Trabajando fractales con winlogo

Sonia Sabogal
Universidad Industrial de Santander
Santander, Colombia
ssabogal@uis.edu.co

Gilberto Arenas
Universidad Industrial de Santander
Santander, Colombia
garenasd@uis.edu.co

Resumen

Después de una breve introducción en la cual se establecerán algunos conceptos teóricos básicos de la geometría fractal, se realizarán talleres en los cuales, con ayuda de las herramientas que trabaja el software WinLogo, se construirán diversos fractales, analizando sus principales características (autosimilitud, dimensión, etc.)

1. Introducción

Existen varias razones por las cuales la geometría fractal llama la atención: en primer lugar por la vistosidad y belleza de las figuras que se estudian en ella; en segundo lugar constituye, comparándola con la geometría euclidiana clásica, una geometría más cercana a la naturaleza. Al respecto veamos lo que dice quien es considerado "padre de los fractales", el matemático Benoit Mandelbrot (1924–): "*Postulo que muchos de los patrones de la naturaleza son tan irregulares y fragmentarios que, comparándolos con la geometría euclidiana, muestran no solo un más alto grado, sino un muy distinto nivel de complejidad. El número de las distintas escalas de extensión de patrones naturales es prácticamente infinito. La existencia de estos patrones nos reta a estudiar esas formas que la geometría euclidiana deja de lado por carecer de forma para investigar, por así decirlo, la morfología de lo amorfo. Respondiendo a este reto, he concebido y desarrollado una nueva geometría de la naturaleza, cuyo uso puede implementarse en diversos campos. Describe muchos de los patrones irregulares a nuestro alrededor, identificando una familia de formas a las que llamaré fractales.*"

Otra razón por la cual los fractales son interesantes, la constituyen las innumerables aplicaciones que se les vienen encontrando no solo en diversas ramas de la ciencia sino también en las artes: en biología, química, física, geología, ingeniería, medicina, diseño, arquitectura, música, pintura, etc.

Mencionaremos una última razón: siendo la geometría fractal un área relativamente joven de la matemática, proporciona una buena cantidad de problemas abiertos, lo cual de por sí es atractivo para cualquiera que se precie de ser un amante de la matemática.

Podríamos decir que existen tres caminos por donde uno se puede aproximar a la teoría de los fractales: el camino de los sistemas iterados de funciones (los fractales son autosemejantes), el camino de la teoría de la medida y la dimensión (los fractales tienen dimensión

"extraña") y a través de la teoría de los sistemas dinámicos (los fractales como conjuntos de Julia). El presente cursillo corresponde a una exposición introductoria y divulgativa sobre los fractales, mediante una aproximación a través del camino de los sistemas iterados de funciones, estableciendo una definición formal de la noción de autosemejanza en el contexto de los espacios métricos y desarrollando unos talleres que hacen uso del programa computacional WinLogo. Pretendemos inquietar y motivar a los participantes, a incursionar en este fascinante campo de las matemáticas.

El programa WinLogo, y el lenguaje Logo en general, resulta particularmente apropiado para trabajar fractales, puesto que una de sus principales características es la *recursividad*, la cual es también característica esencial del proceso de construcción de fractales mediante sistemas iterados de funciones. Por otra parte el lenguaje LOGO fue diseñado específicamente para la enseñanza y sus comando están en español. La versión que se utiliza en los talleres que presentamos en este cursillo, se denomina WINLOGO para WINDOWS y como su nombre lo indica, se trata de una versión diseñada para correr bajo Windows, siguiendo la misma filosofía que éste para el uso de las ventanas, las herramientas y el ratón.

Para quien esté interesado en éste lenguaje de programación, podemos sugerir ver las referencia [11, 12].

El cursillo se desarrolla en tres sesiones: en la primera se presentan los conceptos y resultados teóricos básicos, y en las dos sesiones siguientes se realizan dos talleres en los cuales, con la ayuda de WinLogo se construyen y analizan diversos fractales. Para finalizar esta introducción queremos transcribir a continuación lo que afirma M. Barnsley, (uno de los pioneros en formalizar la geometría fractal y uno de los autores más referenciados en este campo), en la introducción a su libro *Fractals Everywhere* [2]:

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

2. La formalización matemática

Las demostraciones de los teoremas y afirmaciones que usamos en esta sección se pueden encontrar en [1] y [2].

Definición 1 (Espacios métricos). *Un* ***espacio métrico*** *es un conjunto M, no vacío, de objetos (que llamaremos* ***puntos****) dotado de una función*

$$\begin{array}{rccc} d: & M \times M & \longrightarrow & \mathbb{R} \\ & (x,y) & \longmapsto & d(x,y) \end{array}$$

(llamada ***métrica*** *o* ***distancia*** *en el espacio) que satisface los dos siguientes axiomas:*

M1: $\forall x, y \in M : d(x,y) = 0 \Longleftrightarrow x = y.$

M2: $\forall x, y, z \in M : d(x,y) \leq d(x,z) + d(y,z)\,.$

Fácilmente se verifica que, no importa cuáles sean los puntos $x, y, z \in M$*, se cumple* $d(x,y) \geq 0$*,* $d(x,y) = d(y,x)$ *y* $d(x,y) \leq d(x,z) + d(z,y)$*.*

Ejemplo.

