Course: CS30A1570 Complex Systems

Assignment 3: Fractals

Completion Date: 22 March 2024

Submitted by: Turzo Ahsan Sami

Study on Fractals and Fractal Dimensions (Koch Curve, Cantor Set, Sierpinski Carpet)

Introduction

Fractals offer a fascinating area of study in mathematics and computer science. Fractal patterns are intricate and possess self-similarity across scales. This report presents a thorough study on fractals, focusing on the Koch Curve, Cantor Set, and a Sierpinski Carpet. The study explores the box-counting method to approximate the Hausdorff dimension of the Koch Curve and the Cantor Set fractals. This report also investigates the impact of varying parameters on the accuracy of this approximation. The report also discusses the implementation of a Python and NetLogo model for generating and iterating a Sierpinski Carpet fractal.

Research Question:

The primary research question guiding this study is: What are the optimal parameter settings for box counting, including initial box length and increment, to obtain the most accurate approximation of the Hausdorff dimension for the Koch curve, Cantor Set, and Sierpinski Carpet fractals? Additionally, how does increasing the iteration levels of the fractals affect the accuracy of the Hausdorff dimension approximation?

Result:

The study found that the box-counting dimension closely approximates the Hausdorff dimension for the Koch Curve, Cantor Set, and Sierpinski Carpet under certain parameter settings. However, increasing the iteration levels of the fractals seems to decrease the accuracy of the Hausdorff dimension approximation. The study also successfully implemented a NetLogo model to generate and iterate a novel Sierpinski Carpet fractal.

Task 1: Design your own fractal, different from the ones described in the videos and quizzes. Compute the fractal dimension of your fractal.

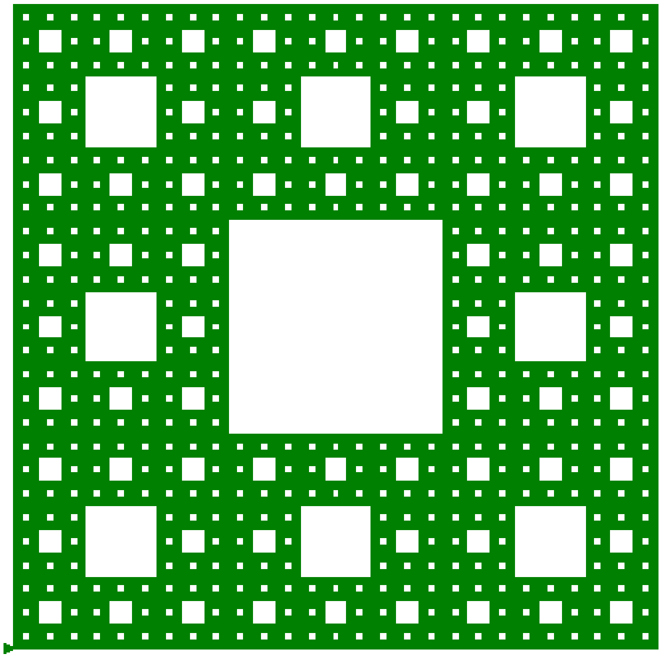
For this task, the Sierpinski Carpet Fractal is modelled in Python using the turtle graphics module (Attachment 1: sierpinski_carpet_fractal.py). The Sierpinski Carpet is a plane fractal first described by Wacław Sierpinski in 1916 (Reference 1). The Sierpinski Carpet is a fractal that results from subdividing a square into 9 equal parts and removing the central one, then recursively repeating the process for the remaining parts (Reference 2).

The fractal dimension of the Sierpinski Carpet can be calculated using the formula: *D = log(N)/log(S) where, N is the number of self-similar pieces, and S is the scale factor.*

For a Sierpinski Carpet having N = 8 (each square is divided into eight new squares) and S = 3 (each new square is 1/3 the length of the original), the fractal dimension D = log(8) / log(3) = 1.89 (Reference 3).

This means that the Sierpinski Carpet is roughly 1.89-dimensional. It's more than a 1-dimensional line but less than a 2-dimensional shape. This is a characteristic of fractals - they can have non-integer dimensions.

