

# Prácticas con Algoritmos Fractales

C.Escribano, A.Giraldo, M.A.Sastre

Dept. de Matemática Aplicada  
Facultad de Informática  
Universidad Politécnica de Madrid  
Boadilla del Monte, 28660-Madrid  
e-mail: [masastre@fi.upm.es](mailto:masastre@fi.upm.es)

## Resumen

Se presentan ideas para el desarrollo de prácticas con técnicas y algoritmos Fractales. Estas técnicas tienen especial interés en el tratamiento de imágenes, tanto en la generación y simulación como en la compresión.

## 1. Introducción

Los algoritmos Fractales se utilizan para describir formas complejas con una reducida cantidad de información mediante sencillos procesos iterativos.

En 1977 B.B. Mandelbrot publicó el libro "The Fractal Geometry of Nature" [9] describiendo numerosas aplicaciones de este tipo de estructuras para la investigación en ciencias aplicadas, siendo una herramienta fundamental en la modelización de un gran número de fenómenos naturales. El término fractal, procedente del latín "fractus" (fragmentado, irregular), fue introducido por Mandelbrot.

El contexto en el que se pueden desarrollar estas prácticas es muy variado, dependiendo del curriculum ofertado por cada escuela o facultad. Se proponen prácticas que están indicadas en asignaturas relacionadas con tratamiento de la señal, simulación y técnicas gráficas y, por supuesto, en geometría fractal. Algunas de estas prácticas encajan en asignaturas relacionadas con teoría de control, sistemas dinámicos y caos, procesos estocásticos, procesos iterativos y programación,... En el caso concreto de la Facultad de Informática de la U.P.M. estas prácticas se enmarcan en las asignaturas optativas de Geometría Fractal y Sistemas Dinámicos.

Las prácticas que se presentan a continuación se pueden encontrar implementadas en Maple6 en el libro "Geometría Fractal, algoritmos y aplicaciones" [7].

## 2. Práctica de Fractales Clásicos

El objetivo de esta práctica es la generación de varios fractales clásicos como el Conjunto de Cantor, Triángulo de Sierpinski, curva de Koch, etc, mediante un algoritmo iterativo determinista, permitiendo la visualización de diferentes etapas de la construcción del fractal. La figura 1 muestra el primer paso de la construcción del triángulo de Sierpinski.

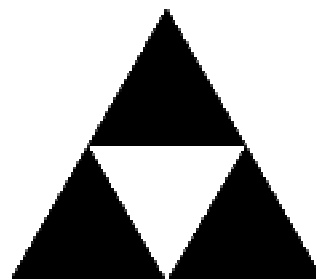


Figura 1. Primera etapa de la construcción del triángulo de Sierpinski  
La figura 2 muestra la novena etapa.

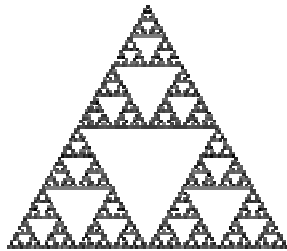


Figura 2. Nueve iteraciones en la construcción del Triángulo de Sierpinski

- *Descripción del algoritmo.* Este algoritmo se conoce como algoritmo de la fotocopiadora múltiple, ya que su funcionamiento es el mismo que una máquina fotocopiadora que contrae el original un número determinado de veces, a diferentes escalas y sitúa las copias en diferentes lugares de la página. Para cada máquina de este tipo, existe un único conjunto que es invariante, es decir, al introducir este conjunto en la máquina, la salida es el mismo conjunto. Lo sorprendente de este algoritmo, es que si empezamos con un conjunto cualquiera y retro-alimentamos la máquina con la salida indefinidamente, el proceso converge al conjunto invariante. De esta forma se pueden visualizar aproximaciones a de los conjuntos fractales que se definen mediante estos procesos recursivos.
- *Conocimientos previos.* Los conceptos que se utilizan son muy básicos: convergencia, semejanzas contractivas, teorema del punto fijo, iteración.
- *Asignaturas en las que se puede realizar esta práctica.* Puesto que los conceptos que se manejan son muy sencillos y los alumnos los manejan desde primer curso, se puede realizar la practica en asignaturas básicas como Cálculo (visualizar el concepto de convergencia, punto fijo), Algebra (visualizar cómo transforma una semejanza o una aplicación lineal), programación (como primeras prácticas para implementar algoritmos recursivos), fractales, etc.
- *Primera variación del algoritmo: Juego del caos* (M.Barnsley [1]). Es una pequeña

