Genetic Algorithms for the Physical Simulation of Flower Pigmentation Patterns

Grant Skaggs

A thesis submitted for the Turing Scholars honors distinction.

The University of Texas at Austin

May 6, 2022

Readers:

Professor Etienne Vouga
Professor Risto Miikkulainen
Professor Devangi Parikh

Abstract

Genetic Algorithms for the Physical Simulation of Flower Pigmentation Patterns

by Grant Skaggs

Modeling flower pigmentation patterns using physical simulation offers the potential for several advantages over texture mapping. These models may, for instance, be used to automatically create a diverse population of flowers by slightly modifying their parameters, geometries, and initial conditions. However, physical simulation of pigmentation patterns may involve finding parameters for instances of a class of differential equations known as reaction-diffusion equations, a process which can be difficult and time-consuming. This thesis analyzes the use of genetic algorithms to search for appropriate parameters for such systems. It demonstrates the effectiveness of this approach via case studies, as well as a relatively unconstrained exploration of the parameter space using an automated fitness function. The author presents both a web application and a command line tool to interface with his genetic algorithms, in order to aid future researchers in finding realistic flower pigmentation patterns.

To my mom and dad

Who taught me to love flowers and math.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to acknowledge Dr. Etienne Vouga who advised me throughout my work on this thesis, and who furthermore instructed me in his computer graphics, physical simulation, and competitive programming courses.

I would also like to acknowledge Dr. Calvin Lin, the director of the Turing Scholars Honors Program, for introducing me to computer science research and for his general support as a mentor and professor.

Section 1: Motivation

The use of physical simulation to model flower pigmentation patterns offers several advantages over texture mapping. In the context of synthesizing images of a flower, this latter process involves two steps. First, a texture must be created, either via a scanned photo or a painted image. Next, a mapping must be made from the texture to the flower's geometry. Though texture mapping can produce reasonable results, it is a fundamentally unmalleable method, in that it suffers from distortions when applied to new geometries. In contrast, physically simulated models do not suffer from any such distortions, and the parameters, geometries, and initial conditions of such models may be mutated slightly to create large populations of distinct individuals.

Additionally, simulations of flowers can offer insight into the natural science which they intend to mimic. Botanists may use models of flower pigmentation as evidence in favor of or against their theories of cellular development and molecular systems.

The current state of the art for the simulation of flower pigmentation patterns is presented in a 2021 paper by Ringham et al. entitled "Modeling Flower Pigmentation Patterns" [10]. The authors identified 7 mechanisms critical to development of flower pigmentation patterns in nature:

1. *Organ identity:* For instance, the labellum and sepals of an orchid may be colored differently than the rest of its petals, as seen in Figure 1.1.
2. *Flower age:* Flowers of different ages may be colored differently and contrast each other when bunched together.
3. *Exposure to light:* "Bud blushing" refers to the behavior some flowers exhibit whereby the parts of the petals exposed to light before bloom are colored differently.
4. *Vasculature:* Proximity to veins often affects local flower pigmentation.
5. *Position within petals:* Different areas of the petal may be colored differently. The outer edges of a flower may have a distinct color. Alternatively, the flower may have patterns which follow axes along the petals' geometries.
6. *Reaction-diffusion patterns:* These are more complex patterns, characterized by the interactions of a few key molecules and resulting in spots, stripes, or maze-like aesthetics, as in Figure 1.2.
7. *Random patterns:* Stochastic processes result in non-uniform patterns as in Figure 1.3.

Most of these processes are simple to implement and understand. Proximity to organ and position with in petal, for instance, reduce to simple interpolations of color based on geometric distances. Modeling can be performed via the space colonization algorithm presented in

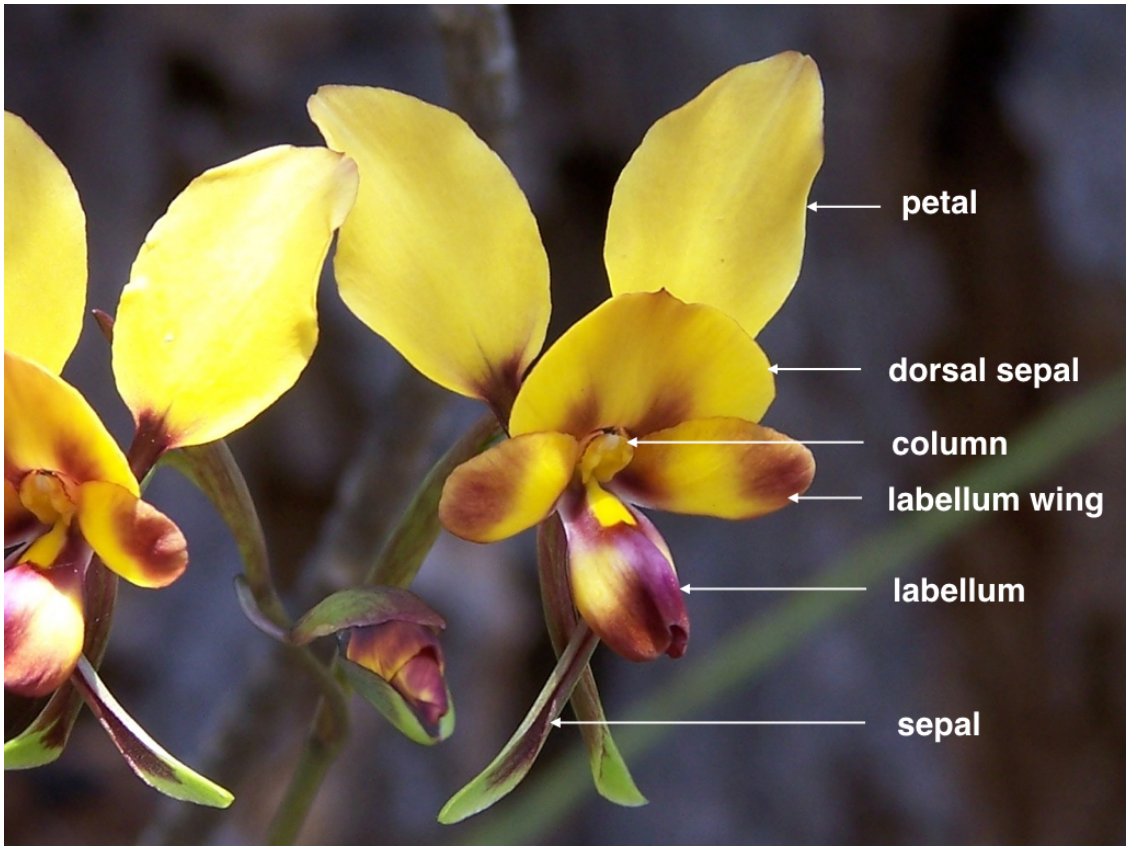

Figure 1.1: An orchid, whose labellum and sepals are colored differently than the other petals [3].

Figure 1.2: Flowers whose pigmentation may result from reaction-diffusion patterns.

Figure 1.3: A flower whose pigmentation appears stochastic in nature.

[11]. This algorithm was originally used to simulate the growth of trees and plant limbs, and involves little more than iteratively growing plant nodes toward programmer-defined points of attraction.

Much more difficult theoretically and practically are the reaction-diffusion patterns which result in the more complex patterns flowers have to offer. Models which produce such patterns are composed of systems of differential equations known as reaction-diffusion equations. These are discussed in detail in Section 2.

While systems of reaction-diffusion equations are powerful in that they can produce a broad range of patterns, finding the right parameters for these models to produce a desired pattern is difficult. In fact, many choices for parameters yield no pattern at all. As a result, Ringham et al. reported that creating their models for a given flower "ranged from hours to days," a period of time "longer than using scanned textures."

