



GRADO EN MATEMÁTICA COMPUTACIONAL

TRABAJO FINAL DE GRADO

Algoritmos de dibujo en dinámica holomorfa

Autor:
Clara SERRANO HERRERA

Tutor académico:
Jordi CANELA SÁNCHEZ

Curso académico 2022/2023

Resumen

Este trabajo aborda los sistemas dinámicos discretos en una variable compleja. Tiene el objetivo de desarrollar diversos algoritmos y la teoría necesaria para poder entenderlos ya que son una de las herramientas básicas para visualizar a día de hoy la dinámica holomorfa y poder entender los conjuntos. Para ello, se empieza introduciendo conceptos preliminares como fracciones continuas, esfera de Riemann y funciones racionales. Luego, se explora la teoría local, definiendo puntos periódicos y su dinámica en entornos cercanos. La teoría global define los conjuntos de Julia y Fatou, tanto para polinomios como de forma general. También se estudia el conjunto de Mandelbrot, sus propiedades y se plantean algunos problemas abiertos. La sección final se dedica a presentar y explicar los tres algoritmos de dibujo: el de escape, el de iteración inversa y el de Henriksen. Se realiza una comparación exhaustiva de los algoritmos, evaluando sus ventajas y desventajas.

Palabras clave

Conjunto de Julia, conjunto de Fatou, conjunto de Mandelbrot, algoritmo de escape, algoritmo de iteración inversa, algoritmo de Henriksen.

Keywords

Julia set, Fatou set, Mandelbrot set, escape algorithm, inverse iteration algorithm, Henriksen algorithm.

Índice general

| | |
|--|-----------|
| Introducción | 1 |
| 1. Conceptos preliminares | 5 |
| 1.1. Fracciones continuas | 5 |
| 1.2. Esfera de Riemann | 8 |
| 1.3. Funciones racionales | 9 |
| 2. Teoría local | 11 |
| 2.1. Definiciones básicas | 11 |
| 2.2. Dinámica en un entorno cerca de puntos periódicos | 13 |
| 3. Teoría global | 21 |
| 3.1. Conjuntos de Julia y Fatou | 21 |
| 3.1.1. Definición para polinomios | 21 |
| 3.1.2. Definición general | 21 |
| 3.1.3. Propiedades | 25 |

| | |
|-------------------------------------|-----------|
| 4. Conjunto de Mandelbrot | 29 |
| 4.1. Definición de \mathcal{M} | 29 |
| 4.2. Propiedades | 30 |
| 4.3. Componentes hiperbólicas | 31 |
| 5. Algoritmos | 35 |
| 5.1. Algoritmo de escape | 36 |
| 5.2. Algoritmo de iteración inversa | 47 |
| 5.3. Algoritmo de Henriksen | 55 |
| 5.4. Comparación de algoritmos | 66 |
| 5.5. Conclusión final | 73 |
| A. Anexo I | 77 |
| A.1. Algoritmo de escape | 77 |
| A.1.1. Conjunto de Mandelbrot | 77 |
| A.1.2. Conjunto de Julia | 78 |
| A.2. Algoritmo de iteración inversa | 79 |
| A.3. Algoritmo de Henriksen | 81 |
| A.3.1. Conjunto de Mandelbrot | 81 |
| A.3.2. Conjunto de Julia | 82 |

Introducción

A finales del siglo XIX, distintos matemáticos tales como Weierstrass, Koch, Lévy o Cantor empezaron a estudiar objetos y comportamientos que iban en contra de la intuición clásica y que ahora conocemos como *fractales*. A principios del siglo XX, matemáticos como Schröder, Poincaré, Fatou y Julia indagaron en la iteración de funciones en el plano euclídeo y del plano complejo dando lugar a los actualmente conocidos *sistemas dinámicos*. Estos sistemas dieron lugar a objetos geométricos parecidos a los estudiados por Koch o Cantor y cumplían propiedades como la *auto-similitud*. Pero, a causa de la limitación de los ordenadores de la época, nunca pudieron visualizar los fractales que estaban manejando. A mediados del siglo XX, aparecieron los primeros ordenadores y su capacidad de cálculo permitió visualizar las figuras geométricas y ayudar a entender y crear nuevas teorías matemáticas.

El método de Newton es un método iterativo muy conocido en el mundo de las matemáticas que se utiliza para calcular las raíces reales o complejas de una ecuación

$$f(z) = 0,$$

siendo f una función derivable y z una variable real o compleja. Las primeras menciones al estudio de iteración de funciones holomorfas se encuentran en los trabajos sobre el método de Newton de Ernst Schröder sobre 1870 [19, 20] y de Arthur Cayley en 1879 [3, 4]. Estos dos matemáticos estudiaron el método de Newton a través de la iteración de la función holomorfa

$$N_f(z) = z - \frac{f(z)}{f'(z)}$$

en el plano complejo, que hasta el momento solo se había considerado en variable real. De esta forma, iterando la función $N(z)$ con una condición inicial z_0 , se obtenía la sucesión

$$z_0, z_1 = N_f(z_0), z_2 = N_f(z_1), \dots, z_n = N_f(z_{n-1}), \dots$$

llamada *órbita de z_0* .

Schröder demostró el teorema conocido por *Teorema del punto fijo* que justifica el funcionamiento del método de Newton ya que las raíces de $f(z)$ son puntos fijos atractores de $N(z)$.

Además, propuso condiciones para garantizar una convergencia más rápida hacia la raíz, lo que lo motivó a buscar métodos alternativos de convergencia y, por ende, a estudiar la iteración de funciones holomorfas en un contexto más amplio.

Otro problema que estudiaron Schröder y Cayley fue iterar el método de Newton en zonas alejadas de los entornos de las raíces. Supongamos que un polinomio de grado 2 tiene dos raíces complejas, a y b . El problema consistía en saber qué condiciones iniciales z_0 convergerían a a y cuáles a b . Esto se conoce como *cuenca de atracción*, es decir, sea α un punto fijo, la cuenca de atracción de α son todos los puntos que convergen a α . De esta forma, se quería saber qué puntos pertenecían a la cuenca de atracción de a y qué puntos a la de b . Ambos matemáticos consiguieron resolver el problema para el caso de polinomios de grado 2 y demostraron que la frontera de ambas cuencas era una recta como podemos ver en la Figura 1(a) donde se ha utilizado la función $f(z) = z^2 - 4$ cuyas raíces son 2 y -2 y los colores para la cuenca de atracción son rojo y verde, respectivamente. Cayley continuó con la investigación y probó con polinomios de grado 3. Sin embargo, las divisiones de las cuencas de atracción de las raíces eran mucho más complicadas. Esto lo podemos ver en la Figura 1(b) donde se ha utilizado la función $f(z) = z^3 - 1$ cuyas raíces son 1, $e^{2\pi\frac{i}{3}}$ y $e^{-2\pi\frac{i}{3}}$ y los colores para la cuenca de atracción son rojo, verde y azul, respectivamente.

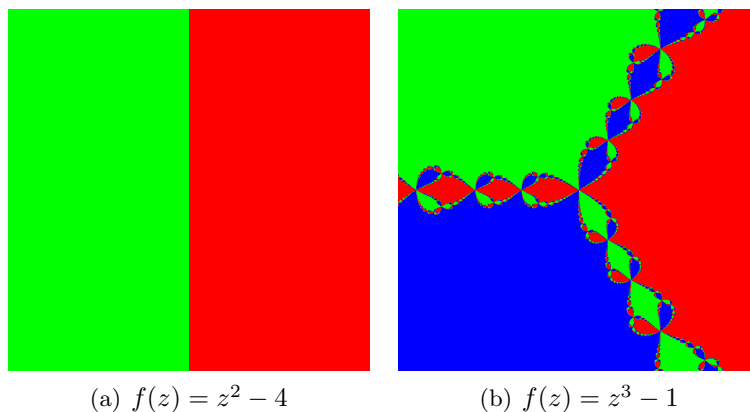


Figura 1: N_f para diversas funciones f .

La teoría global se empezó a estudiar a partir de las investigaciones realizadas por Pierre Fatou y Gaston Julia en la década 1910-1920. Este estudio se centró en la iteración de funciones holomorfas en la esfera de Riemann [9, 10, 7, 8, 13]. La aportación más importante de estos dos matemáticos es la partición de la esfera en dos conjuntos complementarios con un comportamiento dinámico totalmente diferente, uno *estable* conocido como *conjunto de Fatou* y otro *caótico* conocido como *conjunto de Julia*. Por estable nos referimos a que los puntos se comportan de la misma manera que los puntos de su entorno, mientras que los puntos caóticos no.

Fatou y Julia estudiaron detalladamente los conjuntos de Fatou y Julia, pero dejaron también muchos problemas abiertos. En 1982, D. Sullivan resolvió la clasificación de las componentes del conjunto de Fatou usando funciones casiconformes [23]. Esto marcó el inicio de una nueva era en el estudio de la teoría de la iteración holomorfa. El *conjunto de Mandelbrot*, que corresponde a valores $c \in \mathbb{C}$ que hacen que el conjunto de Julia de $P_c(z) = z^2 + c$ sea conexo, atrajo mucha atención, aunque Fatou solo lo mencionó brevemente y no lo estudió. Muchas propiedades de este conjunto son problemas abiertos y siguen siendo objeto de investigación activa.

Debido a la complejidad de los conjuntos de Fatou y Julia y del conjunto de Mandelbrot, el uso de los ordenadores y su capacidad de cálculo han facilitado la representación de los conjuntos y con ello su comprensión. Este trabajo tiene dos objetivos principales: presentar los fundamentos de la teoría de la dinámica compleja e implementar diversos algoritmos de dibujo. Con respecto a la presentación teórica, abordamos tanto la dinámica local como la global y también algunas nociones básicas del conjunto de Mandelbrot. Los fundamentos teóricos presentados son esenciales para sentar las bases y justificar el funcionamiento de los tres algoritmos de dibujo. De cada algoritmo se realiza una implementación y una explicación detallada. También se realizan dibujos para cada caso posible y finalmente se comparan entre sí. El primer algoritmo es el algoritmo de escape, el segundo es el algoritmo de iteración inversa y el último es el algoritmo de Henriksen.

El trabajo está estructurado como sigue. En el capítulo 1 se presentan los conceptos preliminares, abordando temas como las fracciones continuas en la sección 1.1, la esfera de Riemann en la sección 1.2 y las funciones racionales en la sección 1.3. A continuación, en el capítulo 2 se desarrolla la teoría local, empezando con definiciones básicas en la sección 2.1 y explorando la dinámica en un entorno cercano a puntos periódicos en la sección 2.2. El capítulo 3 se centra en la teoría global, específicamente en los conjuntos de Julia y Fatou. Se inicia con la definición de estos conjuntos para polinomios en la sección 3.1.1, seguida de la definición general en la sección 3.1.2. También se discuten las propiedades de estos conjuntos en la sección 3.1.3. El capítulo 4 se dedica al estudio del conjunto de Mandelbrot. Aquí, se establece la definición de este conjunto en la sección 4.1, se exploran sus propiedades en la sección 4.2 y se abordan las componentes hiperbólicas en la sección 4.3. Los algoritmos son el foco del capítulo 5. Se presentan los 3 algoritmos mencionados anteriormente, el algoritmo de escape en la sección 5.1, el algoritmo de iteración inversa en la sección 5.2 y el algoritmo de Henriksen en la sección 5.3. Además, se realiza una comparación de estos algoritmos en la sección 5.4 y se presenta una conclusión final en la sección 5.5. Por último en el Anexo A, se proporcionan todos los códigos utilizados para obtener los dibujos de los 3 algoritmos.

Capítulo 1

Conceptos preliminares

En este capítulo se presentan diversos conceptos que serán de utilidad a lo largo de este trabajo de fin de grado.

1.1. Fracciones continuas

En este apartado vamos a realizar una breve introducción a los números irracionales y a las fracciones continuas. Para una mayor introducción a fracciones continuas ver [16]. Un número es *irracional* si no se puede expresar en forma de fracción debido a que posee infinitas cifras decimales no periódicas. Además, se puede demostrar que si los decimales de un número son periódicos entonces el número es racional. El conjunto de los números irracionales se denota por $\mathbb{R} \setminus \mathbb{Q}$.

Proposición 1.1.1. *Un número con decimal infinito periódico es racional.*

Demostración. Un número r es racional cuando se puede expresar como la división de dos números enteros, es decir, $r = \frac{p}{q}$ siendo $p, q \in \mathbb{Z}$ y $q \neq 0$.

Por definición, un número infinito periódico tiene n cifras de su parte decimal que se repiten infinitamente, con $n \in \mathbb{N}$ y $n \neq 0$. Por tanto, si consideramos $p = 10^n \cdot r - r$ la parte decimal periódica del número se cancela dando como resultado la parte entera. Si $p = 10^n \cdot r - r = r(10^n - 1)$, despejando r obtenemos que $r = \frac{p}{(10^n - 1)}$. En consecuencia, hemos conseguido expresar r como la división de dos números enteros. \square

Un ejemplo de la demostración anterior sería el siguiente. Sea $r = 2,85858585\dots = 2,\overline{85}$ un número racional con decimal infinito. Vemos que $n = 2$ porque el número 85 se repite periódicamente y tiene 2 cifras. Ahora, consideramos

$$p = 10^n \cdot r - r = 10^2 \cdot 2,858585\dots - 2,858585\dots = 285,8585\dots - 2,8585\dots = 283.$$

Por tanto sabemos que $p = 283$. Pero como hemos visto que

$$p = 10^n \cdot r - r = r(10^n - 1) \implies r = \frac{p}{(10^n - 1)},$$

sustituyendo obtenemos

$$r = \frac{283}{10^2 - 1} = \frac{283}{99}.$$

El objetivo principal de esta sección es el estudio de números irracionales. Para ello usaremos fracciones continuas.

Definición 1.1.2. Una *fracción continua simple* $[a_0; a_1, \dots, a_n, \dots]$ es una expresión obtenida a través de la iteración de representar un número como la suma de su parte entera y el recíproco de otro número. Es decir,

$$[a_0; a_1, \dots, a_n, \dots] = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots + \frac{1}{a_n + \dots}}}}$$

con $a_0 \in \mathbb{Z}$ y $a_j \in \mathbb{N} \setminus \{0\}$ si $j > 0$. Se dice que es convergente si la sucesión de convergentes $p_n/q_n := [a_0; a_1, \dots, a_{n-1}]$ es *convergente* (donde $p_n \in \mathbb{Z}$ y $q_n \in \mathbb{N} \setminus \{0\}$ son coprimos). En este caso, definimos el valor de la fracción continua como el límite de p_n/q_n .

El teorema siguiente nos muestra que existe una biyección entre los números irracionales y las fracciones continuas simples. Para ver la demostración consultar [14].

Teorema 1.1.3. *Toda fracción continua simple $[a_0; a_1, \dots]$ es convergente y su límite α es irracional. Recíprocamente, para cada $\alpha \in \mathbb{R} \setminus \mathbb{Q}$ existe una única fracción continua simple $[a_0; a_1, \dots]$ tal que $[a_0; a_1, \dots] = \alpha$.*

En los capítulos siguientes vamos a utilizar el número áureo. Utilizando el *algoritmo de Euclides* podemos obtener su fracción continua. El procedimiento a seguir sería el siguiente: tenemos un número x y lo escribimos como

$$x = a_0 + x_0 = a_0 + \frac{1}{\frac{1}{x_0}},$$

siendo a_0 su parte entera y x_0 su parte decimal. Ahora escribimos $\frac{1}{x_0} = a_1 + x_1$, siendo a_1 la parte entera y x_1 su parte decimal. Repetimos este proceso de forma iterativa, obteniendo a cada paso un nuevo a_n .

Vamos a verlo con el siguiente ejemplo para el número áureo. Sea $\theta = \frac{\sqrt{5}+1}{2}$ el número áureo. Ahora, tenemos que separar el número en parte entera y decimal. Como $\theta \approx 1,61803398$, la parte entera será 1. La parte decimal es distinta de 0, por tanto tenemos que calcular la inversa y repetir el mismo proceso con el resultado.

Por tanto, como hemos comentado primero separamos el número en parte entera y decimal:

$$\frac{\sqrt{5}+1}{2} = 1 + \frac{\sqrt{5}-1}{2}$$

Ahora calculamos la inversa de la fracción que nos queda:

$$1 + \frac{\sqrt{5}-1}{2} = 1 + \frac{1}{\frac{2}{\sqrt{5}-1}} = 1 + \frac{1}{\frac{2(\sqrt{5}+1)}{5-1}} = 1 + \frac{1}{\frac{\sqrt{5}+1}{2}}$$

Repetimos otra vez el mismo proceso, es decir, separamos la parte entera y decimal:

$$1 + \frac{1}{\frac{\sqrt{5}+1}{2}} = 1 + \frac{1}{1 + \frac{\sqrt{5}-1}{2}}$$

Ahora calculamos la inversa:

$$1 + \frac{1}{1 + \frac{\sqrt{5}-1}{2}} = 1 + \frac{1}{1 + \frac{1}{\frac{2}{\sqrt{5}-1}}}$$

y repetimos el mismo proceso mientras se pueda. Por tanto θ es de la forma $\theta = [1; 1, \dots, 1, \dots]$. A lo largo del trabajo utilizaremos en numerosas ocasiones el número irracional θ , pero haremos un abuso de notación y en lugar de referirnos al número áureo, nos referiremos al número $\theta = \frac{\sqrt{5}-1}{2}$. Este número es la parte decimal del número áureo y es con lo que nos interesará trabajar.

Por otra parte, una consecuencia del algoritmo de Euclides es que si tenemos una fracción racional podemos obtener una fracción continua finita. Por ejemplo:

$$\frac{5}{7} = 0 + \frac{1}{\frac{7}{5}} = 0 + \frac{1}{1 + \frac{2}{5}} = 0 + \frac{1}{1 + \frac{1}{\frac{5}{2}}} = 0 + \frac{1}{1 + \frac{1}{2 + \frac{1}{2}}}$$

Vemos que a cada paso el denominador se va haciendo más pequeño, por tanto es finito. La fracción continua resultante sería: $\frac{5}{7} = [0; 1, 2, 2]$.

Veamos a continuación cuáles son los diferentes tipos de números irracionales.

Definición 1.1.4. Se dice que un número irracional $\alpha = [a_0; a_1, \dots]$ es de tipo:

1. acotado, $\alpha \in \mathcal{C}$, si

$$\sup_n a_n < \infty;$$

2. diofántico, $\alpha \in \mathcal{D}$, si

$$\exists K > 0, \exists \mu < \infty : \forall p \in \mathbb{Z}, \forall q \in \mathbb{N} \left| \alpha - \frac{p}{q} \right| > \frac{K}{q^{2+\mu}};$$

3. Brjuno, $\alpha \in \mathcal{B}$, si

$$\sum_{n \geq 1} \frac{\log(q_{n+1})}{q_n} < \infty.$$

En particular, anteriormente hemos visto que la fracción continua del número áureo es $\theta = [0; 1, 1, \dots]$. Por lo tanto, el número áureo es un número irracional de tipo acotado.

El siguiente resultado nos proporciona la relación entre \mathcal{C} , \mathcal{D} y \mathcal{B} . Para una demostración del mismo ver [12].

Proposición 1.1.5. *Se cumple que $\mathcal{C} \subset \mathcal{D} \subset \mathcal{B}$.*

1.2. Esfera de Riemann

Cuando estudiamos sistemas dinámicos en variable compleja es conveniente poder comprender el comportamiento de puntos $z \in \mathbb{C}$ que tienden a ∞ . Es necesario obtener una compactificación adecuada del plano complejo. Por ello, consideramos la esfera de Riemann $\widehat{\mathbb{C}}$ que nos permite estudiar el plano complejo ampliado $\mathbb{C} \cup \{\infty\}$. La esfera de Riemann es una esfera unidad que se obtiene a través de la adición de un punto de infinito en el polo norte, N , al plano complejo. Por tanto, la esfera de Riemann es una representación geométrica de los números complejos extendidos.

La correspondencia utilizada entre los complejos y la esfera se llama *proyección estereográfica*. La idea de construcción se basa en considerar \mathbb{C} como el plano que atraviesa la esfera centrada

en $z = 0$. Ahora, a cada punto $z \in \mathbb{C}$ le tenemos que hacer corresponder un único punto w de la esfera de Riemann. Este punto w se obtiene al intersectar la esfera y la recta entre el polo norte $N = (0, 0, 1)$ y el punto z . Podemos observar que cuanto mayor es el módulo del punto z , más cerca estará w del polo norte, de esta forma la proyección de ∞ será N . Por tanto podemos determinar que los puntos z tales que $|z| > 1$ pertenecen al hemisferio norte y que, recíprocamente, los puntos z tales que $|z| < 1$ pertenecen al hemisferio sur. Además, los puntos en los que $|z| = 1$, se tiene que $w = z$. Podemos ver una representación en la Figura 1.1.

Cuando necesitamos estudiar puntos cuya imagen es ∞ , es conveniente disponer de un atlas adecuado. Un atlas habitual a tal efecto es el formado por las cartas (U, ϕ_1) y (V, ϕ_2) donde $U = \widehat{\mathbb{C}} \setminus \{\infty\}$ y $V = \widehat{\mathbb{C}} \setminus \{0\}$, $\phi_1 : U \rightarrow \mathbb{C}$ con $\phi_1(z) = z$ y $\phi_2 : V \rightarrow \mathbb{C}$ con $\phi_2(z) = \frac{1}{z}$, siendo $\frac{1}{\infty} = 0$. Cabe remarcar que si estudiamos puntos periódicos cuya órbita contiene 0 e ∞ , podrá ser conveniente usar otro atlas donde $V = \widehat{\mathbb{C}} \setminus \{a\}$ y $\phi_2(z) = \frac{1}{z-a}$, donde a no pertenece al ciclo atractor.

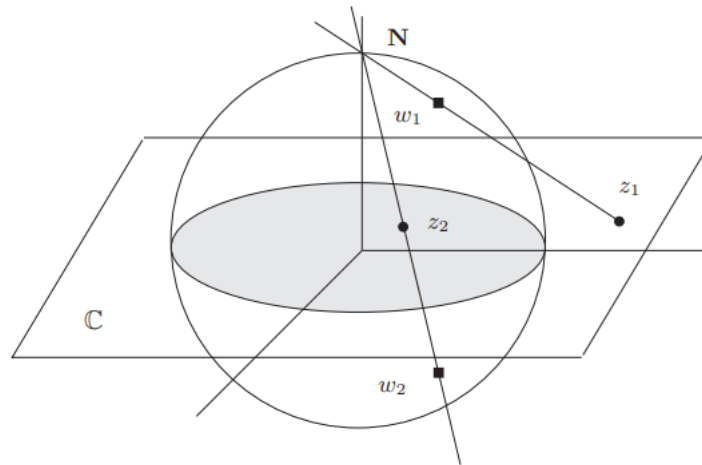


Figura 1.1: Esfera de Riemann. Figura tomada de [17, Figura 7.6]

1.3. Funciones racionales

En primer lugar vamos a estudiar qué es una función holomorfa.

Definición 1.3.1. Se dice que una función continua $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ es derivable en sentido complejo si lo es en \mathbb{R}^2 y cumple las ecuaciones de Cauchy-Riemann

$$u_x(x, y) = v_y(x, y)$$

$$v_x(x, y) = -u_y(x, y).$$

Diremos que f es *holomorfa* (o *analítica*) en un abierto $V \subset \widehat{\mathbb{C}}$ si es derivable en sentido complejo para todo punto $z \in V$. Equivalentemente, f es holomorfa en un abierto $V \in \mathbb{C}$, si existe el límite

$$f'(z) = \lim_{h \rightarrow 0} \frac{f(z+h) - f(z)}{h},$$

donde $f : V \subset \mathbb{C} \mapsto \mathbb{C}$ y $h \in \mathbb{C}$.

Definición 1.3.2. Una función $f : U \rightarrow V$ es *conforme* si es un homeomorfismo y es holomorfa.

Sea $f : \widehat{\mathbb{C}} \rightarrow \widehat{\mathbb{C}}$ una función continua. Si $f(z_0) = \infty$ para algún $z_0 \in \mathbb{C}$ se dice que f es holomorfa en un entorno de z_0 si y solo si $\frac{1}{f(z)}$ es holomorfa en un entorno de z_0 . Además diremos que f es holomorfa en un entorno de $z = \infty$ si $f(\frac{1}{z})$ es holomorfa alrededor de $z = 0$. Destacamos que $z \rightarrow \frac{1}{z}$ es una aplicación conforme que va de $\widehat{\mathbb{C}} \rightarrow \widehat{\mathbb{C}}$.

Definición 1.3.3. Las *funciones racionales* son funciones de la forma $f(z) = \frac{P(z)}{Q(z)}$, donde P y Q son polinomios.

Una propiedad que caracteriza estas funciones es que el grado topológico de la función racional $f(z) = \frac{P(z)}{Q(z)}$ viene dado por el máximo entre los grados de los polinomios P y Q . Es decir,

$$d_f = \max\{d_P, d_Q\}.$$

Cuando decimos que el grado topológico es d_f queremos decir que todo punto $z_0 \in \widehat{\mathbb{C}}$ tiene exactamente d_f preimágenes bajo f (contando multiplicidad). Si f es una función constante tenemos que el grado de la función f es 0. Los puntos críticos de f son los puntos en los que se anula su derivada, es decir, los c tales que $f'(c) = 0$.

Una propiedad importante es la siguiente y podemos encontrar su demostración en [2]

Corolario 1.3.4. *Una función racional f con grado positivo $d_f \geq 1$ tiene $2d_f - 2$ puntos críticos en $\widehat{\mathbb{C}}$. Un polinomio p con grado positivo $d_p \geq 1$ tiene como máximo $d_p - 1$ puntos críticos en \mathbb{C} .*

En el caso polinomial lo que sucede es que visto desde el punto de vista de la esfera de Riemann, el ∞ es un punto crítico de multiplicidad $d_p - 1$ para cualquier polinomio.

Otra propiedad importante de las funciones racionales es que toda función holomorfa que vaya de la esfera de Riemann a ella misma es una función racional. Esto lo vamos a ver en la siguiente proposición. Para ver la demostración de la proposición consultar [18].

Proposición 1.3.5. *Las funciones holomorfas $\widehat{\mathbb{C}} \rightarrow \widehat{\mathbb{C}}$ que no son idénticamente ∞ son funciones racionales de la forma $\frac{P}{Q}$ con P, Q polinomios en \mathbb{C} y $Q \neq 0$.*

Capítulo 2

Teoría local

Los sistemas dinámicos pueden estudiarse en diferentes espacios métricos, pero en este trabajo nos vamos a restringir al estudio de la iteración de funciones holomorfas en la esfera de Riemann. Para ello vamos a introducir los conceptos y definiciones básicas que necesitaremos en los capítulos siguientes.

2.1. Definiciones básicas

Definición 2.1.1. Sea $f : \widehat{\mathbb{C}} \rightarrow \widehat{\mathbb{C}}$ una función racional. La *órbita* de un punto $z_0 \in \widehat{\mathbb{C}}$ es el resultado de aplicar la función f de forma iterativa. Es decir,

$$z_0, f(z_0) = z_1, f(z_1) = z_2, \dots, f(z_n) = z_{n+1}, \dots$$

Denotamos la órbita de z_0 como $\mathcal{O}^+(z_0) = \{z_0, z_1, \dots, z_{n+1}, \dots\}$.

La definición anterior define la órbita positiva de z_0 , mientras que normalmente cuando se itera hacia atrás, se obtiene más de una preimagen que forman una órbita negativa. Cuando nos refiramos a la órbita de z_0 , nos referiremos a la órbita positiva.

Una vez ya sabemos iterar las funciones, vamos a estudiar los puntos cuyas órbitas van a tener un papel fundamental. En primer lugar, empezaremos por la definición de punto fijo.

Definición 2.1.2. Sea f una función holomorfa y $z_0 \in \widehat{\mathbb{C}}$. Se dice que z_0 es un *punto fijo* si y sólo si $f(z_0) = z_0$.

Los puntos fijos son el caso más sencillo de puntos periódicos.

Definición 2.1.3. Dado $z_0 \in \widehat{\mathbb{C}}$, se dice que z_0 es un *punto periódico de periodo p* si existe $p \in \mathbb{N}$ tal que $f^p(z_0) = z_0$ y además para todo $n < p$ se tiene que $f^n(z_0) \neq z_0$. En este caso su órbita es el conjunto finito $\langle z_0 \rangle = \{z_0, z_1, \dots, z_{p-1}\}$.

