UNIVERSITY OF CALGARY

Modelling Natural Phenomenon with Reaction-Diffusion

by

Lee Ringham

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE

CALGARY, ALBERTA
JANUARY, 2020

Abstract

Procedural methods provide an algorithmic way to produce textures for use in computer graphics. One such method, reaction-diffusion, is a powerful mathematical approach that describes natural pattern formation in terms of chemicals known as morphogens. This thesis describes LRDS, an environment for authoring reaction-diffusion models directly on arbitrary surfaces. Morphogens, their behaviours, and the domain in which they reside can be quickly and easily defined. By performing computation on the GPU, the pattern forming simulation can be interacted with in real-time, facilitating productivity and experimentation. Four case studies are presented. The first is a simulation of ladybug pigmentation patterns. The second is a simulation of pigmentation patterns seen on the body of snakes. The third study looks at flower petal pattern modelling. Lastly, a biologically-motivated model of the autoimmune disease psoriasis is presented.

Acknowledgements

I want to acknowledge my supervisor Dr. Prusinkiewicz for all his guidance and support. I would also like to thank the members of the Algorithmic Botany lab, including Andrew Owens, Jeremy Hart, Cory Bloor, Mik Cieslak, Philmo Gu, Desmond Larsen-Rosner, and Pascal Ferraro. Your advice and thoughtful conversations greatly helped me to learn, develop, and enjoy this work. For his work on psoriasis, I also thank Robert Gniadecki. And finally, I thank Robert Munafo for the advice on Gray-Scott reaction-diffusion models and their parameter space.

To Carol, Mark, Andrew, and my loving wife Natalie.

Table of Contents

List of Figures and Illustrations

List of Tables

Chapter 1

Introduction

From the stripes on a zebra to the spots on a leopard, nature provides a wide variety of beautiful patterns (Fig. 1.1). In 1952 Alan Turing proposed a system of partial differential equations (PDEs) aimed at explaining the formation of natural patterns. The patterns are created from chemicals that diffuse and react together through a spatial medium (Turing, 1952). The chemicals are thought to cause various phenomena such as specialization of tissue in a process known as morphogenesis. Therefore, these chemicals are referred to as morphogens.[1] This system is named reaction-diffusion and has since become widely used in mathematical and computational modelling of natural pattern formation. In 1972 Hans Meinhardt and Alfred Gierer independently discovered and advanced reaction-diffusion by focusing on the roles of long-range activation and short-range inhibition (Gierer and Meinhardt, 1972). Since then, advanced reaction-diffusion models have been created and used to explain many different biological patterns (Garzón-Alvarado, Diego A. and Ramírez Martinez, Angelica M., 2011; Fowler et al., 1992; Lefèvre and Mangin, 2010; Meinhardt, 2009).

[1]The term "morphogen" should not be confused with Wolpert's positional signals definition (Wolpert, 1996).

Figure 1.1 – Examples of beautiful patterns found in nature. *Photographs by pixabay.com, licensed under Pixabay License.*

There has been a large body of work focused on simulating reaction-diffusion patterns. These simulations can be used to give insights into the inner workings of nature or provide textures for use in computer graphics. Regular grids represent the space in which most simulations compute the partial differential equations. Chemicals are found in the grid cells, which represent discrete areas. This arrangement offers many advantages, such as ease of evaluation and the use of specialized graphics hardware to accelerate computation.

Witkin and Kass (1991) used reaction-diffusion as a method of texture synthesis for computer graphics and extended the range of possible patterns from traditional reaction-diffusion by introducing anisotropic diffusion and varying diffusion rates in their simulations. The patterns were simulated on a grid, which was subsequently mapped onto a parametric surface as a texture. Grid boundaries were connected in the topology of a torus to avoid

seams in the texture.

Growth of the spatial medium supporting pattern formation can affect the pattern's appearance[2]. Although grid domains can grow, they are not suitable for growth occurring just at a single location. Local growth requires an arbitrary surface not constrained to a rectangular shape.

Lefèvre and Mangin (2010) modelled the growth and folding of a brain using reaction-diffusion. In this work, they simulated a labyrinth pattern on a spherical mesh representing the brain's surface. Chemical concentrations determined the rate of mesh growth. This growth, in turn, provides more space for the pattern to develop. The result is labyrinth-like folds protruding from the surface of the mesh. Harrison et al. (2002); Holloway and Harrison (2007) modelled the growth of plants by coupling reaction-diffusion and surface deformation in the same way. Fowler et al. (1992) modelled patterns found on seashells using a special case of growth. The shell domain starts as a 1D layer representing the initial conditions of the pattern. Layers are accrued over time, where each subsequent layer is a progression through time of the simulation.

Turk (1991) simulated reaction-diffusion on meshes by using a second Voronoi mesh to represent the original mesh surface. This Voronoi mesh was used as the spatial domain and chemicals were stored in its faces. The rate of diffusion across face edges depends on the edge lengths. Using a second mesh avoids modification of the original mesh and allows for the generation of detailed pattern textures. Another benefit of simulating directly on a mesh is that there is no need to correct for pattern distortion that occurs when mapping a grid to an arbitrary surface. Unfortunately, there is no consideration for growth or interaction.

Reaction-diffusion has also been solved directly on triangular meshes (Descombes, Samira Michèle and Dhillon, Daljit Singh and Zwicker, Matthias, 2016), avoiding the need for a Voronoi mesh. This work leveraged the GPU, allowing for much faster simulation progression compared to CPU-based computation. This speed facilitated parameter space exploration

[2]This is an observation that Turing identified but purposely ignored.

and pattern creation. However, Descombes, Samira Michèle and Dhillon, Daljit Singh and Zwicker, Matthias (2016) do not consider growth or anisotropic diffusion.

There have been a number of software applications designed around creating models that can procedurally generate patterns. A program named "TexRD" supports simulation of reaction-diffusion on grid based domains (Luc Decker, 2019). "NetLogo" is a mature and well tested application that allows for fast simulation of PDEs and other pattern forming systems such as cellular automata (Wilensky, U, 1999). A cross-platform application named "READY" is designed around simulating GPU accelerated reaction-diffusion equations on arbitrary domains (GollyGang, 2012). In the listed applications, there is no built-in support for extensions to reaction-diffusion such as anisotropic diffusion on arbitrary triangulated surfaces. Also, parameter specification is only done through editing text files.

In this thesis I present "Lightweight Reaction-Diffusion Simulator" (LRDS), a program that facilitates simulation and exploration of reaction-diffusion patterns, including those formed on grids and arbitrary triangular meshes. These meshes have support for growth, anisotropic diffusion, and parameters that change value in over space and time. Reaction-diffusion is computed directly on triangular meshes by combining the use of the CPU and GPU. Adaptive subdivision and user interaction, which do not benefit from parallelism due to their recursive or sequential nature, use the CPU. Parallelizable algorithms, such as reaction-diffusion equations are calculated efficiently by using the GPU. This mixed approach makes it possible to create fast and interactive simulations.

This thesis is organized as follows. First, I review the principles of reaction-diffusion patterning, and I explain the computational aspects of simulating reaction-diffusion. Next, I describe the implementation details of the LRDS system. On this basis, I present four case studies. The first study analyses ladybug shell patterns. Liaw et al. (2001) produced simulations of ladybug shell patterns on a grid located on a partial sphere. I improve on this study by using arbitrary triangular meshes to replace the partial sphere. These meshes allow for an organic shell shape. I also explore some alternative parameters and initial

conditions to improve on the pattern stability and appearance. The second study produces reaction-diffusion patterns that resemble those found on snakes. Anisotropy is used to align these patterns to the snake mesh and growth is used to create more complicated patterns such as rows of dual spots. The third study explores pigmentation patterns on flowers. I produced models of various flower species on meshes by using varying parameters and anisotropic diffusion. Some flowers considered include foxglove, monkeyflower, and orchids, which display vibrant patterns on attractive curved petals. Although petal pigmentation patterns are visually striking, it is an area that remains largely unexplored despite the large amount of work put into modelling other aspects of flower petals (Owens et al., 2016). Next, I present a case study on modelling the autoimmune disease psoriasis based on contemporary research into cytokine interactions. A computational model of psoriasis has the potential to be a useful tool as it provides a mathematical representation of the disease and a fast, non-invasive way to test the disease's response to treatments. It is hoped this model can be extended to other autoimmune diseases as well as increasing the efficacy of treatments. Finally, I conclude the main contributes of this thesis and discuss future work.

Chapter 2

Principles of Reaction-Diffusion Patterning

Alan Turing proposed a system of differential equations as a model for biological pattern formation (Turing, 1952). In this system, a spatial medium contains an initially homogeneous distribution of chemicals. This arrangement is in an unstable equilibrium and the presence of small chemical perturbations instigates pattern formation. The system responds in a few ways over time: it can gain stability in a patterned state, oscillate between patterns, or settle in a homogeneous non-patterned state.

2.1 Reaction-Diffusion in a Continuous Domain

Reaction-diffusion is formalized as a set of PDEs that represent the change in concentration of morphogens over time. Considering morphogens a and b, a two-substance reaction-diffusion system is defined by the system of equations:

$$\begin{aligned} \frac{\partial a}{\partial t} &= F(a,b) + D_a \nabla^2 a, \\ \frac{\partial b}{\partial t} &= G(a,b) + D_b \nabla^2 b. \end{aligned} \tag{2.1}$$

As the name suggests, reaction-diffusion is composed of two mechanisms, reaction and diffusion. Functions F and G describe the production and decay of a and b, and together constitute the reactions of the system. D_a and D_b are coefficients that control how fast these morphogens diffuse through the domain. Physically, they depend on the morphogen particle size and permeability of the domain. ∇^2 is the Laplacian operator, and in conjunction with the diffusion coefficients, describes the diffusion component of the system. The concentrations of a and b are functions of time and position. All other values are constant parameters.

These equations apply at any position in the domain in which reaction-diffusion is occurring. The size, shape, and growth of this domain may also play a role in pattern formation. Some patterns require a minimum amount of space to form. As domain size increases, the same parameters can produce different patterns. Additionally, the shape of the domain can affect pattern positioning and orientation.

2.2 Reaction-Diffusion Models

2.2.1 The Turing model

Turing (1952) considered reaction-diffusion on both a discrete and continuous 1D ring. Initially, morphogen concentrations would be constant across the domain and the system would be in a stable state. Low-amplitude noise would then instigate pattern formation. The equations he proposed had the form:

$$\begin{aligned} \frac{\partial v}{\partial t} &= s(uv - v - \beta) + D_v \nabla^2 v, \\ \frac{\partial u}{\partial t} &= s(\alpha - uv) + D_u \nabla^2 u. \end{aligned} \tag{2.2}$$

The morphogen u has a base production α. The uv term describes the rate at which the morphogen u is converted into the morphogen v. Changes to s, the reaction rate, scale

the pattern features. For example, a spot pattern will appear larger with smaller values of parameter s. β controls a constant removal of v.

2.2.2 The Activator-Inhibitor Model

The concept of reaction-diffusion was reinvented by Gierer and Meinhardt (1972) (see also (Meinhardt, 1982)). They considered two morphogens, an activator a and an inhibitor h. The activator is autocatalytic, using itself to reproduce. It also spurs the production of the inhibitor, which slows autocatalysis. The interplay between these two actions is what allows for patterns to form. Meinhardt and Gierer advanced the idea that pattern formation depended on is the activator diffusing much slower than the inhibitor. Often this is the case, but such drastic differences in diffusion are not always needed (Gray and Scott, 1984; Marcon et al., 2016). The activator-inhibitor model is defined as:

$$\begin{aligned} \frac{\partial a}{\partial t} &= \rho \frac{a^2}{h} - \mu_a a + \rho_a + D_a \nabla^2 a, \\ \frac{\partial h}{\partial t} &= \rho a^2 - \mu_h h + \rho_h + D_h \nabla^2 h. \end{aligned} \tag{2.3}$$

In these equations, ρ is the reaction rate. μ_a and μ_h are the decay rates of a and h. ρ_a and ρ_h represent the base production of a and h.

2.2.3 The Activator-Depleted Substrate Model

The activator-depleted substrate model is another model proposed by Gierer and Meinhardt (1972). In this model, the inhibitor is replaced by a substrate that the activator uses to perform autocatalysis. The inhibition mechanism results from the depletion through the consumption of the substrate by the activator. This is represented by the equations:

$$\begin{aligned} \frac{\partial a}{\partial t} &= \rho s a^2 - \mu_a a + \rho_a + D_a \nabla^2 a, \\ \frac{\partial s}{\partial t} &= -\rho s a^2 - \mu_s s + \rho_s + D_s \nabla^2 s. \end{aligned} \tag{2.4}$$

Here ρ is the reaction rate, μ_a and μ_s are the decay rates, and ρ_a and ρ_s are the base production rates of a and s.

2.2.4 The Gray-Scott Model

Gray and Scott (1984) investigated the behaviour of a simple irreversible set of reactions and discovered it produced interesting patterns. These reactions occur in an isothermal, continuously stirred tank reactor into which chemicals U is continuously fed. V reacts with U in the tank and decays into an inert product P. This system is known as the Gray-Scott model, and the reactions are formalized as:

$$\begin{aligned} U + 2V &\rightarrow 3V, \\ V &\rightarrow P. \end{aligned} \tag{2.5}$$

When represented as partial differential equations, this system has the form:

$$\begin{aligned} \frac{\partial v}{\partial t} &= uv^2 - (F+k)v + D_v \nabla^2 v, \\ \frac{\partial u}{\partial t} &= -uv^2 + F(1-u) + D_u \nabla^2 u. \end{aligned} \tag{2.6}$$

uv^2 represents $U + 2V \rightarrow 3V$ and the constant k controls the rate at which $V \rightarrow P$ occurs. F is a scalar parameter that controls how much of u is fed into the system and the proportion of u and v that is removed. The feeding and removal of u is controlled by F which has the effect of trying to keep the concentration of u near 1. Remarkably, the diffusivity coefficients use a ratio of 1:2 for the activator and substrate, which is a smaller ratio than considered by Gierer and Meinhardt. Except for a narrow range of parameters, pattern formation in this system is not instigated by noise. A pre-pattern is required to start pattern formation.

The Gray-Scott model is a specific case of the activator-depleted substrate model. Using the substitutions: $a = v$, $s = u$, $\rho = 1$, $\rho_a = 0$, $\mu_a = F + k$, $\rho_s = F$, and $\mu_s = F$, we obtain Eqn. 2.4. Pearson (1993) extensively explored and visualized the Gray-Scott model in 2D

and produced many diverse patterns (Fig. 2.1).

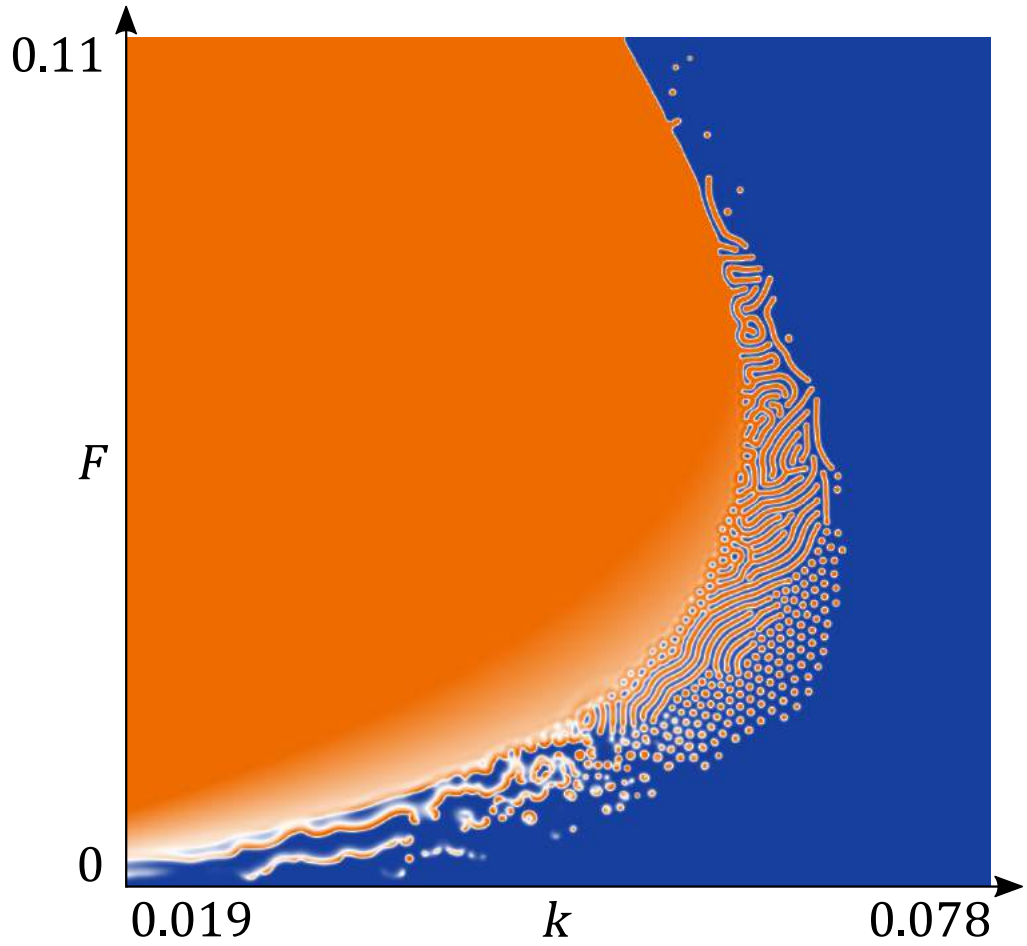


Figure 2.1 – Gray-Scott parameter space for parameters k on the x-axis ranging from $[0.019-0.078]$ and F on the y-axis ranging from $[0-0.11]$. Visualized is the concentration of morphogen a. Low concentration values are blue and high concentrations are orange.

2.3 Extensions to Basic Reaction-Diffusion

The basic reaction-diffusion concept proposed by Turing and contextualized by Gierer and Meinhardt has provided the tools for reasoning about natural pattern formation. From that idealized system, we consider extensions to represent nature and produce realistic patterns more accurately.

One such extension is changing parameters based on their position in the domain. A commonly changed parameter is the diffusion rate. Increasing the diffusion rate along the

domain causes patterns such as meandering stripes to exhibit a preferred orientation (Zheng et al., 2009). Witkin and Kass (1991) also varied diffusion rates to correct for pattern distortion on curved surfaces. The change of parameters in space can be driven by images to create intricate patterns for artistic renderings.

Not only does diffusion vary, it can be anisotropic due to the heterogeneous structure of tissue. Zhou et al. (2007) modelled pigment patterns on flower petals by considering the influence of veins on diffusion. The diffusion between adjacent cells is modified based on the presence of vascular cells. Adding consideration for vein width results in darker pigment patterns along veins, as seen in real flowers such as orchids. Sanderson et al. (2004) used anisotropic diffusion with reaction-diffusion to visualize vector fields. A uniform spot pattern has its diffusion driven by an underlying vector field that produces distorted ovals. In a study of snake pattern diversity, Allen et al. (2013) used anisotropic diffusion to model various snakeskin patterns and associated them with snake behaviour.

Domain growth can affect morphogen patterns. Patterns may form before growth is finished, and subsequent growth will cause patterns to stretch and deform. Alternatively, patterns can develop in tandem with growth, adjusting and migrating with available space. Kondo and Asai (1995) modelled the stripe patterns found on the angelfish *Pomacanthus* by considering pattern formation during growth. The pattern is not affixed to the underlying skin and gains new stripes that insert between existing ones over time. Similarly, J.D. Murray and M.R. Myerscough (1991) modelled the effect of growth on snakeskin pigment pattern formation. Fowler et al. (1992) modelled patterns formed on a growing 1D seashell margin. Pattern-driven growth couples the growth rate of the surface with morphogen concentrations. Lefèvre and Mangin (2010) used pattern-driven growth to model brain development. This coupling has also been used to simulate structure formation in plants (Harrison et al., 2002; Holloway and Harrison, 2007).

Multi-stage models where parameter values change over time have been used to simulate the pigment patterns on leopards and jaguars (Liu et al., 2006). Malheiros and Walter (2017)

simulated moray eel spots by varying diffusion rates and changing morphogen saturation limits on different areas of the body. Changing these parameters had the effect of creating irregular spots that changed into a thicker labyrinth-like pattern. Having more than two morphogens allows for different patterns. Schenk et al. (2000) created a three-morphogen model that produces a pattern of spot clusters that move together as a group. Meinhardt (1982) created a five-morphogen model that produced zebra-like stripe patterns.

Due to high sensitivity to small changes in parameters and the dependence on a difference in diffusion rates, it is hard to justify reaction-diffusion models as a representation of nature. Stationary cells are an example of immobile cell-autonomous factors that play a role in real pattern formation. They can be represented as non-diffusing morphogens but are usually abstracted away when creating a model. Omitting non-diffusing morphogens is done to simplify the model and focus on the obvious morphogens. In a study by Marcon et al. (2016), they found that 70% of three or four-morphogen systems, including non-diffusing morphogens, do not require differing diffusion rates. Also, the patterns formed by these models are much less sensitive to parameter changes. Including these non-diffusing morphogens provides a closer representation of reality and suggests that some basic assumptions, such as long-range inhibition and short-range activation, should be used as a special case rather than the standard when modelling biological patterns.

Chapter 3

Computational Aspects

3.1 Systems of Chemical Reactions

Reaction-diffusion equations are an idealized and abstract representation of reality. They capture two phenomena, chemical reactions and diffusion. Mathematically, this is represented as a system of partial differential equations, whose numerical solution requires discretization of time and space. Computation of the chemical reactions at a specific point in space depends on the elapsed time and previous concentration. Computation of diffusion, however, depends additionally on the concentrations at adjacent points.

3.2 Isotropic Diffusion

Diffusion is a process in which particles of a substance move from areas with high concentration to areas with low concentration. The change in concentration over space is referred to as a concentration gradient. Diffusion was formalized in 1855 by Fick's second law:

$$\frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u). \tag{3.1}$$

The symbol $\nabla\cdot$ is the divergence operator, ∇ is the gradient operator, and u is a scalar field. Given Eqn. 3.1, we see that the diffusion rate depends on the gradient of the concentration. At a given location, the divergence of the gradient of u measures the difference between the concentration at that point and the average of the neighbouring concentrations. Diffusion is proportional to this change in concentration. However, particle size and domain porosity also affect the diffusion rate. This is represented by the diffusivity coefficient D. If diffusivity is the same regardless of direction, this diffusion is said to be isotropic, and we can simplify Eqn. 3.1 to:

$$\frac{\partial u}{\partial t} = D\nabla^2 u, \tag{3.2}$$

where ∇^2 is the divergence of the gradient and is called the Laplacian. Formally, it is the sum of the second spatial derivative in each basis direction x_i.

$$\nabla^2 = \sum_{i=0}^{n} \frac{\partial^2}{\partial x_i^2}. \tag{3.3}$$

3.3 Anisotropic Diffusion

In many biological scenarios, particles do not diffuse at an equal rate in all directions. In that case, the diffusion is called anisotropic, and the diffusivity coefficient changes based on the direction considered. For anisotropic diffusion, a single scalar D no longer suffices. We can represent anisotropy by a tensor Λ. In the two-dimensional case, the tensor has the form:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{3.4}$$

Each λ_i represents the diffusivity in a basis of the space. In practice, λ_1 and λ_2 are used to represent a portion of the desired scalar diffusion rate. To represent isotropic diffusion, $\lambda_1 = 1$ and $\lambda_2 = 1$. Λ is axis-aligned. Arbitrary orientations can be calculated with a

rotation matrix R, allowing us to represent general anisotropic diffusion:

$$D = R^T \Lambda R. \tag{3.5}$$

We can use D to transform the gradient and thereby the diffusion rate, obtaining Eqn. 3.1.

3.4 Discrete Diffusion Operators

To simulate reaction-diffusion, the domain on which it is simulated must be discretized, which in turn means that we must use a discrete version of the Laplacian.

3.4.1 Diffusion on Grids

Grids of square cells are a common representation of domains. This has many benefits as grids are easy to represent, and the required differential operators are simple to implement. Each cell in the grid has an area associated with concentrations of morphogens. Computationally, concentrations are represented by a scalar value assigned to each cell.

