

Advanced Reaction-Diffusion Models for Texture Synthesis

Allen R. Sanderson*, Robert M. Kirby*, Chris R. Johnson* and Lingfa Yang+

*Scientific Computing and Imaging Institute, University of Utah

+Center for Complex Systems, Brandeis University

Abstract

Since the introduction of texture synthesis using a reaction-diffusion model in the early 1990s their use has not been widespread. This is likely due to both the difficulty in selecting parameters that result in stable, useful patterns as well as, the computational costs of producing these patterns. In this paper we present techniques that help overcome the difficult parameter selection process that controls the pattern development. In addition, we expand the basic diffusion model to allow for shaped patterns. Finally, we show that is possible to create multiple and oscillating patterns by coupling two reaction-diffusion systems together. These techniques have been implemented using both explicit and semi-implicit solutions on a CPU and GPU. We provide sample source code of both implementations online.

1 Introduction

In the early 1990s Turk [19], Witkin and Kass [20], and Fowler et. al. [4] introduced texture synthesis using a reaction-diffusion model to computer graphics. Since then, their use has not been wide spread, with few references in the computer graphics literature. At the same time, there have been a significant number of references for the use of reaction-diffusion models in other applications, including mathematical biology [13] and chemical dynamics [2]. In addition, since the original papers on reaction-diffusion texture synthesis, there have been computational and theoretical advances in applying reaction-diffusion methods to a variety of applications [9].

The lack of wide spread use of reaction-diffusion techniques within computer graphics is likely due to both the difficulty in selecting reaction-diffusion parameters that result in stable, useful patterns, as well as the computational costs of producing these patterns using numerical reaction-diffusion techniques. Indeed, because the equations governing reaction-diffusion models are nonlinear and require a sometimes computationally costly iterative process to reach a dynamic equilibrium, creating a useful texture can be very challenging.

In this paper we present techniques that help overcome the difficult parameter selection and computational costs associated with reaction-diffusion texture synthesis. Specifically, we have developed a GPU implementation [17] of an extended reaction-diffusion model that allows one to quickly form useful patterns while at the same time exploring the parameter space. Our source code is available online at the website listed at the end of this paper.

2 Reaction-Diffusion Basics

In 1952, Turing published his now classic paper describing the chemical process between signaling molecules that spread away from their source to form a concentration gradient (morphogens) within a series of cells [18]. Due to instabilities in the system, the morphogens both react and diffuse, which changes their concentration within each cell. Turing suggested that morphogens could

react with each other and diffuse through cells forming patterns through the reaction-diffusion process. The resulting “Turing patterns” from various reaction specifications have been shown to produce a wide array of patterns, including regular patterns reminiscent of those found on many animals (zebra stripes, leopard spots, etc.) [13]. Since Turing’s 1952 paper, others have proposed alternative reaction-diffusion models, including the popular Gray and Scott model [7, 8]. A historical overview can be found in [2].

We now describe some of the basic reaction-diffusion models and some of their characteristics beginning with Turing’s model, where he describes a two morphogen model in the form a set of nonlinear partial differential equations:

$$\frac{\partial a}{\partial t} = F(a, b) + d_a \nabla^2 a \quad (1)$$

$$\frac{\partial b}{\partial t} = G(a, b) + d_b \nabla^2 b \quad (2)$$

where a and b are the morphogens concentration; F and G are the functions controlling the production rate of a and b ; d_a and d_b are the diffusion rates, and $\nabla^2 a$ and $\nabla^2 b$ are the Laplacians of a and b representing isotropic diffusion. Turing [18] further defines F and G for his particular application of interest as:

$$F(a, b) = s(ab - a - \alpha) \quad (3)$$

$$G(a, b) = s(\beta - ab) \quad (4)$$

where a and b again are the morphogen concentration, α and β can be thought of as being a decay and growth rate of a and b respectively, and s is the reaction rate.

Pearson [14] describes the reaction model due to Gray and Scott [7, 8] that has the form:

$$F(a, b) = s(a^2 b - (F + k)a) \quad (5)$$

$$G(a, b) = s(-a^2 b + F(1 - b)) \quad (6)$$

where F and k are the feed and degrading rates, respectively.

These two reactions are considered to be activator-substrate models or depletion models. As the morphogen a activates, the morphogen b is depleted and visa-versa. This leads to patterns that are inverses of one another as shown in Figure (1). Another well known depletion model is the Brusselator model [15] that has the form:

$$F(a, b) = s(\alpha - (1 + \beta)a + a^2 b) \quad (7)$$

$$G(a, b) = s(\beta a - a^2 b). \quad (8)$$

This model, as will be discussed later, is particularly interesting because it can be used in a coupled system to produce double spot patterns.

The other common type of reaction is an activator - inhibitor model [6]. The most well known is the Gierer-Meinhardt model, which has the form:

$$F(a, b) = s(\gamma + a^2/b - \alpha a) \quad (9)$$

$$G(a, b) = s(a^2 - \beta b). \quad (10)$$

Within the reaction-diffusion research literature, one can find many other reaction models that produce patterns, such as the Oregonator model of the Belousov-Zhabotinsky reaction [3], as well as models that contain more than two morphogens. For the purposes of this paper, we have focused on models that contain two morphogens for producing stable patterns or three morphogens for oscillatory models. Further details regarding reaction-diffusion models and their solution can be found in, for example, [2], [9], and [13].

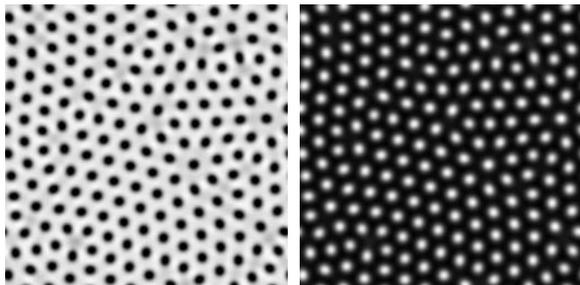


Figure 1: (a) Activator-Depletion Reaction-Diffusion model showing the activator morphogen a and (b) the depletion morphogen b for a spot pattern using Turing's model. Parameters for generating the patterns can be found in the appendix.

3 Controlling a Reaction-Diffusion System

We now examine each part of the reaction-diffusion model and how each can be controlled to form a particular pattern. We look at each part individually so that its affect can be seen. In most cases this is the best way to build a particular pattern. Start with one basic part and then add to it until the desired pattern is formed.

