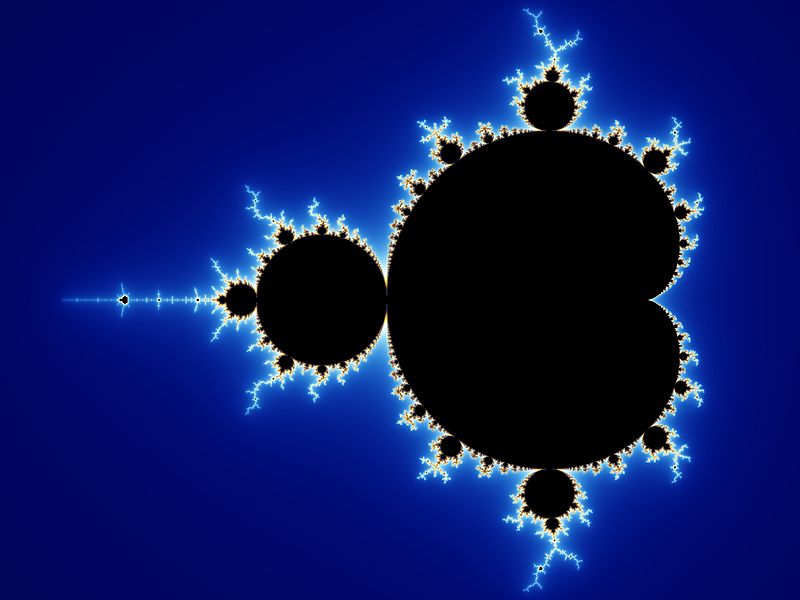
Mathematics HL Internal Assessment

# Fractional Geometry

Julia Bowers



## Introduction

Ever since I learned what fractals were, I’ve loved to doodle the Sierpinski Triangle (also known as the Sierpinski Gasket) and the Koch snowflake, trying to see how many iterations of the pattern I could fit in. It came down to how small of details I could add before my pencil lead became too thick. For different types of fractals, such as Diffusion-Limited Aggregation clusters or the Mandelbrot set, I’ve written computer programs to animate the generation process. The main characteristic of fractals is self-similarity, and this can generate remarkable beauty. As you zoom into a fractal, you are presented with the same (or nearly the same) pattern.

|  |  |
| --- | --- |
|  |  |

Left: Zoom sequence for the Mandelbrot set. It shows ever-finer detail as magnification increases. The set’s boundary contains smaller versions of the main shape, exemplifying the self-similarity property of fractals. Right: the Sierpinski Triangle

However, self-similarity is not what defines fractals. A more indicative characteristic is based on how they scale differently from other geometric figures. Lending “fractals” their name is the concept of fractional dimension (i.e. a dimension that is not an integer), which often exceeds their topological dimension.

In this investigation, I will examine the concept of fractional dimension and determine how to calculate it for some of my favorite fractals.

## What is Dimension?

I know the basics - a point is 0 dimensions, a line 1, a surface 2, and a solid 3. Dimension is typically the number of coordinates that are needed to define a point in a space.

Determining how you can measure an object - its length, area, or volume - can show which dimension the object is. If I try to measure the volume (3 dimension) of a square, I get 0, as the object has a lower dimension than the measuring dimension. If I try to measure the length (1 dimension) of a square, I get infinity. I assume that if you try to measure an object in a higher dimension than the object’s dimensions, you get 0; the other way around, you get infinity. This can help me later to check if the fractional dimension I calculate makes sense.

In order to calculate fractional dimension, I’ll first look at how different integer dimensions affect how an object scales. Though this is very easy to imagine in integer-dimensional space, generalizing this relationship will allow me to apply it to fractals as well.

N = number of self-similar copies

S = scale factor

D = dimension

N = SD

For example, a 3-dimensional cube scaled by a factor of 3 results in 27 self-similar copies of the original cube. This fits the equation where S = 3, D = 3, and N = 27.

