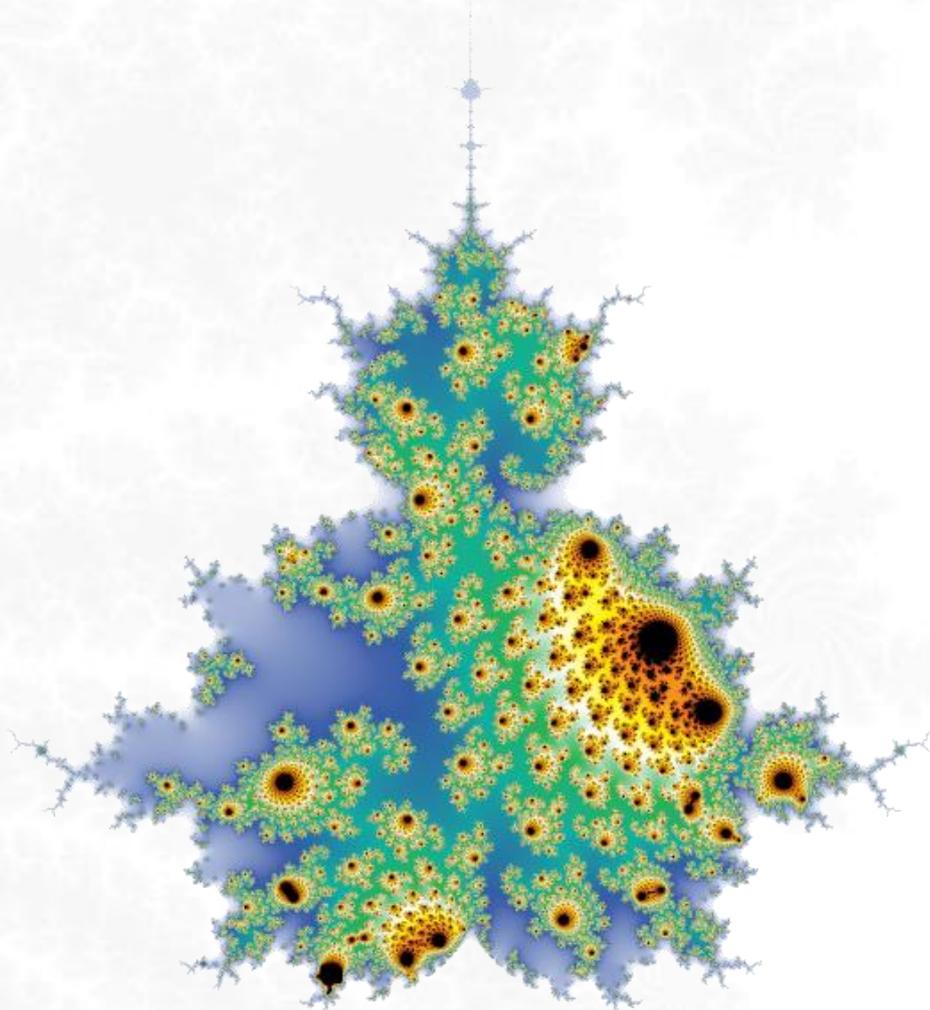


Transformación Juliter

Sergio CT



«Al ser un lenguaje, las matemáticas se pueden utilizar no sólo para informar, sino también, entre otras cosas, para seducir...»

Benoît Mandelbrot, creador de la Teoría de Fractales

<https://fractalfun.es>

Contenido

Introducción.....	3
Algoritmo.....	3
Propiedades.....	5
Reflexión.....	6

A grandes rasgos, la **Transformación Juliter** consiste en alterar el comportamiento del algoritmo de Mandelbrot, de tal forma que C adquiera el valor de una constante J en una iteración I dada. En ese momento, el algoritmo dejará de calcular el Conjunto de Mandelbrot y empleará el resto de las iteraciones en calcular el Conjunto de Julia, que quedará encajado dentro del Mandelbrot calculado hasta el momento, e influenciado por el valor en el que quedó Z tras aquellas primeras iteraciones.