Additionally, even putting aside the problem of choosing parameters, when modeling natural phenomena using reaction-diffusion equations, researchers rely on prior knowledge of the literature to choose which system of reaction-diffusion equations might produce the desired patterns. Programmers unfamiliar with this domain would not have the background knowledge to make such choices and may struggle to produce realistic flower pigmentation patterns as a result.

This thesis attempts to use genetic algorithms to reduce the time and prior knowledge needed to produce flower pigmentation derived by reaction-diffusion patterns. It makes the following contributions to the problem of flower modeling:

1. It proposes a novel generalized system of reaction-diffusion equations called GenRD in hopes of removing the need of the programmer picking a system from the many available in the literature.

2. It analyzes the application of genetic algorithms for finding appropriate parameters for this generalized model, as well as two established models.

3. It makes available both a web application and a python implementation of my genetic algorithms so future researchers may use and build upon my work as a tool to model flower pigmentation patterns (or any other reaction-diffusion patterns).

Section 2: Background

2.1 Reaction-Diffusion Equations and Applications

The notion of reaction-diffusion equations was first introduced in 1952 by Alan Turing [14]. Turing hypothesized that tissue differentiation emerged due to the varying concentrations of compounds he called "morphogens" across a given geometry. He modeled these morphogens as having both reaction and diffusion properties. That is, their concentrations depend on interactions among each other and on their rates of diffusion.

The reaction-diffusion equation for an isolated morphogen is a differential equation of the form:

$$\dot{u} = F(u) + D\Delta u \tag{2.1}$$

Here, $u = u(x,t)$ is the concentration of the morphogen at time t and position $x \in R^n$, $\dot{u}$ is the partial derivative of the concentration of u with respect to time, F is some function meant to represent a reaction mechanism, D is a constant corresponding to the rate of diffusion of u, and Δu is the Laplacian of u.

We may also have systems of N reaction diffusion equations, as in:

$$\dot{u_i} = F_i(u_1, ..., u_N) + D_i\Delta u_i \tag{2.2}$$

Very often, a system of reaction-diffusion equations considers just two morphogens. In these cases, I'll refer to the morphogen concentrations as u and v to simplify notation. One example of such a system is the Gray-Scott model which was introduced in 1984 and is known to produce many interesting patterns when simulated and presented visually [7]. This model is given by equations 2.3 and 2.4.

$$\dot{u} = -uv^2 + F(1-u) + D_u\Delta u \tag{2.3}$$

$$\dot{v} = uv^2 - (F+k)v + D_v\Delta v \tag{2.4}$$

Here, F refers not to a function but to a constant feed rate of the first morphogen, whose concentration is given by u. The existence of the first morphogen may be seen to fuel the growth of the second, whose concentration decays with greater values of constant k. Three example patterns produced using the Gray-Scott model are given in Figure 2.1.

In addition to the Gray-Scott model, another popular model which uses two morphogens and produces interesting visual patterns is the Gierer-Meinhardt model [6], which is given by equations 2.5 and 2.6.

$$\dot{u} = \rho(v^2 - \nu u) + D_u\Delta u \tag{2.5}$$

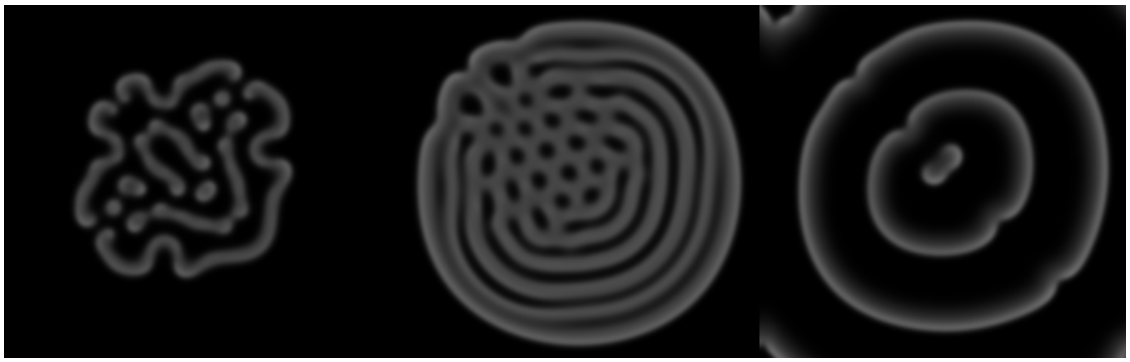

Figure 2.1: Example patterns from Gray-Scott model.

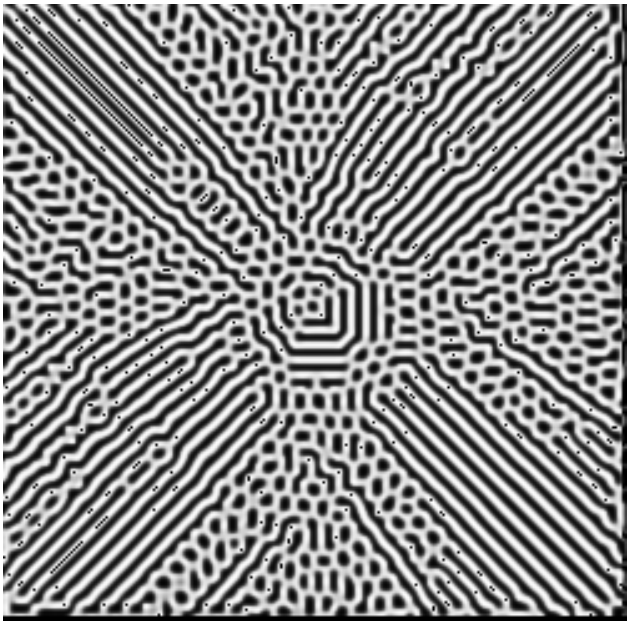

Figure 2.2: Example pattern from Gierer-Meinhardt model.

$$\dot{v} = \rho \left[\frac{v^2}{u(1+\kappa v^2)} - \mu v \right] + D_v \Delta v \tag{2.6}$$

Where ρ, μ, ν, and κ are all constant parameters of the model. An example pattern I produced using the Gierer-Meinhardt system is given in Figure 2.2.

These models and those like them have have been used to describe a broad spectrum of

phenomena in nature, including predator-prey, competition, and symbiotic relationships in population models; reversible reactions and general reactions in chemistry; and serial and plane waves in mathematics [8]. As discussed briefly in Section 1, reaction-diffusion models have been used to describe the pigmentation patterns of flowers [10]. Other applications of reaction-diffusion equations include the simulation of pigmentation patterns for zebras and leopards [15], seashells [4], and fish [12].

2.2 Genetic Algorithms for Reaction Diffusion Equations

Genetic algorithms are a type of evolutionary algorithm in which biologically-inspired functions like fitness, selection, and reproduction are performed over a population of candidate individuals over several generations in order to find solutions to search and optimization problems. In the field of machine learning, they have been used to select hyper-parameters for neural networks [5]. In Artificial Intelligence, they have been used, for instance, to create high-performance Tetris agents, where traditional reinforcement learning algorithms have failed [1]. And in computer graphics, they have been used to evolve textures defined by simple symbolic expression, as well as animations built upon those textures [13].

There is one paper in the literature which attempts to apply genetic algorithms to reaction diffusion equations. Unfortunately, the subject of this paper was not pattern formation, but instead the use of the Gray-Scott model to control the motor patterns of what is called a "minimally-cognitive agent" [2]. Each cell of the agents in this study had associated with them a different set of parameters for the Gray-Scott model, as determined by the cells' chromosomes. The concentrations of morphogens u and v in particular cells controlled motor patterns, and in other cells were affected by the exterior environment. While it's impressive that such a design when evolved using genetic algorithms performed well on certain navigation tasks, this result offers little insight into the use of genetic algorithms for visual pattern formation.