Definición 2.1.4. Dado $z_0 \in \widehat{\mathbb{C}}$, se dice que z_0 es un *punto preperiódico* si z_0 no es periódico pero alguno de los elementos de su órbita sí lo es.

Dado un ciclo $\langle z_0 \rangle$, usamos la regla de la cadena para definir su multiplicador.

Definición 2.1.5. Sea $f : \widehat{\mathbb{C}} \rightarrow \widehat{\mathbb{C}}$ y sea $z_0 \in \mathbb{C}$ un punto periódico de periodo p . Se define el *multiplicador* de la órbita periódica como $(f^p)'(z_0) = \prod_{i=0}^{p-1} f'(z_i) = f'(z_0) \cdots f'(z_{p-1})$.

A continuación introducimos el concepto de conjugación entre dos funciones.

Definición 2.1.6. Sea $f : U \rightarrow U$ y $g : V \rightarrow V$, $U, V \subset \mathbb{C}$. Se dice que son *conjugados* si existe un homeomorfismo $h : U \rightarrow V$ tal que $f = h \circ g \circ h^{-1}$.

Las conjugaciones son cruciales en el estudio de los sistemas dinámicos. La idea es que si dos funciones f y g son conjugadas, entonces presentan la “la misma dinámica” allí donde la conjugación esté definida. Por tanto, las funciones conjugadas tienen las mismas dinámicas cualitativas. En particular, si la conjugación es conforme, entonces envía puntos periódicos a puntos periódicos con el mismo multiplicador. Por simplicidad, tomando f^p y g^p podemos asumir que el punto periódico z_0 es fijo (observar que si z_0 tiene periodo p y multiplicador λ bajo g , entonces z_0 es un punto fijo de multiplicador λ bajo g^p).

Teorema 2.1.7. *El multiplicador de un punto periódico es invariante por conjugación conforme. Es decir, si $g : V \rightarrow V$ tiene un punto fijo z_0 con multiplicador λ , y si $f = h \circ g \circ h^{-1}$ con $h : U \rightarrow V$ conforme, entonces $f : U \rightarrow U$ tiene un punto fijo $w_0 = h(z_0)$ con multiplicador λ .*

Demostración. Tenemos que $f(w_0) = h(g(h^{-1}(w_0))) = h(g(z_0)) = h(z_0) = w_0$. Por tanto, w_0 es un punto fijo de f . Entonces, el multiplicador de w_0 sería $f'(w_0) = h'(z_0)g'(z_0)h^{-1}(w_0)$ por la regla de la cadena. Sabiendo que $h^{-1}(w_0) = \frac{1}{h'(z_0)}$ y sustituyendo en la ecuación anterior tenemos finalmente que $f'(w_0) = h'(z_0)g'(z_0)\frac{1}{h'(z_0)} = g'(z_0) = \lambda$. \square

Observar que cuando hemos definido el multiplicador, no hemos mencionado el caso en el que infinito es un punto periódico. Este caso es un poco más delicado pero se puede trabajar utilizando el teorema anterior. Si ∞ es un punto fijo, para calcular su multiplicador aplicamos una conjugación que lo envíe a un punto acotado. Por ejemplo, conjugando por $z \mapsto \frac{1}{z}$ el multiplicador de infinito es la derivada de $\frac{1}{f(\frac{1}{z})}$ en el 0. Es decir, $g'(0) = \frac{1}{f'(\infty)}$. Esto se puede

ver con más detalle en [2]. Esta definición depende del Teorema 2.1.7 puesto que podríamos haber tomado otra conjugación que enviase infinito a otro punto del plano complejo. Sin embargo, se sigue de este teorema que usando otra conjugación obtendríamos el mismo multiplicador.

Una vez hemos definido el multiplicador, podemos estudiar el comportamiento de los puntos periódicos.

Definición 2.1.8. Sea $z_0 \in \mathbb{C}$ un punto periódico de $f : \widehat{\mathbb{C}} \rightarrow \widehat{\mathbb{C}}$ con multiplicador $\lambda = (f^p)'(z_0)$. Se dice que z_0 es

- *atractor* si $|\lambda| < 1$ (*súper atractor* si $|\lambda| = 0$);
- *repulsor* si $|\lambda| > 1$;
- *indiferente (o neutral)* si $|\lambda| = 1$. En tal caso, decimos que z_0 es:
 - *indiferente parabólico* si $\lambda = e^{2\pi i \frac{p}{q}}$ con $p, q \in \mathbb{N}$, $(p, q) = 1$ y $p < q$;
 - *indiferente irracional* si $\lambda = e^{\pi i \theta}$ con $\theta \in \mathbb{R} \setminus \mathbb{Q}$.

Observación 1. Si $f(z) = a_n z^n + \dots + a_1 z + a_0$ es un polinomio con $n \geq 2$, entonces ∞ es un punto fijo súper atractor (ver [16]). Claramente $f(\infty) = \infty$ y por tanto es un punto fijo. Además su multiplicador es $\frac{1}{f'(\infty)} = \frac{1}{\infty} = 0$.

2.2. Dinámica en un entorno cerca de puntos periódicos

Ahora vamos a estudiar el comportamiento dinámico alrededor de los diferentes tipos de puntos fijos. Un caso particular importante de conjugación es la linearización. Dado un punto fijo, queremos saber si la dinámica cerca de la función es conjugada a la parte lineal.

Definición 2.2.1. Una función es *linearizable* alrededor de un punto fijo con multiplicador $\lambda \neq 0$ si existe una conjugación local alrededor del punto fijo (función conforme que envía el punto fijo a 0) que conjuga la dinámica de f con la rotación rígida $z \rightarrow \lambda z$.

En particular, en esta sección vamos a ver en qué casos la función es linearizable alrededor del punto fijo o periódico. Los teoremas que vamos a enunciar son muy importantes así que en caso de querer ver las demostraciones, consultar [16].

Caso atractor y repulsor

Una proposición relacionada con los puntos atractores y repulsores es la siguiente.

Proposición 2.2.2. Si z_0 es atractor entonces es topológicamente atractor. Si z_0 es repulsor entonces f^{-1} es topológicamente atractor para f^{-1} .

Demostración. Supongamos que z_0 es un punto fijo de f y que la derivada de f evaluada en z_0 es menor que 1, $|f'(z_0)| < 1$. Por tanto, existirá $r > 0$ y $|f'(z_0)| < \mu < 1$ tal que si $z \in \mathbb{D}(z_0, r)$ entonces

$$\frac{|f(z) - f(z_0)|}{|z - z_0|} < \mu < 1.$$

Como sabemos que $f(z) = z$ por ser un punto fijo, la desigualdad anterior quedará de la forma $|f(z) - z_0| < \mu|z - z_0|$. Es decir, el valor $f(z)$ es más próximo a z_0 que el punto z (con un factor $\mu < 1$). Este argumento podemos aplicarlo n veces para obtener $|f^n(z) - z_0| < \mu^n|z - z_0|$. Como $\mu < 1$, concluimos que

$$\lim_{n \rightarrow \infty} f^n(z) = z_0$$

para todo $z \in \mathbb{D}(z_0, r)$.

Los puntos fijos repulsivos se estudian de forma parecida pero a través de f^{-1} ya que si son repulsivos por f , entonces son puntos fijos atractores por f^{-1} . Pero esto no implica que las órbitas se alejen para siempre, se podría llegar a un punto después de varias iteraciones en el que podrían volver. \square

Definición 2.2.3. Sea z_0 un punto fijo atractor de la función f . Se define la *cuenca de atracción* de z_0 como el conjunto de todos los puntos de \mathbb{C} que convergen a z_0 . Es decir,

$$A(z_0) = \{z \in \mathbb{C} \mid f^n(z) \xrightarrow{n \rightarrow \infty} z_0\}.$$

Además, se define la *cuenca inmediata de atracción* $A^*(z_0)$ como la componente conexa de $A(z_0)$ que contiene a z_0 .

Un ejemplo de cuenca de atracción para el polinomio $f(z) = z^2 - 1$ se puede ver en la Figura 2.1. En negro podemos ver la cuenca de atracción del 0 y del -1, $A(0, -1)$. Mientras que en rojo podemos ver la cuenca de atracción del infinito, $A(\infty)$, ya que como hemos visto dado un polinomio, el infinito es un punto súper atractor. Este dibujo es conocido por el nombre de *la basílica*.

En el siguiente teorema Koenigs demostró que si el multiplicador es distinto de 0 y tiene módulo distinto de 1 entonces la función es conjugada a la parte lineal.

Teorema 2.2.4. (Teorema de linealización de Koenigs) Si el multiplicador λ satisface que $|\lambda| \neq 0, 1$, entonces existe un cambio de coordenadas local y holomorfo $w = \phi(z)$, con $\phi(0) = 0$, tal que $\phi \circ f \circ \phi^{-1}$ es la función lineal $w \rightarrow \lambda w$ para todo w en un entorno del origen. Además, ϕ es único módulo de multiplicación por una constante distinta de cero.

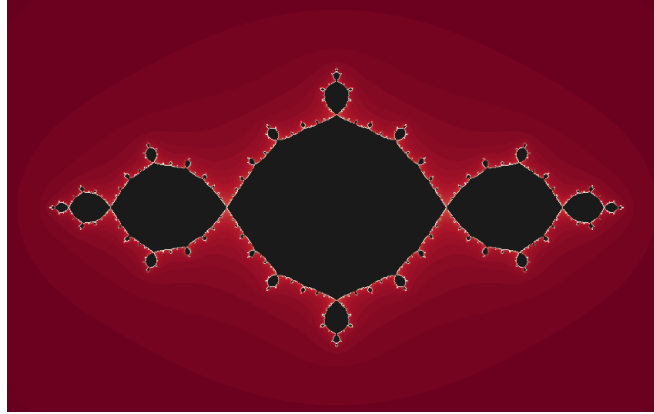


Figura 2.1: Conjunto lleno de Julia del polonimio $f(z) = z^2 + c$ para el valor $c = -1$ donde los puntos negros se corresponden con la cuenca de atracción del $A(0, -1)$ y los rojos con $A(\infty)$.

Ahora vamos a ver un resultado importante para el caso $\lambda = 0$, es decir, para los puntos súper atractores y que fue demostrado por Bottcher en 1904 (ver [16]). Observar que en este caso la parte lineal de $f(z)$ es 0. Por lo tanto, no podemos esperar que la función sea linearizable, a no ser que sea constante igual a 0.

Teorema 2.2.5. (Teorema de Bottcher) *Supongamos que $f(z) = a_n z^n + a_{n+1} z^{n+1} + \dots$, donde $n \geq 2$. Entonces existe un cambio de coordenadas local y holomorfo $w = \phi(z)$ que conjuga f con la aplicación de la n -ésima potencia $w \rightarrow w^n$ en todo el entorno de $\phi(0) = 0$. Además, ϕ es único módulo multiplicación por una raíz $(k-1)$ -ésima de la unidad.*

Caso indiferente parabólico

Los puntos fijos neutrales son aquellos puntos fijos donde la derivada de f evaluada en z_0 es igual a $e^{i\alpha}$, es decir, $f'(z_0) = e^{i\alpha}$ siendo $\alpha \in [0, 2\pi)$. El comportamiento de las órbitas alrededor de estos puntos es más complejo que el de los puntos atractores y repulsores que hemos visto anteriormente. Además influye también el valor de α y de su carácter racional o irracional. Para entenderlo mejor veremos un ejemplo.

Supongamos que $\alpha = 0$ entonces $f'(z_0) = 1$. Sea $f(z) = z + z^2$ con $z_0 = 0$ vemos que es un punto fijo con derivada evaluada en 0 igual a 1, $f'(0) = 1$. Además, podemos comprobar que si consideramos como condición inicial cualquier punto en el eje real positivo, la órbita tiende a infinito. Mientras que si consideramos un punto del eje real negativo como condición inicial, su órbita se acerca al origen sobre el eje real negativo. Por tanto, podemos concluir que en el entorno de un punto neutral hay una mezcla de órbitas que convergen al punto parabólico y que escapan del punto parabólico.

Si $\alpha \in \mathbb{Q}$, α será de la forma $\alpha = \frac{p}{q} \neq 0$ por ser un número racional. En este caso $\lambda = e^{2\pi i \alpha}$ sería una raíz q -ésima de la unidad. Si tomamos la q -ésima iteración de la función, reducimos el caso a $\lambda = 1$. Es decir, aplicando una conjugación que envíe z_0 a 0, consideramos funciones de la forma

$$f(z) = z + az^{m+1} + \mathcal{O}(z^{m+2}),$$

con $m > 0$ y $a \neq 0$. El entero $m + 1$ se conoce como *multiplicidad* del punto parabólico con $\lambda = 1$.

El comportamiento descrito anteriormente nos muestra que la función no va a poder ser conjugada a la parte lineal, es decir, no va a poder ser linearizable. En general un teorema que nos permite entender la dinámica cerca de los puntos parabólicos es el teorema de Leau-Fatou. Para poder ver este teorema necesitaremos ver la definición de pétalo.

Definición 2.2.6. Sea N un entorno de z_0 que no contiene ningún punto crítico, es decir, $f : N \rightarrow f(N) = N'$ y $f^{-1} : N' \rightarrow N$ está bien definida.

Sea U un conjunto abierto y conexo que satisface que $\overline{U} \subset N \cap N'$. Entonces, se dice que U es un *pétalo atractor* en z_0 si:

- $f(\overline{U}) \subset U \cup \{z_0\}$.
- $\bigcap_{k \geq 0} f^k(\overline{U}) = \{z_0\}$.

Se dice que U' es un *pétalo repulsor* si U' es un pétalo atractor para f^{-1} .

El siguiente teorema podemos consultarlo en [1].

Teorema 2.2.7. (Teorema de la flor de Leau-Fatou) Supongamos que f tiene un punto fijo parabólico con multiplicador $\lambda = 1$ en el origen y multiplicidad $m + 1$. Entonces, existen $2m$ pétalos $\{U_j\}_{j=1}^{2m}$, numerados cíclicamente alrededor del origen tal que U_j es atractor o repulsor según j sea impar o par. Cada pétalo U_j intersecta solo con sus dos entornos inmediatos U_{j-1} y U_{j+1} y son disjuntos del resto. Los pétalos se pueden elegir de la forma tal que la unión

$$U_1 \cup \dots \cup U_{2m} \cup \{0\}$$

forma un entorno abierto del origen (ver Figura 2.2).

Definición 2.2.8. Sea z_0 un punto parabólico de la función f . Se define la *cuenca parabólica* de z_0 como el conjunto de todos los puntos de $\widehat{\mathbb{C}}$ que convergen a z_0 , estando z_0 en la frontera.

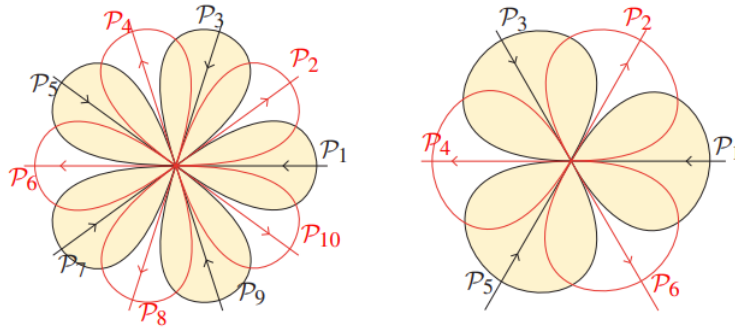


Figura 2.2: Distribución de los pétalos invariantes alrededor del punto parabólico con multiplicador $\lambda = 1$ y multiplicador 5 (figura de la izquierda) y 3 (figura de la derecha). Figura tomada de [1, Figura 3.5]

Caso indiferente irracional

En este caso vamos a ver el gran problema que hay para saber si la dinámica va a ser conjugada a la parte lineal o no. En este caso si la función es conjugada a la parte lineal vamos a tener una rotación rígida. Este es un problema difícil dentro de la dinámica holomorfa y ha sido estudiada por diversas personas.

Cremer en el año 1927 demostró que en algunos casos la dinámica no puede ser conjugada a la parte lineal [16]. En tal caso, decimos que ese punto es un punto de Cremer. Por otra parte, Siegel en el año 1942 [22] demostró que si el punto fijo tiene multiplicador λ de tipo diofántico (ver Definición 2), entonces sí es linearizable. En tal caso decimos que el punto fijo es un punto de Siegel. Más adelante, está condición la mejoró Brjuno con los puntos tipo Brjuno (esto lo estudiaremos en el Teorema 2.2.11).

Definición 2.2.9. Sea f una función holomorfa y z_0 un punto fijo (o periódico) con multiplicador $\lambda = e^{2\pi i\theta}$. Si f no es linearizable alrededor de z_0 decimos que z_0 es un *punto de Cremer*. Si f es linearizable alrededor de z_0 decimos que z_0 es un *punto de Siegel*. El máximo dominio U , con $z_0 \in U$, donde f es conjugada con la función $z \mapsto \lambda z$, es llamado *disco de Siegel* (ver Figura 2.3).

El siguiente teorema nos da una propiedad fundamental de los discos de Siegel y establece su relación con los puntos críticos.

Teorema 2.2.10. *Todo punto de la frontera de un Disco de Siegel maximal U pertenece a la clausura de alguna órbita crítica.*

La idea de la demostración del teorema anterior es que si se coge un disco en la frontera y lo

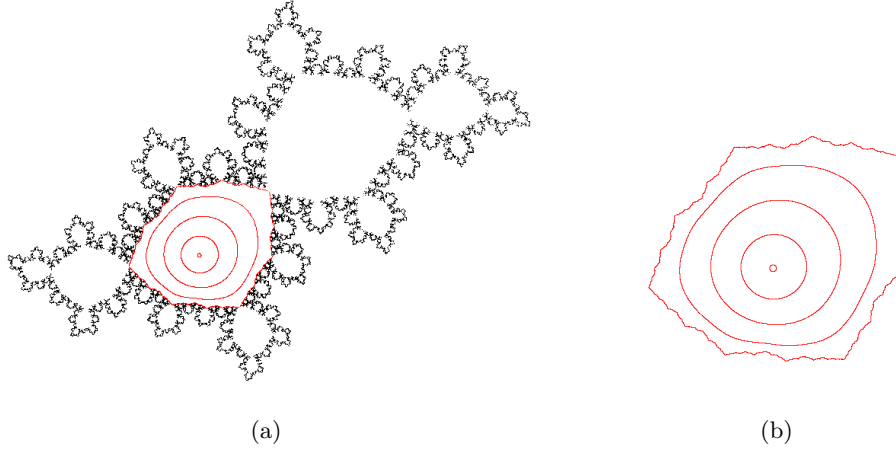


Figura 2.3: Disco de Siegel del polinomio $p(z) = \lambda z + \lambda z^2$ siendo $\lambda = e^{2\pi i\theta}$ y $\theta = \frac{\sqrt{5}-1}{2}$. La frontera se ha obtenido iterando el punto crítico $p = \frac{-\lambda}{2}$. Este dibujo está implementado con el algoritmo de Henriksen explicado en la Sección 5.3.

iteras hacia atrás, si no hay imagen iterada de algún punto crítico, lo podemos repetir cuantas veces queramos hasta cubrir toda la frontera del disco de Siegel. Como no hay puntos críticos, el dominio de linealización se puede ampliar y el disco de Siegel no es maximal. La demostración detallada se puede consultar en [16].

En el siguiente teorema asumimos que el punto fijo es 0, pero no es una falta de generalidad puesto que podemos siempre aplicar una conjugación.

Teorema 2.2.11. (Siegel - Brjuno) Sea $\lambda = e^{2\pi i\alpha}$ con $\lambda \in \mathbb{R} \setminus \mathbb{Q}$. Si α es de tipo Brjuno, entonces dada cualquier función holomorfa de la forma

$$f(z) = e^{2\pi i\alpha} z + a_2 z^2 + \dots$$

es linearizable en un entorno de 0.

La idea de la demostración del teorema anterior está basada en el *problema de los pequeños denominadores*. Este problema consiste en que si desarrollamos en serie de potencias, $f(z) = \lambda z + \sum_{n \geq 2} f_n z^n$, entonces la aplicación linealizante $h(z) = z + \sum_{n \geq 2} h_n z^n$ está determinada por la ecuación

$$f(h(z)) = h(\lambda z)$$

y es única. Despejando orden por orden, se pueden obtener los coeficientes h_n los cuales cumplen la siguiente relación de recurrencia

$$h_n = \frac{1}{\lambda^n - \lambda} P_n(h_2, \dots, h_{n-1}, \lambda, f_2, \dots, f_n),$$

siendo P_n un polinomio. El problema de la demostración viene dado por los términos $\lambda^n - \lambda$ que dificultan la convergencia de la serie formal dada por el límite de $h(z)$.

Siegel consiguió resolver este problema de los pequeños divisores para todo $\alpha \in \mathbb{R}$ que cumpliera una *condición diofántica* en 1942. La idea detrás de por qué para los números diofánticos existe solución se debe a que estos números son los que peor se aproximan por números racionales. Por lo tanto, son los que menos veces $\lambda^n - \lambda$ se acerca a 0. Más tarde, por los mismos métodos, Brjuno consiguió mejorar la condición aritmética de Siegel en la *condición de Brjuno*. Siegel y Brjuno proporcionaron una condición suficiente para saber si una función holomorfa era localmente linealizable cerca de un punto fijo de multiplicador λ . A partir de aquí apareció la cuestión de si era una condición necesaria. ¿Hacía falta que fuera de tipo Brjuno para que fuera linealizable? Este problema en general es muy difícil pero Yoccoz consiguió demostrar que esa condición no solo es suficiente, sino necesaria para la familia cuadrática $P_\lambda(z) = \lambda z(1 - z)$ en 1987 (ver [15]).

Teorema 2.2.12. (Yoccoz) Si $\sum_{n \geq 1} \frac{\log(q_{n+1})}{q_n} = \infty$ (no es de tipo Brjuno), entonces la función cuadrática

$$f(z) = z^2 + e^{2\pi i \alpha} z$$

tiene la propiedad de que todo entorno del origen contiene infinitas órbitas periódicas. En particular, el origen es un punto de Cremer.

Por tanto, el teorema anterior tiene una gran importancia y formó parte del trabajo por el que Yoccoz obtuvo la medalla Fields en 1994. Este teorema se demostró hace aproximadamente 30 años y en la actualidad aun no se ha podido extender nada. Se conjetura que es cierto para cualquier función racional, sin embargo, se trata de un problema abierto que no ha sido resuelto ni para polinomios cúbicos.

Capítulo 3

Teoría global

En este capítulo veremos los conceptos más importantes de la teoría global.

3.1. Conjuntos de Julia y Fatou

3.1.1. Definición para polinomios

Empezaremos explicando la definición de los conjuntos de Fatou y Julia para el caso particular de polinomios.

Definición 3.1.1. El *conjunto lleno de Julia* del polinomio P es el conjunto de puntos con órbita acotada, esto es,

$$K(P) = \{z \in \mathbb{C} \mid P^n(z) \not\rightarrow \infty\} = \mathbb{C} \setminus A(\infty).$$

Definimos el *conjunto de Julia* de P , que denotamos como $J(P)$, como la frontera de $K(P)$. Finalmente, definimos el *conjunto de Fatou* como el complementario de Julia,

$$F(P) = \mathbb{C} \setminus J(P).$$

3.1.2. Definición general

Para poder ver y entender la definición general de Fatou y Julia primero introduciremos algunos conceptos y resultados.

Definición 3.1.2. Sea $\mathcal{F} = \{f^n\}_{n \in \mathbb{N}}$ una familia de funciones holomorfas definidas en un dominio D . Se dice que \mathcal{F} forma una *familia normal* en D si toda sucesión de funciones de \mathcal{F} tiene una parcial que converge uniformemente sobre compactos de D .

Informalmente se puede decir que la familia de iterados $\{f^n\}_{n \in \mathbb{N}}$ es una familia normal si en un entorno de z , todos los puntos se comportan bajo iteración de manera análoga a z .

Definición 3.1.3. Sea f una función racional. Se dice que z_0 es un *punto normal* o *estable* de f si para algún entorno $U \subset \widehat{\mathbb{C}}$ de z_0 la familia de iterados $\{f^n\}_{n \in \mathbb{N}}$ forma una familia normal en U .

Teorema 3.1.4. [26] (**Teorema de Montel**) Sea $\mathcal{F} = \{f^n\}_{n \in \mathbb{N}}$ una familia de iterados de una función holomorfa f definida en un dominio D . Si $\bigcup_{n \in \mathbb{N}} f^n(D)$ omite al menos tres puntos en $\widehat{\mathbb{C}}$ entonces \mathcal{F} es una familia normal en D .

Ahora ya podemos definir los conjuntos de Fatou y Julia.

Definición 3.1.5. Sea f una función holomorfa. Se define el *conjunto de Fatou* de f , $F(f)$, como el conjunto de puntos normales de la función f . Se define el *conjunto de Julia* de f , $J(f)$, como el conjunto de puntos no normales de f , es decir, el complementario del conjunto de Fatou.

Una primera observación de la definición anterior es que el conjunto de Fatou es un conjunto abierto, mientras que el conjunto de Julia es un conjunto cerrado.

Unos resultados importantes que se siguen del teorema de Montel son que toda cuenca de atracción de un punto periódico (súper)atractor o parabólico y todo disco de Siegel pertenece al conjunto de Fatou. En particular, los puntos periódicos (súper)atractores y los puntos de Siegel pertenecen al conjunto de Fatou. Mientras que los puntos parabólicos y los puntos de Cremer pertenecen al conjunto de Julia. Para poder consultar las demostraciones completas es necesario ver [2], pero comentaremos estas propiedades en el apartado 3.1.3.

Ahora vamos a ver una proposición que dice que los conjuntos de Julia y Fatou son invariantes. Podemos consultar la demostración en [17].

Proposición 3.1.6. Los conjuntos de Julia y Fatou son totalmente invariantes. Es decir, si $f^{-1}(J(f)) \subset J(f)$ y $f(J(f)) \subset J(f)$. Y también lo mismo para el conjunto de Fatou, es decir, si $f^{-1}(F(f)) \subset F(f)$ y $f(F(f)) \subset F(f)$. Además, el $F(f) = F(f^p)$.

Definición 3.1.7. Una *componente de Fatou* es cualquier componente conexa del conjunto de Fatou $F(f)$.

Como el conjunto de Fatou es completamente invariante la imagen por f de toda componente de Fatou U es otra componente de Fatou U' . Por tanto, podemos hacer la siguiente clasificación de las componentes del conjunto de Fatou.

Definición 3.1.8. Toda componente de Fatou U de una función holomorfa f es:

- *periódica*, si existe $n \geq 1$ tal que $f^n(U) = U$; o
- *preperiódica*, si U no es periódica pero existe $n > 1$ tal que $f^n(U)$ es una componente periódica; o
- *errante*, si no es periódica ni preperiódica.

El siguiente resultado realizado por Sullivan [24] nos indica que una función racional no tiene componentes de Fatou errantes. Este resultado utiliza la teoría de funciones casiconformes, que es una herramienta muy importante dentro de la dinámica holomorfa pero no lo vamos a ver en este trabajo.

Teorema 3.1.9. (Teorema de ausencia de conjuntos errantes) *Toda componente de Fatou U de una función racional es periódica o preperiódica.*

A continuación queremos clasificar las componentes de Fatou periódicas. Primero definimos anillo de Herman (ver Figura 3.1). Los parámetros utilizados para dibujar el anillo de Herman los hemos obtenido de [17].

Definición 3.1.10. Sea una componente conexa de Fatou. Se dice que U es un *anillo de Herman* de una función racional R si U es conformemente equivalente a un anillo

$$A = \{z \in \mathbb{C} \mid r_1 < |z| < r_2\},$$

con $0 \leq r_1 < r_2$, y $R|_U$ es analíticamente conjugada a una rotación rígida $z \mapsto \lambda z$ con $\lambda = e^{2\pi i\theta}$ y θ irracional.

La frontera del anillo de Herman se obtiene iterando los dos puntos críticos de la función utilizada. Es decir, que una órbita crítica distinta se acumula en cada frontera. Este resultado se puede consultar en [21].

