



# UNIVERSIDAD DE GRANADA

Escuela Técnica Superior de Ingenierías Informática y  
Telecomunicación  
Facultad de Ciencias

## DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS

TRABAJO DE FIN DE GRADO

# Sistemas dinámicos discretos e iteradas de funciones complejas: fractales, caos y estabilidad

Presentado por:  
David Armenteros Soto

Tutor:  
Pedro José Torres Villarroya  
*Departamento de Matemática Aplicada*

Curso académico 2021-2022



# Sistemas dinámicos discretos e iteradas de funciones complejas: fractales, caos y estabilidad

David Armenteros Soto

David Armenteros Soto *Sistemas dinámicos discretos e iteradas de funciones complejas: fractales, caos y estabilidad*.

Trabajo de fin de Grado. Curso académico 2021-2022.

**Responsable de  
tutorización**

Pedro José Torres Villarroya  
*Departamento de Matemática Aplicada*

Doble Grado en Ingeniería  
Informática y Matemáticas

Escuela Técnica Superior  
de Ingenierías Informática  
y Telecomunicación  
Facultad de Ciencias

Universidad de Granada

**DECLARACIÓN DE ORIGINALIDAD**

D./Dña. David Armenteros Soto

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Grado (TFG), correspondiente al curso académico 2021-2022, es original, entendida esta, en el sentido de que no ha utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a 2 de diciembre de 2021

Fdo: David Armenteros Soto



# Índice general

<b>Índice de figuras</b>	<b>IX</b>
<b>Agradecimientos</b>	<b>XIII</b>
<b>Resumen</b>	<b>XV</b>
<b>Summary</b>	<b>XVII</b>
<b>Introducción</b>	<b>XIX</b>
0.1.    Motivación . . . . .	XX
0.2.    Objetivos . . . . .	XX
0.3.    Estructura . . . . .	XXI
<b>1. Sistemas dinámicos discretos</b>	<b>1</b>
1.1.    Primeras definiciones . . . . .	1
1.1.1.    Puntos fijos . . . . .	1
1.1.2.    Puntos Periódicos . . . . .	6
1.2.    Relaciones de recurrencia . . . . .	7
1.2.1.    Ecuaciones en diferencias de primer orden . . . . .	7
1.2.2.    Ecuaciones en diferencias de segundo orden . . . . .	7
1.3.    Sistemas dinámicos discretos lineales . . . . .	9
1.3.1.    Modelo de Leslie . . . . .	9
1.3.2.    Modelo basado en políticas de recolección y sacrificio . . . . .	10
1.4.    Sistemas dinámicos discretos no lineales . . . . .	11
1.4.1.    Mapa tienda . . . . .	11
1.4.2.    El mapa logístico, diagrama de bifurcación y número de Feigenbaum . . . . .	15
1.4.3.    Mapa de Gauss . . . . .	19
1.4.4.    Mapa de Hénon . . . . .	20
<b>2. Introducción a la teoría del caos</b>	<b>23</b>
2.1.    Definición de caos en sistemas dinámicos . . . . .	23
2.2.    El Mapa Shift y la dinámica simbólica . . . . .	24
2.3.    Teorema de Li-Yorke: periodo tres implica caos . . . . .	26
2.4.    Exponente de Lyapunov . . . . .	30
<b>3. Sistemas dinámicos complejos</b>	<b>33</b>
3.1.    El conjunto de Julia . . . . .	33
3.2.    El conjunto de Mandelbrot . . . . .	39
3.2.1.    Límites de las órbitas periódicas. Bulbos del conjunto de Mandelbrot . . . . .	41
3.3.    Aplicación de la teoría de conjuntos de Julia al método de Newton . . . . .	46

## *Índice general*

<b>4. Fractales</b>	<b>49</b>
4.1. Ejemplos de fractales clásicos . . . . .	50
4.1.1. El conjunto de Cantor . . . . .	50
4.1.2. El triángulo de Sierpinski . . . . .	53
4.1.3. La curva de Koch . . . . .	54
4.1.4. El cuadrado de Koch . . . . .	56
4.1.5. Otros fractales . . . . .	58
4.2. Sistema de Lindenmayer (sistema-L) . . . . .	59
4.3. Sistema de funciones iteradas (SFI) . . . . .	67
4.3.1. Algoritmos para la obtención del fractal asociado a un SFI . . . . .	72
<b>5. Dimensión fractal</b>	<b>79</b>
5.1. Dimensión box counting . . . . .	82
<b>6. Visualización de fractales con Ray Tracing</b>	<b>85</b>
6.1. Aplicación del ray tracing a una esfera . . . . .	86
6.1.1. Puntos y vectores de $\mathbb{R}^3$ . . . . .	86
6.1.2. Imágenes en formato PPM . . . . .	87
6.1.3. Rayos . . . . .	88
6.1.4. Intersección de rayos con esferas . . . . .	89
6.1.5. Iluminación y sombreado . . . . .	90
6.1.6. Creación de una escena con múltiples objetos . . . . .	94
6.2. Introducción al ray marching y aplicación a fractales . . . . .	95
6.2.1. Aproximación de la normal de una superficie fractal . . . . .	96
6.2.2. Mandelbulb . . . . .	97
6.2.3. Juliabulb . . . . .	101
6.2.4. Mandelbox . . . . .	103
6.3. Paralelismo basado en procesos y tiempos de renderizado . . . . .	105
<b>7. Conclusiones y trabajo futuro</b>	<b>109</b>
A. Representación tridimensional de las distintas potencias del mapa $f(z) = z^n + c$	111
B. Cuaterniones de Julia	115
C. Estimación del coste y planificación	117
D. Software	121
D.1. Archivos . . . . .	121
D.2. Lenguaje . . . . .	123
D.3. Librerías . . . . .	123
D.4. Entorno de desarrollo . . . . .	124
<b>Bibliografía</b>	<b>125</b>

# Índice de figuras

1.1.	[Python] Función tienda cuando $\mu = \frac{3}{2}$	12
1.2.	[Python] Función tienda cuando $\mu = 2$ y $x_0 = \frac{1}{5}$	13
1.3.	[Python] Función tienda cuando $\mu = \frac{3}{2}$ y $x_0 = \frac{1}{3}$	13
1.4.	[Python] Intersección del mapa tienda $T(x)$ con la diagonal cuando (i) $0 < \mu < 1$ (ii) $\mu = 1$ y (iii) $1 < \mu \leq 2$	14
1.5.	[Python] Intersección de la segunda iterada del mapa tienda $T^2(x)$ con la diagonal para $\mu = 2$ .	15
1.6.	[Python] Representación del mapa logístico para $\mu = 3.5$ .	15
1.7.	[Python] Representación del diagrama de bifurcación o diagrama de Feigenbaum para $x_0 = \frac{1}{2}$ .	18
1.8.	[Python] Representación de los valores obtenidos por las iteradas $f_\mu^n(x)$ a partir de un valor inicial $x_0 = \frac{1}{2}$ (i) $\mu = \frac{1}{2}$ (ii) $\mu = 2$ (iii) $\mu = \frac{16}{5}$ (iv) $\mu = \frac{7}{2}$ (v) $\mu = \frac{37}{10}$ (vi) $\mu = \frac{39}{10}$	18
1.9.	[Python] Mapa de Gauss para $\alpha = 1$ y $\beta = 4$	19
1.10.	[Python] Mapa de Gauss para $\alpha = 4$ y $\beta = 1$	19
1.11.	[Python] Posibles intersecciones de la curva de Gauss con la diagonal.	20
1.12.	[Python] Diagrama de bifurcación del mapa de Gauss con valor inicial $x_0 = 0$ para (i) $\alpha = 4$ (ii) $\alpha = 8$	20
1.13.	[Python] Mapa de Hénon para $\alpha = 1.2, \beta = 0.4$ con valor inicial $(0.1, 0)$	21
2.1.	[Elaoo] Ejemplo de mapa continuo con puntos periódicos de periodo 5 y sin puntos de periodo 3	27
2.2.	[Elaoo] $f(a) = b, f(b) = c$ y $f(c) = a$ entonces $J \subset f(L)$ y $L \subset f(J)$	29
2.3.	[Python] Exponente de Lyapunov y diagrama de bifurcación para $x_0 = 0.6$ con (i) $0 \leq \mu \leq 4$ (ii) $3 \leq \mu \leq 4$	31
3.1.	[Python] Conjunto de Julia relleno $K_0$ asociado al mapa $f_0(z) = z^2$	34
3.2.	[Python] Conjunto de Julia relleno $K_{-0.1+0.8i}$ ( El conejo de Douday) (i) Algoritmo en blanco y negro (ii) Algoritmo de iteración inversa	36
3.3.	[Python] Conjunto de Julia relleno $K_{-0.39-0.58i}$ ( El disco de Siegel) (i) Algoritmo en blanco y negro (ii) Algoritmo de iteración inversa	36
3.4.	[Elaoo] $f_c^{-1}(\Gamma)$ cuando: (i) $c$ es un punto interior de $\Gamma$ (ii) $c$ está en $\Gamma$ (iii) $c$ es un punto exterior de $\Gamma$	37
3.5.	[Python] Conjunto de Julia relleno conexo $K_{0.377-0.248i}$	38
3.6.	[Python] Conjunto de Julia relleno desconexo $K_{-1.2i}$	39
3.7.	[Python] Conjunto de Mandelbrot en blanco y negro	41
3.8.	[Python] Conjunto de Mandelbrot	41
3.9.	[AGo2] Límite de los puntos fijos de M	42
3.10.	[Python] Conjunto de Julia relleno $K_{-0.5+0.5i} \cdot J_{-0.5+0.5i}$ es una curva cerrada y simple	42
3.11.	[AGo2] Límite de los puntos periódicos de periodo dos de M	43

## Índice de figuras

3.12. [Python] Conjunto de Julia relleno $K_{-1.1+0.1i}$ . $J_{-1.1+0.1i}$ ya no es una curva simple. . . . .	44
3.13. [AGo2] Límite de los puntos periódicos de periodo tres y cuatro de M . . . . .	44
3.14. [Python] Conjunto de Julia relleno $K_{-0.2+0.75i}$ , tiene un 3-ciclo atractivo. . . . .	45
3.15. [Elaoo] Mapeo del disco al exterior de $M$ . . . . .	46
3.16. [Python] Conjunto de Julia asociado al método de Newton para la función compleja $p(z) = z^2 - 1$ . . . . .	47
3.17. [Python] Conjunto de Julia asociado al método de Newton para la función compleja $p(z) = z^3 - 1$ . . . . .	48
3.18. [Python] Conjunto de Julia asociado al método de Newton para las funciones complejas: (i) $p(z) = z^4 - 1$ (ii) $p(z) = z^5 - 1$ . . . . .	48
4.1. [Python] Etapas de la construcción del conjunto de Cantor: (i) $C_0$ (ii) $C_1$ (iii) $C_2$ (iv) $C_3$ (v) $C_4$ (vi) $C_5$ (vii) $C_6$ (viii) $C_7$ . . . . .	51
4.2. [Python] Etapas de la construcción del triángulo de Sierpiński: (i) $S_0$ (ii) $S_1$ (iii) $S_2$ (iv) $S_3$ (v) $S_4$ (vi) $S_5$ . . . . .	53
4.3. [Python] Etapas de la construcción de la curva de Koch: (i) $K_0$ (ii) $K_1$ (iii) $K_2$ (iv) $K_3$ (v) $K_4$ (vi) $K_5$ . . . . .	55
4.4. [Python] Copo de nieve de Koch . . . . .	56
4.5. [Python] Etapas de la construcción del cuadrado de Koch: (i) $K'_0$ (ii) $K'_1$ (iii) $K'_2$ (iv) $K'_3$ (v) $K'_4$ (vi) $K'_5$ . . . . .	57
4.6. [Python] Etapas de la construcción de la curva de Lévy: (i) $L_0$ (ii) $L_1$ (iii) $L_2$ (iv) $L_4$ (v) $L_6$ (vi) $L_9$ . . . . .	58
4.7. [Python] Etapas de la construcción de la curva del dragón: (i) $D_0$ (ii) $D_1$ (iii) $D_2$ (iv) $D_6$ (v) $D_9$ (vi) $D_{12}$ . . . . .	59
4.8. [Python] Interpretación gráfica de la palabra $F + Ff - F - Ff + F$ con ángulo $\theta = 90^\circ$ . . . . .	60
4.9. [Python] Interpretación gráfica de los movimientos (i) L (ii) R (iii) S (iv) Z con ángulo $\theta = 90^\circ$ y (v) D (vi) E con ángulo de $\theta = 45^\circ$ . . . . .	61
4.10. [Python] Punta de flecha de Sierpiński . . . . .	63
4.11. [Python] Etapas de la construcción de la curva de Peano: (i) $P_0$ (ii) $P_1$ (iii) $P_2$ . .	64
4.12. [Python] Etapas de la construcción de la curva de Hilbert: (i) $H_0$ (ii) $H_1$ (iii) $H_2$ . .	65
4.13. [Python] Ramificación de la maleza. Axioma: $F$ , reglas de producción: $F \rightarrow F[+F]F[-F]F$ y parámetros: $\theta = 25.7^\circ$ . . . . .	66
4.14. [Python] Ramificación de un arbusto. Axioma: $F$ , reglas de producción: $F \rightarrow FF + [+F - F - F] - [-F + F + F]$ y parámetros: $\theta = 25^\circ$ . . . . .	66
4.15. [Python] Algoritmo aleatorio para la obtención del helecho de Barnsley con (i) 1000 (ii) 10000 (iii) 100000 puntos . . . . .	76
4.16. [Python] Juego del caos para la obtención del triángulo Sierpiński con (i) 1000 (ii) 10000 (iii) 100000 puntos . . . . .	77
5.1. [Python] Conteo de cajas para el triángulo de Sierpiński . . . . .	83
5.2. [Python] Algoritmo box counting para el triángulo de Sierpiński con cajas de dimensiones (i) $2^9$ (ii) $2^8$ (iii) $2^7$ (iv) $2^6$ píxeles. . . . .	84
5.3. [Python] Recta de mejor ajuste de la gráfica logarítmica para el triángulo de Sierpiński . . . . .	84
6.1. [NVI] Conceptos básicos del trazado de rayos . . . . .	85

6.2. [Shiz0] Formato de imagen PPM . . . . .	88
6.3. [Shiz0] Imagen resultante del archivo PPM de la figura 6.2 . . . . .	88
6.4. [Python] Esfera mediante ray tracing básico . . . . .	90
6.5. [Scrb] Vectores que intervienen en la iluminación de una escena . . . . .	90
6.6. [Mathcha] Normal a la esfera en un punto . . . . .	91
6.7. [Scra] Iluminación del área diferencial dA . . . . .	92
6.8. [Scra] Ley del coseno de Lambert . . . . .	93
6.9. [Python] Esfera mediante ray tracing con modelo de reflexión de Phong para iluminación y sombreado. . . . .	94
6.10. [Python] Esferas sobre un tablero de ajedrez. . . . .	95
6.11. [Mathcha] Marcha de rayos . . . . .	96
6.12. [Python] Mandelbulb . . . . .	100
6.13. [Python] Juliabulb (i) $c = (-1, 0.2, 0, 0)$ (ii) $c = (-0.162, 0.163, 0.560, -0.599)$ (iii) $c = (-0.2, 0.8, 0, 0)$ (iv) $c = (-0.125, -0.256, 0.847, 0.0895)$ . . . . .	102
6.14. [Python] Animación del cuaternion de Julia $c = (-0.08, 0.0, -0.8, -0.03)$ . . . . .	103
6.15. [Python] Mandelbox . . . . .	105
6.16. [Mathcha] División de 15 filas de píxeles entre 4 núcleos . . . . .	106
6.17. [Mathcha] Comparativa de tiempos de renderizado . . . . .	106
 A.1. [Python] Representación tridimensional del mapa $f_c(z) = z^n + c$ para (i)(ii) $n = 2$ (iii)(iv) $n = 3$ (v)(vi) $n = 4$ . . . . .	111
A.2. [Python] Representación tridimensional del mapa $f_c(z) = z^n + c$ para (i)(ii) $n = 5$ (iii)(iv) $n = 6$ (v)(vi) $n = 7$ (vii)(viii) $n = 8$ (Mandelbulb) . . . . .	112
A.3. [Python] Representación tridimensional del mapa $f_c(z) = z^n + c$ para (i)(ii) $n = 9$ (iii)(iv) $n = 10$ (v)(vi) $n = 11$ (vii)(viii) $n = 12$ . . . . .	113
 B.1. [Python] Cuaterniones de Julia (i) $c = (-0.291, -0.399, 0.339, 0.437)$ (ii) $c = (-0.2, 0.4, -0.4, -0.4)$ (iii) $c = (-0.213, -0.0410, -0.563, -0.560)$ (iv) $c = (-0.2, 0.6, 0.2, 0.2)$ . . . . .	115
B.2. [Python] Cuaterniones de Julia (i) $c = (-0.445, 0.339, -0.0889, -0.562)$ (ii) $c = (0.185, 0.478, 0.125, -0.392)$ (iii) $c = (-0.450, -0.447, 0.181, 0.306)$ (iv) $c = (-0.218, -0.113, -0.181, -0.496)$ (v) $c = (-0.125, -0.256, 0.847, 0.0895)$ . . . . .	116
 C.1. [Canva] Planificación optimista del desarrollo del trabajo . . . . .	119
C.2. [Canva] Planificación realista del desarrollo del trabajo . . . . .	119



## **Agradecimientos**

Agradezco a mi tutor Pedro José Torres Villarroya por ofrecerme este proyecto y haber estado ayudándome durante el desarrollo del mismo con sus consejos y explicaciones.

En especial, a mis padres por su apoyo no solo durante la elaboración de este trabajo sino en toda esta etapa académica y a todas las personas que me han apoyado a lo largo de los años.

Por supuesto, agradecer a todos los compañeros de clase que me han acompañado en esta etapa, gracias por ayudarme, inspirarme y hacerme este camino más ameno.



## Resumen

En este trabajo se va a realizar un estudio teórico de los sistemas dinámicos discretos. Se definirán y se tratarán aspectos relacionados con la estabilidad, periodicidad, puntos fijos, atracción, repulsión, diagramas de bifurcación, ... Luego, se introducirá la noción de caos partiendo de la definición formal y terminando con el exponente de Lyapunov.

Una vez finalizado el estudio para el caso real, se analizará la dinámica en el caso complejo. Representaremos y estudiaremos los conjuntos de Julia y de Mandelbrot y distintos fractales de Newton. A continuación, explicaremos cómo construir y visualizar fractales de dos dimensiones desde tres perspectivas: construcción clásica (recursividad), sistemas de Lindenmayer y sistemas de funciones iteradas. Terminaremos el estudio con la dimensión de Hausdorff e implementaremos el algoritmo box counting.

Finalmente, se indagará sobre el ray tracing (trazado de rayos) para la visualización de fractales tridimensionales. Comenzaremos con un objeto geométrico sencillo como es la esfera y acabaremos con el renderizado del Mandelbulb, Mandelbox y cuaterniones de Julia.

**PALABRAS CLAVE:** sistema dinámico, caos, conjunto de Julia, conjunto de Mandelbrot, fractal, dimensión de Hausdorff, box counting, ray tracing, Mandelbulb.



## Summary

The dynamical systems theory is a branch of mathematics that deals with the study of the evolution over time of systems that model and simulate nature behaviors. It provides a common language in different scientific fields: biology, ecology, physics, ... We will see which elements compose them and we will give some key notions for its analysis related to periodicity, stability, attractive or repulsive behaviors, ... In addition, linear and non-linear systems, bifurcation diagrams and chaos theory will be present.

Chaos is the term to describe difficult behavior of certain systems. The property of chaos in dynamical systems informs us of certain behaviors to take into consideration in different systems. We will use the formal definition, the Li-Yorke theorem and the calculation of the Lyapunov exponent to detect this quality.

In nature, frequently appears forms that we can't describe. If we look around us, we perceive objects that repeat a pattern indefinitely. One of the purposes of this work is to illustrate the world behind these patterns. Therefore, we will introduce into complex dynamics to examine and represent the Julia and Mandelbrot sets. These sets motivated the appearance of fractal geometry leading to the creation of a new field in mathematics. The precursor of this discipline was Benoît B. Mandelbrot who, in 1975 published his first book on this topic. The fractals generated a lot of interest in other mathematicians such as Georg Cantor, Waclaw Sierpiński, Niels Fabian Helge von Koch, Paul Pierre Lévy, Giuseppe Peano, David Hilbert, ... In this work, we will fully understand the concept of fractal (definition, main characteristics, the most important fractals, dimension), and the mechanisms to generate and visualize them in two and three dimensions. We will use three approaches: classical construction, Lindenmayer systems and SFI to generate them .

The word fractal comes from the latin word 'fractus' which means fractured. This term is very appropriate since these objects do not exist in euclidean dimensions, but they do in fractional dimensions. We will establish a box counting algorithm to calculate the value of this dimension in two-dimensional fractals.

Fractal geometry is closely related to computer graphics. Since its inception, computing mechanisms have been used to generate realistic and high-quality fractal images. Over the years, new technologies and algorithms have emerged and for this reason great improvements have been made in their visualization. The challenge of this work is to apply ray tracing techniques for the visualization of three dimensional fractals and turtle graphics for 2D fractals.

The motivation of this work lies in the variety of applications of dynamical systems and fractals also the interest that is being had on them in recent years. Dynamical systems have a great relevance in all areas of science since they model the behavior of certain phenomenon. They are used to describe the population of certain species, the behavior of the market to

## *Summary*

variations in supply and demand, ...

Beyond discovering a new geometric dimension, Benoît B. Mandelbrot knew that fractals were very useful for many fields of science. If we find a structure in nature to which we can apply the principles of fractal geometry, predictions can be made about how such a structure will behave in the future thanks to the self-similarity property. There are fractal applications from abstract art to health.

- Health: for centuries it has been thought that the human heart beats regularly, it has finally been shown that there is a certain fractal pattern for the beating of a healthy heart. Alterations on the fractal scale may be the basis of some heart disease. Fractals explain how the lungs (a organ with a shape that responds to these same patterns) are ventilated in a homogeneous way. In cancer treatments, fractal geometry is useful to reveal the architecture of tumors and their growth mechanisms. Is the fractal geometry a principle of biological design?
- Ecology, nature and environment: in the environment, fractal patterns are found in natural phenomenon such as earthquakes, tsunamis, flight of birds, ... In ecology they are useful to quantify the amount of  $CO_2$  that forests can process or study how a fire spreads in a forest fire.
- Physical phenomenon: Mandelbrot studied the relief of mountains, the irregular coasts, the distribution of the land, ... reaching the conclusion that the land surface responds to fractal patterns.
- Art: if we like to explain in an artistic point of view, a fractal is the visual expression of a mathematical expression. It leads us to the creation of new worlds.
- Architecture: fractal geometry is very present in architecture. Provides visual beauty, structural ability to support certain weight and optimization of space and materials.

This work is divided into chapters that will detail different issues related to these dynamical systems. At the beginning of each chapter the objectives are specified, preceding the topics to be covered. The chapter 1 begins with the definition of a discrete dynamical system and it presents very useful examples and basic notions about fixed points, periodic points, stability, ... The chapter 2 is an introduction to chaos theory and Lyapunov exponent. In chapter 3 we present the Julia set and the Mandelbrot set motivating the appearance of fractal geometry. Chapter 4 discusses the construction of fractals from different approaches. In chapter 5, we discuss the concept of Hausdorff dimension, its value is calculated with the box counting algorithm. Chapter 6 is for the visualization of three dimensional fractals with ray tracing techniques. Finally, chapter 7 deals with conclusions and future work.

**KEYWORDS:** dynamical systems, chaos, Julia set, Mandelbrot set, fractal, Hausdorff dimension, box counting, ray tracing, Mandelbulb.

# Introducción

La teoría de sistemas dinámicos es una rama de las matemáticas que se ocupa del estudio de la evolución en el tiempo de sistemas que modelizan y simulan comportamientos de la naturaleza. Proporciona un lenguaje común en diferentes ámbitos científicos: biología, ecología, física, ... Veremos que elementos lo componen y daremos algunas nociones claves para su análisis relacionadas con periodicidad, estabilidad, comportamientos atractivos o repulsivos, ... Además, se presentarán distintos sistemas tanto lineales como no lineales introduciendo de esta manera los diagramas de bifurcación y la teoría del caos.

El caos es el término usado para describir el difícil comportamiento de ciertos sistemas. La propiedad de caos en sistemas dinámicos nos informa de ciertas conductas a tener en cuenta en la modelización de sistemas. Utilizaremos la definición formal, el teorema de **Li y Yorke** y el cálculo computacional del exponente de **Lyapunov** para detectar esta cualidad.

En la naturaleza con frecuencia aparecen formas que somos incapaces de describir. Si observamos a nuestro alrededor, percibimos objetos que repiten un patrón indefinidamente. Una de las finalidades de este trabajo es la de ilustrar y comprender el mundo que hay detrás de estos patrones. Por ello, nos sumergiremos en la dinámica compleja para examinar y representar los conjuntos de Julia y de Mandelbrot. Estos conjuntos motivaron la aparición de la geometría fractal dando lugar a la creación de un nuevo campo en las matemáticas. El precursor de esta disciplina fue **Benoît B. Mandelbrot** quien, en 1975 publicó su primer libro al respecto. Los fractales generaron mucho interés en otros matemáticos como **Georg Cantor, Waclaw Sierpiński, Niels Fabian Helge von Koch, Paul Pierre Lévy, Giuseppe Peano, David Hilbert**, ... En este trabajo, se pretende comprender de forma completa el concepto de fractal (su definición, principales características, los fractales elementales más importantes, su dimensión), y los mecanismos para generarlos y visualizarlos tanto en dos como en tres dimensiones. Para generarlos utilizaremos tres enfoques: construcción clásica, sistemas de Lindenmayer y SFI.

La palabra fractal proviene del latín 'fractus' que significa fracturado. Este nombre es muy apropiado ya que estos objetos no existen en las dimensiones euclídeas, sino en las dimensiones fraccionarias. Estableceremos un algoritmo de conteo de cajas para calcular el valor de esta dimensión en fractales de dos dimensiones.

La geometría fractal está estrechamente relacionada con la informática gráfica. Desde su nacimiento, se ha tratado de aprovechar los mecanismos de cómputo para generar imágenes fractales cada vez más realistas y con alta calidad. Conforme han ido pasando los años, nuevas tecnologías y algoritmos han surgido y por eso se han logrado grandes mejoras en la visualización de estos. El gran reto de este trabajo es el de aplicar técnicas de **ray tracing** para la visualización de fractales tridimensionales y gráficos **turtle** para fractales en 2D.

## 0.1. Motivación

La motivación de este trabajo reside en la multitud de aplicaciones que tienen los sistemas dinámicos y los fractales y en el interés que se está teniendo sobre ellos estos últimos años. Los sistemas dinámicos tienen una gran relevancia en todos los ámbitos de la ciencia ya que modelan, explican y ayudan a comprender el comportamiento de ciertos fenómenos. Su estudio ha servido para describir la población de determinadas especies, el comportamiento del mercado sujeto a las variaciones de la oferta y la demanda, ... Cualquier fenómeno a estudiar se puede tratar de simular con un sistema dinámico.

Más allá de descubrir una nueva dimensión geométrica de gran belleza, Benoît B. Mandelbrot sabía que los fractales serían de gran utilidad para muchos campos de la ciencia. Si se identifica una estructura en la naturaleza a la que se le pueden aplicar los principios de la geometría fractal, se pueden hacer predicciones sobre cómo se comportará dicha estructura en el futuro gracias a la propiedad de autosimilitud. Las aplicaciones de los fractales van desde el arte abstracto hasta la salud del cuerpo humano.

- Salud: durante siglos se había pensado que el corazón humano late de forma regular y lineal, finalmente se ha demostrado que existe un patrón fractal determinado para los latidos de un corazón sano. Las alteraciones en la escala fractal pueden ser la base de alguna enfermedad cardiaca. Los fractales explican cómo los pulmones (un órgano con una forma que responde a estos mismos patrones) se ventilan de manera homogénea. En los tratamientos del cáncer, la geometría fractal es útil para desvelar la arquitectura de los tumores y sus mecanismos de crecimiento. ¿ Es la geometría fractal un principio de diseño biológico?
- Ecología, naturaleza y medioambiente: dentro del medio ambiente se encuentran patrones fractales en fenómenos naturales como terremotos, tsunamis, vuelo de las bandadas de pájaros,... En ecología sirven para cuantificar la cantidad de  $CO_2$  que los bosques pueden llegar a procesar o estudiar cómo se extiende un fuego en un incendio forestal.
- Fenómenos físicos: el propio Mandelbrot estudió el relieve de montañas, la irregularidad de las costas , distribución del terreno, ... llegando a la conclusión de que la superficie terrestre responde a patrones fractales.
- Arte: desde un punto de vista artístico, un fractal es la expresión visual de una expresión matemática. Nos conduce a la creación de nuevos mundos.
- Arquitectura: la geometría fractal esta muy presente en la arquitectura no solo por su belleza visual sino por su capacidad estructural para soportar determinado peso u optimización del espacio y materiales.

## 0.2. Objetivos

Los objetivos inicialmente propuestos se han cumplido durante la elaboración de este trabajo.

- Definir, explicar y exponer nociones básicas de los sistemas dinámicos discretos: iteradas, puntos fijos, puntos periódicos, estabilidad, ...

- Ejemplificar la teoría de sistemas con ejemplos lineales y no lineales mediante el uso de herramientas informáticas para la representación de mapas, iteradas, órbitas, puntos fijos y periódicos.
- Introducir la teoría del caos, representar diagramas de bifurcación y calcular el exponente de Lyapunov.
- Explicar y representar los conjuntos de Julia y de Mandelbrot y fractales de Newton.
- Generar y visualizar fractales de dos dimensiones mediante su construcción clásica y sistemas de Lindenmayer.
- Formalizar la construcción de fractales mediante SFI.
- Definir y explicar la dimensión de Hausdorff. Implementar el algoritmo box counting para el cálculo de esta dimensión.
- Visualizar fractales tridimensionales (Mandelbulb, Mandelbox y cuaterniones de Julia) mediante ray tracing.

### 0.3. Estructura

En el desarrollo del presente trabajo se pretende explicar diversos aspectos de la dinámica tanto real como compleja: fractales, caos, estabilidad, diagramas de bifurcación, ... Este documento está dividido en capítulos que detallarán distintos asuntos relacionados con estos sistemas. Al inicio de cada capítulo se precisan los objetivos del mismo, precediendo los temas que se van a tratar.

El capítulo 1 comienza definiendo lo que es un sistema dinámico discreto y expone resultados y ejemplos de gran utilidad. El capítulo 2 es una introducción a la teoría del caos y al exponente de Lyapunov. Se demostrará de manera formal y computacional como determinados sistemas son caóticos. En el capítulo 3 se presentan los conjuntos de Julia y de Mandelbrot motivando la aparición de la geometría fractal. En el capítulo 4 se expone la construcción de fractales desde distintos enfoques. En el capítulo 5 se trata el concepto de dimensión de **Hausdorff** y se calcula su valor con el algoritmo de conteo de cajas. En el capítulo 6 se visualizan fractales tridimensionales con ray tracing. Finalmente, en el capítulo 7 se presentan conclusiones y trabajo futuro.



# 1. Sistemas dinámicos discretos

En este capítulo se van a tratar conceptos básicos sobre sistemas dinámicos discretos. Se introducirán los términos de órbita, punto fijo, punto periódico, así como los conceptos de estabilidad, atracción y repulsión. Se establecerá una relación de recurrencia y se exemplificará la teoría con modelos tanto lineales como no lineales.

Las definiciones y conceptos de interés se encuentran recogidos en [MRKo2], [Lyn18] [Dev89], [Goo13] y [AGo2].

## 1.1. Primeras definiciones

**Definición 1.1.** Sea  $X \subset \mathbb{R}^n$ ,  $X = (X, d)$  un espacio métrico y  $f: X \rightarrow X$  una función continua. Un **sistema dinámico discreto** es un par de la forma  $(X, f)$  donde  $f$  es el **mapa** asociado al sistema.

Dado que el dominio y codominio de  $f$  son el mismo espacio podemos componer  $f$  consigo misma dando lugar a nuevas funciones:  $f^0$  será la función identidad  $id: X \rightarrow X$ ,  $f^1 = f$ ,  $f^2 = f \circ f$ , ...,  $f^n = f \circ f \circ f \circ \dots \circ f$ . Llamaremos a estas funciones las **iteradas de  $f$** . Los sistemas dinámicos estudian el comportamiento de las iteradas de  $f$  a partir de un **valor inicial**  $x_0$ .

**Definición 1.2.** Dado un punto  $x \in X$  denominaremos  $O^+(x)$  a la **órbita** de  $x$  bajo  $f$  donde

$$O^+(x) = \{x, f(x), f^2(x), f^3(x), \dots\}$$

El carácter dinámico de  $f$  se manifiesta cuando consideramos cada órbita  $O^+(x)$  como las distintas posiciones que va tomando un objeto a lo largo del tiempo.

Tiempo	0	1	2	3	...	n
Posición	$x$	$f(x)$	$f^2(x)$	$f^3(x)$	...	$f^n(x)$

Cada punto del dominio da lugar a una órbita, es decir, una secuencia de movimientos. Desde este punto de vista, la función  $f$  genera un sistema dinámico discreto. Por lo tanto, si queremos estudiar las propiedades dinámicas de  $f$ , en realidad, nos interesa conocer como se comportan las órbitas y los puntos que producen.

### 1.1.1. Puntos fijos

**Definición 1.3.** Sea  $f: I \subset \mathbb{R} \rightarrow I$  continua. Se dice que  $x^* \in I$  es un **punto fijo** o **punto de equilibrio** del sistema si  $f(x^*) = x^*$ . Se dice que  $x^* \in I$  es **eventualmente fijo** si  $f^{n+1}(x^*) = f^n(x^*)$  para cierto  $n \in \mathbb{N}$ .

## 1. Sistemas dinámicos discretos

**Teorema 1.1.** Sea  $f: I \subset \mathbb{R} \rightarrow I$  continua. Sea  $x_0 \in I$  un punto cuya órbita converge a un punto  $x^*$ . Entonces  $x^*$  es un punto fijo de  $f$ .

*Demostración.* Como  $\lim_{n \rightarrow \infty} f^n(x_0) = x^* \implies \lim_{n \rightarrow \infty} f^{n+1}(x_0) = \lim_{n \rightarrow \infty} f(f^n(x_0)) = f(x^*)$  (por continuidad de  $f$ ). Como  $\{f^{n+1}(x_0)\}$  es la misma sucesión que  $\{f^n(x_0)\}$  (adelantada), ambas han de tener el mismo límite. Luego  $f(x^*) = x^*$ .  $\square$

**Teorema 1.2.** Sea  $f: I = [a, b] \rightarrow \mathbb{R}$  continua. Si  $I \subset f(I)$  entonces  $f$  tiene al menos un punto fijo en  $I$ .

*Demostración.* Sea  $I = [a, b]$  un intervalo compacto. Como  $f$  es continua, el teorema de Weierstrass asegura que  $f(I)$  es un intervalo cerrado y acotado. Denotamos  $f(I) = [\alpha, \beta]$ . Como  $f$  alcanza su valor máximo y mínimo en  $I$ , entonces existe al menos dos puntos  $a', b' \in I$  tal que:

$$f(a') = \min_{[a,b]} f = \alpha \leq a \leq a' \leq b \leq \beta$$

$$f(b') = \max_{[a,b]} f = \beta \geq b \geq b' \geq a \geq \alpha$$

Tenemos dos situaciones:

$$\alpha = f(a') = a' \quad o \quad f(a') < a'$$

$$\beta = f(b') = b' \quad o \quad f(b') > b'$$

En el caso de que  $f(a') = a'$  o  $f(b') = b'$ , tomamos  $\gamma = a'$  o  $\gamma = b'$  como punto fijo y el teorema quedaría demostrado.

Si consideramos la segunda posibilidad, en la que  $f(a') < a'$  y  $f(b') > b'$ , definimos una función continua  $g$  como  $g(x) = f(x) - x$  tal que:

$$g(a') = f(a') - a' < 0$$

$$g(b') = f(b') - b' > 0$$

Por tanto,  $g$  cambia su signo en el intervalo con extremos  $a', b'$ . Por el teorema de Bolzano, existe  $\gamma \in (a, b')$ , donde  $g(\gamma) = 0$ . Como este intervalo está incluido en  $[a, b]$ , concluimos que existe  $\gamma \in [a, b]$  tal que  $g(\gamma) = 0$ , lo cual implica que  $f(\gamma) = \gamma$ .  $\square$

### 1.1.1.1. Estabilidad, atracción, repulsión e hiperbolicidad

**Definición 1.4.** Sea  $f: I \subset \mathbb{R} \rightarrow I$  continua y  $x^*$  un punto fijo de  $f$ . Se dice que  $x^*$  es un punto fijo **estable** si para todo  $\epsilon > 0$  existe  $\delta > 0$  tal que si  $x \in X$  y  $|x - x^*| < \delta$ , entonces  $|f^n(x) - x^*| < \epsilon$  para todo  $n \geq 0$ . En caso contrario, el punto fijo será **inestable**.

**Definición 1.5.** Sea  $f: I \subset \mathbb{R} \rightarrow I$  continua. Se dice que un punto fijo  $x^*$  es **atractivo** si existe un entorno  $U$  del punto tal que  $\lim_{n \rightarrow \infty} f^n(x) = x^*$  para todo  $x \in U \cap I$ . Al conjunto  $\{x \in I \mid \lim_{n \rightarrow \infty} f^n(x) = x^*\}$  se le llama **cuenca de atracción** de  $x^*$ .

**Definición 1.6.** Se dice que un punto fijo  $x^*$  es **asintóticamente estable** si es a la vez estable y atractor.

**Definición 1.7.** Sea  $f: I \subset \mathbb{R} \rightarrow I$  continua. Se dice que un punto fijo  $x^*$  es **repulsivo** si existe un entorno  $U$  del punto tal que para todo  $x \in U \cap I \setminus \{x^*\}$  existe  $k \in \mathbb{N}$  tal que  $f^k(x) \notin U$ .

**Definición 1.8.** Sea  $f: I \subset \mathbb{R} \rightarrow \mathbb{R}$ ,  $f \in C^1(I)$  y  $x^* \in I$  un punto fijo. Si  $|f'(x^*)| \neq 1$  entonces  $x^*$  es un punto **hiperbólico**. En caso contrario, si  $|f'(x^*)| = 1$  entonces  $x^*$  es un punto **no hiperbólico o indiferente**.

**Teorema 1.3.** Sea  $f: I \subset \mathbb{R} \rightarrow \mathbb{R}$ ,  $f \in C^1(I)$  y  $x^* \in I$  un punto fijo hiperbólico.

- i. Si  $|f'(x^*)| < 1$  entonces  $x^*$  es un punto **atractivo** (asintóticamente estable). En concreto, si  $|f'(x^*)| = 0$  entonces a  $x^*$  se le denomina **superatractivo**.
- ii. Si  $|f'(x^*)| > 1$  entonces  $x^*$  es un punto **repulsivo** (inestable).

*Demostración.* i. Sea  $\lambda > 0$  tal que  $|f'(x^*)| < \lambda < 1$ . Entonces existe un entorno  $\epsilon > 0$  tal que  $(x^* - \epsilon, x^* + \epsilon) \subset I$  y  $|f'(x)| < \lambda$  para todo  $x \in (x^* - \epsilon, x^* + \epsilon)$ . Usamos el teorema del valor medio, para todo  $x \in (x^* - \epsilon, x^* + \epsilon)$ ,  $\exists c \in I(x^*, x)$  (intervalo abierto de extremos  $x$  y  $x^*$ ) tal que  $|f(x) - x^*| = |f'(c)||x - x^*| < \lambda|x - x^*|$ . Entonces  $f(x) \in (x^* - \epsilon, x^* + \epsilon)$  y por tanto  $|f^2(x) - x^*| = |f(f(x)) - x^*| < \lambda|f(x) - x^*| < \lambda^2|x - x^*|$ . En general,  $|f^n(x) - x^*| < \lambda^n|x - x^*| \rightarrow 0$ . Concluimos que  $x^*$  es un atractor (asintóticamente estable).

ii. Sea  $\lambda$  tal que  $|f'(x^*)| > \lambda > 1$ . Entonces existe un entorno  $\epsilon > 0$  tal que  $(x^* - \epsilon, x^* + \epsilon) \subset I$  y  $|f'(x)| > \lambda$  para todo  $x \in (x^* - \epsilon, x^* + \epsilon)$ . Usamos el teorema del valor medio, para todo  $x \in (x^* - \epsilon, x^* + \epsilon)$ ,  $\exists c \in I(x^*, x)$  (intervalo abierto de extremos  $x$  y  $x^*$ ) tal que  $|f(x) - x^*| = |f'(c)||x - x^*| > \lambda|x - x^*|$ . Repitiendo el proceso,  $|f^n(x) - x^*| > \lambda^n|x - x^*|$ . Como ahora  $\lambda^n \rightarrow \infty$  cuando  $n \rightarrow \infty$  las iteraciones de puntos cercanos a  $x^*$  tienden a separarse de  $x^*$ . Concluimos que  $x^*$  es un repulsor (inestable).

□

*Observación 1.1.* El valor de  $|f'(x^*)|$  proporciona una aproximación de la velocidad de convergencia de  $f^n(x)$  a  $x^*$ .

**Teorema 1.4.** Sea  $f: I \subset \mathbb{R} \rightarrow \mathbb{R}$ ,  $f \in C^3(I)$  y  $x^* \in I$  un punto fijo no hiperbólico o indiferente con  $f'(x^*) = 1$ .

- i. Si  $f''(x^*) \neq 0$  entonces  $x^*$  es un punto **semiestable**, es decir, presenta estabilidad por la derecha o por la izquierda.
- ii. Si  $f''(x^*) = 0$  y  $f'''(x^*) < 0$  entonces  $x^*$  es un punto fijo atractivo (asintóticamente estable).
- iii. Si  $f''(x^*) = 0$  y  $f'''(x^*) > 0$  entonces  $x^*$  es un punto fijo repulsivo (inestable).

*Demostración.* i. Si  $f'(x^*) = 1$  entonces  $f(x)$  es tangente a la diagonal  $y = x$  en  $x^*$ . Supongamos que  $f''(x^*) > 0$ , entonces  $f(x)$  es convexa en  $x^*$ . Como  $f \in C^3(I)$ , tenemos que  $f''(x) > 0$  para todo  $x \in (x^* - \delta, x^* + \delta)$ ,  $\delta > 0$ . En particular,  $f'(x)$  debe ser creciente en ese intervalo. Como  $f'(x^*) = 1$ :

$$f'(x) < 1 \quad \forall x \in (x^* - \delta, x^*) \quad f'(x) > 1 \quad \forall x \in (x^*, x^* + \delta)$$

## 1. Sistemas dinámicos discretos

Por continuidad, podemos suponer  $f'(x) > 0$  en ese intervalo. Por el teorema del valor medio aplicado a  $[x, x^*] \subset (x^* - \delta, x^*)$ , existe  $q \in (x, x^*)$  que verifica:

$$f'(q) = \frac{f(x^*) - f(x)}{x^* - x}$$

Como  $0 < f'(q) < 1$  y  $x^* > x$  se tiene:

$$0 < \frac{f(x^*) - f(x)}{x^* - x} < 1 \quad x < f(x) < x^*$$

Repetiendo este argumento, podemos ver que la sucesión  $f^n(x)$  es creciente y acotada superiormente por  $x^*$ , de modo que converge a  $x^*$ . Además, no puede existir otro punto fijo en ese intervalo porque sino por el teorema del valor medio tendríamos que  $f'(q^*) = 1$  para algún  $q^* \in (x, x^*)$ , lo cual es una contradicción. Concluimos con que  $x^*$  es estable y atractor por la derecha.

Por otro lado, si  $[x^*, x] \subset [x^*, x^* + \delta]$ . Aplicando de nuevo el teorema del valor medio se obtiene:

$$f'(q) = \frac{f(x) - f(x^*)}{x - x^*} > 1 \quad f(x) > x > x^*$$

Repetiendo el procedimiento, se tiene que los puntos cerca de  $x^*$  se alejan y por tanto  $x^*$  es inestable por la izquierda.

Cuando  $f''(x^*) < 0$  ( $f$  es cóncava en  $x^*$ ) se utiliza un argumento análogo.

- II. Si  $f''(x^*) = 0$ ,  $f'''(x^*) < 0$  y  $f'(x^*) = 1$  entonces  $x^*$  es un punto de inflexión y por el estudio de la tercera derivada se concluye que  $f'(x)$  tiene un máximo local en dicho punto. Por continuidad, se tiene:

$$f'(x) < 1 \quad \forall x \in (x^* - \delta, x^* + \delta), x \neq x^*, \delta > 0$$

Por lo tanto,  $f''(x) > 0$  para  $x \in (x^* - \delta, x^*)$ , de modo que  $f'(x)$  es creciente en dicho intervalo; y  $f''(x) < 0$  para  $x \in (x^*, x^* + \delta)$ , así que  $f'(x)$  es decreciente en ese intervalo. Como  $f'(x) \neq 1$  si  $x \neq x^*$  y por un método análogo al utilizado en (I.) concluimos que  $x^*$  es asintóticamente estable.

- III. Si  $f''(x^*) = 0$ ,  $f'''(x^*) > 0$  y  $f'(x^*) = 1$  entonces  $x^*$  es un punto de inflexión y por el estudio de la tercera derivada se concluye que  $f'(x)$  tiene un mínimo local en dicho punto. Por continuidad, se tiene:

$$f'(x) > 1 \quad \forall x \in (x^* - \delta, x^* + \delta), x \neq x^*, \delta > 0$$

Como  $f'(x) \neq 1$  si  $x \neq x^*$  y por un método análogo al utilizado en (I.), los puntos cercanos a  $x^*$  se alejan de  $x^*$ . Por tanto, concluimos que  $x^*$  es inestable.

□

**Definición 1.9.** Sea  $f: I \subset \mathbb{R} \rightarrow \mathbb{R}$ ,  $f \in C^3(I)$ . La **derivada schwarziana** de  $f$  es la función  $Sf(x)$  definida:

$$Sf(x) = \frac{f'''(x)}{f'(x)} - \frac{3}{2} \left[ \frac{f''(x)}{f'(x)} \right]^2$$

Cuando  $f'(x) = -1$ ,

$$Sf(x) = -f'''(x) - \frac{3}{2} [f''(x)]^2$$

**Teorema 1.5.** *Sea  $f: I \subset \mathbb{R} \rightarrow \mathbb{R}$ ,  $f \in C^3(I)$  y  $x^* \in I$  un punto fijo no hiperbólico o indiferente con  $f'(x^*) = -1$ .*

- i. Si  $Sf(x^*) < 0$  entonces  $x^*$  es un punto fijo atractivo (asintóticamente estable).
- ii. Si  $Sf(x^*) > 0$  entonces  $x^*$  es un punto fijo repulsivo (inestable).

*Demostración.* i. Sea  $g(x) = f^2(x)$ . Entonces,  $g(x^*) = x^*$  es un punto fijo de  $g$ . Si  $x^*$  es asintóticamente estable respecto de  $g$  entonces tambien lo es respecto de  $f$ .

$$g'(x) = \frac{d}{dx}(f(f(x))) = f'(f(x)) \cdot f'(x)$$

de modo que:

$$g'(x^*) = f'(x^*)f'(x^*) = (-1)(-1) = 1$$

Empezamos viendo si la segunda derivada  $g''$  es igual o distinta de 0 en  $x^*$ :

$$g''(x) = f'(f(x))f''(x) + f''(f(x))(f'(x))^2$$

de modo que:

$$g''(x^*) = f'(x^*)f''(x^*) + f''(x^*)(f'(x^*))^2 = -f''(x^*) + f''(x^*) = 0$$

Estudiamos la derivada tercera:

$$g'''(x) = f''(f(x))f'(x)f''(x) + f'(f(x))f'''(x) + f'''(f(x))(f'(x))^3 + f''(f(x))2f'(x)f''(x)$$

Para  $x = x^*$ :

$$g'''(x^*) = (f''(x^*))^2(-1) - 2f'''(x^*) + 2f''(x^*)(-1)f''(x^*) = -2f'''(x^*) - 3(f''(x^*))^2 = 2Sf(x^*)$$

Por hipótesis,  $Sf(x^*) < 0$ , de modo que  $g'''(x^*) = 2Sf(x^*) < 0$  y por el teorema 1.4 concluimos que  $x^*$  es asintóticamente estable respecto de  $f$ .

- ii. Repitiendo el desarrollo de (I.) para  $g(x) = f^2(x)$ , obtenemos que  $g'(x^*) = 1$  y  $g''(x^*) = 0$ . Ahora  $g'''(x^*) = 2Sf(x^*) > 0$  ya que por hipótesis  $Sf(x^*) > 0$ . Concluimos por el teorema 1.4 que  $x^*$  es inestable.

