

Sistemas complejos: estudio y aplicaciones de modelos logísticos acoplados



Alba Gállego Marín
Trabajo de fin de grado en Matemáticas
Universidad de Zaragoza

Director: Ricardo López Ruiz
6 de julio de 2016

Prólogo

Hay una primera pregunta obligada: ¿qué es un sistema complejo? Un sistema complejo es una agrupación de elementos que interactúan entre ellos con una meta o fin común. Para describirlos hace falta conocer el funcionamiento de las partes interesadas, así como el funcionamiento del sistema una vez relacionadas estas. A raíz de las interacciones entre las partes, aparecen propiedades nuevas que no poseían los elementos aislados. En general, hay una serie de propiedades que llevan a un sistema a desarrollar comportamientos complejos. A grandes rasgos son las siguientes:

1. Los elementos que componen el sistema, sean muchos o pocos, interactúan entre sí de forma no lineal, matemáticamente hablando. Es decir, no hay una relación de proporcionalidad.
2. Las interacciones generan bucles de retroalimentación positiva o negativa.
3. Son sistemas donde constantemente hay intercambio de energía e información entre los elementos que los forman y, en general, con el entorno.

Hay ejemplos de sistemas complejos prácticamente en nuestro día a día: desde colonias de insectos que buscan la supervivencia, hasta las sociedades humanas y su comunicación, pasando por las conexiones del sistema nervioso que nos hacen ser como somos.

En este trabajo, se aplicará el concepto de sistema complejo y lo que él conlleva a poblaciones animales y su evolución. Este tema tiene origen en el siglo XVIII, cuando Thomas Malthus afirmó que la población humana crecía más rápido que los recursos y, así pues, desgraciada e inevitablemente se llegaría a un punto de superpoblación. A partir de este punto comenzaron los intentos de representación y estudio de estos fenómenos poblacionales.

Podemos representar la evolución de una especie aislada mediante un mapa logístico (que es dependiente de un parámetro). Sobre este hecho versará el primer capítulo, donde el objetivo es comprender los conceptos básicos sobre sistemas dinámicos y caos, para poder explicar el comportamiento de una especie aislada en función del parámetro del que depende el mapa logístico.

En el segundo ampliaremos los conocimientos adquiridos sobre sistemas dinámicos y caos al caso bidimensional, y combinaremos dos mapas logísticos para poder así trabajar con dos poblaciones que interactúan entre ellas. Concretamente, se tratarán las tres relaciones básicas entre especies: simbiosis, predador-presa y competición por la supervivencia. Se estudiará el comportamiento de las poblaciones, así como el caos que se presenta en cada uno de los casos.

Finalmente, en el tercer y último capítulo, aparecerán resultados propios. Se dirigirá la atención al caso simbiótico, en el que se introducirá un nuevo parámetro. Se estudiará cómo se modifica el comportamiento de las poblaciones así como el caos que presentan según varía el parámetro introducido.

El trabajo está enfocado de manera práctica, explicado en su mayoría con términos poblacionales, e ilustrado con gráficas de generación propia.

Summary

A complex system is a group of elements that interact among them with a target. In order to describe a complex system, we need to know the behaviour of the parties and the function of the system when the parties are already related. Due to the interactions between the parties, new properties appear.

In this dissertation, animal species will be considered as complex systems. An isolated species can be represented by a logistic map, and that will be the topic of the first chapter. In the second chapter, we will combine two logistic maps for the purpose of studying the basic relationships between species: symbiosis, predator-prey and competition. Finally, in the third and last chapter we will focus attention on the symbiotic case. Introducing a new parameter will give us some new and own results.

Chapter 1. Dynamical systems and one-dimensional logistic maps

The target of this first chapter is to study the behaviour of a logistic map, that represents an isolated species, using the basic concepts of dynamic systems and chaos.

In order to achieve that, we first need to learn the basic concepts about dynamic systems and chaos.

Definition 1. A **dynamical system** is a tuple $(\mathcal{E}, \mathcal{G}, f)$ where \mathcal{E} is the state space, \mathcal{G} is the time space, and $f : \mathcal{G} \times \mathcal{E} \rightarrow \mathcal{E}$ is a map that satisfies:

- f is a continuous map.
- $f(0, x) = x \forall x \in \mathcal{E}$.
- $f(t, f(s, x)) = f(t + s, x) \forall t, s \in \mathcal{G} \text{ y } \forall x \in \mathcal{E}$.

Definition 2. A point x^* is said to be periodic of period n if $f^{(n)}(x^*) = x^*$.

Definition 3. Given an initial condition x_0 , we call **orbit** of x_0 to the sequence $f(x_0), f(f(x_0)), f(f(f(x_0))) \dots$. An orbit is periodic if it is the set of all the iterations over a periodic point.

Theorem 1. *Given a one-dimensional map with f a continuous map, if it has a periodic orbit of period p , it also has periodic orbits of period q , where q greater than p , keeping the following order, called **Sarkovskii's** where each number is less than at its right and each row is less than the following:*

3	5	7	9	11	...	$(2n + 1) \cdot 2^0$...
$3 \cdot 2$	$5 \cdot 2$	$7 \cdot 2$	$9 \cdot 2$	$11 \cdot 2$...	$(2n + 1) \cdot 2^1$...
$3 \cdot 2^2$	$5 \cdot 2^2$	$7 \cdot 2^2$	$9 \cdot 2^2$	$11 \cdot 2^2$...	$(2n + 1) \cdot 2^2$...
$3 \cdot 2^3$	$5 \cdot 2^3$	$7 \cdot 2^3$	$9 \cdot 2^3$	$11 \cdot 2^3$...	$(2n + 1) \cdot 2^3$...
	\vdots					\vdots	
...	2^n	2^{n-1}	...	2^3	2^2	2	1

Based on the previous theorem, a result known as **Li-Yorke's theorem**, says that the existence of a period 3 orbit implies chaos.

In order to detect chaos, we use the **Liapunov exponent**. It measures the separation between next trajectories over time. Chaos is characterized by, at least, one positive Liapunov exponent.

Definition 4. Given the one-dimensional map $x_{n+1} = f(x_n)$, where $x_0 \in [0, 1]$ and its variational equation:

$$y_{n+1} = \frac{df}{dx}(x_n)y_n, \quad (1)$$

Liapunov exponent is defined as:

$$\lambda = \lim_{T \rightarrow \infty} \frac{1}{T} \ln \left| \frac{y_T}{y_0} \right| \quad (2)$$

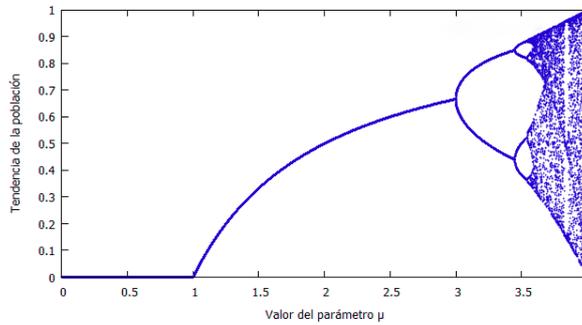
Now it's time to define what a **logistic map** is:

Definition 5. A logistic map is a one-dimensional map in which $f(x_n, \mu) = \mu x_n(1 - x_n)$, giving rise to the following recursive relation:

$$x_{n+1} = \mu x_n(1 - x_n)$$

where $\mu \in (0, 4)$ in order to assure that $x_n \in (0, 1)$.

Considering a logistic map as the representation of an isolated species, in the following image the successive behaviours of the species can be observed. Those are: extinction, equilibrium at a fixed point, period-doubling bifurcation, phase of intermittency, periodic windows and chaos.



Chapter 2. Coupled logistic maps

Now we are going to couple two logistic maps, in order to study the relationship between different species. First, let's see how **Liapunov exponent** is defined in the bidimensional case.

Definition 6. Given two coupled logistic maps, Liapunov exponent is defined as:

$$\sigma := \lim_{T \rightarrow \infty} \frac{1}{T} \ln(|u_T| + |v_T|)$$

where

$$u_{n+1} = \frac{\partial}{\partial x} f_1(x_n, y_n) u_n + \frac{\partial}{\partial y} f_1(x_n, y_n) v_n$$

$$v_{n+1} = \frac{\partial}{\partial x} f_2(x_n, y_n) u_n + \frac{\partial}{\partial y} f_2(x_n, y_n) v_n$$

Symbiosis

In this first case, the species have mutual benefit. The equations are the following:

$$x_{n+1} = \lambda(3y_n + 1)x_n(1 - x_n)$$

$$y_{n+1} = \lambda(3x_n + 1)y_n(1 - y_n)$$

As we did in the one-dimensional case, we are going to enumerate the successive behaviours of the species:

- Extinction of both species.
- Both species reach the equilibrium at a finite population (fixed point), although some initial conditions may promote the extinction.
- Both species oscillate between two fixed points, what means they are in a period 2 orbit. Again, some initial conditions may promote the extinction.
- Synchronization is lost, and both species can survive swinging between infinite states. Periodic windows appear, and as in the last two cases, some initial conditions may promote the extinction.
- The possibility of extinction disappears.
- The system steps in a chaotic regime.

Predator-prey

Now one of the species benefits from preying on the other. The equations are the following:

$$x_{n+1} = \lambda(3y_n + 1)x_n(1 - x_n)$$

$$y_{n+1} = \lambda(-3x_n + 4)y_n(1 - y_n)$$

The successive behaviours of the species are:

- Both species become extinct since the number of preys is not enough.
- Preys can survive but in a number not sufficient to feed predators.
- Predator's strategy becomes efficient, and both species reach the equilibrium at a fixed point.
- Bi-stability: a fixed point coexists with period-doubling bifurcation, and then chaotic bands appear.
- After a Neimark-Hopf bifurcation of the fixed point, the system becomes quasi-periodic.

Competition

In this third case, the species compete for survival. The equations are the following:

$$x_{n+1} = \lambda(-3y_n + 4)x_n(1 - x_n)$$

$$y_{n+1} = \lambda(-3x_n + 4)y_n(1 - y_n)$$

The species present the following behaviours:

- Both species become extinct due to a slight growth.
- The equilibrium is reached at a fixed point, regardless of the initial conditions.
- A period 2 orbit appears, leading then to a period 4 orbit.
- After a Hopf bifurcation the behaviour becomes quasiperiodic.
- The system steps in a chaotic regime.

Chapter 3. Own results: changes and consequences.

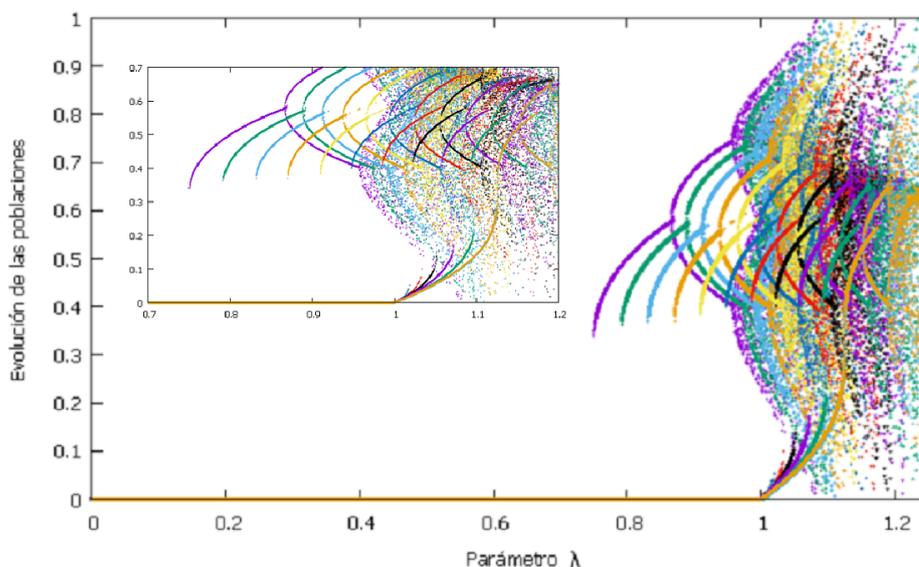
In the last chapter we introduce a new parameter, $\delta \in [0, 1]$, in the symbiotic case. Now the equations will be:

$$\begin{aligned}x_{n+1} &= \lambda(3y_n^{1+\delta} + 1)x_n(1 - x_n) \\ y_{n+1} &= \lambda(3x_n^{1+\delta} + 1)y_n(1 - y_n)\end{aligned}$$

Observe that for the value $\delta = 0$ we have the original symbiotic model. First, we will make a drastic change, introducing $\delta = 1$ in order to observe what differences appear, which are:

- **Bistability:** in the case $\delta = 0$ exists a bistability between life and death, but now it is lost, and in its place a bistability in life appears.
- **Life:** in the original model the first value of life is a finite number, while in this new case life is continuous (it starts from extinction).
- **Chaotic attractor:** when chaos appears, species visit (with different probabilities) successively distinct zones of a chaotic attractor. This chaotic attractor in case $\delta = 0$ presented a bow that now has disappeared.

After studying also the case $\delta = 0.5$, we will be able to conclude that all the models with $\delta \in (0, 1)$ are simply a transition between the cases $\delta = 0$ and $\delta = 1$ with respect to the three properties we have just referred to. We can observe that in the following figure, which shows the changes in the species evolution when we modify the parameter δ of the new symbiotic model. In the figure, each color belongs to a different value of δ .



Índice general

Prólogo	III
Summary	V
1. Introducción: sistemas dinámicos y mapas logísticos unidimensionales	1
1.1. Mapas unidimensionales	1
1.1.1. Sobre órbitas periódicas	2
1.1.2. Sobre órbitas caóticas	3
1.1.3. Rutas hacia el caos	5
1.2. El origen del mapa logístico	6
1.3. Interpretación biológica	7
2. Acoplado de dos mapas logísticos	9
2.1. El acoplado de dos mapas	9
2.2. Simbiosis	9
2.2.1. Puntos fijos, órbitas y curvas invariantes	10
2.2.2. Comportamiento de las poblaciones	13
2.2.3. El caos en mapas bidimensionales	14
2.3. Predador-presa	15
2.3.1. Cálculos esquemáticos y explicación poblacional	15
2.3.2. Evolución al caos	16
2.4. Competición	17
2.4.1. Cálculos esquemáticos y explicación poblacional	17
2.4.2. Evolución al caos	18
3. Resultados propios: alteraciones y consecuencias	19
3.1. Modificación drástica de la tasa de crecimiento	19
3.1.1. Puntos fijos y comportamiento	19
3.1.2. Atractores caóticos	20
3.1.3. Semejanzas y diferencias con el modelo simbiótico original	21
3.2. Modificación gradual de la tasa de crecimiento: un punto intermedio	22
3.2.1. Comportamiento	22
3.2.2. Atractores caóticos	23
3.2.3. Semejanzas y diferencias con los anteriores modelos	24
3.3. Conclusiones	24
Bibliografía	27
Anexos	29

Capítulo 1

Introducción: sistemas dinámicos y mapas logísticos unidimensionales

En este primer capítulo el objetivo es comprender los conceptos básicos sobre sistemas dinámicos, saber cuáles son sus características, y ser capaces con ello de estudiar el comportamiento de un mapa logístico aislado.

