

**TURING PATTERNS IN LINEAR  
CHEMICAL REACTION SYSTEMS  
WITH NONLINEAR CROSS  
DIFFUSION**

**DAVID FRANZ**  
**B.Sc., University of Lethbridge, 2005**

A Thesis  
Submitted to the School of Graduate Studies  
of the University of Lethbridge  
in Partial Fulfilment of the  
Requirements for the Degree

**MASTER OF SCIENCE**

Department of Chemistry and Biochemistry  
University of Lethbridge  
LETHBRIDGE, ALBERTA, CANADA

© David Franz, 2007

## Abstract

Turing patterns have been studied for over 50 years as a pattern forming mechanism. To date the current focus has been on the reaction mechanism, with little to no emphasis on the diffusion terms.

This work focuses on combining the simplest reaction mechanism possible and the use of nonlinear cross diffusion to form Turing patterns. We start by using two methods of bifurcation analysis to show that our model can form a Turing instability. A diffusion model (along with some variants) is then presented along with the results of numerical simulations. Various tests on both the numerical methods and the model are done to ensure the accuracy of the results. Finally an additional model that is closed to mass flow is introduced along with preliminary results.

## Acknowledgments

First I would like to thank my supervisory committee for their help and advice during the course of my thesis. I especially want to single out Dr. Marc Roussel for his faith in me, first hiring me as an undergrad, followed by taking me on as a graduate student. He has always been available, and no matter the problem, he has never been at a loss for a new idea or a new approach.

Of course this entire project would have never been possible without financial support, specifically, Keith and Hope Ferguson Memorial Scholarship, Graduate Student Travel Assistance Award, Mom & Dad, and the University of Lethbridge.

Next I would like to thank my friends and fellow grad students. Regardless of the time of day (or night), they have always been willing to talk for five minutes while I gave my computer a timeout. HJ and the Hills were especially good friends for always willing to have a beer regardless of which day of the week it was. Additionally I should like to congratulate and thank Martin. Not only has he put up with my incessant chatter on points of miscellanea (and for an especially fruitful day where he and I found out as much as we could on traffic interchanges - focusing on traffic circles), but also was a wonderful sounding board for problems.

And of course my parents; in addition to bringing me to this world, they have supported my entire (and to them never ending) post-secondary career.

# Contents

<b>List of Figures</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Bifurcation Analysis</b>	<b>6</b>
2.1 What is Bifurcation Analysis? . . . . .	6
2.2 Linear Stability Analysis . . . . .	6
2.2.1 A Foray into the Phase Plane . . . . .	10
2.3 Turing Bifurcation Analysis . . . . .	11
2.3.1 Diffusion . . . . .	11
2.3.2 Diffusion and Cross Diffusion . . . . .	14
2.4 Two-cell model . . . . .	16
<b>3 Model</b>	<b>20</b>
3.1 The First Working Model . . . . .	20
3.1.1 Alternative Model . . . . .	21
3.2 $\rho$ Model . . . . .	23
3.3 Numerical Simulation . . . . .	24
3.4 Comparison of Numerical Results to Bifurcation Analysis . . . . .	24
<b>4 Tests on the model</b>	<b>27</b>
4.1 Perturbations and Fluxes . . . . .	27
4.2 Noise . . . . .	28
4.2.1 Noise in Space . . . . .	30
4.2.2 Noise on Parameters . . . . .	32
4.3 Higher Dimensions . . . . .	33
<b>5 Closed Model</b>	<b>37</b>
5.1 Linear Stability Analysis . . . . .	37
5.2 Turing Bifurcation Analysis . . . . .	38
5.3 Numerical Results . . . . .	40
<b>6 Conclusion</b>	<b>42</b>
<b>7 Appendix</b>	<b>44</b>
7.1 Numerical Methods . . . . .	44
7.1.1 Finite Difference . . . . .	44
7.1.2 Finite Elements . . . . .	47
<b>References</b>	<b>52</b>

## List of Figures

1.1	Schematic of typical Turing pattern formation . . . . .	2
1.2	Classic animal coat patterns . . . . .	4
2.1	Phase plane diagram of the linear reaction system . . . . .	11
2.2	Plot of the trajectories for the linear reaction system . . . . .	12
2.3	Schematic of a two cell system . . . . .	16
3.1	Plots of the cross-diffusion strength as a function of $u$ and the resulting patterns from numerical simulation. . . . .	22
3.2	A plot of the pattern resulting from equation 3.2. . . . .	23
3.3	A comparison of numerical results with analytical results. . . . .	25
4.1	Plot of single pulse perturbations . . . . .	28
4.2	Plot of multiple and continuous pulse perturbations . . . . .	29
4.3	Random walk . . . . .	31
4.4	Plot of system with noise added . . . . .	32
4.5	Plot of two dimensional system evolution with and without noise . . .	34
4.6	Plot of two dimensional simulation, using two different methods . . .	35
4.7	Plot of three dimensional systems final pattern . . . . .	36
5.1	Phase plane diagram of the closed linear reaction system . . . . .	39
5.2	Three plots of the closed model under different conditions . . . . .	41
7.1	Discretization of time and space . . . . .	45

# Chapter 1

## Introduction

In 1952 Alan Turing<sup>i</sup> published the paper ‘The Chemical Basis of Morphogenesis’.<sup>3</sup> This paper in turn spurred an entire new field of nonlinear dynamics investigating what are now known as Turing patterns.

So just what is a Turing pattern? The typical definition is pattern formation via the combination of chemical reactions with diffusion. Of course that really does not illustrate how they are formed. Rather the way to think of it is as the combination of short range activation with long range inhibition<sup>ii</sup>. In the typical Turing pattern there are two components  $A$  and  $I$ . Component  $A$  is an activator and will stimulate both its own production and that of  $I$ . Component  $I$  will do the opposite: it will inhibit the production of both  $A$  and  $I$ . Now without diffusion our system would just remain at the steady state and nothing would really occur. However, if we add diffusion, specifically where  $I$  diffuses faster, we can actually make patterns.

The pattern formation first requires a single area where the system is slightly away from the steady state, say above the steady state (see figure 1.1). In this area then both  $A$  and  $I$  are produced, however since diffusion is occurring, they both diffuse away, with  $I$  moving faster and further. In this case we get a growing region where the levels of  $A$  are building which is surrounded by a region where  $I$  has coalesced, and has done the opposite and actually lowered the levels of  $A$  and  $I$  (which is sustained by the continuous flow of  $I$  from the previous high point). By the very formation of a low point in both  $A$  and  $I$ , material from the unaffected region outside starts to diffuse to the lower area, and since  $I$  diffuses faster, we again have an area where  $A$  starts to grow, and so on. Now of course, the reactions have to be nonlinear so as to prevent  $A$  from growing without bound, and to prevent  $I$  from driving the concentrations to zero, but beyond that it should be pretty easy create such a system, right? Wrong.

Before we get to the actual experimental results, we should first give a formal

---

<sup>i</sup>Yes, THE Alan Turing of computer science fame. He is often called the father of modern computer science and is responsible for the Turing machine,<sup>1</sup> the Turing test,<sup>2</sup> and in his final work Turing patterns.

<sup>ii</sup>This is not the only mechanism available for pattern formation, just the easiest to explain. One can also have what is known as activator and substrate depletion.<sup>4</sup>

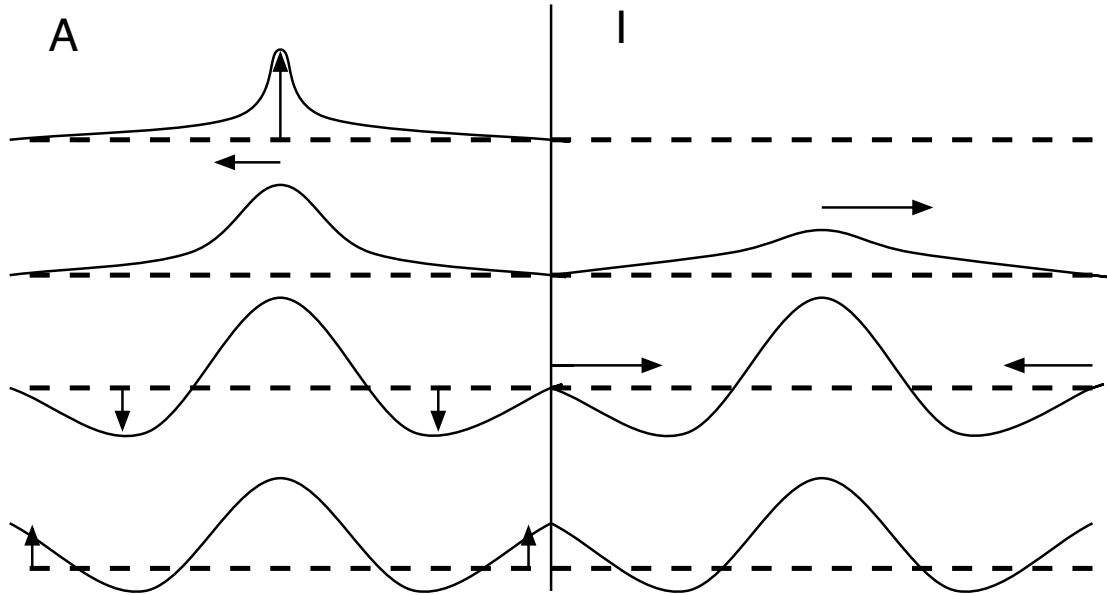


Figure 1.1: A simple example of how the typical activator-inhibitor mechanism works for pattern formation in one dimension, with both  $A$  and  $I$  shown, where the dashed line represents the steady state. In the first panel, an excess of  $A$  is added to a point. This stimulates the production of both  $A$  and  $I$ , and since  $I$  diffuses faster, it does not remain at the high point long enough to inhibit production of  $A$ , but rather diffuses out further as shown in the second panel. In the third panel one sees how away from the high point in  $A$  there is now a lower level of both  $A$  and  $I$  due to the inhibitor, which results in the diffusion from the neighboring areas. Again however,  $I$  moves faster, leaving an area with less inhibitor which in turn results in higher level in  $A$  (final panel).

definition of exactly what a Turing pattern is. In addition to the destabilization of the steady state by the process described above, Turing patterns have the two following properties:<sup>5</sup> spontaneous symmetry breaking and a characteristic wavelength independent of geometry. So what does this mean you ask. Well, spontaneous symmetry breaking is where the system will break free of the steady state on its own, that is, no external effect is required to drive the system to pattern formation. As for a characteristic wavelength independent of geometry, that is where the pattern's wavelength is not related to the size of system, but is instead inherent to the reaction and diffusion terms.

So while Turing patterns were first postulated in 1952, the first actual experimental pattern was created 38 years later in 1990<sup>6</sup> via what is known as a gel reactor. These reactors provide a constant flow of reagents into a gel medium<sup>iii</sup> where the pattern formation actually occurs. The reaction itself was a variant of a chlorite-iodide reaction where chloride and iodide ions in different oxidation states react with each other over time in a multistep mechanism. The progress of the reaction is made visible by the change in color of starch indicator preloaded into the gel, which in this case is the formation of patterns in the gel's surface.

Turing's original paper was aimed at explaining how biological patterns form. The classical examples given are always animal coat patterns (figure 1.2), but Turing patterns are also cited where any sort of repeating pattern occurs, in everything from embryonic development to brain structure.<sup>8</sup> The first real evidence of Turing patterns in biology was in 1995<sup>9</sup> when it was noticed that the patterns on angelfish (*Pomacanthus*) have the same size regardless of animal size, that is, as the animal grows, more stripes are inserted (fitting in perfectly with the characteristic wavelength definition). Furthermore, in 2006 the first *in vivo* experiments<sup>10</sup> were conducted on mice that showed some evidence of a reaction-diffusion mechanism with an inhibitor being responsible for follicle spacing in mice.

All Turing patterns, regardless of the application, require diffusion. The most common way, by far, to represent diffusion is to use what is called Fickian diffusion first developed by Adolf Fick in 1855.<sup>12</sup> The technique he used was measuring the concentrations of salt in a column, and he was able to show that there is a diffusion constant which is independent of concentration, similar to Fourier's heat conductivity. The interesting thing is that the situations under which Fickian diffusion can occur are actually quite rare. For example, in multicomponent system, one can show<sup>13</sup> that the diffusion rate should differ from the Fickian form and that there should be cross diffusion<sup>iv</sup>. Furthermore, when one starts to involve ionic species, the effect becomes even more pronounced, leading to very interesting diffusion effects both with and

---

<sup>iii</sup>The gel medium is an important factor insuring that it is actually diffusion driving the pattern formation, rather than artificial flows such as convection currents. The use of gels also allows the experimentalist the ability to vary the diffusivities of the system,<sup>7</sup> which is an important factor in creating Turing patterns.

<sup>iv</sup>Cross diffusion is where a component is transported via a gradient in another.



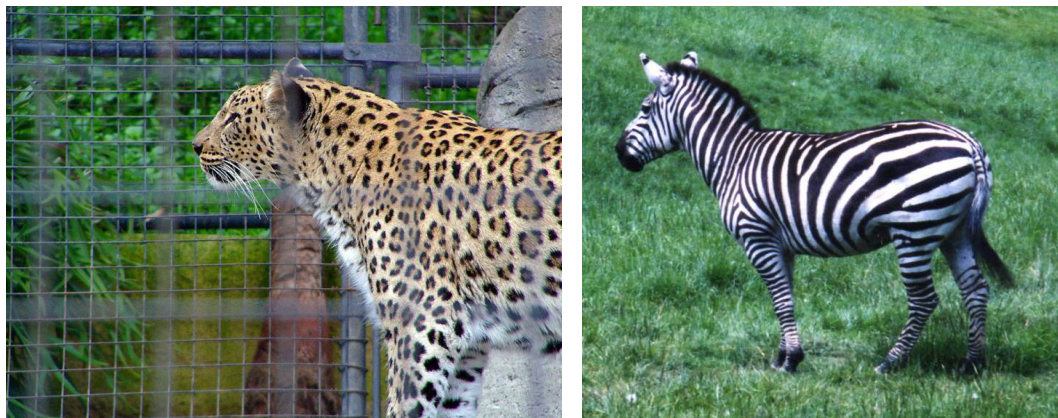


Figure 1.2: The two classic examples of patterns in biology, a leopard with its spots and a zebra with its stripes (images used with permission from wikipedia commons<sup>11</sup>).

without<sup>14,15</sup> external electric fields. Even without making the terms nonlinear, the addition of cross diffusion alone has led to pattern formation in both physics<sup>16</sup> and ecology.<sup>17</sup> Concentration-dependent diffusion coefficients have been studied in the Gray-Scott model<sup>18,19</sup> in a regime where non-Turing spatial patterns are observed.

Another interesting fact is that the vast majority of systems<sup>4,20,21</sup> displaying Turing bifurcations are located (in parameter space) next to Andonov-Hopf bifurcations<sup>v</sup>. Furthermore, it has been shown<sup>22</sup> that any conventional system with either a Turing instability or an Andronov-Hopf bifurcation will be able to form the other. And while these bifurcations are interesting, this means that these systems are already close to a point where they are no longer stable<sup>vi</sup>.

In this thesis we shall introduce a novel mechanism for Turing patterns. The goal here is to form patterns with only linear chemical reactions. To start with we do some pen and paper analysis in chapter 2 to determine the conditions required for a Turing instability<sup>vii</sup> with linear reactions. In addition to the standard bifurcation analysis, a cellular technique is used, and both methods show that cross diffusion is required.

<sup>v</sup>An Andronov-Hopf bifurcation is the point where a system starts or stops (depending on direction) forming limit cycles (continuous oscillations in variable levels).

<sup>vi</sup>Another interesting point is that with current models and experimental systems one can predict the wavelength of Turing patterns based on the average diffusion coefficients and the period of the limit cycle.<sup>7</sup>

<sup>vii</sup>The destabilization of the steady state which is required for any system to evolve into a Turing pattern

With this in hand we present the diffusion model in chapter 3 which was used to numerically create Turing patterns. Of course, running numerical simulations without checks is never good, so in chapter 4 we carry out some tests and run some interesting simulations to really flesh out both our numerical methods and our understanding of our models. For chapter 5, a new model is introduced in which the system is closed to mass flow, and while the results are only preliminary, they are quite exciting. With the exception of chapter 5, a quick summary of the results is available in paper form.<sup>23</sup> We then close with the conclusion and an appendix briefly going through the numerical methods used.