1. *El conjunto de los números reales con la distancia definida por* $d_u(x,y) = |x-y|$ *(distancia usual o euclidiana sobre* $\mathbb{R}$*) forma el espacio métrico* $(\mathbb{R}, d_u)$*.*

2. *Es claro que sobre un conjunto podemos definir muchas distancias, por ejemplo, sobre el conjunto de los números reales podemos definir la distancia*

$$d_{\ln}(x,y) = \ln(1 + |x-y|);$$

no es difícil verificar que $(\mathbb{R}, d_{\ln})$ *es un espacio métrico.*

3. *El conjunto de parejas reales ordenadas* $\mathbf{x} = (x_1, x_2)$ *(o* $\mathbb{R}^2$*) con la distancia*

$$d_u((x_1,x_2),(y_1,y_2)) = \sqrt{(y_1-x_1)^2 + (y_2-x_2)^2},$$

llamada distancia usual o euclidiana sobre $\mathbb{R}^2$*, se denomina* espacio euclídeo de dos dimensiones $\mathbb{R}^2$*,* $(\mathbb{R}^2, d_u)$*.*

4. *Tomando para los elementos de un conjunto arbitrario* X*,*

$$d_{disc}(x,y) = \begin{cases} 0, & si \;\; x = y, \\ 1, & si \;\; x \neq y, \end{cases}$$

obtendremos, evidentemente, que (X, d_{disc}) *es un espacio métrico denominado* espacio métrico discreto*.*

Recordemos que una **sucesión** en X es la imagen de una función $s : \mathbb{N} \to X$. Estos es, $s(X) = \{s(1), s(2), \cdot, \cdot, s(n), \cdot, \cdot\}$. En adelante, una sucesión en X puede considerarse como una lista ordenada de elementos de X, escritos en la forma $(s_1, s_2, \cdots, s_n, \cdots)$, con $s_n = s(n)$ para cada $n \in \mathbb{N}$. Por brevedad, se usa la notación para sucesiones de $\{s_n\}$ (o $\{s_n\}_{n=1}^{\infty}$ si es necesario enfatizar en cual es la variable).

Sea $s = \{s_n\}$ una sucesión infinita, y sea k una función cuyo domino es el conjunto de los enteros positivos y cuyo recorrido es un subconjunto del conjunto de los enteros positivos. Supongamos que k conserva el orden, o equivalentemente, es creciente, esto es, $k(m) < k(n)$ si $m < n$. La función compuesta $s \circ k$ está definida para todo entero $n \geq 1$, y para cada uno de tales n se tiene $(s \circ k)(n) = s_{k(n)}$. Esta función se llama una

subsucesión de s. Por efectos de brevedad, a menudo se utiliza la notación $\{s_{k(n)}\}$ o $\{s_{k_n}\}$ para designar la subsucesión de $\{s_n\}$ cuyo término n-ésimo es $\{s_{k(n)}\}$.

Por ejemplo, $s_{k(n)} = \{1/3^n\}$ es una subsucesión de $s_n = \{1/n\}$. En este caso $k(n) = 3^n$.

Con el fin de precisar las nociones de compacidad y continuidad es necesario definir la noción de convergencia en un espacio métrico. (Decimos que $\{x_n\}$ es una sucesión en un espacio métrico (X, d) si esta es una sucesión en el conjunto X).

Definición 2. *Sea X un espacio métrico y $\{x_n\}_{n\in\mathbb{N}}$ una sucesión en X. La sucesión $\{x_n\}_{n\in\mathbb{N}}$ se dice que **converge** a $x \in X$ (o la sucesión $\{x_n\}$ es **convergente**), simbólicamente $x_n \to x$, si dado $\epsilon > 0$, existe $N \in \mathbb{N}$ tal que si $n \geq N$, entonces $d(x_n, x) < \epsilon$.*

Note que si una sucesión x_n converge a x en X, entonces dado dado $\epsilon > 0$, existe $N \in \mathbb{N}$ tal que si $n \geq N$, entonces $d(x_n, x) < \epsilon/2$. Por tanto, de la desigualdad triangular, si $n, m \geq N$, se sigue que $d(x_n, x_m) < \epsilon$.

En adelante diremos que una sucesión $\{x_n\}_{n\in\mathbb{N}}$ es una **sucesión de Cauchy** si se satisface la propiedad anterior, es decir, si dado $\epsilon > 0$, existe $N \in \mathbb{N}$ tal que si $n, m \geq N$, se tiene que $d(x_n, x_m) < \epsilon$.

Es importante señalar que toda sucesión convergente es de Cauchy como ya se demostró, pero no toda sucesión de Cauchy es convergente.

Por ejemplo: $\{1,\ 1.4,\ 1.41,\ 1.412,\ \ldots\}$ es de Cauchy pero no converge en $\mathbb{Q}$. Otro ejemplo lo constituye la sucesión $\{1/n\}$, es de Cauchy pero no converge en $(0, 1)$ con la métrica usual.

A los espacios métricos X en los cuales toda sucesión de Cauchy sea convergente los llamaremos espacios **completos**.

Teorema 3. *Suponga que $\{x_n\}_{n\in\mathbb{N}}$ es una sucesión convergente en un espacio métrico (X, d). Entonces:*

(i) el límite $x = \lim_{n\to\infty} x_n$ es único;

(ii) cualquier subsucesión de $\{x_n\}_{n\in\mathbb{N}}$ converge también a x;

(iii) $\{x_n\}_{n\in\mathbb{N}}$ es una sucesión de Cauchy.

