Z}, i \neq k} \frac{N_k}{z |i - k|^\mu}, \quad (2.4.30)$$

so we get

$$\begin{aligned} \frac{N_k(t + \tau) - N_k(t)}{\tau} &\approx \frac{\lambda g(A_k)}{\ell^{\mu-1}} \sum_{i \in \mathbf{Z}, i \neq k} \frac{\frac{N_i}{g(A_i)} - \frac{N_k}{g(A_k)}}{z |i - k|^\mu} - \frac{N_i}{|i - k|^\mu} \frac{\mathcal{L}(g(A_i)) \ell^\mu}{z^2 g(A_i)^2} \\ &\approx \frac{\lambda g(A_k)}{\ell^{\mu-1}} \left(\frac{\ell^\mu}{z} \mathcal{L}(N_k/g(A_k)) - N_k \frac{\mathcal{L}(g(A_k)) \ell^\mu}{z g(A_k)^2} \right). \end{aligned} \quad (2.4.31)$$

As we saw before, we can approximate the term $\mathcal{L}(f_k)$ by

$$\mathcal{L}(f_k) = \sum_{j \neq k} \frac{f_j - f_k}{|j - k|^\mu \ell^\mu} \approx \frac{1}{\ell} \int_{-\infty}^{\infty} \frac{f(y) - f(x)}{|y - x|^\mu} dy. \quad (2.4.32)$$

Now we use the same definition of the fractional Laplacian as in equation (2.4.12):

$$\Delta^s f(x) = C_{d,2s} \int_{y \in \mathbf{R}^d} \frac{f(x) - f(y)}{|x - y|^{2s+d}} dy, \quad C_{d,2s} = 2^{2s} \frac{\Gamma(s + d/2)}{\pi^{d/2} |\Gamma(-s)|}, \quad 0 < s \leq 1, \quad (2.4.33)$$

where we used $s = \frac{\mu-d}{2}$. Then, in 1 dimension, we find

$$\ell \mathcal{L}(A_k) \approx C_{1,2s}^{-1} \Delta^s(A). \quad (2.4.34)$$

Hence, the continuum limit for N becomes

$$\begin{aligned} \frac{\partial N}{\partial t} &= \frac{\lambda}{z C_{1,2s}} \left(g(A) \Delta^s \left(\frac{N}{g(A)} \right) - \frac{N}{g(A)} \Delta^s(g(A)) \right), \\ A &= D_s K_s * N - D_l K_l * N. \end{aligned} \quad (2.4.35)$$

Simulations of this equation can be found in section 3.2.2. At this point, one can ask why the scaling as defined in equation (2.4.9) was not used, i.e. with the constant $\frac{\lambda\pi}{\Gamma(\alpha+1)\sin(\alpha\pi/2)}$ in order to simplify the constant $\frac{\lambda}{z C_{1,2s}}$ above. In the first place, the scaling is defined such that the corresponding equation is exactly $N_t = \Delta^s N$, without any diffusion constants. From a modelling perspective we would like to keep these constants as they contain information from the underlying process. For example, if we take $\lambda = 0$, we find that $N_t = 0$. This means that the solution is constant, exactly what we expect because $\lambda = 0$ corresponds to $p_{kk} = 1$, i.e. nothing happens. In the second place, omitting the constants in the scaling enables us to focus on the more important details of the derivation, namely the fact that we should use $\tau \sim h^\alpha$ whereas we are used to $\tau \sim h^2$ in the normal diffusion process and how to use this to derive a continuum equation.

2.4.3 Continuum limit in 2-d

Now we move on to the 2 dimensional model. Fortunately, the derivation is to a great extent the same. In the discrete case, we must change all the summations from a summation over \mathbf{Z} to a summation over \mathbf{Z}^2 . So let us write down the two dimensional discrete model. We use the subscript (kl) instead of (k, l) to prevent the equation from becoming too cluttered:

$$N_{(kl)}(t_{n+1}) = \sum_{(ij) \in \mathbf{Z}^2 \setminus (kl)} p_{(ij),(kl)}(t_n) N_{(ij)}(t_n) + p_{(kl),(kl)} N_{(kl)}(t_n), \quad (2.4.36)$$

$$p_{(kl),(ij)} = (1 - p_{(kl),(kl)}) \frac{w_{(kl),(ij)}}{\sum_{(i'j') \in \mathbf{Z}^2 \setminus (kl)} w_{(kl),(i'j')}}}, \quad (2.4.37)$$

$$w_{(kl),(ij)} = \frac{g(A_{ij})}{\|(ij) - (kl)\|_2^\mu}, \quad (2.4.38)$$

with

$$A_{kl}(t_n) = \frac{D_s}{n_s} \sum_{|(ij)-(kl)| \leq R_s} N_{ij}(t_n) - \frac{D_l}{n_l} \sum_{|(ij)-(kl)| \leq R_l} N_{ij}(t_n), \quad (2.4.39)$$

where $|(ij) - (kl)|$ is the euclidean distance between (kl) and (ij) , and the n_s and n_l in the last line stand for the number of grid points within a circle with radius of the short interaction scale R_s respectively long interaction scale R_l . Simulations of this discrete model can be found in section 3.2.3.

For the attractiveness A , these equations result in the continuum limit in the equation

$$A(x, t) = \frac{D_s}{\pi r_s^2} \int_{B_{r_s}(x)} N(x', t) dx' - \frac{D_l}{\pi r_l^2} \int_{B_{r_l}(x)} N(x', t) dx', \quad (2.4.40)$$

which we again write as

$$A(x, t) = D_s K_s * N(x, t) - D_l K_l * N(x, t), \quad (2.4.41)$$

which means that we interpret the integral as a convolution of N with the indicator function K_s that is $\frac{1}{\pi r_s^2}$ on a circle with radius r_s around the origin.

The derivation of the continuous equation for N can almost completely be copied from the 1-d case, but there is a small detail in the derivation we have to take extra care of. In the 1-d case we defined $z = 2\zeta(\mu)$, which is defined for $\mu > 1$. Furthermore, we used that $\mu = \alpha + 1$ with α the Lévy parameter and 1 the dimension of space. The range $0 < \alpha < 2$, and therefore $1 < \mu < 3$, is exactly the range in which the heavy tail of the distribution is integrable, but has no second moment. In two dimensions, the tail needs to decay faster for a distribution to be integrable and therefore we define $\mu = \alpha + 2$ in \mathbf{R}^2 . Now to redo the derivation in section 2.4.2 we need to define a new z :

$$\begin{aligned} z &= \sum_{(i,j) \in \mathbf{Z}^2 \setminus \{0,0\}} \frac{1}{\|(i,j)\|_2^\mu} \\ &= 4\beta(\mu/2)\zeta(\mu/2), \end{aligned} \quad (2.4.42)$$

which converges, as expected, only for $\mu > 2$, see [2]. $\beta(z)$ is here the Dirichlet beta function. Next, we can define

$$\mathcal{L}(f_{kl}) = \sum_{(i,j) \in \mathbf{Z}^2 \setminus \{k,l\}} \frac{g(A_{kl}) - g(A_{ij})}{\|(i,j) - (k,l)\|^{\alpha+2} \ell^{2(\alpha+2)}}. \quad (2.4.43)$$

Fortunately, in the definition of the fractional laplacian (2.4.33) we find $\mu = 2s + d = \alpha + 2$. Hence, $\ell^{2\mu} \mathcal{L}(f_{kl})$ converges to $\ell^{2\mu-2} \Delta^{\alpha/2}$. Because $2\mu - 2 = 2\alpha$ we now see that we should take as scaling not $\tau \sim \ell^\alpha$ but $\tau \sim \ell^{2\alpha}$ which is what we expect in 2-d. Now putting everything together we arrive at the following equation:

$$\begin{aligned} \frac{\partial N}{\partial t} &= \frac{\lambda}{z C_{2,2s}} \left(g(A) \Delta^s \left(\frac{N}{g(A)} \right) - \frac{N}{g(A)} \Delta^s (g(A)) \right), \\ A &= D_s K_s * N - D_l K_l * N. \end{aligned} \quad (2.4.44)$$

For simulations of this equation, see section 3.2.4.

2.4.4 Simplifications

At this point one can ask why we define the p_{ik} so unnecessarily difficult and take the p_{kk} constant, while in the IBM by De Jager the p_{ik} are independent of A and the p_{kk} do depend on A . Before we answer this question, we write down a model corresponding to the IBM by De Jager. Therefore, we use the same step probabilities as we used in equation (2.4.16) but make the jump rate λ dependent of A , i.e. $p_{kk} = 1 - \lambda(A_k)$. In this case the difference scheme (2.4.7) becomes

$$\begin{aligned} \frac{N_k(t + \tau) - N_k(t)}{\tau} &= \frac{1}{\ell^{\mu-1}} \sum_{i \in \mathbf{Z}, i \neq k} \frac{\lambda(A_i(t))N_i(t)}{z|i-k|^\mu} - \frac{\lambda(A_k(t))}{\ell^{\mu-1}} N_k(t) \\ &= \frac{1}{\ell^{\mu-1}} \sum_{i \in \mathbf{Z}, i \neq k} \frac{\lambda(A_i(t))N_i(t) - \lambda(A_k(t))N_k(t)}{z|i-k|^\mu}, \end{aligned} \quad (2.4.45)$$

which results fractional diffusion equation

$$\frac{\partial N}{\partial t} = \frac{1}{zC_{d,2s}} \Delta^s (\lambda(A)N). \quad (2.4.46)$$

The derivation of this equation, as well as the equation itself, is much easier than the derivation of equation (2.4.44). However, there are multiple reasons we did not pursue this equation in the next chapters. The main reason is the fact that the most obvious choice for $\lambda(A_k)$, which is $\lambda(A_k) = 1 - g(A_k)$, does not produce the desired patterns as is shown in section 3.3, neither in the discrete case nor in the continuum limit. Another less mathematical reason is that some errors in our code for the discrete equations (see section 3.2) resulted in homogeneous solutions instead of patterns independent of the choice of $\lambda(A)$. In section 3.3 we will discuss the choice of $\lambda(A)$ in more detail.

2.5 Langevin Equations

In the introduction of this chapter, we talked about the different levels of modelling. Until now we have treated the mesoscopic and macroscopic approach, but not the stochastic differential equation approach. In a paper by Martínez-García et al. [21] the pattern formation in mussel beds has been modelled via a Langevin equation. Using this approach they derived the following equation:

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= D_0 \Delta (g(\rho)\rho), \\ g(\rho) &= \frac{1 + \tanh[2(a - b\rho_s + c\rho_l) - 1]}{2}, \end{aligned} \quad (2.5.1)$$

where ρ_s and ρ_l are the short respectively long scale densities the same way as we defined them in equation (2.1.2). Simulations of this equation can be found in the paper itself or in figure 3.12.

Their approach is to start with a Langevin equation describing the movement of N individual particles moving according to the equation

$$\mathbf{r}_i = \sqrt{2D(\mathbf{r}_i, \rho_s, \rho_l)} \eta_i(t), \quad (2.5.2)$$

where \mathbf{r}_i is the position of the i -th particle and $D = D_0 g(\rho)$. $\eta_i(t)$ is Gaussian white noise with spatially zero mean and the time correlation matrix is given by $\langle \eta_i(t) \eta_i(t') \rangle = \delta_{ij} \delta(t - t')$ so we can understand $\eta_i(t)$ as a Wiener process W_t . Of course, equation (2.5.2) has no meaning in its present form, but should be understood within the Itô calculus. In the Itô-language the equation becomes

$$dX_i(t) = \sqrt{2D(X_i, \rho_s, \rho_l)} dW_i(t). \quad (2.5.3)$$

If we drop the dependence on i and assume that $X(t)$ is a stochastic process describing the density of our system, we can write down a Fokker-Planck equation immediately:

$$\frac{\partial \rho}{\partial t} = \Delta (D(\rho, \rho_s, \rho_l) \rho), \quad (2.5.4)$$

which is the equation found in [21]. However, we do not have a single mussel but a bunch of them. The authors state that they derived their equation from the Langevin equation by following Dean's approach [9] and applying a mean-field approximation. What this effectively means is the following: we define the density of a single particle as

$$\rho_i(x, t) = \delta(X_i(t) - x) \quad (2.5.5)$$

and the density of all particles as

$$\rho(x, t) = \sum_{i=1}^N \rho_i(x, t). \quad (2.5.6)$$

Dean's approach can be used to derive a stochastic differential equation for this density ρ

$$\frac{\partial \rho(x, t)}{\partial t} = \nabla \left(\sqrt{2D(x, \rho_s, \rho_l)} \eta(t) \sqrt{\rho(x, t)} \right) + \Delta D(x, \rho_s, \rho_l) \rho(x, t). \quad (2.5.7)$$

To make a mean-field approximation, we define $\tilde{\rho} = \langle \rho \rangle$, the average of many different realisations for ρ . For $\tilde{\rho}$ we get

$$\frac{\partial \tilde{\rho}(x, t)}{\partial t} = \Delta \langle D(x, \rho_s, \rho_l) \rho(x, t) \rangle. \quad (2.5.8)$$

We can see that at this point we lose a lot of information about the individual trajectories of the particles. Where ρ is still a summation over δ -peaks, $\tilde{\rho}$ has become a continuous function. Now the last step to take is to find an approximation for $\langle D(x, \rho_s, \rho_l) \rho(x, t) \rangle$. Martínez-García et al. took the most basic approximation and approximated $\langle D(x, \rho_s, \rho_l) \rho(x, t) \rangle$ with $D(x, \tilde{\rho}_s, \tilde{\rho}_l) \tilde{\rho}(x, t)$.

Now the final equation looks similar to equation (2.2.20), but the solutions are very different. More on the difference between the equations can be found in section 3.3.

Numerical analysis of the different equations

In this chapter we will simulate the different models derived in the previous chapter, both the continuous models and the discrete ones. We will focus on the evolution of the equations to their final states, i.e. coarsening, and on the wavelengths of these final states.

3.1 Simulations of the continuous approximations of Liu's IBM

In section 2.2 we derived the following equation as a continuous approximation for Liu's IBM:

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= \frac{3}{2} \Delta \left(\frac{\rho}{g(A)^2} \right), \\ g(A) &= \frac{1}{2} + \frac{1}{2} \tanh(A), \\ A &= D_s K_s * \rho - D_l K_l * \rho. \end{aligned} \tag{3.1.1}$$

See equations (2.2.20),(2.2.3) and (2.2.4). First, we rescale the time t with $3/2$ to get rid of the constant in front. Before we can start simulating, there are three main questions we have to solve. How do we code A , how do we discretize the equation and what kind of boundary conditions do we use? The equation was derived on the whole of \mathbf{R}^2 and therefore we decided to take as domain $\Omega = [0, L]^2$ with periodic boundary conditions. Another advantage of taking periodic boundary conditions is that it directly tells us how to calculate A near the boundary. If a point is closer than r_l to the boundary, we would need information from outside Ω to calculate A . However, when we periodically extend our domain, we can use this extension to calculate A near the border.

