Escuela Técnica Superior de
Ingeniería Informática

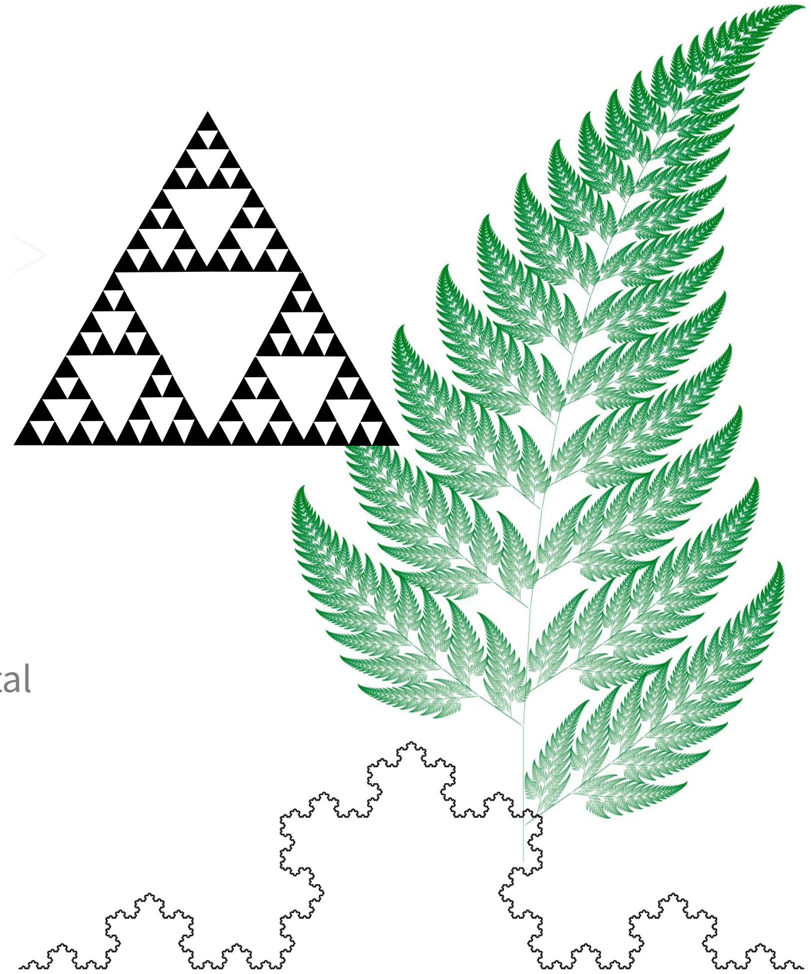
Sistemas de funciones iteradas

Matemáticas para la Computación

Sevilla, Enero 2021

Beatriz Galiana Carballido
Fernando Sánchez Paredes

Índice de contenidos



Estudio teórico: fractales

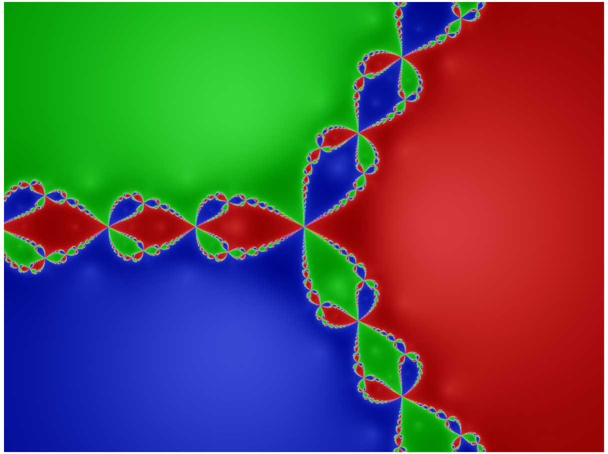
Objeto geométrico cuya estructura básica, fragmentada o aparentemente irregular, se repite a diferentes escalas

Propiedades:

- Autosimilares
- Dimensión fractal > dimensión topológica

Definición. Dimensión de autosemejanza.
Sea un conjunto autosemejante compuesto por N copias reducidas en un factor de escala e, la dimensión d de autosemejanza del conjunto se define como el exponente que verifica

$$N = e^d$$



Estudio teórico: aplicaciones afines

Elementales

Traslación $f\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}a\\b\end{pmatrix}$

Rotación $f\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}$

Simetría respecto del eje x e y.

