

Treball final de grau
GRAU DE MATEMÀTIQUES

Facultat de Matemàtiques
Universitat de Barcelona

**REACTION-DIFFUSION
SYSTEMS AND PATTERN
FORMATION**

Lucia A. Alonso García

Director: Dr. Àlex Haro
Realitzat a: Departament de Matemàtica Aplicada i Anàlisi
Barcelona, January 18, 2016

Abstract

This bachelor's degree final thesis deals with the problem of pattern formation in living forms, with special interest in animal skin. Mathematically, this phenomenon is caused due to perturbations on a system of differential equations initially in equilibrium, the so-called reaction-diffusion system, which is time independent but it becomes spatially unstable when is perturbed. This spatial instability is the one which causes the pattern. Therefore, we have started this project obtaining this mathematical model and then studying under which conditions the system becomes spatially unstable. Particularly, these conditions were given by the mathematician Alan Turing in *The Chemical Basis of Morphogenesis* in 1952. Furthermore, another important point developed in this undergraduate thesis has been to apply this theory on animal body, particularly, to see how the geometry of the body affects the design of the resulting pattern (striped or spotted). Finally, we have made some simulations of the system of differential equations in adapted domains with the characteristics we want to emphasize of animal body (scale, thickness and curvature). In order to do them, we have designed three programs in C which approximate the solution of the system using the explicit scheme of a numerical method, the so-called finite difference method.

Acknowledgments

I would like to express my gratitude to my advisor Àlex Haro who has encouraged me, through his helpful advice and guidance, to always do things better. Furthermore, I want to thank him his time and the confidence he has shown on me.

I would also like to thank Manel Vila for sending me a lot of information which has been really useful for me and Anna Puig for her help with the program *ParaView*.

Last, but far from least, I want to thank my family and friends who, as they always do, have supported me throughout this project.

Contents

1	Introduction	1
2	Reaction-diffusion systems	3
2.1	Modeling of chemical reactions	3
2.1.1	Law of mass action: a simple example	4
2.1.2	Reversible reactions	6
2.1.3	General reactions	7
2.2	Modeling of diffusion	8
2.3	General model of reaction-diffusion systems	10
3	Turing instability. Activator-inhibitor systems	11
3.1	Linear stability: basics	11
3.2	Turing instability	13
4	Pattern formation on animal skin: The importance of the geometry of the body	18
4.1	Murray's theory	18
4.2	Effect of the geometry	19
4.2.1	Effect of the scale	19
4.2.2	Importance of the thickness	20
4.2.3	Short discussion about the curvature of the torus	21
5	Simulation results	23
5.1	Conditions for pattern formation in Schakenberg's model	23
5.2	A finite difference method for reaction-diffusion systems	24
5.2.1	Spatial and temporal approximation	25
5.2.2	Boundary conditions	25
5.2.3	Derivatives approximation	25
5.2.4	Explicit method for reaction-diffusion equations	26
5.3	Simulations	28
6	Conclusions	33
6.1	Conclusions on the approach of the project	33
6.2	Conclusions on the results of the project	34

A	Laplacian on a torus	36
B	Stability of the explicit finite difference method	37
C	The program	40

1 Introduction

This project stems, firstly, from my huge interest in applied mathematics and biology.

Nowadays it is known that lots of biological processes are explained by mathematical models, which enable us to make predictions and to understand better some biological problems with special complexity. However, traditionally these two scientific areas worked separately, and that is the reason why there are more and more research groups trying to give mathematical models to different biological processes.

One of the biological phenomena that is currently studied the most is pattern formation. One looks around and can observe that there is a large number of patterns in nature: desert dunes, spider webs, animal skin (zebras, leopards...), leaves... This observation of the nature led many scientists to wonder why they are originated, and thus some theories started to appear.

However, the most known and accepted mathematical theory and model are the ones given by Alan Turing in 1952 in his article *The Chemical Basis of Morphogenesis*. Particularly, he studied pattern formation in living forms.

This theory states that pattern formation takes place during the morphogenesis, which is the shaping of an organism by embryological processes of differentiation of cells, tissues and organs and the development of organ systems according to the genetic blueprint of the potential organism and environmental conditions. Particularly, Turing affirmed that patterns emerge from the appearance of instabilities over two processes that occur simultaneously: chemical reactions and diffusion, which affect the concentration of a specific substances called morphogens.

Although Turing's theory is the most accepted one for pattern formation, it is quite general in that it includes many species and patterns. This is why many modern scientists have taken this theory as a starting point to study specific cases.

One example of them, and the one that I will study along my project, is the mathematician James D. Murray. He developed and applied Turing's theory in animals, like mammals (zebras, leopards, big cats...), seashells, butterflies...

Particularly, when I was looking for the topic of my project, I discovered an article which started with the question: "How do zebras get its stripes?". I found really interesting to obtain a mathematical explanation of this phenomenon, so I decided to focus my project on mammals and, specifically, study why there are some animals with stripes and some other ones with spots.

Structure of the project

The topic of my project can be summarized as the study of pattern formation on animal skin. From this, I have developed its contents through four more chapters.

The theory of my project starts in **Chapter 2**. I have mentioned before that Alan Turing stated that pattern formation in living forms emerge from a combi-

nation of two biological processes: chemical reactions and diffusion. Therefore, I started this chapter, and so my project, modeling these two processes in order to finally obtain a general model for pattern formation. It consists of a system of partial differential equations, the so-called *reaction-diffusion system*.

Once obtained this model, our next goal, which is developed in **Chapter 3**, was to study how patterns take place from disturbances applied to this reaction-diffusion process. However, it is difficult to deal with this problem directly since it is nonlinear. Therefore, we did a previous linearization of the system around an equilibrium, using tools of dynamical systems. Then, we imposed Alan Turing's hypothesis on the linearized system and obtained some mathematical conditions for pattern formation.

In **Chapter 4** we study pattern formation on animal skin. We start giving Murray's theory, which explains, firstly, the biological process in which patterns are formed and, secondly, how geometry affects the resulting pattern, that is, the appearance of stripes or spots according to body constraints. In particular, he started studying the skin of legs and tails, because it is similar to a cylinder surface, which is well known by the mathematicians and which is useful to show the importance of the thickness. However, because of the kind of boundary conditions that we consider along this project, our study is on a torus, from which we also can observe the effect of the curvature.

Finally, we have included a last section of simulations, **Chapter 5**. We have chosen a particular reaction-diffusion system, the *Schakenberg's model* in three domains (1D, flat torus and torus), from which we select one parameter related with geometry and we tune it in order to obtain different patterns and show its importance. To do the simulations we have used an explicit finite difference method.

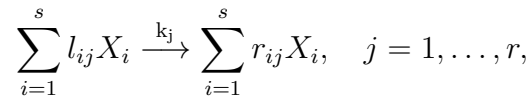
2 Reaction-diffusion systems

In the introduction of the project we have explained that pattern formation in animals takes place during the morphogenesis due to perturbations in two processes that occur simultaneously, chemical reactions and diffusion of substances called morphogens.

Therefore, our goal in this chapter is to obtain a mathematical model of reaction-diffusion systems. We have started modeling the two processes mentioned before: chemical reactions and diffusion. They are developed in sections 2.1 and 2.2 independently, that is, one in the absence of the other one. Then, using this information, in section 2.3 we have been able to give the general model we want to obtain, which we will see that consists of a system of partial differential equations.

2.1 Modeling of chemical reactions

Chemical reactions are the variation of substances concentration due to the interaction with other species and with itself. The usual notation for this process is:



where:

- s denotes the number of substances and index i indicates each substance.
- r denotes the number of reactions and index j indicates each reaction.
- X_i are the substances.
- l_i and r_i are the stoichiometric coefficients (numbers of molecules of each substance).
- k_j is the rate or kinetic constant of each reaction j . We will see what does it mean in 2.1.1.

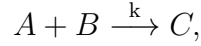
The modeling of this process is based in the *law of mass action*. We will start introducing this law and then we will develop the model of general reactions.

Therefore, our main goal in this section is to study along 2.1.1, 2.1.2 and 2.1.3 the shape of the equations which model chemical reactions. However, from these equations, we have been able to give in 2.1.1 further information which complements the theory of this project.

Observation 1. In order to simplify, in case of reactions with a small number of species, it is usual to denote by capital letters $A, B, C...$ each substance (see subsections 2.1.1 and 2.1.2).

2.1.1 Law of mass action: a simple example

We first consider a very simple reaction:



Let $a=[A]$, $b=[B]$ and $c=[C]$ be the concentrations of substances A, B and C. We usually call the left terms of the reaction (in this case A and B) *reactants* and, the right ones (C), *products*.

Our first assumption is that the rate v of the reaction corresponds to the number of collisions between the reactants A and B (r_1ab , r_1 constant) multiplied by the probability that indeed the reaction happens in case of collision (r_2). Thus,

$$v = r_1r_2ab = kab,$$

that is, the reaction rate is proportional to the products of the concentrations of the reactants (*law of mass action*). Particularly, this proportion corresponds to the kinetic constant k of the reaction.

Therefore, we can describe now the variation of a , b and c by:

$$\Delta a = -kab\Delta t, \quad \Delta b = -kab\Delta t, \quad \Delta c = kab\Delta t.$$

where the signs represent the appearance or disappearance of the substance. Then, if $\Delta t \rightarrow 0$,

$$\frac{da}{dt} = -kab, \quad \frac{db}{dt} = -kab, \quad \frac{dc}{dt} = kab.$$

Finally, as an extension of the project, we introduce the concept of *first integral* and we will see that they have to do with the conservation of mass and we can use them to find solutions of the system of nonlinear differential equations with any initial condition.

Definition 2.1.1.1. A continuously differentiable function $H : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be a *first integral* of the system $\frac{dx}{dt} = f(x)$, $x \in U$, if $H(x(t))$ is constant for any solution $x(t)$ of the system, that is, its orbital derivative satisfies $\dot{H}(x) := DH(x)f(x) = 0 \forall x \in U$.

We now define the functions:

$$H_1(a, b, c) = a + c$$

$$H_2(a, b, c) = b + c$$

These functions satisfy:

$$\dot{H}_1(a, b, c) = \dot{a} + \dot{c} = -kab + kab = 0$$

$$\dot{H}_2(a, b, c) = \dot{b} + \dot{c} = -kab + kab = 0$$

which implies that H_1 and H_2 are constant for any solution $(a(t), b(t), c(t))$, so that, they are first integrals of the system.

Particularly, if we define the function

$$H(a, b, c) = a + b + 2c$$

we observe that its orbital derivative is 0 as well and thus it is a first integral of the system.

Therefore, through these functions we can see that there is a preservation of mass.

These first integrals are also useful to find a solution of the nonlinear system of equations. Given any initial condition

$$a(0) = a_0 \quad b(0) = b_0 \quad c(0) = c_0$$

we obtain:

$$H_1(a_0, b_0, c_0) = a_0 + c_0,$$

$$H_2(a_0, b_0, c_0) = b_0 + c_0.$$

We have seen that these functions are constant. Thus,

$$H_1(a(t), b(t), c(t)) = H_1(a_0, b_0, c_0),$$

$$H_2(a(t), b(t), c(t)) = H_2(a_0, b_0, c_0).$$

It determines two planes and lead to the system of equations:

$$\begin{cases} a(t) + c(t) = a_0 + c_0 \\ b(t) + c(t) = b_0 + c_0 \end{cases}$$

which solutions are lines. We can write:

$$\begin{cases} a(t) = a_0 + c_0 - c(t) \\ b(t) = b_0 + c_0 - c(t) \end{cases}$$

However, we have additional information that we can use to obtain explicit solutions of $c(t)$. We can substitute this expressions of $a(t)$ and $b(t)$ in the differential equation

$$\frac{dc}{dt} = kab,$$

obtaining (for $a_0 \neq b_0$):

$$\frac{dc}{dt} = k(a_0 + c_0 - c(t))(b_0 + c_0 - c(t))$$

Therefore, we have an ODE of only $c(t)$ which we can solve using variables separation and obtain the solution

$$c(t) = \frac{(a_0 + c_0) - (b_0 + c_0)e^{(a_0-b_0)(kt+C)}}{1 - e^{(a_0-b_0)(kt+C)}},$$

where C is a constant determined by the initial condition $c(0) = c_0$.

Thus, from the conservation of mass and the first integrals, we have the solution of the system with initial conditions $a(0) = a_0$, $b(0) = b_0$ and $c(0) = c_0$, which is (for $a_0 \neq b_0$):

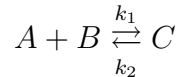
$$a(t) = a_0 + c_0 - \frac{(a_0 + c_0) - (b_0 + c_0)e^{(a_0-b_0)(kt+C)}}{1 - e^{(a_0-b_0)(kt+C)}}$$

$$b(t) = b_0 + c_0 - \frac{(a_0 + c_0) - (b_0 + c_0)e^{(a_0-b_0)(kt+C)}}{1 - e^{(a_0-b_0)(kt+C)}}$$

$$c(t) = \frac{(a_0 + c_0) - (b_0 + c_0)e^{(a_0-b_0)(kt+C)}}{1 - e^{(a_0-b_0)(kt+C)}}.$$

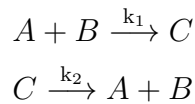
2.1.2 Reversible reactions

We now consider a reversible reaction, i.e,



with reaction rates k_1 and k_2 .

