

Estimación de curvatura para coloración de sistemas dinámicos en el plano complejo

Jacobo Hernández Varela

2 de diciembre de 2019

Índice

1. Introducción	1
2. Conceptos Fundamentales	2
2.1. Iteraciones y Órbitas	2
2.2. Fractales	3
2.2.1. Funciones de Coloración y Paleta	3
2.2.2. Calculando Imágenes Fractales	4
3. Coloraciones Promedio	4
3.1. La Curvatura Promedio para Coloración	4
4. Resultados	5
Referencias	11

1. Introducción

Este trabajo revisa conceptos de las matemáticas para un algoritmo seleccionado que se utiliza para colorear imágenes fractales. En general, los algoritmos de coloración fueron desarrollados para ser instrumentos de el arte fractal. Javier Barrallo y Damien Jones describen el nacimiento del arte fractal y el papel de los algoritmos de coloración de la siguiente manera [2].

Durante la década de 1980, los entusiastas de los fractales comenzaron a explorar los fractales por su mérito artístico, y no por su importancia matemática. Mientras que las matemáticas eran la herramienta, el enfoque era el arte. Como la ecuación fractal en si misma era el elemento matemático más obvio, los artistas fractales experimentaron con nuevas ecuaciones, introduciendo cientos de tipos de fractales diferentes. Al elegir cuidadosamente los parámetros para refinar la forma, el color y la ubicación, estos exploradores introdujeron el concepto de arte fractal.

Después de 1995, se introdujeron pocos nuevos tipos de fractales importantes. Esto se debe a que las innovaciones más recientes en el arte fractal no provienen de cambiar la ecuación fractal, si no de nuevas formas de colorear los resultados de estas ecuaciones.

A medida que estos algoritmos de coloración pasan de simples a complejos, los artistas fractales a menudo vuelven a las ecuaciones fractales clásicas más simples. Con la mayor flexibilidad que proporcionan estos sofisticados algoritmos, hay aún más espacio para la expresión artística personal.

Se han desarrollado innumerables algoritmos de coloración. El proceso de desarrollo de un algoritmo de coloración suele combinar la experimentación con la comprensión de las matemáticas y el análisis de los resultados visuales. Sin embargo, no se requiere una comprensión matemática completa para desarrollar algoritmos interesantes y útiles.

2. Conceptos Fundamentales

Por cuestión del trabajo las definiciones que se presentan a continuación están confinadas al plano complejo en lugar de un espacio métrico arbitrario.

2.1. Iteraciones y Órbitas

La n -ésima composición de f con sigo misma es denotada por $f^n(z)$. Esto es,

$$f^n(z) = f(f(\dots f(z)\dots))$$

donde f es aplicado a z n veces. Los puntos $f(z)$, $f^2(z)$, $f^3(z)$, ... son llamados iteradas de z . Aplicando repetidamente la función a la previa iterada es llamado iteración. Dado un valor inicial z_0 y una función f , la n -ésima iteración $f^n(z_0)$ es denotado por z_n .

Definición 2.1. (*Órbita*). Dada una función $f : \mathbb{C} \rightarrow \mathbb{C}$, la órbita de un punto $z \in \mathbb{C}$ es el conjunto

$$O(z) = \{z, f(z), f^2(z), \dots\}.$$

La órbita $O(z)$ se dice que es periódica si, dados f y z , existe un número $n \in \mathbb{N}$ tal que $f^n(z) = z$. Una órbita contiene infinitos puntos si no es periódica. Una órbita $O(z)$ se escapa si las iteradas convergen a infinito.

