Fractals And Fractal Geometry

adapted from the ThinkQuest page http://library.thinkquest.org/3493/frames/fractal.html

[What is a Fractal? | Building Fractals | Fractal Dimension]
[Famous Fractals | Real-Life Relevance]

What is a Fractal?

Classical Euclidean geometry works with objects which exist in integer dimensions: zero-dimensional points, one-dimensional lines and curves, two-dimensional surfaces like planes, and three-dimensional solid objects such as balls and blocks (think spheres and cubes). However, many things in nature are better described as having a dimension which is not a whole number, because of a property called *self-similarity*: if you magnify some part of the object, you will find that part is identical to the whole object, on a smaller scale. A common example is a fern branch, where each leaf resembles the entire branch in miniature (**figure 0a**).

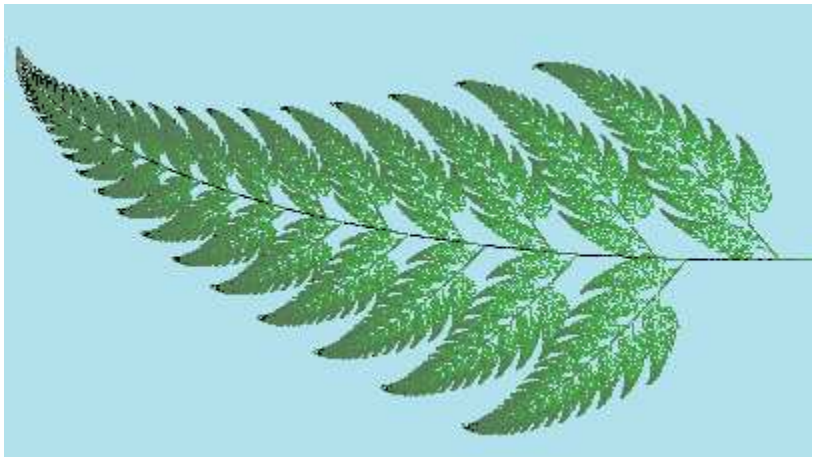

Figure 0a. A fractal fern leaf. *(Source: ThinkQuest)*

The fern leaf, when pressed flat, is part of a two-dimensional plane, and appears to take up two-dimensional space, even though any individual point on the fern appears to be simply part of a one-dimensional curve that helps make up the leaf. One may argue that the fern's dimension is then somewhere between one and two: it doesn't really take up two-dimensional space in the same way the interior of a square does, but it does take up more than just the one dimension of a simple curve. Such an object is said to have *fractal dimension*, by virtue of its self-similarity, and such objects are said to be *fractal*. Benoit Mandelbrot coined this term in 1975 from the Latin adjective *fractus*, meaning fragmented and irregular, to describe objects with the two properties of self-similarity and fractal dimension. (We may unconsciously associate "fractal" with "fractional", but we will see below that fractal dimensions are rarely what we normally think of as fractions, i.e., rational numbers.) While a straight line has a dimension of exactly one, a fractal curve will have a dimension between one and two, depending on how much space it takes up as it curves and twists. The more a fractal curve fills up a plane, the closer it approaches two dimensions. In the same manner of thinking, a fractal surface (like, say, the surface of a cloud) will cover a dimension somewhere between two and three. Hence, a fractal landscape which consists of a hill covered with tiny bumps would be closer to two dimensions, while a landscape composed of a rough surface with many average sized hills would be much closer to the third dimension.

Self-similarity, in its strictest interpretation (also called *scale invariance*), implies that a fractal object has an infinite level of detail, as no matter how much you magnify the fractal, you will see the same amount of detail. **Figure 0b** shows an animation of one well-known example, the *Koch curve* (which we will revisit below), showing exactly this magnification of detail.

Figure 0b. Animation of the Koch curve showing self-similarity. *(Source: Wikipedia)*

Building Fractals

Where Euclidean geometry describes lines, ellipses, circles, etc. with equations, fractal geometry describes objects in terms of algorithms --- sets of instructions on how to create a fractal. One way (arguably the most elementary) to describe fractals is through what are called *iterated function systems*, or IFS. This is the only type of fractal that we shall discuss in detail in this article. IFS follow the general approach of altering a geometric object in a particular way, leaving multiple smaller objects each of which is similar to the original, and then repeating the process on each of those smaller objects to create even smaller parts, and so on. The fractal is the result of carrying this process out infinitely many times. (Of course, in practice, we cannot repeat the process infinitely many times, but in practice at some point we run up against the resolution of our ability to draw the results --- that is, past some point the changes we make are no longer visible, unless we magnify the original figure.)