$f\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}$ $f\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}-1 & 0\\0 & -1\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}$

Homotecia de razón *r* centrada en el origen.

$$f\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}r & 0\\0 & r\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}$$

Contractivas

Las **aplicaciones contractivas** se obtienen a partir de transformaciones elementales donde sus coeficientes son **menores que la unidad**.

$$f\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}a & b\\c & d\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}e\\f\end{pmatrix}$$

Estudio teórico: sistemas de funciones iteradas

Determinista

1. Tomamos un conjunto de puntos en el plano
2. Le aplicamos cada una de las *n* transformaciones afines de nuestro sistema
3. Obtenemos *n* puntos
4. A estos *n* puntos le volvemos a aplicar las *n* transformaciones afines.

Vamos a realizar **tantas iteraciones como sean necesarias** hasta que obtengamos una figura que resultará ser el **atractor del sistema**

Aleatorio

Se aplica a **un solo punto** que vamos dibujando

1. Asignar a cada una de las transformaciones del sistema una probabilidad.
2. En cada iteración del algoritmo escogemos una de las transformaciones en función de la probabilidad asignada
3. Dibujar dicho punto

Iteramos tantas veces como sea necesario para obtener el **atractor** del sistema

Ejemplos clásicos I: *Triángulo de Sierpiński*

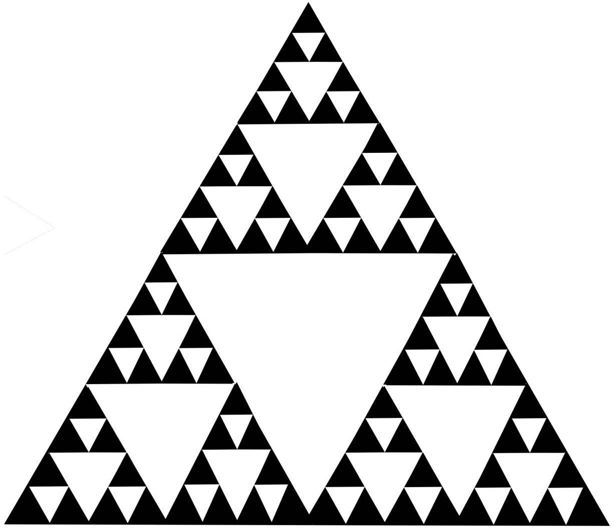
Sistemas de funciones iteradas equiprobables

$$f_1\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{1}{2} & 0\\0 & \frac{1}{2}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}0\\0\end{pmatrix}=(\frac{x}{2},\frac{y}{2})$$

$$f_2\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{1}{2} & 0\\0 & \frac{1}{2}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}\frac{1}{2}\\0\end{pmatrix}=(\frac{x+1}{2},\frac{y}{2})$$

$$f_3\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{1}{2} & 0\\0 & \frac{1}{2}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}\cos 60\\\sin 60\end{pmatrix}=(\frac{x+\cos 60}{2},\frac{y+\sin 60}{2})$$

Todas ellas con ⅓ de probabilidad.



- Puede construirse a partir de cualquier triángulo

Ejemplos clásicos II: *Helecho de Barnsley*

Sistemas de funciones iteradas

La primera función se da con probabilidad 0,01

$$f_1\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}0 & 0\\ 0 & \frac{4}{25}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}0\\0\end{pmatrix}$$

La segunda función se da con probabilidad 0,85

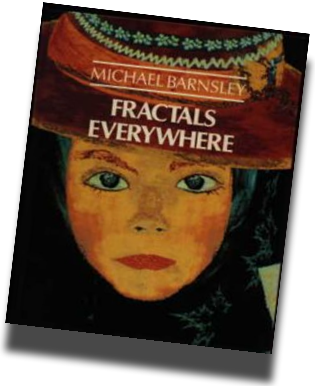
$$f_2\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{17}{20} & \frac{1}{25}\\ -\frac{1}{25} & \frac{17}{20}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}0\\ \frac{8}{5}\end{pmatrix}$$

La tercera función se da con probabilidad 0,07

$$f_3\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{1}{5} & -\frac{13}{50}\\ \frac{23}{100} & \frac{11}{50}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}0\\ \frac{8}{5}\end{pmatrix}$$

La cuarta función se da con probabilidad 0,07

$$f_4\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}-\frac{3}{20} & \frac{7}{25}\\ \frac{13}{50} & \frac{6}{25}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}0\\ \frac{11}{25}\end{pmatrix}$$