Cuando calculamos fractales en una computadora, se necesita una representación finita de una órbita. Esto se hace limitando tanto la magnitud como el número máximo de elementos que deben contenerse en una órbita. Definimos las constantes M y N_{max} como el valor de rescate y el número máximo de iteraciones, respectivamente. Además, definimos el conjunto $\mathbb{C}_{N_{max}}$ como

$$\mathbb{C}_{N_{max}} = \bigcup_{k=1}^{N_{max}+1} \mathbb{C}^k$$

Definición 2.2. (*Órbita truncada*). Sean

$$O(z) = \{z, f(z), f^2(z), \dots\}$$

una órbita y M y $N_{max} \in \mathbb{N}$ constantes dadas. Sea \overline{N} el entero no negativo mas pequeño para el cual $|f^{\overline{N}}(z)| > M$, y definimos $N = \min\{\overline{N}, N_{max}\}$. La órbita truncada $O_T(z) \in \mathbb{C}_{N_{max}}$ es el conjunto

$$O_T(z) = \{z, f(z), f^2(z), \dots, f^N(z)\}.$$

Además del punto inicial z , la órbita truncada contiene N de las primeras N_{max} iteradas de z_n . El número de elementos de una órbita truncada es de $N + 1$. Cabe destacar que $N \leq N_{max}$ y N depende tanto de la función f como del valor inicial z .

Proposición 2.1. (*Desigualdad de Rescate*). Supongamos $1 \leq N \leq N_{max}$ y consideremos los puntos $z_{N-1} \in O_T(z)$ y $z_N \in O_T(z)$. La Desigualdad de Rescate es la desigualdad

$$|z_{N-1}| \leq M < |z_N|. \quad (2.1)$$

Según la Definición 2.2, N es el entero más pequeño para el cual $M < |z_N|$. Así $|z_{N-1}| \leq M$, por lo que la Desigualdad de Rescate es válida.

2.2. Fractales

Como ejemplo, consideremos una función

$$\begin{aligned} f &: \mathbb{C} \longrightarrow \mathbb{C} \\ z &\longmapsto z^p + c \end{aligned} \quad (2.2)$$

donde la constante $p \in \mathbb{N}$, $p \geq 2$ y la semilla $c \in \mathbb{C}$. La función $f(z)$ define el sistema dinámico

$$z_k = z_k^p + c. \quad (2.3)$$

Este sistema dinámico es conocido por presentar un comportamiento caótico; dependiendo del valor constante c y del valor inicial z_0 , las iteraciones se comportan de forma diferente.

2.2.1. Funciones de Coloración y Paleta

Una imagen se representa como una matriz de $m \times n$ de puntos discretos denominados píxeles. A cada pixel en la matriz se le asocia un color RGB. Sea z_0 un punto en el plano complejo que denota la posición correspondiente de un pixel. Para calcular el color *RGB* de un pixel en la imagen fractal, se calcula primero la órbita truncada $O_T(z_0)$. Luego la función de coloración es evaluada.

Definición 2.3. (*Función de Coloración*). Una función de coloración es una función

$$u : \mathbb{C}_{N_{max}} \longmapsto \mathbb{R} \quad (2.4)$$

que mapea la órbita truncada a la recta real.

El número de elementos $N + 1$ en el argumento de u es variable. Usualmente las implementaciones de las funciones de coloración incluyen un bucle que se ejecuta para todos los elementos de $O_T(z_0)$. La coloración $u(O_T(z_0))$ se estudia a menudo en función del punto z_0 , mientras que algunas funciones de coloración solo utilizan la última iteración z_N de $O_T(z_0)$. Esto es denotado como $u(z_N)$.

Una función de índice de color $I(u)$ mapea el valor de u de la función de coloración a un índice de color. El índice de color está contenido en el intervalo $(0, 1)$ o $[0, 1]$. La función de paleta de colores $P(I)$ es usada para mapear el índice de color a el espacio de color RGB. Los componentes rojo, verde y azul del color RGB pueden ser representados como un punto en $[0, 1]^3$.

Definición 2.4. (*Función de paleta*). Una función de paleta $P : \mathbb{R} \longmapsto \mathbb{R}^3$ mapea el índice de color I a el espacio de color RGB. Su dominio es $[0, 1]$ y su rango es $[0, 1]^3$. El dominio puede ser extendido a la recta real usando la parte decimal de I .

La función de índice de color se usa simplemente como medio para ajustar la apariencia visual de coloreo sin necesidad de cambiar la función de paleta.

