

Tema 13

Fractales

En este capítulo se presenta el segundo proyecto de Física Computacional en C, dedicado a los Fractales. Si se ha escogido este proyecto, se deberán hacer los ejercicios propuestos en él.

13.1. Fractales: Teoría y Fenomenología

“Clouds are not spheres, mountains are not cones, coastlines are not circles, and bark is not smooth, nor does lightning travel in a straight line”

Con esta frase comenzaba Benoit Mandelbrot¹ en 1977 su ensayo sobre la geometría irregular de la naturaleza. En este ensayo, Mandelbrot ponía de manifiesto que muchas de las formas y patrones observados en la naturaleza son demasiado irregulares y complejos para poder ser medidos o descritos adecuadamente mediante la geometría euclídea clásica, que utiliza para el proceso de medida de un objeto patrones de comparación tan clásicos como la recta, el círculo, el cuadrado o polígonos/poliedros regulares. Mandelbrot propuso una geometría que permitió describir y caracterizar los objetos irregulares, no sólo naturales, sino también matemáticos, y que denominó *geometría fractal*. El término fractal viene de la raíz latina “fractus”, que significa “roto”, “fracturado” o “fragmentado”, propiedades que describen muy cualitativamente la irregularidad de esas formas pero que esconden importantes implicaciones matemáticas como la no derivabilidad en ningún punto.

Antes del desarrollo de la geometría fractal, la naturaleza era considerada como una geometría euclídea “ruidosa” o “distorsionada”. Por ejemplo, una montaña era básicamente descrita mediante un cono rugoso. Un ejemplo muy claro de este punto de vista puede ser atribuido a Paul Cezanne, que instruía a los jóvenes pintores diciendo: “todo en la naturaleza puede ser visto en términos de conos, cilindros y esferas”. De ahí viene la famosa réplica de Mandelbrot con la que hemos comenzado este capítulo.

El conjunto de Cantor, la curva de Koch, el triángulo de Sierpinski, la esponja de Menger, los conjuntos de Julia y Mandelbrot, los caminos producidos por movimientos brownianos, todos ellos construcciones geométricas que hoy en día son conocidos ejemplos de fractales (y que serán vistos en este curso), ya eran conocidos desde hacía tiempo. Sin embargo, el gran éxito de Benoit Mandelbrot fue el de organizar todas las ideas que habían hasta la fecha y proponer un marco geométrico general para

¹A menudo referido como “padre” de los fractales.

estos objetos, descubriendo que detrás de esta geometría se escondían importantes principios de organización en los fenómenos naturales. Por esta razón, Mandelbrot tituló su ensayo “La geometría fractal de la naturaleza”.²

13.1.1. El proceso de medida

La geometría fractal surge de la necesidad de describir y caracterizar cuantitativamente los objetos irregulares, lo que en definitiva significa responder a una pregunta clásica: ¿cómo podemos medirlos?

Cualquier proceso de medida implica la comparación del objeto que queremos medir con un patrón geométrico cuya dimensión conocemos. Por ejemplo, para medir las dimensiones de una mesa utilizamos una regla o metro. Supongamos que queremos medir un segmento de línea recta y que para ello disponemos de reglas de distinta longitud l . Para un tamaño fijo de regla, l , el número de reglas que necesitaremos para “cubrir” nuestro segmento será inversamente proporcional a l

$$N(l) \sim l^{-1}$$

Por consiguiente, la longitud de nuestro segmento será

$$L(l) = N(l) l = \text{const.}$$

Es importante darse cuenta de que el resultado de nuestra medida no depende de la escala l con la que estamos midiendo. Se dice entonces que hemos “medido bien”, esto es, que hemos empleado la dimensión correcta en nuestro patrón de medida, $d = 1$. Por ejemplo, ¿qué ocurriría ahora si utilizáramos otra dimensión de medida? por ejemplo $d = 2$. Esto querría decir que estamos midiendo la superficie del objeto. Para ello, en lugar de reglas utilizamos cuadrados de lado l . De nuevo tendremos que el número de cuadrados de tamaño l que necesitamos para cubrir el segmento va como $N(l) \sim l^{-2}$. Sin embargo, nuestra medida de la superficie será

$$S(l) = N(l) l^2 \sim l$$

de modo que el resultado de la medida ahora sí que depende fuertemente de la escala y además tiende a 0 cuando la precisión de la medida aumenta (cuando l disminuye). En consecuencia, hemos utilizado una dimensión errónea de medida. Este razonamiento puede extenderse a otros objetos. Por ejemplo, si pretendemos medir la longitud de un cuadrado obtendremos que ésta es infinita, mientras que su volumen se hace cero cuando l disminuye. Sólo cuando empleamos la dimensión correcta $d = 2$ tenemos que la medida no depende de la resolución con la que esta se realiza y además proporciona un resultado finito.

No es necesario que el resultado de nuestra medida sea independiente de l para que podamos afirmar que estamos midiendo correctamente el objeto. Por ejemplo, supongamos que queremos medir la longitud de una circunferencia utilizando el mismo conjunto de reglas de distintos tamaños, como se muestra en la figura 13.1.

Es obvio que el resultado de nuestra medida varía fuertemente con el tamaño de la regla utilizada. Sin embargo, a medida que reducimos el valor de l observamos que este resultado tiende asintóticamente (en el límite $l \rightarrow 0$) hacia un valor finito dado por la longitud real $2\pi r$, como se ilustra en la gráfica 13.2.

²Del inglés The Fractal Geometry of Nature.

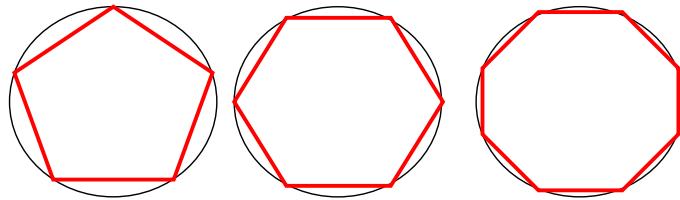


Figura 13.1: Procedimiento de medida de la longitud de una circunferencia utilizando segmentos (indicados mediante el color rojo) de distinta longitud.

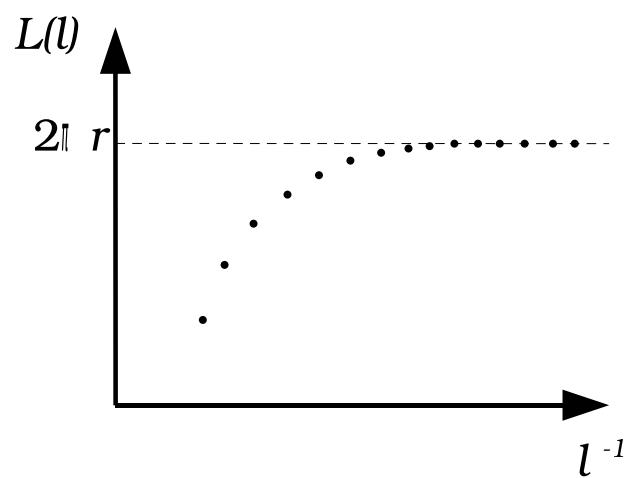


Figura 13.2: Variación del resultado de la medida de la longitud de una circunferencia utilizando segmentos con distinta longitud l .



Figura 13.3: Procedimiento de medida de la longitud de la costa de Inglaterra utilizando segmentos de distinta longitud l . Izquierda: $l = 200$ km, $L = 2400$ km (aprox.). Centro: $l = 100$ km, $L = 2800$ km (aprox.). Derecha: $l = 50$ km, $L = 3400$ km (aprox.).

Este comportamiento quiere decir que nuestra dimensión de medida es correcta pero lo que falla es la geometría del patrón de medida. Lo mismo podría decirse si tratamos de medir la superficie de un cuadrado utilizando pequeños círculos como patrón de medida, parece más lógico utilizar cuadrados, ¿no?

Esta dimensión d , a la que nos hemos referido como dimensión correcta de medida, se denomina *dimensión de Hausdorff* y a su medida (longitud, superficie, volumen,...) se la conoce como *medida de Hausdorff*. Existen otras definiciones de dimensión que están relacionadas con diferentes aspectos de la teoría de la geometría de los objetos, pero a nosotros sólo nos interesa esta.

En general, el volumen $V(l)$ de un objeto arbitrario³ puede ser medido cubriendolo con bolas de tamaño lineal l y volumen l^d . Si necesitamos $N(l)$ bolas para cubrirlo tendremos que

$$V(l) = N(l) l^d$$

Aunque esta definición no es muy precisa, se puede decir que la *dimensión de Hausdorff* del objeto, d , es aquella que hace $N(l) \sim l^{-d}$, ya que el volumen del objeto no cambia si cambiamos la unidad de medida l .

13.1.2. Geometría fractal

Veamos ahora qué ocurre cuando tratamos de medir la longitud de un objeto natural irregular natural, el ejemplo típico es la costa de Inglaterra.