3.1 System Instabilities

The initial concentration for each of the morphogens is set such that they are uniform and they form a steady state solution. In order for the system to react and diffuse and hence form a pattern, there must be some instability or asymmetry that drives it from this initial steady state. Which variables, and how much they are perturbed, greatly affects the resulting pattern.

For instance, in Turing's original implementation the initial morphogen concentrations were spatially perturbed from their initial uniform values. We have found that instead, spatially perturbing the decay and growth rates from their uniform values using a Gaussian deviate drives the pattern formation harder. That is, there is a greater gradient within the reaction that causes the pattern to form more quickly. The perturbing of the decay and growth rates can be thought of as the natural variation within each cell.

For example, the irregular spatiotemporal spot pattern shown in Figure (1) uses Turing's model with spatially perturbed decay and growth rates, α and β . More specifically, they were set to be $12.0 \pm 0.1\%$ and $16.0 \pm 0.1\%$, respectively. The reaction rate and diffusion rates, s , d_a , and d_b were fixed to be $1/128$, $1/16$, and $1/4$

respectively. Finally morphogens, a and b , were each initially set to 4.00.

For other models, such as the Gray-Scott model, it is necessary to perturb the morphogens in a particular manner, for example, uniformly perturbing the center portion of one morphogen [14].

3.2 Parameter Mapping in Reaction-Diffusion Kinetics

In order to use a reaction-diffusion model for texture synthesis it is necessary to understand how each variable, whether fixed or perturbed, affects the pattern being formed. In addition, a stable pattern will not always form unless the parameters fall within a narrow band of values. Trying to predict whether a pattern will form, along with its corresponding shape, is a difficult process. In Figure (2) we show a phase diagram for the reaction kinetics of the decay and growth rates over a 512 by 512 grid for one morphogen, a , of Turing's model. This diagram is formed by using the same parameters in Figure (1) except that the decay and growth are a function of the cell index, $\alpha = f(i) = [8, 20]$ and $\beta = g(j) = [8, 20]$, respectively.

When decay rate dominates, an unstable oscillating pattern is formed (lower right corner of Figure (2)). When the growth rate dominates, no pattern is formed (upper left corner of Figure (2)). Only when the ratio of decay and growth rates is near 0.75 does a stable pattern of lines and spots form.

Given this phase diagram, one might expect a particular pattern to be formed if the decay and growth rates are set within this stable region. While in general this is true, it is not of great help because only two types of patterns are formed - irregular stripe patterns or symmetrical spots - and as will be shown only the latter can be manipulated. As such, the values of the decay and growth factors do not matter as long they are within the stable region that forms spots.

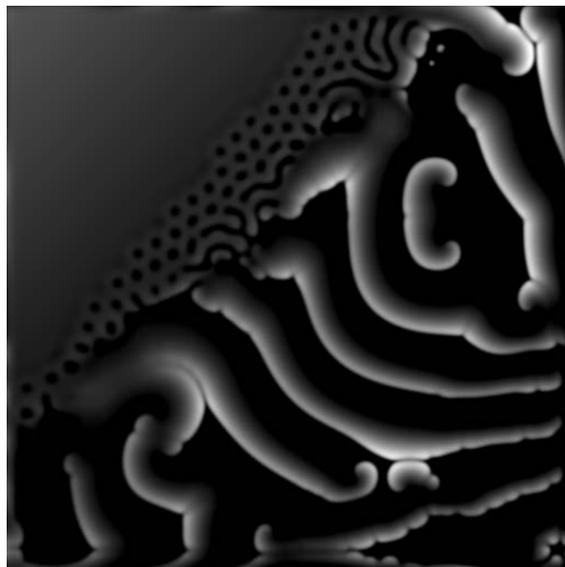


Figure 2: The pattern formed using a variable decay rate, α versus a variable growth rate, β . In each case the range is 8 to 20.

However, it is just not the decay and growth rates that determine the type of pattern formed. The ratio of the diffusion rates also has a dominate role. For example, in Figure (1) the ratio of the diffusion rates, d_a and d_b is 1:4, which forms a spot pattern. If the diffusion rates are changed but maintain a ratio of 1:4 the spot pattern will again be formed with only the size changing. This is fully

demonstrated in the next section. However, if the ratio is changed to 1:16, i.e. d_a and d_b are set to 1/16 and 1, respectively, an irregular spatiotemporal stripe pattern is formed as shown in Figure (3).

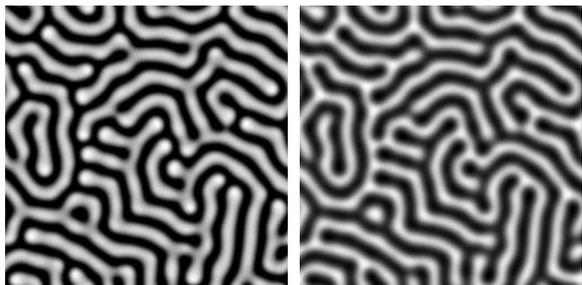


Figure 3: (a) Activator-Depletion Reaction-Diffusion model showing the activator morphogen a and (b) the depletion morphogen b for a stripe pattern using a diffusion ratio of 1:16, $d_a = 1/16$ and $d_b = 1$.

3.3 Reaction Kinetics

The reaction kinetics as written in Equation (1) have three free parameters, the decay and growth rates and the diffusion rate. Typically the decay and growth factors are fixed around a mean value so that a spot pattern will form. But this need not be the case as our phase diagram in Figure (2) shows. Lacalli, et. al. [11] showed that by using a gradient growth or decay rate, they could form a stripe pattern. However, picking values that will produce a pattern beyond uniform stripes that can be controlled based upon a natural parameter can be difficult. In Figure (4) we show one successful example where a pattern that goes from spots to stripes was produced. This was done by setting the decay rate, α to be 12.0 and using a gradient for the growth rate, β between 15.85 and 16.35 along the y axis. The gradient breaks the symmetry, as such the spots are squeezed or distorted resulting in stripes. Such a pattern selection is classically described by coupled amplitude equations in weakly nonlinear theory.

