Sierpiński Gaskets and Carpets: Constructions, Properties, and Visualization

Max Statz

Spring 2023

This is a research project for Math 490: An Invitation to Fractal Geometry. In this paper, two fractal objects will be discussed: Sierpiński gaskets and Sierpiński carpets. The first portion of this paper is dedicated to rigorously defining these objects and discussing their properties. The remaining portion is dedicated to explaining some fractal-generating software that allows the user to visualize Sierpiński carpets and gaskets as well as better understand their construction.

Note that while Sierpiński gaskets and carpets are similar in terms of their construction and even share a number of properties relating to their areas and interiors, these objects are inherently different. For the sake of brevity and reducing the amount of redundant information, this paper at times selectively chooses one of the two fractal objects to demonstrate such properties as their proofs are similar. Significant differences between such proofs will be highlighted when relevant.

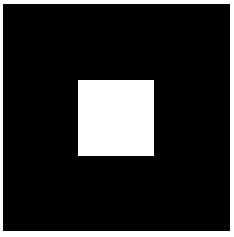
---

1 Carpet and Gasket Construction

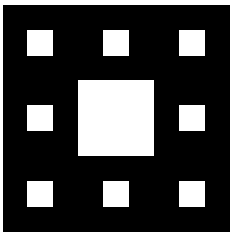
1.1 The Trema Method

We will first construct the Sierpiński carpet using the trema method. Consider a square of side length 1. Call this K_0.

Subdivide K_0 into a 3×3 grid and remove the middle square of side length $\frac{1}{3}$. Call this K_1.



Subdivide each of the remaining 8 squares in K_1 into 3×3 grids and remove their middle squares of side length $\frac{1}{9}$. Call this K_2.



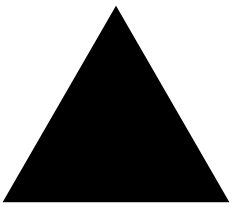
We can define $K_3, K_4, K_5...$ similarly by subdividing the remaining filled-in squares from the previous iteration into 3×3 grids and removing their middle squares. We then define the Sierpiński carpet (SC) to be

$$SC = \bigcap_{n=1}^{\infty} K_n$$

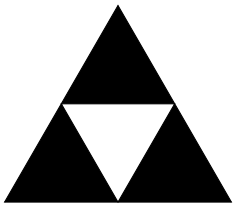
If we consider the sequence $(K_0, K_1, K_2, ...)$, we may also define the Sierpiński carpet (SC) to be

$$SC = \lim_{n \to \infty} K_n$$

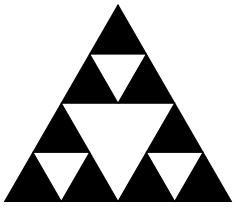
In a similar fashion, we will now construct the Sierpiński gasket using the trema method. Consider an equilateral triangle of side length 1. Call this K_0.



Subdivide K_0 into 4 triangles of equal size by joining the midpoints of the sides and remove the middle triangle of side length $\frac{1}{2}$. Call this K_1.



Subdivide each of the remaining 3 triangles in K_1 into 4 triangles of equal size by joining the midpoints of the sides and remove their middle triangle of side length $\frac{1}{4}$. Call this K_2.



We can define $K_3, K_4, K_5, ...$ similarly by subdividing the remaining filled-in triangles from the previous iteration into 4 triangles of equal size by joining the midpoints of the sides and removing their middle triangles. We then define the Sierpiński gasket (SG) to be

$$SG = \bigcap_{n=1}^{\infty} K_n$$

If we consider the sequence $(K_0, K_1, K_2, ...)$, we may also define the Sierpiński gasket (SG) to be

$$SG = \lim_{n \to \infty} K_n$$

1.2 Iterated Function System

We define the attractor of an iterated function system, call it K, to be

$$K = \bigcup_{i=1}^{n} f_i([0,1]^2)$$

where $f_1, ..., f_n$ is an iterated function system, that is, a family of functions of the form $f_i(x,y) = (r_i x + s_{1_i}, r_i y + s_{2_i})$ with some contraction ratio r_i and pair of translation factors s_{1_i} and s_{2_i}. We can define iterated function systems for which Sierpiński carpets and gaskets are the attractors. For the carpet, we define the following iterated function system:

$$\begin{aligned}
f_1(x,y) &= (\frac{1}{3}x, \frac{1}{3}y) \\
f_2(x,y) &= (\frac{1}{3}x, \frac{1}{3}y + \frac{1}{3}) \\
f_3(x,y) &= (\frac{1}{3}x, \frac{1}{3}y + \frac{2}{3}) \\
f_4(x,y) &= (\frac{1}{3}x + \frac{1}{3}, \frac{1}{3}y) \\
f_5(x,y) &= (\frac{1}{3}x + \frac{2}{3}, \frac{1}{3}y) \\
f_6(x,y) &= (\frac{1}{3}x + \frac{1}{3}, \frac{1}{3}y + \frac{2}{3}) \\
f_7(x,y) &= (\frac{1}{3}x + \frac{2}{3}, \frac{1}{3}y + \frac{1}{3}) \\
f_8(x,y) &= (\frac{1}{3}x + \frac{2}{3}, \frac{1}{3}y + \frac{2}{3})
\end{aligned}$$

where $f_i : [0,1]^2 \to [0,1]^2$ for $i = 1, ..., 8$. So, we have that $SC = \bigcup_{i=1}^{8} f_i([0,1]^2)$.
Now for the gasket, we define the following iterated function system:

$$\begin{aligned}
f_1(x,y) &= (\frac{1}{2}x, \frac{1}{2}y) \\
f_2(x,y) &= (\frac{1}{2}x + \frac{1}{2}, \frac{1}{2}y) \\
f_3(x,y) &= (\frac{1}{2}x + \frac{1}{4}, \frac{1}{2}y + \frac{\sqrt{3}}{4})
\end{aligned}$$

where $f_i : [0,1]^2 \to [0,1]^2$ for $i = 1, ..., 3$. So, we have that $SG = \bigcup_{i=1}^{3} f_i([0,1]^2)$.

2 Properties of Carpets and Gaskets

While their construction is intuitive, these fractal objects reveal a multitude of complexities upon further inspection. This section will discuss their areas, interior points, fractal dimensions, and connections to other fractal objects.

2.1 Area

We claim that both the Sierpiński carpet and gasket have no area. First, let us consider K_0 in the construction of the gasket. We can easily calculate the area of this triangle so we have $\text{Area}(K_0) = 1 \cdot \frac{\sqrt{3}}{2} \cdot \frac{1}{2} = \frac{\sqrt{3}}{4}$. To calculate the area of K_1, we must sum the area of the three remaining triangles after removing the middle triangle from K_0. Doing so, we have $\text{Area}(K_1) = 3 \cdot (\frac{1}{2} \cdot \frac{\sqrt{3}}{4} \cdot \frac{1}{2}) = \frac{3\sqrt{3}}{16}$. To calculate the area of K_2, we must sum the area of the nine remaining triangles after removing the middle triangles from K_1. Doing so, we have $\text{Area}(K_2) = 9 \cdot (\frac{1}{4} \cdot \frac{\sqrt{3}}{8} \cdot \frac{1}{2}) = \frac{9\sqrt{3}}{64}$. Then, we can define a general formula for the area of the n^{th} iteration of the gasket to be

$$\text{Area}(K_n) = \frac{3^n\sqrt{3}}{4^{n+1}}$$

Taking the limit as n tends to infinity yields the following:

$$\begin{aligned}
\lim_{n\to\infty} \text{Area}(K_n) &= \lim_{x\to\infty} \frac{3^n\sqrt{3}}{4^{n+1}} \\
&= \sqrt{3} \cdot \lim_{x\to\infty} \frac{3^n}{4^{n+1}} \\
&= \sqrt{3} \cdot \lim_{x\to\infty} \frac{3^n}{4 \cdot 4^n} \\
&= \frac{\sqrt{3}}{4} \cdot \lim_{x\to\infty} (\frac{3}{4})^n \\
&= \frac{\sqrt{3}}{4} \cdot 0 \\
&= 0
\end{aligned}$$

Therefore, we have that $\lim_{n\to\infty} K_n = SG$ has no area. The proof that the Sierpiński carpet has no area is similar.

