

# Spatial Patterns in a Non-Linear Diffusion Equation

May 19, 2023

## Abstract

Turing patterns arise in reaction-diffusion systems when small perturbations around a steady state result in the development of spatial structures. This studies the emergence of Turing patterns in reaction-diffusion systems with non-constant diffusion coefficients, focusing on the dependence of diffusion coefficients on the species' concentration. Linear stability analysis is used to understand the conditions under which Turing patterns can form in the presence of both reaction and diffusion. We examine the “Brusselator” model as a case study for autocatalytic reaction-diffusion systems. Our central questions are: Can Turing patterns arise in systems with non-constant diffusion coefficients? If so, what are the conditions for their emergence? By answering these questions, we aim to understand the necessary conditions for Turing patterns to arise in reaction-diffusion systems.

## 1 Introduction

In 1952, Alan Turing proposed a mathematical model of chemical reaction that led to formation of stationary patterns in space under suitable conditions. His motivation for studying this was that he claimed it to be the origin of morphogenesis[3]. First evidence of such Turing patterns came from a Russian biochemist in early 1950s, Boris Belousov, who could not get his paper published as his results looked like it violated the Second Law of Thermodynamics that the state cannot evolve to a spatially uniform state [1]. Hence, this phenomenon is also called non-equilibrium pattern formation. Finally, in 1989 the experimental evidence came from Kepper and his team who observed stationary Turing patterns in a chemical reaction of chlorite ions, iodide ions, and malonic acid (CIMA) reaction[1]. It was also observed in other chemical and biological processes[2].

To understand Turing's prediction consider a system of two species concentration  $U$  and  $V$  in the system which have  $D_U$  and  $D_V$  as their respective diffusion constants and the concentration of  $U$  and  $V$  that depend on  $(\vec{x}, t)$  and change due to reaction given by  $f(U, V)$  and  $g(U, V)$  respectively which

in general are non-linear and its own diffusion given by its constants. The reaction-diffusion model can be described by the following set of equations.

$$\begin{aligned} U_t &= D_U \nabla^2 U + f(U, V), \\ V_t &= D_V \nabla^2 V + g(U, V) \end{aligned} \quad (1)$$

$U_t$  and  $V_t$  are partial derivatives of  $U$  and  $V$  with respect to time, and  $\nabla^2$  is a Laplacian operator relating to the space variable  $\vec{x}$ . It can be shown with linear stability analysis of the model that certain conditions need to be fulfilled for Eqn.1 for Turing instability to occur. For example, it will be shown later that the constants  $D_U \neq D_V$  for this to happen, although this is only a necessary condition and not a sufficient one.

In the equation above, the diffusion is assumed to be a constant, but in many interesting physics problem non-linear diffusion takes place like transfer process in porous media, filtering diffusion [4]. In particular, the general porous media equation can be re-written as  $u_t = \nabla(D(u)\nabla u) + f(u)$ (shown later), where  $D(u)$  is diffusion constant function that depends on  $u$  and can be chosen to be non-linear. This paper will be answering the question: *Can Turing pattern be observed as we increase the non-linear parameter  $m$  is of the diffusion coefficient that is dependent on the concentration?* The report seeks to generalize the linear stability test for conditions when  $D(u)$  is a function of different orders of  $u$ . Subsequently, we describe the implementation of numerical simulations to investigate the impact of different diffusion coefficient properties on Turing pattern formation.

## 2 Background and Theory

### 2.1 The General Porous Medium Equation

The General Porous Medium Equation(PME) is given by a non-linear partial differential equation(PDE) of form:

$$\frac{\partial u}{\partial t} = \nabla^2(u^m) + f(u) \quad (2)$$

Here  $u(x, t)$  is a scalar function usually representing concentration in d-dimensional euclidean space where the equation is posed for all  $x \in \mathbb{R}^d$  and  $t \in \mathbb{R}$  for  $t > 0$ .  $\nabla^2$  is a Laplacian operator which acts on the space variable  $x$  in d dimensional space.