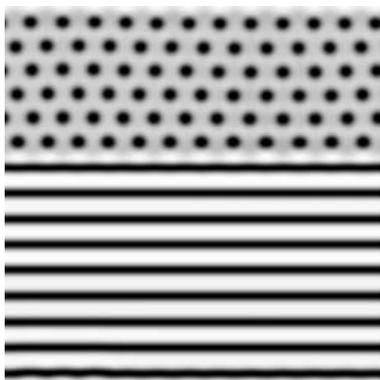


Figure 4: The stripe to spot pattern formed using a constant decay rate, α of 12.0 and a gradient growth rate, β of 15.85 to 16.35.

The other free parameter in the reaction kinetics is the reaction rate, s . Because it appears in both F and G of Equations (3 - 4), respectively it rescales the time and space. As such, it can be used to control the pattern size. For instance, as the reaction rate, s is increased by a quarter and half the original pattern Figure (1a) becomes smaller, Figure (5a) and Figure (5b), respectively. Turk previously noted this and used the surface curvature to vary the spot size [19].

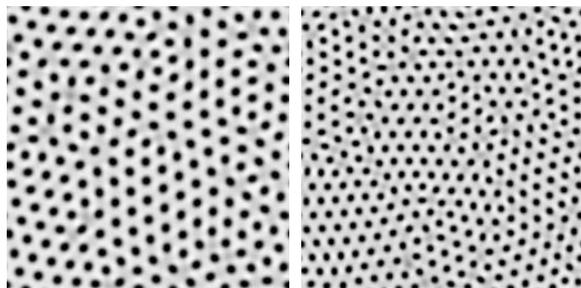


Figure 5: (a) Turing spots with an increased reaction rate of 25% ($s = 1/96$) and (b) 50% ($s = 1/64$).

In a similar fashion it is possible to control the pattern size using the diffusion rates. For instance, if the diffusion rates d_a and d_b of Equations (1 - 2), respectively are increased while the ratio of d_a/d_b remains constant the pattern size will also increase. This shows that it is not the individual rates themselves that control the size, but rather the ratio of the rates. For simplicity and clarity, we vary only the reaction rate for each cell.

3.4 Diffusion Kinetics

3.4.1 Orientation

The diffusion kinetics as written in Equation (1) has just one free parameter, the diffusion rate. This is the case for isotropic diffusion. If we relax this condition and use inhomogeneous, anisotropic diffusion, we are able to create a broader range of patterns. The diffusion equation becomes:

$$\frac{\partial u}{\partial t} = H(u, v) + (\nabla \cdot \sigma_u \nabla) u, \quad (11)$$

where σ_u is a spatially varying symmetric, positive definite diffusion tensor, which we can use to encode the rotation and shaping of a vector field.

Assume that we are working on a finite domain $[a_1, b_1] \times [a_2, b_2] \in \mathcal{R}^2$, and that we are given a regularly-spaced computational grid of size $N_x \times N_y$. At each point (x_i, y_j) , $i = 1, \dots, N_x, j = 1, \dots, N_y$ suppose that we are given an angle $\theta_{ij} \in [0, 2\pi]$. We can now define a rotation matrix and its inverse based upon the angle above:

$$\mathbf{R}_{ij} = \begin{pmatrix} \cos \theta_{ij} & \sin \theta_{ij} \\ -\sin \theta_{ij} & \cos \theta_{ij} \end{pmatrix} \quad (12)$$

$$\mathbf{R}_{ij}^{-1} = \mathbf{R}_{ij}^T = \begin{pmatrix} \cos \theta_{ij} & -\sin \theta_{ij} \\ \sin \theta_{ij} & \cos \theta_{ij} \end{pmatrix}. \quad (13)$$

We now define a principal diffusivity matrix Λ , which is a diagonal matrix and gives the diffusivity coefficients along the two principal axes of diffusion:

$$\Lambda_{ij} = \begin{pmatrix} (\lambda_1)_{ij} & 0 \\ 0 & (\lambda_2)_{ij} \end{pmatrix} \quad (14)$$

where $(\lambda_1)_{ij}$ is the diffusivity in the first principal direction and $(\lambda_2)_{ij}$ is the diffusivity in the second principal direction.

With the definitions above we can define a diffusivity tensor σ_{ij} based on the angle at each grid location as

$$\sigma_{ij} = \mathbf{R}_{ij}^T \Lambda_{ij} \mathbf{R}_{ij}. \quad (15)$$

$$\sigma_{ij} = \begin{pmatrix} (\sigma_{11})_{ij} & (\sigma_{12})_{ij} \\ (\sigma_{21})_{ij} & (\sigma_{22})_{ij} \end{pmatrix}. \quad (16)$$

By manipulating the diffusion matrix coefficients it is possible to create an oriented pattern. We now extend the basic Laplacian by approximating the continuous differential operator as:

$$(\nabla \cdot \sigma(\vec{x}) \nabla) \vec{U} = \frac{\partial}{\partial x} \sigma_{11}(\vec{x}) \frac{\partial \vec{U}}{\partial x} + \frac{\partial}{\partial x} \sigma_{12}(\vec{x}) \frac{\partial \vec{U}}{\partial y} + \frac{\partial}{\partial y} \sigma_{21}(\vec{x}) \frac{\partial \vec{U}}{\partial x} + \frac{\partial}{\partial y} \sigma_{22}(\vec{x}) \frac{\partial \vec{U}}{\partial y} \quad (17)$$

where $\vec{U} = (a(\vec{x}), b(\vec{x}))^T$ is a vector of morphogen concentrations and σ is a diffusivity tensor as defined above. The above equation is then discretized using second order finite differences that takes into account not only the anisotropy but also the inhomogeneity due to the tensor gradients. Witkin and Kass [20] took a similar approach for creating their 2D texture patterns. However, their discretization of the Laplacian was first order and did not take into account the tensor gradients present when using inhomogeneous anisotropic diffusion and as will be demonstrated below, this can affect the quality of the pattern.