Definición 4 (Conjunto compacto). *Un conjunto $A \subset (X, d)$ es **compacto** si toda sucesión en A contiene una subsucesión convergente en A.*

En el caso del espacios métricos $\mathbb{R}^n$ la caracterización más importante es la siguiente:

Teorema 5 (Heine-Borel, caracterización de compactos en $\mathbb{R}^n$). *Sea $A \subset \mathbb{R}^n$. A es compacto si y solo si A es cerrado y acotado.*

Definición 6. *Sean (X, d_1) y (Y, d_2) espacios métricos y $f : X \to Y$. Sea $a \in X$. Diremos que f es **continua** en a, si para todo $\epsilon > 0$ existe $\delta > 0$ tal que siempre que $d_1(x, a) < \delta$ entonces $d_2(f(x), f(a)) < \epsilon$. Diremos que f es continua en X (o simplemente continua), si f es continua en a, para todo $a \in X$.*

Definición 7. *Sea (X, d) un espacio métrico. Una función $f : X \longrightarrow X$ se dice que es una contracción si existe $r \in [0, 1)$ tal que $d(f(x), f(y)) \leq r\, d(x, y)$ para todo $x, y \in X$.*

Teorema 8. *Toda contracción es continua.*

El siguiente teorema, llamado *Teorema del punto fijo para espacios métricos completos*, es de gran importancia para la geometría fractal, pues es justamente el resultado que va a garantizar la existencia y unicidad de cada fractal obtenido por medio de un sistema iterado de funciones.

Teorema 9 (del punto fijo). *Sean M un espacio métrico completo y $f : M \to M$ una contracción de M. Entonces f tiene un único punto fijo, es decir, existe un único $p \in M$ tal que $f(p) = p$. Además, para cualquier $x \in M$ se tiene que:*

$$\lim_{n\to\infty} f^{\circ n}(x) = p.$$

Bosquejo de la demostración. Sea $x \in M$. Puesto que f es una contracción, se puede demostrar que la sucesión

$$(f^{\circ n}(x)) = \left(x, f(x), f^{\circ 2}(x), \ldots, f^{\circ n}(x), \ldots\right)$$

es una sucesión de Cauchy (intuitivamente, cada vez que se aplica la función, las distancias se van reduciendo de modo que los términos de la sucesión se van acercando cada vez más entre si). Como M es completo la sucesión converge, es decir existe un $p \in M$ tal que

$$\lim_{n\to\infty} f^{\circ n}(x) = p.$$

Entonces:

$$f\left(\lim_{n\to\infty} f^{\circ n}(x)\right) = f(p),$$

y como f es continua (por ser contracción) se tiene

$$\lim_{n\to\infty} f\left(f^{\circ n}(x)\right) = f(p),$$

luego

$$\lim_{n\to\infty} f^{\circ n}(x) = f(p),$$

es decir: $p = f(p)$ de modo que p es punto fijo de f. Ahora si q fuese otro punto fijo de f se tendría:

$$d(p, q) = d(f(p), f(q)) \leq r d(p, q)$$

siendo r factor de contracción de f. Entonces

$$d(p, q)(r - 1) = 0$$

y puesto que $0 \leq r < 1$ se concluye que $d(p, q) = 0$. □

2.1. Transformación afín en el plano

Una transformación afín en el plano es un función $\boldsymbol{\omega} : \mathbb{R}^2 \to \mathbb{R}^2$ de la forma

$$\boldsymbol{\omega}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \qquad \text{(Notación Cartesiana)}$$

donde a, b, c, d, B_1 y B_2 son constantes reales. Esta presentación corresponde a coordenadas cartesianas. Una presentación equivalente, en coordenadas geométricas es la siguiente

$$\boldsymbol{\omega}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r\cos\theta & -s\,\mathrm{sen}\,\phi \\ r\,\mathrm{sen}\,\theta & s\cos\phi \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \qquad \text{(Notación geométrica)}$$

en donde (r, θ) y (s, ϕ) son las coordenadas polares de (a, c) y (b, d) respectivamente.

Una transformación afín es entonces una transformación lineal seguida de una traslación y produce sobre cada subconjunto del plano, cierto tipo de efectos geométricos, como: cambios de escala, rotaciones, reflexiones, traslaciones, entre otros. En esta notación los parámetros r y s corresponden a los cambios de escala en los sentidos horizontal y vertical respectivamente, además una reflexión respecto al eje y si $r < 0$, o respecto al eje x si $s < 0$. Los parámetros θ y ϕ corresponden a giros (alrededor del origen) de los ejes x e y respectivamente, y los parámetros B_1 y B_2 indican desplazamientos en los sentidos horizontal y vertical respectivamente.

Un tipo de transformaciones afines muy usado en geometría fractal son la llamadas similitudes que se obtienen cuando $\theta = \phi$ y $|r| = |s|$, es decir:

Las **similitudes** en el plano son transformaciones afines de la forma

$$\boldsymbol{\omega}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r\cos\theta & -r\,\mathrm{sen}\,\theta \\ r\,\mathrm{sen}\,\theta & r\cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix},$$

o de la forma

$$\boldsymbol{\omega}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r\cos\theta & r\,\mathrm{sen}\,\theta \\ -r\,\mathrm{sen}\,\theta & r\cos\theta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix},$$

donde r es el factor de escala y θ es el ángulo de rotación.