The main question is how to code A ? In section 2.4.3 we interpret the integrals in A as convolution of ρ with the step functions K_s and K_l . Convolution is implemented in most programming languages, but one should come up with smart ways to code the kernels K_s and K_l , and one has to take care of the boundary conditions and this can be tricky. However, what the operation K_s* actually does is taking the average around a point and that is exactly what one would do to smooth edges in a picture. Therefore, we can use functions from

the Matlab image processing toolbox¹. The following block of code shows the basics for the computation of A :

```

1 N = 150; %Number of grid points
2 L=3; %length of the grid
3 gridsize =L/N;
4 Ds=1.2;
5 Dl=1;
6 long=1; %length of the long interaction
7 short=1/7; %length of the short interaction
8 nl=floor(long/gridsize); %radius of long scale filter
9 ns=floor(short/gridsize); %radius of short scale filter
10 hl = fspecial('disk',nl); %long scale filter
11 hs = fspecial('disk',ns); %short scale filter
12 Uavl = imfilter(U,hl,'circular'); %Convolution of U with hl
13 Uavs = imfilter(U,hs,'circular'); %Convolution of U with hs
14 A=Ds*Uavs - Dl*Uavl;

```

U is the numerical solution, U_{avl} is ρ_l and U_{avs} is ρ_s . The option `Circular` ensures us that Matlab uses periodic boundaries to compute the convolution. The next part is the discretization of the equation. For this we use Euler forward in time and a spectral method in space. A spectral method means that we use the Fourier transform \mathcal{F} to compute $\Delta(\rho/g(A)^2)$ as follows:

$$\Delta \frac{\rho}{g(A)^2} = \mathcal{F}^{-1} \left(-|k|^2 \mathcal{F} \left(\frac{\rho}{g(A)^2} \right) \right). \quad (3.1.2)$$

To implement this, we first need a discrete representation of $|k|^2 = |(k_x, k_y)|^2$:

```

1 q=2*pi/L*[0:N/2-1,-N/2:-1]; %Vector of wavenumbers in 1-d
2 [X,Y]=meshgrid(q,q);
3 absq2=sqrt(X.^2+Y.^2).^2; %absolute value squared of
4 %the wavenumbers
5 laplace_U=ifft2(-absq2.*fft2(U./gA.^2)); %compute Δ(U/g(A)^2)
6 Unew=U+dt*laplace_U; %One Euler time step forward

```

The rest of the implementation is then straightforward, the complete code can be found in appendix B.

The next thing we need are values for the parameters $\rho_0, r_s, r_l, D_s, D_l$. The density ρ_0 is just as in [19] our main bifurcation parameter so we do not set a value at this point. Furthermore, it is very hard to pin down an exact relation between the densities in the biological experiments and in our simulations. Also, about the experimental values is no agreement in the literature. De Jager [8] uses the following parameters:

$$\begin{aligned} r_s &= 1.1 \text{ cm}, \\ r_l &= 7.5 \text{ cm}, \\ D_s &= 1.3, \\ D_l &= 1.1, \end{aligned} \quad (3.1.3)$$

¹Note: You can only use this toolbox if your institution paid for it. If you work the Mathematical Institute in Leiden you'll have to code it yourself.

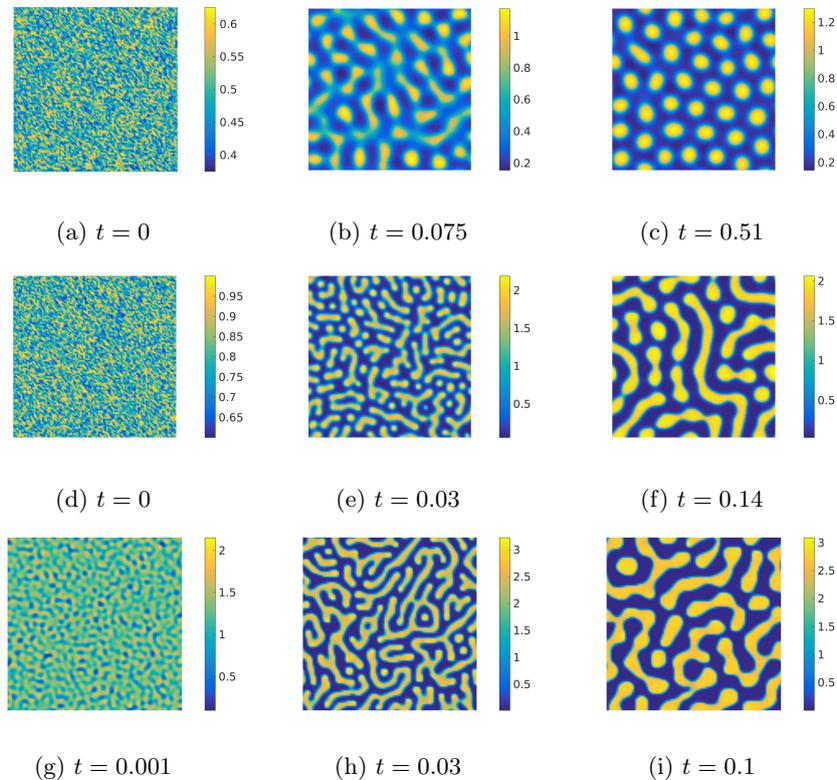


Figure 3.1: Evolution of the solutions of equation (3.1.1) for $\rho_0 = 0.5$ (top 3), for $\rho_0 = 0.8$ (middle 3) and $\rho_0 = 1.2$ (bottom 3). The other parameters are $r_s = 1/14$, $r_l = 0.5$, $D_s = 1.2$, $D_l = 1$ and $L = 4$. The grid contains 100 by 100 points and $dt = 10^{-7}$.

while Liu [20] uses

$$\begin{aligned}
 r_s &= 1.5 \text{ cm}, \\
 r_l &= 6.0 \text{ cm}, \\
 D_s &= 100, \\
 D_l &= 80.
 \end{aligned}
 \tag{3.1.4}$$

As we shall see later on in chapter 4, we can rescale equation (3.1.1) such that the ratios r_s/r_l and D_s/D_l are our only parameters. As we can see, for the ratios D_s/D_l are quite close and the ratios r_s/r_l differ less than a factor 2. In the simulation we will use mostly $D_s = 1.2$, $D_l = 1$, $r_l = 1/2$ and $r_s = 1/14$. The reason we take relatively small values for r_l and r_s is that the computation of the convolution with `imfilter` is considerably slower when we increase r_s and r_l .

Figure 3.1 shows the evolution of a solution for different ρ_0 . We start out with a randomly perturbed homogeneous state with value ρ_0 and observe how the patterns evolve. As we can see, for $\rho_0 = 0.5$ a pattern of evenly distributed clumps of mussels appears. When we increase the initial density to $\rho_0 = 0.8$

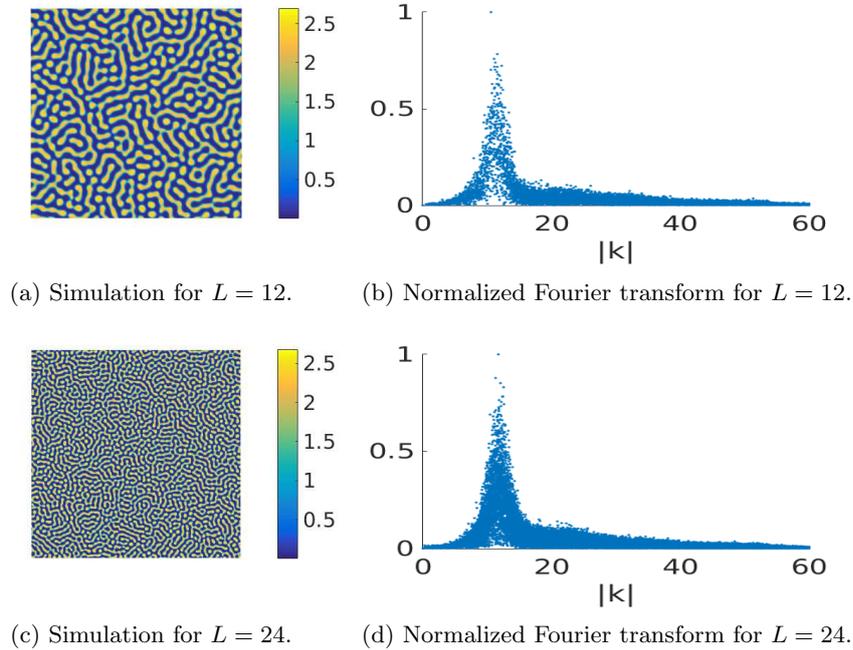


Figure 3.2: Simulation of equation (3.1.1) for large domain size: $L = 12$ and $L = 24$. The other parameters are kept at $\rho_0 = 0.8$, $r_s = 1/14$, $r_l = 0.5$, $D_s = 1.2$ and $D_l = 1$. We show both the simulations and the normalized Fourier transform.

or even $\rho_0 = 1.2$, we see that more elongated and more complicated structures arise, just as we have seen in the mussel beds. An interesting observation is that we definitely see coarsening, especially for $\rho_0 = 0.8$ and $\rho_0 = 1.2$.

Now before we can continue, we must first check whether or not the domain size has any influence on the shape of the patterns. We must be sure that the main wavelength only depends on the interplay between the short- and long-scale interaction. To test this, we increased the domain size from $L = 4$ in the previous simulation to $L = 12$ and $L = 24$, see figure 3.2. Not only visually both solutions are similar, also the normalized Fourier transforms show a striking similarity between both cases, which strengthens our beliefs that the patterns, and especially the dominant wavelength, are independent of the domain size.

3.1.1 Stationary solutions

For the equation

$$\frac{\partial \rho}{\partial t} = \Delta \left(\frac{\rho}{g(A)^2} \right), \quad (3.1.5)$$

a direct computation shows that the (implicit) solution $\rho = cg(A)^2$ for some $c \in \mathbf{R}_+$ is a stationary solution. Since we are interested in the spatial scale of

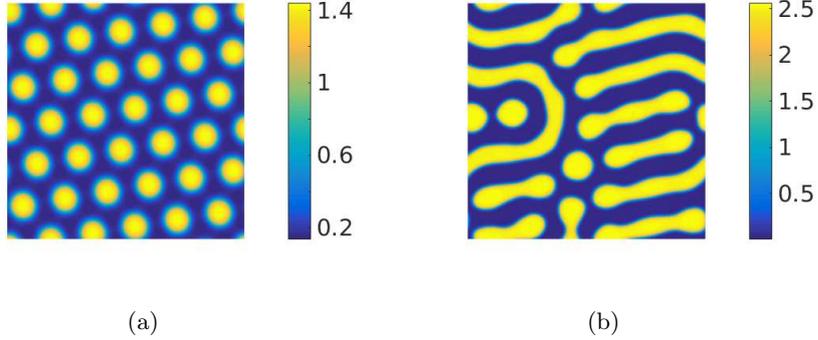


Figure 3.3: Simulations of the stationary solutions of equation (3.1.1) via the iterative method (equation (3.1.7)), $D_s = 1.2$, $D_l = 1$, $r_s = 1/14$, $r_l = 1/2$, $L = 4$ with 144^2 grid points. Both simulations are run until $\|\rho_{n+1} - \rho_n\|_2 < 10^{-6}$. (a) Stationary solution for $\rho_0 = 0.5$. The method converged to a perfect hexagonal pattern very fast. (b) The simulation for $\rho_0 = 0.9$. Convergence in this case is very slow and no distinctive shape has formed.

the stationary solution, we can try to see if we can solve the equation

$$\rho = cg(A)^2, \quad (3.1.6)$$

via an iterative procedure

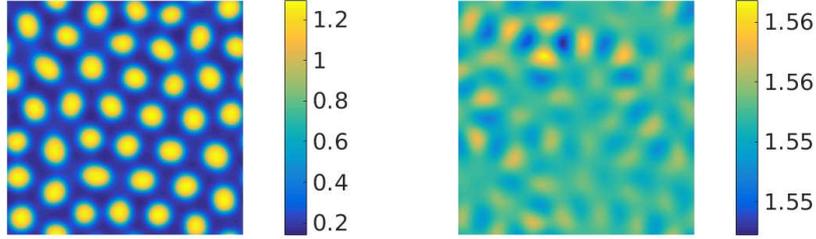
$$\rho_{n+1} = cg(A(\rho_n))^2. \quad (3.1.7)$$

First, we need to determine c . By conservation of mass, we know that when we integrate both sides of equation (3.1.6) over the whole domain we should get the constant ρ_0 . We make the educated guess that the conservation relation also holds for the constant solution ρ_0 so we can compute

$$c = \frac{\rho_0}{(0.5 + 0.5 \tanh(D_s \rho_0 - D_l \rho_0))^2}. \quad (3.1.8)$$

The results of this iterative procedure are shown in figure 3.3.

Next, we must check if the solutions of equation (3.1.1) indeed converge to $\rho/g(A)^2 = c$ with c as given above. Figure 3.4 shows ρ and $\rho/g(A)^2$ after a long run. As figure 3.4b shows, $\rho/g(A)^2$ has almost become constant. Unfortunately, it did not converge to the value for c expected by equation (3.1.8) but to a value close by. For the constants mentioned in figure 3.4, we expect that $c = 1.65$ but we find $c \approx 1.56$. Therefore, we can use the iterative procedure to study the stationary states, but only when we use the experimentally found value for c . One should notice the similarities and differences between the hexagonal patterns in figures 3.3 and 3.4. Via the iterative procedure, perfect circles are formed at a regular distance, while the solution via the PDE is not that regular. This is of course to be expected as we are limited in the time we can simulate the PDE.



(a) Solution at $t = 0.8$.

(b) Plot of $\rho/g(A)^2$ for ρ given in the figure on the left.

Figure 3.4: Simulations of equation (3.1.1) for $\rho_0 = 0.5$, $D_s = 1.2$, $D_l = 1$, $r_s = 1/14$, $r_l = 1/2$ and $L = 4$. We used $dt = 10^{-5}$ and a grid of 100 by 100 points.

3.1.2 Coarsening

Visual inspection of the solutions, especially for $\rho_0 = 0.8$ and $\rho_0 = 1.2$, see figure 3.1, shows us that the solutions evolve from patterns with a small wavelength to patterns with a larger wavelength. To make this more concrete, we redo the simulations and compute the Fourier transform at each time step. Because we are not interested in the specific direction (k_x, k_y) of a wave but rather in its absolute value $|k|$, we plot the Fourier transform versus $|k|$ instead of (k_x, k_y) . Furthermore, to keep the plots more insightful, we set the $k = 0$ mode to zero and then normalize the Fourier transform such that the maximum is 1. In Matlab code:

```

1 uhat = abs(fft2(U)); %absolute value of the 2-d FT
2 uhat(1,1)=0; %set zero mode to 0
3 uhat = uhat/max(max(abs(uhat))); %Normalize transform

```

Figure 3.5 shows the Fourier transform at multiple times in the evolution for $\rho_0 = 1.2$. We can clearly see that at the start multiple wavelengths compete and small wavelengths dominate. As the solution evolves, we can see the coarsening as the larger wavelength starts to dominate.