In Figure (6) we demonstrate the use of inhomogeneous anisotropic diffusion to create a saddle pattern of spots. This was done by setting $(\lambda_1)_{ij} = 1.5$ and $(\lambda_2)_{ij} = 0.5$ and $\theta_{ij} = \arctan(v, u)$, where $u = x/\sqrt{x^2 + y^2}$ and $v = y/\sqrt{x^2 + y^2}$. To more fully demonstrate how a first order versus a second order approximation to the Laplacian affects the resulting pattern when using inhomogeneous anisotropic diffusion we use the same parameters for $(\lambda_1)_{ij}$ and $(\lambda_2)_{ij}$ as before but set $\theta_{ij} = \arctan(v, -u)$ to create circular spot patterns, Figure (7). The figure on the left uses a first order approximation, while the figure on the right uses a second order approximation, which results in spots that are consistently more uniform irregardless of the orientation of the pattern with respect to the grid. We more fully demonstrate the use of inhomogeneous anisotropic diffusion in [16] and in Section 5.

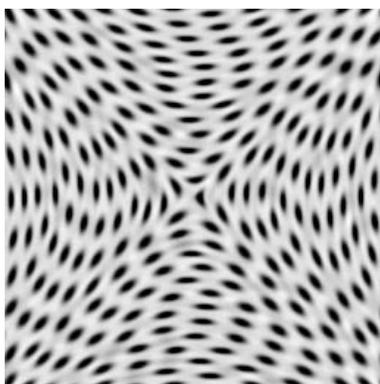


Figure 6: Turing spots using inhomogeneous anisotropic diffusion to create a pattern with a saddle where $(\lambda_1)_{ij} = 1.5$ and $(\lambda_2)_{ij} = 0.5$ and $\theta_{ij} = \arctan(v, u)$.

3.4.2 Shape

We can further control the shape of the pattern formed by moving away from standard linear inhomogeneous anisotropic diffusion to a diffusion tensor that varies as a function of the local gradient. For instance, it is possible to form tear dropped shaped spots by stretching and squishing the pattern based on the dot product of the local morphogen gradient and a vector. As the morphogen concentration changes the gradient and subsequently the shape will also change. This control is accomplished by encoding a change in the principal diffusivity. As before, at each point (x_i, y_j) we are

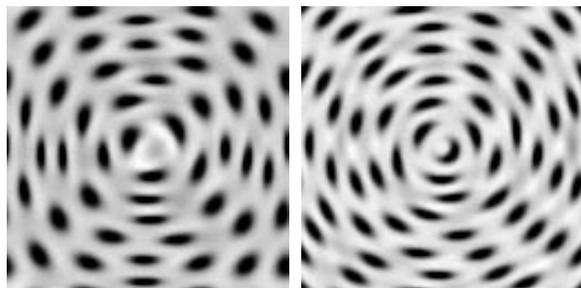


Figure 7: Turing spots using inhomogeneous anisotropic diffusion to create a circular pattern using first order (left) and second order (right) approximation to the Laplacian where $(\lambda_1)_{ij} = 1.5$ and $(\lambda_2)_{ij} = 0.5$ and $\theta_{ij} = \arctan(v, -u)$.

given an angle $\theta_{ij} \in [0, 2\pi]$, which we represent in vector form as: $\vec{v}_{i,j} = (\cos\theta_{ij}, \sin\theta_{ij})$ and ∇u_{ij} . We now define the following operator:

$$d\lambda_{ij}(t) = I \left(\frac{\nabla u_{ij}(t)}{|\nabla u_{ij}(t)|} \cdot \frac{\vec{v}_{ij}}{|\vec{v}_{ij}|} \right) \quad (18)$$

where I is a scaling constant formed by an incomplete gamma function [1]. Inserting this into the principal diffusivity matrix yields:

$$\Lambda_{ij}(t) = \begin{pmatrix} (\lambda_1)_{ij} + d\lambda_{ij}(t) & 0 \\ 0 & (\lambda_2)_{ij} - d\lambda_{ij}(t) \end{pmatrix}. \quad (19)$$

Although not readily apparent due to the signs in equations (18) and (19), the net effect is that if the local gradient is in the opposite direction as the local vector the diffusion is decreased along the principal direction and increased in the secondary direction and visa-versa and shown in Figure (8). In Figure (9) we demonstrate this affect to create a tear drop shaped pattern of spots.

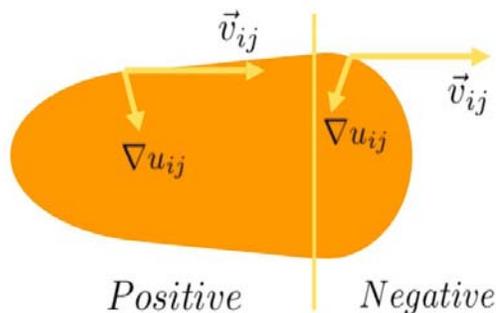


Figure 8: Calculation of the dot product of the local gradient and vector.

Garcke, et. al. [5] took a similar approach but introduced an additional term in their diffusion model rather than encode it directly into the principal diffusivity. By encoding it directly into the principal diffusivity there are fewer terms to deal with, which is important when solving a series of reaction-diffusion equations.

3.5 Mixing Kinetics

In the previous discussion we limited ourselves to a simple two morphogen model. This need not be the case. For example, Turk



Figure 9: Turing tear dropped shaped spots using the dot product of the local gradient and vector to modify the principal diffusivity.

[19] showed how multiple patterns could be formed by using a sequential three step process - form a stable pattern, freeze those cells above a particular concentration level, and then use either the same model with different parameters or use a different model altogether to form a pattern on the unfrozen cells. This technique is able to produce spot on stripe patterns and small spots next to large spots.

Instead of using just one models in a sequential manner, we have explored using two models simultaneously and allowing them to diffuse [21]. Physically, this can be thought of as two layers that are linearly coupled together. Each layer has the same reaction and diffusion kinetics but with different parameters governing their diffusion rates. This system can be describe as:

$$\frac{\partial a_i}{\partial t} = F(a_i, b_i) + \mu(a_j - a_i) + d_{a_i} \nabla^2 a_i \quad (20)$$

$$\frac{\partial b_i}{\partial t} = G(a_i, b_i) + \nu(b_j - b_i) + d_{b_i} \nabla^2 b_i \quad (21)$$

where everything is the same as in Equations (1 and 2) with each layer distinguished by $i, j = 1, 2$ and $i \neq j$, and the diffusion between layers described by the two coupling terms μ and ν . Typically $\mu = \nu$. In Figure (10) we demonstrate this effect using the Brusselator model to create a spot on spot pattern by allowing the diffusion rates in second layer to be approximately three times that in the first layer, that is $d_{a_1} \approx 3d_{a_2}$ and $d_{b_1} \approx 3d_{b_2}$.