To my knowledge, no researchers have explored the use of genetic algorithms to produce visual patterns via reaction-diffusion equations. Until now researchers have been able to search the space of parameters by choosing parameter values (e.g. F and k in the Gray-Scott model) by ad hoc exploration. This may be sufficient to make shell or leopard geometries which are relatively generic. However, based on Ringham et al's report (as relayed in Section 1) about the time taken to parameterize their models, modeling the intricacies of flower pigmentation patterns using reaction-diffusion systems may require more time and human-effort.

Section 3: Methods

3.1 Genetic Algorithm

I applied genetic algorithms to three reaction diffusion models:

1. Gray-Scott, given by equations 2.3 and 2.4.
2. Gierer-Meinhardt, given by equations 2.5 and 2.6.
3. GenRD, given below.

GenRD is a generalization of reaction-diffusion systems of two variables. The observation which led to its creation was the following: All the two-morphogen reaction-diffusion systems used by Ringham et al. to make their flowers in [10] had reaction terms which were degree 2 or smaller rational functions in u and v. The natural generalization of this pattern was GenRD:

$$\dot{u} = \sum_{t=1}^{3} \left[\frac{\sum_{i=0}^{2} \sum_{j=0}^{2} \alpha_{u,t,i,j} u^i v^j}{\sum_{i=0}^{2} \sum_{j=0}^{2} \beta_{u,t,i,j} u^i v^j} \right] + D_u \Delta u \tag{3.1}$$

$$\dot{v} = \sum_{t=1}^{3} \left[\frac{\sum_{i=0}^{2} \sum_{j=0}^{2} \alpha_{v,t,i,j} u^i v^j}{\sum_{i=0}^{2} \sum_{j=0}^{2} \beta_{v,t,i,j} u^i v^j} \right] + D_v \Delta v \tag{3.2}$$

Where α and β are 4-dimensional lists of parameter values. $\alpha_{x,t,i,j}$ and $\beta_{x,t,i,j}$ are fourth order tensors which denote parameter (i, j) for term t for morphogen x.

Each simulation was done on a square grid of size 256 by 256 cells, which was programmed to have periodic boundary conditions. The morphogen concentrations were either initialized via a random uniform distribution over $[0, 1]$ or more complex function which in practice centers patterns and gives them and gives each a line of symmetry about an axis tilted away from either horizontal or vertical. Unless otherwise specified, the results in Section 4 were derived from simulations initialized according to this latter function, which is given here explicitly:

$$u_0^{x,y} = 1 - e^{-80 \cdot ((x+0.05)^2 + (y+0.05)^2)} \tag{3.3}$$

$$v_0^{x,y} = e^{-80 \cdot ((x-0.05)^2 + (y-0.05)^2)} \tag{3.4}$$

Where $u_0^{x,y}$ and $v_0^{x,y}$ are the initial concentrations of u and v for a cell on the grid centered at x, y, supposing the grid itself is a unit square and centered about $(0, 0)$.

Model	Genes	D_u	D_v
Gray-Scott	F, k	0.32768	0.16384
Gierer-Meinhardt	ρ, κ, μ, ν	0.05	2.0
GenRD	$\alpha_{x,t,i,j}$ and $\beta_{x,t,i,j}$ for all defined x, t, i, j	1.0	1.0

Table 3.1: Genes and diffusion constants for each reaction-diffusion model.

Parameter	F	k	ρ	μ	ν	κ	$\alpha_{x,t,i,j}$	$\beta_{x,t,i,j}$
σ^2	0.05	0.05	0.2	0.1	0.1	0.5	0.5	0.5
Initial Range	[.02, .08]	[.04, .06]	[.45, .55]	[.95, 1.0]	[.85, .95]	[.2, .8]	[-1, 1]	[-1, 1]

Table 3.2: Mutation rate for each gene.

Each individual contained a collection of parameter values for a given reaction-diffusion model. These parameters values comprised that individual's genes, and because each reaction-diffusion model has a different number of parameters, the number of genes per chromosome varied depending on the experiment. The genes for each reaction-diffusion model are given by Table 3.1. Note that the diffusion constants D_u and D_v are not treated as genes. Instead, for each model, I used hand-picked values which produced good results in my simulation and, in the case of Gray-Scott and Gierer-Mienhardt models, were also used to produce interesting patterns in [12] and [12], respectively. These values may likewise be seen in Table 3.1.

Crossover was done naively; each chromosome has an equal chance of coming from either parent. The mutation to the value of a gene γ is given by:

$$\gamma := \gamma + N(0, \sigma^2) \tag{3.5}$$

Where σ^2 is the mutation rate for that gene. I experimented with several different mutation rates for my models and determined that those given in Table 4.3 find a useful balance of exploration and stability in my genetic algorithm. That these values are different among the parameters implies that the reaction-diffusion patterns are more sensitive to changes in one parameter as compared to another.

At the start of each evolutionary simulation, the individuals of a population have genes initialized via a random uniform sampling from a gene-specific range of values. Like the learning rate, I had to experiment with different values for these ranges.

I experimented with two different fitness functions for my chromosomes. The first, which I call the modified Dirichlet energy of a grid, was intended to be a hands-off, automated means of discovering interesting reaction-diffusion patterns. This function is based on a quantity know as Dirichlet energy, which measures how variable a function is. It's given by:

$$\frac{1}{2}\int_{\Omega} \|\nabla u(x)\|^2 dx \tag{3.6}$$

When using the Dirichlet energy as a measure of fitness, I found that grids with interesting patterns but low absolute value for morphogen concentrations had lower gradient values, and therefor lower fitness scores, than their less interesting counterparts. To amend this problem, I introduced the modified Dirichlet energy accounts for differences in the absolute value of a grid and is given by:

$$\frac{1}{\max_{\Omega}\|\nabla u(x)\|^2}\int_{\Omega} \|\nabla u(x)\|^2 dx \tag{3.7}$$

The discrete version of the modified Dirichlet energy is given by:

$$\frac{1}{\max_{\Omega}\|\nabla u(x)\|^2}\sum_{\Omega} \|\nabla u(x)\|^2 \tag{3.8}$$

Where ∇x is now the discrete gradient of the discrete vector-valued function u and Ω is now a grid of values instead of a continuous region in space.

The second fitness function I used was selection by the application user. Here, we let a human guide the genetic algorithm toward a desired pattern by picking at the end of each generation a non-null set of chromosomes they feel is in "the right direction" with respect to the goal state. If there are N individuals in the population and S is the list of selected individuals, then the fitness of a given individual is by:

$$f_i := \frac{\lambda}{\text{length}(S)} \text{ if } i \in S, \text{ otherwise } \frac{1-\lambda}{N - \text{length}(S)} \tag{3.9}$$

Where λ is a constant set to .9 in my program.

To produce the next population after a generation is simulated and evaluated, I used fitness proportionate selection, whereby the expected number of times an individual will reproduce is given by:

$$p_i := \frac{f_i}{\sum_{j=1}^{N} f_j} \tag{3.10}$$

Where N is again the size of the population and f_i is the fitness of individual i.

Simulation parameters which the user may change easily in either the text or graphical user interfaces are the simulation length and step size, which are referred to in the remainder of this thesis as T and dt respectively.

3.2 Experiments and Simulation Details

To evaluate the effectiveness of the modified Dirichlet energy as a fitness function which produces interesting reaction-diffusion patterns, I ran my genetic algorithm with 30 individuals and 10 generations, for each of the three models I investigated.

Model	dt	T
Gray-Scott	0.5	2000
Gierer-Meinhardt	0.05	500
GenRD	0.01	50

Table 3.3: dt and T for each reaction-diffusion model.

To evaluate the effectiveness of genetic algorithms as tools for finding specific flower pigmentation patterns, I chose three different flowers as case studies whose patterns I wanted to replicate. Since each of these flowers also appears in Ringham et al.'s "Modeling Flower Pigmentation Patterns," my algorithm's performance on these tasks provides some insight into the value this thesis has for future researchers in the space.

The results of all of these experiments are given in the next section.

