Construction and Properties of the Sierpinski Triangle and other fractals

Piotr Skowroński, Bartosz Smolarz

January 28, 2024

1 Introduction

The Sierpinski triangle is a fascinating fractal that exhibits self-similarity and is generated through a simple recursive process. Understanding its construction and properties sheds light on the beauty and complexity of fractal geometry.

2 Construction of the Sierpinski Triangle

The Sierpinski triangle can be constructed using a recursive algorithm. Let T_0 be the initial equilateral triangle. Then, for each subsequent iteration n, three copies of T_n are created, each scaled down by a factor of $\frac{1}{2}$, and the middle triangle is removed. Mathematically, this process can be represented as:

$$T_{n+1} = \frac{1}{2}\left(T_n - \frac{1}{2}T_n\right)$$

This recursive process continues indefinitely, resulting in the formation of the Sierpinski triangle fractal.

3 Properties of the Sierpinski Triangle

The Sierpinski triangle exhibits several interesting properties, including:

- Self-similarity: At any scale, the Sierpinski triangle resembles itself.
- Infinite perimeter: Despite having finite area, the Sierpinski triangle has an infinite perimeter due to its infinitely repeating pattern.
- Fractal dimension: The Sierpinski triangle has a fractal dimension of approximately 1.585, which lies between the dimensions of a one-dimensional line and a two-dimensional plane.

These properties contribute to the uniqueness and complexity of the Sierpinski triangle.

4 Applications

The Sierpinski triangle has applications in various fields, including:

- Fractal geometry, pioneered by mathematicians such as Benoit Mandelbrot, has revolutionized our understanding of complex and irregular shapes in nature, mathematics, and various scientific fields. It serves as a fundamental example of a self-similar fractal, where intricate patterns repeat at different scales. Fractals are not only fascinating mathematical objects but also powerful tools for modeling and understanding natural phenomena, from coastlines and mountains to biological structures like trees and ferns.

  Fractal geometry has applications in diverse fields such as:

  - Physics: Fractals are used to model phenomena like turbulence, diffusion-limited aggregation, and percolation.
  - Biology: Fractal geometry helps in analyzing complex biological structures such as blood vessels, lungs, and neural networks.
  - Finance: Fractals are applied in modeling stock market fluctuations and analyzing financial data.
  - Art: Fractals inspire artists and designers to create visually stunning and conceptually rich artworks.

- In computer graphics, the recursive structure of fractals is leveraged to create visually appealing patterns, textures, and complex images. Fractals offer a unique way to generate detailed and intricate graphics that exhibit self-similarity at various levels of magnification. Techniques such as fractal landscape generation, flame fractals, and iterated function systems (IFS) are commonly used to produce captivating visual effects in computer-generated imagery (CGI), animation, and digital art.

  Applications of fractals in computer graphics include:

  - Terrain Generation: Fractals are used to simulate natural landscapes, terrain, and geological formations in video games, simulations, and virtual environments.
  - Texture Synthesis: Fractals enable the creation of realistic and visually appealing textures for surfaces, materials, and patterns in computer graphics.
  - Procedural Generation: Fractals are employed to procedurally generate complex and diverse content, including trees, foliage, clouds, and architectural structures.

- Fractals provide a captivating introduction to mathematical concepts such as self-similarity, recursion, and iteration, making them invaluable educational tools. They offer an engaging way to explore mathematical ideas and principles, allowing students to visually observe and interact with abstract concepts. Introducing fractals in education can stimulate curiosity, creativity, and critical thinking skills, fostering a deeper appreciation for mathematics and its applications.

  In educational settings, fractals are used to:

  - Introduce Mathematical Concepts: Fractals serve as concrete examples to illustrate abstract mathematical concepts such as iteration, self-similarity, and infinity.
  - Promote Visual Learning: Visual representations of fractals help students grasp complex mathematical ideas intuitively through observation and exploration.
  - Encourage Interdisciplinary Learning: Fractals bridge the gap between mathematics and other disciplines such as art, science, and technology, fostering interdisciplinary learning and creativity.

- The Sierpinski triangle, due to its unique properties, finds applications in various fields. Here are some specific examples:

  - Cryptography and Computer Security: The Sierpinski triangle can be used to generate random bit sequences that are difficult to predict, which is important in cryptographic algorithms and encryption key generation.
  - Electronics: In the field of electronics, the Sierpinski triangle can be used as an antenna with an unconventional shape, allowing for unique antenna properties such as wide bandwidth, low weight, and ease of production.
  - Materials Engineering: In materials engineering, the Sierpinski triangle can be applied in the design of materials with advanced composite structures, leading to improved material strength, reduced weight, and increased resistance to damage.
  - Biology and Medicine: In the biological sciences and medicine, the Sierpinski triangle can be used to analyze biological structures such as vascular networks, nervous systems, or DNA structures, enabling a better understanding of biological organization and function.
  - Radio Communication: In the field of radio communication, the Sierpinski triangle can be used as a wideband antenna, enabling the transmission of signals at different frequencies with high efficiency.
  - Artificial Intelligence: The Sierpinski triangle can also be applied in machine learning algorithms and artificial intelligence to generate synthetic data, test algorithms, and as a tool for visualization and interpretation of results.