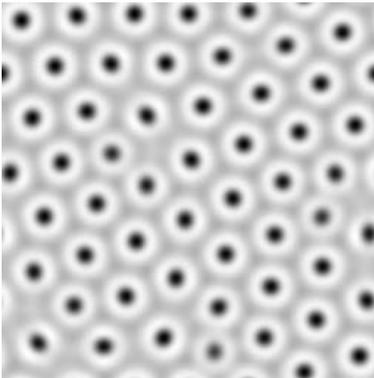


Figure 10: A spot on spot pattern formed using a coupled Brusselator Reaction-Diffusion system with $d_{a_1} = 16.7$, $d_{b_1} = 36.4$ and $d_{a_2} = 49.5$, and $d_{b_2} = 117.6$.

Another option is to extend the model to include a third morphogen, which is then used to couple the reactions together in a three layer system [22]. In this case the top layer is described by:

$$\frac{\partial a}{\partial t} = F(a, b) - \frac{1}{\delta}(a - c) + d_a \nabla^2 a \quad (22)$$

$$\frac{\partial b}{\partial t} = G(a, b) + d_b \nabla^2 b, \quad (23)$$

the middle layer by:

$$\frac{\partial c}{\partial t} = \frac{1}{\delta}(a - c) + \frac{1}{\bar{\delta}}(d - c) + d_c \nabla^2 c, \quad (24)$$

the bottom layer by:

$$\frac{\partial d}{\partial t} = \bar{F}(d, e) - \frac{1}{\delta}(d - c) + d_d \nabla^2 d \quad (25)$$

$$\frac{\partial e}{\partial t} = \bar{G}(d, e) + d_e \nabla^2 e, \quad (26)$$

where the function F and G are from the Oregonator model [3]

$$F(a, b) = \frac{1}{\epsilon}(a - a^2 - fb \frac{a - q}{a + q}) \quad (27)$$

$$G(a, b) = a - b \quad (28)$$

and ϵ, f , and q the kinetics parameters. In addition, F is identical to \bar{F} with the exception of the kinetic parameters $\bar{\epsilon}, \bar{f}$, and \bar{q} .

In this case the morphogen c , which is part of the middle coupling layer, interacts with the top layer containing a and b and the bottom layer containing d and e . This type of reaction-diffusion system is able to produce a stable oscillatory pattern. In Figure (11) we show six frames from an spiral oscillatory pattern.

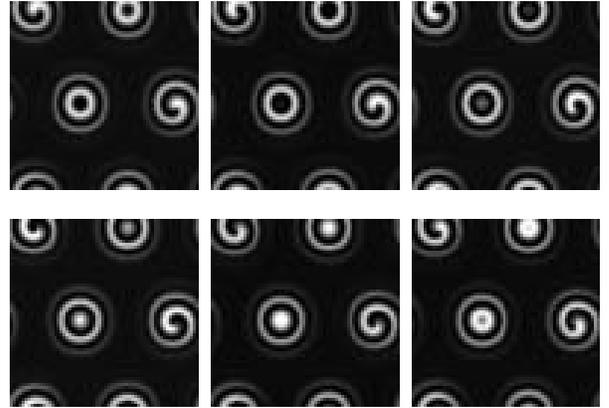


Figure 11: A stable spiral oscillatory pattern formed using a coupled Oregonator Reaction-Diffusion system. Parameters for generating the pattern can be found [22].

4 Solving Reaction-Diffusion Equations

To solve the reaction-diffusion equations we have used second order finite differences in space coupled with either a fully explicit first-order in time forward-Euler integration scheme or a second-order in time semi-implicit Crank-Nicholson scheme that utilizes either Jacobi or Red-Black Gauss-Sidel relaxation for the diffusion terms with forward-Euler for the reactions terms [10].

Because of the sometimes high computational costs associated with solving reaction-diffusion equations, we created a native GPU (GP-CPU) implementation, as well as a CPU implementation for comparison [17].

The GPU implementation is unique in that it is able to do either an explicit or semi-implicit solution. Further, in the case of the semi-implicit solution we have implemented a GPU based convergence test. This allows for all of the operations to take place on the GPU thus the only necessary communication with the CPU is done at start up or when one wishes to view an intermediate result such as when an oscillating pattern is present. To facilitate our implementation we have used the recently developed Framebuffer objects for off screen rendering and Cg [12] for implementing the fragment shaders.

When compared to the CPU implementation, the GPU implementation is approximately 9 and 3 times faster for the explicit and semi-implicit solutions respectively. For example, the 256x256 texture, such as the one shown in Figure (1), which required 50,000 integration steps, can be created in less than 100 seconds on an Intel Xeon P4 running at 3.4Ghz with 2Gb of RAM and 2Gb swap and a nVidia GeForce 6800 Ultra graphics card. Although still computationally expensive these textures only need to be created once and then can be reused. In the case of the oscillating patterns we can get real time frame rates, over 100 fps.

For both integration schemes the largest time step possible that assured stability was used. For the explicit solution, the time step was bounded by the diffusion stability, whereas with the semi-implicit solution, the time step was bounded by the reaction stability. When discretizing the grids we assume a non-dimensional unit spacing which is typical for many reaction models. The one exception is Gray-Scotts' model, which uses a non-dimensional spacing of approximately 0.01. If the diffusion coefficients are adjusted accordingly one can assume unit spacing. We have created patterns using both an explicit and semi-implicit scheme on grids ranging in size from 64 by 64 to 1024 by 1024 with time steps ranging from 0.01 to 12.5 with no qualitative difference in the patterns.

5 Examples

Given the reactions in the previous section we now show some examples of how each part of the kinetics can be used to match those in found in nature. For each of the examples shown we give all the variable values used to produced the figures in the Appendix.

The examples that are perhaps the most difficult are those that use a gradient decay or growth rate with Turing's model. In this case we fix the decay rate and use a gradient growth rate that varies radially from the center, $\beta = f(r) \pm 0.1\%$. At the same we also use a radially increasing reaction rate to increase the pattern size, $s = g(r)$. This results in a circular stripe-spot pattern, Figure (12) that, when colored, is similar to the pattern found around the eye of the White Spotted Puffer Fish, Figure (13) and the Blue Spotted Puffer Fish, Figure (14).

