The Sierpinski Tetrahedron Final Documentation

Wyatt Newberry

December 2015

1 Introduction

The definition of a fractal is often the topic of debate among mathematicians. For the sake of this project, I will be considering fractals as a set with a dimension that is a fraction. The mathematical concept of a fractal was discovered by French mathematician Gaston Julia. Julia was interested in the set of points defined by iteration of functions. In 1918, Julia published a mathematical paper, *Mémoire sur l'itération des fonctions rationnelles*[1] describing iterations of functions, leading to a very important discovery.

2 The Julia Set

Some of Julia's work focused specifically on iterations of a rational function. A rational function is function that is defined by a polynomial fraction, such that the denominator and the numerator are both polynomials. Julia chose some rational function, and generated an image by iterating through it and plotting the points to depict his Julia sets. Mathematically, the rational function is represented as: Let:

$$R(z) = \frac{P(z)}{Q(z)} \text{ where } z \in C$$

and P and Q are polynomials without common divisors. The formula that generates a familiar Julia set is:

$$z_{n+1} = z_n^2 + c$$

for a fixed value c. The Julia set of a fractal serves as the boundary of what points remain in the set in our view verses what point escapes to infinity (which are colored based on how quickly these points escape to infinity) For example:
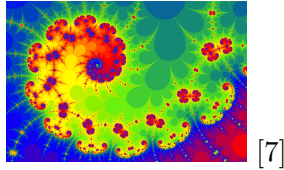
[7]

Taking a closer look at the image, one can soon realize that this image a lot more than a pretty picture. This image has repeating patterns at relatively every scale. This property is known as self-similarity. This image also depicts another quality that I have not mentioned yet. Notice how all of points in the system tend to follow the same pattern, and tend to be in similar spots relatively throughout the image. This is no mistake. This depicts another mathematical term related to fractals, an attractor. An attractor is a set of numerical values that a system tends to take the shape of[2]. This further proves that the images produced by the iteration of the rational function that converges to a Julia set are fractals.

3   The Fractal Dimension

Another special property of fractals I mentioned earlier is their fractional dimension. Fractals do not follow the the traditional pattern of 1, 2, 3, 4, etc. dimensions. Fractals lie between these standard dimensions. How is this possible? Well, lets start with standard Euclidean dimensions. If the linear side of an object residing in Euclidean Dimension D is reduced by $\frac{1}{r}$ in each spatial direction, its unit of measurement (e.g. length, area, or volume) would increase to $N = r^D$ times the original[3]. This is easily represented in this picture:

[8]

2
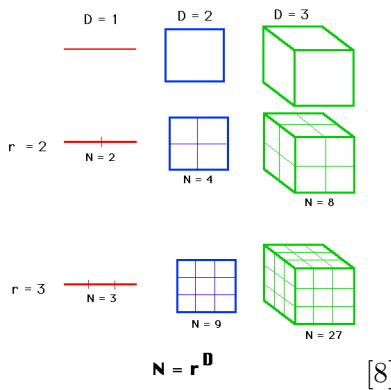
In the case of fractals, their dimensions are denoted by the term 'Hausdorff Dimension'. To calculate a specific fractal's Hausdorff dimension, a simple formula can be followed:

$$N = s^d \text{ and } d = \frac{ln(N)}{ln(s)} \text{ where:}$$

$$N = \text{(number of self-similar pieces) and}$$

$$s = \text{(magnification factor) of each division of the fractal[4].}$$

4   Cantor Fractals

The *Sierpinski Triangle* is a fractal common fractal example that involves taking a triangle and splitting it into 4 equal parts, and removing the center triangle. To understand the triangle, one must first understand its origin. This fractal is in the family of Cantor fractals. These fractals originate from the man they are named after, German mathematician Georg Cantor. It all began in 1883 when he had an idea. Cantor started with a line, and then divided the line into thirds, and removed the middle segment of the line, like so:

[9]

What is fascinating about this set is that this process could be done infinitely. The *Cantor Set* is actually an example of a fractal with a dimension of $\frac{ln(2)}{ln(3)} \approx 0.631$. Another important consideration in this whole process is the endpoints of line segments being removed from the set. If the endpoints were kept, eventually the process would cause original set would be no more. So, it is given in the definition of the *Cantor Set* that the line segments are open sets, which are defined as sets that do not include their endpoints. [5]