Diffusion in 1D

Diffusion on a 1D grid of cells can be computed by representing the Laplacian through finite differencing operations. Because the Laplacian involves second-order derivatives, we can approximate it by computing the difference of two first-order differences for a grid cell centred at i, where i is the i^{th} cell:

$$\begin{aligned} T_0 &= \frac{(u_i - u_{i-1})}{h}, \\ T_1 &= \frac{(u_{i+1} - u_i)}{h}. \end{aligned} \tag{3.6}$$

Here u_{i-1}, u_i, and u_{i+1} are morphogen concentrations at their respective grid cells, T_0 and T_1 are the first order differences, and h is the distance between the centres of adjacent cells.

The discrete Laplacian is then:

$$\nabla^2 u_i = \frac{T_1 - T_0}{h} = \frac{u_{i-1} - 2u_i + u_{i+1}}{h^2}. \tag{3.7}$$

This differencing can be represented as a convolution mask during computation, as shown below:

$$\nabla^2 = \begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline \end{array}$$

Diffusion in 2D

In this case we have two directions to consider. Recall that the Laplacian is the sum of the second derivatives in each principle direction. This allows us to use the summation of the 1D case in both x and y directions, which, for a cell $u_{i,j}$, yields:

$$\nabla^2 u_{i,j} = \frac{u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j}}{h^2}. \tag{3.8}$$

Again, the Laplacian operator can be represented as a convolution mask:

$$\nabla^2 = \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

3.4.2 Diffusion on Arbitrary Triangular Meshes

Triangular meshes are ubiquitous in computer graphics. Meshes allow for a discrete representation of arbitrary surfaces, and there are well-studied algorithms for growing and subdividing them. These properties make meshes a good candidate for reaction-diffusion simulations. As in the grid case, concentrations are associated with cells. These cells are

the faces of the mesh dual graph. Concentrations are stored at the vertices because each vertex is located at the centre of its cell. To allow for easy neighbour identification and area calculation, I represent the mesh as a half-edge data structure (Marschner and Shirley, 2015).

Two half-edges replace each edge in the mesh to build this data structure. Each half-edge stores a pointer to the next half-edge in the same face, a pointer to the vertex that it originates from, and a pointer to the complementary half-edge (Fig. 3.1). Each vertex stores a scalar representing the area of the dual cell. This area is the sum of one-third of each adjacent face's area.

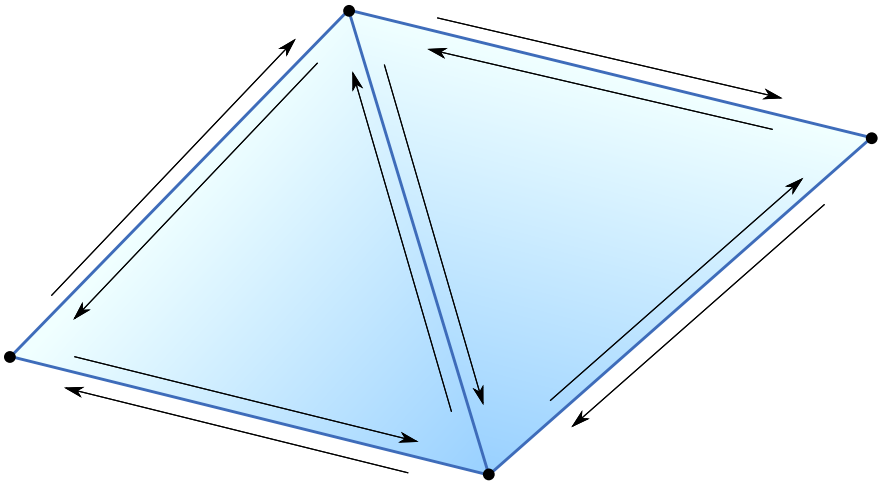


Figure 3.1 – Two triangles and their half-edge representation denoted by black arrows.

Isotropic Diffusion on Meshes

To compute diffusion on an arbitrary triangular mesh, we need a discrete Laplacian. This Laplacian can be a generalization of Eqn. 3.8:

$$(\nabla^2 u)_i = \frac{1}{A} \sum_{i \sim j} w_{ij}(u_i - u_j). \tag{3.9}$$

This generalized equation states that the Laplacian at vertex i of the scalar field u is the sum of the weighted differences between, u_i, the concentration at i, and each neighbouring

concentrations u_j. The weight is w_{ij}, the neighbouring edges of vertex i are denoted $i \sim j$, and the area associated with i is A. The choice of weight determines the behaviour of the discrete Laplacian. Unfortunately, Wardetzky et al. (2007) have shown that no discretization maintains all the properties of the continuous Laplacian. The most common weighting used is the cotangent Laplacian,

$$(\nabla^2 u)_i = \frac{1}{2A} \sum_{i \sim j} (\cot \alpha + \cot \beta)(u_i - u_j). \tag{3.10}$$

A vertex i and a neighbouring vertex j are connected by an edge coloured black in Fig. 3.2. The cotangent weight for the black edge is computed from the angles α and β, which are opposite from the edge. This weight can be derived from the ratio of the edge length of the dual cell, shown in red, and the black edge's length. The black edge length is inversely proportional to the magnitude of the morphogen gradient between vertices i and j. The length of the red edge represents how much of an interface between the area associated with i and the area associated with j exists. Consequently, the amount of diffusion increases when the length of the red edge increases. To find the cotangent Laplacian for the entire mesh, we evaluate Eqn. 3.10 at each vertex. A rigorous derivation is given in (Crane et al., 2013; Herholz, 2013). The drawback of this Laplacian compared to the continuous version is that the cotangent weights can be negative, which occurs when angles are greater than 90°. Consequently, care must be taken when meshing the domain.

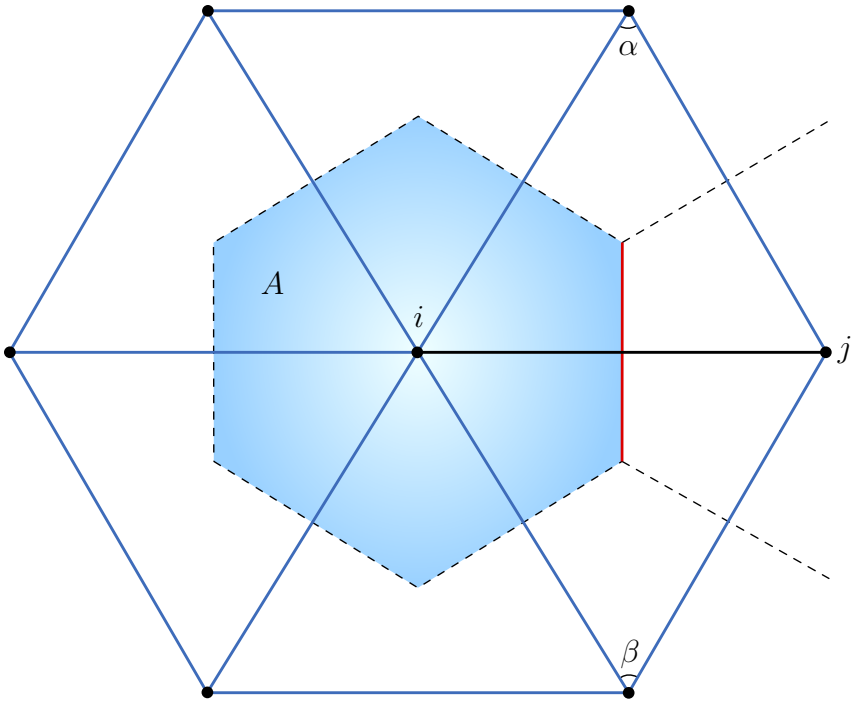


Figure 3.2 – A vertex i and its dual area A. The weighting for the cotangent Laplacian between vertex i and j is dependant on the length of the red and black edges.

Anisotropic Diffusion on Meshes

Computing anisotropic diffusion on a mesh requires Laplacian weights that reflect the change in diffusivity for a given direction. This change is computed by applying a diffusion tensor, D, to the gradient of the morphogen field (Eqn. 3.1). The diffusion tensor at a vertex can be visualized as an ellipse (Fig. 3.3a). The length of the ellipse's axes correspond to the eigenvalues in the diffusivity matrix Λ (Eqn. 3.4). To properly orient Λ, we need a vector denoting the dominant direction of diffusion. For a given vertex, i, a vector, $\vec{\boldsymbol{d}}_{\boldsymbol{i}}$ is specified which provides a notion of direction on the mesh. However, morphogen gradients are defined on triangle faces. To calculate a diffusion direction for a given face, the angle weighted average of its vertex directions, $\vec{\boldsymbol{d}}_{\boldsymbol{i}}$, $\vec{\boldsymbol{d}}_{\boldsymbol{j}}$, and $\vec{\boldsymbol{d}}_{\boldsymbol{k}}$, is computed and projected onto the face to obtain $\vec{\boldsymbol{d}}_{\boldsymbol{ijk}}$ (Fig. 3.3b). The face normal is used with $\vec{\boldsymbol{d}}_{\boldsymbol{ijk}}$ to obtain a rotation matrix R and subsequently, D (Eqn. 3.5).

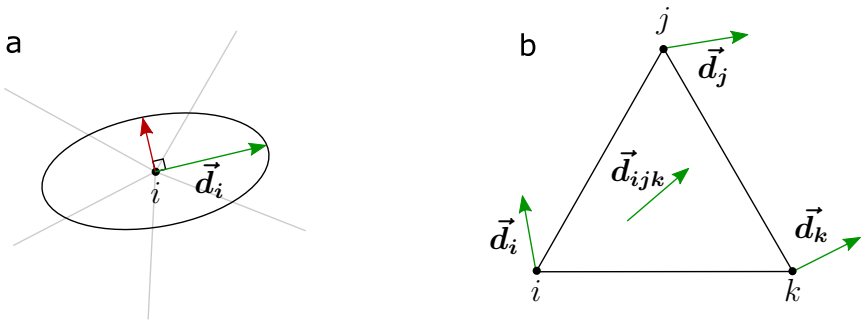


Figure 3.3 – Visualization of a diffusion tensor at a cell and diffusion direction on a face. **a:** the diffusion tensor at vertex i. The green arrow, $\vec{d}_i$, is the primary direction of diffusion and its length corresponds to the first eigenvalue value of Λ. The red arrow is perpendicular to $\vec{d}_i$ and its length corresponds to the second eigenvalue value of Λ. **b:** A triangle with diffusion vectors at each vertex and the resulting face vector.

The morphogen gradient on a face is determined by the sum of the face's edge normals. An edge normal provides the gradient direction and is weighted by the morphogen concentration opposite to that edge. To find an edge normal, we apply a 90° rotation matrix, Q, about the face normal, to the edge. Q is multiplied with D to include this rotation:

$$D^{\perp} = Q^T D Q. \tag{3.11}$$

Using $D^{\perp}$ we calculate the Laplacian weighting and obtain the discrete anisotropic Laplacian:

$$L_{ij} = \begin{cases} -\frac{1}{2A}\sum_t \frac{D^{\perp}e_i}{\|e_i\|} \cdot \frac{e_i}{\|e_i\|}(\cot\alpha + \cot\theta) & \text{if } i = j, \\ \frac{1}{2A}(\Gamma\frac{\cos\gamma}{\sin\alpha} + \Xi\frac{\cos\xi}{\sin\beta}) & \text{if } i \sim j, \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

This matrix is used to compute diffusion when given a vector of morphogen concentrations, $\boldsymbol{u}$, by:

$$L\boldsymbol{u} = \nabla^2\boldsymbol{u}. \tag{3.13}$$

This Laplacian is more complicated than before, and a complete derivation is given by Andreux et al. (2014). The parameters correspond to Fig. 3.4. i and j are vertices, and e_i

and e_j are edges, as shown in Fig. 3.4. t is a triangle that shares vertex i. $\Gamma = \frac{\|D^{\perp}e_j\|}{\|e_j\|}$ and γ is the angle between e_j and $D^{\perp}e_j$. If $D^{\perp} = I$ then $\gamma = \alpha$ and we obtain isotropic diffusion. Ξ and ξ are the same quantities measured on the adjacent triangle with that triangle's $D^{\perp}$.

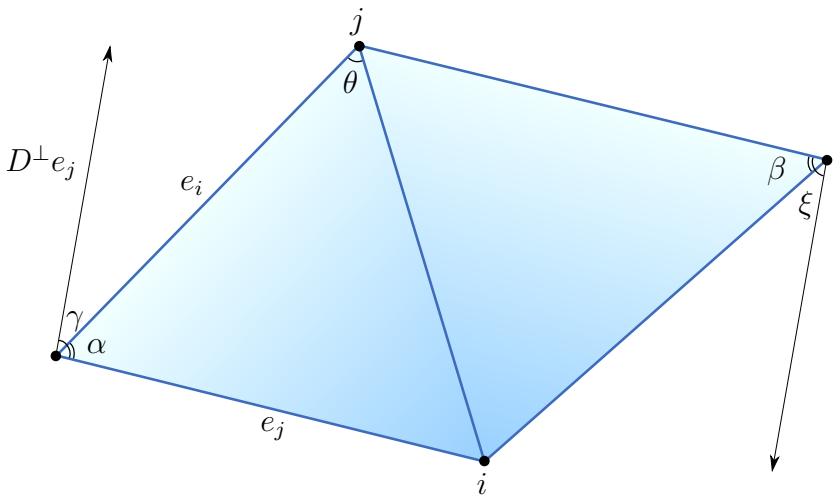


Figure 3.4 – Two triangles with their angles and edges associated with the anisotropic cotangent Laplacian.

3.5 Boundary Conditions

PDEs determine the state of the simulation inside the domain. Interaction between the simulation and the world outside the domain is based on boundary conditions. Dirichlet and Neumann are the two most common boundary conditions. Dirichlet enforces the morphogen values directly at the boundary, and Neumann specifies the rate of change of the morphogens across the boundary. LRDS uses Neumann with a value of zero by default or Dirichlet if specified.

3.6 Systems with Dynamic Structure

In nature, domains grow over time, which can have an appreciable effect on pattern formation. One of them is a diluting effect on chemical concentrations that make up a pattern. When simulating directly on meshes, the face's area represents a minimum size of pattern

detail that can be represented. Thus, it is important to consider the increase in face size from growth. A subdivision algorithm is used to split large triangles into smaller ones to address this. However, subdividing every face in the mesh causes small triangles to be subdivided unnecessarily, and the total number of triangles increases quickly. This is a problem because the simulation's performance decreases as the number of vertices increases.

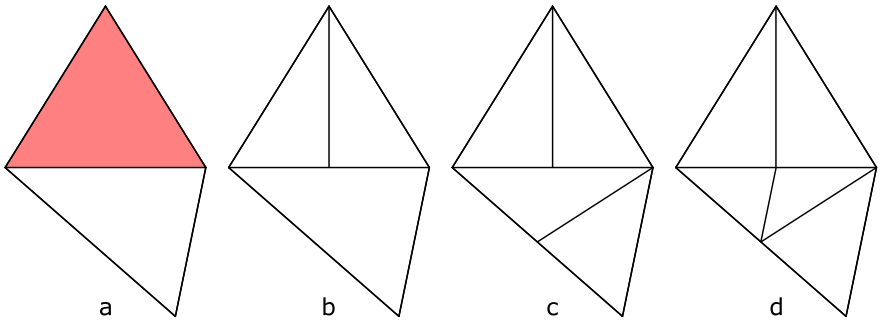


Figure 3.5 – Subdividing a face, shown in red, using longest-edge bisection. **a:** subdivision starts with the red triangle. **b-d:** the progression of algorithm.

Adaptive subdivision is a technique used to only subdivide triangles with an area larger than a given threshold. This approach allows all triangles to maintain a similar area and limits the unnecessary creation of triangles. The choice of adaptive subdivision algorithm used determines the shape of the generated triangles. If the internal angles of a triangle exhibit large deviations from 60°, it can pose a problem for the simulation. The number and magnitude of deviations in a mesh give an informal notion of mesh quality and affects the simulation. Depending on the angle, the cotangent weights can be negative or widely varying due to the behaviour of cotan around 0 and 180 degrees. Negative cotangent weights cause diffusion to move morphogens from low to high concentrations incorrectly (Wardetzky et al., 2007).

To obtain optimal incremental triangulation that tends to produce good quality triangles, we use longest-edge bisection (Rivara and Inostroza, 1998). In this method, faces are only ever subdivided with respect to their longest edge. The red triangle in Fig. 3.5a is too large and must be subdivided. In Fig. 3.5b, the red triangle has been subdivided, but the new edge does not connect to an edge in the adjacent face. To solve this, we also subdivide the

adjacent face along its longest edge. This may fix the issue, or it may cause another triangle to be missing an edge. We proceed by subdividing every face missing an edge in the same way until all the edges are connected. This process is shown in Fig. 3.5 and the algorithm is detailed in Alg. 1.

Subdivision also requires the proper handling of concentrations. The concentration assigned to the new vertex is the average of the neighbouring concentration values that shared the edge.

Algorithm 1: An algorithm to recursively subdivide a triangle and its neighbours based on (Rivara and Inostroza, 1998).

```
Input: Triangle t0
Result: Triangle is subdivided along its longest edge. Adjacent triangles are
        recursively subdivided to share edges.
subdivide(Triangle t0)
    Edge e0 = getLongestEdge(t0)
    bool subdividing = hasAdjacentFace(e0)
    subdivideFace(t0, e0)
    while subdividing do
        if hasAdjacentFace(e0) then
            Triangle t1 = getAdjacentFace(e0)
            Edge e1 = getLongestEdge(t1)
            if e1 == getPairEdge(e0) then
                subdivideFace(t1, e1)
                subdividing = false
            else
                subdivide(t1)
            end
        else
            subdividing = false
        end
    end
end
```

3.7 Numerical Methods

The simulation is advanced by taking small steps through time from a given initial condition. I use the forward Euler method to perform this integration. A reaction-diffusion formula, similar to Eqn 2.1, integrated with forward Euler is:

$$\boldsymbol{x_{i+1}} = \boldsymbol{x_i} + (D\nabla^2 \boldsymbol{x_i} + F(\boldsymbol{x_i}))\Delta t. \tag{3.14}$$

Here $\boldsymbol{x_i}$ is a vector of scalars representing morphogen concentrations at step i and Δt is the time-step. D is diffusivity, ∇^2 is the discrete Laplacian used to compute diffusion, and $F(\boldsymbol{x_i})$ encapsulates the reactions of the system. This method suffers from inaccuracy at larger time-steps because we are assuming $\nabla^2 \boldsymbol{x_i} + F(\boldsymbol{x_i})$ is constant for the whole time-step. This inaccuracy can cause instability in stiff equations by accruing error with each simulation step. Semi-implicit integration schemes (Nie et al., 2006) allowing for larger time-steps, but in practice, it is possible to use small enough time-steps to minimize inaccuracy with forward Euler.

Chapter 4

Software Design and Implementation

4.1 General Requirements

LRDS was designed to fulfil six main requirements.

1. To simulate reaction-diffusion on grids and arbitrary triangulated surfaces. Grids provide a common and simple domain for simulating reaction-diffusion. Meshes afford a more realistic domain because they are not restricted to the plane and are better suited for growth.

2. To allow for interaction with simulation parameters at runtime. Real-time interaction allows for rapid iteration during model creation and pattern exploration. Minimizing delays increases user enjoyment and productivity.

3. To visualize pattern formation over time. Watching a pattern's development allows the user to gain an intuition about simulation behaviour. Also, patterns seen in nature are not necessarily those at a steady state.

4. For easy modification of parameters and the equations themselves. Allowing configurable PDEs dramatically increases the usefulness of LRDS as users can create custom

models. Another benefit of configurable PDEs is that not all users may have access to the program's source code.

5. To support growth, spatially varying parameters, and anisotropic diffusion. These features are essential because their impact on pattern development can lead to more biologically relevant models.

6. For the ability to track the incremental changes made to models. A track record of a model's changes allows the user to see its developmental process. This history provides a list of the avenues examined during the model's creation and future areas of interest.

4.2 Program Architecture

LRDS was implemented on the Windows 10 operating system using C++ and OpenGL. The graphical user interface was developed with the library "Dear ImGui" (Cornut, 2019).

LRDS contains two important abstract classes: **SimulationDomain** and **Simulation**. The former is a class which abstracts the concept of a domain. It contains an array of morphogen concentrations and all the domain-specific functionality that can be performed without knowledge of the spatial relationships within the domain. Pure virtual functions such as Laplacian and gradient are declared in **SimulationDomain**. However, these functions must be implemented by subclasses of **SimulationDomain**. An example subclass is the **HalfEdgeMesh** class, which implements the Laplacian function using Eqn. 3.12 defined in Chapter 3. The **SimulationDomain** base class provides an extensible way to add support for other domain types without having to copy non-domain specific code. The reaction-diffusion models are represented by extending the **Simulation** class. This abstracts and provides all the functionality relevant to the simulation except the PDE formulas. Simulation also contains a **SimulationDomain** member variable, which is how a reaction-diffusion model is associated with a domain. **GPUSim** and **CustomSim** extend **Simulation** and are the two main classes which provide the link between LRDS and the simulation module.

This architecture is shown in Fig. 4.1.

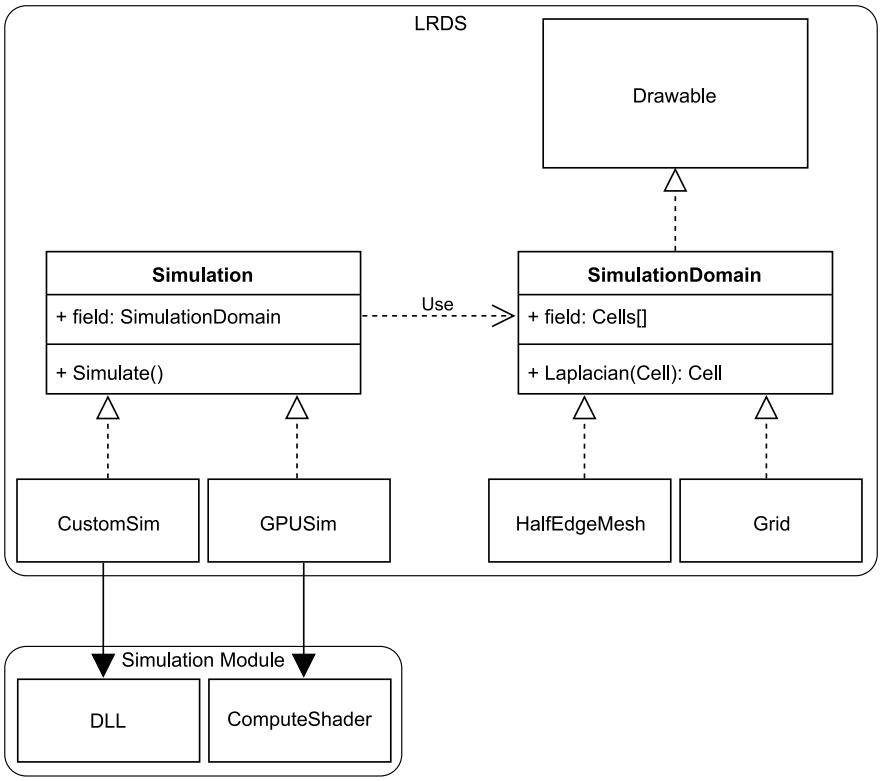


Figure 4.1 – UML diagram representing LRDS's architecture.

4.3 Simulation Creation

Upon start-up, LRDS parses user-provided command-line arguments and reads in a parameter file. The possible command line arguments are described in A.1. The label-value pairs in the parameter file are used to create a symbol table. Then, the window is created along with the camera and scene objects responsible for rendering. Next, an instance of the class **SimulationDomain** is created. This domain instance and symbol table are then used to create a **Simulation** object. The symbol table determines if the simulation module is a DLL or compute shader. After creation, the initial conditions, and boundary conditions are applied from the parsed parameters to the simulation. Next, the simulation and domain instances are added to the scene so they can be updated and rendered. The remainder of the

simulation executes a loop that calls the scene's update and render functions. The update function invokes the simulation module, which computes the PDEs.

4.3.1 GPU Acceleration

Due to the popularity of video games, GPUs have become cheaper and more common. They contain many more cores than CPUs, enabling them to perform highly parallelized processing of triangles and pixels at a much faster rate. With the advent of compute shaders in modern graphics APIs, the power of the GPU can be leveraged to perform general-purpose computation.

