



UNIVERSIDAD
DE GRANADA

Fractales

Fundamentos Matemáticos y
Visualización con Ray-Tracing

Juan Antonio Villegas Recio

27 de junio de 2022

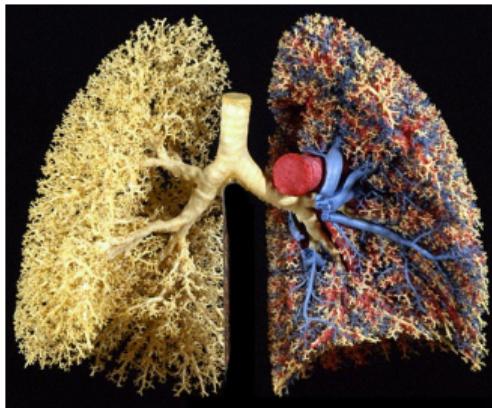
El concepto de ‘fractal’

El concepto de ‘fractal’

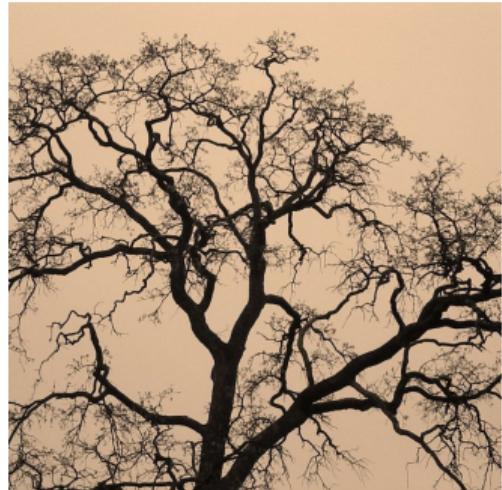
Estructuras que siguen patrones repetitivos



Helecho



Pulmones



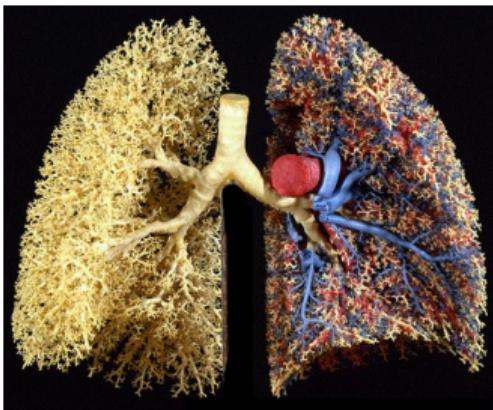
Árbol

El concepto de ‘fractal’

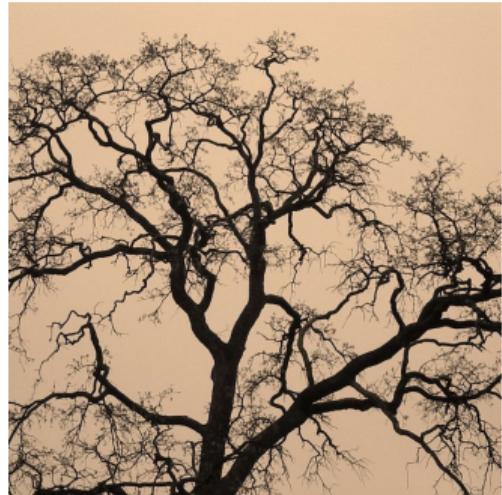
Estructuras que siguen patrones repetitivos



Helecho



Pulmones



Árbol

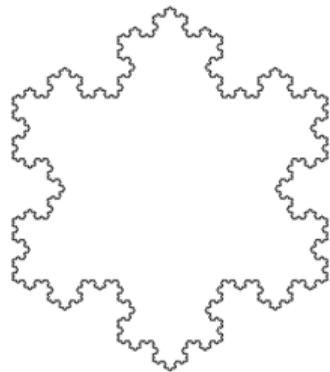
Autosimilaridad

El concepto de ‘fractal’

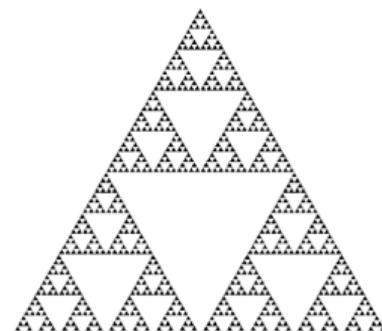
Fractales clásicos



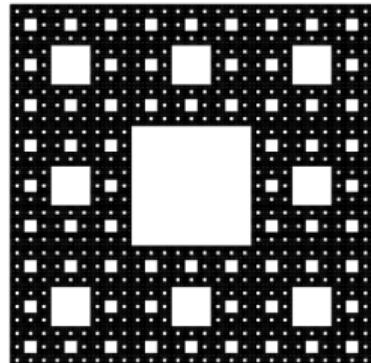
Conjunto de Cantor



Copo de Koch



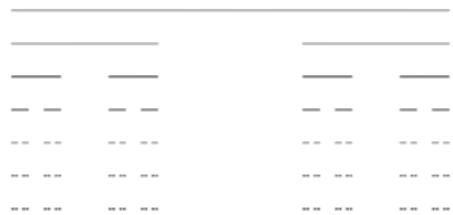
Triángulo de Sierpinski



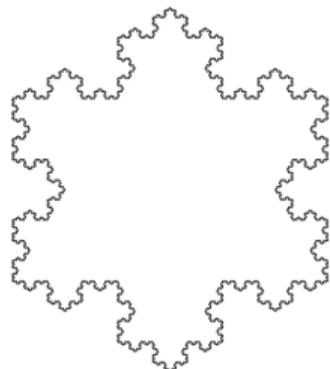
Alfombra de Sierpinski

El concepto de ‘fractal’

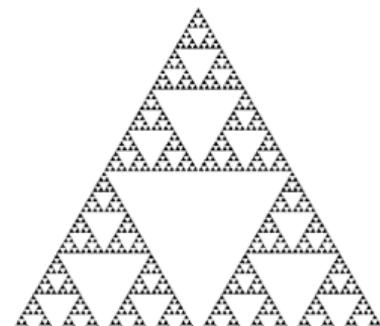
Fractales clásicos



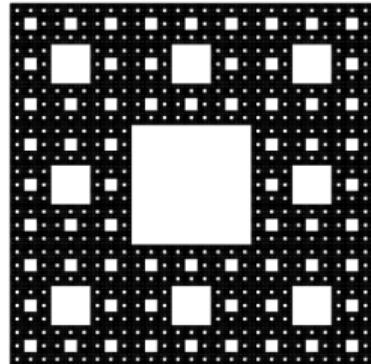
Conjunto de Cantor



Copo de Koch



Triángulo de Sierpinski



Alfombra de Sierpinski

Autosimilares y su dimensión no es un número entero.