The visualizations you will see there of reaction-diffusion patterns are of the concentration v. This decision was arbitrary since visualizing u also results in interesting parameters, and furthermore systems which are interesting when visualizing one of two morphogens are almost inevitably interesting when visualizing its counterpart.

The choice for the simulation length T and time step dt were hand-tuned for each of the three reaction-diffusion models considered. In general, the largest dt which resulted in consistently stable simulations was used. Similarly, the smallest T which resulted in visually interesting patterns was used. These values are given in Table 3.3.

Since some parameter choices for all three models may result in negative values or an unconstrained "blow up" of the concentrations u and v toward either infinity, I clamped the concentrations to the range $[0, 5]$ in my simulation, finding systems which previously took on undesirable values to instead produce grids filled entirely with one end of this range.

Section 4: Results

4.1 Modified Dirichlet Energy

Figures 4.1 and 4.2 show the first and tenth generations of an evolutionary simulation on the Gray-Scott model which uses the modified Dirichlet energy as its fitness function. Figures 4.3 and 4.4 are the same as applied to the Grier-Mienhardt model, and likewise Figures 4.5 and 4.6 are the same as applied to the GenRD model.

4.2 Case Study: Phalaenoris Nanking's 4.55

This flower is a hybrid of two orchid species. It has irregular blotches distributed non-uniformly among its pedals. Although no reaction-diffusion model is known to create this pattern in a stable terminal state, it has been shown to emerge from early iterations of the Gray-Scott model from a random pattern of initial concentrations [9]. I attempted to reproduce these results by applying the genetic algorithm to both the Gray-Scott using user guidance to inform its fitness function. I set the initial concentrations of the morphogens to random values and stopped the simulation at a relatively small T=200. To get results which appeared reasonable took 5 generations.

Figure 4.7 contains three images. The first is the simulation output as visualized using my default coloring, where brighter indicates higher v values. The second image is the same output where each cell is either purple or white depending on its value crossing some threshold which I handpicked. The final image is the that of *Phalaenoris* Nanking's 4.55 which is the ground truth pattern I attempted to replicate.

4.3 Case Study: Kohleria

The *Kohleria* flower is tropical and has striking red and white, maze-like pattern. To reproduce its pattern, Ringham et al. used a reaction-diffusion equation from Turing's original paper. Since this was not one of the models I investigated, I originally intended to run a user-guided genetic algorithm with both GenRD and Grierer-Meinhardt to replicate these results. However, during the fourth generation of an evolutionary simulation for the Gierer-Meinhardt model with the modified Dirichlet energy as fitness function, I discovered a pattern which resembled *Kohleria* remarkably well.

This pattern is the first image in Figure 4.8. The second image in that figure is produced by the same reaction-diffusion simulation, except run for a longer $T = 1000$ instead of the original $T = 500$ and colored with a different palette. The third image is the ground truth picture of *Kohleria*.

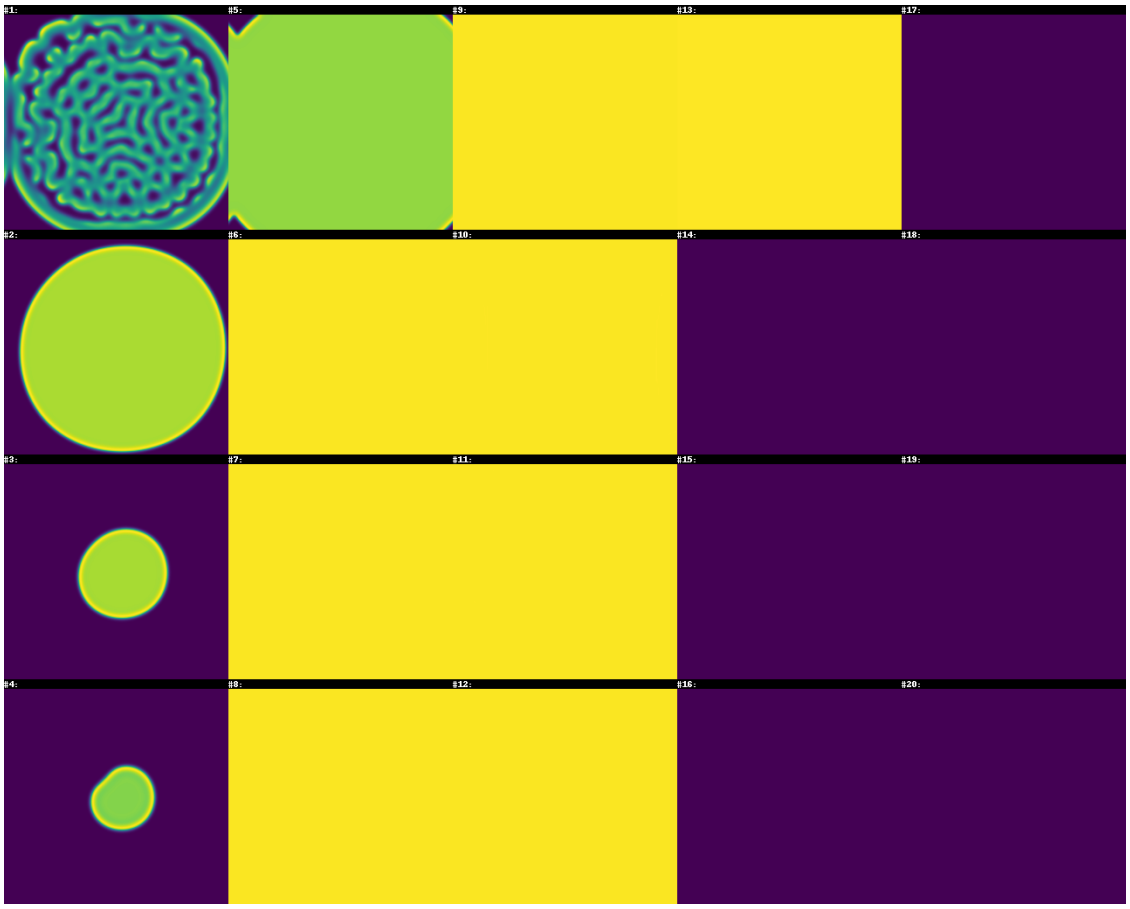

Figure 4.1: Modified Dirichlet energy as fitness function, generation 1: Gray-Scott

4.4 Case Study: Mimulus guttatus

Mimulus guttatus is a wildflower with a distinctive spotted pattern. As in the *Kohleria* case study, I actually found a pattern which approached the spotted pattern of *Mimulus guttatus* in the fourth generation of the same evolutionary simulation for the Gierer-Meinhardt model.

This original pattern I discovered is the first image in Figure 4.8. The second image in that figure is produced by the same reaction-diffusion simulation, except run for a longer $T = 1000$ instead of the original $T = 500$ and colored with a different palette. The third image is the ground truth picture of *Mimulus guttatus*.

