

Fractal Dimension in Image Resolution

Internal Assessment

ADITYA ADVANI

Word Count- 5009

Table of Contents

1.0) Introduction	3
1.1) Rationale	3
1.2) Aim- Research Question.....	3
2.0) Defining Mathematical Processes.....	3
2.1) Hausdorff Dimensionality [□]	3
2.2) Conjecture forming using Geometric Progressions-.....	3
2.3) Isoclines-.....	4
2.4) Taylor Series Approximations-.....	4
3.0) Data Collection	4
3.1) Fractal Selection [□]	4
3.2) Snapshots of Data-.....	5
4.0)Body.....	6
4.1)Derivations.....	6
4.3)Calculating the Derivatives of each function-	7
4.4) An Example- Serpinski's Triangle	7
4.5)Computation of Isoclines of Fractal Vertex Growth Conjectures	11
5.0)Trends Observed- Putting it together.....	11
5.1)Testing the Research Question	11
5.2)A newly apparent trend.....	12
6.0)Math- Taylor Approximations	13
7.0)Evaluation and Conclusion	18
7.1) Conclusion	18
7.2) Strengths	19
7.3) Limitations.....	19
7.4) Implications and Applications	19
7.5) Proposed Extensions	19
8.0) Bibliography	20

1.0) Introduction

1.1) Rationale

Fractals as a concept intrigued me as they exist in dimensions between the 1st and the 2nd and between the 2nd and the 3rd. I had always thought of dimensions as existing as integers, and their existence changed how I looked at the concept as a whole. When studying mathematical concepts, I always found visuals to be more interesting. Fractals were the top of the Serpinski's pyramid of visual concepts that I wished were in the IB portion. I had it set in my mind, that for my Math IA, I would explore them in some way or another.

Because of my interest in the topic, I read about fractals from online websites and watched YouTube videos on them. I remember particularly finding it funny, how a tree fractal is present in the veins of a leaf, while leaves are present on a tree- and I would see fractals everywhere. These ideas culminated in my head when I saw how an image gained resolution through a program. I thought about it, and a fractal program seemed to gain resolution like any other image, as one runs it and zooms in infinitely. This was the intuition, that got me thinking about how fractals and resolution of images. Very soon later, I was able to make the connection of fractal vertices and pixels, and come up with my IA investigation idea. I was sure that I wanted to use fractals to see an image gain resolution, and my curiosity wished to find a trend in the many types of fractals and their effectiveness in this particular application.

1.2) Aim- Research Question

To what extent, can fractals be used in image resolution software?

This application follows the following intuition- each pixel corresponds to a vertex of the fractal, and as a fractal program runs its successive iterations, the increasing number of vertices (corresponding to an increasing number of pixels) lead to an increase in image resolution.

To develop understanding, I would like to look for a trend between Hausdorff dimensionality and fractal growth, so that this concept can be effectively applied.

2.0) Defining Mathematical Processes

2.1) Hausdorff Dimensionality^[1]

This concept is used to determine a concept of a fractal. Hausdorff defines dimensionality as the power to which one raises side length of the object, when its effective mass is increased.

This is defined by the form^[2]

$$\text{Scaling factor}^{\text{Dimensionality}} = \text{Capacity of body}$$

2.2) Conjecture forming using Geometric Progressions-

A Geometric sequence is a series of terms starting from an initial term a , where the next term is defined as the previous term multiplied by some constant r .

When one observes a pattern which contains a geometric sequence, one can use the formula

$$T = ar^{(n-1)}$$

To determine the next point in the series. (The n th term)(T)

If the series involves a sum, one can use the form

$$S_n = \frac{a(r^n - 1)}{r - 1}$$

to determine the sum after n terms have passed.

This is useful in forming a conjecture of how many vertices a fractal has after a certain number of iterations.

¹ "Hausdorff Dimension." Wikipedia. Wikimedia Foundation, 08 Sept. 2019. Web. 08 Sept. 2019.

² Kohavi, Yuval, and Hadar Davdovich. *Topological Dimensions, Hausdorff Dimensions & Fractals*. Publication. N.p., 2009. Web. 8 Sept. 2019.

2.3) Isoclines-

Isoclines are constant slopes applied to graphs which do not have constant slopes.

To find the isocline at a point, one must differentiate the function and plug in the x coordinate of the point.

This gives the slope of the function of the number of vertices of the fractal at that iteration number, which represents the rate of growth of the number of vertices at a certain iteration number.

2.4) Taylor Series Approximations-

These are ways to approximate functions as polynomial series to great accuracy. We can use this, as polynomial functions usually play much better with most mathematical concepts like integration and differentiation, and even coding, than other forms of functions.

Taylor polynomials are considered by approximating a function around a certain x value (a point of interest), and gaining accuracy as one increases the number of terms.

To make an imposter function, we can replicate multiple aspects of the original at a certain point, and we can use simple calculus to gain insight into the aspects of these functions.

We can gain information about the position of the function, the slope at the point of interest, the concavity, etc. This information is given by the successive derivatives.

Matching these in a new function gives a good approximation. This will be explored once conjectures for fractal growth have been established.

3.0) Data Collection

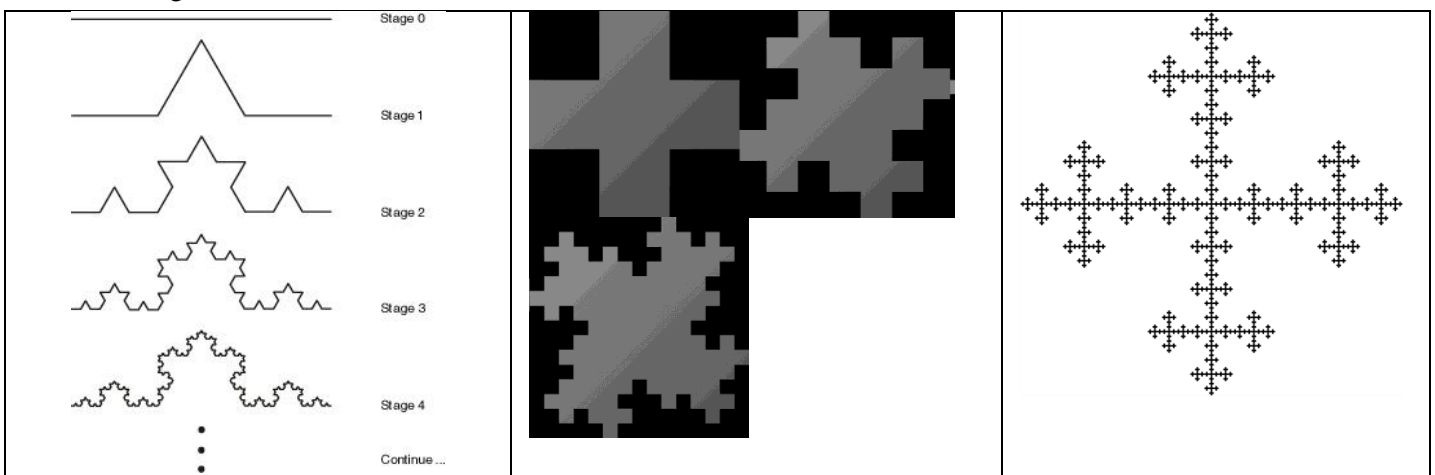
3.1) Fractal Selection^[3]

Research was conducted on fractal dimension from online, and fractal types were selected for the investigation-