El concepto de ‘fractal’

Definición de fractal según B. Mandelbrot



Un fractal es un subconjunto de \mathbb{R}^n que es autosimilar y cuya dimensión fractal excede a su dimensión topológica.

Benoit Mandelbrot



Generación de fractales

Generación de fractales



Existen distintas formas de generar fractales, pero todas se basan en la misma idea.

Generación de fractales



Existen distintas formas de generar fractales, pero todas se basan en la misma idea.

La Iteración

Generación de fractales



Existen distintas formas de generar fractales, pero todas se basan en la misma idea.

La Iteración

- ▶ Iteración de funciones complejas.
- ▶ Iteración de aplicaciones contractivas (SFI).
- ▶ Métodos iterativos de visualización (Sphere-Tracing).

Iteración de funciones complejas

Iteración de funciones complejas



Sean $f : \mathbb{C} \longrightarrow \mathbb{C}$ una función compleja y $z_0 \in \mathbb{C}$.

Iteración de funciones complejas



Sean $f : \mathbb{C} \rightarrow \mathbb{C}$ una función compleja y $z_0 \in \mathbb{C}$.

Sucesión de iteradas:

$$\{f^n(z_0)\} = z_0, f(z_0), f(f(z_0)), \dots$$

Iteración de funciones complejas



Sean $f : \mathbb{C} \rightarrow \mathbb{C}$ una función compleja y $z_0 \in \mathbb{C}$.

Sucesión de iteradas:

$$\{f^n(z_0)\} = z_0, f(z_0), f(f(z_0)), \dots$$

Órbita de z_0 :

$$O_f(z_0) = \{f^n(z_0) : n \in \mathbb{N}\}$$

Iteración de funciones complejas



Sean $f : \mathbb{C} \rightarrow \mathbb{C}$ una función compleja y $z_0 \in \mathbb{C}$.

Sucesión de iteradas:

$$\{f^n(z_0)\} = z_0, f(z_0), f(f(z_0)), \dots$$

Órbita de z_0 :

$$O_f(z_0) = \{f^n(z_0) : n \in \mathbb{N}\}$$

- ▶ **Distintos comportamientos** de la sucesión de iteradas en cada $z_0 \in \mathbb{C}$: convergente, divergente, cíclica o caótica.
- ▶ Dependiendo de dicho comportamiento **asignamos un color**.

Iteración de funciones complejas

Teorema del punto fijo de Banach



Teorema

- ▶ X espacio métrico **completo**.
- ▶ $f : X \rightarrow X$ una aplicación **contractiva**

Entonces

- ▶ f tiene un **único punto fijo** x^* .
- ▶ La sucesión de iteradas $\{f^n(x_0)\}$ converge a x^* para cualquier $x_0 \in X$ inicial.

Iteración de funciones complejas

Teorema del punto fijo de Banach



Teorema

- ▶ X espacio métrico **completo**.
- ▶ $f : X \rightarrow X$ una aplicación **contractiva**

Entonces

- ▶ f tiene un **único punto fijo** x^* .
- ▶ La sucesión de iteradas $\{f^n(x_0)\}$ converge a x^* para cualquier $x_0 \in X$ inicial.

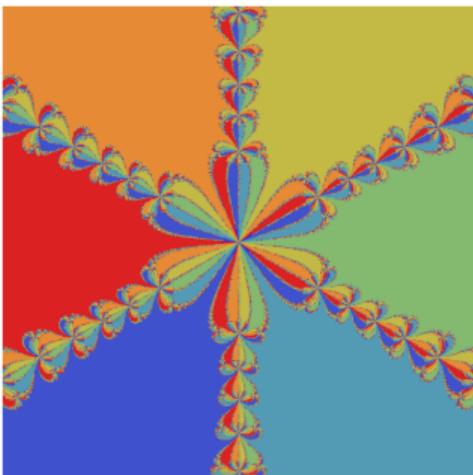
Aplicaciones del teorema del punto fijo: Método de Newton y Sistemas de Funciones Iteradas.

Iteración de funciones complejas

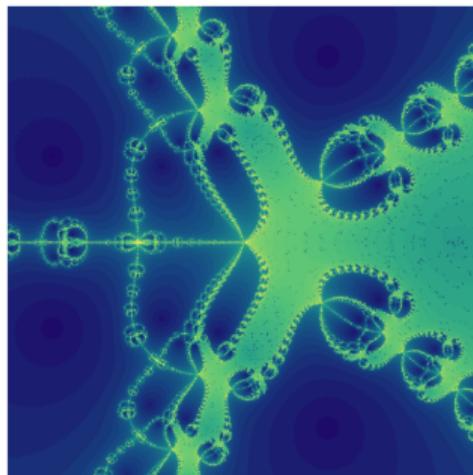


Método de Newton

El **método de Newton** es un proceso iterativo utilizado para calcular raíces de una función $f : \mathbb{C} \rightarrow \mathbb{C}$ a partir de un término $z_0 \in \mathbb{C}$.



