

# Real-Time Visualization of 3D Reaction-Diffusion Systems

Jakob Škornik

University of Ljubljana

Faculty of Computer and Information Science

Slovenia

js6675@student.uni-lj.si

---

## Abstract

*This project originally aimed to develop three-dimensional models of complex structures, such as coral reefs, using reaction-diffusion systems. The objective evolved to include the real-time visualization of these systems using Graphics Shader Language (GLSL) in a web-based application. Leveraging WebGL2, we successfully extended these models from two to three dimensions, utilizing parallel computing capabilities of modern GPUs. We accomplished not only the initial goal, but also developed three different models, namely Gray-Scott, FitzHugh-Nagumo, and Schnakenberg.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: GLSL Shaders & WebGL—3D Reaction-Diffusion

---

## 1. Introduction

Reaction-diffusion systems, a subset of partial differential equations, have been a core element of study in mathematical biology since their beginning in the early 20th century. Reaction-diffusion equations have been proven to be incredibly useful for modelling various natural phenomena such as the behaviour of chemicals during a chemical reaction, the behavior of populations, and the formation of patterns in nature.

The Gray-Scott model, for example, simulates two chemical substances reacting and diffusing. This model has been used for studying pattern formation in chemical reactions. It is best known for its simplicity and is often used as an introduction to reaction-diffusion models.

The FitzHugh-Nagumo model is another significant model that simplifies the Hodgkin-Huxley model of spike generation in neurons. Lastly, the Schnakenberg model is a simple yet robust example of a Turing system, demonstrating conditions for diffusion-driven instability. We will introduce the mathematics behind these systems and solutions to the equations. We will also explain the methods used in solving such systems.

## 2. Goal

While the original goal was to use reaction-diffusion models to create 3D models of organic structures such as coral reefs, our focus shifted towards the real-time and interactive visualization of these models using GLSL shaders.

The goal was adapted to focus on real-time interactive visualization of these models using GLSL shaders and to provide an interface for interacting with the simulation in run-time.

## 3. Implementation

### 3.1. Project

The project is using the following technologies:

- **JavaScript:** The whole project is contained in a JavaScript module embedded in an HTML canvas.
- **glMatrix:** This library provides fast matrix operations and rendering tools.
- **lil-gui:** This library provides a simple but heavily customizable GUI widget.
- **stats.js:** This library provides a simple performance metric widget.

### 3.2. Implementation Process

We initially developed a framework using the Three.js library, a wrapper around WebGL that provides an API for graphical development in JavaScript. However, this API doesn't encompass all capabilities of WebGL. A significant limitation was that 3D textures couldn't be altered after they were created. Consequently, in each iteration, we were required to generate a new texture with updated information.

Creating objects is a resource-intensive process, and the initial implementation struggled to maintain 30 frames per second, even with small textures. We began exploring alternatives and found that using shaders to compute resource-heavy operations with a GPU presented a more viable solution.

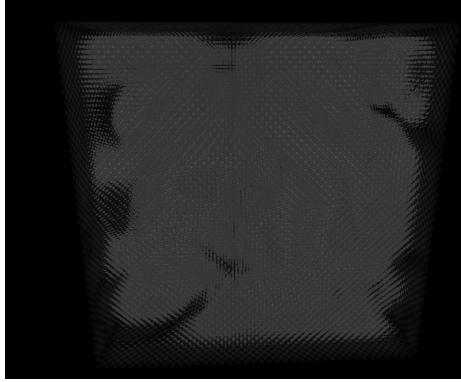


Figure 1: Gray-Scott implementation using Three.js.

Consequently, we discarded the initial implementation and constructed a new framework for rendering shaders in an HTML canvas. Once we had this environment established, we simply reinterpreted the original implementation in GLSL, translating it into fragment shaders. The original version utilized the Runge-Kutta method for approximating the Laplacian, but in the new iteration, we employed the finite differences method.

This also fixed the flickering issue that was present in the original implementation.

## 4. Methodology

Most of these models are originally described in two dimensions, but their expansion to three dimensions is somewhat trivial, namely one must simply change the Laplacian operator with its 3D counterpart. A brief explanation of Laplacian in two and three dimensions is provided at the end of this chapter.

### 4.1. Gray-Scott Model

The Gray-Scott [Pea93] model is a mathematical model describing the behavior of autocatalytic chemical reactions. It

uses two variables,  $u$  and  $v$ , which represent concentrations of two substances. The reaction equations are given by:

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + f(1 - u),$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (f + k)v.$$

Where  $D_u$  and  $D_v$  are the diffusion rates of  $u$  and  $v$ , respectively.  $\Delta$  is the Laplacian operator.  $f$  is the feed rate, and  $k$  is the decay rate.

### 4.2. FitzHugh-Nagumo Model

The FitzHugh-Nagumo [Fit61] model is a simplified version of the Hodgkin-Huxley model that describes the propagation of electrical signals in neurons. The model is described by the following set of equations:

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u + u - u^3 - v,$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + \varepsilon(u - av - b).$$

Here,  $u$  and  $v$  are variables representing the membrane potential and a recovery variable, respectively, and  $\varepsilon$  is an externally applied current. The constants  $a$ ,  $b$ , and  $\tau$  represent specific properties of the neuron.