In LRDS, these shaders are written in the OpenGL Shading Language (GLSL). Since the evaluation of reaction-diffusion equations depends solely on the previous simulation state, reaction-diffusion is easily parallelized. A single execution of the shader evaluates the equations once at a single location on the domain. For an entire step of the simulation, the compute shader is executed conceptually in parallel, across all the GPU cores, until all locations have been processed. Representing the half-edge data structure on the GPU can be done the same way as in RAM, by using structs and arrays. In practice, a simulation step will processes groups of cells in parallel. Each group of cells has access to a small amount of memory. This memory is fast to use but may not be large enough to hold our domain, especially if growth is involved. General-purpose GLSL arrays that are not limited in size are named Shader Storage Buffer Objects (SSBOs). SSBOs are slower to use but allow for arrays big enough to hold all the simulation data. To allow for growth, I allocate SSBOs larger than first needed, providing extra capacity for the simulation to grow. In my GPU half-edge data structure, pointers are replaced by index offsets into their respective SSBOs.

A drawback of compute shaders is the slow transfer rate between RAM and GPU memory. This problem arises with frequent or substantial data transfers. The data are usually the result of algorithms that cannot be parallelized and thus are computed on the CPU. The results are then transferred to the GPU for further processing. The reverse case can also

occur; results are generated on the GPU and require further computation on the CPU. After the CPU is done, the data may then be transferred back to the GPU. Performance loss due to transferring occurs in LRDS due to the recursive nature of the subdivision algorithm used. The CPU is used to perform subdivision when a face becomes too large. Then the GPU is updated. Care has been taken to only transfer the changes subdivision has caused, avoiding performance drops. Nevertheless, if significant changes that affect many locations on the domain are made, users may experience a pause as data synchronizes between GPU memory and RAM.

4.4 Parameter File

When starting the program, the user may specify a configuration file from the command line. If no file is specified, the program will look for "SimConfig.txt". The parameter text file is a customizable model specification. An example parameter file is shown in Fig. 4.2. Text prefixed by a hash symbol denotes a comment. Comments have no effect on the simulation and are used for documentation. A parameter is defined by a label-value pair delimited by a colon. The user can define any number of parameters that do not contain reserved labels. A complete listing of reserved labels is provided in A.2. The *domain* parameter specifies a mesh or grid domain. A Wavefront OBJ filename specifies a mesh domain. Alternatively, "grid" specifies a grid domain. The grid's resolution is defined with the labels *xRes* and *yRes*, and the distance between cells is defined by *cellSize*. The *morphogens* parameter allows the user to specify the names in uppercase, of the morphogens involved in the simulation. These uppercase names are used when defining the initial conditions and PDEs.

User-defined parameters can be declared under the *params* label. Parameters are written inside a pair of curly brackets. The cell indices associated with the parameters are denoted by the *indices* label. In grid-based domains, cells are stored in columns with index 0 representing the bottom left cell. Mesh cells are indexed by the order in which they are declared in the

OBJ file. Valid indices are comma-separated integers, a range defined with a hyphen, "all", or "boundary", which only identifies the boundary indices. A random selection of m indices can be requested using $rand(m)$. If the user wishes each index to represent a larger area, an integer radius can be specified with *radius*, which corresponds to an n-ring region around the cell. For a given cell, a 1-ring region corresponds to the eight adjacent cells on a grid and the cells that can be reached within one edge on a mesh. Multiple entries can be made within the curly brackets to define non-homogeneous parameters. This allows for different sets of indices to be associated with different parameter values.

The *initialConditions* parameter specifies the morphogens associated with each index at start-up. Like *params*, the *initialConditions* are declared in a pair of curly brackets. After the indices are defined, the morphogen concentrations are declared. This can be a number or a random selection from a range of values in $[n, m)$. The latter is specified by using $rand(n, m)$, where n and m are floating-point values. The *simFile* label is used to start the model from a saved simulation state. It accepts a filename with a per-index entry for each morphogen concentration, anisotropic vectors, and principle diffusion rates.

The *boundaryConditions* parameter precedes a pair of curly brackets that determine the PDE behaviour at the domain boundary. First, a set of cell indices is defined. Next, the behaviour for each morphogen is defined. These behaviours only affect the cells defined. The two options are Neumann set to 0 (the default setting) or Dirichlet.

The *rdModel* parameter can be assigned a value of CPU or GPU depending on the desired computation mode. PDEs are declared within curly brackets that follow the *rdModel* parameter. These equations are declared in either GLSL or C++, depending on the computation mode. In the PDEs, user-defined parameter values are accessed by pre-pending "params."[1] to their name. Current morphogen values are used by referencing their names in lower-case. Equation splitting is used to evaluate diffusion before the PDEs are calculated. An array L is predefined and contains the results of the Laplacian of the current morphogen

[1] note the trailing period.

field. When using the GPU, this array is big enough to hold a value for each morphogen at the current cell, and when using the CPU, it is big enough to hold values for every cell and their morphogens. When defining the PDEs, the user only needs to access L with the upper-case morphogen name. This name is then converted to the proper index when the simulation module is created. Similarly, morphogen values are updated by using upper-case morphogen names and writing to a predefined "new" array. An example parameter file is shown in Fig. 4.2.

```
# The following parameters correspond to the Gray-Scott model
domain: icosphere.obj
colorMap: color.map
morphogens: A, S
params:
{
  indices: all
  dt: 0.1
  Da: 0.0005
  Ds: 0.001
  f: 0.03
  k: 0.063
}
initialConditions:
{
  indices: 0-641 # Example of index range, could also be "all"
  A: 0
  S: .5 + rand(0, .5)

  indices: rand(10)
  radius: 1
  A: 1
  S: 1
}
boundaryConditions:
{
  indices: boundary
  A: dirichlet
  S: dirichlet
}
rdModel: GPU
{
  float saa = s*a*a;
  new[A] = a + (params.Da * L[A] + saa - (params.k + params.f) * a) * params.dt;
  new[S] = s + (params.Ds * L[S] - saa + params.f * (1 - s)) * params.dt;
}
```

Figure 4.2 – Example parameter file for the Gray-Scott model. The domain referenced is a unit icosahedron mesh. The mesh has no boundaries. However, boundary conditions are included for completeness.

4.5 User Interface

In conjunction with speed, the integration of a GUI made LRDS an excellent tool for interactively exploring diverse reaction-diffusion models. Exploring different parameter values at runtime is achieved by using graphical control panels shown in Fig. 4.3. They have controls for growth, rendering, and interactive non-homogeneous parameter specification.

Parameters are shown in textboxes generated from the symbol table. Initially, changes to these parameters affect all indices. For more control, subgroups of indices can be selected through painting. Groups of indices with the same set of parameters are given an ID number and can be selected by cycling through numbers with the "Params" textbox.

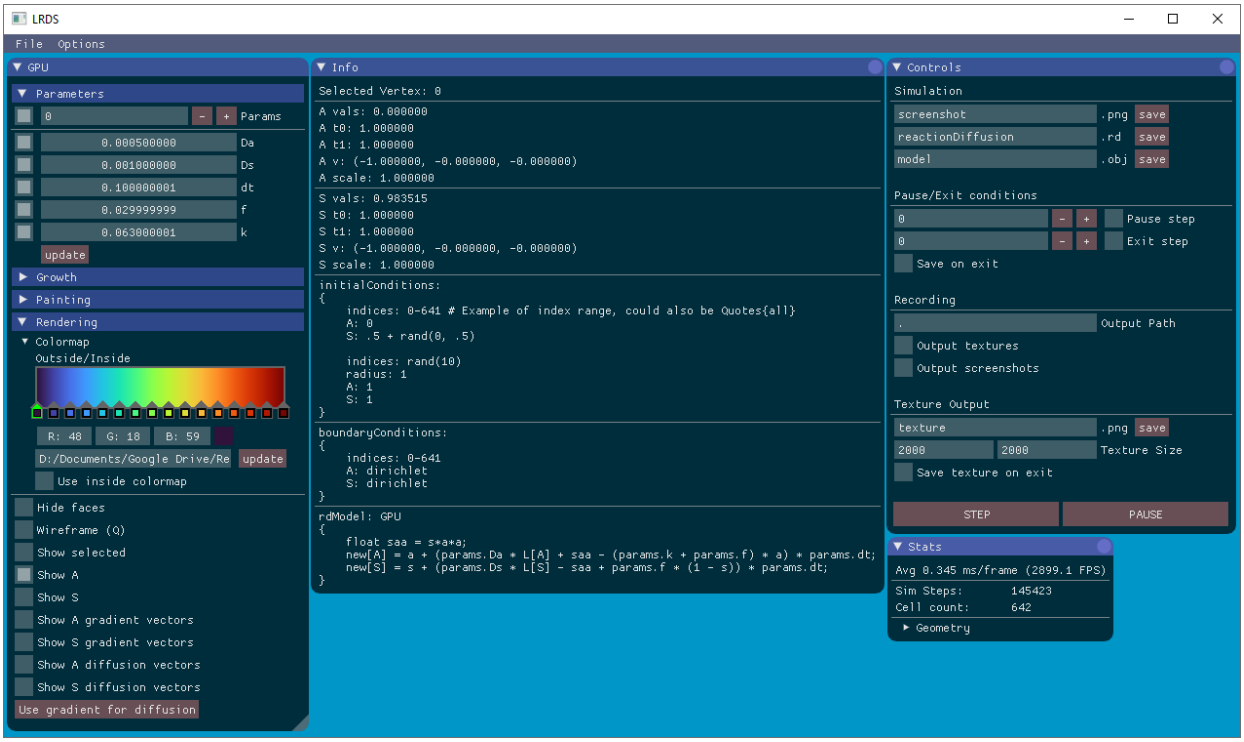


Figure 4.3 – The available control panels. **GPU:** The control panel for modifying parameters as well as collapsible menus for controlling growth, painting, and rendering behaviour. **Info:** information about the model such as the PDEs used and selected vertex attributes. **Controls:** simulation controls allow for saving screenshots and textures. **Stats:** program statistics are shown such as the time per frame, cell count, and total area of the domain.

When specifying parameters locally on a mesh, the user selects a painting mode from the control panel and right-clicks on the mesh to change the values at vertices within a

given radius. When the left mouse button is clicked, and if a mode is selected, a sphere is projected onto the mesh using a raycast. The cell closest to the sphere centre is used to find the vertices that are also affected by the painting operation. From the closest cell, subsequent rings are checked to see if they reside in the sphere. This continues until no more vertices are found. There are three painting modes. The first painting mode is "selection," which allows the user to select groups of vertices and set parameters in bulk. The second painting mode is "morphogen," which allows the user to add to or set the morphogen value at the painted cells. The third painting mode is "anisotropic diffusion," which allows the user to paint the direction of diffusion and the principle diffusion rates. The previous raycast location is recorded and used with the current location to determine the direction the cursor is moving when painting anisotropic diffusion (Fig. 4.4). The principle diffusion rates are specified in the control panel. When painting morphogens or diffusion, the quantity at each cell is modified with a linear falloff from the centre of the sphere.

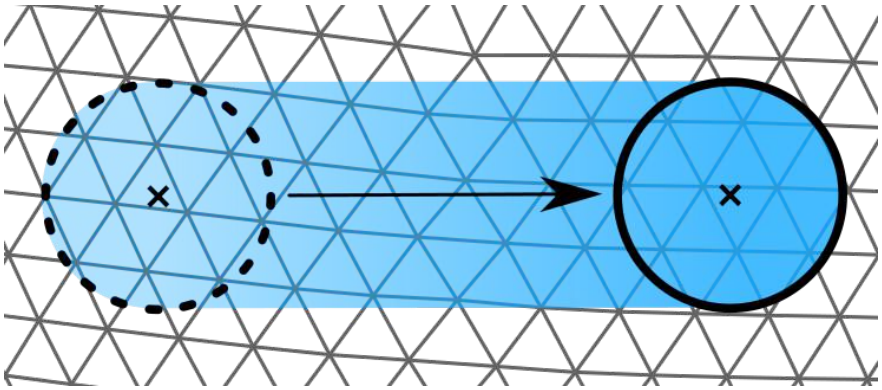


Figure 4.4 – Painting direction is determined by taking the difference in cursor positions from consecutive frames. The **X** inside dashed circle is the initial cursor location, and the **X** inside the solid circle is the current cursor position. Vertices in the blue area are affected by the paint operation.

The sphere radius can be adjusted from the control panel or with the mouse scroll wheel. If not painting, the mouse scroll wheel controls the camera zoom. The left mouse button can be used to rotate the model, and right-click translates the model. Domain orientation and position can be reset by pressing the 1 key or through the control panel. The camera can be moved left, right, up, down, in, and out with the W, A, S, D, R, and F keys, respectively.

4.6 Visualization

The user has a choice of visualizing different morphogen concentrations by actual or normalized value. In the latter case, every morphogen concentration is divided by a user-provided value. The concentration is mapped using a colormap to determine the colour to represent the concentration. A separate colormap can be specified for each side of the domain. This feature can be used to hide the pattern on one side of the mesh. Concentration gradients and the vectors driving anisotropic diffusion can also be visualized as lines (representing vectors) extending from their corresponding vertices. These lines are coloured black at their vertex and transition to green for diffusion vectors or red for gradient vectors (Fig. 4.5). Wireframe mode allows users to see the underlying triangular geometry of the domain. The mesh rendering is enhanced by diffuse lighting to highlight its shape. When selected vertices are visualized, the unselected ones will appear faded. Users can export the generated pattern as a texture for higher quality rendering of models through other software. Exporting requires that the domain mesh comes with texture coordinates. A blank texture is then coloured using the domain mesh texture coordinates.

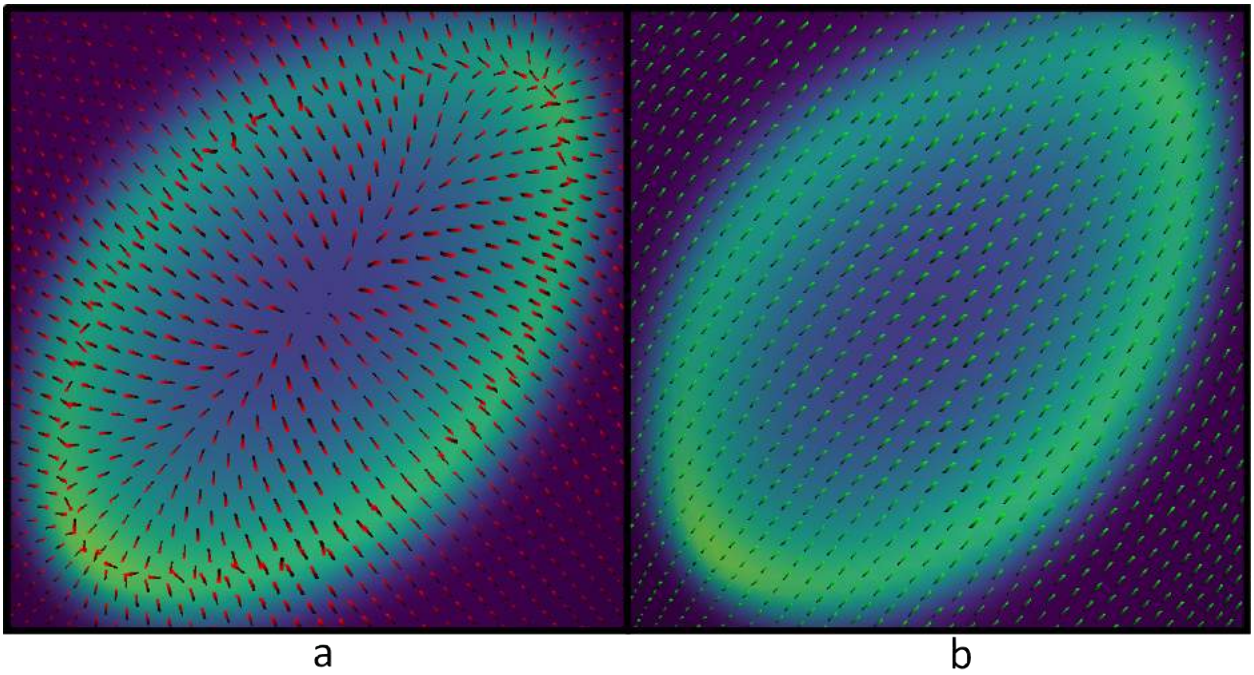


Figure 4.5 – Visualization of vector fields. **a**: The pattern gradient is visualized by lines fading from black to red. **b**: The vector field driving anisotropic diffusion is shown by lines fading from black to green.

4.7 Saving and Loading

After the simulation has loaded, the user can save the simulation at any point, preserving the state of the simulation after pattern development. Another use for saving is if a specific vector field or morphogen configuration is desired, which otherwise might be tedious to define directly with indices. Saving creates a set of files containing the program and simulation state. These files include the colour map, Wavefront OBJ file, and the parameter file. Another created file is "EditorSettings.txt", which contains settings not integral to the simulation's behaviour such as background colour and cursor radius. The saved state of the program is in a ".rd" textfile. This file contains a header that specifies the number of morphogens on the first line and the number of cells on the second line. After the header, the following n lines correspond to the n cells in the domain. The first line is the 0th indexed cell and contains a list of floating-point values separated by spaces. For a model with

morphogens A and S, a line would look like:

$$a\ s\ x_a\ y_a\ z_a\ x_s\ y_s\ z_s\ \lambda_{1a}\ \lambda_{2a}\ \lambda_{1s}\ \lambda_{2s}\ d_a\ d_s.$$

Parameters a and s are the morphogen concentrations and entries with subscripts a and s belong to A and S morphogens respectively. x, y, and z are the components of a cell's anisotropic direction vector. λ_1 and λ_2 are the diffusivities and d is the diffusion scale. A user can load a simulation by specifying the individual files to be used in the parameter file. Using a ".rd" file overrides the model's initial conditions.

4.8 Model Exploration

Designing and exploring a model's pattern forming potential can be challenging. Tracking progress as incremental changes are made is a requirement of model creation with LRDS. To satisfy this, I used the Git version control system. I commit the files associated with a model to a repository periodically during exploration. This allows for easy reproduction of previous patterns and avoids duplication of past efforts. The greatest benefit from this workflow is that the progression through a multi-dimensional parameter space is tracked, providing a mapping of what has previously been explored. Fig. 4.6 shows a visualization of the version control history.

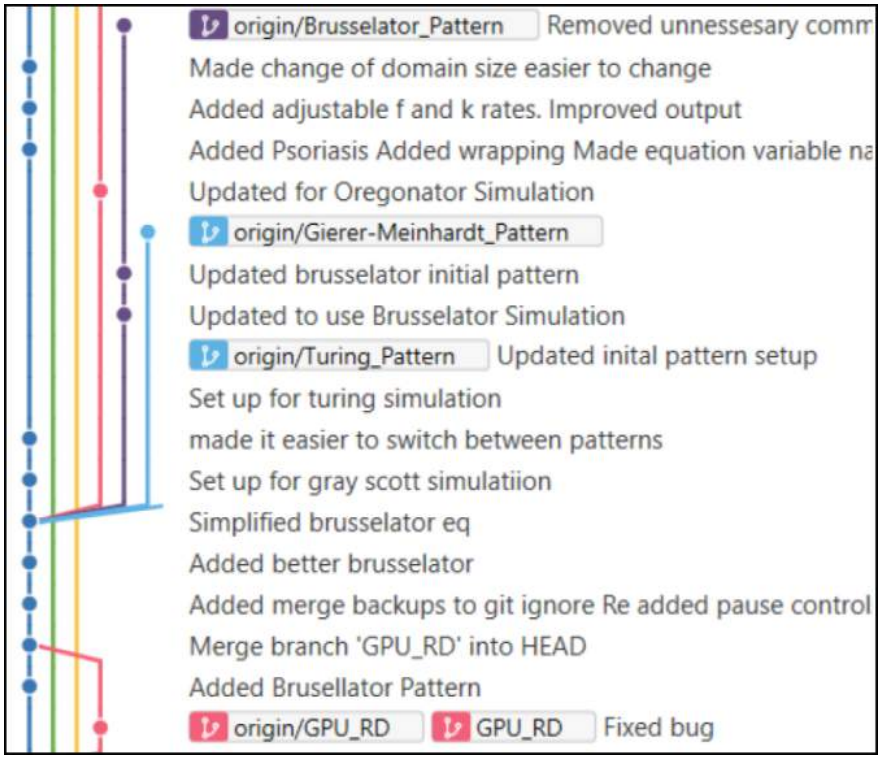


Figure 4.6 – Visualization of the Git repository by the Sourcetree software (Atlassian, 2019). Coloured vertical lines represent models. A dot on the corresponding line represents the state of the model, and a user-provided comment appears on the right. Models derived from others have their vertical line connected by a horizontal line leading to the parent.

Chapter 5

Case Study 1: Ladybug Patterns

5.1 Literature Review

The ladybug (also known as ladybird beetle or lady beetle) is an insect in the family Coccinellidae. They are a small round beetles that range in length from 0.8 to 18mm (Glisan King and Meinwald, 1996) and are often found in leaf piles and gardens. Ladybugs exist throughout the world and display a myriad of spot and stripe pigmentation patterns on their elytra: the two symmetric hard shells on the dorsal side of the insect (Fig. 5.1). The elytra's main purpose is to protect the fragile wings located underneath, and the pattern is thought to deter predators by indicating that the ladybug is bitter tasting (Glisan King and Meinwald, 1996). The pattern on one elytron is a mirror image of the pattern on the other.

Understanding the ladybug life cycle can give insight into how and when their patterns form. The cycle starts with eggs laid on the underside of leaves. These eggs hatch into larvae, which then eat aphids and other food sources until they can pupate and metamorphose into adults. Immediately after pupation, the elytra appear patternless and are a pale-yellow colour. In a timespan of hours to days, dark spots emerge and become black. The yellow transitions to red, giving the ladybug its characteristic appearance. Although certain species of ladybugs are described by the number of spots on their elytra, there can be a variable

Figure 5.1 – A selection of *H. axyridis* ladybugs displaying various spot patterns. © *2019 by Entomart, used with permission.*

number and shape of spots found on ladybugs of the same species.

Ando et al. (2018) explored the genetic mechanisms governing the formation of ladybug patterns. They found a gene, *pannier*, which is responsible for much of the observed pigmentation patterns in *H. axyridis* and *Coccinella septempunctata*. Before any pigment is visible, *pannier* is found in a pre-pattern on the elytra. It then promotes melanin (black) and inhibits carotenoids (red) pigmentation accumulation creating a visible pattern. Future work is needed to identify if other specific genes are involved in pigmentation expression.

Although the specific genetic mechanisms behind pattern formation are not fully known, Liaw et al. (2001) used reaction-diffusion to simulate visually similar ladybug patterns. The

equations used are activator-depleted substrate (Eqn. 2.4) with saturation. Their results were obtained numerically by using forward Euler integration on a grid located on the surface of a partial sphere. A Laplacian defined in spherical coordinates was used to compute diffusion on partial hemispheres of radius 1. It was found that the domain boundary and the curvature change the final position of spots and stripes. Five models were proposed. Three species considered display black spots on a red/orange background. In particular, *Platynaspidius quinquepunctatus*, *Coccinella septempunctata*, and *Epilachna crassimala* display 5, 7, and 10 spots respectively. *Macroilleis hauseri* has brown stripes aligned with the long axis of the shell on a yellow background. *Bothrocalvia albolineata* displays elongated orange loops on a brown background.

5.2 Model Description

I used LRDS to improve these models. The equations remain the activator-depleted substrate formula (Meinhardt, 1982):

$$\begin{aligned} \frac{\partial u}{\partial t} &= \rho_u \frac{u^2 v}{1+\kappa u^2} + \sigma_u - \mu_u u + D_u \nabla^2 u, \\ \frac{\partial v}{\partial t} &= -\rho_v \frac{u^2 v}{1+\kappa u^2} + \sigma_v + D_v \nabla^2 v. \end{aligned} \tag{5.1}$$

The activator is represented by u and is displayed as the pigmentation on the elytra. The substrate is denoted by v. The rate of conversion of v into u is determined by the reaction rate ρ_u. Similarly, ρ_v represents how much v is used in the conversion to u. κ controls saturation, σ_u and σ_v are the base production rates, and μ_u is the decay of u. ∇^2 is the discrete Laplacian with the diffusion rates being D_u and D_v for u and v. The boundary conditions are no-flux, except for the model of *B. albolineata*, which contains a sink along the middle of the domain where the two shell halves would meet.