Según la raíz a la que converge



Según las iteraciones dadas

Iteración de funciones complejas

Conjuntos de Julia



Presentamos la familia de funciones

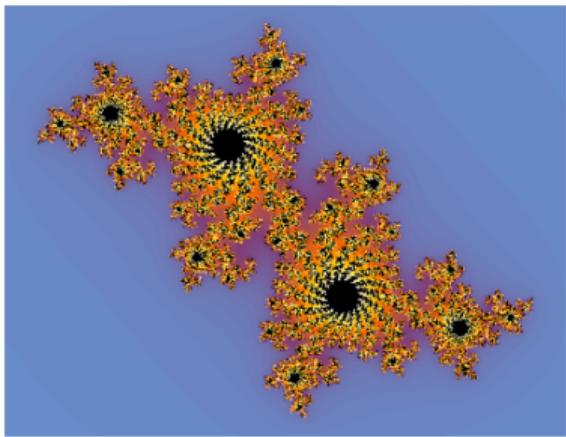
$$P_c(z) = z^2 + c$$

con $z, c \in \mathbb{C}$.

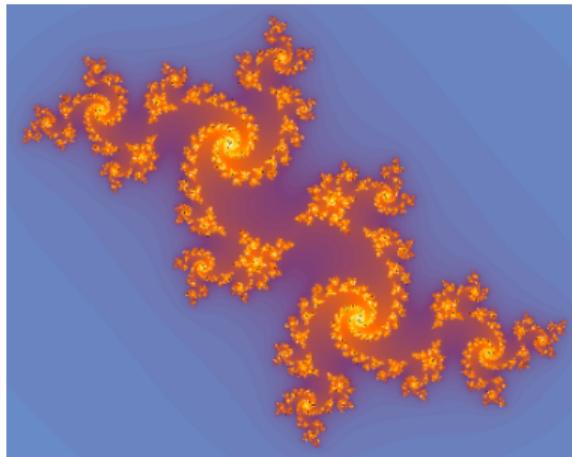
Si fijamos una constante $c \in \mathbb{C}$ e iteramos $P_c(z_0)$ para cada $z_0 \in \mathbb{C}$ obtenemos los llamados **Conjuntos de Julia** \mathcal{J}_c .

Iteración de funciones complejas

Conjuntos de Julia



$$J_{-0,23+0,65i}$$



$$J_{-0,23+0,69i}$$

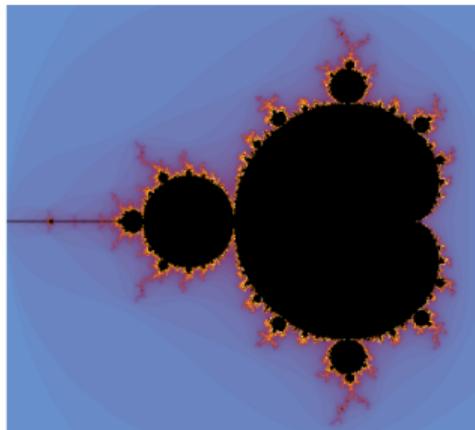
Por tanto, hay tantos conjuntos de Julia como números complejos.

Iteración de funciones complejas

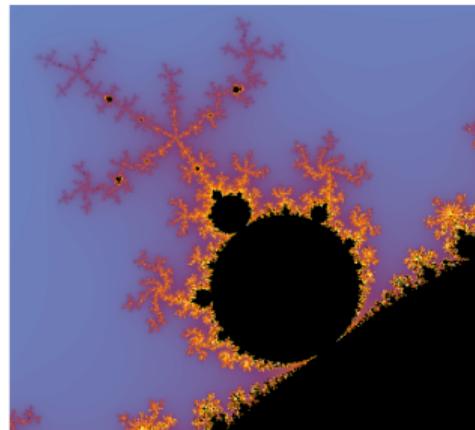
Conjunto de Mandelbrot



Si fijamos una condición inicial $z_0 = 0$ e iteramos para cada posible valor de c visualizaremos el **Conjunto de Mandelbrot** \mathcal{M} .



Conjunto de Mandelbrot



Detalle ampliado de \mathcal{M}

Sistemas de Funciones Iteradas

Sistemas de Funciones Iteradas

Definición



Un **Sistema de Funciones Iteradas (SFI)** está formado por:

- ▶ Un espacio métrico completo X .
- ▶ Un conjunto finito de aplicaciones contractivas $\{w_1, \dots, w_n\}$.

Dado un conjunto compacto $A \subset X$, consideramos aplicación W definida como

$$W(A) = \bigcup_{i=1}^n w_i(A)$$

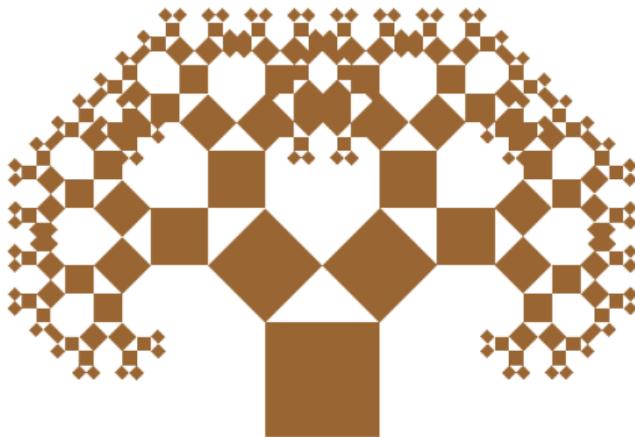
En particular, en \mathbb{R}^2 y mediante transformaciones afines contractivas podemos construir SFI.

Sistemas de Funciones Iteradas

Convergencia de SFI



La iteración infinita de la aplicación W finalmente converge a un **atractor**.



Árbol de Pitágoras



Helecho de Barnsley

Sistemas de Funciones Iteradas

El problema inverso



Obtener imágenes a partir de SFI ✓

¿Obtener SFI a partir de imágenes?

