Total Choas... Total Chosa

An Honors Thesis (Honors 499)

By

Christopher J. Butler

Dr. Kerry Jones

Ball State University

Muncie, Indiana

April 22, 2002

December 2002
Acknowledgments

Many thanks go to the mathematics department of Ball State from the fall of 1998 to the fall of 2002. Without your knowledge, wisdom, and concern for my education this project would never have happened. Special thanks go out to Dr. David Thomas for introducing me to the topic of this paper and Dr. Kerry Jones for not only advising me with this paper but also for teaching me discrete mathematics, knot theory, and my capstone course. Thank you.

I would also like to thank all of my friends at the Newman Center. You have truly been the other half of college. Thank you for putting up with all of my shoptalk and math jokes. Thank you too for putting up with my randomness and for always being there.
Abstract

This paper looks at the topic of fractal geometry and chaos theory. It explores the basics of both fractal geometry and chaos theory. The paper also focuses on the Mandelbrot set and the Julia set. This paper is aimed at high school juniors and seniors. After reading this paper they should be able to begin to understand chaos theory and fractal geometry. The paper includes a review of *The Julia Sets* and *Chaos: The Software*, the two computer programs that are used in the accompanying extended lesson plan. This lesson plan is to help students more fully develop the ideas and understandings of the topics addressed in the paper. By the end of this project the students should have a good feeling of how mathematics is used today and that it is not a dead science.
In the last thirty years, mathematics has made some very big advances. It has radically changed from being taught in the time of Euclid and Newton to the twenty-first century. One of the biggest reasons for this dramatic jump are the new mathematical areas of fractal geometry and chaos theory. Both areas have arisen because of the capabilities that computers allow. Fractal geometry better enables us, as humans, to describe and model nature. Chaos theory shows how seemingly random things, are in fact, quite mathematical.

Fractal geometry was given its name in the mid-1970's by Benoit Mandelbrot. Mandelbrot showed how every day objects in nature from ferns and snowflakes to mountain ranges and coastlines could be described by fractals. "A fractal is a geometric shape that has two special properties: 1. The object is self-similar. 2. The object has a fractal dimension." (Devaney 130) Self-similar means that any part of the design is similar to the whole design, topographically. If any part of the design were "cut" out of the picture all it would take to get back to the original picture would be to isotopically stretch, shrink, or rotate the piece that was cut out. After the cut out piece has been stretched or shrunk to the dimensions of the original design and rotated properly the cut out design will look exactly like the original. When computing fractal images on a computer it is very helpful to use an iterated function system.

Iteration, as defined by *Webster's New World College Dictionary Fourth Edition* is, "to do again or repeatedly" (760). All the computer does in an iterated function system is repeat the same map over and over again. Every map in an iterated function system is dependent on every other map. A map in an iterated function system is the design scaled by some degree of less than one, rotated, and then placed back in the design.
as part of it. Since every map is a shrinking and rotation of the original image, each map is self-similar to the whole image. An iterated function system, then, is simply when an initial value is put into a function repeatedly. Let us look at the function

\[ f(x) = \sqrt{4} \]

as an example. If we choose an initial value of 16 to put into \( f(x) \), we get a new value of 4. The next step in the iterated function system is to take our new value of 4 and put that into our function \( f(x) \). When we do this again we have a new value of 2. This may seem simple, but for this to be a true iterated function this process has to happen a minimum of about a thousand times so that the pattern of the function becomes apparent. This stipulation makes it obvious why computers are helpful when computing iterated function systems.

In its short existence, fractal geometry has been used in a wide variety of ways. Fractal geometry has been used in video games for graphics and in major motion pictures to help design planet surfaces and landscapes. There are three famous examples of fractal geometry that are also fairly simple that we will be looking at. Those are the Cantor set, the Sierpinski triangle, and the Koch snowflake.

