

## **GENERADOR DE ATRACTORES DE IFS: GENIFS.EXE**

*Alberto Rodríguez Santos*

*Profesor de secundaria del Instituto San Cristóbal de los Ángeles,  
Madrid*

*e-mail: [alberto@epsilones.com](mailto:alberto@epsilones.com)*

## GENERADOR DE ATRACTORES DE IFS: GENIFS.EXE

### Resumen:

Los IFS son sistemas que mediante la iteración, es decir, la repetición de ciertos cálculos, sencillos pero fatigosos, dan lugar a imágenes interesantes de apariencia complicada. Nada hay más adecuado para ser automatizado, y esto es lo que hace el programa genifs.exe: automatizar la generación de atractores de IFS. Esta tarea se realiza en dos pasos. En el primero, con movimientos de ratón, se dibujan en pantalla un cierto número de paralelogramos. En el segundo, pulsando un botón, se genera el atractor.

**Palabras clave:** sistemas / automatizado / atractores / programa / iteración

### Abstrac:

The IFS are systems that by means of iteracin, that is to say, repeticin of certain clculos, simple but fatigosos, give rise imgenes interesting of complicated appearance. Nothing is ms suitable to be automated, and this is what genifs.exe makes the program: to automate generacin of atractores of IFS. This task is made in two steps. In first, with movements of ratn, they draw in screen a certain nmero of parallelograms. In the second, pressing botn, is generated the atractor.

**Keywords:** systems / automated / attractors / program / iteracin

## Inicio

Unos de los conjuntos fractales más conocidos son los atractores de IFS (*Iterated Function System*). Dos son las razones principales de esta popularidad. La primera y principal es la espectacularidad gráfica de algunos de dichos atractores. Otra es la sencillez de las ecuaciones que los generan, sencillez que es la base de la compresión fractal. Las siguientes notas son una introducción al programa genifs.exe (entorno Windows), con el cual se pueden generar estos sistemas con simples movimientos de ratón y visualizar sus atractores para el caso particular de IFS compuestos de aplicaciones afines.

Esta introducción se estructura en tres apartados y un apéndice. En el primero se repasa brevemente qué son los IFS; en el segundo se da una idea de cómo operar con el programa; y en el tercero se comenta el teorema del collage de Barnsley. El apéndice menciona algunas cuestiones acerca de las ecuaciones de las aplicaciones afines y explica el interés de los IFS para la compresión de imágenes. Finalmente se incluyen algunas direcciones Web relacionadas con el tema.

## IFS

Un IFS es un sistema finito de funciones contractivas. Cada sistema, considerado como una función sobre el conjunto de los compactos, también es una función contractiva y tiene, por tanto, un punto fijo, que es el atractor del sistema dinámico que resulta de iterar el sistema.

Tras este amenazador párrafo se esconde una idea sencilla. Tomemos un conjunto compacto cualquiera, un cuadrado, un círculo, la forma de la península de Crimea, y apliquémosle una función contractiva, es decir, una función que reduzca la distancia entre los puntos. Con ello obtendremos una imagen de la figura original en la que ha cambiado el tamaño y puede que también la forma, orientación y posición. En la figura, la función simbolizada por la flecha transforma el cuadrado negro en el rojo reduciendo el lado a la mitad.



Si aplicamos de nuevo la función sobre la figura obtenida y repetimos el proceso sucesivamente, es decir, iteramos, generaremos una sucesión de figuras cada vez más pequeñas que en el límite tenderá a un punto, que es el punto fijo de la función. Si consideramos que la función describe un sistema dinámico, es decir, un sistema que cambia con el tiempo (cada figura representa el estado del sistema descrito en un instante de tiempo), a este punto fijo se le llama atractor.