If we are dealing with physical phenomenon like concentration of a chemical or population of a species, the equation leads to assumptions where  $u \geq 0$ . The term  $f(u)$  is a forcing term called activator. Eqn.2 can be re-written as  $|u|^{m-1}u$ . Suppose we say that  $D(u) = |u|^{m-1}$  then Eqn.2 can be written in divergence form.

$$\frac{\partial u}{\partial t} = \text{div} (D(u)\nabla u) + f(u) \quad (3)$$

In the limit of when  $m \rightarrow 1$ , the equation can be considered as the simplest case of a non-linear version of a classical heat equation as the activator term  $f(u)$  can still be non-linear. The diffusion function  $D(u) = D_U$  also becomes a constant, then the modified equation can be written as:

$$\frac{\partial u}{\partial t} = D_U \nabla^2 u + f(u) \quad (4)$$

Similarly, when  $m = 2$ , the diffusion coefficient is linearly dependent on the concentration ( $\therefore D(u) = |u|$ ). When  $m = 3$ , the diffusion coefficient is quadratic in  $u$  dependent on the concentration  $D(u) = u^2$ , etc.

$$\frac{\partial u}{\partial t} = \text{div} (D(u) \nabla u) + f(u) = \nabla (D(u) \nabla u) + f(u) \quad (5)$$

## 2.2 Reaction-Diffusion Equations

Using the theory of functional analysis, the theory can be extended to system of same forms where  $\vec{u} = (u_1, \dots, u_k)$ .

$$\frac{\partial \vec{u}}{\partial t} = D(u) \nabla^2 \vec{u} + \vec{F}(\vec{u}) \quad (6)$$

In this equation, we have extended the scalar form to a matrix form for a system of  $k$  dimensions. Here,  $\vec{u}$  is a  $k$ -dimensional vector,  $D(u)$  is a matrix of diffusion coefficients, and  $\vec{F}$  is a  $k$ -dimensional vector function describing the activation of the system. Explicitly, Eqn.5 can be written as:

$$\begin{pmatrix} U_t \\ V_t \end{pmatrix} = \begin{pmatrix} D_{11} & 0 \\ 0 & D_{22} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x} (u^m \frac{\partial u}{\partial x}) \\ \frac{\partial}{\partial x} (v^m \frac{\partial v}{\partial x}) \end{pmatrix} + \begin{pmatrix} F_1(u, v) \\ F_2(u, v) \end{pmatrix} \quad (7)$$

Here  $D(u)$  is taken to be  $D(u) = D_{ij}(u^m)$  and the constants form the diffusion matrix. We write  $u^{m-1}$  as  $u^m$  contracting the indices for simplicity. Notice that we are missing off-diagonal terms. This is because we ignored cross dependence of both  $u$  and  $v$  in the diffusion coefficient term. A more general analysis should consider cross diffusion terms as well which is also being studied for constant diffusion coefficients[5]. The analysis in this paper is limited to diagonal diffusion coefficient terms.

## 3 Method of solution or investigation

### 3.1 Linear Stability Analysis

The aim of this analysis is to determine if gradients in concentration can arise in the presence of both reaction and diffusion. Let  $u$  and  $v$  represent the concentrations of two species. In the absence of diffusion, the concentration of  $u$  and  $v$  changes only due to reaction, as described by:

$$\begin{pmatrix} u_t \\ v_t \end{pmatrix} = \begin{pmatrix} F_1(u, v) \\ F_2(u, v) \end{pmatrix} \quad (8)$$

We want to identify the properties of  $F_1$  and  $F_2$  that can result in Turing patterns. First, we find the steady state for the system, given by  $F_1(u^{ss}, v^{ss}) = 0$  and  $F_2(u^{ss}, v^{ss}) = 0$ . This steady state is spatially uniform, and the variables form a dynamical system that does not change with space. We assume that this is a stable steady state.

Next, we add diffusion to the system with Laplacian operator ( $\nabla^2$ ), introducing spatial dependence. By law of diffusion, a system homogenizes concentrations; when two species have different concentrations, diffusion tends to make the concentrations uniform over time, rather than creating concentration gradients. Now, we examine the combined effect of reaction and diffusion. We want to determine whether adding diffusion can destabilize the system and result in Turing instability, which could give rise to spatial patterns.

If  $u_{ss}, v_{ss}$  is stable, upon linearization and eigenvalue calculation, the eigenvalues will be negative. This is because the eigenvalues ( $\lambda$ ) of the Jacobian matrix  $J$  have negative real parts, where  $F_{1u}$  is the partial derivative of  $F_1$  with respect to  $u$  which is evaluated at the steady state:

$$J = \begin{pmatrix} F_{1u} & F_{1v} \\ F_{2u} & F_{2v} \end{pmatrix} \quad (9)$$

We find the determinant ( $\det$ ) of  $J - \lambda I = \vec{0}$ , where  $I$  is the identity matrix:

$$\det(J - \lambda I) = \det\begin{pmatrix} F_{1u} - \lambda & F_{1v} \\ F_{2u} & F_{2v} - \lambda \end{pmatrix} = 0 \quad (10)$$

This implies  $(F_{1u} - \lambda)(F_{2v} - \lambda) - F_{1v}F_{2u} = 0$ , then,

$$\lambda^2 - \lambda(F_{1u} + F_{2v}) + (F_{1u}F_{2v} - F_{1v}F_{2u}) = 0$$

This has  $\text{Re}(\lambda) < 0$  if  $F_{1u} + F_{2v} < 0$  is negative and  $F_{1u}F_{2v} - F_{1v}F_{2u} > 0$  is positive. It means the trace of  $J$  is negative and  $\det(J) > 0$ . This is only true for a two-dimensional system. The results will be different for a three-dimensional system because there is no sign change in real part of a root of a quadratic equation when these conditions are fulfilled which is not necessarily true in higher order polynomial equations. The partial derivatives only satisfy Eqn.10 when evaluated at stable state  $u_{ss}, v_{ss}$ .