2.2.2. Calculando Imagenes Fractales

El algoritmo presentado a continuación, iterando la función definida por la ecuación 2.3, muestra una imagen del *conjunto de Julia* [4]. Este algoritmo extiende el algoritmo de tiempo de escape[4] introduciendo las funciones de coloreo, indice y paleta.

Las siguientes operaciones son realizadas para cada pixel en la matriz de $m \times n$ que representa a la imagen.

1. Sea z_0 la posición correspondiente del pixel en el plano complejo.
2. Calcular la órbita truncada iterando la formula $z_n = f(z_{n-1})$ desde z_0 hasta que

- $|z_n| > M$, o
- $n = N_{max}$

donde N_{max} es el máximo número de iteraciones.

3. Usando las funciones de coloración e indice, mapea la órbita truncada resultante a un valor del índice de color.
4. Determina el color RGB del pixel usando la función paleta.

Si $|z_n| > M$, el pixel es un punto exterior. De lo contrario, si $n = N_{max}$, el punto es interior.

El *conjunto de Mandelbrot* puede ser calculado ajustado $z_0 = 0$ y c que corresponda a la posición del pixel en el plano complejo según el paso 1. Además del sistema 2.3, muchas otras funciones muestran un comportamiento caótico. También existen otros varios tipos de fractales y se calculan con algoritmos modificados o completamente diferentes.

3. Coloraciones Promedio

Dada una órbita truncada $O_T(z_0)$, introducimos la notación $Z_i^m = \{z_{i-m}, z_{i-m+1}, \dots, z_i\}$. La notación $Z_i^m(z_0) = Z_i^m$ es equivalente cuando el punto z_0 utilizado para calcular la iteración es importante. La constante m es el numero de iteradas en Z_i^m anteriores a z_i . Las coloraciones promedio calculan un promedio de la función $t : \mathbb{C}^{m+1} \mapsto \mathbb{R}$ evaluados en todos los puntos $Z_i^m(z_0)$ contenidos en una órbita truncada $O_T(z_0)$.

3.1. La Curvatura Promedio para Coloración

La curvatura promedio para coloración fue desarrollada originalmente por Damien M. Jones para Ultra Fractal en 1999.

El coloreado esta basado en la idea de aproximar la curvatura de una curva discreta. La curva discreta, en este caso, esta definida por lo puntos de la órbita truncada $O_T(z_0)$. La aproximación usada es

$$t(Z_N^2) = \left| \arg_{(-\pi, \pi]} \frac{z_n - z_{n-1}}{z_{n-1} - z_{n-2}} \right|$$

donde $\arg_{(-\pi, \pi]} z$ denota el argumento (Análisis Complejo) de z en el intervalo $(-\pi, \pi]$.

Definición 3.1. (*Curvatura Promedio para Coloración*). Definimos la función

$$\begin{aligned} t &: \mathbb{C}^3 \longrightarrow \mathbb{R} \\ Z_n^2 &\longmapsto \left| \arg_{(-\pi, \pi]} \frac{z_n - z_{n-1}}{z_{n-1} - z_{n-2}} \right| \end{aligned} \quad (3.1)$$

Se puede obtener una medida de la curvatura más completa si se promedia la ecuación 3.1 sobre todas las iteraciones Z_i^2 , para $3 \leq i \leq N$.

4. Resultados

Esta idea de coloración de sistemas dinámicos a partir de la curvatura promedio fue programada en el lenguaje de programación *Julia 1.3*, ya que este lenguaje de programación incluye tipos predefinidos tanto para números complejos como racionales, y soporta todas las operaciones matemáticas y funciones elementales estándar en ellos.

Lo primero que se realizó fue generar las paletas de colores, mostradas en la Figura 1, a utilizar para las imágenes fractales.