Introducción

Este documento describe como aplicar la “Transformación Juliter” a fractales representados mediante el Método de Mandelbrot. Este método consiste en analizar el comportamiento de la fórmula iterativa $Z_{n+1} = f(z_n) + C$ para cada punto C del plano complejo, donde f es una función previamente elegida y el valor inicial de Z es el número complejo $(0, 0i)$.

Cuando la sucesión de valores de Z queda acotada, C se considera perteneciente al Conjunto de Mandelbrot y, por lo general, se le asigna el color negro. Por el contrario, en el caso de que la sucesión de valores de Z supere un radio de escape previamente establecido, se le asigna a C un color que depende del número de iteración en el cual la órbita de Z superó dicho límite.

Téngase también en cuenta que el Conjunto de Mandelbrot representa todos los Conjuntos de Julia conexos. Los cuales se calculan con la misma fórmula iterativa, pero en este caso Z sería la variable que recorre el plano complejo, y C tendría un valor constante previamente establecido.

Algoritmo

La “Transformación Juliter” surge durante el desarrollo de nuevas funcionalidades para la aplicación FExplorer, perteneciente al Proyecto FractalFun, y como comprobación de la hipótesis de que se pueden representar fractales con una mezcla regulable entre los métodos de Mandelbrot y de Julia.

La transformación es extremadamente fácil de comprender y de implementar. Por ejemplo, dada la función $f(z) = Z^2$, tenemos la fórmula del Conjunto de Mandelbrot $Z_{n+1} = Z_n^2 + C$, para la cual podemos implementar el siguiente algoritmo que represente el fractal:

pseudo-código:

```

CONST Palette(Lenght) AS COLOR = [custom colour list]
CONST MaxIterations AS INTEGER = 256
FOR EACH Pixel{
  VAR Iteration AS INTEGER = 0
  VAR PixelX AS INTEGER = [X coordinate of the pixel]
  VAR PixelY AS INTEGER = [Y coordinate of the pixel]
  VAR Px AS DOUBLE = [X coordinate of the pixel in the complex plane]
  VAR Py AS DOUBLE = [Y coordinate of the pixel in the complex plane]
  VAR Z as Complex = (0, 0i)
  VAR C as Complex = (Px, Py)
  VAR Bailout AS DOUBLE = 2
  VAR Colour AS COLOR = NULL
  DO WHILE Iteration < MaxIterations AND Z.Magnitude ≤ Bailout {
    Z = Z^2 + C
    Iteration += 1
  }
  IF Iteration = MaxIterations { Colour = Black }
  ELSE { Colour = Palette(Iteration MOD Palette.Lenght) }
  PLOT(PixelX, PixelY, Colour)
}

```

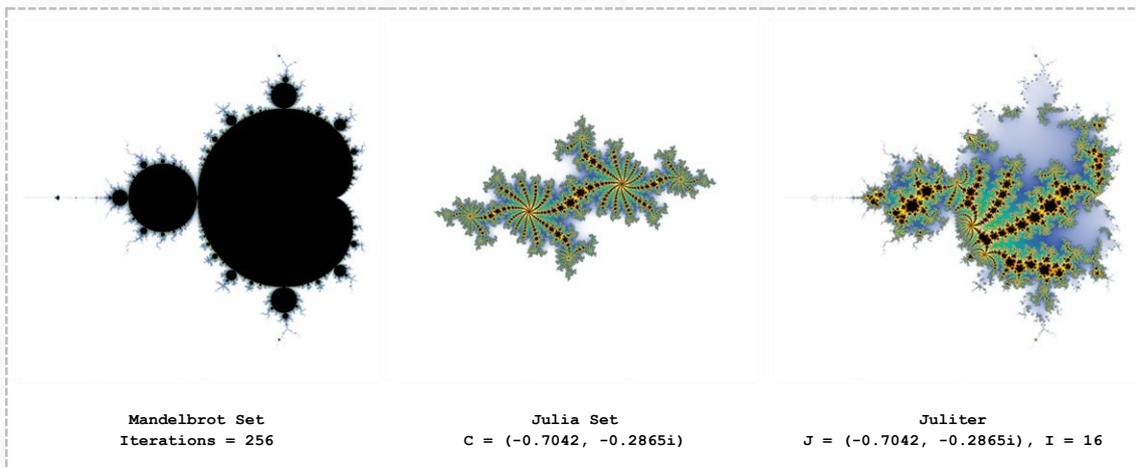
Para añadir la transformación al algoritmo descrito anteriormente necesitamos dos nuevos parámetros, que son los que dan nombre a Juliter: J , que será la constante compleja que representa el Conjunto de Julia que deseamos utilizar en la mezcla; e I , que será el valor de la iteración desde la cual se desea usar la constante J . Para ello tan sólo se debe añadir la siguiente condición después de aumentar el número de iteración en el bucle:

