Reaction-Diffusion Patterns

Romy Williamson

July 2025

1 Introduction

Let's start at the very beginning. The Diffusion equation, also known as the heat equation, is the partial differential equation

$$\frac{\partial u}{\partial t} = \alpha \Delta u, \tag{1}$$

that is, the rate of increase of the concentration of something (e.g. heat, or a chemical that is spreading out) is proportional to the Laplacian (a type of multi-variate second-derivative in space) of the current concentration. The parameter $\alpha > 0$ describes the diffusivity of the substance.

A Reaction-Diffusion system is a system whose state is governed by a semi-linear parabolic partial differential equation of the form

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{D}\Delta \mathbf{q} + \mathbf{R}(\mathbf{q}) \tag{2}$$

where $\mathbf{D}$ is a square diagonal matrix of the same dimension as the vector $\mathbf{q}$ [6].

Notice that, without the $\mathbf{R}$ term, it looks almost the same as the usual Diffusion equation. Without the $\mathbf{R}$ term, this PDE could be describing the concentrations of two (or more) chemicals that are spreading out. (The diagonal elements of $\mathbf{D}$ are the 'diffusion coefficients'.) The $\mathbf{R}$ term, however, encodes the effect of interactions — *reactions* — between the chemicals.

It turns out that these kind of systems occur all the time in nature. The classic example is the situation of two different pigmented chemicals spreading out and interacting over the skin of a developing animal. Knowing that animals come with all different types of patterns (spots, stripes, wiggly patterns etc. − see Figure 1), you should be realising by now that these Reaction-Diffusion equations govern some fairly intriguing dynamics.

A quick note on history: Alan Turing first studied these PDEs in the early 1950s, (shortly after the end of the Second World War, when he was finished decoding the Enigma cipher used by the Nazis). In his 1952 paper "The Chemical Basis of Morphogenesis" [5] he showed that this type of equation can lead to structure emerging from randomness. This was a problem that biologists were stuck on; specifically, they did not understand the how the earliest stages of embryogenesis are possible − how can a blob of cells (a *spherical blastula*, which is symmetrical and structureless) break its symmetry and form a *gastrula*? (See Figure 2.) A subclass of patterns produced by these Reaction-Diffusion equations are named Turing Patterns, in his honour [1].

2 Implementation

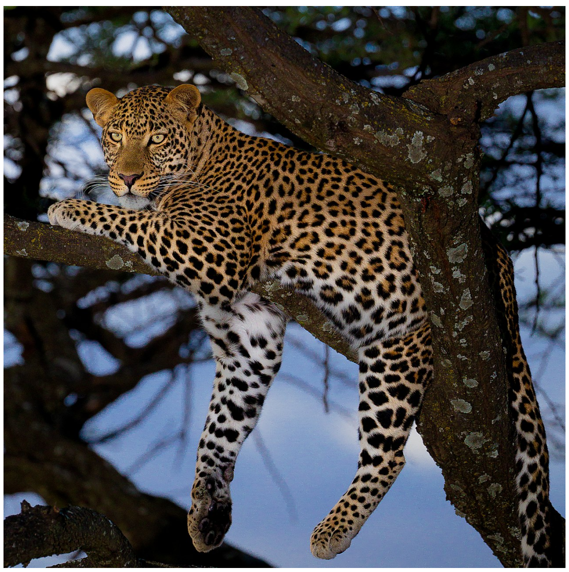
The precise form of the PDE that is used in my implementation is the Gray-Scott model. The state-vector $\mathbf{q}(\mathbf{x}, t)$ is just $(u(\mathbf{x}, t), v(\mathbf{x}, t))$ where u is the concentration of chemical A and v is the concentration of chemical B, at the position $\mathbf{x}$ and time t.

Then

$$\begin{aligned} \tfrac{\partial u}{\partial t} &= D^u \Delta u - uv^2 + F(1-u) \\ \tfrac{\partial v}{\partial t} &= D^v \Delta v + uv^2 - (F+k)v\ . \end{aligned} \tag{3}$$

(a) Spotty Frog

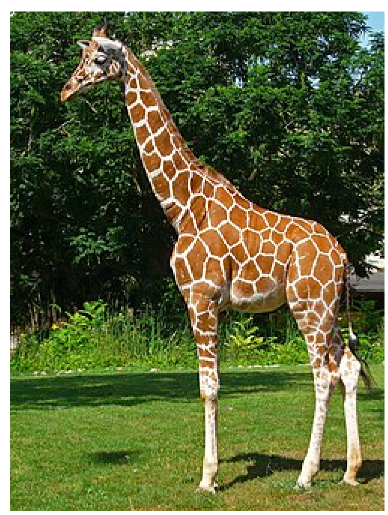
(b) Wiggly Frog



(c) Leopard (with wiggly spots)