2.2 Interior Points

Let E be some arbitrary set. We say that a point $x \in E$ is an interior point if there exists an open ball centered at x that lies entirely in E. We claim that both the Sierpiński carpet and gasket have no interior points. Let K be the Sierpiński carpet, and suppose by way of contradiction that and $z = (x, y)$ is an interior point of K. Then there exists some $r > 0$ such that the open ball $B(z, r)$ centered at z with radius r is contained entirely within K.

Let Q be a closed middle-third square of K that contains z. Since Q is closed and z is an interior point, there exists some $r' > 0$ such that $B(z, r') \subset Q$. Since Q is a closed middle-third square, its side length is $1/3$ times the side length of the parent square that was removed to form K. Thus, Q has area $(1/3)^2 = 1/9$ the area of the parent square. Since $B(z, r') \subset Q$, the area of $B(z, r')$ is at most $1/9$.

However, since z is an interior point of K, there exists some $r'' > 0$ such that $B(z, r'') \subset K$. Since K is the union of closed middle-third squares, there exists a closed middle-third square Q' such that $z \in Q'$ and $Q' \subset B(z, r'')$. But Q' has positive area, since it is a closed middle-third square. This contradicts the fact that $B(z, r'') \subset K$ and every closed middle-third square in K has been removed. Thus, our original assumption must be false z cannot be an interior point of K. Therefore, the Sierpiński carpet has no interior points.

The argument for the Sierpiński gasket is similar: every removed middle-third triangle has positive area, and so if z were an interior point of the gasket, there would exist an open ball centered at z with positive area, which would intersect one of the removed triangles, leading to a contradiction.

2.3 Dimension Calculation

Let $\{f_i\}_{i=1}^n$ be an iterated function system with an attractor K and contraction ratio r. We can calculate the similarity dimension of K, $\dim_{\text{sim}}(K)$, by the following formula:

$$\dim_{\text{sim}}(K) = \alpha \text{ where } \sum_{i=1}^{n} r_i^\alpha = 1$$

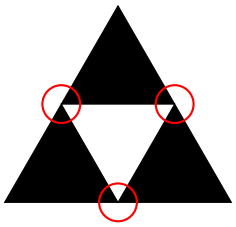
So, we can easily find $\dim_{\text{sim}}(SC)$ and $\dim_{\text{sim}}(SG)$ by solving for α. Using algebra, we obtain the following for the Sierpiński carpet:

$$\begin{aligned}
\sum_{i=1}^{8} (\frac{1}{3})^\alpha &= 1 \\
8(\frac{1}{3})^\alpha &= 1 \\
(\frac{1}{3})^\alpha &= \frac{1}{8} \\
\alpha \log(\frac{1}{3}) &= \log(\frac{1}{8}) \\
\alpha &= \frac{\log(\frac{1}{8})}{\log(\frac{1}{3})} \\
\alpha &= \frac{\log(8)}{\log(3)}
\end{aligned}$$

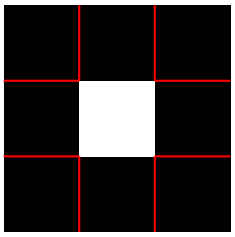
Thus, we have $\dim_{\text{sim}}(SC) = \frac{\log(8)}{\log(3)}$. We can perform the same calculation for the gasket which gives us $\dim_{\text{sim}}(SG) = \frac{\log(3)}{\log(2)}$. Note that for an iterated function system $f_1, ..., f_n$ with attractor K, if all $f_1, ..., f_n$ are pairwise disjoint, then we have

$$\dim_{\text{sim}}(K) = \dim_H(K) = \dim_M(K)$$

However, we can easily see that this is not the case for the Sierpinski carpet and gasket by their constructions. Consider K_1 of the gasket. We can see that there exists non-empty intersections between the functions at the midpoints of the sides of the larger triangle, that is, the corners of the triangles of side length $\frac{1}{2}$ (see figure below).



We can see the same in the case of the carpet where the non-empty intersections between the functions are due to shared points at the inner edges of the 8 squares in K_1 (see figure below).