Adding diffusion, we obtain an equation similar to Eqn.7. Let us work in one-dimensional x-direction. As we will see in the next section, for numerical simulation, we will use Neumann conditions for the boundary values because if we prevent any flux at the boundary, i.e.,  $\partial u / \partial x = \partial v / \partial x = 0$  for all times, then  $u$  or  $v$  will be contained inside the domain. If  $u_{ss}, v_{ss}$  is the steady-state solution of Eqn.8, then it is also the steady-state solution of Eqn.7 with the diffusion term and satisfies the boundary conditions.

Let's consider a small perturbation of the steady state, denoted by  $\tilde{u} = u - u_{ss}$  and  $\tilde{v} = v - v_{ss}$ . Inserting these expressions into Eqn.7 and linearizing,

we get:

$$\begin{pmatrix} \tilde{u}_t \\ \tilde{v}_t \end{pmatrix} = \begin{pmatrix} D_{11} & 0 \\ 0 & D_{22} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial x} (u^m \frac{\partial u}{\partial x} - u_{ss}^m \frac{\partial u_{ss}}{\partial x}) \\ \frac{\partial}{\partial x} (v^m \frac{\partial v}{\partial x} - v_{ss}^m \frac{\partial v_{ss}}{\partial x}) \end{pmatrix} + \begin{pmatrix} F_{1u} \tilde{u} + F_{1v} \tilde{v} \\ F_{2u} \tilde{u} + F_{2v} \tilde{v} \end{pmatrix} \quad (11)$$

$\frac{\partial}{\partial x} (u^m \frac{\partial u}{\partial x} - u_{ss}^m \frac{\partial u_{ss}}{\partial x})$  has four terms (chain rule) and because of linearity of  $\frac{\partial}{\partial x}$  we can write diffusion matrix in the middle as:

$$\begin{aligned} & \frac{\partial^2}{\partial x} (u - u_{ss})(u^m - u_{ss}^m) \left( \frac{mu^{m-1}}{\partial u} - \frac{mu_{ss}^{m-1}}{\partial u} \right) \\ & \frac{\partial^2}{\partial x} (v - v_{ss})(v^m - v_{ss}^m) \left( \frac{mv^{m-1}}{\partial v} - \frac{mv_{ss}^{m-1}}{\partial v} \right) \end{aligned} \quad (12)$$

Combining Eqn.11 and Eqn.13,

$$\begin{aligned} \tilde{u}_t &= D_{11} (u^m - u_{ss}^m) \left( \frac{mu^{m-1}}{\partial u} - \frac{mu_{ss}^{m-1}}{\partial u} \right) \frac{\partial^2}{\partial x^2} \tilde{u} + F_{1u} \tilde{u} + F_{1v} \tilde{v} \\ \tilde{v}_t &= D_{22} (v^m - v_{ss}^m) \left( \frac{mv^{m-1}}{\partial v} - \frac{mv_{ss}^{m-1}}{\partial v} \right) \frac{\partial^2}{\partial x^2} \tilde{v} + F_{1u} \tilde{u} + F_{1v} \tilde{v} \end{aligned} \quad (13)$$

If  $m$  were equal to 0, then the  $u^m$  will not have been expanded and neither its first derivative because the constant is absorbed in the constant  $D_{ii}$  in diffusion Matrix. Hence, Eqn.11 would reduce to:

$$\begin{aligned} \tilde{u}_t &= D_{11} \frac{\partial^2}{\partial x^2} \tilde{u} + F_{1u} \tilde{u} + F_{1v} \tilde{v} \\ \tilde{v}_t &= D_{22} \frac{\partial^2}{\partial x^2} \tilde{v} + F_{1u} \tilde{u} + F_{1v} \tilde{v} \end{aligned} \quad (14)$$

### 3.2 Numerical Simulation

The simulation models a two-dimensional reaction-diffusion system governed by partial differential equations. The system consists of two chemical species,  $U$  and  $V$ , whose concentrations change over time due to reactions and diffusion processes. The simulation uses the nine-point stencil method to approximate the Laplacian of the concentrations and updates the concentrations using Neumann boundary conditions. The reaction-diffusion system is described by parameters  $D_{11}$ ,  $D_{22}$ ,  $a$ , and  $b$ , and is discretized on a  $100 \times 100$  grid with a uniform spacing of  $dx = 1.0$ . The simulation is performed over six different time periods ( $t = 0, 0.1, 0.5, 1, 2$ , and  $5$ ), with a time step of  $dt = 0.001$ .