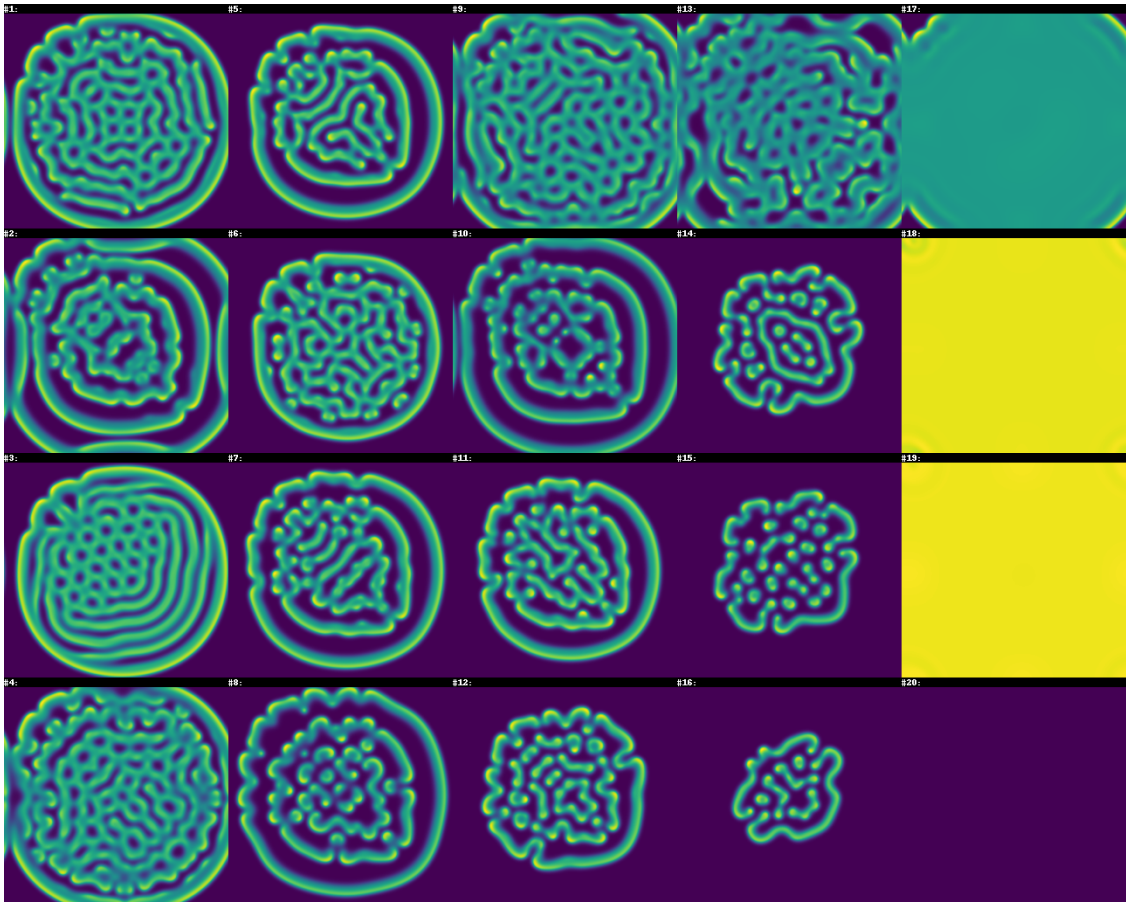

Figure 4.2: Modified Dirichlet energy as fitness function, generation 10: Gray-Scott

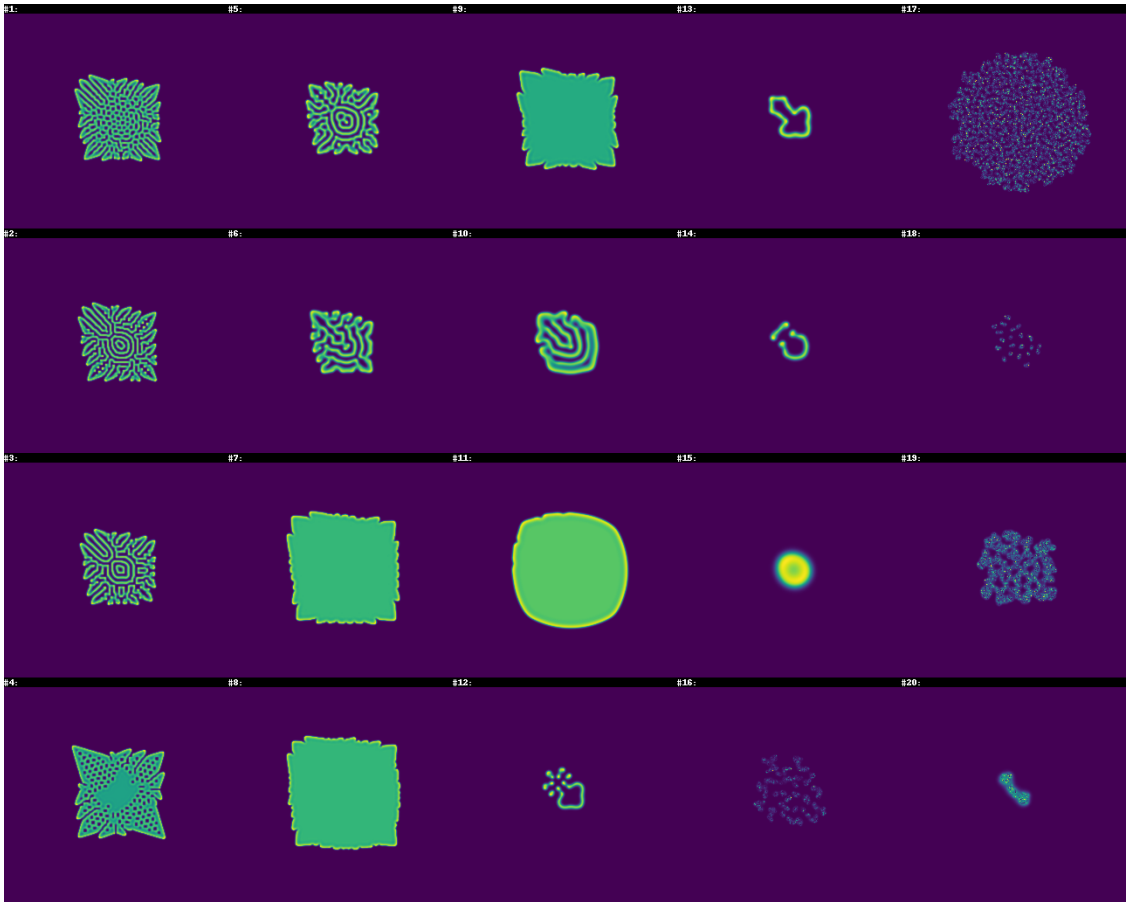

Figure 4.3: Modified Dirichlet energy as fitness function, generation 1: Grier-Meinhardt