Image 1: Sierpinski Carpet Fractal (Python)



Study question: What are the optimal parameter settings for box counting, including initial box length and increment, to obtain the most accurate approximation of the Hausdorff dimension for the Koch curve and Cantor Set fractals? Additionally, how does increasing the iteration levels of the fractals affect the accuracy of the Hausdorff dimension approximation? Finally, which settings yield the closest approximation to the published fractal dimension of the coastline of Great Britain using box counting applied to the coastline image?

After running the Netlogo model (BoxCountingDimension_v_6.1.1) as per the study instructions, the data obtained are presented in Table 1. The optimal parameter settings to obtain the most accurate approximation of the Hausdorff dimension and the Box-counting Diemsnions for the Koch curve and Cantor Set fractals can be inferred from the experiment results.

For the Koch Curve, the box-counting dimension is closest to the Hausdorff dimension at an iteration level of 4 with an initial box length of 5. The box-counting dimension here is 1.252 (Appendix- Image 1), which is very close to the Hausdorff dimension of 1.262. As the initial box length increases to 10 and 20, the box-counting dimension decreases, indicating that a smaller initial box length may yield a more accurate approximation.

For the Cantor Set, the box-counting dimension is closest to the Hausdorff dimension at an iteration level of 4 with an initial box length of 10. The box-counting dimension here is 0.636 (Appendix- Image 2), which is very close to the Hausdorff dimension of 0.631. As the initial box length increases to 20, the box-counting dimension decreases, similar to the Koch Curve.

Increasing the iteration levels from 4 to 6 seems to decrease the accuracy of the Hausdorff dimension approximation for both fractals. This could be due to the increased complexity of the fractals at higher iteration levels, which may make the box-counting method less accurate.

Table 1: Hausdorff Dimension and Box-counting Dimension of the Koch Curve and the Cantor Set.

Model	Iteration	Initial Box Length	Hausdorff Dimension	Box-counting Dimension
Koch Curve	4	5	1.262	1.252
		10	1.262	1.249
		20	1.262	1.243
	6	5	1.262	1.256
		10	1.262	1.228
		20	1.262	1.205
Cantor	4	5	0.631	0.734
		10	0.631	0.636
		20	0.631	0.616
	6	5	0.631	0.616
		10	0.631	0.591
		20	0.631	0.929

The BoxCountingApplied_v_6.1.1.nlogo model was loaded with the coastline fractal and was run 10 times varying the initial box length (each running for 15 iterations). From the data obtained from the experiments (Table 2), it is evident that the initial box length has little to no-effect on the box-counting dimension as the dimension values are close for each experiment set (Appendix- Image 3). However, further study is required to verify this statement.

Table 2: Box-counting Dimension of the Great Britain Coastline fractal

Model	Initial Box Length	Box-counting Dimension
Box-counting Applied (Coastline)	1	1.396
	2	1.388
	3	1.363
	4	1.395
	5	1.379
	6	1.372
	7	1.384
	8	1.376

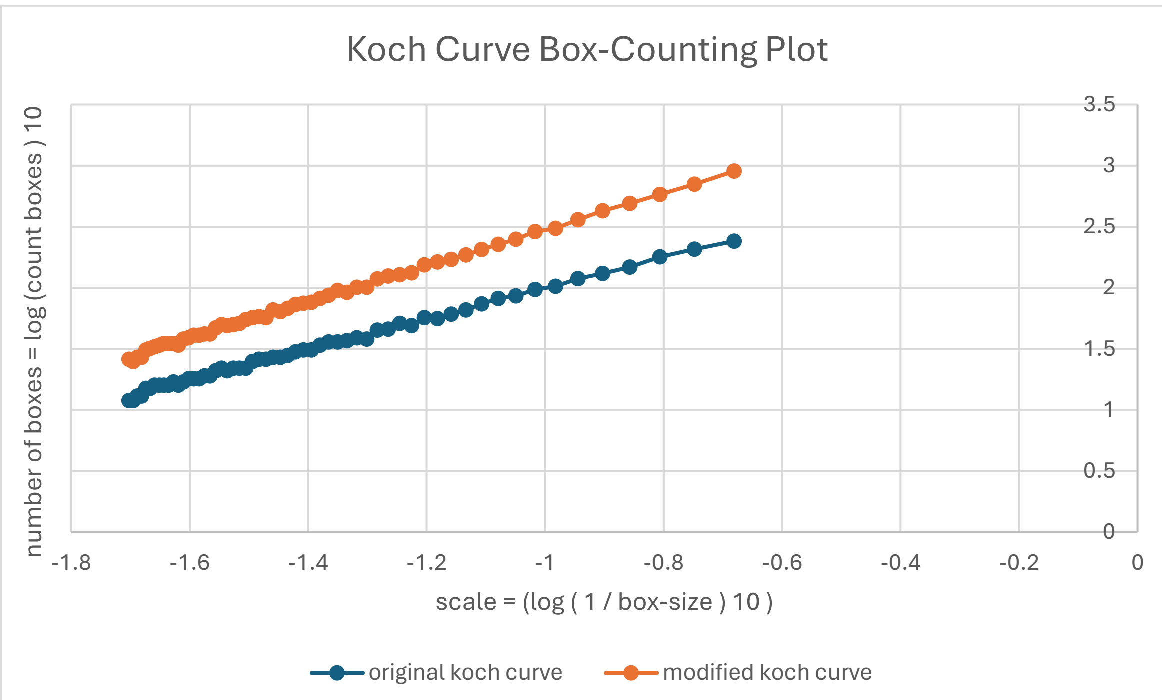
Task 2: How does the box-counting dimension of the variation on the Koch curve, where each segment is replaced by five segments of 1/3 the original length, compare with its Hausdorff dimension? Additionally, what modifications are necessary to implement this variation in the provided NetLogo code for box counting and how do the results differ from the original Koch curve?