□

**Corolario 1.1.** *Sea  $f: I \subset \mathbb{R} \rightarrow \mathbb{R}$ ,  $f \in C^1(I)$  y  $x^* \in I$  un punto fijo no hiperbólico o indiferente.*

- i. Si  $|f'|$  presenta un máximo local en  $x^*$ , entonces  $x^*$  es un punto fijo atractivo (asintóticamente estable).
- ii. Si  $|f'|$  presenta un mínimo local en  $x^*$ , entonces  $x^*$  es un punto fijo repulsivo (inestable).

### 1.1.2. Puntos Periódicos

**Definición 1.10.** Sea  $f: I \subset \mathbb{R} \rightarrow I$  continua. Se dice que  $p \in I$  es un punto **periódico** si existe  $k \in \mathbb{N}$  tal que  $f^k(p) = p$ . El **periodo** de  $p$  se define como el menor natural que cumple la condición anterior. En este caso, nos referimos al punto  $p$  como **periódico de periodo k o k-ciclo**.

*Observación 1.2.* Si  $p$  es un  $k$ -ciclo, los puntos  $\{p, f(p), f^2(p), \dots, f^{k-1}(p)\}$  son periódicos de periodo  $k$ .

**Definición 1.11.** Se dice que  $p$  es **eventualmente periódico** si no es periódico y existe un  $r \in \mathbb{N}$  tal que  $f^r(p)$  es periódico.

*Observación 1.3.*  $p$  es un punto periódico de periodo  $k \in \mathbb{N}$  si y solo si es un punto fijo de  $f^k$  y no es un punto fijo de  $f^i$  con  $1 \leq i < k$ .

Así pues,  $p$  será un punto periódico atractivo o repulsivo de  $f$  si y solo si es un punto fijo atractivo o repulsivo de  $f^k$ . En este caso,  $\{p, f(p), f^2(p), \dots, f^{k-1}(p)\}$  son puntos fijos atractivos o repulsivos de  $f^k$ .

**Teorema 1.6.** Sea  $f \in C^1(I)$  y  $p$  un punto periódico de periodo  $k \in \mathbb{N}$ .

- i. Si  $|(f^k)'(p)| < 1$  entonces  $p$  es punto periódico atractivo (asintóticamente estable).
- ii. Si  $|(f^k)'(p)| > 1$  entonces  $p$  es punto periódico repulsivo (inestable).

*Demostración.* Basta con aplicar la regla de la cadena, que garantiza que para cualquier  $x \in I$ :

$$|(f^k)'(x)| = |f'(f^{k-1}(x)) \cdot (f^{k-1})'(x)| = |f'(x) \cdot f'(f(x)) \cdots f'(f^{k-1}(x))|$$

y aplicar el teorema 1.3. □

**Corolario 1.2.** Sea  $f \in C^1(I)$  y  $p$  un punto periódico de periodo  $k \in \mathbb{N}$ . Si  $p_i = f^i(p)$  para  $i = 0, 1, 2, \dots, k-1$ , entonces:

- i.  $p$  es punto periódico atractivo (asintóticamente estable) si  $|f'(p_0)f'(p_1)f'(p_2) \cdots f'(p_{k-1})| < 1$ .
- ii.  $p$  es punto periódico repulsivo (inestable) si  $|f'(p_0)f'(p_1)f'(p_2) \cdots f'(p_{k-1})| > 1$ .

**Teorema 1.7.** Sea  $f \in C^1(I)$  y  $p$  un punto periódico indiferente de periodo  $k \in \mathbb{N}$  ( $|(f^k)'(p)| = 1$ ):

- i. Si  $|(f^k)'|$  presenta un máximo local en  $p$ , entonces  $p$  es punto periódico atractivo (asintóticamente estable).
- ii. Si  $|(f^k)'|$  presenta un mínimo local en  $p$ , entonces  $p$  es punto periódico repulsivo (inestable).

## 1.2. Relaciones de recurrencia

En esta sección, se introducirán las ecuaciones en diferencias y su relación con los sistemas dinámicos discretos.

### 1.2.1. Ecuaciones en diferencias de primer orden

Un sistema dinámico discreto es una ecuación en diferencias de la siguiente forma:

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

en la que se establece una relación de recurrencia donde  $x_{n+1}$  depende de  $x_n$ . Dado un valor inicial  $x_0$ , las iteradas de  $f$  nos permiten obtener la siguiente sucesión de puntos  $\{x_i : i = 0, \dots, \infty\} = \{x_0, x_1, x_2, \dots, x_n, x_{n+1}, \dots\}$ . Las ecuaciones de esta forma se denominan **ecuaciones en diferencias de primer orden**, ya que los sufijos difieren en uno.

**Teorema 1.8.** *La solución general de la ecuación en diferencias de primer orden:*

$$x_{n+1} = mx_n + n \quad n = 0, 1, 2, \dots \quad (1.1)$$

$$\text{viene dada por: } x_n = m^n x_0 + \begin{cases} \frac{m^n - 1}{m - 1} c & \text{si } m \neq 1 \\ nc & \text{si } m = 1 \end{cases}$$

*Demostración.* Dada la ecuación en diferencias (1.1) y aplicando la relación de recurrencia obtenemos que:

$$\begin{aligned} x_1 &= mx_0 + c \\ x_2 &= mx_1 + c = m^2 x_0 + mc + c \\ x_3 &= mx_2 + c = m^3 x_0 + m^2 c + mc + c \end{aligned}$$

Por lo tanto:

$$x_n = m^n x_0 + (m^{n-1} + m^{n-2} + \dots + m + 1)c$$

Usando la suma de una serie geométrica, sabemos que  $m^{n-1} + m^{n-2} + \dots + m + 1 = \frac{m^n - 1}{m - 1}$  si  $m \neq 1$ . Si  $m = 1$  entonces la suma de la serie geométrica vale  $n$ .  $\square$

*Observación 1.4.* Si  $|m| < 1$  entonces  $\lim_{n \rightarrow \infty} x_n = \frac{c}{1-m}$ .

### 1.2.2. Ecuaciones en diferencias de segundo orden

Las relaciones de recurrencia que involucran términos cuyos sufijos difieren en dos se conocen como **ecuaciones en diferencias de segundo orden**. La forma general con coeficientes constantes es la siguiente:

$$ax_{n+2} = bx_{n+1} + cx_n \quad n = 0, 1, 2, \dots \quad (1.2)$$

**Teorema 1.9.** *La solución general de la ecuación en diferencias de segundo orden (1.2) viene dada por:*

$$x_n = k_1 \lambda_1^n + k_2 \lambda_2^n$$

## 1. Sistemas dinámicos discretos

donde  $k_1$  y  $k_2$  son constantes y  $\lambda_1 \neq \lambda_2$  son las raíces de la ecuación cuadrática  $a\lambda^2 - b\lambda - c = 0$ .

- Si  $\lambda_1, \lambda_2 \in \mathbb{C}$  la solución general se puede expresar como:

$$x_n = k_1\lambda_1^n + k_2\lambda_2^n = k_1(re^{i\theta})^n + k_2(re^{-i\theta})^n = r^n(A \cos(n\theta) + B \sin(n\theta))$$

con  $A$  y  $B$  constantes. En este caso, la solución oscila y es real.

- Si  $\lambda_1 = \lambda_2$  la solución general es de la forma:

$$x_n = k_3\lambda_1^n + k_4n\lambda_1^n$$

*Demostración.* La solución de (1.1) nos da la clave para empezar. Supongamos que  $x_n = \lambda^n k$  es solución y  $\lambda$  y  $k$  son valores a encontrar. Sustituimos en (1.2):

$$a\lambda^{n+2}k = b\lambda^{n+1}k + c\lambda^n k$$

o lo que es lo mismo:

$$\lambda^n k(a\lambda^2 - b\lambda - c) = 0$$

Suponiendo que  $\lambda^n k \neq 0$ , la ecuación tiene solución si:

$$a\lambda^2 - b\lambda - c = 0$$

A esta ecuación la denominaremos **ecuación característica**. La ecuación en diferencias (1.2) tiene dos soluciones, como la ecuación es lineal, una solución viene dado por:

$$x_n = k_1\lambda_1^n + k_2\lambda_2^n$$

donde  $\lambda_1 \neq \lambda_2$  son las raíces de la ecuación característica.

Si  $\lambda_1 = \lambda_2$  entonces la ecuación característica se puede escribir como:

$$a\lambda^2 - b\lambda - c = a(\lambda - \lambda_1)^2 = a\lambda^2 - 2a\lambda_1\lambda + a\lambda_1^2$$

Por lo tanto, sea  $b = 2a\lambda_1$ ,  $c = -a\lambda_1^2$  y supongamos que otra solución es de la forma  $kn\lambda^n$ . Sustituyendo, la ecuación (1.2) se convierte en:

$$ax_{n+2} - bx_{n+1} - cx_n = a(n+2)k\lambda_1^{n+2} - b(n+1)k\lambda_1^{n+1} - cnk\lambda_1^n$$

De modo que:

$$ax_{n+2} - bx_{n+1} - cx_n = kn\lambda_1^n(a\lambda_1^2 - b\lambda_1 - c) + k\lambda_1(2a\lambda_1 - b) = 0$$

Esto confirma que  $kn\lambda^n$  es solución de la ecuación (1.2).

Dado que el sistema es lineal, la solución general es de la forma:

$$x_n = k_3\lambda_1^n + k_4n\lambda_1^n$$

Las constantes  $k_j$  se pueden determinar si se dan los valores  $x_0$  y  $x_1$  que cumplan la ecuación en diferencias (1.2).  $\square$

### 1.3. Sistemas dinámicos discretos lineales

Los sistemas dinámicos discretos lineales se describen mediante relaciones de recurrencia de la forma:

$$x_{n+k} = f(x_n) \quad k \in \mathbb{N}, \quad n = 0, 1, 2, \dots$$

donde el segundo miembro de la ecuación depende linealmente. En esta sección, se plantearán distintos modelos con el fin de ilustrar la utilidad práctica de los mapas lineales.

#### 1.3.1. Modelo de Leslie

El modelo de Leslie se desarrolló para describir la dinámica de una población en función de la porción femenina de una especie. Este modelo puede aplicarse a poblaciones humanas, animales, insectos, ...

Partimos de una población, en la que dividimos la porción femenina en  $n$  clases de edad; por tanto, si  $N$  es la edad máxima alcanzable por una hembra, cada clase de edad abarcará un periodo de  $\frac{N}{n}$ . Para estudiar el modelo de Leslie es necesario definir:

- $t_k$ : periodo de tiempo  $k$ -ésimo; viene dado por  $t_k = \frac{kN}{n}$ . La población se observa en intervalos de tiempo discretos iguales a la longitud de la clase de edad.
- $x_i^{(k)}$ : número de mujeres en la  $i$ -ésima clase de edad después del  $k$ -ésimo periodo de tiempo.
- $b_i$ : número de descendientes femeninos nacidos durante la  $i$ -ésima clase de edad.
- $c_i$ : proporción de mujeres que continúan sobreviviendo desde la  $i$ -ésima hasta la  $i + 1$  clase de edad.

Para que este modelo sea realista, es necesario que se cumplan las siguientes condiciones:

$$(i) \quad b_i \geq 0 \quad 1 \leq i \leq n$$

$$(ii) \quad 0 < c_i \leq 1 \quad 1 \leq i < n$$

Al menos un  $b_i$  tiene que ser estrictamente positivo con tal de asegurar algún nacimiento. Los  $c_i$  no pueden ser cero, en cuyo caso no habría mujeres en la clase de edad  $i + 1$ .

El número de mujeres en la primera clase de edad después del  $k$ -ésimo periodo de tiempo es igual al número de mujeres nacidas de mujeres de las  $n$  clases de edad entre el tiempo  $t_{k-1}$  y  $t_k$ .

$$x_1^{(k)} = b_1 x_1^{(k-1)} + b_2 x_2^{(k-1)} + \dots + b_n x_n^{(k-1)}$$

El número de mujeres de la clase de edad  $i + 1$  en el momento  $t_k$  es igual al número de mujeres de la clase de edad  $i$ -ésima en el momento  $t_{k-1}$  que continúan sobreviviendo.

$$x_{i+1}^{(k)} = c_i x_i^{(k-1)}$$

## 1. Sistemas dinámicos discretos

Las anteriores ecuaciones se pueden escribir de la siguiente forma:

$$\begin{pmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix} = o \begin{pmatrix} b_1 & b_2 & b_3 & \cdots & b_{n-1} & b_n \\ c_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & c_2 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & c_{n-1} & 0 \end{pmatrix} \begin{pmatrix} x_1^{(k-1)} \\ x_2^{(k-1)} \\ x_3^{(k-1)} \\ \vdots \\ x_n^{(k-1)} \end{pmatrix}$$

Dicho de otro modo:

$$X^{(k)} = LX^{(k-1)} \quad k = 1, 2, \dots \quad X \in \mathbb{R}^n$$

La matriz  $L$  se denomina **matriz de Leslie**.

Supongamos que  $X^{(0)}$  es un vector con el número inicial de mujeres de cada una de las clases de edad, entonces:

$$X^{(k)} = LX^{(k-1)} = L^k X^{(0)} \quad k = 1, 2, \dots$$

### 1.3.2. Modelo basado en políticas de recolección y sacrificio

Este modelo se ocupará de las políticas de recolección y sacrificio de especies con una población muy elevada. Las políticas son empleadas para evitar el agotamiento y la extinción de determinados grupos. Una política de recolección o sacrificio es sostenible si la distribución de la población restante es la misma después de cada periodo de tiempo.

LLamaremos  $X$  al vector que muestra la distribución de la población justo antes de aplicar las políticas de recolección o sacrificio. Supongamos que una porción de mujeres a punto de entrar en la clase de edad  $i + 1$  muere, dando como resultado la siguiente matriz:

$$D = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{pmatrix} \quad 0 \leq d_i \leq 1 \quad 1 \leq i \leq n$$

El número de muertes vendrá dado por  $DLX$  y la distribución de la población restante por  $LX - DLX = (I - D)LX$ . Para que las políticas sean sostenibles tiene que ocurrir que  $(I - D)LX = X$ .

Si el valor propio de  $(I - D)L$  es uno, entonces  $X$  será vector propio y la población se estabilizará.

Calculamos  $M = (I - D)L$ , obteniendo:

$$M = \begin{pmatrix} b_1(1-d_1) & b_2(1-d_1) & \cdots & b_n(1-d_1) \\ c_1(1-d_2) & 0 & \cdots & 0 \\ 0 & c_2(1-d_3) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & c_{n-1}(1-d_n) & 0 \end{pmatrix}$$

La matriz  $M$  es una matriz de Leslie, por ello tendrá un valor propio  $\lambda_1 = 1$ , si y solo si:  $(1-d_1)(b_1 + b_2c_1(1-d_2) + b_3c_1c_2(1-d_2)(1-d_3) + \dots + b_nc_1\dots c_{n-1}(1-d_2)\dots(1-d_n)) = 1$

Un posible vector propio asociado al valor propio  $\lambda_1$  es:

$$X_1 = \begin{pmatrix} 1 \\ (1-d_2)c_1 \\ (1-d_2)(1-d_3)c_1c_2 \\ \vdots \\ (1-d_2)\dots(1-d_n)c_1\dots c_{n-1} \end{pmatrix}$$

La población sostenible será  $CX_1$  con  $C$  constante.

Este modelo presenta distintas versiones del mismo, en las que se pueden establecer políticas de recolección o sacrificio uniformemente sostenibles, descarte sostenible de la clase más joven, políticas de aprovechamiento, ... que se pueden observar con más detalle en [Lyn18].

## 1.4. Sistemas dinámicos discretos no lineales

El objetivo de esta sección es el de explicar que son los sistemas dinámicos discretos no lineales. Para ello, se presentarán mapas iterados unidimensionales y bidimensionales no lineales mediante ejemplos concretos. Por último, se introducirá el término de diagrama de bifurcación, motivando de esta manera, la aparición del concepto de caos que se detallará en el capítulo 2.

### 1.4.1. Mapa tienda

**Definición 1.12.** Sea  $T: [0, 1] \rightarrow [0, 1]$  continua. El mapa tienda viene dado por:

$$T(x) = \begin{cases} \mu x & \text{si } 0 \leq x < \frac{1}{2} \\ \mu(1-x) & \text{si } \frac{1}{2} \leq x \leq 1 \end{cases} \quad 0 \leq \mu \leq 2$$

El sistema dinámico discreto unidimensional no lineal asociado al mapa tienda está definido por la relación de recurrencia:

$$x_{n+1} = T(x_n) \quad x_n \in [0, 1] \tag{1.3}$$

## 1. Sistemas dinámicos discretos

La función tienda se construye a partir de dos rectas, esto hace que su análisis sea más sencillo que otros sistemas dinámicos discretos no lineales. A pesar de esto, para ciertos valores de  $\mu$  el sistema puede mostrar un comportamiento complejo, fenómenos caóticos, sensibilidad a las condiciones iniciales y periodicidad.

**Ejemplo 1.1.** Para  $\mu = \frac{3}{2}$

$$(I) \quad x_0 = \frac{3}{5} : \left\{ \frac{3}{5}, \frac{3}{5}, \frac{3}{5}, \dots \right\}$$

$$(II) \quad x_0 = \frac{6}{13} : \left\{ \frac{6}{13}, \frac{9}{13}, \frac{6}{13}, \frac{9}{13}, \dots \right\}$$

$$(III) \quad x_0 = \frac{1}{3} : \left\{ \frac{1}{3}, \frac{1}{2}, \frac{3}{4}, \frac{3}{8}, \dots \right\}$$

Para un mismo valor de  $\mu$  del ejemplo 1.1, se tienen puntos fijos, periódicos de periodo dos y sin periodo. Esto muestra la gran variedad de dinámicas que adopta (1.3)

La función tienda del ejemplo 1.1 se muestra en la figura 1.1.

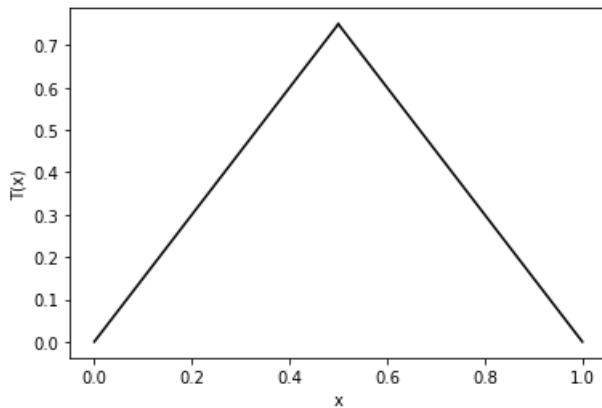


Figura 1.1.: [Python] Función tienda cuando  $\mu = \frac{3}{2}$

*Observación 1.5.* El punto de corte de las dos rectas que forman el mapa tienda se alcanza en  $\frac{\mu}{2}$ , si  $\mu x = \mu(1 - x) \implies x = \frac{1}{2}$  y  $T(\frac{1}{2}) = \frac{\mu}{2}$ .

Las iteraciones numéricas no siempre dan una idea clara de cómo va a ser la secuencia a medida que  $n$  aumenta. Por eso, es conveniente utilizar un **método gráfico** para mostrar las iteraciones.

Partimos del valor inicial  $x_0$ , trazamos la perpendicular al eje de abcisas que pasa por  $x_0$  y tomamos el punto de corte con la función tienda, obteniendo  $T(x_0)$ . Desde este punto, trazamos la perpendicular al eje de ordenadas y tomamos el punto de corte con la diagonal  $y = x$ . La coordenada  $x$  de ese punto corresponde con la iteración  $x_1 = T(x_0)$ . Repetimos el procedimiento tantas veces como iteraciones de la secuencia.

**Proposición 1.1.** *El método gráfico para determinar puntos fijos consiste en identificar las intersecciones de la función  $T(x)$  con la diagonal  $y = x$ .*

**Proposición 1.2.** El tipo de punto fijo se determina a partir del gradiente de la tangente a la función tienda en el punto fijo. Para rectas de ecuación  $y = mx + n$  es fácil ver que:

- (i) Si  $m < -1$  la trayectoria es repelida y se aleja en espiral del punto fijo.
- (ii) Si  $-1 < m < 0$  la trayectoria es atraída y se acerca en espiral al punto fijo.
- (iii) Si  $0 < m < 1$  la trayectoria es atraída y se acerca en forma de escalera al punto fijo.
- (iv) Si  $m > 1$  la trayectoria es repelida y se aleja en forma de escalera del punto fijo.
- (v) Si  $|m| = 1$  el punto fijo no repele ni atrae.
- (vi) Si  $m = 0$  caso trivial.

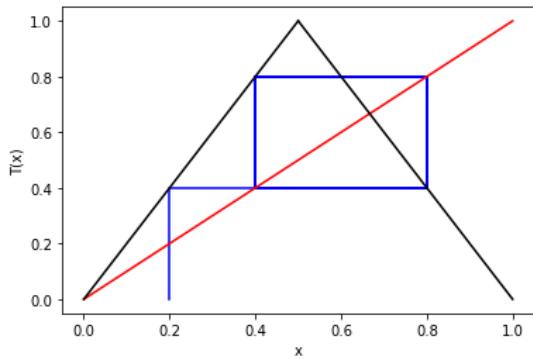


Figura 1.2.: [Python] Función tienda cuando  $\mu = 2$  y  $x_0 = \frac{1}{5}$

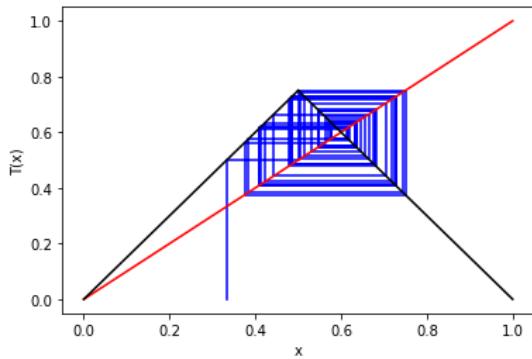


Figura 1.3.: [Python] Función tienda cuando  $\mu = \frac{3}{2}$  y  $x_0 = \frac{1}{3}$

*Observación 1.6.* La proposición 1.2 no es más que un caso específico del teorema 1.3.

**Proposición 1.3.** Es posible determinar los puntos fijos del mapa tienda con la siguiente distinción de casos:

- (i) Si  $0 < \mu < 1$ , el único punto fijo es  $x_{1,1} = 0$  (punto fijo trivial) y es asintóticamente estable.
- (ii) Si  $\mu = 1$ , todo punto  $x \in [0, \frac{1}{2}]$  es fijo.
- (iii) Si  $1 < \mu \leq 2$ , hay dos puntos fijos,  $x_{1,1} = 0$ ,  $x_{1,2} = \frac{\mu}{1+\mu}$ . Ambos son inestables.

**Notación.** El punto periódico dado por  $x_{i,j}$  denota el j-ésimo punto periódico de periodo  $i$ .

*Demostración.* Para hallar los puntos fijos se utilizará el método gráfico (Figura 1.4) y para la estabilidad se aplicará el teorema 1.3.

- (i) Si  $0 < \mu < 1$ , el único punto que pertenece a la intersección de la función tienda con la diagonal es el trivial. Como  $T'(x_{1,1}) < 1 \implies x_{1,1}$  es asintóticamente estable.
- (ii) Si  $\mu = 1$ , la recta  $\mu x$  de  $T(x)$  coincide con la diagonal para todo  $x \in [0, \frac{1}{2}]$ .

### 1. Sistemas dinámicos discretos

(iii) Si  $1 < \mu \leq 2$ ,  $T(x) = x \implies \begin{cases} \mu x = x & \text{si } 0 \leq x \leq \frac{1}{2} \\ \mu(1-x) = x & \text{si } \frac{1}{2} \leq x \leq 1 \end{cases} \implies x = 0, \frac{\mu}{1+\mu}$ . Como  $T'(x_{1,1}) > 1 \implies x_{1,1}$  es inestable y  $T'(x_{1,2}) < -1 \implies x_{1,2}$  es inestable.

□

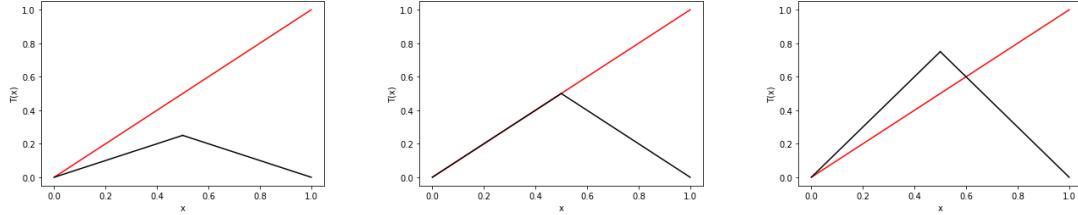


Figura 1.4.: [Python] Intersección del mapa tienda  $T(x)$  con la diagonal cuando (i)  $0 < \mu < 1$   
(ii)  $\mu = 1$  y (iii)  $1 < \mu \leq 2$

**Proposición 1.4.** Se puede extender el método gráfico para determinar los puntos periódicos de periodo  $k \in \mathbb{N}$ . Para ello, es necesario encontrar los puntos de la intersección de  $T^k(x)$  con la diagonal

**Proposición 1.5.** Para determinar los puntos periódicos de periodo dos del mapa tienda es necesario encontrar las intersecciones de la diagonal con:

$$T^2(x) = \begin{cases} \mu^2 x & \text{si } 0 \leq x \leq \frac{1}{2\mu} \\ \mu(1-\mu x) & \text{si } \frac{1}{2\mu} < x \leq \frac{1}{2} \\ \mu(1-\mu + \mu x) & \text{si } \frac{1}{2} < x \leq 1 - \frac{1}{2\mu} \\ \mu^2(1-x) & \text{si } 1 - \frac{1}{2\mu} \leq x \leq 1 \end{cases} \quad 1 \leq \mu \leq 2$$

Si  $1 < \mu \leq 2$ , los puntos periódicos de periodo dos son  $x_{2,1} = \frac{\mu}{1+\mu^2}$ ,  $x_{2,2} = \frac{\mu^2}{1+\mu^2}$ . Si  $0 \leq \mu \leq 1$  no hay puntos de periodo dos.

**Ejemplo 1.2.** Para el caso en el que  $\mu = 2$  se tiene que:

$$T^2(x) = \begin{cases} 4x & \text{si } 0 \leq x \leq \frac{1}{4} \\ 2 - 4x & \text{si } \frac{1}{4} < x \leq \frac{1}{2} \\ 4x - 2 & \text{si } \frac{1}{2} < x \leq \frac{3}{4} \\ 4(1-x) & \text{si } \frac{3}{4} \leq x \leq 1 \end{cases}$$

Los puntos fijos son:  $x_{1,1} = 0$ ,  $x_{1,2} = \frac{2}{3}$  y los puntos de periodo dos:  $x_{2,1} = \frac{2}{5}$ ,  $x_{2,2} = \frac{4}{5}$ . Además, como  $|T'(x)| = 2 > 1$  y  $|(T^2)'(x)| = 4 > 1$ , el teorema 1.3 nos asegura que  $x_{1,1}$ ,  $x_{1,2}$ ,  $x_{2,1}$  y  $x_{2,2}$  son inestables.

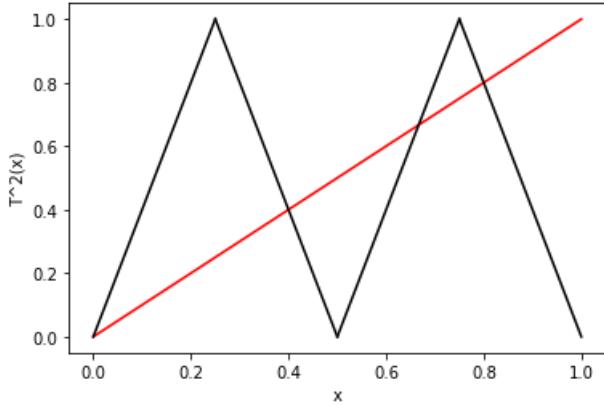


Figura 1.5.: [Python] Intersección de la segunda iterada del mapa tienda  $T^2(x)$  con la diagonal para  $\mu = 2$ .

### 1.4.2. El mapa logístico, diagrama de bifurcación y número de Feigenbaum

El nacimiento de la función logística se remonta a principios de la década de 1970. **R.M. May** investigó las ecuaciones utilizadas por biólogos y entomólogos para modelar las fluctuaciones poblacionales de diversos grupos.

El diagrama de bifurcación contiene toda la información resumida del sistema dinámico, mostrando la periodicidad y los fenómenos caóticos del mismo. **Feigenbaum** estudió el comportamiento de las bifurcaciones del diagrama llegando a descubrir una constante muy relevante en este ámbito.

**Definición 1.13.** Sea  $f_\mu: [0, 1] \rightarrow [0, 1]$  continua. El mapa logístico viene dado por  $f_\mu(x) = \mu x(1 - x)$ ,  $\mu \in [0, 4]$ . El sistema dinámico discreto no lineal asociado al mapa logístico está definido por la relación de recurrencia:

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

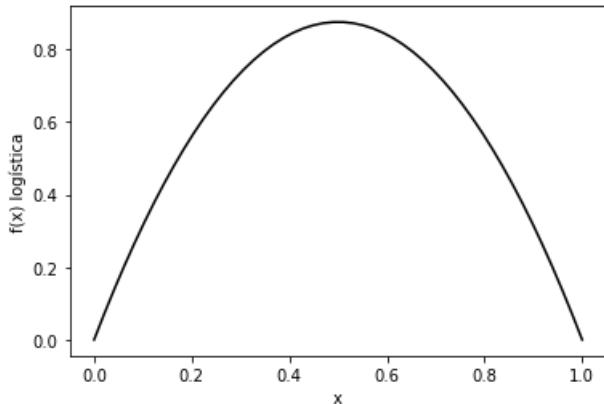


Figura 1.6.: [Python] Representación del mapa logístico para  $\mu = 3.5$ .

### 1. Sistemas dinámicos discretos

*Observación 1.7.* La función logística tiene un máximo que se alcanza en  $\frac{\mu}{4}$ . Como la función es cóncava y  $f'_\mu(x) = \mu(1 - 2x) = 0 \implies x = \frac{1}{2}$ . En  $x = \frac{1}{2}$  se alcanza máximo y  $f_\mu(\frac{1}{2}) = \frac{\mu}{4}$ .

**Proposición 1.6.** Los puntos fijos del mapa logístico son  $x_{1,1} = 0$  y  $x_{1,2} = 1 - \frac{1}{\mu}$ . Por otro lado:

- (i)  $x_{1,1}$  es asintóticamente estable para  $0 < \mu < 1$  e inestable para  $\mu > 1$ .
- (ii)  $x_{1,2}$  es asintóticamente estable para  $1 < \mu < 3$  e inestable para  $\mu < 1$  o  $\mu > 3$ .

*Demostración.* Los puntos fijos del mapa logístico son aquellos que cumplen:  $f_\mu(x) = \mu x(1 - x) = x \implies x_{1,1} = 0, x_{1,2} = 1 - \frac{1}{\mu}$ . Para comprobar la estabilidad se utilizará el teorema 1.3:

- (i) Como  $|f'_\mu(x_{1,1})| = \mu$ ,  $x_{1,1}$  es asintóticamente estable para  $0 < \mu < 1$  e inestable para  $\mu > 1$ .
- (ii) Como  $|f'_\mu(x_{1,2})| = 2 - \mu$ ,  $x_{1,2}$  es asintóticamente estable para  $1 < \mu < 3$  e inestable para  $\mu < 1$  o  $\mu > 3$ .

□

**Proposición 1.7.** Los puntos de periodo dos del mapa logístico son:

$$x_{2,1} = \frac{\mu + 1 + \sqrt{(\mu - 3)(\mu + 1)}}{2\mu} \quad x_{2,2} = \frac{\mu + 1 - \sqrt{(\mu - 3)(\mu + 1)}}{2\mu}$$

Si  $\mu > 3$  el mapa logístico tiene dos puntos periódicos de periodo dos.

*Demostración.* Los puntos de periodo dos del mapa logístico son aquellos que cumplen:

$$f_\mu^2(x) = \mu(\mu x(1 - x))(1 - \mu x(1 - x)) = x \implies x(x - (1 - \frac{1}{\mu}))(-\mu^3 x^2 + (\mu^2 + \mu^3)x - (\mu^2 + \mu)) = 0$$

La ecuación  $-\mu^3 x^2 + (\mu^2 + \mu^3)x - (\mu^2 + \mu) = 0$  tiene dos raíces:

$$x_{2,1} = \frac{\mu + 1 + \sqrt{(\mu - 3)(\mu + 1)}}{2\mu} \quad x_{2,2} = \frac{\mu + 1 - \sqrt{(\mu - 3)(\mu + 1)}}{2\mu}$$

□

**Proposición 1.8.** Los puntos periódicos de periodo dos son asintóticamente estables cuando  $3 < \mu \leq 1 + \sqrt{6}$  y pierden su estabilidad cuando  $\mu > 1 + \sqrt{6}$ .

*Demostración.* Sea  $(f_\mu^2)'(x) = -4\mu^3 x^3 + 6\mu^3 x^2 - 2(\mu^2 + \mu^3)x + \mu^2$ . El teorema 1.6 afirma que los puntos  $x_{2,1}$  y  $x_{2,2}$  son estables cuando:  $|(f_\mu^2)'(x_{2,1})| < 1$  y  $|(f_\mu^2)'(x_{2,2})| < 1$ . Usando el corolario 1.2 se tiene que:

$$\begin{aligned} |(f_\mu^2)'(x_{2,1})| &= |f'_\mu(x_{2,1})f'_\mu(x_{2,2})| < 1 \\ -1 &< \mu^2(1 - 2x_{2,1})(1 - 2x_{2,2}) < 1 \\ -1 &< (-1 - \sqrt{\mu^2 - 2\mu - 3})(-1 + \sqrt{\mu^2 - 2\mu - 3}) < 1 \end{aligned}$$

Se obtienen las siguientes inecuaciones:

$$\mu^2 - 2\mu - 3 > 0 \quad \mu^2 - 2\mu - 5 < 0$$

y resolviendo, se tiene que:  $3 < \mu < 1 + \sqrt{6}$ .

Además cuando  $\mu = 1 + \sqrt{6}$  se tiene:

$$f'_\mu(x_{2,1})f'_\mu(x_{2,2}) = -1$$

de modo que es un ciclo no hiperbólico. Tenemos que la derivada schwarziana :

$$Sf_\mu^2(x_{2,1}) < 0$$

así que por el teorema 1.5 el punto tambien es asintóticamente estable.

Para terminar, es obvio que si  $\mu > 1 + \sqrt{6}$  se tiene que  $|f_\mu^2'(x_{2,1})| > 1$  y  $|f_\mu^2'(x_{2,2})| > 1$ , por tanto son inestables.  $\square$

El comportamiento del mapa logístico, se puede apreciar mediante un diagrama de bifurcación o diagrama de Feigenbaum (Figura 1.7). El diagrama representa los valores de  $\mu \in [0, 4]$  en el eje de abcisas y  $f_\mu^n(x)$  en el eje de ordenadas para un número de iteraciones  $n$  suficientemente grande. Dado un valor inicial  $x_0$ , se calculan las iteraciones de  $f$  a partir de este valor y se representan en el diagrama.

- Cuando  $0 \leq \mu \leq 1$ , el único punto fijo asintóticamente estable es el 0 (proposición 1.6). Sea  $b_0 = 1$ .
- Cuando  $1 < \mu \leq 3$ , las iteraciones se acercan al punto fijo  $1 - \frac{1}{\mu}$  (proposición 1.6). Como consecuencia, la primera bifurcación del diagrama será  $b_1 = 3$ .
- Cuando  $3 < \mu \leq 1 + \sqrt{6}$ , el punto fijo se vuelve inestable y aparecen dos puntos periódicos de periodo dos (proposición 1.7 y 1.8). Por ello, la segunda bifurcación será  $b_2 = 1 + \sqrt{6}$ .
- Cuando  $\mu > 1 + \sqrt{6}$  hasta llegar al valor 3.57 (aproximadamente), el 2-ciclo deja de atraer y aparece un 4-ciclo atractor. Por consiguiente  $b_3 = 3.57$  (aproximadamente).
- En general, al superar un determinado  $b_n$ , se crea un  $2^n$ - ciclo atractor que se mantiene hasta  $b_{n+1}$

## 1. Sistemas dinámicos discretos

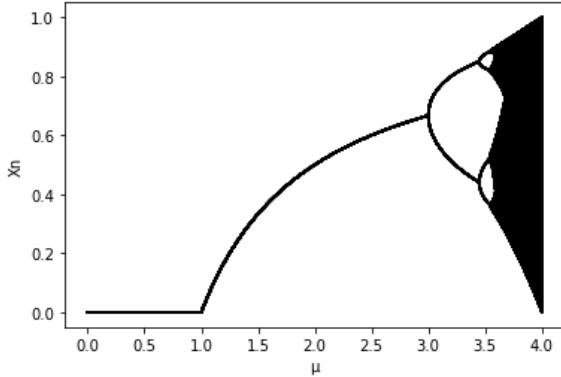


Figura 1.7.: [Python] Representación del diagrama de bifurcación o diagrama de Feigenbaum para  $x_0 = \frac{1}{2}$ .

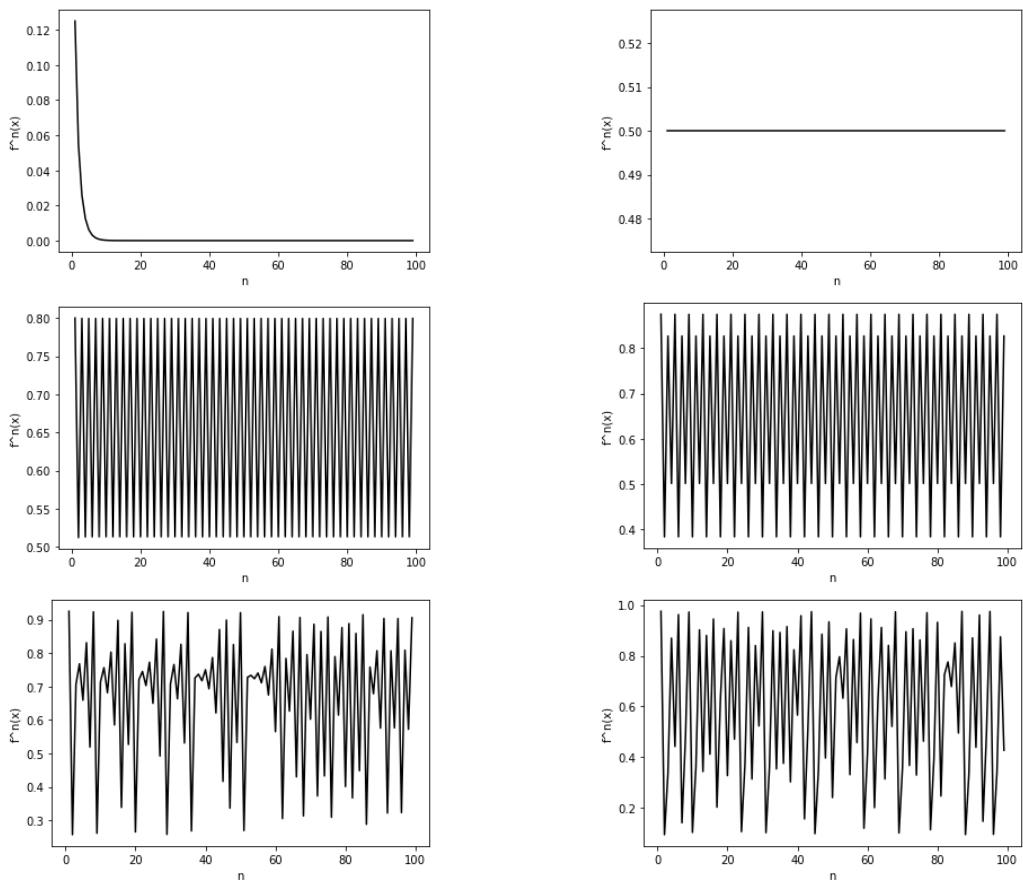


Figura 1.8.: [Python] Representación de los valores obtenidos por las iteradas  $f_\mu^n(x)$  a partir de un valor inicial  $x_0 = \frac{1}{2}$  (i)  $\mu = \frac{1}{2}$  (ii)  $\mu = 2$  (iii)  $\mu = \frac{16}{5}$  (iv)  $\mu = \frac{7}{2}$  (v)  $\mu = \frac{37}{10}$  (vi)  $\mu = \frac{39}{10}$

Sean  $b_1 = 3, b_2 = 1 + \sqrt{6}, b_3 = \dots$  los puntos donde se bifurca el diagrama de Feigenbaum. Sea  $d_k = b_{k+1} - b_k$  la distancia entre dos puntos consecutivos. Calculamos la razón entre cada par de longitudes consecutivas

$$\delta_k = \frac{d_k}{d_{k+1}} = \frac{b_{k+1} - b_k}{b_{k+2} - b_{k+1}}$$

Se puede probar que

$$\delta = \lim_{k \rightarrow \infty} \delta_k = 4.669202\dots$$

A  $\delta$  se le llama constante de Feigenbaum o constante del caos.

Como  $\{d_k\}$  es asintóticamente geométrica de razón  $\delta$  se tiene:

$$b_\infty = \lim_{k \rightarrow \infty} b_k \simeq b_1 + \frac{\delta d_1}{\delta - 1} = 3.57199\dots$$

Una mejor aproximación

$$b_\infty = \lim_{k \rightarrow \infty} b_k \simeq b_2 + \frac{\delta d_2}{\delta - 1} = 3.56987\dots$$

Al punto  $b_\infty$  se le llama **punto de Feigenbaum** o **punto de entrada al caos**.

### 1.4.3. Mapa de Gauss

**Definición 1.14.** Sea  $G: \mathbb{R} \rightarrow \mathbb{R}$  continua. El **mapa de Gauss** viene dado por:

$$G(x) = e^{-\alpha x^2} + \beta \quad \alpha, \beta \in \mathbb{R} \text{ constantes}$$

El sistema dinámico discreto no lineal asociado al mapa de Gauss está definido por la relación de recurrencia:

$$x_{n+1} = G(x_n)$$

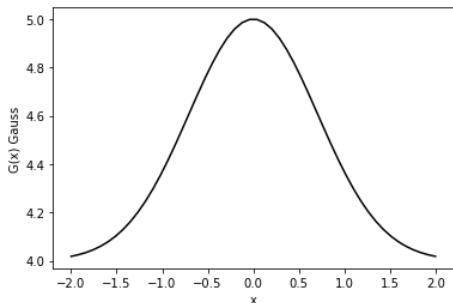


Figura 1.9.: [Python] Mapa de Gauss para  $\alpha = 1$  y  $\beta = 4$

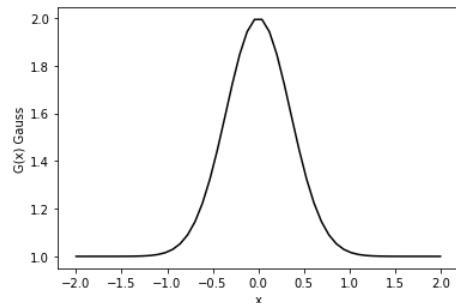


Figura 1.10.: [Python] Mapa de Gauss para  $\alpha = 4$  y  $\beta = 1$

*Observación 1.8.* El parámetro  $\beta$  traslada la función de Gauss en el eje de ordenadas. El parámetro  $\alpha$  tiene relación con la anchura de la curva.

## 1. Sistemas dinámicos discretos

**Proposición 1.9.** Los puntos fijos del mapa de Gauss son las intersecciones de  $G(x)$  con la diagonal  $y = x$ . Dados  $\alpha$  y  $\beta$  constantes, el mapa de Gauss se interseca con la diagonal en uno, dos o tres puntos (Figura 1.11).

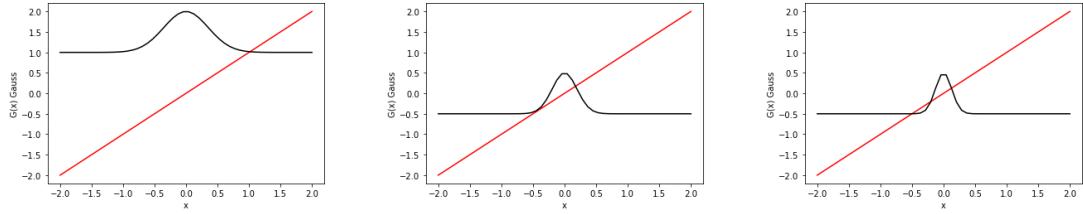


Figura 1.11.: [Python] Posibles intersecciones de la curva de Gauss con la diagonal.

El mapa de Gauss tiene dos puntos de inflexión en  $x = \pm \frac{1}{\sqrt{2\alpha}}$ . Esto significa, que los puntos fijos hacen una transición entre estables e inestables. A medida que  $\beta$  aumenta de  $-1$  los puntos fijos se vuelven inestables y se produce una secuencia de periodo dos y periodo cuatro, hasta finalmente volver a periodo dos. Este comportamiento del mapa de Gauss y otros más, se pueden observar en el diagrama de bifurcación (Figura 1.12).

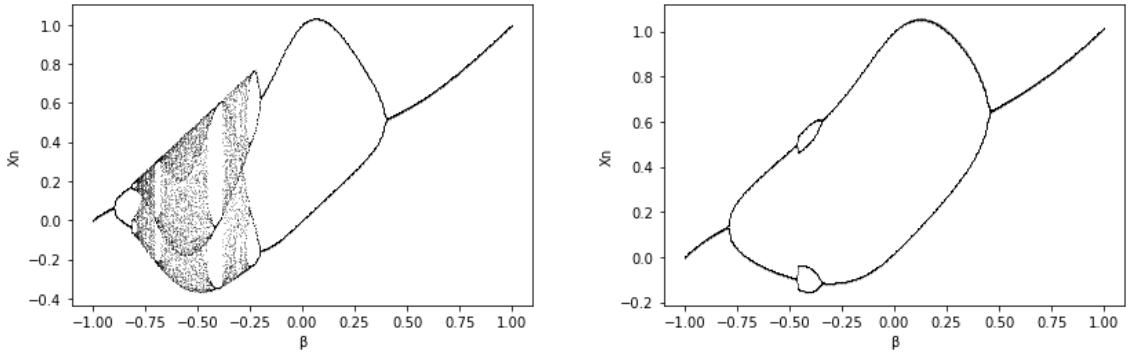


Figura 1.12.: [Python] Diagrama de bifurcación del mapa de Gauss con valor inicial  $x_0 = 0$  para (i)  $\alpha = 4$  (ii)  $\alpha = 8$

### 1.4.4. Mapa de Hénon

**Definición 1.15.** Sea  $P: \mathbb{R} \rightarrow \mathbb{R}$  y  $Q: \mathbb{R} \rightarrow \mathbb{R}$  continuas. El **mapa de Hénon** viene dado por:

$$P(x_n, y_n) = 1 + y_n - \alpha x_n^2$$

$$Q(x_n, y_n) = \beta x_n$$

donde  $\alpha > 0$   $|\beta| < 1$ .