The simulation results are visualized using heatmaps for both species  $U$  and  $V$ . For each time period, the concentration of  $U$  and  $V$  is plotted in separate subplots, resulting in a  $2 \times 3$  grid layout for each species. The simulation is verified by checking if the initial and final total concentrations of the species are approximately equal within a specified tolerance. The heatmaps are displayed and saved as separate images for each species which can be used to analyze the evolution of the reaction-diffusion system over time.

### 3.3 Control Variable

We use Neumann zero flux conditions where the boundaries are impermeable. Neumann boundary conditions specify the derivative of a function at the boundary. We set the top boundary and the bottom boundary values equal to their respective neighboring row values. This means that the difference in values between the top boundary and its neighboring row is zero, and the same for the bottom boundary. Similarly, we do the same with the left and the right boundaries. Setting this Neumann Condition, the total concentration  $U_t + V_t = 0$ , as  $\partial u / \partial x = 0$ , does not change with time, which is used for verification for the simulation.

In this simulation, we aim to verify the conservation of mass in the reaction-diffusion system. To do this, we turn off the reaction functions, setting  $F_1 = 0$  and  $F_2 = 0$ , so that only the diffusion part of the differential equations remains. This allows us to focus solely on the diffusion dynamics and observe whether the total concentration of the two chemical species  $U$  and  $V$  is preserved over time.

Initially, we randomly initialize the  $U$  and  $V$  grids with values between 0 and 1, following a normal distribution. We then evolve the system for different time periods, calculating the total concentration ( $U + V$ ) at each time step. The resulting plot, shown in Figure 1, demonstrates that the sum of the concentrations does not significantly fluctuate and remains within a narrow range between 99.80 and 99.90, even after evolving the system for  $t = 10$ . This observation suggests that the conservation of mass is indeed maintained in the diffusion process, as expected.

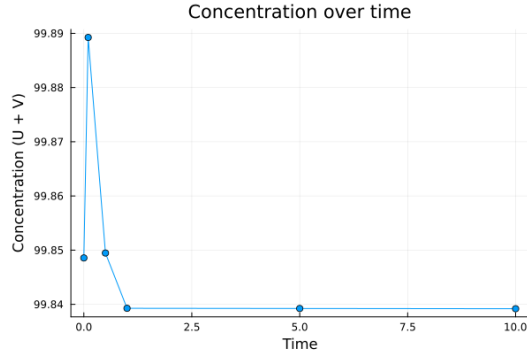


Figure 1: Evolution of the total concentration ( $U + V$ ) in the reaction-diffusion system with reaction functions turned off, showing conservation of mass over time.

### 3.4 Models

These models include definitions of the functions  $F_1$  and  $F_2$ . These are chosen conveniently according to reaction-diffusion models studied in the past.

- “Brusselator” model:

$$\begin{aligned} F_1(u, v) &= a - (b + 1)u + u^2v \\ F_2(u, v) &= bu - u^2v \end{aligned} \quad (15)$$

## 4 Results

### 4.1 Case when $m = 0$

We will consider “Brusselator” model for autocatalytic reaction with the choice of  $F_1$  and  $F_2$  from Eqn.15. Suppose we define following system:

$$\begin{aligned} u_t &= D_{11}u_{xx} + a - (b + 1)u + u^2v \\ v_t &= D_{22}v_{xx} + bu - u^2v \end{aligned} \quad (16)$$

The steady state solution for Eqn.16 is for all  $(x, t)$ . It can be verified as  $F_1 = 0$  and  $F_2 = 0$  in Eqn.15 for steady state in Eqn.17

$$\begin{pmatrix} u_{ss} \\ v_{ss} \end{pmatrix} = \begin{pmatrix} a \\ b/a \end{pmatrix} \quad (17)$$

As done in Eqn.14, we linearize Eqn.16 around steady state Eqn.17,

$$\begin{aligned} u_t &= D_{11}u_{xx} + (b - 1)u + a^2v \\ v_t &= D_{22}v_{xx} - bu - a^2v \end{aligned} \quad (18)$$

$$\begin{bmatrix} u_t \\ v_t \end{bmatrix} = \begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix} \begin{bmatrix} u_{xx} \\ v_{xx} \end{bmatrix} + \begin{bmatrix} b - 1 & a^2 \\ -b & -a^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (19)$$