One of the earliest, and simplest, fractals is the *Cantor set*, or *Cantor dust*, defined by mathematician Georg Cantor (1845-1918). To create it, begin with a line segment (say, the segment from 0 to 1 on a number line), and remove the middle third. Then remove the middle third of each of the resulting pieces. After the first step, there are two segments of length 1/3, after the second, there are four segments of length 1/9, and so on (**figure 0c**). After many steps there appears to be simply a very large collection of tiny points --- so many that they appear to "take up space" on the one-dimensional line.

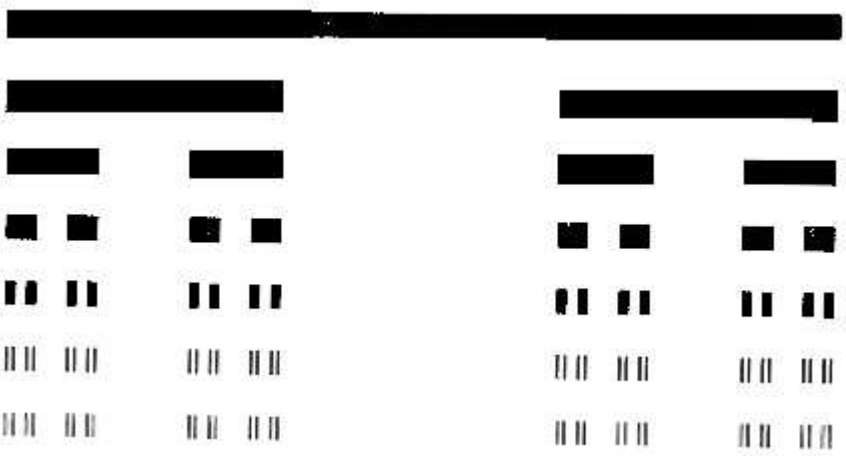

Figure 0C. The Cantor set (points represented with vertical thickness in order to improve visibility)
(Source: ThinkQuest)

We will return to the Cantor set and other fractals shortly, but now we will investigate the question of how exactly one can calculate fractal dimensions. The dimension of the Cantor set should be more than zero but

less than one; the fern leaf should have dimension between one and two. How can we determine what those dimensions are?

Fractal Dimension

Now you understand what fractals are and where they come from, how are their dimensions calculated? For certain objects with which you have dealt all of your life, such as squares, lines, and cubes, it is easy to assign a dimension. You intuitively feel that a line has one dimension, a square (including its interior) has two dimensions, and a cube (with its interior) has three dimensions. You might feel this way because there is one direction in which you can move on a line (that is, the direction parallel to the line), two directions on a square, one direction on a line, and three directions in a cube, but what about fractals? Sometimes you can move in a certain number of directions and sometimes you can move in a different number of directions. This is what causes fractal dimensions to be non-integers.

To derive a formula which will work with all figures, let's first look at how dimensions manifest when we use self-similarity to magnify the Euclidean figures whose dimensions we already know. (We focus our investigation on self-similarity because it is the defining characteristic of fractals.) If we magnify a line segment [dimension 1] by a scale factor of n, then we get $n = n^1$ segments identical to the original line. If instead we magnify a square (with its interior) [dimension 2] by a (linear) scale factor of n, then we get n^2 squares identical to the original square. The same concept holds true for a cube (with interior) [dimension 3]: magnification by a scale factor of n yields n^3 cubes, each identical to the original cube. Therefore, if we magnify a fractal by a (linear) scale factor of n and get some number k of copies of the original fractal, then the dimension of that fractal should be a number D such that $n^D=k$.

We can solve this equation for D using logarithms. (Here we will use "log" to mean the natural logarithm or logarithm base e, although the properties of logarithms can be used to show that it actually does not matter what base is used to calculate the logarithms.) By the properties of logarithms, $log(n^D)=log(k)$ becomes $D\ log\ n=log\ k$, or

$$D = log\ k/log\ n,$$

where n is the magnification scale factor (or, equivalently, the number of equal pieces into which objects are broken at each step) and k is the number of copies of the original object obtained (or created) at each step.