Como se observa en la figura 13.3, al igual que ocurría con el segmento lineal o la circunferencia, a medida que utilizamos reglas de tamaño más pequeño, el número de reglas utilizadas para cubrir la costa aumenta. Sin embargo, cuando representamos la longitud de la costa en función del tamaño de la regla observamos con sorpresa que la longitud no alcanza un valor constante, como en el círculo, sino que parece crecer sin límite a medida que aumenta la precisión de la medida (figura 13.4).

Este comportamiento es debido a que la costa de la isla es irregular, pero fundamentalmente es debido a que esta irregularidad aparece en todas las escalas de longitud, de forma que a medida que utilizamos reglas más pequeñas (aumentamos la

³Con este término nos estamos refiriendo a cualquier tipo de medida de Hausdorff, y no exclusivamente al concepto clásico de volumen.

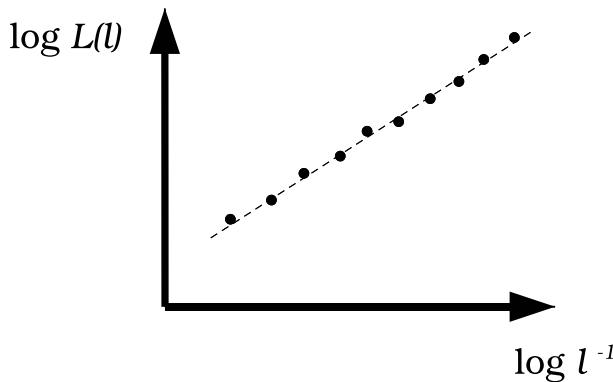


Figura 13.4: En logaritmos, variación del resultado de la medida de la longitud de la costa de Inglaterra utilizando segmentos con distinta longitud l .

resolución de nuestra observación) vamos descubriendo cada vez más detalles de la costa, de modo que el resultado de nuestra medida tiende a infinito. La causa de este comportamiento es la misma del porqué en algunos atlas portugueses aparece que la longitud de la frontera política entre España y Portugal es mayor que la que podemos encontrarnos en los mapas españoles. Como la extensión de Portugal es menor que la de España, en los mapas portugueses, donde sólo se muestra Portugal, la frontera aparece con una escala diferente y un mayor nivel de detalle que en los mapas de España, donde se suele mostrar la Península Ibérica, por lo que no aparecen las irregularidades de la frontera en las escalas pequeñas.

Volviendo al problema de la costa de Inglaterra, el resultado mostrado en la figura anterior nos dice que nuestra medida es incorrecta y que la dimensión de Hausdorff que hemos utilizado en la medida, $d = 1$, no es la adecuada. Sin embargo, si utilizamos la dimensión $d = 2$ e intentamos medir la costa cubriéndola con círculos de distintos diámetros, de nuevo observaremos que nuestra medida tiende a 0 cuando $l \rightarrow 0$. Entonces, ¿cuál es la dimensión correcta de la costa?

13.1.2.1. Dimensión fractal

Como hemos visto antes, para que el resultado de nuestra medida no varíe con la unidad de medida l se debe cumplir que $N(l) \sim l^{-d}$, por consiguiente debe existir una dimensión de Hausdorff, denominada *dimensión fractal* d_f , que verifique $N(l) \sim l^{-d_f}$. Para calcularla podemos tomar logaritmos en ambos lados de la ecuación: $\log N(l) \simeq d_f \log l^{-1}$, de modo que si representamos $\log N(l)$ frente a $\log(1/l)$ debemos obtener una recta cuya pendiente sea la dimensión fractal. Esto es lo que hemos hecho en la Fig. 4 para la costa oeste de Inglaterra. El ajuste a una recta da $d_f = 1,25$.⁴ La definición estricta de dimensión fractal es

$$d_f \equiv \lim_{l \rightarrow 0} \frac{\log N(l)}{\log(1/l)}$$

Es muy común pensar que un fractal es un objeto con una dimensión de Hausdorff fraccionaria, es decir, no entera. Aunque en la mayoría de los casos es así, la definición estricta de fractal es aquel objeto cuya dimensión de Hausdorff es mayor que su

⁴Por ejemplo, la frontera España-Portugal tiene una dimensión fractal de 1,14.

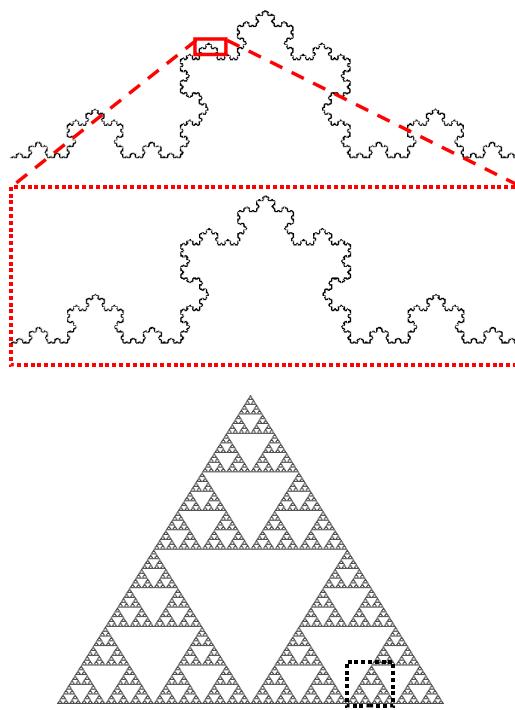


Figura 13.5: (Arriba) Curva de Koch y propiedad de autosimilaridad: la ampliación de cualquiera de sus partes coincide exactamente con la figura total. (Abajo) Triángulo de Sierpinski: si ampliamos la parte seleccionada de nuevo obtendremos exactamente el objeto original.

dimensión topológica: $d_f > d_T$. La dimensión topológica es $d_T = 0$ para un conjunto disconexo de puntos, $d_T = 1$ para una curva, $d_T = 2$ para una superficie y $d_T = 3$ para un sólido.

En la mayoría de los fractales también se cumple que $d_f < d_E$, donde d_E es la *dimensión de embedding*, definida como la menor dimensión del espacio euclídeo en la que podemos insertar el objeto. Por ejemplo, en el caso de la circunferencia o de la costa de Inglaterra tenemos que $d_E = 2$. En el caso de una recta $d_E = 1$. Sin embargo, existen curvas fractales que llenan densamente el espacio de modo que $d_f = d_E = 2$.

13.1.2.2. Autosimilaridad

La propiedad o característica fundamental que define a un fractal es la *auto-similaridad*. Un objeto es auto-similar cuando está compuesto de partes que son similares al todo. Cuando las partes son exactamente iguales al objeto original (después del correspondiente aumento), hablamos de *fractales deterministas*, o fractales perfectos. En la figura 13.5 mostramos dos ejemplos muy conocidos de fractales deterministas, la curva de Koch y el triángulo de Sierpinski (*Sierpinski gasket*).

Cuando las partes no son exactamente iguales al todo, pero sí lo son estadísticamente, es decir, exhiben las mismas propiedades estadísticas, entonces hablamos de *fractales estadísticos* o *fractales aleatorios*. Este tipo de fractales son los que realmente se observan en la naturaleza, pues en su formación participan mecanismos aleatorios y esta aleatoriedad es la que hace que no sean fractales perfectos. El ejem-

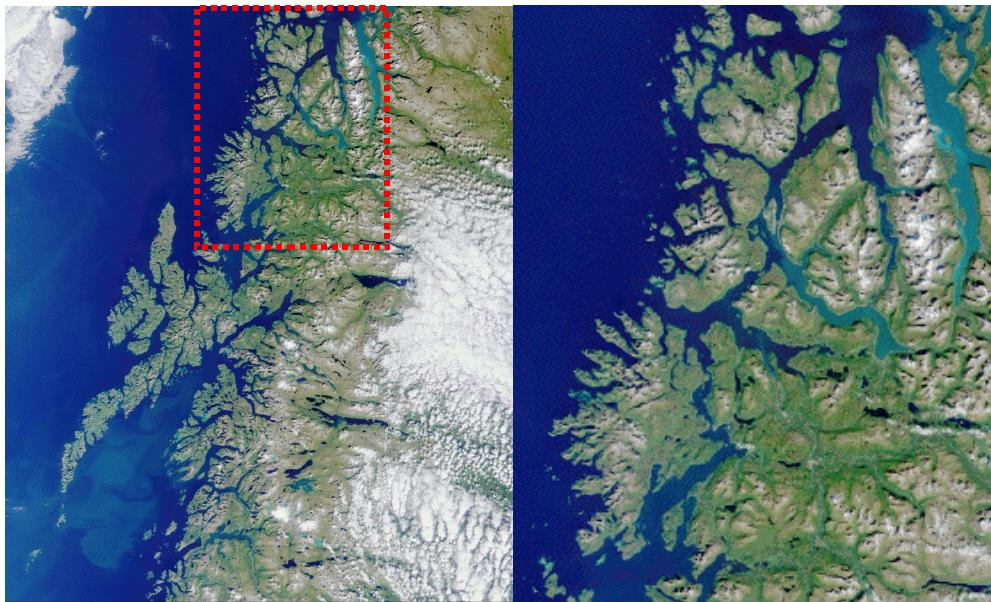


Figura 13.6: (Derecha) Fotografía desde satélite de una linea de costa. (Izquierda) Ampliación de la región resaltada en la figura de la izquierda.

clojero clásico es una línea de costa, como la mostrada en la figura 13.6.