Veamos ahora una clasificación de las componentes de Fatou para funciones racionales que podemos consultar en [2].

Teorema 3.1.11. (Teorema de Clasificación) *Sea U una componente de Fatou periódica de una función racional de grado ≥ 2 . Entonces solo una de las siguientes afirmaciones es cierta:*

1. (caso atractor) U contiene un punto atractor periódico de periodo p ;
2. (caso súper atractor) U contiene un punto súper atractor z_0 ;

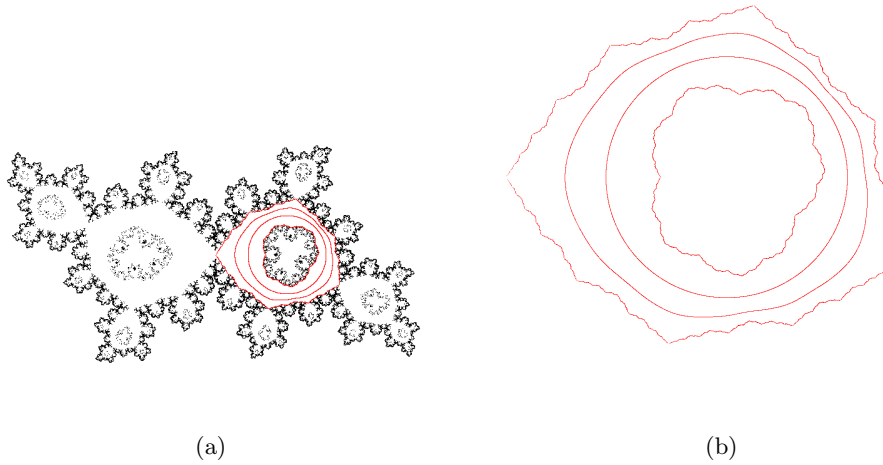


Figura 3.1: Anillo de Herman de la función racional $p(z) = \lambda z^2 \frac{az + 1}{z + a}$ siendo $\lambda = e^{2\pi it}$, $t = 0,61517321588\dots$ y $a = 0,25$. La frontera se ha obtenido iterando sus dos puntos críticos, $p_1 = -1,82793442287247\dots$ y $p_2 = -0,547065577127525\dots$

3. (caso parabólico) ∂U contiene un punto fijo neutral de f^n con multiplicador igual a 1;
4. U es un disco de Siegel;
5. U es un anillo de Herman.

Si en lugar de tener funciones racionales, tuviéramos polinomios no podríamos tener anillos de Herman. Esto se debe al siguiente teorema cuya demostración podemos consultar en [6].

Teorema 3.1.12. (Teorema del módulo máximo) Sea $G \subset \mathbb{C}$ un conjunto acotado y abierto. Supongamos que $f : G \rightarrow \mathbb{C}$ es una función continua en la clausura de G , \bar{G} , y es holomorfa en G . Entonces,

$$\max\{|f(z)| : z \in \bar{G}\} = \max\{|f(z)| : z \in \partial G\}.$$

La idea de la demostración se basa en suponer que tenemos un anillo de Herman de periodo p y en considerar la función f^p . Por tanto tenemos anillos de Herman fijos. Toda curva invariante se enviará a ella misma, entonces por el Teorema del módulo máximo 3.1.2, si f no tiene polos, el interior de esta curva se envía al interior de la curva. Por tanto, por el Teorema de Montel 3.1.4, la región acotada por la curva pertenece a Fatou. Entonces, se forma una componente de Fatou que no puede tener un agujero y no puede ser doblemente conexo.

Otro resultado muy importante es que toda componente de Fatou periódica está relacionada con al menos un punto crítico.

Proposición 3.1.13. *Toda componente de Fatou periódica de una función racional está relacionada con, al menos, un punto crítico:*

- *todo ciclo (súper)atractor tiene un punto crítico en la base inmediata de atracción. En particular, si z_0 es súper atractor, entonces él mismo es un punto crítico;*
- *dado un disco de Siegel U , existe un punto crítico cuya órbita se acumula en la frontera de U ;*
- *dado un anillo de Herman U , existen dos puntos críticos cuya órbita se acumula en la frontera de U .*

La demostración de la proposición anterior y de la siguiente podemos encontrarlas en [21].

Teorema 3.1.14. (Desigualdad de Fatou-Shishikura) *Se cumple que*

$$n_{AB} + n_{PB} + n_{SD} + 2n_{HR} + n_{Cremer} \leq 2(d - 1)$$

donde n_{AB} , n_{PB} , n_{SD} , n_{HR} , n_{Cremer} denota el número de ciclos atractores, parabólicos, de Siegel, de Herman y de Cremer, respectivamente.

3.1.3. Propiedades

En esta sección, veremos cuáles son las propiedades de los conjuntos de Julia y Fatou. A través del análisis de estas propiedades, obtendremos una comprensión más profunda del comportamiento de estos conjuntos y de su importancia en la teoría global. Las demostraciones de las propiedades siguientes las podemos encontrar en [17].

Teorema 3.1.15. *Sea f una función racional con grado mayor o igual que 2. Sea $J(f)$ el conjunto de Julia y sea $F(f)$ el conjunto de Fatou. Se cumple que:*

- (a) *Si $z \in J(f)$ y U es un entorno de z , la unión de todos los iterados de U recubre toda la esfera de Riemann con la excepción de, como mucho, dos puntos.*
- (b) *Los puntos periódicos atractores y sus cuencas de atracción pertenecen al conjunto de Fatou.*
- (c) *Todos los puntos parabólicos pertenecen al conjunto de Julia. Los puntos de Siegel pertenecen al conjunto de Fatou. ya que no son parabólicos.*

- (d) Si z_0 es un punto de $J(f)$, el conjunto de todas sus antiimágenes de z_0 es denso en $J(f)$.
- (e) Los puntos repulsores pertenecen al conjunto de Julia. Además, forman un conjunto denso en $J(f)$.
- (f) El conjunto de Julia de f es distinto del vacío, por tanto $J(f) \neq \emptyset$.

Demostración. Primero demostraremos (a). Esta propiedad es consecuencia del Teorema de Montel (Teorema 3.1.4). Si la familia de iterados omite al menos tres puntos entonces z es un punto normal. Esto es una contradicción con que $z \in J(f)$.

A continuación demostraremos (b). En primer lugar, observamos que la cuenca de atracción de todo punto periódico atractor de periodo $p \geq 1$ es un conjunto abierto. Por tanto, si z pertenece a la cuenca de atracción de un punto fijo z_0 de periodo $p = 1$, para cualquier entorno U de z lo suficientemente pequeño que esté contenido en la cuenca de atracción, los iterados de f en U convergen a la función constante $g(z) \equiv z_0$. Por lo tanto, z es normal y pertenece a $F(f)$. Si z pertenece a la cuenca de atracción de un punto periódico z_0 de periodo $p > 1$, utilizamos el mismo argumento pero en lugar de f tomamos f^p (observad que por la Proposición 3.1.6 tenemos que $F(f) = F(f^p)$). Esto finaliza la demostración de (b).

Ahora, demostraremos (c). Un punto es periódico neutral parabólico cuando el multiplicador de la órbita periódica es $e^{2\pi i \frac{p}{q}}$. Estos puntos son no normales ya que en cualquier entorno de dichos puntos se pueden encontrar direcciones atractoras y repulsoras por tanto no existe el límite continuo de la familia de iterados. Cuando estamos ante un punto de Siegel, existe un entorno en el cual las órbitas giran alrededor del punto sin abandonarlo. Entonces por el Teorema de Montel 3.1.4, la familia de iterados en este entorno sí es normal.

La propiedad (d) es una consecuencia directa de la propiedad (a). Sea $w \in J(f)$ y U un entorno de w . Tenemos que demostrar que U contiene alguna preimagen de z_0 . Dado que las imágenes de U tienen que cubrir todo $\widehat{\mathbb{C}}$ (con la excepción de como mucho dos puntos que pertenecen al conjunto de Fatou [16, lemma 4.9]), por la propiedad (a) debe existir $N > 0$ tal que $z_0 \in f^N(U)$. Esto significa que U tiene algún punto que bajo N iteraciones es enviado a z_0 , es decir, es una preimagen.

Para demostrar (e) lo haremos por contradicción. Por tanto, supongamos que existe un entorno U del punto periódico repulsor z_0 y es normal. Todos los puntos de U se alejan, a excepción de z_0 . Pero, si z_0 pertenece al conjunto de Fatou, entonces la familia de iterados debe converger sobre los compactos a una función holomorfa g de forma que $g(z_0) = z_0$. Pero, como g en z_0 es continua, por definición se debería cumplir que $\lim_{z \rightarrow z_0} g(z) = z_0$. Pero esto es una contradicción ya que z_0 es repulsor y los iterados de U se alejan de él.

Por otra parte, la demostración de que los puntos periódicos son densos en el conjunto de

Julia es más complicada. Se puede consultar en [26]. La demostración está relacionada con una aplicación clásica del Teorema de Montel sugerida por Fatou.

La propiedad (f) es más complicada de demostrar, pero la idea de la demostración se basa en que las funciones racionales tienen infinitos puntos periódicos que pertenecen al conjunto de Julia. La dificultad consiste en comprobar si los puntos periódicos son neutros o repulsores, ya que ver que hay infinitos puntos periódicos es sencillo al resolver la ecuación $f^n(z) = z$. Por la desigualdad de Fatou-Shishikura del Teorema 3.1.14 sabemos que hay como mucho $2d - 2$ ciclos no repulsores. Por tanto $J(f)$ será no vacío. Para ver la demostración completa consultar [2]. □

Capítulo 4

Conjunto de Mandelbrot

En ese capítulo se pretende proporcionar los resultados necesarios que nos permiten justificar los algoritmos utilizados para dibujar el conjunto de Mandelbrot en el capítulo siguiente.

4.1. Definición de \mathcal{M}

En primer lugar veamos un resultado conocido como *dicotomía fundamental* que nos dará la base para poder definir el conjunto de Mandelbrot. Podemos encontrar la demostración en [17].

Teorema 4.1.1. (*Dicotomía fundamental*) *Sea la familia cuadrática $Q_c(z) = z^2 + c$ y sea K_c el conjunto lleno de Julia. Entonces se cumple una de las siguientes afirmaciones:*

1. *o bien la órbita del punto crítico $z = 0$ es acotada y K_c es conexo;*
2. *o bien la órbita del punto crítico $z = 0$ escapa a infinito y K_c es un conjunto de Cantor.*

Por tanto, a partir del teorema anterior surge la definición del conjunto de Mandelbrot ya que es el conjunto de puntos para los que K_c es conexo.

Definición 4.1.2. El *conjunto de Mandelbrot* (ver Figura 4.1) está formado por todos los valores $c \in \mathbb{C}$ cuyas órbitas en el punto 0 no tienden a infinito bajo $Q_c = z^2 + c$. Además, como hemos visto anteriormente, también está formado por todos los valores $c \in \mathbb{C}$ tales que el conjunto de Julia lleno es conexo. Es decir,

$$\mathcal{M} = \{c \in \mathbb{C} \mid Q_c^n(0) \not\rightarrow \infty\} = \{c \in \mathbb{C} \mid K_c \text{ es conexo}\}.$$

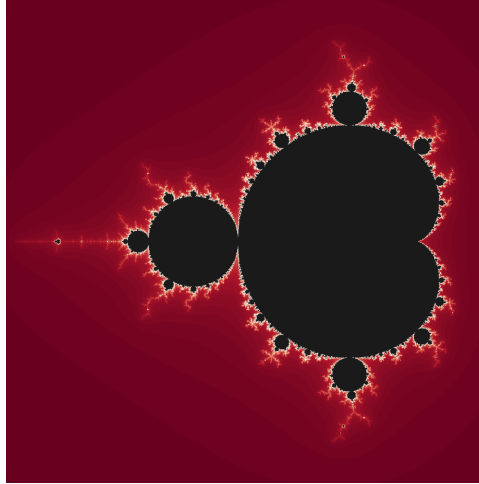


Figura 4.1: Conjunto de Mandelbrot.

4.2. Propiedades

Algunas propiedades que cumple el conjunto de Mandelbrot son las siguientes.

Proposición 4.2.1. *Sea $Q_c(z) = z^2 + c$ y $R = \max\{2, c\}$. Si z satisface $|z| > R$ entonces $\lim_{n \rightarrow \infty} Q_c^n(z) = \infty$.*

Demostración. Podemos observar que si $|z| > R$ entonces $|z| > 2$ y $|z| > c$. Aplicando la desigualdad $|a + b| \geq ||a| - |b||$ tenemos que $\frac{|z^2 + c|}{|z|} \geq |z| - \frac{|c|}{|z|} \geq |z| - 1$. Como $|z| - 1 > 1$ tenemos que $|z^2 + c| > c$. Por tanto, $|Q_c| = |z^2 + c| > (|z| - 1) \cdot |z| > |z|$. En particular, a cada paso $|z| - 1$ se hace más grande y podemos concluir que $Q_c^n(z)$ converge a infinito.

□

Teorema 4.2.2. [17] *El conjunto de Mandelbrot \mathcal{M} cumple que:*

1. \mathcal{M} es cerrado.
2. \mathcal{M} es acotado.
3. \mathcal{M} es conexo.

Demostración. Para demostrar la propiedad 1 vamos a ver que el complementario de \mathcal{M} es abierto. Sea c_0 un punto en el complementario. Vamos a ver que un entorno de puntos U de

c_0 pertenece también al complementario. Es decir, los valores de c para los que $Q_c^n(0)$ escapa a infinito es abierto. Como para un valor de c_0 la órbita tiende a infinito, $Q_{c_0}^n(0) \rightarrow \infty$, existe una $k \in \mathbb{N}$ tal que $Q_{c_0}^k(0) > R$, siendo $R = \max\{2, c_0\}$. Por continuidad, existe un entorno suficientemente pequeño de 0, tal que si c está dentro del entorno entonces $|Q_c^k(0)| > R$. Si definimos $R_c = \max\{2, |c|\}$, por continuidad si c es suficientemente próximo a c_0 entonces pertenece al complementario de \mathcal{M} y es abierto. Por tanto, \mathcal{M} es cerrado.

En la propiedad 2, por la Proposición 4.2.1 y el Teorema 4.1.1 sabemos que si $|c| > 2$ la órbita de 0 escapa a infinito y el conjunto de Julia es un conjunto de Cantor. Por tanto, el conjunto de Mandelbrot es acotado porque está contenido en el disco cerrado de radio 2

La demostración de la propiedad 3 no la vamos a ver. La idea es que se puede caracterizar \mathcal{M} como la intersección de infinitos cerrados conexos encajados (consultar [17]). \square

4.3. Componentes hiperbólicas

Al mirar el dibujo del conjunto de Mandelbrot vemos que está formado por una especie de discos deformados unidos entre ellos y que contienen unas ramificaciones que se parecen a unas antenas. En esta sección describiremos estos discos de forma más precisa. Del lema y la proposición que vamos a enunciar a continuación podemos consultar sus demostraciones en [17].

Lema 4.3.1. *Si el polinomio Q_c tiene un órbita periódica atractora, entonces es única y $c \in \mathcal{M}$.*

Demostración. Esta demostración es trivial. La familia Q_c tiene un único punto crítico dado por $z = 0$, a parte de infinito. Entonces, si existe una órbita atractora, 0 estará en su cuenca inmediata de atracción. Además, 0 no puede escapar a infinito. Como todo ciclo atractor tiene un punto crítico en su cuenca inmediata de atracción (Proposición 3.1.2) concluimos que como máximo hay un ciclo atractor. \square

Ahora ya sabemos que si para un conjunto de parámetros hay una órbita periódica atractora entonces el conjunto pertenece al conjunto de Mandelbrot. Consideramos el conjunto $\mathcal{H} \subset \mathcal{M}$ definido de la forma

$$\mathcal{H} = \{c \in \mathbb{C} \mid Q_c \text{ tiene una órbita periódica atractora}\}.$$

La proposición que vamos a ver ahora asegura que estos parámetros se encuentran en el interior del conjunto de Mandelbrot y no en la frontera.

Proposición 4.3.2. *Si \mathcal{H} y \mathcal{M} son los que hemos definido anteriormente, entonces se cumple que:*

1. El conjunto \mathcal{H} es abierto y por tanto está en el interior de \mathcal{M} .
2. El conjunto \mathcal{H} tiene infinitas componentes conexas. Cada una de ellas posee un invariante: el periodo del ciclo atractor.

La demostración de la proposición anterior la podemos encontrar en [17].

Definición 4.3.3. Se llaman *componentes hiperbólicas de \mathcal{M}* a las componentes conexas de \mathcal{H} .

Estas componentes que hemos definido se corresponden con los discos que hemos mencionado al principio de la sección. Se cree que la unión de todas las componentes de \mathcal{H} forman el interior de \mathcal{M} pero nadie hasta la fecha ha sido capaz de demostrarlo. ¿Sería posible que en algún lugar del conjunto de Mandelbrot hubiera un disco abierto en los que los valores de c no estén asociados a una órbita periódica atractora? La respuesta se cree que es “no”, pero es uno de los problemas abiertos más importantes de la dinámica compleja. Esta conjetura se conoce como la *Conjetura Hiperbólica*.

Conjetura 1. (Conjetura hiperbólica) $\mathcal{H} = \text{int}(\mathcal{M})$.

Alrededor del cardioide principal emergen tantas componentes hiperbólicas como números racionales existen. A su vez, de cada componente hiperbólica del cardioide principal también emergen otras componentes hiperbólicas en su frontera. Por tanto, de cada componente hiperbólica emergen infinitas componentes hiperbólicas asociadas a nuevas bifurcaciones que se conocen como *racionales o de multiplicación del periodo*. Este hecho fue estudiado por Douady y Hubbard en [25] en el que podemos encontrar el siguiente teorema donde se define la parametrización.

Teorema 4.3.4. *Sea Ω una componente hiperbólica de periodo $N > 0$ de \mathcal{M} . Entonces, el multiplicador del ciclo atractor define una función holomorfa y biyectiva*

$$\rho_\Omega : \Omega \rightarrow \mathbb{D}$$

que se extiende de manera continua a la frontera.

Es decir, la parametrización asigna a cada parámetro el multiplicador del ciclo atractor y es biyectiva. Del teorema se deduce que cada Ω de \mathcal{M} tiene un único *centro*, es decir, un único parámetro c para el cual Q_c tiene un ciclo súper atractor. También se deduce que para cada número racional $\frac{p}{q}$, existe un punto c en la frontera de Ω con multiplicador $e^{2\pi i \frac{p}{q}}$. Conectada a este punto, se puede encontrar una componente hiperbólica de periodo qN .

Para mencionar otro de los problemas abiertos que existen sobre el conjunto de Mandelbrot, primero necesitamos introducir unas definiciones.

Definición 4.3.5. Sea un compacto K , se dice que es *localmente conexo en el punto z* si para todo entorno U de z la intersección $K \cap U$ es conexo. Se dice que es *localmente conexo* si lo es en todos sus puntos.

Definición 4.3.6. Se dice que un conjunto $A \in \mathbb{C}$ es *arcoconexo* si dados $a, b \in A$ existe un arco $\gamma : [0, 1] \rightarrow A$ tal que $\gamma(0) = a$ y $\gamma(1) = b$.

Conjetura 2. \mathcal{M} es *localmente conexo (MLC)*.

Esta conjetura que se conoce como MLC y es uno de los problemas más importantes del área de los sistemas dinámicos holomorfos. Se sabe que esta conjetura es cierta para algunos puntos de la frontera del conjunto, pero no se ha podido demostrar para todo el conjunto de Mandelbrot. Otro problema abierto es si \mathcal{M} es arcoconexo.

Si el conjunto de Mandelbrot fuera localmente conexo, esta afirmación implicaría que la conjetura hiperbólica sería cierta, pero esto es muy difícil de demostrar. Otra implicación sería que el diámetro de toda componente hiperbólica tiende a 0 cuando el periodo tiende a infinito. De esto se sigue que la frontera del conjunto de Mandelbrot está contenido en la clausura de la unión de todos los centros de las componentes hiperbólicas. Por lo tanto, dibujando los centros de las componentes hiperbólicas podríamos dibujar el conjunto de Mandelbrot.

Capítulo 5

Algoritmos

En este capítulo se presentarán tres algoritmos: el algoritmo de escape, el algoritmo de iteración inversa y el algoritmo de Henriksen. Para cada uno de ellos vamos a explorar diferentes casos y después seleccionaremos la mejor figura de cada algoritmo para cada caso y las compararemos entre sí. Los distintos casos que vamos a comparar son los siguientes:

- Dos casos hiperbólicos de la función $Q_c(z) = z^2 + c$ con $c = 0,36 + 0,34i$ y $c = -0,672 + 0,337i$.
- Un caso cercano a parabólico con $c = 0,285 + 0,01i$.
- Un caso parabólico de periodo 3 con $f(z) = e^{2\pi i \frac{1}{3}}z + z^2$.
- Un disco de Siegel con $f(z) = \lambda z + \lambda z^2$, siendo $e^{2\pi i \theta}$ y $\theta = \frac{\sqrt{5}-1}{2}$.
- Una dendrita (el punto crítico es preperiódico) con $c = i$.

Además, para el algoritmo de escape y Henriksen exploraremos y compararemos también los siguientes tres casos:

- Un anillo de Herman.
- El conjunto de Mandelbrot.
- Ampliación del conjunto de Mandelbrot.

Los algoritmos están ejecutados en el ordenador Huawei MateBook 14 del año 2020. Dispone de un procesador Intel(R) Core(TM) i7-10510U y 16GB de memoria RAM.

5.1. Algoritmo de escape

Este algoritmo comprueba si se converge a un punto atractor que se conoce de antemano. Por ejemplo sea $p(z)$ es un polinomio, por la Observación 1 sabemos que infinito es súper atractor. Para aplicar el algoritmo de escape en un polinomio miramos si convergemos a infinito. En particular, si $Q_c(z) = z^2 + c$ es la familia cuadrática y $R = \max\{2, c\}$, por la Proposición 4.2.1 podemos verificar si hemos convergido a infinito o no. Pero esto es cierto para esta parametrización. ¿Y si tuviéramos $f(z) = \lambda z + z^2$? La respuesta es no. Por tanto, en general tomamos una constante de escape grande que garantice la convergencia.

Funciones cuadráticas

Sea $Q_c(z) = z^2 + c$, en primer lugar es necesario fijar un número de iteraciones máximas, por ejemplo $N = 50$. Después, fijamos el valor de $c \in \mathbb{C}$ para el cual queremos dibujar el plano dinámico. Ahora, consideramos el cuadrado D centrado en $z = 0$ y de lado $R = \max\{2, c\}$, es decir, $D = \{x + iy \mid -R \leq x, y \leq R\}$. Dentro de este cuadrado consideramos una retícula de 1000×1000 puntos.

Para cada punto de la retícula, consideramos su sucesión de iterados por Q_c . Si para algún $n \leq N$ se obtiene que $|Q_c^n(z)| > R$, sabemos que su órbita escapa a infinito y la pintaremos de la tonalidad del color establecida dependiendo del número de iterados necesarios para escapar. En caso contrario, si una vez alcanzados los N iterados no se ha dado la condición anterior, significa que en ese punto la órbita está acotada en N iterados y la pintaremos de color negro. Es decir, los colores se asignan de color negro si el punto no escapa a infinito. Mientras que si el punto escapa a infinito usamos un escalado de colores en función del número de iterados que ha necesitado (de rojo oscuro si tarda pocos iterados a blanco si tarda muchos iterados). Por tanto, si los colores son más claros significa que ha tardado más en converger, mientras que cuanto más oscuro sea el color menos número de iterados ha necesitado. Cuando el color es negro significa que ha llegado hasta el número de iterados máximo. Finalmente, el conjunto de Julia se intuye en los puntos más blancos.

En este algoritmo podría suceder que un punto tuviera una órbita cuyos primeros N iterados estuvieran acotados pero que finalmente su órbita escapara a infinito. Por tanto, un mayor número de puntos en la retícula y un mayor número de iterados permite una mayor precisión aunque también más tiempo de cálculo. Además el escalado de colores es proporcional al número de iterados, entonces si por ejemplo para un valor de c el punto escapa a infinito en la iteración 20, no tendrá el mismo color si $N = 50$ que si $N = 200$.

Por tanto el algoritmo de escape sería de la forma:

```
def algoritmo_escape(z):
```

```

for i in range(max_iter):
    z = f(z)
    if abs(z) >= 2:
        return i
return max_iter

```

Representar el conjunto de Mandelbrot con este algoritmo se hace forma muy similar a la explicada anteriormente. En general, si queremos dibujar el espacio de parámetros se consigue iterando el punto crítico ya que toda componente de Fatou está relacionada con un punto crítico como hemos visto en el Lema 4.3.1. En particular, para el conjunto de Mandelbrot el 0 es el único punto crítico, por tanto iteramos el punto 0. La diferencia con lo mencionado anteriormente es que en lugar de fijar el valor de $c \in \mathbb{C}$ y variar el valor de z , se fija el valor de z a 0, y c va tomando todos los valores de la retícula, para los cuales se calcula la sucesión de iterados al igual que en el plano dinámico. Esto lo hacemos porque queremos saber si hay componentes de Fatou otras que la cuenca de atracción de infinito. Entonces si escapa a infinito lo pintamos con el color correspondiente y si no escapa a infinito lo pintamos en negro porque puede haber otras componentes de Fatou.

El código de este algoritmo está implementado tanto para representar conjuntos de Julia, como para conjuntos de Mandelbrot y podemos verlos en el Anexo A.1.

Funciones racionales

Si aplicamos el algoritmo de escape que hemos descrito para funciones cuadráticas en funciones racionales los resultados que obtenemos no son los correctos. Esto se debe a que nosotros en la función cuadrática anterior sabemos que el infinito es un punto atractor, pero ¿y si tenemos más de un punto atractor, o si tenemos otros puntos atractores? Realmente, lo que queremos comprobar con el algoritmo de escape es si la función ha “escapado” a los puntos atractores de la función que conocemos de antemano. Por tanto, en caso de que tuviéramos más de un punto atractor, se debería cambiar la condición de parada y tener tantas condiciones como puntos atractores haya.

En el caso de tener una función f racional se tienen que estudiar cuáles son sus puntos atractores. Por ejemplo, en el caso de los anillos de Herman donde $f(z) = \lambda z^2 \frac{az + 1}{z + a}$ siendo $\lambda = e^{2\pi it}$, $t = 0,61517321588\dots$ y $a = 0,25$. Podemos ver que 0 es un punto fijo ya que si sustituimos 0 en $f(z)$ obtenemos

$$f(0) = \lambda \cdot 0^2 \cdot \frac{a \cdot 0 + 1}{0 + a} = 0.$$

Además es súper atractor ya que su multiplicador es $|f'(0)| = 0$. El ∞ también es un punto

fijo súper atractor porque en una función racional infinito es un punto fijo súper atractor si y solo si el grado del numerador supera como mínimo en 2 al grado del denominador. Por tanto sus puntos atractores son el 0 y el ∞ . Entonces, es necesario modificar el algoritmo de escape anterior y comprobar si la órbita ha escapado a alguno de los dos puntos atractores. Para que el algoritmo funcione correctamente sería necesario hacer la siguiente modificación del código del Anexo A.1 de la siguiente manera, siendo $escape \geq 100$ y $\epsilon = 0,001$ (comprobar si ha convergido a 0):

```

def algoritmo_escape(z):
    for i in range(max_iter):
        z = f(z)
        # Comprobamos si ha convergido a infinito
        if abs(z) >= escape:
            return i
        # Comprobamos si ha convergido a 0
        if abs(z) <= epsilon:
            return 0
    return max_iter

```

El motivo por el que hay que modificar el parámetro de $escape$ se debe a que para la función $f(z) = z^2 + c$ tenemos una condición de escape conocida, pero para familias racionales por lo general no se sabe. Por este motivo, se suele jugar con números de escape grandes.

A continuación vamos a ver diferentes casos del algoritmo de escape, variando el número de iterados, N . Todas las imágenes generadas tienen un tamaño de 1000x1000 píxeles.