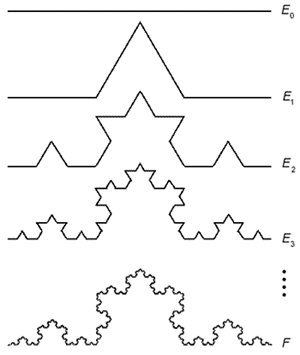
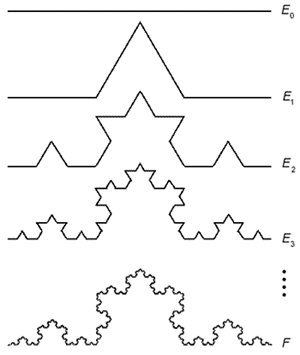
From this relationship, I can write an equation for the dimension D by taking the logarithm of both sides of the equation above, then dividing by log(S):

This leaves me with an formula to calculate the dimension based on the number of self-similar copies N and the scale factor S.

## Dimension of Simple Fractals Generated by Iterative Function Systems

Fractals generated by Iterative Function Systems (IFS) are often self-similar, and have simple rules defining how they are created. The fractal is composed of several copies of itself, each copy being transformed by a function.

One common IFS is the Koch Curve, which is made by starting with a line segment and replacing the middle third of it with two similar line segments. This iterative process is done on each of the four resulting line segments ad infinitum.

 … 

From the image, it is clear that it has an area of 0, so it is not a 2D object. The level of detail is limited by the number of pixels that can be used to represent this object, but I must assume that its detail continues infinitely. I can infer that it has an infinite perimeter, and given that it is in a finite space, I know that it is not simply a 1-dimensional object. I’m going to guess that it has a fractional dimension between 1 and 2.

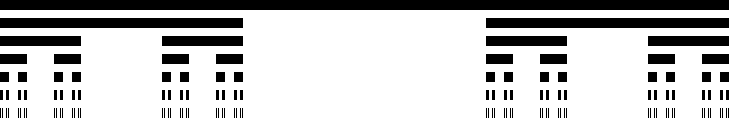
I will try to apply the equation , derived above, to the Koch Curve.

Each self-similar copy is ⅓ the length of the original, so the scale factor S is 3 and the number of self-similar copies N is 4.

Substituting these values for S and N in the dimension equation I derived earlier,

This makes sense, as 1 < 1.262 < 2: the Koch curve is between 1- and 2- dimensional.

I can try to apply this to another fractal, the Cantor ternary set, which has a dimension of less than one.



In this case, the number of self-similar copies, N, is 2 and the scale factor, S, is 3.

The dimension of the Cantor set is approximately 0.631.

## Dimension of non-Iterative Function System Fractals

Finding the dimension of a fractal with constant N and S values wasn’t too difficult, but what about a more complicated fractal-like object, such as the coastline or boundary of a country? The coast*line* itself is not an area (2-dimensional), but it’s not simply 1-dimensional either, just as the Koch curve wasn’t. I’ll guess it has a dimension 1 < D < 2.