Si observamos las dos imágenes de la misma costa pero con diferentes escalas, no habrá forma de distinguirlas entre ellas, es decir, no habrá forma de saber cuál de ellas ha sido tomada con una mayor resolución. Otro ejemplo de fractal aleatorio es un árbol. Si cogemos una rama y la ampliamos hasta que su tamaño coincide con el del árbol veremos que en ambos casos se reproducen las mismas propiedades estadísticas. Podemos repetir el proceso cogiendo una ramita más pequeña de la rama anterior y ampliéndola, y así sucesivamente.

13.1.2.3. Invariancia bajo cambios de escala

Como la principal característica de un fractal es que cada parte es exacta o estadísticamente similar al todo, los fractales no tienen un tamaño o escala característica. Todo lo contrario de los objetos euclídeos, que sí la tienen: los círculos o las esferas tienen diámetros, los cuadrados tienen lados con una longitud característica... incluso los seres vivos también tienen escalas características (la escala característica de un ser humano puede ser su altura, por ejemplo).

Debido a la autosimilitud y a la consiguiente ausencia de una escala característica, los fractales exhiben una simetría distinta a las tres que nos son familiares, que son: simetría bajo traslación, simetría bajo reflexión y simetría bajo rotación. En la Fig. 13.7 hemos ilustrado estos tres tipos de simetrías con imágenes cotidianas.

Por ejemplo, moviéndonos sobre un muro de ladrillos en las dos direcciones del espacio “veremos” siempre lo mismo. Ciertos objetos, por ejemplo el cuerpo humano, tienen una dirección espacial privilegiada de forma que la reflexión de una parte del objeto sobre ese eje reproduce la otra parte, como si de un espejo se tratara. En otros casos, si giramos el objeto alrededor de un eje (por ejemplo una rueda) tampoco observaremos ningún cambio.

Sin embargo, los fractales exhiben una cuarta simetría, la simetría o invariancia



Figura 13.7: Distintos tipos de simetrías

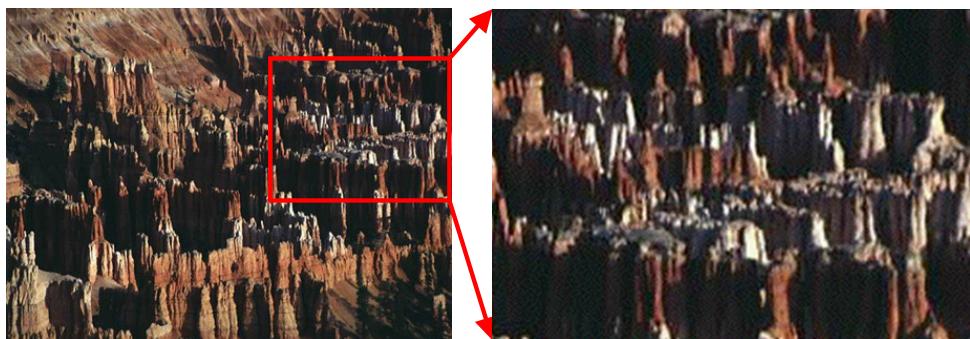


Figura 13.8: Simetría bajo cambio de escala.

bajo cambios de escala. Por ejemplo, en la foto de una formación rocosa en el Cañón del Colorado (figura 13.8) hemos cogido una parte del paisaje y la hemos ampliado en la ventana de la derecha. Estadísticamente ambas imágenes son similares, y sin conocer el lugar no podríamos diferenciar entre la imagen más alejada y la más próxima. Por lo tanto, una forma “casera” de reconocer un fractal es coger una parte del mismo y ampliarla comparándola con el objeto completo, si somos incapaces de discernir entre las dos imágenes podemos sospechar que nos encontramos con un fractal.

13.1.3. Generación de fractales: fractales matemáticos

Existe un conjunto importante de fractales, denominados fractales matemáticos, que pueden ser fácilmente construidos mediante un proceso iterativo de sustitución de formas geométricas elementales. Este proceso repetitivo consta de una semilla o estado inicial y una regla geométrica que se aplica en cada paso, de modo iterativo, sobre el objeto obtenido en el paso anterior. Cuando la regla de sustitución es siempre la misma, el fractal matemático obtenido es determinista; cuando algún aspecto de la regla es aleatorio, el fractal matemático es aleatorio. En la Fig. 13.9 se muestran cómo se construyen algunos de los fractales matemáticos deterministas más conocidos.

En el caso de la curva de Koch partimos ($k = 0$) de un segmento de recta. La regla geométrica consiste en dividir el segmento en tres segmentos iguales y sustituir el del medio por un triángulo equilátero del que quitamos la base, obteniendo cuatro segmentos iguales. Cuando iteramos la regla un número infinito de veces aplicándola sobre cada uno de los segmentos obtenidos en la generación del paso anterior ob-

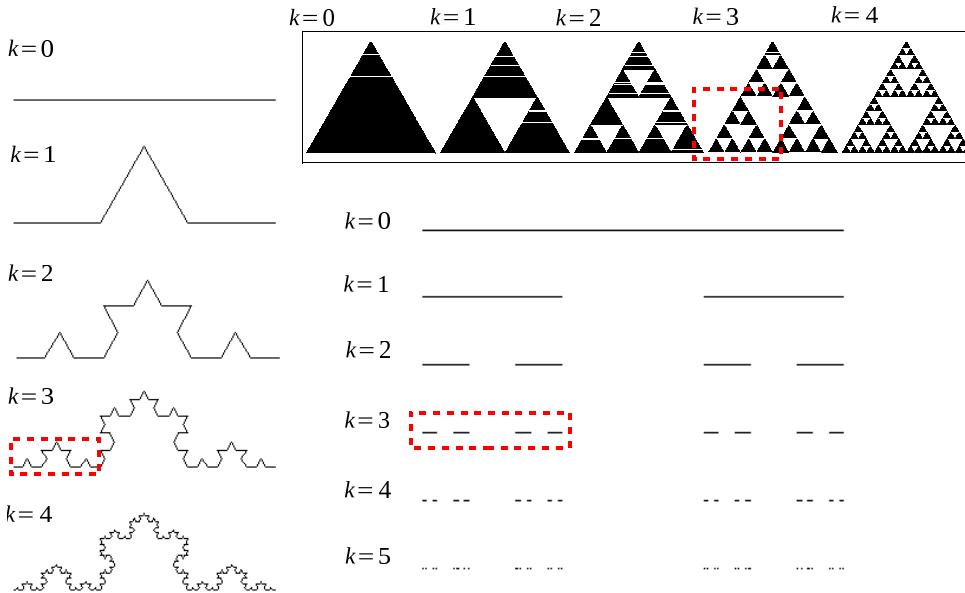


Figura 13.9: Primeros pasos de la generación de algunos fractales matemáticos deterministas: (izquierdo) Curva de Koch; (derecha arriba) Triángulo de Sierpinski; (derecha abajo) Conjunto de Cantor. Cada paso de la iteración en la generación del fractal ha sido denotado mediante k .

tenemos la curva de Koch. Para construir el triángulo de Sierpinski comenzamos con un triángulo equilátero que dividimos en cuatro triángulos y sustraemos el del medio. De nuevo, este proceso es repetido indefinidamente sobre los triángulos obtenidos en cada paso. En el caso del conjunto de Cantor, cortamos el intervalo unidad en tres segmentos iguales y quitamos el del medio.

En todos los casos, en el paso de generación k podemos aumentar una parte del objeto (indicado en la figura mediante el recuadro rojo) por un factor y obtendremos el objeto del paso de generación $k - 1$. Este factor es 3 para la curva de Koch y el conjunto de Cantor, y 2 para el triángulo de Sierpinski.