We can verify that this approach is consistent with the Euclidean notion of dimension for the three simple objects mentioned above:

For a line: $D = log(n^1)/log(n) = 1\ log(n)/log(n) = 1$
For a square: $D = log(n^2)/log(n) = 2\ log(n)/log(n) = 2$
For a cube: $D = log(n^3)/log(n) = 3\ log(n)/log(n) = 3$

Now let us consider some examples based on a line segment. (Since a line segment has dimension 1, we should expect any IFS which removes parts from it to create a fractal with dimension less than 1, and any IFS which adds parts to it to create a fractal with dimension greater than 1.) In the simplest case, suppose we partition the line segment into 2 pieces at each step. If we keep only one of the pieces each time, then after one step we have a line segment of length 1/2, after two steps we have a line segment of length 1/4, and so on (**figure 1a**). Eventually every point except the left endpoint will be thrown away, leaving just a single point. If we apply the formula above, we have that this point should have dimension

$$D = \log k/\log n = \log 1/\log 2 = 0,$$

since at each step we scale down by a factor of $n=2$ but keep only $k=1$ of the pieces. This is consistent with our notion that a single point has dimension zero.

If we keep the scale factor $n=2$ and instead keep both pieces ($k=2$), then of course we are never losing or adding any points, and will end up with the same line segment (**figure 1b**). The dimension is calculated as

$$D = \log k/\log n = \log 2/\log 2 = 1,$$

as it should be. Therefore, in order to generate a true fractal, we will need a scale factor n greater than 2.

The next simplest example is to choose $n=3$ and $k=1$; however, we soon see that keeping only one part ($k=1$) will always reduce the line segment in the end to a single point. Therefore, let us choose $n=3$ and $k=2$, at each step splitting each line segment into thirds and keeping 2 of those thirds. If we keep the first and last third, then we have discovered the "remove the middle third" Cantor set mentioned above (**figure 1c**). This "Cantor dust" has dimension

$$D = \log k/\log n = \log 2/\log 3 \approx 0.6309,$$

which is indeed between 0 and 1. We can make many different "Cantor dust" fractals; **Figure 1d** illustrates what happens if we choose $n=5$ and $k=3$, in which case we get a fractal set with dimension

$$D = \log k/\log n = \log 3/\log 5 \approx 0.6826.$$

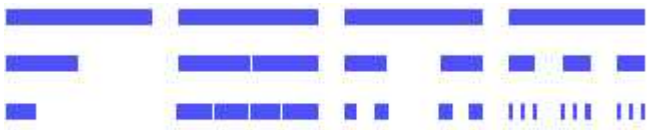

Figure 1. Demonstration of fractal dimensions with Euclidean line segments.

The four examples in Figure 1 all involve removing parts of a line segment. It is also possible to create a fractal by adding parts to a line segment, or through some combination of adding and removing parts. The Koch curve and Koch snowflake described in the next section are examples of fractals created by adding and removing parts of a line segment; in the case of the Koch curve, the adding turns out to outweigh the removing, as the resulting curve begins to take up 2-dimensional space and will be seen to have a dimension greater than 1. Examples of fractals created by only adding parts to a line segment (or curve) include fractal "trees" like the fern leaf in figure 0a.

Fractals can also be created beginning from a two-dimensional object. Each of the following examples divides a unit square with a (linear) scale factor of $n=3$, resulting in 9 smaller squares at each step, of which we keep varying numbers k. To demonstrate the Euclidean point (**figure 2a**), we keep only one square with each iteration, making $k=1$. In the next example (**figure 2b**), we create a Euclidean line segment by keeping only the top three squares with each iteration, making $k=3$. The last Euclidean figure which can be derived from this example is the plane (**figure 2c**). To accomplish this, we keep all the squares with each iteration ($k=9$). The reader can verify that the fractal dimension formula calculates dimensions of 0, 1, and 2, respectively, for these objects.

Here we will give two examples of figures with fractal dimension, both inspired by the classic Cantor set. In **figure 2d**, we keep only the two pieces in the upper left and lower right corner with each iteration. In this case, $k=2$ and $n=3$, reproducing exactly the dimension $\log 2/\log 3 \approx 0.6309$ of the original Cantor dust --- but

if we look at figure 2d, we can see that this fractal will end up as just the original (linear) Cantor set turned 45^{o}. In **figure 2e**, we remove only the middle piece, this time leaving $k=8$ of the nine pieces at each step. This process leaves the fractal known as the *Sierpinski carpet*, whose dimension is $\log 8/\log 3 \approx 1.8928$ or three times that of the Cantor set.