## Chapter 2

### Bifurcation Analysis

#### 2.1 What is Bifurcation Analysis?

Bifurcation analysis is where we determine, via pen and paper, how the behavior of a given system varies as a function of one (or more) of the parameters. These changes in behavior can be jumps in equilibrium values, collapse or creation of limit cycles (oscillating reactions), or any of an ever growing array of possible bifurcations. If the qualitative behavior of the system changes, we call this a bifurcation. In our case the system itself is defined as a series of differential equations (DE's), where each DE determines the behavior, through time, of a variable as a function of itself, other variables, and system parameters.

These system parameters are things that we can change, such as temperature, pressure, rate constants, volume, etc. Depending on the parameter, it is either something that can be changed directly (temperature), or something that we can change indirectly (kinetic parameters, diffusion coefficients<sup>i</sup>). While there are many different types of bifurcations, the type we are looking for is the transition from a stable homogeneous (in space) steady state to a stable spatial pattern via a Turing instability.

We first introduce our linear reaction model and show that in the absence of diffusion, it is globally stable. We then add diffusion followed by cross diffusion and gauge the effects. Finally, a cellular technique is introduced (two-cell model) which will further confirm our results.

For the most part, we follow the general form of such analysis.<sup>24,25</sup> In later chapters, we will be able to compare the predictions made here with numeric results.

#### 2.2 Linear Stability Analysis

We start by constructing a generic two-species mechanism for an open system with only first-order reactions (where  $U$  and  $V$  are the two species and the various  $k$  terms

---

<sup>i</sup>These are both of interest for Turing patterns. Typically these can be altered by changing either the temperature or feed rates of a system (which affect the constant influx or efflux of a component). For the diffusion coefficients, one can also change the type of medium used in the system.

rate constants):



Converting this to rate equations (differential equations for the change in concentration over time -  $T$ )<sup>ii</sup>, where  $U$  and  $V$  represent the concentrations of their respective species; we get:

$$\frac{\partial U}{\partial T} = k_1 + k_4V - k_2U - k_3U, \quad (2.2a)$$

$$\frac{\partial V}{\partial T} = k_5 + k_3U - k_4V - k_6V. \quad (2.2b)$$

In order to simplify further the analysis we nondimensionalize our system. This has two advantages, first of all it typically eliminates one parameter for each measurement scale and it helps reduce the clutter of carrying units around all the time. We begin by determining the units associated with all the parameters and variables:

$$U = V = \text{mol L}^{-1}, \quad (2.3)$$

$$T = \text{s}, \quad (2.4)$$

$$k_1 = k_5 = \text{mol L}^{-1}\text{s}^{-1}, \quad (2.5)$$

$$k_2 = k_3 = k_4 = k_6 = \text{s}^{-1}. \quad (2.6)$$

The goal is then to make all the units cancel and create dimensionless new parameters and variables, where the parameters are now represented by Greek letters and the

---

<sup>ii</sup>We used capital  $T$  here only to differentiate it from the dimensionless time  $t$  which we will use for the remainder of this document.

variables by lowercase letters. We get the following series of equations (noting that there are alternative ways of doing this, with no effect on the system's behavior<sup>iii</sup>):

$$u = \frac{Uk_2}{k_5}, \quad v = \frac{Vk_2}{k_5}, \quad t = Tk_2, \quad (2.7a)$$

$$\alpha = \frac{k_1}{k_5}, \quad \beta = \frac{k_4}{k_2}, \quad \gamma = \frac{k_3}{k_2}, \quad \sigma = \frac{k_6}{k_2}; \quad (2.7b)$$

giving:

$$\frac{\partial u}{\partial t} = \alpha + \beta v - u - \gamma u, \quad (2.8a)$$

$$\frac{\partial v}{\partial t} = 1 + \gamma u - \sigma v - \beta v. \quad (2.8b)$$

By definition, a Turing bifurcation can only occur when the steady state is stable in the absence of diffusion. Therefore we need to determine the steady-state values for  $u$  and  $v$  by solving  $\frac{\partial u}{\partial t} = 0$  and  $\frac{\partial v}{\partial t} = 0$ :

$$u^* = \frac{\beta + \alpha \sigma + \alpha \beta}{\sigma + \sigma \gamma + \beta}, \quad (2.9a)$$

$$v^* = \frac{1 + \gamma + \gamma \alpha}{\sigma + \sigma \gamma + \beta}. \quad (2.9b)$$

Local stability analysis typically involves only the behavior of the system near the steady state, specifically whether the system will return to the steady state after a small displacement in both  $u$  and  $v$  ( $\delta u$  and  $\delta v$ ). The general procedure is to first linearize the system via a Taylor expansion, where we represent  $\frac{\partial u}{\partial t}$  as  $\dot{u}$ :

$$\begin{aligned} \dot{u}(u^* + \delta u, v^* + \delta v) &= \dot{u}(u^*, v^*) + \delta u \frac{\partial \dot{u}(u^*, v^*)}{\partial u} + \delta v \frac{\partial \dot{u}(u^*, v^*)}{\partial v} \\ &+ \frac{(\delta u)^2}{2} \frac{\partial^2 \dot{u}(u^*, v^*)}{\partial u^2} + \delta u \delta v \frac{\partial^2 \dot{u}(u^*, v^*)}{\partial u \partial v} + \frac{(\delta v)^2}{2} \frac{\partial^2 \dot{u}(u^*, v^*)}{\partial v^2} + \dots \end{aligned} \quad (2.10)$$

Since the displacement from the steady state is to be small, any product of  $\delta i \delta j$  will result in a negligible small term and the higher order terms will be even smaller, so

---

<sup>iii</sup>In some cases the scaling is a far greater issue. For example, when trying to apply the steady-state approximation (SSA) to the Michaelis-Menten mechanism,<sup>26</sup> one wants to scale the time such that the SSA is easily applied.

we only consider the first line (as in the linear terms of  $\delta u$  and  $\delta v$ , for which local stability analysis gets its other common name, linear stability analysis). One also sees that the term  $\dot{u}(u^*, v^*)$  by definition should be zero. If we plug in the actual values into equation 2.10 we get (realizing that we are now measuring the change in our displacement from equilibrium):

$$\dot{\delta u} = \delta u(-1 - \gamma) + \delta v(\beta), \quad (2.11a)$$

$$\dot{\delta v} = \delta u(\gamma) + \delta v(-\sigma - \beta). \quad (2.11b)$$

Thus what we really have are two linear equations dictating the behavior of  $\delta u$  and  $\delta v$ . For most systems, these are only accurate in the immediate area around the steady state, for once the system gets too far away from the steady state, the higher order terms are no longer negligible. However, our system is already linear, thus there are no higher order terms. Regardless, we can now represent the linearized system in a matrix form called the Jacobian:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \dot{x}_1}{\partial x_1} & \cdots & \frac{\partial \dot{x}_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \dot{x}_n}{\partial x_1} & \cdots & \frac{\partial \dot{x}_n}{\partial x_n} \end{bmatrix}. \quad (2.12)$$

For our model, then this becomes:

$$\mathbf{J} = \begin{bmatrix} -1 - \gamma & \beta \\ \gamma & -\sigma - \beta \end{bmatrix}. \quad (2.13)$$

The equation

$$\delta x(t) = \delta x(0)e^{\lambda t}, \quad (2.14)$$

gives the behavior of one component for a system of linear differential equations based on its eigenvalue  $\lambda$ .<sup>iv</sup> Thus we solve  $|\mathbf{J} - \lambda \mathbf{I}| = 0$  for  $\lambda$ , where  $|\cdot|$  denotes the determinant:

$$\lambda^2 + \lambda(\sigma + \beta + 1 + \gamma) + \sigma + \beta + \gamma\sigma = 0. \quad (2.15)$$

---

<sup>iv</sup>This is not always the case, if two (or more) components have identical eigenvalues ( $\lambda$ ) the equation becomes slightly more complicated, but nevertheless still solvable with an additional term.<sup>27</sup>

$$\lambda = \frac{1}{2} \left( -(\sigma + \beta + 1 + \gamma) \pm \sqrt{(\sigma + \beta + 1 + \gamma)^2 - 4(\sigma + \beta + \gamma\sigma)} \right). \quad (2.16)$$

Since  $\sigma + \beta + 1 + \gamma > 0$  and  $\sigma + \beta + \gamma\sigma > 0$ , both eigenvalues are always negative, and by equation 2.14 we can tell that the steady state is always stable in the absence of diffusion. This of course is entirely expected for a linear chemical system.<sup>28,29</sup>

### 2.2.1 A Foray into the Phase Plane

For two-dimensional systems, one can visualize the behavior of the system quite easily using phase-plane analysis. A phase plane is a diagram that illustrates the behavior of a system, independent of time. What a phase plane does is that it shows how two variables will change at any given value of themselves (for example, for a given set of parameters,  $u = 2$ , and  $v = 0$  will result in  $u$  decreasing and  $v$  increasing, which in turn will be represented by an arrow in the plane showing a decrease in  $u$  and an increase in  $v$ ). This is done via a diagram with each axis being a variable, and given that paper exists in two dimensions, this works best for two-dimensional systems. We start by determining lines called nullclines. Nullclines are curves where one variable remains constant, determined by solving  $\dot{u} = 0$  or  $\dot{v} = 0$ . Thus along the  $\dot{u} = 0$  nullcline,  $u$  remains constant. The point where the two lines intersect determines where the steady-state is for our system<sup>v</sup>. With the nullclines, we can now create a vector field (see figure 2.1). The vector field is a series of arrows showing in which direction the system will go from the current point, with the size of the arrow showing 'speed' of the change. Thus a small arrow shows a slow change, while a large arrow shows a quick change. The evolution of the system can also be shown with trajectories (see figure 2.2), in which each line shows the history of a single trace (i.e. for each line the system is started out at a point, and the system evolves over time and the line shows how the state of the system changes). The vector field has the advantage of showing how the system evolves from any point in its local neighborhood, while the trajectories show how the system changes over its entire evolution.

---

<sup>v</sup>This is not true of all systems. In some systems the intersection of two nullclines is where an unstable node or focus is. Either way, it is where  $\dot{u} = \dot{v} = 0$ .

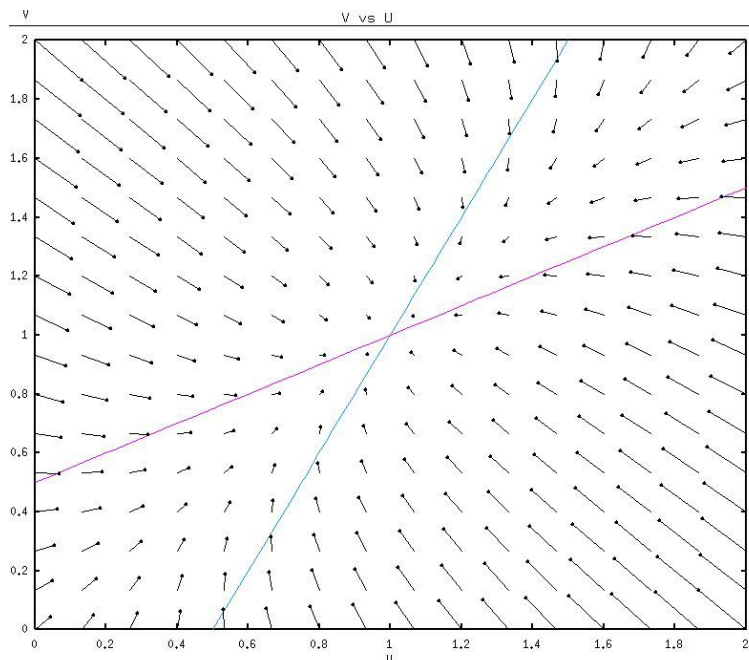


Figure 2.1: Phase plane diagram for the system described in equation 2.8. The left to right nullcline is for  $\dot{u} = 0$ , given by the formula  $u = \frac{\alpha + \beta v}{1 + \gamma}$  and the bottom to top nullcline is for  $\dot{v} = 0$ , given by the formula  $v = \frac{1 + \gamma u}{\sigma + \beta}$ . The actual plot itself is from XPPAUT<sup>30</sup> which uses different sized arrows to represent the magnitudes of  $\dot{u}$  and  $\dot{v}$  at each point. We used a value of 1 for each of the parameters.

## 2.3 Turing Bifurcation Analysis

### 2.3.1 Diffusion

Having determined that in the absence of diffusion the steady state is stable, we next check to see if the addition of diffusion will create a Turing instability. We start by introducing very generic reaction-diffusion equations for  $u$  and  $v$ , where  $J$  represents flux<sup>vi</sup>:

$$\frac{\partial u}{\partial t} = -\frac{\partial J_u}{\partial z} + \alpha + \beta v - u - \gamma u, \quad (2.17a)$$

$$\frac{\partial v}{\partial t} = -\frac{\partial J_v}{\partial z} + 1 + \gamma u - \sigma v - \beta v. \quad (2.17b)$$

<sup>vi</sup>Here and all through the following, we work in only one spatial dimension; it's far easier notationally, and has no bearing on the final results.



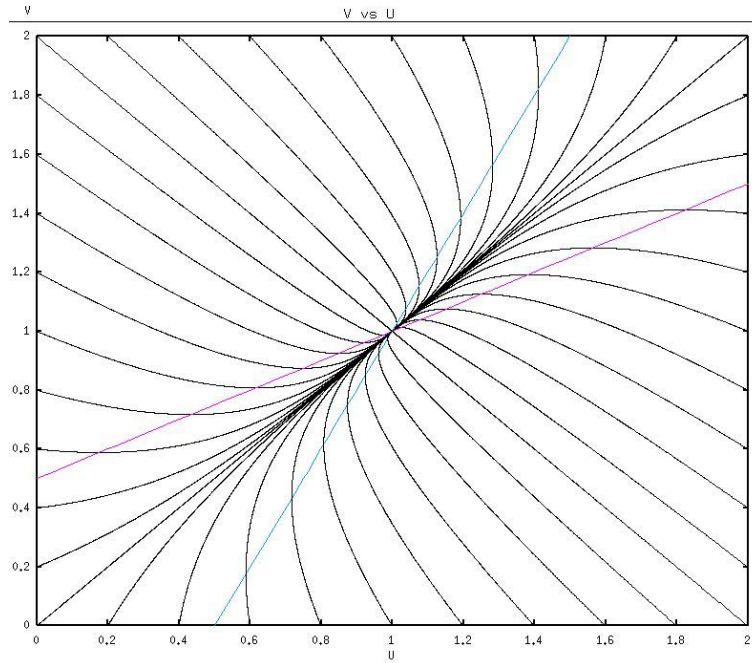


Figure 2.2: A plot of the trajectories, which shows especially clearly how the trajectories gather along the diagonal, then proceed to the equilibrium. Similar to the previous plot, XPPAUT was used with a value of 1 for each the parameters, with the nullcline being the same as the previous figure.