Lets look for solutions that is exponential in time and periodic in space  $x$ , we apply Fourier ansatz.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} e^{\lambda t} \cos(\vec{k} \cdot x) \quad (20)$$

Substituting this into Eqn.19, we get:

$$\lambda \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{bmatrix} -D_{11}k^2 + b - 1 & a^2 \\ -b & -D_{22}k^2 - a^2 \end{bmatrix} \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = J_D \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \quad (21)$$

It is clear that the determinant of modified Jacobian Matrix  $J_D$  is quadratic in  $k^2$ . For spatial patterns we require to have determinant of the matrix in Eqn.21 should be negative.

$$(D_{11}k^2 + b - 1) * (D_{22}k^2 - a^2) + a^2b < 0 \quad (22)$$

We create a range of wave numbers  $k$ , plots the determinant in Eqn.22 as a function of  $k$  for two different  $b$  values (7.5 and 2). When the determinant is negative, it means that the diffusion process has destabilized the steady state,

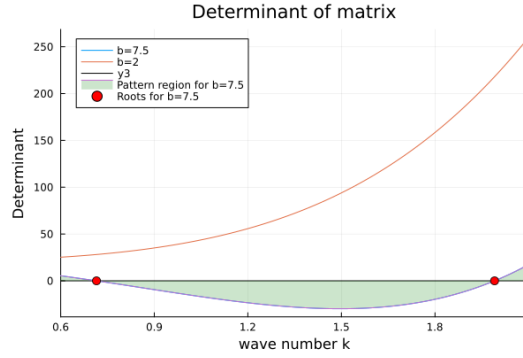


Figure 2: Determinant of the Jacobian Matrix vs. Wave Number ( $k$ ) for Spatial Patterns with Varying  $b$  Values ( $D_{11} = 1, D_{22} = 10, a = 4.5$ )

allowing the system to break the spatial homogeneity and create Turing patterns. Indeed, Figure 2 shows how for appropriate value of  $b$ , we can find a range of wave number  $k \neq 0$ , which means oscillatory solution in time. For the case where  $b = 7.5$  we get the range  $k = 0.2$  to  $k = 2.0$ , where the determinant is negative. This is the region where the Turing patterns are possible. It's important to note that the negative determinant is just one condition that needs to be satisfied for the formation of Turing patterns. Other criteria and properties of the system will determine the actual emergence and shape of the patterns (one such condition is that the diffusion constants must be unequal  $D_{11} \neq D_{22}$ ).



#### 4.1.1 Simulation of the Brusselator model

The Brusselator model simulations are presented as 2-D heat maps for two chemical species, U and V. The first set of simulations (Figures 3 and 4) depict the system with only diffusion terms included, ignoring reaction terms. These figures show the evolution of randomly initialized concentrations U and V over time. In both cases, the diffusion process doesn't lead to stable patterns. Instead, the patterns change continuously, indicating instability in the system.

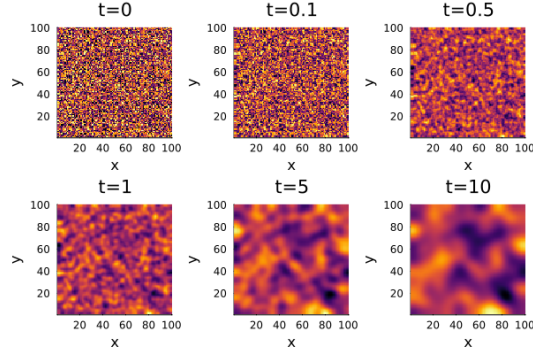


Figure 3: Evolution of Concentration U with Diffusion Only

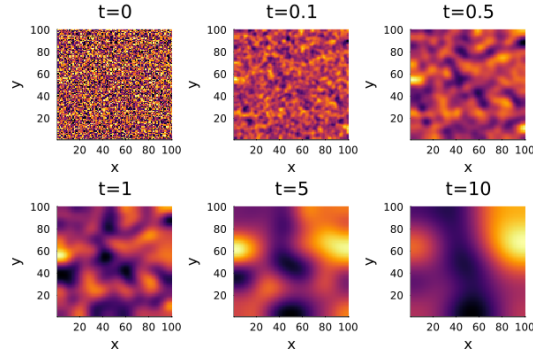


Figure 4: Evolution of Concentration V with Diffusion Only

In the second set of simulations (Figures 5 and 6), both reaction and diffusion terms are included. The parameters  $a$ ,  $b$ ,  $D_{11}$ , and  $D_{22}$  are chosen to satisfy the conditions for Turing pattern formation, as identified through linear analysis in the section before. These figures show the evolution of concentrations U and V over time for different time periods. In contrast to the diffusion-only case, the patterns become stable and remain unchanged as time progresses, indicating that the reaction and diffusion together have led to stable spatial patterns in the system.