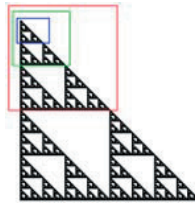


La cosa se pone más interesante si en vez de considerar una sola función consideramos varias, es decir, un sistema de funciones. Al aplicar el sistema sobre la figura original obtendremos varias imágenes, una por cada función, digamos  $n$ , sobre las que podremos aplicar de nuevo el sistema para obtener  $n^2$  imágenes, y después  $n^3$ , y así sucesivamente. La cuestión es que si antes, con una sola función contractiva, el atractor era un único punto, para un sistema de funciones contractivas el atractor es algo más general: un compacto.

En la figura siguiente, además de la función anterior, consideramos dos nuevas funciones, una que transforma el cuadrado negro en el azul y otra que lo transforma en el verde. Las tres funciones reducen el tamaño a la mitad, pero la función azul y la verde además desplazan la figura, la primera hacia la derecha y la segunda hacia arriba. Como son tres las funciones, en el primer paso se obtienen 3 imágenes reducidas del cuadrado original, en el segundo 9, en el tercero 27.



Muchos reconocerán el atractor obtenido: se trata del triángulo de Sierpinsky, introducido por el matemático polaco W. Sierpinsky como versión bidimensional del conjunto de Cantor. En él salta a la vista una de las características esenciales de los conjuntos fractales: la autosimilitud. Este término alude al hecho de que una imagen fractal contiene fragmentos semejantes a sí misma. Esto se entiende perfectamente mirando la figura de la derecha de la última serie: el triángulo verde es una copia, en este caso idéntica, de la figura completa. Pero no solo eso: ampliando la figura se ve que es posible encontrar fragmentos semejantes al triángulo original tan pequeños como queramos.



Otra sorpresa nos espera si en vez de un cuadrado partimos de otra figura, un círculo, por ejemplo:



En efecto, el nuevo atractor es en realidad idéntico al obtenido a partir del cuadrado. Lo cierto es que sea cual sea la forma del compacto de partida, un cuadrado, un círculo, la forma de la península de Crimea, al aplicar el proceso iterativo siempre llegaremos al mismo resultado. Esto explica que a esta forma límite se le llame atractor.

## ***El programa genifs.exe***

Resumiendo, los IFS son sistemas que mediante la iteración, es decir, la repetición de ciertos cálculos, sencillos pero fatigosos, dan lugar a imágenes interesantes de apariencia complicada. Nada hay más adecuado para ser automatizado, y esto es lo que hace el programa genifs.exe: automatizar la generación de atractores de IFS. Esta tarea se realiza en dos pasos. En el primero, con movimientos de ratón, se dibujan en pantalla un cierto número de paralelogramos. En el segundo, pulsando un botón, se genera el atractor. Así de sencillo. El programa incorpora instrucciones detalladas acerca de su uso, así que no me extiendo al respecto. Sin embargo, es interesante saber algo de lo que hay bajo tan sencillas operaciones:

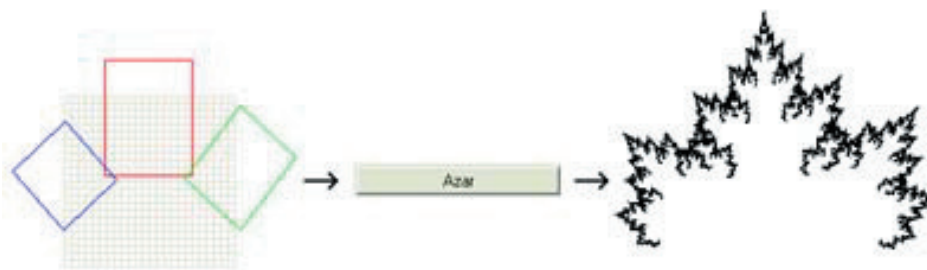
## ***Preparación de las imágenes***

Aunque la única condición que deben cumplir las funciones de un IFS es que sean contractivas, las funciones más frecuentemente utilizadas son las aplicaciones afines, sin duda porque, a pesar de su sencillez, permiten obtener atractores espectaculares, algunos tan austeramente geométricos como el triángulo de Sierpinsky y otros tan sorprendentemente botánicos como el helecho de Barnsley que se muestra a la derecha y que es el atractor de un sistema de tan solo cuatro de estas aplicaciones.