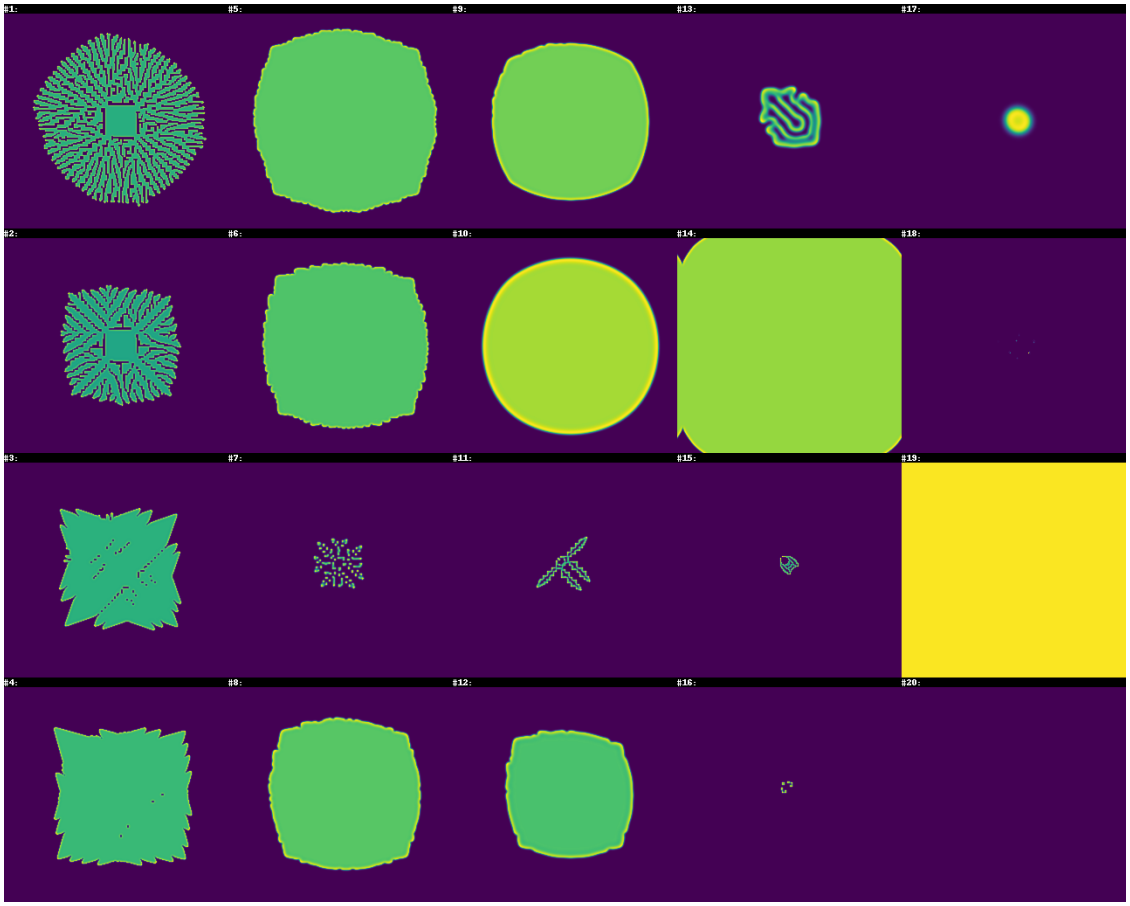

Figure 4.4: Modified Dirichlet energy as fitness function, generation 10: Grier-Meinhardt

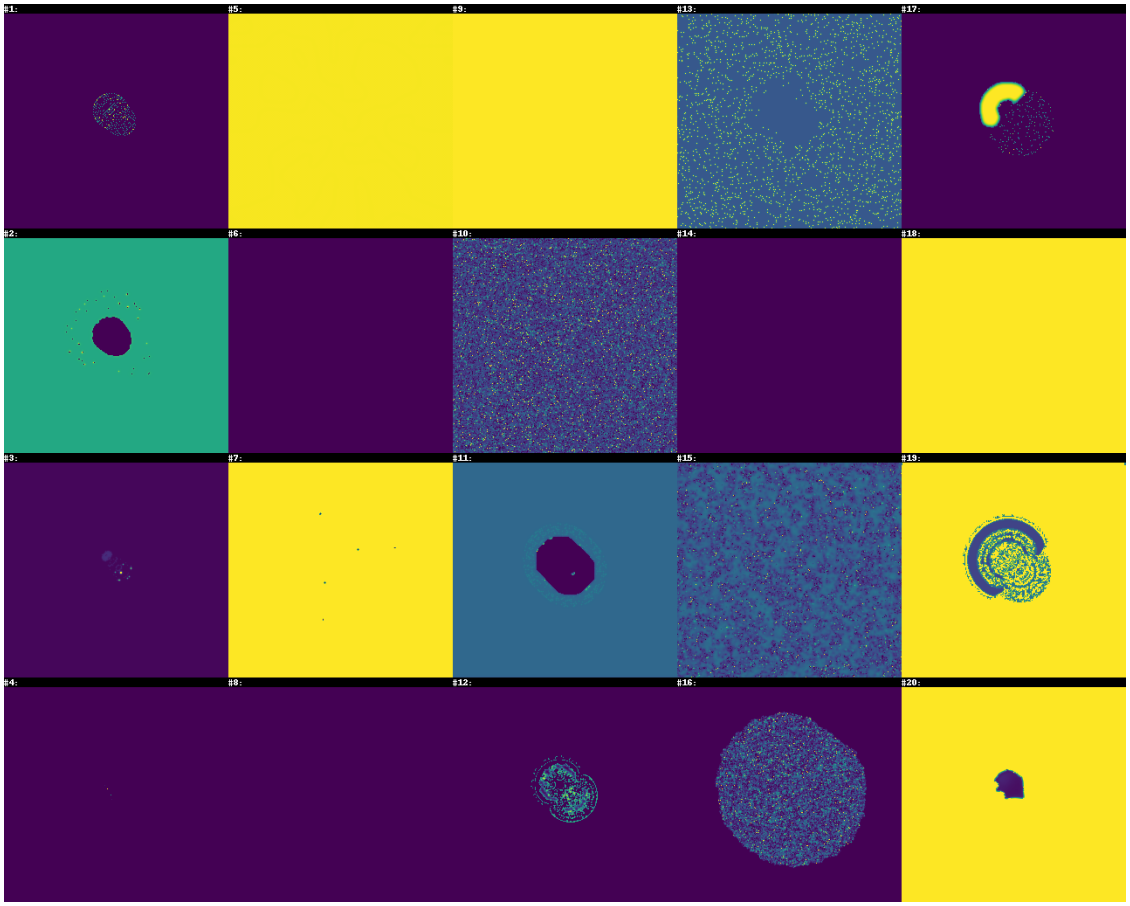

Figure 4.5: Modified Dirichlet energy as fitness function, generation 1: GenRD

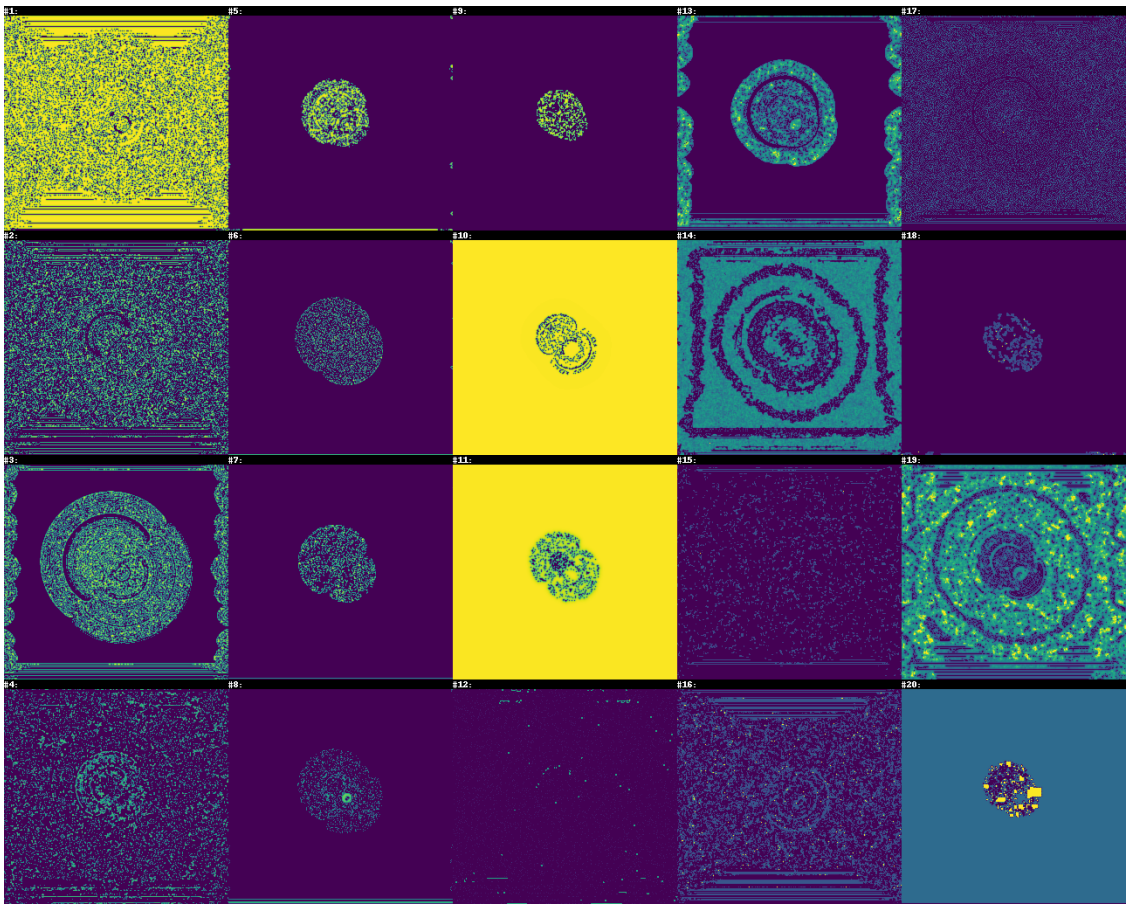

Figure 4.6: Modified Dirichlet energy as fitness function, generation 10: GenRD

Figure 4.7: Original simulation, altered coloring, and ground truth pattern for case study 1.