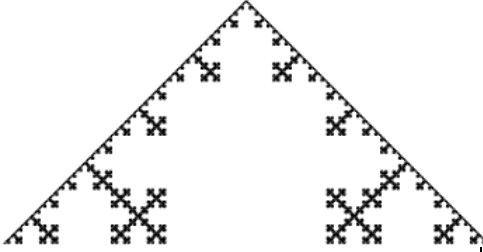
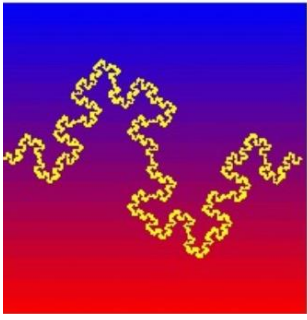
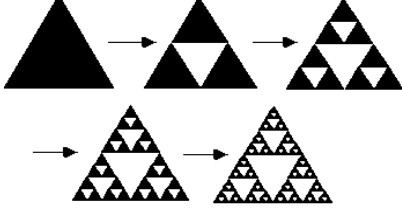
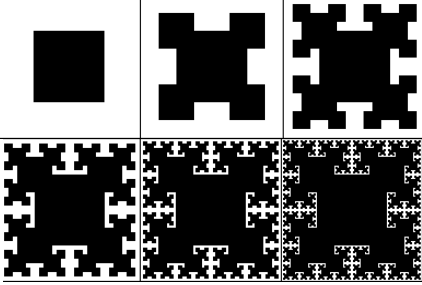
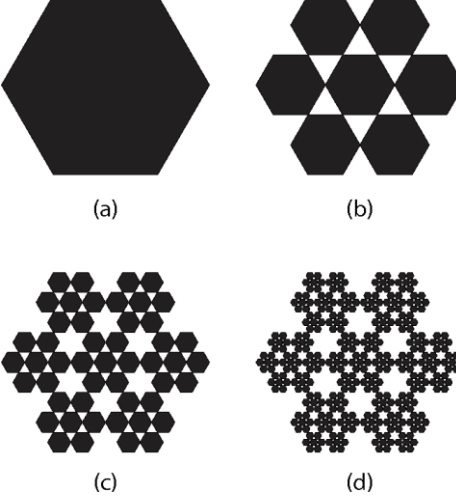
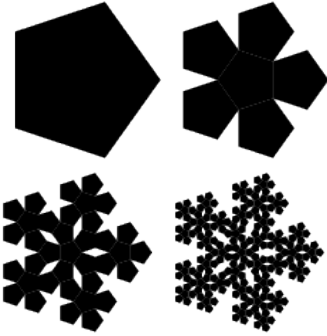
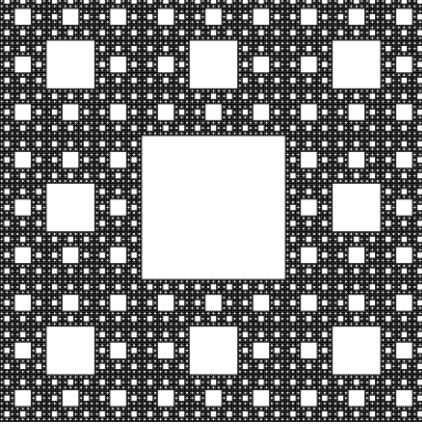
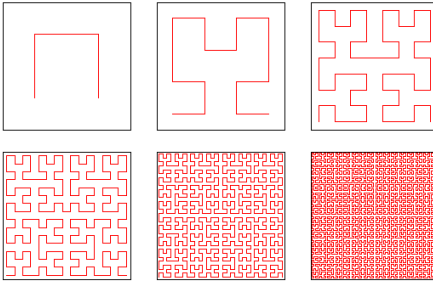
The selected Fractal types are:

- Von Koch Snowflake
- Quadratic Von Koch Crossflake
- Vicsek Fractal
- Quadratic Von Koch Snowflake
- Minkowski Sausage
- Serpinski's Triangle
- Boundary of a T-Square
- Hexaflake
- Pentaflake
- Serpinski's Carpet
- Hilbert's Curve

Here are images of chosen fractals-



³ "List of Fractals by Hausdorff Dimension." *Wikipedia*. Wikimedia Foundation, 06 Sept. 2019. Web. 08 Sept. 2019.

Von Koch Snowflake	Quadratic Von Koch Crossflake	Vicsek Fractal
		
Quadratic Von Koch Snowflake	Minkowski's Sausage	Serpinski's Triangle
		
Boundary of a T-Square	Hexaflake	Pentaflake
		
Serpinski's Carpet	Hilbert's Curve	

*NOTE: It should be noted, that all the fractals are being applied to images, which are 2D, and hence, the fractals being considered must work in less than 2 dimensions for one to be able to make a link between their vertices and the pixels on an image. It is also necessary, that fractals have a dimensionality of more than 1 for this, as fractals with a dimensionality below one work between dimensions 0 and 1.

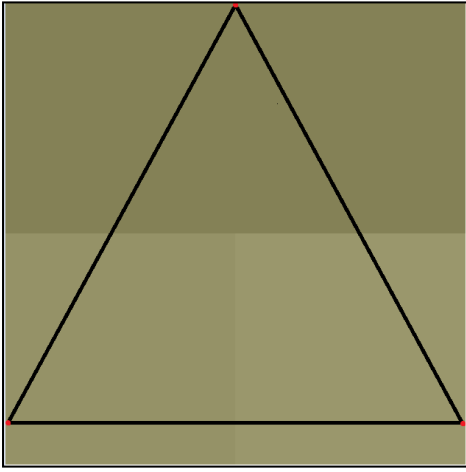
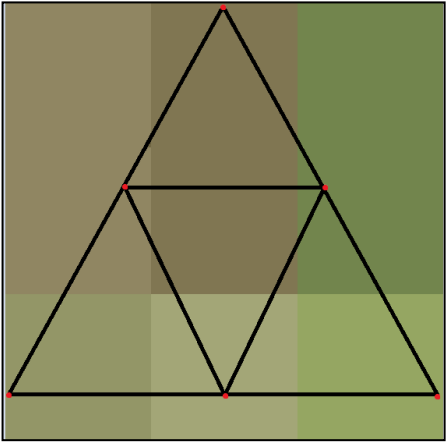
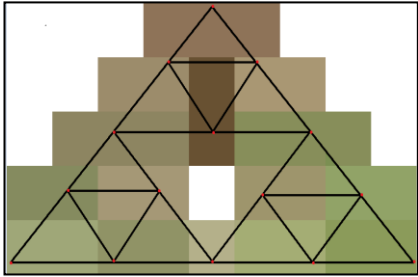
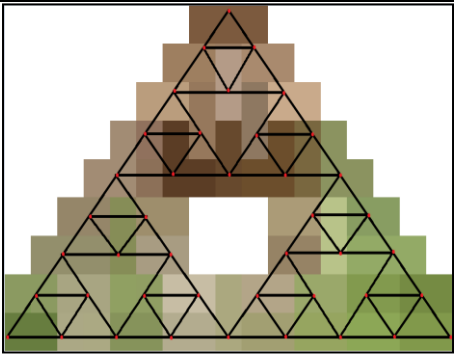
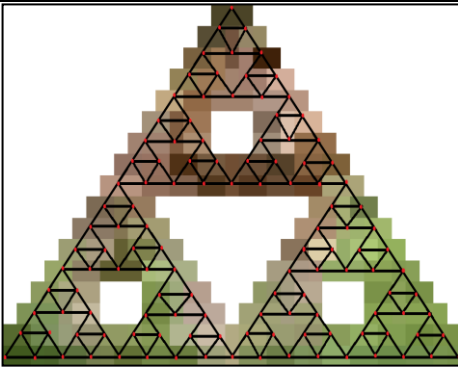
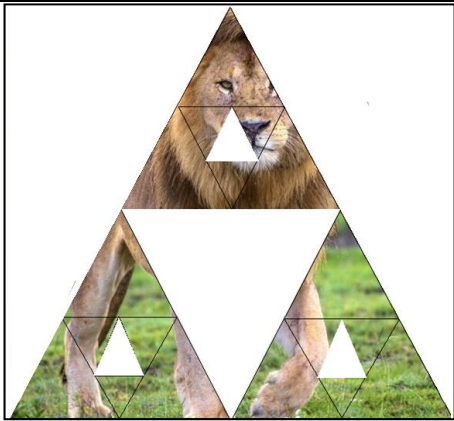
3.2) Snapshots of Data-

Here is an example of how the shortlisted fractals can be applied to resolution as proposed.

The pixels of an image can correspond to a vertices of a fractal, and how this image can then gain resolution by having the fractal repeat successive iterations.

Example: Serpinski's Triangle

Hausdorff Dimensionality = 1.5849

		
Iteration- 0	Iteration- 1	Iteration- 2
Vertices- 3	Vertices- 6	Vertices- 15
		
Iteration- 3	Iteration- 4	Final Iteration
Vertices- 42	Vertices- 123	Vertices- $\lim_{n \rightarrow \infty} n$ (Where $n =$ number of vertices)

A similar example can be considered for each fractal type.^{[4][5]}

4.0)Body

4.1)Derivations

Deriving the Hausdorff Dimensionality for each Fractal Type-

Before one observes and analyzes fractal types, one must have a way to categorize them. This is done through hausdorff dimensionality.

First, I isolated multiple fractals, and found their hausdorff dimensionalities.

To do so, one must redefine dimensions. One can approach this using regular integer dimensions first.

First, one can consider a line of length 1. When scaled by a factor of 2, the line's mass is scaled by a factor of 2.

However, the same cannot be said for a square of side 1. When scaled by a factor of 2, the square's mass is scaled by a factor of 4.

This can be extended to a cube of side 1. When scaled by a factor of 2, the cube's mass is scaled by a factor of 8.