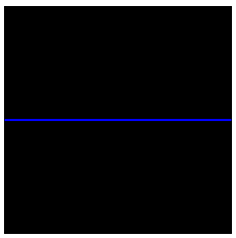
Thus, in both cases we must resort to other ways of calculating the Hausdorff and Minkowski dimensions. Despite this failure to exhibit pairwise disjointness, it can still be shown that these three measures of dimension are equal in both of these cases and we have

$$\dim_{\text{sim}}(SC) = \dim_M(SC) = \dim_H(SC) = \frac{\log(8)}{\log(3)} \quad \text{and} \quad \dim_{\text{sim}}(SG) = \dim_M(SG) = \dim_H(SG) = \frac{\log(3)}{\log(2)}$$

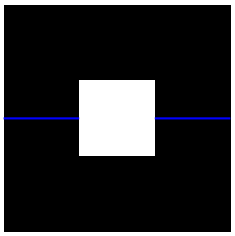
The details of these calculations are beyond the scope of this paper and will not be presented here.

2.4 Hidden Cantor Sets

While these two-dimensional fractal objects seem quite different from the one-dimensional middle thirds Cantor set visually, by analyzing their constructions we find a startling connection between them. We claim that both the Sierpiński carpet and gasket contain infinitely many "slices" of Cantor sets. Let us first consider K_0 of the Sierpiński carpet. If we consider the horizontal slice from $(0, \frac{1}{2})$ to $(1, \frac{1}{2})$ and disregard the y-coordinate we can see that we have K_0 of the middle thirds Cantor set (see blue in figure below).



Now, consider K_1 of the carpet where we have removed the middle square of side length $\frac{1}{3}$. If we consider the same horizontal slice from $(0, \frac{1}{2})$ to $(1, \frac{1}{2})$ and again disregard the y-coordinate we can see that we have K_1 of the middle thirds Cantor set (see blue in figure below).



We can repeat this same process for $K_2, K_3, K_4, ...$ of the carpet which yields $K_2, K_3, K_4, ...$ of the middle thirds Cantor set. Taking the limit of C_i, the horizontal slice of the carpet from $(0, \frac{1}{2})$ to $(1, \frac{1}{2})$ in the i-th iteration of the carpet's construction, as i tends to infinity yields

$$\lim_{i \to \infty} C_i = C_3$$

where C_3 is the middle thirds Cantor set. Note that this is certainly not the only slice of the carpet which reveals a copy of C_3; there are infinitely many! We can take other vertically-translated horizontal slices, vertical slices, or even slices beginning at later iterations of construction; all of which reveal scaled or unscaled copies of the middle thirds Cantor set. We may do the same for the Sierpiński gasket, instead looking at slices of the gasket after removing the middle triangles at a specific iteration.

3 Carpet and Gasket Generating Software

3.1 Description

The following code is written in Python and utilizes the Matplotlib and NumPy libraries. The program uses the gasket and carpets' corresponding iterated function systems to map the objects to a specified number of iterations. The user is given the choice of which object to generate, whether to generate a still or animation of the chosen object, and the number of iterations to generate. For the still generation, each iteration is assigned a unique color that is given to calculated points not seen before in previous iterations.