El sistema dinámico discreto no lineal asociado al mapa bidimensional de Hénon viene definido por las siguientes relaciones de recurrencia:

$$x_{n+1} = P(x_n, y_n)$$

$$y_{n+1} = Q(x_n, y_n)$$

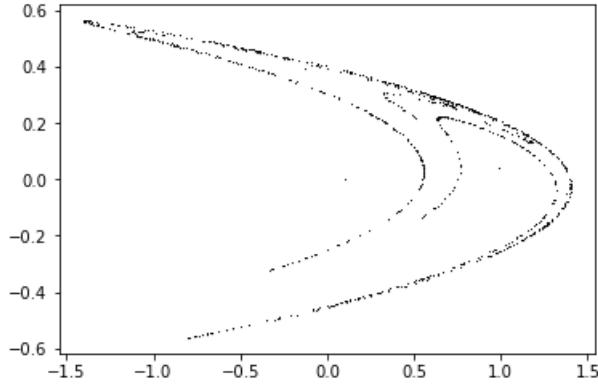


Figura 1.13.: [Python] Mapa de Hénon para  $\alpha = 1.2$ ,  $\beta = 0.4$  con valor inicial  $(0.1, 0)$

**Definición 1.16.** Sea  $P: \mathbb{R} \rightarrow \mathbb{R}$  y  $Q: \mathbb{R} \rightarrow \mathbb{R}$  continuas. Los puntos fijos o puntos de equilibrio del mapa de Hénon son aquellos  $(x^*, y^*) \in \mathbb{R} \times \mathbb{R}$  que cumplen:

$$P(x^*, y^*) = x^*$$

$$Q(x^*, y^*) = y^*$$

**Proposición 1.10.** El sistema dinámico asociado al mapa de Hénon tiene dos puntos fijos si y solo si  $(1 - \beta)^2 + 4\alpha > 0$

*Demostración.* Según la definición 1.16 los puntos fijos del mapa de Hénon son aquellos que cumplen:

$$x^* = 1 - \alpha(x^*)^2 + y$$

$$y^* = \beta x^*$$

La solución del sistema es:

$$x^* = \frac{(\beta - 1) \pm \sqrt{(1 - \beta)^2 + 4\alpha}}{2\alpha} \quad y^* = \beta \left( \frac{(\beta - 1) \pm \sqrt{(1 - \beta)^2 + 4\alpha}}{2\alpha} \right)$$

Si  $(1 - \beta)^2 + 4\alpha > 0$ , el mapa de Hénon tiene dos puntos fijos.  $\square$

**Teorema 1.10.** Sea  $(x^*, y^*) \in \mathbb{R} \times \mathbb{R}$  un punto fijo del mapa de Hénon. Sea la matriz Jacobiana en el punto fijo:

$$J(x^*, y^*) = \begin{pmatrix} \frac{\partial P}{\partial x}(x^*, y^*) & \frac{\partial P}{\partial y}(x^*, y^*) \\ \frac{\partial Q}{\partial x}(x^*, y^*) & \frac{\partial Q}{\partial y}(x^*, y^*) \end{pmatrix} \quad (1.4)$$

Sean  $\lambda_1$  y  $\lambda_2$  los valores propios de la matriz Jacobiana (1.4). Si  $|\lambda_1| < 1$  y  $|\lambda_2| < 1$  entonces el punto fijo  $(x^*, y^*)$  es asintóticamente estable, en caso contrario será inestable.

**Definición 1.17.** Sea  $(x^*, y^*) \in \mathbb{R} \times \mathbb{R}$  un punto fijo del mapa de Hénon. Sean  $\lambda_1$  y  $\lambda_2$  los valores propios de la matriz Jacobiana (1.4). Si  $|\lambda_1| \neq 1$  y  $|\lambda_2| \neq 1$  el punto fijo es hiperbólico. Si  $|\lambda_1| = 1$  o  $|\lambda_2| = 1$  el punto fijo es no hiperbólico.



## 2. Introducción a la teoría del caos

En el capítulo anterior, se hizo una breve mención sobre el concepto de caos en sistemas dinámicos discretos. La finalidad de este, es la de dar respuesta a cuestiones como: ¿Qué es el caos?, ¿Cuáles son sus características?, ¿Qué relación existe con los sistemas dinámicos discretos?, ... Una vez respondidas, se presentarán ejemplos de sistemas dinámicos caóticos, se demostrará el famoso teorema de Li-Yorke y finalizaremos con el exponente de Lyapunov.

Los libros [Dev89], [HN92] y [Lyn18] servirán de apoyo a la teoría descrita en este capítulo.

### 2.1. Definición de caos en sistemas dinámicos

Existen múltiples definiciones posibles de caos, la más extendida y utilizada es la que dió **Robert L. Devaney**, se trata de una visión más topológica.

**Definición 2.1 (Definición R. L. Devaney).** Se dice que un sistema dinámico  $(X, f)$  es caótico si:

- es sensible respecto a las condiciones iniciales.
- es topológicamente transitivo.
- sus puntos periódicos son densos en  $X$ .

**Definición 2.2.** Se dice que un sistema dinámico  $(X, f)$  es **sensible a las condiciones** iniciales si existe un  $\delta > 0$  tal que para todo  $x \in X$  y todo  $\epsilon > 0$  existe  $y \in X$  con  $|x - y| < \epsilon$  y existe  $n \in \mathbb{N}$  tales que  $|f^n(x) - f^n(y)| > \delta$ .

*Observación 2.1.* Un  $(X, f)$  es sensible respecto a las condiciones iniciales, si existe al menos un punto arbitrario cerca de  $x$  que eventualmente se separa de  $x$  mediante una secuencia de iteraciones de  $f$ . En otras palabras, pequeños errores en la estimación de valores de la función se pueden ampliar considerablemente al iterarla.

**Definición 2.3.** Se dice que un sistema dinámico  $(X, f)$  es **topológicamente transitivo** si dados dos abiertos  $U, V \subset X$ , existe  $n \in \mathbb{N}$  tal que  $f^n(U) \cap V \neq \emptyset$ .

*Observación 2.2.* Si  $(X, f)$  es topológicamente transitivo, el sistema no puede descomponerse en dos subconjuntos disjuntos invariantes con interior no vacío.

**Definición 2.4.** Sea  $(X, f)$  un sistema dinámico. Se dice que  $x \in X$  posee una órbita densa si para cualquier  $y \in X$  y  $\forall \epsilon > 0$ ,  $\exists n > 0$  tal que  $d(f^n(x), y) < \epsilon$ .

**Proposición 2.1.** *Si el mapa  $f$  de un sistema dinámico  $(X, f)$  posee una órbita densa, entonces  $(X, f)$  es topológicamente transitivo.*

## 2. Introducción a la teoría del caos

**Definición 2.5.** Los puntos periódicos de  $(X, f)$  son **densos** en  $X$  si para cualquier abierto  $U \subset X$ , existe un punto periódico en  $U$ .

En resumen, si  $(X, f)$  es un sistema dinámico caótico entonces posee tres componentes: impredicibilidad, irreducibilidad y regularidad.

**Definición 2.6.** Se dice que un sistema dinámico  $(X, f)$  es caótico **en sentido de Li-Yorke** si:

- es sensible respecto a las condiciones iniciales.
- sus puntos periódicos son densos en  $X$ .

**Proposición 2.2.** Toda función caótica en sentido de Devaney también lo es en sentido de Li-Yorke, sin embargo el recíproco no es cierto (ejemplo 2.2).

## 2.2. El Mapa Shift y la dinámica simbólica

El mapa Shift es un ejemplo de dinámica simbólica que permite estudiar la dinámica de los sistemas siguiendo el comportamiento de un punto.

**Definición 2.7.**  $\Sigma_2 = \{s = (s_0 s_1 s_2 \dots) | s_j = 0, 1\}$

Se dice que  $\Sigma_2$  es el espacio de secuencias de ceros y unos. Sus elementos son cadenas infinitas de enteros como  $(000\dots)$ ,  $(0101\dots)$ , ...

Para convertir a  $\Sigma_2$  en un espacio métrico, es importante tener presentes las nociones de distancia y espacio métrico [Pay21].

**Definición 2.8.** Una **distancia** en un conjunto no vacío  $X$  es una función  $d: X \times X \rightarrow \mathbb{R}$  verificando las siguientes condiciones:

- (I) *Desigualdad triangular:*  $d(x, z) \leq d(x, y) + d(y, z) \quad \forall x, y, z \in X$
- (II) *Simetría:*  $d(x, y) = d(y, x) \quad \forall x, y \in X$
- (III) *No degeneración:* Para  $x, y \in X$ , se tiene:  $d(x, y) = 0 \iff x = y$

**Definición 2.9.** Si un conjunto no vacío  $X \subset \mathbb{R}^n$  dispone de una distancia  $d: X \times X \rightarrow \mathbb{R}$  decimos  $X = (X, d)$  es un **espacio métrico**.

Dadas dos secuencias  $s = (s_0 s_1 s_2 \dots)$ ,  $t = (t_0 t_1 t_2 \dots) \in \Sigma_2$ , definimos la distancia entre ellas como:

$$d(s, t) = \sum_{i=0}^{\infty} \frac{|s_i - t_i|}{2^i}$$

Dado que  $|s_i - t_i|$  es 0 o 1, esta serie infinita está dominada por la serie geométrica:

$$\sum_{i=0}^{\infty} \frac{1}{2^i} = 2$$

y por tanto converge.

**Proposición 2.3.** *d es una distancia en  $\Sigma_2$ .*

*Demostración.* Claramente,  $d(s, t) \geq 0$  para todo  $s, t \in \Sigma_2$  y  $d(s, t) = 0$  si y solo si  $s_i = t_i$ . Dado que  $|s_i - t_i| = |t_i - s_i|$ , se sigue que  $d(s, t) = d(t, s)$ . Por último, si  $r, s, t \in \Sigma_2$ , entonces  $|r_i - s_i| + |s_i - t_i| \geq |r_i - t_i|$  de donde se deduce que  $d(r, s) + d(s, t) \geq d(r, t)$   $\square$

**Proposición 2.4.** *Sea  $s, t \in \Sigma_2$ ,  $d(s, t) \leq \frac{1}{2^n}$  si y solamente si  $s_i = t_i$  para  $i = 0, 1, 2, \dots, n$ .*

*Demostración.* Si  $s_i = t_i$ , para  $i \leq n$ , entonces:

$$d(s, t) = \sum_{i=0}^n \frac{|s_i - t_i|}{2^i} + \sum_{i=n+1}^{\infty} \frac{|s_i - t_i|}{2^i} \leq \sum_{i=n+1}^{\infty} \frac{1}{2^i} = \frac{1}{2^n}$$

Por otro lado, si  $s_j \neq t_j$  para  $j \leq n$ , entonces:

$$d(s, t) \geq \frac{1}{2^j} \geq \frac{1}{2^n}$$

Consecuentemente, si  $d(s, t) < \frac{1}{2^n}$ , entonces  $s_i = t_i$ , para  $i \leq n$ .  $\square$

**Definición 2.10.** El mapa Shift  $\sigma: \Sigma_2 \rightarrow \Sigma_2$  viene dado por  $\sigma(s_0s_1s_2\dots) = (s_1s_2s_3\dots)$

El mapa shift simplemente olvida la primera entrada y desplaza todas las demás hacia la izquierda.

**Proposición 2.5.** *El mapa  $\sigma: \Sigma_2 \rightarrow \Sigma_2$  es continuo*

*Demostración.* Sea  $\epsilon > 0$  y  $s = s_0s_1s_2\dots$ . Tomamos  $n$  de tal forma que  $\frac{1}{2^n} < \epsilon$ . Sea  $\delta = \frac{1}{2^{n+1}}$ , si  $t = t_0t_1t_2\dots$  satisface  $d(s, t) < \delta$ , entonces por la proposición 2.4 tenemos que  $s_i = t_i$  para  $i \leq n+1$ . La  $i$ -ésima entrada de  $\sigma(s)$  y  $\sigma(t)$  coincide para  $i \leq n$ . Por eso,  $d(\sigma(s), \sigma(t)) \leq \frac{1}{2^n} < \epsilon$ .  $\square$

**Teorema 2.1.** *El mapa Shift es caótico*

*Demostración.* Consideramos la sucesión  $x \in \Sigma_2$  dada por:

$$x = (01|00011011|000001\dots)$$

Observe que dado  $(a_0a_1\dots) \in \Sigma_2$  y  $\epsilon > 0$  podemos hallar  $N \in \mathbb{N}$  tal que  $\epsilon > \frac{1}{2^N}$ . En ese caso, como  $x$  contiene todas las posibles sucesiones finitas de ceros y unos, existe  $M \in \mathbb{N}$  tal que  $\sigma^M(x) = (a_0a_1\dots a_N\dots)$ . De este modo:

$$d(\sigma^M(x), (a_0a_1\dots)) = \sum_{i=N+1}^{\infty} \frac{|x_i - a_i|}{2^i} \leq \sum_{i=N+1}^{\infty} \frac{1}{2^i} = \frac{1}{2^N} < \epsilon$$

Por tanto, la órbita de  $x$  es densa en  $\Sigma_2$ , lo que prueba que el mapa Shift posee una órbita densa y por tanto  $\sigma$  es topológicamente transitivo (proposición 2.1).

A continuación, vamos a probar que las órbitas periódicas son densas en  $\Sigma_2$ . Dado  $(a_0a_1\dots) \in \Sigma_2$ ,  $\epsilon > 0$ ,  $N \in \mathbb{N}$  tal que  $\frac{1}{2^N} < \epsilon$ . Definimos la sucesión periódica  $x^a = (a_0a_1\dots a_Na_0a_1\dots a_Na_0\dots)$ . Por el mismo argumento, se tiene  $d(x^a, a) < \frac{1}{2^N} < \epsilon$ , lo que prueba la densidad de las órbitas

## 2. Introducción a la teoría del caos

periódicas.

Finalmente, respecto a la dependencia sensible a las condiciones iniciales, sea  $(a_0 \dots) \in \Sigma_2$ ,  $\epsilon > 0$  y  $N \in \mathbb{N}$  tal que  $\frac{1}{2^N} < \epsilon$ . Sea  $\bar{a}_k$ , el opuesto de  $a_k$ . Definimos  $y^a = (a_0 a_1 \dots a_N \bar{a}_{N+1} \bar{a}_{N+2} \dots)$ . Se tiene que  $d(a, y^a) < \frac{1}{2^N} < \epsilon$ , como todos los términos desde el  $(N + 1)$ -ésimo en adelante son distintos, se tiene que:

$$d(\sigma^N(y^a), \sigma^N(a)) = \sum_{i=1}^{\infty} \frac{|a_{i+N} - \bar{a}_{i+N}|}{2^i} = \sum_{i=1}^{\infty} \frac{1}{2^i} = 1 > \frac{1}{2}$$

Luego el mapa shift tiene dependencia sensible a las condiciones iniciales con constante de sensibilidad, por ejemplo, igual a  $\frac{1}{2}$ .  $\square$

### 2.3. Teorema de Li-Yorke: periodo tres implica caos

Las fuentes utilizadas para poder enunciar y demostrar el teorema son [Elaoo] [TYL75].

En 1964, **A. N. Sharkovsky** descubrió uno de los teoremas más remarcables para mapas continuos en la recta real. El teorema permaneció totalmente desconocido en occidente hasta que en 1975 **Li y Yorke** publicaron el artículo: "Periodo tres implica caos" en la revista American Mathematical Monthly [TYL75]. En este artículo, se demostró que si un mapa continuo  $f$  tiene un punto periódico de periodo tres, entonces tiene puntos de cualquier periodo. Poco después, se descubrió que el teorema de Li-Yorke es solo un caso del teorema de Sharkovsky.

A. N. Sharkovsky presentó un nuevo orden de los enteros positivos comenzando por el tres. Demostró que si  $k \triangleright r$  y  $f$  tiene un punto periódico de periodo  $k$  entonces  $f$  tiene un punto periódico de periodo  $r$ . El orden de Sharkovsky viene definido de la siguiente forma:

$$\begin{aligned} & 3 \triangleright 5 \triangleright 7 \triangleright \dots \\ & 2 \times 3 \triangleright 2 \times 5 \triangleright 2 \times 7 \triangleright \dots \\ & 2^2 \times 3 \triangleright 2^2 \times 5 \triangleright 2^2 \times 7 \triangleright \dots \\ & 2^n \times 3 \triangleright 2^n \times 5 \triangleright 2^n \times 7 \triangleright \dots \\ & \dots \triangleright 2^n \triangleright \dots \triangleright 2^3 \triangleright 2^2 \triangleright 1 \end{aligned}$$

Primero se listan los números impares, luego los números impares multiplicados por 2, números impares multiplicados por  $2^2$  y, en general, números impares multiplicados por  $2^n$ . Todo ello, seguido de las potencias de 2 en orden descendente. Es fácil ver que este orden abarca todos los enteros positivos.

**Notación**  $m \triangleright n$  denota que  $m$  aparece antes que  $n$  en el orden de Sharkovsky.

### 2.3. Teorema de Li-Yorke: periodo tres implica caos

**Teorema 2.2 (Teorema Sharkovsky).** Sea  $f: I \rightarrow I$  un mapa continuo, donde  $I \subset \mathbb{R}$ . Si  $f$  tiene un punto periódico de periodo  $k$ , entonces tendrá un punto periódico de periodo  $r$  para todo  $r$  con  $k \triangleright r$ .

La demostración del teorema de Sharkovsky requiere múltiples resultados previos para su realización. Debido a su extensión, nos centraremos en otros asuntos relativos al caos. La demostración se encuentra en [Elaoo].

El teorema anterior es óptimo en el sentido de que si  $k \triangleright r$  entonces existen mapas continuos con puntos periódicos de periodo  $r$  que no tienen puntos de periodo  $k$ . Una prueba de ello es el ejemplo 2.1 para  $3 \triangleright 5$ .

**Ejemplo 2.1.** Este ejemplo, muestra un mapa continuo con puntos periódicos de periodo 5 y sin puntos de periodo 3. Se define  $f: [1, 5] \rightarrow [1, 5]$ :

$$f(1) = 3, f(2) = 5, f(3) = 4, f(4) = 2, f(5) = 1$$

Se asume que en cada intervalo  $[n, n+1]$  con  $1 \leq n \leq 4$ ,  $f$  es una recta que une ambos extremos (Figura 2.1).

En primer lugar, aplicamos  $f$  a cada uno de los puntos:

$$1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 1$$

Se observa que todos los puntos son de periodo 5 y que no hay ninguno de periodo 3:

$$f^3([1, 2]) = [2, 5], f^3([2, 3]) = [3, 5], f^3([4, 5]) = [1, 4]$$

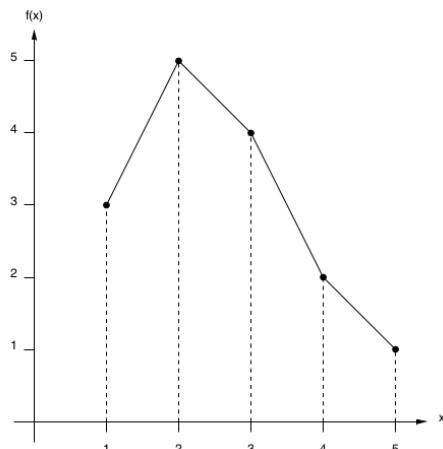


Figura 2.1.: [Elaoo] Ejemplo de mapa continuo con puntos periódicos de periodo 5 y sin puntos de periodo 3

2. Introducción a la teoría del caos

**Teorema 2.3 (Teorema Li-Yorke).** *Sea  $f: I \rightarrow I$  un mapa continuo, donde  $I \subset \mathbb{R}$ . Si  $f$  tiene un punto periódico de periodo 3, entonces:*

- (i)  *$f$  tendrá un punto periódico para cada  $k = 1, 2, \dots$ .*
- (ii) *Existe un subconjunto no numerable  $S \subset I$  (que contiene puntos no periódicos), y satisface las siguientes condiciones:*
  - a) *Para cada  $p, q \in S$  con  $p \neq q$  se cumple:*

$$\limsup_{n \rightarrow \infty} |f^n(p) - f^n(q)| > 0$$

$$\liminf_{n \rightarrow \infty} |f^n(p) - f^n(q)| = 0$$

- b) *Para cada  $p \in S$  y cada punto periódico  $q \in I$ :*

$$\limsup_{n \rightarrow \infty} |f^n(p) - f^n(q)| > 0$$

El objetivo es demostrar (i) del teorema de Li y Yorke, para ello necesitaremos unos resultados previos.

**Lema 2.1.** *Sea  $f: I \rightarrow \mathbb{R}$  continua, donde  $I \subset \mathbb{R}$ . Para cada compacto  $J \subset f(I)$ , existe un intervalo compacto  $Q \subset I$  tal que  $f(Q) = J$*

*Demostración.* Sea  $J = [f(p), f(q)] \subset f(I)$ , con  $p, q \in I$ . Sin perdida de generalidad, suponemos  $p < q$  y tomamos

$$r = \max \{x \in [p, q] : f(x) = f(p)\}$$

$$s = \min \{y \in [r, q] : f(y) = f(q)\}$$

Consideramos el intervalo  $Q = [r, s] \subset I$ . Por el teorema del valor intermedio, se tiene que:

$$J = f(Q)$$

□

**Lema 2.2.** *Sea  $f: I \rightarrow I$  continua y  $\{I_n\}_{n=0}^{\infty}$  una sucesión de compactos con  $I_n \subset I$  y  $I_{n+1} \subset f(I_n)$  para  $n \in \mathbb{N}$ . Entonces, existe una sucesión de compactos  $Q_n$  de tal forma que  $Q_{n+1} \subset Q_n \subset I_0$  y  $f^n(Q_n) = I_n$  para  $n \in \mathbb{N}$ .*

*Demostración.* Sea  $Q_0 = I_0$ . Entonces,  $f^0(Q_0) = I_0$ . Si se ha definido  $Q_{n-1}$  de tal forma que  $f^{n-1}(Q_{n-1}) = I_{n-1}$ , entonces  $I_n \subset f(I_{n-1}) = f^n(Q_{n-1})$ . Aplicando el lema 2.1 en  $f^n$ , existe un compacto  $Q_n \subset Q_{n-1}$  tal que  $f^n(Q_n) = I_n$  □

Estamos preparados para demostrar **Teorema Li-Yorke**

*Demostración.* Supongamos que  $f$  tiene un 3-ciclo  $\{x, f(x), f^2(x)\}$  de la siguiente forma  $\{a, b = f(a), c = f(b)\}$  con  $a < b < c$  o  $a > b > c$ . Por ejemplo, si  $x < f^2(x) < f(x)$ , tomamos  $a = f(x)$ ,  $b = f(a)$ ,  $c = f^2(a)$  y por tanto,  $a > b > c$ . Supongamos que  $c > b > a$ , sean  $J = [a, b]$  y  $L = [b, c]$ . Para  $k > 1$ , sea  $\{I_n\}$  una sucesión de intervalos con  $I_n = L$  para  $n = 0, 1, \dots, k-2$ ,  $I_{k-1} = J$  y  $I_{n+k} = I_n$  para  $n \in \mathbb{N}$ . La sucesión  $\{I_n\}$  se puede ver de esta manera:

$$L, L, \dots, L, J, L, L, \dots, L, J, L, L, \dots, L, J, \dots$$

donde  $L$  aparece  $k-1$  veces antes y después de un  $J$ . Si  $k=1$ , sea  $I_n = L$  para todo  $n \in \mathbb{N}$ . Como  $f(a) = b$ ,  $f(b) = c$  y  $f(c) = a$  entonces por el teorema del valor intermedio  $L, J \subset f(L)$  y  $L \subset f(J)$ .

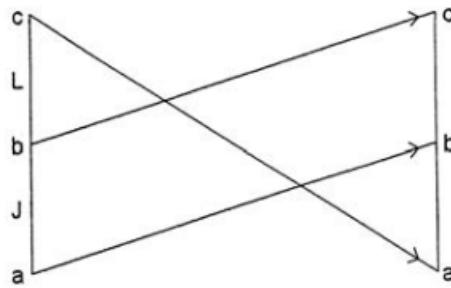


Figura 2.2.: [Elaoo]  $f(a) = b$ ,  $f(b) = c$  y  $f(c) = a$  entonces  $J \subset f(L)$  y  $L \subset f(J)$

Aplicamos el lema 2.2 para producir la sucesión de compactos  $\{Q_n\}$  con  $Q_k \subset Q_0 = L$  y  $f^k(Q_k) = I_k = L$ . Consecuentemente,  $L \subset f^k(L)$ . Por el teorema 1.2, se concluye que  $f^k$  tiene un punto fijo en  $L$  y por tanto,  $f$  tiene un punto de periodo  $k$  en  $I$ .  $\square$

La demostración (ii) del teorema de Li y Yorke se encuentra recogida en [TYL75]. Esta segunda parte, hace referencia a que bajo las hipótesis del teorema,  $f$  tiene dependencia sensible respecto a las condiciones iniciales.

Para finalizar, vamos a ver un ejemplo de una función caótica en sentido de Li y Yorke , pero no en sentido de Devaney.

**Ejemplo 2.2.** Consideramos la función a trozos  $f: [0, 1] \rightarrow [0, 1]$  definida como:

$$f(x) = \begin{cases} x & \text{si } 0 \leq x \leq \frac{1}{2} \\ 2x - \frac{1}{2} & \text{si } \frac{1}{2} < x \leq \frac{3}{4} \\ \frac{5}{2} - 2x & \text{si } \frac{3}{4} < x \leq 1 \end{cases}$$

La función a trozos actúa como el mapa tienda en el intervalo  $(\frac{1}{2}, 1]$ . Por lo tanto,  $f$  tendrá al menos un punto periódico de periodo 3 y será caótica en sentido de Li y Yorke. Sin embargo,  $f$  no es caótica en sentido de Devaney porque no es topológicamente transitiva, no existe un  $x \in [0, \frac{1}{2}]$  tal que  $f^n(x) \in (\frac{1}{2}, 1]$  para algún  $n$  ya que todos los puntos de  $[0, \frac{1}{2}]$  son fijos.

## 2.4. Exponente de Lyapunov

Un método que se utiliza muy a menudo para determinar si un sistema es caótico o no, es el cálculo del exponente de Lyapunov.

En sistemas dinámicos discretos unidimensionales, el exponente de Lyapunov mide la divergencia entre dos órbitas que difieren en  $\delta_0 > 0$  con valor inicial  $x_0$ , es decir,  $x_0 \pm \delta_0$ . Por consiguiente, tras una iteración la distancia entre las dos viene dada por la siguiente aproximación:

$$\delta_1 \approx |f'(x_0)|\delta_0 = M_0\delta_0$$

donde  $M_0$  es el factor de magnificación de la primera iteración. En la segunda iteración se tiene que:

$$\delta_2 \approx |f'(x_1)|\delta_1 = M_1M_0\delta_0$$

Siguiendo este procedimiento, se puede concluir que el factor de magnificación de un ciclo con órbita de periodo  $k$  es el producto:

$$M_0M_1 \cdots M_{k-1}$$

Como este producto es la acumulación de estos factores, tiene sentido considerar el promedio del mismo. El más conveniente es la media geométrica:

$$(M_0M_1 \cdots M_{k-1})^{\frac{1}{k}}$$

Tomando logaritmos obtenemos la media aritmética:

$$\lambda = \ln(M_0M_1 \cdots M_{k-1})^{\frac{1}{k}} = \frac{1}{k}(\ln(M_0) + \cdots + \ln(M_{k-1})) = \frac{1}{k}(\ln|f'(x_0)| + \cdots + \ln|f'(x_{k-1})|)$$

Intuitivamente, la condición de estabilidad de una órbita periódica se verifica cuando el factor de magnificación es menor que 1, es decir, si  $\lambda < 0$  entonces es estable y si  $\lambda > 0$  inestable.

**Definición 2.11.** Sea  $f \in C^1(\mathbb{R})$  y  $x_0$  un valor inicial. El **exponente de Lyapunov**  $\lambda(x_0)$  del mapa  $f$  se define como:

$$\lambda(x_0) = \lim_{k \rightarrow \infty} \frac{1}{k}(\ln|f'(x_0)| + \cdots + \ln|f'(x_{k-1})|)$$

En el caso de que algunas de las derivadas sean 0,  $\lambda(x_0) = -\infty$ .

**Definición 2.12.** El **coeficiente de Lyapunov** se define como la exponencial del exponente de Lyapunov:

$$L(x_0) = e^{\lambda(x_0)}$$

**Definición 2.13.** Se dice que la órbita  $O^+(x_0) = \{x_0, x_1, x_2, \dots\}$  es **asintóticamente periódica** si existe una órbita periódica  $\{y_0, y_1, y_2, \dots\}$  tal que:

$$\lim_{n \rightarrow \infty} |x_n - y_n| = 0$$

**Definición 2.14.** Sea  $f$  un mapa continuo y  $\{x_0, x_1, x_2, \dots\}$  una órbita de  $f$ . Se dice que la órbita es caótica en sentido de Li y Yorke si:

- $O^+(x_0) = \{x_0, x_1, x_2, \dots\}$  no es asintóticamente periódica.
- Ningún exponente de Lyapunov es cero.
- $\lambda(x_0) > 0$ .

Como se puede ver de la definición 2.14, una importante interpretación del exponente de Lyapunov es que mide la información perdida durante el cálculo de las iteraciones de  $f$ .

**Teorema 2.4.** Si al menos uno de los exponentes de Lyapunov es positivo, el sistema es caótico; si el exponente de Lyapunov es negativo, la órbita es periódica y si es cero, ocurre una bifurcación.

**Ejemplo 2.3 (Exponente de Lyapunov en el mapa tienda).** El mapa tienda es uno de los ejemplos en los que el cálculo del exponente de Lyapunov es sencillo. Claramente, se tiene:

$$T'(x) = \begin{cases} \mu & \text{si } 0 \leq x < \frac{1}{2} \\ -\mu & \text{si } \frac{1}{2} < x \leq 1 \end{cases} \quad 0 \leq \mu \leq 2$$

con  $T'(\frac{1}{2})$  indefinido. Para cualquier órbita que no contenga el punto  $\frac{1}{2}$ , se tiene que  $\lambda(x_0) = \ln(\mu)$ . Cuando  $\mu > 1$ , el teorema 2.4 nos informa de comportamiento caótico.

**Ejemplo 2.4 (Exponente de Lyapunov en el mapa logístico).** El exponente de Lyapunov para ciertos mapas puede ser calculado usando una equivalencia con otro mapa cuyo exponente de Lyapunov sea más fácil de calcular. Sin embargo, en este ejemplo, vamos a calcular el exponente de Lyapunov de una forma más computacional. Dado un  $x_0 = 0.6$  y  $f'_\mu(x) = \mu(1 - 2x)$  la derivada del mapa logístico, la siguiente tabla muestra los valores del exponente de Lyapunov para un  $\mu$  fijo:

$\mu$	0.5	1	2.5	3	3.5	3.6	3.97
$\lambda(x_0)$	-0.69	-0.008	-0.69	-0.0045	-0.867	0.183	0.591

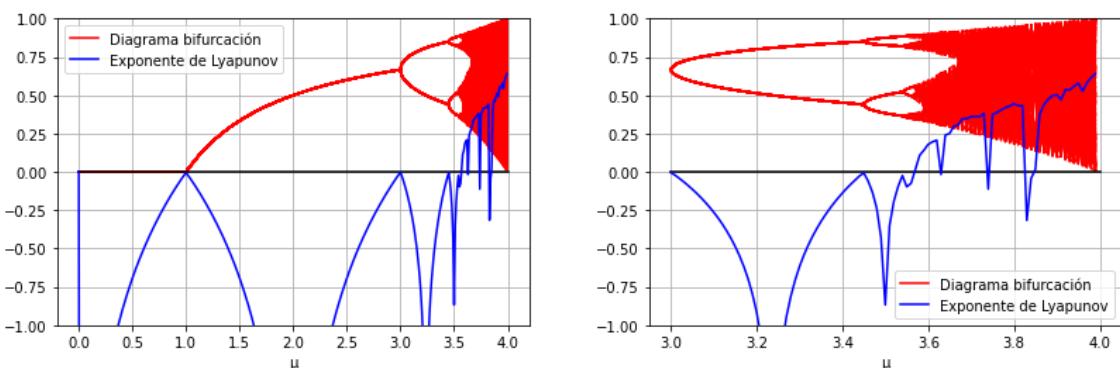


Figura 2.3.: [Python] Exponente de Lyapunov y diagrama de bifurcación para  $x_0 = 0.6$  con  
(i)  $0 \leq \mu \leq 4$  (ii)  $3 \leq \mu \leq 4$

## *2. Introducción a la teoría del caos*

Los resultados numéricos de la tabla anterior y el resultado gráfico de la figura 2.3 concuerdan con el teorema 2.4. Analizando la figura 2.3, se observa que cuanto mayor sea el caos del sistema, mayor es el exponente de Lyapunov . Además, recordemos que  $b_0 = 1, b_1 = 3, b_2 = 1 + \sqrt{6}, \dots$  se conocen como los puntos donde el diagrama se bifurca y es justo donde el exponente de Lyapunov es cero. Es posible apreciar como  $b_\infty = 3.56987\dots$  es el punto de Feigenbaum o punto de entrada al caos del sistema.

## 3. Sistemas dinámicos complejos

Los sistemas dinámicos discretos estudiados hasta el momento están asociados a mapas cuyo dominio es  $\mathbb{R}$ . En este capítulo, se pretende ampliar la teoría descrita en los capítulos 1 y 2 examinando el comportamiento de la dinámica compleja. Se investigarán los conjuntos de Julia y de Mandelbrot analizando sus propiedades y características. Además, se representarán gráficamente ambos conjuntos para respaldar los conocimientos expuestos.

Lo descrito en este capítulo se basa en conceptos detallados en [Fal90], [Bar88], [Elaoo] y [AGo2].

En la dinámica compleja, la manera de analizar y estudiar los sistemas va a ser muy similar a la de los capítulos 1 y 2 con la diferencia de que el dominio de los mapas es  $\mathbb{C}$ . Algunas de las proposiciones, teoremas y definiciones que se utilizarán son una mera extensión de las ya mencionadas para el caso real pero ahora en el plano complejo. Los conceptos de punto fijo, punto periódico, órbita, estabilidad, atracción, hiperbolidad, ... enunciados para  $\mathbb{R}$  se pueden extender para  $z \in \mathbb{C}$ .

**Teorema 3.1.** *Sea  $f: \mathbb{C} \rightarrow \mathbb{C}$ ,  $f \in C^1(\mathbb{C})$  y  $z \in \mathbb{C}$  un punto fijo hiperbólico.*

- (i) *Si  $|f'(z)| < 1$  entonces  $z$  es un punto atractivo (asintóticamente estable). En concreto, si  $|f'(z)| = 0$  entonces a  $z$  se le denomina superatractivo.*
- (ii) *Si  $|f'(z)| > 1$  entonces  $z$  es un punto repulsivo (inestable).*

### 3.1. El conjunto de Julia

Esta sección se dedicará al estudio del conjunto de Julia, uno de los objetos matemáticos más relevantes en teoría de sistemas dinámicos complejos. Este famoso conjunto fue introducido por el matemático francés **Gaston Julia** (1883 - 1978) en su obra [Jul18]

**Definición 3.1.** Sea  $f: \mathbb{C} \rightarrow \mathbb{C}$  un mapa continuo. Se define el **conjunto de Julia relleno** asociado al mapa  $f$ :

$$K(f) = \{z \in \mathbb{C} : O^+(z) \text{ acotada}\}$$

**Definición 3.2.** Se define el **conjunto de Julia**  $J(f)$  (asociado al mapa  $f$ ) como la frontera del conjunto de Julia relleno  $K(f)$ . Equivalentemente,  $J(f)$  se puede definir como la frontera del conjunto de escape:

$$E = \left\{ z \in \mathbb{C} : \lim_{n \rightarrow \infty} |f^n(z)| = \infty \right\}$$

El conjunto de Julia que analizaremos está ligado al mapa cuadrático:

$$z_{n+1} = f_c(z_n) = z_n^2 + c \quad \text{con } c \in \mathbb{C} \tag{3.1}$$

### 3. Sistemas dinámicos complejos

**Notación** El conjunto de Julia relleno y el conjunto de Julia ligados al mapa (3.1) se denotan, respectivamente, como  $K_c$  y  $J_c$ .

Si consideramos la transformación del plano complejo  $\mathbb{C}$  en sí mismo dada por  $f_c(z) = z^2 + c$  donde  $c$  es un número complejo. Es obvio que para un  $z$  suficientemente grande la órbita de  $z$  diverge. Además, si tomamos  $c = 0$ , el conjunto de Julia relleno  $K_0$  asociado al mapa cuadrático  $f_0(z) = z^2$  se corresponde con el disco unidad (Figura 3.1) y  $J_0$  con la circunferencia unidad .

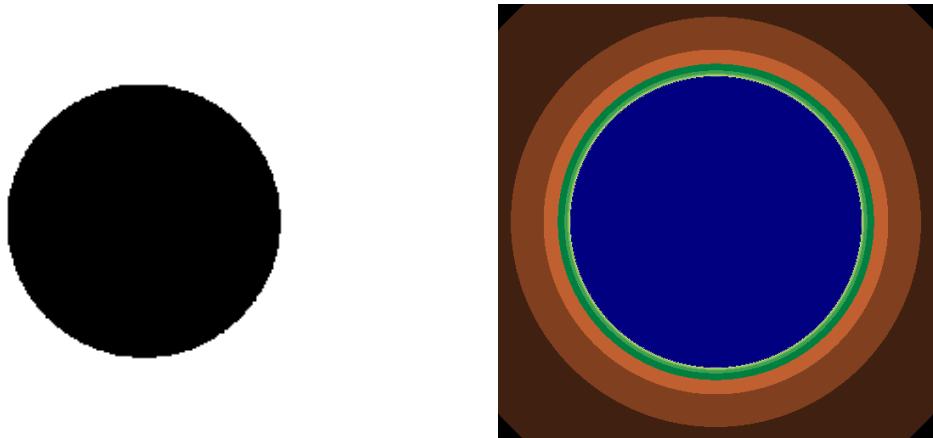


Figura 3.1.: [Python] Conjunto de Julia relleno  $K_0$  asociado al mapa  $f_0(z) = z^2$

$J_0$  es posiblemente el conjunto de Julia más sencillo que existe, por otro lado, cuando  $c$  es un número complejo se obtienen conjuntos con comportamientos más irregulares y difíciles de imaginar a simple vista. Para llevar a cabo el análisis de estos conjuntos se discutirán algunos aspectos teóricos que ayudarán a mejorar la comprensión de  $K_c$  y  $J_c$ . A la hora de generar el conjunto de Julia relleno  $K_c$ , es conveniente conocer el **criterio de escape** que comentaremos a continuación.

**Teorema 3.2 (El criterio de Escape).** *Dado  $z \in \mathbb{C}$  y el mapa  $f_c(z) = z^2 + c$ . Si  $|z| > \max\{2, |c|\}$  entonces:*

$$\lim_{n \rightarrow \infty} |f_c^n(z)| = \infty$$

*Demostración.* Supongamos que  $z \in \mathbb{C}$ ,  $|z| > \max\{2, |c|\}$ . Entonces:

$$\begin{aligned} |f_c(z)| &= |z^2 + c| \geq |z^2| - |c| \\ &\geq |z|^2 - |c| \\ &> |z|^2 - |z| \quad (|z| > |c|) \\ &= |z|(|z| - 1) \end{aligned}$$

Si  $|z| > 2$ , se tiene que  $|z| - 1 = 1 + \eta$ , para algún  $\eta > 0$ . Por tanto se tiene que:

$$|f_c(z)| > (1 + \eta)|z|$$

Por inducción en  $n$  se puede que:

$$|f_c^n(z)| > (1 + \eta)^n |z|$$

Lo cual implica que  $|f_c^n(z)| \rightarrow \infty$  cuando  $n \rightarrow \infty$ . □

### Algoritmo para el conjunto de Julia relleno en blanco y negro

Sea  $r(c) = \max\{|c|, 2\}$ . Se conoce a  $r(c)$  como el **radio umbral** de  $f_c$ . Por el teorema 3.2, para cualquier  $z \in \mathbb{C}$ , si existe  $k \in \mathbb{Z}^+$ , tal que  $|f_c^k(z)| > r(c)$ , entonces la órbita de  $z$  escapa al  $\infty$  y consecuentemente  $z \notin K_c$ . Usando esta observación, vamos a introducir un algoritmo para generar  $K_c$ . Fijamos un número máximo de iteraciones  $N$ . Entonces, para cada  $z$ :

- (I) Si  $|f_c^k(z)| > r(c)$ , para algún  $k \leq N$ , coloreamos el punto  $z$  de blanco.
- (II) Si  $|f_c^k(z)| \leq r(c)$ , para todo  $k \leq N$ , coloreamos el punto  $z$  de negro.

Los puntos negros dan una aproximación del conjunto de Julia relleno  $K_c$

### Algoritmo de iteración inversa (cerco del conjunto de Julia)

Sea  $r(c) = \max\{|c|, 2\}$  el **radio umbral** de  $f_c$ . Se define  $L_0 = \{z \in \mathbb{C} : |z| \leq r(c)\}$ . Sea  $L_{-1} = f_c^{-1}(L_0) = \{z \in \mathbb{C} : f_c(z) \in L_0\}$ . Podemos afirmar que  $L_{-1} \subset L_0$  porque sino entonces existiría  $w \in L_{-1} \setminus L_0$  con  $|w| > r(c)$ . Por el **criterio de escape**, se sigue que  $|f_c(w)| > |w| > r(c)$ , lo cual contradice que  $f_c(w) \in L_0$ . Se define  $L_{-2} = f_c^{-1}(L_{-1})$ ,  $L_{-3} = f_c^{-1}(L_{-2})$ , ...,  $L_{-n-1} = f_c^{-1}(L_{-n})$ . Por tanto, cada  $L_{-n}$  es no vacío ya que contiene los dos puntos fijos de  $f_c$  (son las raíces de  $z^2 - z + c = 0$ ,  $z_{1,1} = \frac{1}{2} + \sqrt{1 - 4c}$ ,  $z_{1,2} = \frac{1}{2} - \sqrt{1 - 4c} \in L_0$ ). Es fácil de ver que:

$$L_0 \supset L_{-1} \supset L_{-2} \supset \cdots \supset L_{-n} \supset \cdots$$

y cada  $L_{-n}$  es cerrado y acotado. De hecho, se tiene que :

$$K_c = \bigcap_{n=0}^{\infty} L_{-n} = \lim_{n \rightarrow \infty} L_{-n}$$

Se observa que  $f_c^{-1}(K_c) = K_c$  desde que  $L_{-n-1} = f_c^{-1}(L_{-n})$ . Además,  $f_c(K_c) = K_c$ . Esto implica que  $K_c$  es invariante. De forma similar, observamos que el conjunto de Julia  $J_c$  es un subconjunto no vacío, cerrado e invariante de  $\mathbb{C}$

**Definición 3.3.** Sea  $f: \mathbb{C} \rightarrow \mathbb{C}$  mapa continuo y  $A \subset \mathbb{C}$ . Se dice que  $A$  es un conjunto invariante si  $f(A) = A$ .

**Teorema 3.3.** Los conjuntos  $K_c$  y  $J_c$  son subconjuntos no vacíos, cerrados, acotados e invariantes de  $\mathbb{C}$ .

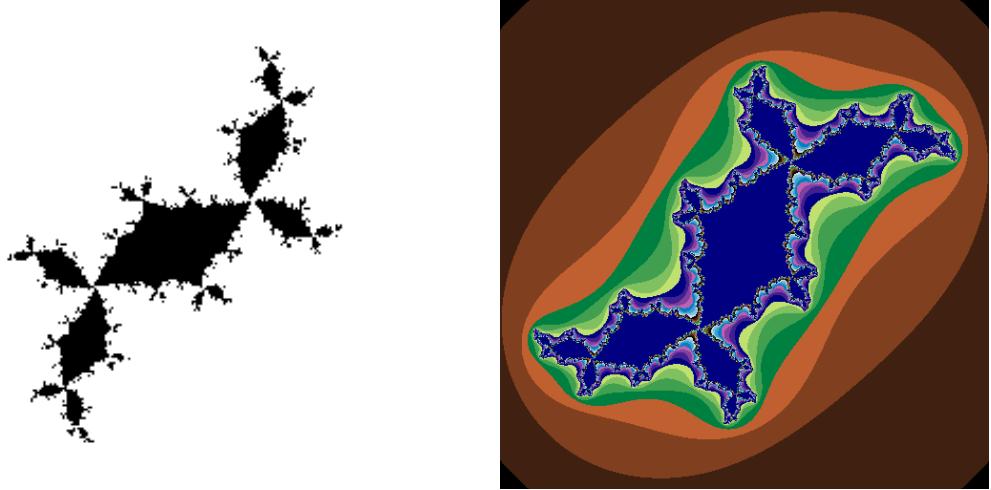


Figura 3.2.: [Python] Conjunto de Julia relleno  $K_{-0.1+0.8i}$  ( El conejo de Douady) (i) Algoritmo en blanco y negro (ii) Algoritmo de iteración inversa

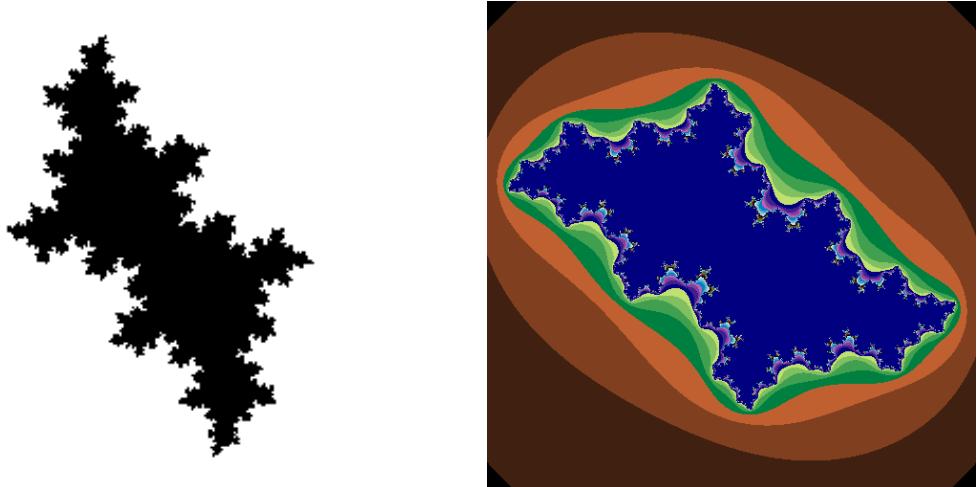


Figura 3.3.: [Python] Conjunto de Julia relleno  $K_{-0.39-0.58i}$  ( El disco de Siegel) (i) Algoritmo en blanco y negro (ii) Algoritmo de iteración inversa

El teorema 3.3 lista una serie de características propias de los conjuntos de Julia. Resulta interesante ampliar estas particularidades con propiedades topológicas. Para ello, comenzaremos con algunas nociones de conexión.

**Definición 3.4.** Un subconjunto  $B$  de  $\mathbb{C}$  se dice que es **arcoconexo** si dados dos puntos  $z_1$  y  $z_2$  en  $B$ , existe una función continua  $f: I = [a, b] \rightarrow B$  tal que  $f(a) = z_1$ ,  $f(b) = z_2$ . Se dice que  $f$  es un **arco** o **camino** de  $z_1$  a  $z_2$ .

**Definición 3.5.** Se dice que un subconjunto  $B$  de  $\mathbb{C}$  es **conexo** si no se puede descomponer como unión de dos abiertos disjuntos y no vacíos de  $\mathbb{C}$

**Proposición 3.1.** *Un conjunto arcoconexo es necesariamente conexo. El recíproco es falso.*

**Lema 3.1.** Sea  $\Gamma$  una curva simple, cerrada y suave de  $\mathbb{C}$ . Entonces,

$$f_c^{-1}(\Gamma) = \{w \in \mathbb{C} : f_c(w) \in \Gamma\}$$

tiene las siguientes propiedades:

- (i) Si  $c$  está en el interior de  $\Gamma$ , entonces  $f_c^{-1}(\Gamma)$  es también una curva simple, cerrada y suave. Existe una correspondencia uno a uno entre el interior de  $f_c^{-1}(\Gamma)$  y el interior de  $\Gamma$ .
- (ii) Si  $c$  está en  $\Gamma$ , entonces  $f_c^{-1}(\Gamma)$  es una curva suave en forma de ocho. Existe una correspondencia uno a uno entre el interior de cada una de las dos hojas y el interior de  $\Gamma$ .
- (iii) Si  $c$  está en el exterior de  $\Gamma$ , entonces  $f_c^{-1}(\Gamma)$  consta de dos curvas suaves y cerradas. Existe una correspondencia uno a uno entre el interior de cada una y el interior de  $\Gamma$ .

*Demostración.* La demostración es muy técnica y requiere grandes conocimientos en análisis complejo. A continuación, se probará el lema cuando  $\Gamma$  sea una circunferencia. Sea  $z \in \Gamma$  y  $z - c = |z - c|e^{i\theta}$ . Entonces,  $w = \pm\sqrt{z - c} = \pm\sqrt{|z - c|}e^{i\theta/2}$

- (i) Si  $c$  está en el interior de  $\Gamma$ , entonces el mapa  $z - c$  traslada el centro de la circunferencia del origen a un punto del interior de  $\Gamma$ .
- (ii) Si  $c$  está en  $\Gamma$ , entonces el mapa  $z - c$  traslada el centro de la circunferencia del origen a un punto de la circunferencia, denotado  $-c$ , y por tanto, tenemos una circunferencia  $\bar{\Gamma}$  que pasa por el origen.
- (iii) Si  $c$  está en el exterior de  $\Gamma$ , entonces el mapa  $z - c$  traslada  $\Gamma$  a una circunferencia  $\bar{\Gamma}$  cuyo origen está en el exterior.