We can look at this in a different way.

⁴ "Online Image Resizer - Crop, Resize & Compress Images, Photos and Pictures for FREE." *Online Image Resizer - Crop, Resize & Compress Images, Photos and Pictures for FREE*. N.p., n.d. Web. 08 Sept. 2019.

⁵ "MS Paint." Web 08 Sept 2019.

The reason the line was scaled by 2, is because it is 1 dimensional- giving a scaling factor of 2^1

The same goes for the square in 2 dimensions,- scaling by 2^2 (=4)

And so, for the cube, in 3 dimensions, scaling by 2^3 .

Hence, the hausdorff dimensionality defines how the capacity of a shape changes as it is scaled by a certain amount.

Hausdorff dimensionality changes what it means to be 1D, 2D, or 3D, we can explain it through a conjecture.

$$\text{Scaling factor}^{\text{Dimensionality}} = \text{Capacity of body}$$

Now, one can extend this definition to fractal shapes.

We can scale the side of the fractal by a constant, and note the change in capacity. Then, plug values into the formula to find a hausdorff dimensionality for that fractal.

The fractals being considered are the ones listed in the previous section. (Data Collection)

*NOTE: It should be noted, that all the fractals are being applied to images, which are 2D, and hence, the fractals being considered must work in less than 2 dimension for one to be able to make a link between their vertices and the pixels on an image. It is also necessary, that fractals have a dimensionality of more than 1 for this, as fractals with a dimensionality below one work between dimensions 0 and 1.

The measure of the hausdorff dimensionality serves as a way to look at fractals in a pattern.

This allows for a way to define fractals using a continuous variable. One can now operationalize the nature of a fractal using this method, which is why this measure of dimensionality was considered for developing a trend.

4.2) Conjectures that will be used:

Forming Conjectures for the growth of the number of vertices in successive iterations of each Fractal type

In this section, we can attempt to analyze the growth of fractals by observing how many vertices are present in each successive iteration of a fractal pattern. To do this, we must first form a conjecture for a fractal pattern which gives the number of vertices after each iteration.

We can go about this, by first collecting data about how many vertices each iteration of a certain fractal has.

Secondly, we can find a pattern for the number of vertices in each fractal iteration using a G.P. Form of $Term =$

$$Initial\ Number \times Common\ Ratio^{(Iteration\ number - 1)}$$

Then, we can arrange this data into a sum, from 0 to any iteration n, which gives the number of vertices present at that iteration.

Lastly, it is a matter of using a sum of terms formula

$$Sum\ of\ vertices\ until\ n\ iterations = initial\ number\ of\ vertices \frac{(Common\ ratio^{(iteration\ number\ n)} - 1)}{Common\ ratio - 1}$$

This gives a conjecture for the number of vertices present in fractal and so the number of pixels in an image

The conjectures formed predict exactly how many vertices are formed by the fractal at a particular iteration number.

This means that all fractals, inherently follow power laws, and hence, follow these power trends.

4.3) Calculating the Derivatives of each function-

Each of these functions show how many pixels a certain fractal can provide an image with at a certain iteration number.

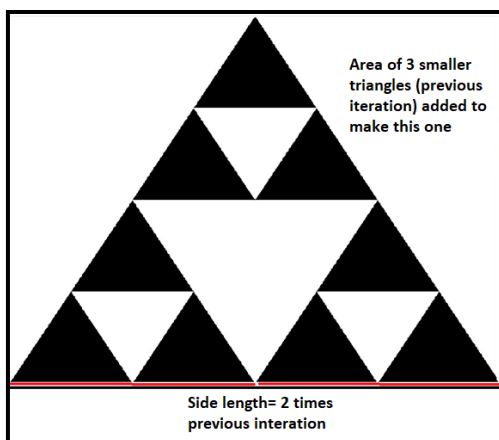
We can do this by calculating the derivative of these functions. This will allow us to plug in iteration numbers and obtain the value of the rate of growth of pixels per iteration at that iteration number- given by an isocline at the iteration number.

4.4) An Example- Serpinski's Triangle

In this section, one can observe the discussed processes applied to an example- Serpinski's Triangle

1) Deriving the Hausdorff Dimensionality of the fractal

When the side is scaled by a factor of 2, Sierpinski's triangle goes to the next iteration, and consists of 3 of the triangles from the previous iteration, arranged with 2 at the base and 1 at the top, forming the scaled up iteration of the triangle. This means that the capacity of the fractal increases by a factor of 3.



$2^D=3$, where D is the Hausdorff dimensionality.^[6]

Plugging into the conjecture formed.

Gives D as 1.5849- the Hausdorff dimensionality of the Sierpinski's Triangle Fractal

The measure of the Hausdorff dimensionality serves as a way to look at fractals in a pattern.

This allows for a way to define fractals using a continuous variable. One can now operationalize the nature of a fractal using this method, which is why this measure of dimensionality was considered for developing a trend.

2) Forming a conjecture for the number of vertices in each fractal iteration

Counted vertices in successive iterations follow the pattern,

$$3, 6, 15, 42, 123$$

Splitting into a sum for the nth iteration,

$$3 + \sum_{n=0}^N 3^n$$

Expressed as a sequence to n terms for the Nth iteration,

$$3 + (3) + (3^2) + (3^3) + (3^4) + \dots + (3^N)$$

Expressing the brackets as a G.P. Summed to n terms

$$3 + \frac{3(3^N - 1)}{3 - 1}$$

Simplifying,

$$\begin{aligned} & 3 + \frac{3^N - 1}{2} \\ & = 3 + 1.5(3^N - 1) \end{aligned}$$

Expressing the number of vertices present in the quadratic von Koch curve after n iterations of the fractal as a conjecture in $3 + 1.5(3^N - 1)$.

This gives a conjecture for the number of vertices present in fractal and so the number of pixels in an image after running N iterations of the fractal.

The conjectures formed predict exactly how many vertices are formed by the fractal at a particular iteration number.

This means that all fractals, inherently follow power laws, and hence, follow these power trends.

⁶ "MS Paint." Web 08 Sept 2019.

Power laws are easier to code than many others due to how one can code an initial function and repeat the code for the next iteration. This means that fractal patterns can be coded very easily and do not stress computers out too much. Implications of this will be considered in a later section.

For example, these codes would be particularly useful if one was considering a homing missile system. The system would not have much power as it would have to be small and light on the missile itself, but with little power, could run a fractal power program (like the ones suggested) and narrow in as the program runs and gains pixels of the target. As an example under this, if the system was running Serpinski's triangle, the system would run iteration 2, and wait for the target to fall into either one of the 3 smaller triangles from iteration 1 that form this. Then it would repeat this for a smaller triangle (split this too, into 3 smaller serpinski triangle fractals) and so, gain resolution and pixels, gaining precision in its mark.

3) Taking the derivative of the conjecture and calculating the isocline at different iteration numbers

The derivative of the above conjecture is given by

$$1.5 \times \ln 3 \times 3^N$$

Which represents the rate of growth of the number of vertices at a point in the function.

To calculate this rate of growth, given by an isocline at a certain iteration value, one can plug in the iteration value into N. For example, the isocline at 3 (the 4th iteration) is given by

$$\begin{aligned} 1.5 \times \ln 3 \times 3^3 \\ = 44.49 \end{aligned}$$

Hence, the growth of pixels at the 4th iteration of Hilbert's Curve is 44 pixels per iteration.

Hence, the isoclines for each fractal were found at n=1, 2, 3, 4, and 5.

These will serve to show some semblance of the rate of growth of pixels for a certain fractal type.

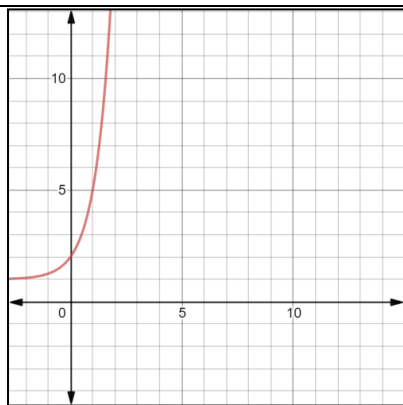
4) Generalizing to other fractal types:

The previous processes were repeated for all other fractal types.

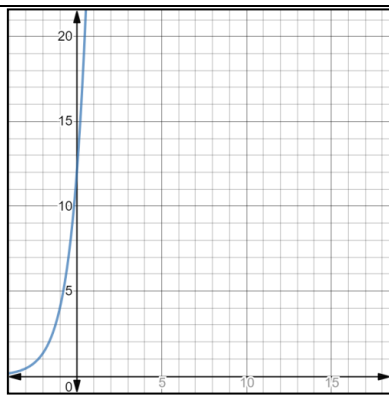
Hence, the following information was deduced.