We allow the diffusion coefficients to depend on  $u$  and  $v$ . Fick's first law<sup>12</sup> would then give the fluxes:

$$J_u = -D_{uu}(u, v) \frac{\partial u}{\partial z}, \quad (2.18a)$$

$$J_v = -D_{vv}(u, v) \frac{\partial v}{\partial z}, \quad (2.18b)$$

where  $D_{uu}$  and  $D_{vv}$  are the diffusion coefficients for  $u$  and  $v$  respectively. We consider an arbitrarily small displacement from equilibrium  $(\delta u, \delta v)$  at a point in  $z$ . First expanding  $D_{uu}(u^* + \delta u, v^* + \delta v)$  via Taylor expansion we get:

$$D_{uu}(u^* + \delta u, v^* + \delta v) = D_{uu}(u^*, v^*) + \delta u \frac{\partial D_{uu}(u^*, v^*)}{\partial u} + \delta v \frac{\partial D_{uu}(u^*, v^*)}{\partial v} + \dots \quad (2.19)$$

We then plug the first terms of this expansion into our flux term to get:

$$J_u = -D_{uu}(u^*, v^*) \frac{\partial(\delta u)}{\partial z} - \delta u \frac{\partial D_{uu}(u^*, v^*)}{\partial u} \frac{\partial(\delta u)}{\partial z} - \delta v \frac{\partial D_{uu}(u^*, v^*)}{\partial v} \frac{\partial(\delta u)}{\partial z}. \quad (2.20)$$

Assuming that  $\delta u$  and  $\delta v$  are smooth in  $z$  and that our diffusion coefficients are smooth near the steady state, we can see that only the first term of the expansion above is not negligible, thus to first order we have:

$$J_u = -D_{uu}^* \frac{\partial(\delta u)}{\partial z}, \quad (2.21a)$$

$$J_v = -D_{vv}^* \frac{\partial(\delta v)}{\partial z}, \quad (2.21b)$$

where we let  $D_{ii}(u^*, v^*) = D_{ii}^*$ . Similarly to linear order, equation 2.17 becomes:

$$\frac{\partial(\delta u)}{\partial t} = D_{uu}^* \frac{\partial^2(\delta u)}{\partial z^2} + \beta \delta v - \delta u - \gamma \delta u, \quad (2.22a)$$

$$\frac{\partial(\delta v)}{\partial t} = D_{vv}^* \frac{\partial^2(\delta v)}{\partial z^2} + \gamma \delta u - \sigma \delta v - \beta \delta v. \quad (2.22b)$$

We can now define a diffusion matrix  $\mathbf{D}$  with our diffusion terms:

$$\mathbf{D} = \begin{bmatrix} D_{uu}^* & 0 \\ 0 & D_{vv}^* \end{bmatrix}. \quad (2.23)$$

Similar to the stability analysis above, we solve for our eigenvalues, this time however, the evolution through time and space is dictated by a different equation:<sup>31</sup>

$$\delta x(t) = \delta x(0) e^{\lambda t} e^{ikz}. \quad (2.24)$$

The added  $e^{ikz}$  term is used to represent a displacement in  $z$ , with a wavenumber  $k$ . This based on Euler's formula:  $e^{ikz} = \cos(kz) + i \sin(kz)$ . Thus we need to solve  $|\mathbf{J} - k^2 \mathbf{D} - \lambda \mathbf{I}| = 0$ , and find the conditions where  $\lambda$  is no longer negative for a real value of  $k$ , giving not only the conditions where the system will become unstable, but also the initial wavelength of the instability.

$$(\lambda + D_{uu}^* k^2 + 1 + \gamma)(\lambda + D_{vv}^* k^2 + \sigma + \beta) + \beta\gamma = 0. \quad (2.25)$$

$$\therefore \lambda = \frac{1}{2}(-B \pm \sqrt{B^2 - 4C}), \quad (2.26a)$$

$$B = k^2(D_{vv}^* + D_{uu}^*) + \sigma + \beta + 1 + \gamma, \quad (2.26b)$$

$$C = k^4(D_{uu}^* D_{vv}^*) + k^2(\gamma D_{vv}^* + \beta D_{uu}^* + \sigma D_{uu}^* + D_{vv}^*) + \sigma + \beta + \sigma\gamma. \quad (2.26c)$$

We can readily see that both the  $B$  and  $C$  terms are positive for all values of  $k$ . Thus both roots of equation 2.26a are negative, meaning that the homogeneous steady state is always stable. Again this is not a great surprise. There is, after all, a reason why Turing patterns are generally thought to be associated with nonlinear kinetics.

### 2.3.2 Diffusion and Cross Diffusion

We repeat the analysis again, but this time with cross diffusion (again given in a general form with dependences on both  $u$  and  $v$ ), so that the flux terms now read as:

$$J_u = -D_{uu}(u, v) \frac{\partial u}{\partial z} - D_{uv}(u, v) \frac{\partial v}{\partial z}, \quad (2.27a)$$

$$J_v = -D_{vv}(u, v) \frac{\partial v}{\partial z} - D_{vu}(u, v) \frac{\partial u}{\partial z}. \quad (2.27b)$$

As before, we only consider the case of a small displacement from the steady state, resulting in the following linearized partial differential equations:

$$\frac{\partial(\delta u)}{\partial t} = D_{uu}^* \frac{\partial(\delta u)}{\partial z^2} + D_{uv}^* \frac{\partial(\delta v)}{\partial z^2} + \beta\delta v - \delta u - \gamma\delta u, \quad (2.28a)$$

$$\frac{\partial(\delta v)}{\partial t} = D_{vv}^* \frac{\partial(\delta v)}{\partial z^2} + D_{vu}^* \frac{\partial(\delta u)}{\partial z^2} + \gamma\delta u - \sigma\delta v - \beta\delta v. \quad (2.28b)$$

In terms of our analysis, the only difference in having cross diffusion is in the diffusion matrix  $\mathbf{D}$ :

$$\mathbf{D} = \begin{bmatrix} D_{uu}^* & D_{uv}^* \\ D_{vu}^* & D_{vv}^* \end{bmatrix}. \quad (2.29)$$

Again  $|\mathbf{J} - k^2\mathbf{D} - \lambda\mathbf{I}| = 0$  is solved:

$$(\lambda + D_{uu}^*k^2 + 1 + \gamma)(\lambda + D_{vv}^*k^2 + \sigma + \beta) - (D_{uv}^*k^2 - \beta)(D_{vu}(u^*, v^*)k^2 - \gamma) = 0. \quad (2.30)$$

$$\therefore \lambda = \frac{1}{2}(-B \pm \sqrt{B^2 - 4C}), \quad (2.31a)$$

$$B = k^2(D_{vv}^* + D_{uu}^*) + \sigma + \beta + 1 + \gamma, \quad (2.31b)$$

$$C = k^4(D_{uu}^*D_{vv}^* - D_{uv}^*D_{vu}^*) + k^2(\gamma D_{uv}^* + \gamma D_{vv}^* + \beta D_{vu}^* + \beta D_{uu}^* + \sigma D_{uu}^* + D_{vv}^*) + \sigma + \beta + \sigma\gamma. \quad (2.31c)$$

It is not possible to make  $B$  negative<sup>vii</sup>, but this time there are two ways to make  $C$  negative. If

$$D_{uv}^*D_{vu}^* > D_{uu}^*D_{vv}^* \quad (2.32)$$

then  $C$  is negative at large wavenumbers. However, the large  $k$  (small wavelength) regime is not one typically considered in studies of Turing bifurcations and we leave it aside<sup>viii</sup>. The second case, and the regime on which we focused this study, is:

$$\gamma D_{uv}^* + \gamma D_{vv}^* + \beta D_{vu}^* + \beta D_{uu}^* + \sigma D_{uu}^* + D_{vv}^* < 0, \quad (2.33)$$

which leads to a Turing instability in the typical closed range of  $k$ . Note that there is no requirement for unequal diffusion coefficients, but the cross diffusion coefficients will have to be negative. Also this analysis only shows that the steady state can be made unstable (Turing instability), not that there will be a stable Turing pattern.

---

<sup>vii</sup>Diffusion coefficients always have to be positive (whether linear or not) but cross-diffusion coefficients can be either negative or positive.

<sup>viii</sup>The reason that this regime is not considered in Turing patterns is that the system, at least close to the steady state, will actually tend to infinite wavenumbers. This makes this regime difficult to model and unlikely to be seen in reality, although it has been investigated in systems with backward diffusion.<sup>32</sup>

## 2.4 Two-cell model

The two-cell model is a way of adding diffusion to a system and using linear stability analysis, but without a diffusion matrix. In Turing's original paper<sup>3</sup> a two cell model was used to illustrate the workings of a system, but typically these are only used from compartmentalized systems, such as membrane problems<sup>33</sup> or cellular networks.<sup>34</sup> To do this we consider two cells joined along a surface with a thickness of  $\Delta z$ . Along this surface, we allow diffusion (and cross diffusion) between the cells. However, since we are considering only two discrete cells, we can express both the kinetics and diffusion terms as ordinary differential equations (rather than partial differential equations).

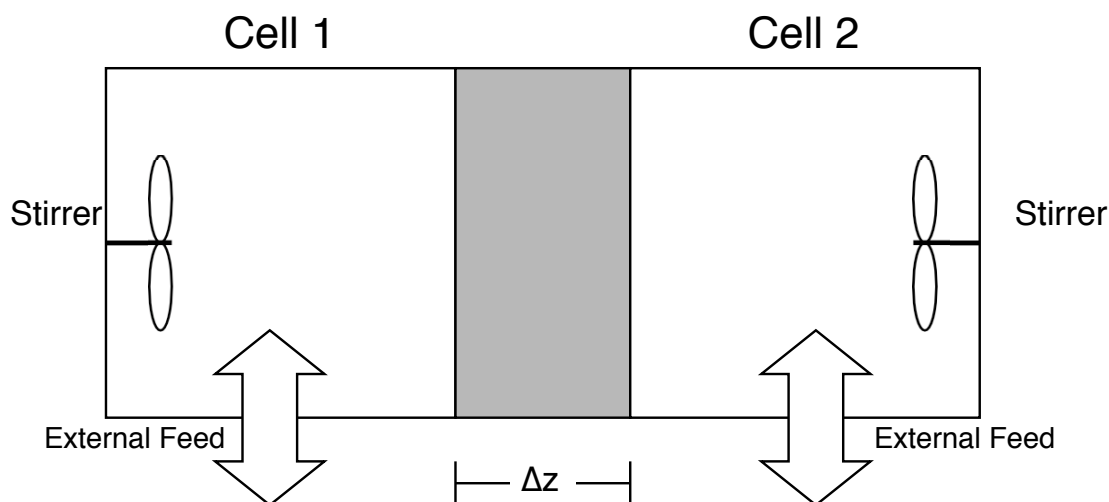


Figure 2.3: A schematic of the two cell model. Each cell is stirred (preventing diffusion within the cell), while the cells are separated by a distance  $\Delta z$ . The external feed adds and removes  $u$  and  $v$  as per the reaction scheme.

Assigning the cells are labels 1 and 2, we can then construct our linearized rate equations:

$$\frac{\partial u_1}{\partial t} = \alpha + \beta v_1 - u_1 - \gamma u_1 + D_{uu}^* \frac{u_2 - u_1}{\Delta z} + D_{uv}^* \frac{v_2 - v_1}{\Delta z}, \quad (2.34a)$$

$$\frac{\partial v_1}{\partial t} = 1 + \gamma u_1 - \sigma v_1 - \beta v_1 + D_{vv}^* \frac{v_2 - v_1}{\Delta z} + D_{vu}^* \frac{u_2 - u_1}{\Delta z}, \quad (2.34b)$$

$$\frac{\partial u_2}{\partial t} = \alpha + \beta v_2 - u_2 - \gamma u_2 + D_{uu}^* \frac{u_1 - u_2}{\Delta z} + D_{uv}^* \frac{v_1 - v_2}{\Delta z}, \quad (2.34c)$$

$$\frac{\partial v_2}{\partial t} = 1 + \gamma u_2 - \sigma v_2 - \beta v_2 + D_{vv}^* \frac{v_1 - v_2}{\Delta z} + D_{vu}^* \frac{u_1 - u_2}{\Delta z}, \quad (2.34d)$$

where  $\Delta z$  is just another parameter (not a spatial term - just scaling the diffusion terms). The homogeneous steady state of this system is the same as calculated before (equation 2.9). So we now find the Jacobian of this very generic system:

$$\mathbf{J} = \begin{bmatrix} -1 - \gamma - \frac{D_{uu}^*}{\Delta z} & \beta - \frac{D_{uv}^*}{\Delta z} & \frac{D_{uu}^*}{\Delta z} & \frac{D_{uv}^*}{\Delta z} \\ \gamma - \frac{D_{vu}^*}{\Delta z} & -\sigma - \beta - \frac{D_{vv}^*}{\Delta z} & \frac{D_{vu}^*}{\Delta z} & \frac{D_{vv}^*}{\Delta z} \\ \frac{D_{uu}^*}{\Delta z} & \frac{D_{uv}^*}{\Delta z} & -1 - \gamma - \frac{D_{uu}^*}{\Delta z} & \beta - \frac{D_{uv}^*}{\Delta z} \\ \frac{D_{vu}^*}{\Delta z} & \frac{D_{vv}^*}{\Delta z} & \gamma - \frac{D_{vu}^*}{\Delta z} & -\sigma - \beta - \frac{D_{vv}^*}{\Delta z} \end{bmatrix}. \quad (2.35)$$

Similar to our first linear stability analysis, we solve  $|\mathbf{J} - \lambda \mathbf{I}| = 0$  to give:

$$\frac{1}{(\Delta z)^2} f(\lambda) g(\lambda) = 0, \quad (2.36)$$

where:

$$f(\lambda) = \lambda^2 + \lambda(1 + \gamma + \beta + \sigma) + \beta + \sigma + \gamma\sigma, \quad (2.37a)$$

$$\begin{aligned} g(\lambda) = & \lambda^2(\Delta z)^2 + \lambda[(\Delta z)^2\beta + 2D_{uu}\Delta z + \gamma(\Delta z)^2 + (\Delta z)^2 + (\Delta z)^2\sigma + 2\Delta z D_{vv}^*] \\ & + [\beta(\Delta z)^2 + 2\beta\Delta z D_{vu}^* + 2D_{uu}^*\beta\Delta z - 4D_{uv}^* D_{vu}^* + 2D_{uu}^*\sigma\Delta z + \gamma(\Delta z)^2\sigma \\ & + 2D_{uv}^*\gamma\Delta z + \sigma(\Delta z)^2 + 2\Delta z D_{vv}^* + 4D_{uu}^* D_{vv}^* + 2\gamma\Delta z D_{vv}^*]. \end{aligned} \quad (2.37b)$$

Since we need equation 2.36 to be zero, either  $f(\lambda)$  or  $g(\lambda)$  must be equal to zero. This gives us then two sets of quadratic equations to solve, with a total of four different

values of  $\lambda$ .  $f(\lambda)$  is the easiest to solve:

$$\lambda_{1,2} = \frac{1}{2}(-B \pm \sqrt{B^2 - 4C}), \quad (2.38)$$

$$B = 1 + \gamma + \beta + \sigma, \quad (2.39a)$$

$$C = \beta + \sigma + \gamma\sigma. \quad (2.39b)$$

Similar to above,  $B$  is always positive, and so is  $C$ , thus there is no way for  $\lambda_{1,2}$  to have any positive values, leaving us with  $g(\lambda)$ , which is a bit more complex:

$$\lambda_{3,4} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}, \quad (2.40a)$$

$$A = (\Delta z)^2, \quad (2.40b)$$

$$B = (\Delta z)^2\beta + 2D_{uu}^*\Delta z + \gamma(\Delta z)^2 + (\Delta z)^2 + (\Delta z)^2\sigma + 2\Delta zD_{vv}^*, \quad (2.40c)$$

$$C = \beta(\Delta z)^2 + 2\beta\Delta zD_{vu}^* + 2D_{uu}^*\beta\Delta z - 4D_{uv}^*D_{vu}^* + 2D_{uu}^*\sigma\Delta z + \gamma(\Delta z)^2\sigma + 2D_{uv}^*\gamma\Delta z + \sigma(\Delta z)^2 + 2\Delta zD_{vv}^* + 4D_{uu}^*D_{vv}^* + 2\gamma\Delta zD_{vv}^*. \quad (2.40d)$$

Yet again we come back to the quadratic equation. This time we have a term not equal to one for  $A$ , but since it is  $(\Delta z)^2$ , it does not affect the sign of the solution (but will affect the magnitude). Likewise,  $B$  will also remain positive, leaving us again to  $C$ . If  $C$  can be made negative, then one solution of  $\lambda$  will be positive, destabilizing the homogeneous steady state.