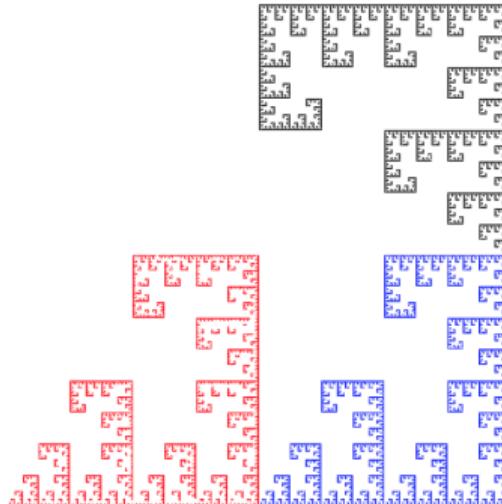
Sistemas de Funciones Iteradas

El problema inverso



Obtener imágenes a partir de SFI ✓

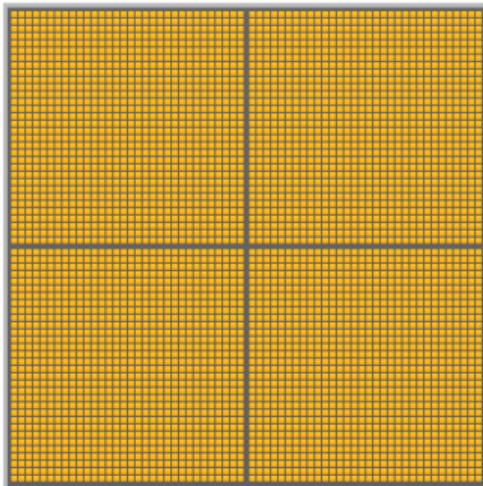
¿Obtener SFI a partir de imágenes? Solución: **El teorema del Collage.**



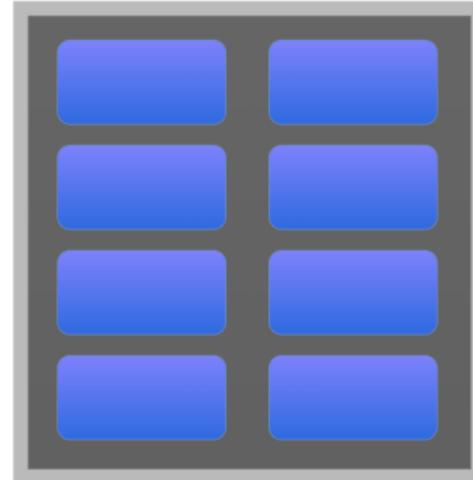
Uso de WebGL

Uso de WebGL

GPU vs CPU



GPU



CPU

WebGL utiliza CPU (JavaScript) y GPU (GLSL)

Visualización de fractales 2D

Visualización de fractales 2D

Conjuntos de Julia y Mandelbrot



Posibilidad de identificar la pantalla con el plano complejo:

Visualización de fractales 2D

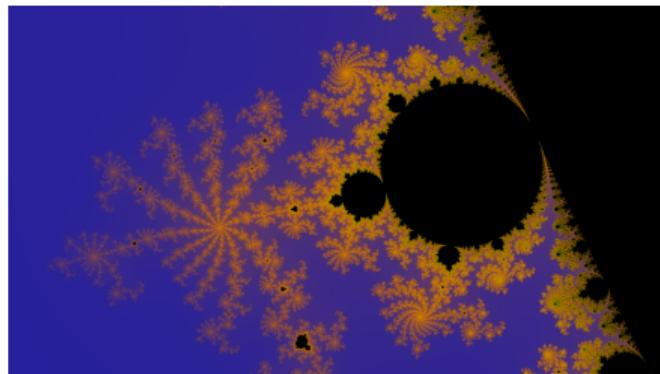
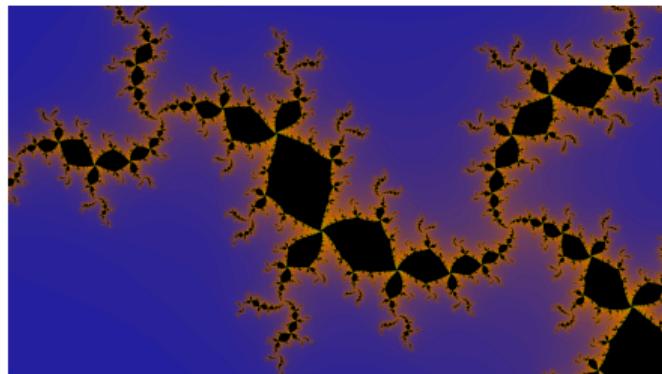
Conjuntos de Julia y Mandelbrot



Posibilidad de identificar la pantalla con el plano complejo:

Cada píxel es un número complejo

Transcribimos el algoritmo a código GLSL y el resultado es ...



Ray-Tracing

Ray-Tracing

Fundamentos



- ▶ **Rasterización:** Identificar qué píxeles ocupan los objetos de la escena.
- ▶ **Ray-Tracing:** Identificar qué objeto se proyectan sobre cada píxel.