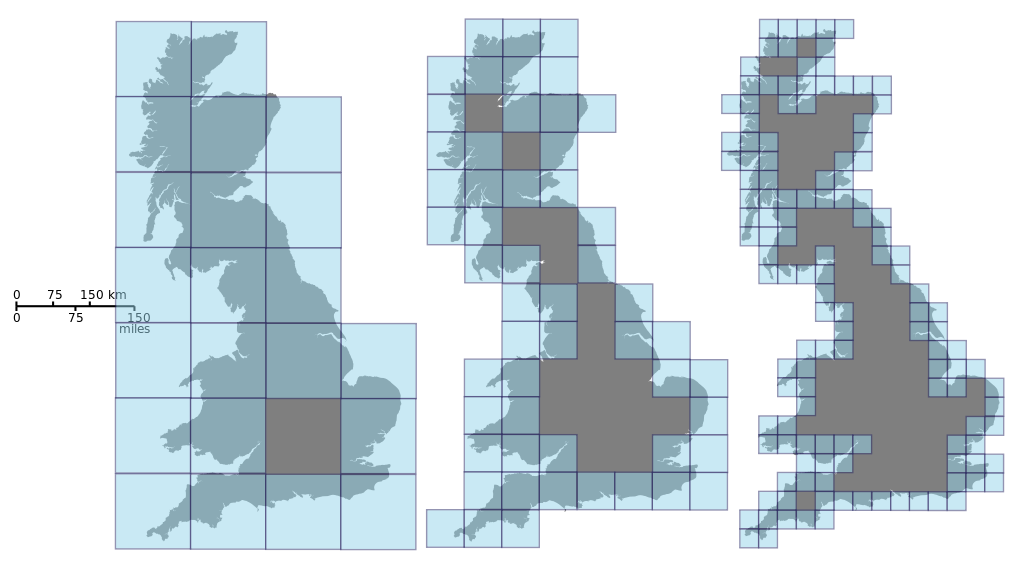
I’ll start off by considering how I would measure a coastline. Given a ruler of length s, the length of the coastline L is s times the number of rulers, N, needed to go around the coast.

L = sN

However, as my ruler gets shorter (measuring in, say, centimeters rather than kilometers), the length becomes greater as the smaller ruler is able to measure rough parts of the coast in more detail. This is akin to scaling the country by a factor of S and the perimeter (the coastline) increasing by a factor of SD where 1<D<2. This is referred to as the **coastline paradox**, originally observed by Lewis Fry Richardson.

I’m going to go back to my equation to try to figure out how I’d find the dimension of a coastline.

Using the box-counting method, I can determine what is called the Minkowski-Bouligand dimension.



The number of self-similar copies is N, the number of boxes I have. For every length of box s, there is a specific value for N, which I will denote as N(s). The scale factor S is 1/s. I then get:

I want my box length s to become as small as possible to measure the greatest amount of detail, so I will take the limit as s approaches zero.

### Determining the Fractional Dimension of Italy’s Northern Border

To figure out the fractional dimension in practice, I will measure the border in increasingly smaller increments. Since of course I can’t actually measure the physical border, I will use a printed map of Italy and rulers of length s = 10, 5, 2, and 1 cm. I will do the box-counting algorithm by hand by making a grid with intervals of s.

I should keep in mind that this border might not be a true fractal, exhibiting self-similarity at all scales, because there are likely to be some straight segments.

|  |  |
| --- | --- |
|  |  |
|  |  |

Map images are a screenshot of Google Maps

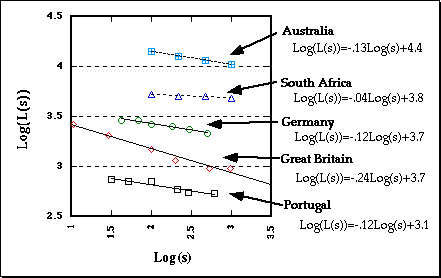
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Length of ruler, cm: s | 10 | 5 | 2 | 1 |
| Number of boxes: N(s) | 4 | 9 | 31 | 64 |

Since the Minkowski-Bouligand dimension is given by , The slope of the trendline generated from an log(N(s))-against-log(1/s) plot will be approximately equal to the fractional dimension of the fractal coastline.



The slope of this trendline is 1.224, therefore Dbox ≅ 1.224. It makes sense that the fractional dimension of Italy’s northern border be between 1 and 2: not quite a 1-dimensional line, but also not a 2-dimensional area.

I was able to find plots for several other countries’ borders[[1]](#footnote-1): Australia, South Africa, Germany, Great Britain, and Portugal. They have been graphed as log(L(s)) vs. log(s), where L = sN(s). The slope ends up being 1-D.