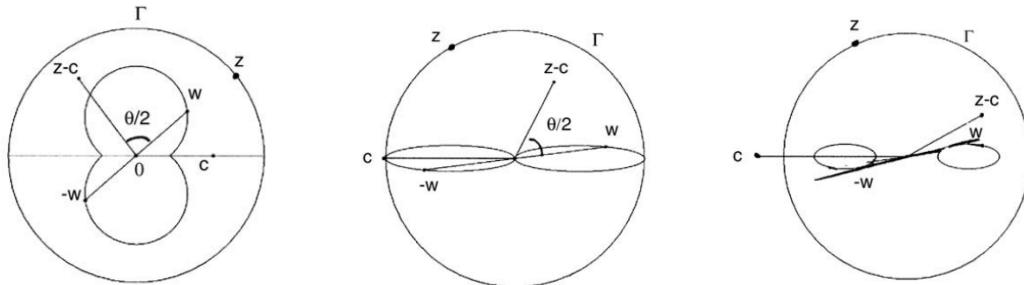


Figura 3.4.: [Elaoo]  $f_c^{-1}(\Gamma)$  cuando: (i)  $c$  es un punto interior de  $\Gamma$  (ii)  $c$  está en  $\Gamma$  (iii)  $c$  es un punto exterior de  $\Gamma$

□

**Teorema 3.4.** Las siguientes afirmaciones sostienen que:

- (i) Si  $0 \in K_c$  entonces el conjunto de Julia  $J_c$  es **arcoc conexo**
- (ii) Si  $0 \notin K_c$  entonces el conjunto de Julia  $J_c$  es **disconexo**

### 3. Sistemas dinámicos complejos

*Demostración.* (i) Supongamos que la órbita  $O^+(0)$  está acotada. Sea  $\Gamma_0$  la circunferencia que encierra a  $O^+(0)$  de tal forma que los puntos del exterior de  $\Gamma_0$  escapan al  $\infty$ . Como  $f_c(0) = c$ ,  $c$  está en el interior de  $\Gamma_0$ . Por el lema 3.1,  $\Gamma_{-1} = f_c^{-1}(\Gamma_0)$  es una curva simple y cerrada que yace en el interior de  $\Gamma_0$ . Se observa que  $f_c(c) = f_c^2(0) \in \Gamma_0$  por hipótesis. Por tanto,  $c = f_c^{-1}$ ,  $f_c(c) \in f_c^{-1}(\Gamma_0) = \Gamma_{-1}$ , y  $c$  está en el interior de  $\Gamma_{-1}$ . Luego  $\Gamma_{-2} = f_c^{-1}(\Gamma_{-1})$  es una curva simple y cerrada que yace en el interior de  $\Gamma_{-1}$ . Continuando el proceso, se crea la sucesión  $\{\Gamma_n\}$  de curvas simples y cerradas, donde  $\Gamma_{-n-1}$  está en el interior de  $\Gamma_{-n}$  y  $c \in \Gamma_{-n}$  para todo  $n \in \mathbb{Z}^+$ . Sea  $L_{-n}$  el cierre de la región acotada por  $\Gamma_{-n}$ . Como consecuencia cada  $L_{-n}$  es cerrado, acotado y arcoconexo. Es fácil ver que:

$$K_c = \bigcap_{n=0}^{\infty} L_{-n} \quad \text{es el conjunto de Julia relleno de } f_c$$

Se sigue diciendo que  $J_c$  es arcoconexo.

(ii) Supongamos que la órbita  $O^+(0)$  no está acotada. Sea  $\Gamma_0$  una circunferencia tal que  $\Gamma_{-1} = f_c^{-1}(\Gamma_0)$  yace en el interior de  $\Gamma_0$  y los puntos fuera de  $\Gamma_{-1}$  escapan al  $\infty$ . Es más, existe  $N \in \mathbb{Z}^+$  tal que  $f_c^N(0) \in \Gamma_0$ ,  $f_c^r(0)$  está en el interior de  $\Gamma_0$  para  $0 \leq r < N$  y  $f_c^r(0)$  está fuera de  $\Gamma_0$  para  $r > N$ . Al igual que en el primer apartado, sea  $\Gamma_{-n-1} = f_c^{-1}(\Gamma_{-n})$ . Para  $n < n_0$ ,  $c$  está en el interior de  $\Gamma_{-n+1}$  y por tanto  $\Gamma_{-n}$  es una curva cerrada y simple del interior de  $\Gamma_{-n+1}$ . Cuando  $n = n_0$ ,  $c \in \Gamma_{-n_0+1}$  y por el lema 3.1,  $\Gamma_{-n_0}$  es una curva en forma de ocho. Ahora, el conjunto de Julia se divide en dos subconjuntos, cada uno de los cuales contenido en una región. Finalmente, podemos asegurar que  $J_c$  es desconexo. □

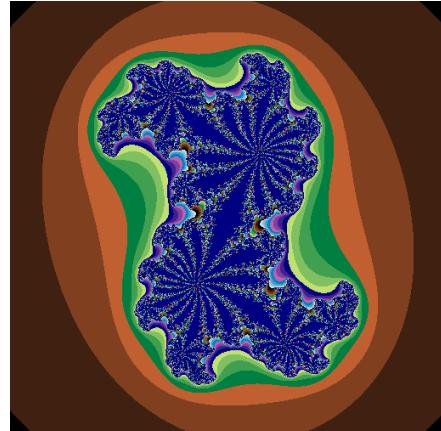
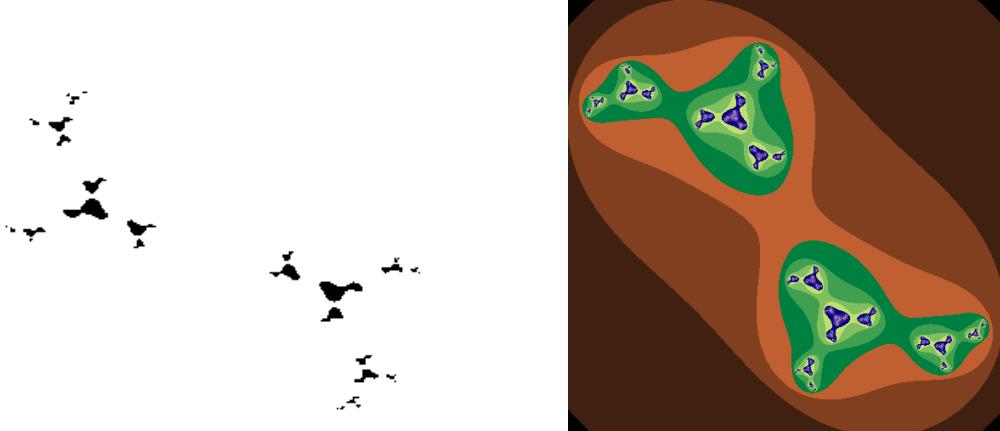


Figura 3.5.: [Python] Conjunto de Julia relleno conexo  $K_{0.377-0.248i}$

Figura 3.6.: [Python] Conjunto de Julia relleno disconexo  $K_{-1.2i}$ 

### 3.2. El conjunto de Mandelbrot

En 1979, **Benoît Mandelbrot** (1924-2010) ideó una forma de distinguir los conjuntos de Julia conexos de aquellos totalmente disconexos. Usó el hecho de que  $J_c$  es conexo si y solo si la órbita de 0 generada por  $f_c$  está acotada. De esta manera, no es difícil trazar el famoso **conjunto de Mandelbrot**.

**Definición 3.6.** El **conjunto de Mandelbrot** consta de todos los valores de  $c \in \mathbb{C}$  para los que la órbita de 0 bajo  $f_c$  está acotada, es decir, no escapa al infinito.

$$M = \{c \in \mathbb{C} : O^+(0) \text{ acotada bajo } f_c\}$$

Como consecuencia inmediata del teorema 3.4, se establece el siguiente **principio de dicotomía**:

**Corolario 3.1.** Para el mapa cuadrático  $f_c$ , se tiene que:

- (i)  $c \in M$  y el conjunto de Julia correspondiente  $J_c$  es **arcoc conexo**.
- (ii)  $c \notin M$  y el conjunto de Julia correspondiente  $J_c$  es **disconexo**.

El corolario 3.1 proporciona una caracterización del conjunto de Mandelbrot:

$$M = \{c \in \mathbb{C} : J_c \text{ es arcoc conexo}\}$$

Ahora, vamos a dar un criterio por el cual un parámetro  $c$  pertenece a  $M$ . Para ello, es conveniente recordar el concepto de punto crítico.

**Definición 3.7.** Se dice que  $z_0 \in \mathbb{C}$  es un punto crítico del mapa  $f \in C^1(\mathbb{C})$  si  $f'(z_0) = 0$

**Proposición 3.2.** El único punto crítico del mapa cuadrático  $f_c$  es 0.

**Teorema 3.5.** Si  $z_0$  es un punto atractivo del mapa complejo  $f$ , entonces  $z_0$  atrae al menos un punto crítico de  $f$ .

**Corolario 3.2.** Si  $f_c$  tiene un punto periódico atractivo entonces  $c \in M$ .

### 3. Sistemas dinámicos complejos

La siguiente tarea consiste en desarrollar un algoritmo para dibujar el conjunto de Mandelbrot con el fin de estudiar su comportamiento. Empezamos con este importante resultado.

**Teorema 3.6.** *El conjunto de Mandelbrot  $M$  está contenido en el disco cerrado de radio 2:*

$$M \subset \{c \in \mathbb{C} : |c| \leq 2\}$$

*Demostración.* Supongamos que  $|c| > 2$ . Entonces, si  $|z| \geq |c|$  se tiene que:

$$|f_c(z)| = |z^2 + c| = |z| \left| z + \frac{c}{z} \right| \geq |z| \left( |z| - \frac{|c|}{|z|} \right) \geq |z|(|c| - 1) = \eta |z|$$

donde  $\eta = |c| - 1 > 1$ .

Seguidamente, vemos que  $f_c(0) = c$ ,  $|f_c^2(0)| = |f_c(c)| \geq |c|\eta$ . Por inducción se puede comprobar que  $|f_c^{n+1}(0)| \geq |c|\eta^n$ . Por tanto, supongamos que  $|f_c^n(0)| \geq |c|\eta^{n-1}$ . Entonces,

$$|f_c^{n+1}(0)| = |f_c(f_c^n(0))| \geq |f_c^n(0)|\eta \geq |c|\eta^n$$

Se sigue con que  $|f_c^{n+1}(0)| \geq |c|\eta^n \rightarrow \infty$  cuando  $n \rightarrow \infty$ . Consecuentemente,  $c \notin M$ , y la demostración queda completada.  $\square$

El teorema 3.6 se emplea para escribir el algoritmo con el cual esbozaremos el conjunto de Mandelbrot  $M$ .

#### Algoritmo de píxeles para el conjunto de Mandelbrot $M$

- (i) Se elige un número de iteraciones máximo  $N$ .
- (ii) Se toma un subconjunto de  $\mathbb{C}$  que contenga al disco cerrado de radio 2:

$$\{c \in \mathbb{C} : |c| \leq 2\}$$

- (iii) Para cada  $c$  del subconjunto, calculamos las primeras  $N$  iteraciones de 0:

$$\{f_c(0), f_c^2(0), \dots, f_c^N(0)\}$$

- (iv) Si en alguna iteración  $k \leq N$ , con  $k \in \mathbb{Z}^+$ ,  $f_c^k(0)$  sale del disco, dejamos de iterar y coloreamos  $c$  de blanco, en otro caso coloreamos  $c$  de negro.

La región negra correspondiente al conjunto de Mandelbrot se aprecia en la figura 3.7.

Para obtener una representación más colorida del conjunto de Mandelbrot, se puede llevar a cabo una modificación del algoritmo. La idea es similar a la ya realizada para el conjunto de Julia. El color que utilizaremos para pintar el punto  $c$  dependerá de la proporción entre el número de iteraciones  $k$  que se han necesitado para escapar del disco de radio 2 y el número de iteraciones máximo  $N$  (Figura 3.8).

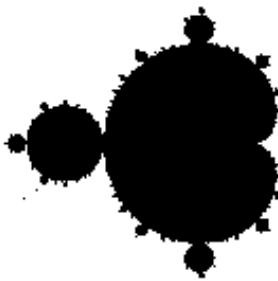


Figura 3.7.: [Python] Conjunto de Mandelbrot en blanco y negro

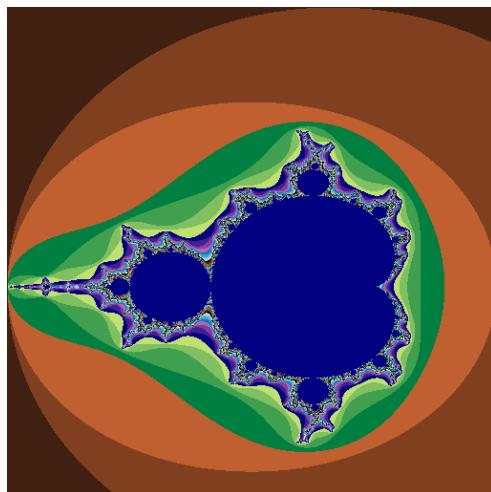


Figura 3.8.: [Python] Conjunto de Mandelbrot

### 3.2.1. Límites de las órbitas periódicas. Bulbos del conjunto de Mandelbrot

Una vez representado el conjunto de Mandelbrot, nos disponemos a analizar sus órbitas periódicas con más detalle. En primer lugar, empezaremos por la gran **cardioide** que forma parte del eje principal de la gráfica.

**Proposición 3.3.**  $f_c$  tiene un punto fijo atractivo si y solo si  $c$  pertenece a la **cardioide** de ecuación  $z = \frac{1}{2}e^{i\theta}(1 - \frac{1}{2}e^{i\theta})$ ,  $0 \leq \theta \leq 2\pi$ .

*Demostración.* Si  $z$  es un punto fijo atractivo de  $f_c$ , se verifica que  $f_c(z) = z^2 + c = z$  y  $|f'_c(z)| = |2z| < 1$ . La frontera de esta región está formada por aquellos puntos que cumplen que  $|z| = \frac{1}{2}$  o  $z = \frac{1}{2}e^{i\theta}$ ,  $0 \leq \theta \leq 2\pi$ . Resolviendo simultáneamente estas ecuaciones

### 3. Sistemas dinámicos complejos

obtenemos para la frontera la ecuación:

$$z = \frac{1}{2}e^{i\theta} - \frac{1}{4}e^{2i\theta} \quad 0 \leq \theta \leq 2\pi$$

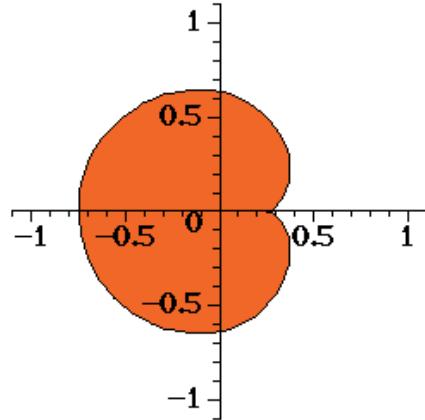


Figura 3.9.: [AGo2] Límite de los puntos fijos de M

□

**Proposición 3.4.** Los conjuntos de Julia  $J_c$  asociados a los parámetros  $c$  que satisfacen la proposición 3.3 son curvas cerradas y simples (Figura 3.10).

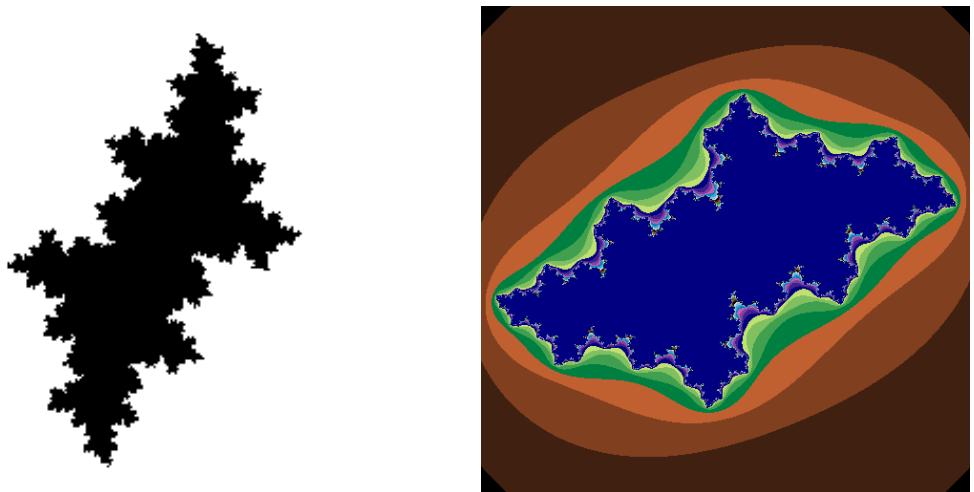


Figura 3.10.: [Python] Conjunto de Julia relleno  $K_{-0.5+0.5i}$ .  $J_{-0.5+0.5i}$  es una curva cerrada y simple

Para los valores de  $c$  que no están en el interior de la cardioide,  $f_c$  no tiene ningun punto fijo atractivo. Vamos a ver si existen valores de  $c$  para los que  $f_c$  tiene un 2-ciclo atractivo.

**Proposición 3.5.**  $f_c$  tiene un 2-ciclo atractivo si y solo si  $|c + 1| < \frac{1}{4}$

*Demostración.* Usando argumentos similares a los anteriores, no es difícil extender el análisis para determinar el límite de los puntos periódicos de periodo dos. Si  $z$  es un punto 2-periódico, será solución de

$$f_c^2(z) = (z^2 + c)^2 + c = z$$

o equivalentemente

$$z^4 + 2cz^2 - z + c^2 + c = 0$$

Como los puntos fijos son solución de la ecuación se tiene que:

$$(z^2 - z + c)(z^2 + z + c + 1) = 0$$

Por tanto, los puntos de periodo dos satisfacen:

$$z^2 + z + c + 1 = 0$$

que tiene como raíces:

$$z_{2,1} = \frac{-1 + i\sqrt{3 + 4c}}{2} \quad z_{2,2} = \frac{-1 - i\sqrt{3 + 4c}}{2}$$

La estabilidad de cada uno de ellos viene determinada por la derivada:

$$f_c'^2(z) = 4z^3 + 4cz = 4z(z^2 + c)$$

Finalmente, como  $f_c'^2(z_{2,1}) = f_c'^2(z_{2,2}) = 4(1 + c)$ , por el teorema 3.1 se tiene que  $f_c$  tiene un 2-ciclo atractivo si y solo si  $|c + 1| < \frac{1}{4}$ .  $\square$

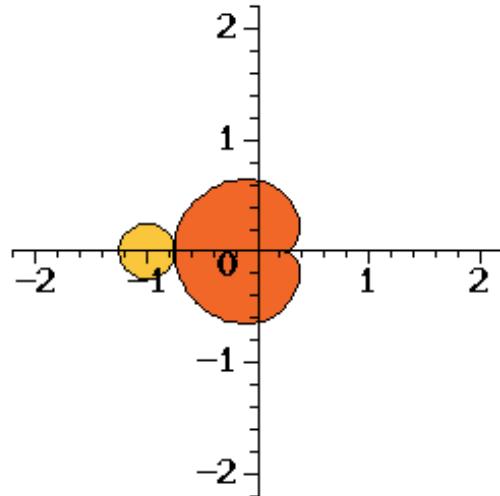


Figura 3.11.: [AGo2] Límite de los puntos periódicos de periodo dos de M

### 3. Sistemas dinámicos complejos

**Proposición 3.6.** Los conjuntos de Julia  $J_c$  asociados a los parámetros  $c$  que satisfacen la proposición 3.5 ya no son curvas simples. Son uniones de una cantidad numerable de curvas cerradas, simples y disjuntas 3 a 3 (Figura 3.12).

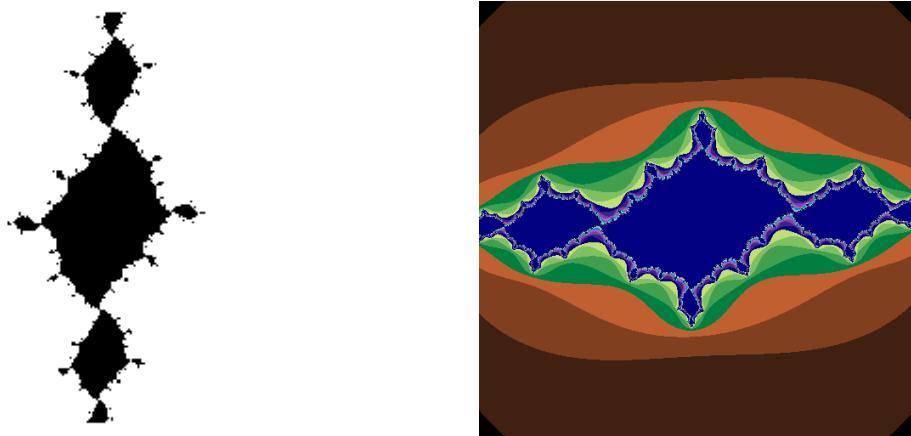


Figura 3.12.: [Python] Conjunto de Julia relleno  $K_{-1.1+0.1i} \cdot J_{-1.1+0.1i}$  ya no es una curva simple.

Siguiendo este proceso, habrá una región de valores de  $c$  para los que exista un ciclo de orden 3 atractivo. Para estos valores, los conjuntos de Julia son uniones de una cantidad numerable de curvas cerradas, simples, disjuntas 4 a 4 y con la propiedad de que donde se tocan 2 se toca también una tercera (Figura 3.14). Análogamente, existen 3 circunferencias y 3 cardioïdes de menor tamaño para los que la dinámica presenta un ciclo de orden 4 atractivo y así sucesivamente. Se obtiene así la figura 3.13 (las cardioïdes de menor tamaño no están en contacto con el cardioide principal, y son tan pequeñas que apenas se ven).

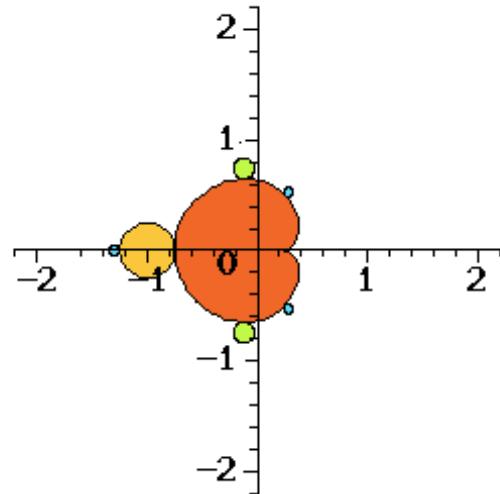


Figura 3.13.: [AGo2] Límite de los puntos periódicos de periodo tres y cuatro de  $M$

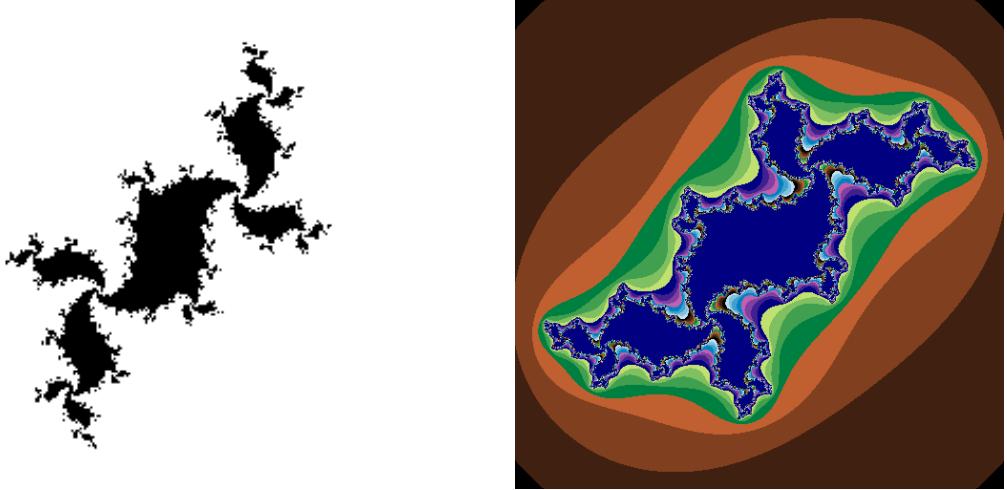


Figura 3.14.: [Python] Conjunto de Julia relleno  $K_{-0,2+0.75i}$ , tiene un 3-ciclo atractivo.

**Teorema 3.7.** *El conjunto de Mandelbrot  $M$  es conexo*

*Demostración.* La demostración se basa en la idea de que el conjunto de Mandelbrot se puede ver como un subconjunto de la esfera de Riemann  $\bar{\mathbb{C}} = \mathbb{C} \cup \infty$  usando la **proyección estereográfica**.

El conjunto de Mandelbrot es simplemente conexo cuando su complementario (en  $\bar{\mathbb{C}}$ ) sea simplemente conexo. El **teorema de representación conforme de Riemann** asegura que un conjunto simplemente conexo diferente al plano complejo es homeomorfo al disco unidad.

Hubbard y Douday mostraron que existe un mapa del disco unidad  $\Delta = \{z : |z| \leq 1\}$  al complementario de  $M$  (en la esfera de Riemann). Como consecuencia, el complementario de  $M$  es simplemente conexo. Usaron el siguiente mapa:

$$\Psi: \Delta \rightarrow \bar{\mathbb{C}}$$

$$\Psi(z) = \frac{1}{z} + b_0 + b_1 z + b_2 z^2 + \dots$$

El mapa  $\Psi$  ofrece una nueva forma de dibujar el conjunto de Mandelbrot, diferente del algoritmo de píxeles. La primera parte del mapa  $\Psi$ ,  $z \rightarrow \frac{1}{z}$  mapea el interior del disco de  $\Delta$  sobre el exterior, invirtiendo la orientación y enviando el origen al  $\infty$ . La serie de potencias  $b_0 + b_1 z + b_2 z^2 + \dots$  añade una pequeña distorsión para hacer la imagen de  $\Psi$  el exterior de  $M$  en lugar del exterior de  $\Delta$ . La imagen bajo  $\Psi$  de un círculo  $|z| = r$ ,  $r < 1$ , es una curva cerrada simple que delimita una región  $M_r$ , que contiene a  $M$  (Figura 3.15).  $\square$

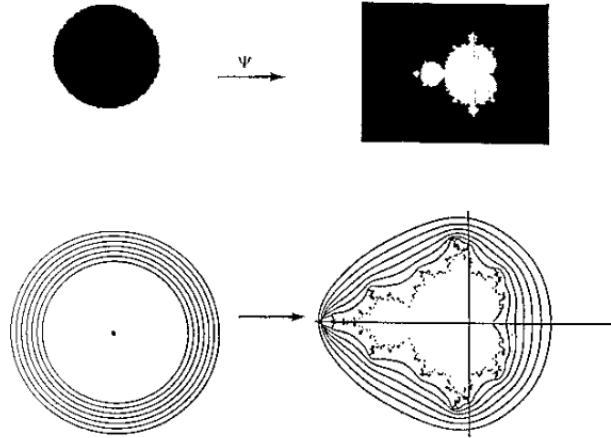


Figura 3.15.: [Elaoo] Mapeo del disco al exterior de  $M$

### 3.3. Aplicación de la teoría de conjuntos de Julia al método de Newton

En análisis numérico, el **método de Newton** o **método de Newton-Rapshon** se utiliza para encontrar las raíces de una ecuación.

**Definición 3.8.** Sea  $p: \mathbb{C} \rightarrow \mathbb{C}$  un polinomio con coeficientes complejos y  $f: \mathbb{C} \cup \{\infty\} \rightarrow \mathbb{C} \cup \{\infty\}$ . Se tiene que:

$$f(z) = z - \frac{p(z)}{p'(z)} \quad (3.2)$$

**Proposición 3.7.** Sea  $f$  una función definida con la fórmula de Newton (3.2). Las iteradas  $f^k(z)$  convergen a la solución de la ecuación  $p(z) = 0$  siempre que se escoja el valor inicial apropiadamente y  $p'(z) \neq 0$ .

Así pues, los puntos fijos de  $f$ , dados por  $\frac{p(z)}{p'(z)} = 0$ , son los ceros de  $p$  junto con  $\infty$ . Derivando, encontramos que:

$$f'(z) = \frac{p(z)p''(z)}{p'(z)^2}$$

por ello, un cero  $z$  de  $p$  es un punto fijo superatractivo de  $f$ , siempre que  $p'(z) \neq 0$ .

Denotaremos

$$A(w) = \left\{ z \in \mathbb{C} \cup \{\infty\} : f^k(z) \rightarrow w \right\}$$

la cuenca de atracción del cero de la ecuación, en otras palabras, el conjunto de valores iniciales que convergen a  $w$  bajo las iteraciones de Newton. Como los ceros son atractores, la cuenca de atracción  $A(w)$  incluye un abierto que contiene a  $w$ .  $J(f)$  es el límite de la cuenca de atracción  $A(w)$  para cada punto fijo atractivo  $w$ . Por lo tanto, se considera que  $J(f)$  es bastante importante en el momento de analizar el dominio de atracción de las raíces del método de Newton.

### 3.3. Aplicación de la teoría de conjuntos de Julia al método de Newton

**Ejemplo 3.1.** Un ejemplo sencillo es el polinomio cuadrático:

$$p(z) = z^2 - c$$

cuyos ceros son  $\pm\sqrt{c}$ . Usando la fórmula de Newton (3.2) obtenemos:

$$f(z) = \frac{z^2 + c}{2z}$$

donde

$$f(z) \pm \sqrt{c} = \frac{(z \pm \sqrt{c})^2}{2z}$$

$$\frac{f(z) + \sqrt{c}}{f(z) - \sqrt{c}} = \left( \frac{z + \sqrt{c}}{z - \sqrt{c}} \right)^2$$

En seguida vemos que si  $\frac{|z+\sqrt{c}|}{|z-\sqrt{c}|} < 1$  entonces  $\frac{|f^k(z)+\sqrt{c}|}{|f^k(z)-\sqrt{c}|} \rightarrow 0$  y  $f^k(z) \rightarrow -\sqrt{c}$  cuando  $k \rightarrow \infty$ , análogamente si  $\frac{|z+\sqrt{c}|}{|z-\sqrt{c}|} > 1$  entonces  $f^k(z) \rightarrow \sqrt{c}$  cuando  $k \rightarrow \infty$ .

El conjunto de Julia  $J(f)$  es la recta  $|z + \sqrt{c}| = |z - \sqrt{c}|$  (la bisectriz perpendicular a  $-\sqrt{c}$  y  $\sqrt{c}$ ) y  $A(-\sqrt{c})$ ,  $A(\sqrt{c})$  son los semiplanos a cada lado. (Para el caso  $c = 1$ , vea la figura 3.16)

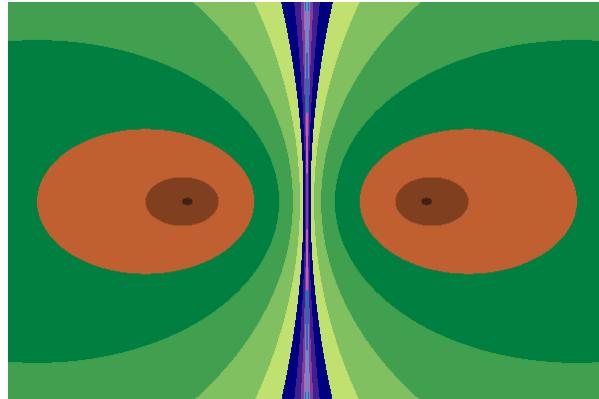


Figura 3.16.: [Python] Conjunto de Julia asociado al método de Newton para la función compleja  $p(z) = z^2 - 1$

El ejemplo cuadrático 3.1 podría llevarnos a pensar que los dominios de atracción bajo la iteración de Newton de los ceros de cualquier polinomio son razonablemente regulares. Sin embargo, para ordenes superiores la situación es muy diferente. Sin profundizar en formalismos teóricos, veamos que ocurre en el siguiente ejemplo.

**Ejemplo 3.2.** El polinomio de grado tres

$$p(z) = z^3 - 1$$

### 3. Sistemas dinámicos complejos

tiene como ceros a  $1, e^{i\frac{2\pi}{3}}, e^{i\frac{4\pi}{3}}$  y la función de Newton:

$$f(z) = \frac{2z^3 + 1}{3z^2}$$

La transformación  $p(z) = ze^{i\frac{2\pi}{3}}$  es una rotación de  $120^\circ$  respecto al origen. Es fácil comprobar que  $f(p(z)) = p(f(z))$ . Esta rotación mapea la cuenca de atracción  $A(w)$  sobre  $A(we^{i\frac{2\pi}{3}})$  por cada uno de los tres ceros  $w$ , por tanto el conjunto de Julia tiene triple simetría respecto al origen. Si  $z \in \mathbb{R}$  entonces  $f^k(z) \in \mathbb{R}$  para todo  $k$  y  $f^k(z)$  converge a 1 excepto para un número contable de reales  $z$ . Como consecuencia  $A(1)$  contiene el eje real salvo un número contable de puntos. Por simetría,  $A(e^{i\frac{2\pi}{3}})$  y  $A(e^{i\frac{4\pi}{3}})$  contienen las rectas que pasan por el origen con un ángulo de  $120^\circ$  y  $240^\circ$  (respectivamente) salvo un número contable de puntos. También se conoce que  $A(w)$  contiene un entorno de  $w$ , de tal forma que cualquier punto de la frontera de uno de los  $A(w)$  está en la frontera de los tres, con incontables puntos triples.

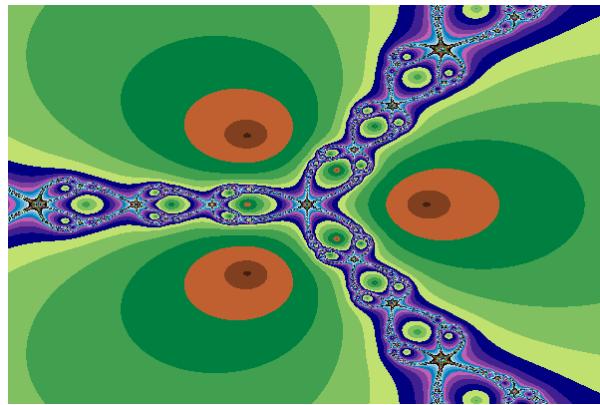


Figura 3.17.: [Python] Conjunto de Julia asociado al método de Newton para la función compleja  $p(z) = z^3 - 1$

Sin entrar en mucho detalle, se puede apreciar que cuando aumentamos el grado del polinomio  $p(z) = z^n - 1$ , se observan ciertos patrones de comportamiento.

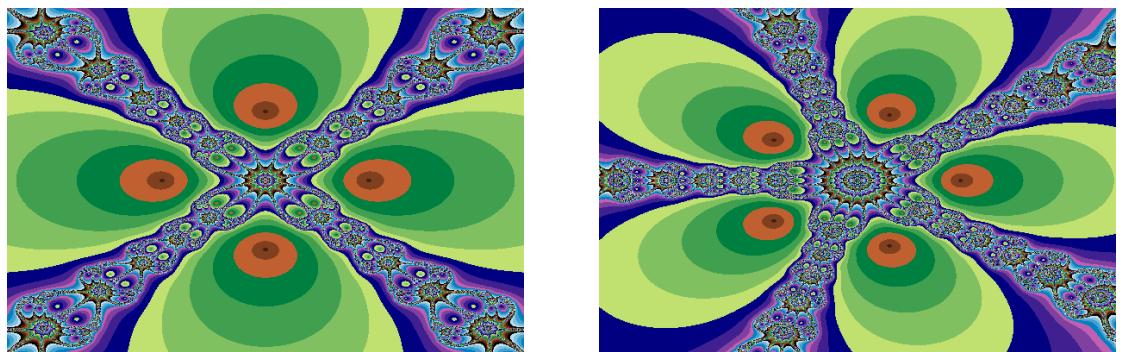


Figura 3.18.: [Python] Conjunto de Julia asociado al método de Newton para las funciones complejas: (i)  $p(z) = z^4 - 1$  (ii)  $p(z) = z^5 - 1$

## 4. Fractales

Después de indagar acerca de los conjuntos de Julia y de Mandelbrot, estamos listos para adentrarnos en el mundo de los fractales. En el capítulo 3 se hizo un análisis de estos conjuntos, sin embargo, aún quedan muchas preguntas por responder. Llegados a este punto, es natural cuestionarse: ¿Qué hay detrás de los conjuntos de Julia y de Mandelbrot?, ¿Con qué objetos geométricos podemos identificarlos?, ... Este capítulo tiene como objetivos definir lo que es un fractal, construir y dar ejemplos de fractales clásicos e introducir los términos de sistema de Lindenmayer (sistema-L) y sistema de funciones iteradas (SFI).

Se usarán los libros [Lyn18], [HOPo4], [Elaoo], [Bar88] y [Man75] para desarrollar los conceptos de este capítulo.

En 1975, **Benoît B. Mandelbrot** introdujo el término fractal en su obra [Man75]. La palabra **fractal** proviene del latín *fractus* que significa fracturado. En líneas generales, se puede dar una definición aproximada de lo que es un fractal.

**Definición 4.1.** Un fractal es un objeto geométrico cuya estructura muestra **autosimilitud** (a veces llamada autosemejanza o autosimilaridad) y que se construye a partir de un patrón o imagen repetida a diferentes escalas.

En 1982, Mandelbrot publicó su nuevo libro [Man82] en el que se podían encontrar ostentosos gráficos creados con la tecnología informática de aquel entonces. En este libro, propone una definición de fractal más rigurosa .

**Definición 4.2.** Un fractal es un subconjunto de  $\mathbb{R}^n$  cuya dimensión de Haussdorff es estrechamente mayor que la dimensión topológica.

La dimensión de Hausdorff o dimensión de Hausdorff-Besicovith es una generalización de la dimensión topológica que define la dimensión fractal como un número no entero. Profundizaremos más en el término en el capítulo 5.

Después de décadas, se han propuesto otras definiciones que también se aproximan, pero hasta el momento no existe ninguna otra que sea más aceptada.

Los objetos que hoy denominamos fractales eran conocidos en las matemáticas desde principios del siglo XX. Su definición fue una gran contribución, ya que en el siglo XIX algunos matemáticos estudiaban objetos cuyas propiedades no se explicaban con la geometría tradicional. Los fractales han generado gran interés desde la llegada de la computadora.

Los conjuntos de Julia y de Mandelbrot tienen estructura fractal. Muchos objetos de la naturaleza muestran autosimilitud a diferentes escalas; por ejemplo: coliflor, helechos, árboles, nubes e incluso las redes de vasos sanguíneos de nuestro propio cuerpo. El análisis fractal se está aplicando en muchas ramas de la ciencia , por ejemplo, gráficos, compresión

#### 4. Fractales

de imágenes por computadora, extracción de petróleo de las rocas, meteorología, ...

*La geometría fractal cambiará a fondo su visión de las cosas. Seguir leyendo es peligroso. Se arriesga a perder definitivamente la imagen inofensiva que tienen las nubes, bosques, galaxias, hojas, plumas, flores, rocas, montañas y muchas otras cosas. Jamás volverá a recuperar las interpretaciones de todos estos objetos que hasta ahora le eran familiares.*

Michael Barnsley

Podríamos seguir dando argumentos por los que la geometría fractal está presente en nuestras vidas y razones por las que es necesario su estudio. No obstante, es mucho más práctico presentar unos cuantos ejemplos.

### 4.1. Ejemplos de fractales clásicos

En esta sección, examinaremos diferentes ejemplos de fractales geométricos, que se obtienen a partir de un proceso de extracción de piezas (segmentos, triángulos, cuadrados,...), partiendo de una inicial. Construiremos paso a paso los patrones básicos de la estructura fractal. En cada uno de ellos, se observará la propiedad de autosimilitud tan característica de los fractales.

#### 4.1.1. El conjunto de Cantor

El **conjunto de Cantor** fue considerado por primera vez por el matemático **Georg Cantor** (1845 – 1918) en 1883. Mandelbrot lo apodó el **polvo de Cantor** debido a su apariencia.

Se construye a partir de un intervalo cerrado  $C_0$  mediante la eliminación sucesiva de una familia de subintervalos. Normalmente, el conjunto inicial es  $C_0 = [0, 1]$ . En la siguiente etapa, obtenemos  $C_1$  eliminando el tercio medio abierto  $(\frac{1}{3}, \frac{2}{3})$  del conjunto inicial  $C_0$ :

$$C_1 = C_0 \setminus \left( \frac{1}{3}, \frac{2}{3} \right) = \left[ 0, \frac{1}{3} \right] \cup \left[ \frac{2}{3}, 1 \right]$$

El siguiente conjunto  $C_2$  se obtiene a partir de  $C_1$  eliminando el tercio medio abierto de cada uno de sus dos intervalos  $[0, \frac{1}{3}]$  y  $[\frac{2}{3}, 1]$ . Por tanto:

$$C_2 = \left[ 0, \frac{1}{9} \right] \cup \left[ \frac{2}{9}, \frac{3}{9} \right] \cup \left[ \frac{6}{9}, \frac{7}{9} \right] \cup \left[ \frac{8}{9}, 1 \right]$$

Se puede observar que, en realidad,  $C_1$  es la unión de dos copias de  $C_0$  mediante una homotecia de razón  $\frac{1}{3}$ . Análogamente,  $C_2$  es la unión de dos copias de  $C_1$  mediante una homotecia de razón  $\frac{1}{3}$ .

Continuando con este proceso hasta el  $k$ -ésimo paso,  $C_k$  está formado por la unión de  $2^k$  intervalos disjuntos y cerrados obtenidos a partir de  $C_{k-1}$  eliminando el tercio medio abierto de cada uno de los  $2^{k-1}$  intervalos (Figura 4.1). De nuevo,  $C_k$  es la unión de dos copias de

$C_{k-1}$  obtenidas mediante una homotecia de razón  $\frac{1}{3}$ . Claramente,  $C_k$  es un conjunto autosimilar exacto, es decir, es idéntico a una copia escalada de sí mismo.

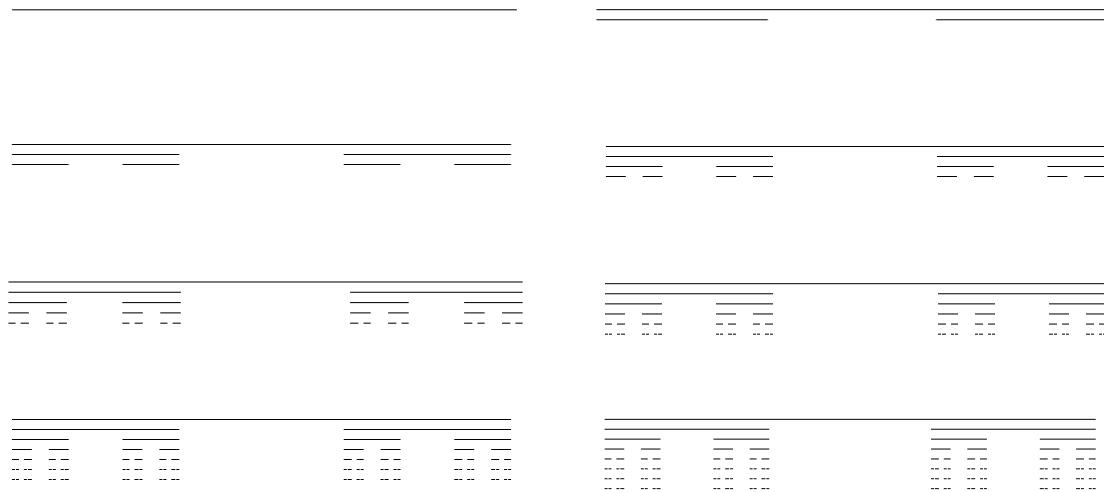


Figura 4.1.: [Python] Etapas de la construcción del conjunto de Cantor: (i)  $C_0$  (ii)  $C_1$  (iii)  $C_2$  (iv)  $C_3$  (v)  $C_4$  (vi)  $C_5$  (vii)  $C_6$  (viii)  $C_7$

Es obvio que con el procedimiento anterior obtenemos una sucesión de intervalos  $C_0 \supset C_1 \supset C_2 \supset C_3 \supset \dots$  En consecuencia, el límite del conjunto de Cantor, está formado por la intersección de los puntos que quedan al repetir el proceso infinitas veces.

**Definición 4.3.** El conjunto de cantor se define como:

$$C = \bigcap_{k=1}^{\infty} C_k$$

Como era de esperar,  $C$  es no vacío ya que contiene los extremos de cada uno de los subintervalos que lo forman. Además,  $C$  es cerrado por ser intersección de conjuntos cerrados y está acotado ya que es un subconjunto del intervalo  $[0, 1]$ .

**Proposición 4.1.** La longitud del conjunto de Cantor es 0.

*Demostración.* Como ya se ha comentado, cada subconjunto  $C_k$  se divide en  $2^k$  intervalos disjuntos y cerrados, cada uno de ellos con longitud  $(\frac{1}{3})^k$ . La longitud total de  $C_k$  es la suma de las longitudes de todos estos subintervalos, es decir,  $(\frac{2}{3})^k$ . Por consiguiente, el límite de la longitud de  $C_k$  es:

$$\lim_{k \rightarrow \infty} \left(\frac{2}{3}\right)^k = 0$$

□

Una consecuencia inmediata de la proposición 4.1 es que el conjunto de Cantor es de medida nula. En particular, no puede contener ningún intervalo de medida no nula.

El conjunto  $C$  consta de un número infinito de puntos de acumulación. Utilizando un sistema numérico con términos de expansión en base tres, es posible clasificar que puntos pertenecen al conjunto de Cantor y cuáles no. A modo de recordatorio, las fracciones se pueden expandir en base tres aplicando un algoritmo simple: triplique el numerador de la fracción en cuestión, realice la división, apunte el factor ternario y continúe con el resto.

**Ejemplo 4.1.** En este ejemplo, expresaremos en términos de expansión en base 3 la fracción  $\frac{4}{7}$ :

Fracción actual	Cociente	Resto	$3 \cdot$ Resto
$\frac{4}{7}$	0	4	12
$\frac{12}{7}$	1	5	15
$\frac{15}{7}$	2	1	3
$\frac{3}{7}$	0	3	9
$\frac{9}{7}$	1	2	6
...	...	...	...

Finalmente, tenemos que  $\frac{4}{7} = (0,1201...)_3$

**Proposición 4.2.** Sea  $x \in [0,1]$ . Entonces,  $x \in C$  si y sólo si tiene una expansión ternaria que no contiene 1's.

*Demostración.* Dividimos el intervalo  $[0,1]$  en tres partes iguales, obteniendo:

$$C_0 = \left[0, \frac{1}{3}\right] \cup \left[\frac{1}{3}, \frac{2}{3}\right] \cup \left[\frac{2}{3}, 1\right]$$

Expresándolos en base tres los 1's se encuentran en el intervalo  $[\frac{1}{3}, \frac{2}{3}]$  ya que  $\frac{1}{3} = (0,100...)_3$  y  $\frac{2}{3} = (0,122...)_3$ . Los extremos  $\frac{1}{3}$  y  $\frac{2}{3}$  tienen otras expresiones ternarias totalmente válidas que no contienen 1's,  $\frac{1}{3} = (0,0222...)_3$  y  $\frac{2}{3} = (0,200...)_3$ . Por tanto, eliminamos dicho intervalo obteniendo  $C_1$ . Repetimos el proceso dividiendo el intervalo  $C_1$  en tres partes iguales:

$$C_1 = \left[0, \frac{1}{9}\right] \cup \left[\frac{1}{9}, \frac{2}{9}\right] \cup \left[\frac{2}{9}, \frac{1}{3}\right] \cup \left[\frac{2}{9}, \frac{7}{9}\right] \cup \left[\frac{7}{9}, \frac{8}{9}\right] \cup \left[\frac{8}{9}, 1\right]$$

Expresándolos en base tres, vemos que los 1's se encuentran en los intervalos  $[\frac{1}{9}, \frac{2}{9}]$  y  $[\frac{7}{9}, \frac{8}{9}]$ :

$$\frac{1}{9} = (0,01)_3 \quad \frac{2}{9} = (0,02)_3 \quad \frac{7}{9} = (0,21...)_3 \quad \frac{8}{9} = (0,22)_3$$

Los extremos  $\frac{1}{9}$  y  $\frac{7}{9}$  tienen otras expresiones ternarias que no contienen 1's. Por tanto, eliminamos dichos intervalos, obteniendo el conjunto  $C_2$ . Aplicando el procedimiento infinitas veces, observamos que:

$$C = \bigcap_{k=1}^{\infty} C_k$$

donde  $C$  está formado por los puntos  $x \in C$  que tienen una expansión ternaria que no contiene 1's.  $\square$

### 4.1.2. El triángulo de Sierpinski

En 1915, el matemático polaco Waclaw Sierpiński (1882-1969) presentó el **triángulo de Sierpiński**. Este fractal ha sido considerado como una versión bidimensional del conjunto de Cantor.