Now we have seen that there actually is coarsening, we should ask ourselves the question how it compares to the coarsening observed by Liu in [19], see e.g. figure 1.5. Therefore, at every time step we compute the Fourier transform (a few are shown in figure 3.5) and determine the absolute value at which the Fourier transform is maximal so that we can plot the dominant wavelength versus time as in figure 3.6. From figure 3.6 it is difficult to jump to any direct conclusions on the coarsening rate. The graph consists mainly of a few large jumps instead of multiple small steps. However, a quick fit with Matlab shows that the best fit to the graph has slope -0.28. Therefore, it is not unreasonable to assume that coarsening on average can be described with a power law with an exponent near -1/3. Furthermore, we should not forget that the computations of the coarsening rate as done in [19], has been done with a far larger number

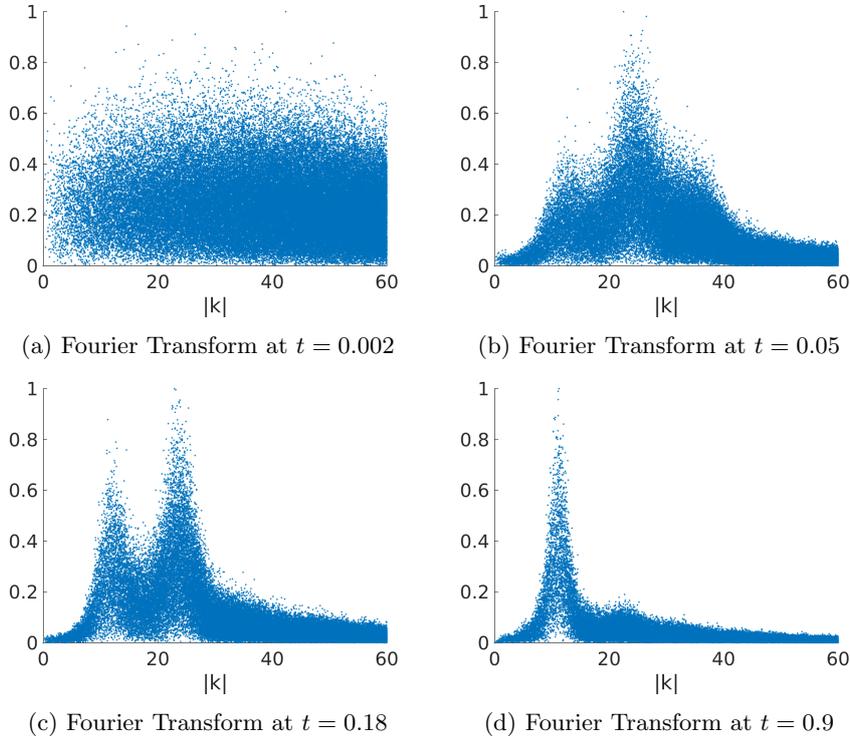


Figure 3.5: Normalized Fourier transforms of the solution to equation (3.1.1) for $\rho_0 = 1.2$, $r_s = 1/14$, $r_l = 0.5$, $D_s = 1.2$, $D_l = 1$ and $L = 24$.

of grid points which allows for a much more precise computation of the Fourier transform.

3.1.3 Wavelength of the stationary solution

In the previous section we have seen that the coarsening in the solution does not go on but stops at a certain wavelength. We would like to see how the wavelength depends on ρ_0 for fixed D, r_s and r_l . Trying to find the stationary solution by simulating the PDE until the stationary solution has been reached, would take far too much computation time. Therefore, we use the iterative method described above to compute the dominant wavelength of the stationary solutions. We see that in figure 1.5 that there is a slight increase in wavelength as ρ_0 increases, although the difference is just a couple of millimetres which is smaller than a mussel. We do not see any significant dependence on the density in our experiments. We are now interested in how the dominant wavelength depends on r_s and r_l . Tables 3.1 shows that when we fix r_l but vary r_s , there is almost no change in wavelength, while keeping r_s fixed and varying r_l gives a major change in wavelength. Therefore, we conclude that the main influence of the wavelength is the long interaction scale. This shouldn't come as a surprise, because the long interaction scale determines when a mussel decides "this group is large enough, I'm out."

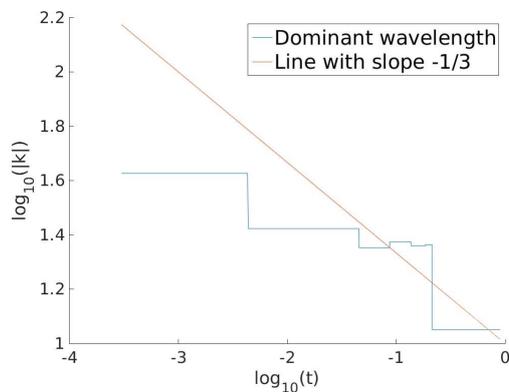


Figure 3.6: Log-Log plot of the dominant wavenumber versus time. The straight line has a slope of $-1/3$.

r_s	0.0800	0.1200	0.1600	0.2000	0.2400	0.2800	0.3200	0.4000
$ k $	5.1144	5.0139	4.8114	4.7065	4.7925	4.7925	4.2226	4.2226

r_l	0.6000	0.6400	0.6800	0.7200	0.8000	0.8400	0.8800	0.9600
$ k $	8.0830	7.6208	6.7776	6.5391	5.9568	5.8185	5.8701	4.7925

Table 3.1: The top table shows the short length scale versus the dominant wavenumber for fixed $r_l = 1$, the bottom table shows the long interaction length r_l versus the dominant wavenumber $|k|$ with r_s fixed at $r_s = 0.12$. The other parameters were $\rho_0 = 0.7$, $D_s = 1.2$, $D_l = 1$ and $L = 4$.

3.1.4 Conclusions

There are a lot of similarities between the simulations of equation (3.1.1) and the experimentally observed patterns. In the first place we notice that for low densities no patterns form, for intermediate densities we see spots and for high densities we see labyrinth-like structures. Both the mussels and the equation exhibit coarsening over time with approximately the same speed. Furthermore, in contrast with the Cahn-Hilliard equation, the coarsening in equation (3.1.1) stops and patterns with fixed wavelengths form just as we see in the mussel beds.

3.2 Simulations of the Lévy models

In this section we simulate the model as derived in section 2.4, starting in 1-d and then move on to 2-d.

3.2.1 Simulations of the discrete model in 1-d

In this section, we simulate equation (2.4.20) describing the mussel density N as derived in section 2.4.1:

$$\begin{aligned}
 N_k(t_{n+1}) &= \sum_{i \in \mathbf{Z}, i \neq k} N_i(t_n) p_{ik} + p_{kk} N_k(t_n), \\
 p_{ki}(t_n) &= (1 - p_{kk}) \frac{w_{ki}}{\sum_{j \neq k} w_{kj}}, \\
 w_{ki}(t_n) &= \frac{g(A_i(t_n))}{|i - k|^{\alpha+1}}, \quad \text{for } 0 < \alpha \leq 2, \\
 g(A) &= \frac{1}{2} + \frac{1}{2} \tanh(A), \\
 A_k(t_n) &= \frac{D_s}{2R_s + 1} \sum_{|i-k| \leq R_s} N_i(t_n) - \frac{D_l}{2R_l + 1} \sum_{|i-k| \leq R_l} N_i(t_n).
 \end{aligned} \tag{3.2.1}$$

The implementation of the discrete equation (3.2.1) is straightforward. After every time step, we compute A , $g(A)$, the vector p_{kk} and the matrix w_{ik} . Next, we normalize w_{ik} by dividing each row by its row sum to get the transition matrix p_{ik} which we can use to update the vector N . The only important detail is that we use a periodic grid, so we need to extend N periodically to make sure that the discrete convolution in equation (2.1.1) is computed correctly. In the derivation of the discrete model we did not specify the p_{kk} , but in section 2.4.2 we made a continuum approximation with p_{kk} constant, i.e. independent of k . Simulations for different μ are shown in figure 3.7, together with the continuum approximation. We will get back to this figure in the next section where we explain the simulation of the continuous equation.

3.2.2 Simulations of the continuous models in 1-d

First, let us state equation (2.4.35) again, which is the continuum approximation for the discrete system (3.2.1) above:

$$\begin{aligned}
 \frac{\partial N}{\partial t} &= \frac{\lambda}{zC_{1,2s}} \left(g(A) \Delta^s \left(\frac{N}{g(A)} \right) - \frac{N}{g(A)} \Delta^s(g(A)) \right), \\
 A &= D_s K_s * N - D_l K_l * N, \\
 K_\mu(x) &= \frac{I_{[-r_\mu, r_\mu]}}{2r_\mu}, \quad \mu = s, l.
 \end{aligned} \tag{3.2.2}$$

Here, $K_s * N$ is the convolution of N with the kernel K_s . Remember that the s in the fractional Laplacian is linked to μ in the discrete equation (3.2.1) via $s = \frac{\mu-d}{2}$ with d the dimension of the space (in this case 1). The implementation of the fractional derivative can be done in multiple ways, but a fast method

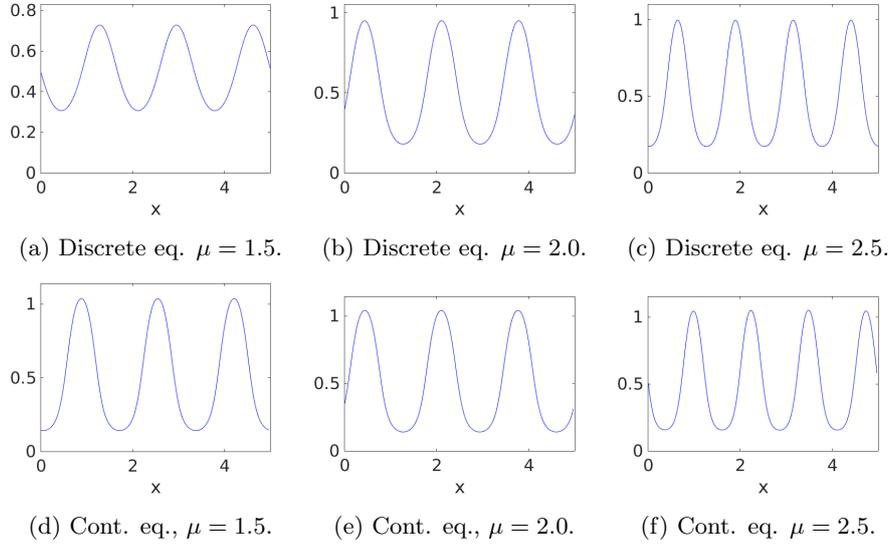


Figure 3.7: Simulations in 1-d of the discrete model (3.2.1) and the continuous model (3.2.2) for different μ . The other parameters are $D_s = 1.2$, $D_l = 1$, $r_s = 1/7$, $r_l = 1$, $\rho_0 = 0.5$ and $p_{kk} = 0.1$.

that is straightforward to code is to approximate the fractional Laplacian via the definition in Fourier space:

$$\mathcal{F}(\Delta^s f) = -|k|^{2s} \mathcal{F}(f). \quad (3.2.3)$$

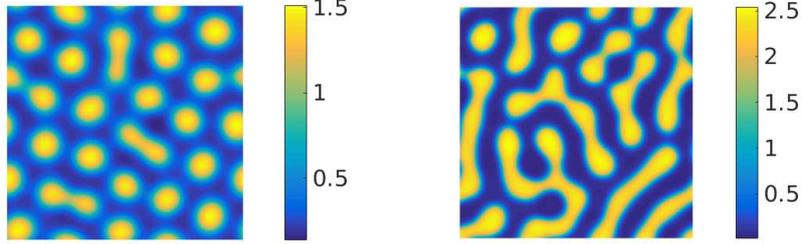
Now we can, for example, compute $\Delta^s(f)$ in matlab as

```

1 q=2*pi/L*[0:K/2-1,-K/2:-1]; %vector of wavenumbers
2 laplaces_f=ifft(-abs(q)^(2*s).*fft(f)) %Delta^s(f)

```

Note that this is exactly the same code as in section 3.1, only the 2 in the exponent of $|k|$ has been replaced with $2 * s$. Some results are shown in figure 3.7. When we compare the top 3 figures in figure 3.7 with the bottom 3 figures, we see that there is a good agreement between the discrete and the continuous case apart from a small phase shift and a small difference in amplitude. The phase shift can be explained due to the fact that the equation is invariant under translation. However, figure 3.7 might be misleading in suggesting that there is one unique solution (up to a phase shift) and that the discrete and continuous equations converge to these qualitatively the same solutions, but this is not the case. Two different solutions are found in the numerics of the discrete and continuous case. One with 3 peaks and one with 4 peaks. Compare in figure 3.7 the solutions for $\mu = 2.5$ with the other cases. Both solutions are stationary solutions in the sense that in the numerics the difference between two time steps has become less than machine precision. Numerically it is difficult to say if one of these is the stable solution and the other a metastable solution. Similar kind of problems have been noticed in the Cahn-Hilliard equation in 1-d. Numerically it was impossible to determine between global and local minimizers of the energy, a problem that disappeared in two dimensions [15]. Therefore, we learned from



(a) Solution after 1300 iterations for $\rho_0 = 0.6$. (b) Solution after 200 iteration for $\rho_0 = 1$.

Figure 3.8: Two simulations of the discrete two-dimensional model. Both simulations were performed on a grid with 100 by 100 points. $D_s = 1.2$, $D_l = 1$, $r_s = 2$, $r_l = 14$ and $\mu = 3$.

this section that there is a good agreement between the discrete and continuous equations, but discussion of the actual shape of the patterns must be postponed to the 2-d equations.

3.2.3 Simulations of the discrete model in 2-d

First, let us write down the two dimensional equivalent of equation (3.2.1), as derived in section 2.4.3, equations (2.4.36-2.4.39):

$$\begin{aligned}
 N_{(kl)}(t_{n+1}) &= \sum_{(ij) \in \mathbb{Z}^2 \setminus (kl)} p_{(ij),(kl)}(t_n) N_{(ij)}(t_n) + p_{(kl),(kl)} N_{(kl)}(t_n), \\
 p_{(kl),(ij)} &= (1 - p_{(kl),(kl)}) \frac{w_{(kl),(ij)}}{\sum_{(i'j') \in \mathbb{Z}^2 \setminus (kl)} w_{(kl),(i'j')}}}, \\
 w_{(kl),(ij)} &= \frac{g(A_{ij})}{\|(ij) - (kl)\|_2^\mu}, \quad 2 < \mu < 4, \\
 A_{kl}(t_n) &= \frac{D_s}{n_s} \sum_{|(ij) - (kl)| \leq R_s} N_{ij}(t_n) - \frac{D_l}{n_l} \sum_{|(ij) - (kl)| \leq R_l} N_{ij}(t_n).
 \end{aligned} \tag{3.2.4}$$

Just as in the 1-d case, we assume $p_{(kl),(kl)}$ to be a fixed constant between 0 and 1 for all (kl) . The numerical implementation for the discrete 2-d case is not as straightforward as the 1-d case. In the 1-d case, the w_{ik} formed an N by N matrix where N is the number of grid points. However, in 2-d we have a grid of N by N points so the $w_{(ij),(kl)}$ now would form a 4 dimensional matrix. We can solve this problem in the following way: For every point (ij) in the sum in the top line of equation (3.2.4) the matrix $p_{(ij),(kl)}$ is computed and then $p_{(ij),(kl)}(t_n) N_{(ij)}(t_n)$ is added to $N_{(kl)}(t_{n+1})$. This process is then repeated for all (ij) in the grid and this makes the computation very slow compared with the 1-d simulations.

The result for two different values of ρ_0 is shown in figure 3.8. As we expect, for a low value of ρ_0 we see spots and for a higher value of ρ_0 labyrinth like structures. Now that we have established that the discrete equations produce

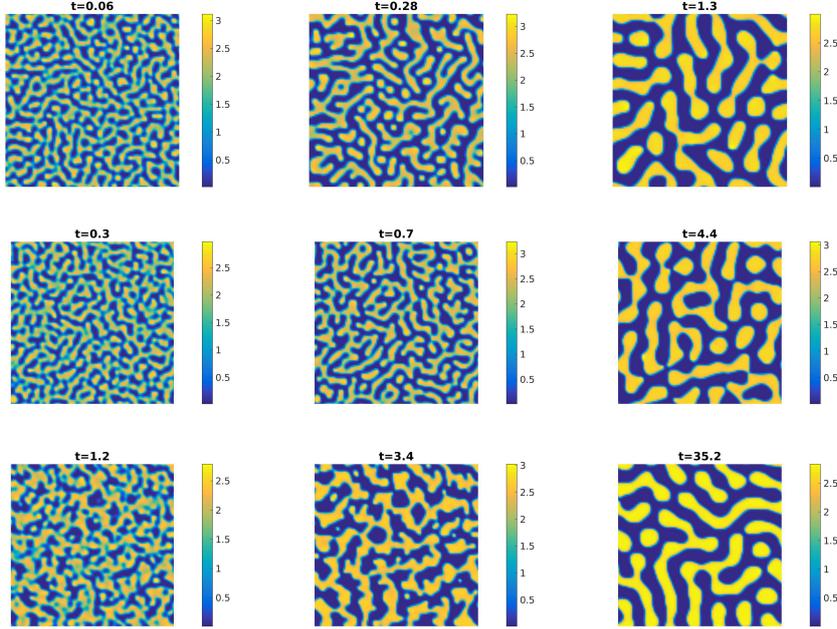


Figure 3.9: Simulation of equation (3.2.5) from a randomly perturbed homogeneous state. The top 3 figures show the evolution for $\mu = 3.5$, the middle 3 figures for $\mu = 3$ and the bottom 3 for $\mu = 2.5$. The other parameters are $D_s = 1.2$, $D_l = 1$, $r_s = 1/14$, $r_l = 1/2$, $L = 4$ and $\rho_0 = 1.2$. The simulations were done on a grid of 100 by 100 points.

patterns of our interest, we can focus on the continuum equations that were derived from these discrete equations.