Caso hiperbólico

En este apartado vamos a ver dos casos hiperbólicos. El primero con el valor de $c = 0,36 + 0,34i$ y el segundo con el valor $c = -0,672 + 0,337i$. Podemos observar los resultados en la Figura 5.1. En ambos casos de dicha figura, se puede apreciar que en las primeras representaciones gráficas hay puntos dibujados en negro que realmente no deberían estar en color negro. Sin embargo, en la última representación se puede apreciar que el dibujo se distingue correctamente. Esta diferencia se debe a que las primeras figuras no tienen un número de iterados lo bastante grande y no se realizan todos los cálculos que en realidad se necesitan. Vemos que cuando $N = 500$ el resultado sí que está completo y se distingue todo correctamente. Por tanto, a medida que se aumenta el número de iterados, la figura se distingue con mayor detalle y precisión.

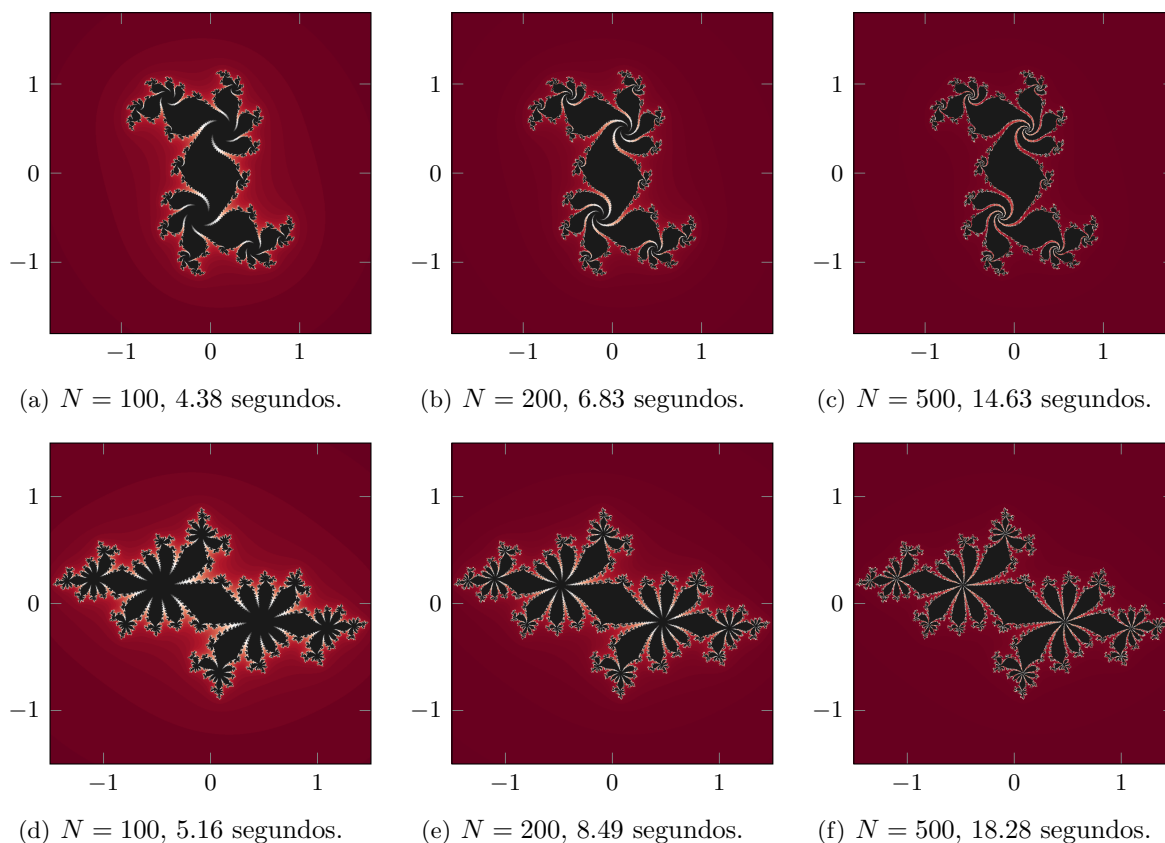


Figura 5.1: Algoritmo de escape para dibujar caso hiperbólico con $c = 0,36 + 0,34i$ y $c = -0,672 + 0,337i$.

Caso cercano a parabólico

En este apartado vamos a ver la influencia del parámetro N para el caso cercano a parabólico $c = 0,285 + 0,01i$. Para el valor de c elegido, tenemos que el punto crítico 0 tiende a infinito. Esto implica que no existe otra componente de Fatou a parte de $A(\infty)$. Al analizar la Figura 5.2, en las primeras iteraciones se observa la aparición de áreas negras. Sin embargo, es importante destacar que estas áreas son incorrectas y no tendrían que aparecer ya que el conjunto de Julia tiene interior vacío y únicamente existe $A(\infty)$. Pero, debido a que nos encontramos en un caso cercano a parabólico, el comportamiento se vuelve más delicado.

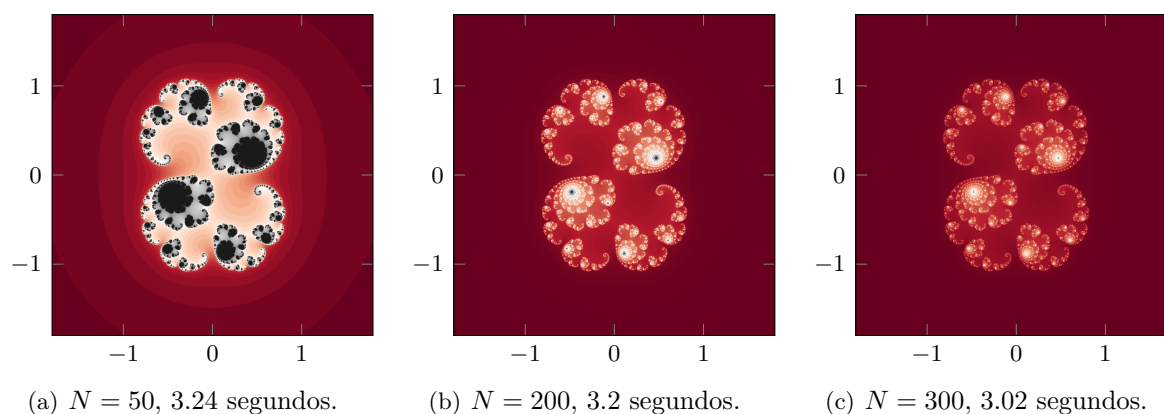


Figura 5.2: Algoritmo de escape para dibujar un caso cercano a parabólico con $c = 0,285 + 0,01i$.

Caso parabólico de periodo 3

Para el caso parabólico de periodo 3 consideramos la siguiente función $f(z) = e^{2\pi i \frac{1}{3}} z + z^2$. Esta figura se conoce por el nombre de *Fat Douady rabbit* (ver [5]). Además, es importante destacar que el punto parabólico se encuentra en la intersección de las tres puntas superiores. Al examinar los resultados obtenidos en la Figura 5.3 se puede apreciar que aunque aumentemos el número de iterados no hay diferencias significativas en la representación de las figuras. Incluso con una gran diferencia en el número de iteraciones entre $N = 100$ y $N = 500$ los cambios visuales que se detectan son mínimos, pero las líneas deberían de acercarse al punto parabólico. Es decir, por más que aumentemos N , las puntas no llegan a tocarse.

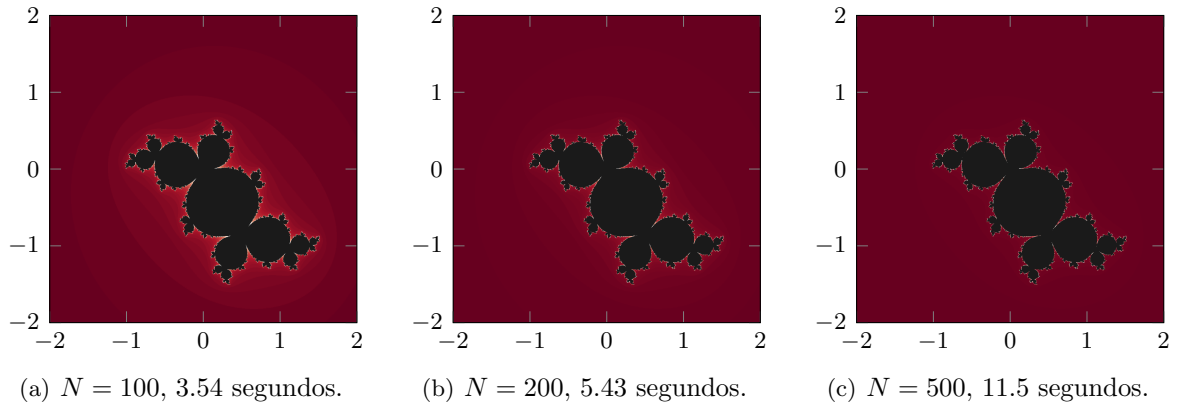


Figura 5.3: Algoritmo de escape para dibujar caso parabólico de periodo 3.

Disco de Siegel

Para estudiar los resultados de dibujar un disco de Siegel tenemos la función $f(z) = \lambda z + \lambda z^2$, siendo $\lambda = e^{2\pi i \theta}$ y $\theta = \frac{\sqrt{5}-1}{2}$. Podemos observar en la Figura 5.4 que en el primer dibujo los bulbos son bastante gordos, mientras que a medida que vamos aumentando el número de iterados los bulbos están un poco más definidos y las puntas se acercan más. Aunque los cambios se pueden apreciar, es cierto que el dibujo mejora muy ligeramente y se necesita un número de N bastante elevado para poder apreciar los resultados.

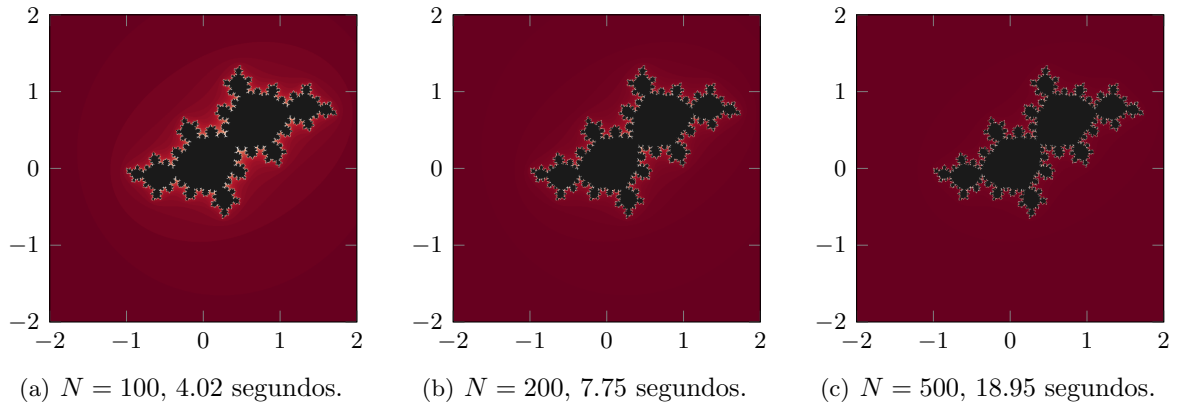


Figura 5.4: Algoritmo de escape para dibujar disco de Siegel.

Caso de una dendrita

Para ver el caso de una dendrita estableceremos el valor de $c = i$. Cuando tenemos una dendrita el punto crítico no escapa a infinito y es preperiódico. Esto podemos verlo fácilmente ya que si $f(z) = z^2 + i$, iterando el punto crítico 0 nos quedaría que

$$\begin{aligned}f(0) &= 0^2 + i = i \\f(i) &= i^2 + i = i - 1 \\f(i - 1) &= (i - 1)^2 + i = -i \\f(-i) &= (-i)^2 + i = i - 1\end{aligned}$$

Por tanto tenemos que $i - 1$ es un ciclo de periodo 2 y el punto crítico es preperiódico. Además el conjunto de Julia tiene interior vacío. En este caso, no se puede dar lugar a ninguna componente de Fatou a parte de $A(\infty)$.

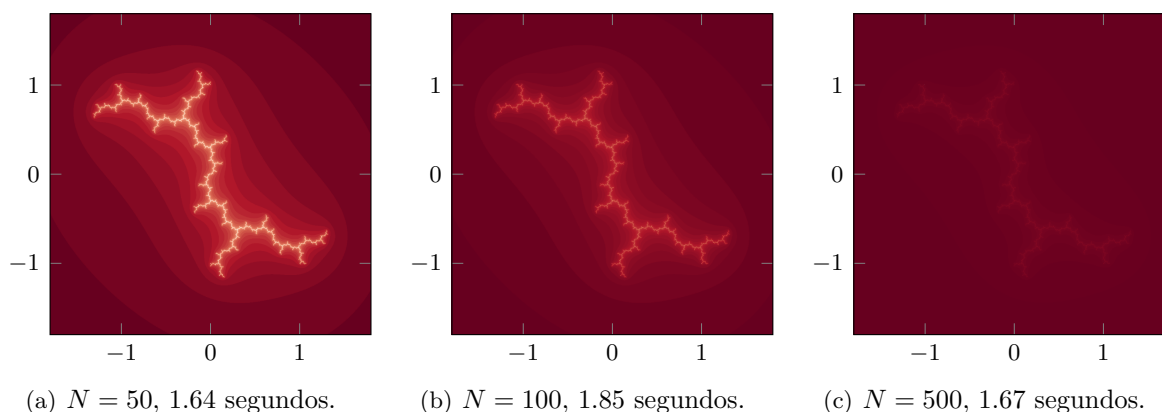


Figura 5.5: Algoritmo de escape para dibujar una dendrita.

En la Figura 5.5 podemos observar que cuando $N = 50$ el dibujo de la dendrita se ve muy bien y con mucha claridad. A medida que se aumenta el número de iterados, el resultado es peor hasta el punto que cuando $N = 500$ cuesta identificar la dendrita. Esto se debe a que como el escalado de colores es proporcional al número de iterados y para este caso tenemos que el conjunto de Julia tiene interior vacío, no se necesita dibujar ningún punto de color negro ya que todos los puntos escapan. Sin embargo, podemos “intuir” el conjunto de Julia como la acumulación de los puntos que tardan más en escapar a infinito. Pero si aumentamos el número de iterados esos puntos se ven menos.

Por tanto, si el dibujo que queremos realizar con el algoritmo de escape tiene interior vacío, sería conveniente utilizar pocos iterados.

Anillo de Herman

Para poder obtener un anillo de Herman utilizaremos la función $f(z) = \lambda z^2 \frac{az + 1}{z + a}$ siendo $\lambda = e^{2\pi it}$, $t = 0,61517321588\dots$ y $a = 0,25$. En primer lugar veremos la figura entera, mientras que después haremos una ligera ampliación donde se encuentra realmente el anillo de Herman.

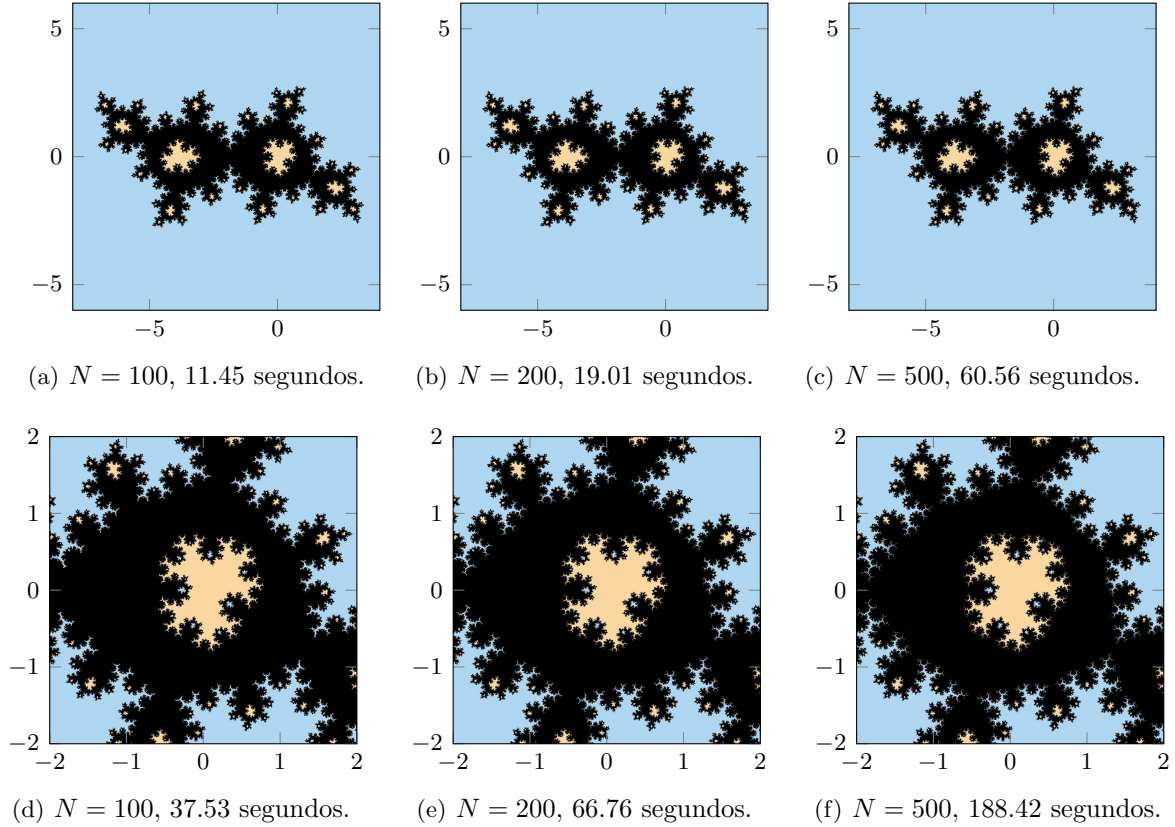


Figura 5.6: Algoritmo de escape modificado para dibujar anillo de Herman.

Las figuras se han obtenido haciendo la modificación mencionada anteriormente sobre el algoritmo de escape para funciones cuadráticas. El resultado de la ejecución podemos verlo en la Figura 5.6. Además, vemos que el resultado es diferente a los resultados obtenidos hasta el momento con el algoritmo de escape, ya que no hay un escalado de colores. Esto se debe a las dificultades que hemos tenido con la escala de colores al haber dos puntos atractores. La solución adoptada ha sido eliminar la escala de colores y pintar los colores que escapan a infinito de azul, los que convergen al 0 de naranja y el resto negro. De esta manera, podemos distinguir el comportamiento de cada punto con mayor facilidad.

En cuanto a los resultados de la ejecución del algoritmo, podemos ver que para $N = 100$

se obtiene un dibujo aceptable pero como podemos comprobar si aumentamos el número de iterados, los bulbos obtienen más detalle y están más definidos, aunque también aumenta el tiempo de ejecución.

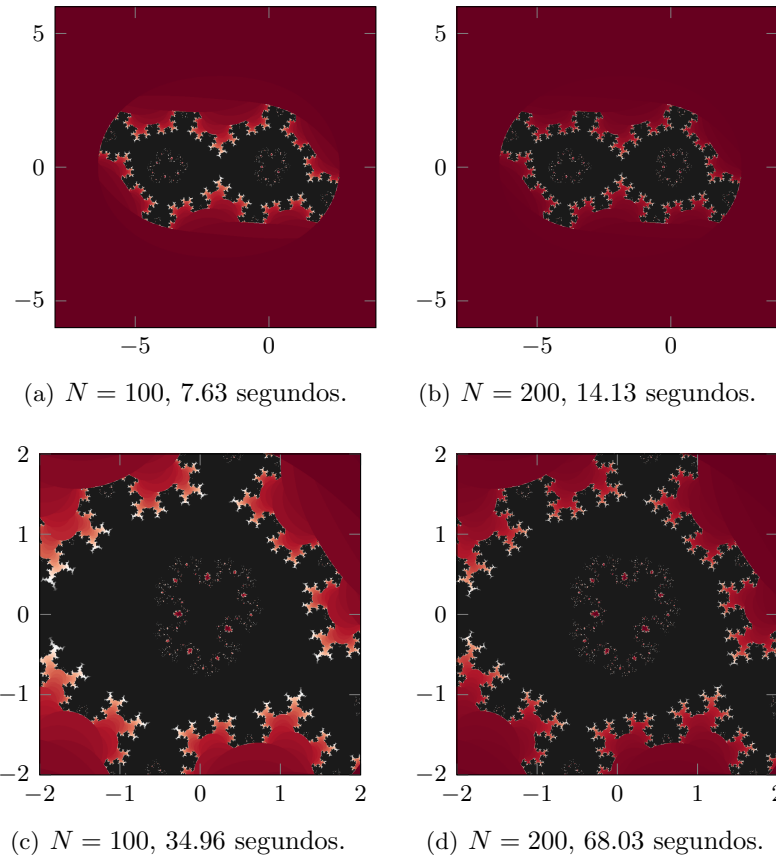


Figura 5.7: Algoritmo de escape sin modificar para dibujar anillo de Herman.

Si no hubiéramos hecho la modificación del algoritmo, el resultado obtenido hubiera sido el de la Figura 5.7. Se puede observar que el centro de la figura está de color negro porque al no considerarse el 0 como punto atractor, no se tiene en cuenta y simplemente se considera que no ha convergido a infinito y por tanto está acotado, como en el caso cuadrático. Además, también se aprecia que hay cortes por la zona de los extremos y que parece que la figura no esté terminada. Esto se debe a que como f es una función racional, la condición de escape es un número mayor que 2. En general, este valor se desconoce, por tanto se suelen probar números grandes como por ejemplo $escape \geq 100$.

Conjunto de Mandelbrot

Hasta ahora hemos visto representaciones del conjunto de Julia o de funciones racionales. Así que en este apartado vamos a ver una representación del conjunto de Mandelbrot y una ampliación de una zona conocida como los *elefantes* (ver [11]). Estos elefantes se encuentran en la frontera del cardioide principal del conjunto de Mandelbrot y reciben este nombre debido a su forma similar con un elefante. Son ampliaciones que se caracterizan por tener autosimilitud, es decir, estructuras similares al conjunto de Mandelbrot pero en una escala más pequeña.

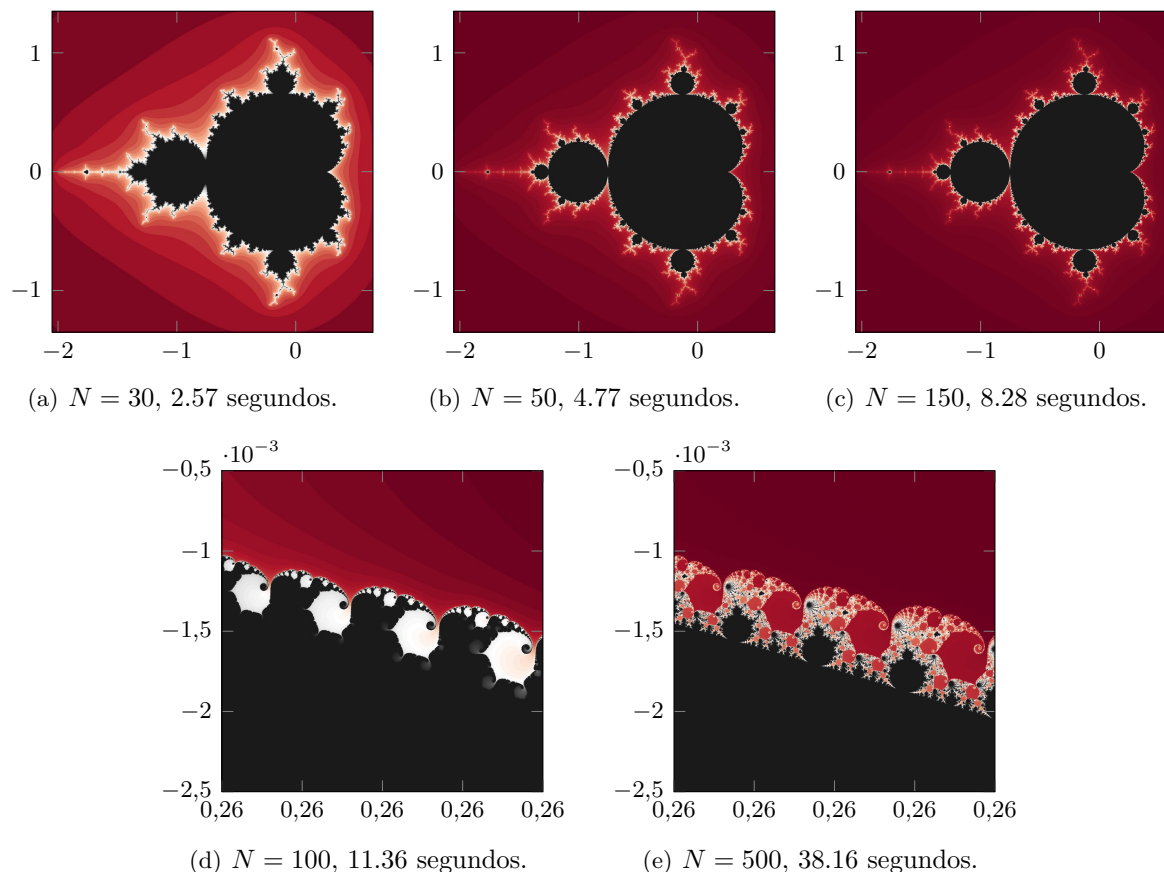


Figura 5.8: Algoritmo de escape para dibujar el conjunto de Mandelbrot.

En la Figura 5.8 podemos ver que los primeros tres dibujos son una representación del conjunto de Mandelbrot. Para $N = 30$ vemos que el dibujo es muy malo porque los bulbos aparecen muy separados. Si aumentamos a $N = 50$, los bulbos se distinguen mejor pero la antena de la punta que se ve a la izquierda ya se empieza a ver peor. Mientras que si $N = 150$, los bulbos se ven muy bien pero la parte blanca de la antena no se distingue correctamente y parece que el conjunto de Mandelbrot no sea un conjunto conexo, pero hemos visto en la

propiedad 3 que sí que lo es.

Por otro lado, en los últimos dos dibujos de la Figura 5.8 podemos ver la ampliación de los elefantes. Se puede apreciar que cuando se utiliza $N = 100$, el dibujo es deficiente a pesar de ser un número de iteraciones considerable. Sin embargo, para poder apreciar los detalles del dibujo es necesario aumentar significativamente el número de iterados, como por ejemplo a $N = 500$.

Ampliación del conjunto de Mandelbrot

En esta sección vamos a ver con detalle las antenas de un bulbo del cardioide principal del conjunto de Mandelbrot.

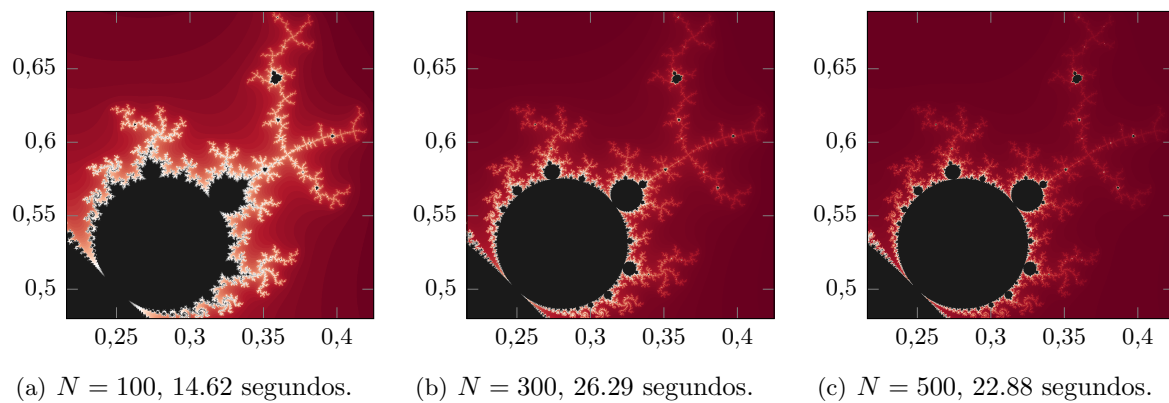


Figura 5.9: Algoritmo de escape para dibujar bulbo del conjunto de Mandelbrot.

En la Figura 5.9, se puede observar que a medida que aumenta el número de iteraciones, el conjunto de Mandelbrot se representa de manera más precisa y detallada. Sin embargo, el espacio de bifurcaciones cada vez es menos visible. Cabe destacar que en este caso el color blanco significa que el punto crítico es cercano a un parámetro que no escapa a infinito, pero al aumentar el número de iterados ya no se distingue correctamente, de forma parecida a lo que pasaba en el caso de la dendrita.