There are two means by which we can make  $C$  negative. This should sound familiar. If we set  $\Delta z$  to be very small, we then require  $D_{uv}^*D_{vu}^* > D_{uu}^*D_{vv}^*$  to create an instability - however this would require the pattern to have a very high wavenumber (see section 3.4), which is typically not where we look for Turing instabilities. Rather if we collect the  $C$  term by  $\Delta z$  we get:

$$\begin{aligned} C = & (\Delta z)^2(\gamma\sigma + \sigma + \beta) \\ & + \Delta z(2D_{uv}^*\gamma + 2D_{uu}^*\beta + 2\gamma D_{vv}^* + 2D_{vv}^* + 2\beta D_{vu}^* + 2D_{uu}^*\sigma) \\ & + 4D_{uu}^*D_{vv}^* - 4D_{uv}^*D_{vu}^* \end{aligned} \quad (2.41)$$

Since we know the kinetic terms to be positive, we need the  $\Delta z$  term to be sufficiently

negative so that for a range of  $\Delta z$  (before the  $(\Delta z)^2$  term gets too large)  $C$  will be negative, giving the inequality:

$$2(D_{uv}^* \gamma + D_{uu}^* \beta + \gamma D_{vv}^* + D_{vv}^* + \beta D_{vu}^* + D_{uu}^* \sigma) < 0 \quad (2.42)$$

Of course we can divide out the 2, which then gives us the exact same condition as equation 2.33.

As will be shown later, the two techniques (linear stability analysis of a spatial system and the two-cell model) do not match perfectly with each other, but we have now shown that both give the same conditions for an instability to occur.



## Chapter 3

### Model

#### 3.1 The First Working Model

In the previous chapter, two separate methods were presented which showed that a linear reaction model combined with both diffusion and cross diffusion should be able to create a Turing instability. However, if one only uses simple (concentration-independent) diffusion and cross diffusion, the concentrations grow without bound. This actually makes quite a bit of sense. In most systems the higher order terms in equation 2.10 become significant as the system gets further away from equilibrium<sup>i</sup>. However, since our system has no higher order terms in the kinetics, with simple diffusion the higher order terms in equation 2.19 are also zero. If the parameters are set to make the steady-state unstable, the system becomes globally unstable.

So what is then needed is a model which has either nonlinear chemical reactions or some nonlinear diffusion coefficients. As one can surmise by the title of this thesis (and by how much the other scenario has been done) we investigated nonlinear diffusion coefficients. By both equations 2.33 and 2.42 we see that we need two things, negative cross diffusion and positive diffusion coefficients. So in order to stabilize the system away from the steady state, we need to either have the diffusion coefficient rise away from the steady state or have the cross-diffusion coefficient drop off away from the steady state. As was stated in chapter 1, to expect diffusion coefficients to not change as a function of concentration is unreasonable, but in cases where the system is well away from the bifurcation point (in diffusion strength), it would take quite a change in diffusion coefficients to restabilize the system.

If we look at cross-diffusion however, we can see quite a few different mechanisms where the cross-diffusion strength could change as a function of concentration, most notably in biological systems. Consider for example, secondary active transport<sup>35</sup> through cells<sup>ii</sup>; the transport rate will increase linearly with concentration until saturation of the transporter sites is approached, at which point the system plateaus,

---

<sup>i</sup>This is the mechanism by which typical Turing pattern forming systems are stabilized.

<sup>ii</sup>Secondary active transport is where the movement of one molecule down a gradient is used to move another molecule against the gradient. The common example used for this mechanism is systems that use sodium ion gradients to power the transport of sucrose.

and any further increase in concentration on one side will not affect the flux in any way (implying that the coefficient will instead drop).

The form that was eventually chosen was in the form of a Hill equation:<sup>36</sup>

$$D_{uv} = \frac{q_u u^m}{K_u^m + u^m} + D_{uv}^\circ, \quad (3.1a)$$

$$D_{vu} = \frac{q_v v^n}{K_v^n + v^n} + D_{vu}^\circ. \quad (3.1b)$$

Now, while both  $D_{uv}$  and  $D_{vu}$  are given, in reality we set one to zero (see section 3.2 below for the exception) to avoid entering the area of parameter space where Turing patterns at large wavenumbers are predicted, thus unless explicitly mentioned, all cross-diffusion terms not given are zero. Also worth mentioning is that all the parameters except for  $m$ ,  $n$ ,  $K_u$ , and  $K_v$ , can be negative or positive.

Referring to figure 3.1 we see the general schemes under which our system works. Under the first scheme, the cross-diffusion goes to zero at low concentration, while under the second scheme the cross-diffusion goes to zero at high concentration. As a general rule, the first scheme has the largest parameter regime, but otherwise the two schemes are for all purposes identical (as will be shown below and in the next chapter).

### 3.1.1 Alternative Model

While the cross-diffusion model given in formula 3.1 is the one mostly used in this thesis, there is a related form that also creates patterns, where we simply change the dependence of the cross-diffusion terms:

$$D_{uv} = \frac{q_u v^m}{K_u^m + v^m} + D_{uv}^\circ, \quad (3.2a)$$

$$D_{vu} = \frac{q_v u^n}{K_v^n + u^n} + D_{vu}^\circ. \quad (3.2b)$$

These models give different looking patterns than the other model, but tend to be far more fickle. All the parameters have to be chosen with great care because the cross diffusion is not controlled by the component being shaped by it; instead the

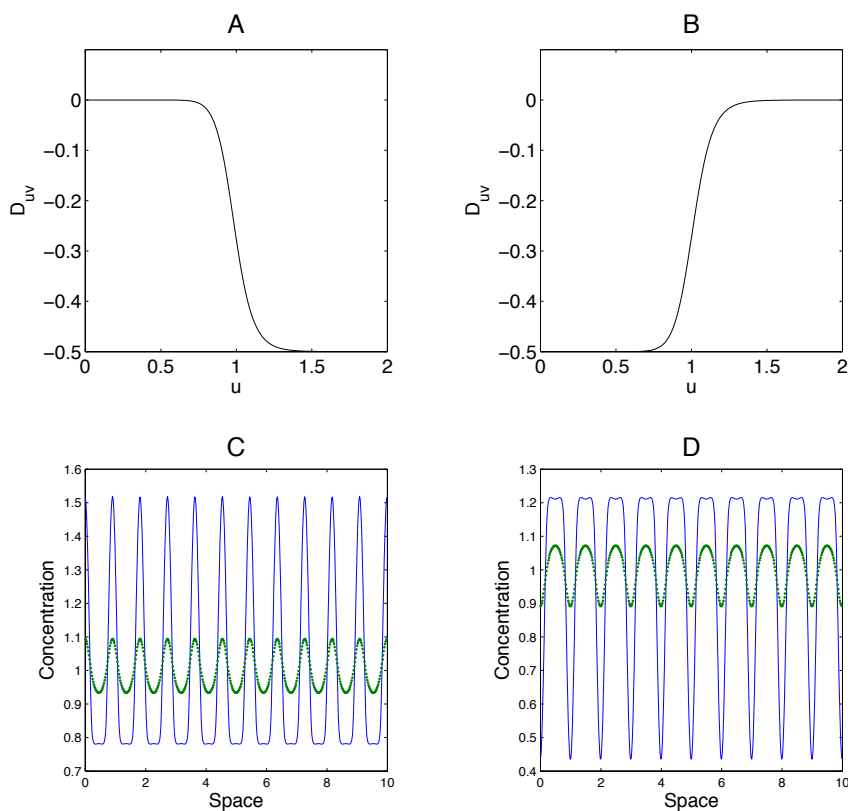


Figure 3.1: Plots of the cross-diffusion strength as a function of  $u$  (A & B) and the resulting patterns (C & D respectively) from numerical simulation. In this figure and all other figures the values for  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\sigma$  are 1 (giving  $u$  and  $v$  both steady-state values of 1). For A and C the parameters used were:  $q_u = -0.5$ ,  $K_u = 0.8$ ,  $D_{uu} = D_{vv} = 0.05$ ,  $D_{uv}^\circ = 0$ ,  $m = 16$ , and for B and D the parameters used were:  $q_u = 0.5$ ,  $K_u = 1.2$ ,  $D_{uu} = D_{vv} = 0.05$ ,  $D_{uv}^\circ = -0.5$ ,  $m = 16$ . The thin line represent  $u$  while the thicker, dotted line represents  $v$ . Note how by inverting the dependence of the cross diffusion on concentration we also invert the final pattern.

control comes from the other component. Since the component not directly affected by cross-diffusion has a much smaller range in concentration (see figure 3.2), the parameters that determine the switching points,  $K_u$  and  $K_v$ , are the most sensitive in this variant model (for example, for the parameters given in figure 3.2 the system can only create patterns for  $0.85 < K_v < 0.9$ ).

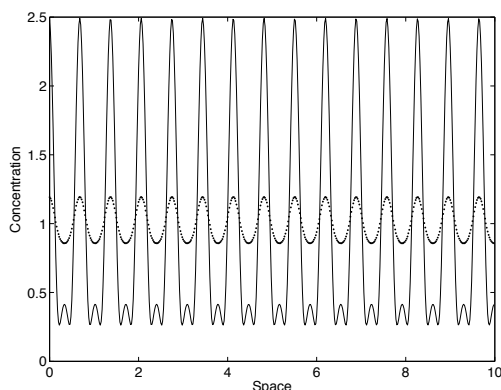


Figure 3.2: A plot of the resulting pattern from equation 3.2, using  $q_v = -0.23$ ,  $K_v = 0.87$ ,  $D_{uu} = 0.05$ ,  $D_{vv} = 0.01$ ,  $D_{vu}^\circ = 0$ ,  $n = 16$ . The dots represent  $u$  whilst the line represent  $v$ . Notice how the plot is very similar to figure 3.1 C.

## 3.2 $\rho$ Model

The  $\rho$  model was an attempt to create a model that would have cross diffusion acting on both components. To do this two new parameters  $\rho$  and  $\chi$  were defined as:

$$\chi = \left( \frac{q_u u^m}{K_u^m + u^m} + D_{uv}^\circ \right) \left( \frac{q_v v^n}{K_v^n + v^n} + D_{vu}^\circ \right), \quad (3.3a)$$

$$D_{uv} = \chi, \quad (3.3b)$$

$$D_{vu} = \rho\chi. \quad (3.3c)$$

Now we have a system that is regulated by both  $u$  and  $v$ , and that has cross-diffusion effects on both components. However when we look at equations 2.32 and 2.33 we see that we if  $\rho$  is too large the system will enter an entirely different regime of infinite wavenumbers, which causes the numerical methods we use to fail. Thus the values of  $\rho$  used are typically very small, in the range of  $\rho < 0.02$ . While this model creates patterns that are identical to the other models, one advantage of this model is that system can be more closely regulated. This is a result of being able to put a lower bound and an upper bound on the range of concentrations for which the cross-diffusion is active, thus even if the cross diffusion is overly strong, it is scaled down at either extreme.

### 3.3 Numerical Simulation

In the previous two sections, various numerical results have been shown and referred to. The results were generated using C++ programs. These programs are finite difference programs that model these system as a series of points. To iterate through time, the simplest technique is used, explicit Euler. In the appendix we go through and present a formal derivation of this technique (and explain why we use it) but suffice to say that it only does first-order integration (which is the norm for reaction-diffusion systems) and is very efficient. In addition to finite difference methods, a finite element program<sup>37</sup> integrated with an implicit Euler algorithm using adaptive step sizes<sup>38</sup> was also used as a check our results. See figure 4.6 to see how they replicate each other.

### 3.4 Comparison of Numerical Results to Bifurcation Analysis

In the previous chapter we had two different formulas (2.31 and 2.42) that we could easily plug numbers into and determine a range of values for our wavenumber (inverse of wavelength) for which the homogeneous steady state is destabilized. Thus we can determine the steady state values of  $D_{uv}$ ,  $D_{vu}$ ,  $D_{uu}$ , and  $D_{vv}$ , for our common parameter set:  $q_u = -0.5$ ,  $K_u = 0.8$ ,  $D_{uu} = D_{vv} = 0.05$ ,  $D_{uv}^o = 0$ ,  $m = 16$ ,  $\alpha = 1$ ,  $\beta = 1$ ,  $\gamma = 1$ , and  $\sigma = 1$ . By equation 2.9 we know the steady-state concentrations of both  $u$  and  $v$ , then by equation 3.1 we find  $D_{uv}^* = -0.4363$ , and  $D_{vu}^* = D_{uu}^* = 0.05$ .

By formula 2.31, we obtain:

$$\lambda = -2 - 0.05k^2 \pm \frac{\sqrt{4 + 1.944k^2}}{2}. \quad (3.4)$$

Referring to figure 3.3 B, we see that there is a range of wavenumbers for which the steady-state is unstable. Specifically our analysis<sup>iii</sup> shows an unstable range from

---

<sup>iii</sup>During the course of the analysis, the value  $k$  was referred to as the wavenumber, as it is in all the literature. However, the value actually corresponds to the number of radians per unit length,  $2\pi k$ , so the all the values given and the values plotted in figure 3.3 are rescaled to reflect the actual wavenumber.

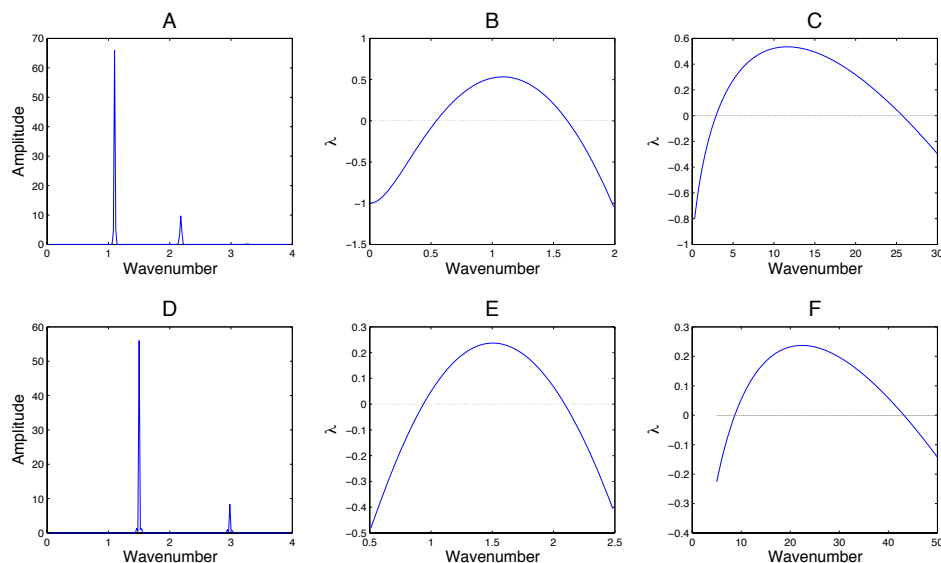


Figure 3.3: A comparison of numerical results vs analytical results. Plots A through C use the parameter regime of figure 3.1 A, while D through F use  $q_u = 0.23$ ,  $K_u = 1.25$ ,  $D_{uu} = 0.01$ ,  $D_{vv} = 0.05$ ,  $D_{uv}^\circ = -0.23$ ,  $m = 16$ . The leading plots (A and D) show the result of a Fourier transform on the final pattern from numerical simulation). B and E show the range of  $k$  for which an instability should arise from Turing bifurcation analysis, while C and F show the same by means of the two cell model. While the latter analytic technique does not agree with numerical simulation (see text), both it and the numerical results increase in wavenumber for the second set of parameters.

$k = 0.54$  to  $1.61$ , with a maximum at  $k = 1.08$ . While the range of  $k$  is important<sup>iv</sup>, we are actually more interested in the value with the largest  $\lambda$ . This will be the dominating wavenumber during the initial Turing instability.

Switching to the two cell model with the same parameters, and formula 2.42, we get:

$$\lambda = -\frac{0.10}{\Delta z} - 2 + \frac{\sqrt{3.89049\Delta z + 4(\Delta z)^2}}{2\Delta z}. \quad (3.5)$$

Now the stability of the system is determined solely by the value of  $\Delta z$ , but to compare it to our previous values, we need it in terms of the wavenumber ( $k$ ), by the

<sup>iv</sup>In systems where the size is too small, as long as there is sufficient space for the smallest unstable wavelength, a Turing instability will occur, but below this size no permanent patterns will form.

conversion  $k = \frac{1}{2\Delta z}$ .<sup>v</sup> then take the inverse to get the wavenumber:

$$\lambda = 2k \left( -0.10 - \frac{1}{k} + \frac{1}{2} \sqrt{\frac{1.9452}{k} + \frac{1}{k^2}} \right). \quad (3.6)$$

Again referring to figure 3.3 C, we have a range of wavenumbers for which the system becomes unstable, specifically, from  $k = 2.92$  to  $25.71$ , with a maximum at  $k = 11.64$ .

