

# Reaction-Diffusion Patterns

Joshua Bainbridge  
MSc Computer Animation and Visual Effects,  
Bournemouth University

June 12, 2015

## Abstract

This paper describes a biological model called reaction diffusion for the spacial dynamics of chemicals. Reaction diffusion equations were first described by Alan Turing in 1952 and have been used since for organic texture synthesis in computer graphics. Here we will review the various reaction diffusion models that have been developed as well as possible simulation parameters for producing different pattern formations. Then we cover the implementation of the reaction diffusion equations and the results of such models.

## 1 Introduction

Texturing has been used extensively within graphics as an efficient way to add information and detail to computer generated imagery. Despite there being research into techniques for synthesising textures, it has not seen as much development as rendering techniques and representation of three dimensional data. There has been work on procedural techniques for noise generation such as that described in Perlin's seminal paper "An Image Synthesizer" [5] or diffusion-limited aggregation that was originally defined by Witten in "Diffusion-Limited Aggregation, a Kinetic Critical Phenomenon" [12]. These techniques have been developed to produce pseudo random variation and are often used to represent natural and biological patterns. They however produce limited variation in the patterns created and have become recognisable due to repeated use.

Another approach is to develop such patterns using a simulation based upon a biological model. This involves solving semi-linear partial differential equations that define the rate of change between chemical species and the concentrations on a cellular level. The idea was developed by Alan Turing in his paper "The Chemical Basis of Morphogenesis" [8] where he defined the original reaction diffusion systems. Since Turing's paper there have been many different models and variations based upon the original equations.

Another practical issue found with solving these equations is that they requires numerical integration which can be very computationally expensive. This is something that procedural approaches such as perlin noise avoid and as a result give fast iterative feedback when used in production. This has improved however with the development of massively parallel computation and respecting languages as the equations support concurrent implementation. Other advancements in numerical integration of reaction diffusion equations were developed by Hundsdorfer in his book "Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations" [2] which focused on variations of higher order integrators.

## 2 Related Work

Alan Turing originally proposed reaction diffusion systems in [8] where he theorised the existence of two or more molecules known as morphogens. During the embryonic phase these would react and diffuse through cellular structures producing chemical substrates that would later define development such as pigment mutation. Research was then primarily found in fields other than graphics such as chemical dynamics and mathematical biology until the early 1990s where it was then used for texture synthesis. Here we will evaluate some of the different models produced and their advantages in relation to graphics.

The first use of reaction diffusion systems in graphics was by Greg Turk in "Generating Synthetic Textures Using Reaction-Diffusion" [9] where he proposed the basic idea of evaluating the equations over a grid. A year later he released another paper called "Generating Textures on Arbitrary Surfaces Using Reaction-diffusion" [10] where he developed upon this and defined the Turk model for the reaction part of Turing's system. In this paper he describes how the model can be used to generate patterns that resemble animal coatings on geometric surfaces with continuity. Another development he made to achieve the complexity needed was in using one simulation to define the parameters across a surface for another simulation.

Andrew Witkin et al. during the same year also released a paper named "Reaction-Diffusion Textures" [11] that defined another alternative model. Within this model he defines three parts that are diffusion, dissipation and reaction where dissipation is specifically separate from reaction. Anisotropic diffusion is also taken into account to produce a wider range of pattern formation as well as better sampling across non-uniform geometry.

Later John Pearson in "Complex Patterns in a Simple System" [4] applied the Gray-Scott model to Turing's equations by introducing a feed rate into the reaction. Pearson extensively tested and recorded the results from changing two parameters. Those being the aforementioned feed rate into the simulation as well as a kill rate that performs a similar role to Witkin's dissipation. Pearson

managed to produce a wide range of results using the Gray-Scott model and has become a popular choice in current implementations.

More recently Alan Sanderson advances upon many of these papers in "Advanced Reaction-Diffusion Models for Texture Synthesis" [7] and describes a range of models and extensions upon the original concept. In a similar manner to Witkin, he uses the Brusselator model to linearly link two simulations, although these run simultaneously rather than sequentially. Sanderson also looks at how a simulation can be controlled using not only the feed and kill parameters but also varying the relating diffusion rates of both morphogens.