5.2. Algoritmo de iteración inversa

Este algoritmo funciona bien siempre para los casos hiperbólicos, discos de Siegel, casos parabólicos, cerca de parabólicos y dendritas. El algoritmo está basado en la propiedad (d) que dice que si z_0 es un punto de $J(f)$, el conjunto de todas las antiimágenes de z_0 es denso en $J(f)$.

Para implementar el algoritmo, lo primero que hay que hacer es buscar sobre qué punto del conjunto de Julia hay que empezar a hacer los cálculos de iteración inversa. Veamos que uno de los dos puntos fijos es repulsor y que los puntos fijos son las soluciones de la ecuación

$$z^2 + c = z,$$

por tanto,

$$\alpha_c = \frac{1}{2}(1 - \sqrt{1 - 4c}),$$

$$\beta_c = \frac{1}{2}(1 + \sqrt{1 - 4c})$$

son las soluciones de la ecuación.

Ahora se puede comprobar que β_c es repulsor excepto para $c = \frac{1}{4}$. Es decir, β_c tiene multiplicador $\lambda_\beta = 2z$, por tanto sustituyendo obtenemos que $\lambda_\beta = 1 + \sqrt{1 - 4c}$. Entonces está claro que $|1 + \sqrt{1 - 4c}| > 1$ ya que o bien $\sqrt{1 - 4c}$ es un número real positivo, o bien es un número imaginario. Si $c = \frac{1}{4}$, el multiplicador será $\lambda_\beta = 1$ dando lugar a un caso parabólico. Como sabemos que los puntos repulsores pertenecen al conjunto de Julia y los puntos parabólicos también, usaremos β_c como punto base. A partir de aquí tenemos que $z_0 = \beta_c$ y empezaremos a calcular las antiimágenes de la forma $z_n = \pm\sqrt{z_{n-1} - c}$ donde todo punto z_n , para cada n , cumple que $Q_c^n(z_n) = \beta_c$.

Tal y como hemos visto en el algoritmo de escape, los puntos que peor se dibujan son los que están cerca de puntos fijos o periódicos cuyo multiplicador es cercano a 1. Por ello, nos interesa centrarnos más en los puntos cuya derivada es pequeña y descartar los puntos que tengan una derivada grande, ya que en general estos puntos se dibujan mejor y dan menos problemas. Además, puede pasar que las antiimágenes no se distribuyan de forma equitativa por el conjunto de Julia y necesitemos un nivel muy alto de n , teniendo que almacenar en la memoria 2^n puntos y que todavía hayan zonas poco exploradas. Para evitar estos problemas vamos a calcular el módulo de la derivada y vamos a iterar los puntos hasta que este módulo alcance un determinado valor, D , descartando los puntos que lo superen. Habrá algunos puntos que lo alcancen con mayor rapidez y otros que necesiten realizar más iteraciones.

La derivada acumulada que hemos mencionado es muy sencilla de calcular. Se calcula utilizando la regla de la cadena. Comenzamos estableciendo el valor de la derivada acumulada en 1 ($derAcu = 1$). Además, sabemos que la derivada de la función $f(z) = z^2 + c$ es $f'(z) = 2z$. A

medida que ejecutamos el algoritmo, guardamos el valor absoluto de la derivada en punto que estamos evaluando, lo sustituimos en $f'(z)$ y lo multiplicamos por el valor actual de la derivada acumulada. Es decir, actualizamos la derivada acumulada multiplicándola por el valor absoluto de $f'(z)$.

En este algoritmo se pueden variar los parámetros de N que es el número de iteraciones inversas máximas que realiza el algoritmo y D que es el valor máximo de derivada acumulada al que se va a llegar para cada valor de c . Así que a continuación vamos a ver cómo influyen los cambios en cada uno de ellos. Además, el código del algoritmo implementado se encuentra en el Anexo A.2 en caso de querer consultarlo, aunque el núcleo del algoritmo es el siguiente.

```
def algoritmo_it_inversa(z, derAcu, nIter):
    if nIter >= max_iter or derAcu > D:
        return False

    derAcu *= abs(der_punto(z)) #calculamos la derivada acumulada

    i, j = get_pos(z.real, z.imag)
    img.putpixel((i, j), (0, 0, 0)) #pintamos el pixel de negro

    inv = inverse_iteration(z) #calculamos la inversa del punto

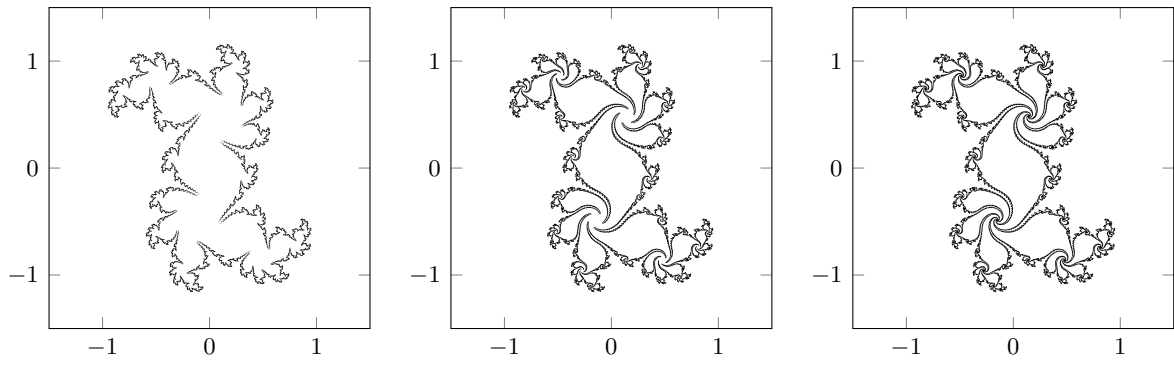
    algoritmo_it_inversa(inv, derAcu, nIter + 1)
    algoritmo_it_inversa(-inv, derAcu, nIter + 1)

    return True
```

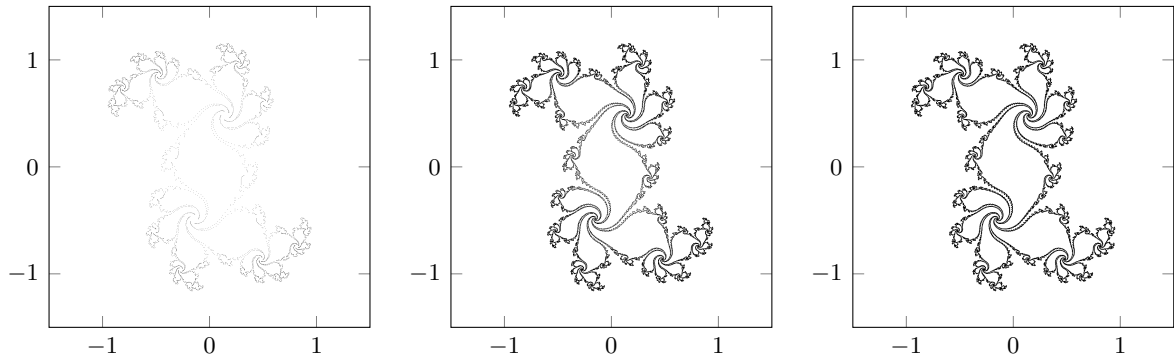
Caso hiperbólico

Veamos como funciona este algoritmo para casos hiperbólicos. En primer lugar vamos a ver los resultados obtenidos para el valor $c = 0,36 + 0,34i$. Como tenemos que estudiar el comportamiento de los parámetros N y D , el procedimiento que hemos seguido ha sido en primer lugar fijar el parámetro D con un valor considerable y aumentar el valor de N para ver cómo influye. Después, hemos fijado el parámetro N con un valor elevado y hemos modificado la D .

En la Figura 5.10 podemos que ver que al fijar el parámetro $D = 50000$, a medida que se aumenta el número de iterados se distinguen más partes del dibujo. Cuando $N = 50$ el interior de la figura está casi todo de color blanco, mientras que cuando $N = 500$, se distinguen las espirales que forman la figura a la perfección. Por tanto, el parámetro N influye directamente en el dibujo y en que se pueda visualizar con sus partes completas.



(a) $N = 50$, $D = 50000$, 8.06 segundos. (b) $N = 150$, $D = 50000$, 15.37 segundos. (c) $N = 500$, $D = 50000$, 16.54 segundos.

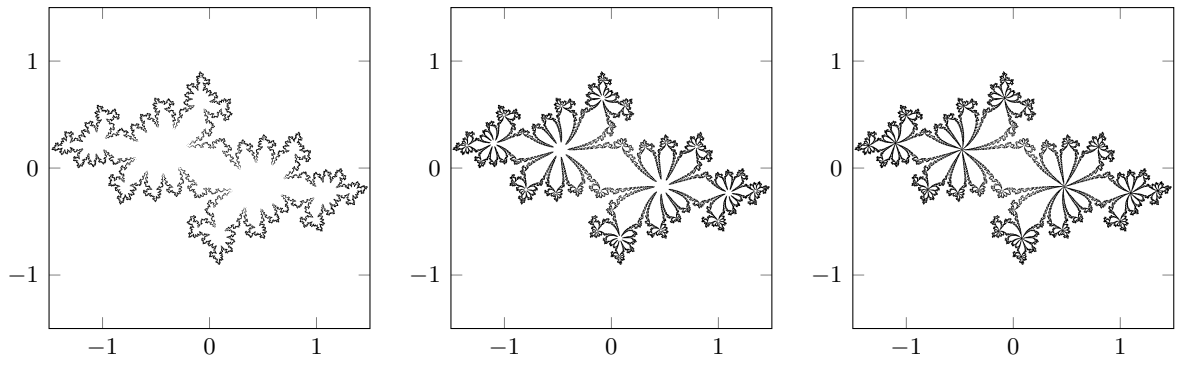


(d) $N = 500$, $D = 1000$, 0.3 segundos. (e) $N = 500$, $D = 10000$, 1.71 segundos. (f) $N = 500$, $D = 30000$, 7.61 segundos.

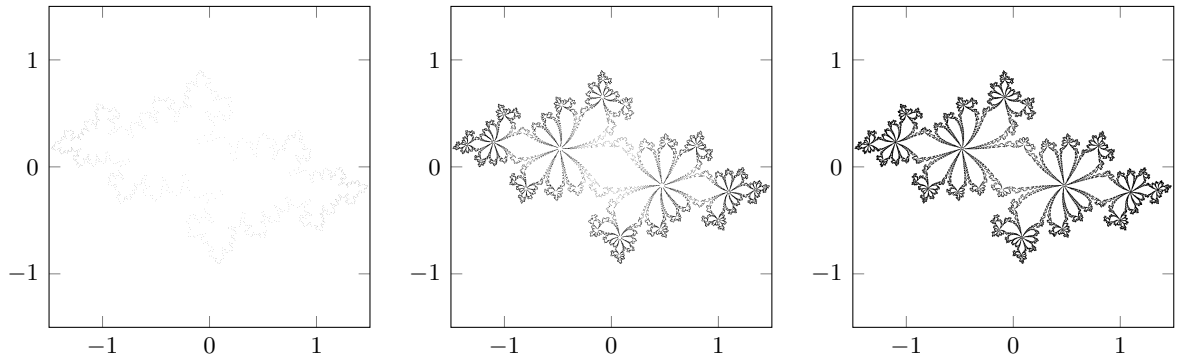
Figura 5.10: Algoritmo de iteración inversa para caso hiperbólico con $c = 0,36 + 0,34i$. En la primera fila se ha fijado el parámetro $D = 50000$ y en la segunda fila el parámetro $N = 500$.

Sin embargo, si fijamos el parámetro $N = 500$ y modificamos el valor D , el resultado obtenido es muy diferente al comentado anteriormente. La forma de la figura y las espirales que contiene se pueden distinguir en todo momento, en lo que influye el parámetro D es en la intensidad. Es decir, si $D = 1000$ se distingue la figura pero el resultado adquiere un color muy claro ya que está formado por pocos puntos. Mientras que a medida que aumenta D , la figura se ve de un color más intenso.

Si modificamos el valor de c anterior a $c = -0,672 + 0,337i$ podemos ver en los resultados de la Figura 5.11 que ocurre lo mismo que para el anterior valor de c .



(a) $N = 50$, $D = 50000$, 0.7 segundos. (b) $N = 150$, $D = 50000$, 0.71 segundos. (c) $N = 500$, $D = 50000$, 5.14 segundos.



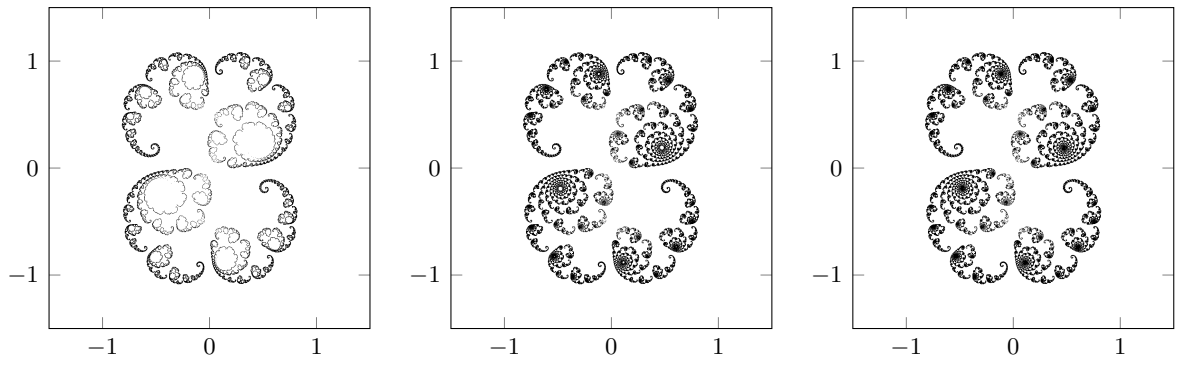
(d) $N = 500$, $D = 1000$, 0.28 segundos. (e) $N = 500$, $D = 10000$, 0.57 segundos. (f) $N = 500$, $D = 30000$, 2.15 segundos.

Figura 5.11: Algoritmo de iteración inversa para caso hiperbólico con $c = -0,672 + 0,337i$. En la primera fila se ha fijado el parámetro $D = 50000$ y en la segunda fila el parámetro $N = 500$.

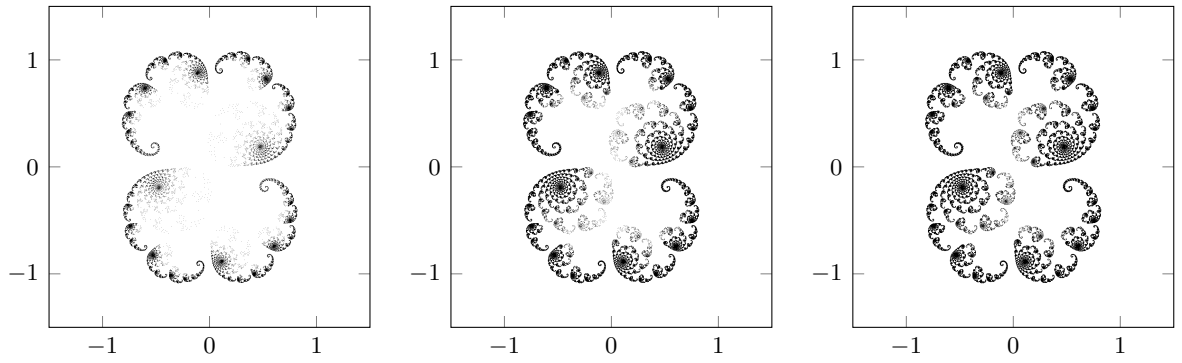
Caso cercano a parabólico

Para el caso cercano a parabólico vamos a establecer $c = 0,285 + 0,01i$ y vamos a ver cómo afectan los parámetros N y D . Vamos a continuar con el mismo procedimiento de primero fijar el valor D y variar el número de iterados y después fijar N y variar el parámetro D .

En la Figura 5.12 podemos observar que cuando fijamos $D = 15000$, si tenemos valores de N bajos la figura no está completa y hay zonas en las que deberían aparecer puntos negros y no aparecen. Esto se debe a que se necesita mayor número de iterados para poder obtener todas las partes de la figura. Por tanto, a medida que aumentamos el parámetro N , el resultado va mejorando hasta obtener una figura en la que la espiral se distingue completamente.



(a) $N = 50$, $D = 15000$, 32.01 segundos. (b) $N = 150$, $D = 15000$, 109.86 segundos. (c) $N = 300$, $D = 15000$, 127.1 segundos.



(d) $N = 400$, $D = 1000$, 1.63 segundos. (e) $N = 400$, $D = 5000$, 22.16 segundos. (f) $N = 400$, $D = 10000$, 74.57 segundos.

Figura 5.12: Algoritmo de iteración inversa para el valor $c = 0,285 + 0,01i$. En la primera fila se ha fijado el parámetro $D = 15000$ y en la segunda fila el parámetro $N = 400$.

Sin embargo, si fijamos el parámetro N y variamos el parámetros D , vemos que la espiral se ve completa desde el primer caso, pero hay zonas que aparecen muy claras y no se distinguen bien. A medida que se aumenta el valor D , aparecen más puntos y aumenta la intensidad en determinadas zonas de la figura, hasta obtener un resultado satisfactorio.

Caso parabólico de periodo 3

Para estudiar el caso parabólico de periodo 3 definimos la función $f(z) = e^{2\pi i \frac{1}{3}} z + z^2$. En este caso nuestro punto inicial es $z_0 = 1 - \lambda$, siendo $\lambda = e^{2\pi i \frac{1}{3}}$. Los resultados obtenidos podemos verlos en la Figura 5.13. En primer lugar, al fijar $D = 100000$ y modificar el valor de N podemos observar que cada vez las puntas se acercan más al centro y la distancia entre ellas es menor.

Aunque, cabe mencionar que se necesita aumentar mucho el valor de N para que se puedan apreciar los cambios, ya que entre las figuras con $N = 50$ y $N = 300$ se ve claramente la mejoría. Pero, entre $N = 300$ y $N = 1000$ el dibujo mejora a un ritmo mucho más lento.

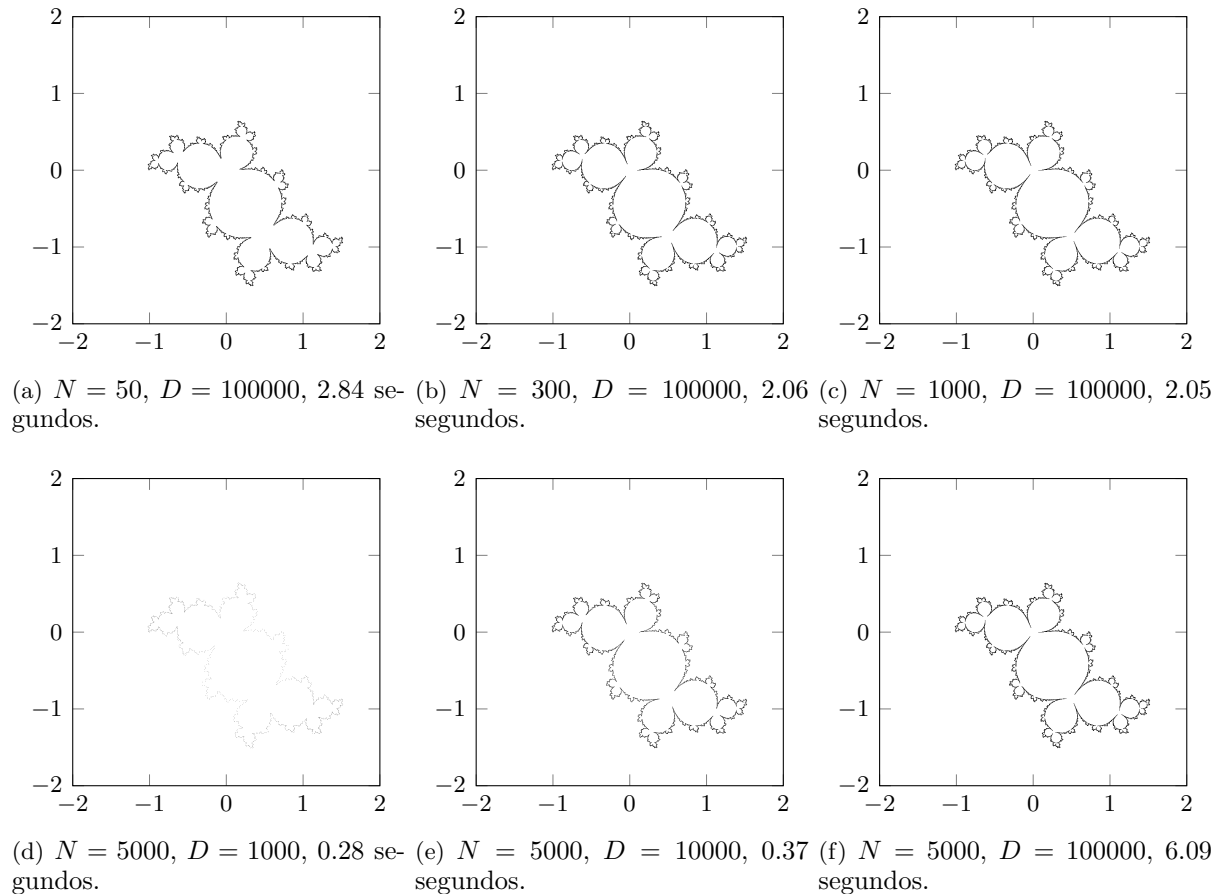


Figura 5.13: Algoritmo de iteración inversa para caso parabólico de periodo 3. En la primera fila se ha fijado el parámetro $D = 100000$ y en la segunda fila el parámetro $N = 5000$.

Por otro lado, cuando fijamos $N = 5000$ y modificamos el valor D , entre las figuras con $D = 1000$ y $D = 10000$ hay una mejoría muy drástica. Pero, entre $D = 10000$ y $D = 30000$ los cambios que se observan son mínimos. Por tanto, la figura con el aumento de los dos parámetros mejora muy lentamente. Además, cabe destacar que los tiempos de ejecución son bastante inmediatos.

Disco de Siegel

Para dibujar un disco de Siegel necesitamos definir la función $f(z) = \lambda z + z^2$, siendo $\lambda = e^{2\pi i\theta}$ y $\theta = \frac{\sqrt{5}-1}{2}$, como ya hemos visto anteriormente. Al igual que antes hemos tomado un punto repulsor inicial, en este caso es fácil de ver que el punto fijo $z_0 = 1 - \lambda$ es repulsor. En la Figura 5.14 podemos ver los resultados obtenidos. En primer lugar vemos que si fijamos $D = 100000$, a medida que se aumenta el parámetro N los bulbos se ven más definidos y el dibujo mejora considerablemente. Pero, si fijamos el parámetro $N = 500$, ocurre lo mismo. Es decir, a medida que aumenta el valor D , se puede apreciar que los bulbos cada vez están más cerrados y las puntas están más cerca de tocarse. Por tanto, podemos decir que en este caso los dos parámetros tienen una influencia bastante parecida.

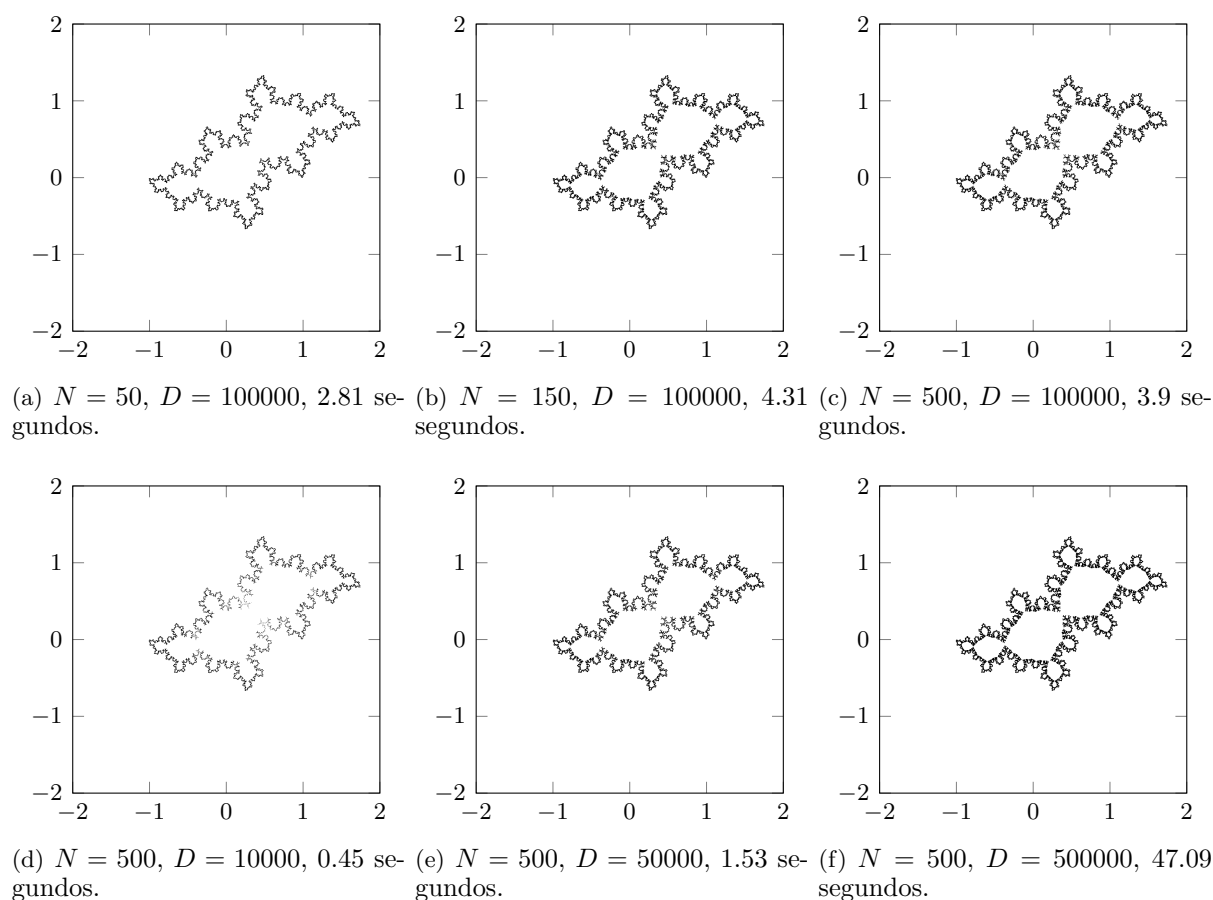
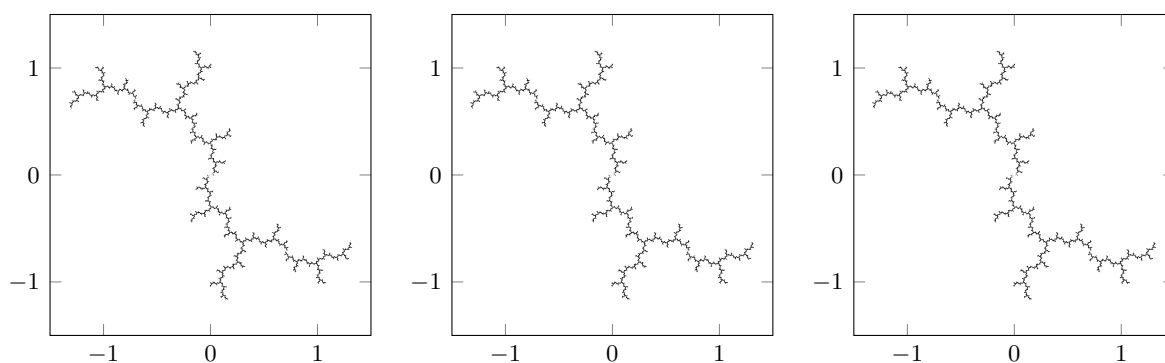


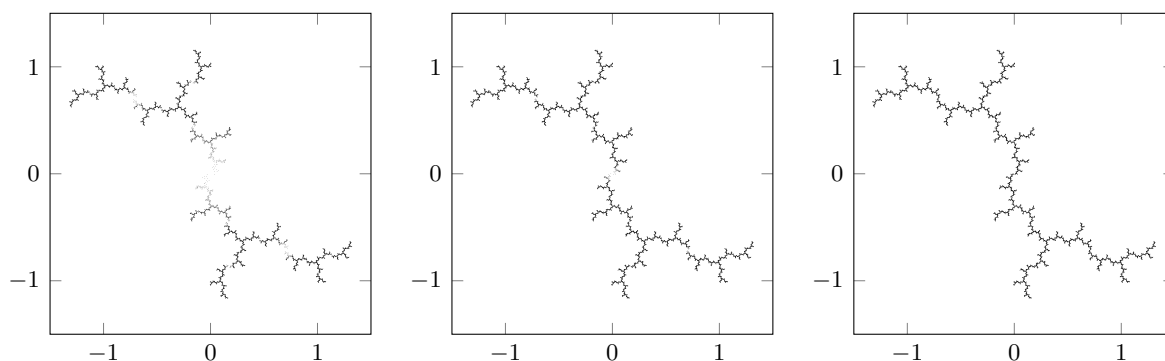
Figura 5.14: Algoritmo de iteración inversa para dibujar un disco de Siegel. En la primera fila se ha fijado el parámetro $D = 100000$ y en la segunda fila el parámetro $N = 500$.