We can consider this reaction as the following system of two reactions:



Therefore, the equations for the reversible reaction are:

$$\frac{da}{dt} = -k_1ab + k_2c, \quad \frac{db}{dt} = -k_1ab + k_2c, \quad \frac{dc}{dt} = k_1ab - k_2c,$$

where the first term of the right side of the equalities corresponds to the first reaction and, the second term, to the second reaction.

Finally, we can observe that this reaction has the same first integrals of the reaction studied before, that is,

$$H(a, b, c) = a + b + 2c$$

$$H_1(a, b, c) = a + c$$

$$H_2(a, b, c) = b + c$$

are constant for all $a(t)$, $b(t)$ and $c(t)$.

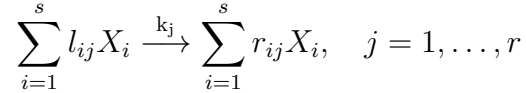
Therefore, one can repeat the same process of the previous case and obtain explicit solutions of $a(t)$, $b(t)$ and $c(t)$ from the differential equation:

$$\frac{dc}{dt} = k_1(a_0 + c_0 - c(t))(b_0 + c_0 - c(t)) + k_2c(t).$$

However, since the procedure is more complex than the first one and it is supplementary information of the project, we just indicate the equation.

2.1.3 General reactions

We now consider the general case that we have given at the beginning of 2.1, which was:



In this case we only give the system of differential equations which models these reactions. Taking $x_i = [X_i]$ as the concentration of each X_i , we obtain from the law of mass action:

$$\begin{aligned} \dot{X}_1 &= -k_1 l_{11} \prod_{i=1}^s X_i^{l_{i1}} + k_1 r_{11} \prod_{i=1}^s X_i^{l_{i1}} + \dots \\ \dot{X}_2 &= -k_1 l_{21} \prod_{i=1}^s X_i^{l_{i1}} + k_1 r_{21} \prod_{i=1}^s X_i^{l_{i1}} + \dots \\ &\vdots \\ \dot{X}_s &= -k_1 l_{s1} \prod_{i=1}^s X_i^{l_{i1}} + k_1 r_{s1} \prod_{i=1}^s X_i^{l_{i1}} + \dots \end{aligned}$$

Therefore, we can write the ODE system for general reactions as:

$$\dot{X}_i = \sum_{j=1}^r k_j (r_{ij} - l_{ij}) \prod_{n=1}^s X_n^{l_{nj}}, \quad i = 1, \dots, s.$$

In conclusion, from these three different cases studied along this section, we can say that the general shape of the equations which model chemical reactions is a nonlinear ODE system which depends on time.

2.2 Modeling of diffusion

The second biological process that we have to study is diffusion, which is the one generated by a substance's random movement. The classical diffusion mathematical modeling is based on the *Fick's laws* and *conservation of mass*.

We start with $u(t, x) \in \mathbb{R}$ as a substance concentration at the point $x \in \mathbb{R}^3$ and time t inside a container.

The model assumes that there is a flux, $J(t, x) \in \mathbb{R}^3$, that is a vector which points into the general direction of movement and $|J(t, x)|$ is proportional to the amount of particles which flow in that direction per unit time.

We choose a test volume Ω with boundary Γ . If no reactions take place, then the only factor which influences the change of concentration in Ω is the flux through Γ , i.e.,

$$\frac{d}{dt} \int_{\Omega} u(t, x) dV = - \int_{\Gamma} J(t, x) dS$$

where dV corresponds to volume integral (\mathbb{R}^3) and dS to surface integration (\mathbb{R}^2).

Let's explain carefully the equality above.

On the one hand, the first part of the equality represents the variation of the concentration in the test volume. On the other hand, we remember that, by definition, we can write the surface integral in the form:

$$\int_{\Gamma} F dS = \int_{\Gamma} F \vec{n} dS$$

where \vec{n} is the unit normal vector to the surface. Hence, the physical interpretation of the second integral is the total flux through the surface Γ .

Therefore, this equality means that the variation of concentration of the particles in Ω corresponds to the quantity of them entering through Γ to the test volume.

We now remember the *Gauss divergence theorem*:

$$\int_{\Gamma} J(t, x) dS = \int_{\Omega} \operatorname{div} J(t, x) dV$$

Using this theorem in the above expression we obtain:

$$\begin{aligned} \frac{d}{dt} \int_{\Omega} u(t, x) dV &= - \int_{\Omega} \operatorname{div} J(t, x) dV \\ \rightarrow \int_{\Omega} \left(\frac{\partial}{\partial t} u(t, x) + \operatorname{div} J(t, x) \right) dV &= 0 \end{aligned}$$

Since this equation is satisfied for all Ω , the integral can be omitted, leading to

$$\frac{\partial}{\partial t} u + \operatorname{div} J = 0 \tag{2.1}$$

the so-called *first law of Fick*.

The elimination of the volume integral is justified by:

Proposition 2.2.0.1. *If $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuous function such that $\forall \Omega \subset U$ compact $\int_{\Omega} f dV = 0$, then $f = 0$.*

Proof. We start supposing that $\exists x \in U$ such that $f(x) > 0$. This implies that $\exists \delta > 0$ such that $\bar{B}_{\delta}(x) \subset U$ and $\forall y \in \bar{B}_{\delta}(x)$ satisfies $f(y) > 0$. Taking $\Omega = \bar{B}_{\delta}(x)$, which is compact since it is a closed ball, we obtain by the integral properties:

$$f(y) > 0 \quad \forall y \in \Omega \implies \int_{\Omega} f dV > 0.$$

If $f(x) < 0$ the proof follows similar lines and we come to $\int_{\Omega} f dV < 0$.

Therefore, we have shown that if $f \neq 0$ then $\int_{\Omega} f dV \neq 0$, from which we can conclude that, if we have the conditions stated in the lemma, $\int_{\Omega} f dV = 0$ implies $f = 0$.

□

We now introduce *Fick's second law*. This law says that flux moves from the side with the highest concentration to the lowest. In order to model it, we remember that the gradient of $u \in \mathbb{R}^3$

$$\nabla u = \begin{pmatrix} \frac{\partial u}{\partial x_1} \\ \frac{\partial u}{\partial x_2} \\ \frac{\partial u}{\partial x_3} \end{pmatrix}$$

points to the region with the highest concentration. Therefore, we can conclude that flux has the direction of the negative gradient of the particle distribution. Hence, we obtain the equality

$$J = -D\nabla u$$

where D is the diffusion coefficient. We assume it is homogeneous, thus D is constant.

Inserting the second law into (2.1), we obtain the diffusion equation

$$\frac{\partial u}{\partial t} = \operatorname{div} \nabla u = D\Delta u,$$

which is also called *heat equation*.

2.3 General model of reaction-diffusion systems

Once seen the mathematical models of chemical reactions and diffusion, we can give a model for the general reaction-diffusion systems. This model consists of a system of partial differential equations which depends on time and space. Specifically, it is given by the expression:

$$\frac{\partial u}{\partial t} = g(u) + D\Delta u$$

where:

- $x \in \mathbb{R}^n$.
- $u(t, x) = (u_1(t, x), \dots, u_m(t, x))$ shows the concentration of each substance.
- $g(u) = (g_1(u_1, \dots, u_m), \dots, g_m(u_1, \dots, u_m))$ is the reaction rate vector.
- D is the diffusion matrix.
- Δ is the Laplacian operator.

In other words, as we have studied in 2.1 and 2.2, $g(u)$ corresponds to the reactive part of the system and, as we have seen in 2.3, $D\Delta u$, to the diffusive one. Particularly, as we have studied in 2.1, each $g_i(u)$ is a nonlinear function which shows the substances concentration which depends on time.

Futhermore, in order to simplify the system, we consider that the diffusion coefficients D_i are constant.

If we want to study this systems in particular domains we can introduce a new parameter γ , which is proportional to the domain size (scale) and we can rewrite our initial system as follows

$$\frac{\partial u}{\partial t} = \gamma f(u) + D\Delta u.$$

That is, we will consider functions $g(u)$ adapted to the dimensions of the domains in which take place the reaction-diffusion systems that we want to study.

3 Turing instability. Activator-inhibitor systems

Alan Turing in *The Chemical Basis of Morphogenesis* in 1952 explained how patterns emerge in living forms. He proposed a morphogenetic mechanism which consists in two simultaneous processes: kinetic reactions and diffusion, which affect the concentration of specific substances called morphogens.

Once given this system (which we have modeled in 2.3), he stated that the pattern is originated from a state in equilibrium and constant which is stable to temporal perturbations, but becomes unstable due to spatial perturbations. Turing proposed, as well, that this instability is due to diffusion.

Therefore, his theory can be given by the following conditions:

(i) In the absence of diffusion, that is, when the system is only given by the chemical kinetics, the state must be stable to small perturbations. We call this equilibrium *base state* and we denote it by u_b .

(ii) When diffusion takes place some oscillatory modes of the system must be unstable to small spatial perturbations but it is time-independent. The instability caused by diffusion is called *diffusion driven instability*.

We have another condition in case of two morphogens:

(iii) The diffusion coefficient of the two substances must differ substantially.

We observe that these conditions are heuristic. Therefore, our goal in this chapter is to obtain mathematical ones which are equivalent to the ones we have already given, that is, conditions that our reaction-diffusion equations must satisfy in order to pattern formation can occur. However, it is complicated to deal with this kind of systems directly and thus we will have to do a previous linear analysis.

The instability that leads to pattern formation is also called *Turing instability*. We will see in the last section that this kind of instability in case of two morphogens leads to an activator-inhibitor system, that is, one morphogen will induce the production of substance and, the other one, the loss.

3.1 Linear stability: basics

This section is a conceptual framework of the general linear analysis that we will apply in our study of the appearance of Turing instability.

We have seen in the second chapter that we will work with systems with dynamics described by partial differential equations of the form

$$\frac{\partial u}{\partial t} = g(u) + D\Delta u,$$

where $g(u) = \gamma f(u)$.

Apart from γ and the coefficients of matrix D , this system can show more parameters in the expression of function f . Then, our linear study will be based on

taking one of these parameters as a control parameter p , which we can tune and obtain different asymptotic behavior of the system.

We will consider that our domain is $[0, A_1] \times \cdots \times [0, A_n]$ and, in addition, we will take periodic boundary conditions.

We start remembering that the first Turing's condition was that, in the absence of diffusion, the state is stable in a base solution u_b , which means:

(i) $g(u_b) = 0$.

(ii) $\mathcal{D}g(u_b)$ has all eigenvalues with negative real part.

We now can write the general solution of the system as:

$$u(t, x) = u_b + v(t, x),$$

where $v(t, x)$ is the correction. Hence, we can rewrite our system of differential equations in the form:

$$\frac{\partial v}{\partial t}(t, x) = g(u_b + v(t, x)) + D\Delta v(t, x).$$

This is the moment when we do the linearization around u_b . Performing it we obtain:

$$\frac{\partial v}{\partial t}(t, x) \approx \mathcal{D}g(u_b)v(t, x) + D\Delta v(t, x). \quad (3.1)$$

Using Fourier series, we can express the solutions of this system with periodic boundary conditions as:

$$v(t, x) = \sum_{q \in \mathbb{Z}^n} v_q(t) e^{2\pi i q \frac{x}{A}},$$

with $\frac{x}{A} = (\frac{x_1}{A_1}, \dots, \frac{x_n}{A_n})$, and so $e^{2\pi i q \frac{x}{A}} = \prod_{j=1}^n e^{2\pi i q_j \frac{x_j}{A_j}}$.

This technique is called *Spectral method* and we have learnt it in the course of *Partial differential equations*.

Substituting the last expression in (3.1):

$$\sum_{q \in \mathbb{Z}^n} \dot{v}_q(t) e^{2\pi i q \frac{x}{A}} = \sum_{q \in \mathbb{Z}^n} \mathcal{D}g(u_b) v_q(t) e^{2\pi i q \frac{x}{A}} - D \sum_{q \in \mathbb{Z}^n} 4\pi^2 \left(\sum_{j=1}^n \frac{q_j^2}{A_j^2} \right) v_q(t) e^{2\pi i q \frac{x}{A}}.$$

This implies that, for each q , we have:

$$\dot{v}_q(t) = \mathcal{D}g(u_b) v_q(t) - D4\pi^2 \left(\frac{q}{A} \right)^2 v_q(t) = (\mathcal{D}g(u_b) - D4\pi^2 \left(\frac{q}{A} \right)^2) v_q(t),$$

where $\left(\frac{q}{A} \right)^2 = \frac{q_1^2}{A_1^2} + \cdots + \frac{q_n^2}{A_n^2}$.