3.2.4 Simulations of the continuous model in 2-d

In section 2.4.3 we derived the following equation as a continuum approximation for the discrete equation (3.2.4):

$$\begin{aligned}
 \frac{\partial N}{\partial t} &= \frac{\lambda}{zC_{2,2s}} \left(g(A) \Delta^s \left(\frac{N}{g(A)} \right) - \frac{N}{g(A)} \Delta^s (g(A)) \right), \\
 g(A) &= \frac{1}{2} + \frac{1}{2} \tanh(A), \\
 A &= D_s K_s * N - D_l K_l * N, \\
 K_\zeta(x) &= \begin{cases} \frac{1}{\pi r_\zeta^2} & |x| < r_\zeta \\ 0 & |x| > r_\zeta \end{cases} \quad \zeta = s, l.
 \end{aligned} \tag{3.2.5}$$

Here, $K_\zeta * N$ stands for the convolution of kernel K_ζ with N . Note that the exponent μ in the discrete equations corresponds to s in the fractional Laplacian Δ^s via $s = \frac{\mu-d}{2}$ where d is the dimension of the space (2 in this case). With the knowledge of the previous sections, implementing this equation in Matlab is now straightforward. We take the code from section 3.1 and change the $|q|^2$ into

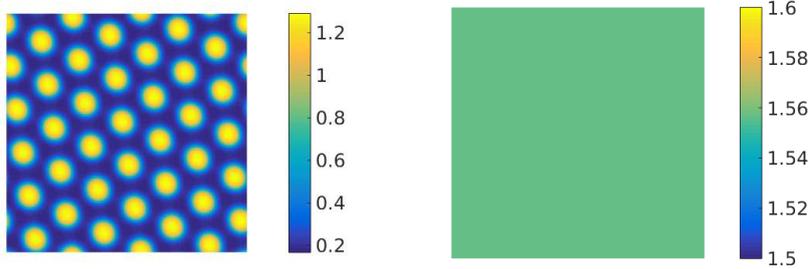


Figure 3.10: Simulation of equation (3.2.5) until convergence. The left figure shows the stationary solution while the figure on the right shows $\rho/g(A)^2$. The parameters are $D_s = 1.2$, $D_l = 1$, $r_s = 1/14$, $r_l = 1/2$, $L = 4$, $\mu = 3.5$ and $\rho_0 = 0.5$.

$|q|^{2s}$. There are multiple questions we would like to answer with this code. First, we would like to see how the Lévy parameter μ influences the solutions. Figure 3.9 shows the evolution of the equation for 3 different values of μ . Note that the 3 most right plots, where the simulations are nearly stationary, are in agreement with the simulation of the discrete model as in figure 3.8. Surprisingly, the shape of these solutions seems independent of the value of μ . In the early stages of the evolution, there is a difference in shape, but this disappears afterwards. We should also notice that the evolution is much slower as μ is lowered. One might think that this is because of the value of the constant $1/(zC'_{2,2s})$. However, the variation in this constant in the range $\mu = 2.5$ to $\mu = 3.5$ is small and can therefore not be responsible for the significant slow down in evolution. Hence, the fractional Laplacian itself is responsible for the slow down. In contrast, De Jager [8] showed that in her Individual Based Model, pattern formation is the fastest when $\mu = 3$, but it is difficult to compare this result with our simulation. The pattern formation rate in [8] is calculated in the following way:

Because we assume that movement speed [of the mussels] is constant, we can calculate the rate of patterning as the normalized inverse of the distance traversed until a pattern is formed.

However, in our continuous equations we cannot follow a single mussel nor can we assume that mussels move with a constant speed.

Another observation we make is that the patterns are very similar to the patterns in section 3.1. This might seem like a coincidence, but the stationary solution of equation (3.2.5) is $\rho = cg(A)^2$ for some $c \in \mathbf{R}_+$. This is exactly the same stationary solution as equation (3.1.6). Furthermore, this stationary solution does not depend on μ .

Now to see whether or not the stationary solutions are indeed the same for the two equations, we run a simulation for equation (3.2.5) with the same parameter values as in figure 3.4. The result is shown in figure 3.10. Not only is the solution identical to the solution in 3.4, but more surprisingly, both solutions converge to the same value for $c = \rho/g(A)^2$. Our simulations have shown that the same holds true for a larger parameter range. Therefore, we conclude

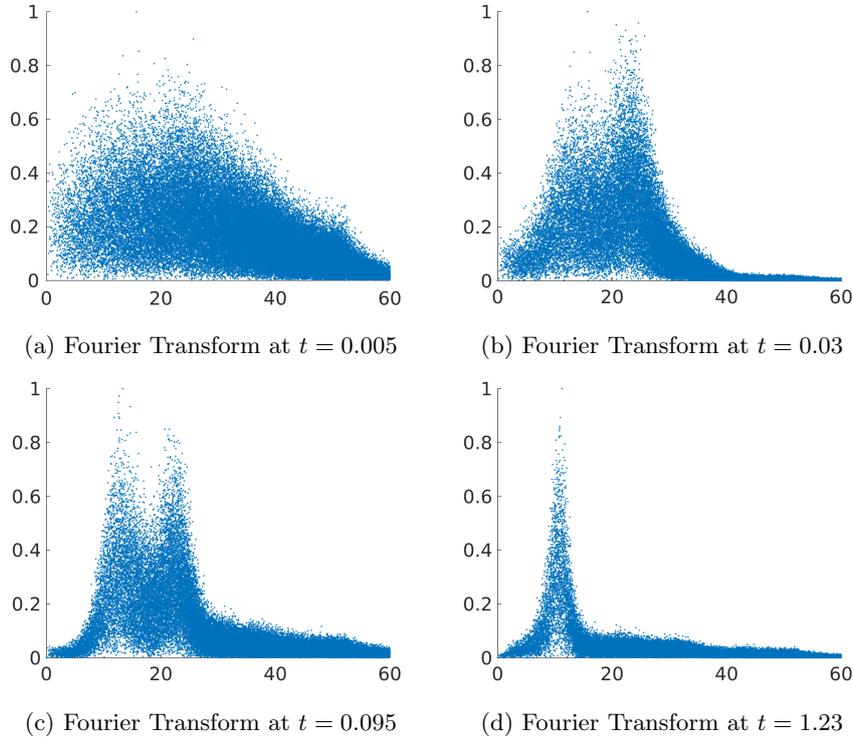


Figure 3.11: Normalized Fourier transforms for a simulation of equation 3.2.5. The parameters are $\rho_0 = 1.2$, $r_s = 1/14$, $r_l = 0.5$, $D_s = 1.2$, $D_l = 1$, $\mu = 3.5$ and $L = 24$.

that both equations converge to the same stationary solution. The meaning of this result is however not clear. Is it just a coincidence or is there a deeper explanation behind it? For the latter one could argue that the stationary solution describes a situation where the short and long scale feedback balance each other, a situation which apparently equally holds for both models. However, the specific shape of both equations depends heavily on the modelling and at this point we are unable to say which assumptions in the modelling are critical to find the specific stationary solution $\rho = g(A)^2$.

3.2.5 Coarsening in Lévy models

As we can see from figure 3.9, the initial formed patterns have a smaller wavelength than the final stages, just as we have seen in the Cahn-Hilliard equation and with the PDE from section 3.1. As we can expect by now, the coarsening behaviour in the equation (3.2.5) is similar to that of equation (3.1.1). To show this, we again plotted the evolution of the Fourier transform, see figure 3.11. When we compare this with figure 3.5, we can see that the evolution of the Fourier transforms are qualitatively the same. In both cases, two peaks emerge and after some time the peak corresponding to the smaller wavelength completely disappears and the peak corresponding to the critical wavelength remains. Again, this behaviour is independent of the value of μ .

3.3 Existence of patterns

In section 3.1 we simulated equation (3.1.1) and found a good comparison with the experimental results. However, in chapter 2 multiple equations were derived similar in shape to equation (3.1.1):

$$\begin{aligned}\frac{\partial \rho}{\partial t} &= \Delta \left(\frac{\rho}{g(A)^2} \right), \\ g(A) &= \frac{1}{2} + \frac{1}{2} \tanh(A).\end{aligned}\tag{3.3.1}$$

In this section we compare equation (3.3.1) with the other equations. As we have shown in section 2.5, in [21] an equation is derived based on the same principles of two scale dependent feedback as we use, but the following equation is derived:

$$\begin{aligned}\frac{\partial \rho}{\partial t} &= \Delta(D(\rho)\rho), \\ D(\rho) &= \frac{1}{2} + \frac{1}{2}(\tanh(2(a - b\rho_s + c\rho_l) - 1)).\end{aligned}\tag{3.3.2}$$

In section 2.2.1 we derived an equation with the following shape:

$$\frac{\partial \rho}{\partial t} = \frac{m_2}{2} \Delta(\lambda(\rho)\rho),\tag{3.3.3}$$

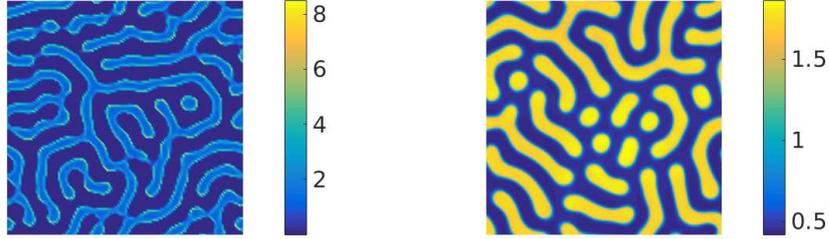
with $\lambda(\rho)$ the jump rate of the mussels and m_2 the second moment of the underlying density *independent* dispersal kernel. In section 2.4.4 we found that this can be generalized to equation (2.4.46).

$$\frac{\partial \rho}{\partial t} = \frac{1}{zC_{2,2s}} \Delta^s(\lambda(\rho)\rho),\tag{3.3.4}$$

in case the second moment of the kernel does not exist. A solution of equation (3.3.2) can be found in figure 3.12a and a solution of equation (3.3.3) for $\lambda(A) = 1 - g(A)$ can be found in figure 3.12c. As we can see, these equations do show pattern formation, but in these patterns, all the mass is concentrated on the edges of the labyrinths or dots and therefore they do not describe the experimental results very well. Hence, we ask the following question: Is there a group of functions $h(A)$ such that the equation

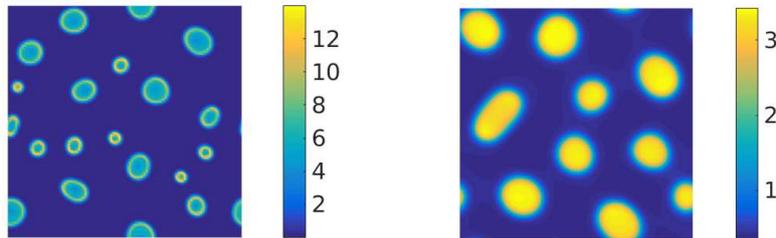
$$\frac{\partial \rho}{\partial t} = \Delta(h(A)\rho),\tag{3.3.5}$$

produces patterns that describe the experimentally found patterns in the mussel beds? To answer this question, we compare equation (3.3.1) with the other equations that do not produce the patterns we are looking for. First, we compare equation (3.3.1) with equation (3.3.2). How do these equations differ, apart from some small differences in the definitions of their D and our $g(A)$? There is a major difference in the range of D and the range of $1/g(A)$. ρ is mapped to $(0, 1)$ by D , while ρ is mapped to $(1, \infty)$ by $1/g(A)^2$. Now what would happen if we replace $D(\rho)$ with $D(\rho) + 1$, i.e. we create an offset such that if a point is very attractive (that is $D \sim 0$) movement is still possible.



(a) Sim. of eq. (3.3.2) with original D .

(b) Sim. of eq. (3.3.2) with $D + 1$.



(c) Sim. of eq. (3.3.3) for $\lambda(A) = 1 - g(A)$.

(d) Sim. of eq. (3.3.3) for $\lambda(A) = 1.1 - g(A)$.

Figure 3.12: Figures (a) and (b) show simulations of equation (3.3.2) for two different D . The parameters are $\rho_0 = 1$, $r_s = 0.05$, $r_l = 0.1$, $a = 1$, $b = 4.4$, $c = 3.9$ and $L = 1$. Figures (c) and (d) show simulations of equation (3.3.3) for two different $\lambda(A)$. The parameters are $\rho_0 = 1$, $r_s = 1/7$, $r_l = 1$, $D_s = 1.2$, $D_l = 1$ and $L = 4$.

As we can see in figure 3.12b, this time we do get patterns that are similar to the ones we have seen before. On the other hand, what if we increase the offset to, say, 5? Then, the diffusion is so strong that the patterns disappear again.

Now how do equations (3.3.3) and (3.3.4) compare with equation (3.3.1) for the most straightforward choice $\lambda(A) = 1 - g(A)$? We again find that the range of $1 - g(A)$ is $(0, 1)$ just as with equation (3.3.2). Hence, we decide to add a small offset and replace $\lambda(A) = 1 - g(A)$ with $\lambda(A) = 1.1 - g(A)$, see figure 3.12d. In this case we also see that introducing an offset produces patterns that are similar to the observed patterns.

Therefore, it is tempting to conclude that there might be a very large group of functions $h(A)$ in the term $\Delta(h(A)\rho)$ that can be used in the modelling, as long as h has a certain offset and is monotonically decreasing in A , such as $1/g(A)^2$, $1.1 - g(A)$ and $D(\rho) + 1$. Proving this might be very difficult, in the first place because we should define mathematically what is meant by “The patterns we are looking for”. One should be able to make a clear distinction between figures 3.12a and 3.12b.

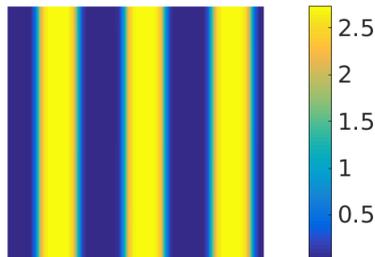


Figure 3.13: Iterative solution of the stationary state of equation (3.1.1) with spectral weighting. The parameters are $r_s = 1/7$, $r_l = 1$, $D_s = 1.2$, $D_l = 1$, $\rho_0 = 1$, $r = 0.15$, $k^* = 4.8$ and $L = 4$.