### 4.3. Schnakenberg

The Schnakenberg [IWW04] model is another reaction-diffusion system that can produce Turing patterns. It is given by the following set of equations:

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u + a - (u + v^2)u,$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + b - (u + v^2)v.$$

Here,  $u$  and  $v$  are the concentrations of two chemical substances, and  $D_u$  and  $D_v$  are their respective diffusion coefficients. The constants  $a$  and  $b$  are parameters of the model.

### 4.4. Finite Differences Method

The Laplace operator or Laplacian is a differential operator given by divergence of gradient on Euclidian space.

The Finite Differences Method is a numerical technique often used for approximating solutions to differential equations. It operates by replacing the derivatives in the differential equations with finite differences approximations.

In our project we used 3D version of Laplacian to model reaction-diffusion systems.

#### 4.5. Extensions

We added two external forces, which act similarly but are in fact different. Through the user interface we can control not only model parameters but also these two external forces.

Wind is the first external force. It adds a constant force in a set direction. Gravity is another force that adds gravitational pull towards a certain point, but only when a specific button is held.

Using external forces changes reaction behaviour in various ways and helps in forming unique patterns.

#### 5. Results

Our implementation is deployed to Vercel and can be accessed [here](#).

We present examples of formations obtained by all three models. The effect of external forces is hard to visualize with images, but is clearly visible in the reaction in real time.

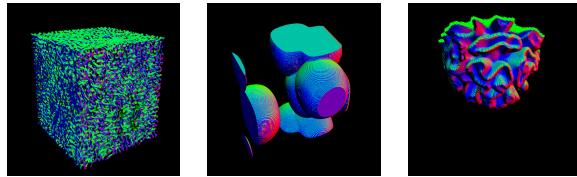


Figure 2: Examples of Gray-Scott models.

Our newly implemented framework computing on the GPU through WebGL shaders yielded great results. We have noted a considerable improvement in performance, by over five times compared to the previous CPU-based implementation.

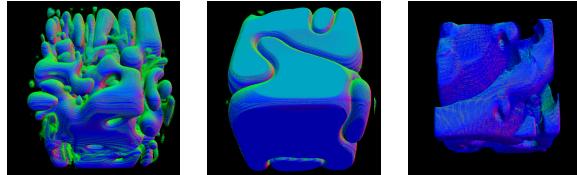


Figure 3: Examples of FitzHugh-Nagumo models.

Additionally, visual anomalies that were present in our earlier models, likely due to an incorrect implementation of the Runge-Kutta method for approximating Laplacian, are now completely resolved.

Overall, our efforts in redesigning the framework and optimizing the implementation of the reaction-diffusion models have successfully led to a smoother, more accurate and much more efficient simulation. The power of parallel GPU

computation has allowed us to overcome the initial problems and achieve our goals.

Additionally we expanded the base Gray-Scott model with two external forces and then used this model as a basis for two additional models; FitzHugh-Nagumo, and Schnakenberg models.

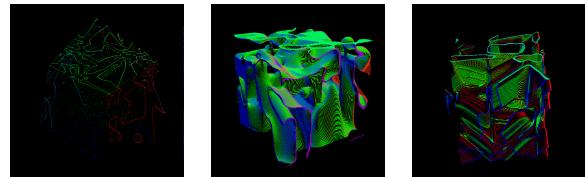


Figure 4: Examples of Schnakenberg models.

UI includes two groups of parameters; base and model-specific parameters. Users can control simulation speed, number of source reactant globs and wrap mode. Wrap mode makes the bounding box wrap around the edges, simulating an infinite space. Once a reaction reaches the end of the bounding box, reaction simply continues on the opposite face.



Figure 5: Example of user interface with Gray-Scott model.

#### 6. Discussion

The transition from a Three.js based framework to a shader-based framework resulted in significant improvements in efficiency and accuracy. Implementing GLSL and employing finite differences methodology, we overcame the initial restrictions posed by Three.js. Furthermore, the visualization errors that were present in our original implementation have now been fixed.

## References

- [Fit61] FITZHUGH R.: Impulses and Physiological States in Theoretical Models of Nerve Membrane. *Biophysical Journal* 1, 6 (July 1961), 445–466. [doi:10.1016/S0006-3495\(61\)86902-6](https://doi.org/10.1016/S0006-3495(61)86902-6). 2
- [IWW04] IRON D., WEI J., WINTER M.: Stability analysis of turing patterns generated by the schnakenberg model. *Journal of mathematical biology* 49 (11 2004), 358–90. [doi:10.1007/s00285-003-0258-y](https://doi.org/10.1007/s00285-003-0258-y). 2
- [Pea93] PEARSON J. E.: Complex patterns in a simple system. *Science* 261, 5118 (1993), 189–192. URL: <https://www.science.org/doi/abs/10.1126/science.261.5118.189>, arXiv:<https://www.science.org/doi/pdf/10.1126/science.261.5118.189>, doi:[10.1126/science.261.5118.189](https://doi.org/10.1126/science.261.5118.189). 2