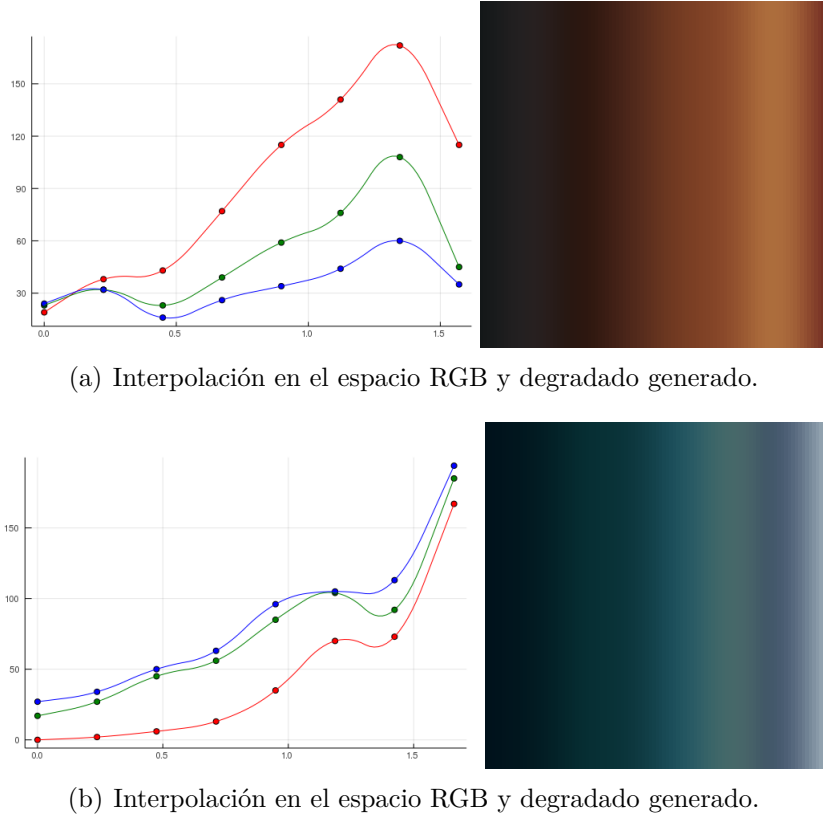


Figura 1: Paletas de colores.

La selección de gradiente es una de las elecciones artísticas más críticas para crear una buena imagen fractal. La selección de color puede enfatizar una parte de una imagen fractal mientras desestima otra. En casos extremos, dos imágenes con los mismos parámetros fractales, pero diferentes esquemas de color parecerán totalmente diferentes.

Con el algoritmo y los términos ahora definidos, todo lo que queda es ilustrar los resultados de algunas combinaciones de opciones.