1.1. Mapas unidimensionales

Un sistema dinámico es un sistema que evoluciona a lo largo del tiempo, y que viene definido por el espacio de posibles estados que pueden obtenerse y la regla que determina la dinámica del sistema.

Definición 1.1. *Un sistema dinámico es una terna $(\mathcal{E}, \mathcal{G}, f)$ donde \mathcal{E} es el conjunto de posibles estados, \mathcal{G} es el conjunto de tiempos, y $f : \mathcal{G} \times \mathcal{E} \rightarrow \mathcal{E}$ es una aplicación que cumple:*

- *f es una aplicación continua.*
- *$f(0, x) = x \quad \forall x \in \mathcal{E}$.*
- *$f(t, f(s, x)) = f(t+s, x) \quad \forall t, s \in \mathcal{G} \text{ y } \forall x \in \mathcal{E}$.*

Existen sistemas dinámicos continuos y discretos. Si \mathcal{G} es un subconjunto de \mathbb{R} o \mathbb{R}^+ tenemos un sistema continuo, y si es un subconjunto de \mathbb{Z} tenemos un sistema discreto. En nuestro caso, nos interesan los segundos, aquellos en los que el tiempo fluye en pasos (que cuando pasemos a dar una interpretación biológica será el paso del tiempo).

Definición 1.2. *Un sistema dinámico discreto unidimensional, o mapa unidimensional, viene representado por una ecuación en diferencias finitas*

$$x_{n+1} = f(x_n), \quad n = 0, 1, 2, \dots \quad x_0 \in S \quad (1.1)$$

donde S es el **espacio de estados**, que en este caso unidimensional generalmente será considerado $S = [0, 1]$ ó $S = [-1, 1]$, y la función es

$$f : S \rightarrow S, \quad S \subset \mathbb{R}$$

Con este mapa unidimensional, el sistema comienza en el estado x_0 (condición inicial), y según el tiempo evoluciona, iterando la función, aparece la secuencia x_0, x_1, x_2, \dots que también puede verse como $x_0, f(x_0), f(f(x_0)), f(f(f(x_0))), \dots$

Definición 1.3. *Para cualquier condición inicial x_0 , la secuencia x_0, x_1, x_2, \dots se denomina **órbita** del punto x_0 .*

Definición 1.4. Veamos algunas clasificaciones para los puntos del sistema:

- Un punto x^* se dice **fijo** si cumple que $f(x^*) = x^*$. Es decir, los puntos fijos de un sistema son las raíces de la ecuación $f(x) = x$.
- Un punto x^* se dice **periódico** de periodo n si cumple que $f^{(n)}(x^*) = x^*$ donde $f^{(n)}$ denota la n -ésima iteración de f . El conjunto de todas las iteraciones de un punto periódico forman una **órbita periódica**.
- Un punto x^* se dice **eventualmente periódico** de período n si x^* no es periódico pero existe $m > 0$ para el cual

$$f^{(n+i)}(x^*) = f^{(i)}(x^*)$$

para todo $i \geq m$. Es decir, $f^{(i)}$ es periódico para $i \geq m$.

- Sea x^* un punto periódico de periodo n . Dicho punto se dice **hiperbólico** si cumple que

$$|(f^{(n)})'(x^*)| \neq 1$$

Una vez calculados los puntos fijos, nos gustaría poder clasificarlos según su estabilidad. Para ello necesitamos la siguiente caracterización:

Definición 1.5. Dado un mapa f diferenciable definido por $x_{n+1} = f(x_n)$, decimos que un punto fijo x^* es **estable** si cumple que:

$$\left| \frac{df}{dx}(x = x^*) \right| < 1 \tag{1.2}$$

Análogamente si $\left| \frac{df}{dx}(x = x^*) \right| > 1$, el punto se determina **inestable**.

A continuación vamos a ir comentando diferentes aspectos de los mapas unidimensionales que nos serán imprescindibles para comprender el comportamiento del mapa logístico, finalidad de este capítulo.

1.1.1. Sobre órbitas periódicas

Ya se ha visto en la definición 1.4 lo que es una órbita periódica. Veamos ahora el llamado **Teorema de Sarkovskii**, que nos habla sobre la existencia de órbitas periódicas.

Teorema 1.1. Dado un mapa unidimensional con f continua, si este tiene una órbita periódica de periodo p , también tiene órbitas periódicas de periodo q , donde q mayor que p , siguiendo el orden \prec , llamado **orden de Sarkovskii**:

$$3 \prec 5 \prec 7 \prec 9 \prec \dots \prec 2^3 \prec 2^2 \prec 2 \prec 1$$

que aparece más detallado en este esquema siguiente, donde cada número es menor que el situado a su derecha, y cada fila es a su vez menor que la siguiente:

3	5	7	9	11	...	$(2n + 1) \cdot 2^0$...
$3 \cdot 2$	$5 \cdot 2$	$7 \cdot 2$	$9 \cdot 2$	$11 \cdot 2$...	$(2n + 1) \cdot 2^1$...
$3 \cdot 2^2$	$5 \cdot 2^2$	$7 \cdot 2^2$	$9 \cdot 2^2$	$11 \cdot 2^2$...	$(2n + 1) \cdot 2^2$...
$3 \cdot 2^3$	$5 \cdot 2^3$	$7 \cdot 2^3$	$9 \cdot 2^3$	$11 \cdot 2^3$...	$(2n + 1) \cdot 2^3$...
	\vdots					\vdots	
...	2^n	2^{n-1}	...	2^3	2^2	2	1

A partir de este teorema, un resultado conocido como **Teorema de Li y Yorke** nos hace ver que la existencia de una órbita periódica de periodo 3 (análogamente un punto periódico de periodo 3), implica la existencia de órbitas de todos los demás periodos, por lo que se dice, como Li y Yorke citaron en su artículo, que “el periodo 3 implica el caos”.

Definición 1.6. Dado un mapa unidimensional f y una condición inicial x_0 , definimos el conjunto ω -límite de la órbita $\{f^n(x_0)\}$ como:

$$\omega(x_0) = \{x : \forall N, \varepsilon \exists n > N \ni |f^n(x_0) - x| < \varepsilon\} \quad (1.3)$$

¿Cómo podemos interpretar este recién definido conjunto? Si $\omega(x_0)$ es el ω -límite de alguna órbita y x_1 es otra condición inicial, diremos que la órbita $\{f^n(x_1)\}$, o el punto x_1 es **atraído** a $\omega(x_0)$ si $\omega(x_1)$ está contenido en $\omega(x_0)$. Se puede interpretar como el conjunto de todos los puntos a los cuales la órbita vuelve infinitamente. Notar que los puntos fijos son un conjunto ω -límite en sí mismos, ya que $x_0 = f^n(x_0) \Rightarrow \omega(x_0) = x_0$.

1.1.2. Sobre órbitas caóticas

En un mapa unidimensional, debido a la pérdida de linealidad, a veces la solución a grandes tiempos se acaba comportando de manera estocástica. A dichas soluciones, no convergentes a ningún conjunto límite, incluido el infinito, se les denomina caóticas. Es decir, las **órbitas caóticas** son aquellas que se comportan como si estuvieran siempre cerca de un punto inestable, es decir, nunca logran encontrar una cuenca o un ciclo que las atraiga. Poseen la propiedad de la sensibilidad a las condiciones iniciales, que consiste en:

Definición 1.7. Un mapa unidimensional dado por $f : S \rightarrow S$ tiene **sensibilidad a las condiciones iniciales** si $\exists \delta$ tal que $\forall x \in S$ (donde S es el espacio de estados) y entorno E de x , existen $y \in E$, $n \geq 0$ tales que $|f^n(x) - f^n(y)| > \delta$.

Intuitivamente, la sensibilidad a las condiciones iniciales indican que dada una condición inicial x , existen puntos arbitrariamente cercanos a ella tales que tras la n -ésima iteración, la separación es de al menos δ .

Definición 1.8. Sea $\{f^n(x_0)\}$ una órbita caótica. Si x_0 está en $\omega(x_0)$, entonces el conjunto $\omega(x_0)$ es denominado **caótico**.

Un **atractor** es un conjunto ω -límite que atrae a una colección de valores iniciales con medida no nula. Este conjunto de condiciones iniciales es denominado **cuenca de atracción** del atractor. Un **atractor caótico** es un conjunto caótico que a su vez es un atractor.

Ejemplo 1.1. Para ver gráficamente un ejemplo de atractor, vamos a considerar el mapa de Henón. Aunque este tema se centra en los mapas unidimensionales, este ejemplo nos servirá para aclarar el concepto de atractor. El mapa de Henón es uno de los sistemas dinámicos discretos más estudiados que muestra comportamiento caótico. En su versión más clásica, viene dado por las ecuaciones:

$$x_{n+1} = 1 - a \cdot x_n^2 + y_n$$

$$y_{n+1} = b \cdot x_n$$

Con el programa de código (Anexo I) se genera la siguiente gráfica, para los valores $a = 1.4$, $b = 0.3$, donde se observa el atractor del mapa de Henón al que van a parar todas las condiciones iniciales de su cuenca de atracción.

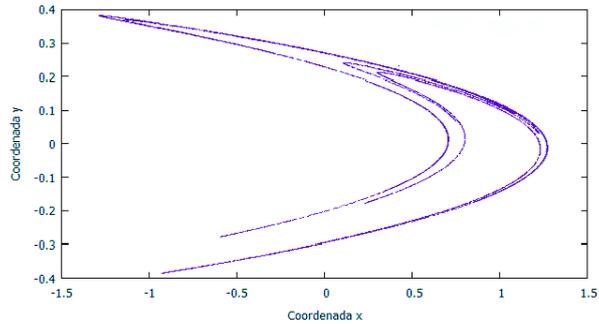


Figura 1.1: Atractor del mapa de Henón.

Si observásemos cómo se va generando la gráfica punto a punto, al principio la gráfica parecería ser una nube caótica con puntos inconexos. Sin embargo, según aumenta el número de iteraciones, la curva se compacta hasta formar el atractor. Una propiedad que lo hace especial es que al acercarnos a cualquier parte de la gráfica, lo que parecían ser líneas individuales, se subdividen en pares de líneas, y así sucesivamente. Es decir, es un atractor con estructura **fractal**. A los atractores con dimensión fractal como este se les denomina **atractores extraños**.

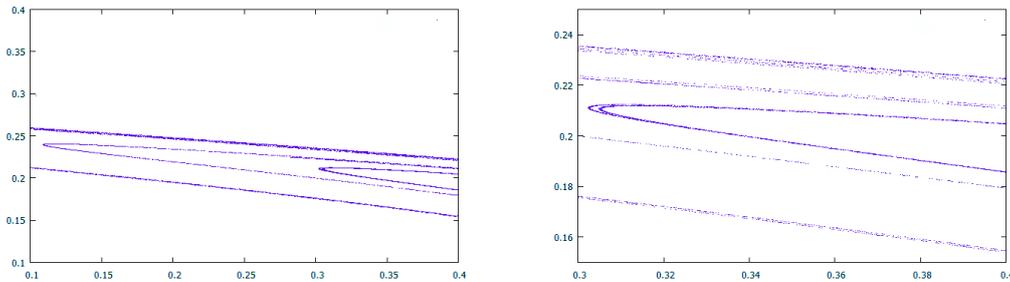


Figura 1.2: Zooms sucesivos del atractor del mapa de Henón.

De acuerdo con el comportamiento de este sistema, una condición inicial del plano será atraída a este atractor extraño, o divergerá al infinito.

Sin embargo, la probabilidad de visita de cada zona del extraño atractor sin embargo no es idéntica. Con los programas de códigos (Anexo II) y (Anexo III) generamos una simulación en el cuadrado $[0, 1] \times [0, 1]$ con un enrejillado de 100×100 puntos asignando a cada celda de la rejilla la probabilidad de ser visitada. El resultado puede visualizarse en la siguiente gráfica:

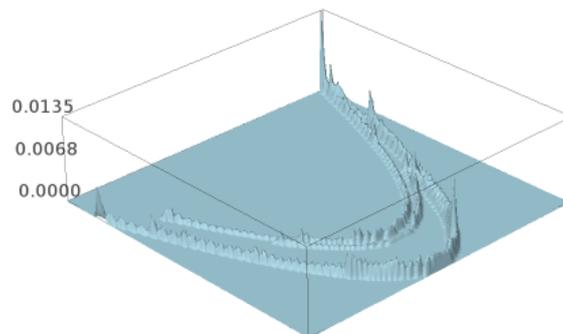


Figura 1.3: Probabilidad de visita de cada una de las zonas del atractor extraño del mapa de Henón.

Ahora que ya conocemos los conceptos de órbita periódica y de atractor, podemos combinarlos para obtener la siguiente definición:

Definición 1.9. En un mapa unidimensional, se denomina **ventana periódica** a aquel intervalo del espacio de parámetros en el que los únicos atractores son órbitas periódicas. Dichas ventanas aparecen normalmente en zonas caóticas.

Para caracterizar cuantitativamente el comportamiento caótico, una de las técnicas usadas es determinar el llamado **exponente de Liapunov**. Grosso modo, el exponente de Liapunov mide la separación con el tiempo entre trayectorias próximas, suponiendo que dicha separación crece exponencialmente. Realmente existe todo un espectro de exponentes de Liapunov, pero lo común es referirse al mayor de ellos, que es el que determina la predictibilidad del sistema. El caos viene caracterizado por un exponente mayor que 0.

Definición 1.10. Dado el mapa unidimensional $x_{n+1} = f(x_n)$, donde $x_0 \in [0, 1]$, y su ecuación variacional:

$$y_{n+1} = \frac{df}{dx}(x_n)y_n, \quad (1.4)$$

el exponente de Liapunov se define como:

$$\lambda = \lim_{T \rightarrow \infty} \frac{1}{T} \ln \left| \frac{y_T}{y_0} \right| \quad (1.5)$$

Ejemplo 1.2. Vamos a considerar el mapa unidimensional dado por $x_{n+1} = 4x_n(1 - x_n)$, donde $f(x) = 4x(1 - x)$, con derivada $\frac{df}{dx} = 4 - 8x$. La ecuación variacional resulta pues:

$$y_{n+1} = (4 - 8x_n)y_n$$

Podemos comprobar que para prácticamente todos los valores iniciales, el exponente de Liapunov para este mapa viene dado por:

$$\lambda = \ln 2$$

1.1.3. Rutas hacia el caos

El caos viene caracterizado por la imposibilidad de predecir el comportamiento futuro con exactitud, aún teniendo las ecuaciones que determinan el sistema dinámico.