Ray-Tracing

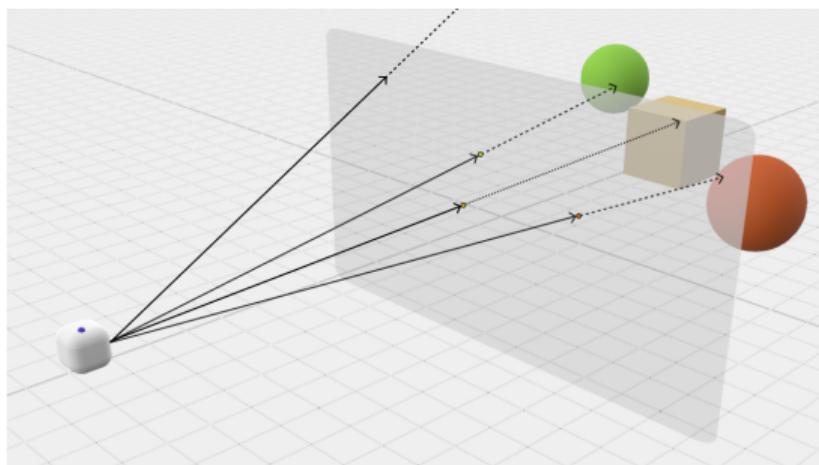
Fundamentos



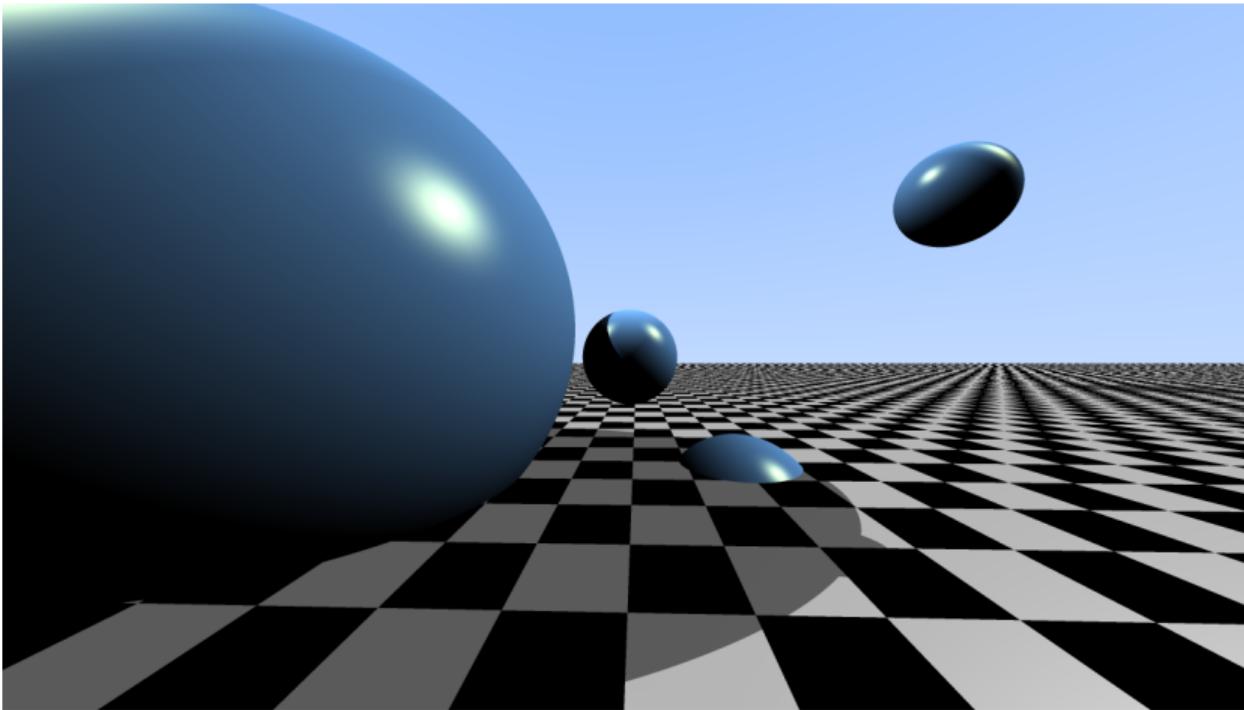
- ▶ **Rasterización:** Identificar qué píxeles ocupan los objetos de la escena.
- ▶ **Ray-Tracing:** Identificar qué objeto se proyectan sobre cada píxel.

Componentes de un programa
'ray-tracer':

- ▶ Generación de rayos primarios
- ▶ Cálculo de intersecciones
- ▶ Modelo de iluminación local