### 3 Morphogenesis

Turing's original work in finding a system that combines a mathematical model to a biological phenomenon has been very influential and is the basis upon which all later models are built. These later models were created while studying spacial dynamics for chemical reactions in a more general field than just that of graphics. The general form describes two morphogen types or chemicals and their reaction within space. One chemical will transform into another while constantly being replenished. The other chemical that is currently being transformed into will slowly dissipate. This happens locally within space and across a time derivative and is the reaction part of the equation. The chemicals are then transformed through space at different rates using diffusion. This is the only non-local part and as result makes adopting space of a higher dimension rather trivial although computationally expensive. This process is then integrated over time producing a derivative that results in what Kondo et al. [3] describes as Turing patterns.

Below in equation one, you can see Turing's original system for producing both the derivative of chemicals  $a$  and  $b$  over time.

$$\begin{aligned}\Delta a &= \frac{\partial a}{\partial t} = F(a, b) + d_a \nabla^2 a \\ \Delta b &= \frac{\partial b}{\partial t} = G(a, b) + d_b \nabla^2 b\end{aligned}\tag{1}$$

Here  $a$  and  $b$  both represent the concentration of a chemical at a specific location in space.  $F$  and  $G$  are functions that dictate change in the chemical's concentration, this represents the reaction and involves both chemicals. Then there is the  $\nabla^2$  that is a laplacian for finding the change of concentration through space. This becomes the diffusion part and  $d$  represents the rate of diffusion for that chemical.

If for example nearby locations were of greater concentration in chemical  $a$ , than that found at the current location, then  $\nabla^2 a$  would be greater than zero. If the nearby locations were of less value than the current location then the laplacian would be less than zero.

Both the parts of the equation can be expanded into the form described in Turing's paper. Due to the computational cost and limited resources available when this was written, the diffusion through space is two dimensional as can be seen below.

$$\begin{aligned}\frac{\partial a_i}{\partial t} &= s(16 - a_i b_i) + d_a(a_{i+1} + a_{i-1} - 2a_i) \\ \frac{\partial b_i}{\partial t} &= s(a_i b_i - b_i - \beta_i) + d_b(b_{i+1} + b_{i-1} - 2b_i)\end{aligned}\tag{2}$$

As before the  $d$  gives the rate of diffusion but the  $s$  also gives the rate of reaction while  $i$  represents an index into the array. This is referred to as the original Turing model.

## 4 Other Models

Another model created by Turk [10] was developed in relation to graphics. In Turk's paper he described how reaction diffusion systems can be used in graphics by simulating across a parametric surface. This requires finding the derivative of the chemical concentration using a two dimensional laplacian. This can be written as

$$\nabla^2 = \frac{\partial^2 a}{\partial x^2} + \frac{\partial^2 a}{\partial y^2}\tag{3}$$

where both  $x$  and  $y$  represent a position in space. The full equation is simply an extension of Turing's model and can be written as

$$\begin{aligned}\frac{\partial a_{ij}}{\partial t} &= s(16 - a_{ij} b_{ij}) + d_a(a_{i+1j} + a_{i-1j} + a_{ij+1} + a_{ij-1} - 4a_{ij}) \\ \frac{\partial b_{ij}}{\partial t} &= s(a_{ij} b_{ij} - b_{ij} - \beta_{ij}) + d_b(b_{i+1j} + b_{i-1j} + b_{ij+1} + b_{ij-1} - 4b_{ij})\end{aligned}\tag{4}$$

where both  $i$  and  $j$  represent indices into a two dimensional array. Turk then repeats the simulation using the previous simulation to perturb current the parameters to produce new patterns.

Witten et al. released another paper on the application the same year as Turk but with a different model. They further generalized the design to include anisotropic and space varying diffusion. They found that this could produce a wider range of patterns. The Witten-Kass model also includes a dissipation part and can be written as

$$C'_{ij} = a^2 \nabla^2 C_{ij} - b C_{ij} + R_{ij}\tag{5}$$

where  $C_{ij}$  is the chemical and  $C'_{ij}$  is its time derivative. The laplacian is represented by  $\nabla^2 C_{ij}$  and  $a$  as its rate of diffusion. The rate of dissipation is  $bC$  where  $b$  is a constant and  $R_{ij}$  represents the reaction part of the equation.