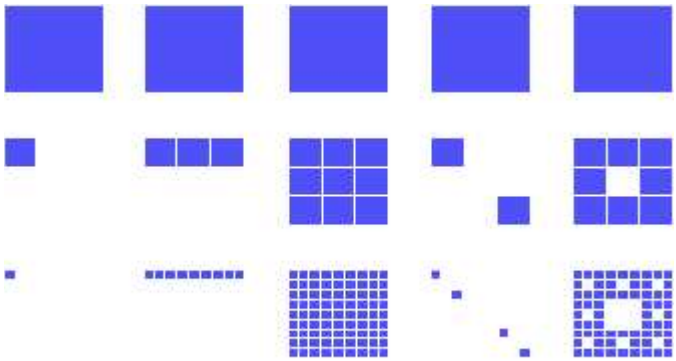

Figure 2. Demonstration of fractal dimensions with Euclidean planes.

Famous Fractals

[The Cantor Set | The Koch Curve | The Koch Snowflake | The Sierpinski Triangle | The Lorenz Model | The Mandelbrot Set | The Julia Set | The Logistic Equation]

Below are several sections, each dealing with an individual fractal. Of course, not all of the fractals in the world are listed below, but only ones which are well known or illustrate an important point which everyone should know. With each fractal, there is a picture, followed by some information about it. For many of the fractals, there is also a link to a C/C++ or BASIC program which will generate a picture of the fractal. For more working source code, visit the Appendix Of Source Code at ThinkQuest. Even if you are not interested in these specific fractals, it is strongly enouraged that you read through each one because many topics other than the specific fractal are reviewed. For example, strange attractors and several applications of fractals to real-life situations are discussed.

The Cantor set

Figure 3. The Cantor set

The Cantor set is a good example of an elementary fractal. The object first used to demostrate fractal dimensions, **figure 1c**, is actually the Cantor set. The process of generating this fractal is very simple. The set is generated by the iteration of a single operation on a line of unit lenght. With each iteration, the middle third from each lines segment of the previous set is simply removed. As the number of iterations increases, the number of separate line segments tends to infinity while the length of each segment approaches zero. Under magnification, its structure is essentially indistinguishable from the whole, making it self-similar.

To calculate the dimension of the Cantor set, we first realize that its magnification factor is three, or the fractal is self-similar if magnified three times. Then we notice that the line segments decompose into two smaller units. Using the formula given in the section entitled Fractal Dimension, we get:

D = log(2) / log(3)
D = 0.6931 / 1.0986
D = 0.6309

The Cantor set has a dimension of 0.6309.

The Koch Curve

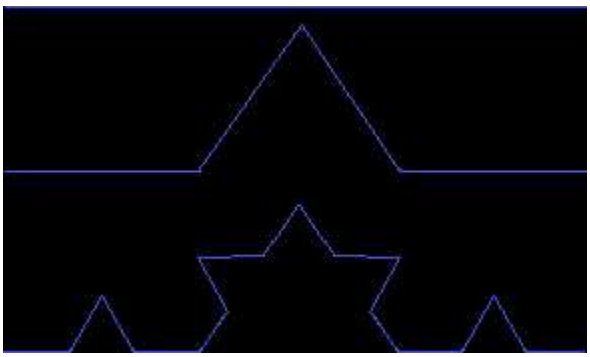

Figure 4. The Koch Curve

So far, all of the examples in this document have dealt with removing pieces from various geometric figures. Fractals, and fractal dimensions can also be defined by adding onto geometric figures. The Koch curve was named after Helge von Koch in 1904. The generation of this fractal is simple. We begin with a straight line of unit length and divide it into three equally sized parts. The middle section is replaced with an equilateral triangle and its base is removed. After one iteration, the length is thus increased by a factor of 4/3 (since there are four segments in place of three). As this process is repeated, the total length of the figure tends to infinity as the length of the side of each new triangle goes to zero. Assuming this could be iterated an infinite number of times, the result would be a figure which is infinitely wiggly, having no straight lines whatsoever.

To calculate the dimension of the Koch Curve, we look at the image of the fractal and realize that it has a magnification factor of three and with each iteration, it is divided into four smaller pieces. Knowing this, we get

D = log(4) / log(3)
D = 1.3863 / 1.0986
D = 1.2619

The Koch Curve has a dimension of 1.2619.