3.2 Limitations

While the program succeeds in visualizing the Sierpiński carpet and gasket, it is limited in terms of efficiency as it can only efficiently generate the carpet out to 5 iterations and the gasket out to 9 iterations. This is due to the exponential increase in the number of point dilations and translations at each subsequent iteration. In the case of the carpet, we see that K_1 requires dilating and translating only 4 points, the four corners of the unit square for a total of $4 \cdot 8 = 32$ computations. K_2 requires dilating and translating each of the 16 points in K_1 for a total of $16 \cdot 8 = 128$ computations. This growth in the number of computations proves to be extremely slow when trying to generate higher numbers of iterations.

```
import matplotlib.pyplot as plt
import numpy as np
import time

carpet_ifs = [
  lambda x, y: (x / 3, y / 3),
  lambda x, y: (x / 3, y / 3 + 1 / 3),
  lambda x, y: (x / 3, y / 3 + 2 / 3),
  lambda x, y: (x / 3 + 1 / 3, y / 3),
  lambda x, y: (x / 3 + 2 / 3, y / 3),
  lambda x, y: (x / 3 + 1 / 3, y / 3 + 2 / 3),
  lambda x, y: (x / 3 + 2 / 3, y / 3 + 1 / 3),
  lambda x, y: (x / 3 + 2 / 3, y / 3 + 2 / 3)
]

gasket_ifs = [
  lambda x, y: (x / 2, y / 2),
  lambda x, y: (x / 2 + 1 / 2, y / 2),
```

```
  lambda x, y: (x / 2 + 1 / 4, y / 2 + np.sqrt(3) / 4)
]

def generate_fractal_object_animate(points, ifs, num_iterations, fractal_name):
  iteration = 0
  plt.plot(points[:, 0], points[:, 1], 'ko', markersize=1)
  plt.title(f"{fractal_name}: Iteration: {iteration}")
  plt.axis('equal')
  plt.draw()
  plt.pause(1)
  plt.close()
  iteration += 1

  for i in range(int(num_iterations)):
    new_points = np.zeros((0, 2))
    for point in points:
      for f in ifs:
        new_points = np.vstack([new_points, f(point[0], point[1])])
    points = new_points
    plt.plot(points[:, 0], points[:, 1], 'ko', markersize=1)
    plt.title(f"{fractal_name}: Iteration {iteration}")
    plt.axis('equal')
    plt.draw()
    plt.pause(1)
    plt.close()
    iteration += 1

def generate_fractal_object_still(points, ifs, num_iterations, fractal_name):
  color_codes = ['ko', 'ro', 'bo', 'go', 'cD', 'mo', 'yo', 'bo', 'ro', 'go']

  fig, ax = plt.subplots() # create a new figure and axis
  iteration = 0
  ax.plot(points[:, 0], points[:, 1], color_codes[iteration], markersize=1)
  iteration += 1
  colors = [iteration - 1] * len(points)

  for i in range(int(num_iterations)):
    new_points = np.zeros((0, 2))
    new_colors = []
    for j, point in enumerate(points):
        for f in ifs:
            new_point = f(point[0], point[1])
            if not np.any(np.all(points == new_point, axis=1)):
              new_points = np.vstack([new_points, new_point])
              new_colors.append(iteration)
    ax.plot(new_points[:, 0], new_points[:, 1], color_codes[iteration], markersize=1)
    colors.extend(new_colors)
    points = np.vstack([points, new_points])
    iteration += 1

  ax.set_title(f"{fractal_name}: Iteration {num_iterations}")
  ax.axis('equal')
  for i, point in enumerate(points):
    ax.plot(point[0], point[1], color_codes[colors[i]], markersize=1)

  fig.canvas.toolbar.pan() # enable panning
  fig.canvas.toolbar.zoom() # enable zooming
```

```
  plt.show()

# Initializes Sierpiński Carpet points and calls fractal generator method
def carpet(num_iterations, present_choice):
  square = np.array([[0, 0], [0, 1], [1, 1], [1, 0]])
  if present_choice == 1:
    generate_fractal_object_still(square, carpet_ifs, num_iterations, "Sierpiński Carpet")
  elif present_choice == 2:
    generate_fractal_object_animate(square, carpet_ifs, num_iterations, "Sierpiński Carpet")

# Initializes Sierpiński Gasket points and calls fractal generator method
def gasket(num_iterations, present_choice):
  triangle = np.array([[0, 0], [1 / 2, np.sqrt(3) / 2], [1, 0]])
  if present_choice == 1:
    generate_fractal_object_still(triangle, gasket_ifs, num_iterations, "Sierpiński Gasket")
  elif present_choice == 2:
    generate_fractal_object_animate(triangle, gasket_ifs, num_iterations, "Sierpiński Gasket")

# Reads user choices from stdin
def main():
  object_choice = input("Please select the fractal object you would like to see generated:\n[1] Sierpiński
  time.sleep(1)
  if object_choice == "1":
    choice_show = "Sierpiński Carpet"
  elif object_choice == "2":
    choice_show = "Sierpiński Gasket"
  present_choice = int(input("Would you like to see a still or animation of the" + choice_show + "?\n[1] St
  num_iterations = int(input("How many iterations of the " + choice_show + " would you like generated?\n"))
  if object_choice == "1":
    carpet(num_iterations, present_choice)
  elif object_choice == "2":
    gasket(num_iterations, present_choice)

if __name__ == "__main__":
  main()
```

4 Reference Materials

Measure, Topology, and Fractal Geometry by Gerald Edgar - Carpet and gasket construction, the implications of pairwise disjointness of an iterated function system's f_i's on dimension calculation

https://wiki.math.ntnu.no/linearmethods/basicspaces/openandclosed - Better understanding the concept of interior points