I checked that this representation follows the definition for Minkowski-Bouligand dimension by deriving from Dbox:

This shows that in fact, the trendline of log(L(s)) vs. log(s) with slope (1 - D) also accurately gives the Minkowski-Bouligand dimension.

The results given by the plot I found online showed that Great Britain’s coastline’s fractional dimension of 1.24 is similar to that of Italy’s northern border (1.224). This made sense to me since they both seem very crinkly or rough. This contrasted with the one given for South Africa, which is 1.04, meaning it’s almost one-dimensional. This is not surprising since, looking at a map of the coastline, it is very smooth. A more “rough” fractal will have a dimension closer to 2, whereas a less rough fractal will have a dimension closer to 1.

## Hausdorff Dimension

The measure of roughness I noticed when comparing coastlines has been formally defined by Hausdorff. This definition is a generalization of the dimensions I’ve been finding throughout this investigation, and a successor to the box-counting, or Minkowski-Bouligand, dimension.

Hausdorff dimension d is defined in relation to the Hausdorff measure of the subset, A, of metric space X. The Hausdorff measure Hd(A) is defined as the infimum of positive numbers y such that for every r > 0, the set A can be covered by a countable family of closed sets, each of diameter less than r, such that the sum of the d-th powers of their diameters is less than y. This is very similar to the box-counting definition, except instead of squares, circles (of diameter less than r) are used.

## Dimension of Diffusion-Limited Aggregation Clusters

I’d like to take a look at some other fractals I am fond of, such as cluster fractals. Diffusion-Limited Aggregation (DLA) cluster fractals are created by placing a “seed” pixel at (0,0), then adding random-walking pixels outside of the cluster that stick to existing stationary pixels.

Similar to how I calculated the fractional dimension of Italy’s northern border, I will calculate the dimension D.

, where N is the number of pixels and 1/s is the radius R of the cluster, which I’m measuring after every iteration.

I decided to write a computer program in the Python language to mimic the creation of a DLA cluster. For every pixel that is added, I record the number of pixels N, and the radius R of the smallest circle that can enclose the cluster. The code is located in the Appendix.

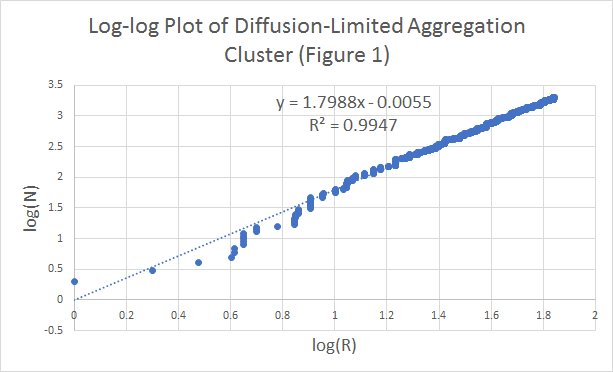
I ran into a challenge trying to make the initial coordinates of a point have a uniform chance of being anywhere outside the enclosing circle of radius R. I solved this issue by placing the point on a circle of radius R+2, and at random determined a degree between 0 and 359. From there, I used the sine and cosine functions to determine the x and y coordinates.

Here is an image of my first DLA cluster (Figure 1), based on 2000 iterations:



The Octave code used to plot it is also located in the Appendix.

I then graphed a log(N)-vs-log(R) plot of the Figure 1 DLA cluster, in order to determine its dimension, which is equal to the slope of the trendline, generated by Microsoft Excel. The fractional dimension of Figure 1 that I have calculated is approximately 1.7988.



I tried this again for another DLA cluster of 2000 iterations (Figure 2, see Appendix for plot and log-log graph), with a fractional dimension of approximately 1.6756. This shows that even with the same method of generation and same number of iterations, two DLA fractals based on random motion can have different dimensions

## Dimension of the Mandelbrot Set

The Mandelbrot set consists of all possible complex numbers C for which the function , starting at z = 0, does not diverge (i.e. it remains bounded). One potential way to compute its fractional dimension would be through the box-counting method, treating the set like a coastline.

