Treball final de grau

GRAU DE MATEMÀTIQUES

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Dynamics of a Discrete Hypercycle

Autor: Júlia Perona García

Directors: Dr. Ernest Fontich, Dr. Josep Sardanyés
Realitzat a: Departament de Matemàtiques i Informàtica
Barcelona, June 27, 2018

Abstract

The concept of the Hypercycle was introduced in 1977 by Manfred Eigen and Peter Schuster within the framework of origins of life and prebiotic evolution. Hypercycle are catalytic sets of macromolecules, where each replicator catalyzes the replication of the next species of the set. This system was proposed as a possible solution to the information crisis in prebiotic evolution. Hypercycles are cooperative systems that allow replicators to increase their information content beyond the error threshold.

This project studies the dynamics of a discrete-time model of the hypercycle considering heterocatalytic interactions. To date, hypercycles' dynamics has been mainly studied using continuous-time dynamical systems. We follow the Hofbauer's discrete model [13]. First, we introduce some important and necessary mathematical notions. Then, we also review the biological concept of the hypercycle and some criticisms that it has received.

We present a complete proof of the fact that the hypercycle is a cooperative system. Also, we present an analytic study of the fixed point in any dimension and its stability. In particular, in dimension three we prove that fixed point is globally asymptotically stable. In dimension four we have obtained a stable invariant curve for all values of the discreteness parameter.

2010 Mathematics Subject Classification. 37C05, 37N25

Acknowledgements

I would like to thank my directors, Ernest and Josep, to support me throughout this project. To Ernest for all the time he dedicated to me, which is a lot, for all the things that I have learnt from him and for making me feel comfortable at every moment. To Josep for his dedication to me, the motivation that he gave me and for providing me with biological knowledge. They are great scientists and better people.

I would also like to express my thanks to my father and my sister, for staying always by my side.

Thanks to my friends to turn every moment into an adventure.

Thanks to Ricardo for staying by my side when I needed it most, for appreciating everything I do and becoming so special to me.

Thanks to all the great professors I have met during these years at university.

I would like to dedicate this project to my mother, I know that she would have liked to see me reach my goals.

Contents

Chapter 1

Introduction

The origin of life is an open problem being also a recurrent and fascinating subject of scientific research. In the context of prebiotic evolution, it is necessary a proper definition of the word *life*. Living systems can be defined as information-coding replicating systems, having the property of so-called open-ended evolution i.e., large capacity to evolve and adapt. Most of the initial investigations about the origin of life are theoretical. Initially, before the 20th century, the origin of life was attributed to the spontaneous generation. Afterwards, Aleksandr Oparin suggested that, millions of years ago on the primitive Earth, the first amino acids (the constituents of the proteins) were formed from random molecular species [22, 20]. Oparin believed that, after that, more complex forms of life would have evolved following the law of natural selection, formulated previously by Darwin, being able to evolve *without limits* (i.e., open-ended evolution previously mentioned). Later, other scientists, such as Miller and Joan Oró carried out important experiments, showing that organic molecules can be synthesized from inorganics ones in prebiotic conditions. Specifically, by simulating in the laboratory the non-oxidative atmosphere, together with strong energy inputs due to lightning, they showed that molecules like amino acids and nucleic acids could be synthesized [19, 21]. Therefore, the first molecular replicators might have been formed, having the properties of life. The involving existent molecules could then have established different interactions between them, when appear the term prebiotic evolution having then the capabilities to form more complex organic structures [22, 8].

Once these organic structures might have been formed, replicators could have been polymerized. Replicators are macromolecules able to make copies of themselves. A good example and a strong candidate for a possible prebiotic replicator is RNA (RiboNucleic Acid)[22]. Some replicators are known to have catalytic ability, for example ribozymes. These are RNA molecules that are able to catalyze different chemical reactions [31]. It is known that this type of replicators have very high mutation rates. It is plausible to think that RNA-based prebiotic replicators might replicate under high mutation rates. That is, due to the lack of error correction mechanisms, these replicators might synthesize different sequences from the parental one. A population of replicators under large mutation rates produce the so-called quasispecies [22, 7, 8]. A quasispecies is a large group of genotypes (i.e., RNA sequences) presenting a very large genetic heterogeneity due to large mutation rates. In other words, they are a group of RNA macromolecules forming a cloud of mutants together with the initial parental sequence. This idea was first introduced by Manfred Eigen, in order to describe the molecular evolution of the primitive sets of replicating molecules [8]. The concept of quasispecies is very powerful and has been used as a modelling framework for different types of biological systems beyond the origin of life problem. For instance, this concept has been applied to investigate the dynamics of RNA viruses (e.g. immunodeficiency virus type 1, HIV-1) [2, 32] or the so-called cancer quasispecies [4]. In the latter case, it is known that cancer cells suffer of very large mutation rates, accumulating a lots of mutations and displaying extreme levels of genetic heterogeneity. Consequently, this allows tumors to escape the restrictions of the normal cell growth, causing an abnormal fast-growing system [4]. The fast replication dynamics of cancer cells, together with their high mutation rates, have been suggested to produce a quasispecies-like population structure [4, 2, 8, 32].

There is an important problem with the quasispecies structure and dynamics, the information threshold [7, 8]. As mentioned, the process of replication in the first prebiotic molecular systems is supposed to be highly error-prone. Eigen conjectured that replicators with large mutation rates and thus a large amount of accumulation of errors have a limit in the accumulation of the information they can encode. That is, there exists the so-called error threshold. In fact, this error threshold is defined as the threshold from which the encoded information of the master copy becomes random. The error threshold imposes a theoretical limit to the amount of information that an error-prone replicator can carry. That is, there is a maximum length of the replicating sequence allowing the maintenance of the information. Moreover, in a quasispecies population, the fastest replicating species will dominate over the entire population due to only-competition dynamics [8, 17]. It is known that, related to the critical sequence length, there exists a critical mutation rate, μ, per nucleotide and per replication cycle beyond which the genetic message melts down and becomes random. This critical mutation rate is given by $\mu_c = v^{-1}$, v being the length of the sequence, defining a mutation limit. As mentioned, if the mutation rate exceeds this critical value, the information is lost and the information of the population becomes random. Alternatively, the critical length of the sequence scales with the mutation rate as $v_c = \mu^{-1}$.

Therefore, the previous constraints cause an information crisis in the prebiotic evolution. This implies that there is a maximum length in the sequences, which is, more or less, of about 100 nucleotides, and thus, prebiotic systems could not increase their genetic information allowing for an increase in complexity [22, 24]. Also, as long as they could have a high information content, the replicators should be capable to encode protein sets responsible for the correction of errors during replication. From these premises, the so-called Eigen paradox arises [22, 8]. The Eigen's paradox can be stated as follows:

To increase the length of replicators, and consequently their genetic information, it is necessary the presence of an enzymatic machinery able to correct errors. However, in order to synthesize molecules responsible for error correction, it is necessary to have long sequences.

This formulation lead us to a chicken-egg problem: *What was first: the chicken or the egg?*.

As a solution to this informational crisis in quasispecies, Eigen and Schuster introduced the system investigated in this project, the hypercycle [7]. The hypercycle is a type of molecular organization that allows replicators to increase their information content under the suppositions of the Eigen's paradox. Specifically, as we will explain in Section 3, the hypercycle allows the coexistence of a set of sequences, each of them below the critical length imposed by the error threshold. Since these sequences cooperate between them, the amount of the information can be much larger than the one allowed by the error threshold.

In this project, we decided to study a discrete-time model of the hypercycle for two reasons. First, there are a lot of investigations about the continuous-time model associated to its dynamics and other concepts such as cooperation. However, there are few investigations about the discrete hypercycle. Hofbauer was the first author who studied the discrete model [13] and we decided to analyze and go in depth into his study. There are different investigations depending on the dimension of the model for time-continuous hypercycles. We will comment on these results in chapter (5). However, the role of the dimension in the dynamics for discrete-time models is scarce in the literature. Second, discrete-time models, in general, are very important in biological systems with processes of cooperation or facilitation (see Figure (3.1)) [25].

When we started to think about which topic we would like to study, we thought that it could be really interesting to study a specific biological model. Since we think that biology is an amazing science and the relation between this model and the origin and evolution of life was a boost for us.

As we will explain later, in chapter (3) there are different models for the hypercycle, the most essential and basic is in which the molecules in the system interact only with the previous molecule. This means that the replication of one molecule depends only on the contribution of its previous molecule. However, there exist other models for the hypercycle including other types of interactions, such as the so-called parasites and short circuits. These other interactions would suppose a problem with the hypothesis that this structure would mend the informational crisis problem.

In this chapter, as we have already seen, an historic and scientific context for the hypercycle is introduced. Also the causes of its appearance and problems that tried to solve. Finally we make a brief summary of the structure of the project and a short introduction for every chapter.

In the second chapter we have the goal of showing and understanding theoretical concepts, both basic and advanced about Dynamical Systems theory, focusing in the branch of discrete dynamical systems, according to our model. We spend the first section of this chapter on introducing some basic concepts about this theory, that we have seen in the *"Models Matemàtics i Sistemes Dinàmics* subject, and also, an important part of this section, we discuss results of Lyapunov Stability, presenting new theory that we have not studied during the degree courses. We give important definitions and three theorems that we will use later to study the dynamics of our system, in chapter (5). In the next section we study an important concept from ordinary differential equations theory that will be also useful in chapter (5). This concept is the Poincaré map. Also we introduce the statement of a theorem about the stability of periodic orbits. The last section in this first chapter is about circulant matrices. We introduce and prove one theorem that gives us an expression for the eigenvalues and eigenvectors of this type of matrices. this will be useful to us in order to find the eigenvalues and eigenvectors of the differential matrix for our system in the fixed point, in section (5.1).

In the third chapter the biological concept of the hypercycle is presented, also we talk about experiments where the existence of this type of structure has been proved. And also, we introduce some criticisms about it. After this, we dedicate a section to deduce, from a system of ordinary differential equations, the discrete model that we will use during our project, and the domain in which is defined our model. Our model is close to the continuous-time model for large values of a parameter introduced in the discrete-model.

The fourth chapter is dedicated to prove that our model is cooperative under a certain hypothesis. As we have already seen, the hypercycle is a structure that admits coexistence between different sequences, then, biologically speaking, cooperation means that none species extinct. We also give an hypothesis by which our system is exclusive, however we will not prove it, because we only have focused in the cooperation case.

In the fifth chapter we study the dynamics of our model and it is divided in three sections. In the first one, we discuss about the unique fixed point in the interior of the domain. First, we show the fixed point, both implicitly and explicitly. Then we apply a change of barycentric coordinates in the continuous-time model, because our model is close to it. This is useful to find the differential matrix of the system and to find its eigenvalues and eigenvectors. Once we have the eigenvalues we discuss the stability of the fixed point in terms of the dimension of our system. To study the stability we use the properties of the differential matrix, such as it is a circulant matrix, results that we prove in section (2.3). In next section we prove that the fixed point is a global asymptotically attractor in third dimension, i.e., it is a point that attracts each point of the domain, this is proved using the Lyapunov asymptotic global stability theorem, presented in section (2.1). Also, we study a particular case that its orbits have period six. Finally, in last section of this chapter, we study the invariant curves of the system for a specific dimension, and also we discuss their stability. To do this, we implement a program with C language, (see appendices), that using a pseudo-Poincaré section finds invariant curves of the system. With prefix pseudo, we refer to a Poincaré section applied to a discrete model, because, as we have said, our system is close to a continuous model, for certain values of a parameter of the model. We introduce the explanation of the program and we introduce a brief description of the main functions of the program. Also, we have some graphics to show the results obtained. We also study the possible bifurcations of our system in terms of a parameter of it. And we arrive to the conclusion that for a certain values there is no bifurcation and for other values it seems that there is a bifurcation.

Finally, in the last chapter we discuss the main results of this project and we also discuss possible projects for the future.

Chapter 2

Some mathematical preliminaries

In this chapter, our aim is to introduce some important mathematical concepts, definitions and theorems that are necessary to develop our project. We have divided this chapter into different sections.

2.1 Discrete dynamical systems and Lyapunov stability

First, we have some important concepts about discrete dynamical systems because our project is mainly focused on this branch of mathematics. All of the concepts in this section can be found in [30] and in the notes of the course *"Models Matemàtics i Sistemes Dinàmics"*, except the proofs of the Lyapunov stability theorems for discrete systems which have been proved in this project.

Definition 2.1. *A discrete dynamical system is a pair (D, f) where $D \subset \mathbb{R}^n$ and $f : D \subset \mathbb{R}^n \longrightarrow D \subset \mathbb{R}^n$ is a continuous function.*

Definition 2.2. *The phase space of a discrete dynamical system is the set D, i.e., the set of possible states of the system.*

Observation 2.3. *$f^k(x)$ denotes the value of the solution of the discrete system at time k starting at x. It is fulfilled that $f^0(x) = x$ and $x_k = f^k(x) = f(f^{k-1}(x))$.*

Definition 2.4. *The orbit of a point $x_0 \in D$ is a sequence:*

$$\Theta(x_0) = \{x_0, x_1 = f(x_0), x_2 = f^2(x_0), \ldots, x_n = f^n(x_0), \ldots\}.$$

Definition 2.5. *A fixed point of the system f is a point $x_0 \in D$ such that $f(x_0) = x_0$.*

Definition 2.6. *A periodic point of period $p \geq 1$ is a point $x_0 \in D$ such that $f^p(x_0) = x_0$ and $f^k(x_0) \neq x_0$ for $0 < k < p$.*

Definition 2.7. *A periodic orbit of period $p \geq 1$ is a sequence of points $\{x_0, x_1, \ldots, x_{p-1}\}$ such that $f(x_i) = x_{i+1}$, for $0 \leq i \leq p-2$, and $f(x_{p-1}) = x_0$.*

Now that we have some important definitions in dynamical systems, we begin with some concepts about Lyapunov stability.

Definition 2.8. *A fixed point, x_0, of the system is a Lyapunov stable fixed point if* $\forall \epsilon > 0$, $\exists \delta > 0$ *such that* $\forall x \in D$ *with* $\|x - x_o\| < \delta$, *we have* $\|f^k(x) - x_0\| < \epsilon$, $\forall k \in \mathbb{N}$. *We also say that the system is Lyapunov stable at x_0.*

Definition 2.9. *A fixed point, x_0, of the system is a Lyapunov unstable fixed point, if x_0 is not a Lyapunov stable fixed point. Similarly, the system is Lyapunov unstable at x_0 if the system is not Lyapunov stable at x_0.*

Definition 2.10. *A fixed point x_0 is asymptotically stable or attractor if it is stable and if $\exists \epsilon > 0$ such that $\forall x \in D$ with $\|x - x_o\| < \epsilon$ it is fulfilled that $\lim_{k\to\infty} f^k(x) = x_0$.*

Definition 2.11. *A fixed point x_0 is a repellor if it is attractor for a local inverse f^{-1}.*

Definition 2.12. *The basin of attraction or domain of attraction of x_0 is the set:*

$$A = \{x \in D \mid \lim_{k\to\infty} f^k(x) = x_0\}.$$

Definition 2.13. *A set X is invariant if $f(X) = X$.*

Observation 2.14. *A set X is positively invariant if $f(X) \subset X$ and, if f is invertible, it is negatively invariant if $f^{-1}(X) \subset X$.*

Theorem 2.15. *Let $D \subset \mathbb{R}^n$ be an open set, $x_0 \in D$ and $f : D \longrightarrow D$ such that:*

1. $f(x_0) = x_0$,
2. *f is differentiable at x_0,*
3. $|\lambda| < 1$, $\forall \lambda \in Spec\{Df(x_0)\}$.

Then, x_0 is asymptotically stable.

Theorem 2.16. *Let $D \subset \mathbb{R}^n$ be an open set, $x_0 \in D$ and $f : D \longrightarrow D$ such that:*

1. $f(x_0) = x_0$,
2. *f is of class $\mathcal{C}^1$,*
3. $\exists \lambda \in Spec\{Df(x_0)\}$ *such that* $|\lambda| > 1$.

Then, x_0 is unstable.

If some eigenvalues have modulus less than 1 and the others have modulus 1, the previous results do not apply. In such case, the concept of Lyapunov function and the theorems of Lyapunov Stability are very useful. So, we are going to introduce them.

Definition 2.17. *Let $V : U \subset \mathbb{R}^n \longrightarrow \mathbb{R}$ be a real valued continuous function. V is a Lyapunov function of the system f in a neighborhood U of the fixed point x_0 if:*

1. $V(x_0) = 0$,
2. $V(x) > 0$, $\forall x \in U \setminus \{x_0\}$,
3. $V \circ f(x) \leq V(x)$, $\forall x \in U \cap f^{-1}(U)$.

Observation 2.18. *If $V : U \longrightarrow \mathbb{R}$ satisfies:*

1. $V(x_0) = C$,
2. $V(x) > C$, $\forall x \in U \setminus \{x_0\}$,
3. $V \circ f(x) \leq V(x)$, $\forall x \in U \cap f^{-1}(U)$.

for some C, *then* $\tilde{V}(x) = V(x) - C$ *is a Lyapunov function.*

Theorem 2.19. ***Lyapunov Local Stability Theorem for maps****.*
Let $D \subset \mathbb{R}^n$ be an open set and $f : D \longrightarrow D$ a continuous discrete system. We suppose that there exists $x_0 \in D$ such that $f(x_0) = x_0$. If there exists a Lyapunov function V for x_0, then, x_0 is a Lyapunov stable fixed point.

Proof. Let $\epsilon > 0$ be arbitrary. Choose $0 < r \leq \epsilon$ such that $\overline{B}(x_0, r) \subset U \cap D$.
We know that f is continuous at x_0, so, we have that:

$$\forall r > 0,\ \exists \delta_1 > 0, \text{ such that } \forall x \in B(x_0, \delta_1) \text{ then } f(x) \in B(x_0, r). \tag{2.1}$$

Now, we define

$$m = \min\{V(y) \mid y \in A = \overline{B}(x_0, r) \setminus B(x_0, \delta_1)\} > 0. \tag{2.2}$$

This minimum, m, exists because $A \subset \mathbb{R}^n$ is closed and bounded, so it is compact. From the existence of m, we can affirm that

$$\exists 0 < \delta_2 < \delta_1 \text{ such that } \forall x \in B(x_0, \delta_2),\ V(x) < m.$$

We choose $x \in B(x_0, \delta_2)$ and take $v = V(x)$. We claim that

$$\forall n \geq 1,\ f^n(x) \in B(x_0, \delta_1) \text{ and } V(f^n(x)) \leq V(x) = v.$$

We are going to prove it by induction:

- $n = 1$:
  By condition (2.1) we have $f(x) \in B(x_0, r)$ and by (2.2) we have $V(f(x)) \leq V(x) = v < m$. This implies that: $f(x) \in B(x_0, \delta_1)$ because if $f(x) \in \overline{B}(x_0, r) \setminus B(x_0, \delta_1)$ it would fulfill that $V(f(x)) \geq m$ and we will have a contradiction. So, we have already proved the result for $n = 1$.