The stability of the system can be studied now from the eigenvalues σ_q of the $n \times n$ matrix $Dg(u_b) - D4\pi^2(\frac{q}{A})^2$.

We have learnt in the courses of *Ordinary differential equations* and *Mathematical models and dynamical systems* that the eigenvalues with $Re(\sigma_q) > 0$ determine instability, while the ones with $Re(\sigma_q) < 0$ determine stability.

Therefore, while $Re(\sigma_q) < 0$ the system is stable. However, when we are tuning control parameter p it could happen that, for a certain value $p = p_c$, $Re(\sigma_q)$ may cross 0 and changes its sign to positive for some values of q . This phenomenon is called *bifurcation*.

We now can introduce two important values of q which have important roles in pattern formation:

- The first value q from which $Re(\sigma(q))$ becomes positive is called *critical wave vector* and we denote it q_c .

- We call q_m the vector wave from which $Re(\sigma(q))$ has the maximum value (not necessarily q_c).

The imaginary part of the growth rate evaluated at the critical wave number determines a critical frequency ω_c given by

$$\omega_c = -\text{Im}(\sigma_{q_c})$$

that defines a characteristic oscillatory time scale for the growing perturbation. We can distinguish two cases:

(a) If $\omega_c = 0$, the instability is towards a stationary state and the pattern will only grow in time. This type of instability is called *stationary* and was the one studied by Turing, and the one which we will study along this project.

(b) If $\omega_c \neq 0$, the instability is oscillatory and the pattern will be modeled by oscillations of period $T = \frac{2\pi}{\omega_c}$. This type of instability is called *Hopf bifurcation*.

3.2 Turing instability

Using the previous conceptual framework of linear analysis, we now give in this part of the chapter the mathematical conditions equivalent to the ones that we have mentioned in the introduction of the chapter, that is, the mathematical conditions needed for pattern formation.

First of all, we introduce some new notation for this section:

$$\begin{cases} M = \mathcal{D}g(u_b) \\ M_q = \mathcal{D}(g(u_b)) - D4\pi^2(\frac{q}{A})^2 = M - D4\pi^2(\frac{q}{A})^2. \end{cases}$$

Furthermore, we denote $m_{ij} = \frac{\partial f_i}{\partial x_j}|_{u_b}$. This implies that the coefficients of matrix M are γm_{ij} for $i = 1, \dots, m$ and $j = 1, \dots, n$.

We now remember that the first hypothesis that Turing stated is that in the

absence of diffusion there is an equilibrium state, called u_b , which we have seen in section 3.1 that satisfies:

(i) $g(u_b) = 0$.

(ii) M has all its eigenvalues σ with negative real part.

These eigenvalues σ are the solution of

$$\det(M - \sigma I) = 0.$$

From now on, in order to simplify, we will consider the two dimensional case with two morphogens.

Two dimensional case with two morphogens

Returning to our previous study, in this case we have the following expression of the above determinant:

$$\det(M - \sigma I) = \sigma^2 - (\text{tr}M)\sigma + \det M.$$

Equating this expression to zero we obtain that these eigenvalues are given by:

$$\sigma = \frac{1}{2}\text{tr}M \pm \frac{1}{2}\sqrt{(\text{tr}M)^2 - 4\det M}.$$

From this expression we can deduce this following lemma (which is justified below in the Figure 1):

Lemma 3.2.0.1. *The real parts of the eigenvalues σ are negative, that is, the stability occurs, when the trace of the matrix is negative and the determinant positive.*

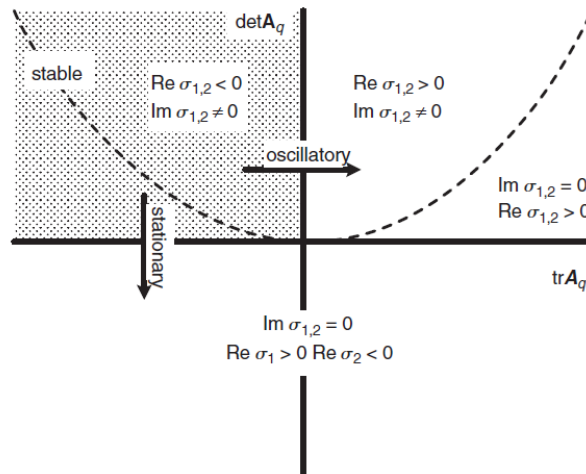


Figure 1: Stability regions in the $\text{tr}M$ - $\det M$ plane, from source: [2]. Particularly, A_q corresponds to our matrix M .

Therefore, applying this lemma to matrix M we obtain two mathematical conditions equivalent to the first Alan Turing's hypothesis:

$$\begin{cases} \operatorname{tr}(M) = \gamma m_{11} + \gamma m_{22} < 0 \\ \det(M) = \gamma^2 m_{11} m_{22} - \gamma^2 m_{12} m_{21} > 0 \end{cases}$$

We now focus on the second condition he gave. This implies that we return to the case with diffusion.

From now on, if coefficient D_1 satisfies $D_1 \neq 0$, matrix D will be expressed in the equivalent form:

$$D = \begin{pmatrix} 1 & 0 \\ 0 & d \end{pmatrix},$$

where $d = \frac{D_2}{D_1}$ is the ratio of diffusion coefficients. We write D in this form in order to have a new parameter d which we will be able to tune and obtain bifurcations.

Now, to study the stability in the presence of diffusion, we have to observe the eigenvalues of matrix M_q .

However, from Lemma 3.2.0.1, it is equivalent to observe the signs of

$$\begin{cases} \operatorname{tr}(M_q) = \gamma m_{11} + \gamma m_{22} - (1+d)4\pi^2 \left| \frac{q}{A} \right|^2 \\ \det(M_q) = d(4\pi^2)^2 \left| \frac{q}{A} \right|^4 - (\gamma d m_{11} + \gamma m_{22}) 4\pi^2 \left| \frac{q}{A} \right|^2 + \gamma^2 (m_{11} m_{22} - m_{12} m_{21}) \end{cases}$$

Comparing this trace with the trace of M , we obtain

$$\operatorname{tr}(M_q) = \gamma m_{11} + \gamma m_{22} - (1+d)4\pi^2 \left| \frac{q}{A} \right|^2 < \gamma m_{11} + \gamma m_{22} < 0$$

Thus, from the Lemma 3.2.0.1, we can conclude that the instability produced by diffusion is caused by a change of sign of the determinant of M_q .

If we observe this determinant we can see that it is a parabola in the quantity $k = 4\pi^2 \left| \frac{q}{A} \right|^2$. One condition for the instability may be asking when the minimum value of the parabola becomes negative. Let's find this value:

$$\det M_q(k) = dk^2 - \gamma(dm_{11} + m_{22})k + \gamma^2(m_{11}m_{22} + m_{21}m_{12}).$$

$$\frac{d(\det(M_q))}{dk} = 2dk - (d\gamma m_{11} + \gamma m_{22})$$

$$\frac{d(\det(M_q))}{dk} = 0 \iff k_m = \frac{d\gamma m_{11} + \gamma m_{22}}{2d} (\iff dm_{11} + m_{22} > 0)$$

$$\det(M_{q_m}) = \gamma^2(m_{11}m_{22} - m_{12}m_{21}) - \frac{(\gamma d m_{11} + \gamma m_{22})^2}{4d}$$

This expression becomes negative when

$$dm_{11} + m_{22} > 2\sqrt{d(m_{11}m_{22} - m_{12}m_{21})}$$

(it is well defined because we have seen that $dm_{11} + m_{22} > 0$)

Therefore, gathering all the information we have developed along this subsection, we can conclude that the two first heuristic conditions that Alan Turing gave for pattern formation are equivalent to the mathematical conditions:

$$(i) \quad m_{11} + m_{22} < 0$$

$$(ii) \quad m_{11}m_{22} - m_{12}m_{21} > 0$$

$$(iii) \quad dm_{11} + m_{22} > 0$$

$$(iv) \quad dm_{11} + m_{22} > 2\sqrt{d(m_{11}m_{22} - m_{12}m_{21})}$$

From (i) and (iii) we obtain that m_{11} and m_{22} must have opposite signs. These two coefficients have to do with the reactive part of the reaction-diffusion system. Besides, we remember that terms m_{ij} have appeared when we have linearized our equations around u_b . Particularly, if we focus on the reactive part of the system, this linearization is:

$$\frac{\partial u_1}{\partial t} \approx u_{1b} + m_{11}(u_1 - u_{1b}) + m_{12}(u_2 - u_{2b}). \quad (3.2)$$

$$\frac{\partial u_2}{\partial t} \approx u_{2b} + m_{21}(u_1 - u_{1b}) + m_{22}(u_2 - u_{2b}). \quad (3.3)$$

Therefore, if we consider, for instance, $m_{11} > 0$ and $m_{22} < 0$, it implies that an increase of u_1 over the uniform state induces its own productions while u_2 inhibits its own production. Thus, we can call u_1 activator and u_2 inhibitor.

Now, from (ii) we obtain that m_{12} and m_{21} have opposite signs, as well. Hence, observing the linearization given by (3.2) and (3.3), there are essentially two kinds of reaction-diffusion models capable of exhibiting diffusion-driven instability with matrix M taking one of the following forms:

$$\begin{bmatrix} + & - \\ + & - \end{bmatrix} \text{ pure activator-inhibitor model,}$$

$$\begin{bmatrix} + & + \\ - & - \end{bmatrix} \text{ cross activator-inhibitor model.}$$

In addition, we remember that the third heuristic condition which Alan Turing gave for pattern formation was that the diffusion coefficients of the two morphogens must differ substantially. Hence, from (iii) and (iv) we can write mathematically this condition as:

$$d \gg \frac{-m_{22}}{m_{11}} > 1.$$

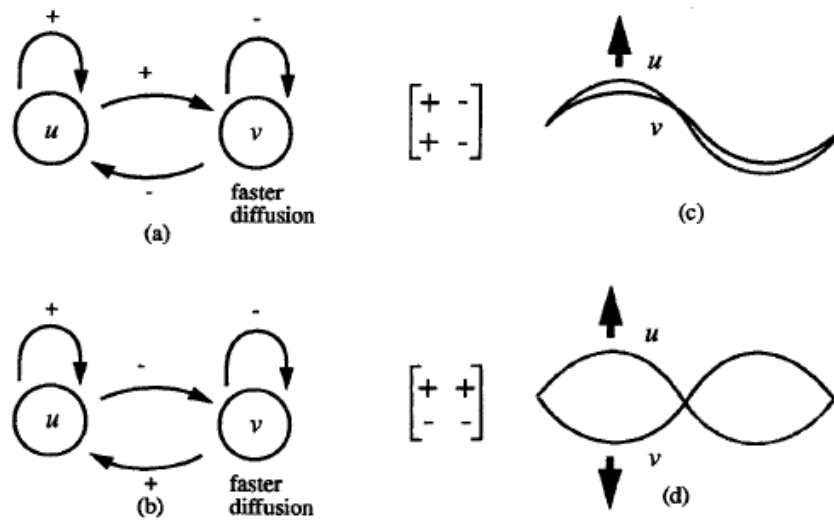


Figure 2: Schematic representation of the two kinds of activator-inhibitor systems, (a) and (c) correspond to the pure one; (b) and (d) to the cross one from the source: [6]. Particularly, u corresponds to our u_1 and, v , to u_2 .

Particularly, this expression means that the inhibitor diffuses much faster than the activator. Gierer and Meinhard in 1972 called this phenomenon *local activation with long-range inhibition*. The range is the mean distance a molecule can travel between its production and disappearance.

Finally, we discuss the two cases mentioned above separately:

(a) *Pure activator inhibitor model*: u_1 is the activator, which is also self-activating, while the inhibitor, u_2 , inhibits not only u_1 , but also itself. Thus, the two species are at high or low density in the same region as the pattern grows.

(b) *Cross activator inhibitor model*: u_2 is the activator but also self-inhibiting and diffuses more quickly. Therefore, u_1 is at high density where u_2 is low and vice versa (“predator-prey model”).

4 Pattern formation on animal skin: The importance of the geometry of the body

Once we have studied how patterns are formed from instabilities performed in a reaction diffusion system initially in equilibrium, our goal in this chapter is to study how patterns take place on animal skin and to see how their geometry (thickness, curvature...) affects the resulting pattern.

First of all, we will introduce the most accepted theory for this phenomenon, which is the one introduced by the mathematician James D. Murray in 1989. He studied how patterns are formed in animals and how their thickness affects their design.

To show the importance of thickness, he studied legs and tails and considered their skin as a cylinder surface. We can obtain this surface taking periodic boundary conditions in two parallel sides of the rectangular domain $[0, A] \times [0, B]$. We observe that this domain and these boundary conditions are similar to the ones that we have considered along all this project, and so we have been able to use some previous information. Therefore, studying the effect of thickness is equivalent to do the same with the narrowness of the rectangle.

However, animals are three dimensional. This is the reason why we have made a short discussion of the importance of the curvature in a real torus. Particularly, we will see that there is a relation between studying the thickness in a flat torus and doing the same with the curvature in a torus.