The Koch Snowflake

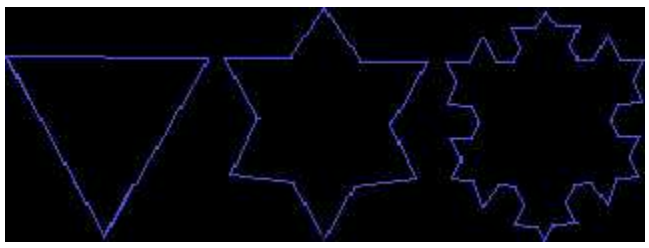

Figure 5. The Koch Snowflake

As would be expected, the Koch Snowflake is generated in very much the same way as the Koch

Curve. The only variation is that, rather than using a line of unit length as the initial figure, an equilateral triangle is used. It is iterated in the same way as the Koch Curve. The length of the resulting figure tends to infinity as the length of the side of each new triangle goes to zero. Iterated an infinite number of times, the Koch Snowflake, like the Koch Curve, has absolutely no straight lines in it. This fractal, if magnified three times in any area, also displays the property of self-similarity.

As mentioned above, the magnification factor of this fractal is three, and as with the Koch Curve, the number of divisions in each magnification is four. With this we get:

D = log(4) / log(3)
D = 1.3863 / 1.0986
D = 1.2619

The Koch Snowflake has a dimension of 1.2619.

The Sierpinski Triangle

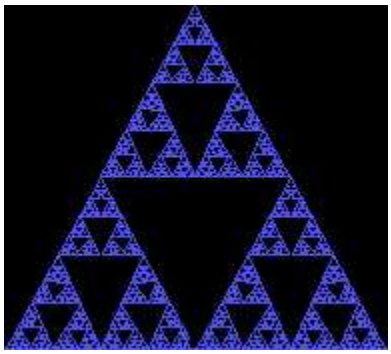

Figure 6. The Sierpinski Triangle

Unlinke the Koch Snowflake, which is generated with infinite additions, the Sierpinski triangle is created by infinite removals. Each triangle is divided into four smaller triangles, each of them half the length (height, etc.) of the original. The center triangle (the only one of the four which is "upside-down") is removed, reducing the area of the whole shape by a factor of 3/4. As this process is iterated an infinite number of times, the total area of the set tends to zero as the size of each new triangle goes to zero.

Since each smaller triangle is reduced in linear dimension from the original by a factor of $n=2$, and we keep $k=3$ smaller triangles at each step, we can calculate the dimension of the Sierpinski triangle as:

D = log(3) / log(2)
D = 1.0986 / 0.6931
D = 1.5850

The Sierpinski Triangle has a dimension of 1.5850.

The Sierpinski Triangle is one of the easiest fractals to generate yourself. This particular program is written in C/C++ and uses the Borland graphics routines, making it easy to port to other languages on different platforms. The source code to this program is available here.

Non-IFS Fractals

The following fractals are included in our "rogues gallery" of famous fractals because they are so well-known, but they cannot be generated by the same kind of IFS as the ones above, so we will not describe

them in the same kind of detail, nor try to calculate their fractal dimensions.

The Lorenz Model

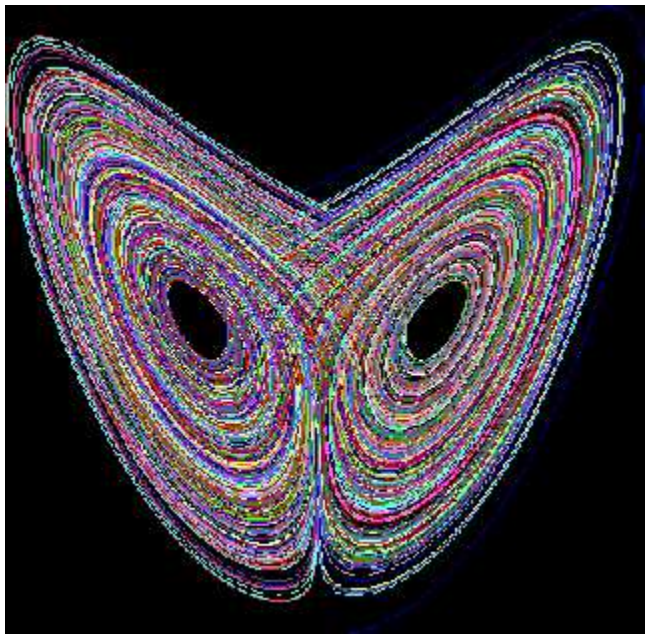

Figure 7. The Lorenz Model

The Lorenz Model, named after E. N. Lorenz in 1963, is a model for the convection of thermal energy. This model was the very first example of another important point in chaos and fractals, dissipative dynamical systems, otherwise known as strange attractors. Strange attractors are covered more in depth here.