5.3 Results

By simulating ladybug patterns on a mesh, I improved on the results in (Liaw et al., 2001). Due to the mesh's flexible representation of 2D surfaces, my model provides a more faithful portrayal of ladybug elytra compared to a partial sphere (Fig. 5.2). A couple of aspects of patterning contributed to the quality of the results. The pattern shape, such as spots or stripes, was essential, and the previous work provided this. Pattern colouration was important as I observed a substantial qualitative increase in patten realism by using images of real ladybug specimens to determine the pattern colours. The pattern's positioning on the mesh, especially for spots, was a necessary pattern feature. It was difficult to predict the final spot positions before the simulation reached a steady-state. Final spot locations often varied based on small changes to the specific initial distribution of morphogens. Being able to run the models quickly helped when picking a morphogen distribution. Fig. 5.3 shows the development of the simulated patterns. During the simulation of spot patterns, the initial pattern will disappear quickly and then reappear as large blotches. The blotches settle split into smaller ovals and then settle as spots. The original parameters provided by Liaw et al. (2001) have been altered to be more mathematically sound, and to account for the differences when simulated on a mesh. A complete list of model parameters is shown in Table 5.1. As noted by Liaw et al. (2001), a starting band of morphogens across the top of the elytra is important for spot alignment. In C, I also used a stripe at the top but ignored the central stripe. This alternative starting pattern produced rows of spots more predictably as individual spots migrated less during pattern formation. I also produced patterns on elytra that were joined in the middle and this produced the same patterns. This is probably due to the high amount of symmetry of the domain and starting pattern. A ladybug rendering is shown in Fig. 5.4.

Model	Species	D_u	D_v	κ	σ_u
A	*P. quinquepunctatus*	0.0005	0.035	0	0
B	*C. septempunctata*	0.0005	0.025	0	0
C	*E. crassimala*	0.0003	0.024 (0.04)	0	0.01 (0)
D	*M. hauseri*	0.000028	0.00168 (0.000168)	0.5 (0.35)	0
E	*B. albolineata*	0.000026	0.00182 (0.000182)	0.45	0.0019

Table 5.1 – Parameter values used for ladybug models on a mesh. The following parameters remain constant for all models $dt = 0.001$, $\sigma_v = 0.1$, $\rho_u = 0.18$, $\rho_v = 0.36$, and $\mu_u = 0.08$. The total number of steps is $1,500,000$ for all models except E where it has been decreased to $500,000$. Parameters in parenthesis are the original values used by (Liaw et al., 2001). Parameter values were changed due to small differences between the patterns formed on a partial sphere and a mesh. In my model of D and E, the parameter D_v has been increased by an order of magnitude to obey the rule that $D_v/D_u \geq 7.8$ (Liaw et al., 2001). The number of spots on C (*E. crassimala*) was less than ten. To rectify this, I lowered D_v from 0.04 to 0.024 and changed the initial morphogen distribution to a stripe at the top. σ_u was then increased from 0 to 0.01, allowing for horizontal lines to form which turn into spots over time. Another discrepancy observed with the initial parameters was that the stripes in D and E turned into spots and irregular lines near the boundary. In D, the initial morphogens propagate as a wave, leaving stripes in its wake. On a mesh, the wave was observed to outpace itself in places, causing it to self-interact and destroy the vertical line pattern. I have increased κ from 0.35 to 0.5, strengthening the tendency to form lines. I also changed the initial distribution from a vertical stripe of u down the centre to include stripes along the boundary (excluding the top). This has the effect of aligning the pattern by reducing the distance the middle wave must travel and avoiding the self-interaction. The total simulation steps was reduced to 500,000 from the original model's 1,500,000 steps to account for lines becoming spots in E. This could also be addressed by decreasing σ_u, which produces a pattern like D.

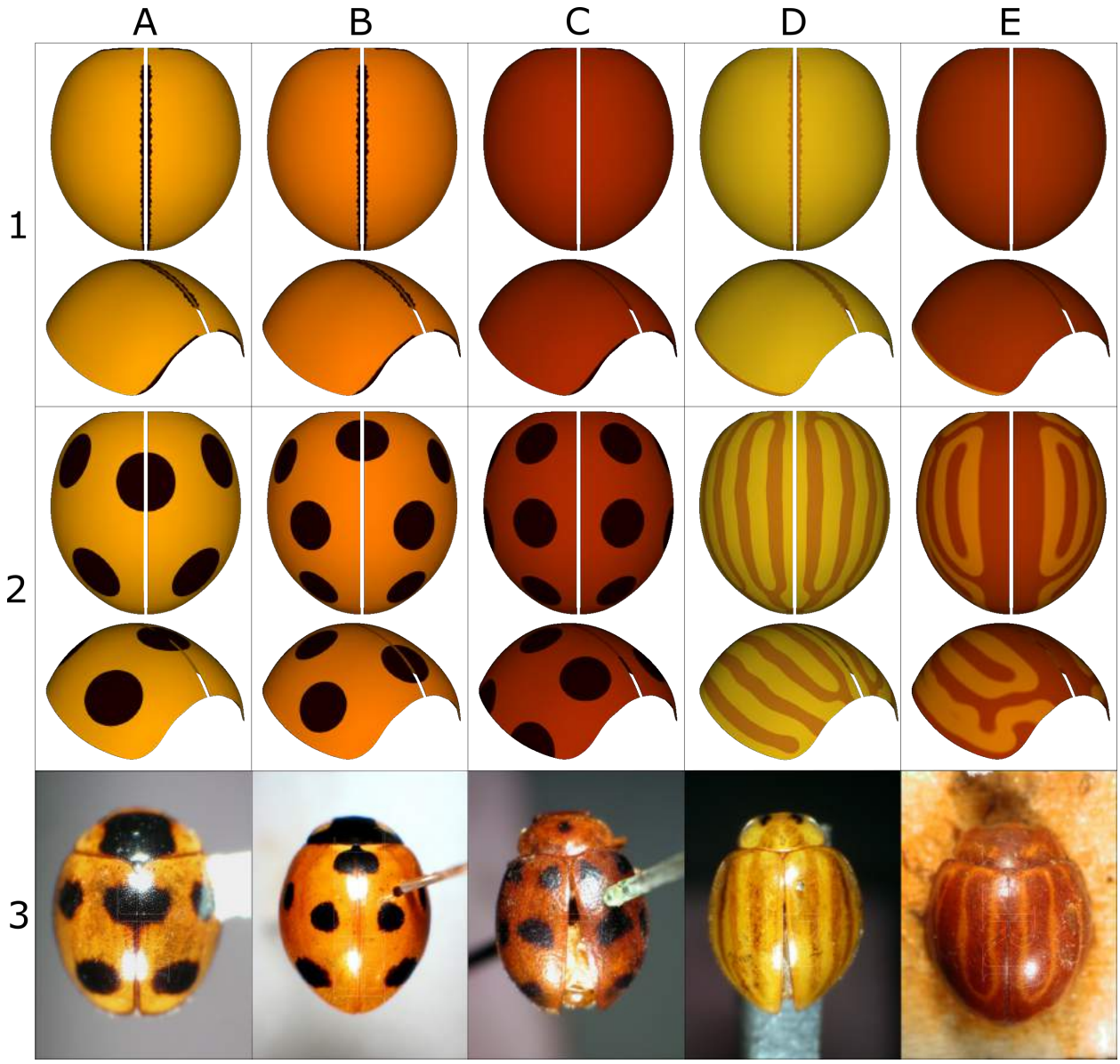


Figure 5.2 – Simulation of ladybug patterns. **Row 1:** initial simulation state. **Row 2:** final simulation state. **Row 3:** collection of the real ladybug species. **A-E:** *P. quinquepunctatus*, *C. septempunctata*, *E. crassimala*, *M. hauseri*, and *B. albolineata* respectively. *Photographs © 2008 by S.-P. Chen, used with permission.*

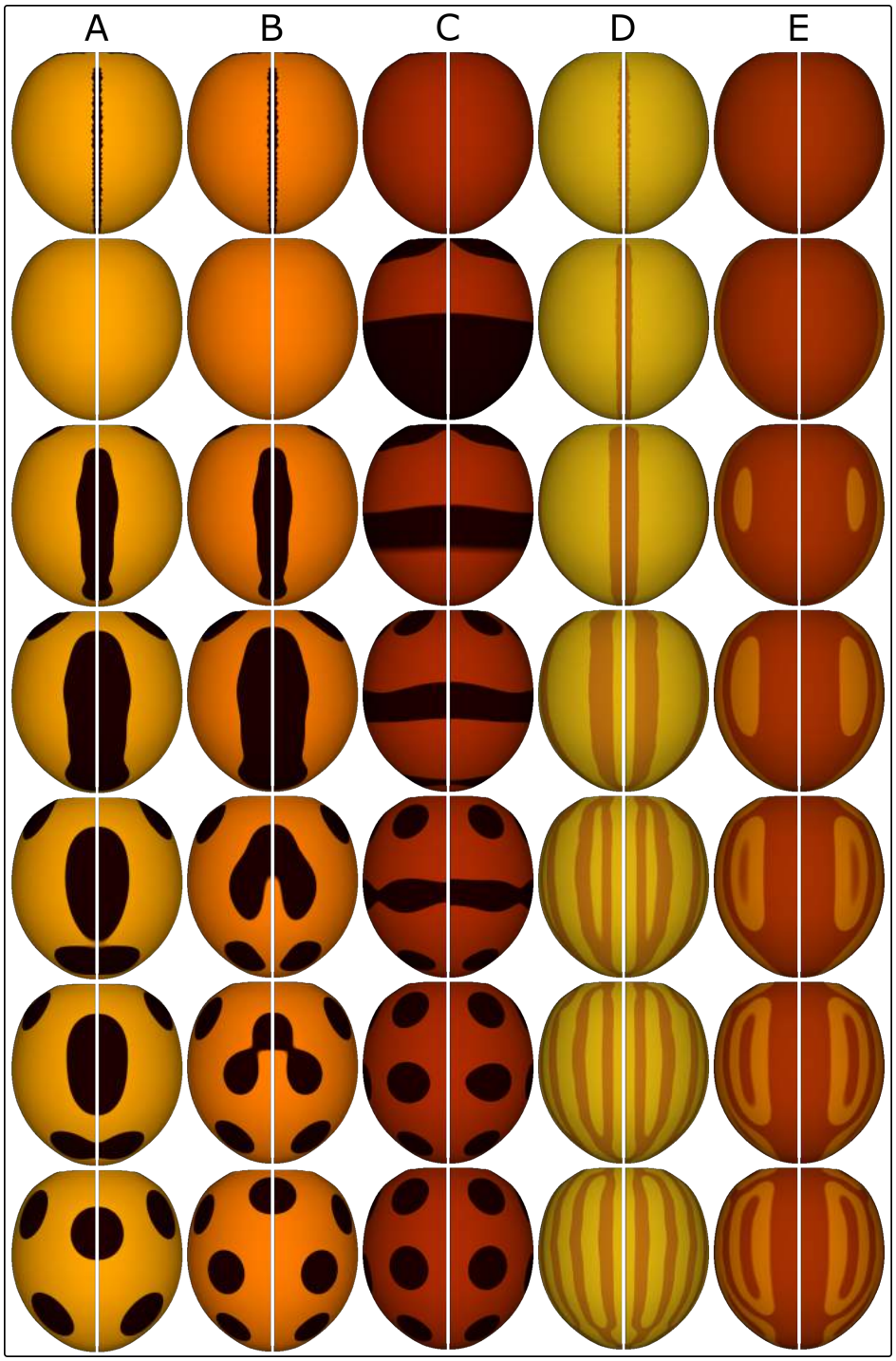


Figure 5.3 – Progression of ladybug patterns over time. **A-E:** *P. quinquepunctatus*, *C. septempunctata*, *E. crassimala*, *M. hauseri*, and *B. albolineata* respectively.

5.4 Discussion and Future Work

Improvements to these models can be made once the biological chemical interactions are fully understood. The most relevant insights would be the real initial distribution of morphogens, the actual reaction behaviour between morphogens, and the rate at which they diffuse throughout the elytra. Some ladybug species can control their pigmentation patterns depending on the season or the surface the ladybug is on. This suggests a simple reaction-diffusion model cannot represent all types of ladybug patterns (ins, 2009).

Further investigation should be made to determine how much the patterns found on the head of the ladybug effect the elytra pattern. Another critical question is if the patterns displayed are at a biological steady-state or does pattern formation stop prematurely. Development of the ladybug patterns over time is less studied, and it would be interesting to see if there exists a real chemical pre-pattern that moves like the models.

Figure 5.4 – A rendering of two ladybugs on a leaf.

Chapter 6

Case Study 2: Snakes

6.1 Biological Background

Snakes display many pigmentation patterns such as speckles, blotches, longitudinal stripes, and transverse stripes. Some of these pattern types are shown in Fig. 6.1. Snakes are entirely covered by protective scales whose top layer is made of translucent keratin. Under this layer is where pigmentation patterns are found. Although the keratin layer is translucent, it may contribute to the pattern's appearance by acting as prisms through which light refracts, producing an iridescent sheen. Scales contribute to pattern formation in other reptiles by limiting diffusion of pigments across the scale boundaries (Manukyan et al., 2017).

Pigmentation patterns perform many useful functions, such as camouflaging the snake from prey and predators or acting as a brightly coloured warning signal that the snake may be venomous. An example warning pattern is seen on the North American coral snake, *Micrurus fulvius*, with its distinctive red, yellow, and black coloured bands (Fig. 6.1a). Some non-venomous snakes employ mimicry, which serves as a defence mechanism by displaying a pattern similar to that of a venomous snake's. A well-known example of this is the scarlet king snake, *Lampropeltis elapsoides*, which also displays red, black and yellow bands like *M. fulvius* (Fig. 6.1b).

The regions of a snake that display pigmentation patterns can be roughly partitioned into the head, body, tail, and underbelly. Here I focus on the most striking patterns found on the body and tail. The similarity between patterns found in these regions and those found on the head or underbelly varies between species. It is not uncommon for the head and underbelly to display a different pattern than the body. A notable example is the ring-necked snake, shown in Fig. 6.1c, which presents a dull colour on its back, serving as camouflage. When the ring-necked snake is provoked, it will display its bright red and orange underside to deter predators.

Figure 6.1 – Examples of snakes with interesting patterns. **a:** An American coral snake, *M. fulvius*. **b:** A scarlet kingsnake, *L. elapsoides*. **c:** The tail of a ring-necked snake. **d:** A python. **e:** A banded kingsnake snake. Photographs **(a)** © *2017 by Trent Adamson, used with permission*, **(b)** *Glenn Bartolotti, licensed under CC BY-SA 3.0*, **(c)** *Peter Paplanus, licensed under CC BY-NC 2.0*, **(d)** *R. Cammauf, licensed under public domain*, **(e)** *pixcove.com, licensed under public domain.*

Snake patterns start development when the animal still in its egg. Murakami et al. (2018) studied the early development of pigmentation patterns in the Japanese four-lined snake (*Elaphe quadrivirgata*). The characteristic pattern *E. quadrivirgata* displays is four

black lateral lines on a light brown background (Fig. 6.2c). In juveniles, these stripes are dark brown and are connected by transverse lines like a ladder. As the snake matures the transverse lines disappear.

To understand how the patterns of *E. quadrivirgata* develop, snakes were observed at different points in time during embryonic development. Pigmentation is measured by observing chromatophore cells, which contain pigmentation molecules. A type of chromatophore, called a melanophore, is responsible for brown to black colouration and is first seen in deeper tissues of *E. quadrivirgata* around 18-21 days after oviposition. These cells do not affect the later development of melanophores seen in the dermal and epidermal layers. Striped pigmentation patterns emerge behind the head after 26-32 days. Stripes then appear on the whole body, first thinly at 29-35 days and then clearly at 35-42 days. Data about how patterns emerge and develop are valuable because they contribute to an understanding of the patterning process and provide testable data points to guide the model's development.

Allen et al. (2013) studied the role behaviour, and ecological factors played in snake pattern diversity by using reaction-diffusion. They classified these patterns by simulating reaction-diffusion with a custom-built program and having users compare the simulated patterns against real snake images. The match between a simulation and a real snake image provided an association between the reaction-diffusion model and a real pattern. Pattern features such as size, complexity, and anisotropy were then represented as parameters.

To gain insight into how patterns are related to the behaviour and environment of snakes, these parameters were associated with ecological and behavioural variables. Examples of ecological variables are those related to habitat like Desert or forest. Behavioural variables correspond to speed, aggression, and hunting strategy. Phylogenetic analysis then revealed to what extent the environment and or snake behaviour was responsible for the diversity of patterns. The results of this analysis suggested snake patterns are mainly correlated with behaviour rather than the environment the snakes inhabited. Allen et al. (2013) found that plain longitudinally- striped snakes are usually smaller and prefer to flee from predators. The

striped pattern makes the snake harder to track by sight while moving. Transverse striped and blotched snakes are often larger and more aggressive. They may also be more venomous and hunt by ambush.

6.2 Previous Modelling Work

Patterns on the ocellated lizard[1] have been simulated by Manukyan et al. (2017). These lizards are covered in a quasi-hexagonal lattice of pigmented scales. A juvenile lizard's scales display white spots on a brown background. This pattern changes in adults whose scales are coloured individually, either a solid green or black. Pattern development continues over the lifetime of the adult, with scales switching between green and black. Manukyan et al. (2017) modelled the adult pattern using a reaction-diffusion system on a grid where multiple cells represented one scale. They lowered the diffusion rates across the scale boundaries simulating thinner skin. Consequently, the diffusion between scales is much slower than inside an individual scale. This reaction-diffusion system behaves like a cellular automaton.

J.D. Murray and M.R. Myerscough (1991) modelled snake patterns by simulating the movement and interaction of chromatophores. Before chromatophores differentiate, they exist as chromatoblasts, which are found uniformly in the dermis. After some time, these cells may become chromatophores by producing pigments, resulting in a visible pattern. Movement of these cells is driven by diffusion and chemotaxis (movement up concentration gradients). The use of chemotaxis makes it possible to generate simple and more complex patterns when calculated on a growing domain. Although J.D. Murray and M.R. Myerscough (1991) proposed that standard reaction-diffusion models may also produce the same patterns. J.D. Murray and M.R. Myerscough (1991) produced more intricate patterns such as staggered and side-by-side spots as well as diamond-shaped patterns by growing the domain. Simulations are carried out on a grid, and only the patterns on the body of the snake

[1] Although lizards are of a different suborder than snakes, they both reside in the squamate reptile family because of their scaled bodies.

are considered.

Pinheiro (2017) created a program for modelling snake patterns by using a combination of textures, cellular automata, and image manipulation. To simulate a transverse stripe pattern, a modeller can define up to four different coloured bands of various thicknesses. Similarly, for longitudinal stripes, the modeller defines the number and the colour of the stripes to be used. Circle textures are randomly distorted in size and position to simulate spots. For other simple patterns, cellular automata are used to generate blotch or zigzag patterns. Generated patterns are unnaturally uniform, so they are distorted using Perlin noise to look more organic. This results in a generated texture that is rendered on a snake mesh and enhanced by using a roughness and height map to provide a scaly appearance. This approach produces compelling results, although Pinheiro (2017) notes that more complex patterns need a phenomenon such as chemotaxis.

6.3 Model Description

I have produced snake patterns using reaction-diffusion on a mesh representing the snake's skin. Most of my models use different parameters for the ventral scales, as there often is a different pattern found there. Snake meshes have been modelled and rendered using the 3D computer graphics software Blender. A normal map is used to add a scaly appearance, and two black spheres are used to represent snake eyes. Models A-E use Gray-Scott reaction-diffusion equations, Eqn. 2.6, and the parameter values are found in Table 6.1. Model F uses the activator-depleted substrate model with saturation, Eqn. 5.1, and its parameters are found in Table 6.2. Models B, D, and F assume anisotropic diffusion. The vector field used runs parallel to the snake's longitudinal axis. The coefficient λ_1 is the diffusivity in the direction of the vector field, and λ_2 is the diffusivity orthogonal to it (Eq. 3.4).

The first snake model displays brightly coloured transverse stripes like those of the Honduran milk snake (*Lampropeltis triangulum hondurensis*). This species is non-venomous but appears similar to other venomous snakes. The initial distribution of morphogens and the final pattern are shown in Fig. 6.2.

Figure 6.2 – Model of the Honduran milk snake. The activator, a, is visualized where concentration values from low to high are represented as dark orange, black, and bright orange. **a:** The initial state: $a = 1$ on the nose and 0 elsewhere, $s = 1$ everywhere. **b:** The final pattern. **c:** An image of a real Honduran milk snake.

The next snake model displays a pattern consisting of four black lateral stripes, similar to those of an adult *E. quadrivirgata*. I have assumed anisotropic diffusion of the activator a, as described in Eqn. 3.1, of a. The diffusivity coefficients are $\lambda_1 = 1$ and $\lambda_2 = 0.502$. Anisotropic diffusion with $\lambda_1 > \lambda_2$ was important because the lines should form parallel to the longitudinal axis of the snake. The ventral scale's parameters vary from those used on the dorsal side by setting $f = 0$ so that extra lines do not form on the underside. The initial and final patterns are shown in Fig. 6.3.

Figure 6.3 – Model of the *E. quadrivirgata*. The activator, a, is visualized where concentration values from low to high are represented as brown to black. **a:** The initial morphogen distribution is $s = 1$ and $a = 0$ except a stripe of $a = 1$ down the dorsal side of the snake. **b:** The final pattern. **c:** An image of a real *E. quadrivirgata*. © *2014 by Anthony Plettenberg Laing, used with permission.*

The third model I made displays a spot or blotch pattern like that of the spotted rock snake (*Lamprophis guttatus*). This pattern is interesting because, in basic spot producing reaction-diffusion models, the spots tend to be distributed over the domain equally. However, the pattern seen on *Lamprophis guttatus* has spots located close together in two rows down the snake's back. This type of pattern can be produced by adding growth and adaptive subdivision to the model. There are two phases during pattern formation. In phase 1, the simulation is initialized from a random placement of morphogens. After initialization, a row of spots forms on the snake. During phase 2, the snake grows uniformly, increasing its surface area by four times. Faces of the mesh start with an average area of 0.52 and are subdivided when they exceed the max face area of 1. This growth allows the spot pattern to form into two rows of spots. Fig. 6.4 shows this model.

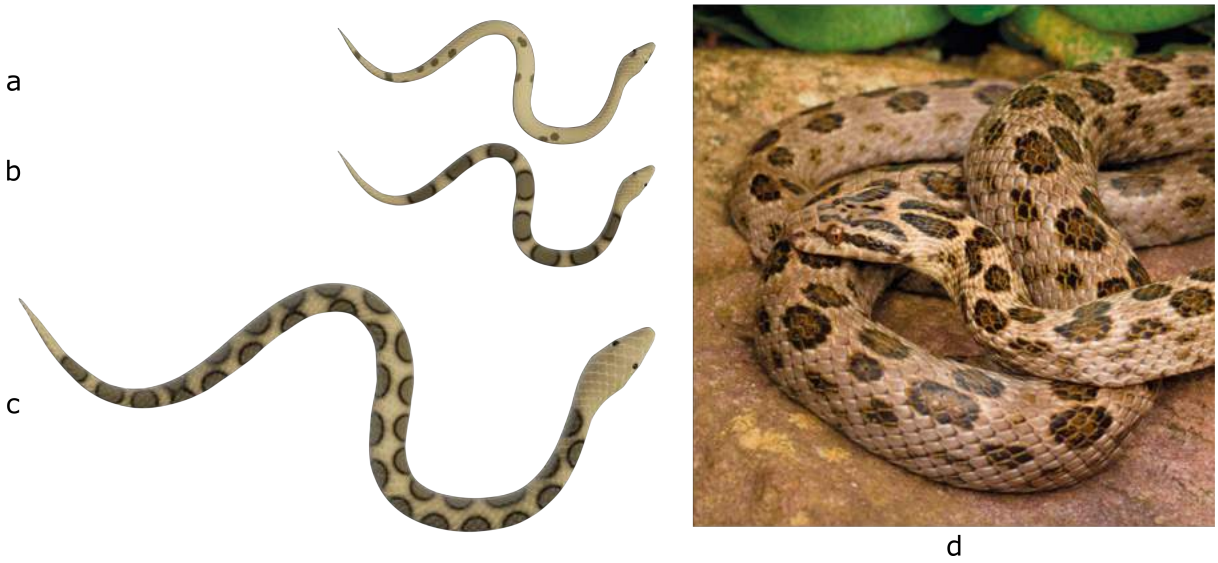


Figure 6.4 – Model of the spotted rock snake. **a:** Start of phase 1 showing the initial morphogen distribution of $s = 1$ and $a = 0$ except for 30 randomly placed spots where $a = 1$. **b:** A pattern of spots has formed along the snake. This is the end of phase 1 and the start of phase 2. **c:** End of phase 2 where rows of spots have formed on an enlarged snake. **d:** An image of a real spotted rock snake. © *2019 by Tyrone Ping, used with permission.*

The common European viper, *Vipera berus*, displays an interesting zigzag pattern. This snake is venomous, and the pattern can serve as a warning signal or as camouflage when the snake is tightly coiled (Lillywhite, 2014). Morphogen a diffuses anisotropically using coefficients $\lambda_1 = 0.81$, $\lambda_2 = 1$. This model is shown in Fig. 6.5.