However, it has been proved by Mitsuhiro Shishikura that the boundary of the Mandelbrot set is 2. This has been done by first showing the relationship between the Mandelbrot set and the Julia set. The Julia set is defined by for a fixed C. For every point z on the complex plane, zn+1 will either remain small or approach infinity. The complex number z is in the Julia set for a particular C value if it does not approach infinity. In his proof, Shishikura used the property that there are is set, M, of values of C where the Julia set is connected (it is not dust-like), and M is the Mandelbrot set.

In his 1998 paper, Shishikura first shows that the dimension of the Mandelbrot set is at least as big as the dimension of his sequence of connected Julia sets, then shows that the dimension of the sequence of sets eventually reaches 2. Hence, the dimension of the Mandelbrot set is 2.

## Conclusion

In this investigation, I have used several different methods to calculate the dimension of various fractals. Sometimes, the simple equation cannot be used, for example if the fractal does not have an easily-identifiable number of self-similar copies and scale factor. In these cases, I instead opted for the box-counting method or my iteration-and-radius method for DLA cluster fractals. In addition, I found that for the DLA cluster fractals, since every fractal produced this way is unique, even the same number of iterations produces fractals of differing dimensions (the ones I calculated were 1.7988 and 1.6756).

In calculating the dimension of the Mandelbrot set, I could have used a method based on data collection, but I found Shishikura’s proof to be more interesting and elegant. In addition, even if the calculated dimension was very close to 2, it is likely that I would not be able to conclusively say it that it was exactly 2 based on my data.

Visualizing the concept of non-integer dimensions can be challenging, but through this study of fractals and the determination of their Minkowski-Bouligand and Hausdorff dimensions, this idea is more clear. Certainly I can now see the relationship between how “rough” a fractal (or fractal-like coastline) appears and its calculated dimension.

# Bibliography

Bown, William. 1991. "Science: Mandelbrot Set Is As Complex As It Could Be". *New Scientist*. https://www.newscientist.com/article/mg13117882-900-science-mandelbrot-set-is-as-complex-as-it-could-be/.

C, Gretchen. 2004. "Model Of Diffusion-Limited Aggregation". https://www.phas.ubc.ca/~berciu/TEACHING/PHYS349/DLA.pdf.

"Fractals & The Fractal Dimension". 2018. *Vanderbilt.Edu*. Accessed February 22. https://www.vanderbilt.edu/AnS/psychology/cogsci/chaos/workshop/Fractals.html.

Joyce, David. 1994. "Julia And Mandelbrot Sets". *Mathcs.Clarku.Edu*. https://mathcs.clarku.edu/~djoyce/julia/julia.html.

Shishikura, Mitsuhiro. "The Hausdorff Dimension of the Boundary of the Mandelbrot Set and Julia Sets." *Annals of Mathematics*, Second Series, 147, no. 2 (1998): 225-67. doi:10.2307/121009.

[Weisstein, Eric W.](http://mathworld.wolfram.com/about/author.html) "Hausdorff Dimension." From [*MathWorld*](http://mathworld.wolfram.com/)--A Wolfram Web Resource. <http://mathworld.wolfram.com/HausdorffDimension.html>

[Weisstein, Eric W.](http://mathworld.wolfram.com/about/author.html) "Minkowski-Bouligand Dimension." From [*MathWorld*](http://mathworld.wolfram.com/)--A Wolfram Web Resource. <http://mathworld.wolfram.com/Minkowski-BouligandDimension.html>

### Images:

Mandelbrot zoom sequence (Image courtesy Wolfgang Beyer)

Beyer, Wolfgang. 2006. *Mandelbrot Zoom Sequence*. Image. https://commons.wikimedia.org/wiki/User:Wolfgangbeyer.