4.1 Murray's theory

James D. Murray stated in 1989 that pattern formation in animals takes place during the *morphogenesis*, which we have defined in the introduction of this project. Besides, he affirmed that in this process occur some perturbations in a reaction-diffusion system which affects the concentration of substances called *morphogens*, which can be hormones, genes...(they are not specified). Finally, he argued that these instabilities are reflected on animal skin due to the subsequent differentiation of the cells to produce melanin, which are the cells that affect animal pigmentation.

Once we have shown in what is based this theory, we can introduce its hypothesis:

(i) The unstable mode that dominates the system and the spatial coat pattern have the same spatial structure.

(ii) The unstable mode is determined by the parameter pair (d, γ) .

(iii) Spatial patterns occur only if d is sufficiently large, and it leads to an unstable eigenvalue range.

(iv) Narrower geometry of the rectangle $[0, A] \times [0, B]$ is easier to produce stripes.

The justification of this last hypothesis is based on the fact that 1D domains can only produce striped patterns and 2D domains can lead not only to striped patterns but to spotted ones. Besides, this statement has an important role in the

justification of some applications in animals body that we will develop at the end of the subsection 4.2.2.

4.2 Effect of the geometry

Apart from the geometry of the domain, we have mentioned the importance of two of the reaction-diffusion system parameters (d and γ) to the final pattern. We also have said that the first hypothesis of Murray's theory states that the unstable mode which dominates the system has the same structure than the spatial coat pattern. Therefore, study the importance of geometry, scale or diffusion is equivalent to see how giving different values to the parameters A , B , γ and d determine what mode will dominate the system. This problem is called *mode selection*.

In the third chapter we have seen that the critical wavenumber k_c of the 2D reaction-diffusion system with two morphogens is given by:

$$k_c = \frac{dm_{11} + m_{22}}{2d_c}.$$

Furthermore, we remember that we have considered:

$$k = 4\pi^2 \left| \frac{q}{A} \right|^2.$$

Particularly, since we have considered $q \in \mathbb{Z}^2$, if we take $q = (m, n)$, we can write $k = 4\pi^2 \left(\frac{m^2}{A^2} + \frac{n^2}{B^2} \right)$.

Hence, for $d > d_c$ there is a range of wavenumbers $k_1^2 < k^2 = 4\pi^2 \left(\frac{m^2}{A^2} + \frac{n^2}{B^2} \right) < k_2^2$. This range is obtained from the zeros from the zeros k_1^2 and k_2^2 of $\det(M_q)(k)$, which are:

$$k_1^2 = \frac{\gamma}{2d} [(dm_{11} + m_{22}) - \{(dm_{11} + m_{22})^2 - 4d(m_{11}m_{22} - m_{12}m_{21})\}^{1/2}], \quad (4.1)$$

$$k_2^2 = \frac{\gamma}{2d} [(dm_{11} + m_{22}) + \{(dm_{11} + m_{22})^2 - 4d(m_{11}m_{22} - m_{12}m_{21})\}^{1/2}]. \quad (4.2)$$

Therefore, we have obtained a range of unstable modes that depends on some parameters which we can tune in order to obtain different patterns. However, owing to the fact that our goal is to study the geometry, we will only study γ , A and B .

4.2.1 Effect of the scale

In this subsection we focus on the scale of the animal body (quantified by the parameter γ), that is, how big or small it is.

From the unstable range exposed above, we observe that if the domain size is too small, that is γ small enough, even the lowest nonzero mode lie outside the unstable range. Thus, no spatial pattern can be generated. This implies that, in general, *very small animals can be expected to be uniform in color*. Examples of this statement are rats, squirrels and small dogs.

Furthermore, as the size of the animal increases, γ reaches some bifurcation values and different patterns take place. However, for very large domains the morphogen concentration distribution is again almost uniform. Thus, *the structure of the pattern of a big animal is very fine*. Examples of this case are elephants and bears.

Therefore, as a conclusion, we can confirm that *medium sized animals are the ones that can show more range of patterns*. We can see a lot of examples in nature, like zebras, giraffes and big cats.

4.2.2 Importance of the thickness

To study how thickness affects the mode that will dominate the system, we have to focus on the geometry of the domain. We remember that we have considered a flat torus, which is represented in a rectangle $[0, A] \times [0, B]$. Furthermore, in the introduction of the chapter it has been said that the study of the thickness is equivalent to observe the narrowness of this rectangle.

Hence, our goal in this part of the project is to study how the narrowness of $[0, A] \times [0, B]$ leads to the appearance of some different patterns. Particularly, we consider B fixed and we tune A .

We remember from the beginning of this section that we have the following range of wavenumbers:

$$k_1^2 < k = 4\pi^2\left(\frac{m^2}{A^2} + \frac{n^2}{B^2}\right) < k_2^2,$$

with k_1 and k_2 given by the expressions (4.1) and (4.2) respectively. Therefore, from this range, we can consider two different cases according to the value of A :

(i) A is small or too large. This implies that there is no mode m that belongs to the unstable range. Therefore, the unstable range only include modes n and thus this case is equivalent to the 1-dimensional one which is the one that leads to the appearance of striped patterns.

(ii) A is sufficiently large. In this case there exist modes n belonging to the unstable range. Hence, we are in the two-dimensional case and we can obtain a pattern which can be spotted or striped.

We now can observe that this theory justify some phenomenons that we see in some mammals body.

On the one hand, tails and legs are usually very thin and so they can only have striped patterns. However, there exists legs that are big enough at one end and become gradually thinner and thinner. In this case, we can see a gradation from a 2D pattern at the thick end to a 1D pattern at the thin end.

On the other hand, animal body is usually thick enough but not too much, so that, their skin is considered a 2D domain.

Therefore, we can conclude:

- There can be animals all striped.

- There can be animals with spotted body and striped tail and legs.
- There can be animals with spotted body and a gradation of spots to stripes in their legs and tail.
- There is no animal with striped body and spotted legs and tail.

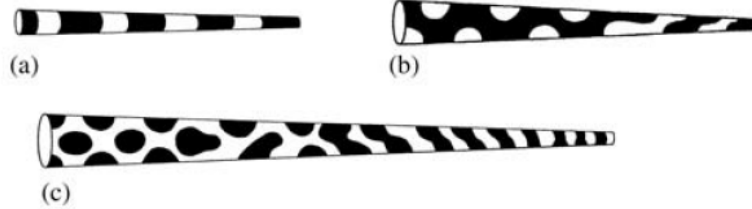


Figure 3: (a) represents a very thin tale which only can have a striped patterns. Conversely, (b) and (c) show the gradation of a spotted pattern (in the thick end) to a striped one (thin end). Source: [6].

4.2.3 Short discussion about the curvature of the torus

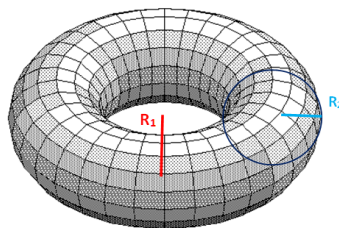
Along all this project we have considered rectangles $[0, A] \times [0, B]$ and periodic boundary conditions. This is the reason why we have been able to study above the effect of the narrowness rigorously using previous information. However, we cannot use this information for the curvature of the torus since its Laplacian operator of the scalar function u is not the same.

We start remembering the parametrization of this surface:

$$\chi(\theta_1, \theta_2) = \begin{pmatrix} (R_1 + R_2 \cos \theta_2) \cos \theta_1 \\ (R_1 + R_2 \cos \theta_2) \sin \theta_1 \\ R_2 \sin \theta_2 \end{pmatrix}$$

where:

- R_1 is the distance between the centre of the torus and the centre of the conduct.
- R_2 is the radius of the conduct.
- θ_1 is the angle along R_1 .
- θ_2 is the angle along R_2 .



With this notation, the Laplacian operator of u in the torus is given by the expression:

$$\Delta u = \frac{1}{(R_1 + R_2 \cos(\theta_2))^2} u_{\theta_1 \theta_1} - \frac{\sin \theta_2}{R_2(R_1 + R_2 \cos \theta_2)} u_{\theta_2} + \frac{1}{R_2^2} u_{\theta_2 \theta_2}. \quad (4.3)$$

This result is justified on Appendix A.

Hence, in order to do an accurate study of the curvature we should have repeated the same process for the corresponding system. However, since we also want to do some simulations and a numerical study of the system, we only have made a comparison between the study of thickness and curvature and obtained some conclusions. To do it, we can compare Laplacian operation of u in the torus (4.3) with the one in the flat torus, which has the expression:

$$\Delta_* u = \frac{4\pi^2}{A^2} \frac{\partial^2 u}{\partial x^2} + \frac{4\pi^2}{B^2} \frac{\partial^2 u}{\partial y^2}.$$

Observing these two formulas one can easily observe that, their second derivatives are multiplied by weights that are inverse to the parameters we want to study (A, B, R_1 and R_2). Hence, we can think that the study of thickness from tuning A and B is similar to the study of curvature from tuning R_1 and R_2 .

We have seen that as thick as a domain is, it is more likely to produce 2D patterns. Therefore, from the above reasoning, we can argue that more curved domains tend more to 2D patterns than less curved ones. Particularly, if we consider parameter R_2 fixed, we can say that if R_1 has a value close to R_2 we will have more 1D patterns and, conversely, if $R_2 \ll R_1$ we will have probably more 2D patterns.

However, to give a stronger justification of the results, we have made some simulations in Chapter 5 tuning R_1 as we have explained in which we can see that this forecast of the effect of the curvature is satisfied.

5 Simulation results

At this time of the project we have given a mathematical model for reaction-diffusion systems, studied Turing instability under which pattern formation takes place and explained how the geometry of animal body affects the appearance of stripes or spots.

Our goal now is thus gather all this information in order to do simulations of a specific reaction-diffusion system, the Schakenberg's model, from which we will tune indendently the parameters A , B and γ to show the importance of the geometry of the body that we have explained in the previous chapter.

Therefore, our first step in this chapter is to obtain the necessary conditions for Turing instability and the range of unstable wavenumbers in the particular model we have chosen. Then, we introduce the numerical method we will use to do our simulations, which is a explicit finite difference method. Finally, we give the results of these simulations and hence we show graphically the effect of γ , A and B (and R_1 and R_2 as well) in three different domains: 1D, flat torus and torus.

5.1 Conditions for pattern formation in Schakenberg's model

We have mentioned in the beginning of the chapter that we will consider in our simulations the Schakenberg's model, which is:

$$\begin{cases} \frac{\partial u_1}{\partial t} = \gamma f_1 + \Delta u_1 \\ \frac{\partial u_2}{\partial t} = \gamma f_2 + d\Delta u_2, \end{cases} \quad (5.1)$$

where:

$$\begin{cases} f_1 = a - u_1 + u_1^2 u_2 \\ f_2 = b - u_1^2 u_2. \end{cases}$$

However, before designing a program which implements it, we need to study under which conditions of this system pattern formation can take place. We now remember that we have to start finding the base state which appears in the absence of diffusion. Therefore, this equilibrium point is found imposing the conditions $f_1 = 0$ and $f_2 = 0$, from which we obtain:

$$u_b = \left(a + b, \frac{b}{(a + b)^2} \right).$$

Now we have to study under which conditions spatial instabilities due to diffusion take place. However, we first have to compute the first partial derivatives of the kinetic functions f_1 and f_2 , that is, the coefficients m_{ij} from the section 3.2, which are:

$$m_{11} = \frac{b-a}{a+b}, \quad m_{21} = \frac{-2b}{a+b},$$

$$m_{12} = (a+b)^2, \quad m_{22} = -(a+b)^2.$$

Therefore, from the section 3.2, we remember that in order to Turing instability occur in this system (and so pattern formation takes place), we need the conditions:

$$(i) \ m_{11} + m_{22} < 0 \implies 0 < b-a < (a+b)^3$$

$$(ii) \ m_{11}m_{22} - m_{12}m_{21} > 0 \implies (a+b)^2 > 0$$

$$(iii) \ dm_{11} + m_{22} > 0 \implies d(b-a) > (a+b)^2 > 0$$

$$(iv) \ dm_{11} + m_{22} > 2\sqrt{d(m_{11}m_{22} - m_{12}m_{21})} \implies [d(b-a) - (a+b)^3]^2 > 4d(a+b)^3.$$

However, in addition to these conditions, we have to choose values of γ , A and B that lead to the existence of modes which belong to the particular unstable range:

$$k_1^2 < k_2 = 4\pi\left(\frac{m^2}{A} + \frac{n^2}{B}\right) < k_2^2,$$

with:

$$k_1 = \gamma \frac{d(b-a) - (a+b)^3 - \sqrt{[d(b-a) - (a+b)^3]^2 - 4d(a+b)^4}}{2d(a+b)}$$

$$k_2 = \gamma \frac{d(b-a) - (a+b)^3 + \sqrt{[d(b-a) - (a+b)^3]^2 - 4d(a+b)^4}}{2d(a+b)}.$$

5.2 A finite difference method for reaction-diffusion systems

Once we have developed all the theory we wanted to study, it is the moment to obtain some numerical results to show it. Hence, this section consists of an explanation of a numerical method, the so-called *finite difference method*, from which we will be able to obtain solutions of reaction-diffusion systems.