3.4 Stability of the stationary states

We have seen by now a lot of stationary states. At least, stationary in the sense that in the numerics the norm between two different time steps, $\|\rho_{n+1} - \rho_n\|_2$, has become small, say 10^{-6} , and there is no visual difference between two different time steps. But how stable are these states? As we have learned from the Cahn-Hilliard equation, it might be possible that solutions become stuck in metastable state. First let us consider the hexagonal dotted solutions of equation (3.1.1). Independent of how long we simulate, how many grid points we use, if we add noise or not, both the simulations via the PDE as well as the iterative method converge to this pattern quite fast up to a precision of 10^{-6} . However, in the case of the labyrinth-like structures the convergence is much slower and there is not a clear single state to which the solutions converge. To solve this problem, we use a method called spectral weighting, [6]. The idea is to focus the evolution of the equation at the dominant wavelength and the corresponding higher modes in the following way: we damp the Fourier coefficients $\rho(k)$ with $\rho(k) \mapsto w(k, k^*)\rho(k)$ where k^* is the dominant wavelength of the system and

$$\begin{aligned}
 w(k, k^*) &= (1 - r) + \\
 &r(\exp(-5(1 - |k|/k^*)^2) + \exp(-5(2 - |k|/k^*)^2) + \exp(-5(3 - (|k|/k^*)^2)),
 \end{aligned}
 \tag{3.4.1}$$

for some r between 0.05 and 0.3. Once structures have formed, we can use this weighting to push the solution into its final form. There is no guarantee that this will actually work and we still have to determine k^* . In the iterative approach, we first run the simulation without the spectral weighting, compute the dominant wavelength and then we rerun the simulation with the weighting for the found k^* . In some cases we get results as in figure 3.13.

Intuitively, this certainly looks more like a stable stationary solution than the labyrinth-like structures we found before. However, it is numerically hard to prove that all solutions converge to a state as in figure 3.13. Furthermore, in order to say that this stationary state is “the best” stationary state, we would, just as in [6], need a functional corresponding to our PDE and show that this

stationary state is a minimizer of the functional. Unfortunately, we do not have such a functional. However, independent of the question what the true stationary state is, we can still conclude that the (near) stationary states have a fixed wavelength which is what we were after.

Analysis of the new model

In chapter 2 we derived multiple models for the pattern formation in mussel beds and we saw a good agreement with the experimental data in chapter 3. In this chapter we study one specific equation in greater depth. We chose equation (2.2.20), a continuum approximation to the Individual Based Model by Liu et al.[20], to study due to the fact that it is relatively simple compared with the other equations and gives good results in comparison to experimental data. Furthermore, equation (2.4.44), a continuum approximation to the Individual Based Model by De Jager et al. [8], has the same stationary solution as equation (2.2.20), so we can hope that the study of equation (2.2.20) gives us some insight in the stationary solutions that can be used for the other equation as well.

First, we rescale the equation to reduce the number of constants. Then, we use linear stability analysis to determine when the homogeneous background becomes unstable and patterns such as hexagonal spots and labyrinths can occur. In section 4.3, we study the stability of the patterns analytically using modulation equations, but this approach will not be successful and we give reasons why.

4.1 Rescaling

First, we state equation (2.2.20):

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= \Delta \left(\frac{\rho}{g(A)^2} \right), \\ A &= D_s K_s * \rho - D_l K_l * \rho, \end{aligned} \tag{4.1.1}$$

where $g(A) = \frac{1}{2} + \frac{1}{2} \tanh(A)$, see equation (2.2.3), and the $*$ stands for convolution, see equation (2.4.41).

To get rid of some of the constants, we rescale the equation. We define $\tilde{x} = \frac{x}{r_l}$ and $\tilde{\rho} = D_l \rho$. This implies that $r_l^2 \Delta_x = \Delta_{\tilde{x}}$. Substituting these scalings into equation (4.1.1) results in

$$\begin{aligned} \frac{\partial \tilde{\rho}}{\partial t} &= \frac{1}{r_l^2} \Delta_{\tilde{x}} \left(\frac{\tilde{\rho}}{g(A)^2} \right), \\ A &= \frac{D_s}{D_l} K_{s/l} * \tilde{\rho} - K_1 * \tilde{\rho}. \end{aligned} \tag{4.1.2}$$

When we define $D = \frac{D_s}{D_l}$, $r_\alpha = r_s/r_l$, $t = r_l^2 \tilde{t}$, we get

$$\begin{aligned} \frac{\partial \tilde{\rho}}{\partial \tilde{t}} &= \Delta_{\tilde{x}} \left(\frac{\tilde{\rho}}{g(A)^2} \right), \\ A &= DK_\alpha * \tilde{\rho} - K_1 * \tilde{\rho}. \end{aligned} \tag{4.1.3}$$

Hence, we are left with an interaction length scale r_α and we have D as a measure for the relative strength of the short scale attraction versus long scale dispersion. However, we should not forget that one of the most important parameters is $\tilde{\rho}_0$ which does not appear in this equation, but only in the initial conditions. To keep the notation more clear, we will drop the tildes from now on.

4.2 Linear stability analysis

In this section we investigate the stability of the homogeneous background. As we have seen in the previous chapters, for certain parameter values the homogeneous background becomes unstable and patterns form. There are 3 parameters we can vary, D , r_α and ρ_0 . In the remaining of this chapter fix D and r_α and vary ρ_0 . The main reason for this choice is the following: In [19], the paper that formed the starting point of this thesis, it was shown that existence of patterns and the shape of the patterns depends of the mussel density. For low densities, no patterns form, for intermediate densities clumps of mussels form and for high densities we observe labyrinth structures. In the previous chapter we have seen numerically that equation (4.1.1) has the same behaviour. Therefore, we will study in the rest of this chapter how ρ_0 influences the existence and shape of the patterns.

4.2.1 Stability curves

Before we can study the patterns in the solutions from equation (4.1.3), we need to determine when patterns can form. We need to find out when which wavenumbers become unstable so we need to find the stability curves or dispersion relation. To compute the stability curves, we substitute $\rho = \rho_0 + \varepsilon e^{\lambda t + i\vec{k} \cdot \vec{x}}$ into equation (4.1.3), Taylor around $\varepsilon = 0$ and look for a dispersion relation $\lambda(\vec{k})$. If $\lambda(\vec{k})$ is negative for all \vec{k} , the solution must decay exponentially to the homogeneous state ρ_0 . The stationary state becomes unstable when there are some \vec{k} such that $\lambda(\vec{k}) > 0$ and patterns can form. The Taylor expansion of equation (4.1.3) around $\varepsilon = 0$ results in long formula's, so we introduce some new constants and functions to keep the notation clear:

$$\begin{aligned} A(\rho) &= DK_\alpha * (\rho_0 + \varepsilon e^{\lambda t + i\vec{k} \cdot \vec{x}}) - K_1 * (\rho_0 + \varepsilon e^{\lambda t + i\vec{k} \cdot \vec{x}}) \\ &= D\rho_0 - \rho_0 + \varepsilon(DK_\alpha * e^{\lambda t + i\vec{k} \cdot \vec{x}} - K_1 * e^{\lambda t + i\vec{k} \cdot \vec{x}}) \\ &:= \gamma_0 + \varepsilon\gamma_1, \end{aligned}$$

and therefore

$$g(A) \approx g(\gamma_0) + \varepsilon\gamma_1 g'(\gamma_0) + h.o.t.$$

Next, we need to approximate the $\rho/g(A)^2$ term:

$$\frac{\rho_0 + \varepsilon e^{\lambda t + i\vec{k}\cdot\vec{x}}}{(g(\gamma_0) + \varepsilon \gamma_1 g'(\gamma_0) + h.o.t.)^2} \approx \frac{\rho_0}{g(\gamma_0)^2} + \varepsilon \frac{g(\gamma_0) e^{\lambda t + i\vec{k}\cdot\vec{x}} - 2\gamma_1 g'(\gamma_0) \rho_0}{g(\gamma_0)^3} + h.o.t. \quad (4.2.1)$$

When we substitute this expansion into equation (4.1.3) we find

$$\varepsilon \frac{\partial e^{\lambda t + i\vec{k}\cdot\vec{x}}}{\partial t} = \Delta \left(\frac{\rho_0}{g(\gamma_0)^2} + \varepsilon \frac{g(\gamma_0) e^{\lambda t + i\vec{k}\cdot\vec{x}} - 2\gamma_1 g'(\gamma_0) \rho_0}{g(\gamma_0)^3} + h.o.t. \right) \quad (4.2.2)$$

The $\mathcal{O}(1)$ equation drops out, so the lowest order equation is the $\mathcal{O}(\varepsilon)$ equation:

$$\lambda e^{\lambda t + i\vec{k}\cdot\vec{x}} = \Delta \frac{g(\gamma_0) e^{\lambda t + i\vec{k}\cdot\vec{x}} - 2\gamma_1 g'(\gamma_0) \rho_0}{g(\gamma_0)^3} \quad (4.2.3)$$

$$= -|k|^2 \frac{g(\gamma_0) e^{\lambda t + i\vec{k}\cdot\vec{x}} - 2\gamma_1 g'(\gamma_0) \rho_0}{g(\gamma_0)^3}. \quad (4.2.4)$$

The last equation holds because the convolution and derivative commute. At this point, we run into our first problem. We would like to divide on both sides by $e^{\lambda t + i\vec{k}\cdot\vec{x}}$ to find an equation for λ as function of \vec{k} independent of \vec{x} , but when we divide $\gamma_1 = DK_\alpha * e^{\lambda t + i\vec{k}\cdot\vec{x}} - K_1 * e^{\lambda t + i\vec{k}\cdot\vec{x}}$ by $e^{\lambda t + i\vec{k}\cdot\vec{x}}$ we do not get rid of the exponential. Fortunately, the convolution in γ_1 has much easier representation in Fourier space because the convolution transforms into a product in Fourier space. Therefore, we divide by $e^{\lambda t}$ on both sides and then take the Fourier transform with respect to \vec{x} :

$$\lambda \mathcal{F}(e^{i\vec{k}\cdot\vec{x}}) = -|k|^2 \left(\frac{1}{g(\gamma_0)^2} - \frac{2\rho_0 g'(\gamma_0)}{g(\gamma_0)^3} (D\hat{K}_\alpha \mathcal{F}(e^{i\vec{k}\cdot\vec{x}}) - \hat{K}_1 \mathcal{F}(e^{i\vec{k}\cdot\vec{x}})) \right). \quad (4.2.5)$$

Then, we integrate $\mathcal{F}(e^{i\vec{k}\cdot\vec{x}}) = \delta(\vec{k})$ out which results in the following equation:

$$\lambda = -|k|^2 \left(\frac{1}{g(\gamma_0)^2} - \frac{2\rho_0 g'(\gamma_0)}{g(\gamma_0)^3} (D\hat{K}_\alpha - \hat{K}_1) \right), \quad (4.2.6)$$

with \hat{K}_μ the Fourier transform of the kernel for $\mu = s, l$:

$$\hat{K}_\mu = \frac{\sin(kr_\mu)}{kr_\mu} = \text{sinc}(kr_\mu) \quad \text{in 1 dim.} \quad (4.2.7)$$

$$\hat{K}_\mu = \frac{2J_1(|k|R_\mu)}{|k|r_\mu} \quad \text{in 2 dim.} \quad (4.2.8)$$

J_1 is the first order Bessel function of the first kind. Notice that the dimension of the space influences equation (4.2.6) only via the shape of the Fouriertransformed kernel. Furthermore, λ only depends on \vec{k} via its absolute value $|k|$. This is not surprising because the kernels and the equation are invariant under rotation.

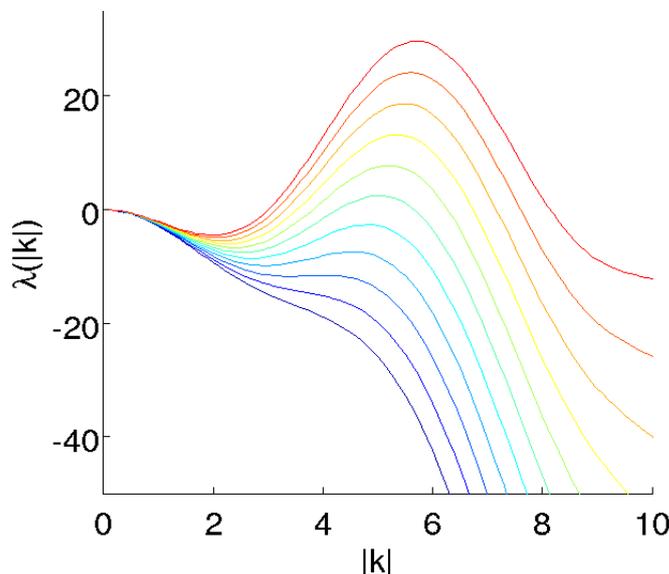


Figure 4.1: Plots of equation (4.2.9) for the 2-d kernel. $D = 1.2$, $r_\alpha = 1/7$ and ρ_0 ranges from 0.3 to 0.6 (from bottom to top). Notice that equation (4.2.9) only depends on \vec{k} via $|k|$ and notice that the curves are negative for small $|k|$.

When we take $g(A) = \frac{1}{2} + \frac{1}{2} \tanh(A)$, we can compute $g(\gamma_0)$ and $g'(\gamma_0)$, so equation (4.2.6) takes the form

$$\lambda = -|k|^2 \left(\frac{1}{\left(\frac{1}{2} + \frac{1}{2} \tanh(\gamma_0)\right)^2} - \frac{\rho_0}{\cosh^2(\gamma_0) \left(\frac{1}{2} + \frac{1}{2} \tanh(\gamma_0)\right)^3} (D\hat{K}_\alpha - \hat{K}_1) \right). \quad (4.2.9)$$

In figure (4.1) we plot equation (4.2.9) for ρ_0 ranging from 0.3 to 0.6. For $\rho_0 = 0.3$, the function decreases monotonically and for $\rho_0 = 0.6$ we see a broad range of unstable waves. Just the visual inspection of these curves raises one important question. The curves seem to be negative for small wavenumbers, is this true and if yes, for which parameter values?

4.2.2 Stability of the small wavenumbers

As we learned in chapter 1 on the Cahn-Hilliard equation, the low wavenumbers are always unstable which allows coarsening to go on. Therefore, we want to investigate the sign of the stability curves near $k = 0$. At $k = 0$ we find that $\lambda(0) = \lambda'(0) = 0$ and $\lambda''(0) = -\frac{2}{g(\gamma_0)^3} (g(\gamma_0) - 2g'(\gamma_0)\rho_0(D-1))$. The first wavenumbers are stable if the second derivative $\lambda''(0)$ is negative, hence when

$$0 > \frac{-2}{g(\gamma_0)^3} (g(\gamma_0) - 2g'(\gamma_0)\rho_0(D-1)). \quad (4.2.10)$$

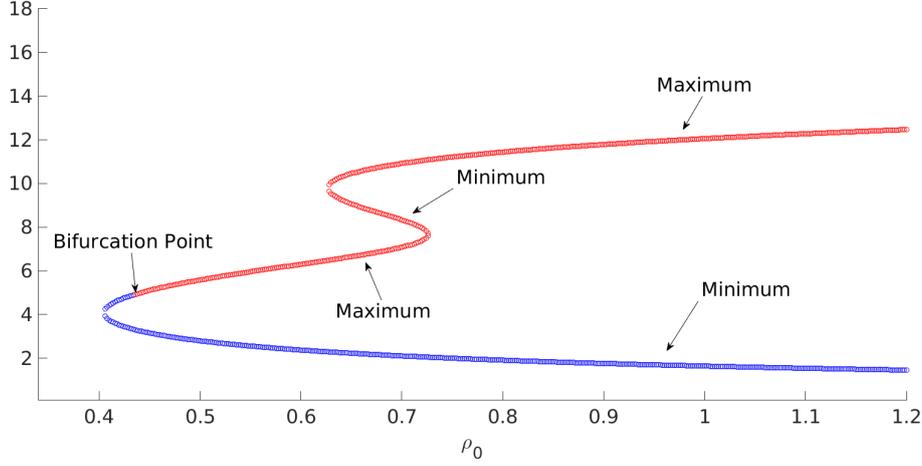


Figure 4.2: This figure shows the location of the extrema of $\lambda(|k|)$ as a function of ρ_0 . D and r_α have a fixed value of $D = 1.2$ and $r_\alpha = 1/7$. Blue denotes a negative extremum, red a positive extremum. The arrows indicate if a branch is a maximum or a minimum. Compare with figure 4.1

By assumption, $g(\gamma_0), g'(\gamma_0) > 0$, so we can rewrite this inequality to

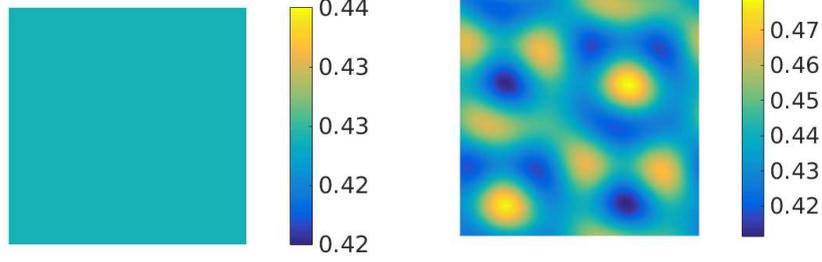
$$D < 1 + \frac{g(\gamma_0)}{2g'(\gamma_0)\rho_0}. \quad (4.2.11)$$

Notice that the rhs of the equation also depends on D via γ_0 but the inequality does not depend on r_α . Numerical calculations show that this implicit inequality always holds for $\rho_0, D \in [0, 10]$. Therefore, the small wavenumbers are stable independent of ρ_0, r_α and D in the parameter regime we are interested in.