The Box-Counting Dimension and the Hausdorff Dimension are both measures of fractal dimension. They provide a quantitative measure of the complexity of a fractal (Reference 5).

For the original Koch curve (Appendix: Image 4), each line segment is replaced by four segments, each of which is 1/3 the length of the original. This results in a box-counting dimension of log(4) / log(3) = 1.26. In each iteration, the length of the curve increases by a factor of 4/3, while the number of squares needed to cover it increases by a factor of 9 (3x3 grid). Because the Koch curve is a self-similar fractal, each part is a small-scale copy of the whole. That is why both box-counting dimension and Hausdorff dimension are equal and is roughly equals to 1.26. And, the Box-counting dimension is 1.307 (Appendix: Image 6).

If the Koch curve is modified in such a way that, so that each segment is replaced by five segments of 1/3 the original length, the fractal becomes more complex (Appendix: Image 5, Attachment 3 & 4). Theoretically, the box-counting and Hausdorff dimensions should remain the same because the scaling behavior of the fractal is still the same (self-similar). However, the results from the box-counting procedure in reality differs from the original Koch curve due to the increased complexity of the fractal. The Hausdorff dimension is increased to 1.631. And, the Box-counting dimension is 1.457 (Appendix: Image 7).

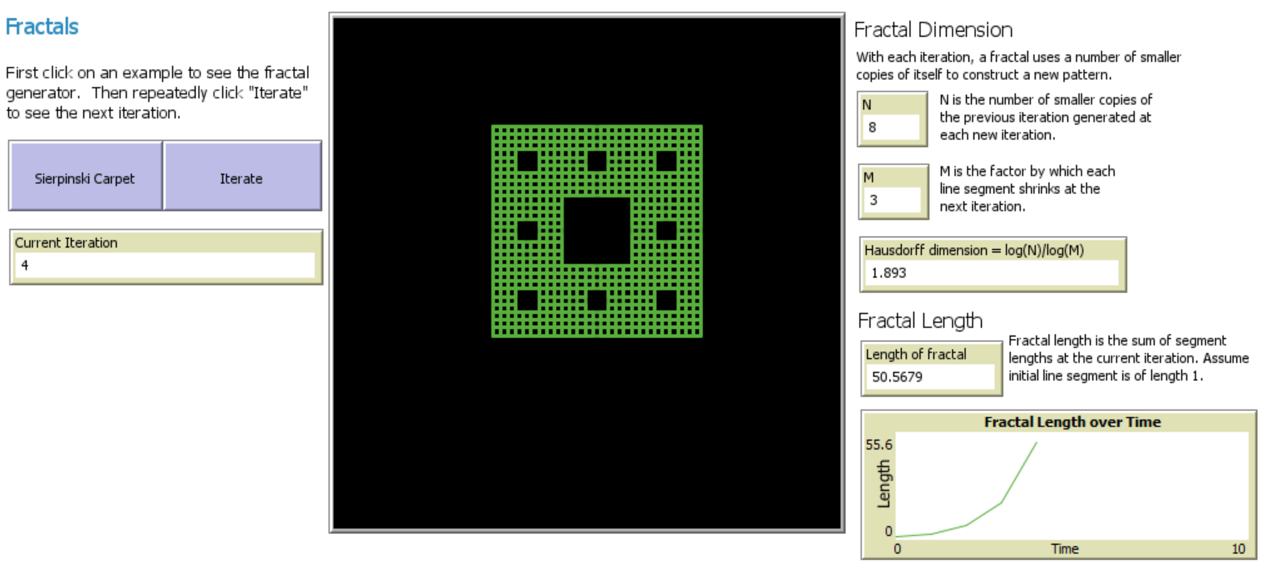
Image 2: Box-counting Plot of the Koch Curve (data obtained from Netlogo model run: Attachment 5 & 6)