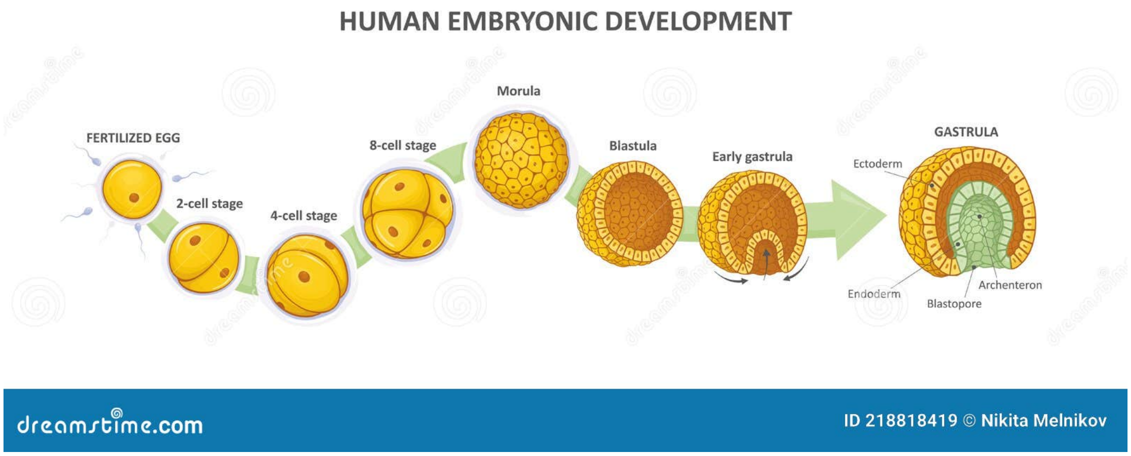
(d) Stripy Zebra with a Spotty Foal

(e) Wiggly Brain Coral



(f) Giraffe (cracked texture)

Figure 1: Various Animal Patterns



(a)

Figure 2: Embryogenesis

Intuitively, D^u and D^v are the diffusivities of the two chemicals (which may be different), F is the 'feed'-rate of chemical A, and k is the 'kill'-rate of chemical B. For a more thorough explanation, see [3]. All of these parameters are considered to be properties of the chemicals, so they are generally kept fixed for the duration of a simulation. I used another simulation tool [2] to help me to select parameter values that make interesting patterns. (Thank you Karl Sims – great tool.) Different parameters lead to different patterns. Just an interesting aside to think about: given our knowledge about the behaviour of different parameter values do we have enough information to work out exactly what went wrong in the zebra foal's development (Figure 1 (d)) to make it spotty and not stripy?

Simulations that work on flat domains (such as the tool I just mentioned) are implemented by evolving the PDE using a finite-difference-based Laplacian on the grid of pixels, and evolving the solution of the PDE using discrete timesteps (usually, forward-Euler).

In my simulation, I wanted to evolve the solution of the PDE on a curved-surface domain (the skin of the frog, or other animals). The animal surfaces are stored as triangle meshes (.obj files). So, any scalar field on the surface can be discretised by sampling its value at every vertex of the mesh. (The mesh vertices take on a similar role as pixels in a 2D image.) We just need to replace the usual grid-based finite-difference Laplace operator matrix by a discrete Laplace-Beltrami operator, called the cotan Laplace-Beltrami matrix [4], which is computed based on the areas and angles of the triangles in the mesh.

The vital part of the code is extremely simple – we just compute the matrix for the Laplace-Beltrami operator, and then iteratively update the chemical concentrations using forward-Euler (this is the most naïve way to discretise a PDE, and can be unstable, but it's accurate enough for our purposes!). The crucial piece of code is shown in Figure 3.

To be a little bit more "creative", I used various initial conditions for (u, v), including bitmap masks of text. I also played with putting the simulation in reverse and 'chaining' simulations, by using the converged result of a simulation with one set of parameters as the initial conditions for a simulation with another set of parameters. This is what you see in the video "fancy-backwards".

3 Reflection

I am aware that some may read this and argue, "but that's not art at all, you are just modelling natural phenomena". I partially agree; it hardly takes a creative genius to implement a numerical solution to a PDE. This addresses a core question of philosophical interest. What is art, and does it always have to be creative? Personally I would favour a more liberal definition (and not just because this is for an art competition).

For thousands of years, humans have looked into nature and seen beauty. In the full moon on a cloudless night, in the plumage of an exotic bird, in the murmurations of the birds in the sky, or in the ripples of the

```python
# Cotangent Laplacian
L = igl.cotmatrix(vertices, faces)

for i in range(steps):
    Lu = L @ U
    Lv = L @ V

    reaction = U * V * V
    U += (Du * Lu - reaction + F * (1 - U)) * dt
    V += (Dv * Lv + reaction - (F + k) * V) * dt
```

Figure 3: The Crucial bit of Code

sand on a deserted beach, in the hypnotic recursion of a branching tree. There is a fallacy often fallen into by creationists, that complex and beautiful things appear to have been designed, therefore they must have been designed by someone, and that someone is the Intelligent Creator.

To others, this just proves that design is an illusion. Physics and maths have already determined all of the structures that can possibly exist in the universe. Maybe as humans, we are merely here to be curators and art-appreciators of the most interesting natural structures.

I think the question of "Does art need to be creative?" is in essence the same as the question of "Is mathematics invented or discovered?". And my best answer to this is that doing art and doing maths are both entirely about uncovering interesting structure. So depending on our mood we could either say that invention and creation are complete illusions. Or more appeallingly, we can redefine 'invention' as a type of 'discovery plus observation and vision'. Now is an appropriate time for a quote:

> *"The sculpture is already complete within the marble block, before I start my work. It is already there, I just have to chisel away the superfluous material."*
>
> – Michelangelo

4 References

References

[1] Wikipedia Contributors. Turing patterns. `https://en.wikipedia.org/wiki/Turing_pattern`, 2025. [Online; accessed July 13, 2025].

[2] Karl Sims. Rd tool. `https://www.karlsims.com/rdtool.html`. [Online; accessed July 13, 2025].

[3] Katharina Käfer and Mirjam Schulz. Gray-Scott model. `https://itp.uni-frankfurt.de/ gros/StudentProjects/Pro`

[4] Keenan Crane. The Laplace Beltrami operator. `https://brickisland.net/DDGSpring2021/wp-content/uploads/202`

[5] Alan M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72, August 1952. doi: 10.1098/rstb.1952.0012.

[6] Wikipedia contributors. Reaction-diffusion systems. https://en.wikipedia.org/wiki/Reaction-diffusion_system, 2025. [Online; accessed July 13, 2025].