

Fractales

Ejemplos, historia y aplicaciones

Antonio Cabrera y Alejandro Gómez

Trabajo para el doble grado de
Ingeniería del Software y Matemática Computacional



Asignatura de topología
U-tad
España
Noviembre 2023

Contents

1	Introducción	3
1.1	¿Qué es un fractal?	3
1.2	Autosimilitud	3
1.2.1	Autosimilitud exacta	3
1.2.2	Cuasiautosimilitud	4
1.3	Definición más formal	5
1.4	Dimensión fractal	5
1.4.1	Cálculo de la dimensión fractal	5
2	Historia	11
2.1	Antecedentes	11
2.2	Primeros fractales	11
2.3	Nuevas matemáticas	12
2.4	Mandelbrot	13
3	L-Systems	14
3.1	¿Qué son los L-Systems?	14
3.2	¿Cómo funcionan?	14
3.3	Ejemplos	15
3.3.1	Algas	15
3.3.2	Copos de nieve	16
3.4	Dibujos de plantas con L-Systems	18
3.4.1	Planta 1	18
3.4.2	Planta 2	19
3.4.3	Planta 3	19
4	Ejemplos de fractales	20
4.1	Copo de Koch	20
4.1.1	Construcción	20
4.1.2	Propiedades	21
4.2	Set de Mandelbrot	25
4.2.1	Construcción	26
4.2.2	Propiedades	27
4.3	Fractales basados en simulaciones físicas	27

4.3.1	Fractal a partir de imanes	27
4.3.2	Fractal a partir de pelotas rebotando en una función	29
4.4	Fractales de papel	32
5	Aplicaciones	34
5.1	Fractales en la naturaleza	34
5.2	Aplicaciones en la biología	34
5.3	Aplicaciones en comunicaciones	35
5.4	Aplicaciones en la geología	36
5.5	Otras aplicaciones	37
A	Código para la visualización del Box Counting	39
A.1	Visualización de la recta y el cuadrado	39
A.2	Visualización del triángulo de Sierpinski	40
A.3	Visualización de Box Counting para el triángulo de Sierpinski	41

List of Figures

1.1	Triángulo de Sierpinski	4
1.2	Copo de Koch	4
1.3	Ejemplos de cuasiautosimilitud en el conjunto de Mandelbrot	5
1.4	Box Counting - Cajas de lado $\frac{1}{2}$	6
1.5	Box Counting - Cajas de lado $\frac{1}{3}$	6
1.6	Box Counting - Cajas de lado $\frac{1}{n}$	7
1.7	Box Counting - Triángulo de Sierpinski - Cajas de lado $\frac{1}{2}$	8
1.8	Box Counting - Triángulo de Sierpinski - Cajas de lado $\frac{1}{4}$	8
1.9	Box Counting - Triángulo de Sierpinski - Cajas de lado $\frac{1}{8}$	9
1.10	Box Counting - Triángulo de Sierpinski - Cajas de lado $\frac{1}{16}$	9
2.1	Función de Weierstrass	11
2.2	Alfombra de Sierspinsky	12
2.3	Curva de Peano	12
2.4	Conjunto de Mandelbrot	13
3.1	Curva de Koch con ángulo de 90 grados - 0 iteraciones	16
3.2	Curva de Koch con ángulo de 90 grados - 1 iteración	17
3.3	Curva de Koch con ángulo de 90 grados - 2 iteraciones	17
3.4	Curva de Koch con ángulo de 90 grados - 3 iteraciones	17
3.5	Planta 1	18
3.6	Planta 2	19
3.7	Planta 3	19
4.1	Iteración cero de la curva de Koch.	20
4.2	Primera iteración de la curva de Koch.	20
4.3	Segunda iteración de la curva de Koch.	21
4.4	Autosilitud exacta del copo de Koch.	21
4.5	Box counting del copo de Koch.	22
4.6	Iteración cero del copo de Koch.	23
4.7	Primera iteración del copo de Koch.	23
4.8	Segunda iteración del copo de Koch.	24
4.9	Iteración n del copo de Koch.	24
4.10	Copo de Koch dentro de un círculo.	25

4.11	Primera imagen publicada del set de Mandelbrot, por Robert W. Brooks y Peter Matelski.	26
4.12	Tres imanes y una bola de acero.	28
4.13	Fractal generado a partir de tres imanes y una bola de acero.	28
4.14	Zoom en el fractal generado a partir de tres imanes y una bola de acero.	29
4.15	Visualización de los botes de la pelota (1,1)	29
4.16	Fractal generado a partir de $f(x) = x^4 - x^2$ y una pelota que rebota en ella.	30
4.17	Zoom a una de las elipses del fractal generado a partir de $f(x) = x^4 - x^2$ y una pelota que rebota en ella.	31
4.18	Zoom a una de las patas del fractal generado a partir de $f(x) = x^4 - x^2$ y una pelota que rebota en ella.	31
4.19	Triángulo de Sierpinsky de papel.	32
4.20	Conjunto de Cantor de papel.	32
4.21	Curva de dragón de papel.	33
5.1	Estructura fractal de los pulmones.	35
5.2	Imagen de como el ADN se compacta en una estructura fractal	35
5.3	Ejemplo de compresión fractal	36
5.4	Lago Nasser, Egipto	36
5.5	Fractal financiero	37
5.6	Fractal que genera una montaña	38

Chapter 1

Introducción

1.1 ¿Qué es un fractal?

La definición más inmediata que tenemos de los fractales es la siguiente:

Definición 1.1.1. *Un fractal es un objeto geométrico cuya estructura básica, fragmentada o irregular, se repite a diferentes escalas. [?]*

Que su estructura básica se repita a diferentes escalas significa que el objeto es autosimilar.

1.2 Autosimilitud

Benoît Mandelbrot la definió como sigue:

Definición 1.2.1. *Un objeto es autosimilar o autosemejante si sus partes tienen la misma forma o estructura que el todo, aunque pueden presentarse a diferente escala y pueden estar ligeramente deformadas. [?]*

Vamos a ver dos tipos de autosimilitud:

1.2.1 Autosimilitud exacta

Definición 1.2.2. *Un objeto es exactamente autosimilar si es exactamente igual a sí mismo a diferentes escalas.. [?]*

Es la más restrictiva de todas y la que vemos en los fractales clásicos.
Algunos ejemplos de objetos exactamente autosimilares son:

- El triángulo de Sierpinski

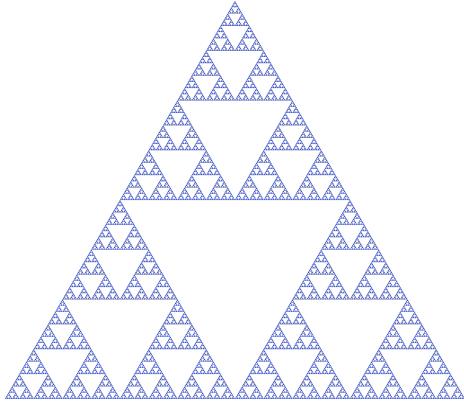


Figure 1.1: Triángulo de Sierpinski

- El copo de Koch

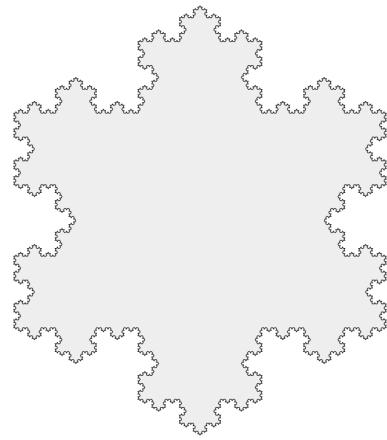


Figure 1.2: Copo de Koch

1.2.2 Cuasautosimilitud

Definición 1.2.3. *Un objeto es cuasautosimilar si es aproximadamente igual a sí mismo a diferentes escalas. [?]*

Los fractales de este tipo contienen copias menores y distorsionadas de si mismos, como ocurre por ejemplo con el conjunto de Mandelbrot.