Task 3: Implement a NetLogo model that allows users to generate, iterate, and draw a fractal that has not been implemented before.

For this task, the Sierpinski Carpet Fractal is modelled in Netlogo (Attachment 2: sierpinski_carpet_fractal.nlogo, Appendix: Image 8). In this program, the 'Sierpinski-Carpet' procedure sets the initial conditions for the fractal and starts the iteration process. The 'Iterate' procedure is responsible for the recursive creation of the fractal. It increases the current iteration, adjusts the length of the line segments, removes old turtles, and runs the memory command on the remaining turtles (Reference 4). The 'Sierpinski-Carpet-Command' procedure sets the angle and scale factor for the fractal and updates the memory with the command string for the fractal.

Image 8: Sierpinski Carpet Fractal (Netlogo)



Discussion:

The results highlight the importance of carefully selecting the initial box length and increment in the box-counting method to accurately approximate the Hausdorff dimension. The decrease in approximation accuracy with increased iteration levels suggests that the complexity of fractals at higher iteration levels may make the box-counting method less accurate. The successful implementation of the Sierpinski Carpet in NetLogo demonstrates the versatility of this platform in exploring novel fractals.

Conclusion:

This study contributes to the understanding of fractal dimensions and the application of the box-counting method in approximating the Hausdorff dimension. The findings underscore the need for further research to improve the accuracy of fractal dimension approximations, particularly at higher iteration levels. The study also opens avenues for further exploration of novel fractals using platforms like NetLogo.

References

1. https://www.geeksforgeeks.org/python-sierpinski-carpet/
2. https://fractalfoundation.org/OFC/OFC-2-1.html
3. https://mathworld.wolfram.com/SierpinskiCarpet.html
4. https://www.wolframalpha.com/input/?i=sierpi%C5%84ski+carpet
5. https://www.math.stonybrook.edu/~scott/Book331/Fractal_Dimension.html

Attachments

1. Sierpinski carpet fractal (python): https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS_Assignment_3/sierpinski_carpet_fractal.py
2. Sierpinski carpet fractal (netlogo): https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS_Assignment_3/sierpinski_carpet_fractal.nlogo
3. Koch curve (modified): https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS_Assignment_3/%5Bmodified%5D%20koch-curve-box-counting-dimension.nlogo
4. Koch curve Box-counting (modified): https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS_Assignment_3/%5Bmodified%5D%20koch-curve-box-counting-dimension.nlogo
5. https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS_Assignment_3/%5Boriginal%5D%20koch-curve-box-counting-dimension%20Box%20Counting%20Plot.csv
6. https://github.com/ahsan-sami-turzo/complex-system-code/blob/main/CS_Assignment_3/%5Bmodified%5D%20koch-curve-box-counting-dimension%20Box%20Counting%20Plot.csv