5 Sierpinski Triangle Visualization

The Sierpinski triangle is a fractal that can be generated recursively. The following Mathematica code generates and visualizes the Sierpinski triangle using recursion:

```
sierpinskiTriangle[points_, depth_] :=
  Module[{midpoints},
   If[depth == 0, {Polygon[points]},
    midpoints = (#[[1]] + #[[2]])/2 & /@ Subsets[points, {2}];
```

```
   Flatten[
    sierpinskiTriangle[#, depth - 1] & /@ {Join[{points[[1]]},
       midpoints[[1 ;; 2]]],
      Join[{points[[2]]}, midpoints[[{1, 3}]]],
      Join[{points[[3]]}, midpoints[[2 ;; 3]]]}]]];

Manipulate[
 Graphics[
  sierpinskiTriangle[{{0, 0}, {1, 0}, {0.5, Sqrt[3]/2}}, depth],
  Frame -> True, PlotRange -> {{0, 1}, {0, Sqrt[3]/2}}], {depth, 0, 8,
   1}]
```

- Function Definition (sierpinskiTriangle):
  - This part defines a recursive function named sierpinskiTriangle.
  - It takes two arguments: points, which represents the vertices of a triangle, and depth, which determines the level of recursion.
  - The function generates the Sierpinski triangle fractal by recursively dividing the input triangle into smaller triangles.
- Module and If Statement:
  - Inside the function, there is a Module construct that encapsulates the local variables and expressions.
  - The If statement checks whether the recursion depth (depth) has reached zero. If so, it returns a single triangle (represented as a Polygon).
- Calculating Midpoints:
  - If the depth is greater than zero, the code calculates the midpoints of the input triangle's edges.
  - It does so by computing the midpoint of each pair of vertices using the formula midpoints = ([[1]] + [[2]])/2 /@ Subsets[points, 2].
- Recursive Calls:
  - After calculating the midpoints, the function recursively calls itself on three smaller triangles formed by joining the original vertices with the midpoints.
  - This recursion continues until the specified depth is reached.
- Manipulate Function:
  - The Manipulate function creates an interactive interface to explore the Sierpinski triangle fractal.
  - It dynamically changes the depth of recursion using a slider (depth, 0, 8, 1) to control the level of detail in the visualization.
- Graphics Output:
  - Finally, the Graphics function displays the generated Sierpinski triangle fractal.
  - It sets options such as Frame -¿ True to add a frame around the plot and PlotRange -¿ 0, 1, 0, Sqrt[3]/2 to specify the plot range.

6 Formation of Fractals, Other Fractals

Fractals can be generated in many different ways, each with its unique characteristics. One popular method is iteratively applying certain transformation functions to an initial set of points. Examples of such fractals include the Mandelbrot set and the Julia set, which are generated by iterating complex functions.

Another method involves the use of Lindenmayer systems (L-systems), which generate fractal structures by iteratively applying simple rules to strings of symbols.

Figure 1: First depth

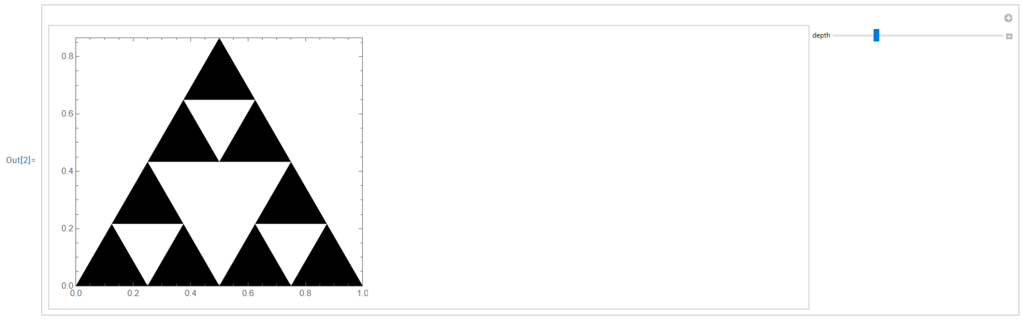

Figure 2: Second depth

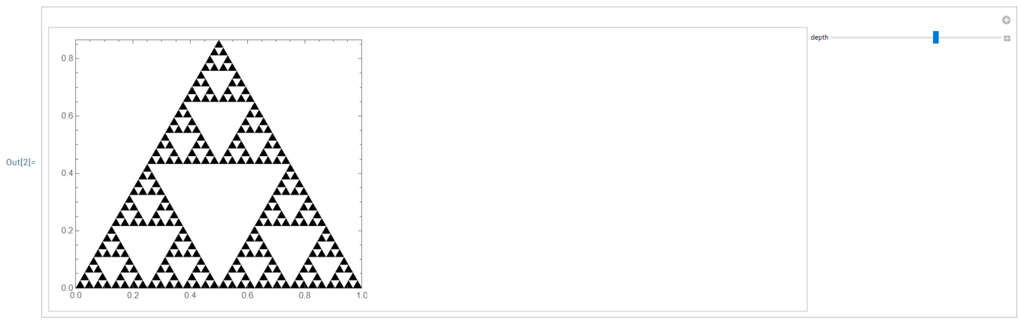

Figure 3: Fifth depth

6.1 Mandelbrot Set

The Mandelbrot set is one of the most well-known fractals, discovered in 1980 by Benoit Mandelbrot. It is generated by iteratively applying the complex function $f(z) = z^2 + c$, where z is a complex number and c is a constant. Points in the complex plane for which the sequence of values generated by this function remains bounded are part of the Mandelbrot set.

6.2 Julia Set

The Julia set is closely related to the Mandelbrot set and is generated by iteratively applying the function $f(z) = z^2 + c$, where z is a complex number and c is a fixed constant, while the values of z vary. The Julia set exhibits a vast variety of shapes and patterns depending on the value of the constant c.

7 Conclusion

The Sierpinski triangle is a captivating fractal with intriguing properties and diverse applications. Its construction using simple recursive rules illustrates the elegance and complexity of fractal geometry, making it a fascinating subject of study and exploration.