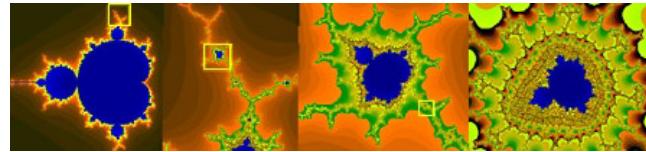


Figure 1.3: Ejemplos de cuasautosimilitud en el conjunto de Mandelbrot

1.3 Definición más formal

En 1982 Benoît Mandelbrot definió los fractales de la siguiente forma:

Definición 1.3.1. *Un fractal es un conjunto cuya dimensión de Hausdorff-Besicovitch es estrictamente mayor que su dimensión topológica.* [?]

La dimensión topológica es la dimensión que todos conocemos, la dimensión de Hausdorff-Besicovitch es una de las formas de calcular la dimensión fractal de un objeto.

1.4 Dimensión fractal

La dimensión fractal es una medida de la complejidad de un objeto fractal. La dimensión topológica de un objeto es un número entero, mientras que la dimensión fractal es un número real. La dimensión fractal es una generalización de la dimensión topológica.

1.4.1 Cálculo de la dimensión fractal

Ya hemos visto que una de las formas de calcular la dimensión fractal de un objeto es la dimensión de Hausdorff-Besicovitch, sin embargo esta resulta algo compleja. Otra forma más sencilla es con el método de *Box Counting* también llamado o también conocido como la dimensión de Minkowski-Bouligard.

Para entender mejor este método vamos a verlo con un ejemplo. Consideraremos un segmento y el borde de un cuadrado ambos de lado 1. [?]

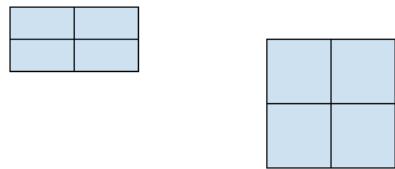


Figure 1.4: Box Counting - Cajas de lado $\frac{1}{2}$

Si intentamos cubrir ambos objetos con cajas de lado $\frac{1}{2}$ vemos que para el segmento necesitamos 2 y para el cuadrado 2^2

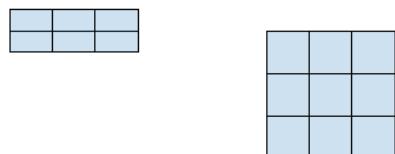


Figure 1.5: Box Counting - Cajas de lado $\frac{1}{3}$

Si utilizamos cajas más pequeñas, de lado $\frac{1}{3}$, vemos que para el segmento necesitamos 3 y para el cuadrado 3^2



Figure 1.6: Box Counting - Cajas de lado $\frac{1}{n}$

Si continuamos con este proceso vemos que para el segmento necesitamos n cajas de lado $\frac{1}{n}$ y para el cuadrado n^2

Vemos que el exponente es un buen indicador de la dimensión de los objetos. Si definimos $N(n)$ como el número de cajas de lado $\frac{1}{n}$ necesarias para cubrir el objeto, entonces tenemos que:

$$N(n) = n^d \quad (1.1)$$

Sacando logaritmos a ambos lados tenemos que:

$$\log N(n) = \log n^d \iff \log N(n) = d \log n \iff d = \frac{\log N(n)}{\log n} \quad (1.2)$$

Definición 1.4.1. La dimensión de Minkowski-Bouligard de un conjunto A es el límite de la expresión $[\dots]$

$$\lim_{\epsilon \rightarrow 0} \frac{\log N(\epsilon)}{\log \frac{1}{\epsilon}} \quad (1.3)$$

donde $N(\epsilon)$ es el mínimo de bolas de radio ϵ necesarias para recubrir el conjunto.

Ahora vamos a ver que ocurre cuando intentamos calcular la dimensión de Minkowski-Bouligard de un fractal, como por ejemplo el triángulo de Sierpinski de base 1.

Si intentamos cubrir el triángulo con cajas de lado $\frac{1}{2}$ vemos que necesitamos 4 cajas.

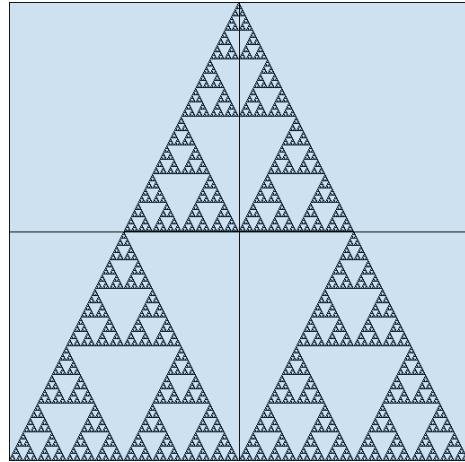


Figure 1.7: Box Counting - Triángulo de Sierpinski - Cajas de lado $\frac{1}{2}$

Si utilizamos cajas más pequeñas, de lado $\frac{1}{4}$, necesitaremos 12 cajas.

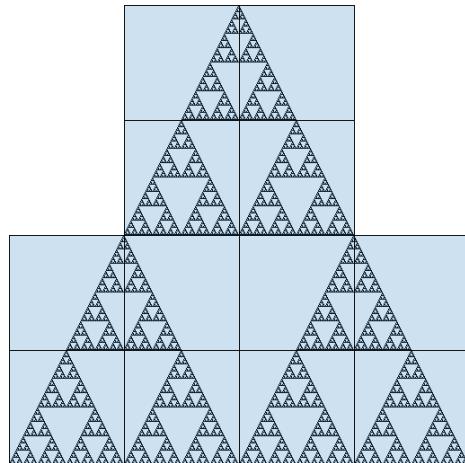


Figure 1.8: Box Counting - Triángulo de Sierpinski - Cajas de lado $\frac{1}{4}$

Con cajas de lado $\frac{1}{8}$ necesitaremos 36 cajas.

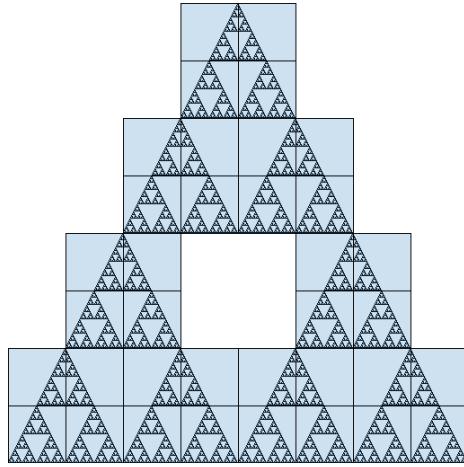


Figure 1.9: Box Counting - Triángulo de Sierpinski - Cajas de lado $\frac{1}{8}$

Y con cajas de lado $\frac{1}{16}$ necesitaremos 108 cajas.