Fractal Name	Hausdorff Dimensionality	Conjecture formed to represent number of vertices at x th iteration	Derivative used to calculate Isocline.
Von Koch Snowflake	1.2619	$1+4^x$	$4^x (\ln 4)$
Quadratic Von Koch Crossflake	1.37	$12(3^x)$	$3^x (12 \ln 3)$
Vicsek Fractal	1.4649	$5^{(x+1)}$	$(5 \ln 5) 5^x$
Quadratic Von Koch Snowflake	1.4649	$1+5^x$	$5^x \ln 5$
Minkowski Sausage	1.5	$2+(6(8^x-1))/7$	$((6 \ln 8)/7) 8^x$
Serpinski's Triangle	1.5849	$3+(3(3^x-1))/2$	$(1.5 \ln 3) 3^x$
Boundary of a T-Square	1.5849	$(8)3^x-4$	$(8 \ln 3) 3^x$
Hexaflake	1.6309	$2+(4)7^x$	$(4 \ln 7) 7^x$
Pentaflake	1.8617	$2+(3(5^{(x+1)}-1))/4$	$(3.75 \ln 5) 5^x$
Serpinski's Carpet	1.8928	$(4)9^x$	$(4 \ln 9) 9^x$
Hilbert's Curve	2	$4^{(x+1)}$	$(4 \ln 4) 4^x$

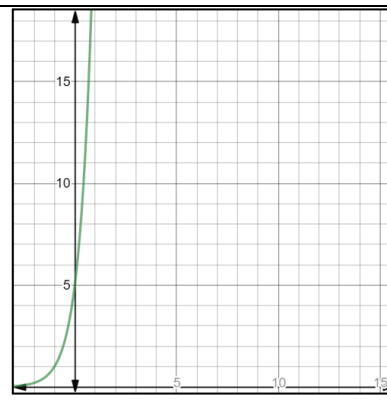
Here are graphs plotted which show how the conjectures look. The number of pixels are plotted on the y axis and the iteration number on the x axis.



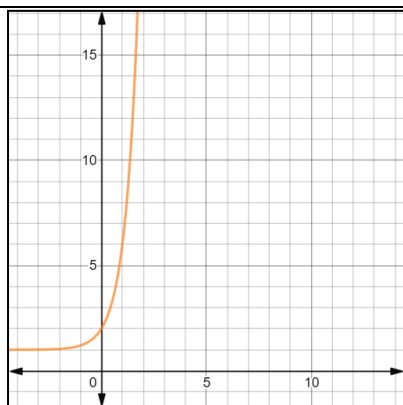
Von Koch Snowflake



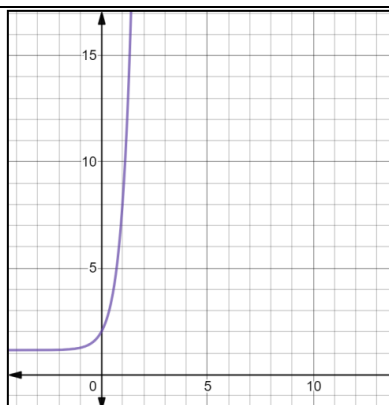
Quadratic Von Koch Crossflake



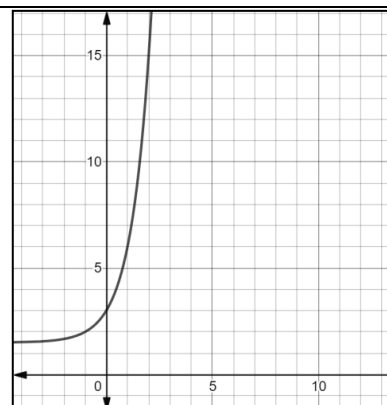
Vicsek Fractal



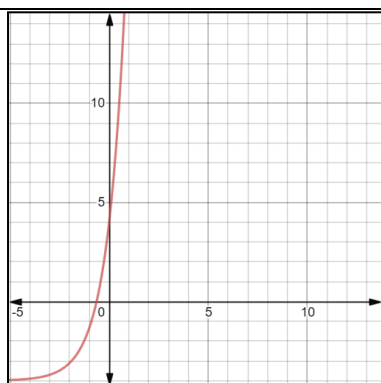
Quadratic Von Koch Snowflake



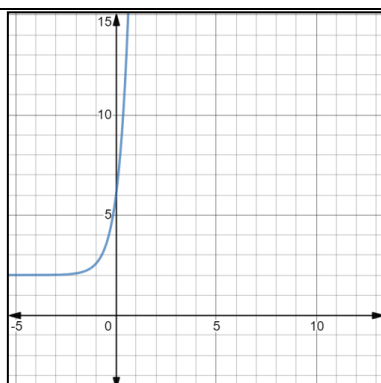
Minkowski Sausage



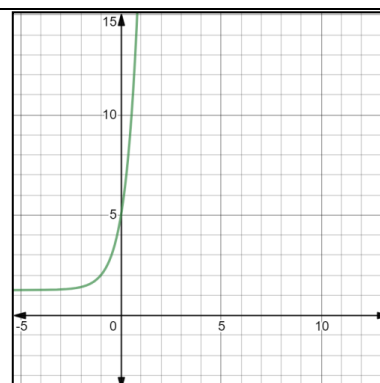
Serpinski's Triangle



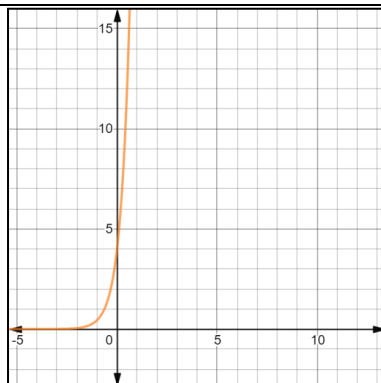
Boundary of a T-Square



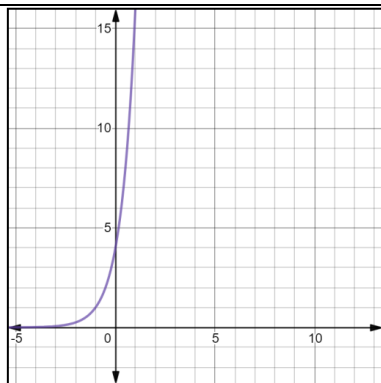
Hexaflake



Pentaflake



Serpinski's Carpet



Hilbert's Curve

The following graphs can all be referenced for further detail at-^[7]

⁷ "Desmos Graphs." *Desmos Graphing Calculator*. N.p., n.d. Web. 08 Sept. 2019 *All of the following graphs were made using the same resource.

<https://www.desmos.com/calculator/grydrmid5x>

For further understanding, the derivative curves were plotted. These however, are unnecessary for the investigation. They can be found at-

<https://www.desmos.com/calculator/fvxdsnbzex>

*Note: When visiting links, the graphs must be selected for them to be viewed. This was done to avoid the clutter of too many graphs on 1 plane.

4.5) Computation of Isoclines of Fractal Vertex Growth Conjectures

For each fractal type, we must find a way to observe pixel amount growth. Since the rates of growth are not constant, we can compare these rates using isoclines at a certain iteration number.

To do this, we must consider the derivative conjectures we observed earlier, and plug in a certain iteration number to each of them. This gives the certain slope of the conjecture we formed at a certain point. Since we are measuring the slope of all fractals at the same iteration, we can use this to compare growth patterns.

The derivatives deduced in the previous section were used to calculate the rate of growth of the number of vertices in the 1st, 2nd, 3rd, 4th and 5th fractal iteration, represented by an isocline at that point.

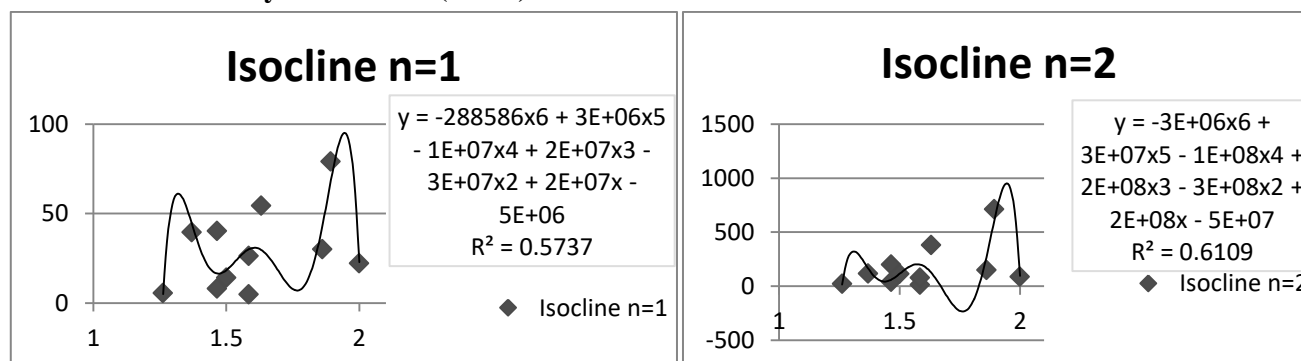
The values are given below.