The result from numerical simulation is a wavenumber maximum at  $k = 1.1$ , while Turing bifurcation analysis predicts a value of 1.08. This shows a remarkable correspondence between numerical results and the standard analytical technique. And while the behavior of the kinetic terms away from the steady state are well behaved, the cross-diffusion coefficients most definitely are not (see figure 3.1), showing how strongly the initial instability drives the pattern formation. The two-cell model is inherently quite limited in its predictive ability. In addition to only predicting the wavenumber at the onset of pattern formation, we are assuming that the behavior of the system can be described with only two cells<sup>vi</sup>. However, one can see in figure 3.3 that the two-cell model does predict an increase in wavenumber for the second set of parameters which is mirrored in the numerical results.

---

<sup>v</sup>The logic behind the  $\Delta z$  to wavelength conversion is simple. If one imagines a sine wave, the distance between the upper and lower peaks would be equivalent to  $\Delta z$ , but the wavelength of the actual peaks is twice this.

<sup>vi</sup>Ideally one would solve the system for the new steady-state values and determine the best value of  $\Delta z$ . However, the sheer complexity of the cross-diffusion coefficients makes this calculation impossible, even with the aid of computer software.

## Chapter 4

### Tests on the model

Having created a model that forms patterns we now want to see what it can do and just how well it works. If the system cannot form patterns with the introduction of perturbations or noise it would not be very robust, and would prove either that the system is flawed, or that the numerical techniques used are in error. In addition, testing the system also gives us some idea on how we might expect this type of system to behave in the real world, and some information on actually implementing it.

In addition to just adding perturbations and noise, there is one more test, adding dimensions. This does many things for us. It tells us just what patterns our system can exhibit, for even in two dimensions one can get dots, hexagons, stripes, labyrinths, and spiral patterns<sup>39,40</sup> with Turing patterns. And while the planar system is a great model for how patterns would form on a surface (say a thin gel or the skin of an animal), three-dimensional simulations tell us the behavior when we go into realistic simulations (i.e. not infinitely thin space).

#### 4.1 Perturbations and Fluxes

The simplest test is to just add a perturbation to the system, where a perturbation is a small “push” to the system. These test the system’s ability to absorb disturbances, for it is rare for systems to be perfectly isolated and regulated. Some systems<sup>41</sup> change their behavior in a major way with the addition of perturbations (like changes in the final pattern and changes in the wavelength), while in other systems these have no discernible or important effect.

There are a variety of ways in which we can perturb our system. The simplest is just a single pulse into the system over a given region of space (figure 4.1). As one can see, the system is quite resilient to a single pulse be it just a single nonspecific pulse, or directed pulses with the aim of starting a new peak, or disturbing a current one. While the pulses shown here are limited in size, larger pulses have no effect, except that the system takes more time to recover the original pattern.

Similarly multiple pulses either have no effect (the time in between being sufficient for the system to recover), or behave similar to continuous flux. If a continuous flux



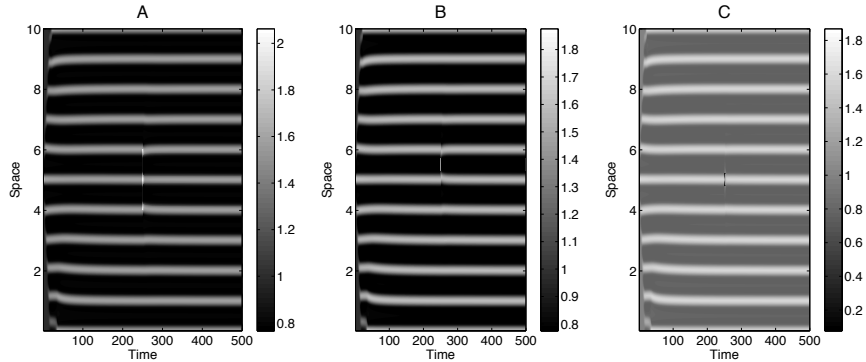


Figure 4.1: A plot (in  $v$ ) of the effect from various perturbations on the system. The parameters for all three plots are  $q_v = -0.5$ ,  $K_v = 0.8$ ,  $D_{uu} = D_{vv} = 0.05$ ,  $D_{vu}^\circ = 0$ ,  $n = 16$ . In panel A a single perturbation was introduced at  $t = 250$  of  $0.5 v$  per unit length over a length of 2. For panel B, the system is hit with a pulse 0.4 wide of  $0.7 v$  per unit length, directly on the gap between the two peaks (0.5 off of center). The final panel, C, has a single negative pulse centered directly on a peak, 0.4 wide of  $-0.8 v$  per unit length. In all three panels we see that that a single pulse does not affect the final pattern in any way.

is added to the system, it can rearrange the final pattern, as show in figure 4.2

Adding a continuous flux into or out of the sides has an almost identical effect as a flux into a point in space. The only difference is that the distortion is more pronounced. This makes sense, since the diffusion mechanisms which would normally help contain the flux can only aid from one side, thus magnifying the effect of the pulses. Similarly, turning the system into a ring has no real effect. While we had initially thought that continuous pulses may be able to produce rotating waves, the system's ability to absorb perturbations makes it able to withstand perturbations locally.

## 4.2 Noise

The addition of noise to a system is a standard approach in nonlinear dynamics, and it is actually very important that we do it. First of all, many of the processes studied in science are not just smooth deterministic occurrences, but rather a series of

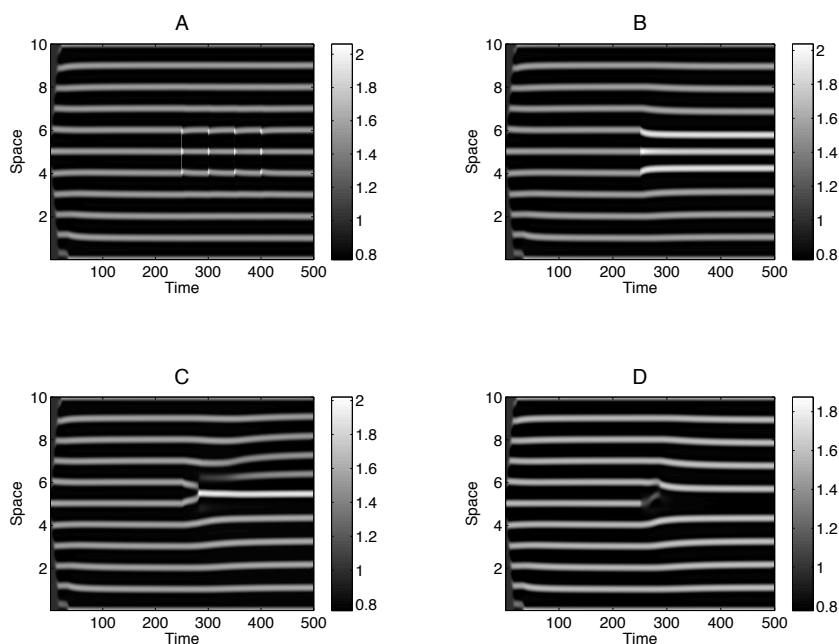


Figure 4.2: A plot (in  $v$ ) of various perturbations on our system of various types. The parameters for all four plots are  $q_v = -0.5$ ,  $K_v = 0.8$ ,  $D_{uu} = D_{vv} = 0.05$ ,  $D_{vu}^\circ = 0$ ,  $n = 16$ . In the first panel, A, a series of single perturbations of  $0.5 v$  per unit length over a length of 2 are introduced. Panel B has a continuous pulse 2 wide of  $0.2 v$  per unit length per time, while in C a smaller pulse 0.4 wide of the same intensity is introduced between two peaks. The final panel, D, has a negative pulse of  $-0.8 v$  per unit length per time introduced, 0.4 wide, centered directly on a peak. One can clearly see that a continuous pulse into (or out of) the system can alter the local pattern, but not the global behavior.

stochastic events that are simply smoothed out by large numbers<sup>i</sup>. In addition most things that we take to be a constant are not. Consider for example, the temperature of solvent in a beaker. While the average is at room temperature, local temperature fluctuations can occur quite readily, and would have the effect of changing various parameters randomly through space and time.

<sup>i</sup>Consider for example a simple chemical reaction of  $A + B$  forming  $C$ . If we were able to follow the evolution of the reaction (say by detecting an NMR signal) we would see a smooth production of  $C$ . However the reality is that each molecule of  $C$  is formed by the collision of one molecule of  $A$  and one molecule of  $B$ . It's just that for even a one molar solution of  $A$  and  $B$  there are  $6.0221 \times 10^{23}$  molecules of each component.

For the actual application of noise to a system, there are three main techniques that can be used. If the number of chemical species is very small, then one would apply a stochastic algorithm.<sup>42</sup> This is where each individual molecule/component is modeled separately and each reaction is based on probabilities coupled to a random number generator. However, given that our system is based on far larger numbers, we instead forgo stochastic simulation, and instead use a second technique, adding noise to each point. We do this by randomly adding or subtracting a small amount to the rate of change of each component at each time step. The third and final technique is to add noise to one or more parameters. For example, one could have a cross-diffusion coefficient that varies slightly with each time step.

The algorithm used to create the noise is actually made of two components. First the Marsaglia Zaman (MZ)<sup>43</sup> random number generator is used to create random numbers over the interval  $(0, 1)$ .<sup>ii</sup> The second component is the Box-Muller algorithm<sup>44</sup> which takes two random numbers from the unit interval, and transforms them to two normal deviates<sup>iii</sup>. Thus our random numbers have a mean of zero and a standard deviation of one.

### 4.2.1 Noise in Space

When we add noise to a point during a timestep, we cannot just add the random noise as is. We have to multiply the noise by a coefficient  $A$  to scale the noise down (effectively changing the standard deviation from one to  $A$ ), for adding or subtracting noise with a standard deviation of one when the equilibrium values are typically one will lead to the noise being really the only factor in the system. In addition, our  $A$  term has to be normalized for the discretization of our system. We do this by<sup>45, 46iv</sup>:

$$\eta = \frac{A}{(dz)^{d/2} \sqrt{dt}}, \quad (4.1)$$

---

<sup>ii</sup>This random number generator is very efficient using only subtraction operators and has an estimated period of  $2^{1407}$ . It actually combines two random number generators so as to avoid any potential flaws with any single algorithm.

<sup>iii</sup>The algorithm itself is really quite simple. Given the two initial random numbers  $U_1$  and  $U_2$  from the interval  $(0, 1)$ , we get our new values,  $X_1$  and  $X_2$  over a normal distribution by  $X_1 = \sqrt{-2 \ln U_1} \cos(2\pi U_2)$  and  $X_2 = \sqrt{-2 \ln U_1} \sin(2\pi U_2)$ .

<sup>iv</sup>Personal communication with Dr. Mikko Karttunen.

where  $A$  would be the idealized amplitude of the noise introduced,  $d$  the number of spatial dimensions, and  $\eta$  the actual value we multiply the Gaussian noise by. The time discretization results in a random walk over time. The average displacement from the start point as a function of time increases as the square root of time, see figure 4.3, thus the longer the timestep, the more averaged out the noise becomes. Similarly, the effect of noise over a larger interval of  $z$  would actually become averaged out, whereas over a smaller space, the noise would become larger. So in a one dimensional simulation with  $dt = 0.001$  and  $dz = 0.02$ , the amplitude of the noise reported could be  $A = 0.0001$ , whereas the actual value used would be  $\eta = 0.0223$ .

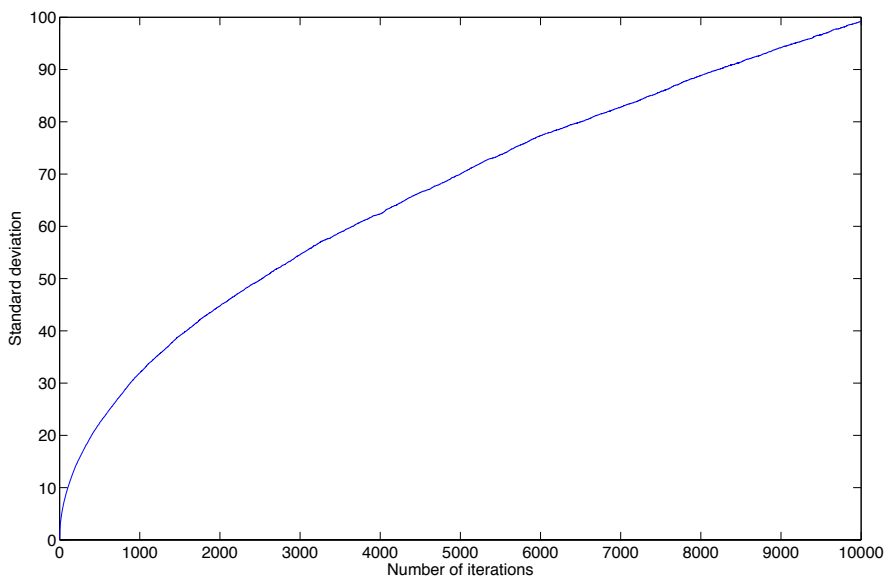


Figure 4.3: A simple plot where a random walk algorithm is run 10000 times for 10000 iterations each time. For each iteration the system randomly takes one step forward or one step back. As one can see, the standard deviation increases with a square root dependence on the number of iterations.

As one can see in figure 4.4, the addition of noise to the rates of change in  $u$  and  $v$  over space has no real effect on the final pattern, but dramatically increases the speed with which a pattern is formed over the entire space. This matches the results of experimental papers<sup>6,47</sup> where they are not initializing pattern formation, but rather the system spontaneously forms patterns on its own due to the inherent

noise of the real world.

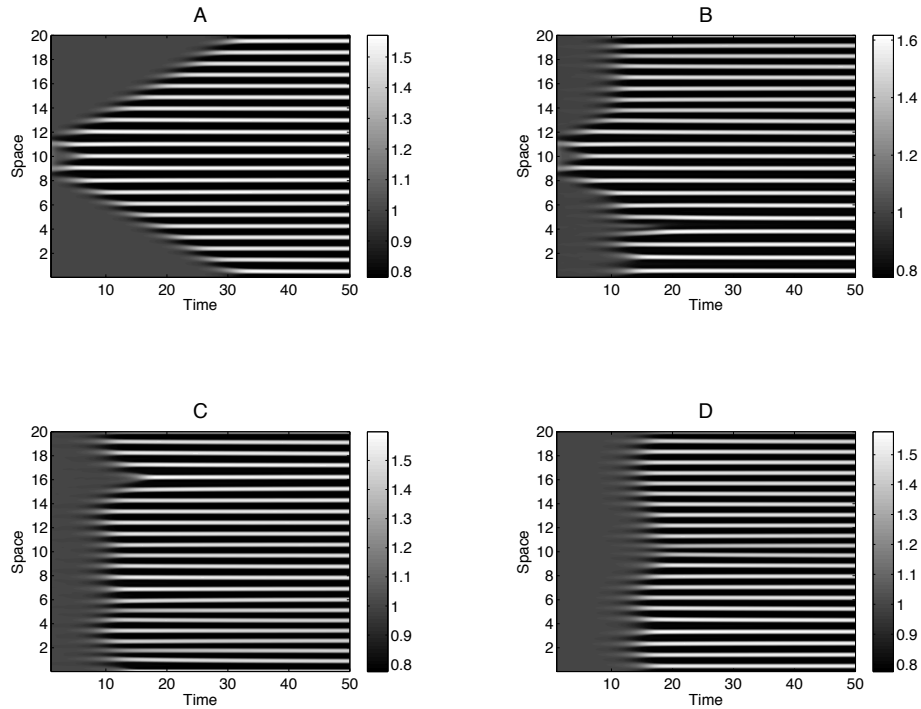


Figure 4.4: A plot (in  $u$ ) comparing the effect that noise has on the system. The parameters for all four plots are  $q_u = -0.5$ ,  $K_u = 0.8$ ,  $D_{uu} = D_{vv} = 0.05$ ,  $D_{uv}^\circ = 0$ ,  $m = 16$ . In panel A there is no noise and just a single perturbation at the beginning of the simulation, while in B we have the same perturbation but add noise with a prenormalized amplitude of  $1.41 \times 10^{-5}$ . For C we remove the initial perturbation and in D we reduce the noise further to  $1.41 \times 10^{-6}$ .