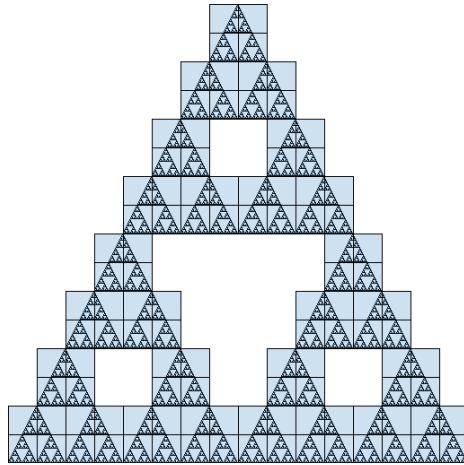


Figure 1.10: Box Counting - Triángulo de Sierpinski - Cajas de lado $\frac{1}{16}$

Si continuamos con este proceso vemos que para el triángulo de Sierpinski necesitamos $4 \cdot 3^{n-1}$ cajas de lado $\frac{1}{2^n}$. Si definimos $N(n)$ como el número de cajas de lado $\frac{1}{2^n}$ necesarias para cubrir el objeto, entonces tenemos que:

$$\begin{aligned}
 d &= \lim_{n \rightarrow \infty} \frac{\log(N(n))}{\log(2^n)} = \lim_{n \rightarrow \infty} \frac{4 \cdot 3^{n-1}}{2^n} \approx \\
 &\approx \lim_{n \rightarrow \infty} \frac{3^n}{2^n} = \frac{\log 3}{\log 2} \approx 1.585
 \end{aligned} \tag{1.4}$$

Como ya hemos comentado antes, nos sale un número real. Nuestra intuición nos sugiere que el triángulo de Sierpinski es "*más que una curva*" pero "*menos que una superficie*". [?]

Chapter 2

Historia

2.1 Antecedentes

Las primeras formas fractales aparecieron en el siglo XIX, cuando el matemático Karl Weierstrass graficó en 1872 su famosa función de Weierstrass. Más tarde en ese mismo siglo, empezaron a surgir conceptos cada vez más cercanos a lo que hoy se consideran fractales, siendo más geométricos y menos algebraicos.[?]

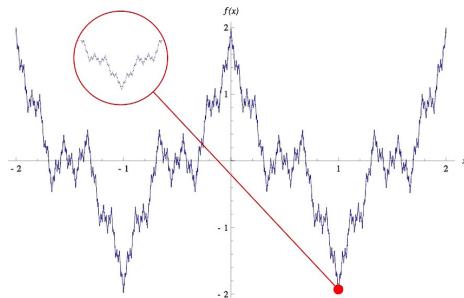


Figure 2.1: Función de Weierstrass

2.2 Primeros fractales

Así, en 1904, Helge von Koch definió su copo de nieve, una curva con propiedades similares a la de Weierstrass. En 1915, Waclaw Sierpinski construyó su triángulo y, un año después, su alfombra. [?]

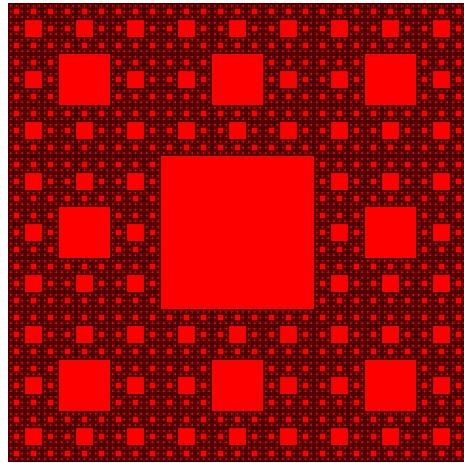


Figure 2.2: Alfombra de Sierspinsky

2.3 Nuevas matemáticas

En el siglo XIX, Georg Cantor desarrolló su conjunto de Cantor y Giuseppe Peano su curva de Peano. Gracias a esto se dieron cuenta de que estos objetos matemáticos no eran casos particulares, si no que había un gran conjunto de funciones que compartían las mismas cualidades que la función de Weierstrass. [?]

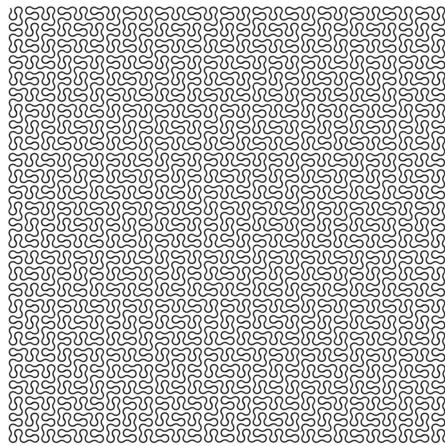


Figure 2.3: Curva de Peano

A pesar de que hubiese un gran conjunto de funciones, la geometría de Euclides y la dinámica Newtoniana no podían describirlas bien. En 1919, el matemático Félix Hausdorff introdujo la primera manera de observar y estudiar estos objetos, la dimensión de Hausdorff-Besicovitch. Años más tarde, Andrei Kolmogorov describió una herramienta similar conocida como la entropía de Kolmogorov. [?]

En 1963, Benoît Mandelbrot empezó a trabajar con los fractales a raíz de otra investigación, lo que le llevó a fundar la geometría fractal. [?]

2.4 Mandelbrot

Mandelbrot es uno de los mayores exponentes en el avance del campo de los fractales, desde identificar una muestra de "*tiempo fractal*" hasta replantearse un problema mundialmente conocido: ¿Cuál es la longitud de la costa de Gran Bretaña? Según Mandelbrot esta distancia es infinita, o mejor dicho, depende de la longitud de la regla con la que se mida. Como ya hemos comentado, la geometría euclídea no es capaz de describir estos objetos, por lo que Mandelbrot utilizó la noción de dimensión, y en concreto, la extraña definición de dimensión fraccionarias. [?]

En 1975, Mandelbrot acuñó el término fractal, que proviene del latín *fractus*, que significa roto o fracturado. El fractal más famoso de Mandelbrot es el conjunto de Mandelbrot, el cual es el más estudiado dentro del campo. [?]

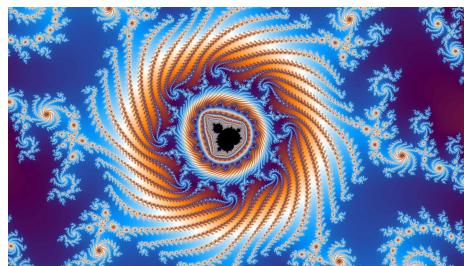


Figure 2.4: Conjunto de Mandelbrot

Chapter 3

L-Systems

3.1 ¿Qué son los L-Systems?

Un sistema-L o también denominados sistemas de Lindenmayer se trata de un conjunto de reglas y símbolos usados principalmente para modelar el proceso de crecimiento de las plantas. [?]

Fueron introducidos en 1968 por Aristid Lindenmayer, el cual estudió los patrones de crecimiento de varias algas y tenían como objetivo realizar una descripción formal del desarrollo de organismos y representar la relación entre células de plantas. [?]

3.2 ¿Cómo funcionan?

Las reglas de estos sistemas se basan en la autosimilitud, por eso las figuras que se suelen modelar suelen representar formas de tipo fractal. [?]

Los sistemas-L se definen como un conjunto $G = \{V, S, \omega, P\}$ [?]

- V es un conjunto de símbolos que contiene elementos que pueden ser reemplazados.
- S es un conjunto de símbolos que contiene elementos que se mantienen fijos.
- ω es una cadena de símbolos de V que definen el estado inicial del sistema (inicio o axioma).

- P es un conjunto de reglas o producciones que definen la forma en la que las variables pueden ser reemplazadas por combinaciones de constantes y otras variables.

Todos los sistemas-L parten de un estado inicial, y se dice que el sistema es libre de contexto si cada iteración se refiere sólo a un símbolo individual y no a sus vecinos. Cuando la aplicación de una regla depende también de sus vecinos, se dice que el sistema-L es sensitivo al contexto. [?]