4.2.3 Turing bifurcation

As can be seen from figure 4.1, for the given D and r_α , all wavenumbers are stable when ρ_0 is small, but when ρ_0 becomes approximately 0.4 wavenumbers around $|k| = 5$ become unstable. To determine this bifurcation point, we need to find the point (ρ_0^*, k_c) where the stability curve touches the axis from below with fixed r_α and D , i.e. we need to find the pair (ρ_0^*, k_c) that fulfils the following conditions:

$$\begin{aligned} \lambda(\rho_0^*, k_c) &= 0, \\ \frac{d\lambda}{d|k|}(\rho_0^*, k_c) &= 0. \end{aligned} \quad (4.2.12)$$



(a) Simulation for $\rho_0 = 0.43$

(b) Simulation for $\rho_0 = 0.44$.

Figure 4.3: Simulation of equation (4.1.3). The parameters are $r_\alpha = 1/7$, $D = 1.2$ and $L = 4$.

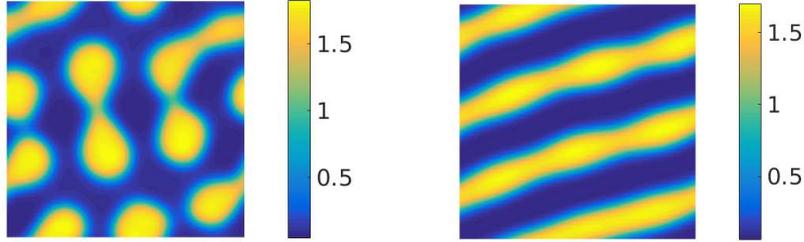
We know $\lambda(|k|)$ from equation (4.2.6), we can compute the derivative (away from $k = 0$) and find:

$$\begin{aligned}
 & g(\gamma_0^*) - 2g'(\gamma_0^*)\rho_0^*(D\hat{K}_\alpha(k_c) - \hat{K}_1(k_c)) = 0, \\
 & -2|k_c| \left(\frac{1}{g(\gamma_0^*)^2} - \frac{2\rho_0^*g'(\gamma_0^*)}{g(\gamma_0^*)^3} (D\hat{K}_\alpha(k_c) - \hat{K}_1(k_c)) \right) - \\
 & |k_c|^2 \left(\frac{4\rho_0^*g'(\gamma_0^*)}{g(\gamma_0^*)^3} \left(D \frac{J_2(r_\alpha|k_c|)}{|k_c|} - \frac{J_2(|k_c|)}{|k_c|} \right) \right) = 0,
 \end{aligned} \tag{4.2.13}$$

with $\gamma_0^* = \rho_0^*(D - 1)$. This set of equations can not be solved analytically, but they can be solved numerically for fixed D, r_α . In figure 4.2 we calculated the positions of the extrema of the stability curves and we can clearly see how the stability curves evolve as function of ρ_0 . For small ρ_0 , the stability curve decreases monotonically to $-\infty$. When ρ_0 increases to approximately 0.4, a local minimum and maximum form. When we increase ρ_0 even more to approximately 0.43, the local maximum becomes positive and we have a Turing bifurcation. Numerical simulations show that indeed for $\rho_0 = 0.43$ a homogeneous solution is found, but for $\rho_0 = 0.44$ hexagonal dots appear, although very slowly and with a small amplitude, see figure 4.3.

As figure 4.4 shows, at $\rho_0 = 0.69$, we still have spots as stationary solution, but at $\rho_0 = 0.70$ we find stripes. However, the stability curves do not give us any information why this happens at this point. There might be other values for D and r_α for which the stability curves do give more information. For example, one could try to find D, r_α and ρ_0 such that $\lambda(|k|)$ has two maxima that become positive at the same time. However, finding the bifurcation points from hexagonal spots to labyrinths is beyond the scope of this thesis.

At this point there remains one question we want to answer: Can we show that the hexagonal pattern that forms after the first bifurcation point is a stable stationary solution and can we find an approximative solution? This question will be treated in the next section.



(a) Simulation for $\rho_0 = 0.69$.

(b) Simulation for $\rho_0 = 0.70$.

Figure 4.4: Simulation of equation 4.1.3. The parameters are $r_\alpha = 1/7$, $D = 1.2$ and $L = 4$.

4.3 Modulation Equations

From the numerical results, we have seen that for $\rho_0 > \rho_0^*$ (but not too large) a numerical stable hexagonal pattern forms. We would like to find an approximation for these patterns and determine their stability analytically. The 2-d case is too difficult to start with and therefore we begin in 1-d. For an introduction to modulation equations, see for example [16].

We want to study the equation in the case $\rho_0 = \rho_0^* + \varepsilon^2 r$ for $0 < \varepsilon \ll 1$ and r of $\mathcal{O}(1)$. Therefore, we propose the following Ansatz for ρ :

$$\rho = \rho_0^* + \varepsilon^2 r + \varepsilon^\alpha A_1(\xi, \tau) e^{ik_c x} + \varepsilon^\beta A_2(\xi, \tau) e^{ik_c x} + \varepsilon^\beta B(\xi, \tau) e^{2ik_c x} + \text{h.o.t} + \text{c.c.} \quad (4.3.1)$$

with $\beta > \alpha$ and we introduced the amplitudes A_1 , A_2 and B which are functions of the slow variables $\xi = \varepsilon x$ and $\tau = \varepsilon^2 t$. H.o.t stands for higher order terms and c.c. stands for complex conjugates. We already introduce the higher order A_2 amplitude because we expect to need it later to solve the $\mathcal{O}(\varepsilon^2)$ equation. The reason for introducing B can be found in the next section.

4.3.1 Simulations for small ε

Before we can apply Ansatz (4.3.1), we should first check the numerics in 1-d for ε small to see if it at all makes sense to look for solutions of period k_c with a slow amplitude $A_1(\xi, \tau)$. We simulate equation (4.1.3) in 1-d for $\rho_0 = \rho_0^* + \varepsilon$. Keep in mind that this ρ_0^* is a different ρ_0^* than computed in the previous section, because in 1-d the kernel has a different Fourier transform, see equation (4.2.7). For $D = 1.2$ and $r_\alpha = 1/7$ the critical density and wavenumber are numerically found to be $(\rho_0^*, k_c) = (0.4063, 4.3413)$. Because we hope to see small perturbations around the stationary solutions, we compute the numerical solution for $\tilde{\rho} = \rho - \rho_0$ with $\rho_0 = \rho_0^* + \varepsilon$. Now, $A(\rho) = D\rho_0 - \rho_0 + A(\tilde{\rho})$ and hence equation (4.1.3) becomes

$$\frac{\partial \tilde{\rho}}{\partial t} = \Delta \left(\frac{\rho_0 + \tilde{\rho}}{g(D\rho_0 - \rho_0 + A(\tilde{\rho}))^2} \right). \quad (4.3.2)$$

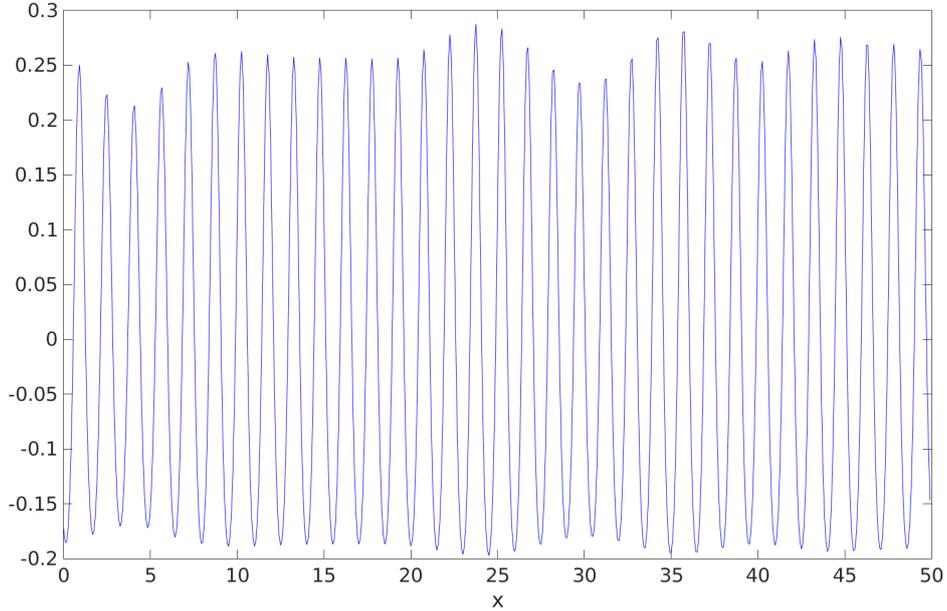


Figure 4.5: Solution of equation (4.3.2) on a large grid, $L = 50$. The other parameters are $\varepsilon = 0.01$, $r_\alpha = 1/7$ and $D = 1.2$.

We want to see the behaviour of the solution on a slow spatial scale and therefore we simulate the equation above on a large grid, see figure 4.5. We see that for ε small, we indeed have fast periodic movement with period near k_c and a slowly varying amplitude. The Fourier transform of the solution in figure 4.5, shown in figure 4.6, shows a large peak around k_c and a small peak near $2k_c$ which justifies the $\varepsilon^\beta B e^{2ik_c x}$ term in Ansatz (4.3.1). Combining the information from the Fourier transform and the solution itself, we can conclude that our Ansatz (4.3.1) is reasonable. However, the numerics do not give any information about the scaling of the variables and the Ansatz.

4.3.2 Slow variables

Before we continue with the modulation equations, we need to find out if the scaling of ξ and τ introduced in Ansatz (4.3.1) is correct. We assumed $\rho_0 = \rho_0^* + \varepsilon^2 r$ and now we introduce $K > 0$ which is of order $\mathcal{O}(1)$ by $k = k_c + \varepsilon^\gamma K$. This means that we assume that there is an interval of width $\mathcal{O}(\varepsilon^\gamma)$ around k_c where the wavenumbers are unstable. Now we substitute the expressions for ρ_0 and k back into the dispersion relation (4.2.6) where we for clarity reasons rewrite γ_0 as $\gamma_0 = (D - 1)(\rho_0^* + \varepsilon^2 r) = \gamma_0^* + \varepsilon^2 \bar{\gamma}_0$ with $\gamma_0^* = \rho_0^*(D - 1)$ and

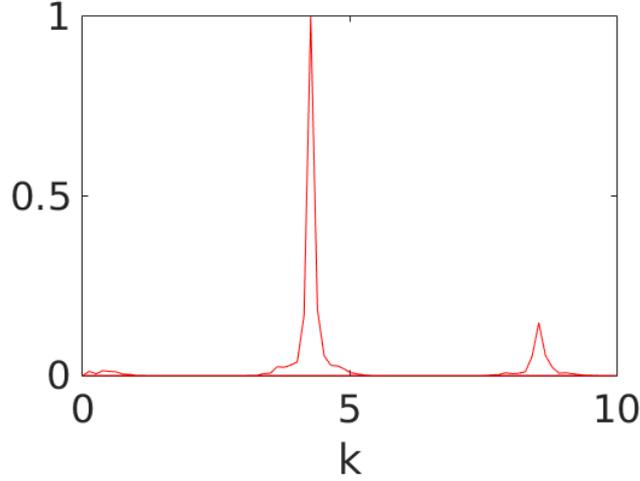


Figure 4.6: Normalized Fourier transform for the solution shown in figure 4.5.

$\bar{\gamma}_0 = r(D - 1)$, and find

$$\lambda = -\frac{|k_c + \varepsilon^\gamma K|^2}{g(\gamma_0^* + \varepsilon^2 \bar{\gamma}_0)^3} \left[g(\gamma_0^* + \varepsilon^2 \bar{\gamma}_0) - 2(\rho_0^* + \varepsilon^2 r)g'(\gamma_0^* + \varepsilon^2 \bar{\gamma}_0) \left(D\hat{K}_\alpha(k_c + \varepsilon^\gamma K) - \hat{K}_1(k_c + \varepsilon^\gamma K) \right) \right]. \quad (4.3.3)$$

When we expand the part within the square brackets we find at $\mathcal{O}(1)$

$$g(\gamma_0^*) + 2\rho_0^*g'(\gamma_0^*)(D\hat{K}_\alpha(k_c) - \hat{K}_1(k_c)) \quad (4.3.4)$$

and at $\mathcal{O}(\varepsilon^\gamma)$

$$-2\rho_0^*g'(\gamma_0^*)(DK\hat{K}'_\alpha(k_c) - K\hat{K}'_1(k_c)). \quad (4.3.5)$$

These two terms form exactly system (4.2.13) at (k_c, ρ_0^*) and are therefore both equal to 0.

When we assume that $\gamma \neq 1$, we find the following $\mathcal{O}(\varepsilon^2)$ term in the square brackets:

$$\bar{\gamma}_0 g'(\gamma_0^*) - 2(\rho_0^* \bar{\gamma}_0 g''(\gamma_0^*) + r g'(\gamma_0^*)) (D\hat{K}_\alpha(k_c) - \hat{K}_1(k_c)). \quad (4.3.6)$$

Therefore, λ is at least of order $\mathcal{O}(\varepsilon^2)$ or $\mathcal{O}(\varepsilon^{2\gamma})$ if $\gamma < 1$. By assumption, we know that λ has two zero's around $k = k_c$ which means that we can choose K such that the lowest order terms in the expansion of λ disappear. If $\gamma < 1$, we would need to eliminate the $\mathcal{O}(\varepsilon^{2\gamma})$ equation and this can only be achieved if $K = 0$ which is against our assumptions, but if $\gamma > 1$, we can only eliminate the $\mathcal{O}(\varepsilon^2)$ equation, which is independent of K , if it is automatically zero by choice of the constants which is not the case. Hence, if we indeed want to find a band of $\mathcal{O}(\varepsilon^\gamma)$ of unstable waves around k_c , we must balance the $\mathcal{O}(\varepsilon^{2\gamma})$ term with the $\mathcal{O}(\varepsilon^2)$ term, i.e $\gamma = 1$, which enables us to eliminate these terms by choosing a suitable nonzero K . Now we know that with k in the unstable band λ is of $\mathcal{O}(\varepsilon^2)$, so we can write $\lambda = \varepsilon^2 \tilde{\lambda}$ and with the choices $\xi = \varepsilon x, \tau = \varepsilon^2 t$ we can conclude that we can approximate $e^{ikx + \lambda t} = e^{ik_c x} e^{i\varepsilon K x + \tilde{\lambda} \varepsilon^2 t} \approx A_1(\xi, \tau) e^{ik_c x}$.