La primer imagen, *At the C-shore* (Figura 2), esta basada en la función

$$f(z) = \sqrt{(1-i)z - 1}.$$

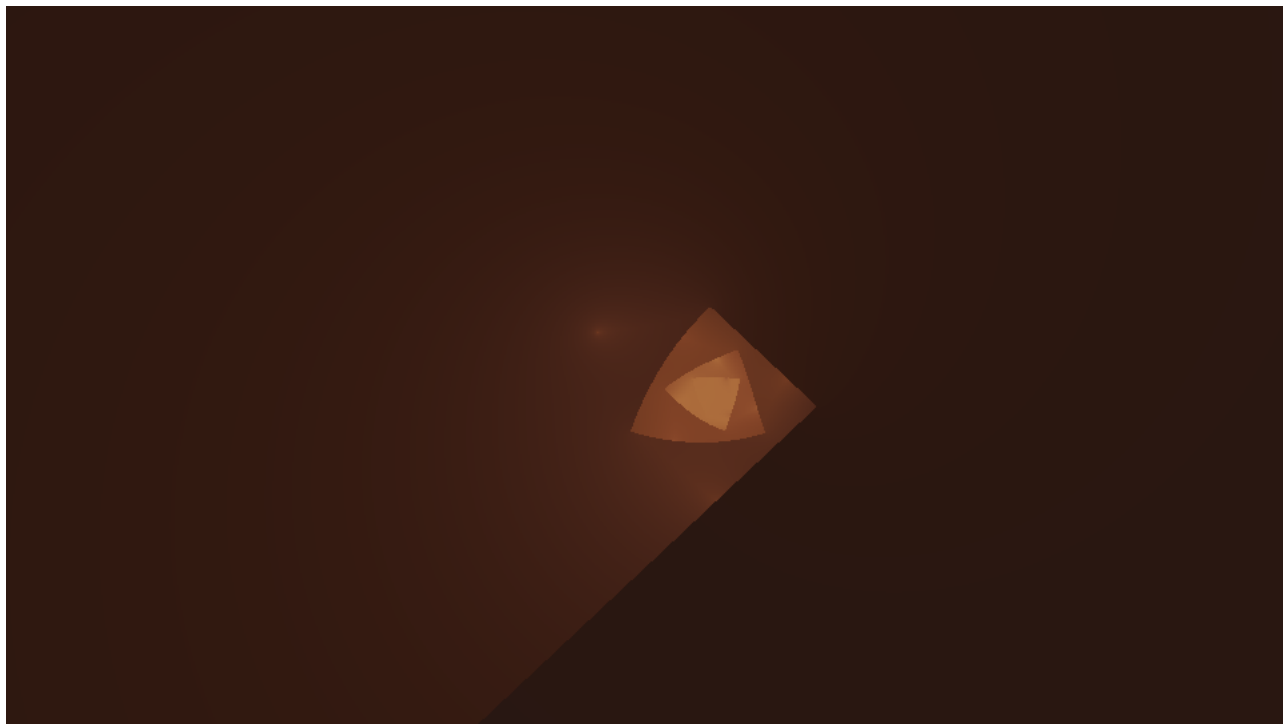


Figura 2: At the C-shore.

La iteración limite para el calculo de la curvatura es $N_{max} = 12$, y la paleta de colores es la mostrada en la Figura 1(a).