3.3 Ejemplos

3.3.1 Algas

Pongamos como ejemplo de sistema-L el primer caso que quiso estudiar Lindenmayer, las algas. [?]

Datos:

- Variables: A, B
- Constantes: ninguna
- Incio: A
- Reglas: $A \rightarrow AB, B \rightarrow A$

Resultado:

- Iteración 0: A
- Iteración 1: AB
- Iteración 2: ABA
- Iteración 3: $ABAAB$
- Iteración 4: $ABAABABA$
- Iteración 5: $ABAABABAABAAB$

3.3.2 Copos de nieve

Los sistemas también pueden ser representados con dibujos, el siguiente ejemplo se refiere a la variante de la curva de Koch con ángulo de 90: [?]

Sintaxis:

$$\begin{aligned} F &\rightarrow \text{Dibujar hacia adelante} \\ + &\rightarrow \text{Girar a la izquierda 90 grados} \\ - &\rightarrow \text{Girar a la derecha 90 grados} \end{aligned} \tag{3.1}$$

Datos:

- Variables: F
- Constantes: $+, -$
- Incio: F
- Reglas: $F \rightarrow F + F - F - F + F$

Resultado:

Para $n = 0$ tenemos F :



Figure 3.1: Curva de Koch con ángulo de 90 grados - 0 iteraciones

Para $n = 1$ tenemos $F + F - F - F + F$:

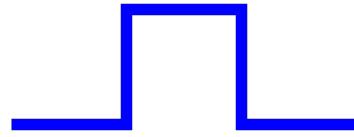


Figure 3.2: Curva de Koch con ángulo de 90 grados - 1 iteración

Para $n = 2$ tenemos $F + F - F - F + F + F + F - F - F + F - F + F - F + F - F - F - F + F - F + F$:

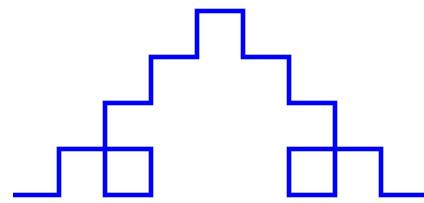


Figure 3.3: Curva de Koch con ángulo de 90 grados - 2 iteraciones

Para $n = 3$ la secuencia es demasiado larga para ponerla aquí, pero el resultado es el siguiente:

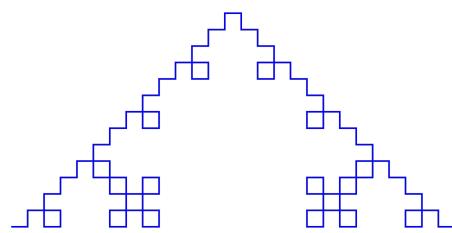


Figure 3.4: Curva de Koch con ángulo de 90 grados - 3 iteraciones

3.4 Dibujos de plantas con L-Systems

3.4.1 Planta 1

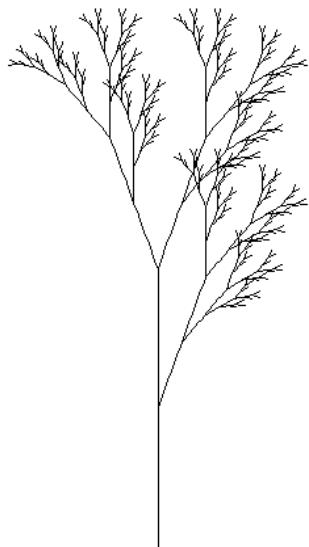


Figure 3.5: Planta 1
[?]

3.4.2 Planta 2

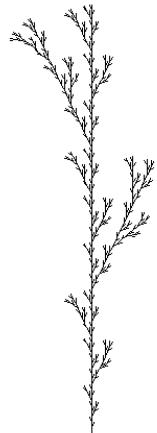


Figure 3.6: Planta 2
[?]

3.4.3 Planta 3

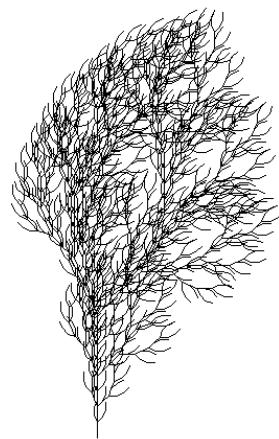


Figure 3.7: Planta 3
[?]

Chapter 4

Ejemplos de fractales

4.1 Copo de Koch

4.1.1 Construcción

Comenzamos con un segmento de longitud 1.

Figure 4.1: Iteración cero de la curva de Koch.

En la primera iteración, dividimos el segmento en tres partes iguales y reemplazamos la parte central por dos segmentos de la misma longitud, formando un ángulo de 60° . [?]

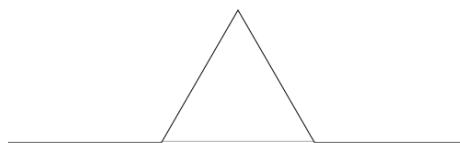


Figure 4.2: Primera iteración de la curva de Koch.

Para siguientes iteraciones, repetimos el mismo proceso para todos los segmentos. [?]

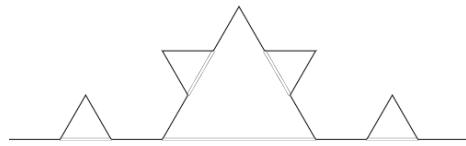


Figure 4.3: Segunda iteración de la curva de Koch.

4.1.2 Propiedades

El copo de Koch es un objeto autosimilar, con autosimilitud exacta.
[?]

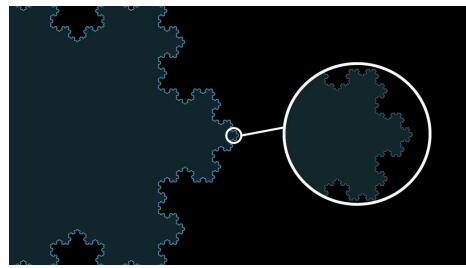


Figure 4.4: Autosilitud exacta del copo de Koch.
[?]

La dimensión fractal del copo de Koch es $\frac{\log 4}{\log 3} \approx 1.26186$ [?]

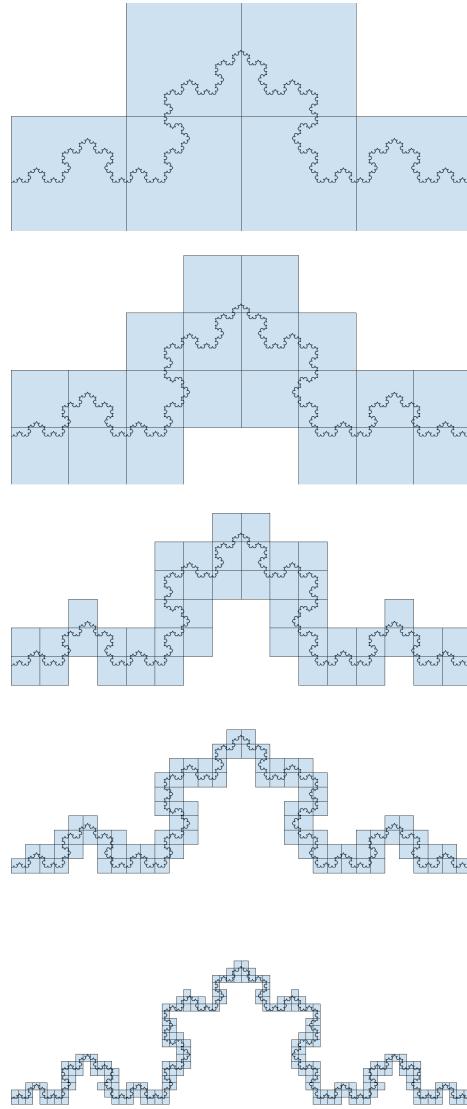


Figure 4.5: Box counting del copo de Koch.