Ejemplos clásicos III: *Curva de Koch*

Sistemas de funciones iteradas equiprobables

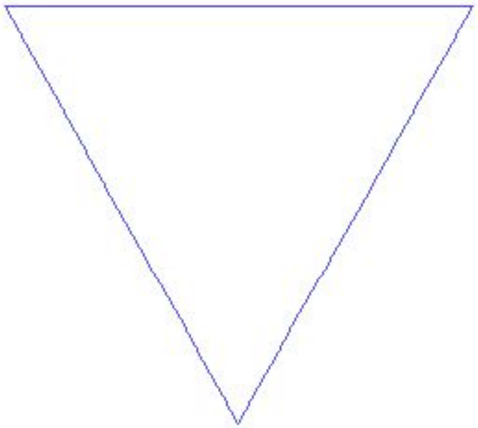
$$f_1\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{1}{3} & 0\\0 & \frac{1}{3}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}0\\0\end{pmatrix}$$

$$f_2\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{1}{6} & 0{,}2\widehat{8}\\0{,}2\widehat{8} & \frac{1}{6}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}\frac{1}{3}\\0\end{pmatrix}$$

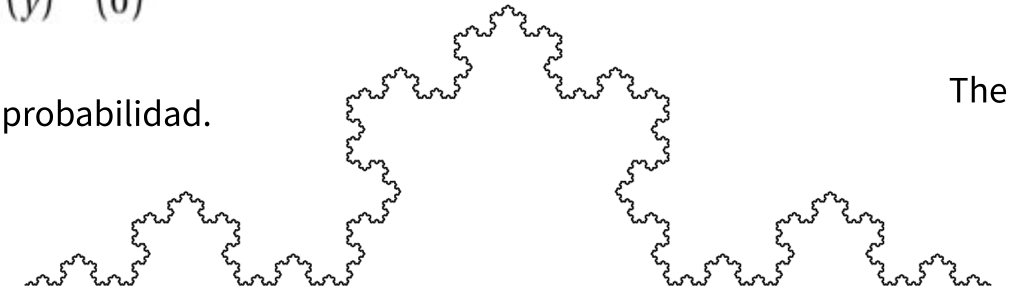
$$f_3\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{1}{6} & 0{,}2\widehat{8}\\0{,}2\widehat{8} & \frac{1}{6}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}\frac{2}{3}\\0\end{pmatrix}$$

$$f_4\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}\frac{1}{3} & 0\\0 & \frac{1}{3}\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix}+\begin{pmatrix}\frac{2}{3}\\0\end{pmatrix}$$

Todas ellas con ¼ de probabilidad.



The Koch Snowflake



El algoritmo: *herramientas y planteamiento*

1. `HTML5`, `CSS` y `JavaScript`
2. `Canvas`: renderización interpretada dinámica de gráficos 2D y mapas de bits y animaciones con estos gráficos

---

- Funciones auxiliares
- Programa principal

Implementación del algoritmo:

funciones auxiliares

```
readIfs()

plot()

next()

findBounds()

newPicture()

drawSomePoints()
```

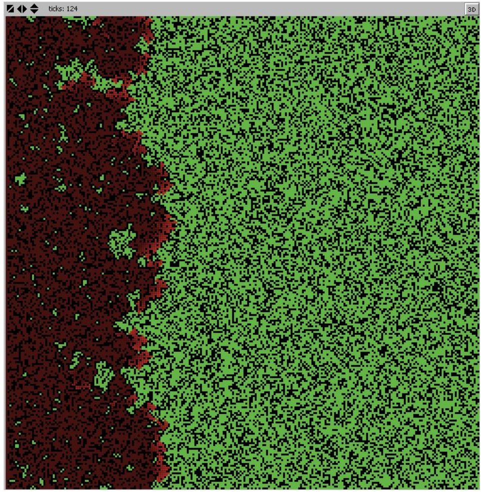
Implementación del algoritmo:

programa principal

- Declaración de variables globales
- Creación de desplegables y cuadro de texto para parámetros del IFS
- Botón ¡Cargar!
- Selección de fractal del desplegable
- Temporizador
- Función `next()` - aplica el sistema de funciones iteradas