La imagen, *i of the Storm* (Figura 3), esta basada en la función

$$f(z) = \frac{(1-i)z^4 + (7+i)z}{2z^5 + 6}.$$

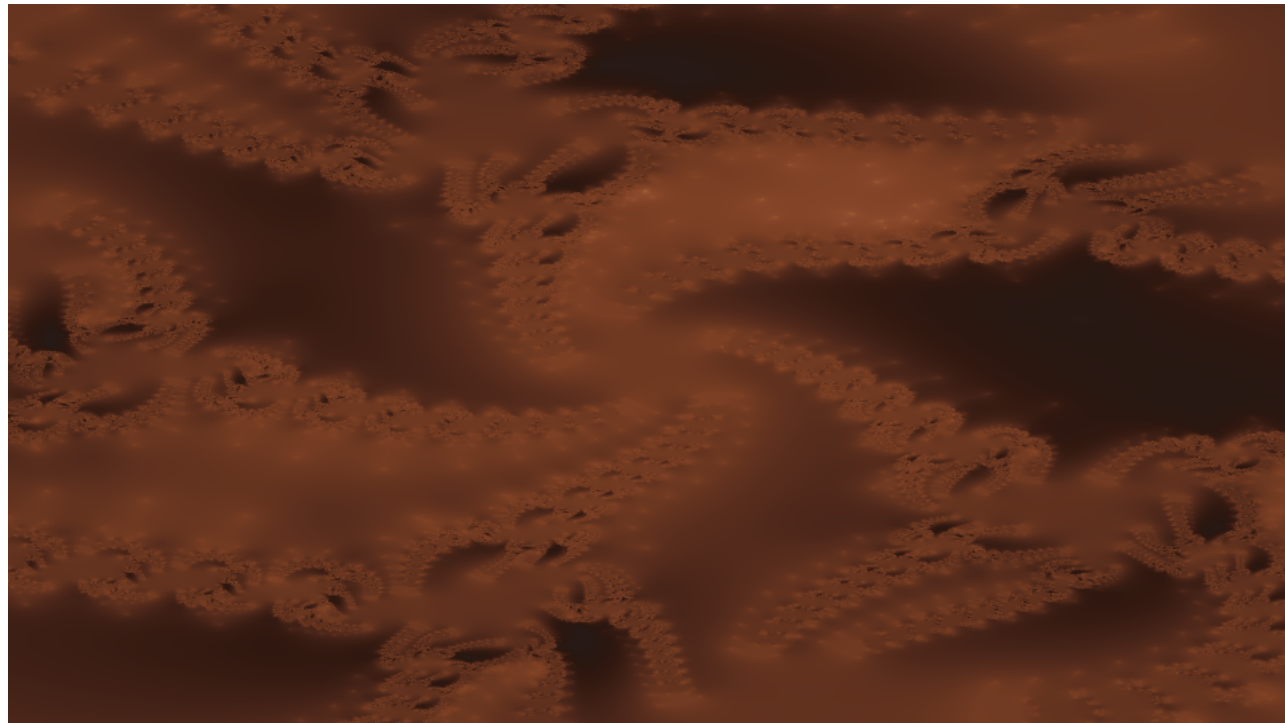


Figura 3: *i of the Storm*.

La iteración límite para el calculo de la curvatura es $N_{max} = 12$, y la paleta de colores es la mostrada en la Figura 1(a).

La siguiente imagen, *i belong to U* (Figura 4), esta basada en la función

$$f(z) = \frac{(1-i)z^6 + (7+i)z}{2z^5 + 6}.$$

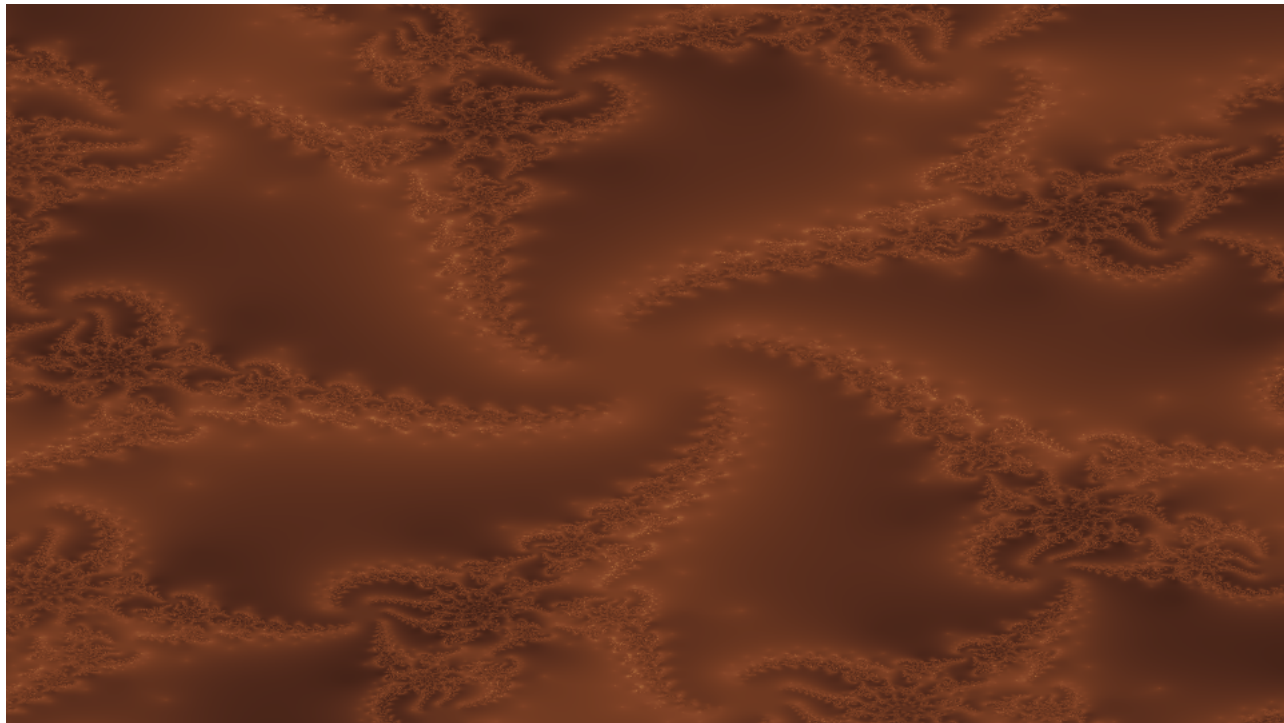


Figura 4: *i belong to U*.