Teorema 1. *El copo de Koch tiene longitud infinita, pero área finita.*
 [?]

Proof. Primero definamos $N(n)$ como el número de lados del polígono en la iteración n y $L(n)$ como la longitud de cada lado del polígono en la iteración n .

En el inicio, $N(0) = 3$ y $L(0) = 3$

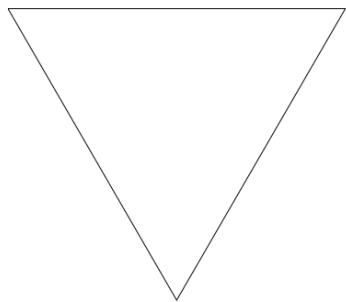


Figure 4.6: Iteración cero del copo de Koch.

En la primera iteración, $N(1) = 3 \cdot 4$ y $L(1) = 3 \cdot \frac{4}{3}$

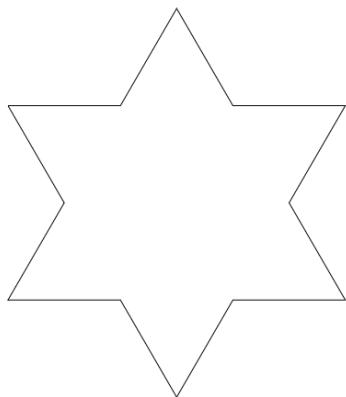


Figure 4.7: Primera iteración del copo de Koch.

En la segunda iteración, $N(2) = 3 \cdot 4^2$ y $L(2) = 3 \cdot \left(\frac{4}{3}\right)^2$

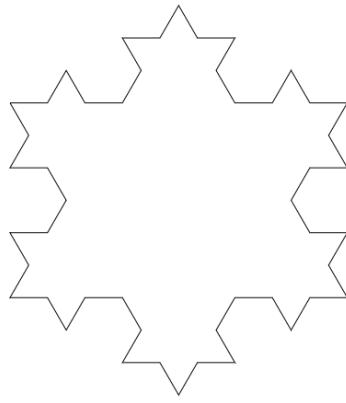


Figure 4.8: Segunda iteración del copo de Koch.

En la iteración n , $N(n) = 3 \cdot 4^n$ y $L(n) = 3 \cdot \left(\frac{4}{3}\right)^n$

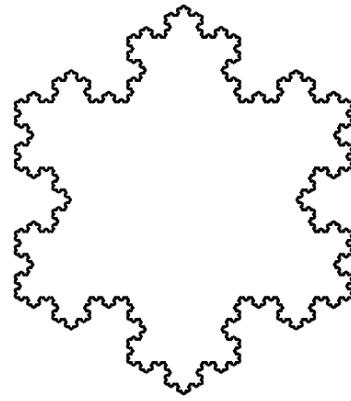


Figure 4.9: Iteración n del copo de Koch.

Viendo esto es facil ver que cuando $n \rightarrow \infty$, $L(n) \rightarrow \infty$

$$\lim_{n \rightarrow \infty} L(n) = \lim_{n \rightarrow \infty} 3 \cdot \left(\frac{4}{3}\right)^n = \infty \quad (4.1)$$

El área claramente es finita ya que podemos contener el copo en un

círculo. [?]

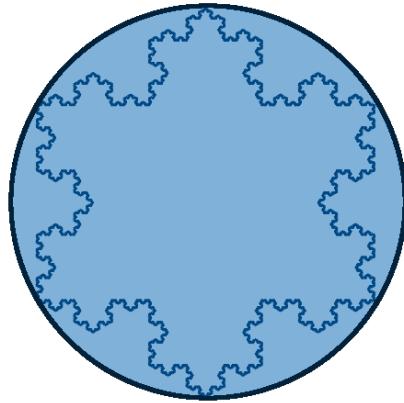


Figure 4.10: Copo de Koch dentro de un círculo.

□

4.2 Set de Mandelbrot

El set de Mandelbrot es un conjunto bidimensional que con una definición muy simple produce estructuras complejas y hermosas.

Aunque el nombre esté relacionado con el matemático Benoît Mandelbrot, el set fue definido por primera vez en 1978 por Robert W. Brooks y Peter Matelski. [?]

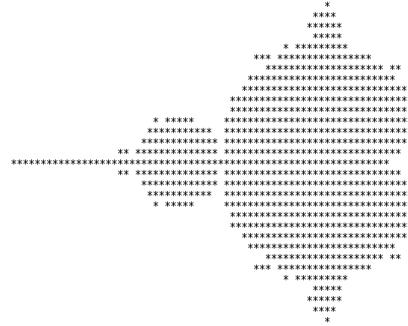


Figure 4.11: Primera imagen publicada del set de Mandelbrot, por Robert W. Brooks y Peter Matelski.

4.2.1 Construcción

Definición 4.2.1. *El conjunto de Mandelbrot es el conjunto de números complejos c para los cuales el método iterativo definido por $z_0 = 0$ y $z_{n+1} = z_n^2 + c$ no tiende al infinito, es decir, no es divergente. [?]*

En las representaciones gráficas se suele mostrar en negro los puntos que no divergen (los que pertenecen al conjunto). El resto de puntos se colorean en función de cuantas iteraciones tardan en exceder un límite en el que ya se supone que divergen.

Para ver si excede el límite podemos usar el módulo del número complejo, el cual representa la distancia de este con el origen. Por ejemplo, si nuestro fractal está definido en $[-1, 1] \times [-1, 1]$, entonces diremos que $\text{mod}(z) > 2 \implies$ divergente. Si completa las 1000 iteraciones, asumimos que converge y lo pintamos de negro.

```

1 def mandelbrot(x, y):
2     c = complex(x, y)
3     z = 0
4
5     for i in range(1, 1000):
6         if abs(z) > 2:
7             return rgb_conv(i)
8         z = z * z + c
9     return (0, 0, 0)
```

4.2.2 Propiedades

Teorema 2. *El conjunto de Mandelbrot es compacto. [?]*

Lema 1. *Un subconjunto cerrado de un compacto es compacto.*

Lema 2. *Un círculo es compacto.*

Lema 3. *El conjunto de Mandelbrot es cerrado.*

Proof. Como el conjunto de Mandelbrot es un conjunto cerrado y es un subconjunto del círculo de radio 2, entonces también es compacto.

□

Teorema 3. *El conjunto de Mandelbrot es conexo. [?]*

En 2001, Jeremy Kahn demostró que efectivamente el conjunto es conexo.

4.3 Fractales basados en simulaciones físicas

Algunos fractales pueden aparecer como resultado de simulaciones físicas. Esto se debe a que estas simulaciones a menudo producen mucho caos (un pequeño cambio en las condiciones inciales produce resultados dispares), y los fractales están muy relacionados con la teoría del caos.[?]

4.3.1 Fractal a partir de imanes

Podemos generar un fractal a partir de tres imanes y una bola de acero colgada de un hilo. Cada pixel representará la posición inicial de la bola y el color (rojo, amarillo o azul) representará el imán en el que termina.

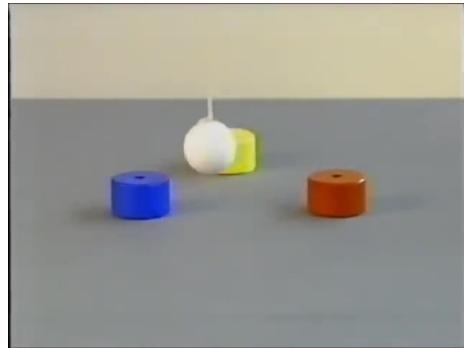


Figure 4.12: Tres imanes y una bola de acero.

