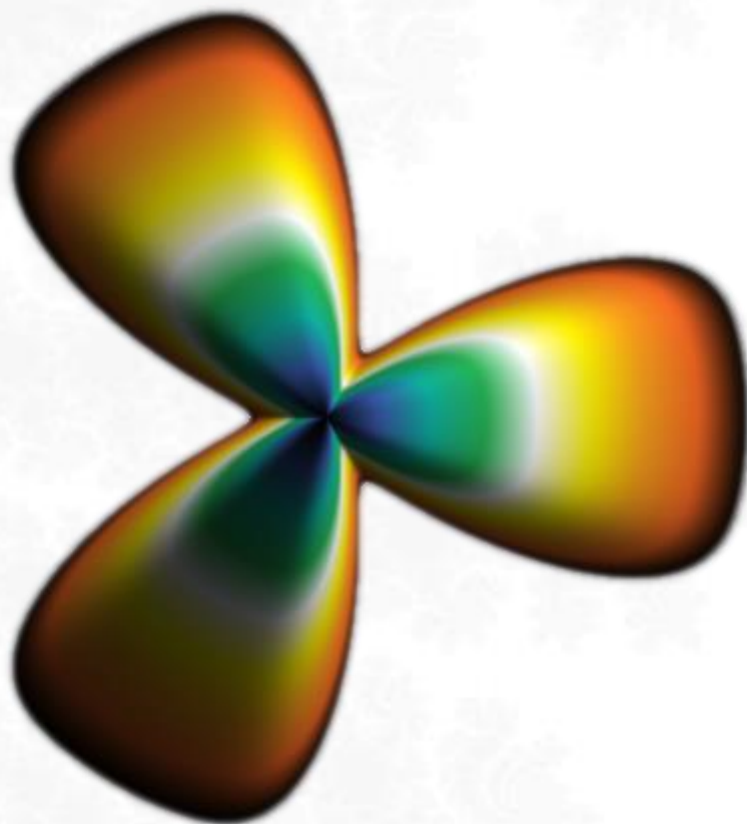


Orbit trap with Superformula

Sergio CT



« Nobody knows the potential of this powerful system; one day he will be able to perform music, compose symphonies and complex graphic designs ... »

Ada Lovelace

<https://fractalfun.es/en>

Content

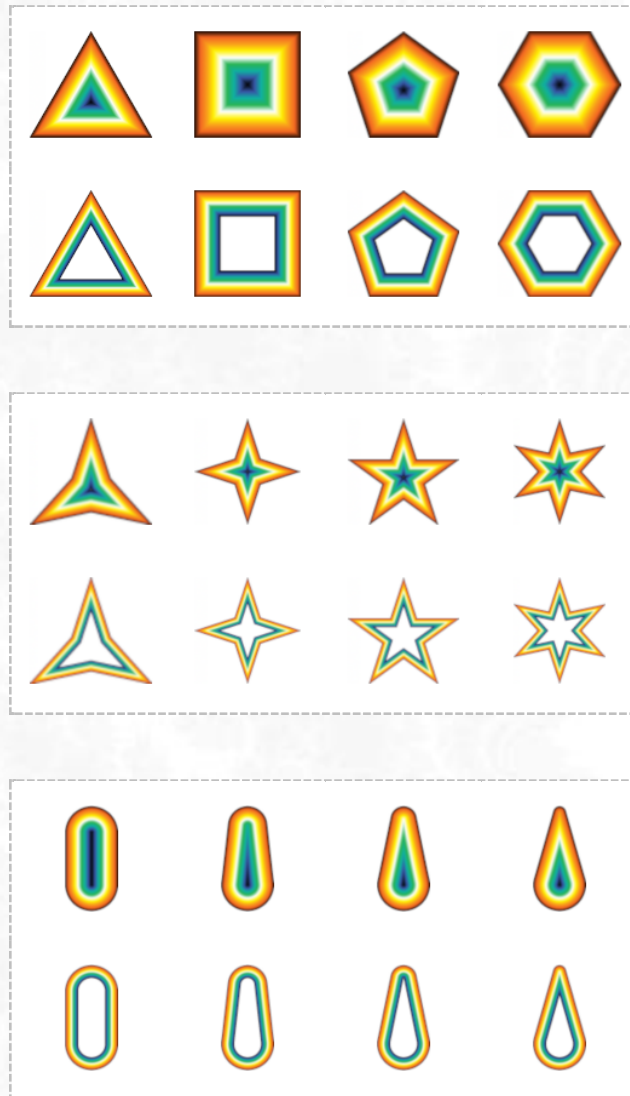
Motivation	3
Equation	4
Algorithm	4
Reflection	5