Another example of using a gradient growth factor is shown in Figure (15). In this case the growth factor is a function of the angle around the center, $s = h(\theta)$. This results in a radial stripe-spot pattern that, when colored, is similar to the pattern found around the eye of the Papua Toby Puffer Fish, Figure (16) and the Map Toby Puffer Fish, Figure (17).

The next example is from a coupled Brusselator system where the ratio of the diffusion rates between the two systems are approximately 1:3, that is $d_{a_1} \approx 3d_{a_2}$ and $d_{b_1} \approx 3d_{b_2}$. This creates a double spot pattern where the diameter of the inner spot is 1/3 the diameter of the outer spot, Figure (18), that, when colored, is similar to the pattern found on the body of the Jewel Moray Eel, Figure (19). To obtain the oblonged spots uniform anisotropic diffusion with a principal diffusivity tensor of $(\lambda_1)_{ij} = 1.25$ and $(\lambda_2)_{ij} = 0.75$ was used.

We show another example also using a coupled Brusselator system. However, in this case we show how the mixing parameter can be used to obtain patterns for the sub-adult and adult phases of the

Painted Sweetlips Fish. We first show each of the systems uncoupled. For the first system, we have fixed the morphogen values of the first and middle rows to 0 and 5 whereas the other rows are random with a mean value of 3. Fixing the two rows causes a stripe pattern to form as shown in Figure (20a). The other system also has the same random values but without the two fixed rows and forms a random spot pattern as shown in Figure (20b). To create asymmetrical spots and stripes which give a more natural pattern we have perturbed the reaction constants. When these two systems are coupled and allowed to mix, $m = 0.25$ an aligned spot pattern is formed, Figure (21) that, when colored, is similar to the pattern found on the body of the sub-adult Painted Sweetlips Fish, Figure (22). To obtain the adult phase the mixing rate is set to zero which decouples the system as shown in Figure (20b) and forms a random pattern that is similar to the pattern found on the body of the adult Painted Sweetlips Fish, Figure (23).

The final example shows how the anisotropic diffusion, variable reaction rate, and shape can be used to create a visualization of a vector field [16]. By mapping the vector magnitude to the reaction rate, the vector orientation to the diffusivity, and the vector direction to the shape we can produce a visualization such as shown in Figure (24) of an electrostatic field.

6 Discussion

When creating the patterns we have typically started with one basic part and then built upon the basic pattern. For instance, when creating the spot stripe patterns in Figure (15) we started first with understanding how to create horizontal stripes. Once we could control the size and density we then wrapped them radially. From there, we added the transition from stripes to spots. Finally, we would adjust the variance to give a more natural look. This approach was taken for almost all of the examples shown above.

Perhaps the most interesting patterns are those that oscillate. Although currently we have not found a similar pattern in nature we believe that future work will lead to other oscillating patterns that can be used.

A future project is to relate the reaction variables to some physical property in an automated fashion. For instance, although we were able to duplicate the patterns around the Puffer Fish eyes, setting the necessary gradients was done by trial and error. It would be interesting if the gradients could be tied to the surface curvature or some other physical property. This would allow for a variety of Puffer Fish patterns to be created each with its own unique pattern without having to fully relying on the random variance typically used.

The other area of future work is to choose parameters based on a physiological property. For instance, for the Painted Sweetlips Fish shown in Figures (22-23), we were able to transition between the sub-adult and adult phases solely using the mixing rate. The next step would be to obtain a pattern using a coupled system for the juvenile Sweetlips Fish, which begins its life with stripes that transition into spots, Figure (25). If possible, then the mixing rate could then be a function of age. This would be of interest for animators who often need to age characters.

7 Acknowledgments

This work was supported, in part, by the DOE SciDAC National Fusion Collaboratory and from grants from NSF, NIH, and DOE. The authors would like to especially thank Jeffrey Jeffords/divegallery.com, Massimo Boyer/edge-of-reef.com, Ken Bondy/kenbondy.com, Rudie Kuitert/Aquatic Photographics, and Robert Delfs/tabula-international.com for permission to use their photographs of marine animals. Douglas Segar, Jeffery Johnson,

and John (Jack) Randell helped with identifying several marine animals. The authors would also like to thank Thurston Lacalli for his insightful discussions on gradients and stripe pattern formation.

8 Appendix

One of the difficulties in using a reaction-diffusion model to form patterns is selecting the necessary parameters. As such, in Table(1) we provide the parameters used in to create Figures(1-24).

Our source code is available online at <http://www.acm.org/jgt/papers/SandersonEtAl06> and can be run on Linux, OS X, and other PC platforms. The code requires the following OpenGL libraries: Glew, Glui, and Glut for the CPU and GPU versions and also nVidia's Cg libraries for the GPU version. You will need to supply some code from Numerical Recipes that can not be distributed due to copyright restrictions.