Las aplicaciones afines transforman rectas paralelas en rectas paralelas o, lo que es lo mismo, paralelogramos en paralelogramos. De hecho, es suficiente conocer la imagen de tres de los vértices de un paralelogramo dado para poder calcular las ecuaciones que describen la aplicación (ver [apéndice](#)).

Esta característica es la que utiliza el programa genifs.exe para trabajar: mediante movimientos de ratón se puede trasladar, girar, comprimir o deformar un cuadrado inicial para obtener un paralelogramo imagen. Por cada paralelogramo imagen que se dibuje, el programa calculará las ecuaciones de la aplicación afín que transforma el cuadrado inicial en dicho paralelogramo. Varios paralelogramos darán lugar a varias funciones y, por tanto, a un IFS. Entonces solo faltará pulsar un botón.



## Algoritmos de generación de atractores de IFS

Una vez preparado un conjunto de imágenes, el programa calculará y mostrará en pantalla el atractor correspondiente. Hay varios algoritmos para generar atractores de IFS. El programa puede aplicar dos de ellos. El usuario únicamente debe elegir cuál.

El primero consiste en seguir las instrucciones al pie de la letra: se dibuja un compacto, en concreto un cuadrado, se aplican sobre él la funciones del sistema, se vuelven a aplicar sobre el resultado del proceso anterior, y así sucesivamente. En este proceso iterativo en cada paso se obtiene un compacto formado por las distintas imágenes de la figura original que resultan de aplicar las funciones del sistema en todos los órdenes posibles. Este algoritmo es el que se ha utilizado para obtener los gráficos anteriores.

Otra forma mucho más económica de generar atractores de IFS hace uso del azar. Se parte de un punto cualquiera y sobre él se aplica una de las funciones del sistema escogida aleatoriamente. Sobre el punto obtenido se aplica otra función, también elegida al azar entre las del sistema, y así sucesivamente. Si hacemos esto en vacío, es decir, sin dibujar, durante un cierto número de pasos, y a continuación continuamos el proceso pero ya dibujando los puntos obtenidos, veremos como el atractor va apareciendo "mágicamente" ante nuestros ojos.

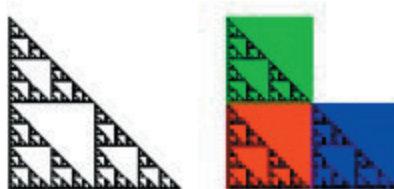
Para mejorar la efectividad del algoritmo conviene asignarle a cada función una cierta probabilidad proporcional al tamaño de la imagen que genera. Es decir: en cuanto más grande sea dicha imagen con más probabilidad se elegirá la función en el proceso iterativo (el programa calcula dichas probabilidades proporcionalmente al determinante de la matriz de los coeficientes de cada aplicación).

## Ingeniería inversa

Gracias al programa, dibujar unos cuantos paralelogramos y ver qué atractor genera el IFS correspondiente es fácil. Algo más difícil es la operación inversa, a saber: dada una cierta figura, encontrar un conjunto de funciones contractivas (en nuestro caso, un conjunto de paralelogramos) que tenga por atractor precisamente esa figura.

Para resolver este problema disponemos del teorema del collage de Barnsley. De modo intuitivo y sin entrar en detalles, este teorema dice que debemos buscar partes de la figura que queremos obtener que sean semejantes a la totalidad y de modo que estas partes, en conjunto, cubran toda la imagen.

La siguiente figura ilustra la idea en el caso del triángulo de Sierpinsky: dado el triángulo agujereado, ¿cómo encontrar un sistema de funciones que lo tengan por atractor? La imagen de la derecha nos muestra la forma: dado que el triángulo completo se puede cubrir mediante tres copias reducidas del mismo, si consideramos las aplicaciones afines que transforman la figura completa en cada una de sus tres copias, el sistema formado por ellas generará el triángulo como atractor. Para comprobar la corrección de nuestra hipótesis bastará abrir el programa, dibujar los cuadrados verde, rojo y azul, y hacerle trabajar.