variación del algoritmo anterior, en el que se introduce un sistema de probabilidades. De esta forma, no se aplican todas las contracciones en cada paso, si no que se escoge de acuerdo a esas probabilidades. Esto permite un algoritmo menos costoso, ya que en el caso anterior la cantidad de puntos crece exponencialmente. Si se da un sistema adecuado de probabilidades se pueden conseguir efectos de sombreado. La descripción del algoritmo es la siguiente: Se toma un punto arbitrario del plano. Se tira un dado. Si sale 1 ó 2 se dibuja un nuevo punto a mitad de camino entre el punto inicial y el punto (0,0), si sale 3 ó 4 se dibuja un nuevo punto a mitad de camino entre el punto inicial y el punto (1,0), y si sale 5 ó 6 se dibuja un nuevo punto a mitad de camino entre el punto inicial y el punto  $(1/2, 3^{1/2}/2)$ . Este juego de tirar el dado se repite con el nuevo punto obtenido. Iteramos este proceso indefinidamente. Aparentemente, este juego debería proporcionarnos una nube amorfa de puntos. La figura 2 muestra el resultado con 10000 puntos.

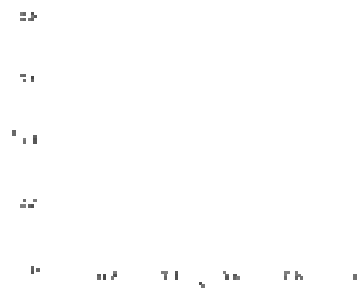


Figura 2. Juego del Caos

- *Segunda Variación: Fractales aleatorios.* Si se introduce azar en los algoritmos que definen los fractales clásicos se obtienen formas más realistas. La figura 4 muestra una isla de Koch aleatoria.

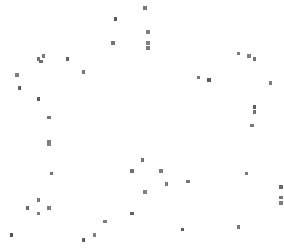


Figura 4. Isla de Koch aleatoria

- *Tercera Variación. Movimiento Browniano unidimensional.* Un movimiento Browniano es un movimiento en el que la partícula se mueve de forma aleatoria. Aquí, la regularidad no es geométrica como en los casos anteriores, si no estadística. Para representar este movimiento se suele utilizar el algoritmo de desplazamiento del punto medio, que consiste en desplazar el punto medio de un segmento mediante una distribución gaussiana, de forma recurrente. La figura 5 muestra un movimiento browniano unidimensional, donde los desplazamientos hacia la izquierda se representa como negativo y hacia la derecha, positivo.



Figura 5. Movimiento browniano unidimensional

- *Movimiento Browniano en dos dimensiones.* Se modeliza con una pequeña variación del algoritmo anterior. En lugar de utilizar segmentos, se utiliza una rejilla (diádica, triangular, etc.) y se desplaza el punto que se

elija. Dando un color a cada altura se pueden simular paisajes, nubes, islas y una gran variedad de texturas, muy útiles en técnicas gráficas. En la figura 6, las alturas inferiores a un cierto nivel se han representado en color azul para simular el mar.

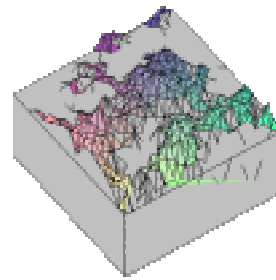


Figura 6. Movimiento browniano 3D

### 3. Conjunto de Mandelbrot

El objetivo de esta práctica es visualizar el conjunto de Mandelbrot mediante el algoritmo de tiempo de escape. Para dibujar el conjunto de Mandelbrot se parte de la función  $F_c(z)=z^2+c$ .

- *Descripción del algoritmo.* Se considera un número  $k$  del orden de 100, y para cada  $c$  de la ventana que vamos a dibujar (eligiendo ventanas más pequeñas, se representa un zoom del conjunto), computamos términos de la sucesión  $f_c^n(0)$ . Si los  $k$  primeros términos verifican todos que  $|f_c^n(0)| < 2$  se decide que  $c$  está en el conjunto de mandelbrot  $M$  y se pinta en negro. Si, por el contrario, existe  $n < k$  con  $|f_c^n(0)| > 2$  se interrumpe la computación para ese  $c$  y se determina que  $c$  está fuera de  $M$ . Si a cada número entre 0 y  $k$  se le asigna un color y se representa cada número  $c$  en el color que corresponde al primer  $n$  tal que  $|f_c^n(0)| > 2$ , es decir, el momento en el que se “escapa” de la zona acotada, se obtienen las típicas imágenes del conjunto de Mandelbrot que todos conocemos. En la figura 7 se muestra

una imagen del conjunto de Mandelbrot hecha con Maple6.