One of the easiest fractals to study to get a better idea of fractals as a whole is the Cantor set where the middle third is removed from each iteration. The Cantor set simply starts out in its initial condition as a line. For this example, the line will be nine units long. The Cantor set involves taking an infinite number of removals. So, after the first step, the Cantor set is now composed of two lines, one from \( 0 \leq x \leq 3 \) and the other from \( 3 \leq x \leq 9 \). (The middle third of our line was \( 3 \leq x \leq 6 \).) In the next step we remove the middle third sections of our two lines that now comprise the Cantor set. This means that we now have four segments in the Cantor set with intervals from \( 0 \leq x \leq 1, 2 \leq x \leq 3, \ldots \).
$6 \leq x \leq 7,$ and $8 \leq x \leq 9$. To take this set one step further we end up with eight sections whose line intervals are from $0 \leq x \leq \frac{1}{3}$, $\frac{2}{3} \leq x \leq 1$, $2 \leq x \leq 2\frac{1}{3}$, $2\frac{2}{3} \leq x \leq 3$,

$6 \leq x \leq 6\frac{1}{3}$, $6\frac{2}{3} \leq x \leq 7$, $8 \leq x \leq 8\frac{1}{3}$, $8\frac{2}{3} \leq x \leq 9$ (Devaney 133).

At first glance it may appear that if this process were completely carried out that the entire line would disappear. This is not the case however, because upon closer inspection we see that the end points of the sections that we remove always remain as part of the line. In the end, then, our Cantor set is a set is a collection points, in this case

$0, \frac{1}{3}, \frac{2}{3}, 1, 2, 2\frac{1}{3}, 3, 6, 6\frac{1}{3}, 6\frac{2}{3}, 7, 8, 8\frac{1}{3}, 8\frac{2}{3}, 9$, because if there was an interval anywhere in our set we would have to remove the middle third of that interval. Not all the points of the Cantor set are end points. The point $\frac{1}{4}$ is also in the Cantor set. The reason for this can be seen if we use a base of three, rather than our usual base ten.

The Cantor set is self-similar. This can be seen by taking the interval of $0 \leq x \leq 3$ and multiplying it by a factor of three (Devaney 134). When this happens we get our original line of nine units long. If we take the interval from $0 \leq x \leq 1$ in the second step and multiply it by a factor of nine, we get our original line. We are in the second step so we have to multiply by a factor of $3^2$.

The Sierpinski triangle (figure 1) is also a fractal image that is made up of an infinite number of removals. To make a Sierpinski triangle, start with a darkened equilateral triangle. For the first step, remove the equilateral triangle that is formed from the three midpoints of the triangle. When this is done there should be four equal equilateral triangles, three that are darkened and one that is missing. The second step of the Sierpinski triangle is the same as the first. We simply remove the triangle that is
made by the midpoints of the three darkened triangles. So after the second step of the Sierpinski triangle there are nine darkened equilateral triangles (Devaney 130). Each step of the Sierpinski triangle increases the number of equilateral triangles by a power of three. Therefore after the first iteration we have three equilateral triangles, after the second—9, and the third—27.

The Sierpinski triangle is also self-similar. If we multiply anyone of the first three equilateral triangles by a factor of three, we end up with the dimensions of our original equilateral triangle. The same goes for any one of the nine triangles in the second iteration provided its multiplied by a factor of $3^2$ because it's the second iteration.

The Koch snowflake is slightly different from the two previous fractal images. Instead of being an infinite number of removals, the Koch snowflake is an infinite number of additions. Just like the Sierpinski triangle, the Koch snowflake starts with an equilateral triangle. The first part of the Koch snowflake involves removing the middle third of each side of the triangle, just like in the Cantor set. After the middle third is removed we then close the gap by adding two pieces that are a third of the total length to each side of the triangle. After our first iteration our equilateral triangle should now look like the Star of David. For the second iteration of the Koch snowflake we remove the middle third of all six sides of our figure and then fill in the gaps just like in the first iteration. Our new figure will now have twelve sides each one-ninth the size of our original figure (Devaney 137).

The Koch snowflake self-similarity can be seen when comparing the sides of the snowflake. If we look at a third of a side of the triangle and multiply it by a factor of three we get the entire image of the triangle. Since each side is the same, and the same
process can be done on each side, then the entire figure is self-similar (Devaney 138).