pseudo-código:

```
IF Iteration = I { C = J }
```

Importante: si el algoritmo necesita modificar C tras el cálculo de Z , esa modificación se deberá realizar después de la transformación, para que esta se aplique correctamente en la siguiente iteración.

En la siguiente imagen se puede ver el resultado de aplicar al Conjunto de Mandelbrot la “Transformación Juliter” con valores de $J = (-0,7042, -0.2865i)$, $I = 16$.

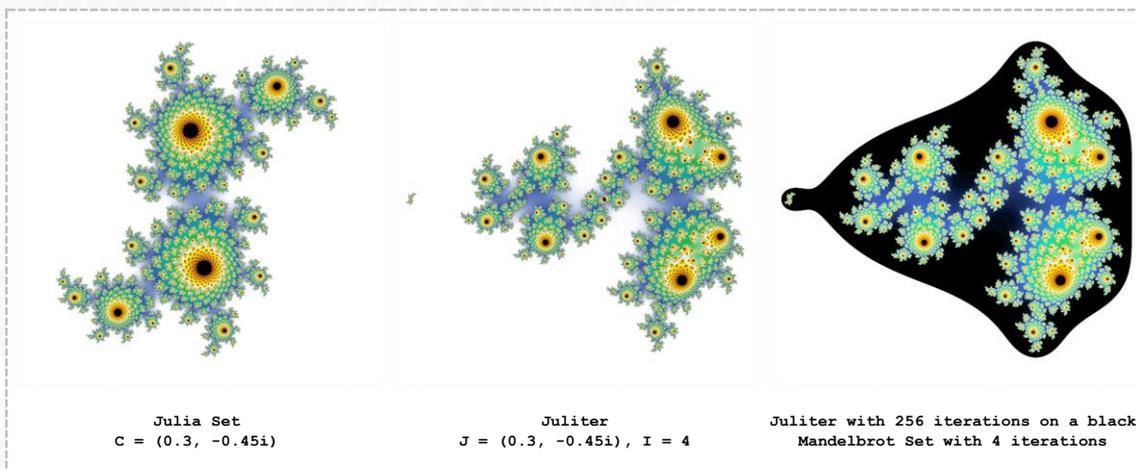


Propiedades

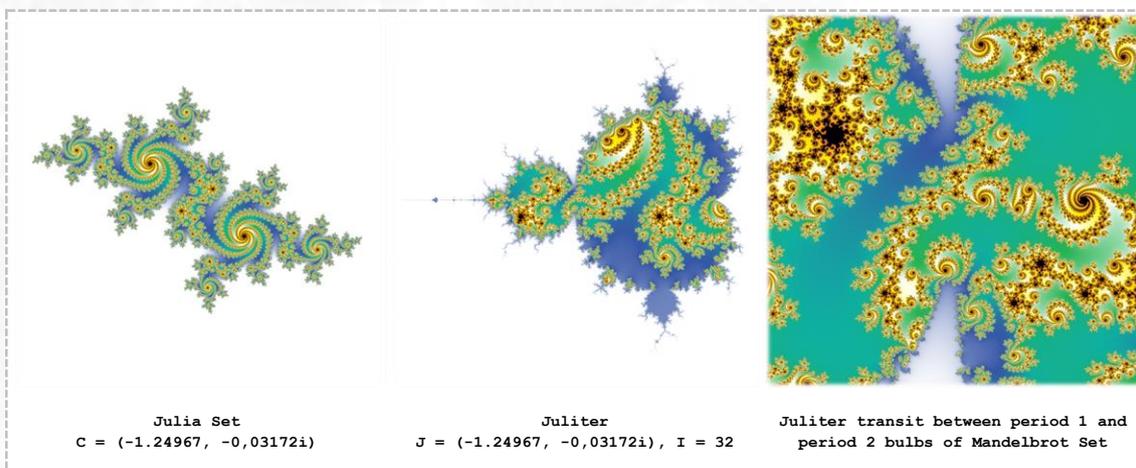
1. Si I es igual a 1 se obtendrá el Conjunto de Julia transformado* cuya identidad es J , y si I es igual al número máximo de iteraciones se obtendrá el Conjunto de Mandelbrot, quedando todos los valores intermedios de I para mezclar ambos conjuntos.