## 4.2.2 Noise on Parameters

The addition of the noise to parameters is the simplest type of noise to implement. Given a parameter  $\omega$ , we would simply use  $\omega = \omega'(1 + \eta N)$ , where  $N$  is the noise term and  $\omega'$  is the original value. Again we use very small amplitude noise to prevent parameters completely changing, as we just want subtle changes in values. The value for the parameter being changed is adjusted each timestep for each location.

In general, the addition of noise to parameters had very little effect, with the largest effect coming from applying the noise to the entire cross diffusion term  $D_{uv}$  rather than to any individual parameter. In these cases the system would become unstable with the application of noise that was too large ( $A > 0.005$ ), but if the parameter range was just outside where pattern formation would occur, the addition of noise would promote pattern formation, but only within certain ranges of  $A$ .

### 4.3 Higher Dimensions

In terms of computation time, each dimension we add to our simulations drastically slows down simulations<sup>v</sup>, but at the same time we must carry out these calculations, for we do not inhabit flatland.<sup>48</sup>

Two-dimensional simulations are the most common simulations to run simply because they best represent the current experiments, and the main inspiration for Turing patterns is natural pattern formation which is typically on the skin or over a surface. As one can see by figure 4.5, our system creates spot type patterns, and similarly to the one dimensional case (figure 3.1), we can either have a series of wells below a plane, or a series of peaks above a plane, figure 4.6. While one can see that the perturbation alone takes some time to create a pattern, the addition of noise in figure 4.5 greatly decreases the time to create the final pattern. This effect of course increases as the system size increases, since it takes longer for the first front to reach the edges of the simulation. We can also see that our modeling methods are accurate by the fact that simulations using two completely different techniques give virtually identical results.

Three-dimensional simulations at first do not seem to make much physical sense. After all, does the chemical reaction scheme not have terms for both the addition and removal of both components regardless of location? The answer is yes, but these reaction steps can still occur in a three-dimensional medium. The influx terms would

---

<sup>v</sup>This is a result of the number of points required and the connections between the points. For an equal density of points, a line 1 long will require 50 points say, with each point having two connections (and thus two calculations), whereas a 1 by 1 sheet will then require 250 points with 4 to 8 connections. The worst case is a 1 by 1 by 1 cube, in which 125000 points are needed, with 6 to 25 connections.

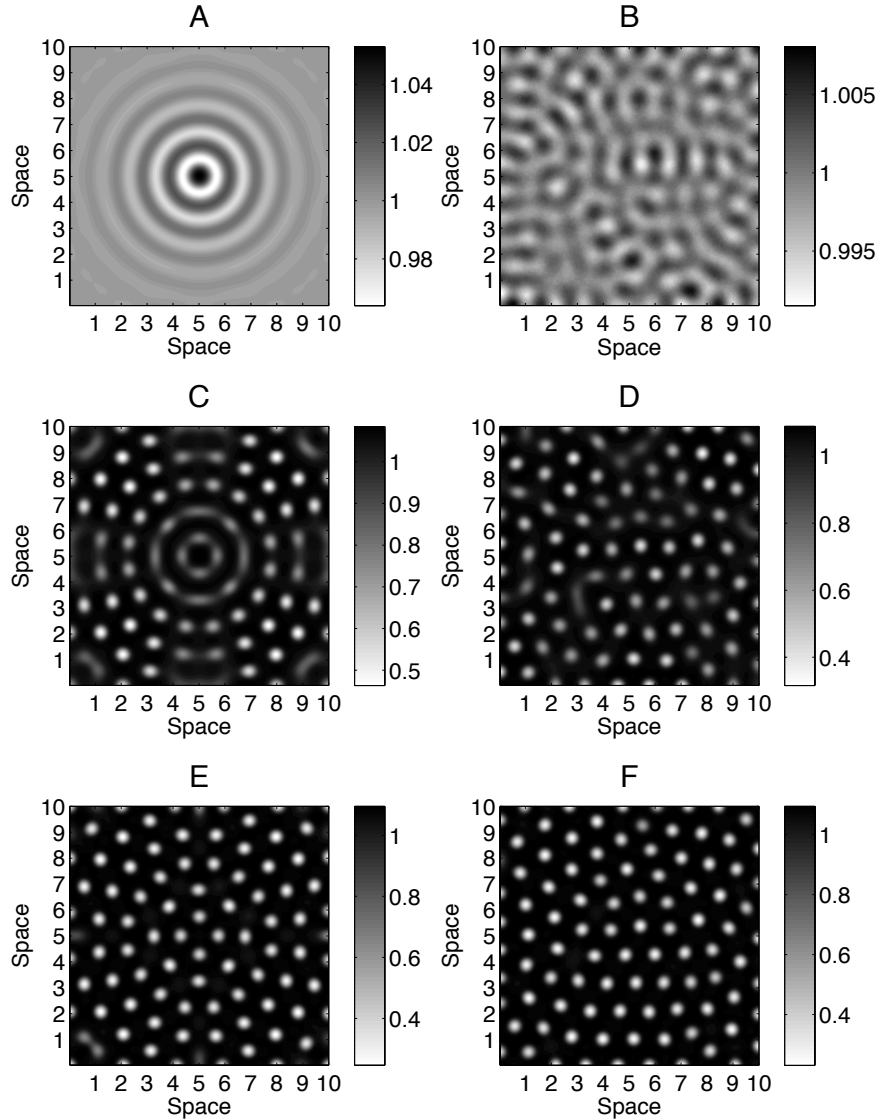


Figure 4.5: Two separate runs of the system are compared to each other. Both are using the parameters  $q_u = 0.45$ ,  $K_u = 1.12$ ,  $D_{uu} = D_{vv} = 0.05$ ,  $D_{uv}^\circ = -0.45$ , and  $m = 16$  (which leads to an inverted pattern compared to figure 4.6). The right column is with only noise with a prenormalized amplitude of  $1.5 \times 10^{-4}$  whilst the left column is without noise. Going from top to bottom, the time shots are at 50, 110, and 140. As one can see, the noise speeds up pattern formation.

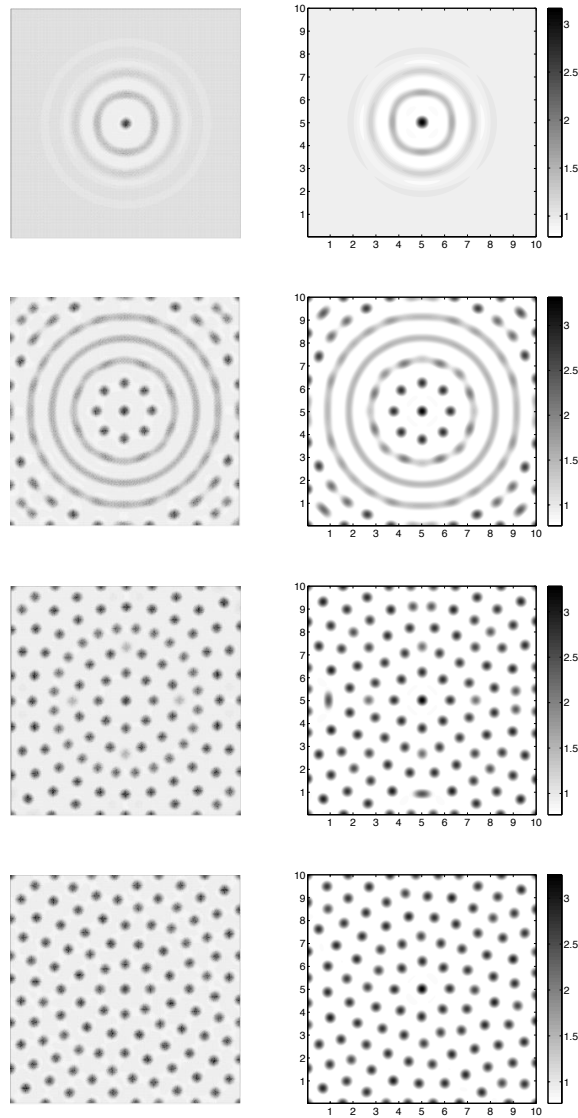
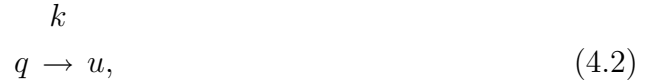


Figure 4.6: Again two separate runs of the system are shown here. Both are using  $q_u = -0.5$ ,  $K_u = 0.8$ ,  $D_{uu} = D_{vv} = 0.05$ ,  $D_{uv}^\circ = 0$ , and  $m = 16$  in a  $10 \times 10$  unit simulation space. The left column is run in FreeFem++<sup>37</sup> while the right is run in C++ with a nine point mesh (see appendix for details). The time shots are at times 10, 30, 50, and 1000 going top to bottom.



actually be the result of the slow degradation of another system component that is highly concentrated. Thus while the actual reaction would be:



if  $q$  is very large, the rate of  $u$  formation would essentially be a constant,  $kq$ . Similarly, the efflux term can be a similar degradation of  $u$  and  $v$ , where the reverse reaction is negligible.

Having rationalized three-dimensional simulations of this model, what is the result? The system invariably produces spheres, and again in the same way that the shapes are produced in lower dimensions we either have spheres of high concentration or spheres of low concentration, of which we have the former case in figure 4.7.

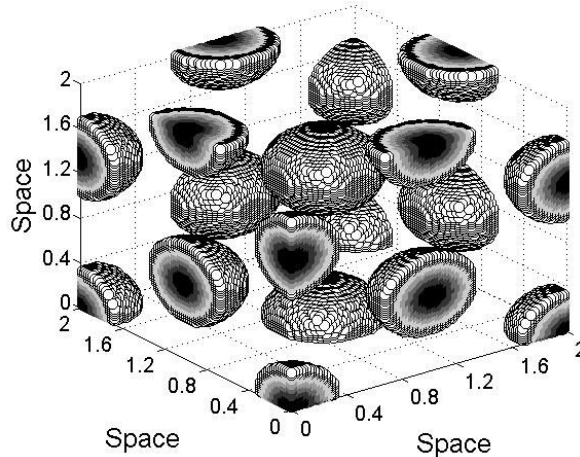


Figure 4.7: A plot of the system in three dimensions for component  $u$ . The color intensity runs from white being a concentration of at least 1 to black being a value of up to 6.5, with all the open spaces having values outside this range. The parameters used were  $q_u = -0.45$ ,  $K_u = 0.8$ ,  $D_{uu} = D_{vv} = 0.05$ ,  $D_{uv}^{\circ} = 0$ ,  $m = 16$ , on a  $100 \times 100 \times 100$  grid.

## Chapter 5

### Closed Model

The closed model is an additional attempt on our part to expand the realm of Turing patterns. In a nutshell, the closed model has the exact same diffusion system as our regular model (equation 2.1), but the chemical reactions are simplified (by simply setting the influx and efflux terms to zero) to:



which in turn gives the rate equations (after nondimensionalization):

$$\frac{\partial u}{\partial t} = \alpha v - u, \quad (5.2a)$$

$$\frac{\partial v}{\partial t} = u - \alpha v. \quad (5.2b)$$

This system is unlike any other used for Turing patterns, with the closest being the temporary patterns in water drops<sup>47i</sup>.

This chapter will then read like a miniature version of the previous chapters, just on a different model. To begin, a bifurcation analysis is carried out, both to show the stability of the reaction system, and the conditions for the onset of a Turing instability. With the conditions required established, we then show the results of numerical simulation.

### 5.1 Linear Stability Analysis

Similar to section 2.2 we carry out a linear stability analysis of our new reaction system, however this time we only cover the main points, since the reader can refer back to the previous iteration for the finer points of this method.

---

<sup>i</sup>The experiment is novel enough that it is worth briefly recounting here. An oil/water mixture is made with soap-like additives added shielding the water drops from the oil, but the drops themselves can contact each other and form water bridges where diffusion can occur. By preloading the system with the reactants for the BZ reaction,<sup>49</sup> the system forms patterns.

We begin by determining the steady-state values for our system. Solving  $\frac{\partial u}{\partial t} = 0$  and  $\frac{\partial v}{\partial t} = 0$ , we get:

$$u = \alpha v. \tag{5.3}$$

This means that steady state of the system will depend on how much  $u$  and  $v$  we start with (which, considering the closed nature of the system, makes perfect sense). We next determine the Jacobian matrix:

$$\mathbf{J} = \begin{bmatrix} -1 & \alpha \\ 1 & -\alpha \end{bmatrix}. \tag{5.4}$$

After solving  $|\mathbf{J} - \lambda\mathbf{I}| = 0$ , the eigenvalue equation is:

$$\lambda^2 + (\alpha + 1)\lambda, \tag{5.5}$$

which has two solutions,  $\lambda = 0$ , and  $\lambda = -\alpha - 1$ . While the latter is obviously always negative, the first eigenvalue really does not give us any information on the behavior of the system. To solve this we could apply manifold theory and analytically solve the system, or take the far simpler approach and do a quick phase plane analysis, figure 5.1. We can see quite clearly that regardless of where the system starts, it reaches a steady state. However, we can also see that unless one perturbs the system in such a way so as to maintain the balance between  $u$  and  $v$ , the steady state will shift. Thus the system will always return to the line of equilibria, but not always to the same point on that line.

## 5.2 Turing Bifurcation Analysis

Having determined that the reaction system alone is stable, we next need to gauge the effects that diffusion and cross diffusion have. Again, only the main points are touched on since this technique was covered in section 2.3. Starting with our linearized reaction-diffusion equation:

$$\frac{\partial(\delta u)}{\partial t} = D_{uu}^* \frac{\partial^2(\delta u)}{\partial z^2} + D_{uv}^* \frac{\partial^2(\delta v)}{\partial z^2} + \alpha \delta v - \delta u, \tag{5.6a}$$

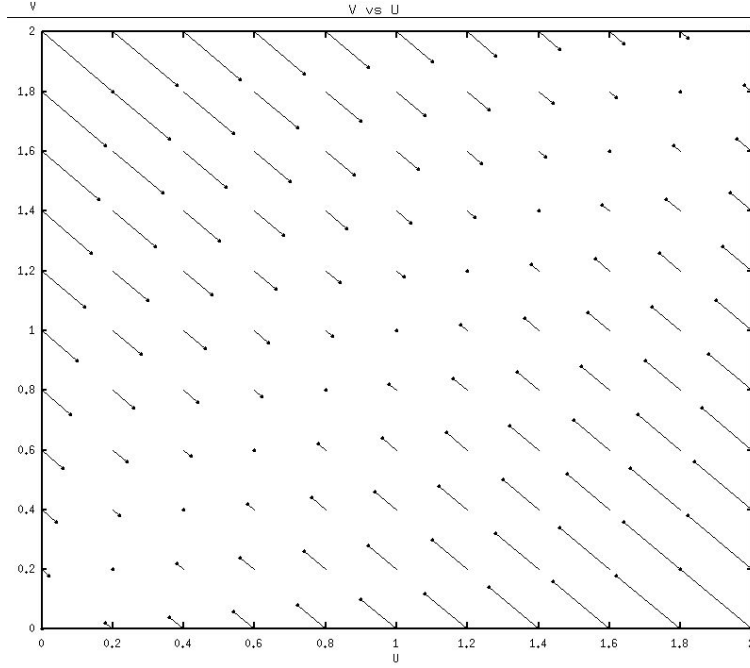


Figure 5.1: Phase plane diagram for the system described in equation 5.2, with a value of  $\alpha = 1$ . The nullcline is not shown so that one can see the actual equilibrium points along the diagonal. Note how the steady-state is dependent on where the system starts, so that we actually have a line of equilibria, where any point is itself is not stable, while as a whole, the line is. The actual plot itself is from XPPAUT<sup>30</sup> which uses different sized arrows to represent the magnitude of  $\dot{u}$  and  $\dot{v}$  at each point.

$$\frac{\partial(\delta v)}{\partial t} = D_{vv}^* \frac{\partial^2(\delta v)}{\partial z^2} + D_{vu}^* \frac{\partial^2(\delta u)}{\partial z^2} + \delta u - \alpha \delta v, \quad (5.6b)$$

we can then solve  $|\mathbf{J} - k^2 \mathbf{D} - \lambda \mathbf{I}| = 0$  again, this time with our new Jacobian, to find:

$$(\lambda + D_{uu}^* k^2 + 1)(\lambda + D_{vv}^* k^2 + \alpha) - (D_{uv}^* k^2 - \alpha)(D_{vu}^* k^2 - 1) = 0. \quad (5.7)$$

$$\therefore \lambda = \frac{1}{2}(-B \pm \sqrt{B^2 - 4C}), \quad (5.8a)$$

$$B = k^2(D_{vv}^* + D_{uu}^*) + 1 + \alpha, \quad (5.8b)$$

$$C = k^4(D_{uu}^* D_{vv}^* - D_{uv}^* D_{vu}^*) + k^2(D_{uv}^* + D_{vu}^* + \alpha D_{vu}^* + \alpha D_{uu}^*). \quad (5.8c)$$

As we can see we need almost the exact same two conditions to make  $\lambda$  negative<sup>ii</sup>. The first condition is exactly the same under high  $k$  with:

$$D_{uv}^* D_{vu}^* > D_{uu}^* D_{vv}^*. \quad (5.9)$$

Of course we try and avoid that regime and instead focus on:

$$D_{uv}^* + D_{vv}^* + \alpha D_{vu}^* + \alpha D_{uu}^* < 0, \quad (5.10)$$

which we can easily reach by having negative cross-diffusion coefficients. Also, compared to the conditions required for the previous system (equation 2.33), this system does not need nearly such negative cross diffusion coefficients to accomplish instability (which is logical considering the analysis above shows a less stable reaction system). One can also see that by setting all the diffusion and coefficients to zero we return to having one eigenvalue equaling zero, thus any diffusion we add only makes the system more stable, whilst any negative cross-diffusion added to the system only destabilizes the steady state.

