

Mirror Plane Transformation

Sergio CT



« I proceeded in a way that theorists abhor, I contemplated and dissected images impossible to forget, using the computer as a microscope ... »

Benoît Mandelbrot, creator of the Theory of Fractals

Content

Introduction	3
Algorithm	3
Reflection	4

The number of fractal, colouring, and orbit trap algorithms included in FFEplorer has been increasing over time; but in the field of complex plane transformations, it only has the basic transformation of "Inverse Plane".

With the new user's formulas editor, FFEplorer has gone from being a software for fractal exploration to a software for experimentation with fractals, facilitating greatly the implementation of new ideas that take advantage of all the application features, without the need for compiling new versions of this. An example of this is the new Mirror Plane Transformation, which owes its name to the fact that, when it is applied to any complex plane fractal, the fractal unfolds into two deformed versions of itself, as if it were being reflected between two slightly curved mirrors.

Introduction

This document describes how to apply the “Mirror Plane” Transformation to fractals represented by the Mandelbrot Method. This method consists of analysing the behaviour of the iterative formula $Z_{n+1} = f(z_n) + C$ for each point C of the complex plane, where f is a previously chosen function and the initial value of Z is the complex number $(0, 0i)$.

When the sequence of values of Z is bounded, C is considered to belong to the Mandelbrot Set and, in general, it is assigned the colour black. On the contrary, if the sequence of values of Z exceeds a previously established escape radius, a colour is assigned to C that depends on the iteration number in which the orbit of Z exceeds said limit.

Also note that the Mandelbrot Set represents all related Julia Sets. Which are calculated with the same iterative formula, but in this case Z would be the variable that runs through the complex plane, and C would have a previously established constant value.

Algorithm

The “Mirror Plane” transformation arises during the tests of the new editor of user's formulas that is included in FFXplorer v12, belonging to the FractalFun Project.

Given the function $f(z) = z^2$, we have the Mandelbrot Set formula $Z_{n+1} = Z_n^2 + C$, for which we can implement the following algorithm that represents the fractal:

pseudo-code:

```

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)
}

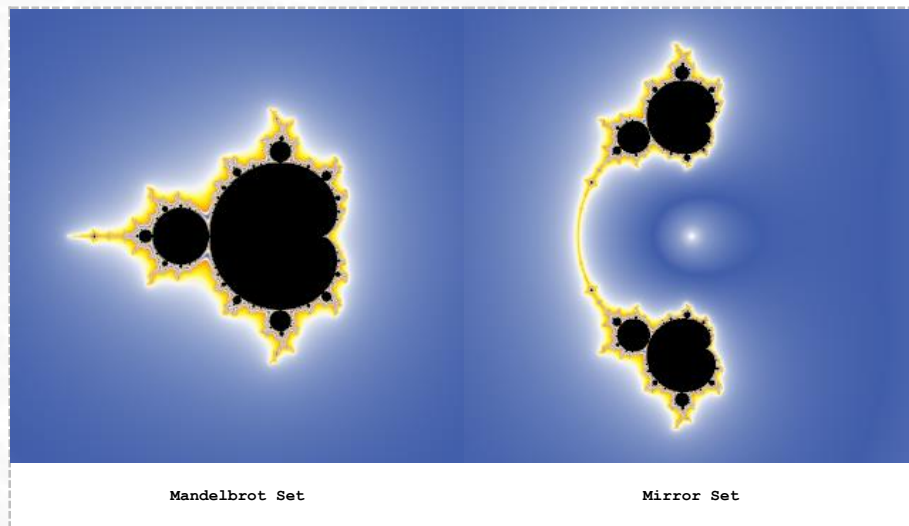
```

To add the “Mirror Plane” transformation to the algorithm described above, we only need to add the inverse of its value to the variable that runs through the plane **before entering the iteration loop**.

pseudo-code:

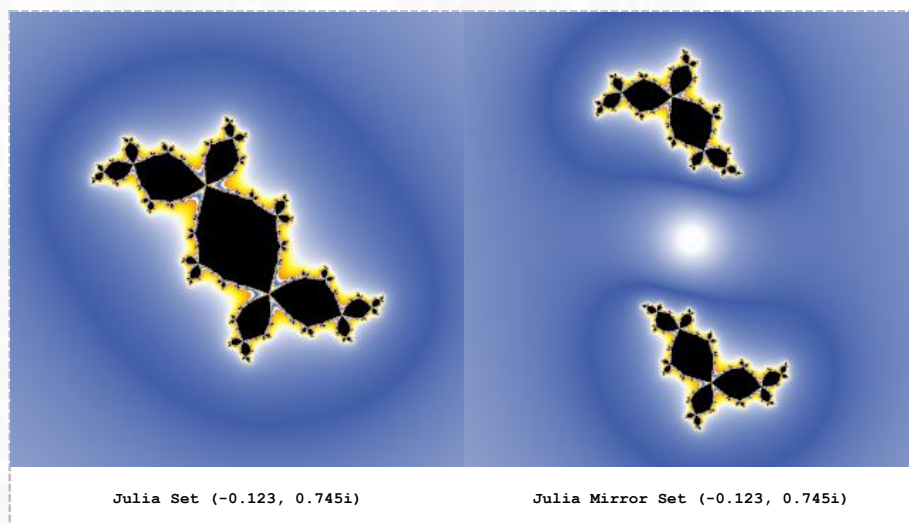
$$c = 1/c + c$$

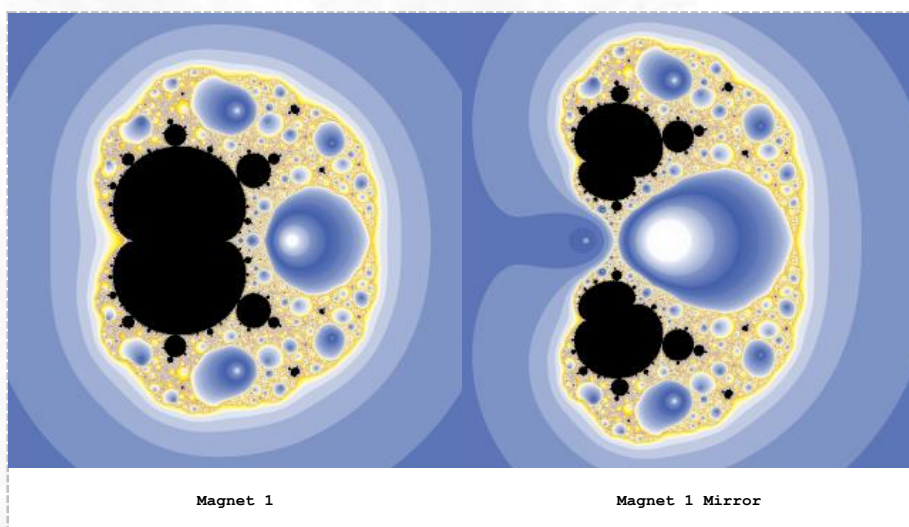
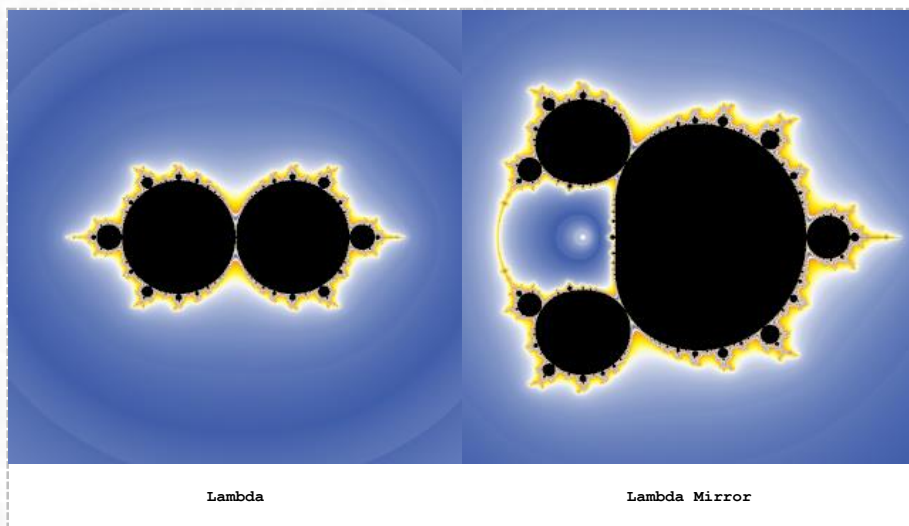