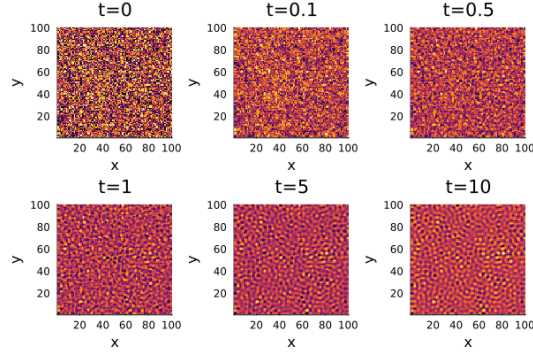


Figure 5: Evolution of Concentration U with Reaction and Diffusion

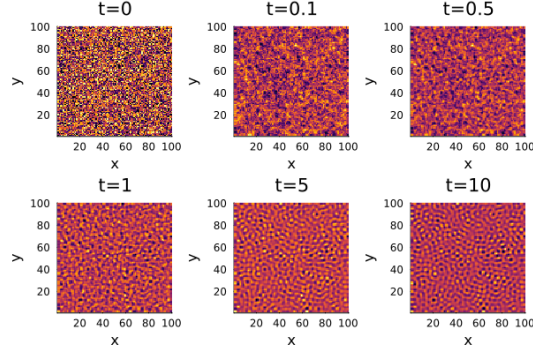


Figure 6: Evolution of Concentration V with Reaction and Diffusion

## 4.2 Case for $m \geq 1$

For case where  $m = 1$ , the diffusion coefficient is linearly dependent to  $D(u) = D_{ii}u$ . This case has been explored in [6], where it is shown that for this particular case:

$$\begin{aligned} D_{11}F_{2v} + D_{22}F_{1v} &\geq 0 \\ (D_{11}F_{2v} + D_{22}F_{1v})^2 - 4 \det D \det F &\geq 0 \end{aligned} \quad (23)$$

We can use Eqn.23 to estimate regions of instability as we did in case with  $m = 1$  (to be done) for the choices of  $F_1$  and  $F_2$  that satisfy Eqn.23.

In the case where  $m = 1$ : if we expand the diffusion matrix at the steady state, we can write the linearized equation like in the case of  $m = 0$  as:

$$\begin{bmatrix} u_t \\ v_t \end{bmatrix} = \begin{bmatrix} D_{11}u_0 & 0 \\ 0 & D_{22}v_0 \end{bmatrix} \begin{bmatrix} u_{xx} \\ v_{xx} \end{bmatrix} + \begin{bmatrix} b-1 & a^2 \\ -b & -a^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (24)$$

We know the steady state solution  $u_0, v_0$  from Eqn.17. For the  $m = 1$  case, the linearized equation at the steady state can be written as in Eqn. 24. By applying the Fourier ansatz, as was done for the  $m = 0$  case, we can derive the

expression in Eqn. 25, which has a modification to include the linear terms  $u_0$  and  $v_0$  in the diffusion matrix.

$$(D_{11}k^2u_0 + b - 1) * (D_{22}v_0k^2 - a^2) + a^2b < 0 \quad (25)$$

By plotting Eqn. 25, we can explore the regions in the parameter space that allow for instability. It is observed that the region is not unstable at  $b = 7.5$ , as it was in the  $m = 0$  case, when using the same value of  $a = 4.5$ . However, instability regions can still be found at higher values of  $b$ . For example, as seen in Figure 7, when  $b = 13$ , there are certain wave numbers that result in oscillatory solutions in time. This demonstrates that the regions of instability change when the diffusion coefficient has a linear dependence on the concentration.

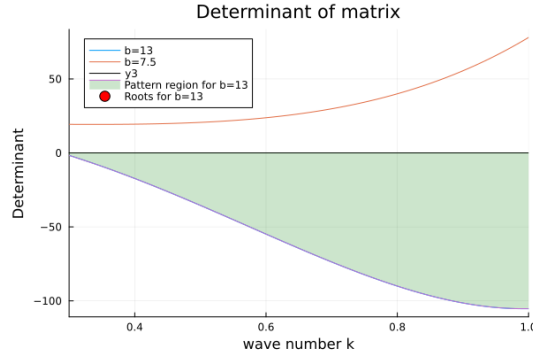


Figure 7:  $m = 1$  case: Determinant of the linearized system as a function of the wave number  $k^2$  for  $a = 4.5$  and  $b = 13$ . The shaded region is highlighted that lead to oscillatory solutions in time, resulting in Turing pattern formation.