The Koch snowflake has another very interesting feature about it; it has a finite area but an infinite perimeter. To be able to find these we first need to find out the general formula for the number of sides in each iteration of the Koch snowflake

\[
\begin{align*}
N_0 &= 3 = 3 \times 4^0 \\
N_1 &= 12 = 3 \times 4^1 \\
N_2 &= 48 = 3 \times 4^2 \\
N_3 &= 192 = 3 \times 4^3
\end{align*}
\]

(Devaney 138).

These first four iterations develop the pattern that each previous iteration is multiplied by a factor of four, which leads to the general equation of

\[
N_z = 4N_{z-1} = 4^z \times 3
\]

(Devaney 138).

Now that we have our number of sides for each iteration, we now need to find the length for each side. For our initial length of our triangle we will use a length of one. Every iteration of the Koch snowflake reduces the sides of the figure by one third. So, if each iteration is a reduction of the previous side by one third then our general formula is

\[
L_z = \frac{1}{3^z}
\]

(Devaney 139).

Now by combining these two formulas, we can find the perimeter \( P_z \) of the Koch snowflake for the \( z \)th iteration. Since \( P_z = N_z \times L_z \) we have the formula

\[
P_z = N_z \times L_z = (4^z \times 3) \times \left(\frac{1}{3^z}\right) = \left(\frac{4}{3}\right)^z \times 3
\]

(Devaney 139).
Each iteration of the Koch snowflake increases the perimeter by a factor of \( \frac{4}{3} \) so the perimeter is always increasing. This means that if we were to start walking around the Koch snowflake, we would never stop.

The area of the Koch snowflake is more difficult to compute but its area is less than \( \frac{4}{3} \) because the Koch snowflake will fit into a square whose sides are \( 2\sqrt{\frac{3}{3}} \). To begin to help to show this, the first four areas of the Koch snowflake are given.

\[
A_0 = \frac{1}{2} \left( \frac{1}{2} \right) \left( \sqrt{3} \right)
\]
\[
A_1 = 3 \left( \frac{1}{2} \right) \left( 0.3333 \times 0.2887 \right)
\]
\[
A_2 = 12 \left( \frac{1}{2} \right) \left( 0.1111 \times 0.0962 \right)
\]
\[
A_3 = 48 \left( \frac{1}{2} \right) \left( 0.0370 \times 0.0321 \right)
\]

Each additional iteration gives the total area of the new triangles that are added onto the Koch snowflake. These four iterations give us an area of .6699, just over half of the total area.

To be able to look at interesting iterated systems we do not have to look any further than the quadratic family. It may seem like we know quadratic functions fairly well but what happens when we iterate the function 1,000 times. Our result may be something that we have never seen before. The really interesting designs found in the quadratic family are a group that is called the Julia set. The Julia set's definition is the boundary of the set of points of a polynomial that escapes to infinity. To be able to understand this definition better, lets talk about the Julia set a little more.
When we iterate the function $V_L = z^2$ it is important to know that $z$ equals the complex number $x + yi$. If we iterate $V_L$ when $z < 1$ we find that our value tends to drift toward 0. Likewise, when we iterate $V_L$ when $z > 1$ we find that our value goes out to infinity. This only leaves the case then of when $z = 1$, which is the Julia set for this function. The Julia set is one because as soon as we put in a value that is greater than one our end result will always be infinity. If we try to compute this we run into a small problem. It will seem that all points either go towards zero or infinity. The reason for this is because of the small round-off error that happens when our computers are working. This very minute amount may not seem to matter, but as the error is placed back into the equation 1,000 times, the error becomes large enough to send our value to either zero or infinity.

We can compute the first few values of $V_L$ by hand, or with the help of a graphing calculator, or a simple mathematics program on a computer. A good first example to start with is the function $W_L = z^2 + c$ where $c$ equals another complex number $x + yi$. This is much more interesting because the Julia set changes for every $c$ value. Even though the Julia set is different for each $c$ value any value outside of the interval $-2 \leq z^2 + c < 2$, on the real axis it will escape towards infinity. This brings up an important point. The Julia sets are graphed on the complex plane. This means that the real numbers are on the horizontal axis and the complex numbers are on the vertical axis.