Along all this project we have considered a rectangle $[0, A] \times [0, B]$ as a domain and periodic boundary conditions. Therefore, we will start developing in detail this case. However, we want to do, as well, 1D simulations and on a torus. Hence, we will include two subsections giving this method in these two domains using the previous information of the flat torus case.

Therefore, our goals are:

(i) To discrete our spatial and temporal domains.

(ii) To use the previous step in order to obtain discrete information from the boundary conditions that are necessary to approximate the solutions of our system.

- (iii) To approximate, independently, spatial and temporal derivatives.
- (iv) To explain the explicit scheme, which we will use to do the simulations.

5.2.1 Spatial and temporal approximation

Let's explain, firstly, spatial approximation. We remember that our domain is a rectangle $[0, A] \times [0, B]$.

We start dividing horizontal and vertical sides of the domain in M_x and M_y intervals with length $h_x = \frac{A}{M_x}$ and $h_y = \frac{B}{M_y}$, respectively. Hence, we obtain a mesh with $(M_x + 1) \times (M_y + 1)$ nodes. We identify each node with the index (i, j) , where i shows the horizontal position and j , the vertical.

Similarly, we divide total time T in N intervals with length $\Delta t = \frac{T}{N}$ enumerated by the index n .

5.2.2 Boundary conditions

We remember that our boundary conditions are periodic. It implies that the right side nodes of the rectangular domain correspond exactly to the left side ones, and the top nodes correspond to the bottom ones. Particularly, we can write:

$$\begin{aligned} u_{0,j} &= u_{M_x,j}, & j &= 1, \dots, M_y - 1 \\ u_{i,0} &= u_{i,M_y}, & i &= 1, \dots, M_x - 1. \end{aligned}$$

Besides, we can generalize this explanation and say that periodic boundary conditions mean that the solution in one node (i, j) is the same that the one in the node $(M_x + i, M_y + j)$.

The choice of boundary conditions has a huge relevance when we apply the finite difference method. We will see in the section below that in order to approximate spatial derivatives in one node (i, j) of the rectangular domain we need the information of its four neighbours $(i - 1, j)$, $(i + 1, j)$, $(i, j - 1)$ and $(i, j + 1)$. However, boundary nodes only have two or three. This is the moment when boundary conditions take importance, we can consider that the missing nodes are the ones displaced $M_x + i$ or $M_y + j$ (or $i - M_x$ or $j - M_y$) positions. For instance, solutions in the nodes (i, M_y) have the neighbours u_{i,M_y-1} and $u_{i,1}$.

5.2.3 Derivatives approximation

In this subsection we approximate spatial and temporal derivatives independently and then, in the following section, we gather them together to get different discrete schemes which give us an approximate solution of our systems.

Particularly, we have obtained the derivatives approximations using finite differences, the scheme of which is introduced in the course of *Numerical methods I*.

Temporal derivative approximation

In order to give an approximation of u_t in one point (x_i, y_j, t_n) , we can perform a first order finite differences formula:

$$u_t(t_n, x_i, y_j) \approx \frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t}.$$

Spatial derivative approximation

We remember that in reaction-diffusion models the spatial derivatives appear in the Laplacian operator. Thus, we have to approximate second derivatives u_{xx} and u_{yy} in a certain node. This approximation of the second derivative is usually applying the central differences scheme:

$$u_{xx}(t_n, x_i, y_j) \approx \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{h_x^2},$$

Performing the same process for u_{yy} we obtain:

$$u_{yy}(t_n, x_i, y_j) \approx \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{h_y^2}.$$

Finally, we can write the Laplacian operator in the following form:

$$\Delta u = u_{xx} + u_{yy} = \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{h_x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{h_y^2}.$$

with $i = 0, \dots, M_x - 1$ and $j = 0, \dots, M_y - 1$.

5.2.4 Explicit method for reaction-diffusion equations

There exist some schemes which enable us to obtain numerical solutions of the discrete system that we have built before. These schemes are classified according to the convergence order of their temporal scheme and the time when Laplacian operator is evaluated. Particularly, we will explain explicit method.

In this scheme, the temporal derivative is given by first order approximation and the right part of the equation is evaluated at time n .

Thus, we write:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} = g(u_{i,j}^n) + D \left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{h_x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{h_y^2} \right).$$

Applying some algebra on this expression we obtain:

$$u_{i,j}^{n+1} = u_{i,j}^n + \Delta t g(u_{i,j}^n) + \Delta t D \left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{h_x^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{h_y^2} \right).$$

where $n = 0, 1, \dots, N - 1$, $i = 0, \dots, M_x - 1$ and $j = 0, \dots, M_y - 1$.

Therefore, from this last expression we can observe that to obtain the solution at a certain time we only have to substitute the previous information. That is, to find the solution at time $n = 1$ we substitute in the last equality the values of u and g at time $n = 0$ that we have from the initial conditions. Thus, performing some iterations, we can get the solution at time $n + 1$ substituting the information we have obtained at time n .

This method is simple but its stability is limited, that is, we have to choose Δt with h^2 -order to have some stability. It means that a detailed solution requires smaller steps, and so, more iterations.

Explicit finite difference method stability is explained and justified in the Appendix B.

Explicit method for the one dimensional case

In this case the domain is an interval $[0, A]$ with periodic boundary conditions, as well. Therefore, the Laplacian of u for this case is:

$$\Delta u = u_{xx}.$$

Hence, the explicit finite difference method for the one dimensional case is:

$$u_i^{n+1} = u_i^n + \Delta t g(u_i^n) + \Delta t D \left(\frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,n}^n}{h_x^2} \right).$$

Explicit method with a torus as a spatial domain

In order to be able to give the explicit finite difference method for the torus, we must have in mind its parametrization, which we have introduced in section 4.4. From this expression, we obtain that the explicit scheme on the torus is quite similar to the flat torus one, but there are two differences.

On the one hand, the spatial domain that we consider is the square $[0, 2\pi] \times [0, 2\pi]$, and the spatial variables are θ_1 and θ_2 instead of x and y . Therefore, spatial steps are:

$$h_{\theta_1} = h_1 = \frac{2\pi}{M_1}$$

$$h_{\theta_2} = h_2 = \frac{2\pi}{M_2},$$

where M_1 and M_2 correspond to the number of horizontal and vertical nodes of the mesh, respectively.

On the other hand, we remember from Chapter 4 that we can write the Laplacian of u on the torus with angles θ_1 and θ_2 by the expression:

$$\Delta u = \frac{1}{(R_1 + R_2 \cos(\theta_2))^2} u_{\theta_1 \theta_1} - \frac{\sin \theta_2}{R_2(R_1 + R_2 \cos \theta_2)} u_{\theta_2} + \frac{1}{R_2^2} u_{\theta_2 \theta_2}.$$

Hence, the discretization of the Laplacian changes in that second derivatives are multiplied by two new factors and it has appeared as well a first derivative, which can be approximated by the central differences scheme:

$$u_{\theta_2}(\theta_{1i}, \theta_{2i}, t_n) \approx \frac{u_{i,j+1}^n - u_{i,j-1}^n}{2h_2}.$$

Therefore, the explicit scheme for this case is:

$$u_{i,j}^{n+1} = u_{i,j}^n + \Delta t g(u_{i,j}^n) + D \left(\frac{1}{(R_1 + R_2 \cos(\theta_{2j}))^2} \frac{\delta_1^2}{h_1^2} + \frac{1}{R_2^2} \frac{\delta_2^2}{h_2^2} - \frac{\sin(\theta_{2j})}{R_2(R_1 + R_2 \cos(\theta_{2j}))} \frac{u_{i,j+1}^n - u_{i,j-1}^n}{2h_2} \right),$$

where:

$$\begin{cases} \delta_1^2 = u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \\ \delta_2^2 = u_{i,j+1} - 2u_{i,j} + u_{i,j-1} \end{cases}$$

5.3 Simulations

We have arrived at the final part of the project, which is the one that shows graphically all the contents developed along it. All the simulations have been done executing three programs in C designed for this project, which are explained and specified in Appendix C, and using *gnuplot* (for the cases 1D and flat torus) and *ParaView* (for the torus case).

In all the simulations we have performed Schakenberg's model (5.1) and considered fixed a , b , d , total time T and temporal step Δt . Particularly, we have taken:

- $a = 0.1$.
- $b = 0.9$.
- $d = 10$.
- $T = 5$.
- $\Delta t = 0.00001$.

These values of the parameters a , b and d are the most frequently used for this kind of simulations since they satisfy, for appropriate values of A , B and γ , Turing's conditions and lead to the existence of modes which belong to the unstable range. Furthermore, it has been chosen this temporal step in order to have stability in our explicit scheme.

Conversely, we have tuned the geometrical parameters independently, according to which characteristic we want to focus. Specifically, γ (scale) has been tuned in 1D simulations, A (thickness) in flat torus ones and R_1 (curvature) on the torus.

One dimensional simulations

As we have mentioned before, we use this case to see the importance of the scale, quantified by γ . In order to show it, we have executed our first program considering fixed $A = 2$, and tuning γ with values: $\gamma = 114$, $\gamma = 1000$, $\gamma = 10000$ and $\gamma = 50000$.

From this pictures we can see how the number of modes grows more and more while γ increases but, for too large values ($\gamma = 50000$), the concentration is almost uniform, that is, the system has reached a new equilibrium and no pattern takes place.

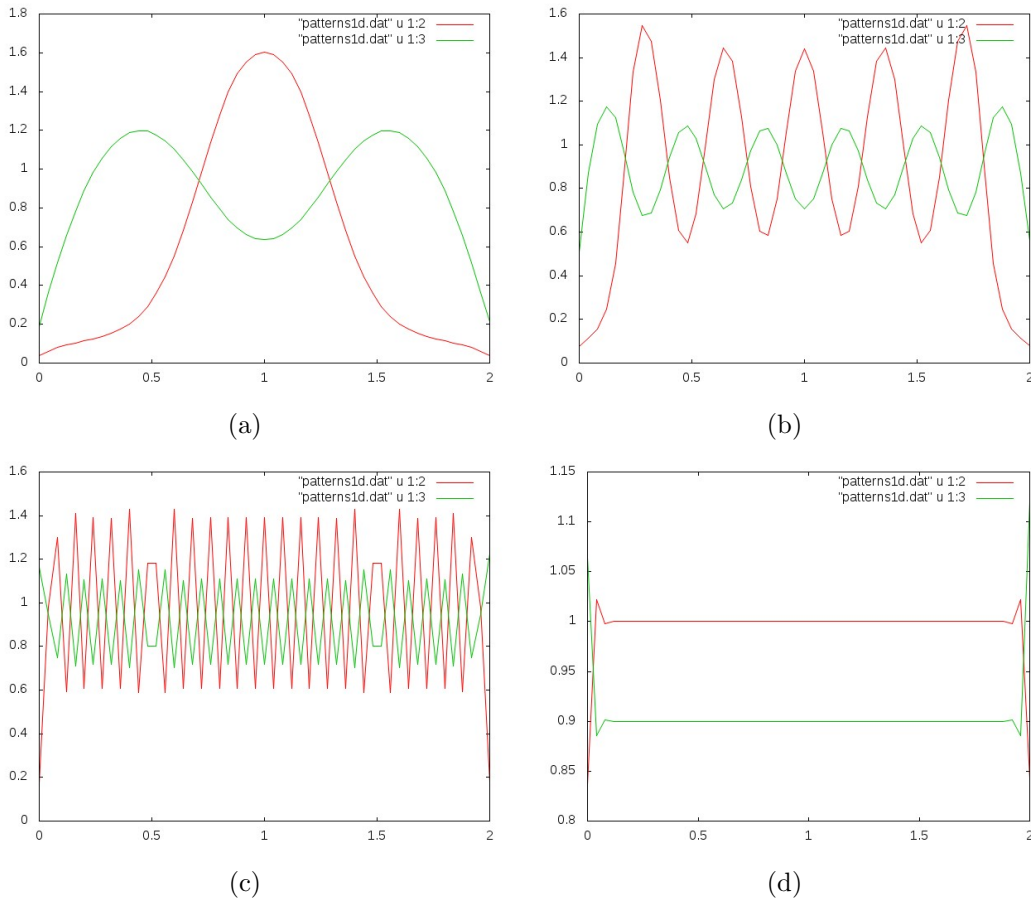


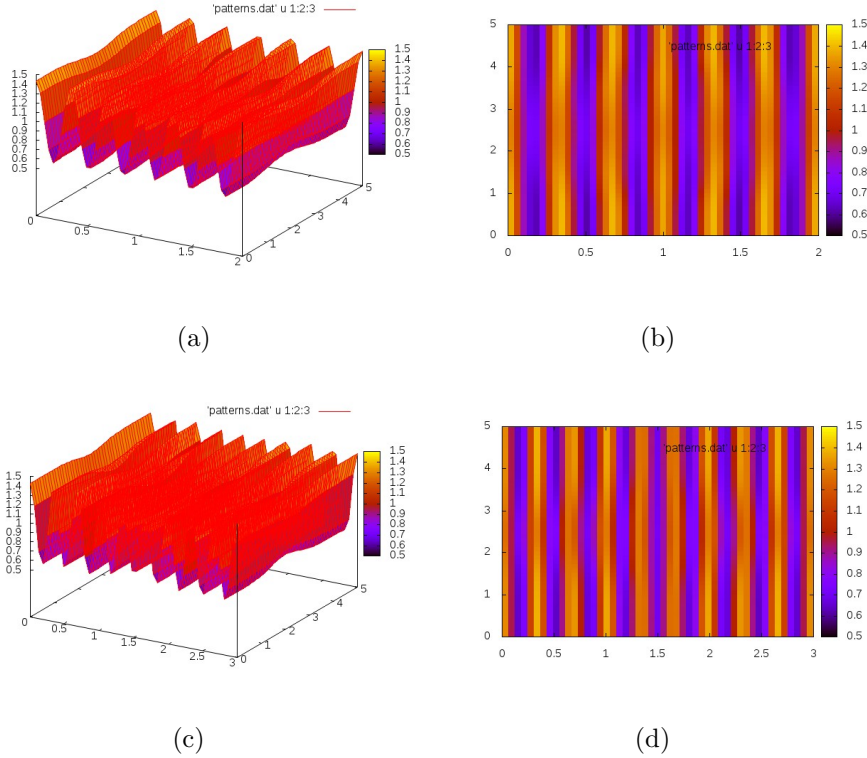
Figure 4: (a) $\gamma = 114$. (b) $\gamma = 1000$. (c) $\gamma = 10000$. (d) $\gamma = 50000$