Figure 4.8: Original simulation, simulation with altered coloring and termination point, and ground truth pattern for *Kohleria*

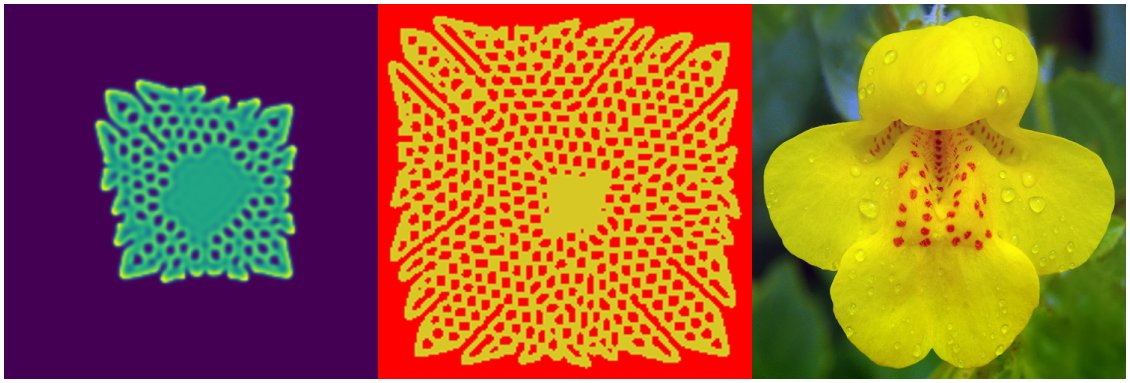

Figure 4.9: Original simulation, simulation with altered coloring and termination point, and ground truth pattern for *Mimulus guttatus*.

Section 5: Software Implementation and Performance

The results obtained in Section 4 were done so using a program I wrote called gard (genetic algorithms for reaction-diffusion). Gard was implemented on top of a python-based Gray-Scott reaction-diffusion simulator written by researchers at ETH Zurich. The original repository for this code may be found at **https://github.com/cselab/gray-scott**. I added to this code the functionality to simulate both the Gierer-Mienhardt model as well as my own GenRD model. The core simulator itself was modified to interface with chromosomes from my genetic algorithm code, which was also implemented in python. The grid of morphogen concentrations were stored in this core simulator as NumPy arrays of data type float64.

To improve the speed of my simulation, I implemented multi-threading of my code where each individual was simulated in parallel, up to the number of threads specified by the user. Doing so resulted in a significant speed-up for simulating all the models I analyzed. Even so, my machine still took longer than a minute to simulate an individual of the GenRD model. Consequently, I implemented a version of all models on the GPU using a library called PyOpenCL. The time my machine takes averaged over fiver iterations to run a generation of 20 individuals for each model is given in Table 5.1. Unsurprisingly, the GPU implementation offers more speedup on the computationally expensive GenRD since the overhead cost of moving data to the GPU is overpowered by the immense gains in computation speed.

The command line interface provided by the original implementation was sufficient for the purposes of a Gray-Scott simulator. Unfortunately, it became clunky with the added complexity of genetic algorithms and user-provided selection. To make the use of my genetic algorithm as a tool feasible, I implemented a graphical user interface to gard using a library called streamlit.

To make gard accessible as a tool to future researchers, I deployed it as a web application with the aforementioned GUI as its interface. You can access this web application at **https://reaction-diffusion.herokuapp.com/**. To avoid paying for a server with GPUs, this online version of gard runs on multi-threaded CPUs and is considerably slower than the

Implementation:	CPU-1	CPU-16	GPU
Avg. Seconds for Gray-Scott:	39.72	27.03	6.39
Avg. Seconds for Gierer-Meinhardt:	18.63	11.58	6.81
Avg. Seconds for GenRD:	118.19	84.31	21.20

Table 5.1: Average time in seconds implementations take to simulate each model.

results one obtains from using a GPU or even faster CPUs. However, the full implementation of gard is available at **https://github.com/gskaggs/reaction-diffusion** for anyone to download and run on their own machines.

Section 6: Discussion and Future Work

6.1 Issues

I encountered two fundamental problems with GenRD. First, the model was prohibitively slow. Terminal time values T greater than 100 could mean minutes to simulate one individual, even on a GPU. This reason using GenRD is so slow is because each time step requires many more computations than either Gray-Scott or Gierer-Meinhardt. There is no getting around this. It was designed to encompass many possible reaction terms and any attempt to reduce the models complexity would also reduce its power of expression.

The second problem I encountered with GenRD was its volatility as a model. The patterns observed in both Figure 4.5 and 4.6 are often predominately yellow (indicating the model "blew-up" and hit ceilings on the concentrations which I imposed) or else they're grainy and resemble static, even if they do appear to look something like a normal reaction-diffusion pattern. No attempt I made to reduce the static phenomenon helped, including reducing the time step dt, increasing the grid resolution, and actively trying to choose non-static patterns when the fitness function was user-input. In fact, the first two of these hypothesized solutions just slowed down the simulation process more, making exploration of the space more difficult. I might have blamed my explicit integration and implemented a semi-implicit approach if I had experienced similar issues with the other two reaction-diffusion models, but this was not the case.

A fundamental problem I experienced with the user-guided approach to genetic algorithms for reaction diffusion patterns was (again) speed. Even when simulating a simple model like Gray-Scott, doing a simulation of 100 individuals for 10 generations may take an hour or more and requires periodic attention from the user. This is certainly faster than blindly searching a space by trying out values until you find patterns resembling what you're looking for. However, it may not be faster than someone familiar with what the literature suggests are good parameters to consider for a given model, and it is much less convenient than having an automatic fitness function go on autopilot.

The main issue I had with my proposed modified Dirichlet energy is that it sometimes produces less interesting geometries in order to maximize the gradient. That's why the tenth generation selected by this function (as in Figures 4.2 and 4.4) contain less diversity than the randomly initialized first generation (Figures 4.1 and 4.3). This explains also why the patterns I declared as best matching the ground truth patterns in the second two cases studies came out of the fourth generation of an evolutionary algorithm using the modified Dirichlet energy, and out of not the tenth or twentieth.

A problem I observed with genetic-algorithms for reaction-diffusion systems in general

was that this approach did not actually reduce the need for prior knowledge about the reaction-diffusion model being simulated. A successful genetic algorithm relies on reasonable learning rates. Too large and you miss interesting elements of the search space. Too small and you do the same. With respect to using the Gray-Scott model, for example, knowing roughly how large the set of values for F which result in patterns makes setting the learning rate simple. If all good F values are between 0.01 and 0.1, then a learning rate of .25 or .0001 for that parameter doesn't make sense. The same goes for any other parameter of a reaction-diffusion system. Without this prior knowledge, a user may be stuck trying different values for the learning rates until they find one which appears to work well. This is exactly the kind of ad hoc exploration my thesis set out to eliminate.