Figure 6.5 – Model of the European viper. **a:** The initial distribution of a is randomly chosen from $[.5, 1)$ for each vertex and $s = 1$ everywhere. The concentration of a seen is multiplied by 0.263 to show the character of the start state pattern. **b:** The final pattern. In this case a multiplied by 0.556. **c:** An image of a real *V. berus*. *Photograph by Benny Trapp, licensed under CC BY 3.0.*

Another snake model is based on the transverse stripes of a southern coral snake (*Micrurus frontalis*). The stripes are brightly coloured and are a warning to others that this snake contains a potent venom. There are two phases: pattern establishment and mesh growth. During phase 1, a simple stripe pattern is established. During phase 2, the mesh grows uniformly, causing the snake's surface area to double. Mesh faces start with an average area of 0.55, and the faces are subdivided when they exceed an area of 1. The snake mesh used is shown in Fig. 6.6.

Figure 6.6 – Model of the *M. frontalis*. **a:** Initially $s = 1$ and $a = 0$ everywhere except on the nose, where $a = 1$. a is visualized from red to black to white and is normalized by 0.520. **b:** end of phase 1 where a basic stripe pattern has formed. **c:** The final pattern after growth. Black stripes have appeared in-between the previous stripes. **d:** A picture of a real *M. frontalis*. *Photograph by William Quatman, licensed under CC BY-SA 2.0.*

The California kingsnake (*Lampropeltis californiae*) contains white and black transverse stripes, which sometimes bifurcate. I have assumed anisotropic diffusion on the snake's body. Morphogen u has diffusivity coefficients $\lambda_1 = 0.75$, $\lambda_2 = 1$, and v has $\lambda_1 = 1$, $\lambda_2 = 0.75$. Anisotropic diffusion was needed to align the stripes perpendicular to the body. The snake's head assumes standard isotropic diffusion as the real snake does not have stripes on its head. This model is shown in Fig. 6.7.

Figure 6.7 – Model of the *L. californiae*. Morphogen u is visualized as white when its value is low and black when high. **a:** The initial distribution is $u = 0$ and $v = 1$ everywhere except a lateral stripe of $u = 1$. **b:** The final pattern. **c:** An image of a real *L. californiae*. © *2015 by David Steen, used with permission.*

6.4 Discussion and Future Work

Snake patterns provide an interesting modelling challenge due to the broad diversity of patterns on a geometrically simple domain. I have produced a variety of patterns based on real snake species. By using the features of LRDS, such as the simulation of growth and anisotropic diffusion, I generated relatively complex pigmentation patterns. My previous models of ladybugs and research into the parameter space of the Gray-Scott model (Eqn. 2.6) helped me identify iconic features of reaction-diffusion patterns that were also seen on snakes. Most of the models produced the same type of pattern on a straight and curved domain. However, simulating model E on a curved domain caused the stripes to split, creating an incorrect forked pattern. Likewise, simulating model F on a straight domain did not allow the stripes to fork correctly. This behaviour raises the question as to what pose a snake assumes throughout its pattern's development. As stated by J.D. Murray and M.R. Myerscough (1991), reaction-diffusion without the effect of chemotaxis is expressive enough to produce some snake patterns. As illustrated in Fig. 6.8, reaction-diffusion models can be

used to create convincing biological patterns for use in computer graphics. Future work may provide an understanding of the role of scales in pattern formation. Models that account for the effect of scale boundaries on diffusion may produce the more delicate details seen in nature, such as the pigmented tips of scales.

One noticeable limitation of the models is seen on the ends of the tails. Stationary reaction-diffusion patterns are frequently standing waves with a fixed wavelength. Thus, the patterns can only fit on a domain if there is enough space for them. A fixed wavelength poses a problem because the end of the tail runs out of space to support the same pattern seen on the body. In nature, the patterns tend to scale down to account for the tail tapering. However, gradually reducing the diffusion rate based on the proximity to the end of the tail might be a solution.

Snake pattern formation is of great interest to herpetologists, who study reptiles and amphibians. They can understand snake evolution through the myriad of patterns snakes display. Snake breeders profit off selling snakes that display striking and unique patterns. Consequently, there is also a financial incentive to predict the effect of breeding on pattern development.

Model	**Species**	$\boldsymbol{D_a}$	$\boldsymbol{D_s}$	$\boldsymbol{f}$	$\boldsymbol{k}$	$\boldsymbol{dt}$	**Total steps (x1000)**
A	*L. triangulum hondurensis*	1.000	2.000	0.026	0.055	0.030	200
B	*E. quadrivirgata*	0.175	0.350	0.078	0.061	0.030	28
C	*L. guttatus*	0.350	0.700	0.022	0.022	0.100	100; 70
D	*V. berus*	0.150	0.300	0.109	0.053	0.300	40
E	*M. frontalis*	1.000	2.000	0.034	0.057	0.030	200; 23

Table 6.1 – Parameters for models A-E using the Gray-Scott equations (Eqn. 2.6). In B and C the ventral scales have $f = 0$. In D the ventral scales have $f = 0$ and $k = 0.08$. In the "Total steps (x1000)" column, values separated by a semicolon denote a multi-phase model with the first and second values representing phase 1 and 2 respectively.

Model	Species	D_u	D_v	κ	ρ_u	ρ_v	σ_u	σ_v	μ_u	dt	Total steps (x1000)
F	*L. californiae*	0.056	3.36	0.5	0.18	0.36	0.001	0.1	0.08	0.01	70

Table 6.2 – Parameter values for the California kingsnake model. This model uses the activator-depleted substrate equations (Eqn. 5.1).

Figure 6.8 – A rendering of *E. quadrivirgata* with iridescence.

Chapter 7

Case Study 3: Flowers Petal Patterns

7.1 Biological Background

To humans, flowers are a symbol of natural beauty. Their spots and stripes enhance this quality (Fig. 7.1). For nature, flower pigment patterns have a more utilitarian purpose. They are used to help plants reproduce by attracting insects and likely evolved to exploit pollinator vision to further this goal. Patterns guide insects to nectar and can appear as paths or landing spots, sometimes visible only in the UV spectrum (Davies et al., 2012). Alternatively, patterns may develop due to external factors such as infections, age, and the environment (Davies et al., 2012; Dana Olivia Robinson and Adrienne HK Roeder, 2015). Thus, understanding them also gives insight into flower reproduction, insect behaviour, and environmental factors. Some patterns mimic the appearance of female bugs, as in the case of the bee orchid (Vereecken and Schiestl, 2008). This deceives bees looking for mates, into pollinating the flower. Surprisingly, given the importance of petal patterns to nature, little attention has been given to simulating them.

Pigments are molecules that provide the colour seen in many natural patterns. Anthocyanins and carotenoids are two examples of pigments found in plants. Anthocyanins can appear as red, blue, and purple, and carotenoids appear red or yellow (Bayer et al., 1966).

Martin and Gerats (1993) studied the role genes play during pigment creation in flowers. These pigments are created in the developing flower petal epidermis. Petals starts as small mounds of cells known as primordia. It is thought that two independent groups of cells grow to form the inner and outer epidermis of the petal. The inner epidermis is the first layer of cells inside the flower. Conversely, the outer epidermis is the first layer of cells on the surface of the flower's exterior. A selection of cells contains the genes that produce pigments in response to proteins called transcription factors. Many transcription factors are involved in complex activation and inhibition relationships together, making identification of the relevant morphogens difficult. As cells of the primordia undergo mitosis, they propagate their genetic information distally. This propagation can create wedge-shaped pigmentation patterns in the presence of transcription factors. In some cases, transcription factors involved in pigment production are supplied by the petal's veins. Therefore, epidermis near a vein will contain more pigment than that which is more distant. 5 to 6 days before the flower has matured, the petals cells have stopped dividing. In the remaining days, cells undergo a process of elongation and this is when pigments are produced.

Figure 7.1 – Examples of pigment patterns on real flowers. *Photographs courtesy of Przemyslaw Prusinkiewicz.*

Yuan (2019) analyzed pattern formation on various species of *Mimulus*(monkeyflowers). Specifically *M. lewisii* and *M. guttatus*. This flower contains red spots located where the flower petals converge into semi-tubular furrows, as shown in Fig. 7.2. The red pigmentation consists of anthocyanin. Through the exploration of the genetic mechanisms controlling pig-

mentation accumulation and distribution, Yuan (2019) identified the key proteins responsible for spot formation. The protein interactions that were also discovered display the key aspects of reaction-diffusion: diffusion, autocatalysis, and inhibition. The proteins are named nectar guide anthocyanin (NEGAN) and red tongue (RTO). NEGAN is a transcription factor responsible for the production of red pigment, and its distribution over the flower can be thought of as a pre-pattern. Through experimentation, Yuan (2019) found that RTO diffuses throughout the flower, and NEGAN is localized in the furrows. The protein NEGAN is shown to be autocatalytic, and in the presence of RTO, this reaction is inhibited. Consequently, RTO is the inhibitor, and NEGAN is the auto-catalytic activator.

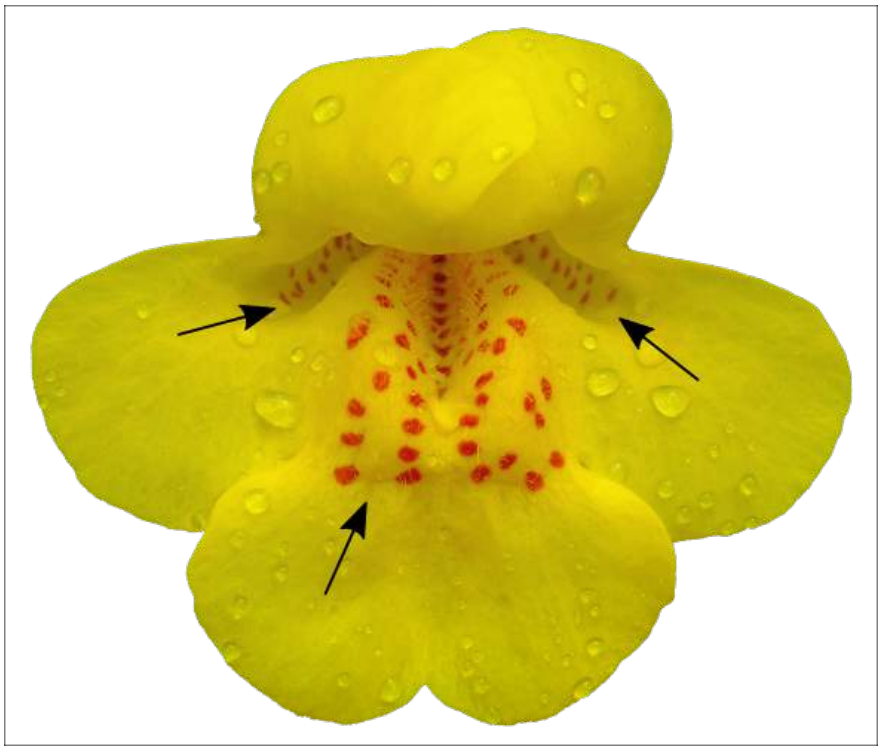


Figure 7.2 – Image of a monkeyflower covered in raindrops. The black arrows denote regions where pattern formation occurs. *Photograph by James Gaither, licensed under CC BY-NC-ND 2.0.*

7.2 Previous Modelling Work

Reaction-diffusion has been used to simulate petal patterns on a grid (Zhou et al., 2007). In this model, petal shape, initial morphogen distribution, and a venation map are input as textures. The diffusion rate at a given point is determined by the distance to a vein. This distance is found by querying the venation map. The reaction-diffusion equations are simulated to generate a final pigment distribution. Finally, colour is determined by mapping the concentration to a user-provided colormap.

Another mathematical approach to flower modelling for use in computer graphics is proposed in (Lu and Song, 2014). This work focuses on modelling flower petal patterns as well as geometric components such as the pistil, stamen, and receptacle in 3-dimensions. These flower components are modelled with parametric ellipsoids and cylinders, and the petal geometry is modelled by deforming rectangular surfaces. Pigment intensity is determined from various combinations of sine functions. The distance from the centre of the flower is used as the argument of these functions, and the resulting values correspond to pigment intensity. This approach provides visually good results, but it does not seem to support irregular or complex patterns.

A simulation of procedurally generated two-dimensional flowers is proposed by Risi et al. (2012). In this model, a flower is represented by a specialized artificial neural network that encodes the flower shape and colour. This network is called a compositional pattern-producing network (CPPN) (Stanley, 2007). It uses a wider range of activation functions compared to a standard neural network to produce symmetric and repeating patterns, which are features often seen in natural flower petals. An example activation function is the sine function, the use of which biases the output toward producing repeating patterns. The CPPN defines a flower by outputting the flower's perimeter and petal colour. The perimeter is found by taking an angle from the x-axis as an input, and it will output a distance value to the perimeter. The CPPN calculates flower colour by accepting a distance value, along with the angle. It then determines the colour at that location on the flower petal. This process

can be repeated for multiple layers that are composited together to create more complex flowers.

This simulation was implemented in the video game Petalz which is based on procedurally generating and sharing flowers of different shapes and colours. Users are given pre-made starting flowers, which they can then breed, display, and sell. Breeding is accomplished through mutating a single flower or cross-pollinating different flowers together. Specifically, the nodes and connections of the CPPN are mixed with another network. This results in a combination of each flower's colours and shape. Selling flowers gives the user in-game currency, which is used to buy flowers from other users, which can then be cross-bred to create new and unique flowers. The social aspect of this game leads to a crowdsourced method of flower creation where the flower attributes are selected for based on their visual appeal.

7.3 Monkeyflower Modelling

The Monkeyflower is a rare case in which actual morphogens and their reactions have been identified. The existence of this theory and the appearance of spots characteristic of reaction-diffusion patterning was a compelling reason to create this model. My model was produced on a triangular mesh using LRDS. The reaction-diffusion equations I selected were activator-inhibitor, Eqn. (2.3), because of the inhibitory relationship between NEGAN and RTO. Here NEGAN plays the role of activator, a, and RTO is the inhibitor, h. In nature, NEGAN is only found in the nectar guides. This holds true in mutant flowers that lack the presence of RTO. Consequently, my model uses two sets of parameters. The difference between the parameters is that the base production of NEGAN, ρ_a, only exists in the nectar guides. The boundary between the regions provides the required noise to instigate pattern formation. Consequently, spots first form on the boundary and continue to form towards the centre. This procession aligns the spots to the shape of the boundary. The full list of parameters is

found in Table. 7.1.

Model	D_a	D_h	ρ	ρ_a	ρ_h	μ_a	μ_h	dt	**Total steps (x1000)**
M. guttatus	$2.5 * 10^{-5}$	0.001	0.05	0.0125, 0	0	0.05	0.08	0.005	1,500

Table 7.1 – Parameters for the monkeyflower model using activator-inhibitor equations (Eqn. 2.3). The ρ_a value after the comma is used at the periphery. The initial morphogen distribution has NEGAN = 0 and RTO = 1 everywhere.

Results

This model produces convincing results as compared to the real picture and has mimicked the spot positioning and general character (Fig. 7.3). The spots seen on the region boundary look artificial because of how neatly they are arranged. This may be due to the sharp transition between regions and can be improved with noise or more care when they are specified. Further study should be done to determine to what extent the separate regions in my model occur in nature. And, if there are two regions, is the boundary shape responsible for the linear arrangement of spots? This model also provides a starting point for models of other monkeyflower species.

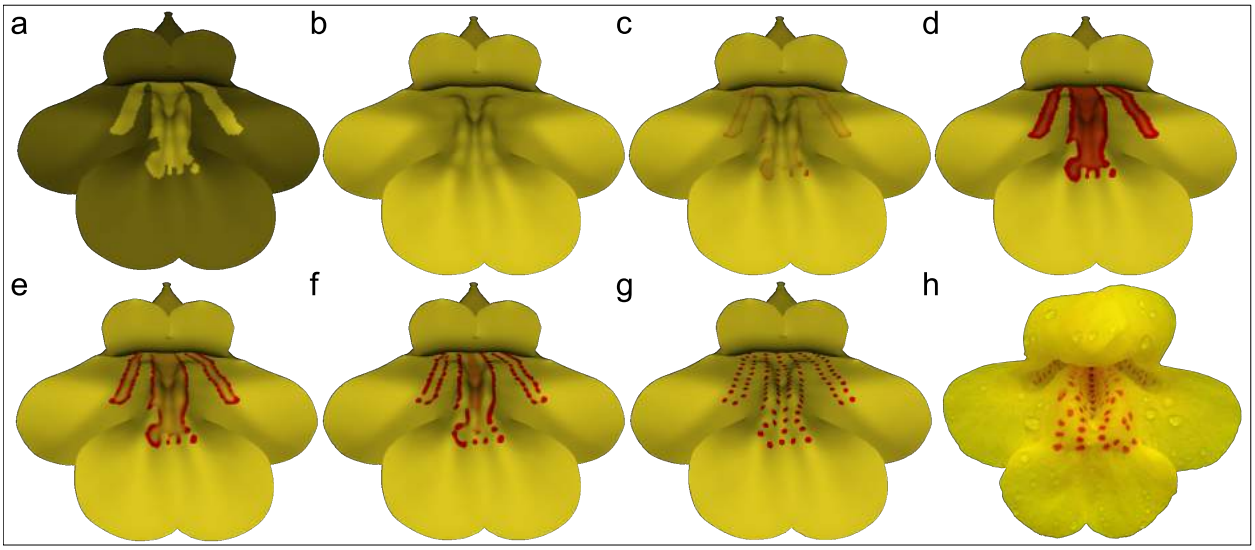


Figure 7.3 – Simulated and real monkeyflowers. **a:** NEGAN base production, ρ_a, is 0 in the dark region, and 0.0125 in the light region. **b-g:** progression of simulated pattern formation. **h:** a picture of a real monkeyflower.

7.4 Other Flower Models

Orchids display many beautiful pigmentation patterns. Examples range from scattered distributions of spots to arrangements of discordant lines, and venation patterns. This wide range of intriguing patterns pose a compelling modelling challenge, which is made more difficult because of their highly varied petal morphology. Patterns on the genus *Digitalis* and *Kohleria* also display interesting spot and stripe patterns. I have used LRDS to model these flower patterns, shown in Fig. 7.4.

Model a: This model is of a *Phalaenopsis* orchid, which displays varying sizes of purple spots on a white background. The *Phalaenopsis* orchid was a modelling challenge due to the variance in spot size and the relatively complex geometry of the flower. A regular spot pattern was simulated to simulate the purple spots. Pattern formation was stopped before the pattern became fully stable, allowing for varying sizes of spots in the resulting pattern. The parameters are listed in Table. 7.2.

Model b: This orchid of the *Encyclia* genus displays orange and yellow stripes across the flower petals. The domain is partitioned into three circular zones that increase the pattern

Model	D_a	D_s	F	k	dt	Total steps
a (*Phalaenopsis*)	0.25	0.5	0.082	0.063	0.3	150

Table 7.2 – Parameters for the *Phalaenopsis* orchid using the Gray-Scott reaction-diffusion system (Eqn. 2.6). At the start, 400 random vertices are initialized with both activator, a and substrate, s values randomly chosen in $[0, 1)$. The remaining vertices have no activator and substrate concentration of 1.

scale as it moves from the centre to the ends of the petals. Initially the morphogens u and v are set to $4 + [-2.0, 2.0)$ everywhere. Anisotropic diffusion (Eq. 3.4) is used to orient the pattern across the petals with both u and v assuming $\lambda_1 = 0.35$, $\lambda_2 = 1$ and the vector field is radiating from the centre. After this pattern has settled, there is a second phase, during which the boundary conditions are changed in some sections to act as sources and sinks. In these sections, u and v use Dirichlet conditions with a value of 0 and 30, respectively. This simulation is then stopped after $4,500$ steps before the pattern fully settles. The parameters for this model are in Table. 7.3.

Model c: Some Kohleria flowers display two distinct patterns: A white background with red spots on the border of the petals, and red oriented lines that branch and radiate from the centre. The outside of the flower appears as a solid light pink. I modelled the inner patterns using anisotropic diffusion (Eq. 3.4). This model uses two morphogens: u and v. Initially they are set to $4 + [-3.0, 3.0)$ everywhere. The model produces spots in the absence of anisotropic diffusion. The region containing line patterns has been modelled by increasing the diffusion rates with respect to the longest axis of the flower. This causes spots to become a series of branching connected lines. The inner region parameters differ by $a = 16$ and u assumes anisotropic diffusion with the coefficients $\lambda_1 = 0.51$, $\lambda_2 = 1$ and vectors radiating from the centre. To form correctly sized spots in the outer region, I use $a = 16.5$ with isotropic diffusion. The parameters for this model are in Table. 7.4.

Model d: *Digitalis* has a scattered pattern of dark purple spots on the bottom inside of its flowers. These spots are surrounded by a white halo that merges with others nearby. Beyond these halos, the rest of the flower appears pink or light purple.

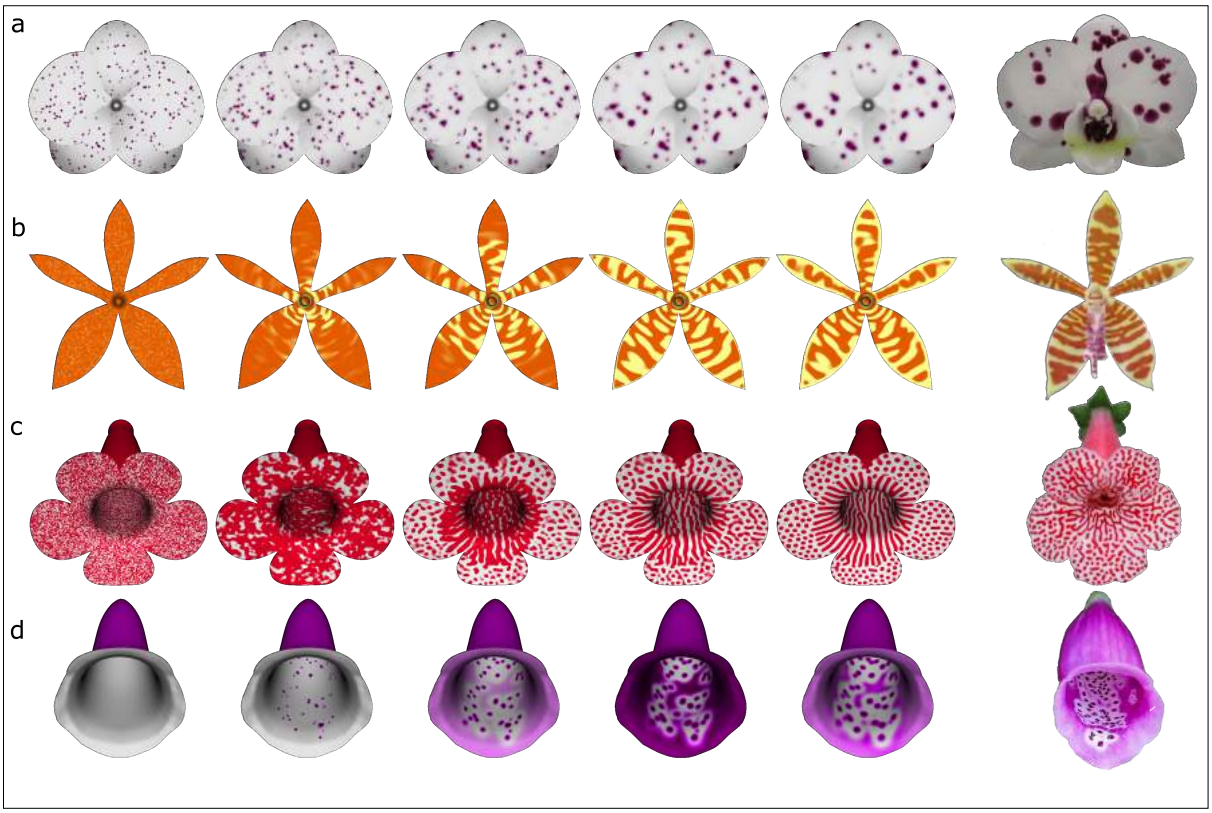


Figure 7.4 – A developmental sequence of the simulated flower models and the corresponding real flowers on the right. **a-d:** *Phalaenopsis*, *Encyclia*, *Kohleria*, *Digitalis*. *Photographs **(a-b)** courtesy of Przemyslaw Prusinkiewicz,* **(c-d)** *by pixabay.com, licensed under Pixabay License.*

Model	D_u	D_v	α	β	s	*uSat*	*dt*	Total steps (x1000)
b (*Encyclia*)	0.4	0.01	16	12	0.005, 0.002, 0.001	6.3	0.05	60; 4.5
c (*Kohleria*)	0.3	0.0625	15.75, 16.5	12	0.035, 0.04	7	0.1	10

Table 7.3 – Parameter values for *Encyclia* and *Kohleria* using Turing reaction-diffusion (Eqn. 2.2). *uSat* represents the maximum value of the u. Values separated by commas are used on different regions of the domain.