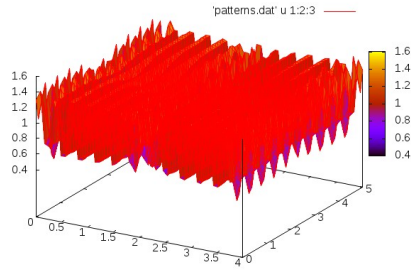
Flat torus simulations

In this subsection we include two kinds of graphics: ones in 3D, which are useful to show the instabilities of the system, and the other ones (with the same parameters) in 2D in which we can see the resulting pattern on the domain. Besides, as we have mentioned before, the parameter which has been tuned is A , from which we can see the importance of the thickness of the body or, equivalently, the effect of the narrowness of the rectangle. Hence, parameters γ and B have been considered fixed, particularly, $\gamma = 1000$ and $B = 5$.

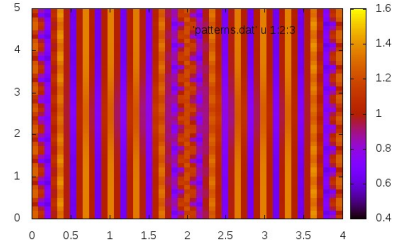
We now remember that, in 4.2.2, we have justified that small values of A lead to 1D patterns (stripes) and, conversely, sufficiently large values of this parameter produce 2D patterns (stripes or spots).

As evidence, we can see in Figure 5 how increasing values of this parameter we have a gradation from a striped pattern to a spotted one.

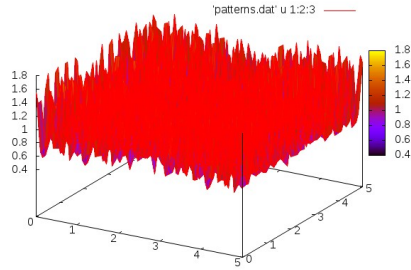




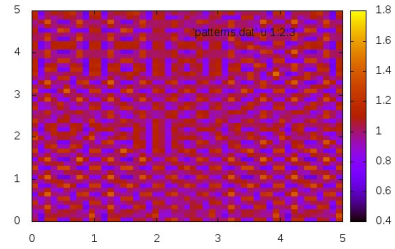
(e)



(f)



(g)



(h)

Figure 5: (a) and (b): $A = 2$; (c) and (d): $A = 3$; (e) and (f): $A = 4$; (g) and (h): $A = 5$.

Torus simulations

In this final simulations we also have started representing two types of graphics: ones which show the instabilities, as the previous case, and another ones which shows the pattern on a torus with the specific radius we consider. As we have said before, we want to show the effect of the curvature tuning the parameter R_1 . Particularly, we have considered fixed γ and R_2 with values $\gamma = 700$ and $R_2 = 0.5$.

We now remmember that, in 4.2.3, we have justified that small values of R_1 lead to 1D patterns (stripes) and, conversely, sufficiently large values of this parameter produce 2D patterns (stripes or spots).

As evidence, we can see in Figure 6 how increasing values of this paramater we have a gradation from a pattern which is quite striped to a totally spotted one.

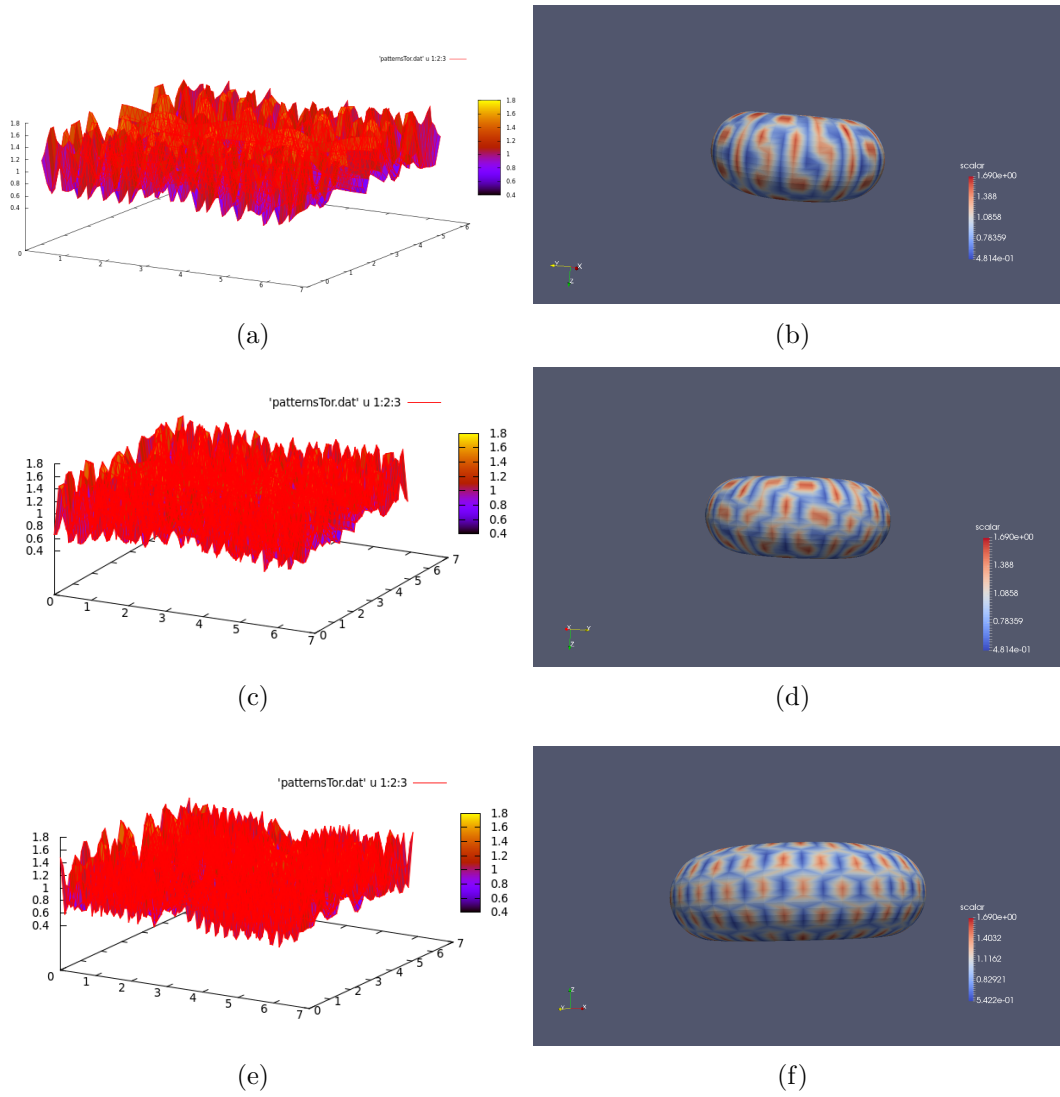


Figure 6: (a) and (b): $R_1 = 0.6$; (c) and (d): $R_1 = 0.75$; (e) and (f): $R_1 = 1$.

6 Conclusions

6.1 Conclusions on the approach of the project

The mathematical problem of pattern formation can be studied by different areas. One can choose one of them and start to develop it with precision focusing on this particular area. However, in this project we have given a general vision of how each mathematical branch interferes with this problem in order to have a global vision of it.

We first have learnt how we obtain the general reaction-diffusion system of equations from studying the **modeling** of chemical reactions and diffusion. If one wants to focus in this area, it could be a good idea to model some examples of chemical reactions which can appear during the morphogenesis. Besides, we have given (as additional information for this project) solutions for the substances concentrations in the simplest case. Therefore, one could have computed, as well, solutions for cases with higher complexity.

In the third chapter the area which acquires more importance is **differential equations and dynamical systems** because we have mainly studied a bifurcation, the so-called *Turing instability*. However, if one wants to deepen this topic, there is the option of studying the *Hopf bifurcation*, which is similar to Turing's one but it is oscillatory instead of stationary.

Once finished this chapter, **topology** and **geometry** get the lead role in Chapter 4. First of all, we have seen that studying the effect of the narrowness of the rectangle $[0, A] \times [0, B]$ that we have considered along this project is equivalent to see how the thickness of the animal body affects the resulting pattern. In addition, we have been interested in study the importance of the curvature, particularly considering a torus as our domain. However, we have not done a rigorous explanation since it requires to repeat most of the contents of the third chapter with the form of Laplacian in the torus. Therefore, if there is an interest to deepen this part one can do a rigorous study in this surface and as well in another ones like spheres, cones... Furthermore, there some researchers studying this importance of the geometry in growing domains.

Finally, we have made a **numerical study** of the problem using the *explicit finite difference method*. This method is clear and easy to program. However, its stability is limited and we have to consider sufficiently small temporal step to have accurate solutions. Therefore, one option to expand the vision of the problem from this area is, on the one hand, to study implicit schemes of the finite difference method or, on the other hand, study directly some other methods with higher complexity.

All in all, although it is more usual to choose one area and focus on it, I am delighted with my decision to make a global vision of the problem, because I think it is very interesting to see how different mathematical areas work together in a same topic giving a high variety of useful information.

6.2 Conclusions on the results of the project

As we have explained above, along all this project we have developed some different aspects about pattern formation (reaction-diffusion models, Turing instability and the influence of the geometry). Furthermore, to show the results graphically, we have designed a program in C using all this information. Then, after executing it tuning some geometrical parameters, we have observed that the simulations correspond to the results we have given in this bachelor's thesis.

Therefore, we could think that our contents are correct. However, we have made a last check to ensure that our program is correct as well. This test has consisted of simulating Figure 5.3 of the source [9] and verifying that the results are the same. Since all the programs designed in this project follow the same structure we have chosen only one of them: flat torus case.

We can see in Figure 5 that we have obtained similar patterns, which is another sign to contemplate that our outcome is correct. In addition, we can also see the effect of the parameter γ in flat torus.

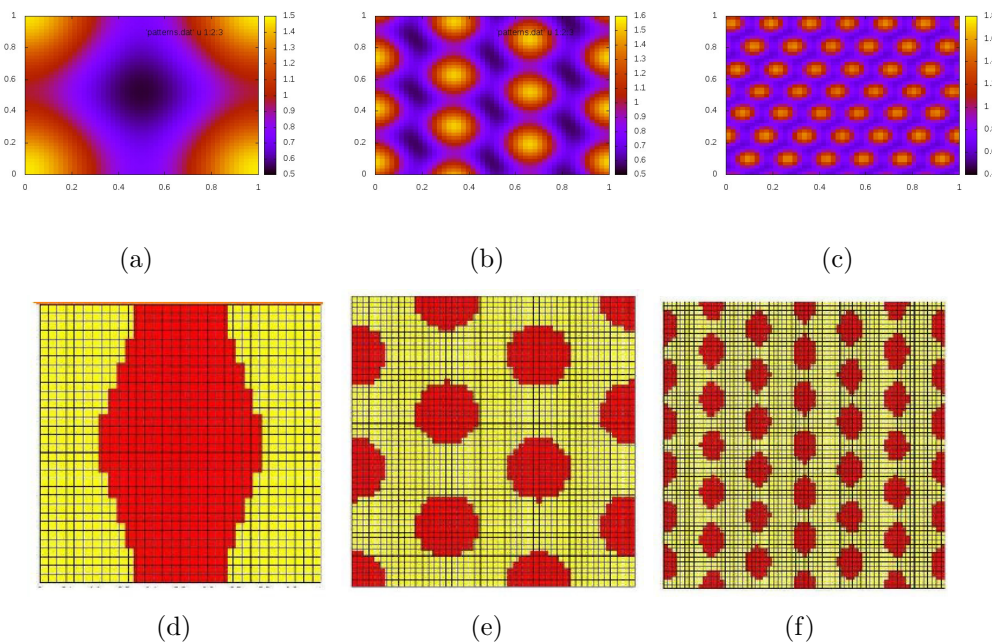


Figure 7: Simulations of the Schakenberg's model with $\gamma = 114$, $\gamma = 1000$ and $\gamma = 5000$. (a), (b) and (c) are obtained from my program and (d), (e) and (f) from source [9].