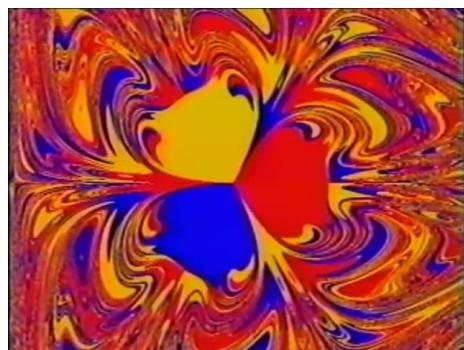


Figure 4.13: Fractal generado a partir de tres imanes y una bola de acero.

En cada línea hay otra de diferente color, por lo que podemos seguir haciendo zoom a priori de forma infinita.

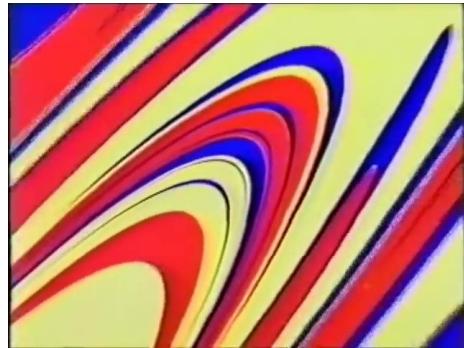


Figure 4.14: Zoom en el fractal generado a partir de tres imanes y una bola de acero.

4.3.2 Fractal a partir de pelotas rebotando en una función

Podemos generar un fractal a partir de una función y una pelota que rebota en ella [?]. Cada pixel representará la posición inicial de la pelota y el color del pixel será cuantos botes tarda en llegar al lado contrario del que empezó. Si la pelota rebota más de 1000 veces suponemos que se queda en bucle y pintamos el pixel de negro

Hemos utilizado la función $f(x) = x^4 - x^2$, pero se podrían usar otras funciones.

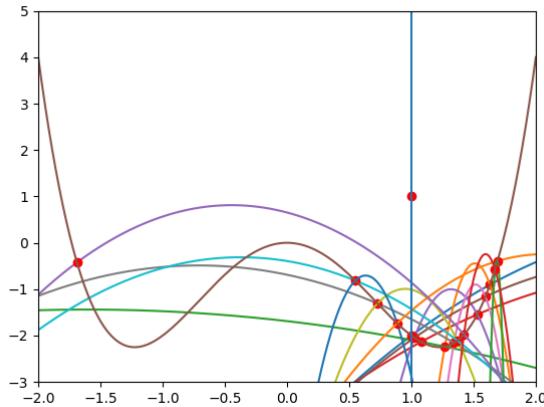


Figure 4.15: Visualización de los botes de la pelota $(1, 1)$

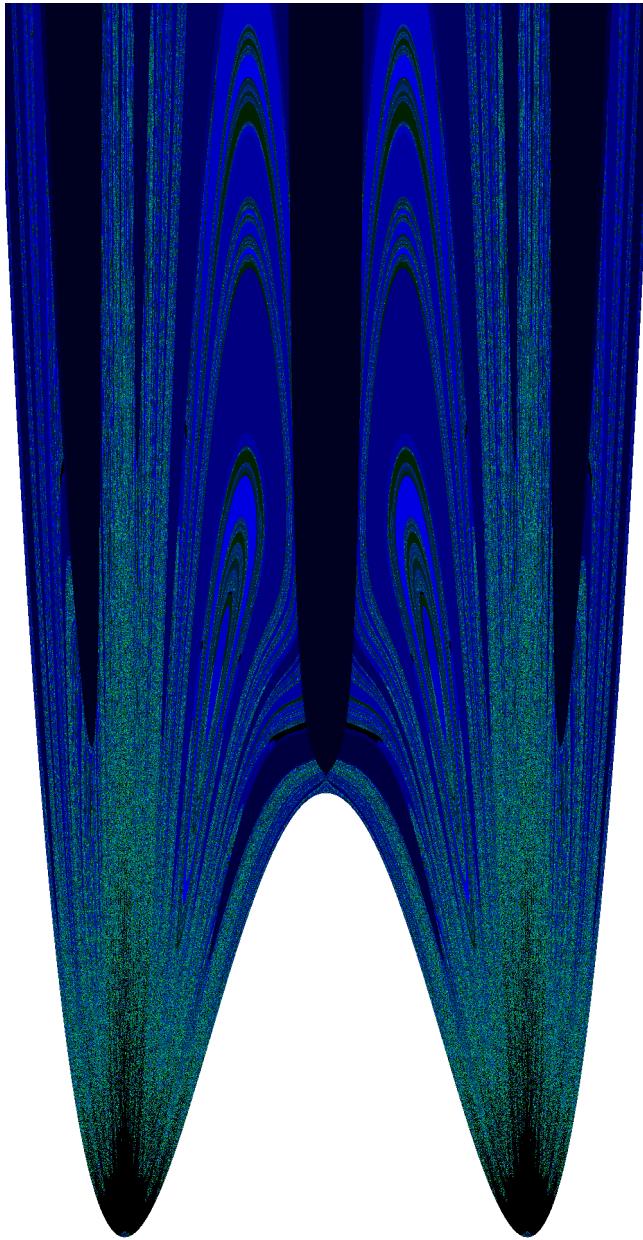


Figure 4.16: Fractal generado a partir de $f(x) = x^4 - x^2$ y una pelota que rebota en ella.

En las elipses del objeto se puede ver una autosimilitud aproximada.

Se parece un poco al fractal magnético.

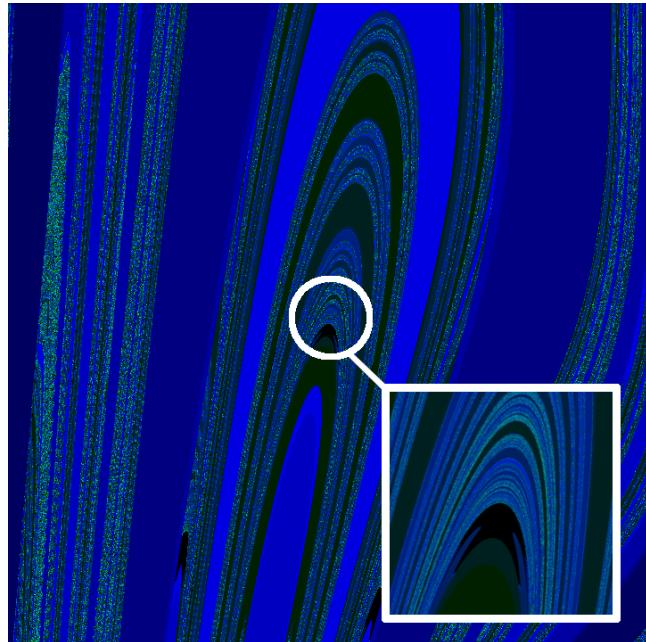


Figure 4.17: Zoom a una de las elipses del fractal generado a partir de $f(x) = x^4 - x^2$ y una pelota que rebota en ella.

Hay otras formas interesantes, aunque no tienen autosimilitud.