References

- [1] William H. Press and Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 1992.
- [2] I.R. Epstein and J.A. Pojman. *An Introduction to Nonlinear Chemical Dynamics*. Oxford University Press, New York, 1998.
- [3] R.J. Field and R.M. Noyes. Oscillations in chemical systems. *Journal of Chemical Physics*, 60(4):1877–1884, 1974.
- [4] Deborah R. Fowler, Hans Meinhardt, and Przemyslaw Prusinkiewicz. Modeling seashells. In *Computer Graphics (SIGGRAPH)*, pages 379–387, New York, NY, USA, 1992. ACM Press.
- [5] H. Garke, T. Preusser, M. Rumpf, A.C. Telea, U. Weokard, and J.J. van Wijk. A phase field model for continuous clustering on vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):230–242, 2001.
- [6] Alfred Gierer and Hans Meinhardt. A theory of biological pattern formation. *Kybernetik*, 12(1):30–39, 1972.
- [7] P. Gray and S.K. Scott. Sustained oscillations and other exotic patterns of behaviour in isothermal reactions. *Journal of Physical Chemistry*, 89(1):22–32, 1985.
- [8] P. Gray and S.K. Scott. *Chemical Oscillations and Instability: Non-linear Chemical Kinetics*. Oxford University Press, 1990.
- [9] Willen Hundsdorfer and Jan Verwer. *Numerical Solutions of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer-Verlag, Berlin, Germany, 2003.
- [10] G.E. Karniadakis and R. M. Kirby II. *Parallel Scientific Computing in C++ and MPI*. Cambridge University Press, New York, 2003.
- [11] T.C. Lacalli, D.A. Wilkinson, and L.G. Harrison. Theoretical aspects of stripe formation in relation to drosophila segmentation. *Development*, 104(1):105–113, September 1988.
- [12] William R. Mark, R. Steven Glanville, Kurt Akeley, and Mark J. Kilgard. Cg: A system for programming graphics hardware in a c-like language. *ACM Transactions on Graphics*, 22(3):896–907, 2003.
- [13] J.D. Murray. *Mathematical Biology*. Springer-Verlag, New York, 1989.
- [14] J.E. Pearson. Complex patterns in a simple system. *Science*, 261:189–192, 1993.
- [15] I. Prigogine and R. Lever. Symmetry breaking instabilities in dissipative systems. *Journal of Chemical Physics*, 48(4):1695–1700, February 1968.
- [16] Allen R. Sanderson, Chris R. Johnson, and Robert M. Kirby. Display of vector fields using a reaction-diffusion model. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 115–122, Washington, DC, USA, 2004. IEEE Computer Society.
- [17] A.R. Sanderson, M. Meyers, C.R. Johnson, and R.M. Kirby. A framework for exploring advection-reaction-diffusion models. *Computing and Visualization in Science*, page Submitted for publication, 2006.
- [18] A.M. Turing. The chemical basis of morphogenesis. *Phil. Trans. Roy. Soc. Lond.*, B237:37–72, 1952.
- [19] Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. In *Computer Graphics (SIGGRAPH)*, pages 289–298, New York, NY, USA, 1991. ACM Press.
- [20] Andrew Witkin and Michael Kass. Reaction-diffusion textures. In *Computer Graphics (SIGGRAPH)*, pages 299–308, New York, NY, USA, 1991. ACM Press.
- [21] Lingfa Yang, Milos Dolnik, Anatol M. Zhabotinsky, and Irving R. Epstein. Spatial resonances and superposition patterns in a reaction-diffusion model with interacting turing modes. *Physical Review Letters*, 88(20):208303–1–4, May 2002.
- [22] Lingfa Yang and Irving R. Epstein. Oscillatory turing patterns in reaction-diffusion systems with two coupled layers. *Physical Review Letters*, 90(17):178303–1–4, May 2003.

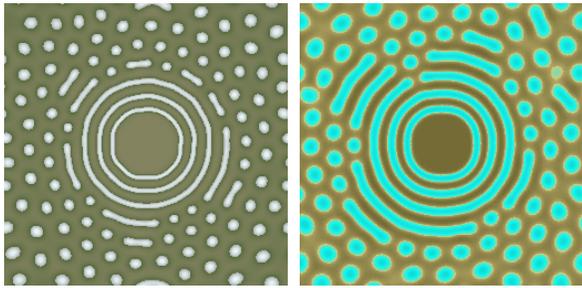


Figure 12: Using radially gradient growth factor, $\beta = f(r) \pm 0.1\%$ and reaction rate, $s = g(r)$ to form a circular stripe-spot pattern.

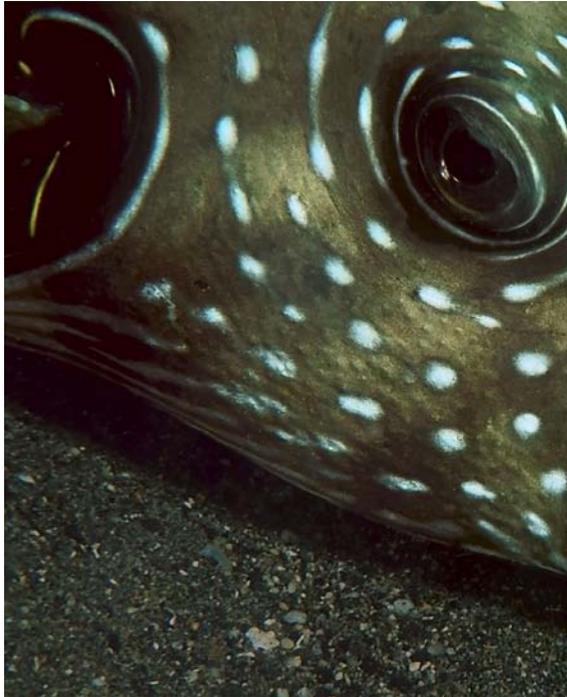


Figure 13: White Spotted Puffer Fish found in the Indo-Pacific region. Image courtesy of Robert Delfs/tabula-international.com.



Figure 14: Blue Spotted Puffer Fish (*Arothron caeruleopunctatus*) found in the Indo-Pacific region. Image courtesy of Jeffrey Jeffords/divegallery.com.

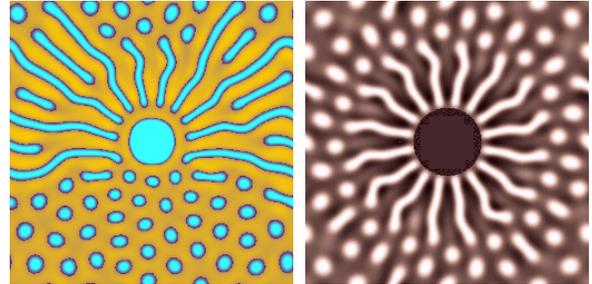


Figure 15: Using radially gradient growth factor, $\beta = h(\theta) \pm 0.1\%$ and reaction rate, $s = g(r)$ to form a radial stripe-spot pattern.



Figure 16: Papua Toby Puffer Fish (*Canthigaster papua*) found in the Western Pacific region. Image courtesy of Massimo Boyer/edge-of-reef.com.



Figure 17: Map Toby Puffer Fish (*Arothron mappa*) found in the Indo-West Pacific region. Image courtesy of Massimo Boyer/edge-of-reef.com.

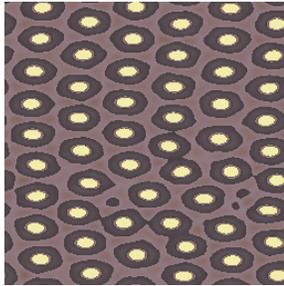


Figure 18: Using a coupled Brusselator system to form a double spot pattern.