4.3.3 Ginzburg-Landau scaling

In this section we will try to derive modulation equations and, as mentioned, we will not succeed. The reasons why will become clear, but let us first explain the formalism behind modulation equations. After we substituted Ansatz (4.3.1) in to equation (4.1.3), we expand the equation around $\varepsilon = 0$ and order the expansion in ‘levels’ of $\varepsilon^i E^j$. As we will see later, the first nonzero term on the left hand side of equation (4.1.3) will be $\varepsilon^{2+\alpha} \partial_\tau A_1 E$. If our Ansatz is correct, we can solve the levels εE , $\varepsilon^2 E$ and $\varepsilon^2 E^2$ in order to find a closed equation (PDE) for A_1 at level $\varepsilon^{2+\alpha} E$. This procedure depends heavily on the choices of α, β and the chosen Ansatz. In the following computation we choose $2\alpha = \beta = 2$ for simplicity. The conclusions will also hold for other choices of α and β . Now let us substitute Ansatz (4.3.1) into equation (4.1.3). For clarity, we will write $E = e^{ik_c x}$. Because we are interested in the behaviour of the solution at two different length scales, we substitute $\partial_x \mapsto \partial_x + \varepsilon \partial_\xi$, so $\partial_{xx} \mapsto \partial_{xx} + 2\varepsilon \partial_x \partial_\xi + \varepsilon^2 \partial_{\xi\xi}$. Next, we need to expand $1/g(\rho)^2$. To prevent things from getting too cluttered, we omit the $\varepsilon^2 B(\xi, \tau) E^2$ term here. However, up to $\mathcal{O}(\varepsilon^2)$ the coefficients for the $\varepsilon^2 A_2(\xi, \tau) E$ term will be the same.

$$\begin{aligned} \frac{1}{g(A(\rho_0^* + \varepsilon^2 r + \varepsilon A_1 E + \varepsilon^2 A_2 E + \text{h.o.t.} + \text{c.c.}))^2} &= \frac{1}{g(\gamma_0^*)^2} - 2\varepsilon \frac{A(A_1 E) g'(\gamma_0^*)}{g(\gamma_0^*)^3} \\ + \varepsilon^2 A(A_1 E)^2 \frac{3g'(\gamma_0^*)^2 - g(\gamma_0^*) g''(\gamma_0^*)}{g(\gamma_0^*)^4} + \varepsilon^2 \frac{-2(A(A_2 E) + \bar{\gamma}_0) g'(\gamma_0^*)}{g(\gamma_0^*)^3} + \text{h.o.t.} + \text{c.c.} \\ &+ \varepsilon^2 2A(A_1 E) A(\bar{A}_1 \bar{E}) \frac{3g'(\gamma_0^*)^2 - g(\gamma_0^*) g''(\gamma_0^*)}{g(\gamma_0^*)^4} \end{aligned} \quad (4.3.7)$$

Therefore, we end up with the equation

$$\begin{aligned} \partial_\tau (\varepsilon^3 A_1 E + \varepsilon^4 A_2 E + \text{h.o.t.} + \text{c.c.}) &= \\ (\partial_{xx} + 2\varepsilon \partial_x \partial_\xi + \varepsilon^2 \partial_{\xi\xi}) (\rho_0^* + \varepsilon^2 r + \varepsilon A_1(\xi, \tau) E + \varepsilon^2 A_2(\xi, \tau) E + \text{h.o.t.} + \text{c.c.}) &= \\ \left(\frac{1}{g(\gamma_0^*)^2} - 2\varepsilon \frac{A(A_1 E) g'(\gamma_0^*)}{g(\gamma_0^*)^3} + \varepsilon^2 A(A_1 E)^2 \frac{3g'(\gamma_0^*)^2 - g(\gamma_0^*) g''(\gamma_0^*)}{g(\gamma_0^*)^4} + \right. & \\ \left. \varepsilon^2 \frac{-2(A(A_2 E) + \bar{\gamma}_0) g'(\gamma_0^*)}{g(\gamma_0^*)^3} + \text{h.o.t.} + \text{c.c.} + \varepsilon^2 2A(A_1 E) A(\bar{A}_1 \bar{E}) \frac{3g'(\gamma_0^*)^2 - g(\gamma_0^*) g''(\gamma_0^*)}{g(\gamma_0^*)^4} \right) &= \end{aligned} \quad (4.3.8)$$

Now the $\mathcal{O}(1)$ drops out because it only contains derivatives of constants. Next, we find at the $\mathcal{O}(\varepsilon)$ level

$$0 = \partial_{xx} \left(\frac{-2\rho_0^* A(A_1 E) g'(\gamma_0^*)}{g(\gamma_0^*)^3} + \frac{A_1 E}{g(\gamma_0^*)^2} \right), \quad (4.3.9)$$

which we can transform into Fourier space as

$$0 = k^2 \left(\frac{-2\rho_0^* g'(\gamma_0^*) \mathcal{F}(A(A_1 E))}{g(\gamma_0^*)^3} + \frac{\mathcal{F}(A_1 E)}{g(\gamma_0^*)^2} \right). \quad (4.3.10)$$

Now we need to determine what to do with the $\mathcal{F}(A(A_1 E))$ and the $\mathcal{F}(A_1 E)$ term. Because we take the Fourier transform with respect to x and not ξ , we

assume A_1 to be constant for the Fourier transform. This means that we can take $\mathcal{F}(A_1 E) = A_1 \mathcal{F}(E)$ and $\mathcal{F}(A(A_1 E)) = A_1 \hat{A}(k) \mathcal{F}(E)$ with \hat{A} the Fourier transform of A . Of course, after we have found A_1 , we need to check if this is indeed a valid assumption. Now we can integrate $\mathcal{F}(E)$ out and find

$$0 = A_1 k_c^2 \left(\frac{-2\rho_0^* g'(\gamma_0^*) \hat{A}(k_c)}{g(\gamma_0^*)^3} + \frac{1}{g(\gamma_0^*)^2} \right). \quad (4.3.11)$$

This is exactly equation (4.2.12) and therefore holds for all A_1 which at this point is a justification of our Ansatz.

At the $\mathcal{O}(\varepsilon^2)$ level we find

$$\begin{aligned} 0 = & \partial_{xx} \rho_0^* \left(A(A_1 E)^2 \frac{3g'(\gamma_0^*)^2 - g(\gamma_0^*)g''(\gamma_0^*)}{g(\gamma_0^*)^4} + 2A(A_1 E)A(\bar{A}_1 \bar{E}) \frac{3g'(\gamma_0^*)^2 - g(\gamma_0^*)g''(\gamma_0^*)}{g(\gamma_0^*)^4} \right) + \\ & \partial_{xx} \left(A_1 E \frac{-2A(A_1 E)g'(\gamma_0^*)}{g(\gamma_0^*)^3} \right) + \\ & \partial_{xx} \left(\frac{A_2 E}{g(\gamma_0^*)^2} - \frac{2\rho_0^*(A(A_2 E) + \bar{\gamma}_0)g'(\gamma_0^*)}{g(\gamma_0^*)^3} \right) + \\ & 2\partial_{x\xi} \left(\frac{A_1 E}{g(\gamma_0^*)^2} - \frac{2\rho_0^*g'(\gamma_0^*)A(A_1 E)}{g(\gamma_0^*)^3} \right). \end{aligned} \quad (4.3.12)$$

By the same arguing as for the $\mathcal{O}(\varepsilon)$ equation, we see that the 3rd and 4th line are zero by equation (4.2.12). At this point our scheme of ordering the expansion in levels $\varepsilon^i E^j$ breaks down. If we would identify lines 3 and 4 above as the $\varepsilon^2 E$ level, we expect these lines to disappear. However, how do we order the first two lines? Can or should we identify $A(A_1 E)A(\bar{A}_1 \bar{E})$ as an $\varepsilon^2 E^0$ term because $E\bar{E} = E^0$? And what about $A(A_1 E)^2$ and $A_1 E A(A_1 E)$, are these $\varepsilon^2 E^2$ terms?

If we identify $A(A_1 E)A(\bar{A}_1 \bar{E})$ as an $\varepsilon^2 E^0$ term, we can solve the $\varepsilon^2 E^0$ equation by setting A_1 to zero, which is against our assumption, or by introducing a zeroth mode term $\varepsilon^2 A_0$ to our Ansatz. When we introduce a zeroth order mode A_0 , we must still be able to solve the $\varepsilon^2 E^0$ equation. However, it will be impossible to find an exact relation between A_0 and A_1 because of the convolution terms. Transforming the equations to Fourier space is also useless at this point because there is no practical way to compute the transform of the nonlinearity $A(A_1 E)A(\bar{A}_1 \bar{E})$.

We find the same problems at the $\varepsilon^2 E^2$ level. When we identify $A(A_1 E)^2$ and $A_1 E A(A_1 E)$ as order $\varepsilon^2 E^2$ terms and remember that we omitted the $B E^2$ terms in the equation above, we still can not solve for B in terms of A_1 in real space because of the convolution and not in Fourier space because there is no practical way to compute $\mathcal{F}(A(A_1 E)^2)$. The question is now whether or not these problems are a consequence of our Ansatz. Maybe another scaling or Ansatz may solve these problems?

Matthews-Cox scaling

When we interpret the $A(A_1 E)A(\bar{A}_1 \bar{E})$ as a level $\varepsilon^2 E^0$ term, we see that we should have introduced an extra $\varepsilon^2 A_0$ amplitude, which can be explained by the

fact that $k = 0$ is just as k_c marginally unstable, i.e. $\lambda(0) = 0$. This leads to the thought that we must use a completely different Ansatz. Therefore, let us introduce the following Ansatz with the scaling as is done by Matthews & Cox [22]:

$$\rho = \rho_0^* + \varepsilon^2 r + \varepsilon^2 f(\xi, \tau) + \varepsilon^{1.5} B(\xi, \tau) E, \quad (4.3.13)$$

where $f(\xi, \tau)$ is the zeroth order mode. For a more extensive discussion of this scaling, see [27]. Now when we substitute this Ansatz into equation 4.1.3, we find the following lengthy expression:

$$\begin{aligned} & \partial_\tau(\varepsilon^{3.5} BE + \varepsilon^4 f + \text{h.o.t.} + \text{c.c.}) = \\ & (\partial_{xx} + 2\varepsilon \partial_x \partial_\xi + \varepsilon^2 \partial_{\xi\xi})(\rho_0^* + \varepsilon^2 r + \varepsilon^{1.5} B(\xi, \tau) E + \varepsilon^2 f(\xi, \tau) + \text{h.o.t.} + \text{c.c.}) \\ & \left(\frac{1}{g(\gamma_0^*)^2} - 2\varepsilon^{1.5} \frac{A(BE)g'(\gamma_0^*)}{g(\gamma_0^*)^3} + \varepsilon^2 \frac{-2(A(f) + \bar{\gamma}_0)g'(\gamma_0^*)}{g(\gamma_0^*)^3} \right) \\ & + \varepsilon^3 (A(BE) + A(\overline{BE}))^2 \frac{3g'(\gamma_0^*)^2 - g(\gamma_0^*)g''(\gamma_0^*)}{g(\gamma_0^*)^4} \\ & + \varepsilon^4 (\bar{\gamma}_0 + A(f))^2 \frac{3g'(\gamma_0^*)^2 - g(\gamma_0^*)g''(\gamma_0^*)}{g(\gamma_0^*)^4} + \text{c.c.} + \text{h.o.t.} \Big). \end{aligned} \quad (4.3.14)$$

In the $\mathcal{O}(1)$ equation the constants drop out just as in the $\mathcal{O}(\varepsilon)$ equation. The $\mathcal{O}(\varepsilon^{1.5})$ drops out because it fulfils equation (4.2.12) with the same reasoning we used for equation (4.3.9). The $\mathcal{O}(\varepsilon^2)$ drops out because $\partial_x f(\xi, \tau) = 0$. The $\mathcal{O}(\varepsilon^{2.5})$ again drops out because of equation (4.2.12). At $\mathcal{O}(\varepsilon^3)$ we find

$$\begin{aligned} 0 = & \partial_{xx} \left(\rho_0^* (A(BE) + A(\overline{BE}))^2 \frac{3g'(\gamma_0^*)^2 - g(\gamma_0^*)g''(\gamma_0^*)}{g(\gamma_0^*)^4} - 2BE \frac{A(BE)g'(\gamma_0^*)}{g(\gamma_0^*)^3} \right) \\ & + 2\partial_{\xi x} \left(\frac{f(\xi, \tau)}{g(\gamma_0^*)^2} + \frac{-2(A(f) + \bar{\gamma}_0)g'(\gamma_0^*)}{g(\gamma_0^*)^3} \right). \end{aligned} \quad (4.3.15)$$

Now the second line drops out again, but the first gives us the same problems we had in the previous section. We cannot classify the terms into levels of order $\varepsilon^i E^j$.

Hence, we come to the conclusion that due to the presence of the operator $A(\rho)$, we cannot organize the expansions into levels of order $\varepsilon^i E^j$, which lies at the basis of the whole modulation equations formalism. Therefore, we should resort other methods such as a center manifold reduction, but it is likely that the convolution operator also causes trouble here.

Conclusion & Outlook

In this thesis we studied several models to explain pattern formation in mussel beds. We started in chapter 1 with the Cahn-Hilliard equation. The Cahn-Hilliard equation, as derived in [19], mimicked the coarsening behaviour of the mussels in the early stages of pattern formation very well, but failed to select a specific dominant wavelength, while the mussels do select a specific wavelength. We concluded that this discrepancy was caused by a misinterpretation of the density dependent movement speed in the mussels. The biological experiments showed that the movement speed of a single mussel depends on the number of mussels in its neighbourhood, while in the derivation of the Cahn-Hilliard equation it was assumed that the movement speed depended on the density at the exact position of the mussel. The Individual Based Models (IBM), as derived in [8, 20] and discussed in section 2.1, take into account that the movement speed of a mussel depends on the amount of mussels in a larger neighbourhood and they can accurately simulate pattern formation. Therefore, we decided to set the Cahn-Hilliard equation aside and made a continuum approximation of the IBM as derived in [20], which resulted in equation (2.2.20):

$$\begin{aligned}\frac{\partial \rho}{\partial t} &= \frac{3}{2} \Delta \left(\frac{\rho}{g(A)^2} \right), \\ g(A) &= \frac{1}{2} + \frac{1}{2} \tanh(A), \\ A &= D_s K_s * \rho - D_l K_l * \rho,\end{aligned}$$

where $K_s * \rho$ and $K_l * \rho$ stand for the average density at a short respectively large length scale. This equation turned out to be successful in simulating patterns that look like the patterns observed in the experiments and turned out to show coarsening.

In the IBM derived in [8], a main feature is the incorporation of a Lévy walk. The main problem with the Lévy walk is the fact that the second moment does not exist which makes the derivation of a continuous equation much more involved. Following [5], we derived equation (2.4.44):

$$\begin{aligned}\frac{\partial \rho}{\partial t} &= \frac{\lambda}{z C_{2,2s}} \left(g(A) \Delta^s \left(\frac{\rho}{g(A)} \right) - \frac{\rho}{g(A)} \Delta^s (g(A)) \right), \\ g(A) &= \frac{1}{2} + \frac{1}{2} \tanh(A), \\ A &= D_s K_s * \rho - D_l K_l * \rho,\end{aligned}$$

where Δ^s is the fractional Laplacian.



(a) Fairy circles in the Namibian desert. (b) Circular and labyrinth-like structures in the great barrier reef.

Figure 4.7: Other possibilities for two scale dependent feedback?