- $n + 1$:
  Suppose that the statement is true for n. Then, $f^{n+1}(x) \in B(x_0, r)$ because (2.1) and $V(f^{n+1}(x)) \leq V(f^n(x)) < m$ because (2.2). This implies that: $f^{n+1}(x) \in B(x_0, \delta_1)$ because if $f^{n+1}(x) \in \overline{B}(x_0, r) \setminus B(x_0, \delta_1)$ it would fulfill that $V(f^{n+1}(x)) \geq m$ and we will have a contradiction. We also have that $V(f^{n+1}(x)) \leq V(x)$.

Finally, we have that:

$$f^n(x) \in B(x_0, \delta_1) \subset B(x_0, r) \subset B(x_0, \epsilon),\ \forall n \geq 1$$

And this implies that the fixed point is Lyapunov stable. □

Definition 2.20. *Let $V : U \subset \mathbb{R}^n :\longrightarrow \mathbb{R}$ be a real valued continuous function. V is a strict Lyapunov function of the system f in a neighborhood U of the fixed point x_0 if:*

1. $V(x_0) = 0$,
2. $V(x) > 0$, $\forall x \in U \setminus \{x_0\}$,
3. $V \circ f(x) < V(x)$, $\forall x \in U \cap f^{-1}(U)$.

Observation 2.21. *If $V : U \longrightarrow \mathbb{R}$ satisfies:*

1. $V(x_0) = C$,
2. $V(x) > C$, $\forall x \in U \setminus \{x_0\}$,
3. $V \circ f(x) < V(x)$, $\forall x \in U \cap f^{-1}(U)$.

for some C, then $\tilde{V}(x) = V(x) - C$ is a strict Lyapunov function.

Theorem 2.22. ***Lyapunov Asymptotic Local Stability Theorem for maps****.*
Let $D \subset \mathbb{R}^n$ be an open set and $f : D \longrightarrow D$ a continuous discrete system. We suppose that there exists $x_0 \in D$ such that $f(x_0) = x_0$. If there exists a strict Lyapunov function V for x_0, then, x_0 is an asymptotically Lyapunov stable fixed point.

Proof. We have already proved the stability of the point in the last theorem. So, we have to prove that $\lim_{n\to\infty} f^n(x) = x_0$, for x in some neighborhood.
Take $r > 0$ such that $\overline{B}(x_0, r) \subset U \cap D$ and $\delta > 0$ such that $\forall x \in B(x_0, \delta) : f^n(x) \in B(x_0, r)$, $\forall n \geq 0$.
We know that $V(f^n(x))$ is a decreasing sequence which is bounded from below. Therefore it has a limit:

$$\lim_{n\to\infty} V(f^n(x)) = \ell \geq 0.$$

Since $\{f^n(x)\} \subset \overline{B}(x_0, r)$, there exists n_k such that $\lim\limits_{n_k\to\infty} f^{n_k}(x) = x_1 \in \overline{B}(x_0, r)$. We suppose that this limit x_1 is not equal to x_0 and we will arrive to contradiction.
We have that $\lim\limits_{n_k\to\infty} V(f^{n_k}(x)) = \ell$, because $f^{n_k}(x)$ is a subsequence of $f^n(x)$. And also, by continuity of V we have that: $\lim\limits_{n_k\to\infty} V(f^{n_k}(x)) = V(\lim\limits_{n_k\to\infty} f^{n_k}(x)) = V(x_1)$. Then, $\ell = V(x_1) > 0$. Now, we have:

$$\lim_{n_k\to\infty} V(f(f^{n_k}(x))) = V(f(x_1)) < V(x_1).$$

And also,

$$\lim_{n_k\to\infty} V(f(f^{n_k}(x))) = \ell.$$

These two statements imply that $\ell < V(x_1)$ and this is a contradiction, which implies that $x_1 = x_0$. Hence all convergent subsequences of $f^n(x)$ converge to x_0.
We have already proved that the fixed point x_0 is asymptotically stable. □

The last two theorems are for a neighborhood of the fixed point x_0, but now, we want to see which are the conditions for the global case.

Theorem 2.23. ***Lyapunov Asymptotic Global Stability Theorem for maps.***
Let $D \subset \mathbb{R}^n$ be open set, $f : D \longrightarrow D$ a continuous discrete system. We suppose that there exist $x_0 \in D$ such that $f(x_0) = x_0$. If there is a strict Lyapunov function V for x_0 defined in D and $\{x \in D \mid V(x) \leq C\}$ is compact $\forall C > 0$, then, x_0 is globally asymptotically Lyapunov stable fixed point, i.e., $\forall x \in D, \lim\limits_{k\to\infty} f^k(x) = x_0$. In other words, the basin of attraction of x_0 is D.

Proof. Let $x \in D$ and $v = V(x)$.
Then $x \in A = \{x \in D \mid V(x) \leq v\}$. The sequence of iterates $\{f^n(x)\}$ is contained in A. We also know that $V(f^n(x))$ is strictly monotone decreasing and $\lim_{n\to\infty} V(f^n(x)) = \ell \geq 0$.
$f^n(x)$ has a convergent subsequence f^{n_k} converging to $x_1 \in A$.
Assume $x_1 \neq x_0$.
We have that $\lim\limits_{n_k\to\infty} V(f^{n_k}(x)) = \ell$, because $f^{n_k}(x)$ is a subsequence of $f^n(x)$. And also, by continuity of V we have that: $\lim\limits_{n_k\to\infty} V(f^{n_k}(x)) = V(\lim\limits_{n_k\to\infty} f^{n_k}(x)) = V(x_1)$. Then, $\ell = V(x_1) > 0$. Now, we have:

$$\lim_{n_k\to\infty} V(f(f^{n_k}(x))) = V(f(x_1)) < V(x_1).$$

And also,

$$\lim_{n_k\to\infty} V(f(f^{n_k}(x))) = \ell.$$

These two statements imply that $\ell < V(x_1)$ and this is a contradiction, which implies that $x_1 = x_0$.
We have already proved that the fixed point x_0 is globally asymptotically stable. $\square$

2.2 Differential equations and Poincaré map

In this section, we are going to introduce some concepts of differential equations theory that will be used in section (5.3). All of the following concepts are in [5] and on the notes of the course *"Equacions Diferencials"*.

Definition 2.24. *Given $f : U \subset \mathbb{R}^n \longrightarrow \mathbb{R}^n$ an autonomous system of differential equations is a system of the form:*

$$\dot{x} = f(x),$$

where t is the independent variable, x is the dependent variable and $\dot{x} = \frac{dx}{dt}$. x is also a vector of state variables. In this setting we say that f is a vector field in U.

Definition 2.25. *A flow is a function*

$$\begin{array}{rccc} \varphi\colon & \Omega \subset \mathbb{R} \times U & \longrightarrow & U \\ & (t, x) & \longmapsto & \varphi(t, x) \end{array}$$

such that,

1. *$\varphi(0, x) = x$,*

2. $\varphi(s, \varphi(t, x)) = \varphi(s + t, x)$.

The solutions of an autonomous system of differential equations generates a flow in some subdomain of $\mathbb{R} \times U$.

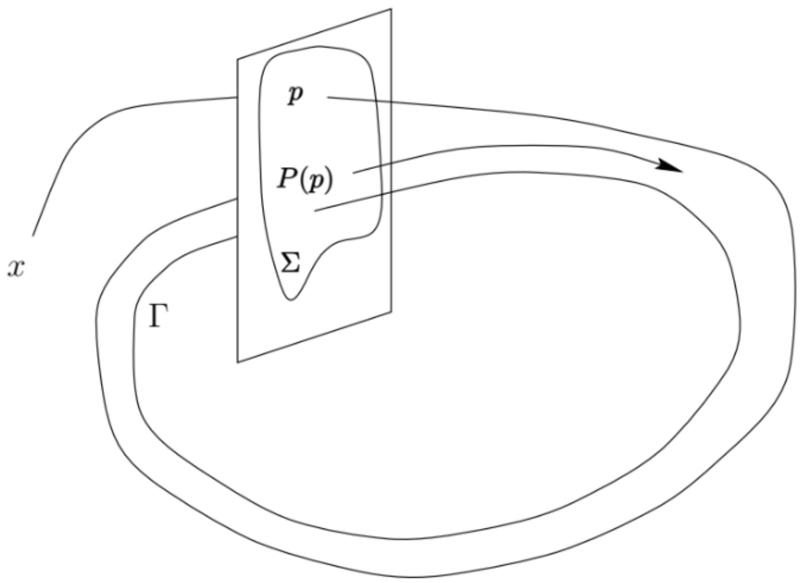


Figure 2.1: Poincaré section, Σ, and Poincaré map, P. x is a point whose trajectory intersects with Σ in point p and returns to the section in $P(p)$. Γ is a periodic orbit. Figure obtained from [5].

Definition 2.26. *A hypersurface* $\Sigma \subset \mathbb{R}^n$ *is a transversal section of a vector field if* $\forall x \in \Sigma$, $T_x M + \langle f(x) \rangle = \mathbb{R}^n$.

Definition 2.27. *A Poincaré section,* Σ*, of a system of differential equations is a* $(n-1)$*-dimensional submanifold such that it is transversal to the flow of the system (see Figure (2.1)).*

Definition 2.28. *A Poincaré map, or return map, is a function* $P : \Sigma \longrightarrow \Sigma$*, such that* $P(x) := \varphi_{T(x)}(x)$*, where* $T(x)$ *is the time of the first return to* Σ *where the solutions crosses* Σ *in the same sense than in* x*. (See Figure (2.1)).*

Basically, the Poincaré map is useful to find periodic orbits of a system of differential equations, because the periodic points of the Poincaré map, are the points of the intersection of the periodic orbits with Σ, i.e., to find periodic orbits we can find points x such that $P(x) - x = 0$, or more generally $P^k(x) - x = 0$.
Another important result is the stability of the periodic orbits:

Definition 2.29. *Let* Γ *be a periodic orbit of a system of differential equations* $\dot{x} = f(x)$*,* P *the corresponding Poincaré map defined on a Poincaré section* Σ *and* $x \in \Gamma \cap \Sigma$*. We say that* Γ *is orbitally stable or asymptotically stable if* x *is stable or asymptotically stable as a fixed point of* P*.*

During the project we use these concepts about Poincaré map and stability of periodic orbits focusing on the similarities of our model with the continuous-time one connecting these concepts with invariant curves of diffeomorphisms.

2.3 Eigenvalues and eigenvectors of a circulant matrix

In this section, we introduce the concept of circulant matrices because we are going to use it in the study of the dynamics of our system, specifically in section (5.1). We also study eigenvalues and eigenvectors of this type of matrices. These results have been taken from [11].

Definition 2.30. *A circulant matrix is a matrix of form $C = (c_{i,j}) \in \mathbb{R}^{n\times n}$ satisfying: $c_{i,j} = c_{(i-j)\, mod\, n}$, i.e.:*

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

Theorem 2.31. *A circulant matrix C diagonalizes, i.e., it has n eigenvalues and n eigenvectors, where eigenvalues are:*

$$\lambda_j = \sum_{k=0}^{n-1} c_k e^{\frac{2\pi ijk}{n}}, \quad j = 0, \dots, n-1,$$

and eigenvectors are:

$$v^{(j)} = \frac{1}{\sqrt{n}}\left(1, e^{\frac{2\pi ij}{n}}, \dots, e^{\frac{2\pi ij(n-1)}{n}}\right), \quad j = 0, \dots, n-1.$$

Proof. First, we are going to calculate eigenvalues and eigenvectors, and once we have them, we will prove that C diagonalizes.
Our eigenvalues and eigenvectors are the solutions of the following system of equations:

$$Cv^{(j)} = \lambda_j v^{(j)}.$$

We can write it in the form:

$$\sum_{k=0}^{n-m-1} c_k v_{k+m}^{(j)} + \sum_{k=n-m}^{n-1} c_k v_{k-(n-m)}^{(j)} = \lambda_j v_m^{(j)}, \quad m = 0, 1, \dots, n-1.$$

Now, choosing $v_k^{(j)} = \frac{\rho^k}{\sqrt{n}}$, for $k = 0, \dots, n-1$, where ρ is a n-th root of unity we have:

$$\sum_{k=0}^{n-m-1} c_k \frac{1}{\sqrt{n}} \rho^{m+k} + \sum_{k=n-m}^{n-1} c_k \frac{1}{\sqrt{n}} \rho^{k-n-m} = \lambda_j \frac{1}{\sqrt{n}} \rho^m, \quad m = 0, 1, \dots, n-1.$$

Simplifying terms and using that $\rho^{-n} = 1$, because ρ is a n-th root of unity, last equation is equal to:

$$\sum_{k=0}^{n-m-1} c_k \rho^k + \sum_{k=n-m}^{n-1} c_k \rho^k = \lambda_j, \quad m = 0, 1, \dots, n-1.$$

So, finally, if $\rho \in \{e^{\frac{2\pi ij}{n}}, j = 0, \ldots, n-1\}$, we have:

$$\lambda_j = \sum_{k=0}^{n-1} c_k e^{\frac{2\pi ijk}{n}},$$

and the corresponding eigenvector is:

$$v^{(j)} = \frac{1}{\sqrt{n}} \left(1, e^{\frac{2\pi ij}{n}}, \ldots, e^{\frac{2\pi ij(n-1)}{n}}\right).$$

Now, we have to prove that the eigenvectors are linearly independent vectors, this means that there are n eigenvectors and consequently our matrix diagonalizes. To prove this, we are going to prove that our matrix:

$$A = (v^{(0)} \mid v^{(1)} \mid \cdots \mid v^{(n-1)}),$$

is invertible. For this, it is sufficient to find a good candidate inverse matrix of A, we call it B, and prove that $BA = Id$.
Let B be the following matrix:

$$B = (v^{(0)} \mid v^{(n-1)} \mid v^{(n-2)} \mid \cdots \mid v^{(1)})^T.$$

Since the product has to be the identity matrix, we will split the problem into two cases. One for the diagonal elements, which have to be one, and the rest of the elements that have to be zero. Now, we will see that this is fulfilled:

$$v^{(n-k)} v^{(k)} = \frac{1}{n} \sum_{q=0}^{n-1} e^{\frac{2\pi i(n-k)q}{n}} e^{\frac{2\pi ikq}{n}} = \frac{1}{n} \sum_{q=0}^{n-1} e^{2\pi iq} = 1,$$

because of the Euler's formula. And, now:

$$v^{(l)} v^{(j)} = \frac{1}{n} \sum_{q=0}^{n-1} e^{\frac{2\pi ilq}{n}} e^{\frac{2\pi ijq}{n}} = \frac{1}{n} \sum_{q=0}^{n-1} e^{\frac{2\pi i(l+j)}{n} q}.$$

Since $l + j \neq n$, we have that $e^{\frac{2\pi i(l+j)}{n}}$ is a n-root of unity and it is different to 1. We call it μ. It is obvious that $\mu^n = 1$.
From the last two lines, we have:

$$\sum_{k=0}^{n-1} \mu^k = 1 + \mu + \mu^2 + \cdots + \mu^{n-1} = \frac{\mu^n - 1}{\mu - 1} = 0.$$

The second equality is easy to see if we multiply both sides by $\mu - 1$. Then, we have the result desired. $\square$

Chapter 3

What is the Hypercycle?

In this chapter, we introduce the hypercycle, both biologically and mathematically. In the first section, we will review the hypercycle theory: why this new system was conceived, why it its importance in biology and, above all, what is its connection with the origin and evolution of life. In the next section, we will first introduce a differential equations model for the hypercycle and we will show the analogue discrete equation. Finally, we will discuss the importance of choosing a discrete equation for our model instead of directly studying the differential equations system.

3.1 Biological concept and its importance

The concept of the Hypercycle appears in 1977, being introduced by Manfred Eigen and Peter Schuster [7]. These scientists asked themselves the following question: *"Why do millions of species, plants and animals, exist, while there is only one basic molecular machinery of the cell: one universal genetic code and unique chiralities of the macromolecules?"* [7].

The hypercycle is a catalytic set of macromolecules, where each molecule or replicator catalyze the replication of the next species of the set (see Figure (3.1) and (3.2)). Precisely, the problem of the information crisis can be solved with this new model of organization, because it is a cooperative system, matter that we deal with later. In the hypercycle, each element is a sequence since all species cooperate and coexist below the critical length imposed by the error threshold. Hence, the information content of the whole system can overcome the error threshold [22, 7, 29, 23, 9, 24].
Several experiments have been carried out to confirm the existence of this type of structures in real replicators. In the following lines we comment on the most important results for these systems:

Peptide self-replication:
This system is a minimum hypercycle based on two self-replicative peptides, rolled up in a helix, R1 and R2 [18]. The replicator R1 is produced as an interaction between a fragment of the electrophilic peptide E and a nucleophilic peptide fragment N1. The replicator R2

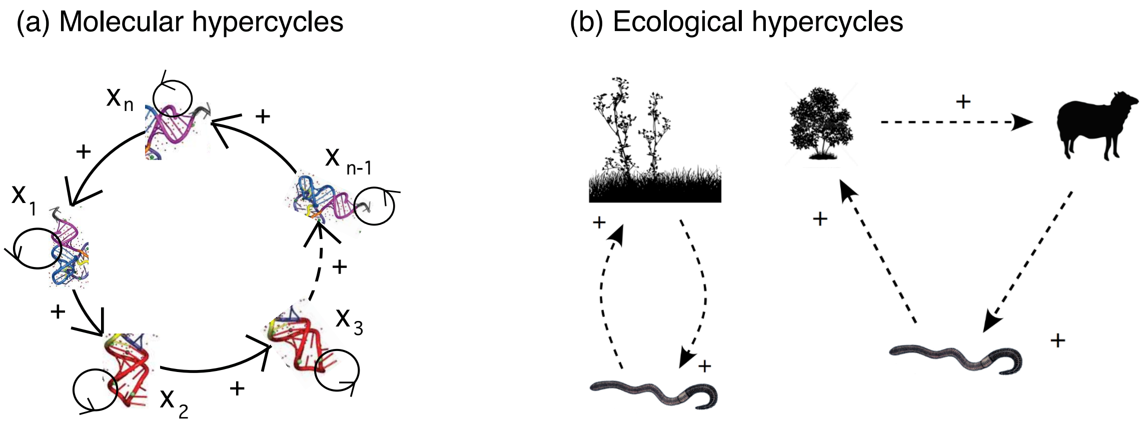


Figure 3.1: Schematic diagram of molecular (a) and ecological (b) hypercycles. Hypercycles are catalytic networks with cyclic structure, in which the species replicate or reproduce with the aid of the previous partner. Molecular hypercycles have been suggested to play a key role in the origins of life scenario, for which ribozymes might have organized hypercyclically to increase the information content. Ecological hypercycles in (b) display the positive feedbacks that introduce non-linearity due to processes of cooperation e.g., mutualism, facilitation between different species.

is made of with a fragment of E and a fragment of a different nucleophilic peptide N2. They created a solution of three fragments, E, N1 and N2. A priori, they thought that it will be a survival situation of the most competent specie, where R2 moves R1, because R2 would consume the common fragment E fastest. At first sight, it seems to them that this hypothesis was accomplished, because it seems that R2 was produced with more abundance than R1. Therefore, they decided to do a new solution, adding this time a 40% more of replicator R1 (in relation to the concentration of the nucleophilic fragments) and what happened was surprising. At the beginning of the reaction, the self-replicating rate of R1 increased a 1.7 more, but the formation rate of the replicator R2 improved in 5.4 its self-replicating rate. This fact made that they concluded that the two replicators were not mutually selective in their growth. R1 catalyzed the formation of R2 and also its formation. Finally, they decided to change the reaction another time, adding 45% more of the replicator R2 and they saw that not only the production rate of R2 has increased in 2.9, even R1 also increased its production rate in 3.5. Clearly, replicators were cooperating and they were acting as a catalysts.