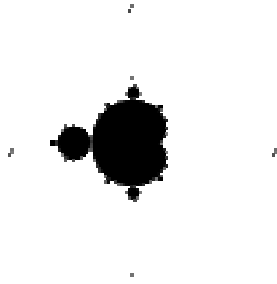


Figura 7. Conjunto de Mandelbrot

- *Variaciones de este algoritmo.* Se pueden realizar numerosas variaciones de este algoritmo, modificando la forma de asignar el color y aplicándolo a otras funciones diferentes. Por ejemplo, para que los cambios de color no sean tan bruscos, se puede introducir una función logarítmica (como en el método “dmj-Smooth” de D.M.Jones). También se puede asignar el color en relación a otras propiedades de los puntos de la órbita  $f^n(c)$ , como puede ser la distancia de los puntos de la órbita a una región dada, por ejemplo un asteroide (método “astroid” de K.Mitchell, 1998). Hay numerosos métodos de coloreado ya implementados en programas para generar imágenes fractales, como “fractint” (“freeware”) o “ultrafractal” (“shareware”), pero siempre se pueden diseñar nuevos métodos estableciendo cualquier propiedad que se nos ocurra para asignar el color. Así se obtienen imágenes fractales espectaculares como la que se pueden ver en concursos y exposiciones de imágenes fractales. Un concurso de este tipo se ha realizado en la facultad de informática (ver <http://www.dma.fi.upm.es/concurso/concurso.html>).
- *Conocimientos previos.* Operaciones con números complejos, métodos iterativos, convergencia.
- *Asignaturas en las que se puede realizar esta práctica.* Esta práctica y sus posibles variaciones son adecuadas en asignaturas relacionadas con técnicas gráficas, sistemas

dinámicos, variable compleja, fractales, análisis y visualización de datos.

#### 4. Compresión de imágenes

Los métodos de “compresión fractal de imágenes digitales” están incluidos dentro de los métodos de compresión con pérdida de información. Por técnicas de compresión fractal de imágenes se entiende los métodos que permiten obtener imágenes fractales (en realidad, lo que se obtiene son las aplicaciones contractivas -SFI-que las generan) parecidas a las imágenes reales que se quieren comprimir. En vez de almacenar en memoria la imagen real, almacenamos el SFI que nos genera la imagen fractal, que es una aproximación de la imagen real (ver [2] y [5]). El algoritmo de descompresión de la máquina fotocopidora múltiple, mencionado en el punto 2. Los algoritmos que se propone implementar en esta práctica se describen en [7].

El objetivo de la práctica es iniciar al alumno en la compresión de imágenes, con un algoritmo de compresión fractal muy básico. A partir de este, se propone analizar su eficacia y los problemas que presenta, para implementar diferentes mejoras.

El problema de la compresión es complejo. Hay que estudiar formas de comprimir que sean reversibles, ya que es necesario recuperar la información original o con alguna modificación que nos permita reconocerla. Para estudiar si un método de compresión-descompresión es bueno, hay que analizar diferentes criterios: ratio de compresión, tiempo de compresión, tiempo de descompresión y calidad de la información que se obtiene tras el proceso de compresión-descompresión. Dependiendo de la utilidad que se vaya a dar a la información comprimida, será necesario renunciar a unas mejoras en favor de otras. Por ejemplo, si se trata de comprimir una imagen para almacenarla o mandarlas por la red, podemos incrementar el tiempo para comprimirla a cambio de conseguir mejor ratio de compresión y buena calidad de la imagen. Este es el caso de la compresión fractal, que se caracteriza por tener un tiempo de compresión muy alto, pero un tiempo de descompresión muy pequeño, buena calidad de imagen y un buen ratio de compresión. Estas características son también muy útiles a la hora de

guardar una información que no va a ser modificada, pero sí consultada asiduamente. Este es el caso de la imágenes que aparecen en una enciclopedia en CD, como la Encarta, cuyas imágenes están comprimidas fractalmente. Sin embargo, si se trata de transmitir en tiempo real, como en una video-conferencia, no será útil esta técnica ya que primará tanto el tiempo de compresión como de descompresión, aunque se pierda algo en calidad de imagen, o incluso de ratio de compresión. En este caso se utilizan técnicas más tradicionales basadas en transformada rápida del coseno, como JPEG.