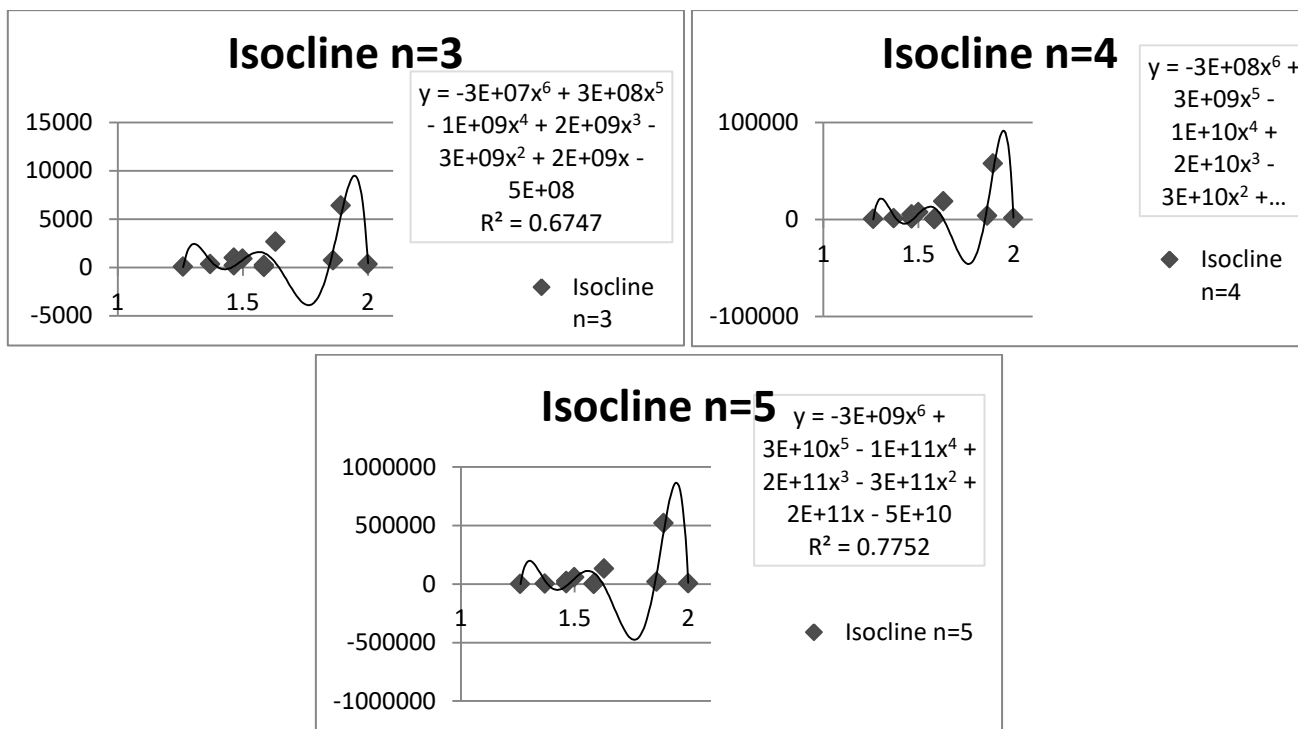
Fractal Name	Isocline: n=1	Isocline: n=2	Isocline: n=3	Isocline: n=4	Isocline: n=5
Von Koch Snowflake	5.545177444	22.18070978	88.72283911	354.8913564	1419.565426
Quadratic Von Koch Crossflake	39.55004239	118.6501272	355.9503815	1067.851145	3203.553434
Vicsek Fractal	40.23594781	201.1797391	1005.898695	5029.493476	25147.46738
Quadratic Von Koch Snowflake	8.047189562	40.23594781	201.1797391	1005.898695	5029.493476
Minkowski Sausage	14.25902771	114.0722217	912.5777737	7300.62219	58404.97752
Serpinski's Triangle	4.943755299	14.8312659	44.49379769	133.4813931	400.4441792
Boundary of a T-Square	26.36669493	79.10008478	237.3002544	711.9007631	2135.702289
Hexaflake	54.48548417	381.3983892	2669.788725	18688.52107	130819.6475
Pentaflake	30.17696086	150.8848043	754.4240215	3772.120107	18860.60054
Serpinski's Carpet	79.10008478	711.9007631	6407.106868	57663.96181	518975.6563
Hilbert's Curve	22.18070978	88.72283911	354.8913564	1419.565426	5678.261703

5.0) Trends Observed- Putting it together

5.1) Testing the Research Question

Now that all the data has been obtained, we can plot a graph of isoclines at n= 1, 2, 3, 4 and 5 (y axis) versus the hausdorff dimensionality of a fractal. (x axis)





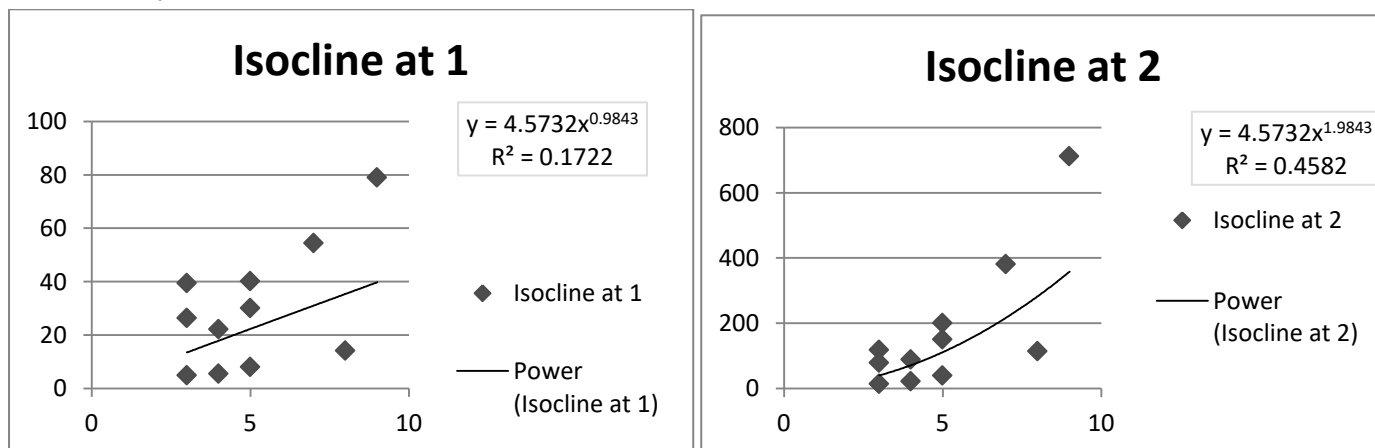
Analysis- The series of graphs shows a very weak trend. Frankly, the trend chosen was a polynomial of higher order because it provided the most flexibility in graphing and hence the highest R² value, but the nature of the graph makes it very clear that there seems to be no relationship between the hausdorff dimensionality and fractal vertex growth. While one can operationalize fractals with a measure of dimensionality, this does not guarantee, in any way, that this measure will relate directly to a growth trend which seems inherent to a fractal. This however, again, does not mean that there is no trend at all. Upon further observation, parts of the definition of dimensionality- the 'capacity' scaling factor discussed, seem to determine the fractal growth patterns.