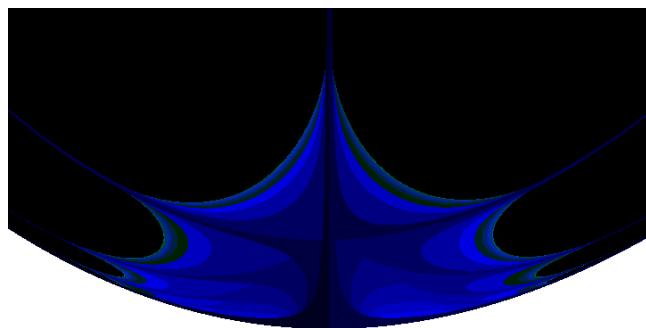


Figure 4.18: Zoom a una de las patas del fractal generado a partir de $f(x) = x^4 - x^2$ y una pelota que rebota en ella.

4.4 Fractales de papel

Los fractales de papel son una forma creativa de crear fractales con nuestras propias manos. A través de pliegues y cortes precisos en el papel podemos manufacturarlos.



Figure 4.19: Triángulo de Sierpinsky de papel.

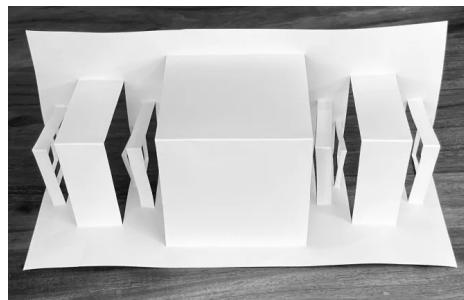


Figure 4.20: Conjunto de Cantor de papel.

Otros fractales los podemos hacer sin cortes, simplemente con tiras de papel.

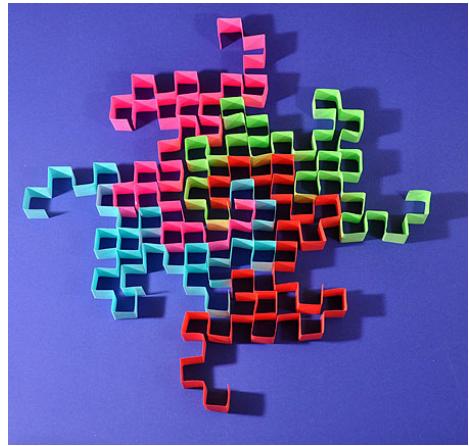


Figure 4.21: Curva de dragón de papel.

Chapter 5

Aplicaciones

Se han podido aprovechar los fractales áreas como: Matemáticas, Física, Química, Comunicaciones, Informática, Geología, Economía, Música, Biología, etc. . .

5.1 Fractales en la naturaleza

Los fractales se pueden encontrar en la naturaleza, por ejemplo, en la forma de las hojas, las ramas de los árboles, las venas de las hojas, las costas, las montañas, los copos de nieve, los relámpagos, los cristales, etc.

Si conseguimos identificar una estructura fractal en un fenómeno natural, podemos estudiar de forma matemática su comportamiento a futuro, permitiendo explicar procesos complejos con fórmulas relativamente simples (como ya vimos en la generación con L-systems)

5.2 Aplicaciones en la biología

Los fractales, además de ser usados en el estudio del crecimiento de las plantas, son usados en:

- El estudio de problemas pulmonares, ya que los pulmones presentan una estructura fractal para repartir el aire de forma homogénea.[?]
]

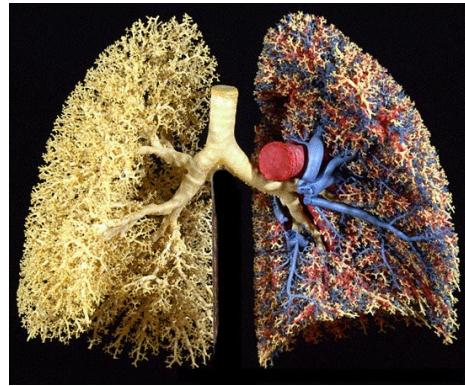


Figure 5.1: Estructura fractal de los pulmones.
[?]

- La evolución de especies depredadoras en entornos controlados pueden representar patrones fractales. [?]
- El estudio del ADN y los enlaces entre nucleótidos. [?].

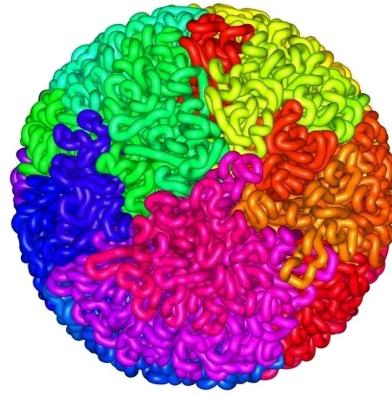


Figure 5.2: Imagen de como el ADN se compacta en una estructura fractal

5.3 Aplicaciones en comunicaciones

En comunicaciones e informática se experimenta con la compresión fractal de archivos para que ocupen menos espacio.[?]



Figure 5.3: Ejemplo de compresión fractal

También se ha experimentado con las redes de ordenadores con estructura fractal para comprobar si generan mejor rendimiento. [?]

5.4 Aplicaciones en la geología

En la geología, se usan para estudiar las irregularidades de la costa y relieves del terrenos, además de la distribución de las islas y continentes.[?]



Figure 5.4: Lago Nasser, Egipto

Los terremotos tienen una relación fractal entre la intensidad del terremoto y la frecuencia del suceso. [?]

5.5 Otras aplicaciones

Los fractales también se usan en:

- La ecología, para cuantificar los niveles de CO₂ que procesan los bosques. [?]
- La astrofísica, para estudiar la formación de estrellas.[?]
- La economía, para estudiar el comportamiento de la bolsa.[?]

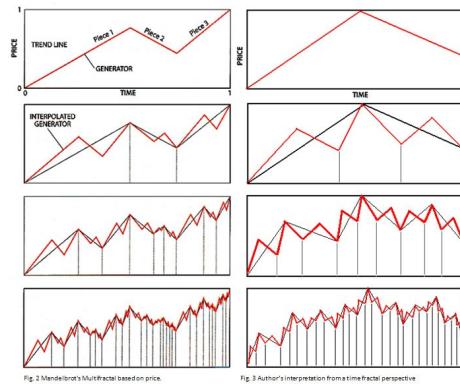


Figure 5.5: Fractal financiero

- Gráficos por ordenador, para generar texturas y paisajes.