Se construye a partir de un triángulo equilátero  $S_0$ , cuyo lado es de longitud 1 (por simplicidad). A continuación, conectamos los tres puntos medios del triángulo  $S_0$ . De esta forma, conseguimos cuatro triángulos equiláteros más pequeños. Para terminar, retiramos el triángulo del medio. Esto nos deja tres triángulos equiláteros de longitud  $\frac{1}{2}$ .

En el próximo paso, repetimos el mismo procedimiento con los tres triángulos restantes para obtener 9 triángulos equiláteros con lados de longitud  $(\frac{1}{2})^2$ . Si continuamos con la construcción, en la  $k$ -ésima etapa tendremos  $3^k$  triángulos equiláteros con lados de longitud  $(\frac{1}{2})^k$  (Figura 4.2).

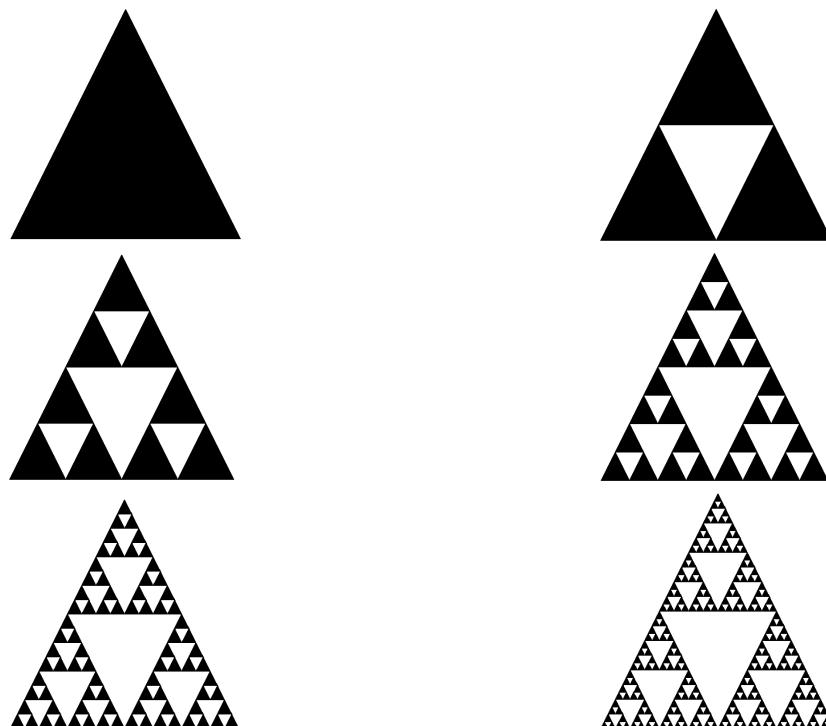


Figura 4.2.: [Python] Etapas de la construcción del triángulo de Sierpiński: (i)  $S_0$  (ii)  $S_1$  (iii)  $S_2$  (iv)  $S_3$  (v)  $S_4$  (vi)  $S_5$

Aplicando el razonamiento anterior, obtenemos una sucesión de conjuntos de la forma  $S_0 \supset S_1 \supset S_2 \supset S_3 \supset \dots$ , tal que, cada nueva iteración  $S_i$  estará incluida en  $S_{i-1}$ . Igual que pasaba con el conjunto de Cantor, el límite del triángulo de Sierpiński está compuesto por la intersección de los triángulos que quedan al repetir el proceso infinitas veces.

**Definición 4.4.** El triángulo de Sierpiński se define como:

$$S = \bigcap_{k=1}^{\infty} S_k$$

**Proposición 4.3.** *El área del triángulo de Sierpiński es cero.*

*Demostración.* Calculemos la suma de las áreas de los triángulos que se han ido eliminando en la construcción del triángulo de Sierpiński. Sea  $A$  el área del triángulo original. En el primer paso, se elimina un triángulo de área  $\frac{1}{4} \cdot A$ , en la segunda etapa se eliminan tres triángulos de área  $(\frac{1}{4})^2 \cdot A$  cada uno. La suma del área de los triángulos eliminados es:

$$\frac{1}{4} \cdot A + 3 \cdot \left(\frac{1}{4}\right)^2 A + 3^2 \cdot \left(\frac{1}{4}\right)^3 A + \dots = A$$

Por tanto, el triángulo de Sierpiński tiene área cero. □

#### 4.1.3. La curva de Koch

La curva de Koch es uno de los fractales más conocidos hasta la fecha. Fue introducido por el matemático sueco **Niels Fabian Helge von Koch** (1870-1924) en 1904 y lleva su nombre.

La curva de Koch  $K$  se construye de diferentes maneras, a continuación, se describirá la más habitual. Se forma a partir de un segmento de longitud 1 (por simplicidad) que recibe el nombre de conjunto inicial  $K_0$ . Para construir  $K_1$ , dividimos el segmento  $K_0$  en tres partes iguales y reemplazamos el tercio medio por otros dos segmentos de la misma longitud formando un ángulo de  $60^\circ$ . En otras palabras, reemplazamos el tercio medio por un triángulo equilátero. En la siguiente etapa, dividimos cada uno de los 4 segmentos que forman  $K_1$  en tres partes iguales y reemplazamos las partes centrales por triángulos equiláteros. En la  $k$ -ésima etapa tendremos  $4^k$  segmentos de longitud  $(\frac{1}{3})^k$  (Figura 4.3).

**Proposición 4.4.** *El perímetro de la curva de Koch es infinito.*

*Demostración.* La longitud de la curva de Koch en la  $k$ -ésima etapa viene dada por la suma de los  $4^k$  segmentos de longitud  $(\frac{1}{3})^k$ , es decir,  $(\frac{4}{3})^k$ . Como:

$$\lim_{k \rightarrow \infty} \left(\frac{4}{3}\right)^k = \infty$$

la curva de Koch consiste en una curva infinitamente larga, es decir, crece indefinidamente conforme  $k \rightarrow \infty$ . □

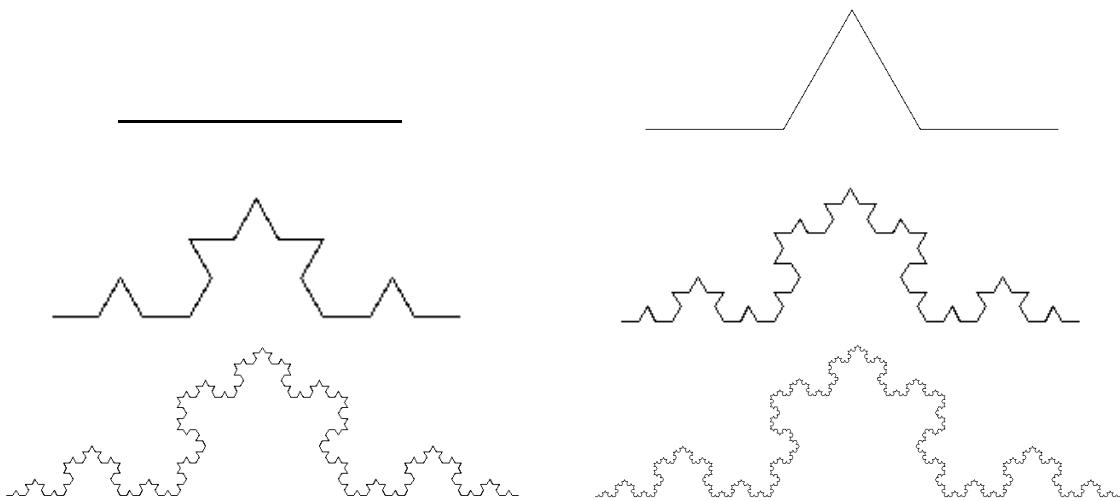


Figura 4.3.: [Python] Etapas de la construcción de la curva de Koch: (i)  $K_0$  (ii)  $K_1$  (iii)  $K_2$  (iv)  $K_3$  (v)  $K_4$  (vi)  $K_5$

Si partimos de un triángulo equilátero y a cada lado le aplicamos el procedimiento para obtener la curva de Koch obtenemos el **copo de nieve de Koch** (Figura 4.4). Visto de otra manera, el copo de nieve está compuesto por la unión de tres curvas de Koch.

**Corolario 4.1.** *El perímetro del copo de nieve de Koch es infinito.*

*Demostración.* Puesto que la curva de Koch tiene perímetro infinito (proposición 4.4). El copo de nieve de Koch es una región cerrada de perímetro infinito.  $\square$

**Proposición 4.5.** *El área del copo de nieve de Koch es finita.*

*Demostración.* Sea  $A_k$  el área del copo de nieve de Koch en la  $k$ -ésima etapa. Si partimos del triángulo equilátero de lado 1, el área del triángulo inicial es  $A_0 = \frac{\sqrt{3}}{4}$ . En el siguiente paso, se añade un triángulo equilátero de lado  $\frac{1}{3}$  en cada lado, por lo tanto:

$$A_1 = \frac{\sqrt{3}}{4} + 3 \cdot \frac{\sqrt{3}}{4 \cdot 3^2} = \frac{\sqrt{3}}{3}$$

En la siguiente etapa tenemos que:

$$A_2 = A_1 + 4 \cdot 3 \cdot \frac{1}{4} \cdot \frac{\sqrt{3}}{3^4} = \frac{\sqrt{3}}{3} + \frac{\sqrt{3}}{3^3}$$

En la  $k$ -ésima etapa:

$$A_k = \frac{\sqrt{3}}{3} + \frac{\sqrt{3}}{3^3} + \frac{4\sqrt{3}}{3^5} + \frac{4^2\sqrt{3}}{3^7} + \frac{4^3\sqrt{3}}{3^9} + \dots$$

#### 4. Fractales

Tomando límite cuando  $k \rightarrow \infty$ :

$$A_\infty = \frac{\sqrt{3}}{3} + \sqrt{3} \cdot \sum_{i=1}^{\infty} \frac{4^{i-1}}{3^{2i+1}}$$

Resolviendo la serie geométrica infinita, el área que encierra el copo de nieve de Koch es:

$$A_\infty = \frac{\sqrt{3}}{3} + \frac{\sqrt{3}}{12} \cdot \frac{4}{5} = \frac{2\sqrt{3}}{5}$$

□

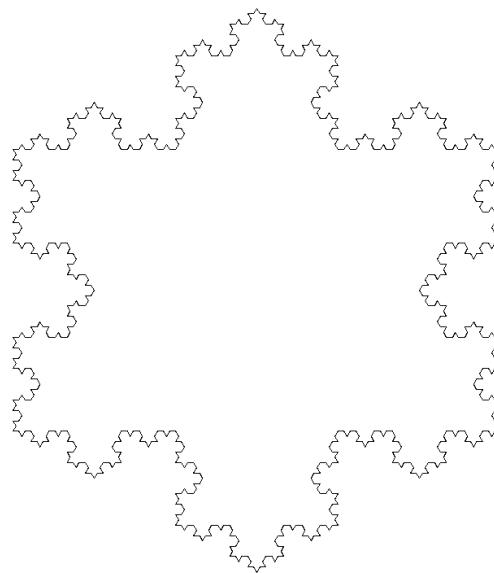


Figura 4.4.: [Python] Copo de nieve de Koch

#### 4.1.4. El cuadrado de Koch

El **cuadrado de Koch**  $K'$  es una variación de la curva de Koch vista en la sección 4.1.3. En esta ocasión, partimos de un cuadrado  $K'_0$  de lado 1 (por sencillez). Las distintas iteraciones se construyen reemplazando cada segmento del conjunto por cinco de longitud  $\frac{1}{3}$ . En la  $k$ -ésima etapa, tendremos  $4 \cdot 5^k$  segmentos de longitud  $(\frac{1}{3})^k$  (Figura 4.5).

**Proposición 4.6.** *El perímetro del cuadrado de Koch es infinito.*

*Demostración.* La longitud del cuadrado de Koch en la  $k$ -ésima etapa viene dada por la suma de los  $4 \cdot 5^k$  segmentos de longitud  $(\frac{1}{3})^k$ , es decir,  $4 \cdot (\frac{5}{3})^k$ . Como:

$$\lim_{k \rightarrow \infty} 4 \cdot \left(\frac{5}{3}\right)^k = \infty$$

el cuadrado de Koch es infinitamente largo, es decir, crece indefinidamente conforme  $k \rightarrow \infty$ .

□

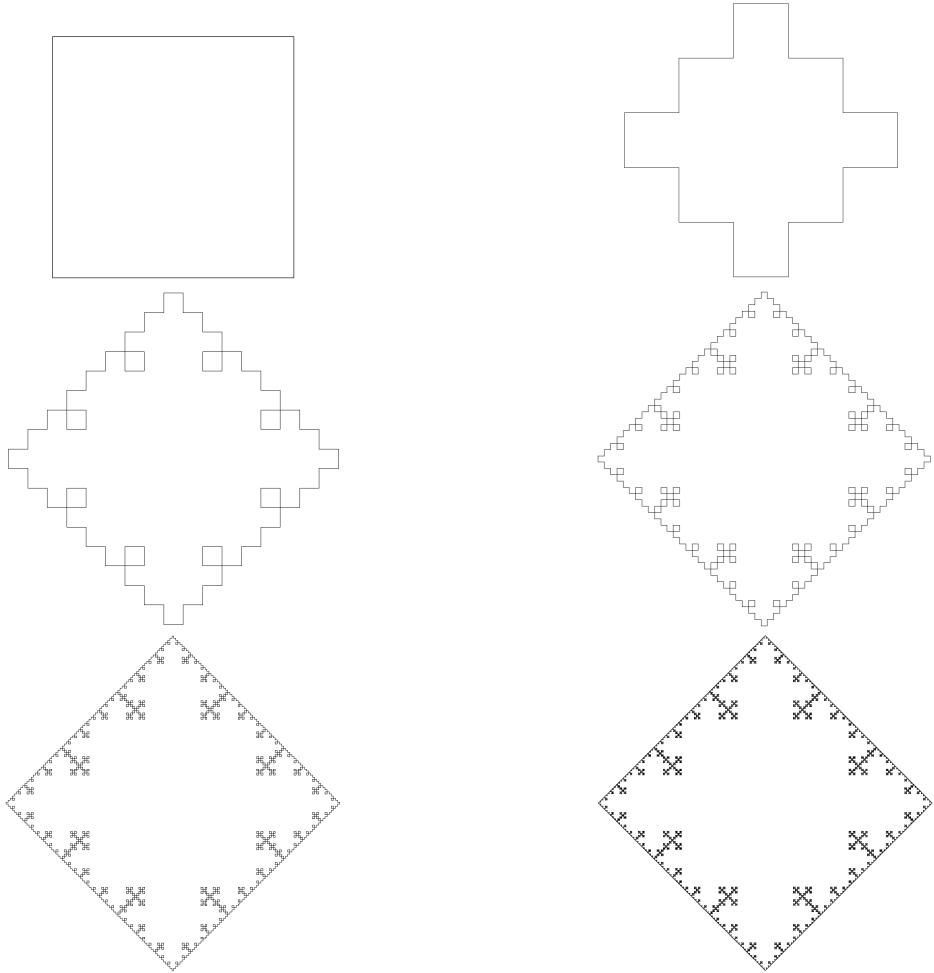


Figura 4.5.: [Python] Etapas de la construcción del cuadrado de Koch: (i)  $K'_0$  (ii)  $K'_1$  (iii)  $K'_2$  (iv)  $K'_3$  (v)  $K'_4$  (vi)  $K'_5$

**Proposición 4.7.** *El área del cuadrado de Koch es 2.*

*Demostración.* Sea  $A_k$  el área del cuadrado de Koch en la  $k$ -ésima etapa. Supongamos que partimos del cuadrado de lado 1, entonces  $A_0 = 1$ . En la siguiente etapa, tenemos que:

$$A_1 = 1 + 4 \cdot \left(\frac{1}{3}\right)^2$$

En la etapa 2, el área viene dada por:

$$A_2 = 1 + 4 \cdot \left(\frac{1}{3}\right)^2 + 4 \cdot 5 \cdot \left(\frac{1}{3}\right)^4$$

Continuando este procedimiento hasta la etapa  $k$ -ésima, tenemos que:

$$A_k = 1 + 4 \cdot \left(\frac{1}{3}\right)^2 + 4 \cdot 5 \cdot \left(\frac{1}{3}\right)^4 + 4 \cdot 5^2 \cdot \left(\frac{1}{3}\right)^6 + \cdots + 4 \cdot 5^{k-1} \cdot \left(\frac{1}{3}\right)^{2k}$$

#### 4. Fractales

Tomando límite cuando  $k \rightarrow \infty$ :

$$A_\infty = 1 + \frac{4}{9} + \sum_{i=1}^{\infty} 4 \cdot 5^i \cdot \left(\frac{1}{9}\right)^{i+1}$$

Se trata de la suma de una serie geométrica infinita, por ello:

$$A_\infty = 1 + \frac{4}{9} + \frac{\frac{4 \cdot 5}{9^2}}{1 - \frac{5}{3^2}} = 2$$

□

#### 4.1.5. Otros fractales

##### 4.1.5.1. La curva de Lévy

En 1906, **Ernesto Cesàro** (1859-1906) expuso por primera vez la **curva de Lévy**. Su nombre se debe al matemático francés **Paul Pierre Lévy** (1886-1971) quien, en 1938, exhibió su propiedad de autosimilaridad.

La curva de Lévy  $L$  se forma a partir de un segmento  $L_0$  de longitud 1 (por sencillez). Para construir  $L_1$ , tomamos este segmento por la hipotenusa de un triángulo rectángulo isósceles, y lo reemplazamos por los otros dos catetos del triángulo. En el siguiente paso, tomamos los dos segmentos anteriores como las hipotenusas de dos triángulos rectángulos isósceles, y los reemplazamos por los dos catetos correspondientes. En la  $k$ -ésima etapa, tendremos  $2^k$  segmentos de longitud  $2^{\frac{k}{2}}$  con respecto al segmento de partida (Figura 4.6).

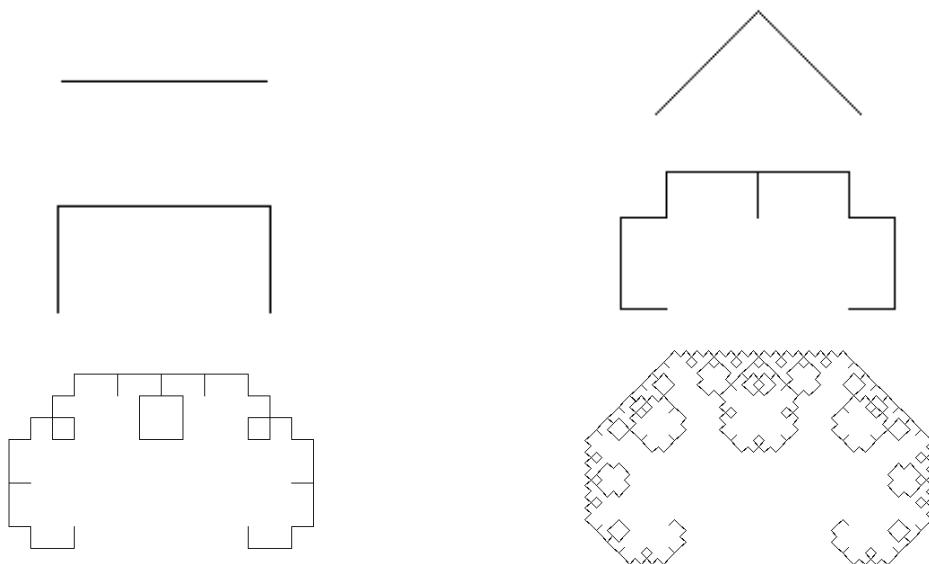


Figura 4.6.: [Python] Etapas de la construcción de la curva de Lévy: (i)  $L_0$  (ii)  $L_1$  (iii)  $L_2$  (iv)  $L_4$  (v)  $L_6$  (vi)  $L_9$

#### 4.1.5.2. La curva del dragón

La construcción de la **curva del dragón** D es muy similar a la de la curva de Lévy. Se forma a partir de un segmento  $D_0$  de longitud 1 (por simplicidad). Tomamos este segmento como la hipotenusa y construimos un triángulo rectángulo e isósceles. Finalmente, eliminamos el segmento inicial. En la siguiente etapa, aplicamos el mismo razonamiento a cada uno de los segmentos alternando la orientación de los triángulos rectángulo e isósceles (Figura 4.7).

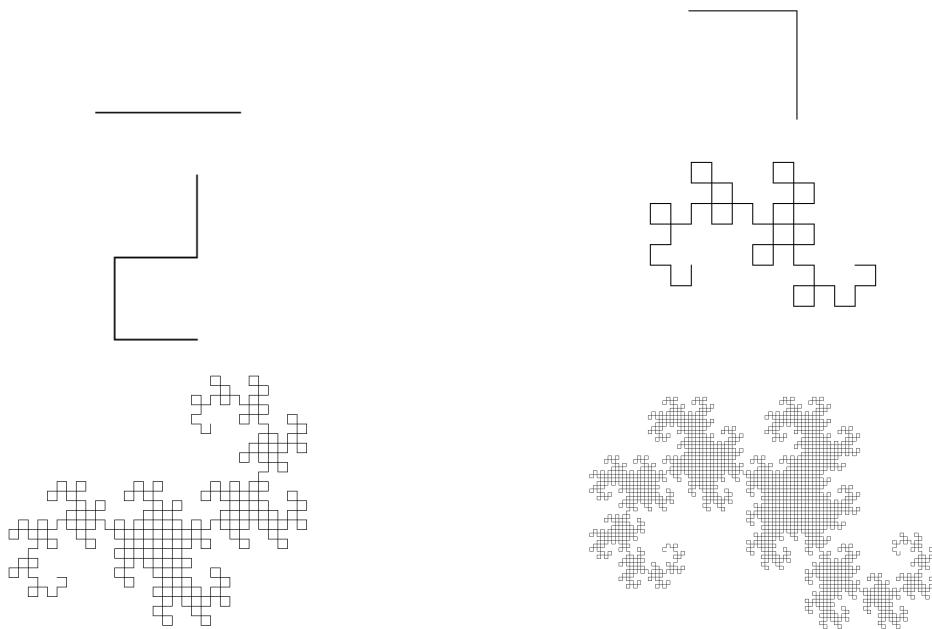


Figura 4.7.: [Python] Etapas de la construcción de la curva del dragón: (i)  $D_0$  (ii)  $D_1$  (iii)  $D_2$  (iv)  $D_6$  (v)  $D_9$  (vi)  $D_{12}$

## 4.2. Sistema de Lindenmayer (sistema-L)

El concepto de **sistema de Lindenmayer** (sistema-L) fue concebido por el biólogo y botánico **Aristid Lindenmayer** (1925-1989) en 1968. Un sistema-L es básicamente un lenguaje, un conjunto de reglas y símbolos que sirve para modelar el proceso de crecimiento de las plantas y para describir una gran variedad de organismos. Todos los conceptos que aparecen en esta sección se encuentran recogidos en [HOPo4].

El concepto central de los sistemas-L es el de **reescritura**. Se trata de una técnica para definir objetos complejos reemplazando sucesivamente partes de un objeto inicial usando reglas de escritura. El mecanismo de reescritura de estos sistemas es útil a la hora de aplicar la recursividad con el fin de lograr fractales complejos.

#### 4. Fractales

Los sistemas de Lindenmayer están compuestos de tres componentes principales:

- **Alfabeto:** conjunto de caracteres válidos que se pueden incluir en una palabra. Por ejemplo, si el alfabeto es  $\{A, B, C\}$  cualquier palabra válida contendrá los tres caracteres: A, B y C.
- **Axioma:** palabra que describe el estado inicial del sistema. Si el alfabeto es  $\{A, B, C\}$ , algunos axiomas de ejemplo son  $AAA, B, ACBAB, \dots$
- **Reglas de producción:** primero se aplican al axioma y luego de forma recursiva, generando nuevas palabras una y otra vez. Una regla del sistema incluye dos palabras, un predecesor y un sucesor. Por ejemplo, con la regla  $A \rightarrow AB$ , siempre que se encuentre una  $A$  en una palabra, se reemplaza por  $AB$ .

La naturaleza recursiva de las reglas de un sistema de Lindenmayer conduce a la auto-similitud, por lo que las formas fractales son fáciles de describir con este modelo. En esta sección presentaremos una interfaz gráfica muy simple basada en los gráficos **turtle** [Pyta]. Con esta interpretación podemos formular las construcciones originales de los fractales clásicos de una manera concisa, es decir, con pocas reglas de producción. Estas reglas deben establecerse con mucho cuidado, de tal manera que el resultado final se tiene que ajustar al objeto fractal.

Imaginemos una tortuga sentada en una hoja de papel mirando hacia una dirección. La cola de la tortuga está un poco sucia por lo que deja un rastro en el papel tan pronto como comience a moverse. La tortuga está completamente entrenada y entiende varios comandos que transmitimos por control remoto. Los comandos se dan en forma de símbolos que pueden ser letras ordinarias del alfabeto o caracteres especiales como  $+$  o  $-$ . Aquí hay un primer conjunto de instrucciones para la tortuga:

- F: avanza un paso (de cierta longitud l) dibujando el camino.
- f: lo mismo que F pero sin dibujar la trayectoria.
- +: gira a la izquierda (sentido antihorario) con un ángulo  $\theta$ .
- -: gira a la derecha (sentido horario) con un ángulo  $\theta$ .

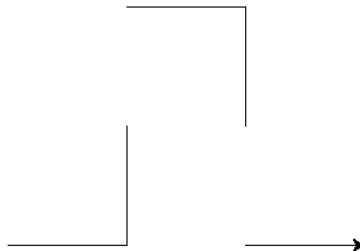


Figura 4.8.: [Python] Interpretación gráfica de la palabra  $F + Ff - F - Ff + F$  con ángulo  $\theta = 90^\circ$

#### 4.2. Sistema de Lindenmayer (sistema-L)

La longitud del paso  $l$  y el ángulo  $\theta$  deben especificarse antes de implementar un sistema-L. Además, necesitamos precisar el **iniciador (axioma)** y las reglas de producción que constituyen el **generador** del fractal.

En términos matemáticos decimos que la tortuga tiene un estado actual que consiste en la posición  $(x, y) \in \mathbb{R}^2$  que ocupa en ese mismo instante y un rumbo actual determinado por un ángulo  $\alpha$ . Esto se describe mediante una tupla de tres elementos  $(x, y, \alpha)$ . El estado de la tortuga cambia cada vez que interpreta un comando. Usando trigonometría elemental, se puede deducir que:

Comando	El estado $(x, y, \alpha)$ pasa a ser
F	$(x + l \cos \alpha, y + l \sin \alpha, \alpha)$
f	$(x + l \cos \alpha, y + l \sin \alpha, \alpha)$
+	$(x, y, \alpha - \theta)$
-	$(x, y, \alpha + \theta)$

Nuestra tortuga también acepta comandos que representan movimientos compuestos. Dicho de otra manera, debería poder interpretar un solo símbolo como una cierta secuencia de comandos de  $F, f, +$  y  $-$ . Para esto, utilizaremos los comandos  $L, R, S, Z, D$  y  $E$ . Todos ellos representan pequeñas curvas cuya interpretación parece obvia a partir de la elección de estos símbolos.

Comando	Interpretación
L	$+F - F - F +$
R	$-F + F + F -$
S	$FF + F + FF - F - FF$
Z	$FF - F - FF + F + FF$
D	$-- F + +F$
E	$F -- F + +$

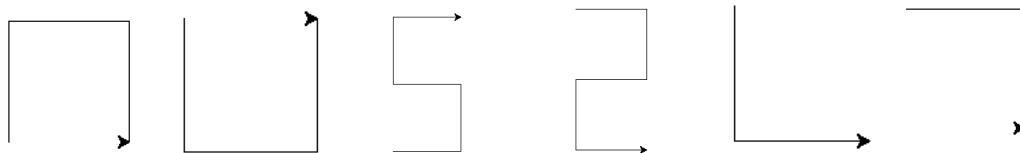


Figura 4.9.: [Python] Interpretación gráfica de los movimientos (i) L (ii) R (iii) S (iv) Z con ángulo  $\theta = 90^\circ$  y (v) D (vi) E con ángulo de  $\theta = 45^\circ$

Una vez que se ha explicado el funcionamiento de los gráficos turtle, estamos listos para explorar la construcción de fractales mediante sistemas de Lindenmayer. A continuación, se formularán unos cuantos ejemplos de sistemas. Es conveniente corroborar la elaboración de algunos de estos ejemplos con la construcción clásica que se dió en la sección 4.1.

**Ejemplo 4.2 (Curva de Koch mediante un sistema-L ).** En la construcción clásica (sección 4.1.3) reemplazamos repetidamente un segmento por una secuencia de cuatro (Figura 4.3 (ii)). Esta secuencia se puede describir con la palabra  $F + F - -F + F$  (donde elegimos el ángulo  $\theta = 60^\circ$ ). La nueva secuencia consiste en un segmento ( $F$ ), un giro de  $60^\circ$  en sentido antihorario (+), otro segmento ( $F$ ), dos giros en sentido horario de  $60^\circ$  cada uno (- -), un segmento ( $F$ ), un giro a la izquierda (+) y otro segmento ( $F$ ).

El proceso de construcción de la curva de Koch se describe mediante el siguiente sistema de Lindenmayer:

- Axioma:  $F$
- Reglas de producción:

$$F \rightarrow F + F - -F + F$$

$$+ \rightarrow +$$

$$- \rightarrow -$$

- Parámetros:  $\theta = 60^\circ$

**Ejemplo 4.3 (Conjunto de Cantor mediante un sistema-L ).** De nuevo, la construcción clásica (sección 4.1.1) comienza con un segmento. Este segmento se divide en tres partes iguales y se elimina el tercio medio abierto (Figura 4.1 (ii)). Este proceso se repite infinitamente. Podemos formalizar esto mediante la regla de producción  $F \rightarrow FfF$ .

El proceso de construcción del conjunto de Cantor se describe mediante el siguiente sistema de Lindenmayer:

- Axioma:  $F$
- Reglas de producción:

$$F \rightarrow FfF$$

$$f \rightarrow fff$$

La regla  $f \rightarrow fff$  garantiza que el tamaño del tercio medio sea el correcto. Si hubieramos usado la regla  $f \rightarrow f$  el espacio intermedio sería demasiado pequeño.

**Ejemplo 4.4 (Curva de la punta de flecha de Sierpiński mediante un sistema-L ).** La construcción de la curva de la punta de flecha de Sierpiński es muy similar a la del triángulo de Sierpiński (sección 4.1.2). En primer lugar, elegimos el ángulo  $\theta = 60^\circ$ . La punta de flecha se forma con la combinación de dos curvas que vienen descritas por  $L$  y  $+R - L - R +$  (Figura 4.10) . De tal manera que este fractal se describe con el siguiente sistema de Lindenmayer:

- Axioma:  $L$
- Reglas de producción:

$$L \rightarrow +R - L - R +$$

$$R \rightarrow -L + R + L -$$

$$+ \rightarrow +$$

$$- \rightarrow -$$

- Parámetros:  $\theta = 60^\circ$

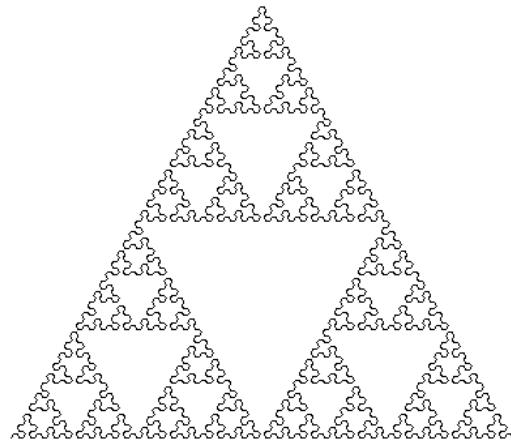


Figura 4.10.: [Python] Punta de flecha de Sierpiński

**Ejemplo 4.5 (Curva del dragón mediante un sistema-L ).** La construcción clásica (sección 4.1.5.2) comienza con una curva en forma de 'L' (Figura 4.7 (ii)). La siguiente etapa se obtiene mediante una rotación de  $45^\circ$  de una copia del conjunto del paso anterior y otra de  $135^\circ$ . Para ello, elegimos el ángulo  $\theta = 45^\circ$ . La curva en forma de 'L' se puede expresar con la palabra  $--F++F$  o  $D$ . La primera copia se puede codificar directamente, en cambio, hay que tener cuidado con la segunda ya que tenemos que tratarla en orden inverso:  $F--F++$  o  $E$ .

El proceso de construcción de la curva del dragón se detalla con el siguiente sistema de Lindenmayer:

- Axioma:  $D$
- Reglas de producción:

$$D \rightarrow -D++E$$

$$E \rightarrow D--E+$$

$$+ \rightarrow +$$

$$- \rightarrow -$$

- Parámetros:  $\theta = 45^\circ$

**Ejemplo 4.6 (Curva de Peano mediante un sistema-L ).** En 1890, Giuseppe Peano (1858 – 1932) presentó la **curva de Peano**. El algoritmo para la construcción de la curva de Peano es muy similar al de la curva de Koch donde cada segmento será remplazado por otros. Partimos de un segmento (Figura 4.11 (i)) de longitud 1. Deducimos 9 nuevos segmentos, cada uno de longitud  $\frac{1}{3}$ , situados tal y como se muestra en la figura 4.11 (ii). Se sustituye cada segmento por una curva poligonal autointersecante compuesta de nueve segmentos iguales. Este proceso se repite indefinidamente.

#### 4. Fractales

Tras dar un paso hacia adelante la tortuga tiene tres opciones: girar a la derecha, girar a la izquierda o continuar recto. Por ejemplo: si gira a la derecha, seguidamente la tortuga debe realizar dos giros a la izquierda con pasos al frente entre medias. Cuando llega a un punto de intersección con la curva puede continuar hacia adelante o girar a la izquierda.

En cada caso, la tortuga puede seguir un sentido horario o antihorario del generador terminando el último segmento hasta el final de la curva. Estas alternativas se describen con las palabras:

$$F - F + F + F + F - F - F - F + F \quad F - F + F + FF + F + F + FF$$

Las opciones para un giro inicial a la izquierda son:

$$F + F - F - F - F + F + F + F - F \quad F + F - F - FF - F - F - FF$$

Si la tortuga sigue recto en el primer punto de decisión, tenemos:

$$FF - F - F - FF - F - F + F \quad FF + F + F + FF + F + F - F$$

La construcción de la curva de Peano se describe mediante el siguiente sistema de Lindenmayer:

- Axioma:  $F$
- Reglas de producción:

$$F \rightarrow FF + F + F + FF + F + F - F$$

$$+ \rightarrow +$$

$$- \rightarrow -$$

- Parámetros:  $\theta = 90^\circ$

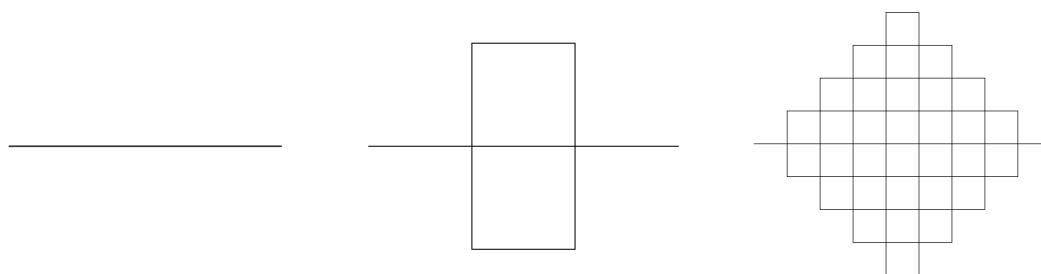


Figura 4.11.: [Python] Etapas de la construcción de la curva de Peano: (i)  $P_0$  (ii)  $P_1$  (iii)  $P_2$

**Ejemplo 4.7 (Curva de Hilbert mediante un sistema-L).** Un año después, en 1891, el matemático alemán **David Hilbert** (1862 – 1943) introdujo la **curva de Hilbert**. Se trata de un tipo de curva que rellena un cuadrado unidad. El inicio de la curva se encuentra en la parte

inferior izquierda y el final en la parte inferior derecha. La curva comienza con 3 segmentos, cada uno de longitud 1, que conectan los centros de los cuatro cuadrantes de un cuadrado (Figura 4.12 (i)). En la segunda etapa, se realizan cuatro copias del primer paso reducidas en  $\frac{1}{3}$ , se colocan en los cuadrantes y finalmente se unen mediante tres segmentos de longitud  $\frac{1}{2}$ . Si proseguimos infinitamente, la curva llenará el plano por completo.

La curva de Hilbert queda explicada mediante el siguiente sistema de Lindenmayer:

- Axioma:  $L$
- Reglas de producción:

$$L \rightarrow +RF - LFL - FR+$$

$$R \rightarrow -LF + RFR + FL-$$

$$F \rightarrow F$$

$$+ \rightarrow +$$

$$- \rightarrow -$$

- Parámetros:  $\theta = 90^\circ$

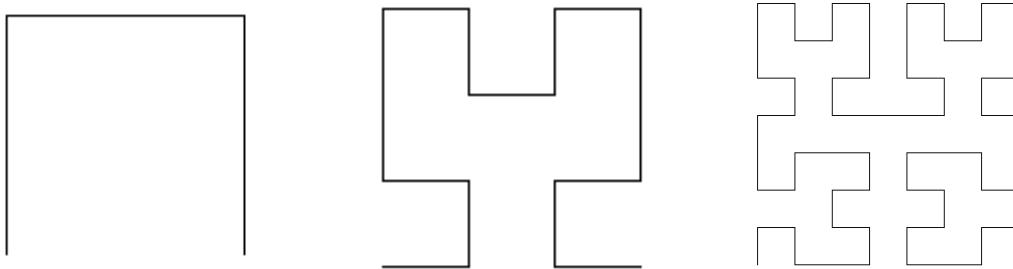


Figura 4.12.: [Python] Etapas de la construcción de la curva de Hilbert: (i)  $H_0$  (ii)  $H_1$  (iii)  $H_2$

La representación de una palabra mediante una cadena de símbolos es sencilla. Pero, ¿cómo podemos representar una estructura ramificada (como es típico en los árboles) mediante una palabra formada secuencialmente por caracteres? Parece ser un problema, sin embargo, existe una solución para ello. Simplemente, introducimos un nuevo símbolo que indique un punto de bifurcación, lo denotaremos corchete izquierdo [. Este corchete se emparejará con un correspondiente corchete de cierre ] indicando que en ese punto se termina la rama. Por ejemplo:

$ABA[BBAA][CCBB]ABA[AABB]ABA$

Eliminando los caracteres entre corchetes, obtenemos la palabra  $ABAABAABA$  que forma el tronco del árbol. De aquí brotan las tres ramas:  $BBAA$ ,  $CCBB$  y  $AABB$ .

Ahora bien, ¿qué debe de hacer nuestra tortuga cuando se encuentre con esos símbolos? En primer lugar, al recibir el comando [ debe recordar su estado actual y dirección. A continuación, la rama se dibuja mediante la interpretación habitual. Finalmente, cuando se active

#### 4. Fractales

el comando `]`, la tortuga debe regresar al punto de ramificación.

La implementación por computadora se lleva a cabo de manera más adecuada mediante el uso de una **pila**. Esta nos permite realizar un seguimiento jerárquico de las ramificaciones. Al encontrar el comando `[`, el estado actual de la tortuga se guarda en la parte superior de la pila. Por otro lado, el comando `]` saca el estado superior y coloca a la tortuga en ese estado. Examinaremos algunos ejemplos sencillos:

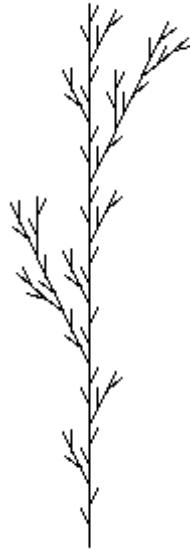


Figura 4.13.: [Python] Ramificación de la maleza. Axioma:  $F$  , reglas de producción:  $F \rightarrow F[+F]F[-F]F$  y parámetros:  $\theta = 25.7^\circ$ .

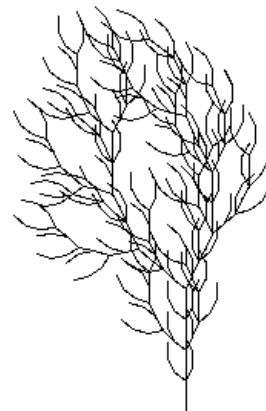


Figura 4.14.: [Python] Ramificación de un arbusto. Axioma:  $F$  , reglas de producción:  $F \rightarrow FF + [+F - F - F] - [-F + F + F]$  y parámetros:  $\theta = 25^\circ$ .

### 4.3. Sistema de funciones iteradas (SFI)

En esta sección, estableceremos la base matemática para la construcción de ciertos tipos de fractales denominados **sistemas de funciones iteradas**. El matemático británico **Michael Barnsley** es el precursor de la teoría de estos sistemas.

La mayor parte de los resultados de esta sección se plantearán en dos dimensiones y tendrán una extensión directa al caso n-dimensional  $\mathbb{R}^n$ . Dichas definiciones se podrán encontrar en [Elaoo].

Comencemos recordando la definición de **transformación lineal** en el plano.

**Definición 4.5.** Se dice que un mapa  $F: \mathbb{R}^2 \rightarrow \mathbb{R}^2$  es una transformación lineal si:

$$(i) \quad F(x + y) = F(x) + F(y) \text{ para todo } x, y \in \mathbb{R}^2.$$

$$(ii) \quad F(\alpha x) = \alpha F(x) \text{ con } \alpha \in \mathbb{R}, x \in \mathbb{R}^2$$

Una transformación lineal  $F$  puede estar representada por una matriz  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$  como se muestra a continuación:

$$F \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix} \quad (4.1)$$

Dado un punto  $\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2$  es posible calcular  $F \begin{pmatrix} x \\ y \end{pmatrix}$ . Para facilitar esta tarea, reescribimos la matriz  $A$  de la siguiente manera:

$$A = \begin{pmatrix} r \cos \theta & -s \sin \phi \\ r \sin \theta & s \cos \phi \end{pmatrix} \quad (4.2)$$

donde

$$\begin{aligned} r &= \sqrt{a^2 + c^2} & \cos \theta &= \frac{a}{\sqrt{a^2 + c^2}} \\ s &= \sqrt{b^2 + d^2} & \cos(\pi - \phi) &= \frac{b}{\sqrt{b^2 + d^2}} \end{aligned}$$

La expresión (4.2) da lugar a numerosos ejemplos, compuestos de rotaciones, reflexiones y homotecias.

**Definición 4.6.** Una transformación lineal **afín** es un mapa  $F: \mathbb{R}^2 \rightarrow \mathbb{R}^2$  representado como sigue :

$$F \begin{pmatrix} x \\ y \end{pmatrix} = A \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} ax + by + e \\ cx + dy + f \end{pmatrix}$$

Para continuar, es importante tener presentes las nociones de norma y espacio normado [Pay21].

**Definición 4.7.** Una **norma** en un espacio vectorial  $X$  es una aplicación  $\|\cdot\|: X \rightarrow \mathbb{R}$ , que a cada vector  $x \in X$  le hace corresponder un número real  $\|x\|$ , verificando las tres condiciones siguientes:

- (I) *Desigualdad triangular:*  $\|x + y\| \leq \|x\| + \|y\| \quad \forall x, y \in X$
- (II) *Homogeneidad por homotecias:*  $\|\lambda x\| = |\lambda| \|x\| \quad \forall x \in X, \forall \lambda \in \mathbb{R}$
- (III) *No degeneración:*  $x \in X, \|x\| = 0 \implies x = 0$

**Definición 4.8.** Un espacio **normado** es un par  $(X, \|\cdot\|)$  donde  $X$  es un espacio vectorial y  $\|\cdot\|$  una norma en  $X$ .

**Definición 4.9.** Si  $X$  es un espacio normado, se define la **distanza** entre dos puntos de  $X$  por:

$$d(x, y) = \|y - x\| \quad \forall x, y \in X \quad (4.3)$$

Para la siguiente definición necesitamos el hecho de que  $\mathbb{R}^2$  es un espacio métrico con la distancia inducida por la norma (4.3).

**Definición 4.10.** Una función  $F: S \rightarrow S$ ,  $S \subset \mathbb{R}^2$  es una **contracción** (o función contractiva) si existe un  $\alpha$  con  $0 \leq \alpha < 1$  que cumpla:

$$\|F(s_1) - F(s_2)\| \leq \alpha \|s_1 - s_2\| \quad \text{para todo } s_1, s_2 \in S$$

La constante  $\alpha$  recibe el nombre de **factor de contracción** de  $F$ .

Para la transformación lineal (4.1), obtenemos la siguiente desigualdad:

$$\|F(s_1) - F(s_2)\| = \|As_1 - As_2\| \leq \|A\| \|s_1 - s_2\|$$

Concluimos, por la definición 4.10, que  $F$  será una contracción de  $\mathbb{R}^2$  si  $\|A\| < 1$  o si los valores propios  $\lambda_1, \lambda_2$  de  $A$  cumplen que  $|\lambda_1| < 1$  y  $|\lambda_2| < 1$ .

En el caso de que  $A$  sea de la misma forma que la matriz (4.2), tenemos que sus valores propios  $\lambda_{1,2} = r \cos \theta \pm i \sin \theta$  y  $|\lambda_{1,2}| = |r|$ . Consecuentemente, si  $|r| < 1$  entonces  $F$  es contractiva.

Ahora dirigiremos nuestra atención al concepto de sistema de funciones iteradas. Sea  $\mathcal{H}(\mathbb{R}^2)$  la colección de todos los subconjuntos cerrados y acotados de  $\mathbb{R}^2$ . Sea  $F_1, F_2, \dots, F_N$  una familia de contracciones de  $\mathbb{R}^2$ . Definimos  $F$  como la unión de las funciones  $F_1, F_2, \dots, F_N$  (comúnmente conocido como el **operador de Hutchinson**):

$$F(S) = F_1(S) \cup F_2(S) \cup \dots \cup F_N(S) \quad S \in \mathcal{H}(\mathbb{R}^2)$$

Si  $F_1(S), F_2(S), \dots, F_N(S)$  son disjuntos (salvo por los límites), y  $S = F(S)$ , entonces se dice que  $S$  es autosimilar. Más adelante, se mostrará como todos los fractales disponen de esta propiedad.

**Definición 4.11.** Sea  $F_1, F_2, \dots, F_N$  una familia de contracciones de  $\mathbb{R}^2$  y  $S \in \mathcal{H}(\mathbb{R}^2)$  un subconjunto cerrado y acotado de  $\mathbb{R}^2$ . Se dice que el sistema  $\{S, F\} = \left\{S, \bigcup_{i=1}^N F_i\right\}$  es un **sistema de funciones iteradas**.

Una vez definido el concepto de SFI, demostraremos que el mapa unión o el operador de Hutchinson  $F$  es una contracción. Posteriormente, utilizaremos el principio de contracción de Banach para mostrar como la secuencia de iteraciones del mapa  $F$  converge a un conjunto cerrado y acotado  $A_F$ , al que denominaremos atractor de  $F$ . Dicho de otra forma,  $F^n(S) \rightarrow A_F$  para cualquier  $S$  y  $F(A_F) = A_F$ . Por esta última igualdad,  $A_F$  es un conjunto invariante. Este conjunto atractor es lo que llamaremos fractal o conjunto autosimilar.

Los siguientes resultados se darán para  $\mathbb{R}^2$  ya que contiene los ejemplos que después comentaremos. Se pueden extender fácilmente a  $\mathbb{R}^n$ .

Comenzamos tomando  $\mathcal{H}(\mathbb{R}^2)$  como la colección de todos los subconjuntos cerrados y acotados de  $\mathbb{R}^2$ . El objetivo que nos atañe es encontrar una norma  $\|\cdot\|$  o distancia  $d$  en  $\mathcal{H}(\mathbb{R}^2)$  que sea capaz de medir la distancia entre dos conjuntos de  $\mathcal{H}(\mathbb{R}^2)$ . Una norma que nos permita asegurar cuando una secuencia de iteraciones  $\{F^n(S)\}$  converge a un conjunto de  $\mathcal{H}(\mathbb{R}^2)$ . Sean  $A, B \in \mathcal{H}(\mathbb{R}^2)$ , la distancia entre un punto  $a \in A$  y  $B$  es:

$$d(a, B) = \inf \{\|a - b\| : b \in B\}$$

La distancia entre los conjuntos  $A$  y  $B$  es:

$$d(A, B) = \sup \{d(a, B) : a \in A\}$$

El principal problema de esta definición es que dados  $A, B \in \mathcal{H}(\mathbb{R}^2)$ ,  $d(A, B) \neq d(B, A)$ . Llegamos a la conclusión de que esta distancia entre conjuntos no es adecuada ya que no es simétrica. Es por ello que introducimos la noción de **distancia de Hausdorff**.