### 5.3 Numerical Results

The actual implementation of this system is not quite as fully developed as the main system used. Rather this system has only been amenable to one-dimensional simulation, as shown in figure 5.2. In the same figure we can see a perfect example of why testing our numerical methods is so important. The first panel shows how this model without noise will lead to only two peaks being formed. Adding very small amounts of noise leads to the system producing multiple peaks, which leads to the next characteristic of this model: uneven peaks and the lack of a single wavelength. While the simulations in a single dimension have proven to be stable, in higher dimensions peaks tend to keep joining together and eventually coalesce into a single peak, followed by the system creating negative concentrations. Even in a single dimension simulations have been far from simple to run, with the parameters used the

---

<sup>ii</sup>This of course only makes sense, since our overall system is essentially just a reduction of our previous system.

figure actually having a predicted wavenumber of zero from the bifurcation analysis.

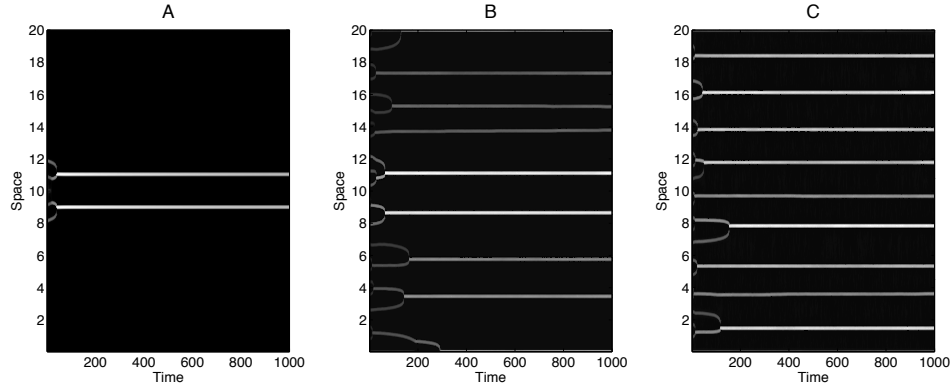


Figure 5.2: Three separate plots of  $u$  in the closed model with  $\alpha = 1$ ,  $q_u = -0.6$ ,  $K_u = 1.2$ ,  $D_{uu} = D_{vv} = 0.05$ ,  $D_{uv}^\circ = 0$ ,  $m = 16$ , over a length of 20 with 1000 mesh points. In panel A, we have only a single perturbation with no noise, where we can see that the system creates two peaks at the edge of the initial perturbation of  $0.5 u$  over a space of 5 units, The general range in concentration is from 1.01 to 3.25. The addition of noise, with the same perturbation in panel B, show that the previous panel is in fact a result of this system not being entirely robust, where the concentration ranges from 0.4 to 10.1. We can say this because the noise is very small, with a prenormalized amplitude of  $1.5 \times 10^{-4}$ , and yet the effect is quite profound. Removing the perturbation altogether in panel C, we see that pattern formation still occurs but in both cases the pattern is not the typical fixed wavelength pattern, but is instead quite irregular. In the last panel, the concentration ranges from 0.4 to 6.0.

While this method is not at all well behaved (numerically), these results still show much promise. This is simply because to the author's knowledge, there have been no methods to date (either proposed or experimental) that have sustained patterns with zero external fluxes.

## Chapter 6

### Conclusion

Turing patterns have been with us for over 50 years, and the field seems to be showing no signs of slowing down any time soon. At the very least, the systems proposed here will be novel additions to the vast repertoire of systems proposed for Turing patterns, but hopefully people will realize that while yes, there is already 50 years of work done in this field, the vast majority of it has focused on the reaction aspect of what is a reaction-diffusion based system. While the systems presented here will most likely not be directly applicable, they show that we can create patterns by using the simplest chemical reactions possible and instead look at the diffusion and cross-diffusion terms. So while the model presented here oversimplifies the chemical reactions, one could argue that most of the work till now oversimplifies the diffusion. Thus one would hope that eventually there will come models that will use experimental data on the diffusion rates (fully modeled, not just a single constant coefficient) for the species involved and be able to use the nonlinearities inherent to both the diffusive process and the kinetics to realistically model systems.

On the biological side, one could see a form similar to the cross diffusion used in this model in a cellular model where transporter proteins are involved (especially the variant detailed in section 3.1.1 where cross-diffusion depends on its own levels) in transporting a component against its own gradient. Even in cases where nonlinear cross-diffusion may not be the best model, many systems have diffusion as a minor component of transport, thus opening the possibility of emulating cellular transport or chemotaxis<sup>50-52</sup> processes with nonlinear diffusion.

Within the realm of the systems presented here, there are still many avenues of investigation available. With the primary model, introducing minor nonlinear terms to the reaction scheme would almost certainly lead to two things, different patterns, and the ability to use a smoother function for the cross diffusion. Furthermore, one could start looking at modeling known reactions, especially those with charged ions.

Another interesting area of research, which has not really been pursued in the study of Turing patterns, is the high  $k$  regime. While these systems may seem quite unrealistic (taken at face value, the condition 2.32 would give infinitely small wavelength by the bifurcation analysis), and given that our system behaves quite

differently away from the steady state, other effects could factor into the system's evolution. Of course, it could simply be the case that all that can occur is a Turing instability which could result in a globally unstable system. Any results in this area would be significant because this represents an entirely new area to Turing patterns, though these small wavelength systems have been previously investigated in backward diffusion systems.<sup>32</sup>

Of course the most interesting area is the closed model. This model, while not being nearly as well behaved as the main system, has the ability to really prove what nonlinear cross diffusion can do and shows that not only can linear systems make patterns but that even closed systems can form patterns. Again, the addition of nonlinear terms to the system (and perhaps the addition of more components to the system) would most likely help stabilize the system.

The models shown in this thesis, and the suggestions for furthering them will not be the mechanism by which the next gel reactor will be run nor will it be referenced 20 years from now as THE model for how a zebra forms its stripes. Rather, the purpose of these models is to show that Turing patterns can be found in areas that have been ignored for some time, and that a shift in the focus of modelers and experimentalists from the reaction mechanisms to the diffusion could make a marked difference.



# Chapter 7

## Appendix

### 7.1 Numerical Methods

Numerical methods are what we use when we have a mathematical problem that cannot be solved analytically. Many problems in many fields fall under this umbrella, such as the three-body problem in physics, multistep and/or multiple component chemical reactions in chemistry, ecological modeling in biology and of course, reaction-diffusion problems. The problem with all these systems is that while the system behavior is defined by a (sometimes very large) set of very exacting rules, there are no mathematical techniques that will give an analytical solution for the state of these systems, with the exception of some very well behaved and limited situations.

So where analytic methods fail, numerical methods step in. While finding the analytical solution to a reaction-diffusion system is generally not possible, we can approximate it using one of many different numerical techniques. These techniques are based on discretizing time and space in some form that is then usable. While two techniques were used here, there is an ever expanding library of techniques, each having advantages and disadvantages.

#### 7.1.1 Finite Difference

Finite difference is the technique predominantly used in the field of reaction-diffusion systems, and was the primary technique used in this research. The key to finite difference is that we treat space as a series of discrete points in space rather than a continuous plane. In a one-dimensional simulation then, each point ( $i$ ) has two neighbors ( $i + 1$  and  $i - 1$ ). Additionally, each point is related in time to the points before and after by a gap in time  $dt$  (see figure 7.1).

It is from these two neighbors that the diffusion is determined, by first determining the flux ( $J$ ) between each point. Assuming only diffusion so that the example is clearer (where  $dz$  is the space between the points),

$$J_i = -\frac{D(x_{i+1} - x_i)}{dz}, \quad (7.1)$$

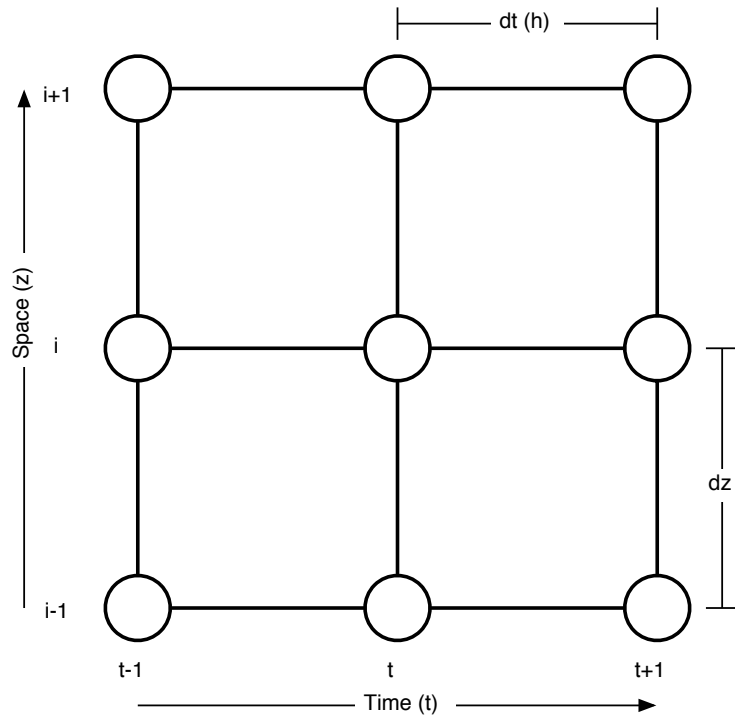


Figure 7.1: A schematic of how the one dimensional systems are run. Each point in space is discrete, and separated by a gap  $dz$ . Similarly, moving through time, each time point is separated by a time  $dt$  (which is labeled  $h$  in the derivations).

$$\frac{dx_i}{dt} = -\frac{J_i - J_{i-1}}{dz}. \tag{7.2}$$

As one can see this almost exactly mirrors our derivation of the diffusion terms in equations 2.18 and 2.17, except that now we have discrete values for our derivatives.

We now have means of connecting our points in space. How do we do this through time? Well, it is actually quite similar: one point at a time. This is done with what is called the explicit Euler method.<sup>38</sup> We can derive this method very easily via a Taylor series. Suppose we have a system that evolves through time as a function of  $u$ , given in the form of a differential equation  $\frac{du}{dt} = f(u)$ , where the state of a system at any point in time is  $u(t)$ , and we are interested in the behavior of the system at

the point  $t + h$ . To determine this we take a Taylor series of  $u(t + h)$ :

$$u(t + h) = u(t) + h \frac{du(t)}{dt} + \frac{1}{2} h^2 \frac{d^2 u(t)}{dt^2} + \dots \quad (7.3)$$

Thus, if we take only the first two terms of the expansion, we can find the state of the system at the next time step based only on the state of the system and the differential equation:

$$u(t + h) = u(t) + hf(u(t)). \quad (7.4)$$

Now the Euler method can only work with very small values of  $h$ , since our error per step is on the order of  $h^2$ .

One interesting side effect of combining our method for diffusion with our method for our time steps is that our timesteps have to be carefully chosen in relation to the diffusive speed and  $dz$ . This condition is known as the Courant-Friedrichs-Lewy condition,<sup>53</sup> which has a very simple basis behind it. Imagine that we are modeling a wave moving across space, at a rate of 1 unit of distance per unit of time. Now let us imagine that our value of  $dz$  is 0.1, meaning that our wave will have to cross 10 mesh points per unit time, but if our time steps are too big (say  $h$  is 0.5), the wave will not be able to cross the space in the time it should have, for the wave can only move a maximum of one mesh point per timestep (and even then, it would be better to have the wave's movement not being just in step with the method, but have numerous timesteps for each movement).

Also of concern is that with this form of integration (which tends to be less accurate than higher order forms) artifacts can be created quite easily, leading to false results.<sup>54</sup> However, there are a few ways by which we can verify that our results are not artifacts:

1. Varying numerical parameters. A wide variety of runs were done changing the values of things such as  $dz$  and  $h$  (while making sure to obey the Courant-Friedrichs-Lewy condition). While our numerical results match up very well with our bifurcation analysis, it is prudent to ensure that our numerical methods support higher  $k$  (i.e. that the mesh points are close enough to support a pattern with high  $k$ ).

2. Adding noise to the system. In addition to speeding up pattern formation (see section 4.2) and more accurately representing the real world, noise ensures that the system is at least somewhat robust (that is that slight variations in the system do not lead to wildly different outcomes).
3. Changing the grid. In two-dimensional simulations the easiest way to link the grid in space is just to the immediate neighbors above and below and to each side (the five point method - counting the middle), but we can also link diagonally to the next four nearest points (the nine point method), and even change the grid geometry via a hexagonal grid design (seven point method). All of these lead to the exact same results in the end.
4. A completely different numerical technique. As will be shown below, we can use a different method altogether which gives the exact same end results.

Having done all four, it can be said with certainty that our results are not just the result of chance, but are actual valid results.

## 7.1.2 Finite Elements

The finite element method works quite differently. Rather than describing the concentration values through space as a series values at a series of points, here we use a series of linear equations<sup>i</sup> to describe the state of the system across space. Thus over a space  $L$ , there may be twenty linear equations each describing the behavior of the concentration of a space of  $\frac{L}{20}$ , where special care is given to ensure that the end of one equation lines up with the beginning of the next equation. This gives us what is then referred to as a piecewise linear approximation. It should be noted now that FreeFem++<sup>37</sup> was used, as it carries out things such as mesh creation and actually solving the spatial problem (but leaving us to integrate through time).

One ‘catch’ about using finite elements is that we cannot solve our equation in the normal (strong) form, rather they have to be in the weak form<sup>ii</sup>. So if we have

---

<sup>i</sup>This is, by far, not the only choice, one can also use Fourier methods (sine waves) or higher order equations such as quadratics. However, this is the easiest and most efficient to implement.