There are a lot of similarities between the solutions of the two equations as is shown in chapter 3. This second equation with the fractional Laplacian produces structures that look identically to the patterns in the first equation and it shows coarsening as well. Most surprisingly however, is the fact that both equations have the same implicit relation for the stationary solution:

$$\rho = cg(A)^2,$$

for some unknown constant c . Numerical simulations showed that both equations evolve to solutions of the implicit equation above, even for the same constant c . The most important conclusion here is that these stationary solutions have a fixed wavelength that depend mainly on the large interaction scale r_l . Therefore, we can conclude that we have found an answer to the main question we started with: “Can we find a way to explain the wavelength selection in mussel beds?”. Yes, we can with the principle of two scale dependent feedback. However, there is one caveat. The only proof we have is numerically.

In chapter 4 we studied the first equation above analytically. We found a Turing bifurcation for the initial density ρ_0 . At the bifurcation point, the spatially homogeneous background becomes unstable and a regular pattern of hexagonal spots formed. However, studying this bifurcation using modulation equations to determine the stability and character of the stationary solutions was not successful. Already in the one dimensional case we got stuck due to the fact that the standard approach to modulation equations does not work for the convolution terms used to define A .

Outlook

There are three main questions that arise from this thesis. The first question is how important the specific shape of equation (2.2.20) is, which is discussed in section 3.3. There are two key ingredients here: we used a specific choice for g and our nonlinearity is $1/g(A)^2$. We would like to figure out if there are any conditions on g and the nonlinearity for pattern formation.

The second question is how to prove the stability of the stationary solutions. An approach that can be used is a center manifold reduction, for example as

is done in [6]. They are able to show the stability of two different types of stationary solutions in a Cahn-Hilliard equation with long-range interaction potential and they can show how the initial density ρ_0 influences the choice for one of these two solutions. However, there is no guarantee this approach will work for equation (2.2.20) for the same reasons the approach via the modulation equation did not work.

The last question is about applicability to other patterns found in nature. For example, the fairy circles found in the Namibian desert (see figure 4.7) show a quite good resemblance with the solutions in [21]. The plants can work together because ground with a lot of roots holds water better in the arid desert, but these patches of grass compete with other patches at long distances, [31, 10].

There might be many other situations in which two scale dependent feedback is the dominant drive for self organisation. However, we must keep in mind we always need a biological justification why an organism would cooperate at a short length scale but compete at large length scales. Therefore, cooperation between biologists, ecologists and mathematicians is crucial, especially when you realise that the Waddensea where the mussels live, the Namibian desert and the great barrier reef, are all very fragile ecosystems that need our better understanding in order to protect them.

Acknowledgements

First of all, I would like to thank my supervisor Vivi Rottschäfer for her guidance, or rather, for her letting me take a completely different track than we planned ahead. I'd like to thank Robbin Bastiaansen for showing in all details necessary that the Cahn-Hilliard equation would not solve my problems. Many thanks to Johan van de Koppel for receiving us at the NIOZ in Yerseke and showing us around the laboratories. However, I should be most thankful for his reminder that ecology, in contrast to mathematics, does not deal with absolute truths, it tries to find mechanisms that together form our amazingly complex nature. Therefore, focussing too much on single details of an equation is, at least from the modelling point of view, not the smartest thing to do. Last but not least, I would like to thank Christian Ludwig from the TU München for learning me enough about the numerics of PDE's to give me the freedom to write my own codes and use these to test all the idea's that popped into my head the last year.

Bibliography

- [1] R. Bastiaansen. Pattern formation in animal populations. Master's thesis, Universiteit Leiden, the Netherlands, 2015.
- [2] J. Borwein. *Pi and the AGM : a study in analytic number theory and computational complexity*. Wiley, New York, 1987.
- [3] J. W. Cahn and J. E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of chemical physics*, 28(2):258–267, 1958.
- [4] J. Carr, M. E. Gurtin, and M. Slemrod. Structured phase transitions on a finite interval. *Archive for rational mechanics and analysis*, 86(4):317–351, 1984.
- [5] S. Chaturapruek, J. Breslau, D. Yazdi, T. Kolokolnikov, and S. G. McCalla. Crime modeling with Lévy flights. *SIAM Journal on Applied Mathematics*, 73(4):1703–1720, 2013.
- [6] R. Choksi, M. Maras, and J. Williams. 2d phase diagram for minimizers of a Cahn-Hilliard functional with long-range interactions. *SIAM Journal on Applied Dynamical Systems*, 10(4):1344–1362, 2011.
- [7] M. de Jager, F. Bartumeus, A. Kölzsch, F. J. Weissing, G. M. Hengeveld, B. A. Nolet, P. M. Herman, and J. van de Koppel. How superdiffusion gets arrested: ecological encounters explain shift from Lévy to Brownian movement. *Proceedings of the Royal Society of London B: Biological Sciences*, 281(1774):20132605, 2014.
- [8] M. de Jager, F. J. Weissing, P. M. Herman, B. A. Nolet, and J. van de Koppel. Lévy walks evolve through interaction between movement and environmental complexity. *Science*, 332(6037):1551–1553, 2011.
- [9] D. S. Dean. Langevin equation for the density of a system of interacting Langevin processes. *Journal of Physics A: Mathematical and General*, 29(24):L613, 1996.
- [10] C. Fernandez-Oto, M. Tlidi, D. Escaff, and M. Clerc. Strong interaction between plants induces circular barren patches: fairy circles. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 372(2027):20140009, 2014.

- [11] B. V. Gnedenko and A. N. Kolmogorov. *Limit distributions for sums of independent random variables*. Addison-Wesley Mathematics Series. Addison-Wesley, Cambridge, MA, 1954. Translated and annotated by K. L. Chung. With an Appendix by J. L. Doob. MR:0062975. Zbl:0056.36001.
- [12] R. Gorenflo, G. De Fabritiis, and F. Mainardi. Discrete random walk models for symmetric Lévy-Feller diffusion processes. *Physica A: Statistical Mechanics and its Applications*, 269(1):79–89, 1999.
- [13] P. Grindrod. *Patterns and Waves, The theory and applications of reaction-diffusion equations*. Oxford University Press, Oxford New York, 2007.
- [14] J. Guckenheimer and P. Holmes. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42. Springer Science & Business Media, 1983.
- [15] K. Hawick. Numerical simulation and role of noise in the Cahn-Hilliard-Cook equation below the critical dimension. In *Proceedings of the IASTED International Conference on Modeling, Identification and Control (MIC)*, pages 24–26, 2010.
- [16] R. B. Hoyle. *Pattern formation: an introduction to methods*. Cambridge University Press, 2006.
- [17] S. Kondo and T. Miura. Reaction-diffusion model as a framework for understanding biological pattern formation. *Science*, 329(5999):1616–1620, 2010.
- [18] I. M. Lifshitz and V. V. Slyozov. The kinetics of precipitation from supersaturated solid solutions. *Journal of Physics and Chemistry of Solids*, 19(1):35–50, 1961.
- [19] Q.-X. Liu, A. Doelman, V. Rottschäfer, M. de Jager, P. M. Herman, M. Rietkerk, and J. van de Koppel. Phase separation explains a new class of self-organized spatial patterns in ecological systems. *Proceedings of the National Academy of Sciences*, 110(29):11905–11910, 2013.
- [20] Q.-X. Liu, P. M. Herman, W. M. Mooij, J. Huisman, M. Scheffer, H. Olf, and J. van de Koppel. Pattern formation at multiple spatial scales drives the resilience of mussel bed ecosystems. *Nature communications*, 5, 2014.
- [21] R. Martínez-García, C. Murgui, E. Hernández-García, and C. López. Pattern formation in populations with density-dependent movement and two interaction scales. *PLoS ONE*, 10(7):e0132261, 07 2015.
- [22] P. Matthews and S. Cox. One-dimensional pattern formation with Galilean invariance near a stationary bifurcation. *Physical Review E*, 62(2):R1473, 2000.
- [23] H. Meinhardt. *The Algorithmic Beauty of Sea Shells, Fourth Edition*. Springer-Verlag New York, Inc., New York, NY, USA, 2009.
- [24] V. Méndez, D. Campos, and F. Bartumeus. *Stochastic foundations in movement ecology: anomalous diffusion, front propagation and random searches*. Springer Science & Business Media, 2013.

- [25] V. Méndez, D. Campos, I. Pagonabarraga, and S. Fedotov. Density-dependent dispersal and population aggregation patterns. *Journal of theoretical biology*, 309:113–120, 2012.
- [26] K. Oldham. *The fractional calculus : theory and applications of differentiation and integration to arbitrary order*. Academic Press, New York, 1974.
- [27] K. F. Poon. *Dynamics of the Nikolaevskiy and related equations*. PhD thesis, Dept. of Mathematics-Simon Fraser University, 2009.
- [28] M. Rietkerk, S. C. Dekker, P. C. de Ruiter, and J. van de Koppel. Self-organized patchiness and catastrophic shifts in ecosystems. *Science*, 305(5692):1926–1929, 2004.
- [29] M. Rietkerk and J. Van de Koppel. Regular pattern formation in real ecosystems. *Trends in ecology & evolution*, 23(3):169–175, 2008.
- [30] M. J. Schnitzer. Theory of continuum random walks and application to chemotaxis. *Physical Review E*, 48(4):2553, 1993.
- [31] E. Sheffer, H. Yizhaq, M. Shachak, and E. Meron. Mechanisms of vegetation-ring formation in water-limited systems. *Journal of theoretical biology*, 273(1):138–146, 2011.
- [32] E. M. Stein. *Singular integrals and differentiability properties of functions*, volume 2. Princeton university press, 1970.
- [33] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 237(641):37–72, 1952.
- [34] J. Van de Koppel, J. C. Gascoigne, G. Theraulaz, M. Rietkerk, W. M. Mooij, and P. M. Herman. Experimental evidence for spatial self-organization and its emergent effects in mussel bed ecosystems. *Science*, 322(5902):739–742, 2008.
- [35] B. P. Vollmayr-Lee and A. D. Rutenberg. Fast and accurate coarsening simulation with an unconditionally stable time step. *Physical Review E*, 68(6):066703, 2003.

Eyre's Unconditionally Stable Scheme

In this appendix we will explain Eyre's unconditionally stable scheme and a specific approach to solve the scheme. A readable introduction for those without much experience in numerical computation can be found in [35], proof of the stability can be found in the appendices therein.

We start with the original Cahn-Hilliard equation in parameterless form:

$$\phi_t = -\Delta(\phi - \phi^3 + \Delta\phi). \quad (\text{A.1})$$

This equation can be discretized in time as

$$\phi_{t+dt} + dt\Delta[(1 - a_1)\phi_{t+dt} + (1 - a_2)\Delta\phi_{t+dt}] = \phi_t + dt\Delta(-a_1\phi_t - a_2\Delta\phi_t + \phi_t^3). \quad (\text{A.2})$$

In [35] it is shown that this scheme is unconditionally Von Neumann stable when $a_2 \leq 0$ and $a_1 > 2$. Therefore, we propose to take $a_2 = 0$. When we introduce $a = a_1 - 1$, equation (A.2) becomes

$$\phi_{t+dt} + dt\Delta[-a\phi_{t+dt} + \Delta\phi_{t+dt}] = \phi_t + dt\Delta(-(a + 1)\phi_t + \phi_t^3). \quad (\text{A.3})$$

The equation above is discrete in time, but not in space. There are many ways to make the equation above discrete in space, but the fact that the equation is linear in ϕ_{t+dt} gives us the opportunity to use a spectral method. Eyre himself wrote a code using spectral methods which can be found on his homepage¹. In the following a short explanation of the code is given.

On a grid with N by N points and grid size h , we know that the eigenvalues of the discrete Laplacian with Neumann boundary conditions are given by

$$\lambda_{mn} = \frac{2}{h^2} \left(\cos\left(\frac{\pi m}{N}\right) - 1 \right) + \frac{2}{h^2} \left(\cos\left(\frac{\pi n}{N}\right) - 1 \right), \quad 0 \leq m, n < N. \quad (\text{A.4})$$

These eigenvalues correspond to the discrete eigenfunctions

$$\begin{aligned} \phi_{mn}(i, j) &= \alpha_m \alpha_n \cos\left(\frac{\pi m(i - 1/2)}{N}\right) \cos\left(\frac{\pi n(j - 1/2)}{N}\right), \\ \alpha_m &= \begin{cases} \sqrt{\frac{1}{N}} & \text{if } m = 0 \\ \sqrt{\frac{2}{N}} & \text{otherwise} \end{cases} \end{aligned} \quad (\text{A.5})$$

¹<http://www.math.utah.edu/~eyre/computing/matlab-intro/ch.txt>

Fortunately, these discrete eigenfunctions ϕ_{mn} are exactly the same functions Matlab uses to compute the 2-dimensional discrete cosine transform `dct2`. Therefore, the following identity holds in Matlab (keep in mind that ϕ_{t+dt} is an $N \times N$ matrix):

$$\mathbf{dct2}(\phi_{t+dt} + dt\Delta[-a\phi_{t+dt} + \Delta\phi_{t+dt}]) = (1 -adt\lambda + dt\lambda^2)\mathbf{dct2}(\phi_{t+dt}), \quad (\text{A.6})$$

where λ is the matrix with eigenvalues λ_{mn} . Therefore, we can now solve equation A.3 for ϕ_{t+dt} :

$$\mathbf{dct2}(\phi_{t+dt}) = \frac{\mathbf{dct2}(\phi_t) + dt\lambda\mathbf{dct2}(\phi_t^3 - (a+1)\phi_t)}{1 -adt\lambda + dt\lambda^2}. \quad (\text{A.7})$$

Note that all multiplications and divisions are pointwise. Hence, we can find ϕ_{t+dt} from ϕ_t by computing `dct2`(ϕ_{t+dt}) from the equation above and then apply the inverse transform `idtc2`.

Code for equation (2.2.20)

The code below is specifically for equation (2.2.20), but can easily be adapted to the other continuous equations.

```

1  L=4;                %length of the grid
2  N=100;             %number of gridpoints
3  gridsize =L/N;
4  long=1;           %length of the long interaction
5  short=1/7;       %length of the short interaction
6  rho=1;           %initial density
7  Ds=1.2;  Dl=1;
8  Nmax=1000000;    %anaximum number of iterations
9  t=0;
10 dt=0.000001;     %timestep
11 visual_update=200; %update plot every 200 steps
12
13 q=2*pi/L*[0:N/2,-N/2+1:-1]; %wavevector in 1-d
14 [X,Y]=meshgrid(q,q);
15 absq2=X.^2+Y.^2; %absolute value of wavenumbers
16                %squared
17 U=rho+(rand(N,N)-0.5)/5; %initial solution
18
19 %This part is for the convolution. hl en hs are matrices
20 %describing the convolution kernel with radius nl and ns.
21 nl=long/gridsize;
22 ns=short/gridsize;
23 hl = fspecial('disk',round(nl));
24 hs = fspecial('disk',round(ns));
25 Uavl = imfilter(U,hl,'circular'); %Av. of U over radius  $r_l$ 
26 Uavs = imfilter(U,hs,'circular'); %Av. of U over radius  $r_s$ 
27
28 figure(1)
29 for i=1:Nmax
30     t=t+dt;
31     A=Ds*Uavs-Dl*Uavl; %compute A
32     gA=(0.5+0.5*tanh(A)); %compute g(A)
33     %This line computes  $\Delta(U/g(A)^2)$ :
34     laplaces_u=ifft2(-absq2.*fft2(U./gA.^2));
35     Unew=U+dt*laplaces_u; %update U
36     %Plot solution
37     if rem(i,visual_update) == 0
38         pcolor(Unew), shading interp, ...
39         axis('off'), axis('equal'),colorbar, drawnow
40     end
41     U=Unew;
42     Uavl = imfilter(U,hl,'circular'); %Update Uavl
43     Uavs = imfilter(U,hs,'circular'); %Update Uavs
44 end

```