Generalmente, el comportamiento de los sistemas se modifica al variar el parámetro (o parámetros) que en la ecuación de este aparece (o aparecen). Con su modificación, el sistema pasa por distintas fases de comportamiento periódico, hasta alcanzar la situación caótica. Dichas fases son las denominadas **rutas al caos**. Van a describirse brevemente a continuación los tres principales procesos que vaticinan que va a llegar el caos:

- **Cascada de duplicaciones de periodo.**

Una cascada aparece como una secuencia infinita de bifurcaciones de duplicación de periodo. Al modificar el parámetro, una órbita periódica estable se desestabiliza duplicando su periodo, y así ocurre sucesivamente.

Para visualizarlo, vamos a considerar por ejemplo el mapa unidimensional dependiente del parámetro a dado por $f_a(x) = a - x^2$, en el que $x_{n+1} = f_a(x_n)$. A continuación pueden apreciarse dos ventanas de duplicación de periodo:

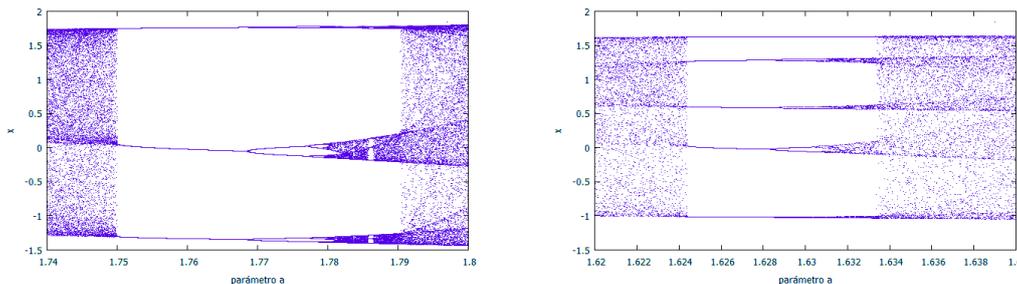


Figura 1.4: Detalle de las duplicaciones de periodo en las ventanas 3 y 5-periódicas.

■ **Intermitencias.**

Este fenómeno sucede cuando al desestabilizarse una órbita periódica, no hay otra órbita estable cercana en el espacio de fases. El sistema entra y sale del estado caótico mediante ciertas ventanas periódicas que aparecen intermitentemente. Esto ocurre con las bifurcaciones saddle-node (punto-silla) en las que un punto de equilibrio estable y uno inestable colisionan, causando la desaparición de ambos y la correspondiente inestabilidad.

Un sencillo ejemplo de una bifurcación saddle-node, ocurre en la ecuación diferencial $\dot{x} = f(x) = a + x^2$. Para deducir los puntos de equilibrio hay que resolver la ecuación $0 = a + x^2$, y para estudiar su estabilidad, evaluar en la derivada $f'(x) = 2x$. Haciendo variar el parámetro los resultados son los siguientes:

- Para $a < 0$ hay dos puntos de equilibrio, que son $-\sqrt{-a}$ y $\sqrt{-a}$, estable ($f'(-\sqrt{-a}) = -2\sqrt{-a} < 0$) e inestable ($f'(\sqrt{-a}) = 2\sqrt{-a} > 0$).
- Para $a = 0$ ambos puntos de equilibrio han colisionado y se ha formado un punto silla.
- Para $a > 0$, tras el fenómeno saddle-node, ya no hay ningún punto de equilibrio.

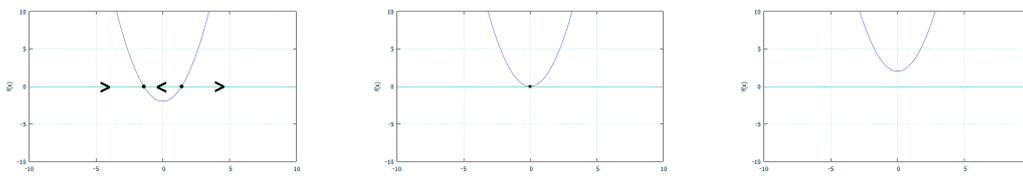


Figura 1.5: Gráficas para $a < 0$, $a = 0$ y $a > 0$.

■ **Cuasiperiodicidad.**

En este caso, se parte de dos frecuencias inconmensurables, y se alcanza el caos al añadir una tercera, como introdujeron Ruelle, Takens y Newhouse en 1978 en su "Teoría de la cuasiperiodicidad a tres frecuencias", que supuso un gran avance en el entendimiento de las turbulencias.

Este fenómeno suele dar lugar a atractores extraños como el que se apreciaba en el ejemplo 1.1.

1.2. El origen del mapa logístico

Como ya se comentó en el prólogo, fue Thomas Malthus (1766-1834) el primero en manejar los conceptos sobre evolución de poblaciones. Sigamos sus pasos.

Supongamos que p es la población actual. El economista y sacerdote Malthus propuso que el crecimiento venía dado por la siguiente ecuación diferencial:

$$\frac{dp}{dt} = k \cdot p \quad (1.6)$$

donde k es la denominada *tasa de crecimiento*. Resolviendo fácilmente esta ecuación diferencial, obtenemos que la solución es:

$$p(t) = e^{k \cdot t}$$

por lo que este primer modelo de Malthus proponía un crecimiento de la población meramente exponencial frente al crecimiento lineal de los recursos.

Fue el matemático belga Pierre-François Verhulst (1804-1849) quien argumentó que dicho modelo era erróneo, pues no se habían tenido en cuenta de forma adecuada los recursos disponibles, que por supuesto limitan el crecimiento. Este propuso que la tasa de crecimiento es proporcional al tamaño de la población, y que además hay un término inhibitor debido a los limitados recursos, que entra en la ecuación siendo proporcional al cuadrado del tamaño de la población:

$$\frac{dp}{dt} = k \cdot p - n \cdot p^2 \quad (1.7)$$

La solución general a esta ecuación diferencial es la siguiente:

$$p(t) = \frac{k}{n} \cdot \frac{e^{ck+kt}}{e^{ck+kt} + 1}$$

que tiende asintóticamente a la constante $\frac{k}{n}$, donde c es una constante de integración. Verhulst llamó a esta propuesta *función logística*, se cree que porque en ese tiempo Verhulst era profesor de la *Royal Military Academy* de Bruselas, y por aquel entonces la palabra logística se asociaba al mundo militar. Hay que esperar hasta finales del siglo XX para que de la mano del físico Robert May se descubra la importancia del trabajo de Verhulst, ya que este modelo logístico ha resultado ser la base de la actual teoría del caos. El siguiente paso es pues dar un salto hasta la versión discreta del modelo que Verhulst propuso.

Definición 1.11. *Un mapa logístico es un mapa unidimensional en el que $f(x_n, \mu) = \mu x_n(1 - x_n)$, dando lugar a la siguiente relación recursiva:*

$$x_{n+1} = \mu x_n(1 - x_n) \quad (1.8)$$

donde el parámetro $\mu \in (0, 4)$ para asegurar que $x_n \in (0, 1)$ en todo momento.

1.3. Interpretación biológica

Los elementos de la ecuación que define a un mapa logístico tienen distintos significados, y combinados nos determinan la dinámica de una población, teniendo en cuenta que hay recursos limitados:

- $x_n \in (0, 1)$ representa la población de una especie aislada tras n generaciones.
- $\mu \in (0, 4)$ es la llamada *tasa de crecimiento*.
- El término μx_n representa la fase de expansión (crecimiento de la población).
- El término $1 - x_n$ representa cómo de alejado está el sistema de la superpoblación.

Mediante simulación podemos estudiar la dinámica del mapa logístico según los distintos valores de μ , siempre en el rango válido:

- Para $0 < \mu < 1$, la tasa de crecimiento no es suficiente para estabilizar a la población, $\lim_{n \rightarrow \infty} x_n = 0$ y con ello la población no puede evitar la **extinción**.

- Para $1 < \mu < 3$, es posible alcanzar un equilibrio entre el crecimiento y la superpoblación, estabilizándose así la población. Como sabemos que los puntos de equilibrio se encuentran entre los puntos fijos, los calculamos. Sabiendo que hay que resolver una ecuación de segundo grado y que $x = 0$ es un punto fijo, el otro que hallemos será el punto de equilibrio que andamos buscando:

$$x = \mu x(1 - x) \Rightarrow \mu(1 - x) = 1$$

con lo que el punto de equilibrio que buscamos, aquel en el que se estabiliza la población independientemente de las condiciones iniciales es:

$$x = \frac{\mu - 1}{\mu} \quad (1.9)$$

- Para $3 < \mu < 3.57$, la población oscila en órbitas de periodo 2^n , donde n varía entre 1 (cuando μ es cercano a 3), e ∞ (cuando μ se aproxima al valor límite 3.57). Se observa pues una **cascada de duplicación de periodos**, una ruta al caos como se comentó en la página 6.
- Para $3.57 < \mu < 3.82$, el sistema va alternando un comportamiento periódico y un régimen caótico, en una fase de **intermitencia**. La población deja de ser predecible.
- Para $3.82 < \mu < 3.85$, aparece una órbita de período 3 para el valor $\mu = 3.82$ después de las impredecibles “explosiones” de la fase de intermitencias. La existencia de una órbita de periodo 3 significa, tal y como vimos en el teorema 1.1, que todos los periodos son posibles para el sistema dinámico.
- Para $3.82 < \mu < 4$, hay un comportamiento caótico con presencia de **ventanas periódicas**.
- Para $\mu = 4$, la dinámica de la población es caótica y nada predecible.

Con el programa de código (Anexo IV) se obtiene la gráfica 1.6.a, donde aparece la evolución de una población según el mapa logístico.

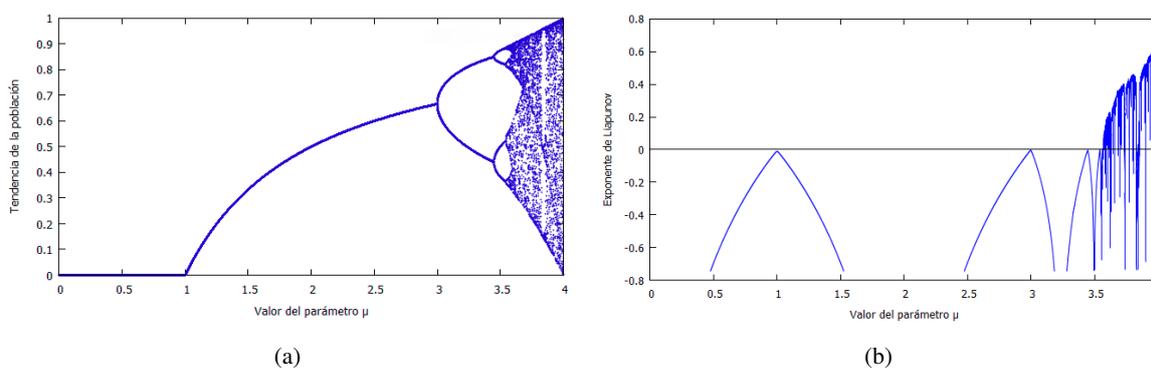


Figura 1.6: (a) Dinámica de la población según el mapa logístico y (b) Exponentes de Liapunov para el mapa logístico.

Para poder visualizar las apariciones caóticas a las que se ha hecho referencia, y compararlas con la gráfica evolutiva, utilizando el programa de código (Anexo V), se obtiene la gráfica 1.6.b, que contiene los diferentes valores del exponente de Liapunov según la evolución del parámetro μ . Recordar que el caos aparece cuando el exponente es positivo.

Capítulo 2

Acoplado de dos mapas logísticos

Tras el estudio de un mapa logístico aislado, pasamos a dar un paso más: acoplar dos. Aunque podría darse a esto un enfoque económico o neuronal, siguiendo con nuestra interpretación biológica, esto significará la convivencia de dos especies, con diferentes intereses según sea el mapa logístico que las representa.

2.1. El acoplado de dos mapas

Para acoplar dos mapas logísticos, necesitamos introducir en cada uno de ellos una tasa de crecimiento que será variable con el tiempo debido a la interacción entre las dos especies, $\mu(z)$, donde $z = x_n$ ó $z = y_n$. Así, este nuevo parámetro depende de la población de la otra especie mediante una constante positiva λ , que mide la fuerza de interacción mutua. Para que la apariencia de la tasa de crecimiento no se complique demasiado, puede tomarse como un crecimiento (1) o como un decrecimiento (2) lineal. Obviamente la población z debe seguir estando en $(0, 1)$, por lo que la tasa debe a su vez seguir estando en $(0, 4)$. Para asegurarlo:

1. $\mu_1(z) = \lambda(3z + 1)$

2. $\mu_2(z) = \lambda(-3z + 4)$

El primer caso puede interpretarse como una población que beneficia a la otra con su existencia, y el segundo, como una interacción competitiva. Según cómo se combinen los parámetros, hay tres situaciones posibles:

- **Simbiosis**, cuando las dos poblaciones presentan una tasa μ_1 . Hay un beneficio mutuo.
- **Predador-presa**, cuando cada una de las poblaciones presenta una de las posibles tasas de crecimiento. Una población es beneficiosa y la otra competitiva.
- **Competición**, cuando ambas poblaciones presentan una tasa μ_2 . Se compete con la otra especie con la finalidad de sobrevivir.

2.2. Simbiosis

Vamos primeramente a estudiar con detalle el caso simbiótico, indagando en los aspectos más importantes, y definiendo conceptos análogos a los del primer capítulo para el caso bidimensional.

Como bien se ha comentado, en este caso la tasa de crecimiento λ es en ambos casos del tipo μ_1 , y es denominada **beneficio mutuo**. Así las dos ecuaciones logísticas que vamos a combinar vienen dadas por las siguientes expresiones:

$$x_{n+1} = \lambda(3y_n + 1)x_n(1 - x_n)$$

$$y_{n+1} = \lambda(3x_n + 1)y_n(1 - y_n)$$

Las simulaciones numéricas muestran que el parámetro λ en este caso solo tiene sentido en el intervalo $(0, 1.0834)$. Téngase en cuenta que si $x_n = 0$, ó $y_n = 0$, se recupera la ecuación de una especie aislada, con $\mu = \lambda$. Los efectos se producen por el acoplamiento de las especies, y no puede deducirse de las propiedades individuales de los mapas logísticos de cada una de ellas. Vamos primero a estudiar el sistema, para poder después explicar qué ocurre según los diferentes valores del parámetro.