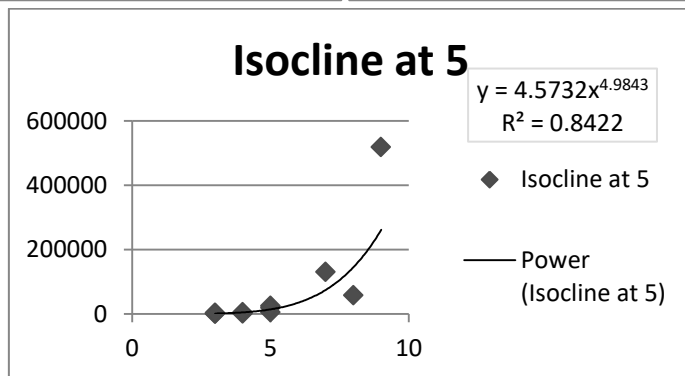
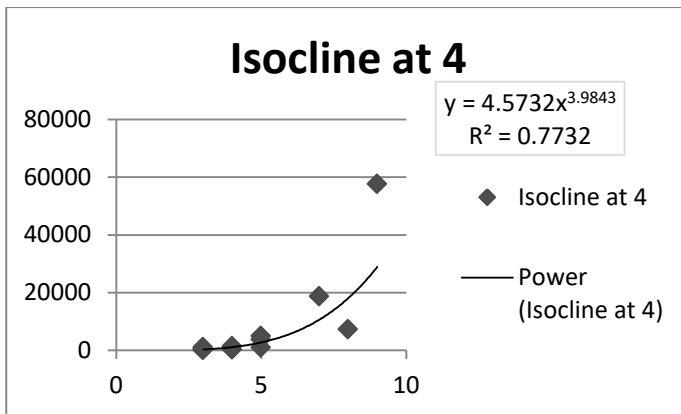
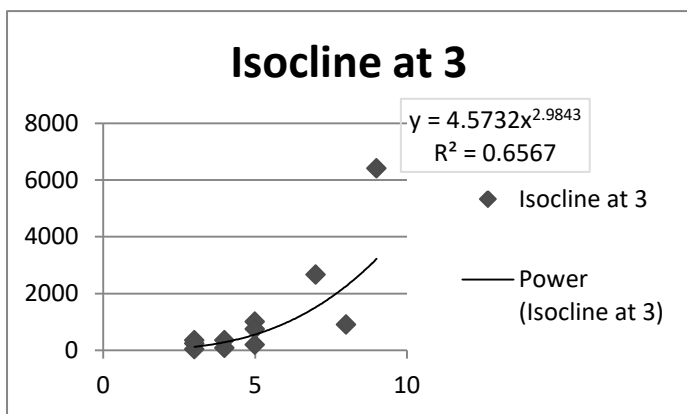
5.2)A newly apparent trend

The new trend references how pixel growth is affected by the number of smaller iterations that make the next fractal pattern.

The following allows for one to observe a trend in the rate of growth of pixels, defined by an isocline, versus the iteration number observed.

The following graphs plot the number of smaller versions of a fractal that form the next iteration or the capacity scaling factor discussed when defining the dimensionality of the fractals on the x axis, and the isoclines at n=1, 2, 3, 4 and 5 on the y axis.

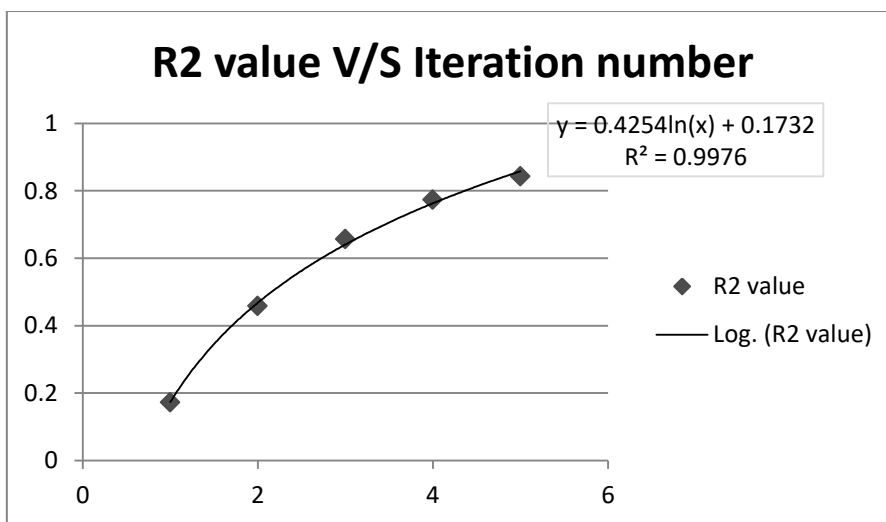




Analysis- In this case, one can, in fact, observe a trend. As the isoclines iteration number increases, the R2 value seems to increase. This implies that at higher iterations there is a trend present. This is possibly as the number of smaller fractals is multiplied by the iteration number in the conjecture formed. This means that at higher iteration numbers, this term will dominate and one will observe a trend for the growth.

For example, the serpinski's carpet consists of 9 smaller versions of itself, and hence, is defined by a term of 9^x in the conjecture $4(9^x)$. At higher iterations, this term becomes larger and impacts the pixel value more than any other terms, which is why it would fit into a power graph.

One must keep in mind, that as iterations increase, one loses area. Hence, running a larger amount of iterations leads to empty and unconsidered spaces in the fractal. One must be able to find a sweet spot to allow for resolution and not the loss of too much area.



The successively increasing R^2 values indicate a trend at a large iteration number. At $n = \infty$, one can obtain a R^2 value of 1, indicating how the power nature of the relationship between the number of smaller fractals that make the next iteration and the growth of the vertices in successive iterations of said fractal.

6.0)Math- Taylor Approximations

Using Taylor Polynomials to approximate the conjectures formed.

Mathematicians often substitute functions that are harder to work with polynomials, for the simple reason that polynomial functions, for most intents and purposes are easier to work with.

The simple animation of any function being replaced by an imposter polynomial is quite satisfying.

One can achieve this by matching various elements of information from the function to the Taylor polynomial. Let's consider something simple.

$f(x) = 1 + 5^x$ around a certain area – let's consider $x = 0$. This is the conjecture form of the Quadratic Von Koch Curve's pixel growth pattern.

One can develop a mirrored Taylor polynomial around this point of interest.

One can limit to a certain amount of terms, before attempting to attain a perfect sense of an infinite series, making understanding easier.

One can consider 5 for now- making the approximation fairly accurate.

Our function, would be defined as an approximation- in polynomial form.

$$P(x) = C_0 + C_1x + C_2x^2 + C_3x^3 + C_4x^4$$

Where the C's are constants one must obtain to fit this.

One can start by matching C_0

C_0 controls the y intercept, or value of the function at our point of interest.

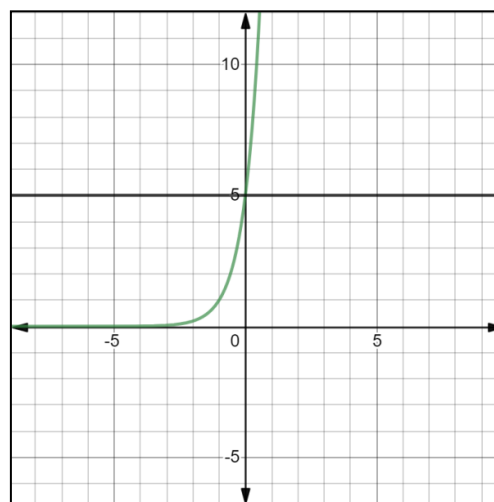
$$\begin{aligned} f(x) &= 1 + 5^x \\ f(0) &= 2 \end{aligned}$$

Hence, we match this in C_0 , as 2.

$$C_0 = 2$$

At this point, the functions, when compared, look like this-

The green function being $f(x)$ and the black one being the Taylor in $P(x)$ [Defined by C_0 only]



At this point $P(x) = 2$

Next, one can consider C_1

One needs information about this point at $x = 0$. Already having considered the value, one can now consider the slope.

This is given by the derivative.

$$\begin{aligned} f'(x) &= (\ln 5)5^x \\ f'(0) &= \ln 5 \end{aligned}$$

Next, one must match this value as the first derivative.

The first derivative of $p(x)$ is given by

$$P'(x) = C_1 + 2C_2x + 3C_3x^2 + 4C_4x^3$$

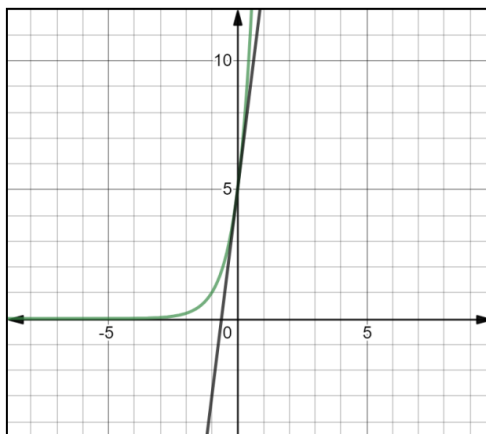
Hence, plugging in $x=0$ gives the value $=C_1$

$$P'(0) = C_1$$

Matching the $\ln 5$ from $f(x)$ means that C_1 must equal $\ln 5$.

$$C_1 = \ln 5$$

Now, our approximation gets better. Here is a visual of how accurate the Taylor is now. Better than before, but, one can still go further.