<sup>ii</sup>Also known as the variational form, the weak form is where we ‘express the problem as infinitely many scalar equations’.<sup>55</sup> In this form we multiply the function by a test function  $a(z)$ , which is

our system (we only carry out the derivation for  $\dot{u}$ , but the same carries over to  $\dot{v}$ ):

$$\frac{\partial u}{\partial t} = \alpha + \beta v - u - \gamma u + D_{uu} \frac{\partial^2 u}{\partial z^2} + D_{uv} \frac{\partial^2 v}{\partial z^2}, \quad (7.5)$$

we can multiply it by a test function  $a(z)$  to get:

$$\frac{\partial u}{\partial t} a(z) = \alpha a(z) + \beta v a(z) - u a(z) - \gamma u a(z) + D_{uu} \frac{\partial^2 u}{\partial z^2} a(z) + D_{uv} \frac{\partial^2 v}{\partial z^2} a(z). \quad (7.6)$$

And now we can integrate with respect to  $z$ :

$$\begin{aligned} \int_0^L \frac{\partial u}{\partial t} a(z) dz &= \int_0^L \alpha a(z) dz + \int_0^L \beta v a(z) dz \\ &- \int_0^L u a(z) dz - \int_0^L \gamma u a(z) dz \\ &+ \int_0^L D_{uu} \frac{\partial^2 u}{\partial z^2} a(z) dz + \int_0^L D_{uv} \frac{\partial^2 v}{\partial z^2} a(z) dz, \end{aligned} \quad (7.7)$$

$$\begin{aligned} \int_0^L \frac{\partial u}{\partial t} a(z) dz &= \int_0^L \alpha a(z) dz + \int_0^L \beta v a(z) dz \\ &- \int_0^L u a(z) dz - \int_0^L \gamma u a(z) dz \\ &- \int_0^L D_{uu} \frac{\partial u}{\partial z} \frac{\partial a(z)}{\partial z} dz - \int_0^L D_{uv} \frac{\partial v}{\partial z} \frac{\partial a(z)}{\partial z} dz, \end{aligned} \quad (7.8)$$

Now we have everything in weak form (the last term being done via integration by parts where by the term  $\frac{\partial u}{\partial x}$  when evaluated at  $x = 0$  and  $x = L$  is zero). We can see now that if our entire system can be linear, as is each term in equation 7.8, we can then represent our entire system with piecewise linear functions. And our test function, of course, will also be piecewise linear. Also of note is that by taking the weak form, we've removed the second derivative (which when the system is represented by piecewise linear equations would always be zero). If there was no time involved, this actually another piecewise linear equation.

system would be easily solved via the Galerkin method<sup>55iii</sup>. However, the time term is actually taken care of with the discretization and implementation of the algorithm below.

The simple explicit Euler algorithm could be used again, but instead we use the implicit Euler algorithm:

$$u(t+h) = u(t) + hf(u(t+h)). \quad (7.9)$$

As one can see, while this method is similar to the explicit Euler method, we now need some way to solve for  $u(t+h)$ . What we do in the code is actually simpler than trying to solve the equation. Instead we run the following:

$$u_{trial} = u(t) + hf(u_{last}). \quad (7.10)$$

The first integration has  $u_{last}=u(t)$  (thus making the first iteration an explicit Euler step), but after the first step we set  $u_{last} = u_{trial}$ , from the previous iteration. For each step we then get an error calculation based on  $u_{trial}-u_{last}$ , the system then repeats over and over until the error shrinks down to a predetermined value. Another advantage of this technique is that the error calculation allows us one further optimization, adaptive step sizes.

To use adaptive step sizes we need a few things. First of all, the numerical technique must support variable  $h$  (as implicit Euler does, but explicit Euler as used in the previous section would not), since not only will adaptive step sizes shrink  $h$  which works well for all techniques, but it also expands  $h$ , running us back to the Courant-Friedrichs-Lewy condition. Also we need a means to gauge the change in the system, for which we use the error calculation.

Before detailing how we implement adaptive step sizes, why do we want to use it? Because it saves computer time, very large amounts of it in fact. When the system is rapidly changing, the difference between  $u(t)$  and  $u_{last}$ , for a given step size, is very large, thus it takes many iterations before the error shrinks to the value for the system to move to the next time step. Similarly, when the system is only

---

<sup>iii</sup>The Galerkin method, in a nutshell, is where the problem is reexpressed in matrix form and solved by linear algebra.

changing slightly, the number of iterations before the step is complete will be very small - so small that the step size ( $h$ ) could be twice as large and still take the same number of iterations. Thus if we can adjust  $h$  such that it is small when the system is undergoing large changes, and large when the system is relatively stable, the time it takes for a numerical simulation is dramatically cut back.

While it does take some tinkering to get the optimal settings for the adaptive step sizes, the time used per run is easily cut by over 50% if not more. The way we carry this out is quite simple. If the error remains within a certain range, the step size  $h$  remains the same. If the error is too small (i.e. well below the value for which the system goes to the next step), we increase  $h$  by 50%. Should the error become too large (i.e. above the range for which  $h$  remains static),  $h$  is decreased by 33%. While this particular algorithm may seem to be biased toward larger  $h$ , it has proven to be quite robust.

Again, we have to ensure that our results are not false:

1. Remove the adaptive step sizes. We set the step size to a fixed value to get the exact same results (albeit taking far longer to run).
2. Changing the numerical parameters. Similarly to the explicit technique, we can change the mesh<sup>iv</sup> that we use, and even the shape of the system. Instead of the generic box that works best for the explicit technique we can use triangles, rectangles, circles, et cetera, all of which work.
3. Implicit vs. explicit Euler. Before implementing the implicit technique, the explicit one was used first, and while it was the least efficient way of simulating the system, there were no changes in the end results.

One common method for improving simulations with finite element methods is the use of mesh adaptation, which increases the mesh density around more interesting areas (so the large flat areas in concentration would be less densely represented whilst the slopes of peaks would have more mesh points). Unfortunately since the peak position is not fixed, especially during the initial pattern formation, a mesh adaptation does

---

<sup>iv</sup>In two dimensions instead of a series of connected equations, represented by lines, we have a web of lines covering the two dimensional shape.

not actually improve results, but instead leads to uneven mesh distribution that does not correlate to the actual pattern, which in turn distorts the results. The better way to ensure a proper mesh for our equations is to err on the side of higher mesh density.



## References

- [1] A. M. Turing. ‘On computable numbers, with an application to the Entscheidungsproblem.’ *Proc. London Math. Soc.*, **42**, 230–265, 1937.
- [2] A. M. Turing. ‘Computing machinery and intelligence.’ *Mind*, **59**, 433–460, 1950.
- [3] A. M. Turing. ‘The chemical basis of morphogenesis.’ *Phil. Trans. R. Soc. Lond. B*, **237**, 37–72, 1952.
- [4] W. Mazin, K. E. Rasmussen, E. Mosekilde, P. Borckmans, and G. Dewel. ‘Pattern formation in the bistable Gray-Scott model.’ *Math. Comp. Sim.*, **40**, 371–396, 1996.
- [5] P. De Kepper, V. Castets, E. Dulos, and J. Boissonade. ‘Turing-type chemical patterns in the chlorite-iodine-malonic acid reaction.’ *Physica D*, **49**, 161–169, 1991.
- [6] V. Castets, E. Dulos, J. Boissonade, and P. De Kepper. ‘Experimental evidence of a sustained standing Turing-type nonequilibrium chemical pattern.’ *Phys. Rev. Lett.*, **64**, 2953–2956, 1990.
- [7] Q. Ouyang, R. Li, G. Li, and H. L. Swinney. ‘Dependence of Turing pattern wavelength on diffusion rate.’ *J. Chem. Phys.*, **102**, 2551–2555, 1995.
- [8] J. H. E. Cartwright. ‘Labyrinthine Turing pattern formation in the cerebral cortex.’ *J. theor. Biol.*, **217**, 97–103, 2002.
- [9] S. Kondo and R. Asai. ‘A reaction-diffusion wave on the skin of the marine angelfish *Pomacanthus*.’ *Nature*, **376**, 765–768, 1995.
- [10] S. Sick, S. Reinker, J. Timmer, and T. Schlake. ‘WNT and DKK determine hair follicle spacing through a reaction-diffusion mechanism.’ *Science*, **314**, 1447–1450, 2006.
- [11] Images in the public domain: <http://en.wikipedia.org/wiki/Image:GtsklLeopard.jpg> and <http://en.wikipedia.org/wiki/Image:GrantsZebra2.jpg>. Sites accessed last on August 20, 2007.
- [12] A. Fick. ‘On liquid diffusion.’ *Phil. Mag. and Jour. Sci.*, **10**, 31–39, 1855.
- [13] C. J. Roussel and M. R. Roussel. ‘Reaction-diffusion models of development with state-dependent chemical diffusion coefficients.’ *Prog. Biophys. Mol. Biol.*, **86**, 113–160, 2004.

- 
- [14] J. Jorné. ‘Negative ionic cross diffusion coefficients in electrolytic solutions.’ *J. Theor. Biol.*, **55**, 529–532, 1975.
- [15] H. Malchow. ‘Dissipative pattern formation in ternary non-linear reaction-electrodifffusion systems with concentration-dependent diffusivities.’ *J. Theor. Biol.*, **135**, 371–381, 1988.
- [16] D. del Castillo-Negrete, B. A. Carreras, and V. Lynch. ‘Front propagation and segregation in a reaction-diffusion model with cross-diffusion.’ *Physica D*, **168**, 45–60, 2002.
- [17] Y. Huang and O. Diekmann. ‘Interspecific influence on mobility and Turing instability.’ *Bull. Math. Biol.*, **65**, 143–156, 2003.
- [18] M. R. Roussel and J. Wang. ‘Transition from self-replicating behavior to stationary patterns induced by concentration-dependent diffusivities.’ *Phys. Rev. Lett.*, **87**, 188302, 1–4, 2001.
- [19] M. R. Roussel and J. Wang. ‘Pattern formation in excitable media with concentration-dependent diffusivities.’ *J. Chem. Phys.*, **120**, 8079–8088, 2004.
- [20] J. E. Pearson and W. Horsthemke. ‘Turing instabilities with nearly equal diffusion coefficients.’ *J. Chem. Phys.*, **90**, 1588–1599, 1989.
- [21] J. H. Merkin, V. Petrov, S. K. Scotts, and K. Showalter. ‘Wave-induced chaos in a continuously fed unstirred reactor.’ *J. Chem. Soc., Faraday Trans.*, **92**, 2911–2918, 1996.
- [22] M. Mincheva and M. R. Roussel. ‘A graph-theoretic method for detecting potential Turing bifurcations.’ *J. Chem. Phys.*, **125**, 204102, 2006.
- [23] D. Franz and M. Roussel. ‘Turing patterns in a linear reaction system with nonlinear cross-diffusion coefficients.’ Submitted to *Physica D*.
- [24] J. D. Murray. *Mathematical Biology, I: An Introduction* (Springer-Verlag, Berlin, 2002).
- [25] M. R. Roussel. ‘Chemistry 5850, Nonlinear Dynamics.’, 2006. [Http://people.uleth.ca/~roussel/nld/](http://people.uleth.ca/~roussel/nld/).
- [26] L. A. Segel and M. Slemrod. ‘The quasi-steady-state assumption: A case study in perturbation.’ *SIAM Rev.*, **31**, 446–477, 1989.
- [27] D. T. Finkbeiner II. *Introduction to matrices and linear transformations* (W.H. Freeman and Company, San Francisco, 1978), third edition.

- 
- [28] J. Z. Hearon. ‘The kinetics of linear systems with special reference to periodic reactions.’ *Bull. Math. Biophys.*, **15**, 121–141, 1953.
- [29] J. Z. Hearon. ‘Theorems on linear systems.’ *Ann. N.Y. Acad. Sci.*, **108**, 36–68, 1963.
- [30] B. Ermentrout. *Simulating, Analyzing, and Animating Dynamical Systems: A Guide to XPPAUT for Researchers and Students* (SIAM, Philadelphia, 2002).
- [31] J. D. Murray. *Mathematical Biology, II: An Introduction* (Springer-Verlag, Berlin, 2002).
- [32] I. L. Kliakhandler, A. V. Porubov, and M. G. Velarde. ‘Localized finite-amplitude disturbances and selection of solitary waves.’ *Phys. Rev. E*, **62**, 4959–4962, 2000.
- [33] A. L. Hodgkin and A. F. Huxley. ‘A quantitative description of membrane current and its application to conduction and excitation in nerve.’ *J. Physiol.*, **117**, 500–544, 1952.
- [34] H. Othmer and L. Scriven. ‘Instability and dynamic pattern in cellular networks.’ *J. Theor. Biol.*, **32**, 507–537, 1971.
- [35] E. M. Wright. ‘Renal  $\text{Na}^+$ -glucose cotransporters.’ *Am. J. Physiol. Renal Physiol.*, **280**, F10–F18, 2001.
- [36] D. Voet and J. G. Voet. *Biochemistry* (Wiley, Hoboken NJ, 2004), third edition.
- [37] O. Pironneau, F. Hecht, and A. L. Hyaric. ‘Freefem++ (v 2.1.8).’
- [38] W. Cheney and D. Kincaid. *Numerical Mathematics and Computing* (Thompson Learning, Belmont CA, 2004), 5th edition.
- [39] P. Borckmans, G. Dewel, A. De Wit, E. Dulos, J. Boissonade, F. Gauffre, and P. De Kepper. ‘Diffusive instabilities and chemical reactions.’ *Int. J. Bifurc. Chaos.*, **12**, 2307–2332, 2002.
- [40] P. K. Maini, K. J. Painter, and H. N. P. Chau. ‘Spatial pattern formation in chemical and biological systems.’ *J. Chem. Soc., Faraday Trans.*, **93**, 3601–3610, 1997.
- [41] Q. S. Li and L. Ji. ‘Control of Turing pattern by weak spatial perturbation.’ *J. Chem. Phys.*, **120**, 9690–9693, 2004.

- 
- [42] D. T. Gillespie. ‘Exact stochastic simulation of coupled chemical reactions.’ *J. Phys. Chem.*, **81**, 2340–2361, 1977.
- [43] G. Marsaglia, B. Narasimhan, and A. Zaman. ‘A random number generator for PC’s.’ *Comput. Phys. Commun.*, **60**, 345–349, 1928.
- [44] G. E. P. Box and M. E. Muller. ‘A note on the generation of random normal deviates.’ *Ann. Math. Statist.*, **29**, 610–611, 1958.
- [45] T. Leppänen, M. Karttunen, K. Kaski, and R. A. Barrio. ‘Dimensionality effects in Turing pattern formation.’ *Int. J. Mod. Phys. B*, **17**, 5541–5553, 2003.
- [46] T. Leppänen, M. Karttunen, R. Barrio, and K. Kaski. ‘The effect of noise on Turing patterns.’ *Prog. Theor. Phys. Supp.*, **150**, 367–370, 2003.
- [47] V. K. Vanag. ‘Waves and patterns in reaction-diffusion systems. Belousov-Zhabotinsky reaction in water-in-oil microemulsions.’ *Physics - Uspekhi*, **47**, 923–941, 2004.
- [48] E. A. Abbot. *Flatland* (Penguin Books, New York, 1998).
- [49] A. Zhabotinskii. ‘Periodic processes of the oxidation of malonic acid in solution (Investigation of the kinetics of the reaction of Belousov).’ *Biofizika*, **9**, 306–311, 1964.
- [50] T. Hillen and K. Painter. ‘A users guide to PDE models for chemotaxis.’ Submitted to *J. Math. Biol.*
- [51] K. J. Painter, P. K. Maini, and H. G. Othmer. ‘A chemotactic model for the advance and retreat of the primitive streak in avian development.’ *Bull. Math. Biol.*, **62**, 501–525, 2000.
- [52] B. N. Vasiev, P. Hogeweg, and A. V. Panfilov. ‘Simulation of *Dictyostelium Discoideum* aggregation via reaction-diffusion model.’ *Phys. Rev. Lett.*, **73**, 3173–3176, 1994.
- [53] R. Courant, K. Friedrichs, and H. Lewy. ‘On the partial difference equations of mathematical physics.’ *Math. Ann.*, **100**, 32–74, 1928.
- [54] I. Barrass, E. J. Crampin, and P. K. Maini. ‘Mode transitions in a model reaction-diffusion system driven by domain growth and noise.’ *Bull. Math. Biol.*, **68**, 981–995, 2006.
- [55] M. S. Gockenbach. *Partial Differential Equations: Analytical and Numerical Methods* (SIAM, Philadelphia, 2002).