Yeast:

In this experiment, the authors performed a study where they proved that it was possible to create a cooperation between two populations of yeast non interaction between them a priori. To do so, they introduced these two populations with a specific genetic modification each one, that consisted in removing different essential nutrients of each population. They realized that populations exchanged this removed essential nutrients between them, i.e., each species of yeast provide other species of yeast with the nutrient that it was extracted. They also observed that cooperation was not very deep, but they demonstrated that even the system was a synthetic simplified cooperative system, with new properties, such as a higher capacity of self-maintainance [27].

E. Coli:
The scientists involved in this experiment decided to do a experiment with two bacterial strains engineered to exchange some essential amino acids, similarly to the experiments with yeast described above. For this case one of the strains **I** could not produce the essentia amino acid isoleucine (iso), but it overproduced leucine (leu). The other strain **L** could not produce leu but overproduced iso. Therefore, the strains were able to exchange these essential amino acids and they could participate in a mutualism allowing growth. Later, they introduced a new replicator called parasite, **P**, that took advantage of the limited resources in the medium, will restrict the growth of the coupled system. This parasite exploited one of the amino acid, iso. The coculture of these three organisms under good conditions gave as a result a strict growth of the strains **I** and **L** [1].

3.2 Criticisms for the Hypercycle

When the concept of the hypercycle appeared, it arose many doubts and criticisms in the scientific community about the problems that this type of organization could have. Some of these problems would be the appearance of short circuits and parasites [22, 14, 28].

In the case of the short circuits, it means that in the hypercycle, appears interactions between the next species, there are interactions between further molecules in the structure [22]. We will give a visual example to understand this concept. Suppose that we have an hypercycle with three molecules I1, I2 and I3, where each one catalyzes the reproduction of the next molecule, but one of them, for example I2 also catalyzes the reproduction of the first molecule, I1 (see Figure (3.2b)). Clearly, inside this hypercycle of three catalytic species, appears a smaller hypercycle with two molecules, I2 and I1 (the short circuit). In the moment that appears this smaller hypercycle it is obviously to think that this set of two molecules will be the hypercycle that grows faster and in consequence the set that will survive, because all can tend to it and the third molecule, I3 could disappear. Therefore, consequently, the larger hypercycle contracts to a less complex form, losing information.

In the case of the parasites, it means that in the set there are macromolecules that receive help from another catalytic species of the set, but they do not act as a catalyst of other species [22, 14, 28]. An easy example could be also an hypercycle with three molecules I1, I2 and I3, but in this case only two molecules have a cyclic structure, i.e., I1 catalyzes the replication of I2 and I2 catalyzes the replication of I1, but, moreover, I2 also catalyzes the replication of I3, and this last molecule do not act as a catalyst of other molecules of the system (see Figure (3.2b)). It is obvious to think that, in time, parasite will dominate, I3, and that I3 will be the unique species that will survive and I1 and I2 would disappear. This problem about the existence of parasites has been studied a lot, first introduced by Boerlijst i Hogeweg. They demonstrated that a hypercycle spatially-extended can self-organizate spatially for reject the parasites that could have in the hypercycle [3].

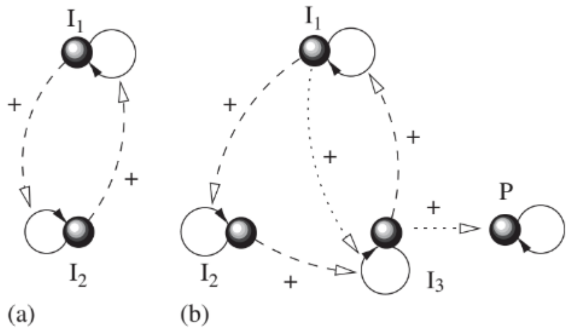


Figure 3.2: (a) Two-member hypercycle. (b) Three-member hypercycle. Catalytic interactions are represented with dashed arrows, curved solid arrows show replicator self-replicating activity. Shortcuts or parasites, P, are indicated with dotted arrows. Figure obtained from [24].

Another concern about the hypercycle, related with the apparition of short circuits, is the problem with the size of the hypercycle. The scientists saw that in the moment that the hypercycle had a high number of macromolecules, it could suppose a problem, because it would be easier to apparition of short circuits [22].

Finally, there are the large fluctuations, i.e., it could be possible that the concentration of one of the elements in the hypercycle collapses to zero. The survival of the hypercycle depends on the existence of all replicators of the set, therefore if this will happen, the whole hypercycle collapse. This happens especially in hypercycles with oscillations, in EDOs, when there are 5 or more species [12].

3.3 Hypercycle with time-discrete dynamics

In this section we will introduce the hypercycle system investigated in this project. To start with, we will study a simple hypercycle taking into account the hetero-catalytic interactions, i.e., it does not take into account interactions between species, beyond the interaction with the previous species. We will specifically focus in a system investigated in [13], building a discrete-time hypercycle model. Before doing so, we will introduce the time-continuous system, which has been widely studied.

A time-continuous, mathematical model for the hypercycle could be the following system of ordinary differential equations:

$$\dot{x}_i = x_i(k_i x_{i-1} - \phi(x)), \quad \text{for } i = 1, ..., n, \tag{3.1}$$

where x_i denotes the concentration of the i-th species, k_i are the kinetic constants that denote the strength of the catalysis of the $i-1$ species on the growth of the i species.
ϕ is a term that keeps the total population constant and introduces competition between all the hypercycle members. Which is the value of ϕ? The fact that the population has to be constant is expressed by: $\sum_{i=1}^{n} x_i = ctant$ (In our case, we suppose that: $\sum_{i=1}^{n} x_i = 1$, to make calculations easier). Then, considering this, it has to be fulfilled: $\sum_{i=1}^{n} \dot{x}_i = 0$.

Therefore:

$$\sum_{i=1}^{n} \dot{x}_i = \sum_{i=1}^{n} x_i(k_i x_{i-1} - \phi(x)) = 0 \Longleftrightarrow \sum_{i=1}^{n} k_i x_i x_{i-1} - \phi(x) \sum_{i=1}^{n} x_i = 0 \Longleftrightarrow \phi(x) = \sum_{i=1}^{n} k_i x_i x_{i-1}.$$

Note, also, that the subindexes i are modulo n, i.e., $x_0 = x_n, x_1 = x_{n+1}, ...$
Another observation is that the ratio of growth $\frac{\dot{x}_i}{x_i}$ is proportional to the concentration of the preceding species x_{i-1}.
We will consider our system on the following n-simplex:

$$S_n = \{x = (x_1, ..., x_n) \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i = 1 \text{ and } x_i \geq 0 \text{ for } i = 1, \cdots, n\}$$

This system of differential equations which has been widely studied by many scientists, was firstly studied by Eigen and Schuster [7].

Now, we want to have a discrete system to model the hypercycle. So, we need a function:

$$\begin{array}{rccl} F(x): & \mathbb{R}^n & \longrightarrow & \mathbb{R}^n \\ & x = (x_1, \cdots, x_n) & \longmapsto & F(x) = (F_1(x), \cdots, F_n(x)) \end{array}$$

with $F_i(x)$ being the value of x_i in the next generation, i.e., $F_i(x)$ represents one step on time. First, $F_i(x)$ has a term proportional to x_i and a term proportional to the product $x_i x_{i-1}$, taking into account that the $(i-1)$-th species contributes to the i-th one. This means that the reproduction function is non-linear:

$$F_i(x) \sim x_i(C + k_i x_{i-1}).$$

This proportionality is equivalent to:

$$F_i(x) = A x_i(C + k_i x_{i-1}),$$

where A will be determined next. We impose the total population to be constant, say equal to one. So if $\sum_{i=1}^{n} x_i = 1$,

$$\sum_{i=1}^{n} F_i(x) = 1 \Longleftrightarrow \sum_{i=1}^{n} A x_i(C + k_i x_{i-1}) = 1 \Longleftrightarrow A \sum_{i=1}^{n} x_i(C + k_i x_{i-1}) = 1 \Longleftrightarrow$$

$$\Longleftrightarrow A(\sum_{i=1}^{n} x_i C + \sum_{i=1}^{n} x_i k_i x_{i-1}) = 1 \Longleftrightarrow A(C + \phi(x)) = 1 \Longleftrightarrow A = \frac{1}{C + \phi(x)}.$$

Therefore, we have the following discrete system:

$$F_i(x) = \frac{C + k_i x_{i-1}}{C + \phi(x)} x_i, \quad i = 1, \cdots, n \quad \text{with} \quad \phi(x) = \sum_{i=1}^{n} k_i x_i x_{i-1} \quad \text{and } C > 0. \tag{3.2}$$

This equation was introduced by Hofbauer [13].

We rewrite the i-th equation, as follows:

$$F_i(x) - x_i = \frac{C + k_i x_{i-1}}{C + \phi(x)} x_i - x_i \Longleftrightarrow F_i(x) - x_i = \frac{C + k_i x_{i-1}}{C + \phi(x)} x_i - \frac{C + \phi(x)}{C + \phi(x)} x_i$$

$$\Longleftrightarrow F_i(x) - x_i = \frac{k_i x_{i-1} - \phi(x)}{C + \phi(x)} x_i \Longleftrightarrow \frac{F_i(x) - x_i}{C^{-1}} = x_i(k_i x_{i-1} - \phi(x)) \frac{C}{C + \phi(x)}.$$

This transformation of (3.2) permits us to interpret C^{-1} as the time interval between two generations. If we think of $x_i = x_i(t)$ and:

$$F_i(x)(t) = x_i(t + C^{-1}),$$

using the definition of the derivative, we have:

$$\lim_{C^{-1} \to 0} \frac{x_i(t + C^{-1}) - x_i(t)}{C^{-1}} = \lim_{C \to \infty} \frac{x_i(t + C^{-1}) - x_i(t)}{C^{-1}} = \lim_{C \to \infty} \frac{F_i(x) - x_i}{C^{-1}}$$

$$= \lim_{C \to \infty} x_i(k_i x_{i-1} - \phi(x)) \frac{C}{C + \phi(x)} = x_i(k_i x_{i-1} - \phi(x)).$$

The last step follows because $\phi(x)$ is bounded and hence:

$$\lim_{C \to \infty} \frac{C}{C + \phi(x)} = 1.$$

Then, expression (3.2) approximates the differential equation (3.1) for $C \to \infty$. Therefore, (3.2) is a good discrete system analogue to (3.1) and for large values of C, the discrete system will have similar properties to those of the differential equation. Finally, we obtain the discrete-time hypercycle model that will be analyzed in this project.

We considered that it would be interesting to study the discrete-time model because there are a lot of investigations about the continuous-time model, but not many about the hypercycle as a discrete-time model.

With continuous-time models we have the number of events per time unit, it means that the generations overlap and the events of birth and death can occur at any time. However, in discrete-time models there are specific moments in which the events of our system can occur and it is not necessary that in these moments only a single event occurs. Discrete-time models are very important in biological models, mostly in population models or in ecological models. Particularly, this type of structure is very useful in ecological models in which species cooperate via mutualistic symbiosis [25] (see Figure (3.1)).

Chapter 4

Cooperation on the Hypercycle

In this section, we will show the cooperation in our model of the hypercycle. In terms of biology, cooperation means that none species of the system dies. From the opposite point of view, the absence of cooperation is called exclusion, i.e., at least one specie of the system will die.

First, we need to give these two important formal definitions:

Definition 4.1. *A dynamical system F defined on the n-simplex S_n is cooperative if the boundary of S_n (∂S_n) is a repellor, i.e., if $\exists \delta > 0$ such that $\forall x \in S_n \setminus \partial S_n$, $\liminf_{k\to\infty} \|F^k(x)\| \geq \delta$.*

Definition 4.2. *The contrary behaviour to cooperation is exclusion, this happens when some orbits go to the boundary and at least one species dies out.*

We recall our hypercycle model:

$$F_i(x) = \frac{C + k_i x_{i-1}}{C + \phi(x)} x_i, \quad i = 1, \cdots, n, \quad \text{with } \phi(x) = \sum_{i=1}^{n} k_i x_i x_{i-1} \quad \text{and } C > 0. \tag{4.1}$$

We write $F^{(N)}(x)$ as the result of iterating (4.1) N times the initial value $F^{(0)}(x) = x$. We will prove the following theorem:

Theorem 4.3. *If we have $C > 0$ and $k_i > 0$ for $i = 1, \cdots, n$, then our system is cooperative.*

Proof. 4.3. We have to prove that, under the theorem conditions, our system is cooperative, i.e., just like we say in the definition (4.1):

$$\exists \delta > 0 \text{ such that } \liminf_{N\to\infty} F_i^{(N)}(x) \geq \delta, \quad \forall i = 1, \cdots, n \quad \text{and} \quad x \in intS_n.$$

We begin by discussing the behaviour of our system on the boundary of S_n:

Lemma 4.4. $\forall x \in \partial S_n$, *it is satisfied that* $\phi(F^{(N)}(x)) \to 0$ *for* $N \to \infty$.

Proof. Let $x \in \partial S_n$. First, we have to determine which is the form of x. As x is in the boundary, some of their components will be 0, but not all. Actually, $\exists i$ such that $x_i = 0, x_{i+1} > 0, \dots, x_{i+k} > 0, x_{i+k+1} = 0$. Let k be the number of consecutive components different of zero, with $1 \leq k < n$, i.e., k is at most $n-1$. To make the problem easy, we can set $i = 0$, because of the cyclic symmetry, and $k_i = 1, \forall i$, the proof also works for arbitrary constants k_i. Therefore, with this simplification of notation, we have: $x_0 = 0, x_j > 0, x_{k+1} = 0,$ with $1 \leq j < k$.

Proposition 4.5. $F_j^{(N)}(x) \to 0$, *and* $\frac{F_j^{(N)}(x)}{F_{j+1}^{(N)}(x)}$ *converges monotonically (asymptotically) to* $\ell_j \in [0, \infty) \cup \{\infty\}$.

Proof. First, we will prove the second part of the statement by induction on j:
The statement is true for $j = 1$:

$$\left.\begin{array}{l} F_0^{(0)}(x) = 0 \Rightarrow F_0^{(N)}(x) = 0 \\ F_1^{(0)}(x) > 0 \Rightarrow F_1^{(N)}(x) > 0 \end{array}\right\} \Rightarrow \frac{F_0^{(N)}(x)}{F_1^{(N)}(x)} = 0 \Rightarrow \frac{F_0^{(N)}(x)}{F_1^{(N)}(x)} \to 0.$$

Now, we assume that is true for j, i.e., $\frac{F_j^{(N)}(x)}{F_{j+1}^{(N)}(x)} \to \ell_j$, monotonically. We have to prove that it is also true for j+1. To prove it we distinguish several cases. Before it, we have the following recursive relations:

$$\frac{F_{j+1}^{(2)}(x)}{F_{j+2}^{(2)}(x)} = \frac{F_{j+1}(F_{j+1}(x))}{F_{j+2}(F_{j+2}(x))} = \frac{C + F_j(x)}{C + F_{j+1}(x)} \frac{F_{j+1}(x)}{F_{j+2}(x)},$$

$$\frac{F_{j+1}^{(3)}(x)}{F_{j+2}^{(3)}(x)} = \frac{F_{j+1}(F_{j+1}^{(2)}(x))}{F_{j+2}(F_{j+2}^{(2)}(x))} = \frac{C + F_j^{(2)}(x)}{C + F_{j+1}^{(2)}(x)} \frac{F_{j+1}^{(2)}(x)}{F_{j+2}^{(2)}(x)},$$

$$\vdots$$

$$\frac{F_{j+1}^{(N+1)}(x)}{F_{j+2}^{(N+1)}(x)} = \frac{F_{j+1}(F_{j+1}^{(N)}(x))}{F_{j+2}(F_{j+2}^{(N)}(x))} = \frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)} \frac{C + F_j^{(N)}(x)}{C + F_j^{(N)}(x)}.$$

Case 1. $\ell_j > 1$

$$\frac{F_j^{(N)}(x)}{F_{j+1}^{(N)}(x)} \to \ell_j > 1 \Rightarrow \frac{C + F_j^{(N)}(x)}{C + F_{j+1}^{(N)}(x)} \to \ell_j > 1.$$

Then, $\exists N_0$ such that $\forall N \geq N_0$: $\frac{C + F_j^{(N)}(x)}{C + F_{j+1}^{(N)}(x)} > 1$. So,

$$\frac{F_{j+1}^{(N+1)}(x)}{F_{j+2}^{(N+1)}(x)} \geq \frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)}.$$

Therefore, $\frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)}$ is an increasing function and it converges to some limit $\ell_{j+1} \geq 0$.

Case 2. $\ell_j = 1$ and $\frac{F_j^{(N)}}{F_{j+1}^{(N)}} \searrow 1$

$$\frac{F_j^{(N)}(x)}{F_{j+1}^{(N)}(x)} \searrow \ell_j = 1 \Rightarrow \frac{C + F_j^{(N)}(x)}{C + F_{j+1}^{(N)}(x)} \searrow \ell_j = 1.$$

Then, $\frac{C+F_j^{(N)}(x)}{C+F_{j+1}^{(N)}(x)} > 1$. So,

$$\frac{F_{j+1}^{(N+1)}(x)}{F_{j+2}^{(N+1)}(x)} > \frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)}.$$

Therefore, $\frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)}$ is an increasing function and it converges to some limit $\ell_{j+1} \geq 0$.

Case 3. $\ell_j < 1$

$$\frac{F_j^{(N)}(x)}{F_{j+1}^{(N)}(x)} \to \ell_j < 1 \Rightarrow \frac{C + F_j^{(N)}(x)}{C + F_{j+1}^{(N)}(x)} \to \ell_j < 1.$$

Then, $\exists N_0$ such that $\forall N \geq N_0$: $\frac{C+F_j^{(N)}(x)}{C+F_{j+1}^{(N)}(x)} < 1$. So,

$$\frac{F_{j+1}^{(N+1)}(x)}{F_{j+2}^{(N+1)}(x)} < \frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)}.$$

Therefore, $\frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)}$ is a decreasing function and it converges to some limit $\ell_{j+1} \geq 0$.