Una notación mas abreviada para las similitudes y que llamamos notación compleja, es

$$\boldsymbol{\omega}(z) = rz\exp(\boldsymbol{i}\theta) + B_1 + B_2\boldsymbol{i}, \qquad \text{o,} \qquad \boldsymbol{\omega}(z) = r\overline{z}\exp(\boldsymbol{i}\theta) + B_1 + B_2\boldsymbol{i}.$$

2.2. Contracciones en el plano

Una función $f : \mathbb{R}^2 \to \mathbb{R}^2$ se dice que es una contracción si existe una constante ρ, con $0 \leq \rho < 1$, tal que para todo par de puntos P y Q del plano,

$$d(f(P), f(Q)) \leq \rho d(P, Q),$$

donde $d(P,Q)$ denota la distancia entre los puntos P y Q.

Si en una similitud el factor de escala r satisface $|r| < 1$, entonces la similitud es una contracción.

2.3. Conjuntos autosimilares

Los conjuntos autosimilares son aquellos que se pueden ver como la unión de sus imágenes por medio de varias contracciones, así, un conjunto A del plano será autosimilar si existen contracciones en el plano, digamos $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$, ..., $\boldsymbol{\omega}_n$ tales que se tiene

$$\boldsymbol{\omega}_1(A) \cup \boldsymbol{\omega}_2(A) \cup \cdots \cup \boldsymbol{\omega}_n(A) = \bigcup_{k=1}^{n} \boldsymbol{\omega}_k(A) = A.$$

De esta manera un conjunto autosimilar queda determinado por un número finito de contracciones que interactúan de determinada manera. Básicamente esto es lo que M. Barnsley llama en [2], un sistema iterado de funciones (SIF); allí (capítulo 3) explica detalladamente la construcción de fractales en espacios métricos, mediante sistemas iterados de funciones, demuestra rigurosamente la existencia y unicidad del conjunto fractal (atractor), para cada SIF. La idea básica consiste en tomar un compacto K cualquiera (no vacío), del plano y aplicarle las contracciones del SIF para obtener $K_1 = \boldsymbol{\omega}_1(K) \cup \cdots \cup \boldsymbol{\omega}_n(K)$; luego repite el proceso sobre el K_1 obtenido para tener $K_2 = \boldsymbol{\omega}_1(K_1) \cup \cdots \cup \boldsymbol{\omega}_n(K_1)$, y así sucesivamente, formando entonces una sucesión de figuras: K_1, K_2, ..., cuyo límite A es el conjunto autosimilar, que en la mayoría de los casos es un conjunto de tipo fractal.

2.4. El hiperespacio $\boldsymbol{\mathcal{H}}(X)$

$\boldsymbol{\mathcal{H}}(X)$ está formado por todos los subconjuntos $A \subseteq X$ compactos y no vacíos, es decir se define:

$$\boldsymbol{\mathcal{H}}(X) =: \{A \subseteq X \mid A \text{ es compacto, } A \neq \emptyset\}.$$

Definiremos a continuación la "distancia" de un punto a un compacto, y la "distancia" de un compacto a otro, con el objetivo de, finalmente, obtener una métrica en $\boldsymbol{\mathcal{H}}(X)$.

Sean $A, B \in \boldsymbol{\mathcal{H}}(X)$, $a \in A$, se define:

- $\hat{d}(a,B) = \inf\{d(a,x) : x \in B\}$
- $\tilde{d}(A,B) = \text{máx}\{\hat{d}(a,B) : a \in A\}$
- $\boldsymbol{h}(A,B) = \text{máx}\{\tilde{d}(A,B), \tilde{d}(B,A)\}$

$\hat{d}(a,B)$ y $\tilde{d}(A,B)$ **no** definen una métrica sobre $\boldsymbol{\mathcal{H}}(X)$ (¿por qué?), mientras que $\boldsymbol{h}(A,B)$ si. Se tiene entonces el siguiente resultado:

Teorema 10. *Si (X,d) es completo, $(\boldsymbol{\mathcal{H}}(X), \boldsymbol{h})$ forma un espacio métrico completo.*

2.5. Sistemas iterados de funciones

Considere X un espacio métrico completo. Llamaremos sistema iterado de funciones (SIF) a una estructura $\{X; \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N\}$, donde cada $\boldsymbol{\omega}_i$ es una contracción de X. Al definir:

$$\begin{array}{rccl} W: & \boldsymbol{\mathcal{H}}(X) & \longrightarrow & \boldsymbol{\mathcal{H}}(X) \\ & K & \longmapsto & W(K) = \bigcup_{i=1}^{N} \boldsymbol{\omega}_i(K) \end{array} \tag{1}$$

es posible demostrar que W es una contracción de $\boldsymbol{\mathcal{H}}(X)$ y dado que este es un espacio completo podemos aplicar el teorema del punto fijo (Teorema 9) para obtener el siguiente resultado:

Teorema 11. *Dado un SIF $\{X; \boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N\}$ existe un único $A \subseteq X$, A compacto, $A \neq \emptyset$ tal que*

$$A = W(A) = \bigcup_{i=1}^{N} \boldsymbol{\omega}_i(A);$$

*este conjunto A se llama el **atractor**, del SIF. Además, para cualquier $K \in \boldsymbol{\mathcal{H}}(X)$ se tiene:*

$$\lim_{n\to\infty} W^{\circ n}(K) = A.$$

Ejemplo. *Consideremos el SIF $\{\mathbb{R}^2; \boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \boldsymbol{\omega}_3\}$, donde*

$$\boldsymbol{\omega}_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \qquad i = 1, 2, \ldots$$

siendo a, b, c, d, B_1, B_2 los valores dados en la tabla.

$$W(K) = \boldsymbol{\omega}_1(K) \cup \boldsymbol{\omega}_2(K) \cup \boldsymbol{\omega}_3(K)$$

i	a	b	c	d	B_1	B_2
1	0.5	0	0	0.5	0	0
2	0.5	0	0	0.5	0.5	0
3	0.5	0	0	0.5	0.5	0.5

Finalizamos esta parte teórica del cursillo, con una definición, ya bien formal desde el punto de vista matemático, de la noción de autosemejanza:

Definición 12. *Diremos que un conjunto es **autosemejante** (o autosimilar) si es el atractor de un SIF.*

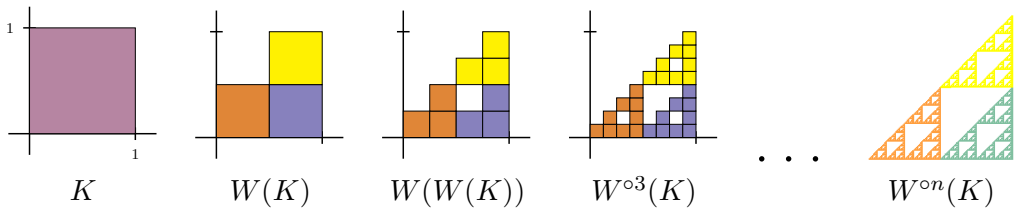


Figura 1. Construcción del triángulo de Sierpiński

2.6. Ejemplos

Veamos a continuación dos ejemplos clásicos, el primero es equivalente al analizado antes, pero se parte de una figura inicial distinta. En el ejemplo visto antes se parte de un cuadrado y en el que vemos a continuación se parte de un triángulo, sin embargo, el conjunto límite es el mismo. Esto ilustra lo afirmado en el Teorema 11 en relación con que el atractor de un SIF es independiente del compacto K elegido como "semilla" o conjunto inicial.

El Triángulo de Sierpiński

Consideremos las contracciones definidas por:

$$\boldsymbol{\omega}_1\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}1/2 & 0\\0 & 1/2\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}; \qquad \boldsymbol{\omega}_2\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}1/2 & 0\\0 & 1/2\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}1/2\\0\end{pmatrix};$$

$$\boldsymbol{\omega}_3\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}1/2 & 0\\0 & 1/2\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0\\1/2\end{pmatrix}.$$

Si se parte por ejemplo de $K = \{(x, y) \in \mathbb{R}^2 : 0 \leq x \leq 1,\ 0 \leq y \leq x\}$ (región triangular de vértices $(0, 0)$, $(1, 0)$ y $(1, 1)$) al aplicar conjuntamente las tres contracciones al triángulo K, se obtiene la figura K_1 formada por tres triángulos; volvemos a aplicar las tres contracciones a K_1 y obtenemos K_2 formada por nueve triángulos y así sucesivamente, la figura límite que se obtiene, se llama Triángulo de Sierpiński.

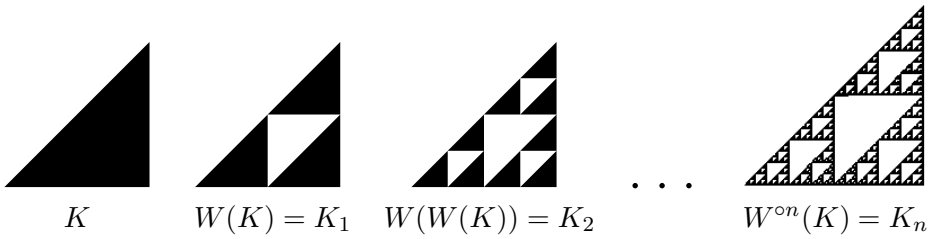


Figura 2. Triángulo de Sierpiński.

La curva de Koch

Para obtener esta figura debemos considerar las siguientes contracciones que presentamos en notación compleja:

$$\boldsymbol{\omega}_1(z) = \frac{1}{3}z; \qquad \boldsymbol{\omega}_2(z) = \frac{1}{3}z\exp\left(\boldsymbol{i}\frac{\pi}{3}\right) + \frac{1}{3};$$
$$\boldsymbol{\omega}_3(z) = \frac{1}{3}z\exp\left(-\boldsymbol{i}\frac{\pi}{3}\right) + \frac{1}{2} + \frac{\sqrt{3}}{6}; \qquad \boldsymbol{\omega}_2(z) = \frac{1}{3}z + \frac{2}{3}.$$

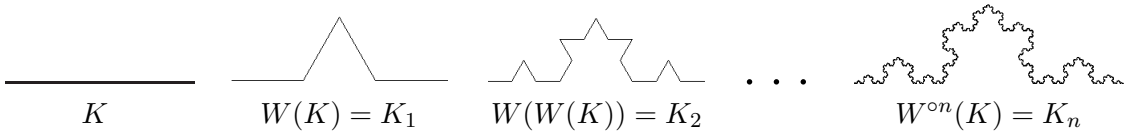


Figura 3. Descripción de la construcción de la *curva de Koch.*

El conjunto autosimilar que se obtiene se llama curva de Koch (ver Figura 3).