Ray-Tracing



Visualización de fractales 3D

Visualización de fractales 3D

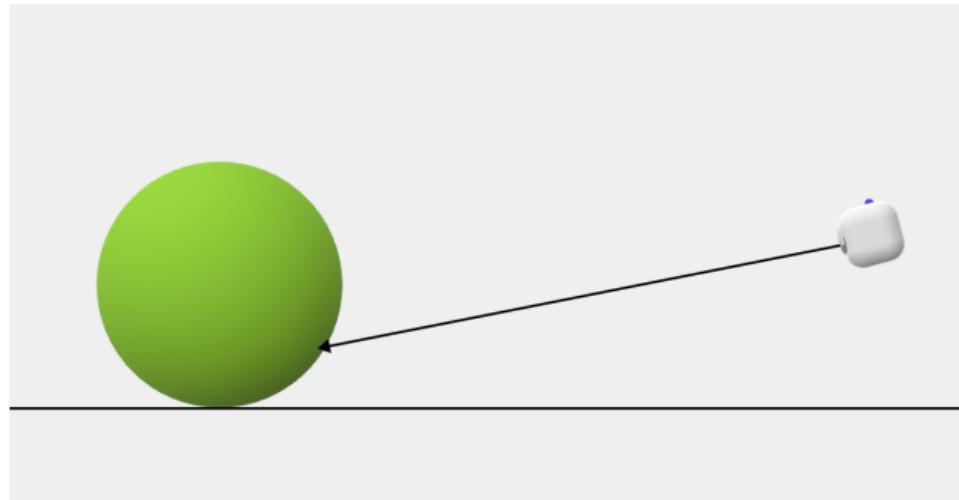
Sphere-Tracing



Es sencillo calcular la intersecciones rayo-esfera y rayo-plano, pero

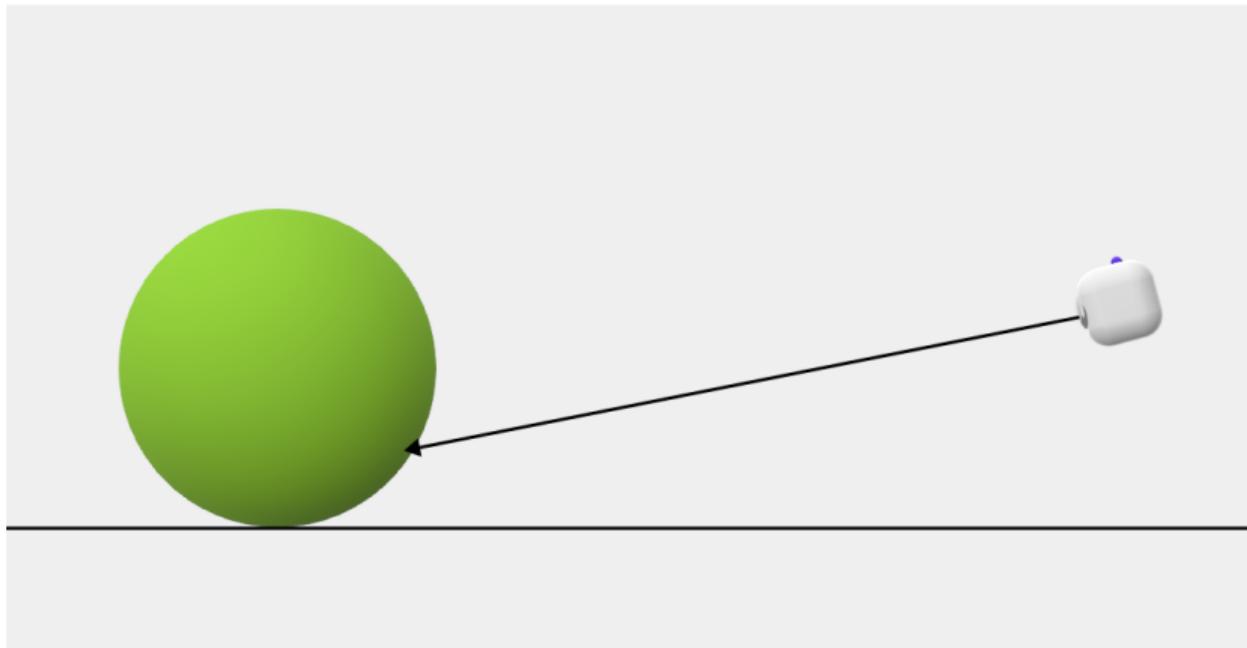
¿Cómo calculamos la intersección con un fractal?

Solución: Con el algoritmo **Sphere-Tracing**.