Figure 19: Jewel Moray Eel (*Muraena lentiginosa*) found in the Baja-Pacific region. Image courtesy of Ken Bondy/kenbondy.com.



Figure 20: Using a Brusselator system with fixed first and middle rows and random morphogen values to form a stripe (left) and random morphogen values to form a spot pattern (right), respectively.

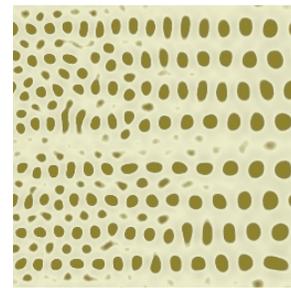


Figure 21: Coupling the Brusselator system from Figure (20) to form an aligned spot pattern.



Figure 22: Sub-Adult Painted Sweetlips Fish (*Diagramma pictum pictum*) found in the Indo-Pacific region. Image courtesy of Rudie Kuiter/Aquatic Photographics.

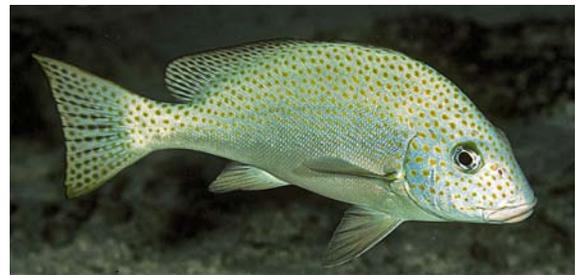


Figure 23: Adult Painted Sweetlips Fish (*Diagramma pictum pictum*) Found in the Indo-Pacific region. Image courtesy of Rudie Kuiter/Aquatic Photographics.

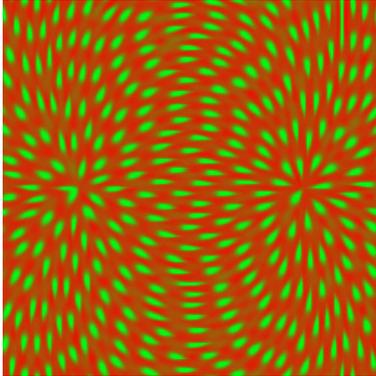


Figure 24: A vector field visualization of an electrostatic field.



Figure 25: Juvenile Painted Sweetlips Fish (*Diagramma pictum pictum*) found in the Indo-Pacific region. Image courtesy of Rudie Kuiter/Aquatic Photographics.

Table 1: Parameters for forming the patterns in each Figure.

Figure	Reaction	$a_{initial}$	$b_{initial}$	α_{ij}	β_{ij}	s	d_a	d_b	Time Step	Mixing
1	Turing	4.0	4.0	$12.0 \pm 0.1\%$	$16.0 \pm 0.1\%$	1/128	1/16	1/4	0.5	NA
2	Turing	4.0	4.0	[8, 20]	[8, 20]	1/128	1/16	1/4	0.5	NA
3	Turing	4.0	4.0	$12.0 \pm 0.1\%$	$16.0 \pm 0.1\%$	1/128	1/16	1	0.5	NA
4	Turing	4.0	4.0	12.0	$15.85 + 0.5j$	1/128	1/16	1/4	0.5	NA
5a	Turing	4.0	4.0	$12.0 \pm 0.1\%$	$16.0 \pm 0.1\%$	1/96	1/16	1/4	0.5	NA
5b	Turing	4.0	4.0	$12.0 \pm 0.1\%$	$16.0 \pm 0.1\%$	1/64	1/16	1/4	0.5	NA
6	Turing	4.0	4.0	$12.0 \pm 0.1\%$	$16.0 \pm 0.1\%$	1/128	1/16	1/4	0.5	NA
7	Turing	4.0	4.0	$12.0 \pm 0.1\%$	$16.0 \pm 0.1\%$	1/128	1/16	1/4	0.5	NA
9	Turing	4.0	4.0	$12.0 \pm 0.1\%$	$16.0 \pm 0.1\%$	1/128	1/16	1/4	0.5	NA
10	Brusselator	$3.0 \pm 1.0\%$	$3.0 \pm 1.0\%$	3.0	9.0	1	16.7/49.4	36.4/117.6	0.0025	0.1
12	Turing	4.0	4.0	12.0	see note a	[0.0028, 0.01]	1/16	1/4	0.5	NA
15	Turing	4.0	4.0	12.0	see note b	[0.0028, 0.01]	1/16	1/4	0.5	NA
18	Brusselator	$3.0 \pm 1.0\%$	$3.0 \pm 1.0\%$	$3.0 \pm 0.25\%$	$9.0 \pm 0.25\%$	1	16.7/49.4	36.4/117.6	0.0025	0.1
20	Brusselator	$3.0 \pm 1.0\%$	$3.0 \pm 1.0\%$	$3.0 \pm 1.0\%$	$9.0 \pm 1.0\%$	[.5, 2]	12.6/47.5	27.5/141.5	0.0015	NA
21	Brusselator	$3.0 \pm 1.0\%$	$3.0 \pm 1.0\%$	$3.0 \pm 1.0\%$	$9.0 \pm 1.0\%$	[.5, 2]	12.6/47.5	27.5/141.5	0.0015	0.25
24	Turing	4.0	4.0	$12.0 \pm 0.1\%$	$16.0 \pm 0.1\%$	1/128	1/16	1/4	0.5	NA

Note a: Values used for β_{ij} in Figure (12)

$$r \in [0, 0.25] : \beta_{ij} = 16 - .5 * r / .25$$

$$r \in [0.25, 0.50] : \beta_{ij} = 16 - .5 * (1 - r / .25)$$

$$r \in [0.50, \text{lim inf}] : \beta_{ij} = 16 \pm 0.1\%$$

Note b: Values used for β_{ij} in Figure (15)

$$r \in [0, 0.125] : \beta_{ij} = 16 - .5 * r / .25$$

$$r \in [0.125, 0.25] : \beta_{ij} = 16 - .5 * (1 - r / .25)$$

$$r \in [0.25, 1.0] : \beta_{ij} = 16 - \sin^2(8\theta)$$

$$R \in [1.0, \text{lim inf}] : \beta_{ij} = 16 \pm 0.1\%$$

Note c: For Figures (6, 7, 9, 18, and 24) inhomogeneous anisotropic diffusion was used.