No siempre resulta tan fácil como en este caso. Por eso el programa permite ensayar nuestras hipótesis y volver a la fase de preparación para modificar o borrar o añadir imágenes nuevas según los resultados obtenidos. También permite grabar el trabajo realizado en un fichero para poder recuperarlo posteriormente.

La finalidad del programa genifs es jugar con estos sorprendentes atractores fractales. Sin embargo, me gustaría pensar que también puede servir para visualizar el aspecto más interesante de los IFS. Los sistemas de función iterada son una muestra de cómo formas de apariencia complicada pueden surgir de procesos sencillos. La complicada hoja de un helecho o el poroso triángulo de Sierpinsky pueden generarse mediante la aplicación iterativa de un puñado de aplicaciones afines. Dicho de otro modo: son suficientes unas pocas ecuaciones, unos pocos números, para codificar formas de apariencia altamente compleja. Muestran, en resumen, unos de los caminos de emergencia de la complejidad. Que todo esto tenga además una aplicación tan práctica como es la compresión fractal de imágenes no es más que otra sorpresa (ver apéndice).

## Apéndices

### 1. Las ecuaciones de una aplicación afín

Una aplicación afín queda definida por un par de ecuaciones de la forma:

$$\begin{aligned} \cdot x' &= a_{11} \cdot x + a_{12} \cdot y + b_1 \\ \cdot y' &= a_{21} \cdot x + a_{22} \cdot y + b_2 \end{aligned}$$

donde  $(x, y)$  son las coordenadas de un punto cualquiera y  $(x', y')$  las de su imagen.

Sea, por ejemplo, la aplicación afín  $f(x)$  de ecuaciones:

$$\begin{aligned} \cdot x' &= 0,5 \cdot x + 0,125 \cdot y + 0,25 \\ \cdot y' &= 0 \cdot x + 0,25 \cdot y + 0,125 \end{aligned}$$

Dados los puntos A (0, 0); B (1, 0) y C (0, 1), para obtener las coordenadas de sus imágenes se sustituyen en las ecuaciones  $x$  e  $y$  por las coordenadas del punto original. Así, para el punto A:

$$\begin{aligned} \cdot x' &= 0,5 \cdot 0 + 0,125 \cdot 0 + 0,25 = 0,25 \\ \cdot y' &= 0 \cdot 0 + 0,25 \cdot 0 + 0,125 = 0,125 \end{aligned}$$

es decir,  $f(A) = A' = (0,25, 0,125)$

Haciendo lo mismo para los puntos B

$$\begin{aligned} \cdot x' &= 0,5 \cdot 1 + 0,125 \cdot 0 + 0,25 = 0,75 \\ \cdot y' &= 0 \cdot 1 + 0,25 \cdot 0 + 0,125 = 0,125 \end{aligned}$$

y C

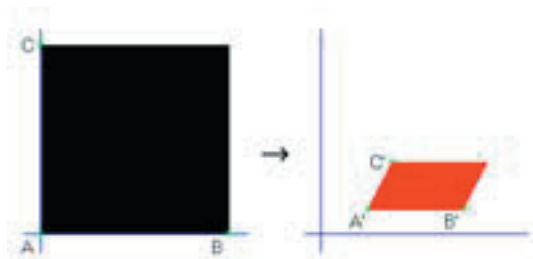
$$\begin{aligned} \cdot x' &= 0,5 \cdot 0 + 0,125 \cdot 1 + 0,25 = 0,375 \\ \cdot y' &= 0 \cdot 0 + 0,25 \cdot 1 + 0,125 = 0,375 \end{aligned}$$

se tiene

$$f(B) = B' = (0.75, 0.125)$$

$$f(C) = C' = (0.375, 0.375),$$

Lo anterior, expresado gráficamente, significa que la aplicación  $f(x)$  transforma el cuadrado negro en el paralelogramo rojo de la figura:



## 2. Obtención de las ecuaciones a partir de la imagen de tres puntos

El problema inverso, es decir, obtener las ecuaciones de una aplicación afín conocidas las imágenes de tres puntos, se resuelve mediante dos sistemas de tres ecuaciones. Supongamos conocidas las imágenes de los puntos A, B y C:

- $f(0, 0) = (0.25, 0.125)$
- $f(1, 0) = (0.75, 0.125)$
- $f(0, 1) = (0.375, 0.375)$

Sustituyendo en

- $x' = a_{11}x + a_{12}y + b_1$
- $y' = a_{21}x + a_{22}y + b_2$  se tiene el sistema de ecuaciones
- $0.25 = a_{11} \cdot 0 + a_{12} \cdot 0 + b_1$
- $0.125 = a_{21} \cdot 0 + a_{22} \cdot 0 + b_2$
- $0.75 = a_{11} \cdot 1 + a_{12} \cdot 0 + b_1$
- $0.125 = a_{21} \cdot 1 + a_{22} \cdot 0 + b_2$
- $0.375 = a_{11} \cdot 0 + a_{12} \cdot 1 + b_1$
- $0.375 = a_{21} \cdot 0 + a_{22} \cdot 1 + b_2$

Agrupando las ecuaciones pares por un lado y las impares por el otro quedan dos sistemas de tres ecuaciones con tres incógnitas que una vez resueltos proporcionan las ecuaciones de la aplicación afín.

## 3. Expresión matricial de una aplicación afín

Las ecuaciones de una aplicación afín  $f(x)$

- $x' = a_{11}x + a_{12}y + b_1$
- $y' = a_{21}x + a_{22}y + b_2$

se expresan matricialmente



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

En esta forma, y en lenguaje vectorial, los vectores columna  $(a_{11}, a_{21})$  y  $(a_{12}, a_{22})$  son las imágenes de los vectores de la base  $(1, 0)$  y  $(0, 1)$ , y el vector columna  $(b_1, b_2)$  es el vector que aplicado sobre el origen nos da su imagen.

La matriz de los coeficientes es la causante de las deformaciones de la imagen, mientras que la matriz columna de los términos independientes es la causante del desplazamiento de la figura.

#### 4. Compresión fractal

Los IFS tienen una aplicación práctica. Se trata de la compresión fractal, y su idea básica es sencilla. Observando las ecuaciones del punto 1. Se ve que una aplicación afín en el plano queda completamente determinada por seis números:  $a_{11}$ ,  $a_{12}$ ,  $b_1$ ,  $a_{21}$ ,  $a_{22}$ ,  $b_2$ . Dado que el IFS que tiene por atractor, por poner un ejemplo, el triángulo de Sierpinsky consta de tan solo tres aplicaciones, resulta que son necesarios  $3 \cdot 6 = 18$  números para codificarla, los cuales, almacenándolos en campos de coma flotante de doble precisión, ocuparán 144 bytes de memoria.

Por otra parte, almacenar dicho atractor en formato GIF en un tamaño de 500x500 pixeles exige más de 9000 bytes. Es decir, que una imagen que ocupa más de 9000 bytes puede almacenarse en tan solo 144 bytes. Además, almacenar la imagen en formato IFS permite reproducirla con cualquier nivel de detalle que se desee.

El interés del uso de los IFS para la compresión de imágenes es evidente.

### ***Sitios de interés***

Algunas direcciones para saber más:

- Wikipedia: [http://en.wikipedia.org/wiki/Iterated\\_function\\_system](http://en.wikipedia.org/wiki/Iterated_function_system)
- MathWorld: <http://mathworld.wolfram.com/IteratedFunctionSystem.html>
- Cut The Knot!: <http://www.cut-the-knot.org/ctk/ifs.shtml>
- David J. Wright: <http://www.math.okstate.edu/mathdept/dynamics/lecnotes/node47.html>
- David Arnold: <http://online.redwoods.cc.ca.us/instruct/darnold/ifs/>
- Epsilones: <http://www.epsilones.com/paginas/a-bestiario.html#bestiario-fractal>