* Hay que tener en cuenta que la función se procesa antes de que se aplique la transformación por primera vez, por lo que el Conjunto de Julia resultante para $I = 1$ podría no coincidir con el Conjunto de Julia estándar para $C = J$. Su aspecto dependerá de la función y/o del algoritmo.

2. Al asignarle a C el valor constante J en la iteración I del proceso de iteración, el Conjunto de Julia Transformado quedará acotado dentro de la representación con I iteraciones del Conjunto de Mandelbrot, empleándose el resto de las iteraciones en la definición del Conjunto de Julia.



3. El Conjunto de Julia Transformado muestra continuidad al extenderse entre zonas convergentes de diferente periodo del Conjunto de Mandelbrot. Entendiéndose por “continuidad” que no aparecen cortes ni saltos en la imagen del fractal.



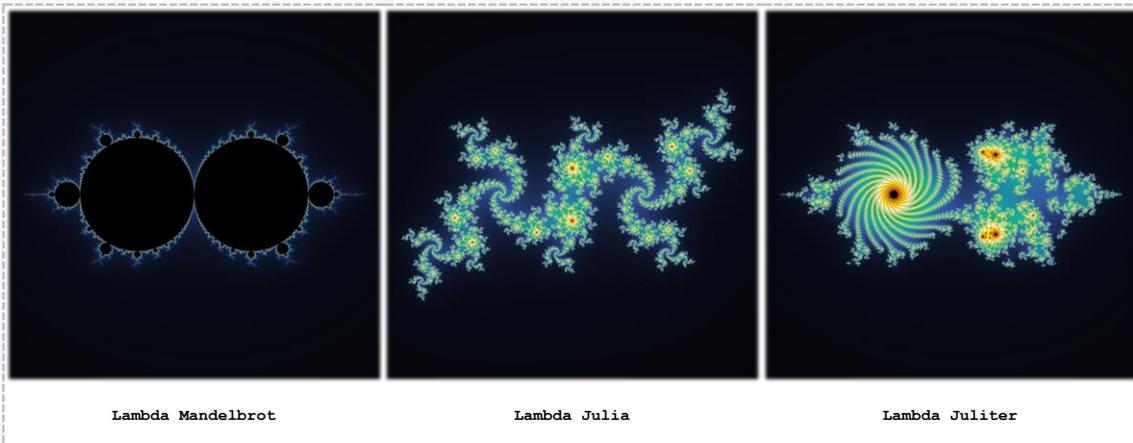
4. Dependiendo de los valores de J e I el proceso de cálculo del fractal puede agilizarse notablemente, reduciéndose el tiempo de cálculo con respecto al de un Conjunto de Mandelbrot con el mismo número de iteraciones.