Caso de una dendrita

Para estudiar el caso de una dendrita establecemos el valor $c = i$. En este caso, como podemos ver en la Figura 5.15, si fijamos el valor D y aumentamos el número de iterados, los cambios que se pueden apreciar son mínimos. Los resultados son prácticamente iguales y los cambios no se aprecian a simple vista. Sin embargo, si fijamos el parámetro $N = 500$, y variamos el parámetro D , sí que se aprecian cambios. Cuando $D = 10000$, la figura tiene zonas más claras por la zona central y superior. Pero a medida que se aumenta el valor D , estas zonas claras cada vez son menores hasta que con $D = 500000$ la figura está completa y toda la figura está de color negro.



(a) $N = 50$, $D = 100000$, 2.72 segundos. (b) $N = 150$, $D = 100000$, 3.11 segundos. (c) $N = 500$, $D = 100000$, 2.71 segundos.



(d) $N = 500$, $D = 10000$, 0.41 segundos. (e) $N = 500$, $D = 50000$, 1.24 segundos. (f) $N = 500$, $D = 500000$, 19.67 segundos.

Figura 5.15: Algoritmo de iteración inversa para $c = i$. En la primera fila se ha fijado el parámetro $D = 100000$ y en la segunda fila el parámetro $N = 500$.

5.3. Algoritmo de Henriksen

Este algoritmo está basado en la propiedad (e) del apartado 3.1.3 que dice que los puntos repulsores pertenecen al conjunto de Julia y forman un conjunto denso en $J(f)$. La idea general del algoritmo es que en vez de iterar un punto, se van a iterar todos los puntos a la vez de cada píxel. Es decir, se va a iterar una bola de cierto radio. Asumimos que el tamaño de la bola se multiplica por la derivada y esto linealmente, cerca de cada punto es aproximadamente cierto.

Para dibujar el conjunto de Julia, dado un píxel, queremos iterarlo para saber si contiene un punto periódico, es decir si $f^n(z) = z$ para algún $n \in \mathbb{N}$, siendo $n \geq 1$. Pero, debemos tolerar un error en este cálculo, ya que estamos considerando un píxel que tiene un determinado tamaño y no únicamente un punto, por tanto al hacer los cálculos si el punto está dentro del píxel consideraremos que sí se cumple la condición. Es decir, queremos comprobar si $f^n(z + \epsilon) = z + \epsilon$, donde ϵ es comparable al tamaño del píxel y lo hemos definido previamente.

Entonces, aproximando por su serie de Taylor de orden 1, queremos comprobar que

$$f^n(z + \epsilon) \approx f^n(z) + \epsilon(f^n)'(z) = z + \epsilon \iff \frac{z - f^n(z)}{(f^n)'(z) - 1} = \epsilon.$$

Por lo tanto, nuestra condición de parada consistirá en comprobar si $\left| \frac{z - f^n(z)}{(f^n)'(z) - 1} \right| < \epsilon$.

Además, para calcular $(f^n)'(z)$ utilizaremos la regla de la cadena ya que sabemos que $f^n(z) = f(f^{n-1}(z))$. Por tanto quedaría, $(f^n)'(z) = f'(f^{n-1}(z)) \cdot (f^{n-1})'(z)$. Para implementarlo en el código del algoritmo, sería suficiente con inicializar una variable a 1 y en cada iteración ir multiplicándola por la derivada acumulada.

Pero si lo que queremos dibujar es el conjunto de Mandelbrot, lo que buscamos son los centros. Es decir, queremos encontrar los puntos en los que $0 = f^n(c)$ ya que si se cumple la Conjetura 4.3 de conectividad local del conjunto de Mandelbrot, estos puntos se acumulan en la frontera del conjunto de Mandelbrot. Por tanto, de la misma forma que lo hemos demostrado antes, aproximando por su serie de Taylor de orden 1, comprobamos que

$$f^n(c + \epsilon) = f^n(c) + \epsilon(f^n)'(c) = 0 + \epsilon \iff \frac{f^n(c)}{1 - (f^n)'(c)} = \epsilon.$$

Si inicializamos la derivada a 1 ($der = 1$), y $f(c) = c^2 + c$, derivando quedaría que $f'(c) = 2c \cdot der + 1$. Por tanto, generalizando por la regla de la cadena tendríamos que

$$(f^n)'(c) = 2 \cdot f^{n-1}(c) \cdot (f^{n-1})'(c) + 1.$$

Por tanto, nuestra condición de parada consistirá en comprobar si $\left| \frac{f^n(c)}{1 - (f^n)'(c)} \right| < \epsilon$. Además, en ambos casos, hay una condición de parada de escape a infinito.

Ahora vamos a ver diferentes casos con este algoritmo. Para el estudio del comportamiento de los parámetros N y ϵ , seguiremos el mismo procedimiento que en la sección anterior. En primer lugar, fijaremos el parámetro ϵ con un valor y aumentaremos el valor de N . A continuación, fijaremos el parámetro N con un valor elevado y modificaremos el valor de ϵ . El código completo para generar las figuras se encuentra en A.3.

Casos hiperbólicos

En este apartado vamos a ver como se comporta este algoritmo para dibujar casos hiperbólicos. En primer lugar vamos a ver los resultados obtenidos con $c = 0,36 + 0,34i$ en la Figura 5.16.

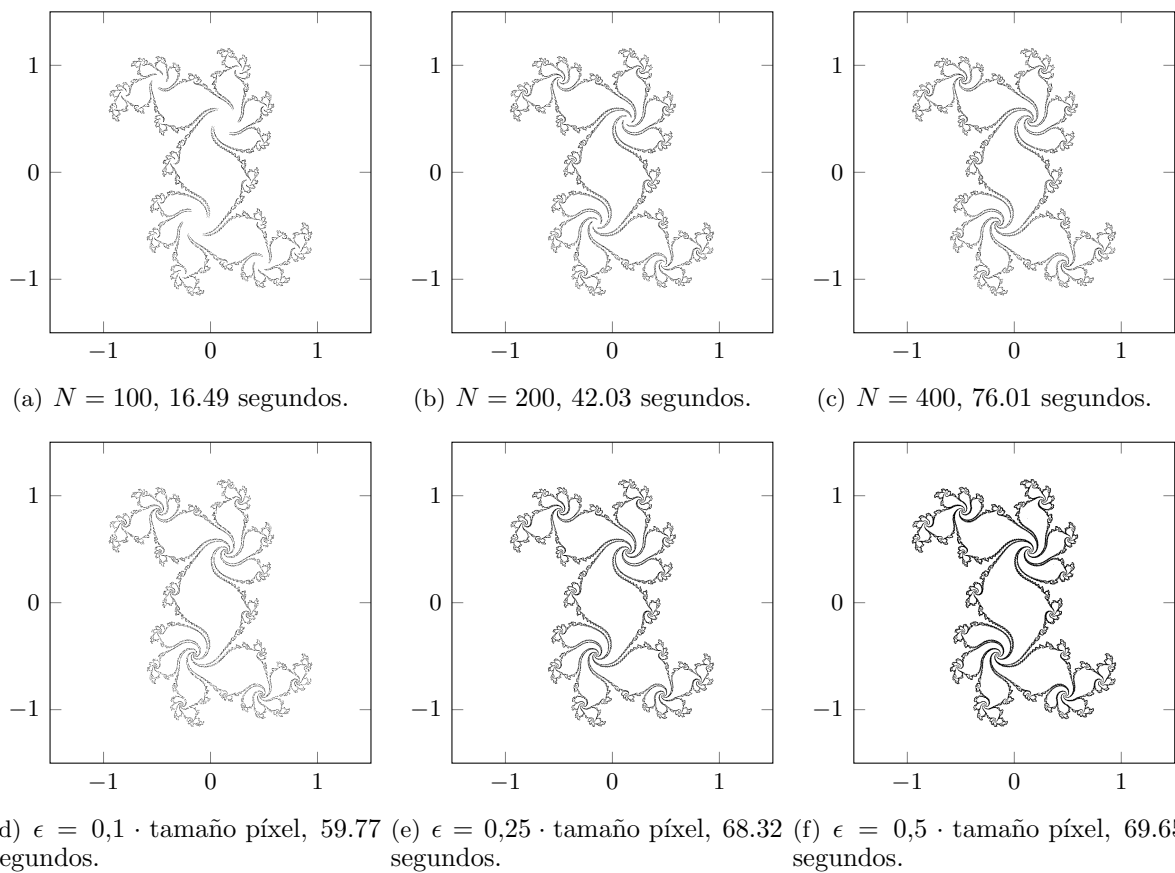


Figura 5.16: Algoritmo de Henriksen para caso hiperbólico con $c = 0,36 + 0,34i$. En la primera fila se ha fijado el parámetro $\epsilon = 0,15 \cdot \text{tamaño píxel}$ y en la segunda fila el parámetro $N = 400$.

Cuando fijamos el parámetro ϵ y variamos el valor de N , podemos observar que cuando

$N = 50$ hay zonas por el interior de la figura que están de color blanco pero que no deberían estarlo. Esto se debe a que el número de iteraciones es demasiado bajo para poder dibujar la figura al completo. Por este motivo, a medida que se aumenta el valor de N , vemos que aparecen más zonas dibujadas hasta que los bulbos se terminan cerrando.

Sin embargo, cuando fijamos el parámetro N y modificamos el valor de ϵ , se puede observar que aumenta el grosor de la figura. Esto se debe a que como el valor de ϵ es más grande, se mira en un entorno más grande si hay puntos periódicos. Como los entornos son más grandes, hay más puntos que cumplen que $f^n(z + \epsilon) = z + \epsilon$, por tanto hay mayor cantidad de puntos dibujados en color negro.

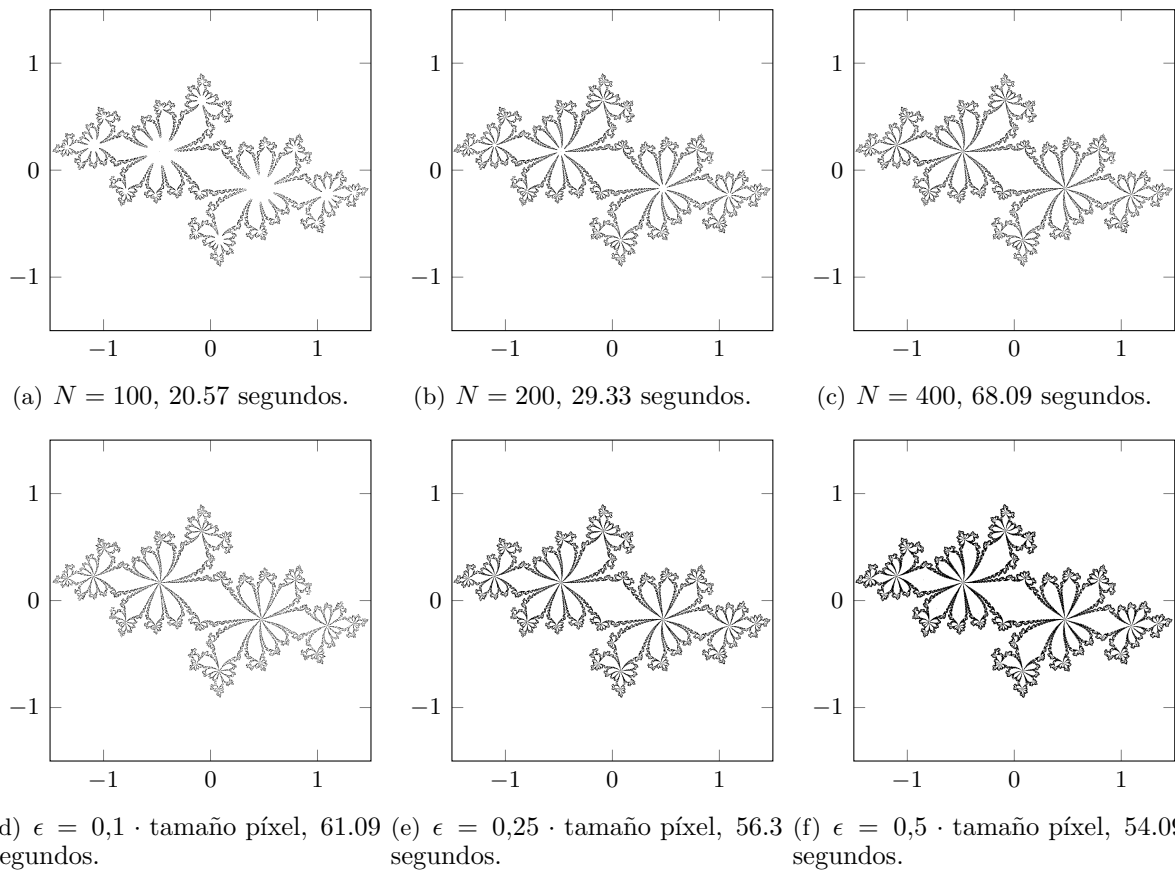


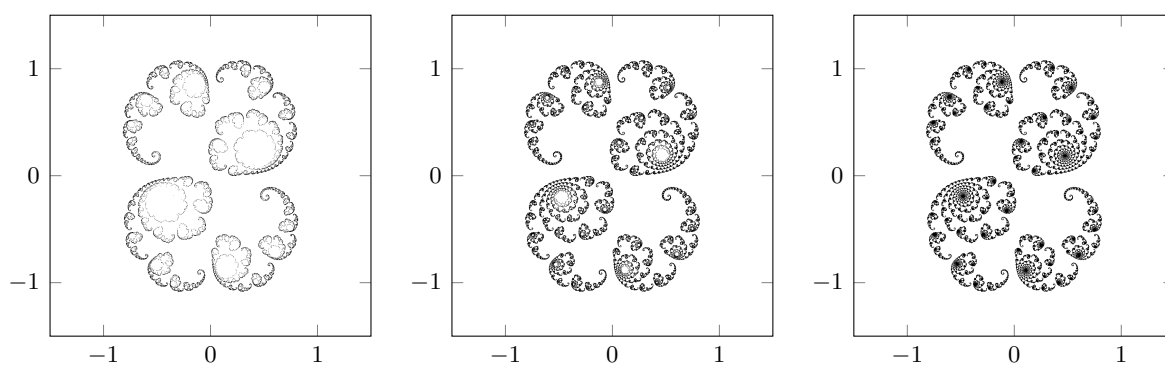
Figura 5.17: Algoritmo de Henriksen para caso hiperbólico con $c = 0,36 + 0,34i$. En la primera fila se ha fijado el parámetro $\epsilon = 0,15 \cdot \text{tamaño píxel}$ y en la segunda fila el parámetro $N = 400$.

Si miramos el resultado para el valor $c = -0,672 + 0,337i$ en la Figura 5.17, podemos ver que ocurre lo mismo que para el anterior valor de c .

Caso cercano a parabólico

Para estudiar el caso cercano a parabólico establecemos el valor $c = 0,285 + 0,01i$. En la Figura 5.18 podemos ver que, al igual que en el caso anterior, si fijamos ϵ y modificamos el número de iterados, aparecen zonas en blanco que no deberían. Pero, si aumentamos N , cada vez hay más puntos en negro en la figura que hacen que finalmente esté completa.

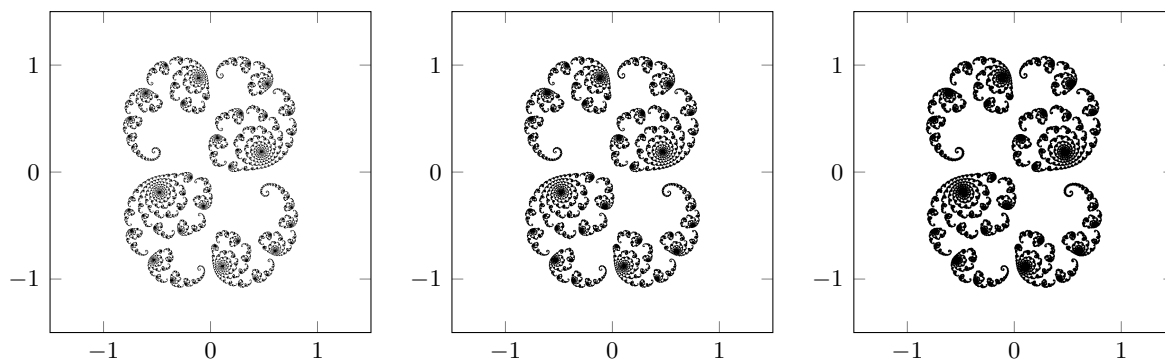
Por otro lado, si fijamos N y modificamos el valor ϵ , podemos observar que hay que ir con cuidado. Si el valor de ϵ es demasiado grande, aparece la frontera de la figura con un grosor bastante grande. Por tanto, en las zonas que aparecen espirales, al estar las líneas muy juntas se pierde gran cantidad de detalles. Por este motivo es recomendable escoger un valor ϵ adecuado.



(a) $N = 50$, 9 segundos.

(b) $N = 100$, 10.9 segundos.

(c) $N = 300$, 10.67 segundos.



(d) $\epsilon = 0,1 \cdot \text{tamaño píxel}$, 13.03 segundos.

(e) $\epsilon = 0,25 \cdot \text{tamaño píxel}$, 12.1 segundos.

(f) $\epsilon = 0,5 \cdot \text{tamaño píxel}$, 11.66 segundos.

Figura 5.18: Algoritmo de Henriksen para dibujar el conjunto de Julia para $c = 0,285 + 0,01i$ que es un caso cercano a parabólico. En la primera fila se ha fijado el parámetro $\epsilon = 0,15 \cdot \text{tamaño píxel}$ y en la segunda fila el parámetro $N = 300$.

Caso parabólico de periodo 3

Para estudiar el caso parabólico definimos la función $f(z) = e^{2\pi i \frac{1}{3}} z + z^2$. En la Figura 5.19 podemos ver que si fijamos el parámetro ϵ , al aumentar el número de iterados cada vez se distinguen más detalles del dibujo y las puntas están más cerca de tocarse.

Mientras que si fijamos el valor $N = 700$ y modificamos el valor de ϵ , podemos ver que cuando $\epsilon = 0,1 \cdot \text{tamaño píxel}$ la figura se ve muy clara y cuesta distinguir todas las partes del dibujo con facilidad. Pero, a medida que aumentamos el tamaño de ϵ hasta tomar $\epsilon = 0,5 \cdot \text{tamaño píxel}$, cada vez el grosor del dibujo es más grande y se distingue mejor el dibujo. Aunque, cabe destacar que no se debe tomar un tamaño de ϵ demasiado grande porque sino se pueden perder detalles de la figura.

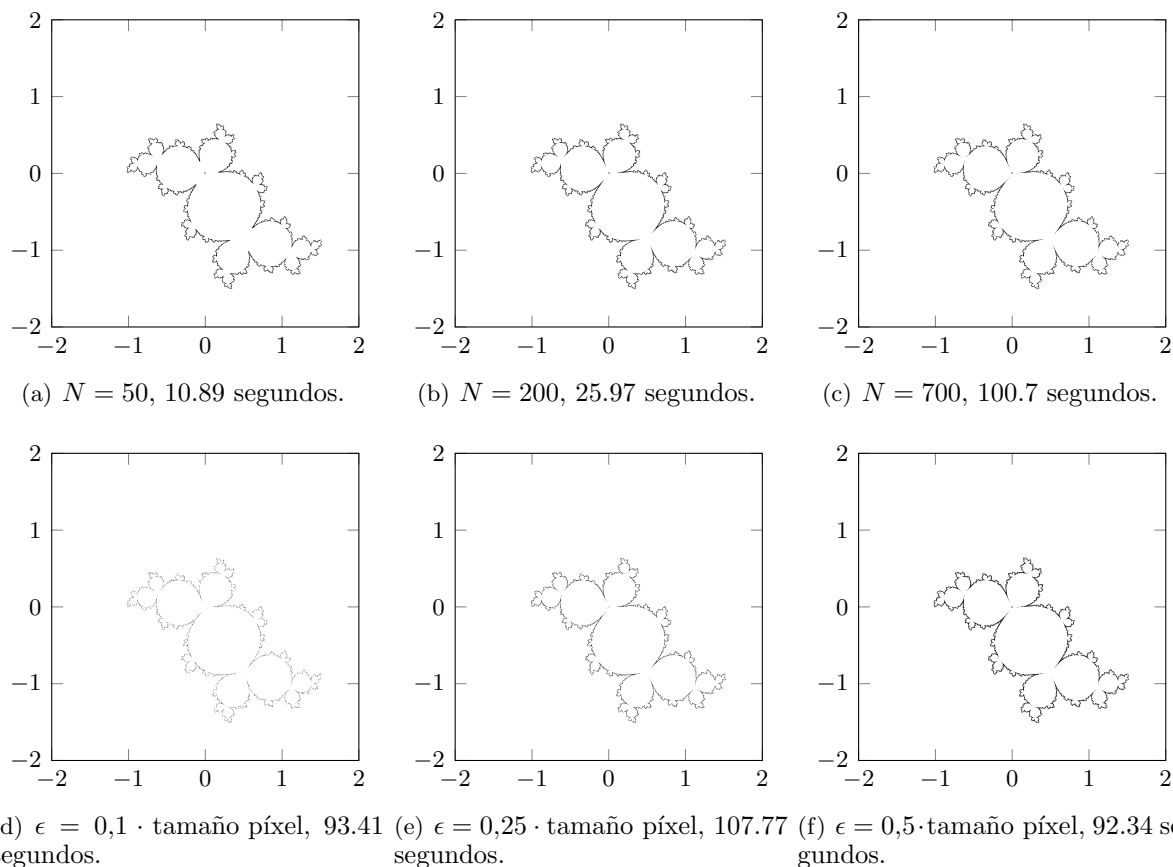
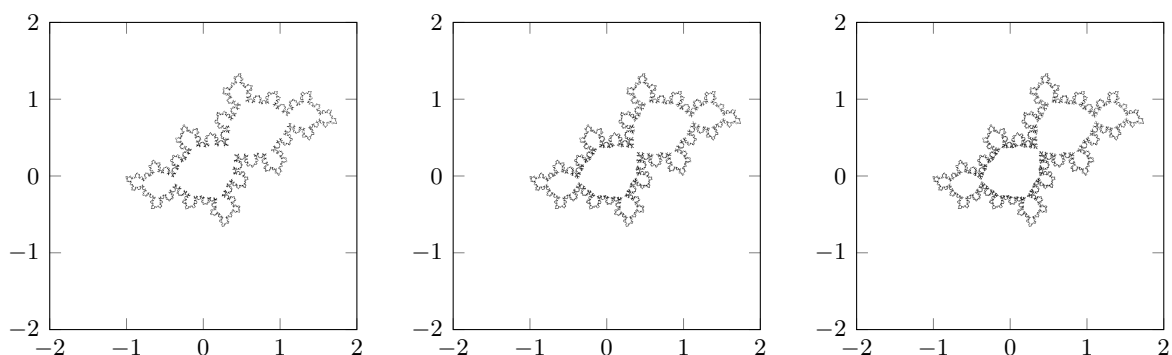


Figura 5.19: Algoritmo de Henriksen para dibujar caso parabólico de periodo 3. En la primera fila se ha fijado el parámetro $\epsilon = 0,4 \cdot \text{tamaño píxel}$ y en la segunda fila el parámetro $N = 700$.

Disco de Siegel

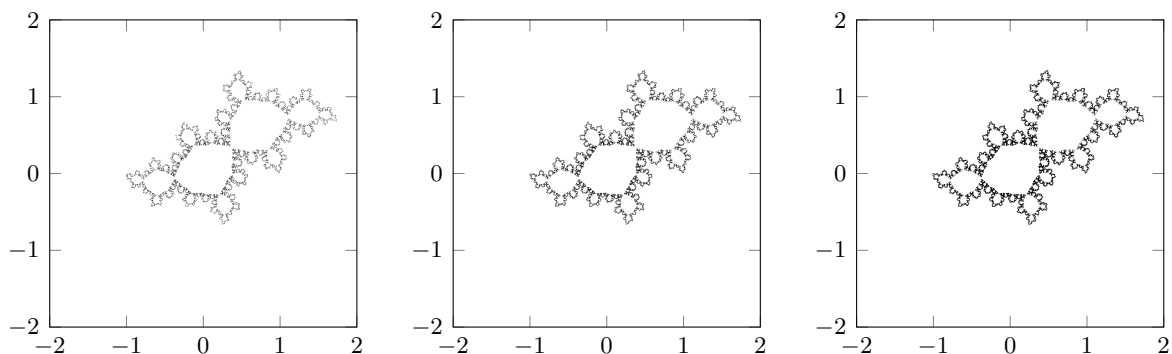
Para poder dibujar los discos de Siegel necesitamos la función $f(z) = \lambda z + \lambda z^2$, siendo $\lambda = e^{2\pi i\theta}$ y $\theta = \frac{\sqrt{5}-1}{2}$. En la Figura 5.20 podemos ver que ocurre lo mismo que hemos comentado en el caso anterior. Si fijamos ϵ y variamos el número de iterados, podemos ver que la figura mejora, los bulbos están más definidos y aumenta la cantidad de detalles. Mientras que si fijamos N y modificamos ϵ , si el valor es muy pequeño la figura se ve muy clara ya que los puntos que la forman son muy pequeños. A medida que se va aumentando el valor, el grosor de la figura aumenta.



(a) $N = 100$, 13.87 segundos.

(b) $N = 300$, 39.06 segundos.

(c) $N = 700$, 65.46 segundos.



(d) $\epsilon = 0,1 \cdot \text{tamaño píxel}$, 76.09 segundos. (e) $\epsilon = 0,25 \cdot \text{tamaño píxel}$, 77.35 segundos. (f) $\epsilon = 0,5 \cdot \text{tamaño píxel}$, 73.44 segundos.

Figura 5.20: Algoritmo de Henriksen para dibujar disco de Siegel. En la primera fila se ha fijado el parámetro $\epsilon = 0,2 \cdot \text{tamaño píxel}$ y en la segunda fila el parámetro $N = 700$.

Caso de una dendrita

Para el caso de la dendrita, establecemos el valor de $c = i$. En la Figura 5.21 podemos observar los resultados obtenidos. En primer lugar, vemos que cuando fijamos ϵ y el número de iterados es $N = 20$ hay zonas por la parte central y superior de la figura que no están bien definidas. Esto se debe a que el número de iteraciones es demasiado bajo y no se puede obtener la figura de forma completa. Cuando aumentamos N vemos que la figura para $N = 50$ es la misma que para $N = 100$, es decir, no se aprecia ninguna diferencia. Además, los dos dibujos tardan aproximadamente lo mismo en ejecutarse. Esto se debe a que aunque el número de iteraciones aumente, la figura convergerá a infinito en la misma iteración y se dejará de iterar el bucle para iterar otro punto debido a la condición de escape a infinito. Por tanto, harán el mismo número de iteraciones aunque tengan valores de N distintas. Por otro lado, si fijamos N y modificamos ϵ , vemos que el grosor de la figura varía dependiendo del ϵ escogido.