**Definición 4.12.** Sean  $A, B \in \mathcal{H}(\mathbb{R}^2)$ . La **distancia de Hausdorff** entre  $A$  y  $B$  se define como:

$$D(A, B) = \max \{d(A, B), d(B, A)\}$$

La distancia de Hausdorff se puede observar desde una perspectiva más geométrica. Para  $\epsilon > 0$  definimos un entorno  $N_\epsilon(A)$  de  $A$ :

$$N_\epsilon(A) = \left\{ x \in \mathbb{R}^2 : d(x, a) \leq \epsilon \text{ para cierto } a \in A \right\}$$

Dados  $A, B \in \mathcal{H}(\mathbb{R}^2)$ , tenemos que:

$$d(A, B) = \inf \{\epsilon > 0 : A \subset N_\epsilon(B)\}$$

$$d(B, A) = \inf \{\epsilon > 0 : B \subset N_\epsilon(A)\}$$

**Proposición 4.8.**  $(\mathcal{H}(\mathbb{R}^2), D)$  es un espacio métrico.

*Demostración.* Sean  $A, B, C \in \mathcal{H}(\mathbb{R}^2)$ . Comprobemos que la distancia de Hausdorff cumple las siguientes tres propiedades:

- (i) *Simetría:* claramente  $D(A, B) = \max \{d(A, B), d(B, A)\} = \max \{d(B, A), d(A, B)\} = D(B, A)$ .
- (ii) *No degeneración:* si  $A = B$ ,  $D(A, B) = D(A, A) = d(A, A) = \sup \{d(a, A) : a \in A\} = 0$ . Como  $A$  y  $B$  son compactos,  $D(A, B) = d(a, b)$  para un cierto  $a \in A$  y  $b \in B$ . Por tanto,

$0 \leq D(A, B) < \infty$ . Si  $A \neq B$ , asumimos que existe un  $a \in A$  tal que  $a \notin B$ . Entonces  $D(A, B) > d(a, B) > 0$ .

(III) *Desigualdad triangular:* para comprobar la desigualdad triangular, primero tenemos que probar que  $d(A, B) \leq d(A, C) + d(C, B)$ . Para cualquier  $a \in A$ :

$$\begin{aligned} d(a, B) &= \inf \{\|a - b\| : b \in B\} \\ &\leq \inf \{\|a - c\| + \|c - b\| : b \in B\} \quad \forall c \in C \\ &= \|a - c\| + \inf \{\|c - b\| : b \in B\} \quad \forall c \in C \\ d(a, B) &\leq \inf \{\|a - c\| : c \in C\} + \sup \{\inf \{\|c - b\| : b \in B\} : c \in C\} \\ &= d(a, C) + d(C, B) \end{aligned}$$

Consecuentemente, queda probado que  $d(A, B) \leq d(A, C) + d(C, B)$ . Del mismo modo,  $d(B, A) \leq d(B, C) + d(C, A)$ . Por tanto:

$$\begin{aligned} D(A, B) &= \max \{d(A, B), d(B, A)\} \\ &\leq \max \{d(B, C), d(C, B)\} + \max \{d(A, C), d(C, A)\} \\ &= D(B, C) + D(A, C) \end{aligned}$$

□

Para formular el principio de contracción de Banach deben definirse unos términos adicionales.

**Definición 4.13.** Una sucesión  $\{x_n\}$  en un espacio métrico  $X$  es de **Cauchy** si para todo  $\epsilon > 0$  existe un  $n_0 \in \mathbb{Z}^+$  tal que  $d(x_n, x_p) < \epsilon$  para todo  $n, p \geq n_0$ .

**Definición 4.14.** Una sucesión  $\{x_n\}$  en un espacio métrico  $X$  es **convergente** si existe un  $x \in X$  tal que  $\{d(x_n, x)\} \rightarrow 0$ , en cuyo caso  $x$  es único. Le llamamos límite de la sucesión  $\{x_n\}$  y escribimos  $\{x_n\} \rightarrow x$ .

**Definición 4.15.** Un espacio métrico  $(X, d)$  es **completo** si toda sucesión de Cauchy del espacio  $X$  es convergente.

**Proposición 4.9.**  $(\mathcal{H}(\mathbb{R}^2), D)$  es un espacio métrico completo.

Puesto que la demostración de la proposición 4.9 es extensa, simplemente comentaremos que se puede encontrar en [Bar88].

Debido a la completitud de  $(\mathcal{H}(\mathbb{R}^2), D)$  estamos listos para enunciar y demostrar el principio de contracción de Banach.

**Teorema 4.1 (Principio de contracción de Banach).** Sea  $F: X \rightarrow X$  una función contractiva con factor de contracción  $\alpha$  en un espacio métrico completo  $(X, d)$ . Entonces  $F$  tiene un único punto fijo  $x^* \in X$  que es un atractor global, es decir, para cada  $x \in X$ ,  $F^n(x) \rightarrow x^*$  cuando  $n \rightarrow \infty$ .

*Demostración.* Sea  $x \in X$  un punto arbitrario. Para  $m < n$ :

$$\begin{aligned} d(F^m(x), F^n(x)) &= d(F^m(x), F^m(F^{n-m}(x))) \\ &\leq \alpha^m d(x, F^{n-m}(x)) \\ &\leq \alpha^m [d(x, F(x)) + d(F(x), F^2(x)) + \cdots + d(F^{n-m-1}(x), F^{n-m}(x))] \\ &\leq \alpha^m d(x, F(x)) [1 + \alpha + \cdots + \alpha^{n-m-1}] \\ &< \frac{\alpha^m d(x, F(x))}{1 - \alpha} \end{aligned}$$

Como por hipótesis  $\alpha < 1$ , es evidente que la sucesión  $\{F^n(x)\}$  es una sucesión de Cauchy y por la completitud de  $X$ ,  $\lim_{n \rightarrow \infty} F^n(x) = x^*$ , para cierto  $x^* \in X$ . La continuidad de  $F$  nos asegura que  $x^*$  es un punto fijo:

$$F(x^*) = F(\lim_{n \rightarrow \infty} F^n(x)) = \lim_{n \rightarrow \infty} F^{n+1}(x) = x^*$$

Veamos que el punto fijo  $x^*$  es único. Supongamos  $x^*, y^* \in X$  puntos fijos, o sea,  $F(x^*) = x^*$  y  $F(y^*) = y^*$ . Se tiene que:

$$d(x^*, y^*) = d(F(x^*), F(y^*)) \leq \alpha d(x^*, y^*)$$

Por tanto,  $0 \leq (1 - \alpha)d(x^*, y^*) \leq 0$ . De donde se sigue que  $x^* = y^*$ .  $\square$

**Lema 4.1.** Si  $F$  es una contracción de  $\mathbb{R}^2$  entonces es una contracción en  $\mathcal{H}(\mathbb{R}^2)$ .

*Demostración.* Sean  $A, B \in \mathcal{H}(\mathbb{R}^2)$ . Para un cierto  $a \in A$  existe un  $b \in B$  tal que  $\|a - b\| = d(a, B)$ . Para probar esto, notamos que el conjunto  $G = \{d(a, x) : x \in B\}$  está acotado inferiormente por 0. Por el principio de buena ordenación, tiene un límite inferior  $\delta \geq 0$  y  $d(a, B) = \delta$ . Es más, existe una sucesión  $\{b_n\}$  de  $B$  tal que  $d(a, b_n) \rightarrow \delta$  cuando  $n \rightarrow \infty$ .

Como  $B$  es cerrado y acotado, existe una subsucesión  $\{b_{n_i}\}$  de  $\{b_n\}$  que converge a  $b \in B$ . Ahora,

$$d(a, b) \leq d(a, b_{n_i}) + d(b_{n_i}, b)$$

Como  $d(a, b_{n_i}) \rightarrow \delta$ ,  $d(b_{n_i}, b) \rightarrow 0$ , tenemos que  $d(a, b) \leq \delta$ , pero por la definición de  $\delta$ ,  $d(a, b) \geq \delta$ . Así pues,  $d(a, b) = \delta = d(a, B)$ . Sea  $F$  una contracción de  $\mathbb{R}^2$  con factor de contracción  $\alpha$ . Entonces:

$$\|F(a) - F(b)\| \leq \alpha \|a - b\| = \alpha d(a, B) \leq \alpha D(A, B)$$

Por tanto,  $d(F(a), F(b)) \leq \alpha D(A, B)$  y consecuentemente  $D(F(A), F(B)) \leq \alpha D(A, B)$ .  $\square$

#### 4. Fractales

**Lema 4.2.** Si  $F_1, F_2, \dots, F_N$  es una familia de contracciones con factores de contracción  $\alpha_i$ ,  $1 \leq i \leq N$  en  $\mathcal{H}(\mathbb{R}^2)$ , entonces su unión  $F = \bigcup_{i=1}^N F_i$  es una contracción con factor  $\alpha = \max\{\alpha_i : 1 \leq i \leq N\}$ .

*Demostración.* En primer lugar, si  $A, B \in \mathcal{H}(\mathbb{R}^2)$ , entonces  $D\{F_1(A) \cup F_2(A), F_1(B) \cup F_2(B)\} \leq \max\{D(F_1(A), F_1(B)), D(F_2(A), F_2(B))\}$ . De forma que:

$$D(F_1(A) \cup F_2(A), F_1(B) \cup F_2(B)) \leq \alpha D(A, B)$$

donde  $\alpha = \max\{\alpha_1, \alpha_2\}$ .

Para completar la demostración del lema, utilizaremos inducción en  $N$ . Supongamos que para  $N = k$ ,  $\bigcup_{i=1}^k F_i$  es una contracción:

$$D(\bigcup_{i=1}^k F_i(A), \bigcup_{i=1}^k F_i(B)) \leq \alpha(k) D(A, B)$$

donde  $\alpha(k) = \max\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ , entonces:

$$\begin{aligned} D(\bigcup_{i=1}^{k+1} F_i(A), \bigcup_{i=1}^{k+1} F_i(B)) &= D(\bigcup_{i=1}^k F_i(A) \cup F_{k+1}(A), \bigcup_{i=1}^k F_i(B) \cup F_{k+1}(B)) \\ &\leq \alpha(k+1) D(A, B) \end{aligned}$$

donde  $\alpha(k+1) = \max\{\alpha(k), \alpha_{k+1}\}$ .

Por lo que  $F = \bigcup_{i=1}^N F_i$  es una contracción para todo  $N$ . □

**Teorema 4.2.** Si  $F_1, F_2, \dots, F_N$  son contracciones de  $\mathbb{R}^2$ , entonces existe un atractor global  $A \in \mathcal{H}(\mathbb{R}^2)$  para el mapa unión u operador de Hutchinson  $F = \bigcup_{i=1}^N F_i$ . Más concretamente, para cada  $B \in \mathcal{H}(\mathbb{R}^2)$ ,  $F^n(B)$  converge a  $A$  con la distancia de Hausdorff.

*Demostración.* Por el lema 4.1,  $F_1, F_2, \dots, F_N$  son contracciones de  $\mathcal{H}(\mathbb{R}^2)$ . Usando el lema 4.2, tenemos que  $F = \bigcup_{i=1}^N F_i$  es también una contracción en  $\mathcal{H}(\mathbb{R}^2)$ . Dado que  $(\mathcal{H}(\mathbb{R}^2), D)$  es un espacio métrico completo, este teorema queda demostrado aplicando el Principio de contracción de Banach (teorema 4.1). □

#### 4.3.1. Algoritmos para la obtención del fractal asociado a un SFI

Legados a este punto, es útil dar algunos ejemplos para ilustrar la teoría de los SFI. Estos ejemplos nos resultarán familiares, ya que algunos de ellos se describieron en la sección 4.1. Afrontaremos la construcción de los mismos desde una perspectiva más rigurosa y utilizaremos los resultados anteriormente explicados para conseguirlo. Existen dos tipos de algoritmos para generar fractales: **determinista** y **aleatorio**.

##### 4.3.1.1. SFI determinista

Sea  $\{S, F = \bigcup_{i=1}^N F_i\}$  un SFI. El algoritmo determinista se basa en calcular directamente la sucesión  $\{F^n(S)\}$ . Aplicando el teorema 4.2, la sucesión anterior converge a un atractor global  $A_F$  con la distancia de Hausdorff.

**Ejemplo 4.8 (El conjunto de Cantor).** Sea  $C_0 = [0, 1]$  (Figura 4.1 (i)) y  $F_1, F_2$  la siguiente familia de contracciones:

$$F_1(x) = \frac{1}{3}x$$

$$F_2(x) = \frac{1}{3}x + \frac{2}{3}$$

Es fácil comprobar que  $\|F_i(u) - F_i(v)\| = \frac{1}{3}\|u - v\|$  para  $i = 1, 2$ . Entonces, el sistema de funciones iteradas  $\{C_0, F = \cup_{i=1}^2 F_i\}$  genera el conjunto de Cantor  $C$  como el límite de  $F^n(C_0) \rightarrow C$  y  $F(C) = C$ . Aplicando el proceso iterativo de obtención del atractor obtenemos:

$$C_0 = [0, 1]$$

$$F(C_0) = F_1(C_0) \cup F_2(C_0) = [0, \frac{1}{3}] \cup [\frac{2}{3}, 1]$$

$$F^2(C_0) = F_1(F(C_0)) \cup F_2(F(C_0)) = [0, \frac{1}{9}] \cup [\frac{2}{9}, \frac{1}{3}] \cup [\frac{6}{9}, \frac{7}{9}] \cup [\frac{8}{9}, 1]$$

Una característica interesante es que el atractor  $C$  no depende del conjunto inicial. Por tanto, en lugar de tomar  $[0, 1]$  podríamos considerar el intervalo  $[0, \frac{1}{2}]$ , el conjunto de los racionales o incluso un solo punto.

La construcción informal del conjunto de Cantor se encuentra en la sección 4.1.1.

**Ejemplo 4.9 (Triángulo de Sierpiński).** Sea  $S_0 = \Delta$  un triángulo equilátero (Figura 4.2 (i)) y  $F_1, F_2, F_3$  la siguiente familia de contracciones:

$$F_1 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$F_2 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix}$$

$$F_3 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{4} \\ \frac{\sqrt{3}}{4} \end{pmatrix}$$

Es fácil probar que  $\|F_i(u) - F_i(v)\| = \frac{1}{2}\|u - v\|$  para  $i = 1, 2, 3$ . Entonces  $\{S_0, F = \cup_{i=1}^3 F_i\}$  es un sistema de funciones iteradas. Adicionalmente,  $F_1(S_0) = T_1, F_2(S_0) = T_2, F_3(S_0) = T_3$  y por tanto,  $F(S_0)$  es la unión de  $T_1, T_2, T_3$ . Los conjuntos  $T_1, T_2, T_3$  son los tres triángulos que aparecen en la primera etapa de la construcción del triángulo de Sierpiński (Figura 4.2 (ii)).

Repitiendo este proceso, tenemos que  $F_1(T_1) = T_{11}, F_2(T_1) = T_{21}, F_3(T_1) = T_{31}, F_1(T_2) = T_{12}, F_2(T_2) = T_{22}, F_3(T_2) = T_{32}, F_1(T_3) = T_{13}, F_2(T_3) = T_{23}, F_3(T_3) = T_{33}$  donde  $T_{11}, \dots, T_{33}$  son los nueve triángulos que aparecen en la segunda etapa de la construcción del triángulo de Sierpiński (Figura 4.2 (iii)). La sucesión  $\{F^n(S_0)\}_{n=1}^\infty$  converge al triángulo de Sierpiński  $S$ .

#### 4. Fractales

Otra forma de expresar las transformaciones, en función de  $a, b, c, d, e, f$ , es con el siguiente código:

	a	b	c	d	e	f
$F_1$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	0	0
$F_2$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	0
$F_3$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{\sqrt{3}}{4}$

La construcción menos técnica del triángulo de Sierpiński se encuentra en la sección 4.1.2.

**Ejemplo 4.10 (La curva de Koch).** Tomemos el intervalo  $K_0 = [0, 1]$  como el segmento de partida de la curva de Koch (Figura 4.3 (i)). El generador de la curva consiste en las cuatro rectas de longitud  $\frac{1}{3}$  que aparecen en la primera etapa de la construcción (Figura 4.3 (ii)). Necesitamos cuatro contracciones  $F_1, F_2, F_3, F_4$  para generar esos segmentos.

Para obtener el primero (empezando por la izquierda), contraemos el segmento  $[0, 1]$  mediante un factor de  $\frac{1}{3}$ :

$$F_1 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Para conseguir el segundo segmento, contraemos el conjunto inicial en un factor de  $\frac{1}{3}$ , rotamos con un ángulo de  $\frac{\pi}{3}$  en sentido antihorario y nos trasladamos  $\begin{pmatrix} \frac{1}{3} \\ 0 \end{pmatrix}$ :

$$\begin{aligned} F_2 \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \frac{1}{3} \cos \frac{\pi}{3} & -\frac{1}{3} \sin \frac{\pi}{3} \\ \frac{1}{3} \sin \frac{\pi}{3} & \frac{1}{3} \cos \frac{\pi}{3} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{3} \\ 0 \end{pmatrix} \\ F_2 \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \frac{1}{6} & -\frac{\sqrt{3}}{6} \\ \frac{\sqrt{3}}{6} & \frac{1}{6} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{3} \\ 0 \end{pmatrix} \end{aligned}$$

El tercer segmento se obtiene mediante una contracción con factor de  $\frac{1}{3}$ , una rotación de  $\frac{\pi}{3}$  en sentido horario y una translación de  $\begin{pmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{6} \end{pmatrix}$ :

$$\begin{aligned} F_3 \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \frac{1}{3} \cos(-\frac{\pi}{3}) & -\frac{1}{3} \sin(-\frac{\pi}{3}) \\ \frac{1}{3} \sin(-\frac{\pi}{3}) & \frac{1}{3} \cos(-\frac{\pi}{3}) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{6} \end{pmatrix} \\ F_3 \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \frac{1}{6} & \frac{\sqrt{3}}{6} \\ -\frac{\sqrt{3}}{6} & \frac{1}{6} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{6} \end{pmatrix} \end{aligned}$$

Finalmente, el cuarto y último segmento se genera mediante una contracción con factor de  $\frac{1}{3}$  y una translación de  $\begin{pmatrix} \frac{2}{3} \\ 0 \end{pmatrix}$ :

$$F_4 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{2}{3} \\ 0 \end{pmatrix}$$

Concluimos, dando el código del SFI:

	a	b	c	d	e	f
$F_1$	$\frac{1}{3}$	0	0	$\frac{1}{3}$	0	0
$F_2$	$\frac{1}{6}$	$-\frac{\sqrt{3}}{6}$	$\frac{\sqrt{3}}{6}$	$\frac{1}{6}$	$\frac{1}{3}$	0
$F_3$	$\frac{1}{6}$	$\frac{\sqrt{3}}{6}$	$-\frac{\sqrt{3}}{6}$	$\frac{1}{6}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{6}$
$F_4$	$\frac{1}{3}$	0	0	$\frac{1}{3}$	$\frac{2}{3}$	0

La curva de Koch se obtiene aplicando las transformaciones  $F_i$  a los segmentos que van apareciendo. Por ejemplo,  $F_i(\hat{K}_j) = \hat{K}_{ij}$  con  $1 \leq i, j \leq 4$ . La curva  $K$  se alcanza como el atractor del conjunto, es decir, como el límite de la sucesión  $\{F^n(K_0)\}$  donde  $F$  es la unión de las contracciones  $F_1, F_2, F_3, F_4$ .

La construcción menos rigurosa de la curva de Koch se encuentra en la sección [4.1.3](#).

#### 4.3.1.2. SFI aleatorio y el juego del caos

El cálculo del atractor de un SFI mediante el algoritmo determinista es bastante lento, consume más tiempo y carece de interés computacional. Por ello, resulta muy útil el algoritmo aleatorio, también conocido como el juego del caos.

Dado un un SFI  $\{S, F = \cup_{i=1}^N F_i\}$ , asignamos a cada contracción  $F_i$  una probabilidad  $p_i > 0$  con  $i = 1, 2, \dots, N$  donde  $\sum_{i=1}^N p_i = 1$ . Esta probabilidad  $p_i$  representa la oportunidad de  $F_i$  de ser seleccionada para aplicarla en el siguiente paso. Se elige un  $x_0 \in \mathbb{R}^2$  arbitrario. A continuación, se escoge un  $x_1 \in \{F_1(x_0), \dots, F_N(x_0)\}$  donde  $F_i(x_0)$ ,  $i = 1, 2, \dots, N$ , tiene una probabilidad  $p_i$  de ser seleccionado. Análogamente e independiente del paso anterior se escoge  $x_2 \in \{F_1(x_1), \dots, F_N(x_1)\}$ . Cuando tenemos construido  $\{x_0, x_1, \dots, x_k\}$ , se determina  $x_{k+1}$  mediante el proceso anterior.

Si queremos conseguir el atractor mediante este algoritmo, nos conviene obtener la mejor aproximación con un número de puntos adecuado. Si las contracciones  $F_i$  tienen aproximadamente la misma masa o medida entonces nos interesa tomar  $p_i = \frac{1}{N}$  donde  $i = 1, 2, \dots, N$ . En caso de que las contracciones  $F_i$  sean transformaciones afines de la forma:

$$F_i \begin{pmatrix} x \\ y \end{pmatrix} = A_i \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

una manera natural de elegir las probabilidades  $p_i$  es:

$$p_i = \frac{|det(A_i)|}{\sum_{i=1}^N |det(A_i)|}$$

#### 4. Fractales

**Ejemplo 4.11 (El helecho de Barnsley).** Barnsley hizo uso de la teoría de los SFI para crear uno de los fractales más distinguidos, **el helecho de Barnsley**, con sólo cuatro transformaciones. El primer punto dibujado se situa en el origen y los nuevos se calculan iterativamente aplicando aleatoriamente una de las cuatro transformaciones.

A continuación, daremos el código de las contracciones con su probabilidad:

	a	b	c	d	e	f	Probabilidad
$F_1$	0	0	0	0.16	0	0	0.01
$F_2$	0.85	0.04	-0.04	0.85	0	1.6	0.85
$F_3$	0.2	-0.26	0.23	0.22	0	1.6	0.07
$F_4$	-0.15	0.28	0.26	0.24	0	0.44	0.07

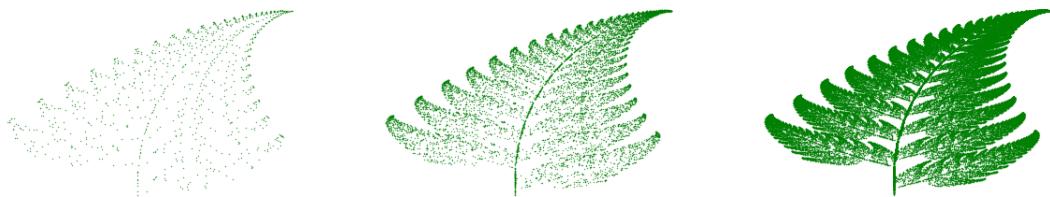


Figura 4.15.: [Python] Algoritmo aleatorio para la obtención del helecho de Barnsley con (i) 1000 (ii) 10000 (iii) 100000 puntos

La transformación  $F_1$  se elige con una probabilidad de 0.01. Simplemente asigna cualquier punto a uno de la base del tallo. La contracción  $F_2$  se escoge el 85 % de las veces, genera copias sucesivas del tallo y hojas inferiores para formar el helecho.  $F_3$  y  $F_4$  se eligen el 14 % del tiempo (7 % cada una) y dibujan las hojas de la parte inferior izquierda y derecha respectivamente.

**Ejemplo 4.12 (El triángulo de Sierpiński aleatorio).** Para obtener el triángulo de Sierpiński mediante el algoritmo aleatorio, simplemente asignamos a cada una de las tres contracciones del ejemplo 4.9 una probabilidad:

$$p_1 = \frac{1}{3} \quad p_2 = \frac{1}{3} \quad p_3 = \frac{1}{3}$$

Sin embargo, el objetivo de este ejemplo es construir este fractal jugando al llamado juego del caos con un dado. En primer lugar, consideramos un triángulo equilátero con vértices  $A, B, C$  y un punto inicial  $x_0$  en su interior. Las reglas del juego del caos son muy sencillas.

Comenzamos con un punto inicial  $x_0$  dentro del triángulo.

- (i) Lanzamos un dado con seis caras numeradas del 1 al 6.
- (ii) Si el número obtenido es 1 o 2, dibujamos el punto medio del segmento  $\overline{AB}$ . Si el número es 3 o 4, dibujamos el punto medio del segmento  $\overline{AC}$ . Finalmente, si es 5 o 6, dibujamos el punto medio del segmento  $\overline{BC}$ .
- (iii) Comenzando por los nuevos puntos generados en el segundo paso, volvemos al principio.

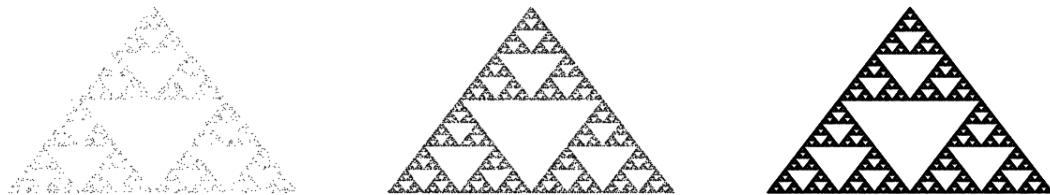


Figura 4.16.: [Python] Juego del caos para la obtención del triángulo Sierpiński con (i) 1000  
(ii) 10000 (iii) 100000 puntos



## 5. Dimensión fractal

En el capítulo 4 abordamos la cuestión de definir un fractal. En esta definición interviene el término de dimensión de Hausdorff. El objetivo de este capítulo, es culminar la teoría fractal exponiendo el concepto de dimensión de Hausdorff y especificando un método computacional para el cálculo de esta dimensión.

Las definiciones y conceptos de interés se encuentran recogidos en [Bar88] y [Lyn18].

En el capítulo anterior, vimos como la autosimilitud es una característica importante de los fractales. Sin embargo, esta propiedad es compartida por objetos que no lo son como rectas, cuadrados, cubos, ... Además, a veces es complicado detectar esta propiedad de autosemejanza en fractales con patrones complejos. Por ello, necesitamos encontrar otra característica exclusiva de los fractales. Esto nos lleva a la definición 4.2 que dió Mandelbrot.

La noción de dimensión que utilizamos habitualmente es la **dimensión topológica** y es siempre un número natural. Para un espacio formado por un punto, la dimensión topológica es 0, para la recta real es 1, para el plano euclídeo es 2,...

**Definición 5.1.** Un conjunto  $A \subset \mathbb{R}^n$  tiene dimensión topológica 0 si para cada punto de  $A$  podemos encontrar un entorno muy próximo de  $\mathbb{R}^n$  cuya frontera no intersecte el conjunto  $A$ .

**Ejemplo 5.1.** El conjunto de Cantor tiene dimensión topológica 0.

Si  $x$  es un punto del conjunto de Cantor,  $x \in C$ , entonces existen  $a < x$  y  $b > x$ , arbitrariamente cerca de  $x$ , no pertenecientes al conjunto. Entonces el intervalo  $[a, b]$  es un entorno de  $x$  cuya frontera no intersecta a  $C$ .

**Definición 5.2.** Un conjunto  $A \subset \mathbb{R}^n$  tiene dimensión topológica  $k > 0$ , denotada por  $D_t(A)$ , si cada punto de  $A$  posee entornos de  $\mathbb{R}^n$  cuya frontera intersecta el conjunto  $A$  en un subconjunto de dimensión  $k - 1$  y además  $k$  es el número natural más pequeño con esta propiedad.

**Ejemplo 5.2.** Un segmento tiene dimensión topológica 1.

Se satisfacen las condiciones para  $k \leq 1$ , ya que toda bola de radio suficientemente pequeño intersectará en 1 o 2 puntos, que es un conjunto de dimensión topológica 0.

**Ejemplo 5.3.** El triángulo de Sierpiński tiene dimensión topológica 1.

Para cualquier punto  $x$  del triángulo de Sierpiński existen bolas cuyo interior contiene a  $x$  y cuya frontera pasa por los vértices de uno de los triángulos que aparecen en la construcción iterativa. La frontera de este entorno intersecta en tres puntos, como el conjunto es autosimilar, siempre podemos encontrar bolas de radio arbitrariamente pequeño.

## 5. Dimensión fractal

Los objetos fractales necesitan otro tipo de dimensión. De hecho, no existen en las dimensiones euclídeas, sino en las dimensiones fraccionarias, que son un descubrimiento del matemático alemán **Félix Hausdorff** y que más tarde completó **Besicovitvh**.

**Definición 5.3.** Sea  $(X, d)$  un espacio métrico completo. Sea  $A \in \mathcal{H}(X)$  un subconjunto no vacío y compacto de  $X$ . Dado  $\epsilon > 0$ , sea  $B(x, \epsilon)$  la bola cerrada de radio  $\epsilon$  y centro  $x \in X$ . Se define el entero  $N(A, \epsilon)$  como el número de bolas cerradas de radio  $\epsilon$  necesarias para cubrir el conjunto  $A$ . Es decir:

$$N(A, \epsilon) = \text{entero positivo más pequeño } M \text{ tal que } A \subset \bigcup_{n=1}^M B(x_n, \epsilon)$$

para un conjunto de puntos distintos  $\{x_n : n = 1, 2, \dots, M\} \subset X$ .

La idea intuitiva que hay detrás de la dimensión fractal es que el conjunto  $A$  tiene dimensión fractal  $D$  si:

$$N(A, \epsilon) \approx C\epsilon^{-D} \quad \text{para una constante positiva } C$$

Aquí usamos la notación  $\approx$  de la siguiente manera. Sean  $f(\epsilon)$  y  $g(\epsilon)$  funciones de variable real con  $\epsilon > 0$ . Entonces  $f(\epsilon) \approx g(\epsilon)$  significa que  $\lim_{\epsilon \rightarrow 0} \left\{ \frac{\ln(f(\epsilon))}{\ln(g(\epsilon))} \right\} = 1$ .

Si lo resolvemos para  $D$  encontramos que:

$$D \approx \frac{\ln N(A, \epsilon) - \ln C}{\ln(\frac{1}{\epsilon})}$$

Observamos como  $\frac{\ln C}{\ln(\frac{1}{\epsilon})}$  se aproxima a cero cuando  $\epsilon \rightarrow 0$ . Esto nos lleva a la siguiente definición.

**Definición 5.4.** Sea  $A \in \mathcal{H}(X)$  donde  $(X, d)$  es un espacio métrico. Para todo  $\epsilon > 0$ , sea  $N(A, \epsilon)$  el menor número de bolas cerradas de radio  $\epsilon > 0$  necesarias para cubrir  $A$ . Si

$$D = \lim_{\epsilon \rightarrow 0} \left\{ \frac{\ln(N(A, \epsilon))}{\ln(\frac{1}{\epsilon})} \right\}$$

existe entonces  $D$  es la dimensión fractal de  $A$ . Usaremos la notación  $D = D(A)$  y diremos que  $A$  tiene dimensión fractal  $D$ .

El siguiente teorema simplifica el proceso de cálculo de la dimensión fractal. Nos permite reemplazar la variable continua  $\epsilon > 0$  por una variable discreta.

**Teorema 5.1.** Sea  $A \in \mathcal{H}(X)$  y  $(X, d)$  un espacio métrico. Sea  $\epsilon_n = Cr^n$  para  $0 < r < 1$  con  $r \in \mathbb{R}$ ,  $C > 0$  y  $n = 1, 2, 3, \dots$ . Si

$$D = \lim_{n \rightarrow \infty} \left\{ \frac{\ln(N(A, \epsilon_n))}{\ln(\frac{1}{\epsilon_n})} \right\}$$

entonces  $A$  tiene dimensión fractal  $D$ .

*Demostración.* Sea  $r, C \in \mathbb{R}$ , y la sucesión  $E = \{\epsilon_n : n = 1, 2, 3, \dots\}$  definida como en el enunciado. Se define  $f(\epsilon) = \max \{\epsilon_n \in E : \epsilon_n \leq \epsilon\}$ . Supongamos que  $\epsilon \leq r$ . Entonces,

$$f(\epsilon) \leq \epsilon \leq \frac{f(\epsilon)}{r}$$

$$N(A, f(\epsilon)) \geq N(A, \epsilon) \geq N(A, \frac{f(\epsilon)}{r})$$

Como  $\ln(x)$  es una función positiva y creciente para  $x \geq 1$ , tenemos que:

$$\left\{ \frac{\ln(N(A, \frac{f(\epsilon)}{r}))}{\ln(\frac{1}{f(\epsilon)})} \right\} \leq \left\{ \frac{\ln(N(A, \epsilon))}{\ln(\frac{1}{\epsilon})} \right\} \leq \left\{ \frac{\ln(N(A, f(\epsilon)))}{\ln(\frac{r}{f(\epsilon)})} \right\} \quad (5.1)$$

Asumimos que  $N(A, \epsilon) \rightarrow \infty$  y  $\epsilon \rightarrow 0$ , sino el teorema no es cierto. Por un lado tenemos que:

$$\lim_{\epsilon \rightarrow 0} \left\{ \frac{\ln(N(A, f(\epsilon)))}{\ln(\frac{r}{f(\epsilon)})} \right\} = \lim_{n \rightarrow \infty} \left\{ \frac{\ln(N(A, \epsilon_n))}{\ln(\frac{r}{\epsilon_n})} \right\} = \lim_{n \rightarrow \infty} \left\{ \frac{\ln(N(A, \epsilon_n))}{\ln(r) + \ln(\frac{1}{\epsilon_n})} \right\} = \lim_{n \rightarrow \infty} \left\{ \frac{\ln(N(A, \epsilon_n))}{\ln(\frac{1}{\epsilon_n})} \right\}$$

Por otro lado:

$$\lim_{\epsilon \rightarrow 0} \left\{ \frac{\ln(N(A, \frac{f(\epsilon)}{r}))}{\ln(\frac{1}{f(\epsilon)})} \right\} = \lim_{n \rightarrow \infty} \left\{ \frac{\ln(N(A, \epsilon_{n-1}))}{\ln(\frac{1}{\epsilon_n})} \right\} = \lim_{n \rightarrow \infty} \left\{ \frac{\ln(N(A, \epsilon_{n-1}))}{\ln(\frac{1}{r}) + \ln(\frac{1}{\epsilon_{n-1}})} \right\} = \lim_{n \rightarrow \infty} \left\{ \frac{\ln(N(A, \epsilon_n))}{\ln(\frac{1}{\epsilon_n})} \right\}$$

Cuando  $\epsilon \rightarrow 0$ , el primer y el último término de (5.1) se aproximan al mismo valor. Por el teorema del sándwich el límite cuando  $\epsilon \rightarrow 0$  del término del medio existe y es igual al mismo valor. Esto completa la demostración.  $\square$

La definición 5.4 nos proporciona la explicación formal y matemática de la noción de dimensión de Hausdorff. Existe una definición alternativa cuyo resultado es exactamente el mismo. La diferencia reside en la interpretación de los elementos que intervienen en la definición.

**Definición 5.5.** Dado un fractal  $F$ , la dimensión fractal  $D_F$  viene dada por el cociente:

$$D_F = \frac{\ln N(l)}{\ln(\frac{1}{l})}$$

donde  $N(l)$  es el número de veces que se reproduce el objeto inicial y  $l$  es la escala en la que se encuentran.

A efectos prácticos, utilizaremos la definición 5.5 ya que resulta muy sencillo identificar los parámetros  $N(l)$  y  $l$  de un fractal. No obstante, conviene tener siempre presente el trasfondo de la definición 5.4 ya que en esta reside el significado más riguroso de dimensión fractal.

**Ejemplo 5.4 (Dimensión fractal del conjunto de Cantor).** En la figura 4.1 se representa la construcción del conjunto de Cantor. En cada etapa, cada segmento se reemplaza por dos de longitud  $\frac{1}{3}$ . Por tanto, en este caso,  $N(l) = 2$  y  $l = \frac{1}{3}$ . El conjunto de Cantor  $C$  tiene

## 5. Dimensión fractal

dimensión fractal:

$$D_F = \frac{\ln 2}{\ln 3} \approx 0.6309$$

**Ejemplo 5.5 (Dimensión fractal de la curva de Koch).** En la figura 4.3 se representa la construcción de la curva de Koch. En cada etapa, cada segmento se reemplaza por cuatro de longitud  $\frac{1}{3}$ . Por tanto, en este caso,  $N(l) = 4$  y  $l = \frac{1}{3}$ . La curva de Koch  $K$  tiene dimensión fractal:

$$D_F = \frac{\ln 4}{\ln 3} \approx 1.2619$$

**Ejemplo 5.6 (Dimensión fractal del cuadrado de Koch).** En la figura 4.5 se representa la construcción del cuadrado de Koch. En cada etapa, cada segmento se reemplaza por cinco de longitud  $\frac{1}{3}$ . Por tanto, en este caso,  $N(l) = 5$  y  $l = \frac{1}{3}$ . El cuadrado de Koch  $K'$  tiene dimensión fractal:

$$D_F = \frac{\ln 5}{\ln 3} \approx 1.4650$$

**Ejemplo 5.7 (Dimensión fractal del triángulo de Sierpiński).** En la figura 4.2 se representa la construcción del triángulo de Sierpiński. En cada etapa, se elimina el triángulo central de cada triángulo equilátero que lo forma. Los lados de los triángulos más pequeños miden la mitad que los del orginal y cada triángulo se reemplaza por tres. Por tanto, en este caso,  $N(l) = 3$  y  $l = \frac{1}{2}$ . El triángulo de Sierpiński  $S$  tiene dimensión fractal:

$$D_F = \frac{\ln 3}{\ln 2} \approx 1.5850$$

### 5.1. Dimensión box counting

Las dimensiones calculadas hasta ahora han sido para hipotéticos objetos fractales que no se encuentran en el mundo real. Mandelbrot anticipó que, la naturaleza presenta estas figuras de una forma más irregular y con una cierta autosimilitud. Ante la necesidad de medir la dimensión de estos objetos surgió el algoritmo **box counting** o **de conteo de cajas**.

Este algoritmo consiste en cubrir el fractal con cajas de diferentes tamaños, hacer un recuento del número de cajas que cubren el objeto y estimar la dimensión.

**Teorema 5.2 (Teorema box counting).** Sea  $A \in \mathcal{H}(\mathbb{R}^m)$ , donde  $(\mathbb{R}^m, d)$  es un espacio con la métrica Euclídea. Cubrimos  $\mathbb{R}^m$  con cajas cuadradas de longitud  $\frac{1}{2^n}$ . Sea  $N_n(A)$  el número de cajas de longitud  $\frac{1}{2^n}$  que intersectan con el atractor. Si

$$D = \lim_{n \rightarrow \infty} \left\{ \frac{\ln(N_n(A))}{\ln 2^n} \right\}$$

entonces  $A$  tiene dimensión box counting  $D$ .

*Demostración.* Observamos que para  $m = 1, 2, 3, \dots$

$$2^{-m}N_{n-1}(A) \leq N(A, \frac{1}{2^n}) \leq N_{k(n)}(A) \quad \text{para todo } n = 1, 2, 3, \dots$$

donde  $K(n)$  es el entero más pequeño  $k$  que satisface  $k \geq n - 1 + \frac{1}{\log_2 m}$ . La primera desigualdad se cumple porque una bola de radio  $\frac{1}{2^n}$  se intersecta como máximo con  $2^m$  cajas

de longitud  $\frac{1}{2^{n-1}}$ . La segunda se deriva del hecho de que una caja de lado  $s$  cabe dentro de una bola de radio  $r$  siempre que  $r^2 \geq (\frac{s}{2})^2 + (\frac{s}{2})^2 + \dots + (\frac{s}{2})^2 = m(\frac{s}{2})^2$  por el teorema de Pitágoras. Ahora

$$\lim_{n \rightarrow \infty} \left\{ \frac{\ln N_k(n)(A)}{\ln 2^n} \right\} = \lim_{n \rightarrow \infty} \left\{ \frac{\ln 2^{k(n)}}{\ln 2^n} \frac{\ln N_{k(n)}(A)}{\ln 2^{k(n)}} \right\} = D$$

debido a que  $\frac{k(n)}{n} \rightarrow 1$  cuando  $n \rightarrow \infty$  y ya que también

$$\lim_{n \rightarrow \infty} \left\{ \frac{\ln 2^{-m} N_{n-1}(A)}{\ln 2^n} \right\} = \lim_{n \rightarrow \infty} \left\{ \frac{\ln N_{n-1}(A)}{\ln 2^{n-1}} \right\} = D$$

El teorema 5.1 para  $r = \frac{1}{2}$  completa la demostración.  $\square$

A continuación, explicaremos el algoritmo de conteo de cajas para el cálculo de la dimensión de Hausdorff de imágenes fractales en  $\mathbb{R}^2$ . Utilizaremos la definición de dimensión sugerida en el teorema 5.2 para nuestro propósito. En el ejemplo 5.8 se detallan los pasos del algoritmo.

**Ejemplo 5.8 (Dimensión box counting del triángulo de Sierpiński ).** En este ejemplo, explicaremos paso a paso el método para calcular la dimensión fractal del triángulo de Sierpiński mediante el conteo de cajas en  $\mathbb{R}^2$ . Para empezar, cubrimos la estructura fractal con cajas cuadradas de diferentes dimensiones formando una cuadrícula.

El teorema 5.2 toma las longitudes de las cajas como  $\frac{1}{2^n}$ . Si intentamos aplicar lo enunciado en este teorema al caso práctico de una imagen, tenemos que entender estas longitudes como las proporciones entre las dimensiones de las cajas y la de la imagen. En este caso, tomaremos una imagen del triángulo de Sierpiński de 1280x979 píxeles y cajas de dimensiones  $2^k$  con  $k = 1, 2, \dots, 9$  (Figura 5.2). El resultado de contar las cajas que cubren el fractal es el siguiente:

Dimensión de la caja (píxeles)	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$
Número de cajas	5	15	48	160	470	1352	3991	10993

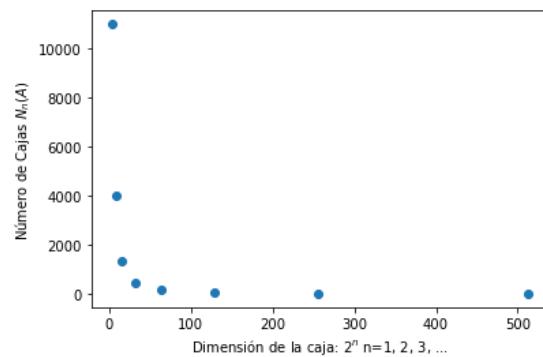


Figura 5.1.: [Python] Conteo de cajas para el triángulo de Sierpiński

## 5. Dimensión fractal

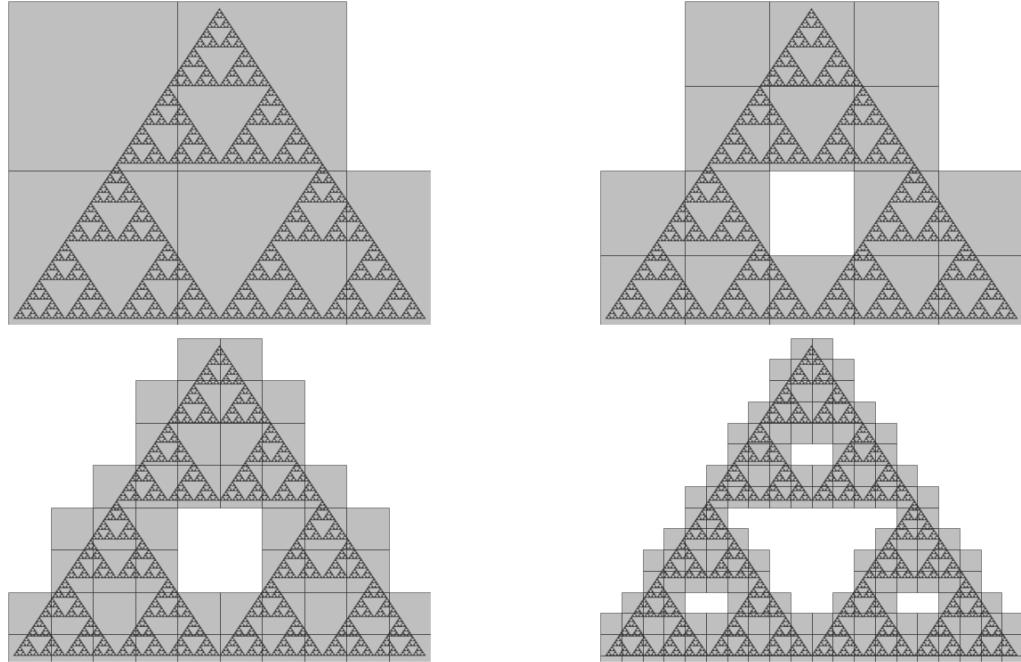


Figura 5.2.: [Python] Algoritmo box counting para el triángulo de Sierpiński con cajas de dimensiones (i)  $2^9$  (ii)  $2^8$  (iii)  $2^7$  (iv)  $2^6$  píxeles.

Posteriormente, aplicamos logaritmos a las dimensiones de las cajas y a los resultados de contarlas. Representamos los puntos y usamos el método de regresión de mínimos cuadrados para obtener la recta de mejor ajuste de esta gráfica logarítmica. El resultado de la dimensión fractal viene dado por la pendiente de esta recta.

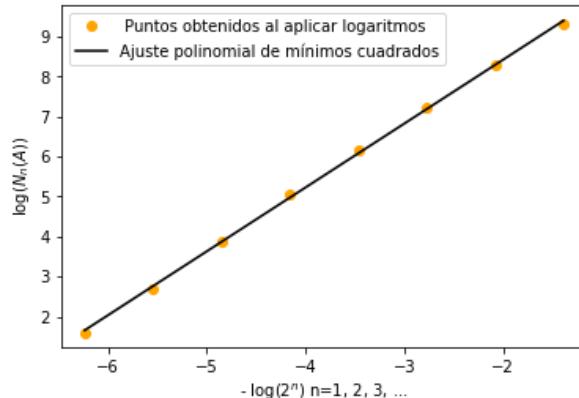


Figura 5.3.: [Python] Recta de mejor ajuste de la gráfica logarítmica para el triángulo de Sierpiński

La pendiente de la recta de mejor ajuste de la figura 5.3 y la dimensión box counting coinciden, su valor es 1.59. La dimensión fractal teórica se encuentra en el ejemplo 5.7.

## 6. Visualización de fractales con Ray Tracing

Por el momento, se han generado imágenes de fractales en 2D. A pesar de que esta dimensión nos permite apreciar la belleza de la geometría fractal, la tercera dimensión proporciona un añadido en su visualización. Por ello, este capítulo está dedicado a la visualización de fractales en 3D. En primer lugar, definiremos el algoritmo de ray tracing y lo aplicaremos a una esfera. Continuaremos con el ray marching y lo pondremos en práctica para generar el Mandelbulb, Mandelbox y distintos cuaterniones de Julia. Finalmente, evaluaremos los tiempos obtenidos y comentaremos de qué forma se han optimizado.

Las explicaciones del ray tracing y su aplicación a una esfera se encuentran recogidas en [Shi20], [Afl20] y [Buc19].

El **Ray Tracing** (o trazado de rayos) es una técnica de renderizado que puede simular de manera realista la iluminación de una escena y sus objetos al generar reflejos, refracciones, sombras e iluminación indirecta. Es capaz de producir imágenes con alto grado de realismo. Para explicar el algoritmo necesitamos configurar una escena con los siguientes elementos:

- **Espacio 3D:** utilizamos tres coordenadas para colocar los objetos en el espacio.
- **Objetos:** por ejemplo, una esfera.
- **Fuente de luz:** pueden ser uno o varios puntos que emiten luz en todas las direcciones.
- **Cámara:** punto de observación de la escena.
- **Imagen:** pantalla a través de la cual la cámara está observando.

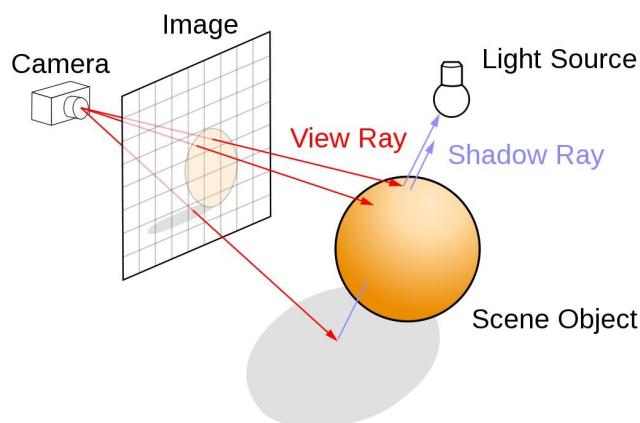


Figura 6.1.: [NVI] Conceptos básicos del trazado de rayos

## 6. Visualización de fractales con Ray Tracing

El trazado de rayos genera imágenes al rastrear el camino de luz desde la cámara (vista de la escena), a través del plano de visión 2D, hacia la escena 3D y de regreso a la fuente de luz. A medida que atraviesa la escena, la luz puede reflejarse de un objeto a otro (provocando reflejos), bloquearse por objetos (provocando sombras) o atravesar objetos transparentes (provocando refracciones). Todas estas interacciones se combinan para producir el color final y la iluminación de cada píxel que se muestra por pantalla.