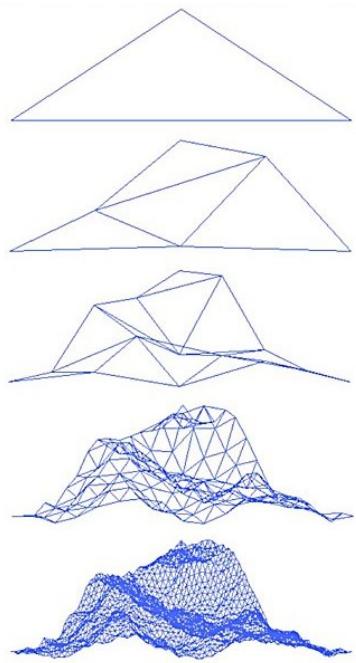


Figure 5.6: Fractal que genera una montaña

Appendix A

Código para la visualización del Box Counting

A.1 Visualización de la recta y el cuadrado

```
1 from PIL import Image, ImageDraw
2
3 def generar_imagen(n, path):
4
5     A = (100,100)
6     B = (200,100)
7
8     a = (300,100)
9     b = (400,100)
10    c = (300,200)
11    d = (400,200)
12
13    img = Image.new('RGB', (500,300), color = 'white')
14
15    # Segmento AB
16
17    draw = ImageDraw.Draw(img, 'RGBA')
18    draw.line((A,B)*100, fill='black', width=1)
19
20    # Cuadrado abcd
21
22    draw.line((a,b), fill='black', width=1)
23    draw.line((b,d), fill='black', width=1)
24    draw.line((d,c), fill='black', width=1)
25    draw.line((c,a), fill='black', width=1)
26
27    # Cajas del segmento AB
28
29    for i in range(0,n):
30        a = (100+i*(1/n)*100,100+100*i/(2*n))
31        b = (100+i*(1/n)*100,100-100*i/(2*n))
32        c = (100+(i*100+100)*(1/n),100-100*i/(2*n))
```

```

33         d = (100+(i*100+100)*(1/n),100+100*1/(2*n))
35             draw.polygon([a,b,c,d], fill=(0,100,180,50), outline='black', )
37     # Cajas del cuadrado abcd
39     for i in range(0,n):
40         for k in range(0,n):
41             a = (300+k*1/n*100,100+i*1/n*100)
42             b = (300+k*1/n*100,100+(i+1)*1/n*100)
43             c = (300+(k+1)*1/n*100,100+(i+1)*1/n*100)
44             d = (300+(k+1)*1/n*100,100+i*1/n*100)
45
46             draw.polygon([a,b,c,d], fill=(0,100,180,50), outline='black'
47             ')
48
49     img.save(path)
50
51 if __name__ == "__main__":
52     N = (2,3,16)
53
54     for i in N:
55         generar_imagen(i, 'boxcounting-' + str(N.index(i) + 1) + '.png')
56

```

A.2 Visualización del triángulo de Sierpinski

Créditos a [umnikos](#) por el código.

```

from PIL import Image, ImageDraw
2 import numpy as np
3
4 size = (512, 512) # the size of the final image
# create a new black and white image
5 im = Image.new("1", size, 1)
px = im.load() # create a pixel access object
6
7 # takes a 4-tuple range of where to draw the sierpinski triangle
8 def triangle(xy, fill=0, depth=5):
# if recursion is over, fill the top left pixel
9     if depth == 0:
10         px[xy[0], xy[1]] = fill
11         return
12     # otherwise, draw 3 smaller sierpinski triangles
13     triangle(
14         (
15             (3 * xy[0] + xy[2]) // 4,
16             xy[1],
17             (xy[0] + 3 * xy[2]) // 4,
18             (xy[1] + xy[3]) // 2,
19         ),
20         fill,
21         depth - 1,
22     )
23
24

```

```

26     )
27     triangle(
28         (
29             xy[0],
30             (xy[1] + xy[3]) // 2,
31             (xy[0] + xy[2]) // 2,
32             xy[3],
33         ),
34         fill,
35         depth - 1,
36     )
37     triangle(
38         (
39             (xy[0] + xy[2]) // 2,
40             (xy[1] + xy[3]) // 2,
41             xy[2],
42             xy[3],
43         ),
44         fill,
45         depth - 1,
46     )
47
48 # draw the triangle (smaller depths results in a dotty output)
49 triangle((0, 0, size[0], size[1]), depth=10)
50 im.save("sierspinsky.png", "PNG") # save the image

```

A.3 Visualización de Box Counting para el triángulo de Sierpinski

```

1 import numpy as np
2 from PIL import Image, ImageDraw
3
4 triangle = Image.open('sierspinsky.png')
5
6 # Cargar imagen como matriz de pixeles
7 img_array = np.array(triangle)
8
9 def check_pixel_square(top_point, size):
10     for i in range(0, size):
11         for k in range(0, size):
12             if img_array[top_point[1]+ i][top_point[0] + k] == False:
13                 return True
14     return False
15
16 def generate_image(n, path):
17     img = Image.new('RGB', (512,512), color = 'white')
18     img.paste(triangle, (0,0))
19
20     draw = ImageDraw.Draw(img, 'RGBA')

```

```

22     size = int(512 / n)
24
26     num_cajas = 0
28
29     for i in range(0, n):
30         for k in range(0, n):
31             if check_pixel_square((i*size,k*size), size):
32                 draw.rectangle([(i*size,k*size),((i+1)*size,(k+1)*size)], fill=(0,100,180,50), outline='black', width=1)
33                 num_cajas += 1
34
35     img.save(path)
36     print(num_cajas)
37
38 if __name__ == "__main__":
39
40     N = (2,4,8,16)
41
42     for i in N:
43         generate_image(i, 'boxcounting-sierspinsky-' + str(N.index(i)) +
44                         1) + '.png')

```

Bibliography

- [1] Varios autores. Los fractales una aplicación de las matematicas en la imagen, 2006. URL: <https://www.repo-ciie.dfie.ipn.mx/pdf/1060.pdf>.
- [2] Lucas Damian Catalano. Compresión fractal de imágenes digitales, noviembre 2011. URL: <https://mate.dm.uba.ar/~umolter/materias/geo-fractal-2-11/apunte-lucas.pdf>.
- [3] Wikipedia contributors. Fractal. <https://es.wikipedia.org/wiki/Fractal>, noviembre 2023. Accessed: 29-11-2023.
- [4] Wikipedia contributors. Mandelbrot set. https://en.wikipedia.org/wiki/Mandelbrot_set, noviembre 2023. Accessed: 02-12-2023.
- [5] Dmytro Fedoriaka. L-systems — dmitriy fedoriaka's website. <https://fedimser.github.io/l-systems.html>, 2023.
- [6] Dory Gascueña. Aplicaciones de la geometría fractal: del cambio climático al cáncer, agosto 2020. URL: <https://www.bbvaopenmind.com/ciencia/matematicas/aplicaciones-la-geometria-fractal-del-cambio-climatico-al-cance>
- [7] Emil Alvarado González. Historia de los fractales y de su geometría, marzo 2017. URL: <https://complejidad.net/2017/03/21/historia-de-los-fractales-y-de-su-geometria/>.
- [8] Antonio Cabrera Landín. balls. <https://github.com/Pastafarista/balls>, octubre 2023.

- [9] Miguel Ángel Morales Medina. ¿qué es el conjunto de mandelbrot?: historia y construcción, noviembre 2011. URL: <https://www.gaussianos.com/%C2%BFque-es-el-conjunto-de-mandelbrot-historia-y-construccion/>.
- [10] Mates Mike. Dimensión fractal: El copo de nieve de koch. https://www.youtube.com/watch?v=eKY_1j9VrEA, 2022.
- [11] Hiro Shimoyama. The relationship between chaos, fractal and physics. <https://www.youtube.com/watch?v=C5Jkgvw-Z6E>, 2016.
- [12] Vanessa Atencia Toro. *Fractales matemáticos*. Tfg, Facultad de Matemáticas - Universidad de Barcelona, enero 2014. URL: <https://deposit.ub.edu/dspace/bitstream/2445/54228/1/memoria.pdf>.
- [13] Donald L. Turcotte. Fractals in geology: What are they and what are they good for?, enero 1991. URL: <https://rock.geosociety.org/net/gsatoday/archive/1/1/pdf/i1052-5173-1-1-sci.pdf>.
- [14] Wikipedia. Sistema-l — wikipedia, la enciclopedia libre, 2022. [Internet; descargado 20-febrero-2022]. URL: <https://es.wikipedia.org/w/index.php?title=Sistema-L&oldid=141801563>.
- [15] Wikipedia. Copo de nieve de koch — wikipedia, la enciclopedia libre, 2023. [Internet; descargado 15-octubre-2023]. URL: https://es.wikipedia.org/w/index.php?title=Copo_de_nieve_de_Koch&oldid=154637466.
- [16] Eduardo Sánchez Ortega y Laura Vargas-Parada. Nuevas formas para determinar fractales en el adn, julio 2018. URL: <https://www.c3.unam.mx/boletines/boletin26.html>.