Case 4. $\ell_j = 1$ and $\frac{F_j^{(N)}}{F_{j+1}^{(N)}} \nearrow 1$

$$\frac{F_j^{(N)}(x)}{F_{j+1}^{(N)}(x)} \nearrow \ell_j = 1 \Rightarrow \frac{C + F_j^{(N)}(x)}{C + F_{j+1}^{(N)}(x)} \nearrow \ell_j = 1.$$

Then, $\frac{C+F_j^{(N)}(x)}{C+F_{j+1}^{(N)}(x)} < 1$. So,

$$\frac{F_{j+1}^{(N+1)}(x)}{F_{j+2}^{(N+1)}(x)} < \frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)}.$$

Therefore, $\frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)}$ is a decreasing function and it converges to some limit $\ell_{j+1} > 0$.

We have proved the second part of the statement. Now we are going to prove the first part. We also distinguish two cases:

Case 1. $\ell_j > 0$:

We know that $F_j^{(N)}(x) \to 0$, and $\frac{F_j^{(N)}(x)}{F_{j+1}^{(N)}(x)}$ converges monotonically (asymptotically) to $\ell_j > 0$. Then, $F_{j+1}^{(N)}(x)$ has to converge to 0 too.

Case 2. $\ell_j = 0$:
In this case, we assume that $F_{j+1}^{(N)}(x) \nrightarrow 0$ and we will realize that there is a contradiction. So, we have:

$$F_{j+1}^{(N)}(x) \nrightarrow 0 \Rightarrow \exists \text{ a subsequence } F_{j+1}^{(N_m)}(x) \subset F_{j+1}^{(N)}(x) \text{ such that } F_{j+1}^{(N_m)}(x) > \eta > 0.$$

We also know that $F_j^{(N)}(x) \to 0$, this implies that every subsequence of $F_j^{(N)}(x)$ tends to 0.
So, we have:

$$\frac{C + F_j^{(N_m)}(x)}{C + F_{j+1}^{(N_m)}(x)} < \frac{C}{C+\eta}.$$

For this, we know that:

$$\frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)} = \frac{C + F_j^{(N-1)}(x)}{C + F_{j+1}^{(N-1)}(x)} \frac{C + F_j^{(N-2)}(x)}{C + F_{j+1}^{(N-2)}(x)} \cdots \frac{C + F_j(x)}{C + F_{j+1}(x)} \frac{C + x_j}{C + x_{j+1}} \frac{x_{j+1}}{x_{j+2}}$$

$$= \frac{x_{j+1}}{x_{j+2}} \prod_{k=0}^{N-1} \frac{C + F_j^{(k)}(x)}{C + F_{j+1}^{(k)}(x)}.$$

this quotient tends to 0 when $N \to \infty$.
Now, $F_{j+2}(x)^{(N)} \le 1$, because it is in the simplex S_n. Therefore $F_{j+1}^{(N)}(x) \le \frac{F_{j+1}^{(N)}(x)}{F_{j+2}^{(N)}(x)} \to 0 \Rightarrow$ $F_{j+1}^{(N)}(x) \to 0$. So, there is a contradiction between this result and the initial supposition. So, this concludes the proof of Proposition 4.5. □

Now, we have that $\forall x \in \partial S_n$, $F^{(N)}(x)$ converges to a subface of the boundary where if $x_j > 0$ then $x_{j+1} = 0$. On such a face we have $\phi(x) = 0$. Note that the set $\{x \mid \phi(x) = 0\}$ is the set of the fixed points in the boundary. So, we have the following claim:

Claim 4.6. $\Gamma = \{x \mid \phi(x) = 0\}$ *is the set of the fixed points of* F *in the boundary.*

Proof. 4.6. We have to prove two implications.
$\Leftarrow$: On the one hand, we have to prove: $x \in \partial S_n$ such that $F(x) = x \Rightarrow \phi(x) = 0$.

$$x \in \partial S_n \text{ such that } F(x) = x \Leftrightarrow x_i = x_i \frac{C + k_i x_{i-1}}{C + \phi(x)}, \forall i.$$

We know that

$$\left.\begin{array}{l} \sum x_i = 1 \\ x \in \partial S_n \end{array}\right\} \Rightarrow \exists l \text{ such that } x_l = 0 \text{ and } x_{l+1} \neq 0.$$

Now,

$$F(x) = x \Rightarrow \begin{cases} x_i = 0, \\ \text{or} \\ \frac{C+k_i x_{i-1}}{C+\phi(x)} = 1, \end{cases}$$

From the condition $\exists l$ such that $x_l = 0$ and $x_{l+1} \neq 0$, we have:

$$x_l = 0 \Rightarrow \frac{C + k_{l+1} x_l}{C + \phi(x)} = \frac{C}{C + \phi(x)} = 1 \Rightarrow \phi(x) = 0.$$

$\Rightarrow$: On the other hand, we have to prove: $\phi(x) = 0 \Rightarrow F(x) = x$.

$$\phi(x) = 0 \Rightarrow \sum k_i x_i x_{i-1} = 0 \Rightarrow x_i x_{i-1} = 0, \forall i \Rightarrow F_i(x) = x_i \frac{C + k_i x_{i-1}}{C}$$

$$= x_i + \frac{k_i x_i x_{i-1}}{C} = x_i \ , \forall i \Rightarrow F(x) = x.$$

With this, we have proved that Γ is the set of fixed points of F on the boundary. $\square$

This finishes the proof of Lemma 4.4. $\square$

Now, we consider the following function: $P(x) = x_1 x_2 \cdots x_n$. Notice that $P(x) = 0 \Leftrightarrow x \in \partial S_n$.

Thus, $P(x)$ is a measure of the distance from x to the boundary of the simplex S_n. Let us consider the set:

$$I(p) := \{x \in \mathring{S}_n \mid P(x) \leq p\} \text{ with } p > 0.$$

Keeping in mind our definition of cooperation, we know that it is equivalent to the existence of a layer $I(p)$ which is left for ever after some time by each orbit starting in $\mathring{S}_n$. We have to note that,

$$\frac{P(F(x))}{P(x)} = \frac{F_1(x)F_2(x)\cdots F_n(x)}{x_1 x_2 \cdots x_n} = \frac{\prod_{i=1}^{n}(C + k_i x_{i-1}) \prod_{i=1}^{n} x_i}{(C + \phi(x))^n \prod_{i=1}^{n} x_i} = \frac{\prod_{i=1}^{n}(C + k_i x_{i-1})}{(C + \phi(x))^n}$$

$$\geq \frac{C^n + C^{n-1} \sum_{i=1}^{n} k_i x_{i-1}}{(C + \phi(x))^n}.$$

This quotient is positive on S_n and greater than 1 on the set of fixed points Γ. So, $\frac{P(F(x))}{P(x)} > 1$ on Γ. Now, S_n is a compact set, because we are in $\mathbb{R}^n$ and a set that is closed and bounded is a compact set. So, we have the following statements:

$$\exists m > 0 \text{ such that } \frac{P(F(x))}{P(x)} \geq m > 0 \ , \forall x \in S_n. \tag{4.2}$$

$$\frac{P(F(x))}{P(x)} \geq M > 1 \text{ in some neighbourhood } A \text{ of the set } \Gamma. \tag{4.3}$$

We choose $r \in (0, 1)$ such that $K := M^r m^{1-r} > 1$.

Definition 4.7. *For each $x \in \partial S_n$ we define $N(x)$ as the smallest number $N \geq 1$ such that:*

$$\frac{card\{0 \leq k < N \mid F^{(k)}(x) \in A\}}{N(x)} \geq r.$$

Less formally, this means that the probability that $F^{(k)} \in A$ for $k < N(x)$ is at least r.

Observation 4.8. *Obviously, $N(x) = 1 \Leftrightarrow x \in A$.*

Observation 4.9. *We also know from Lemma 4.4 that $\forall x \in \partial S_n$, $F^{(k)}(x) \in A$, for k large enough. This guarantees the existence of $N(x)$, for each $x \in \partial S_n$.*

Lemma 4.10. *The function $x \mapsto N(x)$ can be extended to some neighbourhood $I(p)$ of the boundary.*

Proof.

$$\begin{array}{rrcl} F(x): & S_n & \longrightarrow & S_n \\ & x = (x_1, \cdots, x_n) & \longmapsto & F(x) = (F_1(x), \cdots, F_n(x)) \end{array}$$

is a continuous function, and thus $\forall N, F^{(N)}(x)$,$\forall n$ is also a continuous function, because it is a composition of continuous functions. Now, we know that $\forall x \in \partial S_n$, $\exists k < N(x)$ such that $F^{(k)}(x) \in A$, because the last observation. Then, by continuity, $\exists U_k$ an open neighbourhood of x such that $\forall y \in U_k$, $F^{(k)}(y) \in A$. So, if we have $y \in U(x) = \bigcap U_k$, where U_k are the neighbourhoods of all x for $k < N(x)$, we have that $F^{(k)}(y) \in A$, whenever $F^{(k)}(x) \in A$, and therefore $N(y) = N(x)$. On the other hand, if y belongs to a smaller intersection, i.e., of some U_k (considering the sets of a sequence), we would have that $F^{(k)}(y) \in A$, but the number $N(y)$ would be smaller than $N(x)$. Thus, we have that $N(y) \leq N(x)$.
Now, if $\forall x \in \partial S_n$ we have a set $U(x)$, then the family of sets $U(x)$ is an open covering of ∂S_n and since S_n is a compact set, ∂S_n is also compact. So, $\exists$ a finite subcover of these $U(x)$ that covers ∂S_n.
The union of these subcovers contain a layer $I(p)$ for some $p > 0$. Furthermore, $N(x)$ is bounded on $I(p)$ for a certain number $\overline{N}$, where $\overline{N} = \max N(x)$ since for each $U(x)$ we have a finite number $N(x)$.
This proves Lemma 4.10. □

The last step is to prove Theorem 4.3. We need to do it into two parts. Hence, we have decided to divide it into the following two claims:

Claim 4.11. *If $x \in I(p)$, then $\exists N$ with $F^{(N)}(x) \notin I(p)$, i.e., any orbit starting in $I(p)$ leaves this layer after some time.*

Proof. We will prove this claim by contradiction. So, suppose that $\forall N$, $F^{(N)}(x) \in I(p)$, i.e., $d := \sup_{k \geq 0} P(F^k(x)) \leq p$. Now, we choose N_1 such that $y = F^{(N_1)}(x)$ satisfies $P(y) > \frac{d}{\sqrt{K}}$.

Using the definition of $N(y)$ and the statements 4.2 and 4.3 we have:

$$\frac{P(F^{(N(y))}(y))}{P(y)} = \frac{P(F^{(N(y))}(y))}{P(F^{(N(y)-1)}(y))}\frac{P(F^{(N(y)-1)}(y))}{P(F^{(N(y)-2)}(y))}\cdots\frac{P(F^{(2)}(y))}{P(F(y))}\frac{P(F(y))}{P(y)}$$

$$= \prod_{k=0}^{N(y)-1}\frac{P(F^{(k+1)}(y))}{P(F^{(k)}(y))} \geq M^{rN(y)}m^{(1-r)N(y)} = K^{N(y)} \geq K. \tag{4.4}$$

This implies:

$$P(F^{(N(y))}(y)) \geq KP(y) > K\frac{d}{\sqrt{K}} = d\sqrt{K} > d.$$

This contradicts the definition of d. So, we have already proved the claim. □

Claim 4.12. *There exists $q < p$ such that the layer $I(q)$ is never reached for large times, i.e., if $x \notin I(p)$, then $\nexists N \geq 0$ such that $F^{(N)}(x) \in I(q)$.*

Proof. Let $\overline{N} = \max\{N(x) \text{ s.t. } x \in I(p)\}$ and we choose $q = pm^{(1-r)\overline{N}+1}$.
Suppose that $x \notin I(p)$ and we also choose N_2 to be the first iteration such that $z = F^{(N_2)}(x) \in I(p)$. So, it is true that:

$$\frac{P(F^{(N_2)}(x))}{P(F^{(N_2-1)}(x))} > m \text{ and } P(F^{(N_2-1)}(x)) > p.$$

Then, it is fulfilled that $P(z) > mp$.
By the definition of $N(z)$, we have:

$$card\{0 \leq k < N \mid F^{(k)}(z) \notin A\} \leq (1-r)N(z) \leq (1-r)\overline{N}.$$

So, using the same argument as in claim 4.11, we have:

$$\frac{P(F^{(k)}(z))}{P(z)} = \frac{P(F^{(k)}(z))}{P(F^{(k-1)}(z))}\frac{P(F^{(k-1)}(z))}{P(F^{(k-2)}(z))}\cdots\frac{P(F^{(2)}(z))}{P(F(z))}\frac{P(F(z))}{P(z)}$$

$$= \prod_{i=0}^{k-1}\frac{P(F^{(i+1)}(z))}{P(F^{(i)}(z))} \geq m^{(1-r)\overline{N}}.$$

Therefore, we have that:

$$P(F^{(k)}(z)) \geq m^{(1-r)\overline{N}}P(z) > m^{(1-r)\overline{N}}mp = m^{(1-r)\overline{N}+1}p = q$$

is true $\forall k \leq N(z)$, i.e., $F^{(k)}(z) \notin I(q)$ for $k \leq N(z)$. Now, we want to prove it also for $k = N(z)$. Using the bound (4.4), we have that:

$$P(F^{(N(z))}(z)) \geq KP(z).$$

This means that after $N(z)$ iterations, $P(z)$ has increased by the factor $K > 1$. We also know that orbits $F^{(k)}(z)$ never enter the layer $I(q)$. So, if we repeat the argument, the orbit will leave the layer $I(p)$ again, and it will never enter the layer $I(q)$. And the argument will begin again: we would choose a new $x \notin I(p)$ and we would apply the same argument. □

With this, we conclude the proof of Theorem 4.3. We have already proved that for $C > 0$ our system is cooperative. $\square$

Observation 4.13. *For* $C = 0$ *there is exclusion in our system, i.e., some species will go to extinction.*

Chapter 5

Dynamics

In this chapter we are going to study the dynamics of our discrete model for the hypercycle. As we have already said, the continuous-time model has been widely studied. The dependence of the dynamics on the size of the hypercycle (dimension) has been studied in previous works. It is known that two-member hypercycles ($n = 2$) have a stable node coexistence attractor [7, 25]. Numerical results showed that with three-member hypercycles ($n = 3$) there exists coexistence via strongly damped oscillations. For the case of four-member hypercycles ($n = 4$) there exists coexistence via weakly damped oscillations. Finally, hypercycles with five or more members are governed by attracting, invariant periodic orbits.
In discrete-time model, we are going to show that there exists a fixed point in the interior of the domain. We will show that this point is asymptotically global stable for the case of three members ($n = 3$). Also, we will prove that for the case $C = 0$ and $n = 3$ the system has period six. Finally, we will show and explain our study about the invariant curves of the system for $n = 4$.

5.1 The fixed point

Our model for the discrete hypercycle, equation (3.2), has only one fixed point in the interior of S_n. We want to compute this point. Given that it is a discrete system, one step in time has to coincide with the value of the concentration of the i-th species in the previous step, therefore, this is equivalent to:

$$F_i(x) = x_i \ , \ \forall i \Longleftrightarrow \frac{C + k_i x_{i-1}}{C + \phi(x)} x_i = x_i \ , \ \forall i \Longleftrightarrow \frac{C + k_i x_{i-1}}{C + \phi(x)} = 1 \ , \ \forall i \Longleftrightarrow$$

$$\Longleftrightarrow C + k_i x_{i-1} = C + \phi(x) \ , \ \forall i \Longleftrightarrow k_i x_{i-1} = \phi(x) \ , \ \forall i.$$

So, the fixed point of the system in the interior of S_n, Q, is determined by:

$$k_2 x_1 = k_3 x_2 = \cdots = k_n x_{n-1} = k_1 x_n = \phi(x),$$

or what is the same as:

$$k_i x_{i-1} = \phi(x) = \sum_{i=1}^{n} k_i x_i x_{i-1} \ , \ \forall i.$$

The fixed point also can be explicitly expressed, let us calculate it:

$$x_1 = \frac{k_1}{k_2}x_n, x_2 = \frac{k_1}{k_3}x_n, \dots, x_{n-1} = \frac{k_1}{k_n}x_n, x_n = \frac{k_1}{k_1}x_n.$$

Knowing that the sum of $x_i, \forall i$ must be equal to 1, we have:

$$\sum_{i=1}^{n} x_i = 1 \iff k_1 x_n \left(\frac{1}{k_2} + \frac{1}{k_3} + \cdots + \frac{1}{k_n} + \frac{1}{k_1}\right) = 1 \iff k_1 x_n \left(\sum_{i=1}^{n} \frac{1}{k_i}\right) = 1$$

$$\iff x_n = \frac{1}{k_1 \sum_{i=1}^{n} \frac{1}{k_i}}.$$

So, every component of the fixed point can be expressed as follows:

$$x_i = \frac{1}{k_{i+1} \sum_{i=1}^{n} \frac{1}{k_i}}, \forall i.$$

To calculate the differential matrix of F, i.e., the Jacobian matrix, at the fixed point Q, is a difficult task. To do so, we will proved as follows. First, we compare our map with a simpler one and then we pass to barycentric coordinates. Let us write our system in the form:

$$F_i(x) - x_i = \frac{k_i x_{i-1} - \phi(x)}{C + \phi(x)} x_i.$$

As we have already seen in the previous chapter, the right-hand side of this equation differs on the vector field of the differential equation of the hypercycle, (3.1), only by the factor $(C + \phi(x))^{-1}$. Then, the Jacobian matrices of both systems and their eigenvalues will be proportional differing by the factor $(C + \phi(x))^{-1}$. So, we will work with the differentiable system of equation (3.1).