De esta manera se puede construir muchos conjuntos de tipo fractal. Otros ejemplos similares se pueden encontrar en [2], [4] o [7], entre otros.

3. Talleres

Taller 1: Manejo básico de WinLogo

1. Imagine que en la pantalla hay un tablero donde vamos a dibujar, y para esto disponemos de una pequeña tortuga que vive en el tablero de la pantalla, tiene lápices de colores y se caracteriza por ser muy obediente. Para limpiar el tablero y hacer que la tortuga aparezca, digite las letras **bp** (borrar pantalla), luego oprima la tecla enter.

2. La tortuga es capaz de obedecer órdenes expresadas en el lenguaje denominado "idioma de la tortuga". La orden **avanza** hace que la tortuga se mueva en línea recta en la dirección hacia donde mira; para indicarle cuanto debe avanzar, la orden ***av*** (avanza) debe ir seguida de un número que corresponda a los pasos que avanzará. Por ejemplo: dígale a la tortuga que avance 30 de sus pasos, con la instrucción:

   Avanza 30 o ***av*** 30

   Ella cumple la orden al oprimir la tecla enter.

3. Ensaye: borre la pantalla y dígale a la tortuga que avance, con diferentes números de pasos.

4. Otra orden que obedece la tortuga es ***re*** (retrocede) la cual le indica que debe caminar hacia atrás, como en la orden anterior se requiere informarle con un número

la cantidad de pasos que debe retroceder. Por ejemplo: **Retrocede** 60 (enter) o ***re*** 60 (enter).

5. Ensaye: borre la pantalla y pídale a la tortuga que avance al borde inferior de la misma y posteriormente al borde superior.

6. Ahora veamos otras dos órdenes que la tortuga acepta, las cuales modifican la dirección en que mira la tortuga sin afectar su ubicación; **giraderecha**, **giraizquierda**, hacen que la tortuga vuelva su mirada en otra dirección. Igual que en avanza una orden de giro debe contener un número que indique el ángulo que debe girar la tortuga (medido en grados). Por ejemplo:

   Giraderecha 45 (enter) o ***gd*** 45 (enter).

   La tortuga gira, sin cambiar de posición 45 grados hacia la derecha.
   Dígale ahora

   giraizquierda 60 (enter) o ***gi*** 60 (enter).

   Antes de continuar tenga en cuenta las siguientes observaciones: es posible que al comunicarnos con la tortuga se cometan errores tales como: Avanza50 (enter), por el cual el computador le responde (al lado derecho de la parte inferior de la pantalla): "No sé cómo hacer Avanza50", o: Av (enter) y de la misma forma el computador le responderá: "Faltan datos para Av".

7. Digite las siguientes instrucciones y observe. (No olvide oprimir la tecla enter al final de cada una)

   - bp
     gd 90
     av 80

   - bp gd 90 av 80

   - bp
     gd 150
     re 70
     av 90
     gi 70

   - bp
     gi 60
     av 120
     gd 80
     re 50
     gd 20
     av 70

8. Con estas instrucciones podemos trazar muchas figuras, por ejemplo:

   - bp av 20 gd 90 av 20 gi 90 av 20 gd 90 av 20 gi 90 av 20 gd 90 av 20
     ¿Qué figura se obtiene en la pantalla?

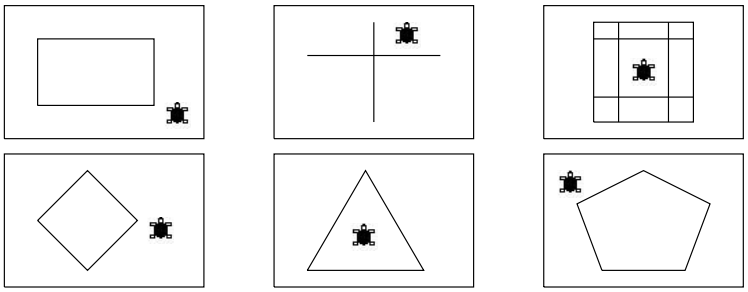
   - bp av 40 gd 90 av 40 gd 90 av 40 gd 90 av 40
     ¿Qué figura se obtiene en la pantalla?

   - bp gd 30 av 40 gd 120 av 40 gd 120 av 40

¿Qué figura se obtiene en la pantalla?

9. Cuando necesite que la tortuga avance sin dejar trazo, utilize la orden ***sl*** (sube lápiz) y no olvide para cuando ya desee el trazo, decirle ***bl*** (baja lápiz). Ensaye esas instrucciones.

10. Dibuje cada una de las gráficas siguientes: (no olvide primero limpiar la pantalla)



11. Digite las siguientes instrucciones: bp gd 90 av 30 sl av 15 bl av 10 sl av 10 bl av 10

    ¿Qué figura observa?

12. Intente darle a la tortuga las órdenes necesarias para dibujar un cuadrilátero regular (un cuadrado) y un triángulo equilátero.

    Observe: ¿algunas órdenes se repiten varias veces?, ¿qué órdenes se repiten y cuántas veces?