Appendices

Image 1: The Kock Curve Box-counting plot

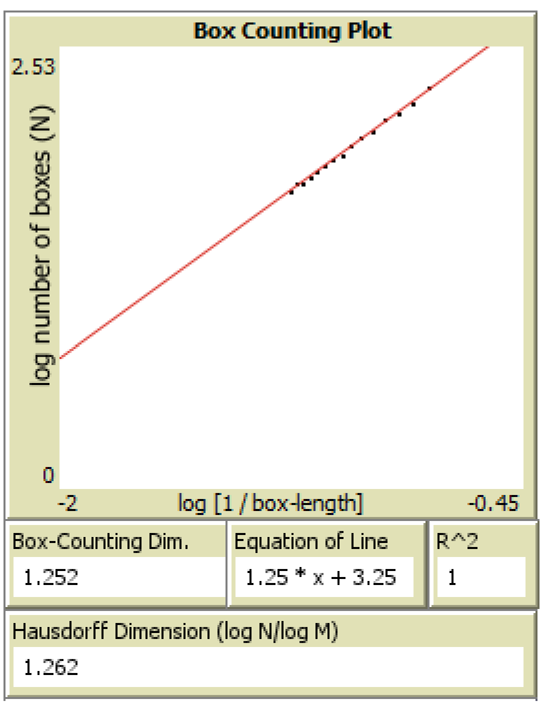


Image 2: The Cantor Set Box-counting plot

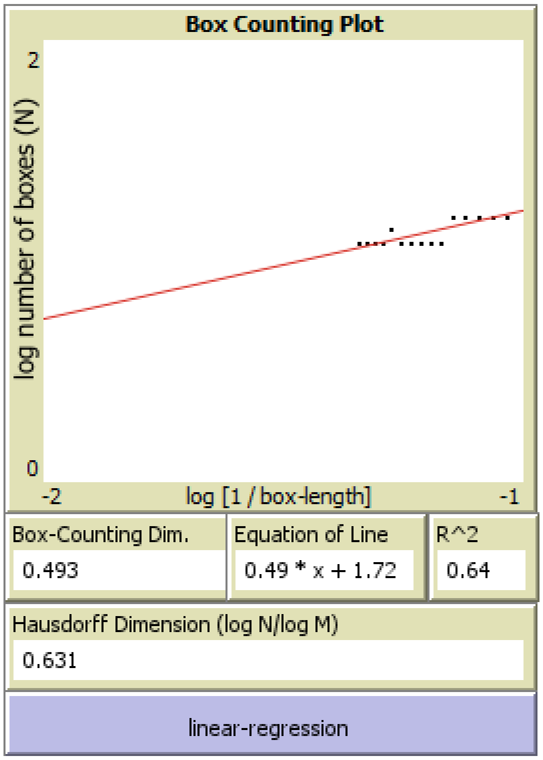


Image 3: Box counting plot for the Great Britain Coastline fractal

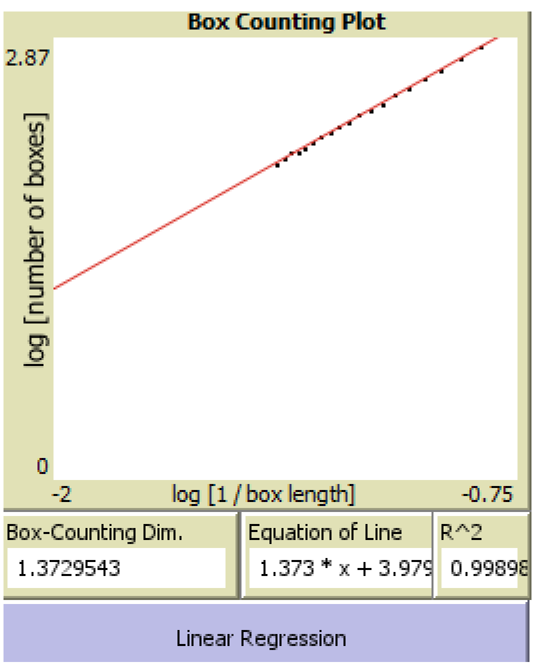


Image 4: Original Koch Curve