References

- [1] Castillo J.A.; Padilla P.; Sánchez F.: Patrones de Turing-Hopf en dominios con crecimiento, *La gaceta de la RSME*, 18(2): 269-286, 2015.
- [2] Cross, M.; Greenside, H.: *Pattern formation and Dynamics in Nonequilibrium systems*, Cambridge University Press, 2009.
- [3] Juárez Valencia L.H.: *Diferencias finitas e Ecuaciones diferenciales parciales* [course notes], 2006.
- [4] Kuttler C.: *Reaction-diffusion equations with applications* [course notes], 2011.
- [5] Maini P.K.; Benson D.L.; Sherratt J.A.: Pattern formation in reaction-diffusion models with spatially inhomogeneous diffusion coefficients, *IMA Journal of Mathematics Applied in Medicine and Biology*, 9, 197-213, 1992.
- [6] Murray, J.D.: *Mathematical Biology: Spatial models and Biomedical Applications*, third edition, Springer-Verlag, 2003.
- [7] Plaza R.G.; Sánchez-Garduño F.; Padilla P.; Barrio R.A.; Maini P.K.: The Effect of Growth and Curvature on Pattern Formation, *Journal of Dynamics and Differential equations*, 16(4), 2004.
- [8] Turing, A.M.: The Chemical Basis of Morphogenesis, *Phil. Trans. R. Soc. London*, 237: 37-72, 1952.
- [9] Ledesma Durán A.: Patrones de Turing en sistemas biológicos, *Universidad Autónoma Metropolitana - Iztapalapa*, 2012.

A Laplacian on a torus

We start with the general case of a surface parametrized by $\chi : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ in orthogonal coordinates (ζ, η) , which satisfy:

$$\left\langle \frac{\partial \chi}{\partial \zeta}, \frac{\partial \chi}{\partial \eta} \right\rangle = 0$$

with $\langle \cdot, \cdot \rangle$ denoting the inner product.

In this case, the Laplacian operator of a scalar function u is expressed by:

$$\Delta u = \frac{1}{h_1 h_2} \left[\left(\frac{h_2}{h_1} u_\zeta \right)_\zeta + \left(\frac{h_1}{h_2} u_\eta \right)_\eta \right],$$

where the subindex denotes derivation ($u_\zeta = \frac{\partial u}{\partial \zeta}$) and:

$$\begin{cases} h_1 = \left| \frac{\partial \chi}{\partial \zeta} \right| \\ h_2 = \left| \frac{\partial \chi}{\partial \eta} \right| \end{cases}$$

In section 4.4. we have parametrized torus as follows

$$\chi(\theta_1, \theta_2) = \begin{pmatrix} (R_1 + R_2 \cos \theta_2) \cos \theta_1 \\ (R_1 + R_2 \cos \theta_2) \sin \theta_1 \\ R_2 \sin \theta_2 \end{pmatrix}.$$

Therefore, we can compute the laplacian we want to obtain using the first formula of this appendix. We thus need the expressions of h_1 and h_2 :

$$h_1 = \left| \frac{\partial \chi}{\partial \theta_1} \right| = \left| \begin{pmatrix} -(R_1 + R_2 \cos \theta_2) \sin \theta_1 \\ (R_1 + R_2 \cos \theta_2) \cos \theta_1 \\ 0 \end{pmatrix} \right| = R_1 + R_2 \cos \theta_2$$

$$h_2 = \left| \frac{\partial \chi}{\partial \theta_2} \right| = \left| \begin{pmatrix} -R_2 \cos \theta_1 \sin \theta_2 \\ -R_2 \sin \theta_1 \sin \theta_2 \\ R_2 \cos \theta_2 \end{pmatrix} \right| = R_2$$

Replacing h_1 and h_2 in the first formula we obtain:

$$\begin{aligned} \Delta u &= \frac{1}{R_2(R_1 + R_2 \cos \theta_2)} \left[\left(\frac{R_2}{R_1 + R_2 \cos \theta_2} u_{\theta_1} \right)_{\theta_1} + \left(\frac{R_1 + R_2 \cos \theta_2}{R_2} u_{\theta_2} \right)_{\theta_2} \right] \\ &= \frac{1}{(R_1 + R_2 \cos(\theta_2))^2} u_{\theta_1 \theta_1} - \frac{\sin \theta_2}{R_2(R_1 + R_2 \cos \theta_2)} u_{\theta_2} + \frac{1}{R_2^2} \end{aligned}$$

B Stability of the explicit finite difference method

We will develop this section around the one dimensional case

$$\frac{\partial u}{\partial t} = g(u) + D \frac{\partial^2 u}{\partial y^2}.$$

However, as it is not the main goal of this project, we will simplify it omitting nonlinear terms, that is, considering $g(u)=0$ and obtaining the equation

$$\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial y^2},$$

which is the *diffusion equation in one dimension*.

We now remember from the fifth chapter that the explicit scheme of finite difference method for reaction-diffusion equation in 1D with $g(u) = 0$ is:

$$u_j^{n+1} = u_j^n + \frac{D\Delta t}{h^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n).$$

Specifically, in matrix notation:

$$\begin{pmatrix} u_0^{n+1} \\ u_1^{n+1} \\ \vdots \\ u_{m-1}^{n+1} \end{pmatrix} = \begin{pmatrix} u_0^n \\ u_1^n \\ \vdots \\ u_{m-1}^n \end{pmatrix} + \frac{D\Delta t}{h^2} \begin{pmatrix} -2 & 1 & 0 & \dots & 1 \\ 1 & -2 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 2 \end{pmatrix} \begin{pmatrix} u_0^n \\ u_1^n \\ \vdots \\ u_{m-1}^n \end{pmatrix}$$

Calling C the previous matrix, we can rewrite the explicit scheme in the form:

$$u^{n+1} = \left(I + \frac{D\Delta t}{h^2}C\right)u^n.$$

Therefore we can write for any time n :

$$u^n = \left(I + \frac{D\Delta t}{h^2}C\right)^n u^0.$$

Now, in order to have stability, we have to impose that the eigenvalues σ_j of matrix $\left(I + \frac{D\Delta t}{h^2}C\right)$ must satisfy:

$$|\sigma_j| \leq 1,$$

for $j = 0, \dots, n-1$. Hence, we need to compute these eigenvalues. First of all, we give a definition and a lemma which will be useful to do this calculation.

Definition B.0.0.1. *A circulant matrix is a $n \times n$ matrix whose rows are composed of cyclically shifted versions of a list (c_0, \dots, c_{n-1}) , that is, a matrix which has the form:*

$$\begin{pmatrix} c_0 & c_{n-1} & c_{n-2} & \dots & c_1 \\ c_1 & c_0 & c_{n-1} & \dots & c_2 \\ c_2 & c_1 & c_0 & \dots & c_3 \\ \dots & \dots & \dots & \dots & \dots \\ c_{n-1} & c_{n-2} & c_{n-3} & \dots & c_0 \end{pmatrix}$$

Lemma B.0.0.1. *The corresponding eigenvalues λ of a circulant matrix C with components (c_0, \dots, c_{n-1}) are:*

$$\lambda_j = c_0 + c_{n-1}\omega_j + c_{n-2}\omega_j^2 + \dots + c_1\omega_j^{n-1},$$

where $\omega_j = \exp(\frac{2\pi ij}{n})$, $i = \sqrt{-1}$ and $j = 0, 1, \dots, n-1$.

We now observe that our matrix C of the explicit scheme is a circulant matrix with coefficients:

$$\begin{cases} c_0 = 2 \\ c_1 = c_{n-1} = 1 \\ c_2 = \dots = c_{n-2} = 0 \end{cases}$$

Therefore, its eigenvalues λ_j are:

$$\lambda_j = -2 + \omega_j + \omega_j^{n-1} = -2 + \exp(\frac{2\pi ij}{n}) + \exp(\frac{2\pi ij(n-1)}{n}).$$

However, the eigenvalues that we want to study are the ones of $(I + \frac{D\Delta t}{h^2}C)$. One can easily observe that they are:

$$\sigma_j = 1 + \frac{D\Delta t}{h^2}\lambda_j = 1 + \frac{D\Delta t}{h^2}(-2 + \exp(\frac{2\pi ij}{n}) + \exp(\frac{2\pi ij(n-1)}{n})).$$

We now can impose the stability condition:

$$\begin{aligned} |\sigma_j| \leq 1 &\iff \left| 1 + \frac{D\Delta t}{h^2}(-2 + \exp(\frac{2\pi ij}{n}) + \exp(\frac{2\pi ij(n-1)}{n})) \right| \leq 1 \\ \stackrel{(1)}{\iff} \left| (1 - 2\frac{D\Delta t}{h^2}) + \frac{D\Delta t}{h^2}(\exp(\frac{2\pi ij}{n}) + \exp(\frac{-2\pi ij}{n})) \right| \leq 1 &\stackrel{(2)}{\iff} \left| (1 - 2\frac{D\Delta t}{h^2}) + 2\frac{D\Delta t}{h^2} \cos(\frac{2\pi j}{n}) \right| \leq 1 \\ \stackrel{(3)}{\iff} \left| 1 - 4\frac{D\Delta t}{h^2} \sin^2(\frac{2\pi j}{2n}) \right| \leq 1 &\iff 0 \leq 4\frac{D\Delta t}{h^2} \sin^2(\frac{2\pi j}{2n}) \leq 2 \\ &\iff 0 \leq \frac{D\Delta t}{h^2} \leq \frac{1}{2 \sin^2(\frac{2\pi j}{2n})}. \end{aligned}$$

The equivalences are justified by:

(1) The complex exponential is $2\pi i$ periodic.

(2) $e^{yi} + e^{-yi} = 2\cos(y)$.

(3) $\cos 2\alpha = 1 - 2\sin^2 \alpha$.

The last condition we have obtained is equivalent to:

$$0 \leq \frac{D\Delta t}{h^2} \leq \frac{1}{2} \leq \iff \Delta t \leq \frac{h^2}{2D}.$$

This is the reason why in 5.2.4 we have said that we have to choose Δt with h^2 -order to have some stability.

C The program

We have designed three programs in C, which differ in what domain we have taken in each one: 1D, flat torus and torus. However, they have the same structure:

- We have considered a mesh of 50 intervals (or 50×50 in 2D case).
- The program asks you to give the values of the parameters of the model, the dimensions of its domain, total time and temporal step.
- The initial conditions are oscillations around the base state. Particularly, we have used the following ones (from [9]):

One dimensional case

$$u_1(0, x) = u_{1b} + 0.001 \sum_{j=1}^{25} \frac{\cos(2\pi jx)}{j},$$

$$u_2(0, x) = u_{2b} + 0.001 \sum_{j=1}^{25} \frac{\cos(2\pi jx)}{j}.$$

Two dimensional cases

$$u_1(0, x, y) = u_{1b} + 0.0016 \cos(2\pi(x + y)) + 0.01 \sum_{j=1}^8 \cos(2\pi j),$$

$$u_2(0, x, y) = u_{2b} + 0.0016 \cos(2\pi(x + y)) + 0.01 \sum_{j=1}^8 \cos(2\pi j).$$

- We have taken periodic conditions. It implies that solutions in one node i or (i, j) are the same ones of the ones in the node $M + i$ or $(M + i, M + j)$ respectively. This is the reason why we have used the modulus operator in C % along all the program in expressions like $u[(M + i)\%i]$

- We have applied the explicit scheme explained in section 5.2.4.