2.2.1. Puntos fijos, órbitas y curvas invariantes

Primero nos centramos en la búsqueda de puntos fijos, ya que la existencia de puntos fijos no triviales (distintos de $(0, 0)$) nos asegura una evolución no trivial del sistema. La restricción de los mapas logísticos acoplados a la diagonal $x_n = y_n$, nos reduce el sistema a un mapa unidimensional cúbico, dado por la ecuación:

$$x_{n+1} = \lambda(3x_n + 1)x_n(1 - x_n) \quad (2.1)$$

Por otro lado, la restricción a los ejes nos lleva al mapa logístico unidimensional:

$$x_{n+1} = \lambda x_n(1 - x_n) \quad (2.2)$$

Así obtenemos de momento 5 puntos fijos: p_0 , p_3 , y p_4 en la diagonal, y p_1 , y p_2 en los ejes. Sus valores, igualando a x_n y resolviendo las ecuaciones (2.1) y (2.2) son:

$$p_0 = (0, 0), p_3 = \left(\frac{1 - \sqrt{4 - \frac{3}{\lambda}}}{3}, \frac{1 - \sqrt{4 - \frac{3}{\lambda}}}{3} \right), p_4 = \left(\frac{1 + \sqrt{4 - \frac{3}{\lambda}}}{3}, \frac{1 + \sqrt{4 - \frac{3}{\lambda}}}{3} \right)$$

$$p_1 = \left(\frac{\lambda - 1}{\lambda}, 0 \right), p_2 = \left(0, \frac{\lambda - 1}{\lambda} \right)$$

De hecho, puede comprobarse con diversas herramientas como *Sage*, que no existen más puntos fijos, ya que todo punto fijo deberá satisfacer el sistema de ecuaciones (2.3), y este solo cuenta con las 5 soluciones p_0, \dots, p_5 que ya hemos hallado con métodos más sencillos.

$$\begin{cases} x = \lambda(3y + 1)x(1 - x) \\ y = \lambda(3x + 1)y(1 - y) \end{cases} \quad (2.3)$$

Si queremos analizar la estabilidad de puntos fijos en espacios multidimensionales como este, necesitamos nuevos conceptos.

Definición 2.1. Dado un mapa multidimensional $f = (f_1, \dots, f_n)$ en \mathbb{R}^n y un punto $p \in \mathbb{R}^n$, la **matriz jacobiana** de f en p se define como:

$$J(p) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

En el caso particular de tener un mapa bidimensional $f = (x_{n+1}, y_{n+1})$ la matriz jacobiana queda:

$$J(p) = \begin{pmatrix} \frac{\partial x_{n+1}}{\partial x_n} & \frac{\partial x_{n+1}}{\partial y_n} \\ \frac{\partial y_{n+1}}{\partial x_n} & \frac{\partial y_{n+1}}{\partial y_n} \end{pmatrix}$$

Teorema 2.1. Sea f un mapa en \mathbb{R}^n y p un punto fijo de este.

1. Si el módulo de todos los valores propios de $J(p)$ es menor que 1, entonces p es estable y un **atractor** para el sistema.
2. Si el módulo de todos los valores propios de $J(p)$ es mayor que 1, entonces p es inestable y un **repulsor** para el sistema.

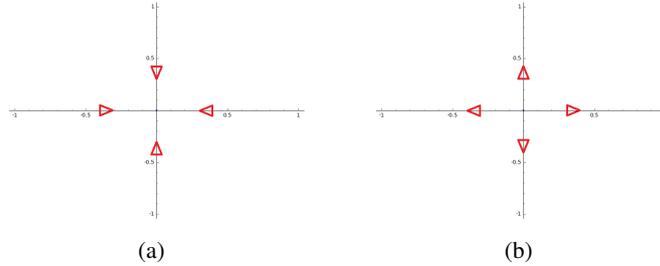


Figura 2.1: (a) Atractor y (b) repulsor en el caso de valores propios reales.

Definición 2.2. Sea f un mapa en \mathbb{R}^n y p un punto fijo de este. El punto p se dice **hiperbólico** si ninguno de los valores propios de $J(p)$ tiene módulo 1. Además, si p es hiperbólico y al menos uno de los valores propios de $J(p)$ tiene valor absoluto mayor que 1, p es un **punto silla**, que siempre es inestable.

Calculamos pues para empezar con el estudio la matriz jacobiana de nuestro sistema:

$$J(x_n, y_n) = \begin{pmatrix} \lambda(3y_n + 1)(1 - 2x_n) & 3\lambda x_n(1 - x_n) \\ 3\lambda y_n(1 - y_n) & \lambda(3x_n + 1)(1 - 2y_n) \end{pmatrix} \quad (2.4)$$

A continuación se evaluarán los puntos fijos que queremos clasificar:

$$J(p_0) = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \quad (2.5)$$

El único valor propio de la matriz (2.5) es el que aparece repetido en la diagonal, λ , así que el punto fijo $(0,0)$ es estable cuando $\lambda < 1$, por lo que en el rango del parámetros $0 < \lambda < 1$ actúa como un atractor, y fuera de él como un repulsor.

$$J(p_1) = \begin{pmatrix} 2 - \lambda & 3\frac{\lambda - 1}{\lambda} \\ 0 & 4\lambda - 3 \end{pmatrix} \quad (2.6)$$

Para la matriz (2.6), los valores propios son $2 - \lambda$ y $4\lambda - 3$ que son mayores que 1 para $\lambda < 1$, así que p_1 es inestable en dicho rango.

$$J(p_2) = \begin{pmatrix} 4\lambda - 3 & 0 \\ 3\frac{\lambda - 1}{\lambda} & 2 - \lambda \end{pmatrix} \quad (2.7)$$

Los valores propios de la matriz (2.7) son $4\lambda - 3$ y $2 - \lambda$, los mismos del caso anterior, así que también p_2 es inestable.

$$J(p_3) = \begin{pmatrix} 2 + \lambda(-2 + \sqrt{4 - \frac{3}{\lambda}}) & \frac{1}{3}(3 - \lambda(2 + \sqrt{4 - \frac{3}{\lambda}})) \\ \frac{1}{3}(3 - \lambda(2 + \sqrt{4 - \frac{3}{\lambda}})) & 2 + \lambda(-2 + \sqrt{4 - \frac{3}{\lambda}}) \end{pmatrix} \quad (2.8)$$

Los valores propios de (2.8) son $\lambda \sqrt{\frac{4\lambda-3}{\lambda}} - 2\lambda + 2 \pm \frac{1}{3} \sqrt{8\lambda^2 + 2(2\lambda^2 - 3\lambda) \sqrt{\frac{4\lambda-3}{\lambda}} - 15\lambda + 9}$. Vamos con el último punto fijo:

$$J(p_4) = \begin{pmatrix} 2 - \lambda(2 + \sqrt{4 - \frac{3}{\lambda}}) & \frac{1}{3}(3 - \lambda(2 - \sqrt{4 - \frac{3}{\lambda}})) \\ \frac{1}{3}(3 - \lambda(2 - \sqrt{4 - \frac{3}{\lambda}})) & 2 - \lambda(2 + \sqrt{4 - \frac{3}{\lambda}}) \end{pmatrix} \quad (2.9)$$

Los valores propios de (2.9) son $-\lambda \sqrt{\frac{4\lambda-3}{\lambda}} - 2\lambda + 2 \pm \frac{1}{3} \sqrt{8\lambda^2 - 2(2\lambda^2 - 3\lambda) \sqrt{\frac{4\lambda-3}{\lambda}} - 15\lambda + 9}$.

Usando los cálculos y conclusiones que acabamos de ver, podemos explicar más detalladamente la evolución dinámica del sistema.

Cuando $0 < \lambda < 1$, ya hemos deducido arriba que p_0 es un nodo atractivo, por lo que efectivamente la extinción es un estado posible. Para el resto de valores del parámetro, es repulsivo. Como también habíamos concluido ya, por ser los valores propios asociados a p_1 y p_2 siempre alguno con módulo mayor que 1, dichos puntos son inestables. Los puntos p_3 y p_4 no existen para $0 < \lambda < 0.75$, así que no intervienen hasta pasado dicho rango de valores. Cuando $\lambda = 0.75$, se produce una bifurcación saddle-node en la que se generan p_3 y p_4 (es decir, pasan a ser reales). A partir de este momento, p_3 es un punto silla y p_4 un nodo atractivo, hasta que en $\lambda = \frac{\sqrt{3}}{2} \cong 0.866$, el nodo p_4 sufre una bifurcación de doblamiento de período, generando una órbita de período 2 que alterna entre los puntos p_5 y p_6 ya fuera de la diagonal. Las coordenadas de dichos puntos son combinaciones de las soluciones de la ecuación cuadrática $\lambda(4\lambda + 3)x^2 - 4\lambda(\lambda + 1)x + (\lambda + 1) = 0$, que proviene de combinar las ecuaciones del sistema (2.10), teniendo en cuenta que los puntos son simétricos:

$$\begin{cases} y = \lambda(3y + 1)x(1 - x) \\ x = \lambda(3x + 1)y(1 - y) \\ x \neq y \end{cases} \quad (2.10)$$

Las soluciones de la ecuación resultante son:

$$x_{5,6} = \frac{2\lambda(\lambda + 1) \pm \sqrt{\lambda(\lambda + 1)(4\lambda^2 - 3)}}{\lambda(4\lambda + 3)}$$

Teniendo en cuenta que no podemos combinar dos soluciones iguales para formar los puntos p_5 y p_6 dado que no pertenecen a la diagonal, es obvio que las dos únicas opciones para los puntos son:

$$p_5 = \left(\frac{2\lambda(\lambda + 1) + \sqrt{\lambda(\lambda + 1)(4\lambda^2 - 3)}}{\lambda(4\lambda + 3)}, \frac{2\lambda(\lambda + 1) - \sqrt{\lambda(\lambda + 1)(4\lambda^2 - 3)}}{\lambda(4\lambda + 3)} \right)$$

$$p_6 = \left(\frac{2\lambda(\lambda + 1) - \sqrt{\lambda(\lambda + 1)(4\lambda^2 - 3)}}{\lambda(4\lambda + 3)}, \frac{2\lambda(\lambda + 1) + \sqrt{\lambda(\lambda + 1)(4\lambda^2 - 3)}}{\lambda(4\lambda + 3)} \right)$$

Definición 2.3. Dado un mapa bidimensional f , se dice que este sufre una bifurcación de Neimark-Hopf cuando un punto fijo pierde la estabilidad debido a que un par de valores propios complejos conjugados del jacobiano sobrepasan el círculo unidad. Dicha bifurcación da lugar a una solución cuasi-periódica.

Los puntos p_5 y p_6 , que tienen los mismos valores propios asociados,

$$\frac{6\lambda^2 - \lambda \pm \sqrt{100\lambda^4 + 100\lambda^3 - 75\lambda^2 - 66\lambda + 9} - 6}{4\lambda + 3}$$

forman una órbita de periodo 2 en el rango $0.866 < \lambda < 0.957$. Cuando se alcanza el punto $\lambda = 0.957$, el sistema pierde la estabilidad mediante una bifurcación de Neimark-Hopf. Dicha bifurcación da lugar

a un par de curvas cerradas invariantes, cuya obtención se escapa de la finalidad de este trabajo, y a la aparición de ventanas periódicas.

Para visualizar lo anterior, componiendo la aplicación consigo misma, damos lugar a una aplicación que tiene a p_5 y p_6 como puntos fijos, podemos ver cómo sus valores propios asociados están dentro del círculo unidad en $0.866 < \lambda < 0.957$ (donde la órbita de período 2 es atractora), y en el punto $\lambda = 0.957$ lo cortan para salir de él.

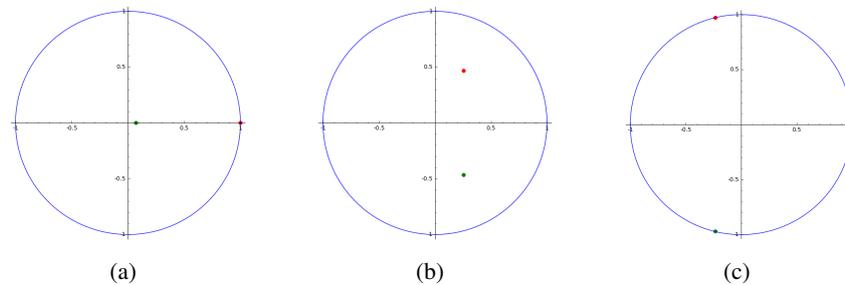


Figura 2.2: Valores propios para (a) $\lambda = 0.866$, (b) $\lambda = 0.9$ y (c) $\lambda = 0.957$.

2.2.2. Comportamiento de las poblaciones

Podemos dar ya, en términos poblacionales, una explicación a las variaciones en las especies, usando lo estudiado en la subsección anterior:

- Para $0 < \lambda < 0.75$, el beneficio mutuo es demasiado pequeño como para mantener a las poblaciones. Ambas desaparecerán tras el paso de pocas generaciones.
- Para $0.75 < \lambda < 0.86$, las poblaciones son capaces de sincronizarse hasta alcanzar un punto de equilibrio si los valores iniciales sobrepasan un cierto límite. Si dicho límite no se supera, ambas poblaciones pueden extinguirse como en el caso anterior.
- Para $0.86 < \lambda < 0.957$, ambas especies oscilan entre los dos mismos puntos fijos. De trata de una órbita de período 2 estable. De nuevo, si las poblaciones iniciales son muy bajas, puede igualmente llegarse a la extinción.
- Para $0.957 < \lambda < 1$, se pierde la sincronización, y ambas especies se pueden mantener con vida oscilando entre infinitos estados. También aparecen ventanas periódicas. Sin embargo, si las poblaciones iniciales son muy cercanas a 0, pueden extinguirse como en el resto de casos.
- Para $1 < \lambda < 1.03$, el comportamiento es análogo al del intervalo anterior, salvo que en este caso la extinción no es factible, sea cual sea la población inicial.
- Para $1.03 < \lambda < 1.08$, el sistema entra en un régimen caótico, y también es posible que el sistema diverja dependiendo de la condición inicial.

En la siguiente gráfica, generada con el programa de código (Anexo VI) que contiene para cada valor de λ los límites de las poblaciones con condición inicial aleatoria, pueden observarse los distintos atractores existentes según las variaciones del parámetro. En ella, las líneas verticales marcan los cambios de intervalo. Podemos por ejemplo ver cómo, tal y como habíamos dicho, las poblaciones tienen la opción de tender a 0 solamente cuando el parámetro es $\lambda < 1$.

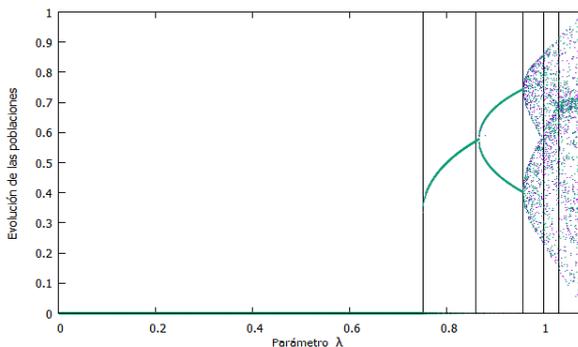


Figura 2.3: Evolución de las poblaciones según mapas logísticos acoplados en situación simbiótica.