Otra forma de pensar en el trazado de rayos es mirando a nuestro alrededor, ahora mismo. Los objetos que estamos viendo están iluminados por rayos de luz. Ahora seguimos el camino de esos rayos desde nuestro ojo hasta los objetos con los que interactúa la luz. Eso es trazado de rayos.

### 6.1. Aplicación del ray tracing a una esfera

Antes de describir las técnicas de representación de fractales en 3D, es conveniente comenzar con objetos más sencillos. En esta sección, aclararemos cuestiones relativas al algoritmo de trazado de rayos aplicado a una esfera. Estas consideraciones se tendrán en cuenta posteriormente en el contexto de la geometría fractal.

#### 6.1.1. Puntos y vectores de $\mathbb{R}^3$

Todo ray tracer (trazador de rayos) necesita representar objetos en un espacio tridimensional, es decir, en  $\mathbb{R}^3$ . Por tanto, es necesario conocer este espacio y los elementos que lo componen.

**Definición 6.1.** Un espacio vectorial es una terna  $(V, +, \cdot)$  donde  $V$  es un conjunto no vacío y  $+, \cdot$  son dos operaciones del tipo  $+ : V \times V \rightarrow V$ ,  $\cdot : \mathbb{R} \times V \rightarrow V$  a las que llamaremos ‘suma de vectores’ y ‘producto por escalares’ respectivamente y con las siguientes propiedades: denotando  $+(u, v) = u + v$  y  $\cdot(\lambda, v) = \lambda v$ ,

- (I)  $u + (v + w) = (u + v) + w \quad \forall u, v, w \in V$  (asociativa)
- (II)  $u + v = v + u \quad \forall u, v \in V$  (comutativa)
- (III) Existe  $e \in V$  tal que  $e + v = v + e = v, \forall v \in V$  (elemento neutro)
- (IV) Para cada  $v \in V$  existe  $w$  tal que  $v + w = w + v = e$  (elemento opuesto)
- (V)  $\lambda(\mu v) = (\lambda\mu)v, \forall v \in V, \forall \lambda, \mu \in \mathbb{R}$  (pseudo-asociativa)
- (VI)  $\lambda(u + v) = \lambda u + \lambda v$  y  $(\lambda + \mu)v = \lambda v + \mu v, \forall u, v \in V$  y  $\forall \lambda, \mu \in \mathbb{R}$  (distributiva)
- (VII)  $1v = v, \forall v \in V$  (unimodular)

De forma abreviada diremos que  $V$  es un espacio vectorial. A los elementos de  $V$  los llamamos vectores y a los de  $\mathbb{R}$ , escalares.

**Definición 6.2.** Dados dos vectores  $u = (u_1, u_2, u_3), v = (v_1, v_2, v_3) \in \mathbb{R}^3$  definimos la **suma de vectores** denotada por  $u + v$  como:

$$u + v = (u_1 + v_1, u_2 + v_2, u_3 + v_3)$$

## 6.1. Aplicación del ray tracing a una esfera

**Definición 6.3.** Dado un vector  $v = (v_1, v_2, v_3) \in \mathbb{R}^3$  y un escalar  $\lambda \in \mathbb{R}$  definimos el **producto por escalares** denotado por  $\lambda v$  como:

$$\lambda v = (\lambda v_1, \lambda v_2, \lambda v_3)$$

**Proposición 6.1.**  $\mathbb{R}^3$  con las operaciones definidas en 6.2 y 6.3 es un espacio vectorial.

Los vectores de  $\mathbb{R}^3$  serán la base de nuestro trazador de rayos e intervendrán en todos los cálculos. Es imprescindible conocer ciertas operaciones que nos serán de gran utilidad.

**Definición 6.4.** Dados dos vectores  $u = (u_1, u_2, u_3), v = (v_1, v_2, v_3) \in \mathbb{R}^3$  definimos su **producto escalar** denotado por  $u \cdot v$  como:

$$u \cdot v = u_1 v_1 + u_2 v_2 + u_3 v_3$$

**Definición 6.5.** Dados dos vectores  $u = (u_1, u_2, u_3), v = (v_1, v_2, v_3) \in \mathbb{R}^3$  definimos su **producto vectorial** denotado por  $u \times v$  como:

$$u \times v = (u_2 v_3 - u_3 v_2, u_3 v_1 - u_1 v_3, u_1 v_2 - u_2 v_1)$$

También es habitual escribir la igualdad anterior como el determinante:

$$u \times v = \begin{vmatrix} i & j & k \\ u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \end{vmatrix}$$

**Definición 6.6.** Dado un vector  $v = (v_1, v_2, v_3) \in \mathbb{R}^3$  definimos el **módulo** (o longitud) del vector denotado por  $\|v\|$  como:

$$\|v\| = \sqrt{v \cdot v} = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

**Definición 6.7.** Definimos la **normalización** de un vector como el proceso de convertirlo en unitario (módulo uno) manteniendo su dirección y sentido. Para normalizar un vector  $v = (v_1, v_2, v_3) \in \mathbb{R}^3$  se divide cada una de sus componentes por su módulo:

$$\hat{v} = \frac{v}{\|v\|}$$

### 6.1.2. Imágenes en formato PPM

Los puntos y los vectores pueden ser fundamentales para un trazador de rayos, pero sin una forma de convertirlos en algo visual, este carece de interés. En esta sección, hablaremos de cómo transformar escenas en imágenes que realmente se puedan ver. Para llegar hasta allí, discutiremos sobre los colores y cómo representarlos.

Las imágenes que vamos a mostrar se encuentran en **modo de color RGB**, es decir, cada píxel de la imagen está compuesto por una combinación de tres colores: rojo (Red), verde (Green) y azul (Blue). Si tomamos estos tres colores y los mezclamos en diferentes cantidades, obtenemos los demás. Las imágenes RGB contienen un total de 24 bits por píxel y usan tres canales de información, uno por cada color. El valor de los canales oscila entre 0 y 255

## 6. Visualización de fractales con Ray Tracing

en decimal o entre 00 y FF en hexadecimal. De esta forma los colores primarios se expresan de la siguiente forma:

	Decimal	Hexadecimal
Rojo	(255,0,0)	#FF0000
Verde	(0,255,0)	#00FF00
Azul	(0,0,255)	#0000FF

La forma más sencilla de escribir los valores de los píxeles es en un archivo. Existen multitud de formatos, en este caso utilizaremos ficheros **PPM** (Portable Pixel Map).

Expliquemos el formato PPM mediante la figura 6.2.

```
P3
3 2
255
255 0 0 0 255 0 0 0 255
255 255 0 255 255 255 0 0 0
```

Figura 6.2.: [Shi2o] Formato de imagen PPM

Cada archivo PPM comienza con un encabezado que consta de tres líneas de texto. La primera línea es la cadena *P3* (que es el identificador del formato PPM que estamos usando). La segunda línea (3 2) consiste en dos números que describen el ancho y alto de la imagen en píxeles. La tercera línea (255) especifica el valor máximo de color, lo que significa que cada valor rojo, verde y azul oscila entre 0 y 255. Inmediatamente después de este encabezado están los datos de los píxeles, representados por tres números enteros (RGB). Cada valor debe estar separado por un espacio.

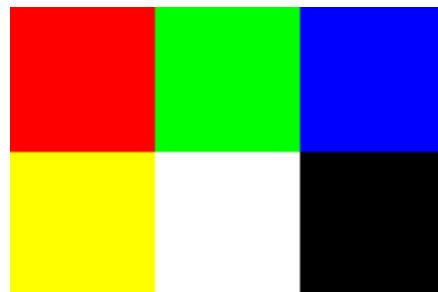


Figura 6.3.: [Shi2o] Imagen resultante del archivo PPM de la figura 6.2

### 6.1.3. Rayos

En el ray tracing, crear rayos y calcular la intersección de esos rayos con los objetos de la escena es muy importante para localizar el punto de impacto y producir el color correcto

para el píxel correspondiente.

Un rayo es en realidad otra forma de referirnos a una recta. Cada rayo tiene una posición de **origen**  $O$ , una posición de **destino**  $D$  y un vector **dirección**  $d$  que dice hacia dónde apunta. De esta forma, definimos un rayo como la recta:

$$r(t) = O + \frac{D - O}{\|D - O\|}t = O + dt \quad O, D, d \in \mathbb{R}^3, t \in \mathbb{R}$$

En el ray tracing, los rayos comienzan en la posición de la cámara ( $C$ ) y van en la dirección del píxel ( $P$ ) apuntado actualmente, por tanto:

$$r(t) = C + \frac{P - C}{\|P - C\|}t \quad P, C \in \mathbb{R}^3, t \in \mathbb{R}$$

#### 6.1.4. Intersección de rayos con esferas

Una **esfera** se define como el conjunto de puntos de  $\mathbb{R}^3$  que están a la misma distancia  $r$  (radio) de un punto dado  $C = (C_x, C_y, C_z) \in \mathbb{R}^3$  (centro).

$$S = \left\{ (x, y, z) \in \mathbb{R}^3 : (x - C_x)^2 + (y - C_y)^2 + (z - C_z)^2 = r^2 \right\}$$

Otra expresión válida es:

$$S = \left\{ P \in \mathbb{R}^3 : \|P - C\|^2 = r^2 \right\}$$

Conocemos la ecuación del rayo y sabemos que condición debe satisfacer un punto para que pertenezca a la esfera. Todo lo que tenemos que hacer es encontrar ese punto de intersección, en otras palabras, tenemos que calcular  $t \in \mathbb{R}$  para que  $r(t)$  esté en la esfera.

$$\begin{aligned} \|r(t) - C\|^2 &= r^2 \\ \|O + dt - C\|^2 &= r^2 \\ (dt + O - C) \cdot (dt + O - C) &= r^2 \\ t^2 d \cdot d + 2td \cdot (O - C) + (O - C) \cdot (O - C) &= r^2 \\ t^2 \|d\|^2 + 2td \cdot (O - C) + \|O - C\|^2 - r^2 &= 0 \end{aligned}$$

Se trata de una ecuación cuadrática ordinaria de la forma:

$$at^2 + bt + c = 0$$

donde

$$a = \|d\|^2 = 1$$

$$b = 2d \cdot (O - C)$$

$$c = \|O - C\|^2 - r^2$$

## 6. Visualización de fractales con Ray Tracing

Dado que  $d$  es un vector unitario, tenemos que  $a = 1$ . Una vez que calculamos el discriminante  $\Delta = b^2 - 4ac$ , hay tres posibilidades:

- $\Delta < 0$ : el rayo no interseca la esfera.
- $\Delta = 0$ : el rayo interseca la esfera en un punto.
- $\Delta > 0$ : el rayo interseca la esfera en dos puntos.

Por tanto, usaremos este criterio para detectar intersecciones. Nos quedaremos con la más cercana.

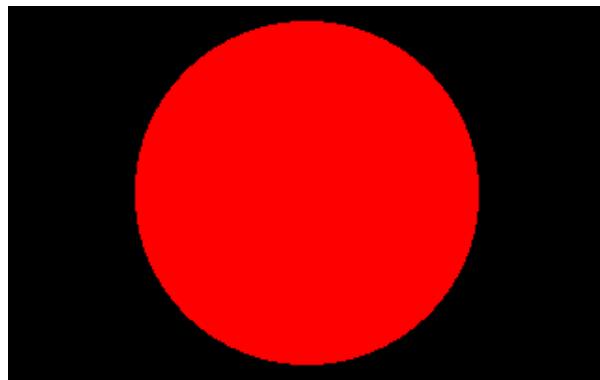


Figura 6.4.: [Python] Esfera mediante ray tracing básico

### 6.1.5. Iluminación y sombreado

Por el momento, hemos creado una esfera, mediante un trazador de rayos muy simple, cuyo realismo deja mucho que desear (Figura 6.4). En esta sección incluiremos la **iluminación** de la escena para conseguir un aspecto más natural y preciso en nuestras imágenes.

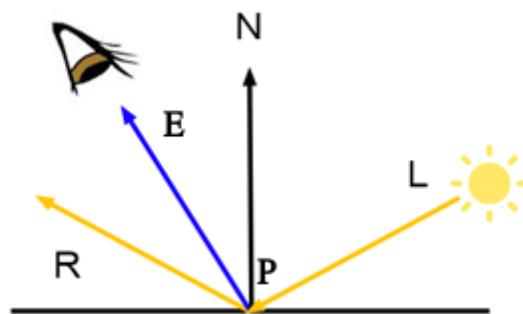


Figura 6.5.: [Scrb] Vectores que intervienen en la iluminación de una escena

## 6.1. Aplicación del ray tracing a una esfera

Sea  $P$  el punto de intersección entre el rayo y un objeto, los cuatro vectores de la figura 6.5 se definen:

- $E$  es el vector de la cámara, que apunta desde  $P$  al origen del rayo.
- $L$  es el vector de la luz, que apunta desde  $P$  a la fuente de luz.
- $N$  es el vector normal a la superficie, que es perpendicular a la superficie en  $P$ .
- $R$  es el vector de reflexión, que apunta a la dirección de la luz entrante al rebotar o reflejar.

Tenemos herramientas para calcular los dos primeros vectores:

- Para calcular  $E$ , cambiamos de signo el vector dirección del rayo girándolo para que apunte hacia atrás.
- Para calcular  $L$ , restamos  $P$  de la posición de la fuente de luz.

Antes de poder usar el vector normal en modelos de iluminación, lo definiremos y comentaremos cómo calcularlo en el caso de una esfera.

### 6.1.5.1. Normal a una superficie

La **normal a una superficie** es un vector con módulo uno que apunta perpendicularmente a una superficie en un punto dado. Una superficie plana tendrá la misma normal en todos los puntos de la misma. Las cosas se complican un poco cuando hablamos de superficies no planas, como la esfera.

#### Cálculo del vector normal a la esfera

Sea una esfera  $S$  con radio  $r$  y centro  $C$  y supongamos que deseamos encontrar el vector normal a un punto de la superficie  $P$ . Resulta que el vector que apunta desde el centro de la esfera al punto  $P$  es perpendicular a la superficie.

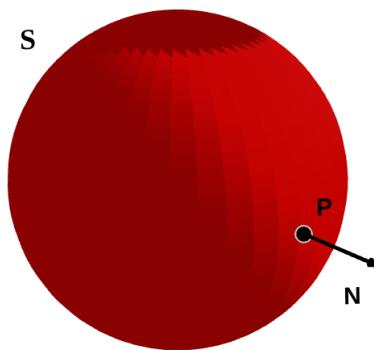


Figura 6.6.: [Mathcha] Normal a la esfera en un punto

## 6. Visualización de fractales con Ray Tracing

Para obtener el vector normal, solamente tenemos que normalizarlo.

$$N = \frac{P - C}{\|P - C\|}$$

### 6.1.5.2. Modelo de reflexión de Phong

Existen muchos algoritmos que pueden simular el reflejo de la luz, el que utilizaremos se llama **modelo de reflexión de Phong**. Fue desarrollado por **Tuong Phong** y simula la interacción entre tres tipos de iluminación:

- **Ambiental:** es la iluminación de fondo, se trata de la luz reflejada de otros objetos en el medio ambiente. El modelo de Phong la trata como constante.
- **Difusa:** es la luz reflejada desde una superficie. Depende del ángulo entre la fuente de luz y la superficie normal.
- **Especular:** es la reflexión de la propia fuente de luz y da como resultado un reflejo especular (punto brillante en una superficie). Depende del ángulo entre el vector de reflexión y el vector de la cámara. Está controlado por un parámetro al que llamaremos brillo, cuanto mayor sea el brillo más pequeño será el reflejo especular.

Dado que la iluminación ambiental es constante, nos centraremos en estimar las reflexiones difusa y especular mediante el modelo de sombreado de Lambert [Scra] y el modelo de sombreado de Blinn-Phong [Scrb] respectivamente.

#### Reflexión difusa. Modelo de sombreado de Lambert

El **modelo de sombreado de Lambert** es el más conocido en superficies difusas. Lo primero que debemos entender es de qué forma interactúa la luz sobre una superficie. Cuando sombrearemos un punto  $P$ , asumimos que en realidad no es un punto sino una superficie muy pequeña que llamamos **área diferencial** y que denotamos  $dA$ . Ahora, lo que realmente consideramos es la cantidad de luz que cae sobre la superficie de esta área alrededor de  $P$ . Esta luz no se reduce a un solo rayo ya que no nos interesa el punto singular sino la pequeña región alrededor de ese punto. La luz que cae sobre la región está contenida dentro de un volumen perpendicular a  $P$  (Figura 6.7).

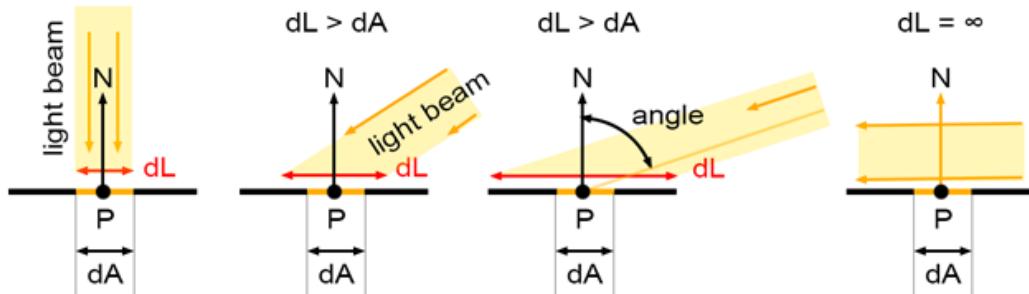


Figura 6.7.: [Scra] Iluminación del área diferencial  $dA$

Resulta fácil comprender que un punto de la superficie  $dA$  recibe un máximo de energía luminosa cuando el haz es perpendicular a la superficie, es decir, cuando la normal  $N$  y la dirección de la luz  $L$  son paralelas. También, podemos deducir que la superficie se oscurecerá a medida que el ángulo entre  $N$  y  $L$  aumenta. En última instancia, cuando  $N$  y  $L$  son perpendiculares,  $dA$  no recibe luz en absoluto. De manera más general, afirmamos que la luminosidad es proporcional a la relación de  $dA$  sobre  $dL$ . Dado que  $dL$ , sección transversal oblicua del haz de luz, aumenta a medida que lo hace el ángulo entre  $N$  y  $L$ , esta proporción disminuye a medida que aumenta el ángulo.

Este principio se conoce como la **Ley del coseno de Lambert**. La cantidad de luz que recibe una superficie es directamente proporcional al ángulo entre la normal de la superficie  $N$  y la dirección de la luz  $L$ .

$$\cos(\theta) = N \cdot L$$

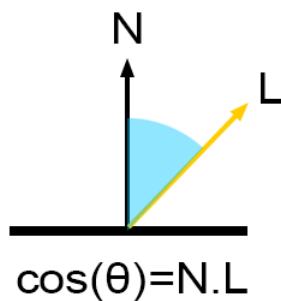


Figura 6.8.: [Scra] Ley del coseno de Lambert

La cantidad de luz que refleja una superficie y lo luminosa que parece depende directamente de la cantidad de luz que recibe. Por lo tanto, el color final de una superficie difusa es de alguna manera proporcional al coseno del ángulo entre  $N$  y  $l$ .

$$\text{Color superficie difusa} \propto \text{Luz incidente} \cdot \cos(\theta)$$

$$\propto \text{Luz incidente} \cdot N \cdot L$$

#### Reflexión especular. Modelo de reflexión de Blinn-Phong

El **modelo de reflexión de Blinn-Phong** se utiliza para generar reflejos especulares en superficies. Estos puntos brillantes se generan mediante la reflexión de la luz en una superficie donde los rayos incidentes se reflejan.

Phong observó que era posible simular este efecto calculando la dirección de reflexión ideal de un rayo y calculando el producto escalar entre este rayo reflejado y la dirección real de la vista.

$$\text{Reflexión Especular}_{Phong} \propto (E \cdot R)^n$$

donde  $E$  es el vector de la cámara,  $R$  es el vector del rayo reflejado y  $n$  es el coeficiente (o exponente) especular (Figura 6.5).

## 6. Visualización de fractales con Ray Tracing

El modelo de Phong es una aproximación excelente y muy eficiente de la iluminación, pero sus reflejos especulares se rompen en ciertas condiciones, específicamente cuando el valor del exponente es pequeño y el ángulo entre la vista y el vector de reflexión supera los 90°. Si es mayor de 90°, el resultado del producto escalar será negativo y por tanto el término especular se convertirá en 0. Podríamos pensar que esto no es un problema, porque no deberíamos ver ninguna luz por encima de los 90°. Sin embargo, esto solo se aplica a la iluminación difusa. En la iluminación specular, no medimos el ángulo entre la fuente de luz y la normal, sino entre la línea de visión y el vector de reflexión.

Por este motivo, **James F. Blinn** lo modificó presentando el modelo de **Blinn-Phong**. Blinn solucionó el problema de Phong mediante el cálculo de la bisectriz del ángulo que forma el vector de la cámara y el vector de la luz.

$$H = \frac{L + E}{\|L + E\|}$$

El resultado de aplicar el modelo de Blinn-Phong es el producto escalar entre el vector  $H$  y la normal a la superficie  $N$  elevado al coeficiente especular.

$$\text{Reflexión Especular}_{Blinn} \propto (H \cdot N)^n$$

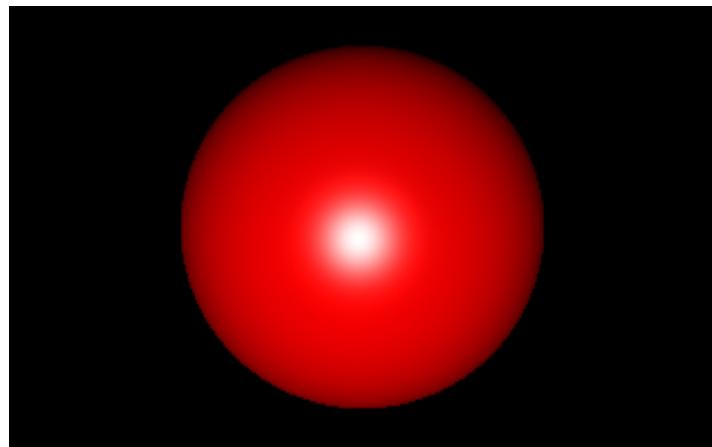


Figura 6.9.: [Python] Esfera mediante ray tracing con modelo de reflexión de Phong para iluminación y sombreado.

### 6.1.6. Creación de una escena con múltiples objetos

Nuestro trazador de rayos es capaz de dibujar una esfera simulando el comportamiento de la luz. Esto es solo el comienzo, se pueden crear escenas más complejas donde intervienen múltiples objetos.

En estas escenas, es necesario tener en cuenta que los rayos de la fuente de luz se pueden reflejar de unos objetos a otros. Para lograr estos reflejos tenemos que encontrar el rayo de reflexión y llamar recursivamente al trazador de rayos un determinado número de veces.

Según la figura 6.5, la dirección  $d$  y el origen  $O$  del rayo de reflexión  $R(t) = O + dt$  viene dado por:

$$d = L - 2(L \cdot N)N$$

$$O = P + N\delta$$

donde  $\delta > 0$ . El término  $N\delta$  se añade a la posición  $P$  para que el rayo de reflexión no comience exactamente en el mismo punto de intersección. Esto último se hace para prevenir que el algoritmo considere que se ha vuelto a intersecar el objeto.

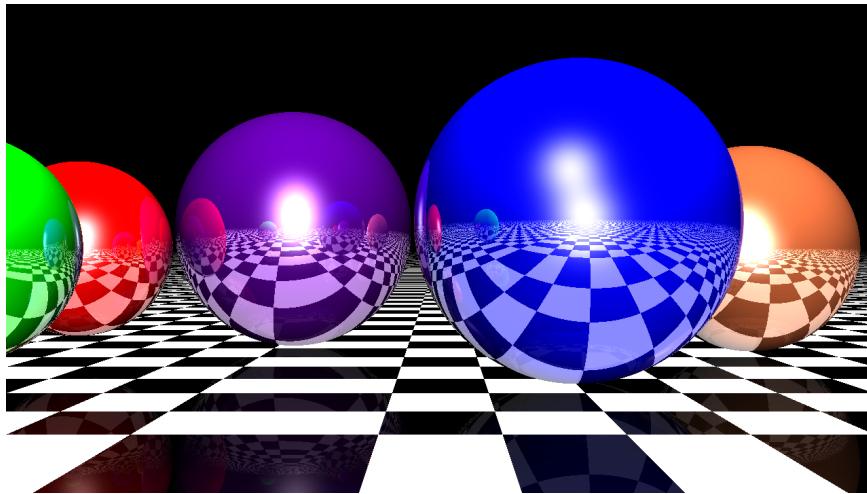


Figura 6.10.: [Python] Esferas sobre un tablero de ajedrez.

## 6.2. Introducción al ray marching y aplicación a fractales

El trazado de rayos clásico dispara uno (o más) rayos por píxel y calcula dónde se intersecan esos rayos con los objetos de la escena. Normalmente, estos objetos se describen matemáticamente de manera sencilla y sus puntos de intersección son fáciles de encontrar. Sin embargo, en el campo de la geometría fractal, las intersecciones con los rayos son difíciles de calcular, requieren mucho tiempo de cómputo y en algunos casos no existe analíticamente una forma de hallarlos. Para este problema existe una solución y es el algoritmo de ray marching [Chr11a].

El **ray marching** (o marcha de rayos) es una técnica con ciertas similitudes al ray tracing que renderiza objetos a través de campos de distancia. En lugar de calcular la intersección exacta entre el rayo de la cámara y el objeto, se avanza en pequeños pasos a lo largo del rayo y se verifica cómo de cerca se está del objeto. Cuando se está más próximo que un determinado umbral, el algoritmo se detiene. Para hacer esto, debemos tener una función que nos indique esta distancia del objeto llamada **estimador de la distancia** o SDF. El valor del estimador de la distancia nos dice cómo de grande es el paso que se nos permite marchar a lo largo del rayo, con la garantía de no golpear nada dentro de ese radio.

## 6. Visualización de fractales con Ray Tracing

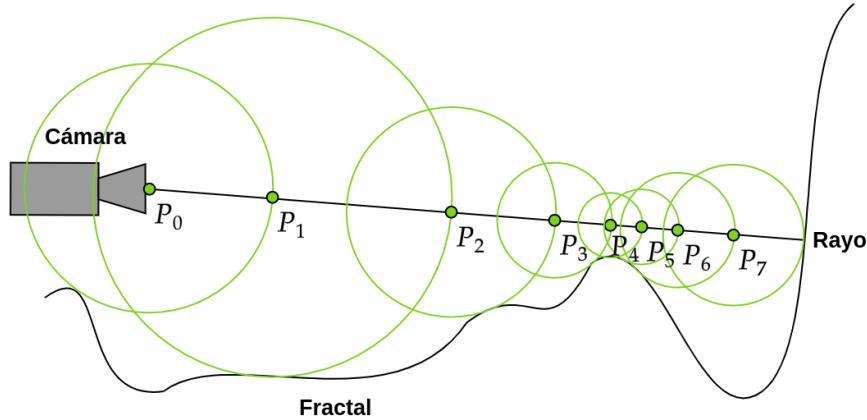


Figura 6.11.: [Mathcha] Marcha de rayos

En la figura 6.11,  $P_0$  es el punto de origen de la marcha de rayos (cámara). La distancia a la que se encuentra la superficie está ilustrada con la circunferencia centrada en  $P_0$ . Una vez obtenida, movemos el siguiente punto en la dirección del rayo, una distancia equivalente a la calculada. De esta forma obtenemos  $P_1$ . Repetimos el proceso, obteniendo  $P_2, P_3, \dots$  hasta que la distancia sea más pequeña que un determinado valor umbral.

Esta técnica funciona muy bien con objetos definidos mediante estimadores de distancia o SDF. Un claro ejemplo es la propia esfera cuyo estimador  $f: \mathbb{R}^3 \rightarrow \mathbb{R}$ :

$$f(p) = \|p - C\| - r \quad p, C \in \mathbb{R}^3(\text{centro}), r \in \mathbb{R}(\text{radio})$$

Si  $p$  pertenece a la esfera el valor del estimador es 0, si está en el interior el valor es negativo y si se encuentra en el exterior la distancia es positiva.

La estimación de la distancia suele ser menos costosa que el cálculo del punto de intersección, lo cual permite un renderizado en tiempo real. En la geometría fractal existen algunos objetos cuyo SDF es conocido como en el caso del **Mandelbulb**, **Mandelbox** y **cuaterniones de Julia**. El objetivo principal de este capítulo es conseguir un renderizado de alta calidad de estos fractales.

Antes de continuar, necesitamos poder calcular la normal de una superficie fractal para usarla en el modelo de iluminación de Phong (sección 6.1.5.2). El método que presentaremos se encuentra en el artículo de John C. Hart [Har89].

### 6.2.1. Aproximación de la normal de una superficie fractal

Las superficies de los fractales en 3D no son diferenciables y por tanto, no tienen definida una normal. Las normales se pueden aproximar examinando la relación de un punto de la superficie con su alrededor. Se han encontrado dos aproximaciones bastante útiles: **Z-buffer** y cálculo del **gradiente**. Utilizaremos el método del gradiente para el cálculo de la normal.

## Gradiente

Un método común para aproximar la normal es calcular el gradiente de un punto en la superficie. Se calcula de la siguiente forma:

$$N_x = D_{x+\epsilon,y,z} - D_{x-\epsilon,y,z}$$

$$N_y = D_{x,y+\epsilon,z} - D_{x,y-\epsilon,z}$$

$$N_z = D_{x,y,z+\epsilon} - D_{x,y,z-\epsilon}$$

donde  $\epsilon > 0$  y  $D_{x,y,z}$  es el valor del estimador de la distancia en el punto  $(x, y, z)$ . Dando como resultado el vector normal  $N = (N_x, N_y, N_z)$ .

### 6.2.2. Mandelbulb

El **Mandelbulb** es probablemente el fractal tridimensional más famoso que existe. Se construyó en 1997 por **Jules Ruis** y se desarrolló en 2009 por **Daniel White** y **Paul Nylander** utilizando coordenadas esféricas. Es una representación del conjunto de Mandelbrot en tres dimensiones.

Las páginas web personales de Paul Nylander [[Nylo9](#)] y Daniel White [[Whio9](#)] y el blog de Mikael Hvidtfeldt Christensen [[Chr11b](#)] serán de gran utilidad para las explicaciones de esta sección.

El conjunto de Madelbrot original (sección [3.2](#)) es un fractal bidimensional. Se ajusta a dos dimensiones porque los números complejos tienen dos componentes. La cuestión que queremos tratar está en cómo podemos extenderlo a tres dimensiones. ¿Existe un sistema numérico similar para tres dimensiones?

El mapa asociado al conjunto de Mandelbrot  $f_c(z) = z^2 + c$ ,  $z, c \in \mathbb{C}$  implica dos operaciones: sumar y elevar al cuadrado. Crear un espacio con números de n componentes donde sea posible sumar es fácil. Nos referiremos a esto como un espacio vectorial donde la suma de componentes funciona.

El mapa  $f_c(z)$  también implica elevar al cuadrado, lo que requiere un operador de multiplicación, un producto vectorial en el espacio vectorial. Un espacio vectorial con un producto vectorial (bilineal) es un álgebra y sus elementos se denominan **números hipercomplejos**.

En cuanto tratamos de crear un sistema numérico tridimensional, introduciendo una tercera componente  $j$ , nos percatamos de que no existe un análogo tridimensional del espacio bidimensional de los números complejos. Por consiguiente, no existe una representación canónica tridimensional del conjunto de Mandelbrot. Por otro lado, es posible construir conjuntos de Mandelbrot en cuatro dimensiones utilizando cuaterniones y números bicomplejos.

White y Nylander consideraron las propiedades geométricas de los números complejos para definir una especie de producto de elementos en el espacio tridimensional. Usando esta multiplicación el mapa  $f_c(z) = z^n + c$  nos lleva al Mandelbulb. La fórmula de White y

## 6. Visualización de fractales con Ray Tracing

Nylander para la potencia n-ésima del punto  $p = (x, y, z)$  es

$$p^n = r^n(\sin(n\theta) \cos(n\phi), \sin(n\theta) \sin(n\phi), \cos(n\theta))$$

donde

$$r = \|p\| \quad \theta = \text{atan2}(\|(x, y)\|, z) = \arccos\left(\frac{z}{r}\right) \quad \phi = \text{atan2}(y, x)$$

La función  $\text{atan2}(y, x)$  devuelve el valor de la  $\arctan\left(\frac{y}{x}\right)$  en radianes. Devuelve el ángulo correspondiente al formado por el eje x positivo y la recta que pasa por el origen y el punto  $(x, y)$ .

$$\text{atan2}(y, x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{si } x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{si } x < 0 \ y \geq 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & \text{si } x < 0 \ y < 0 \quad (x, y) \neq (0, 0) \\ \frac{\pi}{2} & \text{si } x = 0 \ y > 0 \\ -\frac{\pi}{2} & \text{si } x = 0 \ y < 0 \end{cases}$$

Como podemos ver, tenemos dos expresiones equivalentes para el ángulo  $\theta$ . Usando identidades trigonométricas es fácil comprobar que  $\text{atan2}(\|(x, y)\|, z) = \arccos\left(\frac{z}{r}\right)$ . Como  $\|(x, y)\| \geq 0$ , tenemos que:

$$\text{atan2}(\|(x, y)\|, z) = \begin{cases} \arctan\left(\frac{\|(x, y)\|}{z}\right) & \text{si } z > 0 \\ \arctan\left(\frac{\|(x, y)\|}{z}\right) + \pi & \text{si } z < 0 \quad (z, \|(x, y)\|) \neq (0, 0) \\ \frac{\pi}{2} & \text{si } z = 0 \end{cases}$$

- Si  $z = 0$ ,  $\text{atan2}(\|(x, y)\|, z) = \frac{\pi}{2} = \arccos(0)$ .
- Si  $z > 0$ , tenemos que  $\frac{\|(x, y)\|}{z} > 0$  y utilizando la identidad trigonométrica:

$$\arctan(x) = \arccos\left(\frac{1}{\sqrt{1+x^2}}\right) \quad x > 0$$

para  $x = \frac{\|(x, y)\|}{z}$  la igualdad queda comprobada.

- Si  $z < 0$ , tenemos que  $\frac{\|(x, y)\|}{z} < 0$  y utilizando las identidades trigonométricas:

$$\pi + \arctan(x) = \operatorname{arccot}\left(\frac{1}{x}\right) \quad x < 0$$

$$\operatorname{arccot}(x) = \arccos\left(\frac{x}{\sqrt{1+x^2}}\right) \quad x \in \mathbb{R}$$

para  $x = \frac{\|(x, y)\|}{z}$  la igualdad queda comprobada.

**Estimador de la distancia para el Mandelbulb**

Dado que el Mandelbulb es un objeto tridimensional, el algoritmo de marcha de rayos utilizará volúmenes como esferas o elipsoides en lugar de circunferencias (Figura 6.11). Trataremos de encontrar un estimador de la distancia, que nos diga cómo marchar a lo largo del rayo con la garantía de no golpear la frontera del fractal. En otras palabras, encontraremos los radios de estos volúmenes. **John C. Hart** en su artículo [Har89] asegura que para la familia cuadrática, tenemos la siguiente aproximación:

$$d(z) = \frac{|f^n(z)|}{2|f'^n(z)|} \log |f^n(z)|$$

---

**Algoritmo 1 Estimador de la distancia para el Mandelbulb**

---

**Entrada:** *pos, Potencia***Salida:** *DE*

```

1: z  $\leftarrow$  pos
2: dr  $\leftarrow$  1.0
3: r  $\leftarrow$  0.0
4: para i  $\leftarrow$  0, Iteraciones hacer
5:   r  $\leftarrow \|z\|
6:   si r  $>$  2.0 entonces
    break
7:   fin si
8:    $\theta \leftarrow \text{arc cos}\left(\frac{z.z}{r}\right)$ 
9:    $\phi \leftarrow \text{atan2}(z.y, z.x)$ 
10:  dr  $\leftarrow r^{\text{Potencia}-1.0} \times \text{Potencia} \times dr + 1.0$ 
11:  zr  $\leftarrow r^{\text{Potencia}}$ 
12:   $\theta \leftarrow \theta \times \text{Potencia}$ 
13:   $\phi \leftarrow \phi \times \text{Potencia}$ 
14:  z  $\leftarrow zr \times (\sin(\theta) \cos(\phi), \sin(\phi) \sin(\theta), \cos(\theta))$ 
15:  z  $\leftarrow z + pos$ 
16: fin para
17: DE  $\leftarrow 0.5 \times \log(r) \times \frac{r}{dr}$ 
18: devolver DE$ 
```

---

## 6. Visualización de fractales con Ray Tracing

Paul Nylander sugirió usar esta misma aproximación para las distintas potencias del mapa  $f_c(z) = z^n + c$ . En ese instante surgió el Mandelbulb, vinculado a la octava potencia (para  $n = 8$ ).

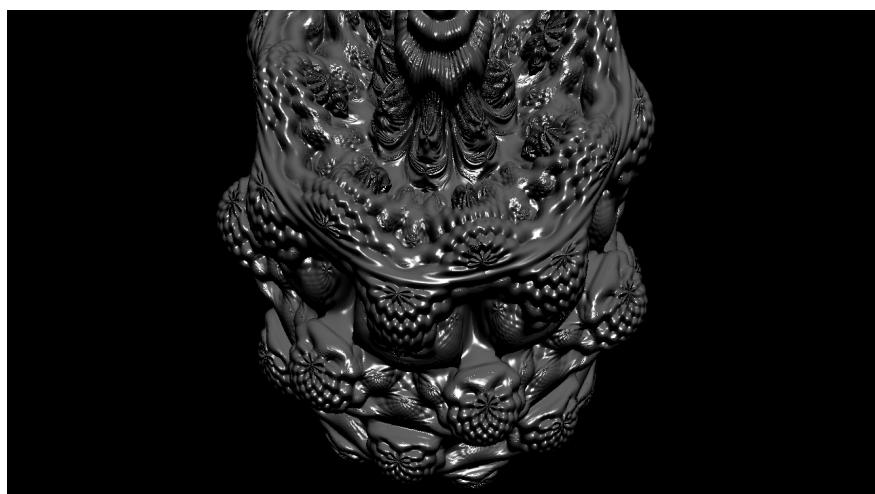
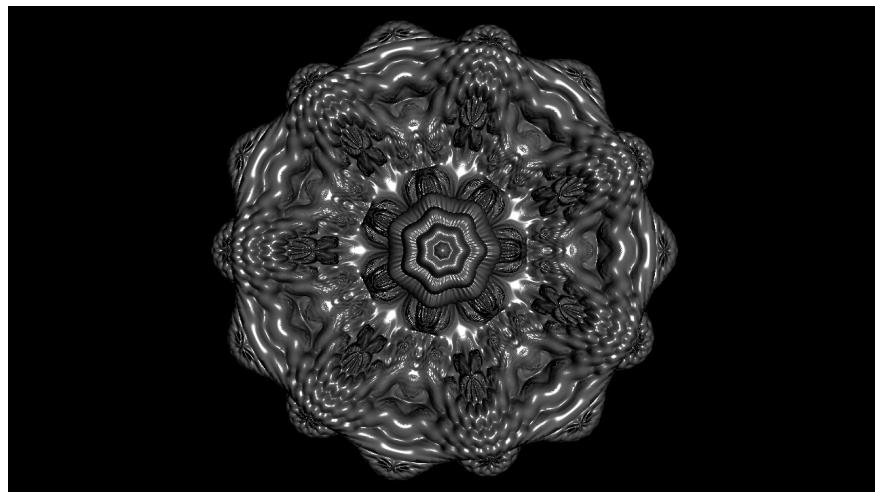


Figura 6.12.: [Python] Mandelbulb

En el apéndice A, se pueden ver más imágenes de las representaciones tridimensionales de las distintas potencias del mapa  $f(z) = z^n + c$ .

### 6.2.3. Juliabulb

Los cuaterniones de Julia se construyen según el mismo principio que el conjunto de Julia tradicional. Utiliza números complejos de cuatro dimensiones en lugar de complejos de dos. Un número complejo se escribe como  $z = r + ai$  donde  $i^2 = -1$ . Un cuaternion tiene dos componentes complejas más y se escribe como  $q = r + ai + bj + ck$  donde  $r, a, b, c \in \mathbb{R}$ . Existen relaciones un poco más complicadas entre  $i, j$  y  $k$ .

$$i^2 = j^2 = k^2 = -1$$

$$\begin{aligned} ij &= k & jk &= i & ki &= j \\ ji &= -k & kj &= -i & ik &= -j \end{aligned}$$

El conjunto de los cuaterniones puede expresarse como:

$$\mathbb{Q} = \{r + ai + bj + ck : r, a, b, c \in \mathbb{R}\}$$

**Definición 6.8.** Dados dos cuaterniones  $q_1 = r_1 + a_1i + b_1j + c_1k, q_2 = r_2 + a_2i + b_2j + c_2k \in \mathbb{Q}$  definimos la **suma de cuaterniones** denotada por  $q_1 + q_2$  como:

$$q_1 + q_2 = r_1 + r_2 + (a_1 + a_2)i + (b_1 + b_2)j + (c_1 + c_2)k$$

**Definición 6.9.** Dados dos cuaterniones  $q_1 = r_1 + a_1i + b_1j + c_1k, q_2 = r_2 + a_2i + b_2j + c_2k \in \mathbb{Q}$  definimos el **producto de cuaterniones** denotado por  $q_1 \cdot q_2$  como:

$$\begin{aligned} q_1 \cdot q_2 &= (r_1r_2 - a_1a_2 - b_1b_2 - c_1c_2) + (r_1a_2 + a_1r_2 + b_1c_2 - c_1b_2)i \\ &\quad + (r_1b_2 + b_1r_2 + c_1a_2 - a_1c_2)j + (r_1c_2 + c_1r_2 + a_1b_2 - b_1a_2)k \end{aligned}$$

El producto entre cuaterniones es asociativo pero no es commutativo.

**Definición 6.10.** Dado un cuaternion  $q = r + ai + bj + ck \in \mathbb{Q}$  definimos el **módulo** (o longitud) del cuaternion denotado por  $\|q\|$  como:

$$\|q\| = \sqrt{r^2 + a^2 + b^2 + c^2}$$

Dado que es bastante difícil dibujar objetos de cuatro dimensiones, necesitamos una forma de representar cuaterniones en una pantalla 2D. El enfoque utilizado es interseccar el fractal con un hiperplano, en esencia, esto hace que una de las componentes del cuaternion dependa de las otras tres. Usaremos el hiperplano  $c = 0$  para renderizar distintos cuaterniones de Julia.

#### Estimador de la distancia para el Juliabulb

Partiendo del artículo [Har89] de **John C. Hart** y de las explicaciones dadas para el estimador de la distancia del Mandelbulb presentamos el algoritmo para estimar la distancia del Juliabulb. Usaremos la aproximación de **Iñigo Quilez** [Qui].

## 6. Visualización de fractales con Ray Tracing

---

### Algoritmo 2 Estimador de la distancia para los cuaterniones de Julia

---

**Entrada:**  $pos, c$

**Salida:**  $DE$

```
1:  $z \leftarrow (pos.x, pos.y, pos.z, 0.0)$ 
2:  $dr \leftarrow 1.0$ 
3:  $r \leftarrow \|z\|^2$ 
4: para  $i \leftarrow 0$ , Iteraciones hacer
5:    $dr \leftarrow dr \times 4.0 \times r$ 
6:    $z \leftarrow z^2 + c$ 
7:    $r \leftarrow \|z\|^2$ 
8:   si  $r > 4.0$  entonces
     break
9:   fin si
10:  fin para
11:   $DE \leftarrow 0.25 \times \log(r) \times \sqrt{\frac{r}{dr}}$ 
12:  devolver  $DE$ 
```

---

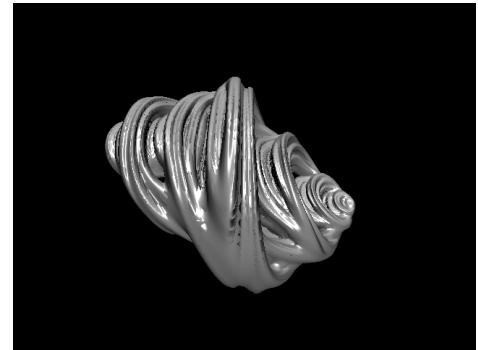
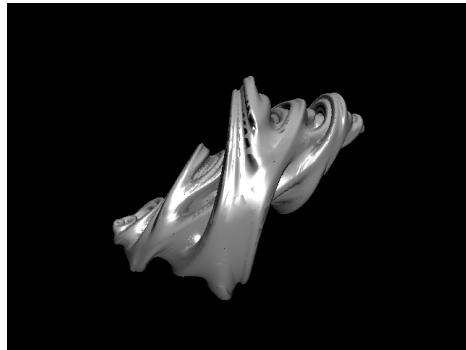
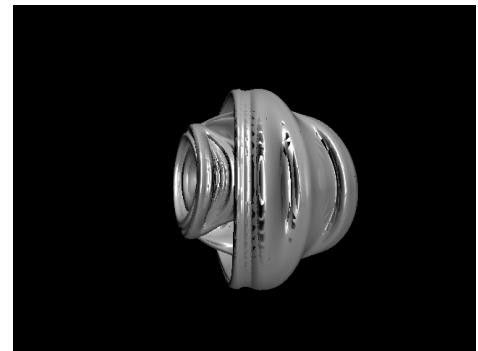
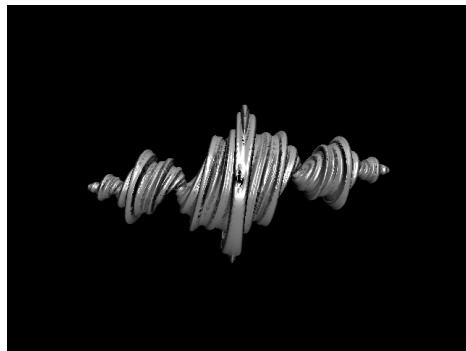


Figura 6.13.: [Python] Juliabulb (i)  $c = (-1, 0.2, 0, 0)$  (ii)  $c = (-0.162, 0.163, 0.560, -0.599)$  (iii)  $c = (-0.2, 0.8, 0, 0)$  (iv)  $c = (-0.125, -0.256, 0.847, 0.0895)$

En el apéndice B, se pueden encontrar más representaciones de cuaterniones.

Si intersecamos progresivamente un cuaternión de Julia con distintos hiperplanos, obtenemos una animación formada por una secuencia de imágenes. A continuación, se muestran 9 cortes del cuaternión  $c = (-0.08, 0.0, -0.8, -0.03)$  obtenidos a partir del desplazamiento de un hiperplano a lo largo de uno de los ejes en pasos de 0.1.