La iteración límite para el cálculo de la curvatura es $N_{max} = 12$, y la paleta de colores es la mostrada en la Figura 1(a).

Ahora pasemos a algo mas conocido, *el Conjunto de Mandelbrot* (Figura 5). Este fractal esta basado en la función

$$f(z, c) = z^2 + c.$$

El procedimiento para generar este fractal fue mencionado anteriormente.

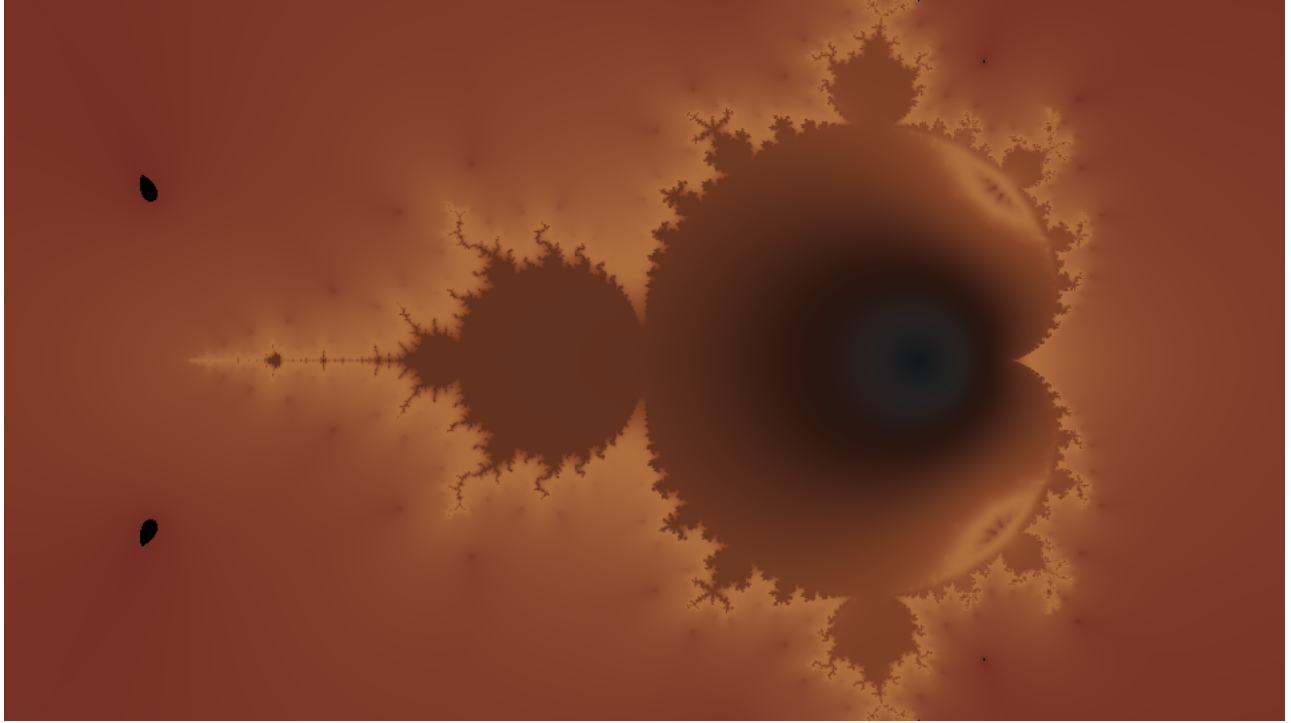


Figura 5: Mandelbrot Set.

La iteración limite para el calculo de la curvatura es $N_{max} = 32$, y la paleta de colores es la mostrada en la Figura 1(a).

La ultima imagen, *Burning Ship* (Figura 6), esta basada en la función

$$f(z, c) = (|Re(z)| + i|Im(z)|)^2 + c.$$

El procedimiento es el mismo que se utilizo para el Conjunto de Mandelbrot.