Because the Julia sets are functions, that means that each Julia set has fixed points just like all other functions. Because they are iterated though these fixed points, they have a few new qualities to them. The fixed points can either be attracting fixed points or repelling fixed points. Around each fixed point is what is called a boundary of basin.
This is a small area around the fixed point that does one of two things, it either draws the value into the fixed point and locks it in there, or it repels the value away from the fixed point keeping it from ever reaching the fixed point unless it hits it directly. Another new aspect that our fixed points have is that they can cause the function to be periodic. What this means is that our instead of our function sticking on one fixed point it could possibly end up jumping backing and forth between two, three, or more fixed points in a never ending periodic cycle. A Julia set with period two would mean that it alternates between two fixed points.

Julia sets are fractals, which means that they are self-similar. This can be shown most easily by zooming in on part of an image of a Julia set. When this is done, the self-similarity will show itself. Not all the Julia sets are totally connected, meaning that they have multiple pieces. Those sets that are not totally connected are said to be composed of fractal dust. All this means is that the set is composed totally of disconnected pieces. In other words, the Julia set is made up exactly like the Cantor set. A quick way to find out if a Julia set is totally connected or is composed of fractal dust is to look at where the initial value of zero goes. If zero escapes towards infinity, then the Julia set is composed of fractal dust, and if it does not then it is totally connected.

There is a way to find all the Julia sets in the quadratic family. They can be found in an image called the Mandelbrot set. The Mandelbrot set is the set of all c-values for which the critical orbit of \( W_c \) does not tend towards infinity. Every point inside and on the boundary of the Mandelbrot set corresponds to an entirely different Julia set. This means that each point in the image is a completely different function. It is kind of like every point in the image of the Mandelbrot is a doorway to a completely different world.
Recall now that every Julia set for the quadratic family lives inside the interval of 
\[-2 \leq z^2 + c < 2\]. Since the Mandelbrot set is the collection of all the Julia sets of the 
quadratic family, it also means that the entire Mandelbrot set lives inside of a circle of 
radius 2. The Mandelbrot set is also symmetric about the x-axis (real numbers). The 
reason for this is because of the complex conjugates of the y-axis (complex numbers). 
Remember that the complex conjugate of the number \(x + yi\) is simply the change in sign 
from addition to subtraction \(x - yi\). When zooming in on the Mandelbrot set it is 
important to remember that unless we zoom in on the real number axis the symmetry of 
the Mandelbrot set no longer holds true. Although it may not appear so in some pictures, 
the Mandelbrot set is actually totally connected. If we zoom in far enough on some of the 
seemingly isolated islands in the Mandelbrot set we will find very thin strands connecting 
these islands to the main body.

The Mandelbrot set is also self-similar. Any part of the Mandelbrot set that we 
zoom in on, when magnified, looks just like the original image. The last interesting 
aspect of the Mandelbrot set that we discuss is the fact that every bulb in the Mandelbrot 
set has its very own corresponding periodic number. In the main bulb, which looks 
somewhat like a round, fat heart, the periodic number is one. Every Julia set that is found 
in this cardioid-shaped area has one fixed point that it centers on. The bulb that is 
directly connected to its left has a period of two meaning that these Julia sets jump back 
and forth between two fixed points. The bulb directly on top of the main cardioid has a 
period of three, which means it goes from the first fixed point, to the second, to a third 
and then returns to the first fixed point to start all over again. Each bulb has its own 
period, but a pattern has yet to be discovered in the relationship between the bulbs.
position and its period.

The computer program *The Julia Sets* written by Evgeny Demidov gives a very good first look at the Mandelbrot set. The first two things seen on the screen are two pictures, one of the Mandelbrot set, and the other of the Julia set that corresponds to the point where the cursor is at on the Mandelbrot set. The cursor can be moved anywhere on the Mandelbrot set by clicking on a spot on it. The cursor actually moves to the spot from its previous location so it takes a little time to travel to the new point. On top of the pictures is information about the animation, iterations of the Julia set and the number of steps it takes to compute it, the equation at the point the cursor is at, as well as the period of that point. The quick introduction of this program does a very good job of describing both the Mandelbrot set as well as the Julia set. It is clear for the most part with only a few spots that require more intuitive thinking. Those spots, in particular, are the last two parts of the quick introduction.