Model	D_a	D_s	ρ_a	ρ_s	μ_a	μ_s	ρ	*dt*	Total steps (x1000)
d (*Digitalis*)	0.00004	0.0015	0.0125	0.05	0.05	0.08	0	0.005	20.7

Table 7.4 – Parameters for the *Digitalis* model using the activator-depleted substrate formula (Eqn. 2.4). This model uses two morphogens s and a. The substrate s is 1 everywhere and a is 0 except for a few vertices at the bottom of the flower. These cells have a value of $a = 1$ and will progress over time to become the dark purple pigment spots.

7.5 Discussion and Future Work

As far as I know, this study represents the only reaction-diffusion simulation on arbitrary triangulated surfaces of flower petal patterns. I have also implemented a biologically-motivated model of monkeyflower spots and their formation. A rendering of the monkeyflower is shown in Fig. 7.5. This simulation provides a reason for the linear arrangement of spots arising from the shape of a parameter boundary. I have also produced other flower models using various reaction-diffusion equations. These models highlight the usefulness and flexibility of reaction-diffusion and LRDS. Anisotropic diffusion was critical for aligning patterns. Future work should identify to what extent anisotropic diffusion occurs in real petals.

There are many more flowers with striking patterns to be modelled. The visual appearance of patterns is affected by the shape of the flower petal cells. Incorporating this phenomenon into renderings of flower petal patterns will provide a more realistic simulation appearance. Flowers attract insects through pigmentation patterns only visible in ultraviolet light. These types of patterns would be an exciting modelling challenge, especially if real insects were attracted to the simulated images. More insight into how genes interact and affect pattern formation would help specifying model PDEs. Future works may investigate the role of growth and vasculature structures on pattern formation.

Figure 7.5 – A rendering of a monkeyflower.

Chapter 8

Case Study 4: Psoriasis

Reaction-diffusion has been widely used to model natural patterns that appear as part of normal development. Reaction-diffusion can also represent other patterns, specifically patterns of disease. This chapter describes such an application. It is an edited version of (Ringham et al., 2019)[1].

8.1 Introduction

Most skin diseases manifest themselves with reproducible patterns of skin lesions, which are conventionally described in terms of lesion morphology (e.g. macules, papules, plaques, etc) and distribution on the skin surface (Nast et al., 2016). The biological basis of pattern formation is only understood in a few special cases. For instance, the segmental pattern of herpes zoster reflects dermatomal viral reactivation through sensory nerves, and the linear pattern in Blaschko lines represents genetic mosaicism. In most cases, however, the mechanisms by which pathological processes in the skin generate reproducible patterns remain virtually unknown (Nast et al., 2016). The majority of skin diseases are inflammatory, which explains why the lesions are often red, elevated and scaly (resulting from, respectively: vasodilation

[1]P.P. and R.G. designed the research, L.R. and P.P. created the mathematical model and performed computer simulations, L.R., P.P. and R.G. wrote the paper.

and hyperemia, inflammatory infiltrate and edema, and pathologically increased epidermal keratinization secondary to inflammation). The skin has a large surface (average 1.5 m^2 - 2.0 m^2) compared to its thickness (0.5 mm-4 mm; the surface-to-volume ratio of approximately 650 m^2/m^3) (Leider, 1949), and is therefore ideally suited to study the mechanisms of spatial propagation of inflammatory processes in a tissue. Psoriasis, a chronic, autoimmune inflammatory skin disease affecting $2\% - 3\%$ of the population in Western countries (Parisi et al., 2013) provides a particularly useful model. The lesions are sharply demarcated, scaly, and distributed symmetrically on the body (Christophers, 2001; Griffiths and Barker, 2007; Nestle et al., 2009). The plaques evolve from pinpoint papules by centrifugal growth, which explains an oval contour of mature lesions (Farber et al., 1985; Soltani and Van Scott, 1972). Individual plaques may merge producing polycyclic contours (Christophers, 2001; Farber et al., 1985). In some instances the plaques have the appearance of rings (referred to as annular, arciform or circinate patterns) (Christophers, 2001; Nast et al., 2016), which is the predominant morphological feature in approximately 5% of patients (Morris et al., 2001). The mechanisms responsible for these patterns are not readily explainable in terms of the lateral propagation of inflammation, in which one would expect gradual attenuation of inflammation due to the dilution of proinflammatory agents that diffuse in the skin. In contrast, in psoriatic lesions the intensity of inflammation is preserved throughout the whole plaque and sharply suppressed at its margin over the distance of a few millimeters. We show that the phenotypic features of psoriasis can be explained in terms of interactions between key pathogenic cytokines consistent with a reaction-diffusion model. This model captures all cardinal phenotypic features of psoriasis and may provide a wider framework to understand the patterning and maintenance of inflammation in other skin diseases.

8.2 Results

8.2.1 Classification of Psoriasis Plaque Patterns

The patterns repetitively identified in the literature are listed in Fig. 8.1, see the full paper for additional details.

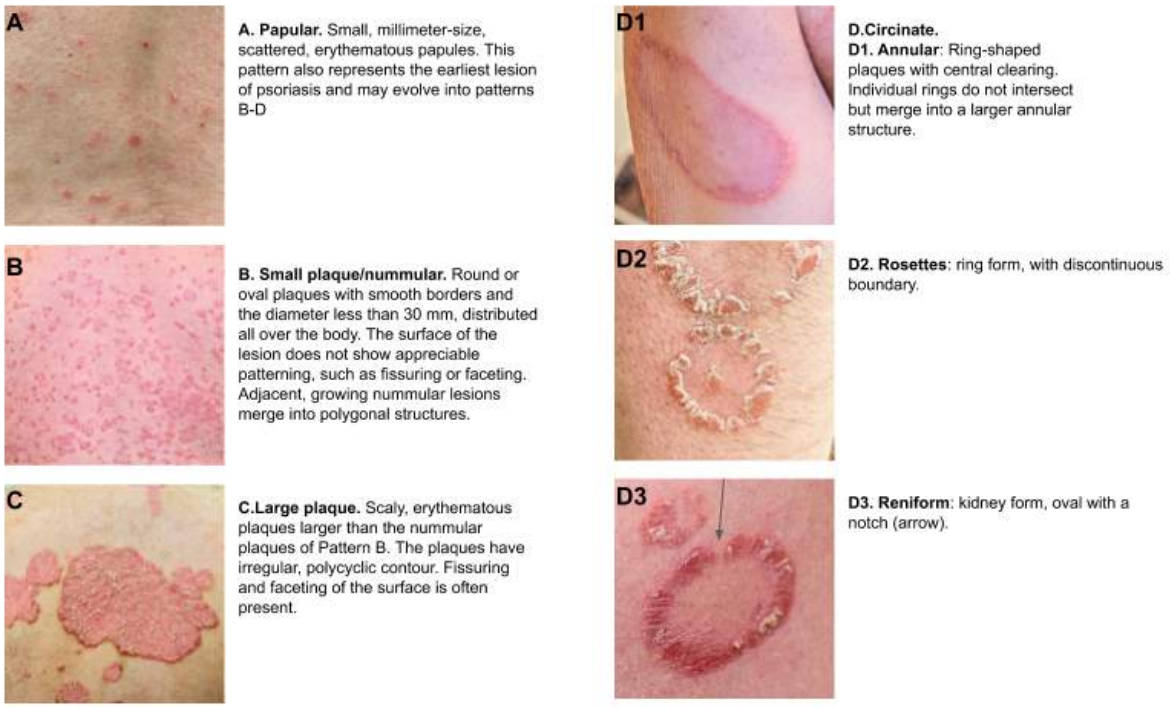


Figure 8.1 – Patterns of skin lesions psoriasis.

8.2.2 Model of Cytokine Interactions in Psoriasis

Cytokines IL-23, IL-17 and TNFα are central mediators in the psoriatic plaque formation, as underscored by the fact that pharmacological blockade of either cytokine by monoclonal antibodies causes clinical remission in a large proportion of patients (Jabbar-Lopez et al., 2017). Interactions between the cytokines inferred from the available data are shown schematically in Fig. 8.2A. The most important pathogenic cytokines are those of the IL-17 family being produced primarily by the T_H17 lymphocytes (interaction **0**) (Krueger et al., 2012).

These cells require IL-23 for expansion and activation (Cosmi et al., 2008; Wilson et al., 2007; Zheng et al., 2007), and amplify the inflammatory process by inducing other proinflammatory cytokines, the most important of which is TNFα (Boehncke and Schön, 2015). Psoriatic plaques contain both dendritic cells producing IL-23 and T_H17 cells expressing the IL-23 receptor (Cosmi et al., 2008; Lee et al., 2004; Tillack et al., 2014; Wilson et al., 2007). Treatment with guselkumab, a selective therapeutic monoclonal antibody inhibiting IL-23, attenuates IL-17s in psoriatic plaques and in serum in patients with psoriasis (interaction **1**) (Hawkes et al., 2018; Sofen et al., 2014; Tillack et al., 2014). This attenuation is correlated with the clinical clearing of psoriasis lesions (Sofen et al., 2014). IL-17 and TNFα synergize with each other (Alzabin et al., 2012; Krueger et al., 2012; Xu et al., 2017): IL-17 increases the expression of TNFα (Jovanovic et al., 1998) (interaction **2**), whereas therapeutic TNFα inhibition blocks IL-17 in responding patients (interaction **3**) (Zaba et al., 2007, 2009). The positive feedback of IL-17 cytokines on their own production (interactions **2** and **3**) is further demonstrated by the findings that IL-17A induces IL-17C (Xu et al., 2018), and that the therapeutic inhibition of the IL-17 receptor with brodalumab reduces the expression of the IL-17 cytokine (IL-17A, C, F) (Russell et al., 2014). TNFα downregulates IL-23 (interaction **4**) either directly (Notley et al., 2008; Zakharova and Ziegler, 2005) or indirectly via inhibition of interferons (Palucka et al., 2005; Tillack et al., 2014). Disturbance of this negative interaction is probably responsible for paradoxical induction of psoriasis in patients with rheumatoid arthritis and inflammatory bowel disease treated with TNFα antibodies (Palucka et al., 2005; Tillack et al., 2014). That induction is readily reverted by therapeutic inhibition of the excess of IL-23 by ustekinumab, an antibody binding to the p40 chain of IL-23 (Tillack et al., 2014).

8.2.3 Computational Model Construction

To analyze whether the molecular-level interactions depicted in Fig. 8.2A can account for the observed plaque patterns and the response of the disease to treatment, we constructed a

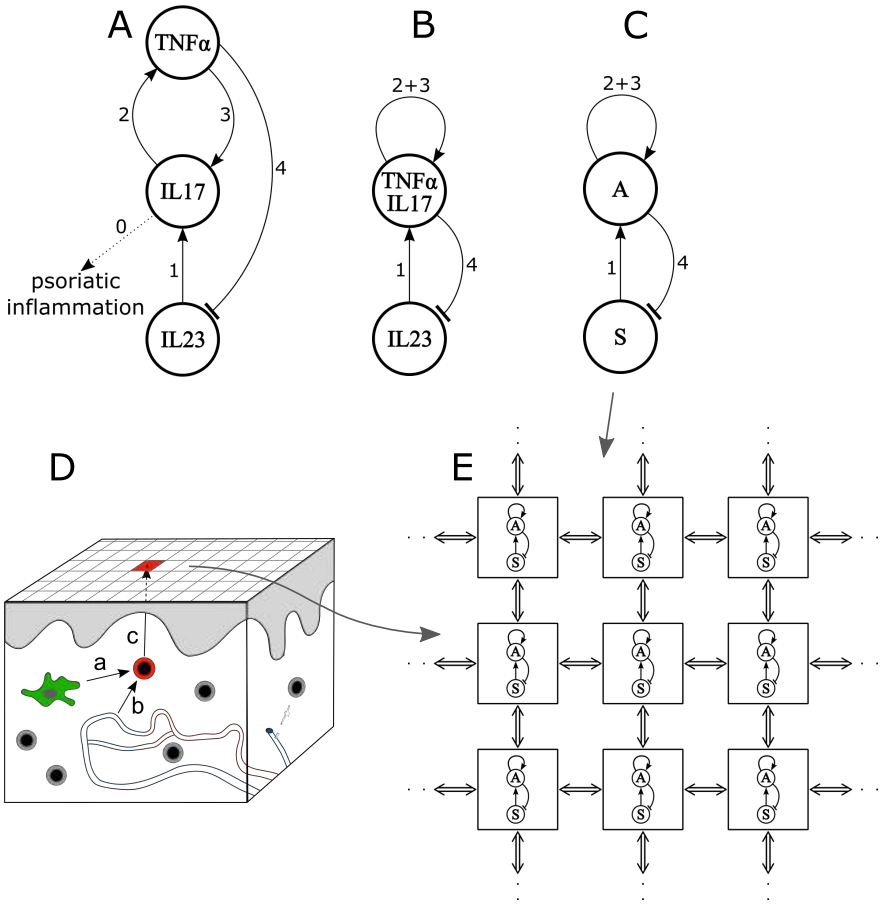


Figure 8.2 – Modelling plaque formation in psoriasis. **A)** Interactions between key cytokines involved in psoriasis plaque formation. Labels 0-4 refer to the observations from which these interactions have been inferred (see Results). **B)** A simplified diagram of interactions, in which cytokines IL-17 and TNFα are considered jointly. **C)** Diagram B relabeled as an activator (A) - depleted substrate (S) system. **D)** Skin representation and simulation initialization. The skin surface is partitioned into square regions. A lesion is initiated by an activated T_H17 cell (red) which is either a resident memory T-cell activated by a dendritic cell (green, interaction a) or has migrated from circulation through a capillary wall (interaction b). The area of microinflammation around the activated T_H17 cell is considered as a "seed" region and its projection to the surface (arrow c) is colored in red. Epidermis, the upper layer of the skin is shaded in grey, capillaries in the dermis are colored in red (arterioles) and blue (venules). Skin resident memory T-cells are marked in grey. **E)** Detail of skin surface representation. Each region is two-dimensional projection of the underlying activator-depleted substrate system of proinflammatory cytokines and represents a computational cell implementing reaction system (C). These computational cells are interconnected (double arrows), allowing for the diffusion of cytokines.

mathematical model. We followed the standard method of simplifying the modeled system to focus on its essence and make the model more amenable to analysis (Bak, 1996; Gaines, 1977; Prusinkiewicz, 1998). This simplification reduced the size of the parameter space and thus, to the extent possible, the use of parameters for which quantitative data are currently unavailable. It has also related the problem of plaque pattern formation to a known class of reaction-diffusion systems, which provided guidance for the exploration of the parameter space, and facilitated the analysis and interpretation of the results.

We have pursued the following train of thought. The mutual promotion of cytokines IL-17 and TNFα, represented by interactions **2** and **3** in Fig. 8.2A, suggests that their concentrations may change in concert. Assuming this is the case, we reduced the three-substance graph in Fig. 8.2A by representing IL-17 and TNFα jointly. The resulting two-substance graph (Fig. 8.2B) has the structure of an activator-depleted substrate reaction-diffusion model (Gierer and Meinhardt, 1972; Marcon et al., 2016)(Fig. 8.2C). In this model, the substrate S with concentration s is locally converted into the activator A with concentration a according to the canonical equations (Gierer and Meinhardt, 1972; Meinhardt, 1982):

$$\begin{aligned} \frac{\partial a}{\partial t} &= ka^2 s + \rho_{a0} - \mu_a a + D_a \nabla^2 a \\ \frac{\partial s}{\partial t} &= -ka^2 s + \rho_{s0} - \mu_s s + D_s \nabla^2 s \end{aligned} \tag{8.1}$$

The term $ka^2 s$ indicates that the conversion is autocatalytically promoted by the activator, with the rate controlled by parameter k. Its concentration increases at the expense of the substrate, thus the activator downregulates the substrate. Parameters ρ_{a0} and ρ_{s0} are the rates of the base production of the activator and the substrate, and μ_a and μ_s control their turnover. The remaining terms, $D_a \nabla^2 a$ and $D_s \nabla^2 s$, represent diffusion of the activator and substrate at the rates controlled by parameters D_a and D_s, respectively (for simplicity, diffusion is not explicitly represented in Figs. 8.2A-C). Consistent with figures 8.2B and C, we identify variable a with the concentration of cytokines TNFα and IL-17, and s with the

concentration of IL-23:

$$a = [TNF\alpha, IL17],\ s = [IL23].$$

In the simulations, a patch of skin surface (Fig. 8.2D) is represented by an array of interconnected computational "cells", each of which performs local computation according to Equations (8.1) (Fig. 8.2E). The initial state in all simulations is a uniform distribution of IL-23 in the whole array, except for randomly distributed small "seed" areas with a high concentration of IL-17 and TNFα. These areas represent IL-17-secreting cells (such as the T_H17-cell) that either have been activated in situ (Fig. 8.2D, interaction **a**) or have migrated from the circulation to the skin or (Fig. 8.2D, interaction **b**) (Krueger et al., 2012).

8.2.4 Exploration of the Model Parameter Space

Currently, it is not feasible to measure the diffusion of cytokines in human skin and consequently, there are no experimental data to provide suggestions for the parameter values of the model. Consequently, we adopted a reverse strategy where we explored the model parameter space by searching for values that would yield psoriasis patterns observed in patients (Fig. 8.1). To guide this search, we referred to the Gray-Scott reaction-diffusion system (Gray and Scott, 1984), for which the parameter space has been thoroughly explored:

$$\begin{aligned} \frac{\partial a}{\partial t} &= a^2 s - (f + c)a + D_a \nabla^2 a \\ \frac{\partial s}{\partial t} &= -a^2 s + (1 - s)f + D_s \nabla^2 s \end{aligned} \tag{8.2}$$

We observe (see also (Yamamoto and Miorandi, 2010; Yamamoto et al., 2011)) that Equations (8.2) are a special case of Equations (8.1), where

$$k = 1,\ \rho_{a0} = 0,\ \mu_a = f + c,\ \rho_{s0} = f,\ \text{and}\ \mu_s = f.$$

The parameter space and details of six patterns obtained for specific parameter values are shown in Fig. 8.3. These patterns correspond visually to the six types of psoriasis identified in patients (Fig. 8.1). Note that, consistent with the common assumption of the Gray-Scott reaction-diffusion model, the ratio of the diffusion rates of substrate and activator was set to $D_s : D_a = 2$ (Pearson, 1993). This is a departure from the much larger ratios typically used in reaction-diffusion models (Diego et al., 2018; Gierer and Meinhardt, 1972; Kondo and Miura, 2010; Lengyel and Epstein, 1991; Marcon et al., 2016; Vastano et al., 1987). On biochemical grounds, this departure is justified by the commensurate, small size of the three cytokines, implying comparable diffusion rates (see Table 8.1). The small ratio of diffusion rates does not preclude Turing instability and spontaneous pattern emergence for carefully chosen values of the remaining parameters (see Fig. 8.4). Nevertheless, the parameter values leading to the formation of plaque patterns are compatible with the "filtering" operation mode, in which the patterns do not emerge spontaneously in a homogeneous medium and elaborate initial pre-patterns instead (Diego et al., 2018; Lee et al., 1993; Muratov and Osipov, 2000; Pearson, 1993). This latter mode is more pertinent to the development of psoriasis plaques, which is initiated by an activated T_H17 cell in the skin (Fig. 8.2D).

Molecule	**MW [kDa]**	$\boldsymbol{D_{tiss}[\mu m^2/s]}$
TNFα	26	154.4
IL17	35	123.6
IL23	54.1	89.1

Table 8.1 – Diffusion coefficients for the three cytokines involved in our model using the empirical formula $D_{tiss} = 1.778\text{x}10^{-4}\text{xMW}^{-0.75}$ (Swabb et al., 1974)(Equation F in their paper). The actual rates of macromolecule transport in a tissue may differ from these estimates, as other factors may also play a role. These include convection, which may run in the direction opposite to the concentration-gradient-driven diffusion (Swabb et al., 1974), and cell proliferation, which may be relevant to the transport of cytokines otherwise mostly confined to their mother cells.

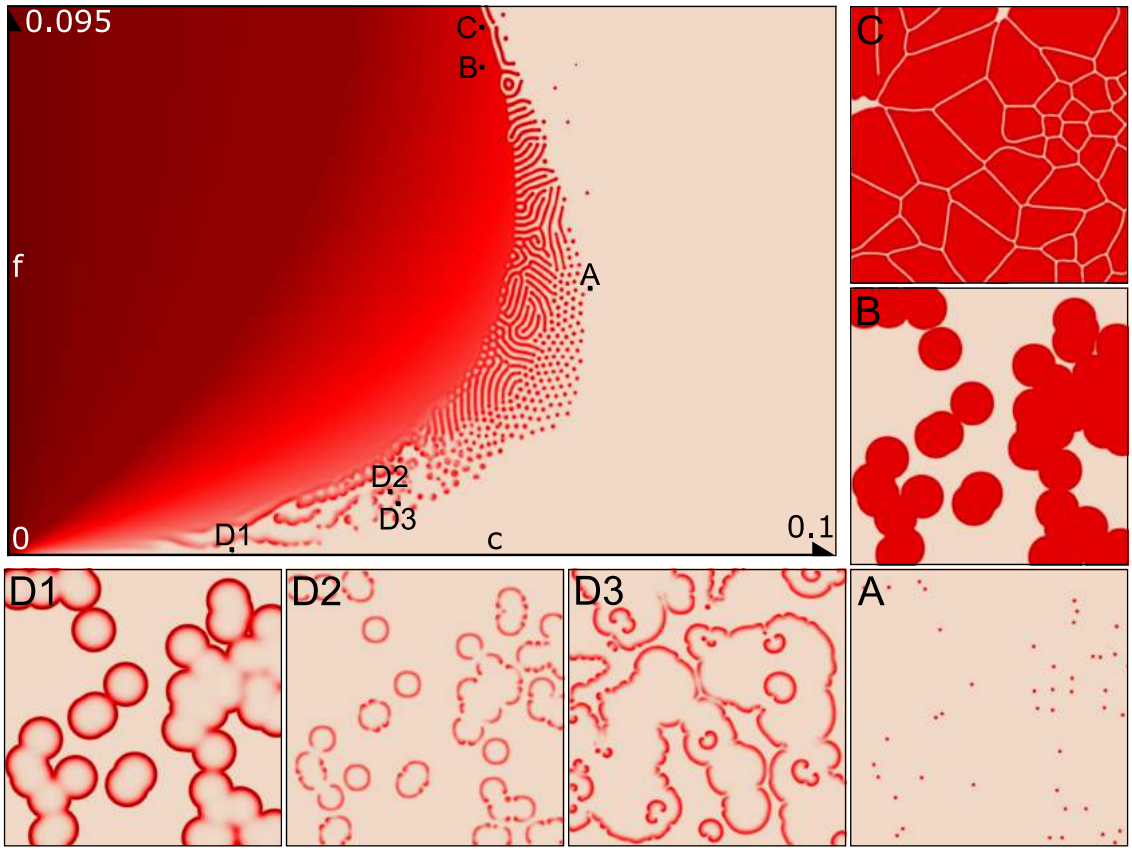


Figure 8.3 – Parameter space of the model and selected patterns. Top left: A comprehensive representation of the range of patterns generated using Equations 8.2 for different values of the synthetic parameters c and f. **A-D**: magnified views of patterns generated using select parameter values. These labels and patterns correspond to the patterns of psoriatic skin lesions identified in Fig. 8.1.