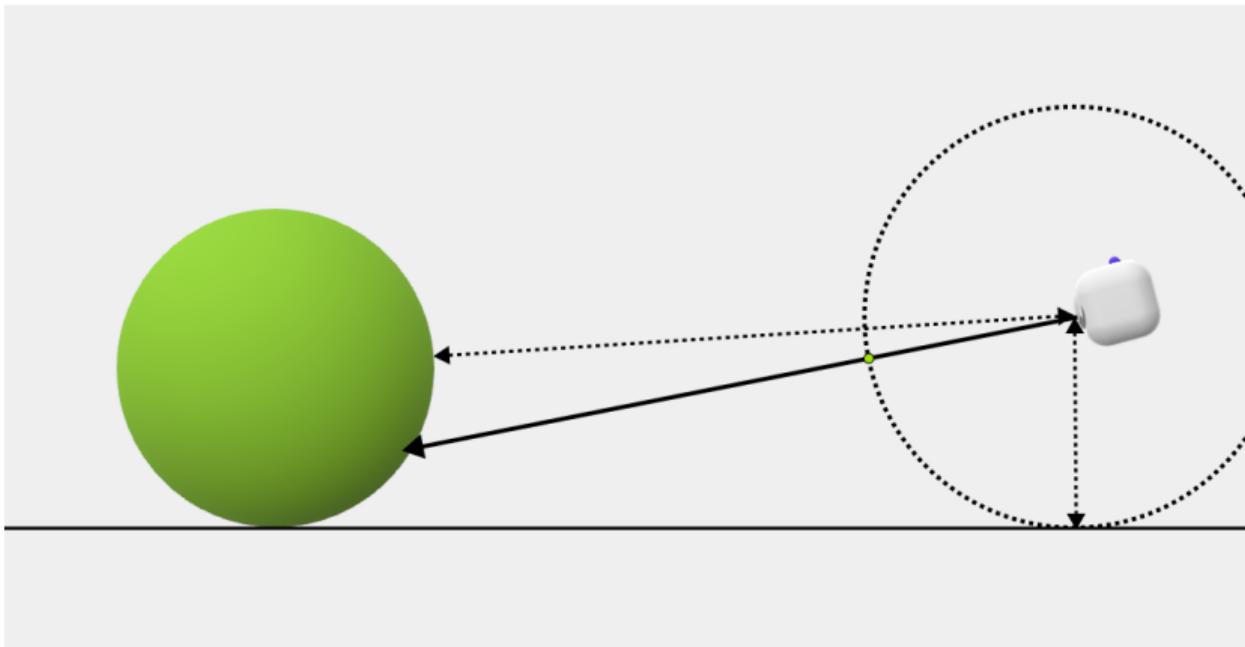
Visualización de fractales 3D

Sphere-Tracing



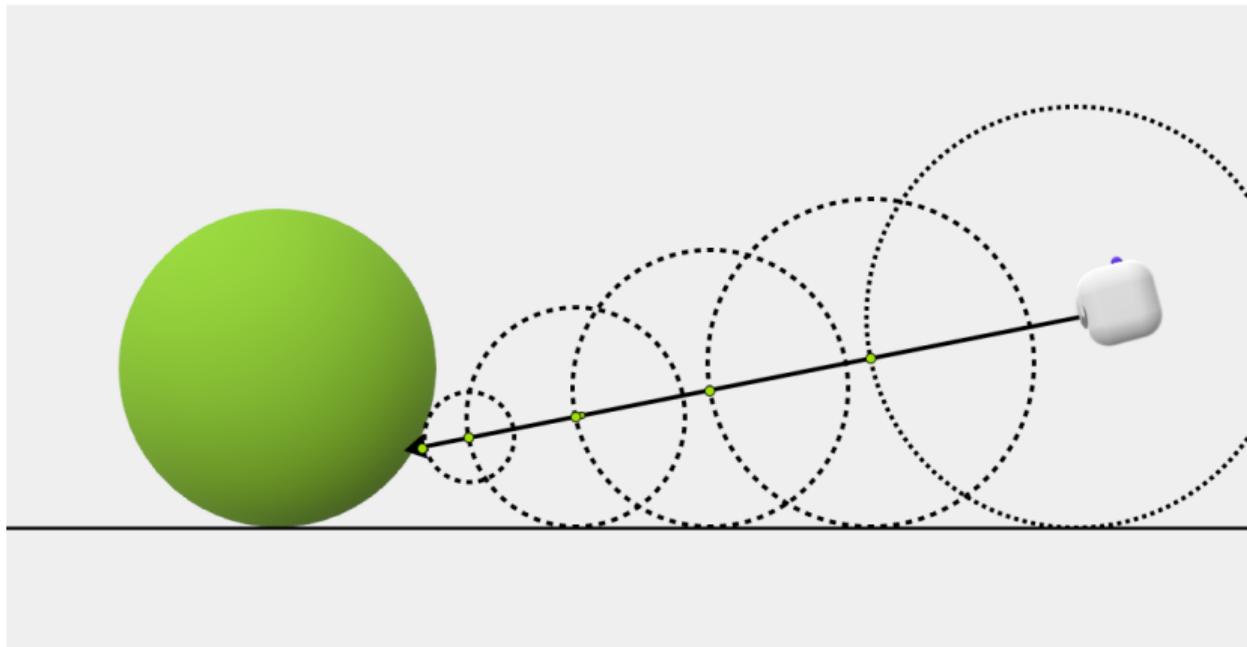
Visualización de fractales 3D

Sphere-Tracing



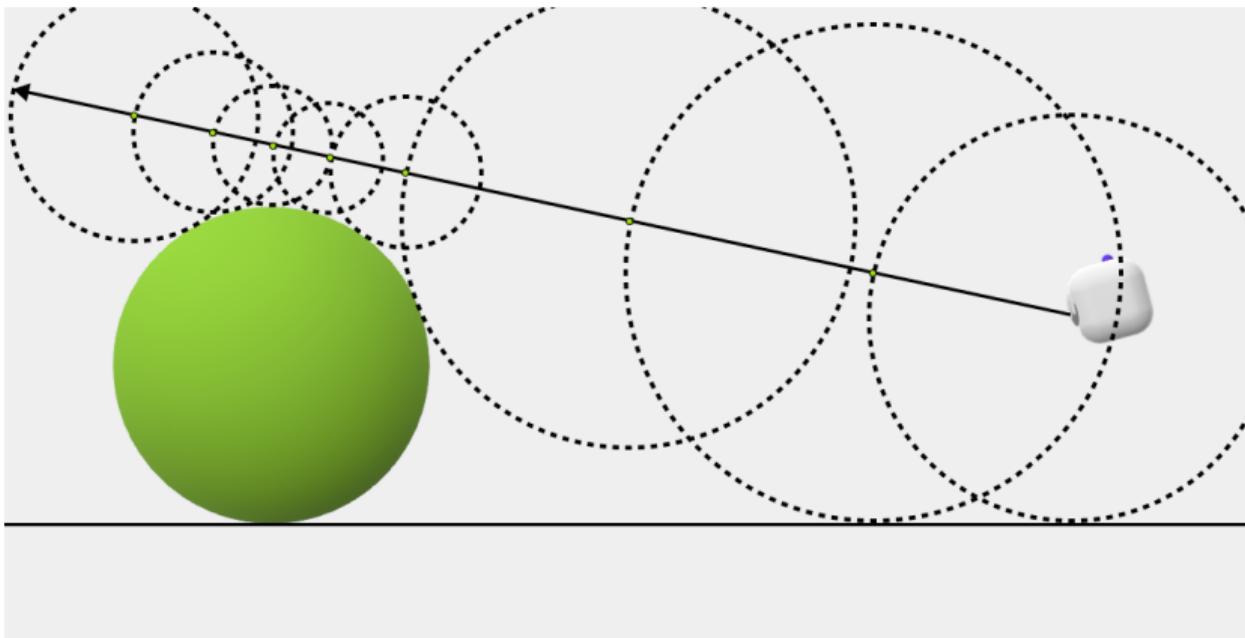
Visualización de fractales 3D

Sphere-Tracing



Visualización de fractales 3D

Sphere-Tracing



Visualización de fractales 3D

BDFs y SDFs



Necesitamos estimar la distancia de un punto cualquiera al fractal... o al menos una cota inferior.

Visualización de fractales 3D

BDFs y SDFs



Necesitamos estimar la distancia de un punto cualquiera al fractal... o al menos una cota inferior.

Para ello contamos con las **BDF** (Bounding Distance Function), que son funciones $f : \mathbb{R}^3 \longrightarrow \mathbb{R}$ tales que

$$|f(x)| \leq d(x, f^{-1}(0)) \quad \forall x \in \mathbb{R}^3$$

Es decir, acotan la distancia a la superficie implícita que definen.

Si se cumple la igualdad, entonces se les denomina **SDF** (Signed Distance Function).

Visualización de fractales 3D

Conjuntos de Julia y Mandelbrot 3D



Generalización de los complejos \mathbb{C} : los **cuaternios** \mathbb{H} .

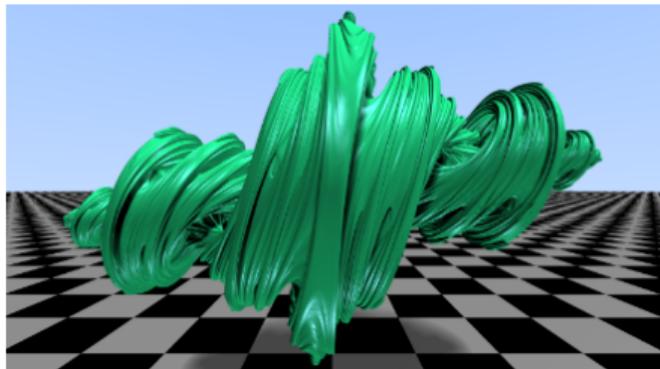
Visualización de fractales 3D

Conjuntos de Julia y Mandelbrot 3D

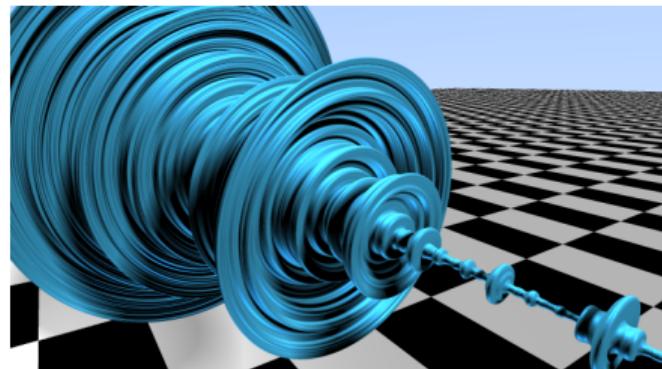


Generalización de los complejos \mathbb{C} : los **cuaternios** \mathbb{H} .
Podemos extender la función $P_c(z) = z^2 + c$ a