Para estos fractales matemáticos podemos calcular analíticamente la dimensión fractal. En el caso del conjunto de Cantor, la forma natural para medir la longitud del conjunto es utilizando segmentos de longitud $l_k = (1/3)^k$, con $k = 1, 2, \dots, \infty$. Por ejemplo, cuando $l = 1$ necesitamos sólo 1 segmento para cubrirlo: $N(1) = 1$. Cuando $l = 1/3$, necesitamos dos segmentos, $N(1/3) = 2$; cuando $l = (1/3)^2$, tenemos que $N((1/3)^2) = 4$. Generalizando obtenemos que $N(l_k) = 2^k$. Esto se ve muy fácil en la figura anterior cuando consideramos que k indica el paso de generación: cada paso de generación k representa el fractal cubierto con segmentos de longitud $l_k = (1/3)^k$. Si ahora sustituimos estas expresiones en la definición de dimensión fractal

$$d_f \equiv \lim_{l \rightarrow 0} \frac{\log N(l)}{\log(1/l)} = \lim_{k \rightarrow \infty} \frac{\log N(l_k)}{\log(1/l_k)}$$

obtenemos que la dimensión fractal del conjunto de Cantor es $d_f = \log 2 / \log 3 \simeq 0,639\dots$. Como $d_f > d_T = 0$, el conjunto de Cantor es un fractal. También se cumple que $d_f < d_E = 1$.

En el caso del triángulo de Sierpinski, la medida óptima se realiza cubriendo el fractal con triángulos de lado $l_k = (1/2)^k$. En este caso es fácil darse cuenta de que

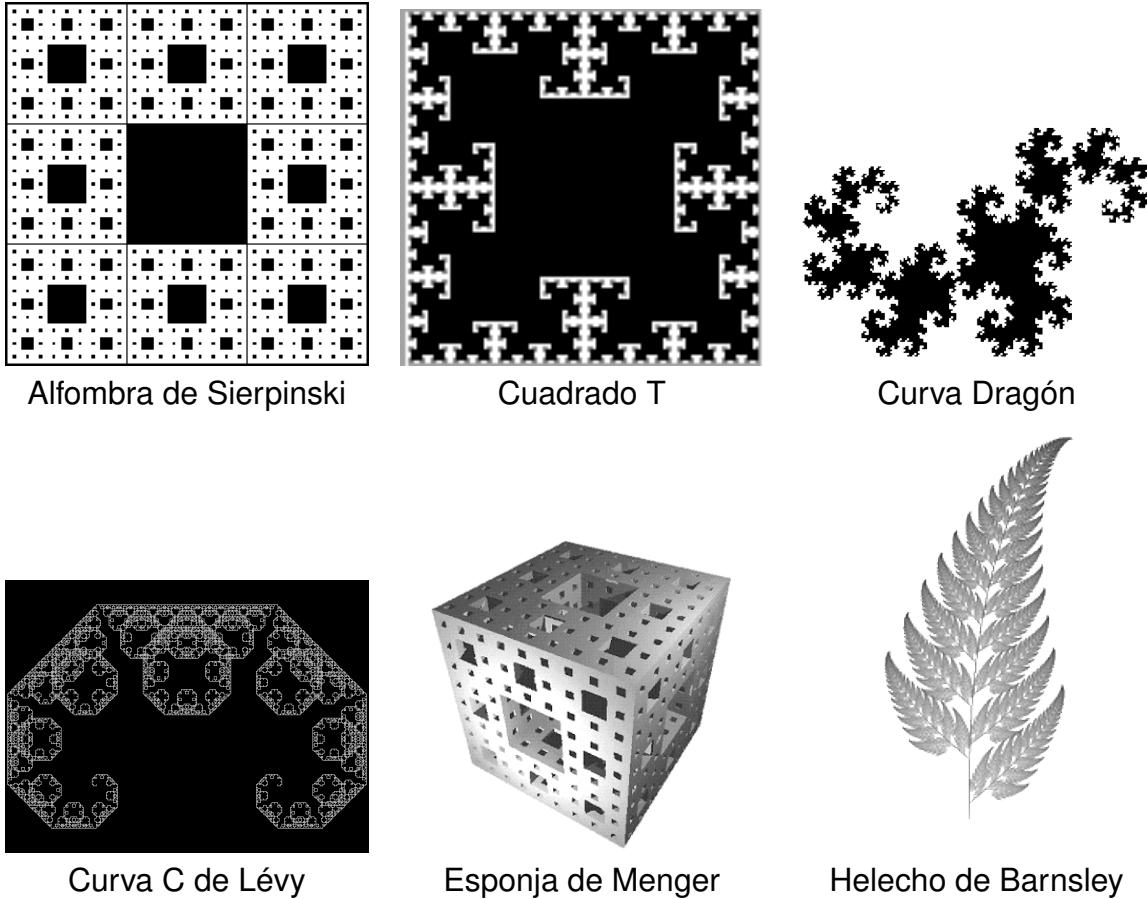


Figura 13.10: Fractales matemáticos deterministas generados mediante funciones iteradas.

$N(l_k) = 3^k$ y que por tanto su dimensión fractal es de $d_f = \log 3 / \log 2 = 1,585 \dots$, que es menor que su dimensión de embedding $d_E = 2$ y mayor que su dimensión topológica $d_T = 1$.

Por último, para la curva de Koch tenemos que si medimos con segmentos de longitud $l_k = (1/3)^k$ necesitaremos $N(l_k) = 4^k$ segmentos para cubrirlo, de modo que $d_f = \log 4 / \log 3 = 1,262 \dots$, también satisfaciendo $1 = d_T < d_f < d_E = 2$.

En la Fig. 13.10 se muestran otros fractales matemáticos muy conocidos también obtenidos a partir de funciones iteradas (sistemas de función iterada). Todos estos fractales son exactamente auto-similares: observamos exactamente la misma forma geométrica en todas las escalas.

Hay muchas formas de generar un fractal matemático, y dentro de cada forma infinitas posibilidades. Como veremos en el capítulo dedicado a los sistemas dinámicos, los fractales aparecen asociados a sistemas dinámicos no-lineales, y muy especialmente a los sistemas caóticos: los atractores caóticos tienen la forma de fractales más o menos autosimilares.

Por ejemplo, consideremos la relación matemática de recurrencia $z_{n+1} = z_n^2 + c$, con $z_0 = 0$. Para un punto c del plano complejo ($c = x + iy$), esa relación recursiva produce una sucesión de valores de z_n que puede crecer ilimitadamente a medida que n aumenta, o puede mantenerse acotada por debajo de un cierto valor. Si ahora representamos en el plano complejo (x, y) mediante color negro los valores de c que dan lugar a una sucesión acotada, y mediante una escala de azules aquellos puntos que

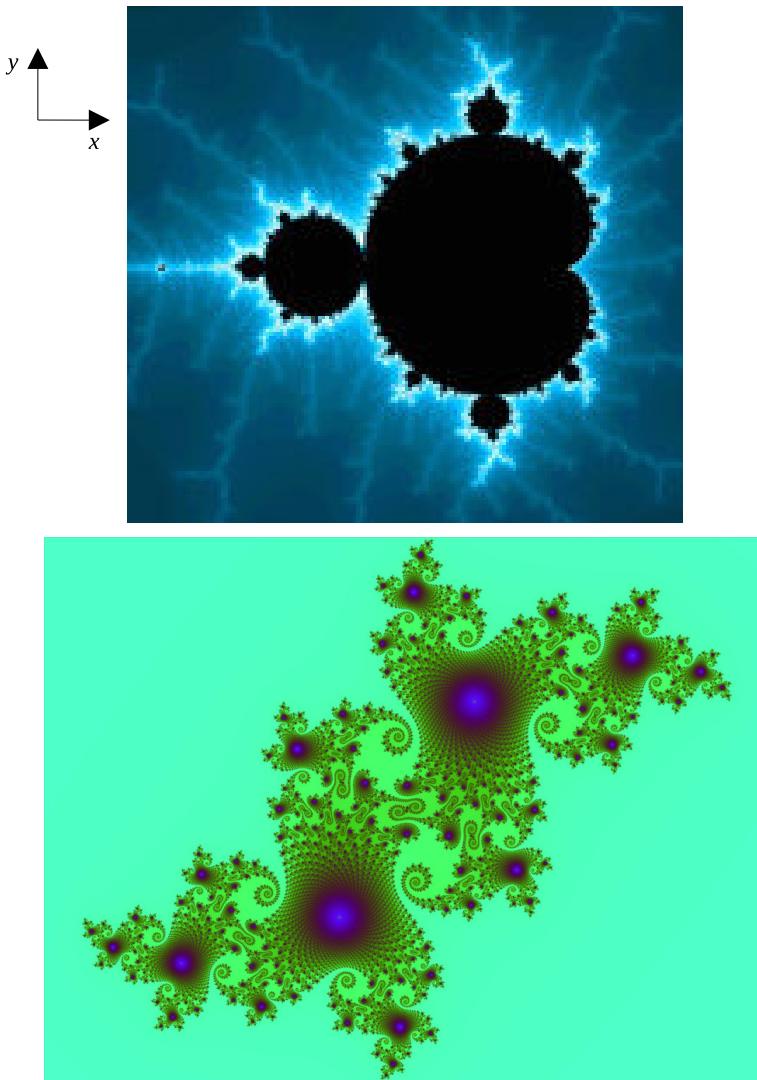
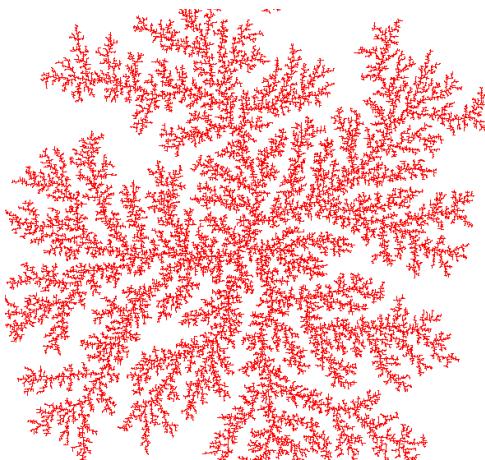