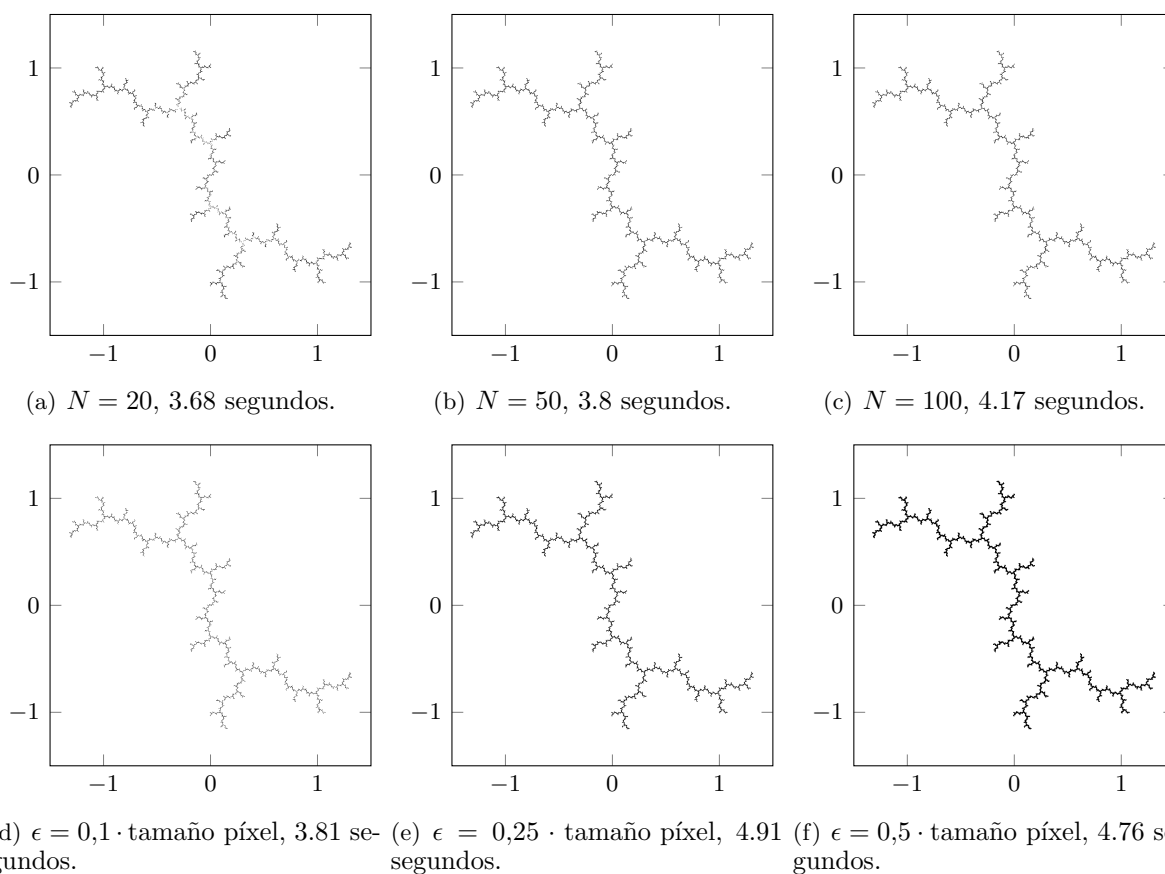
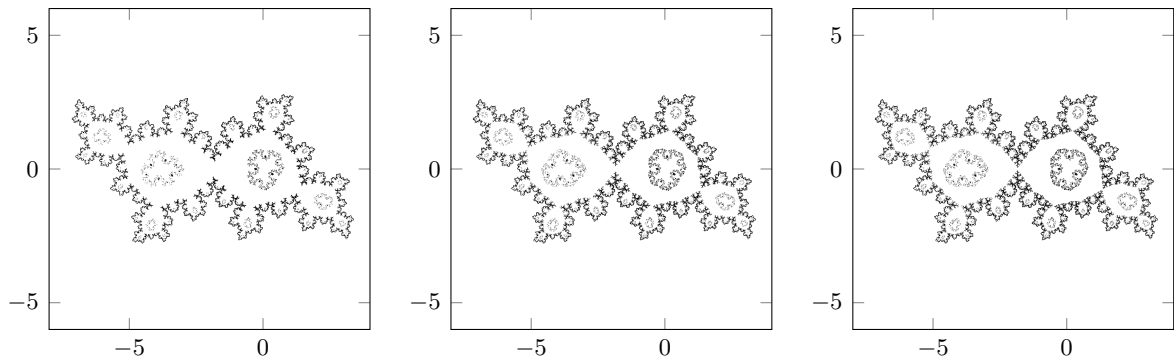


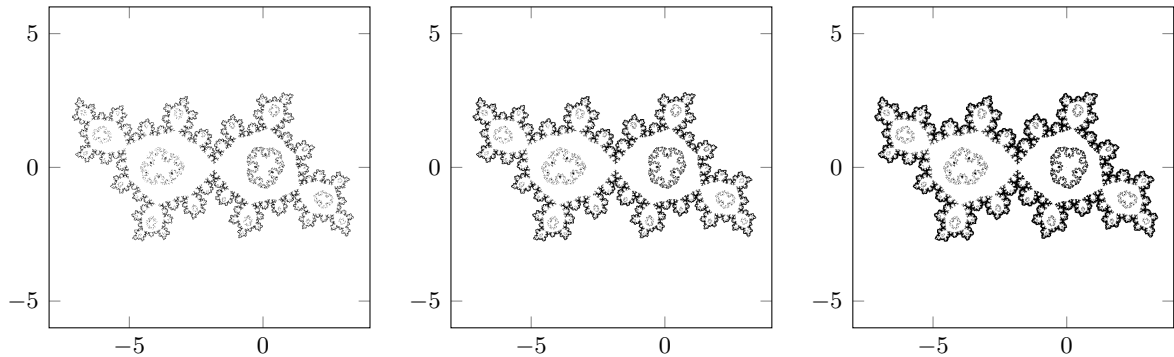
Figura 5.21: Algoritmo de Henriksen para dibujar el conjunto de Julia para $c = i$ que es una dendrita. En la primera fila se ha fijado el parámetro $\epsilon = 0,15 \cdot$ tamaño píxel y en la segunda fila el parámetro $N = 100$.

Anillo de Herman

Para dibujar un anillo de Herman necesitaremos la función $f(z) = \lambda z^2 \frac{az + 1}{z + a}$ siendo $\lambda = e^{2\pi it}$, $t = 0,61517321588\dots$ y $a = 0,25$. Primero veremos la figura entera, mientras que después haremos una ligera ampliación donde se encuentra realmente el anillo de Herman. En la Figura 5.22 podemos ver que si fijamos ϵ y aumentamos el número de iterados la figura mejora considerablemente. Mientras que si fijamos el valor $N = 500$ y modificamos ϵ , el grosor de la figura aumenta proporcionalmente. Podemos ver que hacen falta muchos iterados para que los dibujos se vean bien, pero el resultado es muy bueno. Cabe destacar que los tiempos de ejecución son muy elevados ya que para $N = 500$ ha tardado aproximadamente 15 minutos y para $N = 1500$ ha tardado aproximadamente 40 minutos en terminarse.



(a) $N = 100$, 50.55 segundos. (b) $N = 500$, 196.97 segundos. (c) $N = 1500$, 601.99 segundos.



(d) $\epsilon = 0,1 \cdot \text{tamaño píxel}$, 231.74 segundos. (e) $\epsilon = 0,25 \cdot \text{tamaño píxel}$, 284.3 segundos. (f) $\epsilon = 0,5 \cdot \text{tamaño píxel}$, 254.57 segundos.

Figura 5.22: Algoritmo de Henriksen para dibujar un anillo de Herman. En la primera fila se ha fijado el parámetro $\epsilon = 0,15 \cdot \text{tamaño píxel}$ y en la segunda fila el parámetro $N = 500$.

Si hacemos una ampliación y miramos con más detalle donde se encuentra realmente al anillo de Herman obtenemos los resultados de la Figura 5.23. Podemos observar que ocurre lo

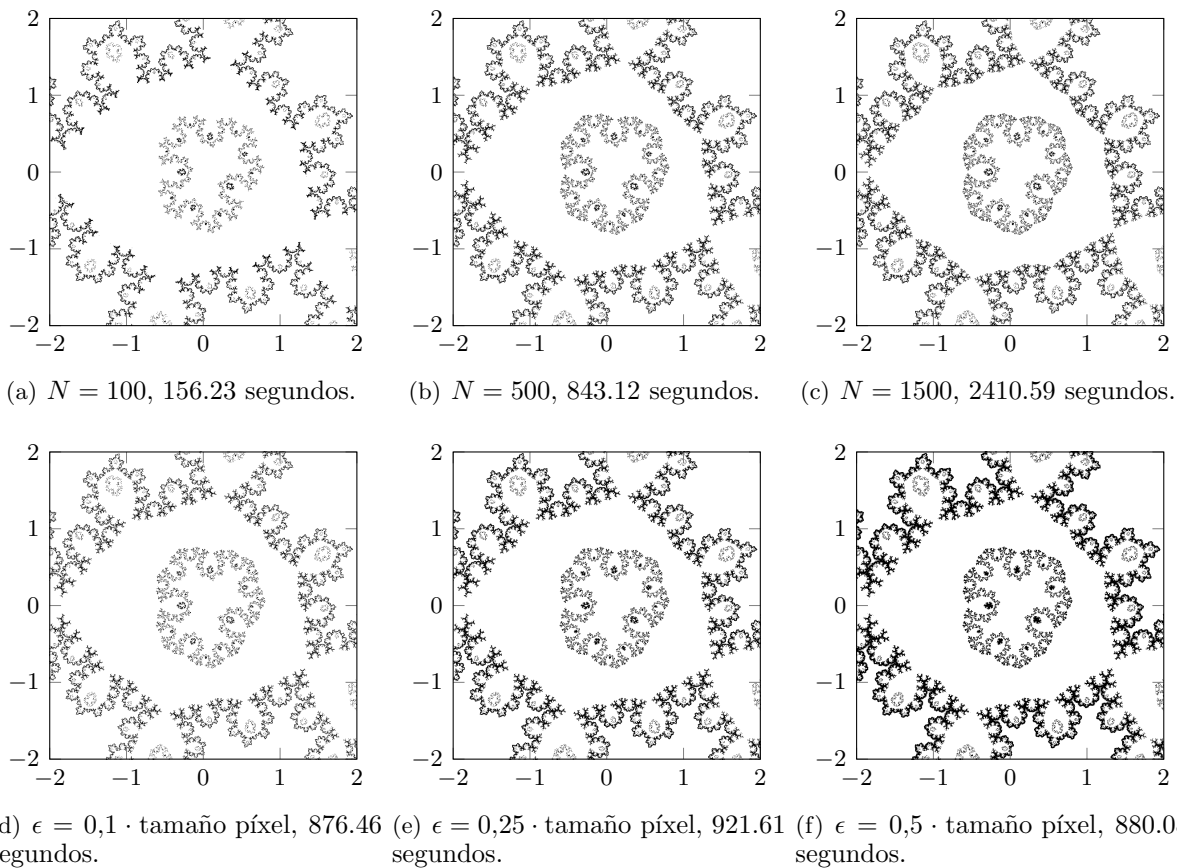


Figura 5.23: Algoritmo de Henriksen para dibujar un anillo de Herman. En la primera fila se ha fijado el parámetro $\epsilon = 0,15 \cdot \text{tamaño píxel}$ y en la segunda fila el parámetro $N = 500$.

mismo que si no hacemos la ampliación.

Conjunto de Mandelbrot

Para dibujar el conjunto de Mandelbrot con el algoritmo de Henriksen es necesario hacer una modificación del código utilizado para dibujar el conjunto de Julia ya que como hemos visto en el principio de la sección, no se utiliza la misma condición ni se calcula de la misma forma la derivada acumulada.

En la Figura 5.24 podemos ver varias representaciones del conjunto de Mandelbrot. En primer lugar, podemos observar que a medida que aumenta el número de iteraciones los bulbos de las figuras están más cerrados y mejor definidos. Aunque cabe destacar que con el valor de ϵ

escogido no se obtienen buenos dibujos ya que la antena de la izquierda parece que sea disconexa pero no es cierto ya que sabemos que el conjunto de Mandelbrot es conexo.

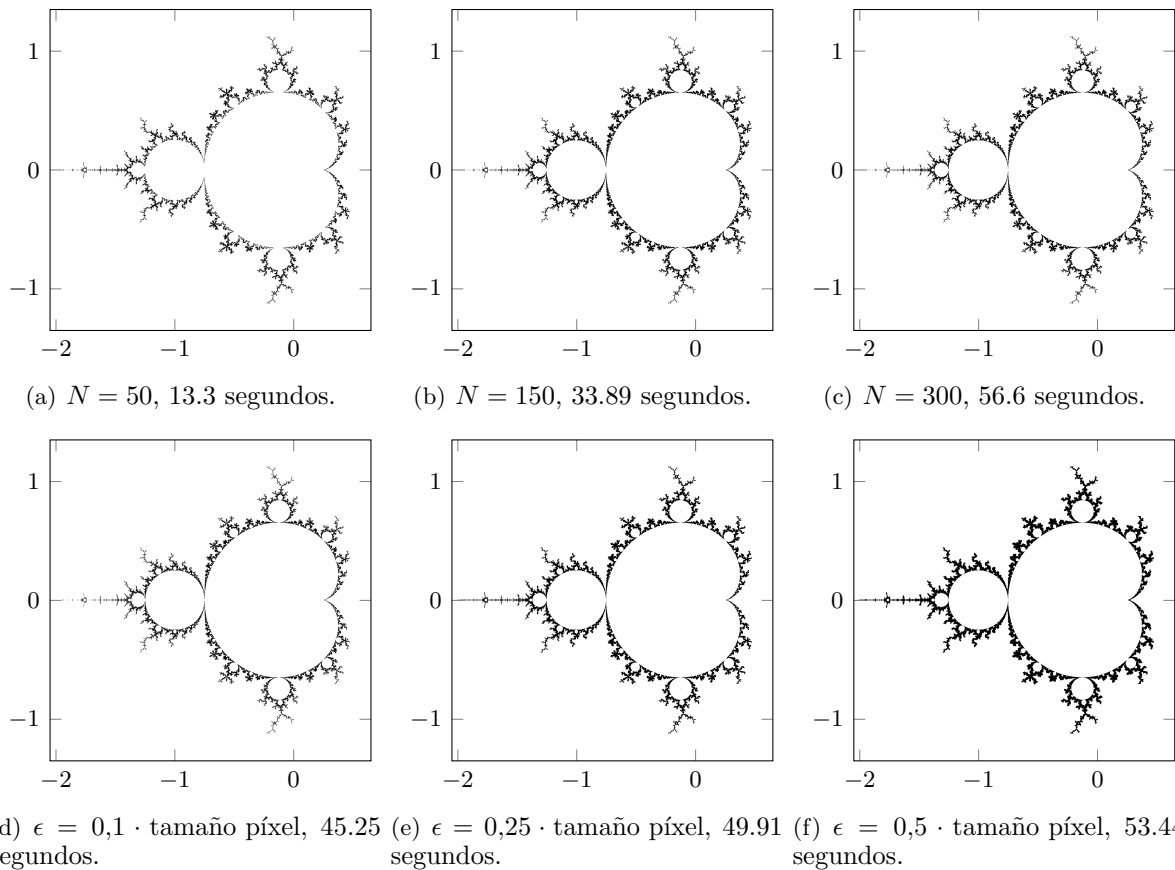
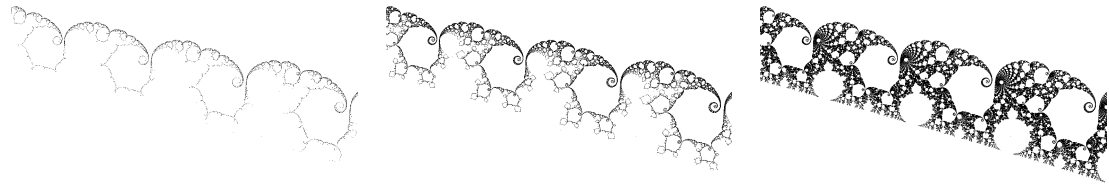


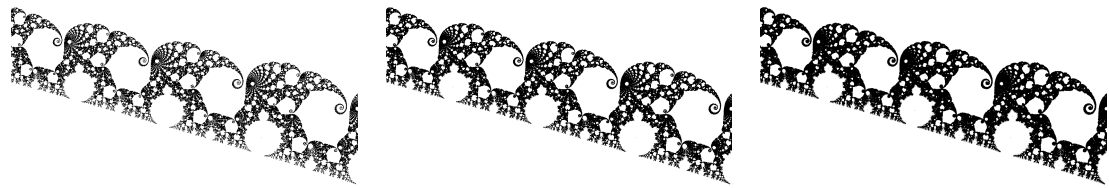
Figura 5.24: Algoritmo de Henriksen para dibujar el conjunto de Mandelbrot. En la primera fila se ha fijado el parámetro $\epsilon = 0,15 \cdot \text{tamaño píxel}$ y en la segunda fila el parámetro $N = 300$.

Si aumentamos el tamaño de ϵ y fijamos el valor N , podemos observar que el grosor de la figura aumenta en proporción y se soluciona el problema que hemos comentado antes. A partir de $\epsilon = 0,25 \cdot \text{tamaño píxel}$, la antena de la izquierda ya sí se ve conexa y unida al resto de la figura.

A continuación vamos a ver unos dibujos sobre los elefantes en la Figura 5.25. En primer lugar, se observa que se necesita un valor de N elevado para que el dibujo se distinga bien y se puedan apreciar los elefantes. Pero, cuando fijamos el valor N y variamos el valor ϵ , vemos que los dibujos tienen un grosor mayor, aunque cuando $\epsilon = 0,5 \cdot \text{tamaño píxel}$ las zonas negras no se distinguen de la misma manera que para los otros valores y se pierden detalles.



(a) $N = 100$, 59.43 segundos. (b) $N = 150$, 73.39 segundos. (c) $N = 500$, 240.38 segundos.



(d) $\epsilon = 0,1 \cdot \text{tamaño píxel}$, (e) $\epsilon = 0,25 \cdot \text{tamaño píxel}$, (f) $\epsilon = 0,5 \cdot \text{tamaño píxel}$, 203.72 segundos.
197.79 segundos. 215.88 segundos.

Figura 5.25: Algoritmo de Henriksen para dibujar los elefantes. En las figuras (a), (b) y (c) se ha fijado el parámetro $\epsilon = 0,15 \cdot \text{tamaño píxel}$ y en las figuras (d),(e) y (f) el parámetro $N = 500$.

Ampliación del conjunto de Mandelbrot

En este caso vamos a ver cómo funciona este algoritmo para dibujar las antenas de los bulbos del cardioide principal del conjunto de Mandelbrot. Los resultados obtenidos son los de la Figura 5.26 y como en los casos anteriores, si aumentamos el número de iterados se puede apreciar que el bulbo está más cerrado y aparecen más detalles. Mientras que si modificamos el parámetro ϵ , aumenta el grosor de la figura. En general, podemos comprobar que cuantas más iteraciones mejor se ve el dibujo y la calidad de las antenas se mantiene. Es decir, mejoran todas las partes por igual, ya sean las antenas o el bulbo en cuestión.

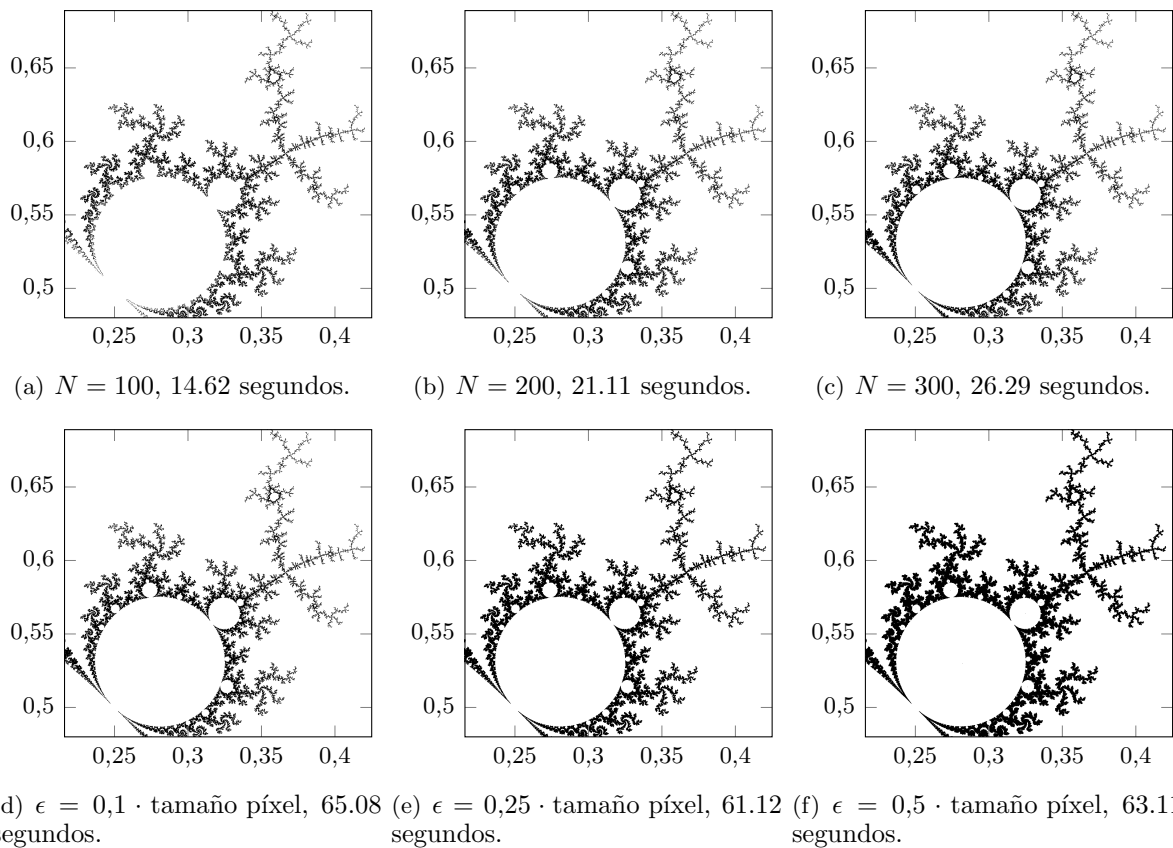


Figura 5.26: Algoritmo de Henriksen para dibujar bulbo del conjunto de Mandelbrot. En la primera fila se ha fijado el parámetro $\epsilon = 0,15 \cdot \text{tamaño píxel}$ y en la segunda fila el parámetro $N = 300$.

5.4. Comparación de algoritmos

En esta sección vamos a hacer una comparación de los diferentes algoritmos para todos los casos que hemos visto en los apartados anteriores. Para ello, cogeremos los mejores dibujos obtenidos variando los diferentes parámetros. El orden de las imágenes de las figuras que a partir de aquí vamos a comparar van a ser, en primer lugar el resultado del algoritmo de escape, después del algoritmo de iteración inversa y finalmente del algoritmo de Henriksen.

Empezaremos comparando el caso hiperbólico en la Figura 5.27 para $c = 0,36 + 0,34i$ (a), (b) y (c) y para $c = -0,672 + 0,337i$ (d), (e), (f). Podemos comprobar que el algoritmo que más tiempo tarda en ejecutarse es el algoritmo de Henriksen, mientras que el que menos tiempo de ejecución necesita es el algoritmo de escape. Los resultados del algoritmo de iteración inversa y de Henriksen son muy similares, aunque los tres resultados obtenidos son bastante buenos.

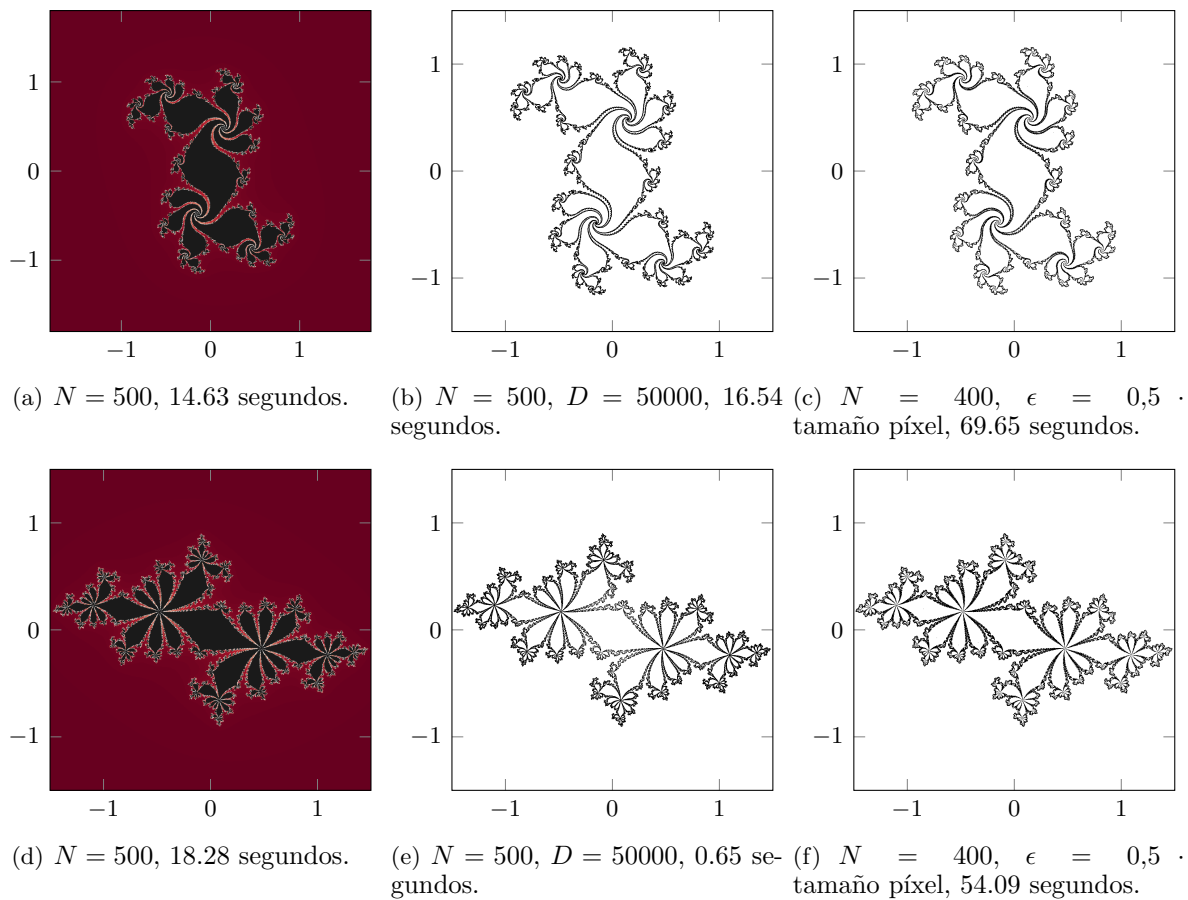


Figura 5.27: Comparación caso hiperbólico.

Ahora vamos a comparar algún caso cercano a parabólico obtenido a través del valor $c = 0,285 + 0,01i$ en la Figura 5.28. Podemos ver que con el algoritmo de iteración inversa y el de Henriksen se obtienen unos resultados muy parecidos aunque Henriksen es ligeramente mejor. Lo único a destacar es la diferencia de tiempo de ejecución entre los dos algoritmos ya que, a diferencia del caso anterior, en este caso podemos ver que el algoritmo que más tiempo de ejecución ha necesitado ha sido el algoritmo de iteración inversa. También podemos ver que en el algoritmo de escape no se pueden apreciar los detalles del dibujo tal y conforme podemos verlos en los otros dos algoritmos. Por este motivo el algoritmo no es la mejor opción para representar este caso ya que es bastante delicado.