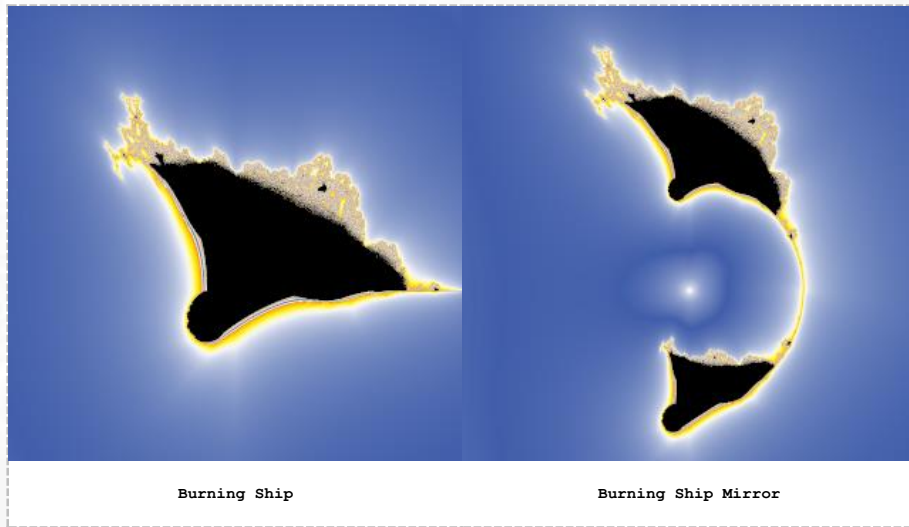
The resulting set of applying the transformation of “Plane Mirror” on the Mandelbrot Set has been given the name of “Mirror Set”:

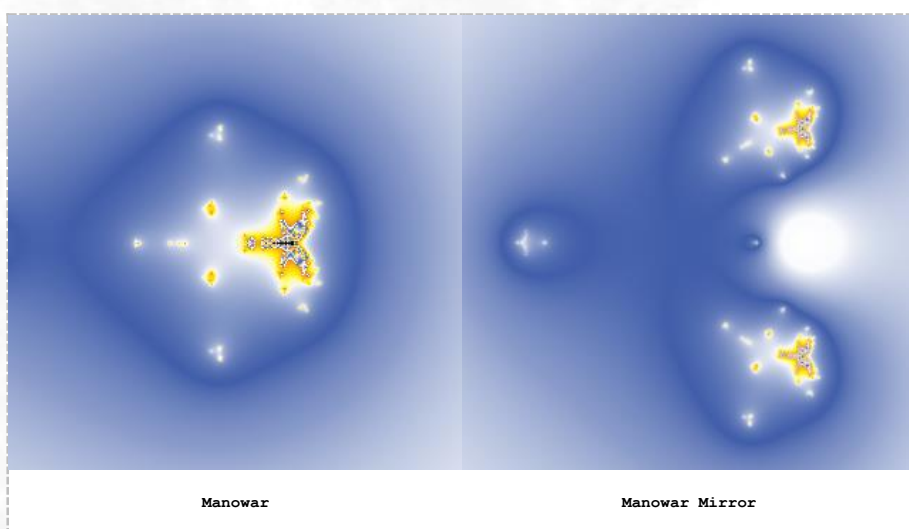
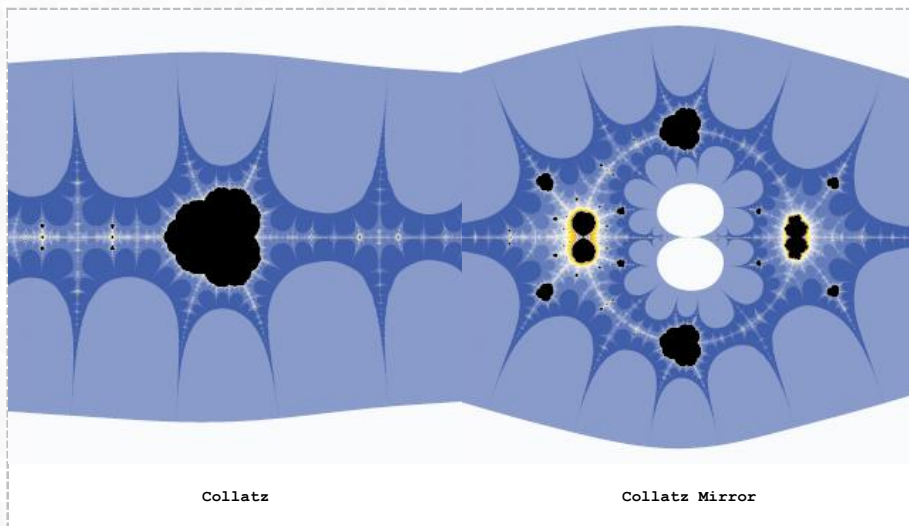
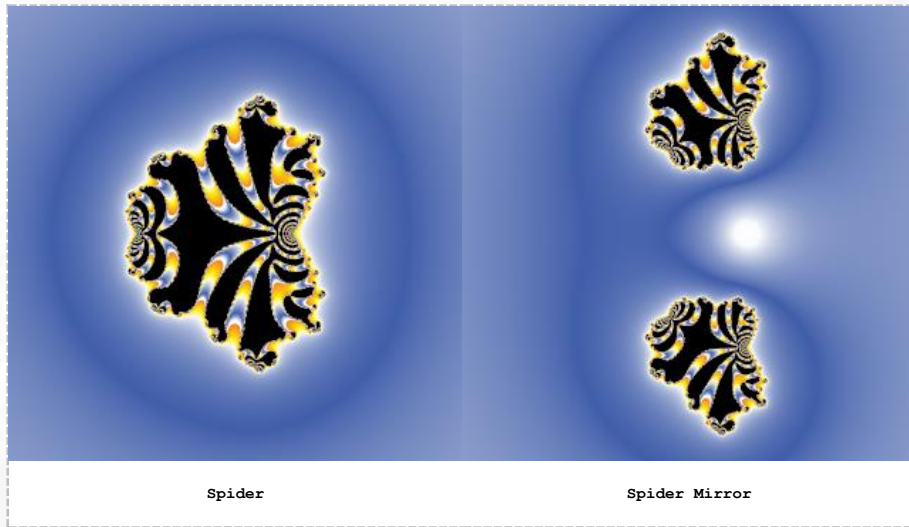