The other parts of this website quickly get into higher level thinking. The topics covered to help explain the Mandelbrot set and the Julia sets are: Iterations of Real functions, Dynamics on the Complex plane, Fixed point theory, Periodic and preperiodic point in the Mandelbrot set, Mandelbrot and Julia-bulb scaling theory, and $Z^{3,4,N}$ and rational $Q_M(Z)/P_N(Z)$ Mandelbrot and Julia sets. The program also has an illustration component where some of the known more beautiful images can be found like the sea horse and the elephants. Also included are some interesting and different colorings of some of these pictures as well.

This program does an excellent job of being both simple enough to understand at its most basic level and yet still challenging enough to keep coming back to learn more.
about the Mandelbrot and Julia sets. Each subsection in this software is loaded with graphics to help the user visualize and understand the mathematics and explanations being discussed. Each subsection is short and to the point. The content page of the website allows for very quick and easy access to any section of the site.

*Chaos: The Software* is another software program that looks at fractal geometry and chaos theory. *Chaos: The Software* has six different areas that explore fractal geometry. The first area is simply the Mandelbrot set. It is a full picture of the set with the ability to zoom in and explore different areas. This program gives the user a better chance to explore the Mandelbrot set than *The Julia Sets* program. This part of the software also allows the user to look at a Julia set, a Rudy set, the cubic Mandelbrot, and a cubic Julia set. This first part is only a look at the appearance of these sets.

The second part of this program is entitled Magnets and Pendulums. In this part of the software, the user can explore attracting and repulsing boundary areas. The program has several different preset placements of magnets, positive for attracting and negative for repulsing, for the user to look at. At the same time though the user can add, delete, and even move the magnets to any place on the screen. The user can also set the strength of the magnets to simulate the varying degrees of pulling and pushing that these boundaries can have.

The third section of this program is labeled Strange Attractors. The four types of strange attractors that it can display are the Lorenz, Yorke, Henon, and Henon Horseshoe. The default type is the Lorenz, which has two attractors on the screen and several points flowing around them. The points are represented by X’s and have tails so that the user can see where the point has been. The user can also add more points by left clicking
anywhere on the screen to see where that point will eventually fix itself to.

The Chaos Game is the title of the fourth section. This section also has preset images such as the nautilus, fern, Sierpinski's triangle, Peitgen tree, zigzag, spirals, and the word chaos, among others. The best part about this section is that it is here that the user can make their own fractal image. By tweaking an existing image they can make their own. Although it helps to have an understanding of linear algebra to do this, the set for each matrix is easy enough that anyone can do it. The user can change the dimensions of any matrix simply by scrolling through, picking one, and then changing the values of that matrix. The first thing that the user will learn in this section is how quickly the entire picture can change simply by altering one matrix. This is the section of the program where the user will learn the most about how fractal images are part of the whole and the whole is the part.

The fifth section, entitled Fractal Forgeries, shows how fractals can be used to make natural objects like planets, clouds, and mountains. As in each previous section, the user can tweak these images to create new and different planets, clouds and mountains. This section can give the user a beginning understanding of how the movie industry uses computer technology to do everything from adding objects into movies to making an entire movie from a computer. The sixth section, which is called Toy Universes, has four games to play.

*Chaos: The Software* covers aspects of fractal geometry and Chaos theory that *The Julia Sets* does not. Its help sections are somewhat helpful, but the only way to really get a sense and feel for the program is by trial and error. It is also confusing in sections because the program starts off assuming that its user is familiar with Chaos
theory and fractal geometry. The good side to this minor problem though is that it is a very easy program to use and easy to move around in, even in the beginning.