Figura 13.11: (Arriba) Conjunto de Mandelbrot. (Abajo) Conjunto de Julia.

producen una sucesión divergente (cuanto más intenso es el azul más rápidamente se produce la divergencia), obtendremos el conocidísimo conjunto de Mandelbrot mostrado en la Fig. ???. Resulta fascinante viajar dentro del conjunto de Mandelbrot y observar cómo ese mundo se reproduce en todas las escalas: cada pequeño círculo de la periferia reproduce un mundo similar al mostrado en la figura, y lo mismo para los circulitos aun más pequeños de la periferia de esos pequeños círculos, y así hasta el infinito. Otro conjunto muy conocido generado de forma similar es el conjunto de Julia (Fig. 13.11). En estos casos tenemos fractales quasi auto-similares, ya que el fractal aparece aproximadamente (aunque no exactamente) idéntico en diferentes escalas. Este tipo de fractales quasi auto-similares contienen copias del fractal completo de forma distorsionada o degenerada.

Por último tenemos los fractales matemáticos aleatorios (véase Fig. 13.12). Típicos ejemplos de estos fractales aleatorios son los generados por procesos estocásticos como la *difusión browniana* (aleatoria) de partículas que se agregan al contacto con otras partículas (DLA, *diffusion-limited aggregation*), produciendo formaciones dendríticas en todas las escalas como la que se muestra en la figura. También podemos generar fractales matemáticos aleatorios utilizando funciones geométricas iteradas en



Clúster formado por DLA



Costas fractales



Montañas fractales

Figura 13.12: Fractales matemáticos aleatorios.

las que en cada paso se produce una pequeña alteración aleatoria de la regla de sustitución. De esta forma se generan los paisajes montañosos o relieves naturales mostrados en la figura. Como hemos dicho, en este caso la auto-similaridad es estadística, es decir, el fractal es estadísticamente el mismo en todas las escalas ya que existen medidas estadísticas que se mantienen en todas las escalas. Esta auto-similaridad estadística produce fractales que nos recuerdan a muchos a los patrones observados en la naturaleza.

13.1.4. Cómo medir la dimensión fractal

Uno de los problemas prácticos más importantes es determinar la dimensión fractal de un fractal que bien ha podido ser generado computacionalmente, o bien es una imagen digitalizada. Aquí mostramos los dos métodos más utilizados: el *método sandbox* y el método de *box counting*.

13.1.4.1. Método sandbox

Este método hace uso de la siguiente relación de escala:

$$M(R) \sim R^{d_f}$$

donde $M(R)$ es la masa del fractal contenida en un círculo de radio R centrado en cualquier punto del mismo. Para determinar d_f debemos escoger un punto del fractal (o un píxel de la imagen) y trazar en torno a él n círculos de radios $R_1 < R_2 < \dots < R_n$, donde R_n debe ser más pequeño que el tamaño del fractal, es decir, todos los círculos deben caer dentro del fractal. Para cada círculo R_i contamos el número de puntos (o píxeles) del fractal que se encuentran dentro del mismo. El valor obtenido será la masa del fractal $M_1(R_i)$. Dependiendo de la geometría, a veces es más conveniente utilizar cuadrados de lado $L_1 < L_2 < \dots < L_n$ en lugar de círculos. Este proceso es repetido

en otros puntos del fractal, de modo que para el punto j obtendremos $M_j(R_i)$. Una vez que hemos concluido el análisis sobre m puntos, calculamos la masa media $M(R_i)$ dentro del círculo de radio R_i promediando sobre todos los puntos analizados:

$$M(R_i) = \frac{1}{m} \sum_{j=1}^m M_j(R_i)$$

A continuación representamos $M(R_i)$ frente a R_i en una gráfica doble logarítmica. La pendiente de la curva, para valores grandes de R_i , determina la dimensión fractal. Como hemos dicho, para evitar efectos de contorno el radio de los círculos debe ser menor que el radio o longitud del fractal, y los centros de los círculos deben ser escogidos bien dentro del fractal, de modo que los círculos de mayor radio caigan dentro del objeto. Cuanto más grande sea el fractal y mayor sea m , mejor será la estadística. En el caso de fractales aleatorios, también se puede considerar un único punto (por ejemplo el centro del DLA) y promediar sobre diferentes realizaciones (construcciones) del fractal.

13.1.4.2. Método de box-counting

Este método consiste en dividir la región del espacio ocupada por el fractal en cajas de lado l . La dimensión de estas cajas (segmentos, cuadrados o cubos) será d_E . Por ejemplo, para hacer un box counting sobre el conjunto de Cantor sólo necesitaremos segmentos, pero para hacer un box-counting sobre el triángulo de Sierpinski o la curva de Koch necesitaremos cajas cuadradas. A continuación contamos cuántas de estas cajas cubren el fractal, este número será $N(l)$. Para ello, sólo tenemos en cuenta aquellas cajas en las que caiga alguna parte del fractal, aunque sea un único punto. Repetimos el proceso para diferentes tamaños de caja, tomando como tamaño máximo de caja l_{\max} el tamaño del fractal, y como tamaño mínimo l_{\min} el tamaño de un punto o píxel de la imagen.

Como sabemos, $N(l)$ escala con l de la forma:

$$N(l) \sim l^{-d_f}$$

de modo que obtendremos la dimensión fractal representando $N(l)$ frente a $1/l$ en una gráfica doble logarítmica y realizando el ajuste de los puntos a una recta mediante regresión lineal. La pendiente nos dará nuevamente d_f .

13.1.5. Fractales en la naturaleza

Debido a que los fractales poseen un infinito grado de irregularidad y detalle que se repite en todas las escalas, ningún objeto natural puede ser un fractal. Sin embargo, los objetos naturales pueden mostrar “fractalidad” en un rango limitado de escalas, hablamos entonces de fractales naturales. En la figura 13.13 mostramos ejemplos de fractales naturales.

13.1.6. Propiedades prácticas de los fractales

Los fractales tienen importantes propiedades geométricas. Por ejemplo, la curva de Koch es una curva continua pero no diferenciable (o derivable) en ninguno de sus



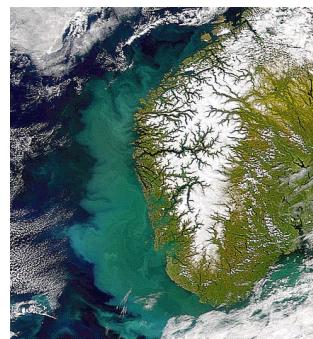
Árboles



Sistemas montañosos



Rayos



Ríos



Helechos



Venación en las hojas



Corales



Rocas



Coliflor



Brécol Romanescu

Figura 13.13: Fractales naturales.

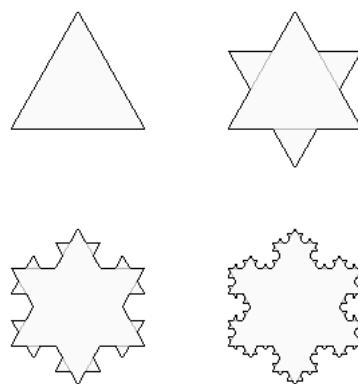


Figura 13.14: Cuatro primeros pasos de la generación del copo de nieve de Koch, compuesto de tres curvas de Koch.

puntos. Estas propiedades geométricas hacen que en muchos casos los fractales sean la solución escogida por la naturaleza ante importantes problemas de optimización. Fijémonos, por ejemplo, en el copo de nieve de Koch (figura 13.14): tiene un perímetro infinito encerrando un área finita.

Esto quiere decir que la fractalidad proporciona estructuras que tienen una medida infinita en una dimensión dada (longitud infinita en el caso del copo de nieve de Koch), pero que están encerradas en un espacio finito de dimensión superior (área finita para el copo de nieve de Koch). Esta propiedad tiene importantes consecuencias en la naturaleza y en la industria, ya que proporciona la solución a un problema de espacio. La estructura ramificada de los árboles es la que proporciona, dentro de un volumen limitado, una mayor área de intercambio de $\text{CO}_2 - \text{O}_2$ con el aire y una mayor exposición para la captura de la radiación solar. Lo mismo se puede decir de nuestro sistema circulatorio o del árbol bronquial que constituye nuestros pulmones. ¿Cuál es la forma óptima de asegurar el riego sanguíneo a todas las partes de nuestro? ¿Y la de maximizar la absorción del oxígeno por parte de la sangre en nuestros alvéolos pulmonares dentro de un volumen limitado? La respuesta está en una estructura fractal ramificada, como se muestra en la Fig. 13.15. El sistema nervioso, el forro intestinal, los conductos biliares son otros ejemplos.