The “Superformula” equation appeared in April 2003, in number 90 of the *American Journal of Botany*, in an article by biologist *Johan Gielis*. It was obtained as a generalization of the equation of the superellipse, created by the Danish mathematician *Piet Hein*, which, in turn, is a generalization of the equation of the ellipse. At first, the use of “Superformula” was proposed to describe different complex shapes and curves present in nature, but in this document I propose its use in colouring algorithms for fractals, in order to generate complex orbit traps.

Motivation

The FFEplorer application has several algorithms for the generation of orbit traps with complex shapes, which, by means of specific parameters, can be asked to draw shapes with the desired number of tips, sides, etc.

As the flexibility of these algorithms is somewhat limited, for usability reasons I decided to generate several closed sets of well-defined, solid and annular shapes, leaving to the user the possibility of choosing the one they want, to adjust only its properties at the orbit trap level (that is, the properties that have nothing to do with the shape itself: position, size, rotation, opacity, and colouration).



Some of the 78 predefined shapes of FFEplorer v11.1

However, when I became aware of the existence of the "Superformula" equation, it seemed to me that it was versatile enough to adapt it to an orbit trap generation algorithm that would offer the user all control over the parameters involved in the definition of the shape itself.

Equation

The "Superformula" equation in polar form, with r for the radius and φ for the angle, is represented as follows:

$$r(\varphi) = \left(\left| \frac{\cos\left(\frac{m\varphi}{4}\right)}{a} \right|^{n_2} + \left| \frac{\sin\left(\frac{m\varphi}{4}\right)}{b} \right|^{n_3} \right)^{\frac{1}{n_1}}$$

Where:

- m , n_1 , n_2 and n_3 are positive real numbers.
- a and b are positive real numbers, excluding zero.

As "Superformula" (a , b , m , n_1 , n_2 y n_3) is a generalization of "Superellipse" (a , b y n) it can be made to work like the latter using the values a , b , $m = 4$ y $n_1 = n_2 = n_3$.

There is a variant of "Super Formula" in which m has independent values, x and y , for each term:

$$r(\varphi) = \left(\left| \frac{\cos\left(\frac{x\varphi}{4}\right)}{a} \right|^{n_2} + \left| \frac{\sin\left(\frac{y\varphi}{4}\right)}{b} \right|^{n_3} \right)^{\frac{1}{n_1}}$$

Algorithm

The algorithm developed for FFExplorer performs a complex plane analysis using the equation described above, as follows:

1. Initializes the values of m_1 , m_2 , n_1 , n_2 , n_3 , a y b with the values entered by the user.
2. It traverses the plane and, for each point to be analysed p , gives φ the value of its phase, with which it solves the equation and obtains a radius r .
3. Find the modulus of the complex number represented by the values of r and φ in its polar form.

4. Check if the modulus of p is less than or equal to the modulus calculated in the previous step, in that case the algorithm considers that the analysed point is inside the shape and proceeds to colour it, otherwise it is not coloured.

Note: The implementation is so simple and intuitive that it is not considered necessary to accompany these instructions with an explanatory pseudo-code.