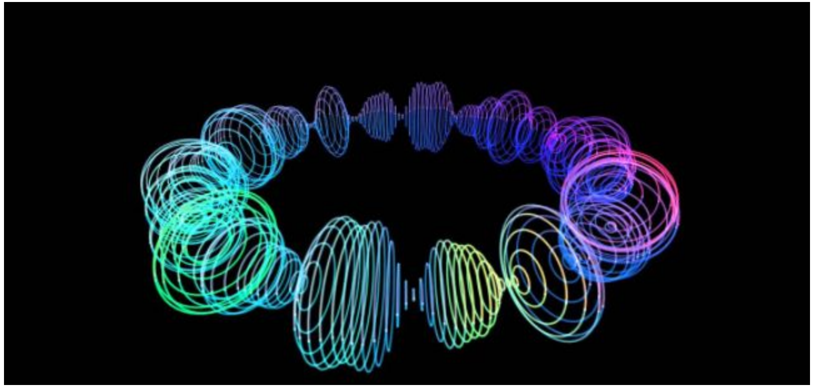
Aplicaciones

- Medicina
- Geografía
- Bioquímica

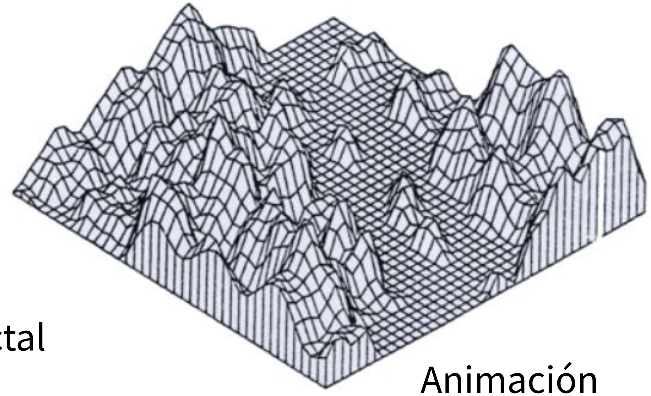


Ecología

Arqueología



Música fractal



Animación

Aplicación web: *HTML5, CSS y JavaScript*

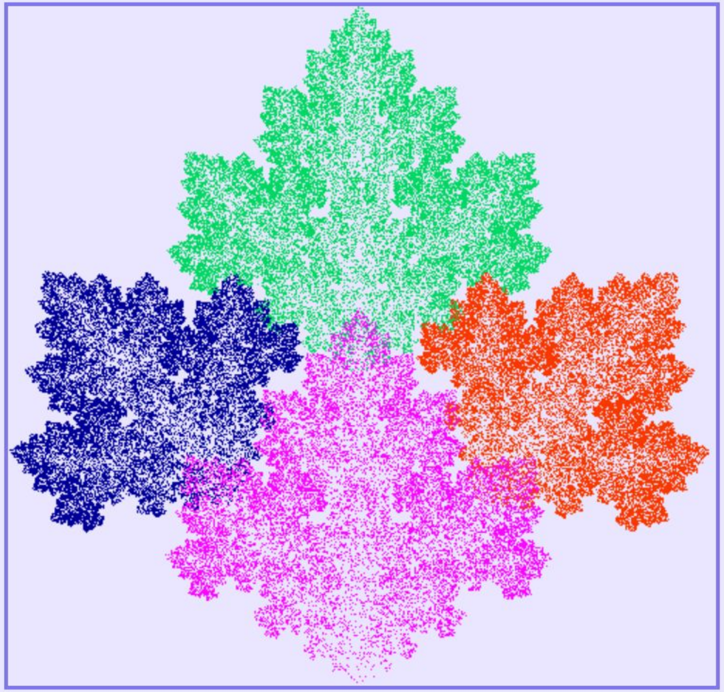
Renderizador de Sistemas de Funciones Iteradas

Elige uno de los siguientes ejemplos clásicos o modifica los parámetros tú mismo: leaf.ifs

```
0.555  0.000  0.000  0.555   0.000  0.000   0.20
0.550  0.000  0.000  0.550   0.000  0.185   0.30
0.353  0.281 -0.295  0.336   0.068  0.112   0.25
0.353 -0.281  0.295  0.336  -0.068  0.112   0.25
```

¡Cargar!

Échale un vistazo al [repositorio]() del código en GitHub



Referencias

Repositorio en GitHub:

https://github.com/beagaliana/mc2021

Enlace a la memoria en formato .pdf:

https://hdvirtual.us.es/discovirt/index.php/s/aa2Z62yJzoYn4pL

¡Muchas gracias por vuestra atención!

¿Alguna duda? :-)

Beatriz Galiana Carballido
Fernando Sánchez Paredes