Se ha demostrado que la auto-similaridad de las estructuras fractales fisiológicas las hace más robustas. Por ejemplo, la fractalidad de los pulmones los hace ser más tolerantes a posibles fallos durante el crecimiento. La fractalidad del sistema circulatorio humano debilita los “martillazos” producidos por la violencia del bombeo de nuestro corazón. Si hubiera cualquier resonancia en la circulación de la sangre podríamos morir. Fuera de nuestro cuerpo humano, también se ha observado que la estructura fractal de la costa debilita al máximo el impulso de las olas, por lo que representa la mejor forma de “defenderse” de su acción erosiva.

Esta propiedad geométrica de máxima superficie en un volumen finito tiene importantísimas aplicaciones tecnológicas y económicas. Un ejemplo de esto son los catalizadores, sustancias que desempeñan el papel de “intermediarios” en ciertas reacciones químicas de gran importancia, activando, facilitando o acelerando su desarrollo. La presencia de un catalizador en el proceso de reacción se limita a modificar la ve-

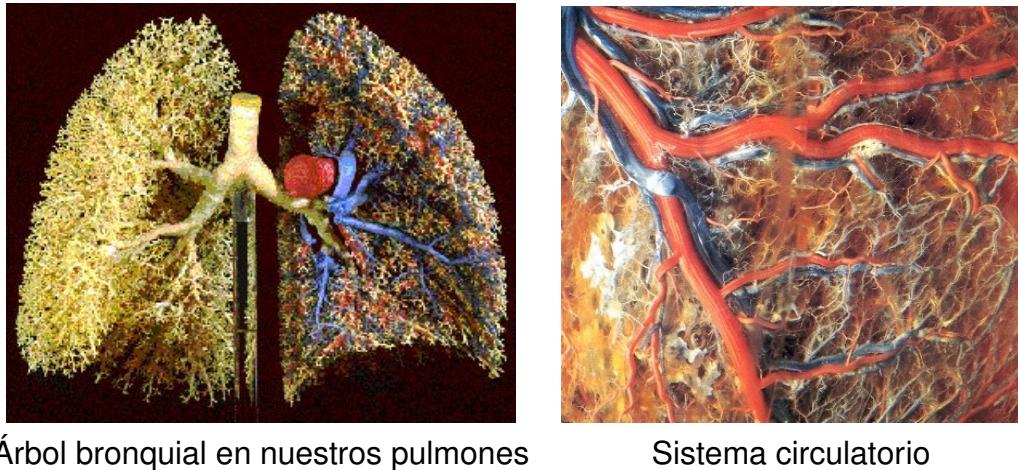
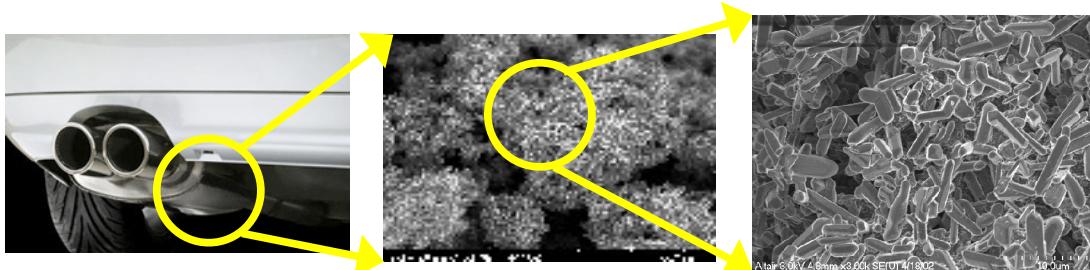


Figura 13.15: Fractales dentro del cuerpo humano.

Figura 13.16: Estructura de cristales de Rutilo TiO_2 utilizada como catalizador o soporte de un catalizador.

locidad de la transformación, el catalizador no se considera ni reactivo ni producto en la reacción. En la catálisis heterogénea, estos catalizadores suelen aparecer sobre superficies o a veces es la propia superficie la que tiene un efecto catalizador sobre los compuestos químicos que se encuentran en el medio (generalmente gaseoso). Las especies químicas son adsorbidas sobre la superficie activa del catalizador, donde reaccionan entre sí dando lugar a los productos deseados. Cuanto mayor sea la superficie del catalizador dentro de un volumen fijo, mayor será el rendimiento de la reacción. De nuevo, la solución óptima consiste en utilizar estructuras muy rugosas con propiedades fractales. En la Fig. 13.16 se ha ilustrado este hecho mostrando un catalizador situado en el interior del tubo de escape de un coche. La función del catalizador es descomponer las sustancias presentes en el humo producido por la combustión de la gasolina (como el monóxido de carbono CO, los óxidos de nitrógeno NO_x y los hidrocarburos HC), y transformarlas en productos menos nocivos reduciendo así la contaminación atmosférica.

En telefonía móvil encontramos otra aplicación de los fractales. Utilizando antenas fractales (Fig. 13.17), la multitud de escalas permite la recepción de una banda más ancha de frecuencias en un espacio compacto

También se han desarrollado fibras ópticas empaquetadas de forma fractal para lograr guías de onda con una distorsión muy baja (figura 13.18).

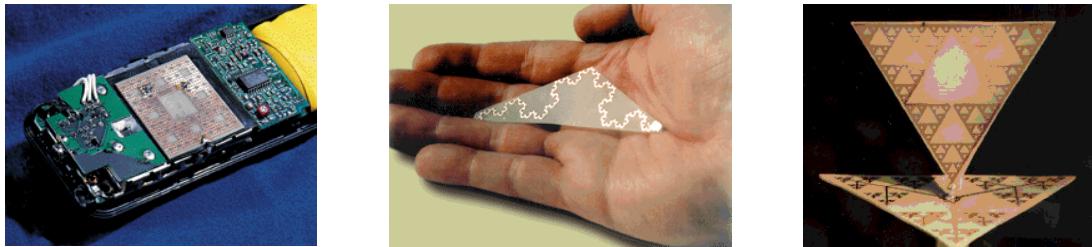


Figura 13.17: Antenas fractales.

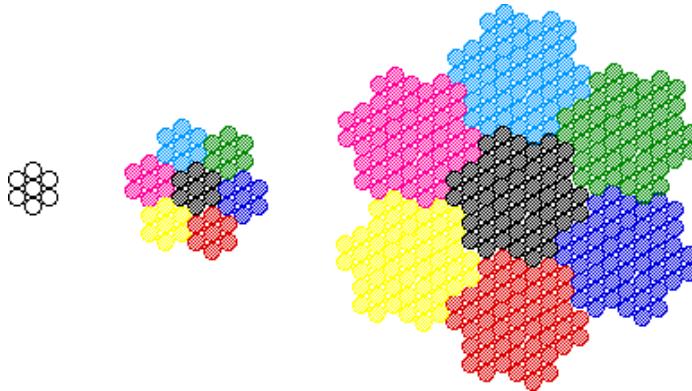


Figura 13.18: Empaqueamiento fractal de fibras ópticas.

13.2. Ejemplos y ejercicios en C

13.2.1. Conjunto de Cantor

El conjunto ternario de Cantor es uno de los fractales más sencillos de construir de manera geométrica. Se trata de un conjunto de puntos de la recta que se obtiene partiendo de un intervalo cerrado, digamos el $[0; 1]$. A este intervalo se le quita su subintervalo central, de modo que queda el conjunto cerrado $[0; \frac{1}{3}] \cup [\frac{2}{3}; 1]$. En cada uno de estos subintervalos se repite la misma operación, quedando $[0; \frac{1}{9}] \cup [\frac{2}{9}; \frac{1}{3}] \cup [\frac{2}{3}; \frac{7}{9}] \cup [\frac{8}{9}; 1]$. Esta misma regla se repite una y otra vez, de modo que, de cada vez, se duplica el número de subintervalos, a la vez que se reduce la longitud de cada uno de ellos a la tercera parte (ver figura ??).

Matemáticamente se demuestra que esta sucesión de reglas genera una sucesión de conjuntos que converge a un conjunto formado por infinitos puntos disjuntos (esto es, entre cualesquiera dos de ellos se halla una infinidad de números reales). Este conjunto se llama conjunto de Cantor y se dice que es el atractor del proceso iterativo.