Chaos theory and fractal geometry are perfect counterexamples to the idea that mathematics has not grown since the time of Newton. Although in its infancy as compared to other fields of mathematics, Chaos theory has already proven itself as not only a true field of study but as an interesting and exciting field, too. Chaos theory will continue to grow and develop as technology becomes better and better, and more people start to study this fascinating field of mathematics.
Figure 1
Sierpinski's Triangle
Figure 2
Fern
Figure 3
Pietgen’s Tree
Works Cited


http://www.mathcs.sjsu.edu/faculty/rucker/chaos.htm

http://www.ipm.sci-nnov.ru/~DEMIDOV/MSet/Contents.htm
Chaos Lesson Plan

Standards: Technology
            Geometry

Goals: To have students know there is more to math than paper and pencil.
       To have a beginning understanding of Chaos theory.
       To have a beginning understanding of fractal geometry.

Objectives: To know what the Mandelbrot set is.
            To know how the Mandelbrot set is made.
            To know what Julia sets are.
            To know some special Julia sets.

Materials:  Chaos: the Software
            The Julia Sets
            A computer for every two or three students.

Instruction:

(It is important that you know how to operate and have an understanding of these two computer programs before taking the students to the lab. The worksheet is self-explanatory so your main role is to answer questions as they arise from the students self exploration of the programs.)

Start giving the paper Total Chaos...Total Chosa to read as homework.

The next day take a few minutes to answer any questions about the paper itself.

Have them meet in the computer lab and group themselves to two or three's depending on how many computers are available.

Pass out the Chaos Worksheet.

Have them go through the quick tour of The Julia Sets.

When finished have them start answering the questions on the work sheet, switching to Chaos: the Software when the worksheet calls for it.

(This worksheet should take two days [three at most] to complete in the lab.)
Chaos Worksheet

Directions: Open up, *The Julia Sets* program. Read through the quick tour and then answer the following questions.

1) What is the Mandelbrot set symmetrical about?

2) What are the periods of the four main bulbs of the Mandelbrot set?

3) What do the Julia sets look like in general when you pick a point well inside the Mandelbrot set?

4) What do the Julia sets look like in general when you pick a point well outside the Mandelbrot set?

5) What do the Julia sets look like in general when you pick a point close to the boarder and inside the Mandelbrot set?
6) What do the Julia sets look like in general when you pick a point close to the border but outside the Mandelbrot set?

7) Where do you think that some of the pictures such as the elephants and the seahorses lie? Why?

8) Using the opening two applets zoom in and find the both the elephants and the seahorses. Print out a picture of each and write the equation of each Julia set as well.

Directions: Close The Julia Sets program and open the program Chaos: the Software. Go to the Strange Atractors section and answer the following questions.

9) Put several new points on the screen, starting in different locations. Do the points form some sort of pattern as they approach these attractors? If so what does it look like?

10) Do the starting points seem to have a pattern of where they go? (Does one area go to one attractor, or is it random where the points go.)
Directions: Exit the strange Attactors and now go the Pendulums and Magnets section.

11) Go the second layout. Start the pendulum in the lower left-hand part of the screen. Does a pattern seem to develop? If so what?

12) What eventually happens to the screen? Why do you think this happens?

13) Go to the third layout and change all the odd magnets to a charge of negative two. Start several pendulums in different locations. What seems to happen to the pendulums when they are started?

14) Scroll to the fifth layout. Start a pendulum outside of the negative circle. What pattern seems to develop?

15) Does this pattern remind you of anything?

16) What happens when the pendulum is started inside the circle?
17) What happens when the charges of all the magnets are reversed?

Directions: Go to Chaos the game.

18) Press F4. What happens to each individual matrix when the parameters A through F are "tweaked" positively?

A: 
B: 
C: 
D: 
E: 
F: 

19) What happens when the are "tweaked" negatively?

A: 
B: 
C: 
D: 
E: 
F: 

20) As each individual matrix is altered what happens to the picture as a whole?
21) Starting with one of the preset images, make your own image. Tell me which image you started with. You may add and delete matrices. (You must have a minimum of six matrices.

22) Write a minimum of a two-page report telling me what you learned from this math lab. Include in your report some things that this sort of technology could be used for, and where it might go in the future.