- *Algoritmo en Blanco y negro.* Se divide la imagen en rangos  $R_i$  de 8x8 pixels disjuntos. Se divide la imagen en todos los posibles dominios  $D_i$  de 16x16 pixels. Se amplía cada rango  $R_i$  al doble de su tamaño y se compara el rango ampliado con cada uno de los dominios de tamaño doble seleccionando el dominio  $D_i$  más parecido. Si la diferencia entre  $R_i$  y  $D_i$  es menor que 15 pixels, almacenamos la transformación que lleva  $D_i$  en  $R_i$ . Si la diferencia entre  $R_i$  y  $D_i$  es mayor que 15 pixels se divide  $R_i$  en cuatro subrangos de 4x4, y cada dominio  $D_i$  en cuatro subdominios de 8x8 pixels y se repite el proceso con los nuevos rangos y dominios. Este algoritmo es de implementación sencilla, pero tiene la desventaja de que tanto el ratio como el tiempo de compresión son malos.
- *Asignaturas en las que se puede realizar esta práctica.* Tratamiento de imágenes, geometría fractal.
- *Variación 1.* Compresión con quad-trees. Este algoritmo es análogo al anterior, pero en lugar de tomar todos los rangos iguales se hace la búsqueda de rangos y dominios por quad-trees. Este algoritmo mejora notablemente, en la mayoría de los casos, el ratio y el tiempo de compresión.
- *Variación 2.* Niveles de grises. El algoritmo anterior se puede aplicar a imágenes con diferentes niveles de grises. La complejidad del algoritmo aumenta al introducir una métrica más compleja que aumentar los cálculos por cada rango.
- *Variación 3.* Preclasificación de dominios. Clasificando previamente los dominios se

reducen notablemente las búsquedas y los cálculos por cada rango, por lo que mejora el tiempo de compresión.

#### 4. Valoración

El interés mostrado por los alumnos a la hora de realizar estas prácticas sobrepasa los objetivos que nos habíamos fijado inicialmente. Por ejemplo, en compresión fractal de imágenes se presentaron prácticas realizadas en LINUX y en JAVA, en las que se introdujeron mejoras que no se habían explicado en clase. En el tema de fractales clásicos, se presentaron prácticas en JAVA con applets que permiten al usuario modificar algunos parámetros y experimentar qué sucede. Algunos de estos alumnos siguen trabajando en estos temas con el fin de realizar su TFC.

El éxito del concurso "Imágenes Fractales, ¿Cómo se hacen?" realizado en la Facultad de Informática de la U.P.M. durante el Año Mundial de las Matemáticas, hace patente el interés de los alumnos por estas técnicas.

#### Referencias

- [1] Barnsley M.F., *Fractals Everywhere*. Academic press, 1993
- [2] Barnsley M.F., Hurd L.P., *Fractal Image Compression*. AK Peters, Ltd., 1993
- [3] Bunde A. Y Havlin S. (Eds.) , *Fractals in Science*. Springer-Verlag, Berlin, 1994.
- [4] Falconer K.J., *Fractal Geometry*. Wiley, New York, 1990.
- [5] Fisher Y., *Fractal Image Compression*. Springer-Verlag, 1994.
- [6] Flake G-W., *The computational Beauty of Nature*. Bradford Book, Massachusetts, 1999.
- [7] Giraldo A. Y Sastre M.A. *Geometría Fractal, algoritmos y aplicaciones. Publicaciones de la Facultad de Informática de la UPM. Madrid 2001.*
- [8] Guzman M., Martín M.A., Morán M., Reyes M. *Estructuras fractales y aplicaciones*. Labor, Barcelona, 1993.
- [9] Mandelbrot B., *Fractals Geometry of Nature*. W.H. freeman and Co., New york, 1982.

- [10] Peitgen O.H., Jürgens H. y Saupe D., *Chaos and Fractals. New Frontiers of Science*. Springer-Verlag, Berlin 1992.
- [11] H.-O.Peitgen y P.H.Richter, *The beauty of fractals*. Springer-Verlag, Berlin, 1986.
- [12] Stewart, I. *The Magical Maze. Seeing the World Through Mathematical Eyes*. John Wiley & Sons, Inc. 1998.