 [10]
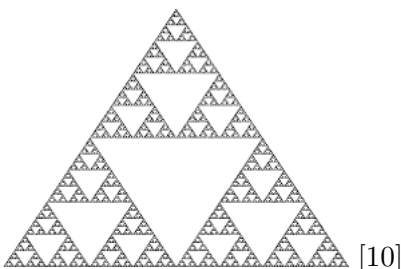
The *Sierpinski Triangle* divides into 3 self-similar pieces, and has a magnification factor of 2, therefore its Hausdorff dimension is $\frac{ln(3)}{ln(2)} \approx 1.58$. This triangle can be generated easily in two different ways. The first way is the Chaos Game[11] (more properly known as an Iterative Function System) method. The corners of the triangles could be considered as p_1, p_2, p_3 and a random point could be named v_1. Then, set $v_{n+1} = \frac{1}{2}(v_n + p_{r_n})$ where r_n is a random number 1, 2, or 3. Then this is iterates for values v_1 to v_∞. If the original point v_1 in on the triangle, then all points v_n will lie on the triangle. If v_1 that is within the perimeter of the triangle is not on the Sierpinski triangle, none of the points will be on the triangle. But, these points will converge on the attractor, which is the Sierpinski Triangle[6].

Another method of generating this triangle is via recursion. Recursion is at the most basic definition a repeated application of a procedure. This method is used in computer programming to generated this type of fractal. A set of instructions can be provided as code that divides the triangle, determines the new vertices of the interior triangles, and recursively call these instructions to implement the fractal.

5   The Complete Project

I started the project with 2 Dimensional representations of the Sierpinski Triangle. I generated the triangle via The Chaos Game and Recursion to compare and contrast the difference of generating the fractal. The Chaos Game in 3 Dimensions is not an easy feat (and coincidentally the 3 Dimensional Chaos Game was another student's project). So for my final version of my RTICA, I used recursion to depict the Sierpinski Tetrahedron. But, I also had to do the recursion in a "reverse" manner. Basically, the tetrahedrons had to be drawn in the proper size and arrangement, so when the base case was reached, then the program actually begins to generate the geometry. For this project I used JavaScript/HTML and also a library for JavaScript called THREE.JS. This library is a very powerful tool for 3D graphics in JavaScript. Without this library, my project would have been

significantly more difficult and would have required much more time. I have also included a screenshot of the project below.
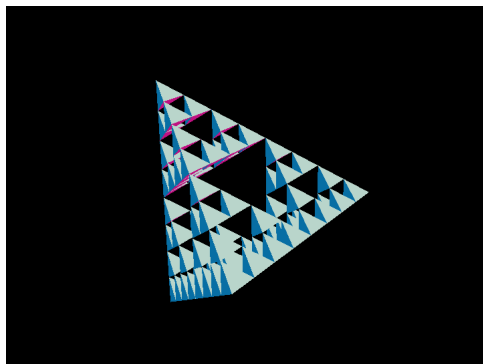
References

[1] Wikipedia, *Gaston Julia* (`http://en.wikipedia.org/wiki/Gaston_Julia`)

[2] Wolfram MathWorld, *Attractors* (`mathworld.wolfram.com/Attractor.html`)

[3] Vanderbilt University, *Fractals and the Fractal Dimension* (`vanderbilt.edu/AnS/psychology/cogsci/chaos/workshop/Fractals.html`)

[4] Boston University Mathematics, *Chaos Game* (`math.bu.edu/DYSYS/chaos-game/node6.html`)

[5] Wikipedia, *Georg Cantor* (`en.wikipedia.org/wiki/Cantor_set`)

[6] Wikipedia, *Sierpinski Triangle* (`en.wikipedia.org/wiki/Sierpinski_triangle#Chaos_game`)

[7] Relativity Book, *Julia Set* (`http://www.relativitybook.com/CoolStuff/julia_set_alt/alt_julia_zoom_03.png`)

[8] Vanderbilt, *Hausdorff Dimension* (`http://www.vanderbilt.edu/AnS/psychology/cogsci/chaos/workshop/Hausdorff.GIF`)

[9] Wikiepdia, *Cantor Set* (`https://upload.wikimedia.org/wikipedia/`
`commons/thumb/5/56/Cantor_set_in_seven_iterations.svg/`
`729px-Cantor_set_in_seven_iterations.svg.png`)

[10] Wolfram MathWorld, *Sierpinski Sieve* (`http://mathworld.wolfram.`
`com/images/gifs/sier90.jpg`)

[11] Wikipedia, *The Chaos Game* (`https://en.wikipedia.org/wiki/`
`Chaos_game`)