To calculate the Jacobian matrix we introduce a barycentric change of coordinates in our system:

$$y_i = \frac{k_{i+1} x_i}{\sum_{j=1}^{n} k_{j+1} x_j}. \tag{5.1}$$

This change of coordinates maintains the properties of our system:

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \frac{k_{i+1} x_i}{\sum_{j=1}^{n} k_{j+1} x_j} = \frac{\sum_{i=1}^{n} k_{i+1} x_i}{\sum_{j=1}^{n} k_{j+1} x_j} = 1,$$

$$x_i > 0 \text{ , } k_i > 0 \Rightarrow y_i > 0.$$

This transformation sends S_n into itself, transforming Q into the "center point" $\overline{Q}$ of the simplex S_n:

$$\overline{Q} = \left(\frac{1}{n}, \cdots, \frac{1}{n}\right)$$

Also, it is differentiable in a neighbourhood of S_n and has a differentiable inverse. Let us calculate it:

$$y_i = \frac{k_{i+1} x_i}{\sum_{j=1}^{n} k_{j+1} x_j} \iff x_i = \frac{y_i \sum_{j=1}^{n} k_{j+1} x_j}{k_{i+1}}. \tag{5.2}$$

It has to be fulfilled that $\sum_{i=1}^{n} x_i = 1$. So, we add a summation on the two sides of the second equation. Then, we have:

$$\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} \frac{y_i \sum_{j=1}^{n} k_{j+1} x_j}{k_{i+1}} \iff \sum_{i=1}^{n} \frac{y_i \sum_{j=1}^{n} k_{j+1} x_j}{k_{i+1}} = 1 \iff \sum_{j=1}^{n} k_{j+1} x_j \sum_{i=1}^{n} \frac{y_i}{k_{i+1}} = 1 \iff$$

$$\iff \sum_{j=1}^{n} k_{j+1}x_j = \frac{1}{\sum_{i=1}^{n} \frac{y_i}{k_{i+1}}}.$$

Therefore, replacing this in (5.2), we have the inverse of our change of coordinates:

$$x_i = \frac{y_i M}{k_{i+1}}, \text{ where } M = \Big(\sum_{j=1}^{n} \frac{y_j}{k_{j+1}}\Big)^{-1}.$$

If we differentiate $y_i = \frac{k_{i+1}x_i}{\sum_{j=1}^{n} k_{j+1}x_j} = \frac{k_{i+1}x_i}{M}$, where $M = \sum_{j=1}^{n} k_{j+1}x_j$, we have:

$$\dot{y_i} = \frac{k_{i+1}\dot{x_i}M - k_{i+1}x_i\dot{M}}{M^2} = \frac{k_{i+1}\dot{x_i}M - k_{i+1}x_i\sum_{j=1}^{n} k_{j+1}\dot{x_j}}{M^2}.$$

Now, we replace with $\dot{x_i} = x_i(k_i x_{i-1} - \phi(x))$:

$$\dot{y_i} = \frac{k_{i+1}x_i(k_i x_{i-1} - \phi(x))M - k_{i+1}x_i\sum_{j=1}^{n} k_{j+1}x_j(k_j x_{j-1} - \phi(x))}{M^2}.$$

We also know that $x_i = \frac{y_i M}{k_{i+1}}$ and using this we have: $\phi(x) = \sum_{i=1}^{n} k_i x_i x_{i-1} = M^2 \sum_{i=1}^{n} \frac{y_{i-1}y_i}{k_{i+1}}$.
So:

$$\dot{y_i} = \frac{k_{i+1}\frac{y_i M}{k_{i+1}}(k_i\frac{y_{i-1}M}{k_i} - M^2\sum_{i=1}^{n}\frac{y_{i-1}y_i}{k_{i+1}})M - k_{i+1}\frac{y_i M}{k_{i+1}}\sum_{j=1}^{n} k_{j+1}\frac{y_j M}{k_{j+1}}(k_j\frac{y_{j-1}M}{k_j} - M^2\sum_{i=1}^{n}\frac{y_{i-1}y_i}{k_{i+1}})}{M^2}$$

$$= \frac{y_i M(y_{i-1}M - M^2\sum_{i=1}^{n}\frac{y_{i-1}y_i}{k_{i+1}})M - y_i M\sum_{j=1}^{n} y_j y_{j-1}M^2 + y_i M\sum_{j=1}^{n} y_j M^3\sum_{i=1}^{n}\frac{y_{i-1}y_i}{k_{i+1}}}{M^2}$$

$$= \frac{y_i M^2}{M^2}\Big(y_{i-1}M - M^2\sum_{i=1}^{n}\frac{y_{i-1}y_i}{k_{i+1}} - M\sum_{j=1}^{n} y_j y_{j-1} + M^2\sum_{j=1}^{n} y_j\sum_{i=1}^{n}\frac{y_{i-1}y_i}{k_{i+1}}\Big)$$

$$= y_i M(y_{i-1} - M\sum_{i=1}^{n}\frac{y_{i-1}y_i}{k_{i+1}} - \sum_{j=1}^{n} y_j y_{j-1} + M\sum_{i=1}^{n}\frac{y_{i-1}y_i}{k_{i+1}} = y_i M(y_{i-1} - \sum_{j=1}^{n} y_j y_{j-1}).$$

Therefore, we already have transformed differential equation.

Claim 5.1. *The Jacobian matrix of our system evaluated at the fixed point is a circulant matrix.*

Proof. First, we have to compute the fixed point of our system:

$$\dot{y_i} = 0 \iff y_i M(y_{i-1} - \sum_{j=1}^{n} y_j y_{j-1}) = 0 \iff y_{i-1} - \sum_{j=1}^{n} y_j y_{j-1} = 0.$$

Note that $M > 0$ and $y_i > 0$, $\forall i$.
So, the fixed point is the point such that $y_{i-1} = \sum_{j=1}^{n} y_j y_{j-1}$. We denote the vector field of the system for $\dot{y_i} = Y_i(y_1, ..., y_n)$.
Now, we will compute the components of the Jacobian matrix of our system:

$$\frac{\partial Y_i}{\partial y_j} = \frac{\partial(y_i M)}{\partial y_j}\Big(y_{i-1} - \sum_{j=1}^{n} y_j y_{j-1}\Big) + y_i M\frac{\partial}{\partial y_j}\Big(y_{i-1} - \sum_{j=1}^{n} y_j y_{j-1}\Big).$$

Now, we evaluate it in the fixed point. We know that it fulfills $y_{i-1} = \sum_{j=1}^{n} y_j y_{j-1}$ and it can expressed by $\overline{Q} = \left(\frac{1}{n}, \cdots, \frac{1}{n}\right)$.

$$\left.\frac{\partial Y_i}{\partial y_j}\right|_{\overline{Q}} = y_i M \frac{\partial}{\partial y_j}\left(y_{i-1} - \sum_{j=1}^{n} y_j y_{j-1}\right)\Bigg|_{\overline{Q}} = \frac{1}{n}\frac{1}{\sum_{j=1}^{n}\frac{1}{nk_{j+1}}}\left(\delta_{i-1,j} - \frac{2}{n}\right)$$

$$= \frac{1}{n}\frac{1}{\frac{1}{n}\sum_{j=1}^{n}\frac{1}{k_j}}\left(\delta_{i-1,j} - \frac{2}{n}\right) = \frac{1}{\sum_{j=1}^{n}\frac{1}{k_j}}\left(\delta_{i-1,j} - \frac{2}{n}\right).$$

Where, $\delta_{i-1,j}$ is the Kronecker delta.
Therefore,

$$\left.\frac{\partial Y_i}{\partial y_j}\right|_{\overline{Q}} = \left.\frac{\partial Y_{i+1}}{\partial y_{j+1}}\right|_{\overline{Q}}$$

So, the Jacobian matrix of the system at the fixed point $\overline{Q}$ is a circulant matrix. □

Observation 5.2. *The first row of the Jacobian matrix is given by:*

$$\left(N\left(\frac{-2}{n}\right), \cdots, N\left(\frac{-2}{n}\right), \cdots, N\left(1 - \frac{2}{n}\right)\right),$$

where $N = \sum_{j=1}^{n}\frac{1}{k_j}$.

Proposition 5.3. *The eigenvalues of the Jacobian matrix are:*

$$\lambda_0 = -N \ , \ \lambda_j = N\exp\left(\frac{2\pi i j}{n}\right) \ , \quad j = 1, \cdots, n-1.$$

So, the eigenvalues of our discrete system are:

$$\omega_0 = \frac{C}{C+N} \ , \ \omega_j = 1 + \frac{N}{C+N}\exp\left(\frac{2\pi i j}{n}\right) \ , \quad j = 1, \cdots, n-1.$$

Proof. Using Theorem (2.31), we have our result to compute the values of λ_j , $j = 0, \cdots, n-1$. And using that our discrete equation differs from the differential equation only by the factor $(C + \phi(x))^{-1}$ we have the result for the values of ω_j , $j = 0, \cdots, n-1$. □

Now, we are going to discuss the character of the fixed point in terms of its eigenvalues, when $C > 0$, i.e., when there is cooperation. Observe that $\frac{N}{C+N} \le 1$. We will divide the study in three cases:

1. Case $n = 2$: in this case, we have only two eigenvalues ω_0, ω_1. Let us calculate their norm:

$$|\omega_0| = \left|\frac{C}{C+N}\right| < 1,$$

$$|\omega_1| = |1 + \frac{N}{C+N}\exp(\pi i)| = |1 - \frac{N}{C+N}| < 1.$$

   Then, the fixed point is an attractor.

2. Case $n = 3$: in this case, we have three eigenvalues $\omega_0, \omega_1, \omega_3$. Let us calculate their norm:
$$|\omega_0| = \left|\frac{C}{C+N}\right| < 1,$$
$$|\omega_1|^2 = \left|1 + \frac{N}{C+N}\exp\left(\frac{2\pi i}{3}\right)\right|^2 = \left|1 + \frac{N}{C+N}\left(\cos\left(\frac{2\pi}{3}\right) + i\sin\left(\frac{2\pi}{3}\right)\right)\right|^2$$
$$= \left(1 + \frac{N}{C+N}\cos\left(\frac{2\pi}{3}\right)\right)^2 + \left(\frac{N}{C+N}\right)^2 \sin^2\left(\frac{2\pi}{3}\right) = 1 + \frac{2N}{C+N}\cos\left(\frac{2\pi}{3}\right)$$
$$+\left(\frac{N}{C+N}\right)^2 = 1 - \frac{N}{C+N} + \left(\frac{N}{C+N}\right)^2 < 1 \Longrightarrow |w_1| < 1,$$
$$|\omega_2|^2 = \left|1 + \frac{N}{C+N}\exp\left(\frac{4\pi i}{3}\right)\right|^2 = \left|1 + \frac{N}{C+N}\left(\cos\left(\frac{4\pi}{3}\right) + i\sin\left(\frac{4\pi}{3}\right)\right)\right|^2$$
$$= \left(1 + \frac{N}{C+N}\cos\left(\frac{4\pi}{3}\right)\right)^2 + \left(\frac{N}{C+N}\right)^2 \sin^2\left(\frac{4\pi}{3}\right) = 1 + \frac{2N}{C+N}\cos\left(\frac{4\pi}{3}\right)$$
$$+\left(\frac{N}{C+N}\right)^2 = 1 - \frac{N}{C+N} + \left(\frac{N}{C+N}\right)^2 < 1 \Longrightarrow |w_1| < 1,$$
Then, the fixed point is an attractor.

3. Case $n = 4$:
$$|\omega_1|^2 = \left|1 + \frac{N}{C+N}\exp\left(\frac{2\pi i}{4}\right)\right|^2 = 1 + \left(\frac{N}{C+N}\right)^2 > 1$$
Then, the fixed point is unstable.

4. $n > 4$:
$$|\omega_1| \geq |\mathrm{Re}(\omega_1)| = 1 + \frac{N}{C+N}\mathrm{Re}\left(\exp\left(\frac{2\pi i}{n}\right)\right) > 1$$
Then, the fixed point is unstable.

5.2 Dimension $n = 3$. An asymptotically stable fixed point

For $n = 3$ we have a fixed point $p = (x_1, x_2, x_3)$ in the interior of S_3, determined by:
$$k_1 x_3 = k_2 x_1 = k_3 x_2 = \phi(x) = \sum_{i=1}^{3} k_i x_i x_{i-1}.$$
In the last section we have seen that the fixed point is an attractor, and in this section we want to prove that this point is a global attractor. For this, we are going to use the results that we have developed in section (2.1). More exactly, we are going to use the theorem of global stability.

In this subsection we denote by D the interior of S_3.

Theorem 5.4. *Assume $C > 0$. The function:*

$$\begin{array}{rccc} V: & D & \longrightarrow & \mathbb{R} \\ & x = (x_1, x_2, x_3) & \longmapsto & V(x) = V(x_1, x_2, x_3) \end{array}$$

defined by,

$$V(x) = \left(\sum_{i=1}^{3} \frac{1}{k_{i+1}x_i} \right) \left(\sum_{i=1}^{3} k_{i+1}x_i \right) = \sum_{i,j=1}^{3} \frac{k_{i+1}x_i}{k_{j+1}x_j}.$$

is a global strict Lyapunov function for our model of the hypercycle $F(x)$ for $n = 3$, i.e., $V(x)$ fulfills:

1. $V(p) = 9$,
2. $V(x) > 9$, $\forall x \in D$ *and* $x \neq p$,
3. $V \circ F(x) < V(x)$, $\forall x \in D$ *and* $x \neq p$,

and also, the set $\{x \in S_3 \mid V(x) \leq \alpha\}$ is compact, $\forall \alpha \geq 9$.
Then, the fixed point p is globally asymptotically stable.

Proof. First, we introduce a change of variable: $y_i = k_{i+1}x_i$. The new fixed point is $y = (y_1, y_2, y_3)$ then we have a new expression of $V(x)$ and a new expression of $F(y)$:

$$V(x) = \sum_{i,j=1}^{3} \left(\frac{y_i}{y_j} \right) \text{ and } F_i(y) = k_{i+1}F_i(x) = k_{i+1}x_i \frac{C + k_i x_{i-1}}{C + \phi(x)} = y_i \frac{C + y_{i-1}}{C + \phi(x)}.$$

Now, we know that the fixed point p fulfills $k_i x_{i-1} = k_{i+1}x_i$, $\forall i \in \{1, 2, 3\}$. Therefore $y_1 = y_2 = y_3$. So, we have $V(p) = 9$.
Next, we want to see the second statement of our theorem:

$$V(x) = \sum_{i,j=1}^{3} \left(\frac{y_i}{y_j} \right) = 3 + \sum_{i \neq j} \left(\frac{y_i}{y_j} \right) = 3 + \sum_{i<j} \left(\frac{y_i}{y_j} + \frac{y_j}{y_i} \right).$$

If we prove that, if $y \neq p$ $\sum_{i<j} \left(\frac{y_i}{y_j} + \frac{y_j}{y_i} \right) > 6$, we will have our statement. Therefore, is sufficient to demonstrate the following claim:

Claim 5.5. *If $y_i, y_j > 0$ and $y_i \neq y_j$ then $\frac{y_i}{y_j} + \frac{y_j}{y_i} > 2$, $\forall i, j \in \{1, 2, 3\}$.*

Proof. We will prove this claim by contradiction.
Suppose that $\frac{y_i}{y_j} + \frac{y_j}{y_i} < 2$, $\forall i, j \in \{1, 2, 3\}$.

$$\frac{y_i}{y_j} + \frac{y_j}{y_i} \leq 2 \Longrightarrow \frac{y_i}{y_j} \leq 2 - \frac{y_j}{y_i} \Longrightarrow \frac{y_i}{y_j} \leq \frac{2y_i - y_j}{y_i} \Longrightarrow \frac{y_i^2}{y_j} + y_j \leq 2y_i \Longrightarrow \frac{y_i^2 + y_j^2}{y_j} \leq 2y_i \Longrightarrow$$

$$\Longrightarrow y_i^2 + y_j^2 - 2y_i y_j \leq 0 \Longrightarrow (y_i + y_j)^2 \leq 0.$$

We have a contradiction, so we have the result desired.

□

Now, we want to see the third and last statement of the theorem, i.e., $V(F(y)) < V(y)$, $\forall y \in D \setminus \{p\}$. So, it is sufficient to see that $V(y) - V(F(y)) > 0$, $\forall y \in D \setminus \{p\}$. First, we have:

$$V(F(y)) = \sum_{i,j=1}^{3} \frac{F_i(y)}{F_j(y)} = \sum_{i,j=1}^{3} \frac{y_i \frac{C+y_{i-1}}{C+\phi(x)}}{y_j \frac{C+y_{j-1}}{C+\phi(x)}} = \sum_{i,j=1}^{3} \frac{y_i}{y_j} \frac{C+y_{i-1}}{C+y_{j-1}}.$$

Therefore,

$$V(y) - V(F(y)) = \sum_{i,j=1}^{3} \frac{y_i}{y_j} - \sum_{i,j=1}^{3} \frac{y_i}{y_j} \frac{C+y_{i-1}}{C+y_{j-1}} = \sum_{i,j=1}^{3} \left(\frac{y_i}{y_j} - \frac{y_i}{y_j} \frac{C+y_{i-1}}{C+y_{j-1}} \right)$$

$$= \sum_{i,j=1}^{3} \frac{y_i}{y_j} \left(1 - \frac{C+y_{i-1}}{C+y_{j-1}} \right) = \sum_{i,j=1}^{3} \frac{y_i}{y_j} \left(\frac{y_{j-1} - y_{i-1}}{C+y_{j-1}} \right) = \sum_{i,j=1}^{3} \frac{y_i y_{j-1}}{y_j (C+y_{j-1})}$$

$$- \sum_{i,j=1}^{3} \frac{y_i y_{i-1}}{y_j (C+y_{j-1})} = \left(\sum_{i=1}^{3} y_i \right) \sum_{j=1}^{3} \frac{y_{j-1}}{y_j (C+y_{j-1})} - \left(\sum_{i=1}^{3} y_i y_{i-1} \right) \sum_{j=1}^{3} \frac{1}{y_j (C+y_{j-1})}.$$

Now, we multiply by the common denominator $\prod_{j=1}^{3} y_j (C+y_{j-1})$. Then, our expression is now:

$$\left(\sum_{i=1}^{3} y_i \right) \sum_{j=1}^{3} y_{j-1} y_{j-1} (C+y_{j+1}) y_{j+1} (C+y_j) - \left(\sum_{i=1}^{3} y_i y_{i-1} \right) \sum_{j=1}^{3} y_{j-1} (C+y_{j+1}) y_{j+1} (C+y_j)$$

$$= \left(\sum_{i=1}^{3} y_i \right) \sum_{j=1}^{3} y_{j-1}^2 y_{j+1} (C+y_j)(C+y_{j+1}) - \left(\sum_{i=1}^{3} y_i y_{i-1} \right) \sum_{j=1}^{3} y_j y_{j+1} (C+y_j)(C+y_{j-1}).$$

We introduce the following abbreviations:

$$S = \sum_{i=1}^{3} y_i \ , \ R = \sum_{i=1}^{3} y_i y_{i-1} \ , \ P = \prod_{i=1}^{3} y_i.$$

We develop our last expression because we want to have an expression in function of the parameter C:

$$S \left[\sum_{i=1}^{3} (y_{i-1}^2 y_{i+1} C + y_{i-1}^2 y_{i+1} y_i)(C+y_{i+1}) \right] - R \left[\sum_{i=1}^{3} (y_i y_{i+1} C + y_i^2 y_{i+1})(C+y_{i-1}) \right]$$

$$= S \left[\sum_{i=1}^{3} (y_{i-1}^2 y_{i+1} C^2 + y_{i-1}^2 y_{i+1}^2 C + y_{i-1}^2 y_{i+1} y_i C + y_{i-1}^2 y_{i+1}^2 y_i) \right]$$

$$-R \left[\sum_{i=1}^{3} (y_i y_{i+1} C^2 + y_i y_{i+1} y_{i-1} C + y_i^2 y_{i+1} C + y_i^2 y_{i+1} y_{i-1}) \right]$$

$$= S \left[C^2 \left(\sum_{i=1}^{3} y_{i-1}^2 y_{i+1} \right) + C \left(\sum_{i=1}^{3} (y_{i-1}^2 y_{i+1}^2 + y_{i-1}^2 y_{i+1} y_i) \right) + \sum_{i=1}^{3} y_{i-1}^2 y_{i+1}^2 y_i \right]$$

$$-R \left[C^2 \left(\sum_{i=1}^{3} y_i y_{i+1} \right) + C \left(\sum_{i=1}^{3} (y_i y_{i+1} y_{i-1} + y_i^2 y_{i+1}) \right) + \sum_{i=1}^{3} y_i^2 y_{i+1} y_{i-1} \right].$$