$$P(q) = q^2 + c, \text{ con } q, c \in \mathbb{H}$$



$\mathcal{J}_{-0,71-0,31i}$



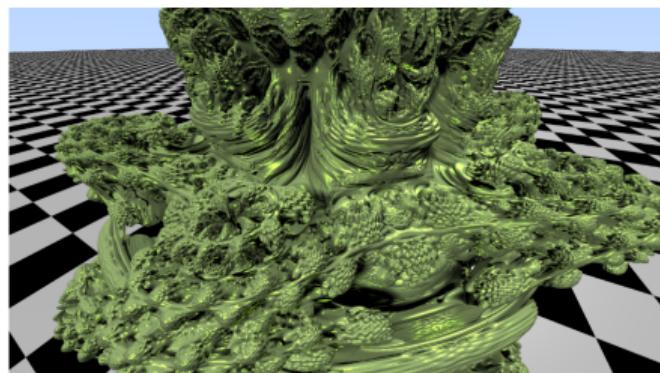
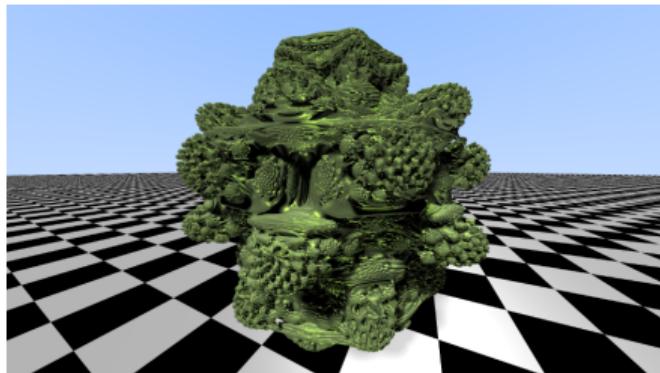
\mathcal{M} generalizado

Visualización de fractales 3D

Conjunto de Mandelbub



Generalización de \mathcal{M}_8 (el conjunto de Mandelbrot utilizando la función $z^8 + c$) utilizando aritmética poco rigurosa.



Conclusión

Conclusión



La geometría fractal tiene aplicaciones en:

- ▶ Desarrollo de videojuegos y producción de películas
- ▶ En otras áreas de las matemáticas como la estimación de parámetros mediante ecuaciones diferenciales e integrales
- ▶ Dinámica de poblaciones
- ▶ Procesos biológicos
- ▶ ...

Diferencia entre la generación de imágenes en CPU y el uso de la GPU.

El siguiente paso sería optimizar la aplicación para que pudiera ejecutarse en tiempo real en ordenadores más convencionales.

Bibliografía fundamental I



-  Rubiano, G.N. (2013). *Iteración y Fractales (con Mathematica ®)*. Universidad Nacional de Colombia.
-  Mandelbrot, B. (1983). *The Fractal geometry of nature*. Freeman, New York.
-  Barnsley, M (1993). *Fractals everywhere*. Academic Press, 2nd edition.
-  Shirley, P. (2020). *Ray Tracing in One Weekend*. Recuperado 28 de mayo de 2022, de <https://raytracing.github.io/books/RayTracingInOneWeekend.html>.
-  Hart, J., Sandin, D. y Kauffman, L. (1989). Ray tracing deterministic 3-D fractals. *ACM SIGGRAPH Computer Graphics*, 23(3):289–296. Disponible en <https://doi.org/10.1145/74334.74363>.

Bibliografía fundamental II



-  Hart, J. (1995). Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces. *The Visual Computer*, 12:527–545. Disponible en <https://doi.org/10.1007/s003710050084>.
-  Crane, K. (2005). Ray Tracing Quaternion Julia Sets on the GPU. *University of Illinois at Urbana-Champaign*. Disponible en <https://www.cs.cmu.edu/~kmcrane/Projects/QuaternionJulia/paper.pdf>.
-  Douady, A. y Hubbard, J. (2009). *Exploring the Mandelbrot set. The Orsay Notes*. Cornell University, Ithaca, NY. Disponible en <https://pi.math.cornell.edu/~hubbard/OrsayEnglish.pdf>.



¡Gracias por escuchar!

The screenshot shows a web page titled "Visualización de Fractales". At the top, there is a navigation bar with links for "Home", "Fractales 2D", and "Fractales 3D". Below the navigation, there are two large images: one labeled "Fractales 2D" showing a 2D fractal pattern, and another labeled "Fractales 3D" showing a 3D fractal landscape. A small explanatory text at the bottom left states: "Esta web ha sido desarrollada como parte del 'Trabajo de Fin de Grado' de Juan Antonio Villegas Recio, alumno de Ingeniería Informática y Matemáticas en la universidad de Granada, durante el curso 2021/2022. Si lo deseas, puedes consultar la memoria, la documentación o el código fuente del proyecto en GitHub." A blue button at the bottom left says "Ir al proyecto en GitHub".

Web disponible en
[jantoniov.github.io/
Geometria-Fractal/](https://jantoniov.github.io/Geometria-Fractal/)

The screenshot shows a GitHub repository page for "JAntonioVR/Geometria-Fractal". The repository is described as "TFC de Juan Antonio Villegas Recio". It includes links to the GitHub page (jantoniov.github.io/Geometria-Fractal/) and the GitHub repository ([github.com/JAntonioVR/
Geometria-Fractal](https://github.com/JAntonioVR/Geometria-Fractal)). On the right side of the page, there is a large, detailed fractal image.

Repositorio de GitHub disponible en
[github.com/JAntonioVR/
Geometria-Fractal](https://github.com/JAntonioVR/Geometria-Fractal)