8.2.5 The Development of Lesions and Response to Treatment

The simulated development of psoriasis lesions and the response to treatment are shown in Fig. 8.5. The development was simulated by using the forward Euler method to advance the state of the reaction-diffusion model over time, given an initial random distribution of small papules. The parameter values and the initial conditions for each of these simulations are listed in Table 8.2, with additional information characterizing the sensitivity of simulations to the variation of (individual) parameter values collected in Table 8.3. Minimum values of the activator A, representing cytokines IL-17 and TNFα, needed to initiate pattern formation are collected in Table 8.4. The simulated patterns shown in Fig. 8.5 A-D3 have

striking resemblance to the actual patterns of psoriatic skin lesions shown in Fig. 8.1. Next, we simulated the effect of therapy by increasing the decay rate of cytokines IL-17 and TNFα (activator A), which mimics real-life treatment with an anti-cytokine antibody. Interestingly, the simulated lesion clearing was not simply a time-reversal of the processes of plaque formation: the interior of the plaques cleared first, producing annular lesions (Fig. 8.5, row 5). The residual lesions dispersed slowly, eventually disappearing entirely or leaving residual spots (Fig. 8.5, row 6).

Finally, to verify that the modelling results do not critically depend on the reduction of the three-substance system in Fig. 8.2A to the two-substance system in Fig. 8.2B, we have constructed a simulation model corresponding directly to Fig. 8.2A (see Supplementary Text). Guided in part by parameter values found for the two-substance model (Tables 8.2 and 8.3), we found values for which the three-substance model produces qualitatively the same plaque patterns (Table B.1). This result validates the simplification underlying the two-substance model.

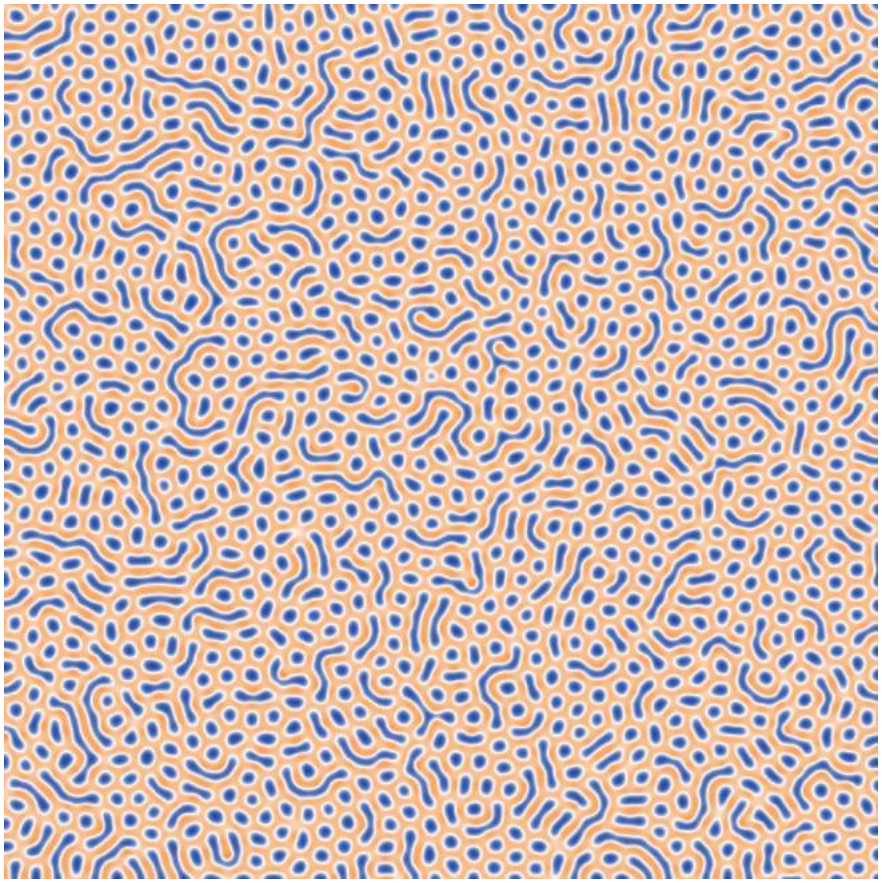


Figure 8.4 – (Example of a pattern generated de novo using the Gray-Scott model (Equation 8.2) after 6000 iterations. The concentration is visualized from blue to orange. Parameter values: $f = 0.042, c = 0.06, D_a = 0.25, D_s = 0.5, dt = 1$. The initial conditions are a homogeneous distribution everywhere, with the addition of a small amount of noise: $a = 0.22557, s = 0.45219 \pm 0.000001$.

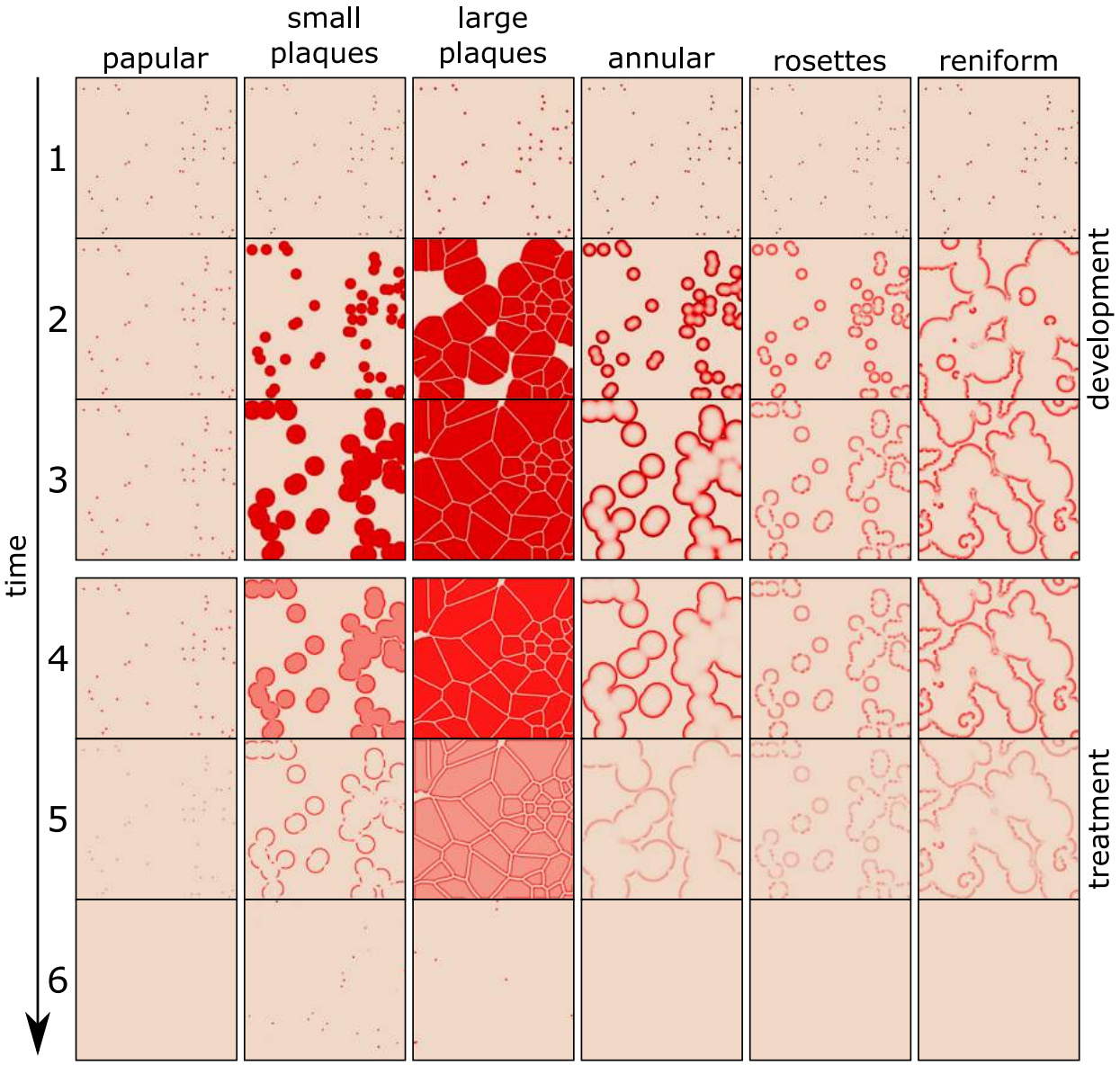


Figure 8.5 – The simulated progression of different types of psoriatic lesions. Rows 1-3: Development of the lesions. The earliest stage of a papule (Row 1) consists of randomly distributed small seed areas. Later forms of the disease (Rows 2 and 3) correspond to patterns identified in Figs. 1 and 3. Rows 4-6: The effect of treatment simulated by increasing the decay rate of IL-17 and TNFα. Note that the treatment does not result in a simple reversal of the original pattern development, but produces residual lesions with more activity at the margin of the plaques (Row 5). In some instances, residual papules persist (Row 6).

8.3 Discussion

Since the foundation of dermatology as a medical specialty in the beginning of the 19th century, morphological patterns provided a useful and robust criterion for the diagnosis and classification of skin diseases. However, the mechanisms through which skin diseases produce diverse patterns remained unknown. We have shown that all major morphological types of the common skin disease psoriasis (papular, small plaque, large plaque, and different forms of circinate patterns) can be generated by a reaction-diffusion model with different parameter values. The model is based on the currently known up- and down-regulating interactions between three proinflammatory cytokines: TNFα, IL-23 and IL-17. These interactions are not direct chemical reactions, but are mediated by the immunologically active cells stimulating or inhibiting the release and proliferation of intermediary cytokines. The model has a spatio-temporal character, explaining the emergence of patterns during disease development and their disappearance during subsequent treatment. Reaction-diffusion thus provides a promising framework for studying mechanisms underlying the progress and treatment of psoriasis. As detailed data regarding the interaction and diffusion of cytokines involved in psoriasis become available, more elaborate models may be constructed to recreate the actual biological processes in the skin with an increased accuracy. Recent advances in the theoretical understanding of reaction-diffusion (Diego et al., 2018) suggest that the resulting models may also become more robust to parameter changes, currently limited to narrow ranges. Inflammatory patterns related to psoriasis are found in other diseases as well. For example, annular lesions are seen in erythema multiforme, dermatophytosis and erythema annulare centrifugum; reniform patterns in erythema gyratum repens, urticaria and lupus erythematosus; and rosettes in granuloma annulare. We thus hypothesize that reaction-diffusion models can be applied further to explain the patterns of other inflammatory skin diseases, and suggest their treatment by selective cytokine inhibition. Eventually, reaction-diffusion models could provide a framework for understanding the pathogenesis and pharmacologic intervention of a broad spectrum of skin diseases.

Name	**Papular**	**Small Plaque**	**Large Plaque**	**Annular**	**Rosette**	**Reniform**
$\mu_{[IL23]} = \rho_{[IL23]0}$	0.046	0.084	0.091	0.001	0.009	0.011
$\mu_{[TNF\alpha]}$ (*before treatment*)	0.116	0.141	0.148	0.028	0.056	0.057
$\mu_{[TNF\alpha]}$ (*during treatment*)	0.120	0.1467	0.153	0.04	0.0625	0.065
maxSteps	12,620	14,000	143,000	4,500	3,900	15,500
treatSteps	12,000	12,000	140,000	1,700	2,700	13,000

Table 8.2 – Parameter values used to generate the six classes of psoriasis plaque patterns shown in Fig. 8.5. In all simulations $k = 1, \rho_{a0} = 0, D_a = 0.25$, and $D_s = 0.5$. Simulations were carried out using forward Euler methods with time-step $dt = 0.4$ for $maxSteps$ iterations, with the treatment starting after $treatSteps$ iterations. The textures used in all simulations had a resolution of 500 x 500 texels, with each texel representing a sample point of a discretized patch of the skin. Parameters of individual simulations are collected in Table 8.1. We assumed Neumann boundary conditions set to 0, i.e., no diffusion of activator A and substrate S across the boundary. The initial activator concentration a was set to 0 in each texel except for 50 seed spots, placed randomly across the domain. Each spot was represented by a 3x3 array of texels with a concentration of 0.5 (See Table 8.3 for the minimum values). The initial concentration of the substrate s was 1.0 everywhere. All concentrations were represented with 32-bit floating point accuracy.

Name	Papular	Small Plaque	Large Plaque	Annular	Rosette	Reniform
ρ_{s0}	[0.04510, 0.04705]	[0.08375, 0.15000]	[0.09060, 0.09110]	[0.00075, 0.00285]	[0.00875, 0.00910]	[0.01085, 0.01112]
ρ_{a0}	[0.00000, 0.00075]	[0.00000, 0.00525]	[0.00000, 0.00002]	[0.00000, 0.00022]	[0.00000, 0.00015]	[0.00000, 0.00008]
μ_s	[0.04435, 0.04725]	[0.03475, 0.08475]	[0.09085, 0.09175]	[0.00000, 0.00125]	[0.00875, 0.00910]	[0.01085, 0.01120]
μ_a	[0.11265, 0.11899]	[0.10000, 0.14180]	[0.14785, 0.14870]	[0.01500, 0.03500]	[0.05360, 0.05650]	[0.05620, 0.05750]
D_s	[0.46000, 0.57500]	[0.42500, 0.80000]	[0.46100, 0.50500]	[0.00000, 1.05000]	[0.47500, 0.52500]	[0.42500, 0.61500]
D_a	[0.22000, 0.27000]	[0.17500, 0.29000]	[0.24500, 0.27000]	[0.15000, 0.75000]	[0.23500, 0.27500]	[0.22500, 0.28500]
k	[0.95900, 1.05000]	[0.98800, 1.60000]	[0.99000, 1.00200]	[0.75000, 1.35000]	[0.98500, 1.06500]	[0.97500, 1.01500]

Table 8.3 – Ranges of parameter values resulting in patterns visually similar to those shown in Fig. 8.5. For each varied parameter all remaining values are as in Table 8.2.

Pattern	A	B	C	D1	D2	D3
Minimum initial concentration of the activator at the spots	0.208	0.244	0.256	0.088	0.126	0.127

Table 8.4 – Minimum values of the activator A needed to initiate the formation of patterns shown in Fig. 8.5.

8.4 Limitations of the Study

The main limitation of this study is that the validity of the proposed model cannot be confirmed by direct measurements of cytokine concentration gradients in the skin. Although the diffusion rates of the cytokines are expected to be similar to each other (Table 8.1), which is consistent with the Gray-Scott-type model, the range of diffusion is likely to be much larger than the predicted millimeter scale due to the accrual of cytokine-secreting cells to the inflammatory infiltrate and centrifugal cell movement. Currently, the large-scale measurements of cytokine gradients in human skin are not technically feasible.

Chapter 9

Software Performance

A critical aspect in the design of LRDS was real-time performance. In order to support pattern formation at interactive rates, the PDEs and extensions to reaction-diffusion were required to be calculated quickly. Efficient simulation was achieved by leveraging the GPU to perform computation in a highly parallelized fashion. As shown in Table 9.1, the high degree of parallelization provided by the GPU was integral for increasing simulation speed in medium to large domains. And performance of LRDS can be improved greatly by upgrading the GPU used for computation.

This speed facilitated interaction by decreasing the time between user provided input and LRDS' response. GPU integration allows a user to directly manipulate their model and explore how parameter changes effect pattern details. Another benefit was the added ability to observe pattern formation in real-time.

User productivity is also affected by software performance. Studies have shown that, when using websites, a delay greater than 1 second interrupts the user's flow of thought, and if a delay is greater than 10 seconds, the user will want to do something else (Nielsen, 1994). Thus, it was important to minimize delays after user's actions in order to increase user productivity.

Cell Count	GPU - NVIDIA GTX 850M (ms)	CPU - INTEL i7-4810MQ (ms)
642	21,117	22,628
10,242	21,964	91,805
21,728	24,387	194,066
40,962	33,250	378,590
163,842	94,017	2,104,099

Table 9.1 – Analysis of LRDS performance. Shown is the time taken to perform 10,000 iterations of a reaction-diffusion simulation. For small domains consisting of less than 1000 cells, the CPU and GPU exhibit similar performance. When simulating on medium to large domains, the GPU outperforms the CPU by an order of magnitude or more as the number of cells increases. This benefit is still seen, even with the modest graphics card used for this test. Timings of the specific models presented in this thesis are found in Table. C.1.

Chapter 10

Conclusions

10.1 Contributions

In this thesis, I have introduced LRDS, an environment for quick and efficient authoring and simulation of reaction-diffusion models on grids and triangulated manifolds. To create a model, users easily specify their equations and parameters in a text file. These models are then simulated on the CPU or by using parallelized computation on the GPU. LRDS provides advanced features like anisotropic diffusion, non-homogeneous parameters, and domain growth with adaptive subdivision to give users flexibility when creating patterns. Integration of these features and the speed of the GPU has made LRDS an powerful tool for interactively exploring a wide array of reaction-diffusion models. Furthermore, these patterns can then be used to create textures or animations of pattern development for use in computer graphics, video games, and films. LRDS also has applications in a scientific setting, allowing users to model their observations and test their biological hypotheses quickly. These models can then be used to gain a better understanding of nature.

Using LRDS, I have produced models of natural patterns on grids and triangular meshes in a series of case studies. The first study concerns simulating ladybug elytra patterns. These patterns are composed of spots, stripes, and loops and are coloured in vivid red,

black, brown, and yellow. To represent the elytra, I used a triangular mesh. Using a mesh provides a flexible and natural representation of the elytra's curvature and shape compared to grid-based simulations. Pattern formation occurs directly on the mesh, avoiding any mapping distortion. The colour of the simulated patterns was chosen by referencing real ladybug images.

The next study concerns pigmentation patterns seen on snakes. These patterns can be spots and stripes or more complex patterns composed of zigzags and blotches. Growth and anisotropic diffusion augment simple patterns to create more complex ones. Pattern simulation occurs on a snake-shaped mesh.

The third study deals with simulating flower petal patterns. Although the colourful patterns seen on flower petals are a defining aspect of their appearance, few studies research simulating them. I created novel models of orchids by referencing parameters from other works as well as independently searching the parameter space. The models were expanded by varying parameters spatially and by using anisotropic diffusion. I also modelled the spotted purple flower *Digitalis*, among others. Using existing protein interactions that determine pigment formation, I modelled the monkeyflower species. The monkeyflower is a rare case where the real world morphogens responsible for pattern formation are known.

In the final case study, I presented a biologically-motivated model of the autoimmune disease psoriasis. This study took a departure from modelling pigmentation patterns and ventured into the domain of medicine. A hallmark of psoriasis is the red lesions that appear on the skin with a variety of geometric patterns. These patterns are an essential characteristic of the disease, yet the mechanisms through which they arise remain unknown. We modelled the interactions between the main pathogenic cytokines, TNFα, IL17, and IL23, to produce all known patterns of psoriasis. From this, we simulated the treatment of the disease through cytokine targeting. This computational model offers an exciting approach to understanding psoriasis better because of the rapid rate in which psoriasis can be simulated compared to the actual disease. Modelling also provides an avenue for testing treatments and possibly a

future cure.

10.2 Future Work

From this thesis, two main avenues can be explored as future work. First, LRDS can be extended with more features. Coupling growth and patterning has been shown to create compelling structures in 3D Harrison et al. (2002); Holloway and Harrison (2007). This feature would be easy to add as the essential components are already implemented. Support for volumetric domains would allow for the exploration of 3D patterning. An area of research that would benefit from this feature is the simulation of vein formation. Supporting arbitrary values for Neumann boundary conditions and more boundary conditions in general would be useful as well. Active-transport is an alternative to diffusion that could also be a useful feature in LRDS for modelling biological systems.

Concerning future reaction-diffusion work, the effect of more sophisticated methods for computing reaction-diffusion, like higher order differential operators and time-stepping schemes, should be researched to see what effect they have on pattern formation. Another future work is to simulate the same patterns on meshes of varying resolution and quality to determine how triangle shape and size effects patterning.

The second area of future work concerns the technical aspects of LRDS. Support for CPU multi-threading, SIMD, or distributed computing would increase the software's performance and productivity of users. Being able to edit all aspects of the parameter file directly inside LRDS would increase usability. Currently, the only supported operating system is Windows, adding cross-platform support would allow LRDS to reach a more extensive user base. Finally, I intend to open-source LRDS to allow users to explore and build off my results.

Bibliography

(2009). *Encyclopedia of Insects*. Elsevier Science, 2 edition.

Allen, W. L., Baddeley, R., Scott-Samuel, N. E., and Cuthill, I. C. (2013). The evolution and function of pattern diversity in snakes. *Behavioral Ecology*, 24(5):1237–1250.

Alzabin, S., Abraham, S. M., Taher, T. E., Palfreeman, A., Hull, D., McNamee, K., Jawad, A., Pathan, E., Kinderlerer, A., Taylor, P. C., et al. (2012). Incomplete response of inflammatory arthritis to TNFα blockade is associated with the Th17 pathway. *Annals of the rheumatic diseases*, 71(10):1741–1748.

Ando, T., Matsuda, T., Goto, K., Hara, K., Ito, A., Hirata, J., Yatomi, J., Kajitani, R., Okuno, M., Yamaguchi, K., Kobayashi, M., Takano, T., Minakuchi, Y., Seki, M., Suzuki, Y., Yano, K., Itoh, T., Shigenobu, S., Toyoda, A., and Niimi, T. (2018). Repeated inversions within a pannier intron drive diversification of intraspecific colour patterns of ladybird beetles. *Nature Communications*, 9(1):3843.

Andreux, M., Rodola, E., Aubry, M., and Cremers, D. (2014). Anisotropic Laplace-Beltrami Operators for Shape Analysis. In *NORDIA'14 - Sixth Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment*.

Atlassian (2019). Sourcetree. `https://www.sourcetreeapp.com/`.

Bak, P. (1996). How nature works: the science of self-organized criticality.

Bayer, E., Egeter, H., Fink, A., Nether, K., and Wegmann, K. (1966). Complex formation and flower colors. *Angewandte Chemie International Edition in English*, 5(9):791–798.

Boehncke, W.-H. and Schön, M. P. (2015). Psoriasis. *The Lancet*, 386(9997):983–994.

Christophers, E. (2001). Psoriasis- epidemiology and clinical spectrum. *Clinical and experimental dermatology*, 26(4):314–320.

Cornut, O. (2019). Dear ImGui. `https://github.com/ocornut/imgui`.

Cosmi, L., De Palma, R., Santarlasci, V., Maggi, L., Capone, M., Frosali, F., Rodolico, G., Querci, V., Abbate, G., Angeli, R., et al. (2008). Human interleukin 17–producing cells originate from a CD161+CD4+ T cell precursor. *Journal of Experimental Medicine*, 205(8):1903–1916.

Crane, K., de Goes, F., Desbrun, M., and Schröder, P. (2013). Digital Geometry Processing with Discrete Exterior Calculus. In *ACM SIGGRAPH 2013 courses*, SIGGRAPH '13, New York, NY, USA. ACM.

Dana Olivia Robinson and Adrienne HK Roeder (2015). Themes and variations in cell type patterning in the plant epidermis. *Current Opinion in Genetics & Development*, 32:55 – 65. Developmental mechanisms, patterning and organogenesis.

Davies, K., Albert, N., and Schwinn, K. (2012). From landing lights to mimicry: The molecular regulation of flower colouration and mechanisms for pigmentation patterning. *Functional Plant Biology*, 39:619–638.

Descombes, Samira Michèle and Dhillon, Daljit Singh and Zwicker, Matthias (2016). Optimized CUDA-Based PDE Solver for Reaction Diffusion Systems on Arbitrary Surfaces. In Wyrzykowski, Roman and Deelman, Ewa and Dongarra, Jack and Karczewski, Konrad and Kitowski, Jacek and Wiatr, Kazimierz, editor, *Parallel Processing and Applied Mathematics*, pages 526–536, Cham. Springer International Publishing.

Diego, X., Marcon, L., Müller, P., and Sharpe, J. (2018). Key features of Turing systems are determined purely by network topology. *Physical Review X*, 8(2):021071.

Farber, E. M., Nall, L., and Strefling, A. (1985). Psoriasis: a disease of the total skin. *Journal of the American Academy of Dermatology*, 12(1):150–156.

Fowler, D. R., Meinhardt, H., and Prusinkiewicz, P. (1992). Modeling seashells. *ACM SIGGRAPH Computer Graphics*, 26(2):379–387.