Para calcular este conjunto, construiremos los intervalos que se generan al aplicar las sucesivas “extracciones del tercio central”. Para ello definimos una estructura de datos,

```
struct Intervalo {
    double a, b;
};
```

que representa uno de estos intervalos, $[a; b]$, y creamos un *array* de *Intervalos* que iremos llenando. El primero de estos intervalos será el $[0; 1]$, del que partimos. Lo declaramos como

```
intervalo [0].a=0.0;
intervalo [0].b=1.0;
nintervalos=1;
```

inicializando el contador de Intervalos, nintervalos, a 1. Después, llevamos a cabo las iteraciones sobre todos los intervalos construidos con anterioridad (la primera vez, sólo sobre el $[0; 1]$). Estas iteraciones consisten en construir, a partir de un intervalo dado, uno nuevo, el “tercer tercio” del de partida, que se añade al array:

```
intervalo [m].a=intervalo[n].a
    +2*(intervalo [n].b-intervalo [n].a)/3;
intervalo [m].b=intervalo[n].b;
```

Posteriormente, sustituimos el intervalo que estamos transformando por su “primer tercio”

```
intervalo [n].a=intervalo[n].a; intervalo [n].b=intervalo[n].a
    +(intervalo [n].b-intervalo [n].a)/3;
```

Así, para todos los intervalos que habíamos construído en la iteración anterior (o para el Intervalo $[0; 1]$ creado inicialmente).

De este modo, se rellena el array `intervalo[NINTERVALOS]` que es una aproximación a la solución del problema tanto mejor cuanto mayor sea NINTERVALOS (esto es, el número de iteraciones permitidas). El código de este programa se encuentra en el listado 13.1.

Listado 13.1: El programa que genera los intervalos del conjunto de Cantor.

```
1 #include <stdio.h>
2
3 #define NINTERVALOS 100000
4
5 struct Intervalo
6 {
7     double a, b;
8 };
9
10 int main(int argc, char** argv)
11 {
12     struct Intervalo intervalo[NINTERVALOS];
13     int n, m, nintervalos;
14
15     intervalo[0].a=0.0;
16     intervalo[0].b=1.0;
17     nintervalos=1;
18
19     /* bucle de iteraciones (tantas como se pueda) */
20     do {
21         /* aplicar la transformación a todos
22            los intervalos existentes      */
23         for(n=nintervalos-1;
24             n>=0 && nintervalos<NINTERVALOS; n--)
```

```

25    {
26        m=nintervalos;
27
28        /* segundo subintervalo */
29        intervalo[m].a=intervalo[n].a+2*(
30            intervalo[n].b
31            -intervalo[n].a)/3;
32        intervalo[m].b=intervalo[n].b;
33
34        /* primer subintervalo */
35        intervalo[n].a=intervalo[n].a;
36        intervalo[n].b=intervalo[n].a+(
37            intervalo[n].b
38            -intervalo[n].a)/3;
39
40        nintervalos++;
41    }
42
43 } while(nintervalos<NINTERVALOS);
44
45 /* exportar las cotas de los intervalos */
46 for(n=0; n<nintervalos; n++)
47 {
48     printf(" %g\t%g\n", intervalo[n].a,
49             intervalo[n].b);
50 }
51
52 return 0;
53 }
```

Ejercicio 13.1. Representar estos Intervalos con Gnuplot a distintas resoluciones. Suponiendo que se hayan guardado en un fichero llamado “cantor3.dat”, se puede utilizar el siguiente comando:

```
plot 'cantor3.dat' u 1:(0) with points pt 6 ps 0.1
```

Luego, se puede utilizar el “zoom interactivo” de Gnuplot (usualmente arrastrando el ratón sobre la gráfica mientras se mantiene pulsado el botón derecho del ratón) para ampliar este conjunto. Se verá que el aspecto de las partes ampliadas del conjunto es similar al del propio conjunto: *el conjunto de Cantor es un fractal*. Para volver al “zoom” inicial, indicaremos a Gnuplot: `reset`.

Ejercicio 13.2. Representar, ahora, los Intervalos usando la instrucción de Gnuplot

```
plot 'cantor3.dat' u 1:2 with points pt 6 ps 0.1
```

Para órdenes de generación bajos (`NINTERVALOS` pequeño) los puntos no se encuentran “alineados” con la diagonal. Explique qué está sucediendo y qué indica esta anomalía.

La dimensión fractal del conjunto de Cantor se puede calcular de una manera sencilla a partir del proceso iterativo de construcción del conjunto. Como se ha explicado, al hablar del método de *box-counting*, la dimensión fractal viene dada por la pendiente de la recta que se obtiene al representar el logaritmo del mínimo número de cajas disjuntas de un tamaño dado que contienen puntos del fractal frente al logaritmo del tamaño de dichas cajas. En el proceso iterativo de construcción del conjunto de Cantor, el número mínimo de cajas de un tamaño dado ($1/3^k$) que recubren el fractal es igual al número intervalos (2^k), generados en la k -ésima iteración.

De este modo, si en un paso de iteración (al final, pero dentro, del bucle `do...while`) escribimos a un archivo el logaritmo de la longitud de los intervalos “`log(intervalo[0] .- b-intervalo[0] . a)`” (da igual cuál, ya que todos tendrán la misma longitud) frente al logaritmo del número de intervalos definidos (nuestro *box counting*), “`log((double)nintervalos)`”, entonces obtendremos la recta buscada: su pendiente será la dimensión fractal del conjunto de Cantor.

Ejercicio 13.3. Implementar este algoritmo para obtener un archivo con las parejas de valores `(-log(intervalo[0].b-intervalo[0].a), log((double)nintervalos))` y calcular la dimensión fractal utilizando el programa ajuste a una recta del listado 11.29. Comprobar que esta dimensión fractal del conjunto ternario de Cantor es igual a $D_F = \log(2) / \log(3) \simeq 0,63$.

Nota: Obsérvese que es irrelevante la base de los logaritmos que se esté usando en el *box-counting* (¡mientras sea siempre la misma!).

13.2.2. Triángulo de Sierpinski

El triángulo de Sierpinski es uno de los fractales bidimensionales más famosos. Igual que el fractal de Cantor, existe una manera geométrica de construirlo. Se parte de un triángulo \widehat{ABC} y de éste se eliminan los puntos contenidos en el triángulo determinado por los tres puntos centrales de sus lados, que serán $\frac{1}{2}(A + B)$, $\frac{1}{2}(B + C)$, $\frac{1}{2}(C + A)$. Con los tres nuevos triángulos (unidos por sus vértices) en que quedan repartidos los puntos “supervivientes”, se vuelve a hacer la misma operación: eliminar los triángulos centrales. De este modo, como en el ejemplo del conjunto de Cantor, los triángulos considerados en cada iteración convergen hacia un conjunto fractal, formado por puntos del plano.

Ejercicio 13.4. Escribir un programa similar al del conjunto de Cantor para generar el triángulo de Sierpinski. Para ello, defínase una estructura `Triangulo` que contenga las coordenadas de los tres puntos de un triángulo

```
struct Triangulo {
    float xA, yA;
    float xB, yB;
    float xC, yC;
};
```

y aplíquese luego la regla de dividir cada `Triangulo` en tres, creando dos nuevos y modificando el original, de modo que en cada triángulo nuevo permanezca uno de los tres vértices del de partida.

13.2. EJEMPLOS Y EJERCICIOS EN C

13-21

Guárdense las coordenadas de cada triángulo en cada línea del archivo de texto (igual que en el ejemplo anterior se guardaban los intervalos): $(x_A; y_A; x_B; y_B; x_C; y_C)$. Represéntense los puntos con Gnuplot. Bastará hacerlo con los primeros vértices, que vendrán dados por las coordenadas $(x_A; y_A)$ de las dos primeras columnas del archivo (aunque se logrará más resolución empleándolas todas).

Una manera alternativa de generar el triángulo de Sierpinski es mediante el método XOR (en un tema posterior, veremos el significado de este método). Este método genera un conjunto de puntos que se pueden relacionar con los (centros de los) triángulos que existen en cada iteración del proceso geométrico anterior. Sin embargo, a diferencia de los métodos anteriores que construían todo el conjunto fractal al mismo tiempo, aumentando la resolución, el método XOR construye el triángulo de Sierpinski “línea a línea”. La idea es la siguiente:

- Se parte de una matriz bidimensional con todos sus elementos puestos a cero.

Nota: En C la forma más simple de hacer esto (sin añadir ningún detalle más como la posición de los puntos, que no es necesaria aquí) es considerar un *array* de elementos booleanos: 0 y 1. Como en C no existe este tipo, tenemos dos opciones: elegir un tipo ya existente (por ejemplo el “*unsigned char*”, que es de los que menos ocupa en memoria) o bien hacer un *typedef* (a “*unsigned char*”, por insistir en este tipo de dato pequeño) que se llame *BOOL*.