If we order the terms according powers of C, we would obtain the following expression:

$$I_0(y) + I_1(y)C + I_2(y)C^2.$$

First, we have:

$$I_0(y) = S\left[\sum_{i=1}^{3} y_{i-1}^2 y_{i+1}^2 y_i\right] - R\left[\sum_{i=1}^{3} y_i^2 y_{i+1} y_{i-1}\right] = \sum_{i=1}^{3} y_i \sum_{i=1}^{3} y_{i-1}^2 y_{i+1}^2 y_i$$
$$- \sum_{i=1}^{3} y_{i-1} y_{i+1} \sum_{i=1}^{3} y_i^2 y_{i+1} y_{i-1} = 0.$$

Now, we want to see that the other two terms are positive:

$$I_1(y) = S\left[\sum_{i=1}^{3} y_{i-1}^2 y_{i+1}(y_{i+1} + y_i)\right] - R\left[\sum_{i=1}^{3} y_i y_{i+1}(y_{i-1} + y_i)\right] = S\left[\sum_{i=1}^{3} y_{i-1}^2 y_{i+1} y_i + \sum_{i=1}^{3} y_{i-1}^2 y_{i+1}^2\right]$$
$$-R\left[\sum_{i=1}^{3} y_i y_{i+1} y_{i-1} + \sum_{i=1}^{3} y_i^2 y_{i+1}\right] = S(PS + \sum_{i=1}^{3} y_i y_{i+1}^2) - R(3P + \sum_{i=1}^{3} y_i^2 y_{i+1}) = P(S^2 - 3R)$$
$$+S\sum_{i=1}^{3} y_i y_{i+1}^2 - R\sum_{i=1}^{3} y_i^2 y_{i+1} = P\Big(\sum_{i=1}^{3} y_i^2 - R\Big) + S\sum_{i=1}^{3} y_i y_{i+1}^2 - R\sum_{i=1}^{3} y_i^2 y_{i+1}$$
$$= P\Big(\sum_{i=1}^{3} y_i^2 - R\Big) + \sum_{i=1}^{3} y_i^3 y_{i+1}^2 + \sum_{i=1}^{3} y_i^2 y_{i+1}^2 y_{i-1} + \sum_{i=1}^{3} y_i^3 y_{i-1}^2 - \sum_{i=1}^{3} y_i^2 y_{i-1}^3 - \sum_{i=1}^{3} y_i^2 y_{i+1}^2 y_{i-1}$$
$$+\sum_{i=1}^{3} y_i^3 y_{i+1} y_{i-1} = P\Big(\sum_{i=1}^{3} y_i^2 - R\Big) + \sum_{i=1}^{3} y_i^2 y_{i+1}^3 + P\sum_{i=1}^{3} y_i^2 = \sum_{i=1}^{3} y_i^2 y_{i+1}^3 - PR.$$

Doing some calculations, the last expression is equal to:

$$\sum_{i=1}^{3} (y_i - y_{i+1})^2 y_{i-1}^3 y_i.$$

And this expression is positive for all $y \in D \setminus \{p\}$.
Finally, let see if the last term is positive:

$$I_2(y) = S\Big(\sum_{i=1}^{3} y_{i-1}^2 y_{i+1}\Big) - R\Big(\sum_{i=1}^{3} y_i y_{i+1}\Big) = S\Big(\sum_{i=1}^{3} y_i y_{i+1}^2\Big) - R^2$$
$$= \Big(\sum_{i=1}^{3} y_i\Big)\Big(\sum_{i=1}^{3} y_i y_{i+1}^2\Big) - \Big(\sum_{i=1}^{3} y_i y_{i+1}\Big)^2.$$

We want to see if this expression is positive. So we can prove if the following expression is true:

$$\Big(\sum_{i=1}^{3} y_i y_{i+1}\Big)^2 < \Big(\sum_{i=1}^{3} y_i\Big)\Big(\sum_{i=1}^{3} y_i y_{i+1}^2\Big).$$

We choose:

$$z = (\sqrt{y_1}, \sqrt{y_2}, \sqrt{y_3}),$$
$$w = (\sqrt{y_1} y_2, \sqrt{y_2} y_3, \sqrt{y_3} y_1).$$

If we use the Cauchy-Schwarz inequality:

$$\sum_{i=1}^{3} \Big(y_i y_{i+1}\Big)^2 = \|(z \cdot w)\|^2 \leq \|z\|^2 \|w\|^2 = \Big(\sum_{i=1}^{3} y_i\Big)\Big(\sum_{i=1}^{3} y_i y_{i+1}^2\Big).$$

Observation 5.6. *We have already proved that* $I_2(y) \geq 0$ *and we wanted this term to be positive and not equal to* 0. $I_2(y) = 0 \Leftrightarrow y_1 = y_2 = y_3$, *i.e. the fixed point.*

Finally, since $\lim_{x \to \partial S_3} V(x) = \infty$, we clearly have that $\{x \in S_3 \mid V(x) \leq \alpha\}$ is a compact set $\forall \alpha \geq 9$.
We have already proved that our fixed point p is a globally asymptotically stable fixed point. $\square$

Observation 5.7. *When* $C = 0$, F *is defined in* D *and* V *is a first integral because* $V(F(x)) = x$, $\forall x \in D$.

Proposition 5.8. *For* $n = 3$ *and* $C = 0$, *all orbits of the system are periodic of period six.*

Proof. For $C = 0$, the transformed system is:

$$F_i(x) = \frac{k_i x_i x_{i-1}}{\phi(x)},$$

which is not defined on the part of the boundary of S_3 where $\phi(x) = 0$.
If we multiply this equation by k_{i+1} in both sides and using $y_i = x_i k_{i+1}$, we have:

$$k_{i+1} F_i(x) = k_{i+1} \frac{k_i x_i x_{i-1}}{\phi(x)} \Longleftrightarrow F_i(y) = \frac{y_i y_{i-1}}{\phi(y)},$$

where $\phi(y)$ is something depending on y. Now, let us calculate some quotients to prove that the system has period six:

$$\frac{F_1(y)}{F_2(y)} = \frac{y_3}{y_2}, \quad \frac{F_2(y)}{F_3(y))} = \frac{y_1}{y_3}, \quad \frac{F_3(y)}{F_1(y)} = \frac{y_2}{y_1},$$

$$\frac{F_1^2(y)}{F_2^2(y)} = \frac{F_3(y)}{F_2(y)} = \frac{y_3}{y_1}, \quad \frac{F_2^2(y)}{F_3^2(y)} = \frac{F_1(y)}{F_3(y)} = \frac{y_1}{y_2}, \quad \frac{F_3^2(y)}{F_2^2(y)} = \frac{F_2(y)}{F_1(y)} = \frac{y_2}{y_3}.$$

If we calculate until the iterate six, we have:

$$\frac{F_1^6(y)}{F_2^6(y)} = \frac{y_1}{y_2}, \quad \frac{F_2^6(y)}{F_3^6(y)} = \frac{y_2}{y_3}, \quad \frac{F_3^6(y)}{F_2^6(y)} = \frac{y_3}{y_1}.$$

Hence, we have:

$$\frac{F_1^6(y)}{y_1} = \frac{F_2^6(y)}{y_2} = \frac{F_3^6(y)}{y_3} = \alpha \Longleftrightarrow \Big(F_1^6(y), F_2^6(y), F_3^6(y)\Big) = \alpha\Big(y_1, y_2, y_3\Big).$$

Now, the iterates have to be in the domain:

$$1 = \sum_{i=1}^{3} F_i^6(y) = \alpha \sum_{i=1}^{3} y_i \Longrightarrow \alpha = 1.$$

We have proved that all orbits have period six. $\square$

5.3 Invariant curves and study of bifurcations

During the extensive reading of papers about the subject of our project, we have read a lot about the possible existence of invariant curves in our system for values of $n \geq 4$. But, in none of them it is proved this existence, nor analytically neither numerically. So, we have decided to do a numerical study of invariant curves and their stability. We have written the codes in C language (see Appendices for the implementation in C).
First, we are going to describe how we have found invariant curves for our system with values $k_i = 1, \forall i = 1, \ldots, n$, in terms of the parameter $C > 0$.
This is not an easy task, but remember that in section (3.3) we have proved that our system is close to the continuous system for large values of C. Moreover, for large values of C, the points and their images are close. Hence, we have decided to construct a map equivalent to the Poincaré map for continuous systems.
As we have seen in section (5.1), the fixed point, for this values of k_i, is $p = (\frac{1}{n}, \ldots, \frac{1}{n})$. So, we have decided to choose a Poincaré section at the hyperplane $x_1 = \frac{1}{n}$. Once this section is fixed, we choose one point in it, x, keeping in mind that the last component of this point has to be $x_n = 1 - \sum_{i=1}^{n-1} x_i$ and the first component $x_1 = \frac{1}{n}$. Hence, it is only necessary to introduce the remaining $n-2$ components.
Next, this point is iterated with the map of our system until it crosses twice the section. Before the point crosses the section, we keep the last three iterates, and once this point crosses the section, we keep the next three iterates. With these six points, we do an interpolation to find the set of interpolation polynomials. This set has n polynomials: $p_1(t), \ldots, p_n(t)$. The aim of this interpolation is to find the intersection point between the polynomial and the Poincaré section. By definition, this point is the image of a map, that we val g, which is the analogous of a Poincaré map that we also call Poincaré map.
To find $g(x)$, we have the n polynomials and we know that the first polynomial has to take the value $\frac{1}{n}$. With this information we apply Newton's method for one variable to find the solution of $p_1(t) - \frac{1}{n} = 0$. We choose $t = 0$ as an initial condition of the method, because we adjust the values of the table of interpolation to make this possible.
Now, once we have the value t, we substitute this value in the next $n-2$ polynomials and we have the first $n-1$ components of the intersection point. To compute the last component, as before, we have the relation $x_n = 1 - \sum_{i=1}^{n-1} x_i$. By doing in this way, we get our desired point $g(x)$.
Then, to find an invariant curve of the system, we have to find a fixed point of $g(x) = x$. To do so, we use Newton's method for $n-2$ variables. We want to find a solution of the equation $g(x) - x = 0$, where $g(x), x \in \mathbb{R}^{n-2}$. That is only using $n-2$ components, not counting the first and the last ones, because the first component of $g(x) - x$ is automatically 0, since the image point $g(x)$ has the first component equal to $\frac{1}{n}$, since it is in the Poincaré section. The last component is calculated with the property that all of the components have to sum 1.
Therefore, Newton's method returns us the fixed point of the function $g(x)$. And we expect that this point is the intersection between the Poincaré section and an invariant curve.

We have taken the idea of using a polynomial interpolation to get the intersection of the invariant curve with the Poincaré section from Ref. [10].

Now, with the purpose of making easier for the reader the understanding of the code we have implemented, we are going to explain the main functions:

1. **Poincaré map - void ginters()**:
   Let x be a point from the Poincaré section, this function iterates this point until the iterated point cross twice the Poincaré section. It keeps the last three points before the iterate crosses the section and the first three points after the iterate crosses the section and it makes an interpolation (with interpolation function) to obtain an intersection between the polynomials approximation and the section. To find the intersection point, it finds zeros of the first polynomial minus $\frac{1}{n}$, i.e., it solves $P_1(t) - \frac{1}{n} = 0$. We have a function to do this (newton1var).

2. **Newton's method for more than one variable - void newton()**:
   This function calculates the point x such that $g(x) - x = 0$. We also have a function whose point is the solution of $g(x) - x$. We denote this function by called $h(x)$. So, this method calculates the iterate x^{k+1} such that $x^{k+1} - x^k < tol = 10^{-12}$ with a maximum of 15 iterations of the method. In order to calculate the next iterate of a point, we solve the following equation:

$$Dh(x_k)(x^{k+1} - x^k) = -h(p).$$

3. **The model - void fx()**:
   This function calculates the iterate of a point with our model.

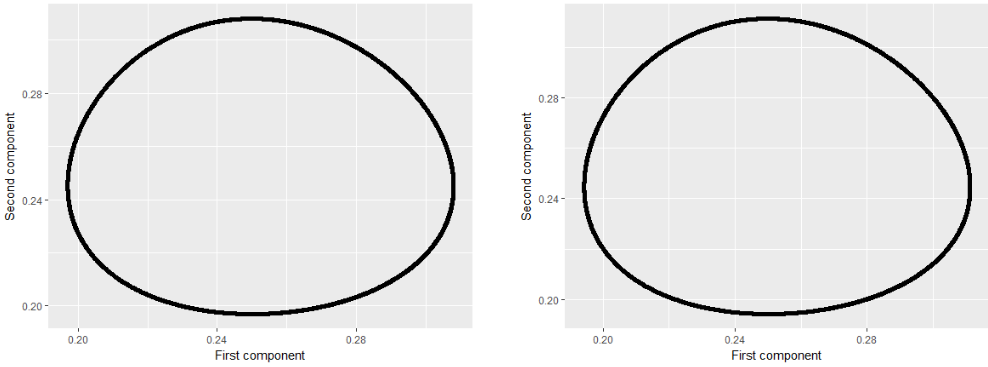
4. **Differential function - void differential()**:
   This function calculates an approximation of the differential $Dh(x)$. It computes the partial derivatives according to the following formula:

$$\frac{\partial h_i}{\partial x_j}(x) \approx \frac{h_i(x + he_j) - h_i(x - he_j)}{2h}, \qquad \text{with } h = 10^{-3}.$$

Now, we are going to comment on the results. We have decided to choose $n = 4$, because is the minimum dimension for which we expect to have invariant curves. As we said before, we take $k_i = 1, \forall i$. Once, for different values of the parameter C, we have the fixed point of the Poincaré map, we iterate this point and we obtain the invariant curve. With the objective of representing this curve, we project the curve on the plane generated by two variables, for example the first and the second x_1, x_2 (see Figures (5.1) and (5.2)).

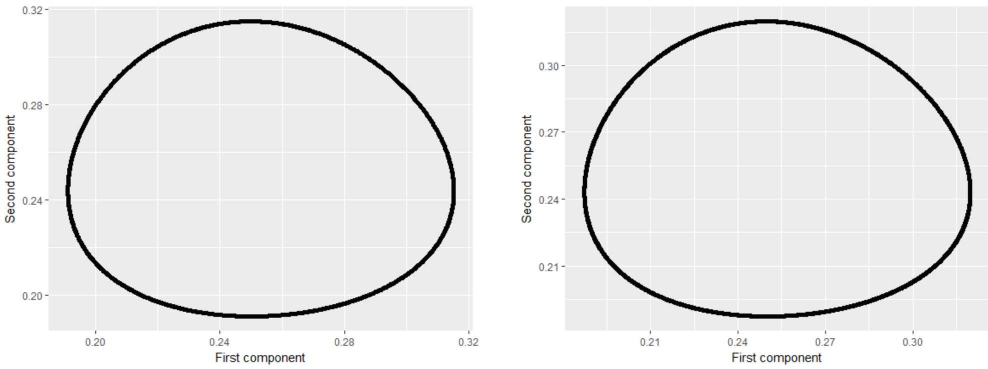
We thought that it could be interesting to study the stability of the invariant curves. To do so, we modified a little bit our program. We made the function newton to calculate the eigenvalues of the differential matrix at the fixed point of the Poincaré map. We have calculated them for values of C from 0.1 to 300 (see Figures (5.3) and (5.4)).

We can observe that the value of the first eigenvalue, λ_1, is always positive for all values of $0.1 \leq C \leq 300$. Also, $\lambda_1 < 1$, but from approximately $C = 100$, the eigenvalue is close to 1. We can also observe that the value of the eigenvalue λ_2 is also always positive



(a) (b)

Figure 5.1: Invariant curve obtained with (a) $C = 50$, initial point: $x = (0.25, 0.15, 0.15, 0.45)$ and the fixed point: $p = (0.25, 0.19694, 0.24527, 0.30779)$, (b) $C = 45$, initial point: $x = (0.25, 0.19694, 0.24527, 0.30779)$ and the fixed point: $p = (0.25, 0.19425, 0.24477, 0.31098)$.



(a) (b)

Figure 5.2: Invariant curve obtained with (a) $C = 40$, initial point: $x = (0.25, 0.19425, 0.24477, 0.31098)$ and the fixed point: $p = (0.25, 0.19110, 0.24414, 0.31476)$, (b) $C = 35$, initial point: $x = (0.25, 0.19110, 0.24414, 0.31476)$ and the fixed point: $p = (0.25, 0.18733, 0.24334, 0.31933)$.

and close to 0 for all values of $0.1 \leq C \leq 300$. Hence, for these values of C, we can expect that the invariant curves are asymptotically stable, using (2.29).

Finally, we want to discuss about an interesting thing. We have read in [15] that when $C \to \infty$ appears a Hopf Bifurcation in our model. This means that for C tending to infinity, the invariant curves collapse into a fixed point. For check this, we do again a little modification in our program and now, we find the fixed point of de Poincaré map for different values of C, but this time, we consider values of C very big and we see that the fixed point of our Poincaré map is very close to the fixed point of our system: $p = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$. See Figure (5.5) to observe the different graphics with the value of C and the value of each component, except the first component that it is always equal to $\frac{1}{4}$.

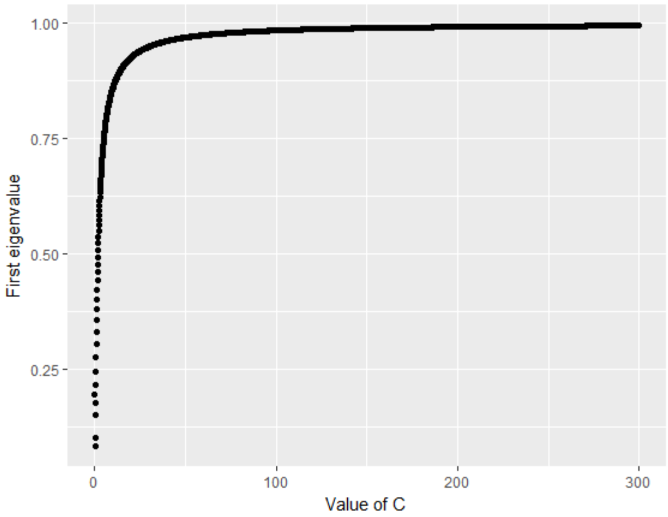


Figure 5.3: The first eigenvalue in terms of the parameter C.