Reflexión

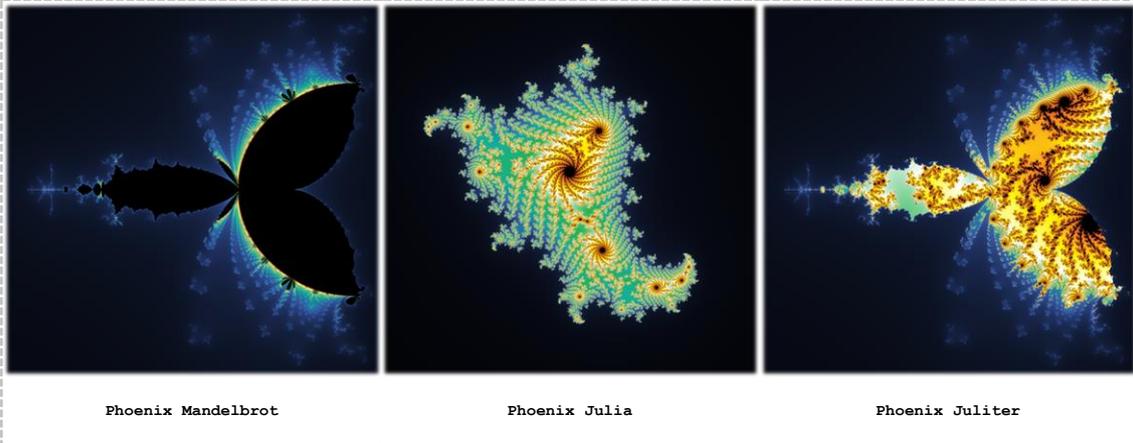
La “Transformación Juliter” es capaz de crear zonas explorables dentro de las zonas de convergencia que con algunos algoritmos de coloración carecen de interés estético. Es una transformación sencilla de implementar y por lo tanto se puede incluir en cualquier aplicación de representación gráfica de fractales sin mucho esfuerzo.

Las pruebas realizadas con FExplorer parecen indicar que la transformación ofrece buenos resultados para cualquier función manejada mediante el método de Mandelbrot. Hasta el momento se han probado con éxito las siguientes: Mandelbrot, Lambda, Phoenix, Manowar, Magnets, Burning Ship, Nova y Spider.