Although still constructed in similar manner to the previous models, this is substantially different and does not require multiple chemicals. Witten also expresses the laplacian as

$$\nabla^2 C_{ij} \approx \frac{C_{i+1j} + C_{i-1j} + C_{ij+1} + C_{ij-1} - 4C_{ij}}{h^2} \quad (6)$$

where  $h$  is the distance between the current and neighbouring samples. Witten wrote that it could be written as a convolution of the chemical array

$$L = \frac{1}{h^2} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (7)$$

where  $L$  is used as a mask and the values in the matrix are the coefficients from equation 6. As for anisotropic diffusion, Witten first constructs a diffusion matrix taking into account the rotation

$$A = \begin{bmatrix} a_1^2 \cos^2 \theta + a_2^2 \sin^2 \theta & (a_2^2 - a_1^2) \cos \theta \sin \theta \\ (a_2^2 - a_1^2) \cos \theta \sin \theta & a_2^2 \cos^2 \theta + a_1^2 \sin^2 \theta \end{bmatrix} \quad (8)$$

where  $a_1$  is the diffusion in the  $[\cos \theta, \sin \theta]$  direction and  $a_2$  is the diffusion in the  $[-\sin \theta, \cos \theta]$  direction. They then use this to construct an anisotropic mask in the form

$$M = \begin{bmatrix} -a_{12} & 2a_{22} & a_{12} \\ 2a_{11} & -4(a_{11} + a_{22}) - 2h^2b & 2a_{11} \\ a_{12} & 2a_{22} & -a_{12} \end{bmatrix} \quad (9)$$

where  $a$  is the element within the diffusion matrix. They also introduce another two dimensional array called a diffusion map. This array allows the diffusion matrix to vary with position by representing it's three coefficients at each finite position in space. This can also be much coarser than the chemical array and be used with bilinear interpolation.

Pearson [4] defines another variation based upon the Gray-Scott [1] model. This model describes the reactions between three chemicals and is originally written in the form



where both operations are irreversible, ultimately resulting in  $P$  as a product. Pearson then adds a feed term to  $U$  while  $U$  and  $V$  are both removed during the process, expressing the model in terms of a reaction diffusion system. This can be written as

$$\begin{aligned} \frac{\partial U}{\partial t} &= d_U \nabla^2 U - UV^2 + f(1 - U) \\ \frac{\partial V}{\partial t} &= d_V \nabla^2 V + UV^2 - (f + k)V \end{aligned} \quad (11)$$

where  $U$  and  $V$  are both the reacting chemicals. The feed and kill coefficients are represented by  $f$  and  $k$  respectively. This model has the effect of being activator-substrate meaning that as one chemical becomes more concentrated the other becomes weaker resulting in an inverse pattern formation. This will be demonstrated during a later part of this paper.

In a more recent paper [7], Sanderson describes a reaction diffusion system that is linearly dependent on another and visa-versa. This coupling approach is similar to that described by Turk [10] but runs concurrently rather than sequentially. It can be written in the form of

$$\begin{aligned}\frac{\partial a_i}{\partial t} &= F(a_i, b_i) + \mu(a_j - a_i) + d_{a_i} \nabla^2 a_i \\ \frac{\partial b_i}{\partial t} &= G(a_i, b_i) + \nu(b_j - b_i) + d_{b_i} \nabla^2 b_i\end{aligned}\tag{12}$$

and also another identical system with the  $i$  and  $j$  indices reversed. Both  $\mu$  and  $\nu$  are the coupling terms and are typically  $\mu = \nu$ . Sanderson also implemented the Brusselator model [6] for the reaction part of the equation, which can be written in the form

$$\begin{aligned}F(a, b) &= s(\alpha - (1 + \beta)a + a^2b) \\ G(a, b) &= s(\beta a - a^2b)\end{aligned}\tag{13}$$

where  $\alpha$  and  $\beta$  are both the kill and feed rates respectively while  $s$ , as before is the rate of reaction.

## 5 Implementation

For implementation I first developed a C++ graphics environment to allow instant feedback of the simulation within a frame buffer. It also allowed me to control the time of each iteration during the integration very accurately and allow synchronisation of computation and visualisation. The submitted code is currently written so that the simulation will run at 60 frames a second and compensate for delta time steps. As reaction diffusion systems are naturally highly concurrent I decided to implement the code on a GPU for massively parallel computation. I started by creating a working simulation on the CPU that was highly vectorized and this made it simple to transfer the code to both OpenCL and CUDA compute languages for the GPU. As for submission you will find the implementation for OpenCL as the runtime compilation of kernels make it simpler to compile. For further information on GPU implantation there is a paper [7] written by Sanderson following his previous research on reaction diffusion systems.