For cases with  $m = 1$ , a linear analysis would also involve quadratic terms in the steady-state solution,  $u_0^2$  and  $v_0^2$ . This would yield conditions analogous to Eqn.23 that functions  $F_1$  and  $F_2$ , and their Jacobian should satisfy. However, to obtain a comprehensive stability analysis, techniques such as Hopf Instability should be employed, which are not covered in this analysis.

## 5 Conclusion

In this study, we investigated the formation of Turing patterns for different values of  $m$ , where  $m$  characterizes the power-law dependence of the diffusion coefficients on the concentrations of reacting species. We used linear stability analysis to derive the conditions for pattern formation in the Brusselator model for the case  $m = 0$ . The simulation results, showing the formation of stable spatial patterns, support our analytical findings.

For the case when  $m = 1$ , we analyzed the impact of linear dependence of diffusion coefficients on concentrations. The linearized equation and instability conditions were modified accordingly, and we found that the regions of

instability change when the diffusion coefficients have a linear dependence on concentration. This demonstrates the importance of considering the effect of  $m$  on the pattern formation process.

For cases with  $m > 1$ , a more technical analysis like Hopf Instability is required to establish proper parametric conditions for pattern formation. In addition, nonlinear bifurcation theory could be employed as an extension of this analysis, providing a qualitative approximation of the system's behavior[1]. Further investigations could also explore the applicability of non-equilibrium thermodynamics models to understand the formation of Turing patterns in reaction-diffusion systems[1] which this project omits.

## 6 Code

### 6.1 Code for determinant and wave number

```

1 using Plots
2 using LinearAlgebra
3 using Roots
4
5 # determinant of the Brusselator model
6 function model_determinant(k, d11, d22, a, b)
7     return (-d11*k^2+b-1)*(-d22*k^2-a^2) + a^2*b
8 end
9
10 # wave numbers range
11 k = range(0.6, 2.1, length = 100)
12
13 plot()
14
15 # find roots of the determinant equation when b=7.5
16 b_value = 7.5
17 f(k) = model_determinant(k, 1, 10, 4.5, b_value)
18 roots = find_zeros(f, 0.6, 2.1)
19
20 #b = 7.5
21 plot!(k, model_determinant.(k, 1, 10, 4.5, b_value), label="
    b=7.5")
22
23 #b = 2
24 plot!(k, model_determinant.(k, 1, 10, 4.5, 2), label="b=2")
25 hline!([0], color=:black)
26 title!("Determinant of matrix")
27 xlabel!("wave number k")
28 ylabel!("Determinant")
29 legend=:topleft
30
31 # shade the region where determinant is negative (spatial
    pattern occurs)

```

```

31 plot!(x->model_determinant(x, 1, 10, 4.5, b_value),
      fillrange=0, fillalpha=0.2, fillcolor=:green, label="
      Pattern region for b=7.5", xlims=(0.6, 2.1))
32
33 # display roots on the plot
34 scatter!(roots, [0, 0], color=:red, markersize=5, label="
      Roots for b=7.5")
35
36 savefig("det_extended.png")

```

## 6.2 Simulation

```

1 using Plots
2
3 function del2_9(a::Matrix{Float64}, dx=1.0)
4     #=
5     Returns a finite-difference approximation of the
6     laplacian of the array a,
7     with lattice spacing dx, using the nine-point stencil:
8
9             1/6  2/3  1/6
10            2/3 -10/3 2/3
11            1/6  2/3  1/6
12     =#
13     del2 = zeros(size(a))
14     del2[2:end-1, 2:end-1] .= (4.0 * (a[2:end-1, 3:end] + a
15                                     [2:end-1, 1:end-2] +
16                                     a[3:end, 2:end-1] + a
17                                     [1:end-2, 2:end-1])
18                                +
19                                (a[1:end-2, 1:end-2] + a[1:
20                                end-2, 3:end] +
21                                a[3:end, 1:end-2] + a[3:end,
22                                3:end]) -
23                                20.0 * a[2:end-1, 2:end-1])
24                                ./ (6.0 * dx^2)
25
26     return del2
27 end
28
29 function apply_neumann_conditions!(U::Matrix{Float64}, V::
30 Matrix{Float64})
31     for Z in (U, V)
32         Z[1, :] = Z[2, :]
33         Z[end, :] = Z[end-1, :]
34         Z[:, 1] = Z[:, 2]
35         Z[:, end] = Z[:, end-1]
36     end
37 end