    Repite es un nuevo comando que le va a ayudar a dibujar polígonos regulares, de manera fácil y rápida. Aquí tiene un ejemplo de cómo se usa este nuevo comando.

    repite n [**com1 arg1 com2 arg2** etc. . . .]

    El comando **repite** necesita tener corchetes. Los corchetes contienen los comandos y los argumentos que hemos de repetir. (Recuerde que el comando es el **qué hacer** y el argumento es **cuántas veces**). Intente usar este nuevo comando para hacer diferentes polígonos regulares (triángulo, cuadrado, pentágono, hexágono, etc.).

13. Hasta ahora usted ha trabajado en el modo de "comandos directos". Ha llegado el momento de que le enseñe a la tortuga nuevos procedimientos que ella pueda recordar. Por ejemplo, hacer un polígono de n lados de una longitud de lado ℓ. Para ello necesitará dos nuevos comandos: **para** *nombre del procedimiento* `opciones`, para iniciar el modo de enseñanza, y **fin** para acabar con la lección.

    a) Escriba "**para** *poligono* `:n :lado`" y pulse enter.

    b) Escriba sus comandos para hacer un polígono, por ejemplo:

    repite :n [av :lado gd 360/:n]

    y pulse enter.

c) Ahora escriba **fin** para acabar la lección de su tortuga y pulse enter.
La **ventana de trabajo** mostrará un mensaje diciendo "poligono definido".

d) Escriba **poligono 6 100** y pulse enter, ¿qué dibuja la tortuga?

e) Haga que la tortuga dibuje un pentágono regular y un eneágono regular.

Para finalizar le presentamos una tabla para que recuerde algunos de los comandos básicos del "idioma de la tortuga".

PRIMITIVA	CÓDIGO	DESCRIPCIÓN
Borrar pantalla	**bp**	Borra todo lo que esté en el área de gráficos
Avanza n	**av *n***	Adelante, mueve la tortuga n pasos hacia delante
Retrocede n	**re *n***	Retrocede, mueve la tortuga n pasos hacia atrás
Gira a la derecha n	**gd *n***	Gira la tortuga hacia la derecha n grados
Gira a la izquierda n	**gi *n***	Gira la tortuga hacia la izquierda n grados
Baja lápiz	**bl**	Baja la pluma
Sube lápiz	**sl**	Sube la pluma
Goma	**goma**	Cambia la pluma por el borrador
Oculta la tortuga	**ot**	Hace invisible la tortuga
Muestra la tortuga	**mt**	Hace visible la tortuga
Repite	**Repite *n*** [. . .]	Permite repetir n veces una lista de instrucciones

Taller 2: Creando fractales con WinLogo

1. Inventar una semilla y una producción que generen una figura de tipo fractal. Llamaremos también a la semilla el "paso 0" y a la producción el "paso 1" del proceso de construcción del fractal.

2. Escribir las instrucciones en WINLOGO para graficar el "paso 1" de los siguientes fractales:

   - La curva de Koch.
   - El triángulo de Sierpiński.
   - La isla de Koch.
   - La carpeta de Sierpiński.
   - El fractal que usted ideó en el punto 1.

3. A continuación se encuentran programas en lenguaje LOGO que generan la curva de Koch y otros fractales. Cópielos en el computador (no es muy importante por ahora entender la estructura del programa, ya que para esto se necesitan instrucciones del lenguaje LOGO más avanzadas); y ¡deléitese observando la forma como la tortuga construye el fractal!

⊛ **Programa 1: *La curva de Koch***

```
para koch :paso :lado
si :paso = 0 [av :lado alto]
koch :paso-1 :lado/3  gi 60
koch :paso-1 :lado/3  gd 120
koch :paso-1 :lado/3  gi 60
koch :paso-1 :lado/3
fin
```

Este programa trabaja con dos variables **:paso** y **:lado**. Para ejecutar el programa debe "llamarlo" e indicar dos valores determinados para las variables; por ejemplo digite:

koch 0 100

y oprima la tecla enter. ¿Qué dibuja la tortuga?

Siga ensayando con los siguientes u otros valores arbitrarios que usted elija:
koch 1 100; **koch 2 150;** **koch 5 300;** **koch 7 250.**

Observando lo que dibuja la tortuga en cada caso, explique el significado de las variables **:paso** y **:lado**.

⊛ **Programa 2: *Isla de Koch***

```
para isla :paso :lado
repite 3 [koch :paso :lado gd 120]
fin
```

Observe que para realizar este programa es necesario haber definido con anterioridad el programa que genera la curva de Koch. Repita el mismo proceso que se realizó en el programa 1, escogiendo valores arbitrarios:
isla 1 100; **isla 2 150;** **isla 5 300;** **isla 4 200.**

⊛ **Programa 3: *Pentágono***

```
para pentagono :paso :lado
haz "contraccion (suma 3 raizcuadrada 5)/2
si :paso = 0 [av :lado alto]
gi 36 pentagono :paso-1 :lado/:contraccion  gi 72
pentagono :paso-1 :lado/:contraccion  gd 144
pentagono :paso-1 :lado/:contraccion  gd 72
pentagono :paso-1 :lado/:contraccion  gi 72
pentagono :paso-1 :lado/:contraccion  gi 72
pentagono :paso-1 :lado/:contraccion  gd 36
fin
```

Repita el mismo proceso, escogiendo valores arbitrarios:
pentagono 1 100; **pentagono 2 150;**

pentagono 3 300; **pentagono 5 250.**

¿Qué dibuja la tortuga en cada caso?