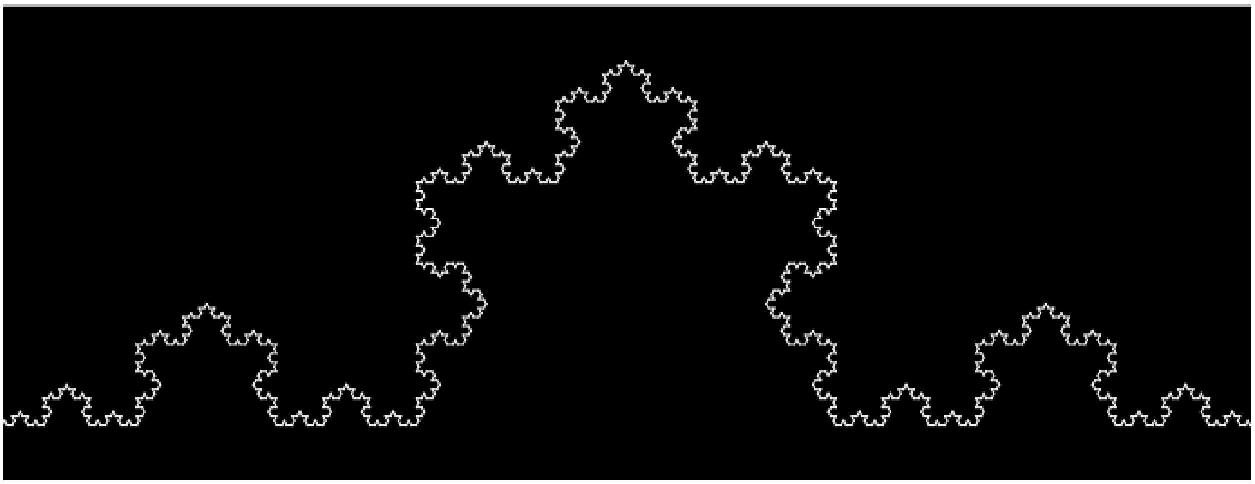


Image 5: Modified Koch curve, where each segment is replaced by five segments of 1/3 the original length

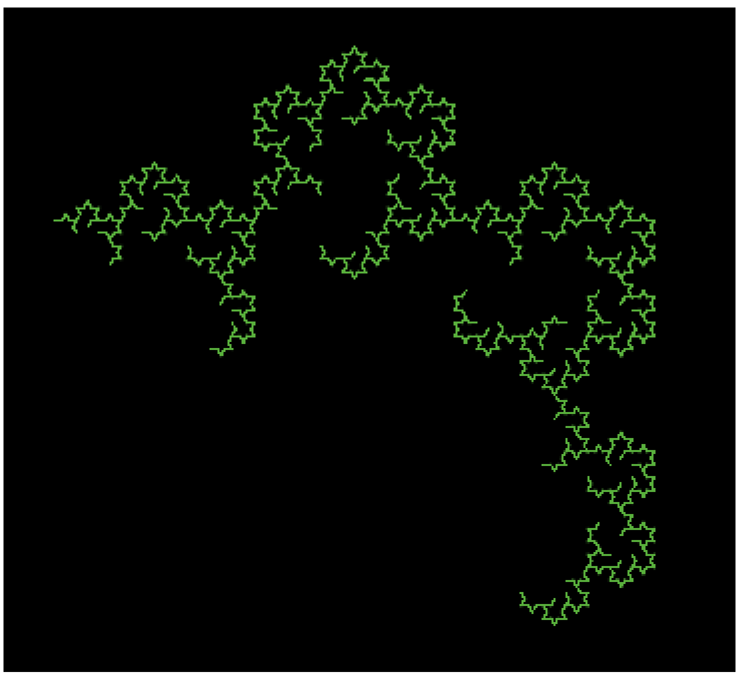


Image 6: Original Koch Curve Box Counting Plot

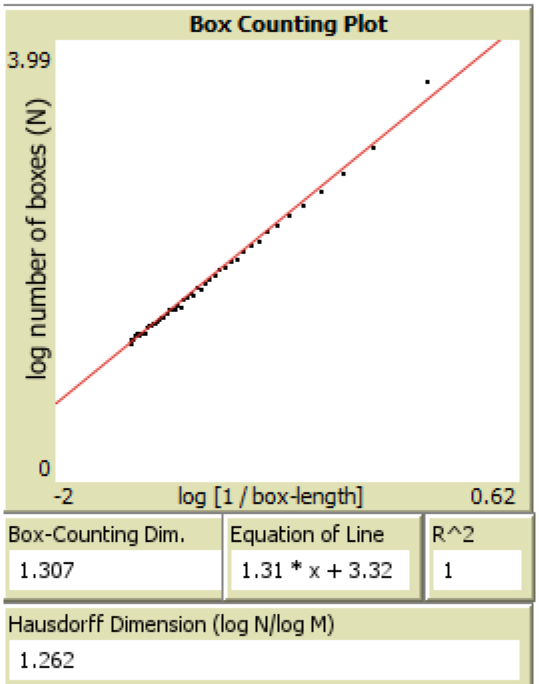


Image 7: Modified Koch Curve Box Counting Plot