2.2.3. El caos en mapas bidimensionales

Como dijimos en el capítulo anterior, una de las herramientas que tenemos para detectar el caos es el exponente de Liapunov. Si queremos estudiarlo, necesitamos pues definir el concepto para el caso bidimensional:

$$\begin{aligned} x_{n+1} &= f_1(x_n, y_n) \\ y_{n+1} &= f_2(x_n, y_n) \end{aligned}$$

Suponiendo que las funciones f_1, f_2 que componen el mapa f son derivables, las ecuaciones variacionales en este caso vienen dadas por:

$$\begin{aligned} u_{n+1} &= \frac{\partial}{\partial x} f_1(x_n, y_n) u_n + \frac{\partial}{\partial y} f_1(x_n, y_n) v_n \\ v_{n+1} &= \frac{\partial}{\partial x} f_2(x_n, y_n) u_n + \frac{\partial}{\partial y} f_2(x_n, y_n) v_n \end{aligned}$$

Definición 2.4. En caso de tener dos mapas logísticos acoplados, el **exponente de Liapunov** viene dado por la fórmula:

$$\sigma := \lim_{T \rightarrow \infty} \frac{1}{T} \ln(|u_T| + |v_T|)$$

Para verificar que efectivamente en el último intervalo las poblaciones entran en un régimen caótico, utilizando la definición 2.4 calculamos los valores del exponente de Liapunov en función del parámetro λ (recordemos que el caos viene determinado por un exponente de Liapunov positivo). Dichos valores se aprecian en las siguientes gráficas, generadas con el programa de código (Anexo VII):

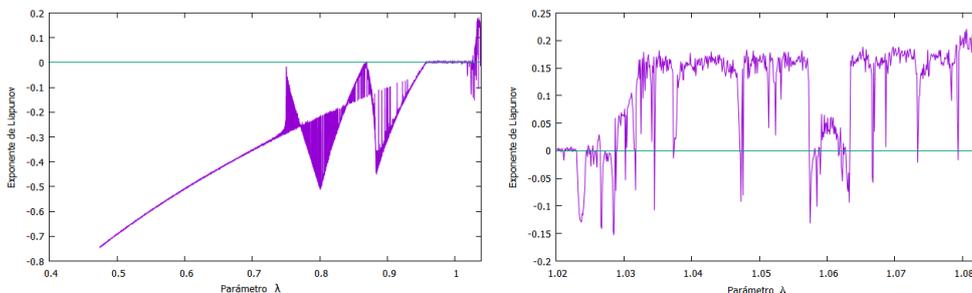


Figura 2.4: Exponente de Liapunov en función del parámetro λ .

En la primera gráfica puede apreciarse cómo cada uno de los máximos relativos, donde $\sigma = 0$, corresponden a aquellos valores del parámetro en los que se ha producido una bifurcación. En la segunda

se ve en detalle la evolución del exponente de Liapunov en la zona caótica (determinada por valores positivos de este, como puede observarse).

Para visualizar el atractor caótico, fijamos el mayor valor posible del parámetro, $\lambda = 1.0834$, para el cual ya hemos dicho que hay un comportamiento caótico. Utilizando la técnica que se empleó para la gráfica 1.3, con la simulación generada por el programa de código (Anexo VIII) obtenemos la siguiente figura en la que se aprecian las probabilidades de visita de cada una de las zonas del atractor:

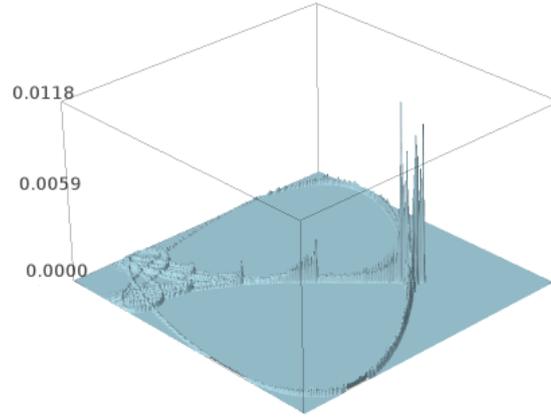


Figura 2.5: Visualización del atractor caótico para $\lambda = 1.0834$ con las respectivas probabilidades de visita.

A continuación, se describirán brevemente, sin tanto detalle como en el caso simbiótico, otros dos modelos poblacionales como son el de predador-presa y el de competición.

2.3. Predador-presa

En este segundo modelo, cada población tiene un tipo de tasa distinta. Supondremos que x_n es el predador, con tasa del tipo μ_1 , y que y_n es la presa, con tasa del tipo μ_2 . La primera se beneficia depredando a la segunda, y la segunda población queda disminuida a raíz de dichos ataques. Hay una interacción de ataque-defensa, y de nuevo cambia el significado de la tasa, que ya no mide el crecimiento si no la interacción entre las especies. El máximo valor que dicha tasa puede alcanzar es 1.21 para que las poblaciones se mantengan en el rango considerado. Las ecuaciones logísticas a combinar en este caso son:

$$x_{n+1} = \lambda(3y_n + 1)x_n(1 - x_n) \quad (2.11)$$

$$y_{n+1} = \lambda(-3x_n + 4)y_n(1 - y_n) \quad (2.12)$$

2.3.1. Cálculos esquemáticos y explicación poblacional

Hallamos los puntos fijos como hicimos en el caso simbiótico, haciendo $y_{n+1} = y_n$, $x_{n+1} = x_n$ en (2.11) y (2.12), que son:

$$p_0 = (0, 0), p_1 = \left(\frac{\lambda - 1}{\lambda}, 0\right), p_2 = \left(0, \frac{4\lambda - 1}{4\lambda}\right)$$

$$p_3 = \left(\frac{8\lambda + 4\sqrt{4\lambda^2 + 9} + 9}{3(2\lambda + \sqrt{4\lambda^2 + 9} + 3)}, \frac{\lambda - \sqrt{4\lambda^2 + 9} - 3}{3\lambda}\right), p_4 = \left(\frac{8\lambda - 4\sqrt{4\lambda^2 + 9} + 9}{3(2\lambda - \sqrt{4\lambda^2 + 9} + 3)}, \frac{\lambda + \sqrt{4\lambda^2 + 9} - 3}{3\lambda}\right)$$

$$J(x_n, y_n) = \begin{pmatrix} \lambda(3y_n + 1)(1 - 2x_n) & 3\lambda x_n(1 - x_n) \\ -3\lambda y_n(1 - y_n) & \lambda(-3x_n + 4)(1 - 2y_n) \end{pmatrix}$$

A partir de la anterior matriz jacobiana correspondiente al sistema, podemos estudiar (análogo al caso simbiótico) los valores propios asociados, y con ello deducir la estabilidad y comportamiento de los puntos. Con los datos que se obtienen, (que omitiremos para simplemente visualizar los resultados), podemos explicar el comportamiento del sistema en términos poblacionales:

- Para $0 < \lambda < 0.25$, el crecimiento de las presas no es suficiente para compensar la combinación de las muertes naturales y las muertes a causa de los predadores, por lo que ambas especies se extinguen.
- Para $0.25 < \lambda < 0.4375$, la población de presas sobrevive en un número limitado, que no es suficiente para servir de alimento a los predadores, que acaban desapareciendo.
- Para $0.4375 < \lambda < 1.051$, la estrategia de los predadores es eficiente y el sistema llega a un equilibrio en el punto p_4 . El crecimiento de λ permite la coexistencia de las especies, hasta que en $\lambda = 1.051$ aparece una inestabilidad.
- Para $1.051 < \lambda < 1.0851$, aparece una cascada de duplicaciones de periodo, a partir de una órbita de periodo 3, lo cual anuncia que se alcanzará una situación caótica. El sistema pasa a ser bi-estable en la vida, ya que dependiendo de las condiciones iniciales, las poblaciones pueden oscilar en las órbitas periódicas, o equilibrarse en el punto fijo.
- Para $1.0851 < \lambda < 1.0997$, es posible una dinámica aperiódica. La cascada de duplicaciones de periodo ha dado lugar a bandas caóticas.
- Para $1.0997 < \lambda < 1.1758$, las poblaciones pueden diverger o alcanzar el equilibrio en p_4 , aunque en algunos casos se pasa antes por una zona caótica.
- Para $1.1758 < \lambda < 1.21$, el sistema sufre una bifurcación de Neimark-Hopf, que da lugar a una curva invariante, y acaba apareciendo una dinámica divergente al superar $\lambda = 1.21$, lo que de nuevo puede interpretarse como algún tipo de catástrofe natural.

2.3.2. Evolución al caos

Modificando los programas de códigos (Anexo VI) y (Anexo VIII) para adaptarlos al nuevo modelo, obtenemos las dos siguientes gráficas. En 2.6.a se observan las tendencias de ambas poblaciones en función del parámetro λ , donde las líneas verticales hacen hincapié en los cambios de intervalo. En 2.6.b, se aprecia el atractor caótico y las probabilidades de ser visitada por cada una de las zonas que lo componen.

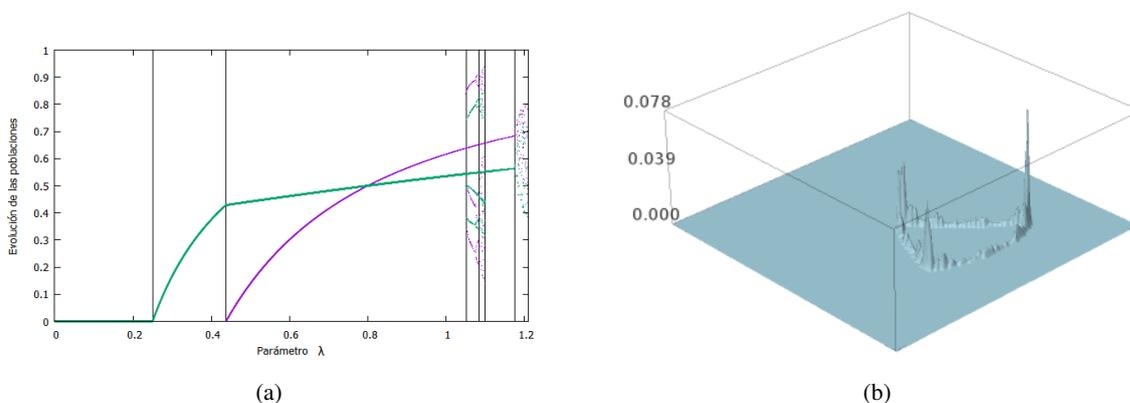


Figura 2.6: (a) Evolución de las poblaciones en función del parámetro λ . (b) Visualización del atractor caótico para $\lambda = 1.21$ con las respectivas probabilidades de visita.

Si queremos verificar las bandas caóticas y el resto de zonas en las que aparece caos, modificando el programa de código (Anexo VII) podemos obtener las siguientes gráficas que representan la evolución del exponente de Liapunov:

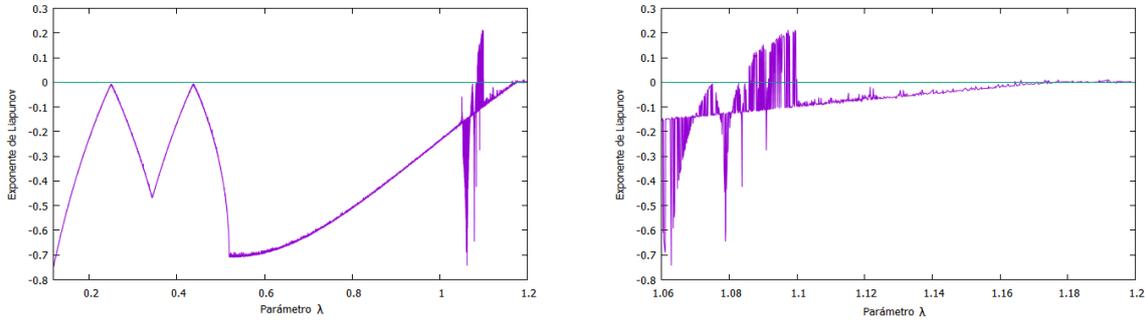


Figura 2.7: Exponente de Liapunov en función del parámetro λ .

2.4. Competición

En el tercer modelo, dos poblaciones compiten por la supervivencia. Ambas especies cuentan con una tasa del tipo μ_2 , y buscan sobrevivir compitiendo con la contraria. En esta ocasión, el significado de la tasa es el nivel de competitividad, y como máximo puede alcanzar el valor $\lambda = 1.206$. Las ecuaciones logísticas a combinar en el caso competitivo son:

$$x_{n+1} = \lambda(-3y_n + 4)x_n(1 - x_n) \quad (2.13)$$

$$y_{n+1} = \lambda(-3x_n + 4)y_n(1 - y_n) \quad (2.14)$$

2.4.1. Cálculos esquemáticos y explicación poblacional

Daremos esta vez directamente los puntos fijos y la matriz jacobiana que han de utilizarse para repetir los cálculos necesarios para obtener de nuevo las gráficas que nos muestran la evolución de las poblaciones y el atractor caótico:

$$p_0 = (0, 0), p_1 = \left(\frac{4\lambda - 1}{4\lambda}, 0\right), p_2 = \left(0, \frac{4\lambda - 1}{4\lambda}\right)$$

$$p_3 = \left(\frac{2(2\sqrt{\lambda^2 + 12\lambda} - 2\lambda - 3)}{3(\sqrt{\lambda^2 + 12\lambda} - \lambda)}, \frac{7\lambda - \sqrt{\lambda^2 + 12\lambda}}{6\lambda}\right), p_4 = \left(\frac{2(2\sqrt{\lambda^2 + 12\lambda} + 2\lambda + 3)}{3(\sqrt{\lambda^2 + 12\lambda} + \lambda)}, \frac{7\lambda + \sqrt{\lambda^2 + 12\lambda}}{6\lambda}\right)$$

$$J(x_n, y_n) = \begin{pmatrix} \lambda(-3y_n + 4)(1 - 2x_n) & -3\lambda x_n(1 - x_n) \\ -3\lambda y_n(1 - y_n) & \lambda(-3x_n + 4)(1 - 2y_n) \end{pmatrix}$$

Con los cálculos omitidos y los datos que estos proporcionan la siguiente explicación, que aparece resumida, en términos poblacionales:

- Para $0 < \lambda < 0.25$, el valor de la tasa es pequeño, lo que no solo implica una ligera competición, sino que también el crecimiento es leve, demasiado como para que las poblaciones consigan sobrevivir.
- Para $0.25 < \lambda < 0.9811$, ambas poblaciones sobreviven y se equilibran en el punto fijo p_4 , independientemente de las condiciones iniciales.

- Para $0.9811 < \lambda < 1.1743$, el punto fijo se vuelve inestable, y se bifurca dando lugar a una órbita de período 2 que hace que las poblaciones oscilen entre dos puntos de la diagonal.
- Para $1.1743 < \lambda < 1.1875$, la órbita anterior se vuelve inestable, y aparece una órbita de período 4 fuera de la diagonal.
- Para $1.1875 < \lambda < 1.1924$, tiene lugar una bifurcación de Hopf, y la órbita de período 4 da lugar a una curva cerrada invariante. El comportamiento es cuasiperiódico.
- Para $1.1924 < \lambda < 1.201$, la dinámica se vuelve compleja y aparece el caos. Las poblaciones visitan sucesivamente un atractor formado por cuatro anillos caóticos.
- Para $1.201 < \lambda < 1.206$, las cuatro bandas caóticas anteriores pasan a convertirse en solo dos grandes bandas caóticas .