Gaines, B. R. (1977). System identification, approximation and complexity. *International Journal of General System*, 3(3):145–174.

Garzón-Alvarado, Diego A. and Ramírez Martinez, Angelica M. (2011). A biochemical hypothesis on the formation of fingerprints using a turing patterns approach. *Theoretical Biology and Medical Modelling*, 8(1):24.

Gierer, A. and Meinhardt, H. (1972). A theory of biological pattern formation. *Kybernetik*, 12(1):30–39.

Glisan King, A. and Meinwald, J. (1996). Review of the Defensive Chemistry of Coccinellids. *Chemical Reviews*, 96(3):1105–1122. PMID: 11848782.

GollyGang (2012). READY. `https://github.com/GollyGang/ready`.

Gray, P. and Scott, S. (1984). Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system A + 2B $\rightarrow$ 3B; B $\rightarrow$ C. *Chemical Engineering Science*, 39(6):1087 – 1097.

Griffiths, C. E. and Barker, J. N. (2007). Pathogenesis and clinical features of psoriasis. *The Lancet*, 370(9583):263–271.

Harrison, L. G., Wehner, S., and Holloway, D. M. (2002). Complex morphogenesis of surfaces: theory and experiment on coupling of reaction–diffusion patterning to growth. *Faraday Discussions*, 120:277–293.

Hawkes, J. E., Yan, B. Y., Chan, T. C., and Krueger, J. G. (2018). Discovery of the IL-23/IL-17 signaling pathway and the treatment of psoriasis. *The Journal of Immunology*, 201(6):1605–1613.

Herholz, P. (2013). General discrete Laplace operators on polygonal meshes.

Holloway, D. M. and Harrison, L. G. (2007). Pattern selection in plants: coupling chemical dynamics to surface growth in three dimensions. *Annals of botany*, 101(3):361–374.

Jabbar-Lopez, Z. K., Yiu, Z. Z., Ward, V., Exton, L. S., Mustapa, M. F. M., Samarasekera, E., Burden, A. D., Murphy, R., Owen, C. M., Parslew, R., et al. (2017). Quantitative evaluation of biologic therapy options for psoriasis: a systematic review and network meta-analysis. *Journal of Investigative Dermatology*, 137(8):1646–1654.

J.D. Murray and M.R. Myerscough (1991). Pigmentation pattern formation on snakes. *Journal of Theoretical Biology*, 149(3):339 – 360.

Jovanovic, D. V., Di Battista, J. A., Martel-Pelletier, J., Jolicoeur, F. C., He, Y., Zhang, M., Mineau, F., and Pelletier, J.-P. (1998). IL-17 stimulates the production and expression of proinflammatory cytokines, IL-β and TNF-α, by human macrophages. *The Journal of Immunology*, 160(7):3513–3521.

Kondo, S. and Asai, R. (1995). A reaction diffusion wave on the skin of the marine angelfish Pomacanthus.

Kondo, S. and Miura, T. (2010). Reaction-diffusion model as a framework for understanding biological pattern formation. *science*, 329(5999):1616–1620.

Krueger, J. G., Fretzin, S., Suárez-Fariñas, M., Haslett, P. A., Phipps, K. M., Cameron, G. S., McColm, J., Katcherian, A., Cueto, I., White, T., et al. (2012). IL-17A is essential for cell activation and inflammatory gene circuits in subjects with psoriasis. *Journal of Allergy and Clinical Immunology*, 130(1):145–154.

Lee, E., Trepicchio, W. L., Oestreicher, J. L., Pittman, D., Wang, F., Chamian, F., Dhodapkar, M., and Krueger, J. G. (2004). Increased expression of interleukin 23 p19 and p40 in lesional skin of patients with psoriasis vulgaris. *Journal of Experimental Medicine*, 199(1):125–130.

Lee, K. J., McCormick, W., Ouyang, Q., and Swinney, H. L. (1993). Pattern formation by interacting chemical fronts. *Science*, 261(5118):192–194.

Lefèvre, J. and Mangin, J.-F. (2010). A reaction-diffusion model of human brain development. *PLoS computational biology*, 6(4):e1000749.

Leider, M. (1949). On the weight of the skin. *Journal of Investigative Dermatology*, 12(3):187–191.

Lengyel, I. and Epstein, I. R. (1991). Modeling of Turing Structures in the Chlorite—Iodide—Malonic Acid—Starch Reaction System. *Science*, 251(4994):650–652.

Liaw, S. S., Yang, C. C., Liu, R. T., and Hong, J. T. (2001). Turing model for the patterns of lady beetles. *Physical Review E*, 64(4):041909.

Lillywhite, H. B. (2014). *How snakes work: structure, function and behavior of the world's snakes*. Oxford University Press.

Liu, R., Liaw, S., and Maini, P. (2006). Two-stage Turing model for generating pigment patterns on the leopard and the jaguar. *Physical review E*, 74(1):011914.

Lu, L. and Song, W. L. (2014). Simulation Research for Petal Color. In *Advances in Computers, Electronics and Mechatronics*, volume 667 of *Applied Mechanics and Materials*, pages 237–241. Trans Tech Publications.

Luc Decker (2019). TexRD. `https://www.texrd.com/`.

Malheiros, M. d. G. and Walter, M. (2017). Pattern formation through minimalist biologically inspired cellular simulation. In *Proceedings of the 43rd Graphics Interface Conference*, pages 148–155. Canadian Human-Computer Communications Society.

Manukyan, L., Montandon, S. A., Fofonjka, A., Smirnov, S., and Milinkovitch, M. C. (2017). A living mesoscopic cellular automaton made of skin scales. *Nature*, 544(7649):173.

Marcon, L., Diego, X., Sharpe, J., and Mueller, P. (2016). High-throughput mathematical analysis identifies Turing networks for patterning with equally diffusing signals. *eLife Sciences*, 5.

Marschner, S. and Shirley, P. (2015). *Fundamentals of computer graphics*. CRC Press.

Martin, C. and Gerats, T. (1993). Control of pigment biosynthesis genes during petal development. *The Plant Cell*, 5(10):1253.

Meinhardt, H. (1982). Models of biological pattern formation. *New York*.

Meinhardt, H. (2009). *The algorithmic beauty of sea shells*. Springer Science & Business Media.

Morris, A., Rogers, M., Fischer, G., and Williams, K. (2001). Childhood psoriasis: a clinical review of 1262 cases. *Pediatric dermatology*, 18(3):188–198.

Murakami, A., Hasegawa, M., and Kuriyama, T. (2018). Developmental mechanisms of longitudinal stripes in the Japanese four-lined snake. *Journal of morphology*, 279(1):27–36.

Muratov, C. and Osipov, V. V. (2000). Static spike autosolitons in the Gray-Scott model. *Journal of Physics A: Mathematical and General*, 33(48):8893.

Nast, A., Griffiths, C., Hay, R., Sterry, W., and Bolognia, J. (2016). The 2016 International League of Dermatological Societies' revised glossary for the description of cutaneous lesions. *British Journal of Dermatology*, 174(6):1351–1358.

Nestle, F. O., Kaplan, D. H., and Barker, J. (2009). Psoriasis. *New England Journal of Medicine*, 361(5):496–509.

Nie, Q., Zhang, Y.-T., and Zhao, R. (2006). Efficient Semi-implicit Schemes for Stiff Systems. *J. Comput. Phys.*, 214(2):521–537.

Nielsen, J. (1994). *Usability engineering*. Elsevier.

Notley, C. A., Inglis, J. J., Alzabin, S., McCann, F. E., McNamee, K. E., and Williams, R. O. (2008). Blockade of tumor necrosis factor in collagen-induced arthritis reveals a novel immunoregulatory pathway for Th1 and Th17 cells. *Journal of Experimental Medicine*, 205(11):2491–2497.

Owens, A., Cieslak, M., Hart, J., Classen-Bockhoff, R., and Prusinkiewicz, P. (2016). Modeling dense inflorescences. *ACM Transactions on Graphics (TOG)*, 35(4):1–14.

Palucka, A. K., Blanck, J.-P., Bennett, L., Pascual, V., and Banchereau, J. (2005). Cross-regulation of TNF and IFN-α in autoimmune diseases. *Proceedings of the National Academy of Sciences*, 102(9):3372–3377.

Parisi, R., Symmons, D. P., Griffiths, C. E., Ashcroft, D. M., et al. (2013). Global epidemiology of psoriasis: a systematic review of incidence and prevalence. *Journal of Investigative Dermatology*, 133(2):377–385.

Pearson, J. E. (1993). Complex patterns in a simple system. *Science*, 261(5118):189–192.

Pinheiro, J. M. (2017). A procedural model for snake skin texture generation.

Prusinkiewicz, P. (1998). *In search of the right abstraction: the synergy between art, science, and information technology in the modeling of natural phenomena*. na.

Ringham, L., Prusinkiewicz, P., and Gniadecki, R. (2019). Skin patterning in psoriasis by spatial interactions between pathogenic cytokines. *iScience*.

Risi, S., Lehman, J., D'Ambrosio, D. B., Hall, R., and Stanley, K. O. (2012). Combining search-based procedural content generation and social gaming in the petalz video game. In *Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Rivara, M.-C. and Inostroza, P. (1998). USING LONGEST-SIDE BISECTION TECHNIQUES FOR THE AUTOMATIC REFINEMENT OF DELAUNAY TRIANGULATIONS. *International Journal for Numerical Methods in Engineering*, 40(4):581–597.

Russell, C. B., Rand, H., Bigler, J., Kerkof, K., Timour, M., Bautista, E., Krueger, J. G., Salinger, D. H., Welcher, A. A., and Martin, D. A. (2014). Gene expression profiles normalized in psoriatic skin by treatment with brodalumab, a human anti–IL-17 receptor monoclonal antibody. *The Journal of Immunology*, 192(8):3828–3836.

Sanderson, A. R., Johnson, C. R., and Kirby, R. M. (2004). Display of vector fields using a reaction-diffusion model. In *IEEE Visualization 2004*, pages 115–122.

Schenk, C., Liehr, A., Bode, M., and Purwins, H.-G. (2000). Quasi-particles in a Three-dimensional Three-component Reaction-diffusion System. In *High Performance Computing in Science and Engineering'99*, pages 354–364. Springer.

Sofen, H., Smith, S., Matheson, R. T., Leonardi, C. L., Calderon, C., Brodmerkel, C., Li, K., Campbell, K., Marciniak Jr, S. J., Wasfi, Y., et al. (2014). Guselkumab (an IL-23–specific mAb) demonstrates clinical and molecular response in patients with moderate-to-severe psoriasis. *Journal of Allergy and Clinical Immunology*, 133(4):1032–1040.

Soltani, K. and Van Scott, E. J. (1972). Patterns and sequence of tissue changes in incipient and evolving lesions of psoriasis. *Archives of dermatology*, 106(4):484–490.

Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162.

Swabb, E. A., Wei, J., and Gullino, P. M. (1974). Diffusion and convection in normal and neoplastic tissues. *Cancer research*, 34(10):2814–2822.

Tillack, C., Ehmann, L. M., Friedrich, M., Laubender, R. P., Papay, P., Vogelsang, H., Stallhofer, J., Beigel, F., Bedynek, A., Wetzke, M., et al. (2014). Anti-TNF antibody-induced psoriasiform skin lesions in patients with inflammatory bowel disease are characterised by interferon-γ-expressing Th1 cells and IL-17A/IL-22-expressing Th17 cells and respond to anti-IL-12/IL-23 antibody treatment. *Gut*, 63(4):567–577.

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 237(641):37–72.

Turk, G. (1991). Generating textures on arbitrary surfaces using reaction-diffusion. *ACM SIGGRAPH Computer Graphics*, 25:289–298.

Vastano, J. A., Pearson, J. E., Horsthemke, W., and Swinney, H. L. (1987). Chemical pattern formation with equal diffusion coefficients. *Physics Letters A*, 124(6-7):320–324.

Vereecken, N. J. and Schiestl, F. P. (2008). The evolution of imperfect floral mimicry. *Proceedings of the National Academy of Sciences*, 105(21):7484–7488.

Wardetzky, M., Mathur, S., Kälberer, F., and Grinspun, E. (2007). Discrete Laplace operators: No free lunch. volume 07, pages 33–37.

Wilensky, U (1999). NetLogo. `https://ccl.northwestern.edu/netlogo`.

Wilson, N. J., Boniface, K., Chan, J. R., McKenzie, B. S., Blumenschein, W. M., Mattson, J. D., Basham, B., Smith, K., Chen, T., Morel, F., et al. (2007). Development, cytokine profile and function of human interleukin 17–producing helper T cells. *Nature immunology*, 8(9):950.

Witkin, A. and Kass, M. (1991). Reaction-diffusion Textures. *SIGGRAPH Comput. Graph.*, 25(4):299–308.

Wolpert, L. (1996). One hundred years of positional information. *Trends in Genetics*, 12(9):359–364.

Xu, M., Lu, H., Lee, Y.-H., Wu, Y., Liu, K., Shi, Y., An, H., Zhang, J., Wang, X., Lai, Y., et al. (2018). An interleukin-25-mediated autoregulatory circuit in keratinocytes plays a pivotal role in psoriatic skin inflammation. *Immunity*, 48(4):787–798.

Xu, T., Ying, T., Wang, L., Zhang, X. D., Wang, Y., Kang, L., Huang, T., Cheng, L., Wang, L., and Zhao, Q. (2017). A native-like bispecific antibody suppresses the inflammatory cytokine response by simultaneously neutralizing tumor necrosis factor-alpha and interleukin-17A. *Oncotarget*, 8(47):81860.

Yamamoto, L. and Miorandi, D. (2010). Evaluating the robustness of activator-inhibitor models for cluster head computation. In *International Conference on Swarm Intelligence*, pages 143–154. Springer.

Yamamoto, L., Miorandi, D., Collet, P., and Banzhaf, W. (2011). Recovery properties of distributed cluster head election using reaction–diffusion. *Swarm Intelligence*, 5(3-4):225–255.

Yuan, Y.-W. (2019). Monkeyflowers (Mimulus): new model for plant developmental genetics and evo-devo. *New Phytologist*, 222(2):694–700.

Zaba, L. C., Cardinale, I., Gilleaudeau, P., Sullivan-Whalen, M., Suárez-Fariñas, M., Fuentes-Duculan, J., Novitskaya, I., Khatcherian, A., Bluth, M. J., Lowes, M. A., et al. (2007). Amelioration of epidermal hyperplasia by TNF inhibition is associated with reduced Th17 responses. *Journal of Experimental Medicine*, 204(13):3183–3194.

Zaba, L. C., Suárez-Fariñas, M., Fuentes-Duculan, J., Nograles, K. E., Guttman-Yassky, E., Cardinale, I., Lowes, M. A., and Krueger, J. G. (2009). Effective treatment of psoriasis with etanercept is linked to suppression of IL-17 signaling, not immediate response TNF genes. *Journal of Allergy and Clinical Immunology*, 124(5):1022–1030.

Zakharova, M. and Ziegler, H. K. (2005). Paradoxical anti-inflammatory actions of TNF-α: inhibition of IL-12 and IL-23 via TNF receptor 1 in macrophages and dendritic cells. *The Journal of Immunology*, 175(8):5024–5033.

Zheng, Y., Danilenko, D. M., Valdez, P., Kasman, I., Eastham-Anderson, J., Wu, J., and Ouyang, W. (2007). Interleukin-22, a TH17 cytokine, mediates IL-23-induced dermal inflammation and acanthosis. *Nature*, 445(7128):648–651.

Zheng, Y., Zhang, L., Wang, Y., Liang, P., and Kang, J. (2009). Modeling and analyzing stripe patterns in fish skin. In *2009 International Conference on Optical Instruments and Technology: Optoelectronic Imaging and Process Technology*, volume 7513, page 75131M. International Society for Optics and Photonics.

Zhou, N., Wang, J., Dong, W., and Paul, J.-C. (2007). Modeling and Visualization of Flower Color Patterns. pages 150 – 155.

Appendix A

Program Inputs

A.1 Command Line Arguments

ModelsPath= Path to where OBJ models are located.

ShadersPath= Path to where shaders are located.

ColorMapsPath= Path to where colormaps are located.

SavePath= Path to save output to.

ConfigFile= SimConfig filename (eg. SimConfig.txt).

SimFile= Filename containing the starting morphogen concentrations.

Steps= Integer number of simulation steps until program exits.

SaveOnExit Enable saving the model when program exits.

Run Start the simulation running.

A.2 Reserved Labels Used in a Parameter File

ModelsPath: Filepath to folder containing OBJ models.

ColorMapsPath: Filepath to folder containing colormaps.

ShadersPath: Filepath to folder containing shaders.

camera: Six comma separated floating point values representing the position and look at points used to orient the camera.

model: Nine comma separated floating point values representing the X, Y, Z vectors used to orient the domain.

domain: The domain as either an OBJ filename (eg. model.obj) or grid. Only manifold OBJ meshes are supported. Grid domains look for two other parameters *width* and *height* which denote grid resolution.

xRes: Integer representing width in squares of a grid domain.

yRes: Integer representing height in squares of a grid domain.

cellSize: A float representing spatial width of a single square in a grid domain.

simFile: A filename of a text file containing all per vertex values such as morphogen concentrations, vector directions and principle diffusivities. (eg. simfile.rd)

colorMap: A filename of binary file containing a 256 RGB colormap. This is used for both inside and outside the mesh (eg. color.map).

colorMapOutside: The colormap for the outside of the mesh.

colorMapInside: The colormap for the inside of the mesh.

growthTickLimit: An integer representing the number of simulation steps before the domain is grown.

growing: *true* or *false* to turn growth on or off.

growthX: A float percentage representing growth percentage on global X axis.

growthY: A float percentage representing growth percentage on global Y axis.

growthZ: A float percentage representing growth percentage on global Z axis.

maxFaceArea: The face area threshold as a float for adaptive subdivision.

pauseAt: Integer number of simulation steps until program pauses.

exitAt: Integer number of simulation steps until program exits.

morphogens: A comma separated list of morphogen names in uppercase (eg. A, S, U, V).

initialConditions: The start of initial condition specification.

rdModel: Either GPU or CPU depending on desired computation mode. Also denotes the

indices: Specifies integer indices used to define initial conditions and parameters. Valid values are: all or 1,2,3 or 1-3.

Appendix B

Psoriasis

B.1 Three-Substance Model of Psoriasis

To show that the results obtained for the two substance system in Fig. 8.2B also hold for the three-substance system in Fig. 8.2A, we have constructed a simulation model corresponding directly to Fig. 8.2A. The equations have the form:

$$\begin{aligned}
\frac{\partial[TNF\alpha]}{\partial t} &= \rho_{[TNF\alpha]0} - \mu_{[TNF\alpha]}[TNF\alpha] + \eta[IL17] - k[TNF\alpha]^2[IL23] \\
&\quad + D_{[TNF\alpha]}\nabla^2[TNF\alpha] \\
\frac{\partial[IL17]}{\partial t} &= \rho_{[IL17]0} - \mu_{[IL17]}[IL17] + k[TNF\alpha]^2[IL23] + D_{[IL17]}\nabla^2[IL17] \\
\frac{\partial[IL23]}{\partial t} &= \rho_{[IL23]0} - \mu_{[IL23]}[IL23] - k[TNF\alpha]^2[IL23] + D_{[IL23]}\nabla^2[IL23]
\end{aligned} \tag{B.1}$$

Parameter values resulting in the different pattern classes shown in Fig. 8.5 are collected in Table B.1.

Name	**Papular**	**Small Plaque**	**Large Plaque**	**Annular**	**Rosette**	**Reniform**
[IL23]=[IL23]0	0.04	0.045	0.055	0.001	0.009	0.011
$\mu_{[TNF\alpha]}$ (before treatment)	0.103	0.103	0.115	0.028	0.055	0.054
$\mu_{[TNF\alpha]}$ (during treatment)	0.107	0.1087	0.12	0.04	0.0615	0.062
maxSteps	13,000	14,000	143,000	4,500	3,900	15,500
treatSteps	12,000	12,000	140,000	1,700	2,700	13,000

Table B.1 – Parameter values for generating the six classes of psoriasis plaque patterns shown in Fig. 8.5 using the three-substance model. In all simulations: $k = 1, \eta = 2, \rho_{[TNF\alpha]0} = \rho_{[IL17]0} = 0, \mu_{[IL17]=1, D_{[IL23]}=0.5, D_{[TNF\alpha]}=D_{[IL17]}=0.25}$.

Appendix C

Model Timings

Name	Cell Count	Time (sec)
P. quinquepuncatuts	6238	2744.32
C. septempuncata	6238	2752.61
E. crassimala	6238	2770.58
M. Hauseri	6238	2615.42
B. albolineata	6238	926.58
L. triangulum hondurensis	10593	393.84
E. quadrivirgata	10593	54.12
L. guttatus	10593	502.13
V. berus	10593	89.32
M. frontalis	10593	490.8
L. californiae	10593	147.09
M. guttatus	49700	6229.07
Phalaenopsis	10504	0.41
Encyclia	11025	119.38
Digitalis	12223	48.59
Kohleria	24561	25.36

Table C.1 – Model cell counts and the associated time to simulate each model. The computer used contained an NVIDIA GTX 850M GPU and an INTEL i7-4810MQ CPU. Computation of the PDEs was performed on the GPU.

Appendix D

Copyright Permissions

應動組-李奇峯 Wed, Jul 10, 2019 at 1:19 AM
To: Lee Ringham

Dear Colleague,

It is no problem with using digital images at my institute for research. Those images should be cited as follow: Chen, S.—P. 2008. Digital Insect of Taiwan Agricultural Research Institute. Taiwan Agricultural Research Institute, Taichung, Taiwan.

Best Regards,

Chi-Feng Lee

David Steen Thu, Oct 3, 2019 at 4:25 PM
To: Lee Ringham

Hi Lee,
I think that would be fine. Unfortunately I have little knowledge or expertise in pigmentation.
Cheers,
D

****

David A. Steen Ph.D. (he/him)

Tyrone Ping Wed, Oct 2, 2019 at 10:41 PM
To: Tyrone Ping Website

Good Day Lee,

Thank you for your message and inquiry.

Do let me know which image you'd like and I'll be able to send you. Decent resolution version to use in your thesis.

Cheers

Tyrone.

Kind regards,

Robert Coral Fri, Nov 29, 2019 at 2:18 PM
Reply-To: Robert Coral
To: Lee Ringham

Hello,

Yes, you may.

Thanks,

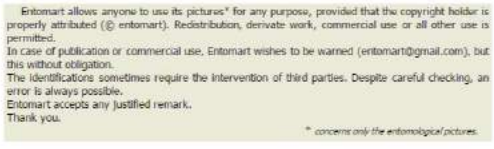
Robert Coral
President & CEO | The Serpentarium, Inc

This email and any attachments are intended only for use by the addressee(s) named herein and may contain legally privileged and/or confidential information. It is the property of The Serpentarium, Inc. If you are not the intended recipient of this email, you are hereby notified that any dissemination, distribution or copying of this email, any attachments thereto, and any use of the information contained is strictly prohibited. If you have received this email in error, please notify me at and permanently delete the original and any copy thereof.

Entomart Fri, Dec 6, 2019 at 2:36 AM
To: Lee Ringham

Hello
You can freely use Entomart images for your master's thesis.

Best regards,
Entomart,
Claude Galand


Entomart allows anyone to use its pictures* for any purpose, provided that the copyright holder is properly attributed (© entomart). Redistribution, derivate work, commercial use or all other use is permitted.
In case of publication or commercial use, Entomart wishes to be warned (entomart@gmail.com), but this without obligation.
The identifications sometimes require the intervention of third parties. Despite careful checking, an error is always possible.
Entomart accepts any justified remark.
Thank you.
* concerns only the entomological pictures.

Anthony von Plettenberg Laing Thu, Oct 3, 2019 at 7:33 AM
To:

Hi Lee,

Thank you for getting in touch.
You're more than welcome to use any of my imagery for non-commercial purposes, but please always credit it accordingly.
I do also have photos of juvenile four lined ratsnakes which may also be of interest to you, they can be found on my facebook page instead.

I'd be very interested in reading your thesis once you are finished and if you would be happy to share it.

Best wishes,
Anthony

Hello,

My name is Lee Ringham and I am writing a master's thesis on simulating snake patterning.
May I please have permission to use your photo of Micrurus fulvius in my thesis? It is for educational purposes only.

Thank you,
Lee Ringham

Lee Ringham

Hey, yeah that's fine with me!

Trent Adamson