```

```

29
30 function simulate_reaction_diffusion(D11, D22, a, b, U, V,
    dt, dx, steps, fun=1)
31     for t in 1:steps
32         Uxx = del2_9(U, dx)[2:end-1, 2:end-1]
33         Vxx = del2_9(V, dx)[2:end-1, 2:end-1]
34
35         u = U[2:end-1, 2:end-1]
36         v = V[2:end-1, 2:end-1]
37         if fun == 0
38             U[2:end-1, 2:end-1] = u + dt * (D11 * Uxx)
39             V[2:end-1, 2:end-1] = v + dt * (D22 * Vxx)
40         else
41             U[2:end-1, 2:end-1] = u + dt * (D11 * Uxx + (b -
                1) * u + a^2 * v)
42             V[2:end-1, 2:end-1] = v + dt * (D22 * Vxx - b *
                u - a^2 * v)
43
44         end
45
46         apply_neumann_conditions!(U, V)
47     end
48     return U, V
49 end
50
51 concentrations_over_time = []
52 function verify_simulation(U0::Matrix{Float64}, V0::Matrix{
    Float64}, U::Matrix{Float64}, V::Matrix{Float64}, tol=1e
    -3)
53     init_concentration = (sum(U0) + sum(V0))/n
54     curr_concentration = (sum(U) + sum(V))/n
55
56     append!(concentrations_over_time, curr_concentration)
57     return isapprox(init_concentration, curr_concentration,
        atol=tol)
58 end
59
60 # parameters
61 D11 = 1.0
62 D22 = 10.0
63 a = 4.5
64 b = 7.5
65
66 # grid size 2D domain
67 n = 100
68 dx = 1.0
69
70 # Time
71 dt = 0.001
72 total_time = [0, 0.1, 0.5, 1, 5, 10] #six time periods

```

```

73
74 steps_per_plot = 1000*total_time
75
76 # initialize matrices U and V with random values between 0
   and 1
77 U0 = rand(Float64, n, n)
78 V0 = rand(Float64, n, n)
79 #append!(concentrations_over_time, sum(U0) + sum(V0)/n)
80 # initialize a plots for u and v separately
81 plt = plot(layout=(2, 3), legend=false, grid=false, xlabel="
   x", ylabel="y", title=["t=0" "t=0.1" "t=0.5" "t=1" "t=5"
   "t=10"])
82 plt1 = plot(layout=(2, 3), legend=false, grid=false, xlabel=
   "x", ylabel="y", title=["t=0" "t=0.1" "t=0.5" "t=1" "t=5"
   "t=10"])
83
84 # simulate the reaction-diffusion system and plot for
   specified time steps
85 for (i, time) in enumerate(total_time)
86     steps = steps_per_plot[i]
87     U, V = simulate_reaction_diffusion(D11, D22, a, b, U0,
        V0, dt, dx, steps, 1)
88
89     # set f = 0
90     #U, V = simulate_reaction_diffusion(D11, D22, a, b, U0,
        V0, dt, dx, steps, 0)
91     # verify the simulation, prints false if initial
        conditions do not match the final conditions
92     is_verified = verify_simulation(U0, V0, U, V)
93     if !is_verified
94         println("Simulation not verified at t=$time")
95     end
96
97     # heatmap of U, V to the corresponding subplots
98     heatmap!(plt[i], U)
99     heatmap!(plt1[i], V)
100 end
101 concentration_plot = plot(total_time,
    concentrations_over_time, xlabel="Time", ylabel="
    Concentration (U + V)", title="Concentration over time",
    legend=false, marker=:circle)
102 display(concentration_plot)
103 savefig(concentration_plot, "concentration_over_time.png")
104
105 display(plt)
106 savefig(plt, "U_resultss_with_reaction.png")
107 display(plt1)
108 savefig(plt1, "V_resultss_with_reaction.png")

```

## References

- [1] T. Leppanen. *Current Topics in Physics*. Imperial College Press, London, 2004. page.199- and references therein.
- [2] Philip K. Maini, Kevin J. Painter, and Helene Nguyen Phong Chau. Spatial pattern formation in chemical and biological systems. *Journal of the Chemical Society, Faraday Transactions*, 93:3601–3610, 1997.
- [3] Alan Mathison Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237:37–72, 1952.
- [4] Juan Luis Vazquez. *The Porous Medium Equation: Mathematical Theory*. University Autonoma de Madrid, Madrid, 2007. page:1-15.
- [5] E.P. Zemskov. Nonlinear analysis of a reaction-diffusion system: Amplitude equations. *Journal of Experimental and Theoretical Physics*, 115(4):729–732, 2012.
- [6] E.P. Zemskov. Turing instability in reaction-diffusion systems with nonlinear diffusion. *Journal of Experimental and Theoretical Physics*, 117:764–769, 2013.