At this point,

$$P(x) = 2 + \ln 5x$$

More information can be determined from $x = 0$. Namely, the concavity given by the second derivatives. This can help find a value for C_2

$$f''(x) = (\ln 5)^2 5^x$$

$$f''(0) = (\ln 5)^2$$

Again, one must mirror this in the second derivative of the polynomial

$$P''(x) = 2C_2 + 6C_3x + 12C_4x^2$$

At $x=0$, this equals $2C_2$.

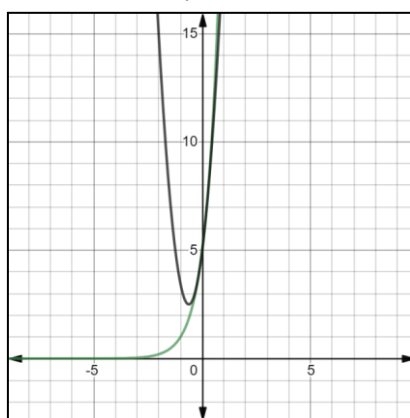
$$P''(0) = 2C_2$$

To obtain a value for C_2 , we must equate these second derivatives.

$$2C_2 = (\ln 5)^2$$

$$C_2 = \frac{(\ln 5)^2}{2}$$

At this point, our approximation holds 3 terms, defined by constants C_0 , C_1 and C_2 . It seems to get fairly accurate.



Now,

$$P(x) = 2 + \ln 5x + \frac{(\ln 5)^2}{2} x^2$$

We can extend this to match the next few terms and find constants in similar ways.

Finding C_3 ,

$$f'''(0) = (\ln 5)^3$$

Matching in the polynomial

$$P'''(x) = 6C_3 + 24C_4x$$

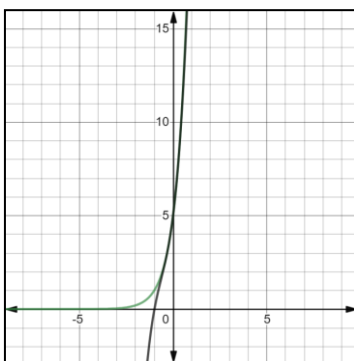
$$P'''(0) = 6c_3$$

Hence, matching the values of the 3rd derivative,

$$6C_3 = (\ln 5)^3$$

$$C_3 = \frac{(\ln 5)^3}{6}$$

Now, the function is very accurate around the point of interest $x=0$



The approximation improves, defining

$$P(x) = 2 + \ln 5x + \frac{(\ln 5)^2}{2}x^2 + \frac{(\ln 5)^3}{6}x^3$$

Finding C_4 .

$$f''''(0) = (\ln 5)^4$$

Matching in the polynomial

$$P''''(x) = 24C_4$$

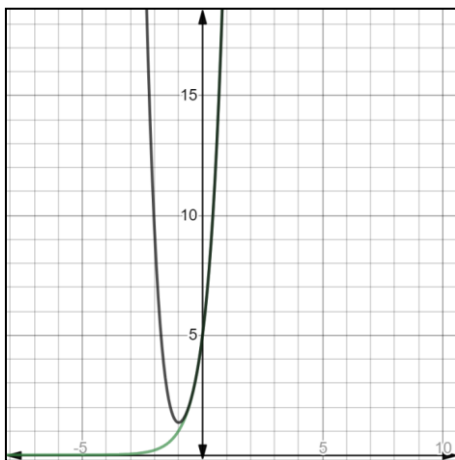
$$P''''(0) = 24C_4$$

Hence, matching the 4th derivative

$$24c_4 = (\ln 5)^4$$

$$C_4 = \frac{(\ln 5)^4}{24}$$

Finally, our Polynomial has 5 terms, leaving it a fairly accurate substitution.



Finally,

$$P(x) = 2 + \ln 5x + \frac{(\ln 5)^2}{2}x^2 + \frac{(\ln 5)^3}{6}x^3 + \frac{(\ln 5)^4}{24}x^4$$

Leaving us with all our constants for the polynomial-

$$C_0 = 2$$

$$C_1 = \ln 5$$

$$C_2 = \frac{(\ln 5)^2}{2}$$

$$C_3 = \frac{(\ln 5)^3}{6}$$

$$C_4 = \frac{(\ln 5)^4}{24}$$

The stitched imposter polynomial made to replicate $f(x)$ around 0 being

$$P(x) = C_0 + C_1x + C_2x^2 + C_3x^3 + C_4x^4$$

$$P(x) = 2 + \ln 5x + \frac{(\ln 5)^2}{2}x^2 + \frac{(\ln 5)^3}{6}x^3 + \frac{(\ln 5)^4}{24}x^4$$

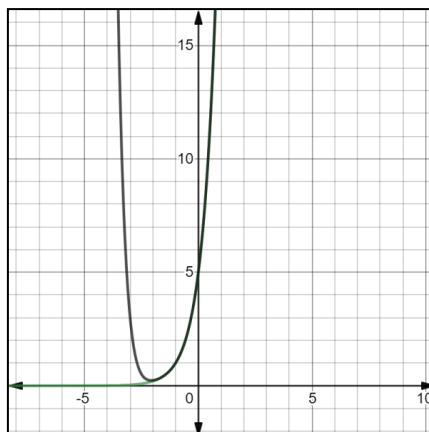
One can observe, that as terms are added, the series becomes a better and better approximation.

Hence, an infinite series is considered a limit of a perfect imposter polynomial function, stitched to replicate any other.

$$P(x) = \lim_{n \rightarrow \infty} \left[2 + \ln 5x + \frac{(\ln 5)^2}{2}x^2 + \frac{(\ln 5)^3}{6}x^3 + \frac{(\ln 5)^4}{24}x^4 + \dots + \frac{(\ln 5)^n}{n!}x^n \right]$$

This limit represents a perfect approximation of the polynomial $f(x) = 1 + 5^x$

Here is the same polynomial, extended to 10 terms.



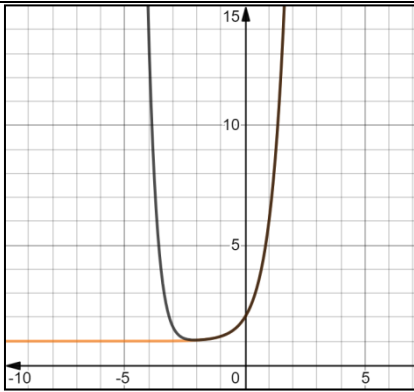
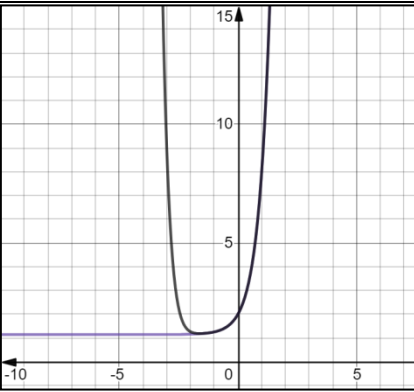
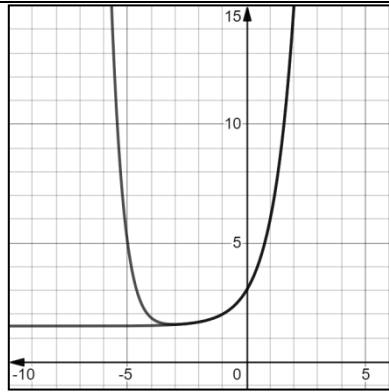
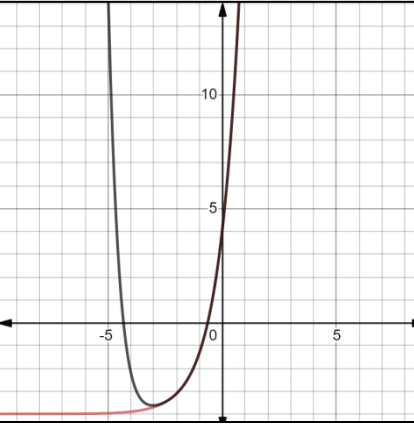
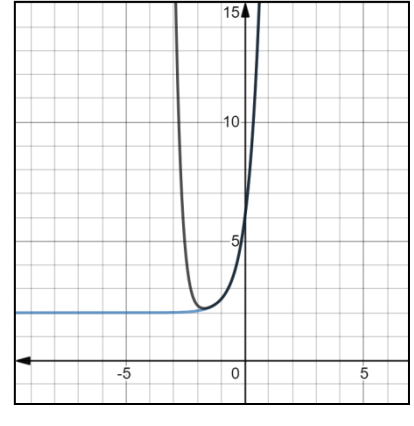
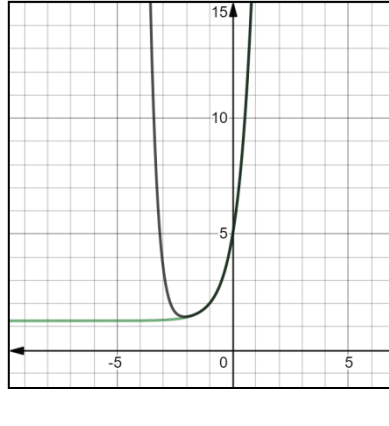
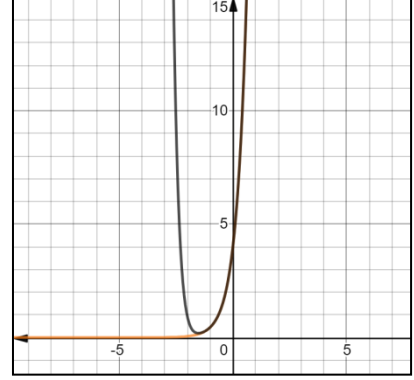
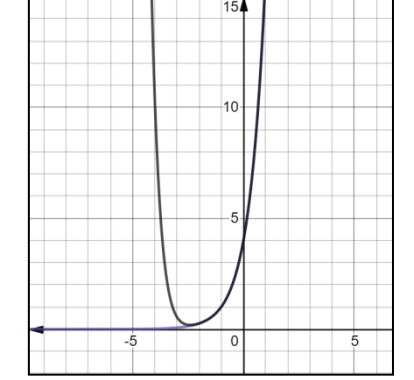
We can form a general form of this equation, where $P(x)$ is an approximation of $f(x)$ around $x=a$