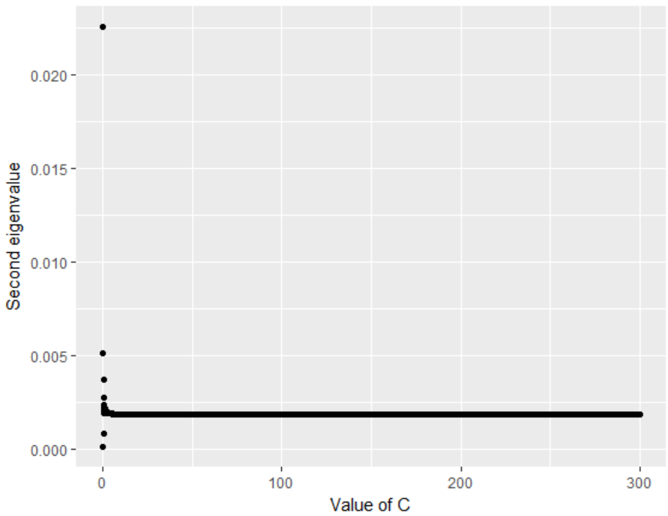
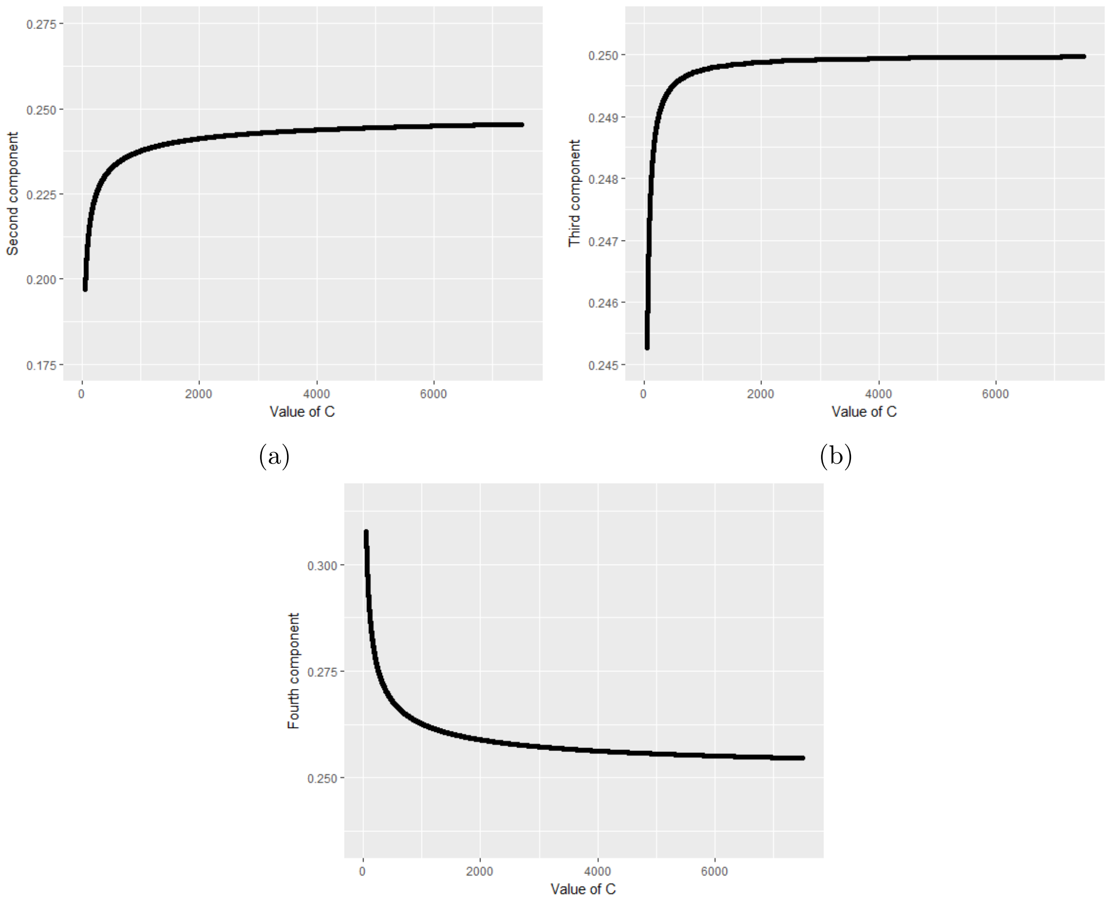


Figure 5.4: The second eigenvalue in terms of the parameter C.

Indeed, we can see in Figure (5.5b) that third component is very close to the value $\frac{1}{4}$, and the other components (see Figure (5.5a) and (5.5c)) are close too, but not so much. That is due to computational problems.

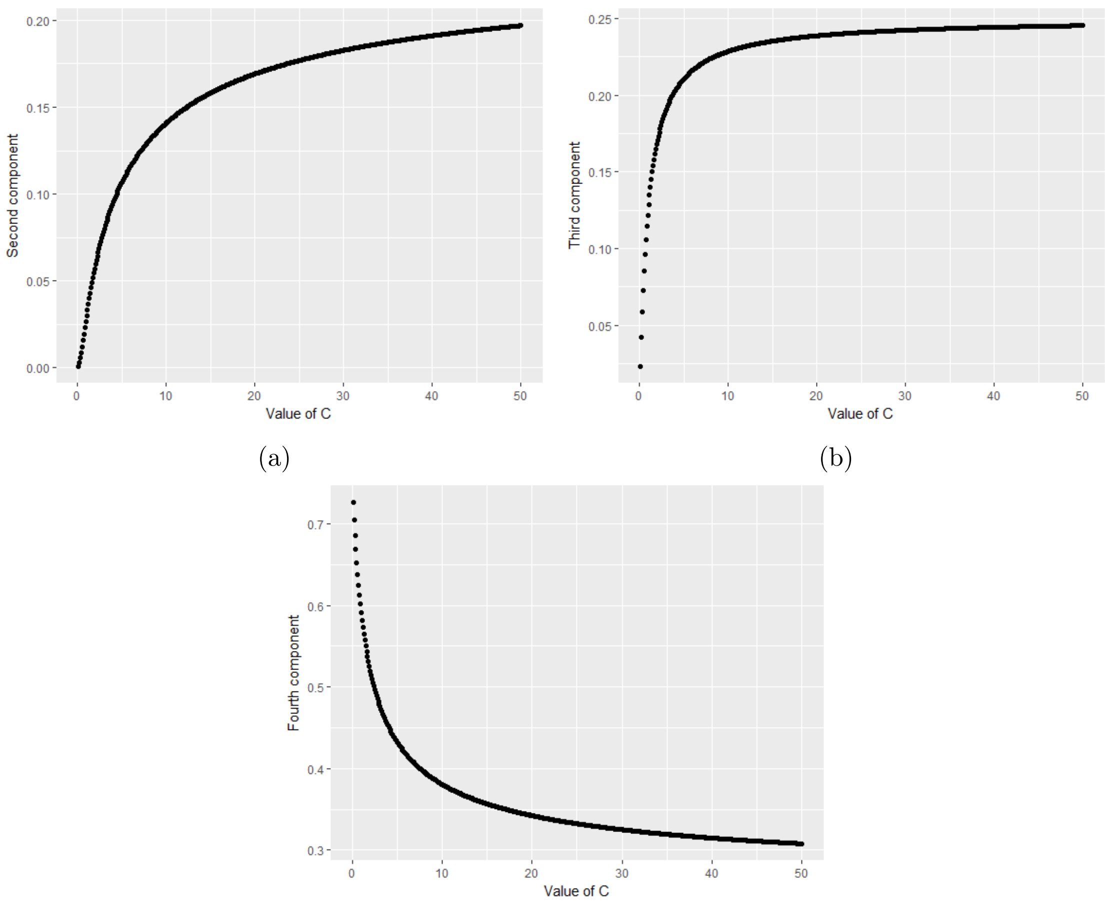
Lastly, we have studied what happens when $C \to 0$, and we have seen that we can continue the invariant curve and there is no bifurcation in this case (see Figure (5.6)).



(a) (b)

(c)

Figure 5.5: Values of each component of fixed point when $C \to \infty$.



(a) (b)

(c)

Figure 5.6: Values of each component of fixed point when $C \to 0$.

Chapter 6

Conclusions

One of the aims of this project was to study the dynamics of a specific discrete-time model for the hypercycle, understanding the biological and mathematical concept. The system that we have studied is the result of a continuous-time system with a discreteness parameter C^{-1}, following the paper of Hofbauer [13].

In the first part of this project, we have introduced different concepts about mathematics, such as Lyapunov stability and Poincaré map, needed to develop our project. We have also introduced the biological interpretation of the hypercycle and we have studied the cooperation between the species in the catalytic architecture.

In the second part, we have studied its dynamics in terms of the dimension of the hypercycle. We have proved that the two-member hypercycle has an asymptotically stable fixed point ensuring the all-species coexistence, and the three-member one has a globally asymptotically stable fixed point. We have also studied, with a numerical implementation, the invariant curves of the four-member hypercycle. This topic had not been studied until now. We have realized that there are asymptotically stable invariant curves for each value of C. Moreover when $C \to \infty$ it seems that the invariant curves tend to collapse into the fixed point.

When we were studying the main paper [13] we have detected some small mistakes that we have been able to correct.

This project has been an opportunity to learn about the extraordinary world of biology and of mathematical modeling for biological processes. The connection between the hypercycle and the origin and evolution of life has surprised me a lot.

Finally, for future projects, it would be interesting to study the invariant curves for larger dimensions of the hypercycle and even to study different discrete-time models with parasites and short circuits.

Bibliography

[1] Amor, D. R., Montañez, R., Duran-Nebreda, S. and Solé, R. V. *Spatial dynamics of synthetic microbial mutualists and their parasites.* PLoS Comput Biol. 2017. vol. 13, num. 8, e15689.

[2] Andino, R. and Domingo, E. *Viral quasispecies.* Elsevier. 2015, vol. 479-480, p. 46-51.

[3] Boerlijst, M. C. and Hogeweg, P. *Spiral wave structure in prebiotic evolution: Hypercycles stable against parasites.* Physica D. 1991, vol. 48, p. 17-28.

[4] Castillo, V., Lázaro, J. T. and Sardanyés, J. *Dynamics and bifurcations in a simple quasispecies model of tumorigenesis.* Comp. Appl. Math. DOI 10.1007/s40314-015-0234-3.

[5] Chicone, C. *Ordinary Differential Equations with Applications.* United States of America: 2006, 1999, Springer Science+Business Media, Inc. eISBN: 0-387-22623-0. p. 82-84, 214-215.

[6] Eigen, M., Biebricher, C. K. and Gebinoga, M. *The Hypercycle. Coupling of RNA and Protein Biosynthesis in the Infection Cycle of an RNA Bacteriophage.* Biochemistry. 1991, vol. 30, num. 46, p. 11005–11018.

[7] Eigen, M. and Schuster, P. *The Hypercycle. A Principle of Natural Self-Organization. Part A.* Naturwissenschaften.1977, vol. 64, num. 11, p. 541–565.

[8] Eigen, M. *Selforganization of Matter and the Evolution of Biological Macromolecules.* Naturwissenschaften.1971, vol. 58, num. 10, p. 465-523.

[9] Farré, G., Sardanyés, J., Guillamon, A. and Fontich, E. *Coexistence stability in a four-member hypercycle with error tail through center manifold analysis.* Nonlinear Dyn. 2017, vol. 90, p. 1873-1883.

[10] Gelfreich, V. and Vieiro, A. *Interpolating vector fields for near identity maps and averaging.* Preprint, arXiv:1711.01983v1.

[11] Gray, R. M. *Toeplitz and circulant matrices: A review.* 2001. Available at: http://ee.stanford.edu/ gray/toeplitz.pdf

[12] Guillamon, A., Fontich, E. and SardanyÃ©s, J. *Bifurcations analysis of oscillating hypercycles.* Journal of Theoretical Biology. 2015, vol. 387, p. 23-30.

[13] Hofbauer, J. *A Difference Equation Model for the Hypercycle.* SIAM Journal on Applied Mathematics. 1984, vol. 44, p. 762-772.

[14] Hofbauer, J. and Sigmund, K. *The theory of evolution and Dynamical Systems.* London Mathematical Society Student Texts. 1995, vol. 7.

[15] Hofbauer, J. and Iooss, G. *A Hopf Bifurcation Theorem for Difference Equations. Approximating a Differential Equation.* Mh. Math. 1984, vol. 98, p. 99-113.

[16] Iggidr, A. and Bensoubaya, M. *New Results on the Stability of Discrete-Time Systems and Applications to Control Problems.* Journal of Mathematical Analysis and Applications. 1998, vol. 219, p. 392-414.

[17] Kauffman, S. A. *The origins of order: self-Organization and selection in Evolution.* Oxford University Press, Inc. 1993.

[18] Lee, D. H., Severin, K., Yokobayashi, Y. and Reza Ghadiri, M. *Emergence of symbiosis in peptide self-replication through a hypercyclic network.* Nature. 1997, vol. 390, p. 591-594.

[19] Miller, S. L. *A production of amino acids under possible primitive earth conditions.* Science. 1953, vol. 117, p. 528-529.

[20] Oparin, A. I. *Life. Its Nature, Origin and Development.* Academic Press Inc., Publishers. 1962.

[21] Oró, J. *Mechanisms of synthesis of adenine from hydrogen cyanide under possible primitive Earth conditions.* Nature. 1961, vol. 190, p. 389-290.

[22] Sardanyés, J. *Dynamics, Evolution and Information in Nonlinear Dynamical Systems of Replicator.* PhD thesis, Universitat Pompeu Fabra, 2009.

[23] Sardanyés, J., and Solé, R. V. *Spatio-temporal dynamics in simple asymmetric hypercycles under weak parasitic coupling.* Physica D. 2007, vol. 231, p. 116-129.

[24] Sardanyés, J. and Solé, R. V. *Bifurcations and phase transitions in spatially extended two-member hypercycles.* Journal of Theoretical Biology. 2006, vol. 243, p. 468-482.

[25] Sardanyés, J. *The Hypercycle: From molecular to ecosystem dynamics.* In: Landscape Ecology Research Trends, chap. 6. Nova Science Publishers, Inc. 2008.

[26] Schuster, P., Sigmund, K. and Wolff, R. *Mass Action Kinetics of Selfreplication in Flow Reactors.* Journal of Mathematical Analysis and Applications. 1980, vol. 78, p. 88-112.

[27] Shou, W., Ram, S. and Vilar, J. M. G. *Synthetic cooperation in engineered yeast populations.* PNAS. 2007, vol. 104, num. 6, p. 1877–1882.

[28] Smith, J. M. and Szathmáry, E. *The major transitions in evolution.* Oxford University Press. 1995.

[29] Szostak, N., Wasik, S. and Blazewicz, J. *Hypercycle*. PLOS Comput Biol. 2016, vol.12, num. 4, p. 1-13.

[30] Teschl, G. *Ordinary Differential Equations and Dynamical Systems*. American Mathematical Society. Graduate Studies in Mathematics. 2012, vol. 140. ISBN: 978-0-8218-8328-0. Parts 2-3.

[31] Vlassov, A. V. *Mini-ribozymes and freezing environment: a new scenario for the early RNA world*. Biogeosciences Discussions. 2005, vol. 2, p. 1719-1737.

[32] Youn Lee, H., Perelson, A. S., Park, S. C. and Leitner, T. *Dynamic Correlation between Intrahost HIV-1 Quasispecies Evolution and Disease Progression*. PLOS Comput Biol. 2008, vol. 4, num. 12, e1000240.

Appendices

A numerical implementation to find invariant curves

First, we have the code of the program that find the fixed point of the Poincaré map.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void fx(double *x0, double *k, double c, int n, double *x);
void ginters(double *x0, double *k, double c, int n, double *g);
void diferential(double *p, double *k, double c, int n, double **diferencial);
void hx(double *p, double *k, double c, int n, double *h);
void gauss(int n, double **A, double *b, double tol);
void newton(double *q0, double *k, double c, int n, double *q);
double polinsubst(double *coef, double *punt, double subst, int n);
void interp(int n, double *x, double *p,double *c);
double newton1var(double x0, double *x, double *coef, int n);
double der(double *coef, double *x, double subst, int n);

int main(void) {

        double *k, c, *condinicial, *puntimatge;
        int n, i;

        printf("Dona'm n\n");
        scanf("%d", &n);

        k = (double*)malloc(n*sizeof(double));
        condinicial = (double*)malloc(n*sizeof(double));
        puntimatge = (double*)malloc(n*sizeof(double));

        if(k == NULL || condinicial == NULL || puntimatge == NULL){
                printf("No hi ha prou memoria \n");
                exit(1);
        }
```

```
        printf("Dona'm els coeficients k_i\n");
        for(i = 0; i < n; i++) {
                scanf("%le", &k[i]);
        }

        printf("Dona'm el valor del coeficient C\n");
        scanf("%le", &c);

        condinicial[0] = 1./n;

        printf("Dona'm les n-2 components del punt inicial\n");
        for(i = 1; i < (n-1); i++) {
                scanf("%le", &condinicial[i]);
        }

        condinicial[n-1] = 1;
        for(i = 0; i < n-1; i++) {
                condinicial[n-1] -= condinicial[i];
        }

        newton(condinicial, k , c, n, puntimatge);

        for(i=0;i<n;i++) printf("%le ",puntimatge[i]);
        printf("\n");

        free(k);
        free(condinicial);
        free(puntimatge);

        return 0;
}
void fx(double *x0, double *k, double c, int n, double *x) {
        double phi = 0;
        int i;

        phi = k[0]*x0[0]*x0[n-1];
        for(i = 1; i < n; i++) {
                phi += k[i]*x0[i]*x0[i-1];
        }
        x[0] = x0[0]*((c + k[0]*x0[n-1])/(c+phi));
        for(i = 1; i < n; i++) {
                x[i] = x0[i]*((c + k[i]*x0[i-1])/(c+phi));
        }
        return;
```