2.4.2. Evolución al caos

De manera análoga a como se hizo en los modelos anteriores, obtenemos las siguientes gráficas para el caso competitivo:

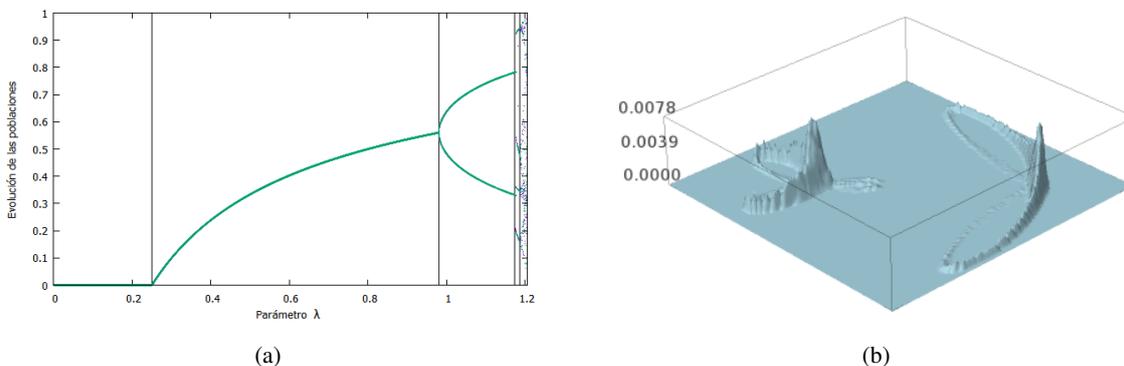


Figura 2.8: (a) Evolución de las poblaciones en función del parámetro λ (b) Visualización del atractor caótico para λ = 1.206 con las respectivas probabilidades de visita.

De nuevo, si queremos verificar las zonas caóticas podemos hacerlo mediante el exponente de Liapunov, modificando, como en el caso anterior, el programa de código (Anexo VII):

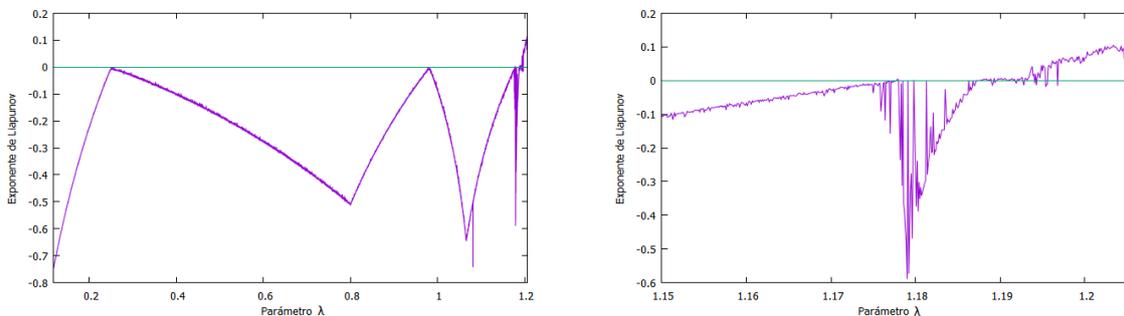


Figura 2.9: Exponente de Liapunov en función del parámetro λ.

Capítulo 3

Resultados propios: alteraciones y consecuencias

En este tercer y último capítulo, vamos a centrarnos en el caso simbiótico. Alteraremos las ecuaciones con las que contábamos en el capítulo anterior, y observaremos a qué modificaciones dan lugar los cambios realizados. Todos los resultados se han obtenido de manera experimental.

3.1. Modificación drástica de la tasa de crecimiento

Recordemos que en el capítulo anterior consideramos ambas tasas de crecimiento de la forma $\mu_\delta(z) = \lambda(3z + 1)$. Vamos a ver qué ocurre si tomamos ahora una tasa de la forma:

$$\mu(z) = \lambda(3z^{1+\delta} + 1)$$

donde puede darse $\delta \in (-\infty, \infty)$, aunque en este caso vamos a considerar el caso $\delta \in [0, 1]$, ya que de este modo tendremos una transición entre el caso ya considerado y el caso $\mu(z) = (3z^2 + 1)$, que es una modificación drástica que vamos a pasar a estudiar.

Primero vamos pues a considerar el caso en el que las ecuaciones a combinar son:

$$x_{n+1} = \lambda(3y_n^2 + 1)x_n(1 - x_n) \quad (3.1)$$

$$y_{n+1} = \lambda(3x_n^2 + 1)y_n(1 - y_n) \quad (3.2)$$

A partir de la simulación numérica se comprueba que solo es válido el rango $\lambda \in (0, 1.2563)$.

3.1.1. Puntos fijos y comportamiento

Como hicimos en los casos anteriores, vamos a calcular los puntos fijos y el jacobiano. En la restricción a la diagonal, imponiendo $x_n = y_n$, contamos con 2 puntos fijos reales:

$$p_0 = (0, 0), p_1 = \left(\frac{1}{3} \left(\frac{8\lambda - 9}{\lambda} \right)^{\frac{1}{3}} + \frac{1}{3}, \frac{1}{3} \left(\frac{8\lambda - 9}{\lambda} \right)^{\frac{1}{3}} + \frac{1}{3} \right)$$

Si ahora nos restringimos a los ejes, obtenemos otros dos puntos fijos reales:

$$p_2 = \left(0, \frac{\lambda - 1}{\lambda} \right), p_3 = \left(\frac{\lambda - 1}{\lambda}, 0 \right)$$

El resto de puntos fijos no se tienen en cuenta en el modelo, ya que son números complejos. Ahora calculamos la matriz jacobiana:

$$J(x_n, y_n) = \begin{pmatrix} \lambda(3y_n^2 + 1)(1 - 2x_n) & 6\lambda x_n y_n(1 - x_n) \\ 6\lambda x_n y_n(1 - y_n) & \lambda(3x_n^2 + 1)(1 - 2y_n) \end{pmatrix}$$

La primera diferencia con el modelo del capítulo anterior la encontramos con el intervalo en el que la extinción es la única posibilidad:

$$J(p_0) = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \quad (3.3)$$

El único valor propio asociado es λ , lo que hace que la extinción se dé en el intervalo $\lambda \in (0, 1)$, donde p_0 es estable. La diferencia es que esta vez, en dicho intervalo es lo único que puede ocurrir. Esto puede observarse en la gráfica siguiente, generada con el programa de código (Anexo IX), donde se muestra la tendencia de la población según varía el parámetro:

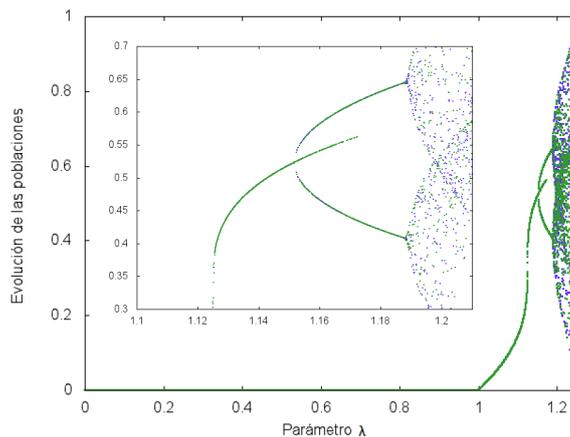


Figura 3.1: Tendencia de las poblaciones en función del parámetro λ .

Más detalladamente, el comportamiento que siguen ambas poblaciones en distintos intervalos es:

- Para $0 < \lambda < 1$, la tasa no es suficiente para mantener a las poblaciones con vida, y la única posibilidad es la extinción.
- Para $1 < \lambda < 1.1524$, las poblaciones consiguen sincronizarse y se equilibran en el punto fijo p_1 .
- En $\lambda = 1.1524$ aparece una bifurcación, dando lugar a una órbita de periodo 2 estable, y para $1.1524 < \lambda < 1.1723$, el sistema se vuelve bi-estable: puede visitar la órbita de periodo 2, o mantenerse en equilibrio en el punto fijo p_1 .
- Para $1.1723 < \lambda < 1.1886$, se pierde la biestabilidad, y la única opción para cualquier condición inicial es la órbita de periodo 2, aunque dependiendo de la condición inicial el sistema puede diverger.
- Para $\lambda > 1.1886$, la situación se vuelve caótica.

3.1.2. Atractores caóticos

Otra manera de observar el comportamiento de las poblaciones es representar los atractores para cada uno de los valores de λ . Esto nos proporciona de algún modo una vista superior de la evolución, desde la extinción en $p_0 = (0, 0)$, hasta el atractor caótico. En la gráfica, cada color corresponde a un valor distinto del parámetro λ .

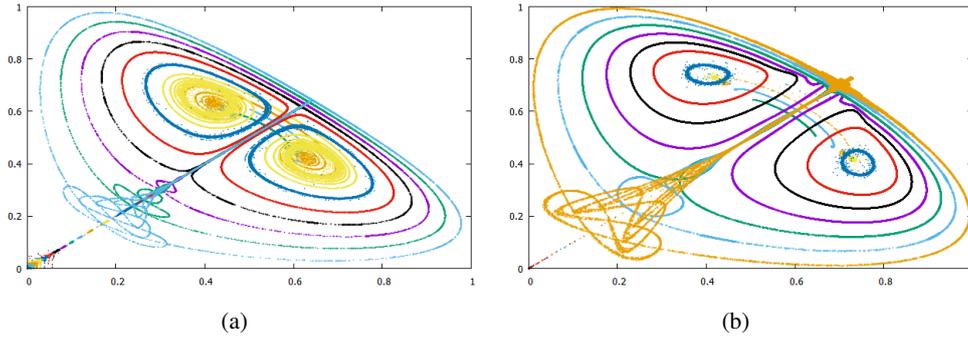


Figura 3.2: Atractores según varía el parámetro λ para (a) caso modificado ($\delta = 1$) y (b) caso original ($\delta = 0$).

Ahora, como hicimos en los capítulos anteriores, para detectar las zonas caóticas vamos a emplear el exponente de Liapunov, con la definición 2.4. El mayor valor es $\sigma = 0.34824$ (que destaca en la gráfica, aunque sin ningún motivo aparente), y es obtenido para el valor $\lambda = 1.19270$. También de los datos generados podemos deducir que, aunque hay picos positivos aislados, la situación caótica aparece con el valor $\lambda = 1.18860$, donde el exponente de Liapunov es $\sigma = 0.00312$.

Todo lo anterior se puede observar en las gráficas siguientes (completa y ampliación de la zona caótica, respectivamente). Los datos se obtienen modificando el programa de código (Anexo VII).

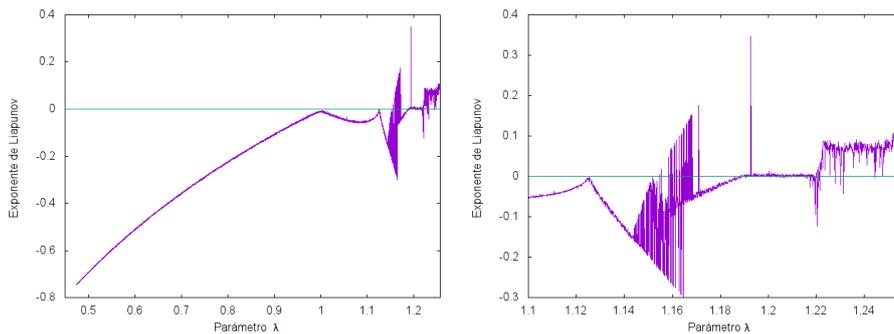


Figura 3.3: Exponente de Liapunov en función del parámetro λ .

3.1.3. Semejanzas y diferencias con el modelo simbiótico original

Se puede ver casi simplemente comparando las gráficas 2.3 y 3.1 que el modelo original y el modificado comparten los esquemas de bifurcación en el estado vivo.

Sin embargo, hay importantes diferencias entre ambas situaciones:

- En el modelo original había **biestabilidad** entre los estados de vida y muerte, mientras que en el nuevo modelo esta biestabilidad vida-muerte se pierde, y pasa a ser entre dos estilos distintos de supervivencia.
- En el modelo original la **vida** aparecía en un **valor finito**, mientras que ahora la vida es continua, empieza en un valor nulo relevando a la extinción. Además, en un momento dado experimenta un crecimiento rápido.
- Desaparece la **lizada caótica** que se aprecia en la esquina superior derecha en la gráfica 2.5, que es la zona en la que se acumulaba la mayor **probabilidad de visita**. ¿Dónde se acumula ahora dicha probabilidad?

Con la siguiente gráfica tridimensional, análoga a la 2.5, podemos responder a la última pregunta. La probabilidad resulta seguir acumulándose en la antigua lazada caótica, que ahora es una *zona en T*, y de manera más drástica, ya que si antes el mayor valor alcanzado era 0.0118, ahora es 0.28, haciendo que las probabilidades de visita en el resto de zonas sean mucho más cercanas a 0 que antes.

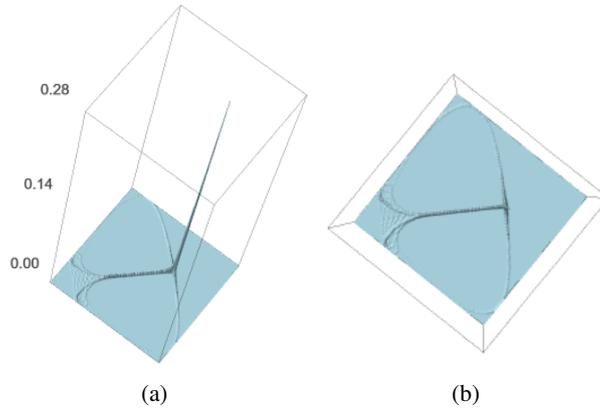


Figura 3.4: Visualización del atractor caótico para $\lambda = 1.256$ en vista (a) lateral y (b) superior.

3.2. Modificación gradual de la tasa de crecimiento: un punto intermedio

De los dos estudios para $\delta = 0$ y $\delta = 1$ que se han hecho, parece deducirse que todos los valores $\delta \in (0, 1)$ restantes proporcionarán un cambio gradual de una situación a otra. A continuación va a estudiarse el modelo para el caso intermedio $\delta = 0.5$, para confirmar que hay un cambio gradual en los puntos intermedios, y en la siguiente sección se buscará el valor del parámetro δ para el que se pierde la biestabilidad vida-muerte.