I used Pearson's adaptation of the Gray-Scott model [4] as it produced a wide range of results and was relatively stable. It was then simulated using an implicit Euler integration. To improve stability I altered Witten's original convolution mask to the form:

$$M = \frac{1}{h^2} \begin{bmatrix} 0.05 & 0.2 & 0.05 \\ 0.2 & -1 & 0.2 \\ 0.05 & 0.2 & 0.05 \end{bmatrix} \quad (14)$$

This allowed for a larger delta time and faster computation. Using data gathered by Pearson [4] I was able to simulate many different patterns and formations. Below are just some examples of the results that were obtained. I also experi-

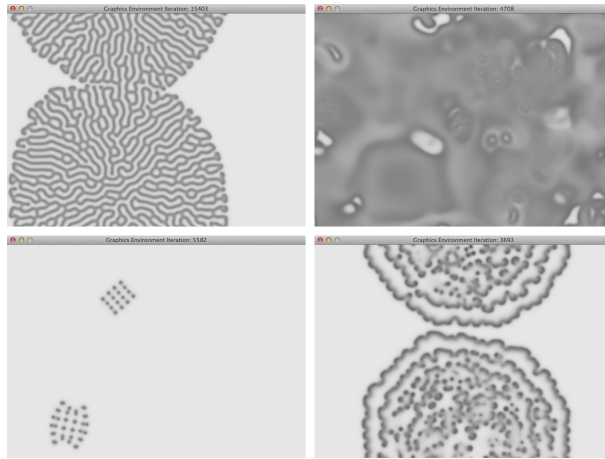


Figure 1: Example Simulations

mented with diffusion parameters for each chemical and found the same results as those recorded by Sanders [7]. Changing the diffusion relative between the chemicals has no effect other than altering the scale of the simulation. This



Figure 2: Relative Diffusion

can be seen in the left image of Figure 2 as opposed to the bottom left image of Figure 1. Both these simulations have a diffusion ratio of  $\frac{1}{2}$  in contrast the ratio of  $\frac{1}{16}$  seen in the right image of Figure 2. Finally I also visualised both the

chemical concentrations to find an inverse pattern is produced, validating the activator-substrate behaviour of the Gray-Scott model. The parameters for such patterns can be found in the `paramters.txt` file within the source root.

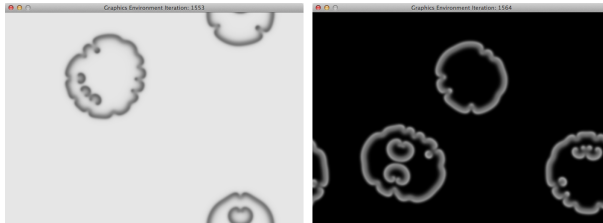


Figure 3: Activator-Substrate Behaviour

## 6 Conclusion

As demands for more organic and innovative approaches to texture synthesis develop and computational power increase, reaction diffusion equations could become more widely used and researched. They can also be extended into higher dimensional space relatively easily. Current supporting research in fields such as chemical dynamics and mathematical biology have already developed a range models that could be further experimented within graphics with minimal work. Another area of potential research would be in interactivity, allowing artists control over parameters to develop textures in more organic ways.

## References

- [1] P. Gray and S.K. Scott. Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Isolats and other forms of multistability. *Chemical Engineering Science*, 38(1):29–43, 1983.
- [2] W. Hundsdorfer and J.G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer Series in Computational Mathematics. Springer, 2003.
- [3] Shigeru Kondo and Takashi Miura. Reaction-diffusion model as a framework for understanding biological pattern formation. *Science*, 329(5999):1616–1620, 2010.
- [4] J. E. Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, 1993.
- [5] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, July 1985.



- [6] I. Prigogine and R. Lefever. Symmetry breaking instabilities in dissipative systems. *J. Chem. Phys.*, 48:1695–1700, 1968.
- [7] Allen R. Sanderson, Robert M. Kirby, Chris R. Johnson, and Lingfa Yang. Advanced reaction-diffusion models for texture synthesis. *J. Graphics Tools*, 11(3):47–71, 2006.
- [8] Alan Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society B*, 237:37–72, 1952.
- [9] Greg Turk. Generating synthetic textures using reaction-diffusion. Technical report, Chapel Hill, NC, USA, 1990.
- [10] Greg Turk. Generating textures on arbitrary surfaces using reaction-diffusion. *SIGGRAPH Comput. Graph.*, 25(4):289–298, July 1991.
- [11] Andrew Witkin and Michael Kass. Reaction-diffusion textures. *SIGGRAPH Comput. Graph.*, 25(4):299–308, July 1991.
- [12] T. A. Witten and L. M. Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. *Phys. Rev. Lett.*, 47:1400–1403, Nov 1981.