```
}

void ginters(double *x0, double *k, double c, int n, double *g) {
        double recta, *x, *x1, *x2, *p1, *p2, *p3, *p4, *p5, *p6, *aux,
        *aux2, *aux3, *aux4, *aux5, **punts, **matriucoef, *coef, *valors,
        primervalor;
        int i, l, j, cont;

        punts = (double**)malloc(n*sizeof(double*));
        for(i = 0; i < n; i++) {
                punts[i] = (double*)malloc(6*sizeof(double));
        }

        matriucoef = (double**)malloc(n*sizeof(double*));
        for(i = 0; i < n; i++) {
                matriucoef[i] = (double*)malloc(6*sizeof(double));
        }

        x = (double*)malloc(n*sizeof(double));
        x1 = (double*)malloc(n*sizeof(double));
        x2 = (double*)malloc(n*sizeof(double));
        p1 = (double*)malloc(n*sizeof(double));
        p2 = (double*)malloc(n*sizeof(double));
        p3 = (double*)malloc(n*sizeof(double));
        p4 = (double*)malloc(n*sizeof(double));
        p5 = (double*)malloc(n*sizeof(double));
        p6 = (double*)malloc(n*sizeof(double));
        aux = (double*)malloc(n*sizeof(double));
        aux2 = (double*)malloc(6*sizeof(double));
        aux3 = (double*)malloc(6*sizeof(double));
        aux4 = (double*)malloc(6*sizeof(double));
        aux5 = (double*)malloc(6*sizeof(double));
        coef = (double*)malloc(6*sizeof(double));
        valors = (double*)malloc(6*sizeof(double));
        if(x == NULL || x1 == NULL || x2 == NULL || p1 == NULL || p2 == NULL
        || p3 == NULL || p4 == NULL || p5 == NULL || p6 == NULL || aux == NULL
        || aux2 == NULL || aux3 == NULL || aux4 == NULL || aux5 == NULL ||
        coef == NULL || valors == NULL){
                printf("No hi ha prou memoria \n");
                exit(1);
        }

        recta = x0[0];
        fx(x0, k, c, n , x);
        if(x[0] > recta) {
```

```
do {
        fx(x, k, c, n, x1);
        for(i = 0; i < n; i++) x[i]=x1[i];
} while(x1[0] > recta);

for(i = 0; i < n; i++) {
        x2[i] = x[i];
}
cont = 0;
do {
        if(cont == 0) {
                for(i = 0; i < n; i++) p1[i] = x2[i];
        }
        if(cont == 1){
                for(i = 0; i < n; i++) p2[i] = x2[i];
        }
        if(cont == 2) {
                for(i = 0; i < n; i++) p3[i] = x2[i];
        }
        if(cont > 2) {
                for(i = 0; i < n; i++) p1[i] = p2[i];
                for(i = 0; i < n; i++) p2[i] = p3[i];
                for(i = 0; i < n; i++) p3[i] = x2[i];
        }
        fx(x2, k, c, n, aux);
        for(i = 0; i < n; i++) x2[i] = aux[i];
        cont++;
} while(x2[0] < recta);
for(i = 0; i < n; i++) {
        p4[i] = x2[i];
}
fx(x2, k, c, n, p5);
fx(p5, k, c, n, p6);
} else {
do {
        fx(x, k, c, n, x1);
        for(i = 0; i < n; i++) x[i] = x1[i];
} while(x1[0] < recta);
for(i = 0; i < n; i++) {
        x2[i] = x[i];
}
cont = 0;
do {
        if(cont == 0) {
                for(i = 0; i < n; i++) p1[i] = x2[i];
```

```
                }
                if(cont == 1){
                        for( i = 0; i < n; i++) p2[i] = x2[i];
                }
                if(cont == 2) {
                        for(i = 0; i < n; i++) p3[i] = x2[i];
                }
                if(cont > 2) {
                        for(i = 0; i < n; i++) p1[i] = p2[i];
                        for(i = 0; i < n; i++) p2[i] = p3[i];
                        for(i = 0; i < n; i++) p3[i] = x2[i];
                }
                fx(x2, k, c, n, aux);
                for(i = 0; i < n; i++) x2[i] = aux[i];
                cont++;
        } while(x2[0] > recta);
        for(i = 0; i < n; i++) {
                p4[i] = x2[i];
        }
        fx(x2, k, c, n, p5);
        fx(p5, k ,c, n, p6);
}
for(i = 0; i < n; i++) {
        punts[i][0] = p1[i];
        punts[i][1] = p2[i];
        punts[i][2] = p3[i];
        punts[i][3] = p4[i];
        punts[i][4] = p5[i];
        punts[i][5] = p6[i];
}

for(i = 0; i < 6; i++) {
        valors[i] = -2+i;
}

for(i = 0; i < n; i++) {
        for(l = 0; l < 6; l++) {
                aux2[l] = punts[i][l];
        }
        interp(5, valors, aux2, coef);
        for(j = 0; j < 6; j++) {
                matriucoef[i][j] = coef[j];
        }
}
```

```
        primervalor = 0.;
        for(i = 0; i < 6; i++) {
                aux3[i] = matriucoef[0][i];
        }
        aux3[0] -= 1./n;
        for(i = 0; i < n; i++) {
                aux4[i] = punts[i][0];
        }
        primervalor = newton1var(primervalor, valors, aux3, 5);

        for(i = 0; i < n-1; i++) {
                for(j = 0; j < 6; j++) {
                        aux5[j] = matriucoef[i][j];
                }
                g[i] = polinsubst(aux5, valors, primervalor, 5);
        }
        g[n-1] = 1.;
        for(i = 0; i < n-1; i++) g[n-1] -= g[i];
        free(p4);
        free(p5);
        free(p6);
        free(p1);
        free(p2);
        free(p3);
        free(x2);
        free(aux);
        free(aux2);
        free(aux3);
        free(aux4);
        free(aux5);

        return;

}

double newton1var(double x0, double *x, double *coef, int n){

        double tol = 1.e-12, xn, distancia, resultat, resultatder;
        int cont = 0;

        do {
                resultat = polinsubst(coef, x, x0, n);
                resultatder = der(coef, x, x0, n);
                xn = x0 - (resultat/resultatder);
                distancia = fabs(xn - x0);
```

```
                x0 = xn;
                cont++;
        } while(distancia > tol && cont < 15);

        return x0;
}

double der(double *coef, double *x, double subst, int n){

        double h = 1.e-3, resultat, resul1, resul2;
        resul1 = polinsubst(coef, x, subst+h, n);
        resul2 = polinsubst(coef, x, subst-h, n);
        resultat = (resul1 - resul2)/(2*h);

        return resultat;
}

double polinsubst(double *coef, double *punt, double subst, int n) {

        int i;
        double resul;
        resul = coef[n];
        for(i = (n-1); i >= 0; i--) resul = (resul*(subst - punt[i])) + coef[i];

        return resul;
}

void interp(int n, double *x, double *p, double *c) {

        int i, j;

        for(i = 0; i <= n; i++) c[i] = p[i];

        for(i = 0; i < n; i++){
                for(j = n; j > i; j--){
                        c[j] = (double) (c[j]-c[j-1])/(x[j]-x[j-i-1]);
                }
        }

        return;
}

void diferential(double *p, double *k, double c, int n, double **diferencial) {

        double h = 1.e-3,*pa, *ps, *hi1, *hi2;
```

```
        int i, j, l;

        hi1 = (double*)malloc(n*sizeof(double));
        hi2 = (double*)malloc(n*sizeof(double));

        pa = (double*)malloc(n*sizeof(double));
        ps = (double*)malloc(n*sizeof(double));

        if(hi1 == NULL || hi2 == NULL || pa == NULL || ps == NULL){
                printf("No hi ha prou memoria \n");
                exit(1);
        }
        for(i = 1; i < (n-1); i++) {
                for(j = 0; j < n; j++) {
                        pa[j] = p[j];
                        ps[j] = p[j];
                }
                for(j = 1; j < (n-1); j++) {
                        pa[j] -= h;
                        pa[n-1] = 1.;
                        for(l = 0; l < (n-1); l++) pa[n-1] -= pa[l];
                        ps[j] += h;
                        ps[n-1] = 1.;
                        for(l = 0; l < (n-1); l++) ps[n-1] -= ps[l];
                        hx(pa, k, c, n, hi1);
                        hx(ps, k, c, n, hi2);
                        diferencial[i-1][j-1] = (hi2[i] - hi1[i])/(2*h);
                        for(l = 0; l < n; l++) {
                                pa[l] = p[l];
                                ps[l] = p[l];
                        }
                }
        }
        free(ps);
        free(pa);
        free(hi1);
        free(hi2);

        return;

}

void hx(double *p, double *k, double c, int n, double *h) {

        double *g;
```

```
        int i;

        g = (double*)malloc(n*sizeof(double));
        if(g == NULL){
                printf("No hi ha prou memoria \n");
                exit(1);
        }

        ginters(p, k, c, n, g);
        for(i = 0; i < n; i++)  h[i] = g[i] - p[i];

        free(g);

        return;
}

void gauss(int n, double **A, double *b, double tol) {

        int i, j, k;
        double mult;

        for(i = 0; i < n-1; i++) {
                if(fabs(A[i][i]) < tol) {
                        printf("No es pot resoldre el sistema.\n");
                }
                for(j = (i+1); j < n; j++) {
                        mult = A[j][i] / A[i][i];
                        for(k = (i+1); k < n; k++) {
                                A[j][k] = A[j][k] - (mult*A[i][k]);
                        }
                        b[j] = b[j] - (mult*b[i]);
                        A[j][i] = mult;
                }
        }

        if(fabs(A[n-1][n-1]) < tol) {
                printf("No es pot resoldre el sistema.\n");
        }
        for(i = (n-1); i >= 0; i--) {
                for(j = (n-1); j > i; j--) {
                        b[i] -= A[i][j]*b[j];
                }
                b[i] /= A[i][i];
        }
```

```
        return;
}

void newton(double *q0, double *k, double c, int n, double *q) {

        double tol = 1.e-12, *delta, *h, **matrixdif, *h1, *h_mod;
        int i, cont = 0;
        double distance = 10;

        delta = (double*)malloc(n*sizeof(double));
        h = (double*)malloc(n*sizeof(double));
        h1 = (double*)malloc(n*sizeof(double));
        h_mod = (double*)malloc((n-2)*sizeof(double));

        if(delta == NULL || h == NULL || h1 == NULL || h_mod == NULL){
                printf("No hi ha prou memoria \n");
                exit(1);
        }

        matrixdif = (double**)malloc((n-2)*sizeof(double*));
        for(i = 0; i < (n-2); i++){
                matrixdif[i] = (double*)malloc((n-2)*sizeof(double));
        }

        do {
                distance=0;
                hx(q0, k, c, n, h);
                for(i = 0; i < (n-2); i++) h_mod[i]=h[i+1];
                for(i = 0; i < (n-2); i++) {
                        h_mod[i] = -h_mod[i];
                }

                diferential(q0, k, c, n, matrixdif);
                gauss((n-2), matrixdif, h_mod, tol);
                q[0] = q0[0];
                for(i = 0; i < (n-2); i++) {
                        q[i+1] = q0[i+1] + h_mod[i];
                }
                q[n-1] = 1.;
                for(i = 0; i < (n-1); i++) q[n-1] -= q[i];
                for(i = 0; i < n; i++) {
                        q0[i] = q[i];
                }
                for(i = 0; i < (n-2); i++) {
                        distance += h_mod[i]*h_mod[i];
```

```
            }
            distance = sqrt(distance);
            cont++;
    } while(distance > tol && cont < 15);

    free(delta);
    free(h);
    free(h1);
    for (i = 0; i < (n-2); ++i) {
            free(matrixdif[i]);
    }
    free(matrixdif);

    return;
}
```

The following code is to calculate the iterates of a point. We use it to find invariant curves.

```
#include <stdio.h>
#include <stdlib.h>

int main(void){

        int n, i, t;
        double *x0, *x, *k, c, phi;
        FILE *fileWrite;

        fileWrite = fopen("results.dat", "w");
        if(fileWrite == NULL) {
                printf("\nNo es pot obrir el fitxer!\n");
                exit(0);
        }

        printf("Dona'm la dimensio, n:\n");
        scanf("%d", &n);

        x0 = (double*)malloc(n*sizeof(double));
        if(x0 == NULL) {
                printf("\nNo hi ha prou memoria\n");
                exit(1);
        }

        x = (double*)malloc(n*sizeof(double));
        if(x == NULL) {
                printf("\nNo hi ha prou memoria\n");
```

```
		exit(1);
	}

	printf("Dona'm el punt.\n");
	for(i = 0; i < n; i++) scanf("%le", &x0[i]);

	k = (double*)malloc(n*sizeof(double));
	if(k == NULL) {
		printf("No hi ha prou memoria\n");
		exit(1);
	}

	printf("Dona'm els valors de k.\n");
	for(i = 0; i < n; i++) scanf("%le", &k[i]);

	printf("Dona'm el valor de C.\n");

	scanf("%le", &c);

	phi = 0;
	phi = k[0]*x0[0]*x0[n-1];
	for(i = 1; i < n; i++) {
		phi += k[i]*x0[i]*x0[i-1];
	}

	fprintf(fileWrite, "iteration ");
	for(i = 0; i < n; i++) {
		fprintf(fileWrite, "x_%d \t", i+1);
	}
	fprintf(fileWrite, "\n");

	fprintf(fileWrite, "0\t");
	for(i = 0; i < n; i++) {
		fprintf(fileWrite, "%.5lf ", x0[i]);
	}
	fprintf(fileWrite, "\n");

	t=1;
	while(t <= 100000){

		x[0] = x0[0]*((c + k[0]*x0[n-1])/(c+phi));
		for(i = 1; i < n; i++) {
			x[i] = ((c + k[i]*x0[i-1])/(c+phi))*x0[i];
		}
		fprintf(fileWrite, "%d \t", t);
```

```
                for(i = 0; i < n; i++) {
                        fprintf(fileWrite, "%.5lf ", x[i]);
                }
                fprintf(fileWrite, "\n");
                for(i = 0; i < n; i++) x0[i]=x[i];
                phi = 0;
                phi = k[0]*x0[0]*x0[n-1];
                for(i = 1; i < n; i++) {
                        phi += k[i]*x0[i]*x0[i-1];
                }

                t++;
        }

        free(x);
        free(x0);
        fclose(fileWrite);
        free(k);

        return 0;

}
```

Next, we modify the newton function and the main function in the first program to calculate the stability of the invariant curves. Also, we add a function that, using the Newton function, calculates the different fixed points for different values of C ("punt fix").

```
void newton(double *q0, double *k, double c, int n, double *q, double *vaps) {

        double tol = 1.e-12, *delta, *h, **matrixdif, *h1, *h_mod, **a;
        int i, cont = 0;
        double distance = 10;

        delta = (double*)malloc(n*sizeof(double));
        h = (double*)malloc(n*sizeof(double));
        h1 = (double*)malloc(n*sizeof(double));
        h_mod = (double*)malloc((n-2)*sizeof(double));

        if(delta == NULL || h == NULL || h1 == NULL || h_mod == NULL){
                printf("No hi ha prou memoria \n");
                exit(1);
        }
        matrixdif = (double**)malloc((n-2)*sizeof(double*));
        for(i = 0; i < (n-2); i++){
                matrixdif[i] = (double*)malloc((n-2)*sizeof(double));
        }
```

```
a = (double**)malloc((n-2)*sizeof(double*));
for(i = 0; i < (n-2); i++){
        a[i] = (double*)malloc((n-2)*sizeof(double));
}

do{
        distance=0;
        hx(q0, k, c, n, h);
        for(i = 0; i < (n-2); i++) h_mod[i]=h[i+1];
        for(i = 0; i < (n-2); i++) {
                h_mod[i] = -h_mod[i];
        }

        diferential(q0, k, c, n, matrixdif);
        gauss((n-2), matrixdif, h_mod, tol);
        q[0]=q0[0];
        for(i = 0; i < (n-2); i++) {
                q[i+1] = q0[i+1] + h_mod[i];
        }
        q[n-1]=1.;
        for(i=0; i<(n-1); i++) q[n-1] -= q[i];
        for(i = 0; i < n; i++) {
                q0[i] = q[i];
        }
        for(i = 0; i < (n-2); i++) {
                distance+=h_mod[i]*h_mod[i];
        }
        distance=sqrt(distance);
        cont++;
}while(distance > tol && cont < 15);

diferential(q, k, c, n, a);
for(i = 0; i < (n-2); i++) a[i][i] += 1.;

vaps[0] = ((a[0][0]+a[1][1]) + sqrt(((a[0][0]+a[1][1])*(a[0][0]+a[1][1]))
        -4*((a[0][0]*a[1][1])-(a[0][1]*a[1][0]))))/2;
vaps[1] = ((a[0][0]+a[1][1]) - sqrt(((a[0][0]+a[1][1])*(a[0][0]+a[1][1]))
        -4*((a[0][0]*a[1][1])-(a[0][1]*a[1][0]))))/2;

free(delta);
free(h);
free(h1);
for (i = 0; i < (n-2); ++i) {
        free(matrixdif[i]);
```

```
        }
        free(matrixdif);
        for(i = 0; i < (n-2); i++) {
                free(a[i]);
        }
        free(a);

        return;
}

void puntfix(double c, double x1, double x2, int n, double *resul, double *vaps){

        double *k, *condinicial;
        int i;

        k = (double*)malloc(n*sizeof(double));
        condinicial = (double*)malloc(n*sizeof(double));

        for(i = 0; i < n; i++) {
                k[i]=1.;
        }

        condinicial[0] = 1./n;

        condinicial[1] = x1;
        condinicial[2] = x2;

        condinicial[n-1] = 1;
        for(i = 0; i < n-1; i++) {
                condinicial[n-1] -= condinicial[i];
        }

        newton(condinicial, k , c, n, resul, vaps);

        free(k);
        free(condinicial);

        return;
}

int main(void) {

        int n = 4, i;
        double x1, x2, c, *ptfix, *vaps;
        FILE *fileWrite;
```

```
fileWrite = fopen("resultsestabilitat.dat", "w");
if(fileWrite == NULL) {
        printf("\nNo es pot obrir el fitxer!\n");
        exit(0);
}

ptfix = (double*)malloc(n*sizeof(double));
if(ptfix == NULL) {
        printf("\nNo hi ha prou memoria\n");
        exit(1);
}

vaps = (double*)malloc((n-2)*sizeof(double));
if(vaps == NULL) {
        printf("\nNo hi ha prou memoria\n");
        exit(1);
}


fprintf(fileWrite, "C  ");
for(i = 0; i < n; i++) {
        fprintf(fileWrite, "x_%d \t", i+1);
}
fprintf(fileWrite, "lambda_1 \t lambda_2 \t");
fprintf(fileWrite, "\n");

x1 = 0.00280;
x2 = 0.04219;
c=0.1;
while(c < 300) {
        puntfix(c, x1, x2, n, ptfix, vaps);
        fprintf(fileWrite, "%le \t", c);
        for(i = 0; i < n; i++) {
                fprintf(fileWrite, "%.5lf ", ptfix[i]);
        }
        for(i = 0; i < (n-2); i++) {
                fprintf(fileWrite, "%.5lf ", vaps[i]);
        }
        fprintf(fileWrite, "\n");
        c=c+0.1;
        x1 = ptfix[1];
        x2 = ptfix[2];
}
free(ptfix);
```

```
        fclose(fileWrite);
        free(vaps);

        return 0;
}
```

Finally, we modify our main function, with the "puntfix" function, to calculates the fixed point for iterates of C.

```
int main(void) {

        int n = 4, i;
        double x1, x2, c, *ptfix;
        FILE *fileWrite;

        fileWrite = fopen("resultsbif.dat", "w");
        if(fileWrite == NULL) {
                printf("\nNo es pot obrir el fitxer!\n");
                exit(0);
        }

        ptfix = (double*)malloc(n*sizeof(double));
        if(ptfix == NULL) {
                printf("\nNo hi ha prou memoria\n");
                exit(1);
        }

        fprintf(fileWrite, "C  ");
        for(i = 0; i < n; i++) {
                fprintf(fileWrite, "x_%d \t", i+1);
        }
        fprintf(fileWrite, "\n");

        x1 = 0.15;
        x2 = 0.15;
        c=50;
        while(c > 0) {
                puntfix(c, x1, x2, n, ptfix);
                fprintf(fileWrite, "%le \t", c);
                for(i = 0; i < n; i++) {
                        fprintf(fileWrite, "%.5lf ", ptfix[i]);
                }
                fprintf(fileWrite, "\n");
                c=c-0.1;
                x1 = ptfix[1];
                x2 = ptfix[2];
```

```
    }

    free(ptfix);
    fclose(fileWrite);

    return 0;
}
```