Lambda



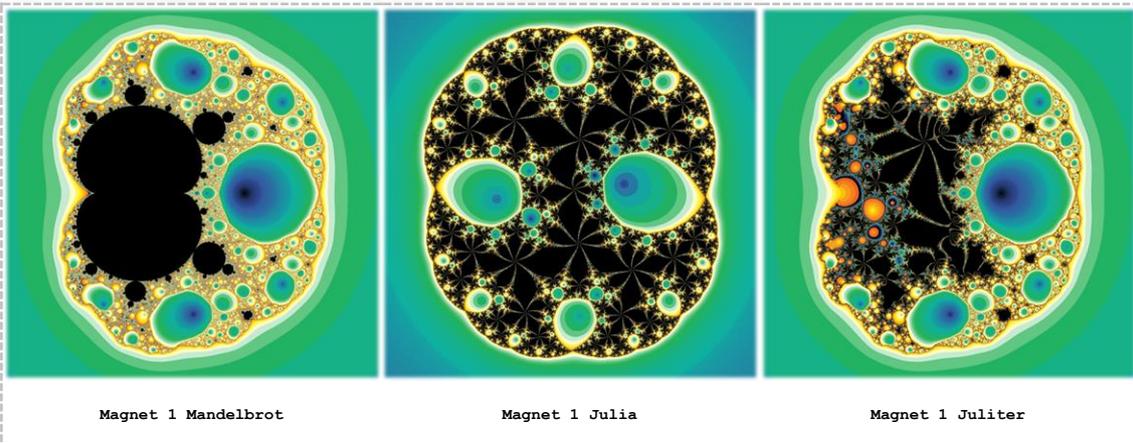
Phoenix



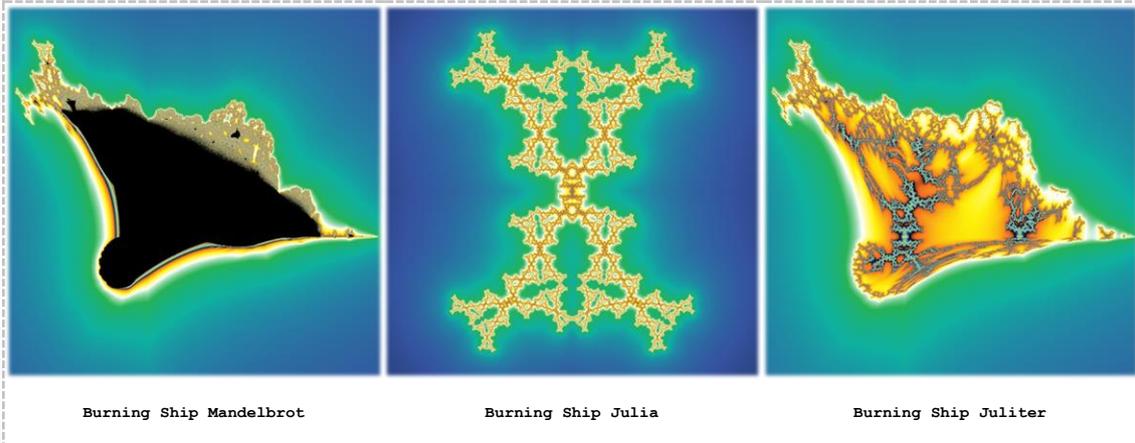
Manowar



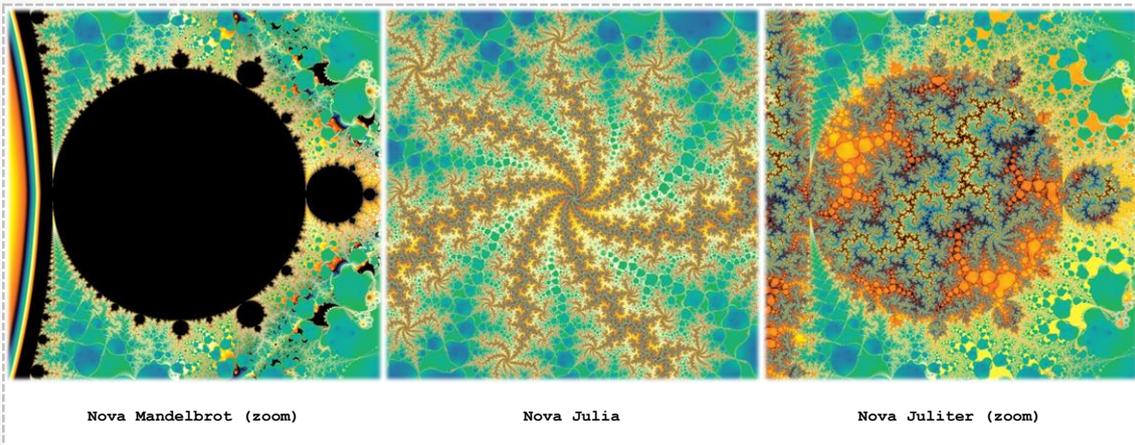
Magnet 1



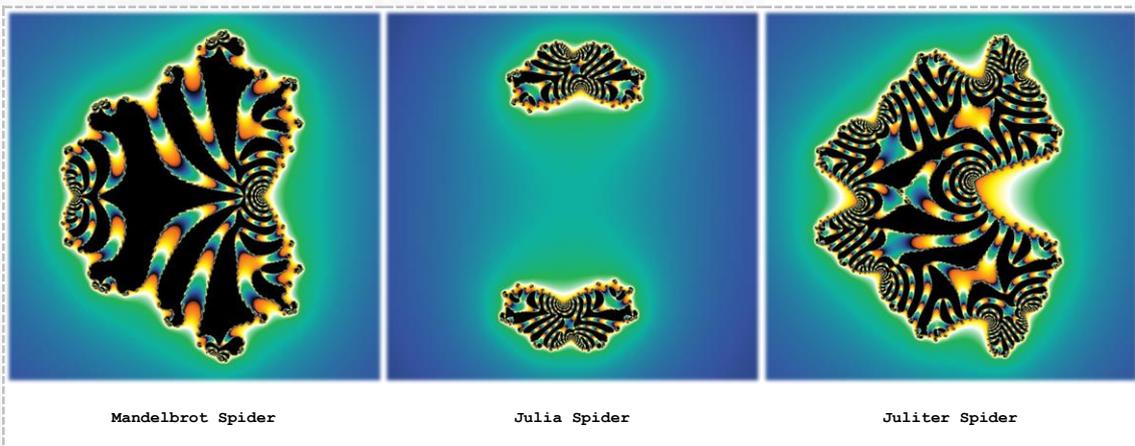
Burning Ship



Nova



Spider



Esta obra está autorizada bajo la Licencia Internacional Reconocimiento-NoComercial-CompartirIgual 4.0 de Creative Commons. Para ver una copia de esta licencia, visite <https://creativecommons.org/licenses/by-nc-sa/4.0/>