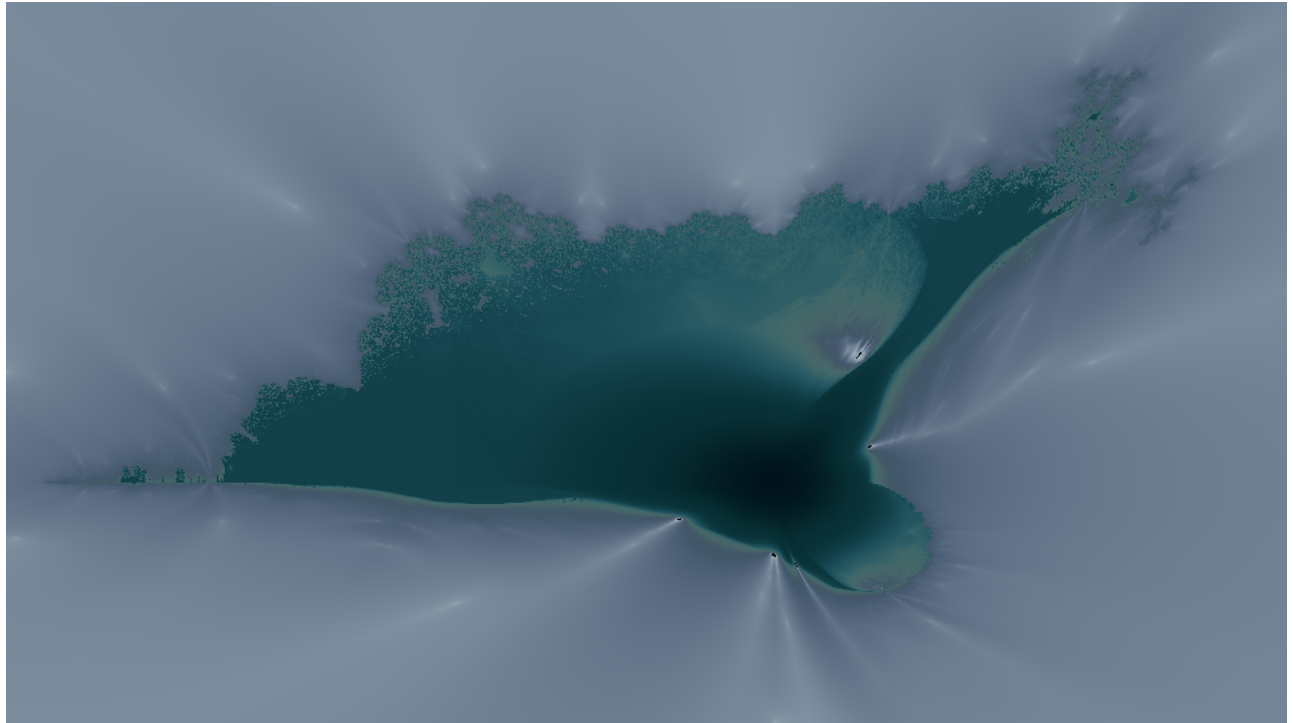


Figura 6: Burning Ship.

La iteración limite para el calculo de la curvatura es $N_{max} = 32$, y la paleta de colores es la mostrada en la Figura 1(b).

Tanto el programa como las imágenes antes mostradas pueden ser encontradas en el siguiente enlace:

https://github.com/JacobHdez/Fractal_Coloring

Referencias

- [1] P. D. SISSON “*Fractal art using variations on escape time algorithms in the complex plane*” Louisiana State University in Shreveport, USA
- [2] Javier Barrallo & Damien M. Jones “*COLORING ALGORITHMS FOR DYNAMICAL SYSTEMS IN THE COMPLEX PLANE*” The University of the Basque Country, 2009 San Sebastián. SPAIN
- [3] Jussi Härkönen “*On Smooth Fractal Coloring Techniques*” Abo Akademi University, 2007
- [4] Michael F. Barnsley “*FRACTALS EVERYWHERE*” Academic Press, 1998