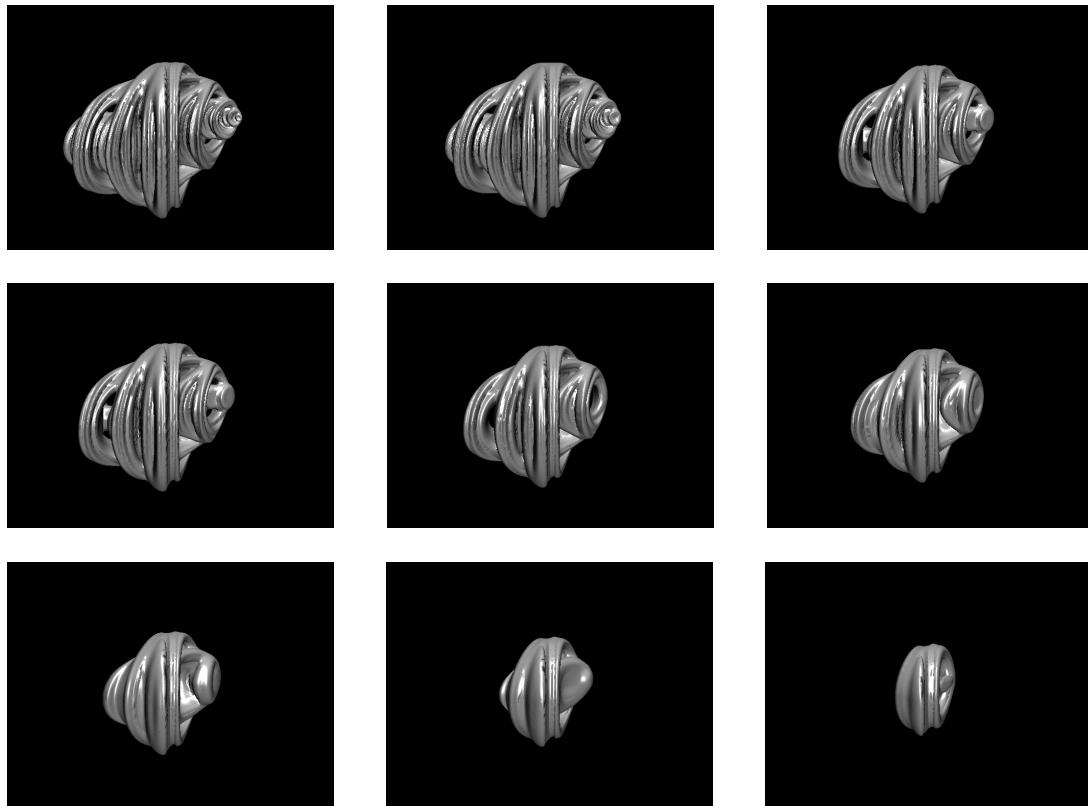


Figura 6.14.: [Python] Animación del cuaternión de Julia  $c = (-0.08, 0.0, -0.8, -0.03)$

#### 6.2.4. Mandelbox

El mandelbox es un fractal con forma de caja que fue presentado por **Tom Lowe** en 2010. A diferencia del Mandelbulb, el mandelbox usa una transformación un poco más compleja:

$$z_{n+1} = \text{scale} \times \text{sphereFold}(\text{boxFold}(z_n)) + c$$

El interés de este fractal es sobre todo visual, nos limitaremos a dar un SDF para su representación en lugar de comentar su construcción.

##### Estimador de la distancia para el Mandelbox

Usaremos la aproximación de **Mikael Hvidtfeldt Christensen** [[Chr11c](#)].

6. Visualización de fractales con Ray Tracing

---

**Algoritmo 3 Estimador de la distancia para el Mandelbox**

---

**Entrada:**  $pos$ ,

**Salida:**  $DE$

```
1:  $offset \leftarrow pos$ 
2:  $dr \leftarrow 1.0$ 
3:  $scale \leftarrow 2.0$ 
4: para  $i \leftarrow 0$ , Iteraciones hacer
5:    $pos, dr \leftarrow boxFold(pos, dr)$ 
6:    $pos, dr \leftarrow sphereFold(pos, dr)$ 
7:    $pos \leftarrow scale \times pos + offset$ 
8:    $dr \leftarrow dr \times |scale| + 1.0$ 
9: fin para
10:  $r \leftarrow \|pos\|$ 
11:  $DE \leftarrow \frac{r}{|dr|}$ 
12: devolver  $DE$ 
```

---

---

**Algoritmo 4 boxFold**

---

**Entrada:**  $pos, dr$

**Salida:**  $p, dr$

```
1:  $px \leftarrow max(min(pos.x, 1.0), -1.0)$ 
2:  $py \leftarrow max(min(pos.y, 1.0), -1.0)$ 
3:  $pz \leftarrow max(min(pos.z, 1.0), -1.0)$ 
4:  $pxyz \leftarrow (px, py, pz)$ 
5:  $p \leftarrow pxyz \times 2.0 - pos$ 
6: devolver  $p, dr$ 
```

---

---

**Algoritmo 5 sphereFold**

---

**Entrada:**  $pos, dr$

**Salida:**  $pos, dr$

```
1:  $r2 \leftarrow \|pos\|^2$ 
2: si  $r2 < 0.5$  entonces
3:    $temp \leftarrow 2.0$ 
4:    $pos \leftarrow pos \times temp$ 
5:    $dr \leftarrow dr \times temp$ 
6: si no, si  $r2 < 1.0$  entonces
7:    $temp \leftarrow \frac{1.0}{r2}$ 
8:    $pos \leftarrow pos \times temp$ 
9:    $dr \leftarrow dr \times temp$ 
10: fin si
11: devolver  $p, dr$ 
```

---

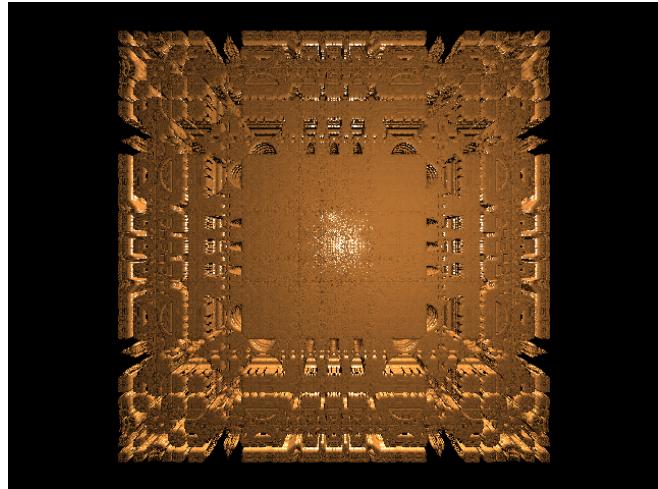


Figura 6.15.: [Python] Mandelbox

### 6.3. Paralelismo basado en procesos y tiempos de renderizado

Hasta el momento, se han dado una serie de explicaciones con las que somos capaces de construir un trazador de rayos complementariamente funcional. Sin embargo, no hemos tenido en cuenta un aspecto muy importante y es el tiempo de renderizado. En esta sección, veremos de qué forma se puede mejorar.

Cuando ejecutamos un programa en python, por defecto solo utiliza un núcleo del procesador. Esto significa que el resto de núcleos no tienen ninguna tarea asignada por parte del trazador de rayos. Si seguimos este paradigma, obtendremos peores tiempos de renderizado. Para optimizarlos utilizaremos todos los núcleos de nuestra CPU, en otras palabras, haremos uso del paralelismo [Pytb].

Para paralelizar nuestro trazador de rayos repartiremos el trabajo de renderizado en diferentes procesos que se ejecutarán en los núcleos de nuestro procesador. Para ello, dividiremos el número de píxeles de alto de la imagen a renderizar entre los distintos procesos. Dicho de otra manera, cada núcleo se encargará de renderizar unas determinadas filas de píxeles de la imagen. Una vez que terminen todos los procesos, bastará con agrupar todo el trabajo realizado por separado en una sola imagen.

## 6. Visualización de fractales con Ray Tracing

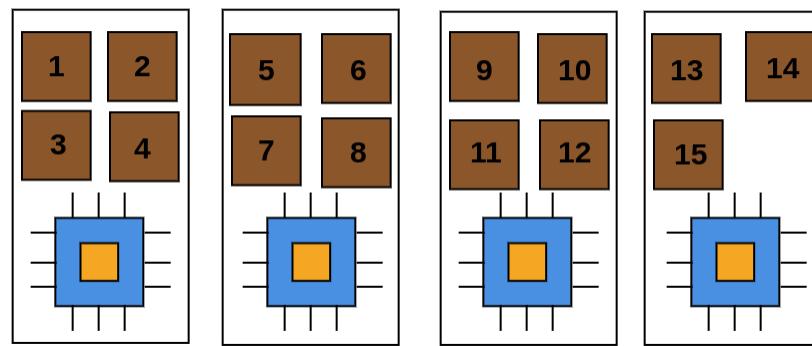


Figura 6.16.: [Mathcha] División de 15 filas de píxeles entre 4 núcleos

Para demostrar que con esta idea se logran mejores resultados, que mejor forma que ponerla en práctica y analizar los tiempos obtenidos. Para la siguiente prueba se ha utilizado un portátil **MSI GP62 7RE Leopard Pro** con un procesador **Intel(R) Core(TM) i7-7700HQ 2.80GHz** de **8 núcleos**. Las imágenes se han renderizado con alta calidad **HD** y parámetros de error muy ajustados para ver de qué es capaz el trazado de rayos. Para optimizar al máximo la parallelización se han lanzado 8 procesos, debido a que la CPU dispone de 8 núcleos. Compararemos estos resultados con los adquiridos en la ejecución con un solo proceso.

	1 proceso	8 procesos
Esferas (Figura 6.10)	1min 13s	24s
Mandelbulb (Figura 6.12)	5min 30s	1m 46s
Juliabulb(Figura 6.13(i))	13min 5s	4min 40s
Mandelbox(Figura 6.15)	1h 2min	20min 15s

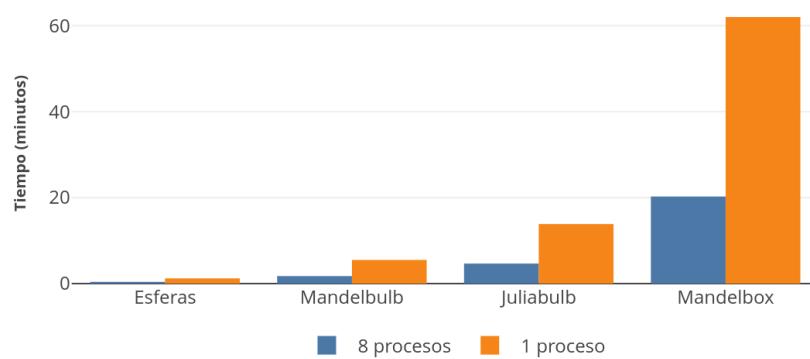


Figura 6.17.: [Mathcha] Comparativa de tiempos de renderizado

### *6.3. Paralelismo basado en procesos y tiempos de renderizado*

Las resoluciones utilizadas para este experimento no son las que usa la aplicación por defecto. El programa ajusta la resolución de las imágenes para que el usuario pueda obtener imágenes fractales en apenas 2 minutos, sin grandes esperas. Por eso, se reducirán las resoluciones del Juliabulb y Mandelbox a SD y QVGA respectivamente.



## 7. Conclusiones y trabajo futuro

En este último capítulo se van a exponer las conclusiones obtenidas a lo largo del desarrollo del trabajo. Además, se comentarán algunas propuestas y trabajos futuros para continuar con este estudio.

Los objetivos principales de este trabajo son: analizar y estudiar los sistemas dinámicos discretos, la dinámica compleja y los fractales, implementar programas que sirvan de complemento a la teoría descrita y visualizar fractales en 3D. Recapitulando todo el trabajo realizado, podemos asegurar que los objetivos inicialmente propuestos se han cumplido.

El campo de los sistemas dinámicos se complementa muy bien con la informática. Las herramientas informáticas nos son de gran ayuda para modelizar sistemas, estudiar el comportamiento de los mapas, calcular graficamente puntos fijos y periódicos, analizar óbitas,... Nos permite observar la estabilidad de puntos del sistema y estudiar comportamientos caóticos. Por ello, su uso es realmente beneficioso en este ámbito.

En el mundo fractal, tanto las matemáticas como la informática juegan un papel muy importante. Los conjuntos de Julia y de Mandelbrot nacieron de la iteración de una función compleja. Analizando matemáticamente este mapa complejo podemos comprender cómo se comporta bajo ciertas circunstancias y estudiar todas sus propiedades. Los fractales se construyen mediante procesos recursivos, sistemas de Lindenmayer, sistemas de funciones iteradas, ... Todos estos conceptos ayudan a entender cómo se originaron y de qué forma se construyen. Dado que la dimensión topológica es la más utilizada, resulta muy interesante la aparición de la dimensión de Hausdorff para la dimensión de los fractales.

A lo largo de los años, la geometría fractal ha generado cada vez más interés. Esto, en parte, ha sido gracias al uso de la informática en la visualización de estos objetos. Las representaciones de los conjuntos de Julia y de Mandelbrot y de fractales no pasan desapercibidas para nadie. Las razones de su atractivo reside en la curiosidad que causan en la persona que las está observando.

Con este trabajo, nos percatamos del potencial que tiene el ray tracing en la visualización de objetos en 3D. Gracias a este algoritmo, las escenas se ven más realistas y nos permiten representar fractales de alta calidad. Los estimadores de la distancia ayudan a encontrar la distancia a la superficie fractal y a renderizar estos en tiempo real.

Como futuros trabajos, sería de gran interés considerar mejoras y nuevas implementaciones en el proyecto de ray tracing. El siguiente paso sería crear un controlador para poder desplazar y rotar el punto de visión (cámara), aumentar y disminuir la calidad de renderizado,... durante la ejecución. También podríamos crear animaciones, con el continuo movimiento de la cámara permitiendo la visualización completa desde todos los ángulos y perspectivas. Esto último permitiría sumergirnos en el propio fractal y observarlo desde todos los rincones. Para crear animaciones de los cuaterniones de Julia, sería necesario un

## *7. Conclusiones y trabajo futuro*

estudio más profundo de los hiperplanos de corte. Respecto a los conjuntos de Julia y Mandelbrot podríamos representarlos de tal manera que fuera posible realizar zoom para verlos con más detalle.

Si vamos más alla, sería interesante no solo ser capaces de visualizarlos sino también de poseerlos en el mundo real. Con esto me refiero a la posibilidad de crear modelos tridimensionales de los fractales e imprimirlos con una impresora 3D.

## A. Representación tridimensional de las distintas potencias del mapa $f(z) = z^n + c$

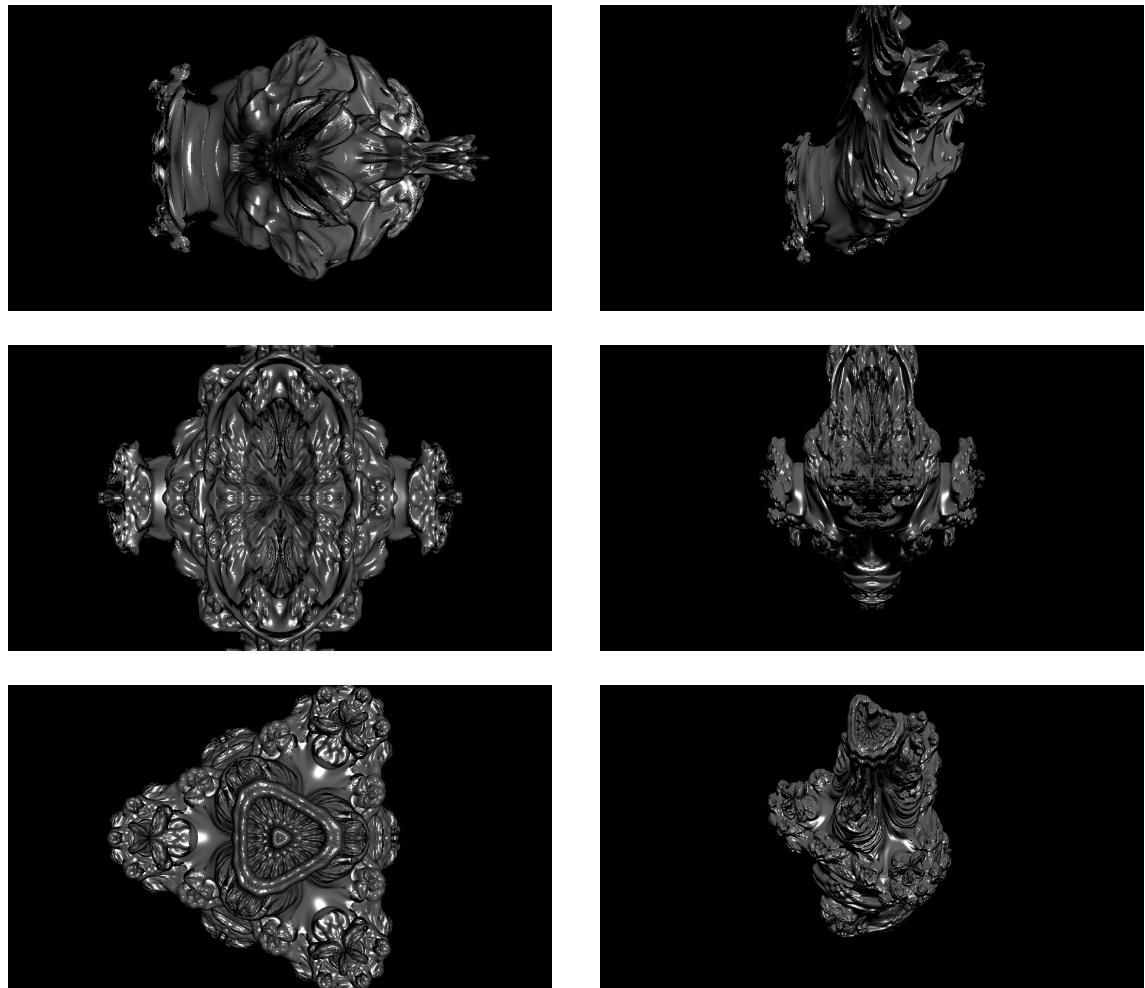


Figura A.1.: [Python] Representación tridimensional del mapa  $f_c(z) = z^n + c$  para (i)(ii)  $n = 2$  (iii)(iv)  $n = 3$  (v)(vi)  $n = 4$

A. Representación tridimensional de las distintas potencias del mapa  $f(z) = z^n + c$

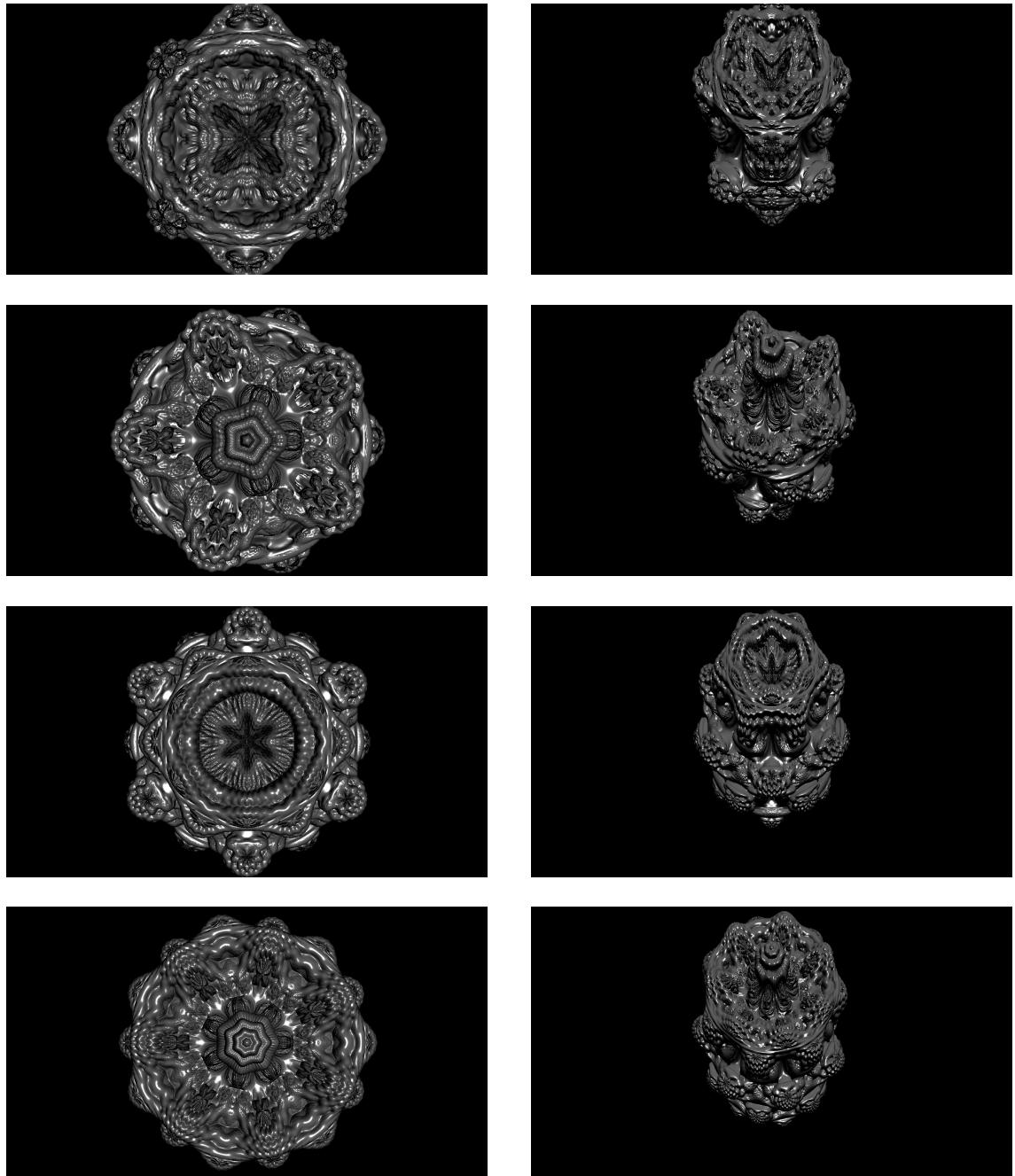


Figura A.2.: [Python] Representación tridimensional del mapa  $f_c(z) = z^n + c$  para (i)(ii)  $n = 5$  (iii)(iv)  $n = 6$  (v)(vi)  $n = 7$  (vii)(viii)  $n = 8$  (Mandelbulb)

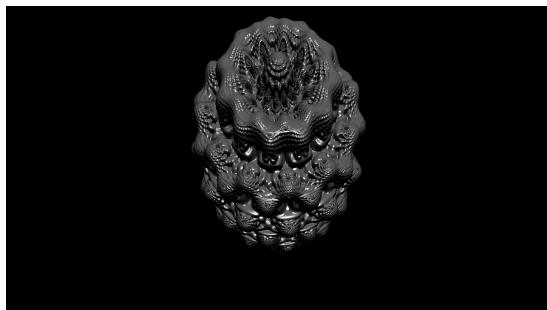
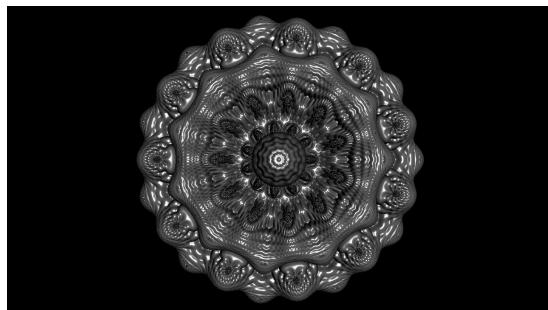
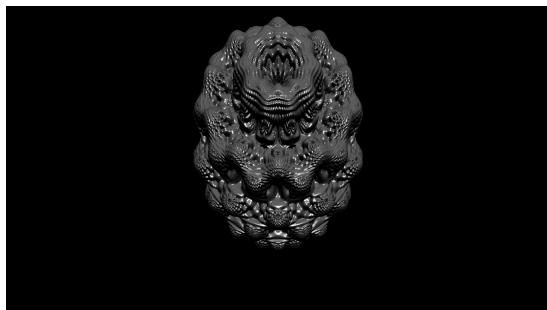
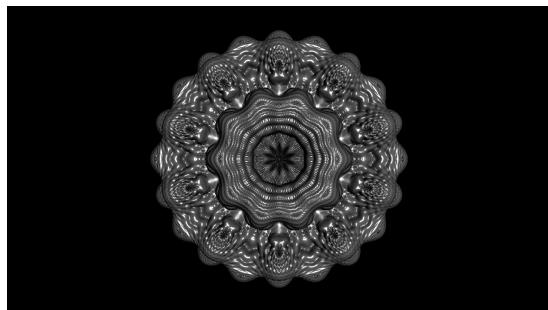
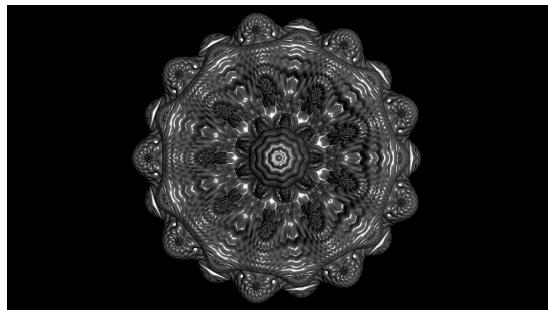
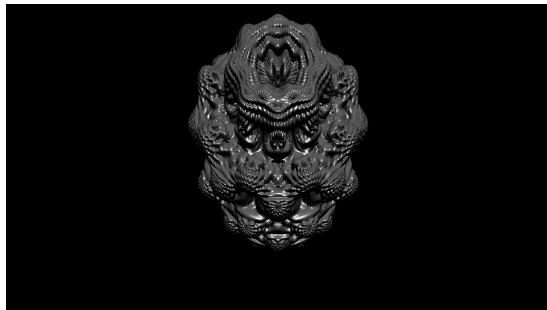
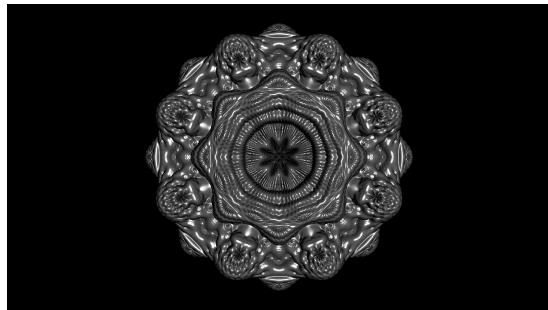


Figura A.3.: [Python] Representación tridimensional del mapa  $f_c(z) = z^n + c$  para (i)(ii)  $n = 9$  (iii)(iv)  $n = 10$  (v)(vi)  $n = 11$  (vii)(viii)  $n = 12$



## B. Cuaterniones de Julia

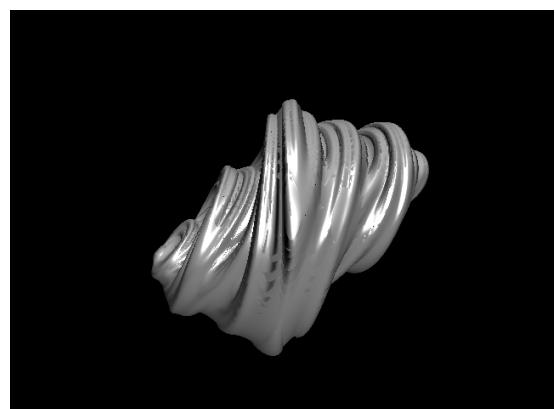
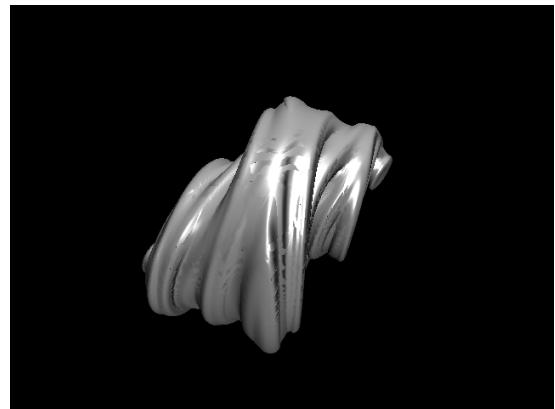
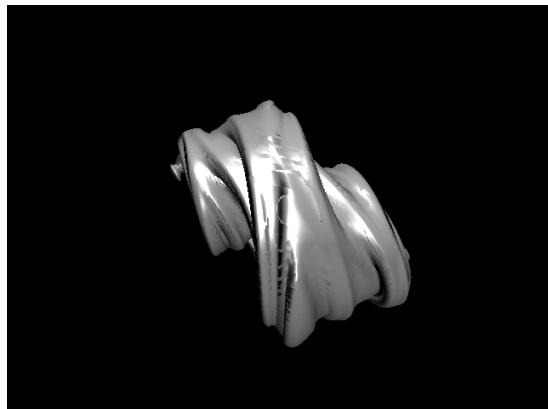


Figura B.1.: [Python] Cuaterniones de Julia (i)  $c = (-0.291, -0.399, 0.339, 0.437)$  (ii)  $c = (-0.2, 0.4, -0.4, -0.4)$  (iii)  $c = (-0.213, -0.0410, -0.563, -0.560)$  (iv)  $c = (-0.2, 0.6, 0.2, 0.2)$

B. Cuaterniones de Julia

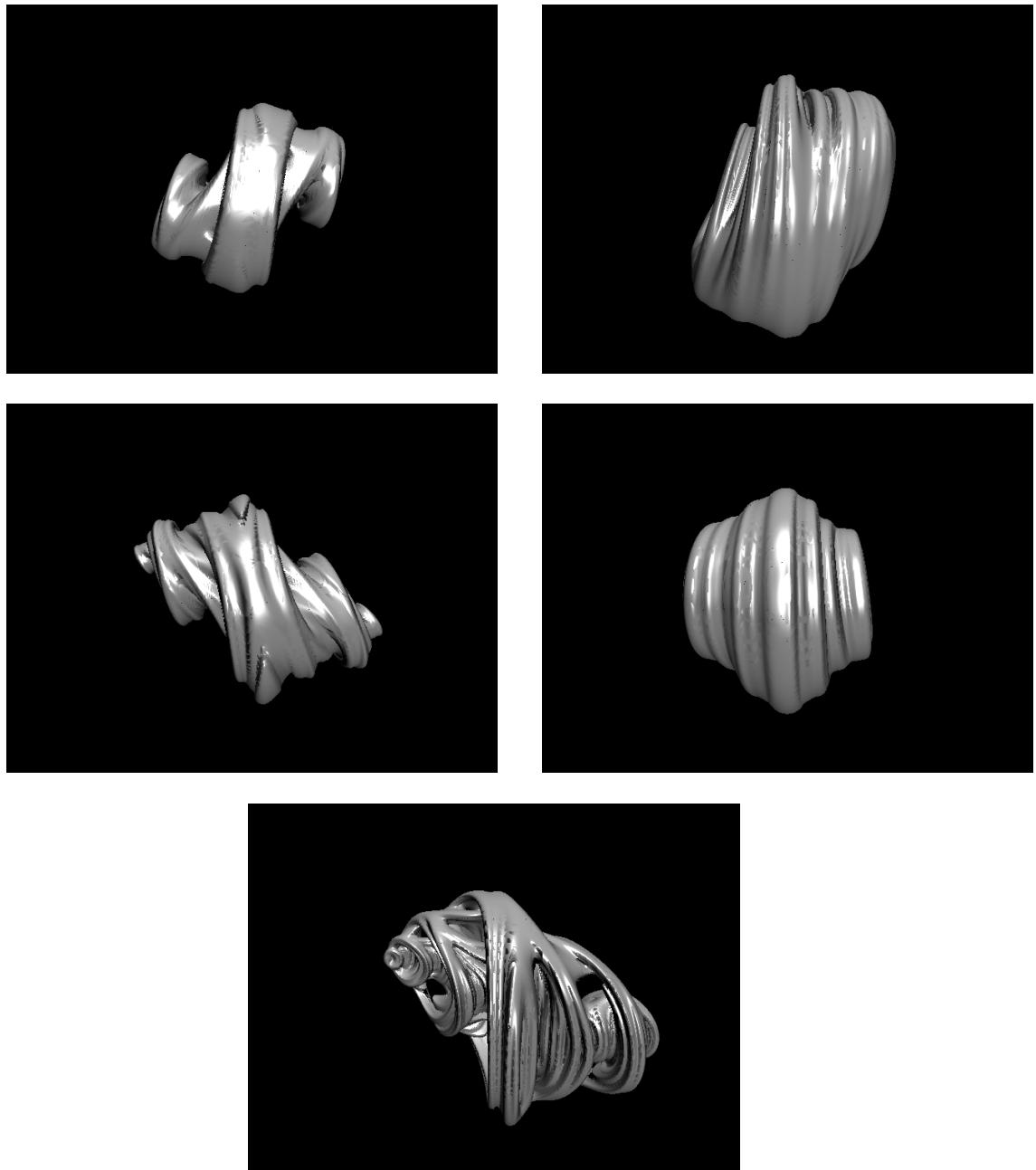


Figura B.2.: [Python] Cuaterniones de Julia (i)  $c = (-0.445, 0.339, -0.0889, -0.562)$  (ii)  $c = (0.185, 0.478, 0.125, -0.392)$  (iii)  $c = (-0.450, -0.447, 0.181, 0.306)$  (iv)  $c = (-0.218, -0.113, -0.181, -0.496)$  (v)  $c = (-0.125, -0.256, 0.847, 0.0895)$

## C. Estimación del coste y planificación

En esta sección, se realiza un presupuesto del coste del trabajo realizado en función de las horas dedicadas a cada tarea. Se estima un precio de 30€/hora para este proyecto.

Para la realización del presupuesto se han dividido los costes en directos o indirectos. Los costes directos son aquellos que guardan una estrecha relación con el desarrollo del proyecto. Estos costes cubren los gastos de tareas de índole teórica o práctica. Los costes indirectos son aquellos que no se atribuyen directamente al proyecto. Están relacionados con el mantenimiento, conexión a internet, factura de la luz, ... Se tasan en un 4 % del total de costes directos.

Las tareas teóricas son aquellas asociadas al estudio y análisis de los contenidos teóricos. Se tienen en cuenta tareas que van desde la consulta bibliográfica hasta su síntesis final con sus correspondientes correcciones y reuniones con el tutor. La tareas prácticas son aquellas relacionadas con el análisis, validación e implementación del código. Tenemos en consideración el equipo de trabajo (MSI GP62 7RE Leopard Pro con Intel(R) Core(TM) i7 – 7700HQ 2.80GHz de 8 núcleos y GTX 1050 Ti) y el porcentaje de su uso para el proyecto con respecto a la vida útil del mismo. Se estima el coste por porcentaje de uso de 195 €. Todos los recursos software que se han usado son gratuitos.

C. Estimación del coste y planificación

Concepto	Cantidad (horas)	Coste unidad (€/hora)	Coste total (€)
Costes directos			
Tareas teóricas			
Estudio conceptos previos	20	30	600
Estudio sistemas dinámicos reales	15	30	450
Estudio sistemas dinámicos complejos	25	30	750
Estudio fractales	35	30	1050
Redacción memoria	120	30	3600
Corrección memoria	25	30	750
Reuniones con el tutor	10	30	300
Tareas prácticas			
Equipo de trabajo: MSI GP62 7RE Leopard Pro			195
Implementación sistemas dinámicos reales	15	30	450
Implementación cálculo exponente de Lyapunov	2	30	60
Implementación conjuntos de Julia y Mandelbrot	5	30	150
Desarrollo código construcción de fractales	30	30	900
Implementación algoritmo box counting	5	30	150
Implementación ray tracing	112	30	3360
Desarrollo cuaderno jupyter	15	30	450
Costes indirectos			
Mantenimiento, conexión internet, luz, ...			528,6
Total			13743,6

Tabla C.1.: Estimación del coste del proyecto (IVA no incluido)

En la figura C.1 se encuentra la planificación optimista del desarrollo del trabajo. En ella se han dividido las tareas en teóricas (color azul) y prácticas (color naranja). La redacción y corrección de la memoria estarán presentes a lo largo de todo el desarrollo del proyecto. En la planificación se ha tenido en cuenta el periodo de exámenes.

En algunas tareas, las previsiones iniciales no se han cumplido. En la figura C.2 se puede observar una planificación realista del trabajo. En ciertos meses, la realización de tareas teóricas y prácticas ha sido de manera simultanea ya que tanto la parte teórica como práctica se complementan muy bien.



Figura C.1.: [Canva] Planificación optimista del desarrollo del trabajo



Figura C.2.: [Canva] Planificación realista del desarrollo del trabajo



## D. Software

### D.1. Archivos

Todos los archivos del [proyecto](#) se encuentran en la plataforma Github.

El siguiente código pone en práctica los conocimientos adquiridos en cada capítulo. Los ordenaremos según su relación con conceptos y resultados teóricos expuestos.

#### Capítulo 1: Sistemas dinámicos discretos

- [TentMap.py](#): representa la función tienda según distintos parámetros de  $\mu$  y calcula puntos fijos, puntos periódicos de periodo dos y la órbita de un valor inicial  $x_0$  (Figuras 1.1, 1.2, 1.3, 1.4 y 1.5).
- [BifurcationdiagramLogisticMap.py](#): representa el mapa logístico según distintos parámetros de  $\mu$  y el diagrama de bifurcación o Feigenbaum asociado a un valor inicial  $x_0$  (Figuras 1.6, 1.7 y 1.8).
- [BifurcationdiagramGaussMap.py](#): representa el mapa de Gauss según distintos parámetros de  $\alpha$  y  $\beta$ , calcula las posibles intersecciones con la diagonal y traza el diagrama de bifurcación o Feigenbaum asociado a un valor inicial  $x_0$  (Figuras 1.9, 1.10, 1.11 y 1.12).
- [HenonMap.py](#): representa el mapa de Hénon según distintos parámetros de  $\alpha$  y  $\beta$  (Figura 1.13).

#### Capítulo 2: Introducción a la teoría del caos

- [LyapunovExponent.py](#): calcula el exponente de Lyapunov del mapa logístico y representa los valores obtenidos junto a su diagrama de bifurcación (Figura 2.3).

#### Capítulo 3: Sistemas dinámicos complejos

- [JuliaSet.py](#): representa el conjunto de Julia relleno  $K_c$  en blanco y negro y el cerco de este conjunto (Figuras 3.1, 3.2, 3.3, 3.5, 3.6, 3.10, 3.12 y 3.14).
- [MandelbrotSet.py](#): representa el conjunto de Mandelbrot en blanco y negro y el cerco de este conjunto (Figuras 3.7 y 3.8).
- [ApplicationNewton-RapshonToJulia.py](#): utiliza el método de Newton-Rapshon para calcular las raíces de un polinomio y representar fractales de Newton (Figuras 3.16, 3.17 y 3.18).

## Capítulo 4: Fractales

Los siguientes programas construyen paso a paso distintos fractales:

- [CantorSet.py](#): conjunto de cantor (Figura 4.1).
- [SierpinskiTriangle.py](#): triángulo de Sierpiński (Figura 4.2).
- [KochCurve.py](#): curva de Koch (Figura 4.3) y copo de nieve de Koch (Figura 4.4).
- [KochSquare.py](#): cuadrado de Koch (Figura 4.5).
- [LevyCurve.py](#): curva de Lévy (Figura 4.6).
- [DragonCurve.py](#): curva del dragón (Figura 4.7).
- [L-system.py](#): interpreta axiomas y reglas de producción para construir fractales mediante sistemas de Lindenmayer (Figuras 4.8, 4.9, 4.10, 4.11, 4.12, 4.13 y 4.14).
- [BarnsleyFern.py](#): construye el helecho de Barnsley a partir de un SFI aplicando el algoritmo aleatorio (Figura 4.15).
- [SierpinskiTriangleChaosGame.py](#): construye el triángulo de Sierpiński a partir de un SFI mediante el juego del caos (Figura 4.16).

## Capítulo 5: Dimensión fractal

- [BoxCountingDimension.py](#): calcula la dimensión de fractales en 2D usando el algoritmo de box counting (Figuras 5.1, 5.2 y 5.3)

Todas las imágenes que se obtienen como resultado de la ejecución del código anterior se encuentran en el directorio [Images](#). Además, se ha realizado un [cuaderno jupyter](#) para una lectura más cómoda. Este cuaderno incluye todo el código anterior salvo aquellos programas que utilicen gráficos turtle ([CantorSet.py](#), [SierpinskiTriangle.py](#), [KochCurve.py](#), [KochSquare.py](#), [LevyCurve.py](#), [DragonCurve.py](#) y [L-system.py](#)).

Para poder utilizar los gráficos turtle en el cuaderno es necesario utilizar la herramienta `ipypturtle`, sin embargo, este widget no dispone de comandos y funcionalidades que son imprescindibles para el proyecto. Por ello, no se ha podido obtener una versión en línea de estos programas. No obstante, todo este código fuente se encuentra disponible en el repositorio.

## Capítulo 6: Visualización de fractales con ray tracing

Todo los archivos del trazador de rayos se encuentran en [RayTracing3DFractals](#). La estructura del proyecto es la siguiente:

- [vector3D.py](#): se encarga de manipular vectores de  $\mathbb{R}^3$ . Contiene la clase Vector3D.
- [image.py](#): gestiona imágenes en formato PPM. Contiene la clase Image.
- [color.py](#): representa los colores de los píxeles de la imagen. Contiene la clase Color.
- [point3D.py](#): representa puntos de  $\mathbb{R}^3$ . Contiene la clase Point3D.

- [sphere.py](#): representa el objeto esfera. Contiene la clase Sphere.
- [ray.py](#): crea tanto los rayos que se dirigen a la escena como rayos de luz, de reflexión, ... Contiene la clase Ray.
- [scene.py](#): almacena la información de la escena. Contiene la clase Scene.
- [rayTracer.py](#): se encarga del renderizado mediante ray tracing. Contiene la clase Ray-Tracer.
- [light.py](#): proporciona la iluminación de la escena. Contiene la clase Light.
- [composition.py](#): incluye información de los materiales de los que están compuestos los objetos que participan en la escena. Contiene las clases Composition y ChessBoard-Composition.
- [rayMarcher.py](#): se encarga del renderizado mediante ray marching. Contiene la clase RayMarcher.
- [mandelbulb.py](#): representa el fractal Mandelbulb. Contiene la clase Mandelbulb.
- [quaternion.py](#): se encarga de manipular cuaterniones. Contiene la clase Quaternion.
- [juliabulb.py](#): representa el fractal Juliabulb. Contiene la clase Juliabulb.
- [mandelbox.py](#): representa el fractal Mandelbox. Contiene la clase Mandelbox.
- [settings.py](#): contiene los ajustes de la aplicación.
- [main.py](#): controla la ejecución del programa.

En el directorio [RenderedImages](#) se encuentran las imágenes fractales renderizadas tanto en formato PPM como PNG.

## D.2. Lenguaje

El código ha sido implementado en *Python 3.8.5* y *Python 3* en el cuaderno de jupyter.

## D.3. Librerías

Los paquetes utilizados son los siguientes:

- [numpy](#): para trabajar con arrays.
- [matplotlib](#): permite crear y personalizar gráficos.
- [math](#): proporciona acceso a funciones matemáticas.
- [PIL](#): para la edición de imágenes.
- [turtle](#): para la creación de gráficos mediante instrucciones.
- [random](#): para generar números aleatorios.

#### D. Software

- *sys*: provee acceso a algunas variables usadas o mantenidas por el intérprete.
- *re*: proporciona operaciones para expresiones regulares.
- *multiprocessing*: se trata de un paquete que permite crear procesos para aprovechar al máximo los núcleos de la CPU.
- *tqdm.auto*: imprime barras de progreso.
- *tempfile*: genera directorios y archivos temporales.
- *pathlib*: representa rutas de archivos.
- *shutil*: ofrece una serie de operaciones de alto nivel en archivos y colección de archivos.

#### D.4. Entorno de desarrollo

En el desarrollo del proyecto se ha utilizado el navegador *Anaconda* para acceder a los entornos *spyder* y *jupyter notebook*.

# Bibliografía

Las referencias se listan por orden alfabético. Aquellas referencias con más de un autor están ordenadas de acuerdo con el primer autor.

- [Afl20] Omar Aflak. Ray tracing from scratch in python, 2020. <https://medium.com/swlh/ray-tracing-from-scratch-in-python-41670e6a96f9>. [Citado en pág. 85]
- [AGo2] M. Asunción Sastre, Antonio Giraldo. *Sistemas dinámicos discretos y caos: teoría, ejemplos y algoritmos*. Fundación General de la UPM de Madrid, 2002. [Citado en págs. IX, IX, X, 1, 33, 42, 43, and 44]
- [Bar88] Michael Barnsley. *Fractals everywhere*. Academic Press, 1988. [Citado en págs. 33, 49, 70, and 79]
- [Buc19] Jamis Buck. *The Ray Tracer Challenge*. Andy Hunt, 2019. [Citado en pág. 85]
- [Chr11a] Mikael Hvidtfeldt Christensen. Distance estimated 3d fractals, 2011. <http://blog.hvidtfeldts.net/index.php/2011/06/distance-estimated-3d-fractals-part-i/>. [Citado en pág. 95]
- [Chr11b] Mikael Hvidtfeldt Christensen. Distance estimated 3d fractals (v): The mandelbulb different de approximations, 2011. <http://blog.hvidtfeldts.net/index.php/2011/09/distance-estimated-3d-fractals-v-the-mandelbulb-different-de-approximations/>. [Citado en pág. 97]
- [Chr11c] Mikael Hvidtfeldt Christensen. Distance estimated 3d fractals (vi): The mandelbox, 2011. <http://blog.hvidtfeldts.net/index.php/2011/11/distance-estimated-3d-fractals-vi-the-mandelbox/>. [Citado en pág. 103]
- [Dev89] R.L. Devaney. *An introduction to chaotic dynamical systems*. Addison-Wesley, 1989. [Citado en págs. 1 and 23]
- [Ela00] S. Elaydi. *Discrete chaos*. Champan - HALL/CRC Press, 2000. [Citado en págs. IX, IX, IX, X, 26, 27, 29, 33, 37, 46, 49, and 67]
- [Fal90] Kenneth Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. John Wiley and Sons, 1990. [Citado en pág. 33]
- [Goo13] Geoffrey R. Goodson. *Lecture notes on dynamical systems, chaos and fractal geometry*. University of Towson, 2013. [Citado en pág. 1]
- [Har89] John C. Hart. Ray tracing deterministic 3-d fractals. 1989. [Citado en págs. 96, 99, and 101]
- [HN92] Yoshikazu Baba, Hiroyuki Nagashima. *Introduction to chaos: Physics and Mathematics of Chaotic Phenomena*. Institute of Physics, London, 1992. [Citado en pág. 23]
- [HOPo4] Dietmar Saupe, Heinz-Otto Peitgen, Hartmut Jürgens. *Chaos and Fractals. New frontiers of Science*. Springer-Verlag New York, Inc., 2004. [Citado en págs. 49 and 59]
- [Jul18] Gaston Julia. *Mémoire sur l'itération des fonctions rationnelles*. Gallica, Cellule MathDoc, 1918. [Citado en pág. 33]
- [Lyn18] Stephen Lynch. *Dynamical Systems with Applications using Python*. Springer, Birkhäuser, 2018. [Citado en págs. 1, 11, 23, 49, and 79]
- [Man75] Benoît B. Mandelbrot. *Les objets fractals: Forme, hasard et dimension*. Flammarion, 1975. [Citado en pág. 49]
- [Man82] Benoît B. Mandelbrot. *The fractal geometry of nature*. W. H. Freeman and Co., 1982. [Citado en pág. 49]

## Bibliografía

- [MRKo2] Orlando Merino, Mustafa R.S. Kulenovic. *Discrete Dynamical Systems and Difference Equations with Mathematica*. CHAPMAN - HALL/CRC Press, 2002. [Citado en pág. 1]
- [NVI] NVIDIA. Ray tracing. <https://developer.nvidia.com/discover/ray-tracing>. [Citado en págs. x and 85]
- [Nylo9] Paul Nylander. Hypercomplex fractals, 2009. <http://www.bugman123.com/Hypercomplex/index.html>. [Citado en pág. 97]
- [Pay21] Rafael Payá. Apuntes de análisis funcional del grado de matemáticas (ugr), 2020-2021. [http://www.ugr.es/~rpaya/documentos/Funcional/2020-21/Apuntes\\_01.pdf](http://www.ugr.es/~rpaya/documentos/Funcional/2020-21/Apuntes_01.pdf). [Citado en págs. 24 and 67]
- [Pyta] Python. Documentación de la librería de gráficos turtle de python. <https://docs.python.org/3/library/turtle.html>. [Citado en pág. 60]
- [Pytb] Python. multiprocessing paralelismo basado en procesos. <https://docs.python.org/es/3.9/library/multiprocessing.html>. [Citado en pág. 105]
- [Qui] Iñigo Quilez. 3d julia set fractals. <https://iquilezles.org/>. [Citado en pág. 101]
- [Scra] Scratchapixel. Introduction to shading. <https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/diffuse-lambertian-shading>. [Citado en págs. xi, xi, 92, and 93]
- [Scrb] Scratchapixel. The phong model, introduction to the concepts of shader, reflection models and brdf. <https://www.scratchapixel.com/lessons/3d-basic-rendering/phong-shader-BRDF>. [Citado en págs. xi, 90, and 92]
- [Shi20] Peter Shirley. Ray tracing in one weekend, 2020. <https://raytracing.github.io/books/RayTracingInOneWeekend.html#outputanimage/theppmimageformat>. [Citado en págs. xi, xi, 85, and 88]
- [TYL75] James A. Yorke, Tien-Yien Li. Period three implies chaos. 1975. [Citado en págs. 26 and 29]
- [Whio9] Daniel White. The unravelling of the real 3d mandelbulb, 2009. <https://www.skytopia.com/project/fractal/mandelbulb.html>. [Citado en pág. 97]