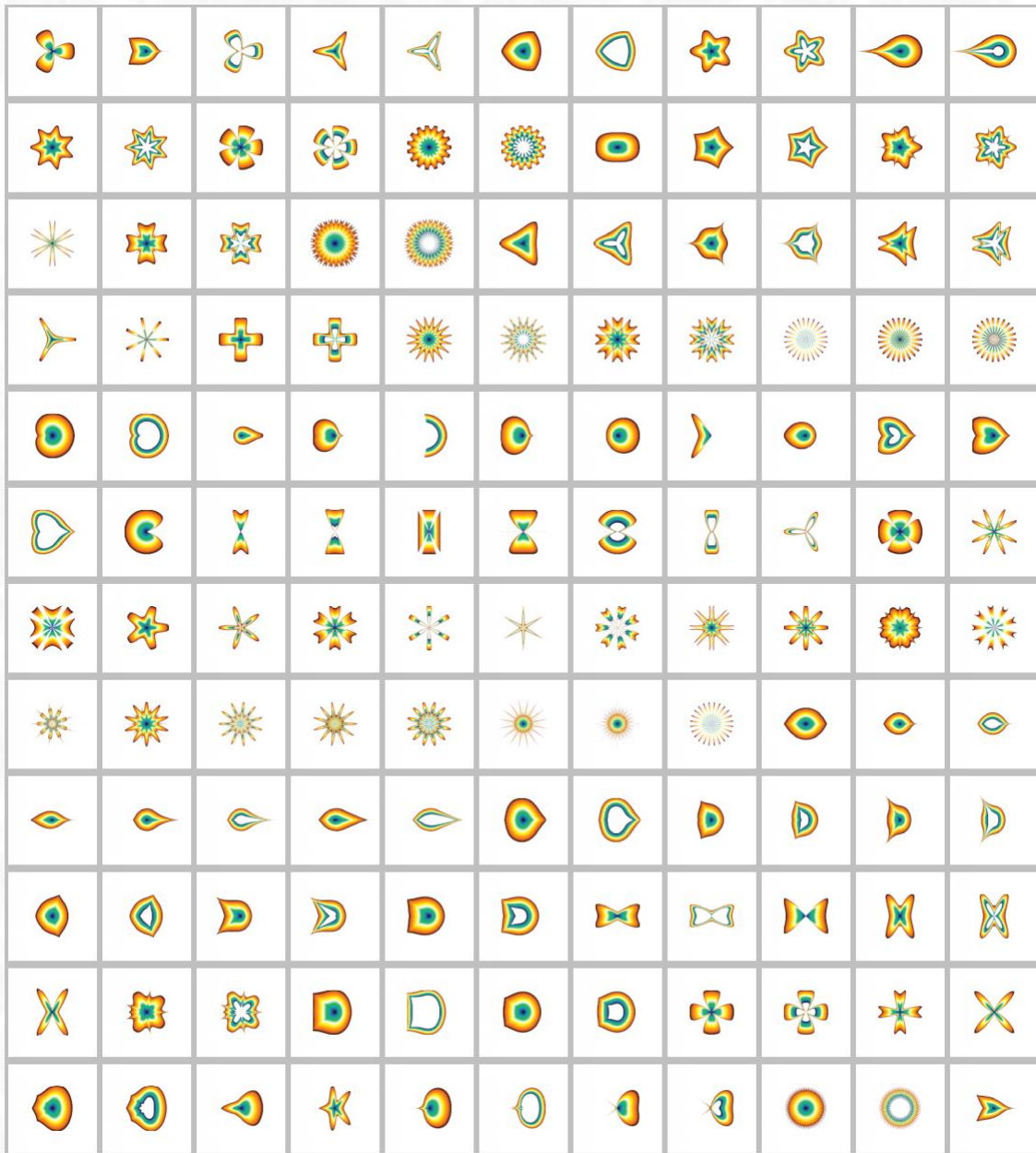
Reflection

The "Superformula" orbit trap is a very good option to allow users to design, use and share their own custom Orbit traps. However, FFE Explorer already had similar options available by using custom orbit traps in PNG image files, with the only limitation being the resolution of the image file itself:



Example of a fractal made with FFE Explorer using a PNG-type orbit trap

Testing with FExplorer has produced an enormous number of shapes, which have provided excellent results, especially using them in the distance-based colouring algorithm with lighting:

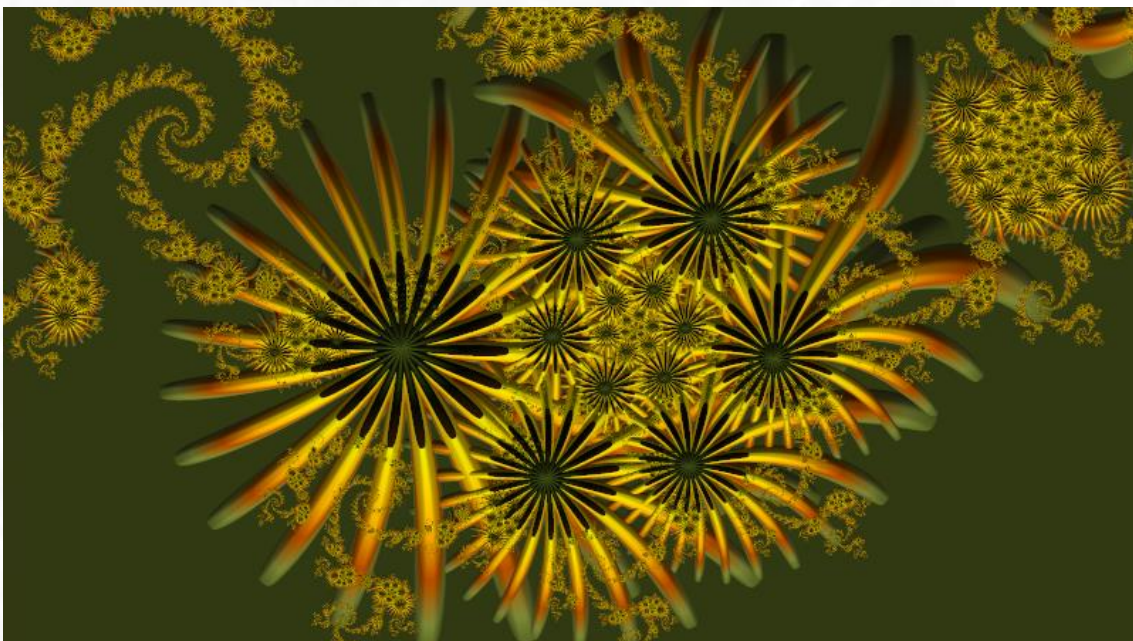


Some of the "Supershapes" obtained with the FExplorer "Superformula" algorithm

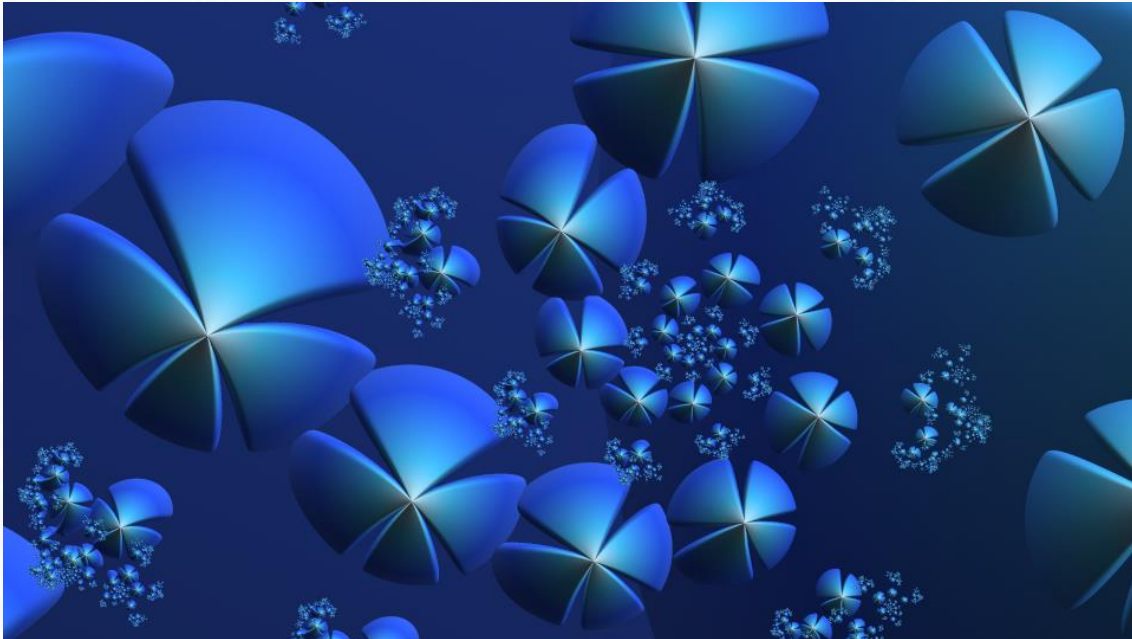
The following examples show the results of using "Supershape" type orbit traps for colouring fractals:



Fractal made by Elensegu with FFXplorer, using a "Supershape" type orbit trap



Fractal made by Elensegu with FFXplorer, using a "Supershape" type orbit trap



Fractal made by JLFractals with FFEExplorer, using a "Supershape" type orbit trap



Fractal made by Sergio CT with FFEExplorer, using a "Supershape" type orbit trap



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/>