One dimensional case

```
1 #include <stdio.h>
  #include <math.h>
3 #include <stdlib.h>
5 #define pi 3.1415926535897932
  #define M 50
7
  double gm= 0;
9
  /*FUNCTIONS OF THE MODEL*/
11 double f1(double a, double u1, double u2) {
  double f1;
```

```

13 f1=gm*(a-u1*(1-u1*u2));
   return f1;
15 }

17 double f2(double b,double u1,double u2){
   double f2;
19 f2=gm*(b-u1*u1*u2);
   return f2;
21 }

23 int main(void) {
   int n, i, j, N;
25 double a, b, l, d, h, T, pastemp, x[M+1], sum[M+1], u1[M+1], u2[M+1],
       u1b[M+1], u2b[M+1];

27 printf("Introduce parameters a, b, gm and d:\n");
   scanf("%lf %lf %lf %lf", &a, &b, &gm, &d);
29
   printf("Introduce length, total time T and temporal step\n");
31 scanf("%lf %lf %lf", &l, &T, &pastemp);

33 /*SPATIAL STEP*/
   h=l/M;
35
   /*DISCRETE DOMAIN*/
37 x[0]=0.;
   for(i=0;i<M;i++){
39 x[i+1]=x[i]+h;
   }
41
   /*NUMBER OF TEMPORAL ITERATIONS*/
43 N=(int)((T/pastemp)+1);

45 /*BASE STATE*/
   for(i=0; i<M+1; i++){
47 u1b[i]= a+b;
   u2b[i]=b/((a+b)*(a+b));
49
   }
51
   /*INITIAL CONDITIONS*/
53 for(i=0; i<M+1; i++){
   sum[i]=0.;
55 for(j=0; j<25;j++){
   sum[i]+=(cos(2*pi*(j+1)*x[i]))/(j+1));
57 }
   u1[i]=u1b[i]+0.001*sum[i];
59 u2[i]=u2b[i]+0.001*sum[i];
   }

61 FILE *patterns1din;
63 patterns1din=fopen("patterns1din.dat","w");
   if(patterns1din==NULL) exit(1);
65
   for(i=0;i<M+1;i++){

```

```

67 fprintf(patterns1d, "%lf %lf %lf \n", x[i], u1[i], u2[i]);
69 }
71 for(n=0; n<N; n++){
72   for(i=0; i<M+1; i++){
73     u1[i]=u1[i%M]+(pastemp*f1(a, u1[i%M], u2[i%M]))+(pastemp*(1./(h*h))*(u1
      [(i+1)%M]-2*u1[i%M]+u1[(i-1)%M]));
74     u2[i]=u2[i%M]+(pastemp*f2(b, u1[i%M], u2[i%M]))+(d*pastemp*(1./(h*h))*(
      u2[(i+1)%M]-2*u2[i%M]+u2[(i-1)%M]));
75   }
76 }
77 FILE *patterns1d;
78 patterns1d=fopen("patterns1d.dat", "w");
79 if(patterns1d==NULL) exit(1);
81 for(i=0; i<M+1; i++){
82   fprintf(patterns1d, "%lf %lf %lf \n", x[i], u1[i], u2[i]);
83 }
84 }
85 return 0;
86 }

```

Flat torus

```

1 #include<stdio.h>
2 #include<math.h>
3 #include<stdlib.h>
4
5
6 #define pi 3.1415926535897932
7 #define Mx 50
8 #define My 50
9
10 double gm= 0;
11
12
13 /*FUNCTIONS OF THE MODEL*/
14 double f1(double a, double u1, double u2){
15   double f1;
16   f1=gm*(a-u1*(1-u1*u2));
17   return f1;
18 }
19
20 double f2(double b, double u1, double u2){
21   double f2;
22   f2=gm*(b-u1*u1*u2);
23   return f2;
24 }
25
26 int main(void) {

```

```

27 int n, i, j, k, N;
double a, b, lx, ly, d, hx, hy, T, pastemp, x[Mx+1], y[My+1], sum[Mx
+1][My+1], u1[Mx+1][My+1], u2[Mx+1][My+1], u1b[Mx+1][My+1], u2b[Mx
+1][My+1];
29
printf("Introduce parameters a, b, gm and d:\n");
31 scanf("%lf %lf %lf %lf", &a, &b, &gm, &d);

33 printf("Introduce length, height, total time T and temporal step\n");
scanf("%lf %lf %lf %lf", &lx, &ly, &T, &pastemp);
35
/*SPATIAL STEP*/
37 hx=lx/Mx;
hy=ly/My;
39
/*DISCRETE DOMAINS*/
41 x[0]=0.;
for (i=0; i<Mx; i++){
43 x[i+1]=x[i]+hx;
}
45 y[0]=0.;
for (j=0; j<My; j++){
47 y[j+1]=y[j]+hy;
}
49
/*NUMBER OF TEMPORAL ITERATIONS*/
51 N=(int)((T/pastemp)+1);

53 /*BASE STATE*/
for (i=0; i<Mx+1; i++){
55 for (j=0; j<My+1; j++){
u1b[i][j]= a+b;
57 u2b[i][j]=b/((a+b)*(a+b));
}
59 }

61 /*INITIAL CONDITIONS*/
for (i=0; i<Mx+1; i++){
63 for (j=0; j<My+1; j++){
sum[i][j]=0.;
65 for (k=0; k<8; k++){
sum[i][j]+=(cos(2*pi*k*x[i]));
67 }
u1[i][j]=u1b[i][j]+0.0016*cos(2*pi*(x[i]+y[j]))+0.001*sum[i][j];
69 u2[i][j]=u2b[i][j]+0.0016*cos(2*pi*(x[i]+y[j]))+0.001*sum[i][j];
}
71 }

73 FILE *patternsIn;
patternsIn=fopen("patternsIn.dat", "w");
75 if (patternsIn==NULL) exit(1);

77 for (i=0; i<Mx+1; i++){
for (j=0; j<My+1; j++){

```



```

79 fprintf(patternsIn , "%lf %lf %lf %lf \n" , x[i] ,y[j] ,u1[i][j] ,u2[i][j])
    ;
    }
81 }
83
84 for (n=0; n<N; n++){
85
86     for (i=0; i<Mx+1; i++){
87         for (j=0; j<My+1; j++){
88             u1[i][j]=u1[i%Mx][j%My]+(pastemp*f1(a,u1[i%Mx][j%My],u2[i%Mx][j%My]))
                +(pastemp*(1./(hx*hx))*(u1[(i+1)%Mx][j%My]-2*u1[i%Mx][j%My]+u1[(Mx+
                i-1)%Mx][j%My]))+(pastemp*(1./(hy*hy))*(u1[i%Mx][(j+1)%My]-2*u1[i%
                Mx][j%My]+u1[i%Mx][(My+j-1)%My]));
89             u2[i][j]=u2[i%Mx][j%My]+(pastemp*f2(b,u1[i%Mx][j%My],u2[i%Mx][j%My]))
                +(d*pastemp*(1./(hx*hx))*(u2[(i+1)%Mx][j%My]-2*u2[i%Mx][j%My]+u2[(
                Mx+i-1)%Mx][j%My]))+(d*pastemp*(1./(hy*hy))*(u2[i%Mx][(j+1)%My]-2*
                u2[i%Mx][j%My]+u2[i%Mx][(My+j-1)%My]));
91         }
92     }
93
94 }
95
96 FILE *patterns;
97 patterns=fopen("patterns.dat","w");
98 if(patterns==NULL) exit(1);
99
100
101 for (i=0;i<Mx+1;i++){
102     for (j=0;j<My+1;j++){
103         fprintf(patterns , "%lf %lf %lf %lf \n" , x[i] ,y[j] ,u1[i][j] ,u2[i][j]) ;
104     }
105 }
106
107 return 0;
108 }

```

Torus

```

#include<stdio.h>
2 #include<math.h>
#include<stdlib.h>
4
#define pi 3.1415926535897932
6 #define M1 50
#define M2 50
8
double gm=0;
10
/*FUNCTIONS OF THE MODEL*/
12 double f1(double a,double u1,double u2){

```

```

double f1;
14 f1=gm*(a-u1*(1-u1*u2));
return f1;
16 }

18 double f2(double b,double u1,double u2){
double f2;
20 f2=gm*(b-u1*u1*u2);
return f2;
22 }

24 int main(void) {
int n, i, j, k, N;
26 double a, b, R1, R2, d, h1, h2, T, pastemp, theta1[M1+1], theta2[M2
+1], sum[M1+1][M2+1], u1[M1+1][M2+1], u2[M1+1][M2+1], u1b[M1+1][M2
+1], u2b[M1+1][M2+1], x[M1+1][M2+1], y[M1+1][M2+1], z[M1+1][M2+1];

28 printf("Introduce parameters a, b, gm and d:\n");
scanf("%lf %lf %lf %lf", &a, &b, &gm, &d);
30
printf("Introduce R1, R2, total time T and temporal step\n");
32 scanf("%lf %lf %lf %lf", &R1, &R2, &T, &pastemp);

34 /*SPATIAL STEP*/
h1=2*pi/M1;
36 h2=2*pi/M2;

38 /*DISCRETE DOMAINS*/
theta1[0]=0.;
40 for (i=0;i<M1;i++){
theta1[i+1]=theta1[i]+h1;
42 }
theta2[0]=0.;
44 for (j=0;j<M2;j++){
theta2[j+1]=theta2[j]+h2;
46 }

48 /*NUMBER OF TEMPORAL ITERATIONS*/
N=(int)((T/pastemp)+1);
50 printf("number of iterations: %d", N);

52 /*BASE STATE*/
for (i=0; i<M1+1; i++){
54 for (j=0; j<M2+1; j++){
u1b[i][j]= a+b;
56 u2b[i][j]=b/((a+b)*(a+b));
}
58 }

60 /*INITIAL CONDITIONS*/
for (i=0; i<M1+1; i++){
62 for (j=0; j<M2+1; j++){
sum[i][j]=0.;
64 for (k=0; k<8;k++){
sum[i][j]+=(cos(2*pi*k*theta1[i]));

```

```

66 }
   u1[i][j]=u1b[i][j]+0.0016*cos(2*pi*(theta1[i]+theta2[j]))+0.001*sum[i
   ][j];
68 u2[i][j]=u2b[i][j]+0.0016*cos(2*pi*(theta1[i]+theta2[j]))+0.001*sum[i
   ][j];
   }
70 }

72 FILE *patternsTorIn;
   patternsTorIn=fopen("patternsTorIn.dat","w");
74 if(patternsTorIn==NULL) exit(1);

76 for(i=0;i<M1+1;i++){
   for(j=0;j<M2+1;j++){
78 fprintf(patternsTorIn, "%lf %lf %lf %lf \n", theta1[i],theta2[j],u1[i
   ][j],u2[i][j]);
   }
80 }

82 for(n=0; n<N; n++){
84 for(i=0; i<M1+1; i++){
86 for(j=0; j<M2+1; j++){
   u1[i][j]=u1[i%M1][j%M2]+(pastemp*f1(a,u1[i%M1][j%M2],u2[i%M1][j%M2]))
   +(pastemp*(1./((R1+(R2*cos(theta2[j])))*(R1+(R2*cos(theta2[j])))))
   *(1./(h1*h1))*(u1[(i+1)%M1][j%M2]-2*u1[i%M1][j%M2]+u1[(M1+i-1)%M1][j
   %M2]))+(pastemp*(1./(R2*R2))*(1./(h2*h2))*(u1[i%M1][(j+1)%M2]-2*u1[
   i%M1][j%M2]+u1[i%M1][(M2+j-1)%M2]))-(pastemp*(sin(theta2[j]))*(u1[i
   %M1][(j+1)%M2]-u1[i%M1][(j-1)%M2])/(2*h2*R2*(R1+(R2*cos(theta2[j]))
   )));
88 u2[i][j]=u2[i%M1][j%M2]+(pastemp*f2(b,u1[i%M1][j%M2],u2[i%M1][j%M2]))
   +(d*pastemp*(1./((R1+(R2*cos(theta2[j])))*(R1+(R2*cos(theta2[j])))))
   *(1./(h1*h1))*(u2[(i+1)%M1][j%M2]-2*u2[i%M1][j%M2]+u2[(M1+i-1)%M1
   ][j%M2]))+(d*pastemp*(1./(R2*R2))*(1./(h2*h2))*(u2[i%M1][(j+1)%M2
   ]-2*u2[i%M1][j%M2]+u2[i%M1][(M2+j-1)%M2]))-(d*pastemp*(sin(theta2[j
   ]))*(u2[i%M1][(j+1)%M2]-u2[i%M1][(j-1)%M2])/(2*h2*R2*(R1+(R2*cos(
   theta2[j])))));
90 }
   }
92 }

94 }

96 FILE *patternsTor;
   patternsTor=fopen("patternsTor.dat","w");
98 if(patternsTor==NULL) exit(1);
   fprintf(patternsTor, "x coord, ycoord, z coord, scalar\n");
100
   FILE *patternsTorGnuplot;
102 patternsTorGnuplot=fopen("patternsTorGnuplot.dat","w");
   if(patternsTorGnuplot==NULL) exit(1);
104

```

```

106 for (i=0;i<M1+1;i++){
108 for (j=0;j<M2+1;j++){
x[i][j] = (R1+R2*cos(theta2[j]))*cos(theta1[i]);
110 y[i][j] = (R1+R2*cos(theta2[j]))*sin(theta1[i]);
z[i][j] = R2*sin(theta2[j]);
112 fprintf(patternsTorGnuplot, "%lf %lf %lf %lf \n", theta1[i], theta2[j]
], u1[i][j], u2[i][j]);
fprintf(patternsTor, "%lf, %lf, %lf, %lf \n", x[i][j], y[i][j], z[i][j]
], u1[i][j]);
114 }
116 }
118 return 0;
}

```