- En la primera línea de la matriz se coloca una semilla (valor 1) que dará lugar al fractal. Por simetría esta semilla conviene colocarla en la mitad de la fila.
- Para construir la siguiente línea del conjunto se recorre la siguiente fila de la matriz y se asigna a cada elemento booleano el XOR de los elementos correspondientes al anterior y posterior en la línea anteriormente construida (que en el primer paso sólo tendrá un punto distinto de 0, el elegido en la primera línea).

Nota: XOR significa “eXclusive OR”, esto es, la disyunción lógica “O pero no Y”. La tabla de verdad de esta operación lógica es la siguiente:

XOR	0	1
0	$0 \wedge 0 = 0$	$0 \wedge 1 = 1$
1	$1 \wedge 0 = 1$	$1 \wedge 1 = 0$

En C, los operadores del álgebra booleana se construyen con un único carácter, así “O” es “|”, “Y” es “&”, “O pero no Y” es “~”. Estos operadores actúan al nivel de los bits de los números; por eso aquí emplearemos 0 (todos sus bits son cero) y 1 (todos sus bits son cero menos uno, que es 1).

En la figura 13.19 se ilustra cómo se genera con este método el triángulo de orden $k = 3$. Obsérvese que si queremos construir la k -ésima generación del fractal, la matriz deberá tener las dimensiones $2^k \times (2 \times 2^k - 1)$.

El listado 13.2 muestra el código de este programa que implementa el algoritmo para generar el triángulo de Sierpinski.

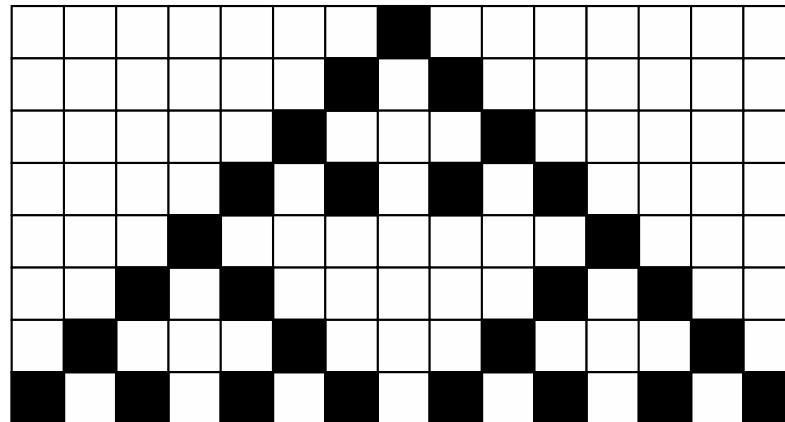


Figura 13.19: Triángulo de Sierpinski crecido a partir de una semilla en una matriz usando el método XOR.

Listado 13.2: El programa que genera el conjunto de Sierpinski mediante el método XOR.

```

1 #include <stdio.h>
2
3 #define ANCHURA 1024*2-1
4 #define ALTURA 1024
5
6 typedef unsigned char BOOL;
7
8 #define ON      (BOOL)1
9 #define OFF     (BOOL)0
10
11 int main(int argc, char** argv)
12 {
13     BOOL spky[ALTURA][ANCHURA];
14     int i, j;
15
16     for(i=0; i<ALTURA; i++)
17     {
18         for(j=0; j<ANCHURA; j++)
19         {
20             spky[i][j]=OFF;
21         }
22     }
23
24     spky[0][(ANCHURA-1)/2]=ON;
25
26     for(i=1; i<ALTURA; i++)
27     {
28         for(j=1; j<ANCHURA-1; j++)
29         {
30             spky[i][j]=
31                 ( spky[i-1][j-1] ^ spky[i-1][j+1] );
32         }
33     }
34 }
```

```

32         }
33     }
34
35     for(i=0; i<ALTURA; i++)
36     {
37         for(j=0; j<ANCHURA; j++)
38         {
39             if( spky[i][j] )
40                 printf("%d\t%d\n", j,i);
41         }
42     }
43
44     return 0;
45 }
```

- Nótese que los puntos que están en los extremos de las filas de la matriz (columna $j=0$ y columna $j=ANCHURA-1$) no son recorridos por el bucle ya que no tienen vecinos a izquierda o derecha, respectivamente.

13.2.2.1. Cálculo de la dimensión fractal por el método “*sandbox*”

La forma autosimilar del triángulo de Sierpinski, cuyos puntos acabamos de generar línea a línea, se puede aprovechar para hacer uso de una definición alternativa de la dimensión fractal: la dimensión másica o de “*sandbox*”. A pesar de su nombre, casi siempre coincide con la dimensión fractal de “*box-counting*” (igual que ésta coincide con la dimensión fractal de Haussdorf, que es la que matemáticamente está mejor establecida).

La dimensión fractal másica busca ajustar la siguiente relación entre el tamaño lineal r de un fractal y su masa $m(r)$:

$$m(r) \sim r^{D_{FM}}$$

Obsérvese que para una figura plana “sólida” (sin agujeros), como un triángulo, esta relación nos proporciona $D_{FM} = 2$ (la masa, proporcional al área, proporcional, a su vez, al cuadrado del tamaño del triángulo, descrito, por ejemplo, por su altura), que corresponde al concepto de bidimensionalidad que asociamos al triángulo.

Ahora veremos cuál es la dimensión fractal másica del triángulo de Sierpinski. Para ello asignaremos masa unidad a cada elemento activo de la matriz, y consideraremos que el tamaño lineal del fractal (r) viene dado por el número de líneas del mismo que se hayan construido, comenzando desde la línea donde se plantó la semilla. Así, simplemente, contaremos, para cada línea, el número total de elementos con valor 1 activados en la matriz, desde la semilla del fractal, hasta finalizar el cálculo de la fila r en cuestión, tal como se ilustra en la figura 13.20.

Ejercicio 13.5. Representar la masa frente a la altura del triángulo de Sierpinski.

Representar los datos con Gnuplot en coordenadas log-log (doble logarítmicas).

La pendiente de la recta resultante es la dimensión fractal buscada.

En Gnuplot se puede obtener esta pendiente a partir de los datos (guardados en un archivo “*sierpinskixordf.dat*” mediante el siguiente comando:

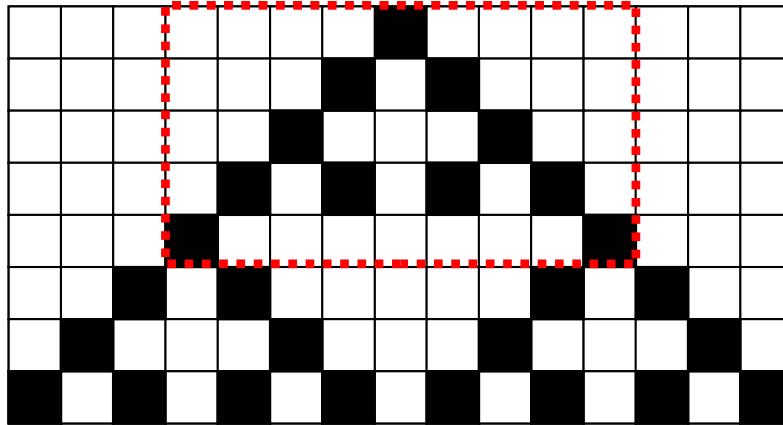


Figura 13.20: Método de “sandbox” en el triángulo de Sierpinski. Destacada, la región de tamaño $r = 5$, que da una masa $m(5) = 11$.

```
fit M0+Df*x 'sierpinskixordf.dat' u ($log($1)):(log($2)) via M0, Df
```

El comando “fit” de Gnuplot proporciona un ajuste por mínimos cuadrados de la expresión $M0+Df*x$ a los datos (que se leen del archivo y se convierten en sus logaritmos); el valor de Df proporcionado debería ser próximo a 1,6. Para comprobar la bondad del ajuste, se puede hacer

```
plot M0+Df*x, 'sierpinskixordf.dat' u ($log($1)):(log($2))
```

Aunque el ajuste por mínimos cuadrados de Gnuplot es muy completo y flexible, nosotros ya tenemos un programa que calcula regresiones lineales (ejercicio en la sección 11.10), y las regresiones a leyes de potencias se pueden convertir en regresiones a rectas, como acabamos de ver, sin más que aplicar a las magnitudes relacionadas por la ley de potencias la función logaritmo, así:

$$m(r) = m_1 \times r^{D_{FM}}$$

se convierte en

$$\log m(r) = \log m_1 + D_{FM} \times \log r$$

Ejercicio 13.6. Modificar la función de regresión lineal del ejercicio de la sección 11.10 para que ajuste una ley de potencias. Integrar esta función en el programa del ejercicio anterior para que, desde el mismo programa, se calcule la dimensión fractal del triángulo de Sierpinski.

Compárese el resultado obtenido por este procedimiento con el obtenido con el método de ajuste iterativo de Gnuplot. ¿Cuál es mejor? o ¿tiene sentido esta pregunta?