A continuación vamos a comparar el caso parabólico de periodo 3 en la Figura 5.29. Podemos ver que el algoritmo de iteración inversa es casi inmediato, mientras que el algoritmo de Henriksen ha tardado casi dos minutos en ejecutarse. Además, en ambos dibujos las puntas casi llegan a tocarse pero hace falta tener un número de iteraciones muy elevado. Sin embargo, el

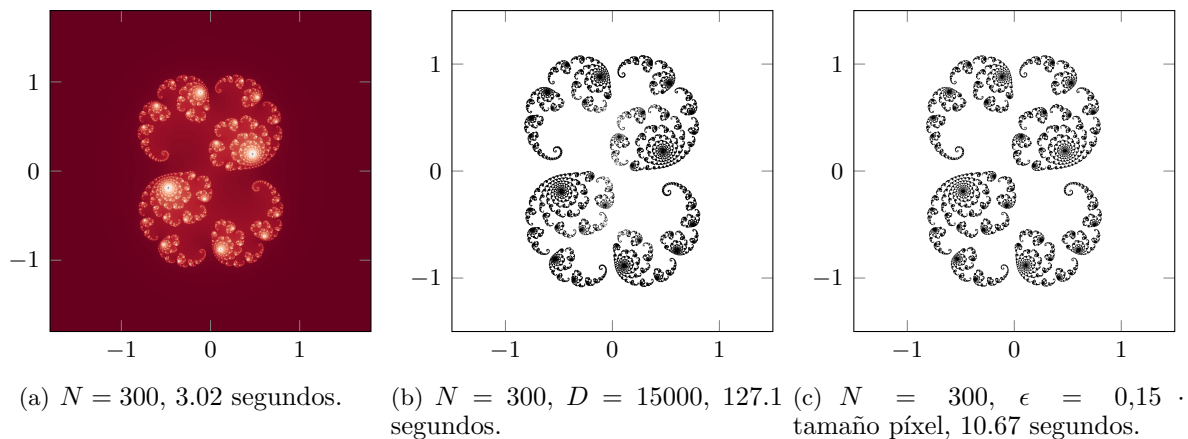


Figura 5.28: Comparación caso cercano a parabólico.

algoritmo de escape no es bueno para este tipo de casos ya que aunque aumentemos el número de iteraciones, no se aprecian cambios significativos en la figura.

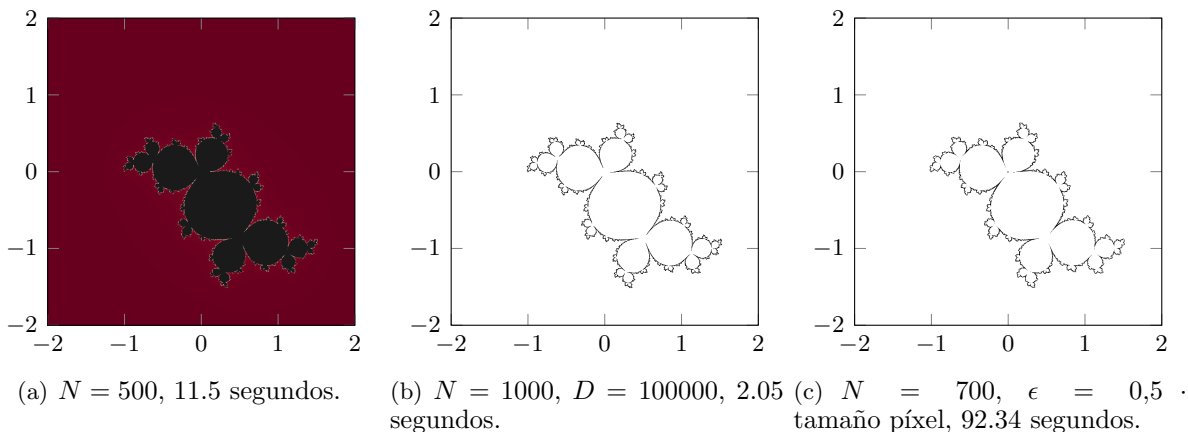


Figura 5.29: Comparación caso parabólico de periodo 3.

Ahora vamos a comparar los resultados obtenidos al dibujar los discos de Siegel en la Figura 5.30. Este caso es muy similar al caso anterior. Es decir, con el algoritmo de escape no obtenemos un dibujo tan bueno como con los otros dos algoritmos. Y, entre los algoritmos de iteración inversa y Henriksen los resultados son muy similares, mientras que los tiempos de ejecución son bastante distantes.

La siguiente comparación que vamos a realizar es para el caso de una dendrita que se obtiene a través del valor $c = i$ en la Figura 5.31. En este caso podemos observar que tanto con algoritmo de escape como con el algoritmo de Henriksen se ve la dendrita a la perfección y sin necesidad de

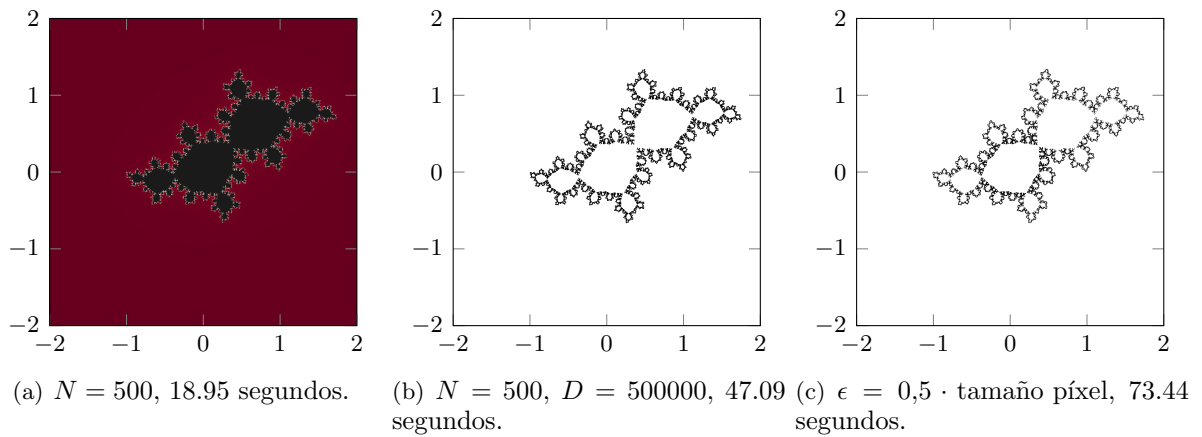


Figura 5.30: Comparación de los discos de Siegel.

utilizar parámetros muy elevados. Mientras que el resultado del algoritmo de iteración inversa necesita un valor D muy grande para poder conseguir que la figura de la dendrita esté unida por la zona central.

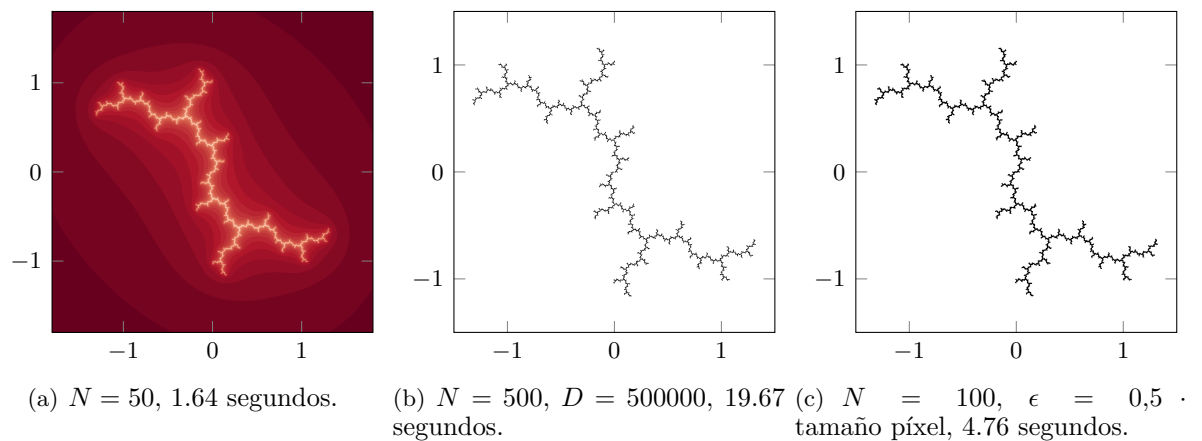


Figura 5.31: Comparación del caso de una dendrita con valor $c = i$.

A partir de aquí las comparaciones que vamos a continuar haciendo van a ser del algoritmo de escape y del algoritmo de Henriksen. Esto se debe a que el algoritmo de iteración inversa no lo vamos a utilizar para representar el conjunto de Mandelbrot. Además, la función f para poder dibujar los anillos de Herman tiene grado 3, por lo que el algoritmo de iteración inversa queda descartado debido a la gran dificultad para resolver la ecuación y obtener la función inversa. Por tanto, el algoritmo de iteración inversa solo funciona para funciones cuadráticas en las que se pueda resolver la ecuación para obtener la inversa de forma explícita y esto es una gran limitación frente a los otros dos algoritmos.

A continuación vamos a comparar el conjunto de Mandelbrot. Los resultados obtenidos en la Figura 5.32 son ambos de buena calidad y se obtienen muy buenos resultados con los dos algoritmos, aunque los tiempos de ejecución sean muy distantes. Pero, cabe destacar que con el algoritmo de escape tenemos la limitación que cuando aumentamos el número de iterados para que el dibujo tenga más calidad, las antenas se distinguen peor y se dejan de ver con tanta claridad. La antena de la izquierda puede parecer que sea disconexa del resto de la figura, pero en realidad está todo conectado.

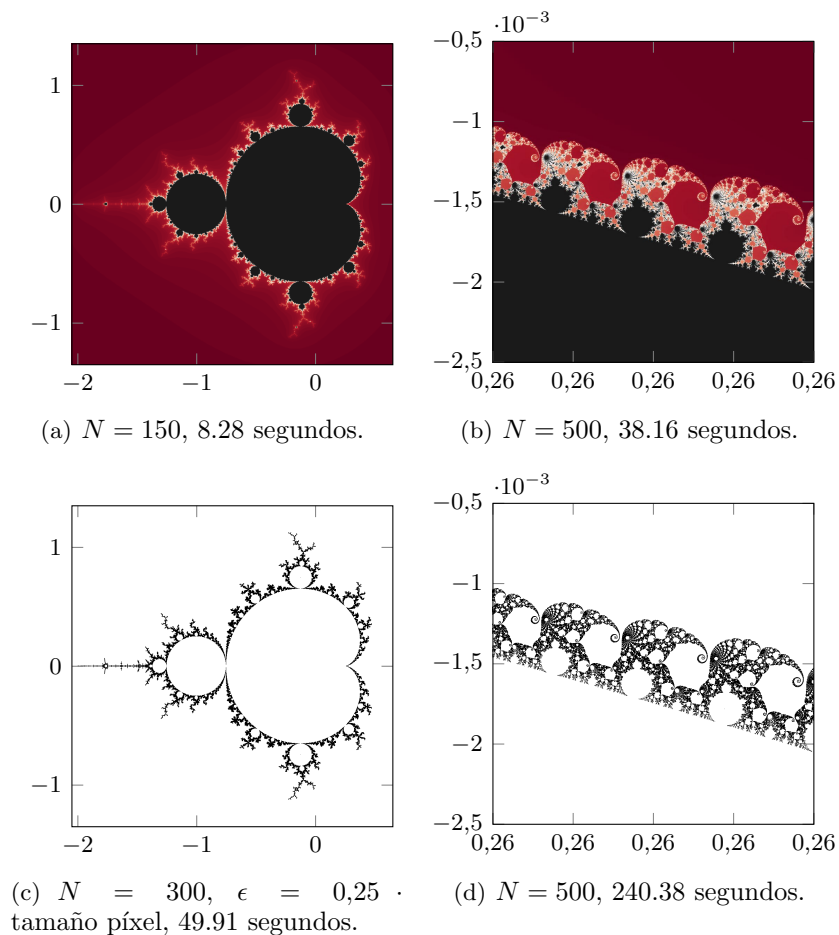


Figura 5.32: Comparación del conjunto de Mandelbrot.

Los resultados obtenidos haciendo una ampliación del conjunto de Mandelbrot son los que podemos ver en la Figura 5.33. La gran diferencia que se puede apreciar es la calidad de las antenas ya que con el algoritmo de escape es difícil distinguir detalles ya que aparecen con muy poca intensidad. Mientras que con el algoritmo de Henriksen se ven todas las partes por igual y se distinguen a la perfección todos los detalles.

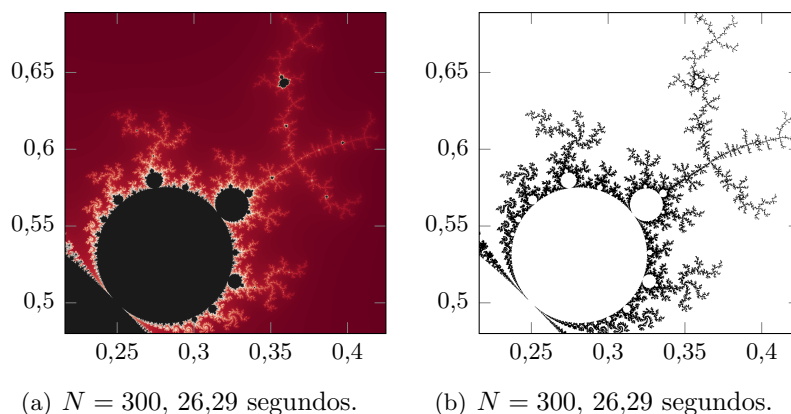


Figura 5.33: Comparación de una ampliación de las antenas del conjunto de Mandelbrot.

Los resultados de los anillos de Herman los podemos ver en la Figura 5.34. Observamos que el dibujo del algoritmo de escape es aceptable. Puede servir perfectamente para tener una idea de cómo es el anillo de Herman y distinguir su forma. Pero, a comparación del resultado obtenido con el algoritmo de Henriksen, los resultados no tienen nada que ver. Se ve claramente que con Henriksen se distinguen todas las partes mucho mejor y se obtiene una gran cantidad de detalles que con el algoritmo de escape no se pueden apreciar.

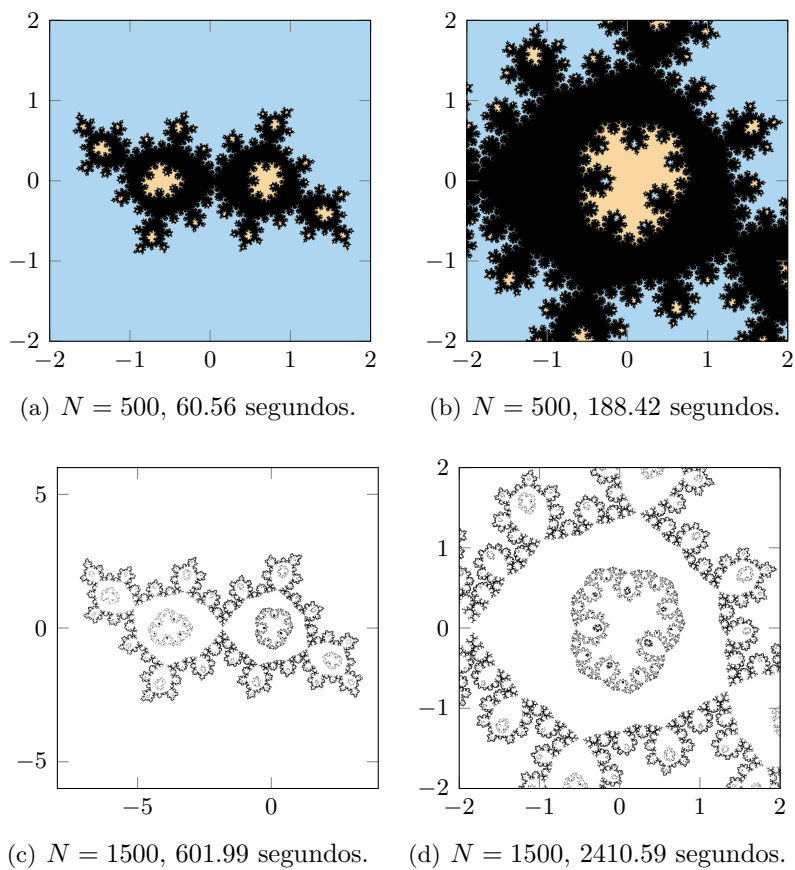


Figura 5.34: Comparación de los anillos de Herman.

5.5. Conclusión final

En resumen, hemos visto que el algoritmo de escape funciona perfectamente para casos hiperbólicos, ya que únicamente aumentando el número de iterados se obtienen resultados muy buenos y el tiempo de ejecución es relativamente pequeño para la cantidad de cálculos realizados. Sin embargo, para los casos en los que los parámetros están cerca de puntos parabólicos o si son puntos periódicos indiferentes, no se consigue distinguir bien todas las partes del dibujo y aunque aumentemos el número de iterados el resultado no mejora. En el caso de los discos de Siegel, se obtienen resultados aceptables pero no tiene nada que ver con el resultado obtenido con Henriksen. Para dibujar las dendritas y el conjunto de Mandelbrot este algoritmo funciona muy bien menos cuando aparecen antenas ya que no se pueden distinguir con todo el detalle posible. Por último, en cuanto a los anillos de Herman se obtiene un dibujo aceptable que mejora mucho al haber realizado la modificación de la condición de parada.

En cuanto al algoritmo de iteración inversa, hemos visto que funciona perfectamente para funciones cuadráticas. Su tiempo de ejecución en casos hiperbólicos es casi instantáneo y el resultado es muy bueno siempre que se elijan bien los parámetros. Mientras que, en casos cercanos a parabólicos o en casos parabólicos, el tiempo de ejecución es más variable y depende del parámetro utilizado, aunque finalmente el resultado sea muy bueno. En el caso de los discos de Siegel, los tiempos de ejecución son muy pequeños y el resultado es muy similar al del algoritmo de Henriksen, teniendo en cuenta que el algoritmo de Henriksen tarda mucho más tiempo en ejecutarse. En el caso de dibujar una dendrita hemos visto que necesita valores de los parámetros excesivamente altos para que se consiga cerrar por la zona del centro, aunque se obtienen buenos resultados también. Por último, los anillos de Herman no se pueden obtener a través de este método ya que tiene una limitación. Para ejecutar este algoritmo es necesario poder calcular la ecuación inversa y si el grado de la función es mayor que 2, esto supone un grave problema porque la dificultad para resolverla es muy alta o incluso imposible.

Finalmente, el algoritmo de Henriksen es el más versátil y nos garantiza que va a funcionar en todos los casos, aunque el tiempo de ejecución pueda variar. Además, todos los resultados obtenidos son buenos y si no lo son, aumentando el número de iterados y la resolución se consigue una mejora importante. Por tanto, con este algoritmo sabemos que siempre vamos a obtener dibujos muy satisfactorios y de buena calidad.

Bibliografía

- [1] N. Fagella B. Branner. *Quasiconformal Surgery in Holomorphic Dynamics*. Cambridge University Press, 2014.
- [2] Alan F. Beardon. *Iteration on Rational Functions*. Springer-Verlag, 1990.
- [3] A. Cayley. Applications of the newton-fourier method to an imaginary root of an equation. *Quat. J. of Pure and App. Math.*, 16:179–85, 1879.
- [4] A. Cayley. The newton-fourier method imaginary problem. *Am. J. of Math.*, 2:97, 1879.
- [5] A. Chéritat. Near parabolic renormalization for unicritical holomorphic maps. *Arnold Mathematical Journal*, 8 (2):169–270, 2022.
- [6] John B. Conway. *Functions of One Complex Variable I*. Springer-Verlag, second edition, 1978.
- [7] P. Fatou. Sur les substitutions rationnelles. *Comp. Rend. heb. S. Acad. Sci.*, 164:806–808, 1917.
- [8] P. Fatou. Sur les substitutions rationnelles. *Comp. Rend. heb. S. Acad. Sci.*, 165:992–995, 1917.
- [9] P. Fatou. Sur les équations fonctionelles. *Bull. Soc. mat. Fr.*, 47:161–271, 1919.
- [10] P. Fatou. Sur les équations fonctionelles. *Bull. Soc. mat. Fr.*, 48:33–94 y 208–314, 1920.
- [11] I. Hargittai and C.A. Pickover. *Spiral Symmetry*. World Scientific, 1992.
- [12] C. Henriksen. *Holomorphic dynamics and Herman rings*. PhD thesis, Technical university of Denmark, 1997.
- [13] G. Julia. Mémoire sur l’iteration des fonctionsa rationnelles. *J. Math. Pur. Appl.*, 8:47–245, 1918.
- [14] A. Ya. Khinchin. *Continued Fractions*. Dover Publications, 1997.

- [15] R. Pérez Marco. Las medallas fields. *La Gaceta de la RSME*, 21:407–416, 2018.
- [16] J. Milnor. *Dynamics in one complex variable*. Princeton University Press, third edition, 2006.
- [17] X. Jarque N. Fagella. *Iteración compleja y fractales*. Vicens Vives, 1991.
- [18] W. Schlag. *A concise course in complex analysis and Riemann surfaces*.
- [19] E. Schröder. Ueber unendlich viele algorithmen zur auflösung der gleichungen. *Mathematische Annalen*, 2:317–65, 1870.
- [20] E. Schröder. Ueber iterite functionen. *Mathematische Annalen*, 3:296–322, 1871.
- [21] M. Shishikura. On the quasiconformal surgery of rational functions. *Annales scientifiques de l'É.N.S.*, 20:1–29, 1987.
- [22] C. L. Siegel. Iterations of analytic functions. *Ann. of Math.*, 43:607–612, 1942.
- [23] D. Sullivan. Quasiformal homeomorphisms and dynamics. *Annals of Math*, 122:401–418, 1985.
- [24] D. Sullivan. Quasiformal homeomorphisms and dynamics i: Solution on fatou-julia problem on wandering domains. *Ann. Maths.*, 122:401–418, 1998.
- [25] A. Douady y J. Hubbard. *Étude dynamique des polinomes complexes*. Pub. Math d'Orsay, 1984.
- [26] L. Carleson y T. W. Gamelin. *Complex Analysis*. Springer, 2001.

Anexo A

Anexo I

A.1. Algoritmo de escape

A.1.1. Conjunto de Mandelbrot

```
pixels = 1000

# definimos los limites del dibujo
width = 4
height = 4

minX = -2
minY = -2
maxX = minX + width
maxY = minY + height

def f(z):
    return z*z + c

def algoritmo_escape(c):
    z = 0.0 + 0.0j
    for i in range(max_iter):
        z = f(z)
        if abs(z) >= 4:
            return i
```

```

    return max_iter

max_iter = 500

# creamos la malla de 0s
grid = np.zeros([pixels, pixels])

# ahora almacenamos en la malla el numero de iteracion
# en la que escapa a infinito
for fila, x in enumerate(np.linspace(minX, maxX, num=pixels)):
    for col, y in enumerate(np.linspace(maxY, minY, num=pixels)):
        c = complex(x, y)
        grid[col][fila] = algoritmo_escape(c)

# definimos las pulgadas de la imagen
plt.figure(dpi=250)

#definimos la escala de color
plt.imshow(grid, cmap='RdGy')
plt.axis('off')

# guardamos la imagen como un archivo PNG
plt.savefig("ConjuntoMandelbrot.png", bbox_inches='tight', pad_inches=0)
plt.show()

```

A.1.2. Conjunto de Julia

```

pixels = 1000

# definimos los limites del dibujo
width = 4
height = 4

minX = -2
minY = -2
maxX = minX + width
maxY = minY + height

def f(z):
    return z*z + c

```

```

def algoritmo_escape(z):
    for i in range(max_iter):
        z = f(z)
        if abs(z) >= 4:
            return i
    return max_iter

max_iter = 100
c = complex(-1, 0)

# creamos la malla de 0s
grid = np.zeros([pixels, pixels])

# ahora la rellenos la malla de 0s
for fila, x in enumerate(np.linspace(minX, maxX, num=pixels)):
    for col, y in enumerate(np.linspace(maxY, minY, num=pixels)):
        z = complex(x, y)
        grid[col][fila] = algoritmo_escape(z)

# definimos las pulgadas de la imagen
plt.figure(dpi=250)
plt.imshow(grid, cmap='RdGy')
plt.axis('off')

# guardamos la imagen como un archivo PNG
plt.savefig("ConjuntoJulia.png", bbox_inches='tight', pad_inches=0)
plt.show()

```

A.2. Algoritmo de iteración inversa

```

# creamos una nueva imagen de 1000x1000 pixeles con fondo blanco
pixels = 1000
img = Image.new('RGB', (pixels, pixels), color='white')

# definimos los limites del dibujo
width = 3
height = 3

minX = -1.5

```

```

minY = -1.5
maxX = minX + width
maxY = minY + height

# se calculan parametros para obtener posicion matriz img
divX = width / pixels
divY = height / pixels

def inverse_iteration(z):
    return np.sqrt(z - c)

# calcula la posicion de un punto en la matriz
def get_pos(x, y):
    i = (maxX + x) / divX
    j = (maxY - y) / divY
    return round(i), round(j)

def algoritmo_it_inversa(z, derAcu, nIter):
    if nIter >= max_iter or derAcu > D:
        return False

    derAcu *= abs(der_punto(z))

    i, j = get_pos(z.real, z.imag)
    img.putpixel((i, j), (0, 0, 0))

    inv = inverse_iteration(z)

    algoritmo_it_inversa(inv, derAcu, nIter + 1)
    algoritmo_it_inversa(-inv, derAcu, nIter + 1)

    return True

# definimos parametros
max_iter = 100
D = 20000

c = complex(-1, 0)

# definimos z_0 (punto repulsor)
z_0 = 0.5 * (1 + np.sqrt(1 - 4 * c))

```

```

# calculamos la derivada de  $z^2 + c$ 
z = symbols('z')
der = diff(z * z + c, z)
der_punto = lambdify(z, der)

algoritmo_it_inversa(z_0, 1, 0)
plt.imshow(img)

# guardamos la imagen como un archivo PNG
img.save('prueba.png')

plt.show()

```

A.3. Algoritmo de Henriksen

A.3.1. Conjunto de Mandelbrot

```

def f(z):
    return z * z + c

# creamos una nueva imagen de 1000x1000 pixeles con fondo blanco
pixels = 1000
img = Image.new('RGB', (pixels, pixels), color='white')

#definimos los limites del dibujo
width = 4
height = 4

minX = -2
minY = -2
maxX = minX + width
maxY = minY + height

#definimos parametros
max_iter = 400
escape = 5

```

```

pixel_size = (maxX - minX) / pixels
epsilon = 0.20 * pixel_size

# calculamos derivada
z = symbols('z')
der = diff(z * z, z)
der_punto = lambdify(z, der)

# Henriksen
for fila, x in enumerate(np.linspace(minX, maxX, num=pixels)):
    for col, y in enumerate(np.linspace(maxY, minY, num=pixels)):
        c = complex(x, y)

        z_0 = c + 0j
        z = c + 0j

        derAcu = 1 + 0j

        for n in range(1, max_iter):
            derAcu = derAcu*2*z+1
            z = f(z)

            # comprobamos si z_0 es un punto periodico
            if abs(z) < abs(epsilon*derAcu):
                img.putpixel((fila, col), (0, 0, 0))
                break

            # comprobamos si el punto ha escapado a infinito
            if abs(z) > escape:
                break

img.show()

# guardamos la imagen como un archivo PNG
img.save(ConjuntoMandelbrot.png)

```

A.3.2. Conjunto de Julia

```

def f(z):
    return z * z + c

```



```

# creamos una nueva imagen de 1000x1000 pixeles con fondo blanco
pixels = 1000
img = Image.new('RGB', (pixels, pixels), color='white')

#definimos los limites del dibujo
width = 3
height = 3

minX = -1.5
minY = -1.5
maxX = minX + width
maxY = minY + height

# definimos parametros
max_iter = 250
escape = 5

pixel_size = (maxX - minX) / pixels
epsilon = 0.15 * pixel_size

c = complex(-1, 0)

# calculamos la derivada de  $z^2 + c$ 
z = symbols('z')
der = diff(z * z + c, z)
der_punto = lambdify(z, der)

# Henriksen
for fila, x in enumerate(np.linspace(minX, maxX, num=pixels)):
    for col, y in enumerate(np.linspace(maxY, minY, num=pixels)):
        z_0 = complex(x, y)
        z = complex(x, y)
        derAcu = 1 + 0j

        for n in range(1, max_iter):
            derAcu *= der_punto(z)
            z = f(z)

            if abs((z_0 - z) / (derAcu - 1)) < epsilon:

```

```
        img.putpixel((fila , col), (0, 0, 0))
        break

    if abs(z) > escape:
        break

plt.imshow(img)

# guardamos la imagen como un archivo PNG
img.save(ConjuntoJulia.png)
```