6.2 Visual Performance

Analyzing the degree to which a simulated reaction-diffusion pattern matches nature is difficult. I rely on visual comparison, as did Ringham et al. in their work. In the first case study, that which examined *Phalaenoris* Nanking's 4.55, it's clear that the morphology of the splotches in my pattern did not match that of the ground truth which were more circular. However, my pattern honored that the splotches varied in size and were non-uniformly dispersed. In the second case study, which analyzed the patterns of the *Kohleria* flower, my simulation produced moderately better results. My model had a mixture of maze-like lines and dots which were also observed in the ground truth, even if they were mixed together in a very different way. That is, while the flowers' dots are seen more toward the petal's edges, my pattern's dots are seen throughout the maze-like lines. Lastly, in reproducing the *Mimulus guttatus* flower, my models succeeded in producing uniformly distributed dots, similar to those seen in the actual plant. However, I was not able capture the spacing of these dots with respect to each other, nor was I able to capture how the dots only appeared in specific parts of the petal.

6.3 Limitations

The shortcomings of my patterns in these case studies are explained by a number of mechanisms which Ringham et al. used to great effect and which my model ignored. These mechanisms are:

- *Petal geometry:* My models existed on square grids as opposed to on triangular meshes.
- *Wrap-around:* To be consistent with prior simulations using 2D grids, my models used a wrap-around morphology where concentrations at borders affected concentrations on the other side of the grid. Petals do not have this functionality.
- *Variable model parameters:* The values of my model parameters were consistent when applied throughout the grid. To get more complex patterns, Ringham et al. used

what they called an "intermediate morphogen" to construct a gradient across a petal geometry, and then used that gradient to gradually vary the parameters for their reaction-diffusion simulations.

- *Isotropic diffusion:* The diffusion terms in my models were all isotropic. That is, morphogens diffused equally along both x and y axes. Diffusion in which this is not the case is called anisotropic and is essential to modeling certain flower patterns.

- *Proximity to vasculature:* In some flowers, good models result from varying the morphogen concentrations in close proximity to vasculature, which I did not model.

These differences meant my model was unable to capture all the reaction-diffusion patterns necessary for some flower pigmentation patterns. One potential avenue for future research, then, would be to explore the use genetic algorithms for a more realistic set of models. A researcher might, for instance, synthesize a set of flowers by hand and by an evolutionary approach in order to determine which is faster. Alternatively, she might attempt to model some of these more complex mechanisms using genes and add them to individuals' chromosomes.

6.4 Conclusion

The aim of this thesis was to evaluate genetic algorithms as a mechanism for determining good parameters for models of flower pigmentation. Since I did not use the same models to simulate flower geometry as Ringham et al., I cannot determine with absolute certainty if the insights I gained through my experimentation would hold for their development process as well. That said, it seems the issues outlined in Section 6.2 were fundamental to any evolutionary approach to this problem, and consequently I conclude that the genetic algorithm I developed, as well as the generalized reaction-diffusion model I introduced, are not in general likely to speed up the process of parameter selection for reaction-diffusion models.

However, I do not feel the issues I encountered are insurmountable and would not advocate for giving up on the potential advantages of either an evolutionary approach to this problem or a generalized reaction-diffusion model. Instead, I propose the following iterations upon my work as potential avenues toward more fruitful research:

- Exploring other automatic fitness functions besides the modified Dirichlet energy I used. Automatic fitness functions makes simulation time less of an issue since the user does not have to be around to give it periodic attention. A fitness function which was able to reward interesting patterns and dismiss static, for instance, might've done much to expedite my exploration of GenRD's parameter space. Of course, an automatic fitness function which could be initialized to target patterns which look similar to a photograph or drawing would be ideal.

- Designing algorithms which modulate the learning rate for a parameter based on the degree to which changing its value affects simulations. Ideally this would be done automatically, but feedback from the user on how much or little patterns changed in between generations could also be used. In either case, this would lessen the advantage of having prior knowledge about a given system of reaction-diffusion equations.

- Using combinations of popular reaction-diffusion systems instead of a more complex generalized system like GenRD. For instance, a linear combination of the reaction terms of the Gray-Scott and Gierer-Meinhardt models might be able to express more patterns than either of them on their own, all while mainting expressiveness of all patterns each is known to produce. The computation and simulation time necessary for such a model would be much less than something more bulky like GenRD.

Bibliography

[1] Niko Bohm, Gabriella Kókai, and Stefan Mandl. "An Evolutionary Approach to Tetris". In: 2005.

[2] Kyran Dale and Phil Husbands. "The Evolution of Reaction-Diffusion Controllers for Minimally Cognitive Agents". In: *Artificial life* 16 (Oct. 2009), pp. 1–19. DOI: 10.1162/artl.2009.16.1.16100.

[3] Geoff Derrin. *Diuris (labeled)*. web. licence: https://commons.wikimedia.org/wiki/File:Diuris_(labelled).jpg. July 2016.

[4] Deborah R. Fowler, Hans Meinhardt, and Przemyslaw Prusinkiewicz. "Modeling Seashells". In: *SIGGRAPH Comput. Graph.* 26.2 (July 1992), pp. 379–387. ISSN: 0097-8930. DOI: 10.1145/142920.134096. URL: https://doi.org/10.1145/142920.134096.

[5] Keshav Ganapathy. *A Study of Genetic Algorithms for Hyperparameter Optimization of Neural Networks in Machine Translation*. 2020. DOI: 10.48550/ARXIV.2009.08928. URL: https://arxiv.org/abs/2009.08928.

[6] Alfred Gierer and Hans Meinhardt. "A theory of biological pattern formation. Kybernetik 12, 30- 39". In: *Biological Cybernetics* 12 (Jan. 1972), pp. 30–39. DOI: 10.1007/BF00289234.

[7] Peter Gray and Stephen K. Scott. "Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system A + 2B → 3B; B → C". In: *Chemical Engineering Science* 39 (1984), pp. 1087–1097.

[8] Christina Kuttler. "Chapter 4 - Reaction–Diffusion Equations and Their Application on Bacterial Communication". In: *Disease Modelling and Public Health, Part B*. Ed. by Arni S.R. Srinivasa Rao, Saumyadipta Pyne, and C.R. Rao. Vol. 37. Handbook of Statistics. Elsevier, 2017, pp. 55–91. DOI: https://doi.org/10.1016/bs.host.2017.07.003. URL: https://www.sciencedirect.com/science/article/pii/S0169716117300123.

[9] Lee Ringham, Przemyslaw Prusinkiewicz, and Robert Gniadecki. "Skin Patterning in Psoriasis by Spatial Interactions between Pathogenic Cytokines". In: *iScience* 20 (Oct. 2019). DOI: 10.1016/j.isci.2019.10.008.

[10] Lee Ringham et al. "Modeling Flower Pigmentation Patterns". In: *ACM Trans. Graph.* 40.6 (Dec. 2021). ISSN: 0730-0301. DOI: 10.1145/3478513.3480548. URL: https://doi.org/10.1145/3478513.3480548.

[11] Adam Runions, Brendan Lane, and Przemyslaw Prusinkiewicz. "Modeling Trees with a Space Colonization Algorithm". In: Jan. 2007, pp. 63–70. DOI: 10.2312/NPH/NPH07/063-070.

[12] Allen Sanderson et al. "Advanced Reaction-Diffusion Models for Texture Synthesis". In: *J. Graphics Tools* 11 (Jan. 2006), pp. 47–71. DOI: 10.1080/2151237X.2006.10129222.

[13] Karl Sims. "Artificial Evolution for Computer Graphics". In: *SIGGRAPH Comput. Graph.* 25.4 (July 1991), pp. 319–328. ISSN: 0097-8930. DOI: 10.1145/127719.122752. URL: https://doi.org/10.1145/127719.122752.

[14] A. M. Turing. "The Chemical Basis of Morphogenesis". In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 237.641 (1952), pp. 37–72. ISSN: 00804622. URL: http://www.jstor.org/stable/92463 (visited on 04/29/2022).

[15] Greg Turk. "Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion". In: *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '91. New York, NY, USA: Association for Computing Machinery, 1991, pp. 289–298. ISBN: 0897914368. DOI: 10.1145/122718.122749. URL: https://doi.org/10.1145/122718.122749.