$$P(x) = f(a) + \frac{df(a)}{dx} \frac{x-a}{1!} + \frac{d^2f(a)}{dx^2} \frac{(x-a)^2}{2!} + \frac{d^3f(a)}{dx^3} \frac{(x-a)^3}{3!} + \dots, \quad -\infty < x < \infty$$

Hence, one can apply this to each fractal type and form conjectures for polynomials.

Here are polynomials which substitute for the conjectures formed- written to the 10th iteration. The conjectures which each of the polynomials substitute for are provided below the graphs.

<p>Von Koch Snowflake-$1+4^x$</p>	<p>Quadratic Von Koch Crossflake-$12(3^x)$</p>	<p>Vicsek Fractal-5^{x+1}</p>

		
<p>Quadratic Von Koch Snowflake-$1+5^x$</p>	<p>Minkowski Sausage-$2+(6(8^x-1))/7$</p>	<p>Serpinski's Triangle-$3+(3(3^x-1))/2$</p>
		
<p>Boundary of a T-Square-$(8)3^x-4$</p>	<p>Hexaflake-$2+(4)7^x$</p>	<p>Pentaflake-$2+(3(5^{x+1}-1))/4$</p>
		<p>*NOTE: The displayed Taylor polynomials are expressed until 10 terms. The coloured graphs are the conjectures formed for each fractal, with the black graph being the corresponding Taylor polynomial.</p>
<p>Serpinski's Carpet-$(4)9^x$</p>	<p>Hilbert's Curve-4^{x+1}</p>	

These graphs can be referenced with animations of how the Taylor polynomial gets better with successive iterations at <https://www.desmos.com/calculator/qhhzz6xya4>

Note- One must highlight corresponding graphs. This cannot be done automatically as there are too many graphs- so one must select the graph and its corresponding polynomial (Both under the same fractal type) and highlight this to observe the animation, which runs from the 0th to the 10th iteration.

7.0) Evaluation and Conclusion

7.1) Conclusion

This internal assessment has explored fractal growth and hausdorff dimensionality. Although no trend was initially found, a second trend became apparent, with the starting vertices of the fractal. This power trend grows stronger as more iterations of the fractal are observed, due to the fact that the power term in the growth of number of vertex conjecture begins to get very large and dominate at higher iterations.

Fractals as a concept fall within the realm of recreational math, and are rarely seen applied to the real world. I found this intriguing, as one can see fractals everywhere in nature, and hence, one should be able to apply fractals in multiple

situations. Thus, this IA focuses on the application of fractals, so as to make this interesting concept more useful with modern pixel resolution.

The internal assessment is a representation of how mirroring patterns primarily noticed in nature can usually lead to beneficial applications, which can be optimized rather well. Fractals for example, due to their nature, are computationally less demanding than other code (as they follow power laws). This adds a lot of significance to the investigation, allowing for many applications.

While in theory, the application does work, it comes with its strengths and limitations, discussed below.

7.2) Strengths

1. Fractal conjectures follow power laws, and hence, are easy to code- This makes it so that not much computational power is necessary. Hence, fractals can be used as pixel resolution programs, where the vertices of the fractal correspond to the pixels of an image (as suggested) and hence aim to aid image resolution.
2. The exploration is not limited to 2 dimensional images- If fractals which operate in more than 2 dimensions (like Serpinski's tetrahedron) are chosen, the same concept can be explored in 3 dimensions or more, to help with holographic resolution.

7.3) Limitations

1. Fractals usually tend to grow with repetitive holes, or do not grow in traditional image shapes (which tend to be squares or rectangles. This means that in their application, one can find that while certain parts of an image have gained resolution, others have not. This limits their potential usefulness in image resolution programs.

Very few fractals, like serpinski's carpet or Hilbert's curve, can potentially fit images as they grow in a square or a rectangular shape.

2. Modern technology is excellent at image resolution. Fractals will find it hard to compete with the already existing technology, which might require more computing power, but can definitely perform a similar task faster and perhaps more efficiently.

7.4) Implications and Applications

Fractals tend to be easy to code, as they follow power laws, whose coding involved the repeated multiplication of a specific constant. This makes it so that they do not use up too much computing power, and make them perfect for image resolution programs where computing power is scarce. Though their application is situational, it is completely possible and useful.

1. One could consider a missile system, which applies the fractal iteration and hence zooms in on its target while in mid air. The missile would not require a large computational load and would gain accuracy as time passed. As an example, if the missile ran serpinski's triangle iterations, it would lock once the target was in one of the 3 sub parts of the triangle and zoom, and continue to repeat this process until it lands accurately.
2. Another application of the concept is in nanotechnology, where this computing power is scarce. Here, it may be useful in regular nanocamera technology for that very reason. The lack of computing power may make it hard to use many resolution softwares, opening up opportunities for fractals to be applied.
3. The fractal concepts explored can be used for hologram resolution in 3 dimensions, if fractals that expand in 3 dimensions are chosen. Since this area of resolution is less explored, perhaps fractals like Serpinski's tetrahedron can be useful in generating resolution in holograms.

7.5) Proposed Extensions

1. It would be interesting, to explore the concept of space filling curves, in the context of image resolution, and compare the efficiency of its application in image resolution to the fractal application in this internal assessment.
2. One may find it beneficial to explore fractals and graph theory, exploring a trend in how algorithms like Kruskal's minimum spanning tree's or Djikstra's algorithm's total weight change as fractals gain vertices in successive iterations.

8.0 Bibliography

"Desmos Graphs." *Desmos Graphing Calculator*. N.p., n.d. Web. 08 Sept. 2019.

"Fractal Lab." *Fractal Lab - Interactive WebGL Fractal Explorer*. N.p., n.d. Web. 08 Sept. 2019.

"Hausdorff Dimension." *Wikipedia*. Wikimedia Foundation, 08 Sept. 2019. Web. 08 Sept. 2019.

Hilbert's Curve: Is Infinite Math Useful? Dir. 3Blue1Brown. *YouTube*. YouTube, 21 July 2017. Web. 08 Sept. 2019.

Kohavi, Yuval, and Hadar Davdovich. *Topological Dimensions, Hausdorff Dimensions & Fractals*. Publication. N.p., 2009. Web. 8 Sept. 2019.

"List of Fractals by Hausdorff Dimension." *Wikipedia*. Wikimedia Foundation, 06 Sept. 2019. Web. 08 Sept. 2019.

"Online Image Resizer - Crop, Resize & Compress Images, Photos and Pictures for FREE." *Online Image Resizer - Crop, Resize & Compress Images, Photos and Pictures for FREE*. N.p., n.d. Web. 08 Sept. 2019.

"MS Paint." Web 08 Sept 2019.

Aditya Advani