Ahora vamos a considerar la tasa de crecimiento $\mu(z) = \lambda(3z^{1+\frac{1}{2}} + 1)$, con lo que queda:

$$x_{n+1} = \lambda(3y_n^{\frac{3}{2}} + 1)x_n(1 - x_n) \tag{3.4}$$

$$y_{n+1} = \lambda(3x_n^{\frac{3}{2}} + 1)y_n(1 - y_n) \tag{3.5}$$

3.2.1. Comportamiento

Esta vez vamos directamente a observar el comportamiento, ya que se ha repetido el proceso en varias ocasiones. Con simulación numérica se deduce que solo tienen sentido los valores $\lambda \in (0, 1.1575)$. La gráfica que nos muestra la evolución es la siguiente:

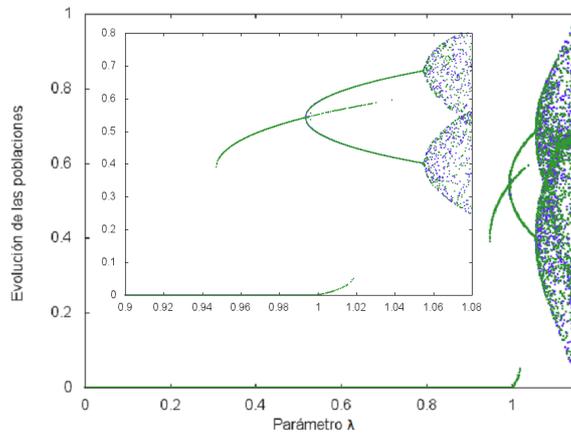


Figura 3.5: Tendencia de las poblaciones en función del parámetro λ .

El comportamiento que siguen según el parámetro λ , como puede verse en la gráfica anterior es, resumidamente, el siguiente:

- Para $0 < \lambda < 0.9472$, la única opción es la extinción.
- Para $0.9472 < \lambda < 0.9939$, dependiendo de la condición inicial se da la extinción, o puede darse equilibrio de ambas poblaciones en el mismo punto.
- Para $0.9939 < \lambda < 1$, según la condición inicial hay tres opciones: la extinción, el equilibrio en un punto, o la entrada en una órbita de periodo 2.
- Para $1 < \lambda < 1.0185$, vuelve a haber tres opciones, el equilibrio y la órbita de periodo 2 del intervalo anterior, y el equilibrio en un nuevo punto que parte desde la extinción de manera continua.
- Para $1.0185 < \lambda < 1.0303$, las poblaciones pueden equilibrarse en el punto que apareció en los dos casos anteriores, o en la órbita de periodo 2.
- Para $1.0303 < \lambda < 1.0537$, ambas poblaciones caen en la órbita de periodo 2.
- Finalmente, para $\lambda > 1.0537$, el comportamiento pasa a ser caótico.

3.2.2. Atractores caóticos

Antes de visualizar el atractor caótico, para comprobar visualmente que es una transición entre los dos estudiados anteriormente, vamos a corroborar mediante el exponente de Liapunov el comienzo de la zona caótica.

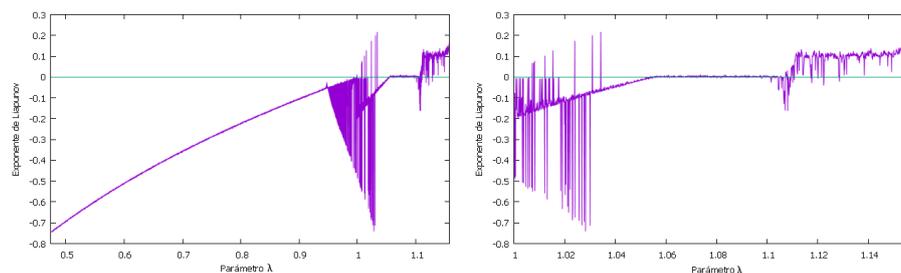


Figura 3.6: Exponente de Liapunov en función del parámetro λ .

Efectivamente, obviando de nuevo los picos positivos aislados, el caos aparece con un exponente de Liapunov $\sigma = 8.1744 \cdot 10^{-4}$, correspondiente a $\lambda = 1.0537$, como ya habíamos predicho.

Ahora vamos a visualizar la evolución de los atractores hasta llegar al atractor caótico, para compararlo visualmente con los de los modelos original y drásticamente modificado.

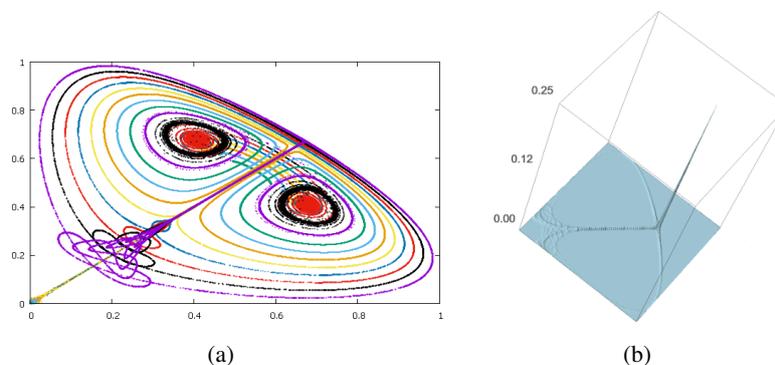


Figura 3.7: (a) Atractores según varía el parámetro λ y (b) atractor caótico para el valor $\lambda = 1.1575$ con sus respectivas probabilidades de visita.

3.2.3. Semejanzas y diferencias con los anteriores modelos

Como la intuición ya nos podía indicar, el caso $\delta = 0.5$ es simplemente un corte intermedio en el cambio gradual desde $\delta = 0$ y $\delta = 1$. Es decir, entre otras cosas, el máximo valor de λ válido, $\lambda = 1.1577$ es intermedio entre los anteriores ($\lambda = 1.0834$ y $\lambda = 1.2563$), y la gráfica es una transición entre las anteriores. Otros aspectos a destacar son:

- La lazada del atractor caótico que había en el caso $\delta = 0$, y que para $\delta = 1$ habíamos visto que desaparecía, para $\delta = 0.5$ apenas queda representada por un pequeño pico. Además, como en el caso $\delta = 0$, la probabilidad de visita está drásticamente acumulada en la nueva *zona en T*.
- Se mantiene la biestabilidad vida-muerte, como en el caso $\delta = 0$, y además aparece biestabilidad en la vida, como en el caso $\delta = 1$.

3.3. Conclusiones

En el caso general, las ecuaciones a combinar son:

$$x_{n+1} = \lambda(3y_n^{1+\delta} + 1)x_n(1 - x_n) \quad (3.6)$$

$$y_{n+1} = \lambda(3x_n^{1+\delta} + 1)y_n(1 - y_n) \quad (3.7)$$

El jacobiano dependiente del parámetro δ tiene la forma siguiente:

$$J_\delta(x_n, y_n) = \begin{pmatrix} \lambda(3y_n^{1+\delta} + 1)(1 - 2x_n) & 3\lambda(1 + \delta)y_n^\delta x_n(1 - x_n) \\ 3\lambda(1 + \delta)x_n^\delta y_n(1 - y_n) & \lambda(3x_n^{1+\delta} + 1)(1 - 2y_n) \end{pmatrix}$$

Si evaluamos en $p_0 = (0, 0)$, que es un punto fijo para cualquier valor de δ , al evaluar obtenemos siempre el mismo resultado:

$$J_\delta(p_0) = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$$

Por lo tanto, hay algo común a todos los modelos. Independientemente del valor de δ , la extinción tiene lugar en el intervalo $\lambda \in (0, 1)$. Nos interesa saber en qué valor del parámetro δ se pierde la biestabilidad vida-muerte, y en cuál aparece la biestabilidad en la vida. Además, estudiaremos también para qué valor δ desaparece la lazada del atractor caótico.

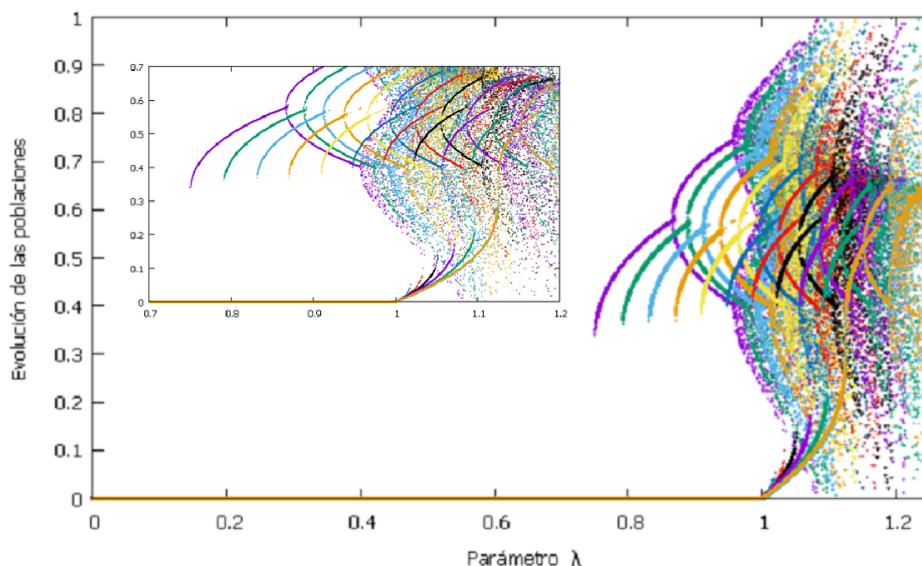


Figura 3.8: Cambios en la evolución de las poblaciones según el parámetro λ , donde cada color corresponde a un valor del parámetro δ distinto.

Realizando sucesivas simulaciones, cada vez para valores del parámetro con más cifras decimales (para ajustar el valor lo máximo posible), obtenemos que se pierde la biestabilidad vida-muerte en el valor $\delta = 0.6419$. En cuanto a la biestabilidad en la vida, con un procedimiento similar, hallamos que aparece en el valor $\delta = 0.0350$.

Ahora, vamos a estudiar para qué valor desaparece la lazada del atractor caótico. Con simulación numérica, podemos obtener cortes del intervalo $\delta \in [0, 1]$, y sus respectivos valores máximos de λ , que son necesarios para representar el atractor caótico.

Parámetro δ	Máximo λ	Parámetro δ	Máximo λ
0	1.0834	0.55	1.1665
0.05	1.0904	0.6	1.1752
0.1	1.0968	0.65	1.1847
0.15	1.1038	0.7	1.1941
0.2	1.1105	0.75	1.2037
0.25	1.1176	0.8	1.2138
0.3	1.1252	0.85	1.2240
0.35	1.1329	0.9	1.2345
0.4	1.1410	0.95	1.2453
0.45	1.1491	1	1.2563
0.5	1.1575		

Cuadro 3.1: Tabla con 20 valores del parámetro δ y sus correspondientes valores máximos de λ .

Con los datos de la tabla 3.1, podemos representar pues los atractores caóticos correspondientes a cada modelo con tasa $\mu_\delta(z) = \lambda(3z^{1+\delta} + 1)$, para su correspondiente máximo λ . En la siguiente figura podemos observar la evolución de dichos atractores sucesivos según variamos el parámetro λ :

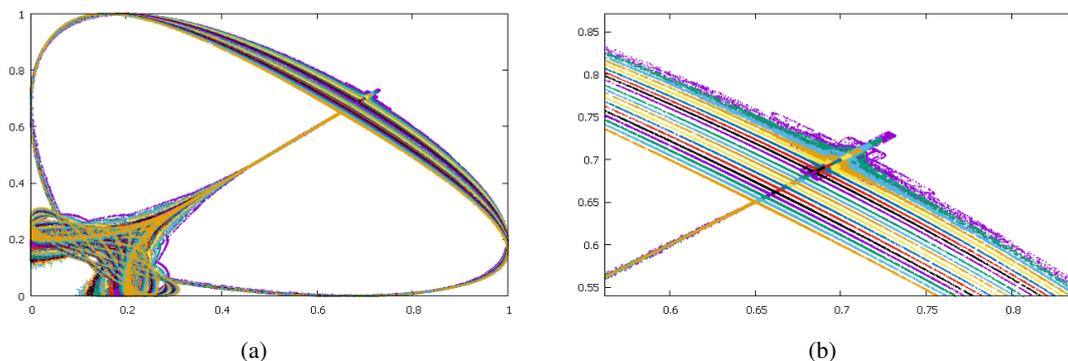


Figura 3.9: (a) Evolución de los atractores caóticos y (b) ampliación de la zona de la lazada caótica, donde cada color corresponde a un valor de δ distinto.

Tras deducir numéricamente que la pérdida de la lazada se encuentra en el intervalo $\delta \in (0.55, 0.60)$, repitiendo el proceso con valores más ajustados, se llega a la conclusión de que la completa desaparición de la lazada y consiguiente aparición de la *zona en T* se dan en torno al valor $\delta = 0.5990$.

Así pues, el aumento de δ provoca una aparición de la biestabilidad para valores de λ mayores debido a que el acoplo simbiótico es más pequeño. Además, una vez alcanzado el caos, según aumenta el valor de δ , se va concentrando alrededor de un solo punto la probabilidad de visita de las zonas del correspondiente atractor caótico.

Bibliografía

- [1] [AD] M. AUSLOOS, M. DIRICKX (editores), *The Logistic Map and the Route to Chaos: from the Beginnings to Modern Applications*, colección Understanding Complex Systems, Springer, 2006.
- [2] [ASY] K. T. ALLIGOOD, T.D. SAUER, J. A. YORKE, *Chaos: an Introduction to Dynamical Systems*, Textbooks in Mathematical Sciences, Springer, 1996.
- [3] [D] R. L. DEVANEY, *An Introduction to Chaotic Dynamical Systems*, 2^a edición, Addison-Wesley Publishing Company, 1989.
- [4] [FZL] FU JING-CHAO, ZHANG ZHONG-HUA, LIU CHUN-LI, *Chaos Control of a Discrete Coupled Logistic Model for the Symbiotic Interaction of Two Species*, Chinese Control and Decision Conference, 2011.
- [5] [LF] R. LÓPEZ-RUIZ, D. FOURNIER-PRUNARET, *Complex Behavior in a Discrete Coupled Logistic Model for the Symbiotic Interaction of Two Species*, Mathematical Biosciences and Engineering 1: págs. 307-324, 2004.
- [6] [LP] R. LÓPEZ-RUIZ, C. PÉREZ-GARCÍA, *Dynamics of Two Logistic Maps with a Multiplicative Coupling*, International Journal of Bifurcation and Chaos, Vol. 2, No. 2: págs 421-425, 1992.
- [7] [S] M. SAMBASIVAM, *Dynamical Behavior of Logistic Equation through Symbiosis Model*, International Journal of Scientific and Research Publications, Volume 3, Issue 6, June 2013.
- [8] [W] W. STEEB, *The nonlinear workbook*, 1^a edición, World Scientific, Singapur, 1999.
- [9] [WK] T. WILLIAMS, C. KELLEY, *Gnuplot 4.6: An Interactive Plotting Program*, versión 4.6.6, 2014.

Anexos

Aparecen a continuación parte de los códigos empleados para generar las simulaciones que dan lugar a las gráficas que aparecen en la memoria. Tras generar los datos de la simulación y plasmarlos en un fichero (fase realizada en lenguaje JAVA), estos se han representado usando GNUplot y SAGE-math.