⊛ **Programa 4: *Dragón de Sierpiński***

```
para dragon :paso :lado :signo
si :paso = 0 [av :lado alto]
gi 60*:signo
dragon :paso-1 :lado/2 :signo gd 60*:signo
dragon :paso-1 :lado/2 :signo gd 60*:signo
dragon :paso-1 :lado/2 :signo gi 60*:signo
fin
```

En este programa ":signo" toma los valores de -1 o 1. Repita el mismo proceso que se realizó en los programas anteriores. Interprete el significado de la variable ":signo".

⊛ **Programa 5**

El siguiente algoritmo genera un triángulo rectángulo y es auxiliar del programa principal.

```
para triangulo :lado
av :lado gd 90 av :lado gd 135 av :lado*(raizcuadrada 2) gd 135
fin
```

Programa principal:

```
para tri :paso :lado
si :paso=0 [triangulo :lado alto]
tri :paso-1 :lado/2 sl av :lado/2 bl
tri :paso-1 :lado/2 sl gd 90 av :lado/2 gi 90 bl
tri :paso-1 :lado/2 sl re :lado/2 gi 90 av :lado/2 gd 90 bl
fin
```

"Bautice" con un nombre apropiado el programa 5.

⊛ **Programa 6: *Triángulo de Sierpiński***

Se presenta primero un algoritmo auxiliar que hace un cuadrado de color azul y se usará en el programa principal.

```
para cuadrado :lado
poncl 5 repite 4 [av :lado gd 90]
sl av :lado/2 gd 90 av :lado/2 bl rellena
sl re :lado/2 gd 90 av :lado/2 gd 180 bl
fin
```

Programa principal:

```
para sierpinski :paso :lado
si :paso=0 [cuadrado :lado alto]
sierpinski :paso-1 :lado/2 gd 90 sl av :lado/2 gi 90 bl
sierpinski :paso-1 :lado/2 sl av :lado/2 bl
sierpinski :paso-1 :lado/2 sl re :lado/2 gi 90 av :lado/2 gd 90 bl
fin
```

Compare los programas 5 y 6. ¿Qué similitud y qué diferencias encuentra? Discuta con sus compañeros y escriba alguna conclusión.

⊛ **Programa 7:** ***Carpeta de Sierpiński***

Algoritmo auxiliar:

```
para cuadrorr :lado
poncl 5 repite 4 [av :lado gd 90] sl av :lado/3 gd 90 av :lado/3 gi 90 bl rellena
poncl 16 repite 4 [av :lado/3 gd 90] sl av :lado/9 gd 90 av :lado/9 bl rellena
sl re 4*:lado/9 gi 90 re 4*:lado/9 bl poncl 5
fin
```

Programa principal:

```
para carpeta :paso :lado
si :paso=0 [cuadrado :lado alto]
si :paso=1 [cuadrorr :lado alto]
repite 4 [carpeta :paso-1 :lado/3 cuadrorr :lado/3 av :lado/3
          carpeta :paso-1 :lado/3 cuadrorr :lado/3 av 2*:lado/3 gd 90]
fin
```

Observe que este programa hace uso de dos programas auxiliares: el programa ***cuadrado*** del programa 6 y el programa ***cuadrorr***. Explique cómo hace la tortuga para rellenar de color una figura y qué hace el programa ***cuadrorr***.

4. Escriba un programa que genere el fractal que usted inventó en el punto 1.

Bibliografía

[1] APOSTOL, T. *Análisis Matemático*. Barcelona España, Editorial Reverte, 1972.

[2] BARNSLEY, M. *Fractals Everywhere*. Academic Press, Inc. San Diego, 1988.

[3] BARNSLEY, M., DEMKO, S. *Iterated Function Systems and the Global Construction of fractals*. Proc. Roy. Soc. London, Ser. **A 399** 243–275, 1985.

[4] CASTRO, F. *Geometría fractal en el bachillerato*. Monografia de grado, Licenciatura en Matemática, UIS, 1994.

[5] DE GUZMÁN, M., MARTÍN, A., MORÁN, M. y REYES, M. *Estructuras fractales y sus aplicaciones*. Editorial Labor, S.A., Barcelona, 1993.

[6] FLOREZ, E. *Una reseña histórica de la geometría fractal*. Monografía de grado, Licenciatura en Matemáticas, UIS, Bucaramanga, 1995.

[7] Grupo Fractales. Departamento de Matemáticas – UIS, Taller de fractales, Material pre-impreso, Bucaramanga, 1993.

[8] MANDELBROT, B. *Los objetos fractales. Forma, azar y dimensión.* Tusquets Editores, S.A. Barcelona, 1993.

[9] MANDELBROT, B. *The fractal Geometry of Nature*. Freeman San Francisco, 1992.

[10] RUBIANO, G. N. *Fractales para profanos*. Universidad Nacional de Colombia, Editorial Unibiblos, Bogotá, 2002.

[11] WINLOGO, *Entorno de programación para el desarrollo y aprendizaje con lenguaje Logo:*
`http://www.wlogo.com/`

[12] El paraiso de Logo, *Página de aficionados al lenguaje Logo:*
`http://mondragon.angeltowns.net/paradiso`