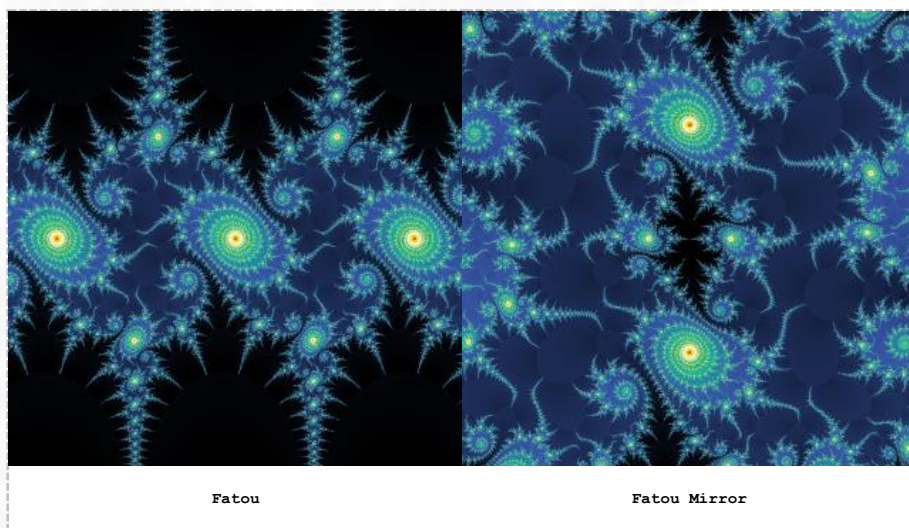
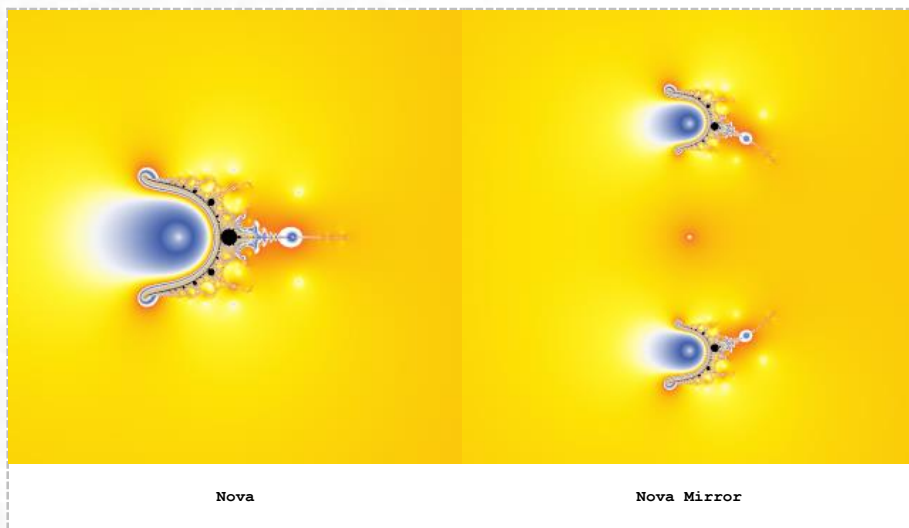
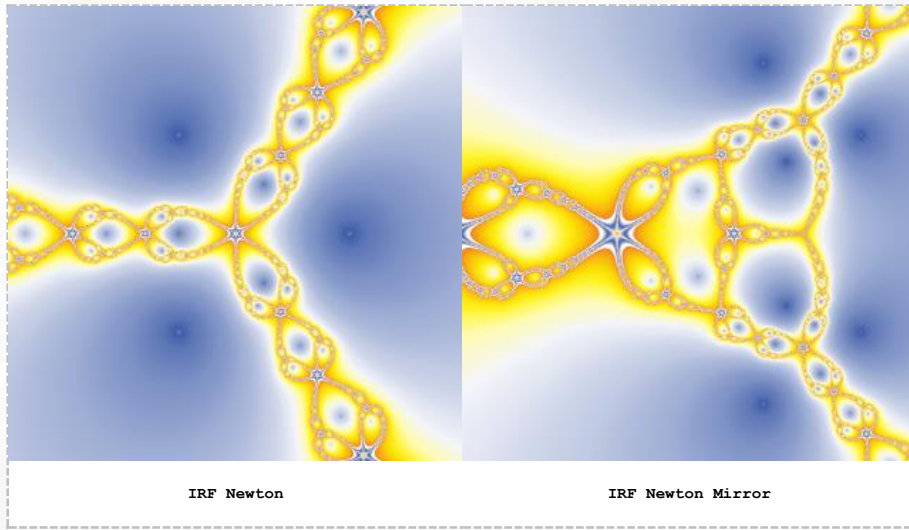
Reflection

The “Mirror Plane” transformation can also be applied to fractals calculated using the Julia Method and provides especially interesting results when the two deformed copies of the set have their ends connected. Here are several examples:











This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>