Sierpinski Triangle

*Sierpinski Gasket*. 2018. Image. Accessed February 22. http://www.math.ubc.ca/~cass/courses/m308-03b/projects-03b/skinner/ex-dimension-sierpinski\_gasket.htm.

Koch curve

Falconer, K. *Koch curve*. 1990, *Fractal Geometry: Mathematical Foundations and Applications* (Chichester: John Wiley and Sons). Accessed from http://www.scielo.br/scielo.php?script=sci\_arttext&pid=S0103-97331998000200007

Cantor ternary set

*Cantor Ternary Set*. 2007. Image. https://commons.wikimedia.org/wiki/File:Cantor\_set\_in\_seven\_iterations.svg.

Coast of Britain

Monnerot-Dumaine, Alexis. 2010. *Estimating The Box-Counting Dimension Of The Coast Of Great Britain*. Image. https://commons.wikimedia.org/wiki/File:Great\_Britain\_Box.svg.

Log-log plot of several coastlines

*The Richardson Effect*. 2018. Image. Accessed February 22. https://www.vanderbilt.edu/AnS/psychology/cogsci/chaos/workshop/Fractals.html.

Mandelbrot set (Image courtesy Wolfgang Beyer)

Beyer, Wolfgang. 2013. *Mandelbrot Set*. Image. https://commons.wikimedia.org/wiki/File:Mandel\_zoom\_00\_mandelbrot\_set.jpg.

# Appendix

### DLA-cluster Python code:

import random  
import math  
  
def write\_out(point\_list, filename):  
 f = open(filename, 'w')  
 for z in point\_list:  
 f.write(str(z[0])+ "," + str(z[1]) + '\n')  
  
def checkBeside(x, y, list):  
 for z in list:  
 if (z[0] == x):  
 if (z[1] == y+1) or (z[1] == y-1):  
 return True  
 elif (z[1] == y):  
 if (z[0] == x+1) or (z[0] == x-1):  
 return True  
 return False  
   
points = []  
## a point will have [x, y, d]  
  
data = []  
## [N, r] where n is the number of points and r is the radius of the enclosing circle  
  
points.append([0,0,0])  
r = 0 #max radius  
  
print("N: \tR:")  
  
i = 2000  
  
while len(points) < i:  
 # make a new point  
   
 # choose degree randomly  
 rad = (math.pi/180) \* random.randint(0, 359)  
   
 # calculate x and y  
 limit = math.ceil(r + 2)  
 newX = int(limit \* math.sin(rad))  
 newY = int(limit \* math.cos(rad))  
   
 # move this point until it sticks  
 while not checkBeside(newX, newY, points):  
 #print("move")  
 #print(newX, newY)  
 move = random.randint(0, 3)  
 if (move == 0) and (newX < limit):  
 newX += 1  
 elif (move == 1) and (newX > -limit):  
 newX -= 1  
 elif (move == 2) and (newY < limit):  
 newY += 1  
 elif (move == 3) and (newY > -limit):  
 newY -= 1  
   
 d = (newX\*\*2 + newY\*\*2)\*\*(1/2.0)  
   
 if d > r:  
 r = d  
   
 points.append([newX, newY, d])  
 data.append([len(points),r])   
   
 print(str(len(points)) +"\t" + str(r))  
  
#write pixels to csv file   
write\_out(points, "dla\_points\_1\_" + str(i) + ".csv")  
write\_out(data, "dla\_data\_1\_" + str(i) + ".csv")

### Octave plotting code:

data = load('dla\_points\_1\_2000.csv');

%plot data

plot(data(:,1),data(:,2), ".", "markersize", 12);

%you may need to vary markersize to get more elegant results

### Figure 2 (DLA cluster):

|  |  |
| --- | --- |
|  |  |

## 

1. <https://www.vanderbilt.edu/AnS/psychology/cogsci/chaos/workshop/Fractals.html> [↑](#footnote-ref-1)