Anexo I: Atractor caótico del mapa de Henón

```
package henonmap;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import static java.lang.Math.*;

public class HenonMap {

    public static double iteracionx(double xn, double yn){
        return 1-1.4*pow(xn,2)+yn;
    }

    public static double iteraciony(double xn, double yn){
        return 0.3*xn;
    }

    public static void main(String[] args) throws IOException {
        int limit=100000;
        double xn=Math.random(), yn=Math.random(), xn1, yn1;
        FileWriter fichero;
        fichero = new FileWriter("henon_attractor.txt");
        PrintWriter out = new PrintWriter(
            new FileWriter("henon_attractor.txt"), true);

        for(int i=0; i<limit; i++){
            xn1=xn; yn1=yn;
            xn=iteracionx(xn1,yn1);
            yn=iteraciony(xn1,yn1);
            out.println(xn + "\t" + yn);
        }

        fichero.close();
    }
}
```

Anexo II: Valores para el atractor caótico del mapa de Henón

Antes de hacer la simulación que nos genera el atractor caótico del Henon map, necesitamos fijar unas condiciones iniciales y conocer entre que valores máximos y mínimos nos encontramos. Así, podremos a continuación hacer un enrejillado transformando nuestro intervalo en el intervalo del enrejillado.

```
package grafica_atractor_henon_valores;
import static java.lang.Math.pow;

public class Grafica_atractor_henon_valores {

    public static double iteracionx(double xn, double yn){
        return 1-1.4*pow(xn,2)+yn;
    }

    public static double iteraciony(double xn){
        return 0.3*xn;
    }

    public static void main(String[] args) {

        double xn=0.5, yn=0.5, xn1, yn1;
        double maxx,maxy,minx,miny;
        int N=100, Ntot=100000;

        for(int z=0; z<=100; z++){
            xn1=xn; yn1=yn; xn=iteracionx(xn1,yn1); yn=iteraciony(xn1);
        }

        maxx=xn; maxy=yn; minx=xn; miny=yn;

        for(int z=0; z<=Ntot; z++){
            xn1=xn; yn1=yn;
            xn=iteracionx(xn1,yn1);
            yn=iteraciony(xn1);

            if(xn>maxx) maxx=xn;

            if(xn<minx) minx=xn;

            if(yn>maxy) maxy=yn;

            if(yn<miny) miny=yn;

        }

        System.out.println(minx+" "+maxx+" "+miny+" "+maxy);

    }
}
```

Anexo III: Atractor caótico del mapa de Henón 3d

A continuación aparece el código JAVA para generar la simulación que a cada uno de los 10000 puntos de nuestro enrejillado de $[0,1] \times [0,1]$ le asocia su probabilidad de atracción.

```

package grafica_atractora_henon;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.lang.Math;
import static java.lang.Math.abs;
import static java.lang.Math.pow;

public class Grafica_atractora_henon {

    public static double iteracionx(double xn, double yn){
        return 1-1.4*pow(xn,2)+yn;
    }

    public static double iteraciony(double xn){
        return 0.3*xn;
    }

    public static void main(String[] args) throws IOException {

        double xn=0.5, yn=0.5, xn1, yn1;
        double a=-1.2846594639345508;
        double b=1.2729729582739004;
        double c=-0.38539783918036524;
        double d=0.3818918874821701;

        int N=100;
        int Ntot=100000;

        FileWriter fichero;
        fichero = new FileWriter("grafica_atractor_henon.txt");
        PrintWriter out = new PrintWriter(
            new FileWriter("grafica_atractor_henon.txt"), true);

        for(int z=0; z<=100; z++){
            xn1=xn; yn1=yn;
            xn=iteracionx(xn1,yn1);
            yn=iteraciony(xn1);
        }

        double [] [] mat=new double[N] [N];

        for(int z=0; z<=Ntot; z++){
            xn1=xn; yn1=yn;
            xn=iteracionx(xn1,yn1);
            yn=iteraciony(xn1);
            //Trasladamos el intervalo a [0,100]
            int corx=(int) Math.floor(((xn-a)/(b-a))*(N-1));
            int cory=(int) Math.floor(((yn-c)/(d-c))*(N-1));
            mat[corx] [cory]++;
        }

        for(int i=0; i<N; i++){

```

```

        for(int j=0; j<N; j++){
            mat[i][j]=mat[i][j]/Ntot;
        }
    }

    for(int i=0; i<N; i++){
        for(int j=0; j<N; j++){
            out.println(i + ";" + j + ";" + mat[i][j]);
        }
    }

    fichero.close();

}
}

```

El siguiente paso es cargar nuestro fichero de datos en SAGE-math y representar la simulación. Aparece el código a continuación para tener una muestra del proceso (en el resto de casos no aparecerá).

```

datos2=DATA+'grafica_atractor_henon.txt'

f2 = open(datos2, 'r')

xArr=[]
yArr=[]
zArr=[]

line=f2.readline()

while(line !=''):
    xyz=line.split(';')
    xArr.append(float(xyz[0]))
    yArr.append(float(xyz[1]))
    zArr.append(float(xyz[2]))
    line=f2.readline()

list_plot3d(zip(xArr,yArr,zArr),aspect_ratio=(1,1,2000))

```

Anexo IV: Evolución del mapa logístico

```
package logisticmap1;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

public class LogisticMap2 {

    static double iteracion(double xn,double mu){
        return 0.5*xn*(1.0-xn);
    }

    public static void main(String[] args) throws IOException {

        int n=1000;

        FileWriter fichero;
        fichero = new FileWriter("ecuación_logística.txt");
        PrintWriter out = new PrintWriter(
            new FileWriter("ecuación_logística.txt"), true);

        for (double mu=0;mu<=4;mu+=0.1/n){

            double xn=0.3;

            for (int i=0;i<1000;i++){
                xn=iteracion(xn,mu);
            }

            xn=iteracion(xn,mu);
            out.println(mu + "\t" + xn);
        }

        fichero.close();
    }
}
```

Anexo V: Exponente de Liapunov del mapa logístico

```
package liapunov2;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import static java.lang.Math.*;

public class Liapunov2 {

    static double iteracionx(double xn,double mu){
        return mu*xn*(1.0-xn);
    }

    static double iteraciony(double yn,double xn,double mu){
        return (mu-2*mu*xn)*yn;
    }

    public static void main(String[] args) throws IOException {

        FileWriter fichero;
        fichero = new FileWriter("liapunov_ecuación_logística.txt");
        PrintWriter out = new PrintWriter(
            new FileWriter("liapunov_ecuación_logística.txt"), true);

        for (double mu=0;mu<=4;mu+=0.1/1000){

            double xn=0.3; double yn=1.0;

            for (int i=0;i<1000;i++){
                xn=iteracionx(xn,mu);
                yn=iteraciony(yn,xn,mu);
            }

            xn=iteracionx(xn,mu);
            yn=iteraciony(yn,xn,mu);

            double lambda=log(abs(yn))/1000;
            out.println(mu + "\t" + lambda);

        }

        fichero.close();
    }
}
```

Anexo VI: Evolución de las poblaciones en el caso simbiótico

```
package lambda1;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

public class Lambda1 {

    static double iteracionx(double l,double xn, double yn){
        return l*(3*yn+1)*xn*(1-xn);
    }

    static double iteraciony(double l,double xn, double yn){
        return l*(3*xn+1)*yn*(1-yn);
    }

    public static void main(String[] args) throws IOException {

        FileWriter ficherox,ficheroy;
        ficherox = new FileWriter("lambda1x.txt");
        ficheroy = new FileWriter("lambda1y.txt");
        PrintWriter outx = new PrintWriter(
            new FileWriter("lambda1x.txt"), true);
        PrintWriter outy = new PrintWriter(
            new FileWriter("lambda1y.txt"), true);

        for (double mu=0;mu<=1.084;mu+=0.0001){

            double xn=Math.random();
            double yn=Math.random();
            double xn1=0, yn1=0

            for(int n=0; n<2000; n++){
                xn1=xn; yn1=yn;
                xn=iteracionx(mu,xn1,yn1);
                yn=iteraciony(mu,xn1,yn1);
            }

            outx.println(mu + "\t" + xn);
            outy.println(mu + "\t" + yn);
        }

        ficherox.close();
        ficheroy.close();
    }
}
```

Anexo VII: Exponente de Liapunov bidimensional (caso simbiótico)

```

package liapunovbidimensional;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import static java.lang.Math.*;

public class LiapunovBidimensional {

    static double iteracionx(double xn, double yn, double l){
        return l*(3*yn+1)*xn*(1-xn);
    }

    static double iteraciony(double xn, double yn, double l){
        return l*(3*xn+1)*yn*(1-yn);
    }

    static double iteracionu(double xn, double yn, double un, double vn, double l){
        return l*(3*yn+1)*(1-2*xn)*un+3*l*xn*(1-xn)*vn;
    }

    static double iteracionv(double xn, double yn, double un, double vn, double l){
        return 3*l*yn*(1-yn)*un+l*(3*xn+1)*(1-2*yn)*vn;
    }

    public static void main(String[] args) throws IOException {

        FileWriter fichero;
        fichero = new FileWriter("liapunov_bidimensional.txt");
        PrintWriter out = new PrintWriter(
            new FileWriter("liapunov_bidimensional.txt"), true);

        for (double l=0 ; l<=1.083 ; l+=0.0001){
            double xn=Math.random(); double yn=Math.random();
            double un=Math.random(); double vn=Math.random();
            double xn1=0, yn1=0, un1=0, vn1=0;

            for (int i=0;i<1000;i++){
                xn1=xn; yn1=yn; un1=un; vn1=vn;
                xn=iteracionx(xn1,yn1,l);
                yn=iteraciony(xn1,yn1,l);
                un=iteracionu(xn1,yn1,un1,vn1,l);
                vn=iteracionv(xn1,yn1,un1,vn1,l);
            }
            out.println(l + "\t" +
                (Math.log(Math.abs(un)+Math.abs(vn)))/(1000));
        }

        fichero.close();
    }
}

```

Anexo VIII: Atractor caótico 3d del caso simbiótico

```

package graficaatractor;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.lang.Math;

public class GraficaAtractor {

    static double iteracionx(double xn, double yn, double l){
        return l*(3*yn+1)*xn*(1-xn);
    }
    static double iteraciony(double xn, double yn, double l){
        return l*(3*xn+1)*yn*(1-yn);
    }

    public static void main(String[] args) throws IOException {

        double x0=Math.random(), y0=Math.random();
        double xn=x0, yn=y0, xn1, yn1;
        int N=100, Ntot=100000;
        double l=1.0834;
        double [][] mat=new double[N][N];
        FileWriter fichero;
        fichero = new FileWriter("grafica_atractor.txt");
        PrintWriter out = new PrintWriter(
            new FileWriter("grafica_atractor.txt"), true);

        for(int b=0; b<=100; b++){
            xn1=xn; yn1=yn; xn=iteracionx(xn1,yn1,l); yn=iteraciony(xn1,yn1,l);
        }

        for(int b=0; b<=Ntot; b++){
            xn1=xn; yn1=yn; xn=iteracionx(xn1,yn1,l); yn=iteraciony(xn1,yn1,l);
            mat[(int) Math.floor(xn*N)][(int) Math.floor(yn*N)]++;
        }

        for(int i=0; i<N; i++){
            for(int j=0; j<N; j++){
                mat[i][j]=mat[i][j]/Ntot;
            }
        }

        for(int i=0; i<N; i++){
            for(int j=0; j<N; j++){
                out.println(i + ";" + j + ";" + mat[i][j]);
            }
        }
        fichero.close();
    }
}

```

Anexo IX: Evolución caso simbiótico modificado

```
package cap3delta1;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

public class Cap3Delta1 {

    static double iteracionx(double l,double xn, double yn){
        return l*(3*yn*yn+1)*xn*(1-xn);
    }

    static double iteraciony(double l,double xn, double yn){
        return l*(3*xn*xn+1)*yn*(1-yn);
    }

    public static void main(String[] args) throws IOException {

        FileWriter ficherox,ficheroy;
        ficherox = new FileWriter("pobx.txt");
        ficheroy = new FileWriter("poby.txt");
        PrintWriter outx = new PrintWriter(
            new FileWriter("pobx.txt"), true);
        PrintWriter outy = new PrintWriter(
            new FileWriter("poby.txt"), true);

        for (double l=0;l<=1.25629;l+=0.00001){
            double xn1=0, yn1=0;
            double xn=Math.random();
            double yn=Math.random();
            for(int n=0; n<1000; n++){
                xn1=xn; yn1=yn;
                xn=iteracionx(l,xn1,yn1);
                yn=iteraciony(l,xn1,yn1);
            }
            outx.println(l + "\t" + xn);
            outy.println(l + "\t" + yn);
        }

        ficherox.close();
        ficheroy.close();

    }

}
```

Anexo X: Sucesión de atractores

```

package cap3atraccion;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.text.DecimalFormat;

public class Cap3Atraccion {

    static double iteracionx(double l,double xn, double yn){
        return l*(3*yn*yn+1)*xn*(1-xn);
    }

    static double iteraciony(double l,double xn, double yn){
        return l*(3*xn*xn+1)*yn*(1-yn);
    }

    public static void main(String[] args) throws IOException {

        int indice=1;
        //Basta con elegir uno de los valores de lambda que llevan a la
        //extinción, tras comprobar que efectivamente todos lo hacen
        FileWriter fichero0;
        fichero0 = new FileWriter("lambda0.txt");
        PrintWriter out0 = new PrintWriter(
            new FileWriter("lambda0.txt"), true);

        for(int n=0; n<=20000; n++){
            double xn=Math.random(), yn=Math.random();

            for(int b=0; b<=100; b++){
                double xn1=0, yn1=0;
                xn1=xn; yn1=yn;
                xn=iteracionx(0.5,xn1,yn1);
                yn=iteraciony(0.5,xn1,yn1);
            }
            out0.println(xn + "\t" + yn);
        }

        fichero0.close();

        for(double i=1; i<1.3; i+=0.05){
            indice++;
            double xn1=0, yn1=0;

            FileWriter fichero;
            fichero = new FileWriter("lambda"+indice+".txt");
            PrintWriter out = new PrintWriter(
                new FileWriter("lambda"+indice+".txt"), true);

            for(int n=0; n<=20000; n++){
                double xn=Math.random(), yn=Math.random();

                for(int b=0; b<=100; b++){
                    xn1=xn; yn1=yn;

```

```
        xn=iteracionx(i,xn1,yn1);
        yn=iteraciony(i,xn1,yn1);
    }
    out.println(xn + "\t" + yn);
}
fichero.close();
}
}
```

A continuación, para generar la gráfica de la figura 3.2 simplemente representamos en gnuplot todos los ficheros juntos. Para crear un gif que muestre la evolución, modificamos el código de arriba, generando un mayor número de ficheros, y después empleamos el siguiente script:

```
set terminal gif animate enhanced transparent delay 10

set output "attractorinator.gif"
unset key
set style line 1 lc rgb '#ff9900' lt 1 lw 2 pt 5 # blue

do for[i=1:59]{
    set xrange [0:1]
    set yrange [0:1]
    plot "lambda".i.".txt" w p ls 1
}

set output
```