The Lorenz Model is not a particularly difficult fractal to generate graphically. The source code for a program, written in C/C++, which will generate an image of the Lorenz Model is available here.

The Mandelbrot set

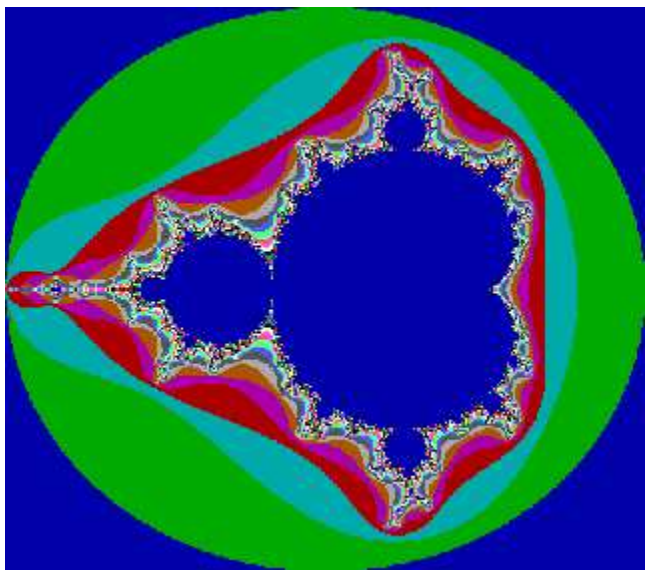

Figure 8. The Mandelbrot set

Named after Benoit Mandelbrot, The Mandelbrot set is one of the most famous fractals in existance. It

was born when Mandelbrot was playing with the simple quadratic equation $z=z^2+c$. In this equation, both z and c are complex numbers. In other words, the Mandelbrot set is the set of all complex c such that iterating $z=z^2+c$ does not diverge.

To generated the Mandlebrot set graphically, the computer screen becomes the complex plane. Each point on the plain is tested into the equation $z=z^2+c$. If the iterated z stayed withen a given boundry forever, convergence, the point is inside the set and the point is plotted black. If the iteration went of control, divergence, the point was plotted in a color with respect to how quickly it escaped.

When testing a point in a plane to see if it is part of the set, the initial value of z is always zero. This is so because zero is the critical point of the equation used to generate the set. Critical points are explained in depth here.

For a program, again written in C/C++, which will generate the Mandlebrot set graphically and allow you to explore the set, go here. For a program writting in BASIC which will generate the Mandelbrot set, go here.

The Julia set

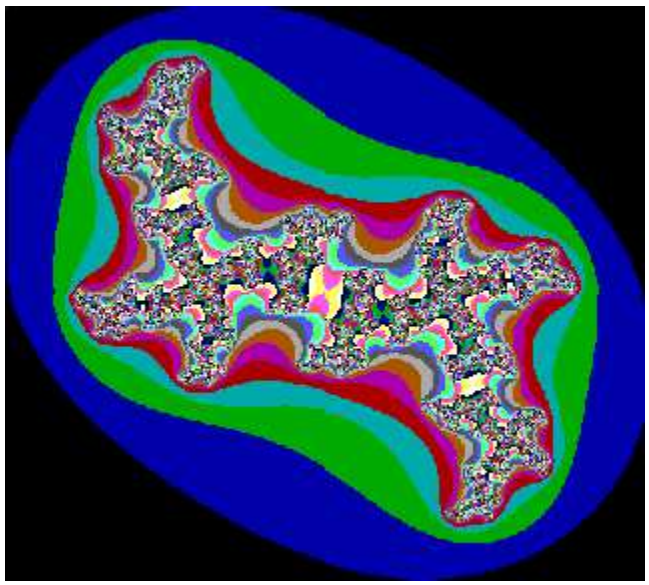

Figure 9. The Julia set

The Julia set is another very famous fractal, which happens to be very closely related to the Mandelbrot set. It was named after Gaston Julia, who studied the iteration of polynomials and rational functions during the early twentieth century, making the Julia set much older than the Mandelbrot set.

The main difference between the Julia set and the Mandelbrot set is the way in which the function is iterated. The Mandelbrot set iterates $z=z^2+c$ with z always starting at 0 and varying the c value. The Julia set iterates $z=z^2+c$ for a fixed c value and varying z values. In other words, the Mandelbrot set is in the parameter space, or the c-plane, while the Julia set is in the dynamical space, or the z-plane.

A program writtin in C/C++ which will graphically generate the Mandelbrot set can be found here. Its BASIC counterpart is located here.

Logistic Equation

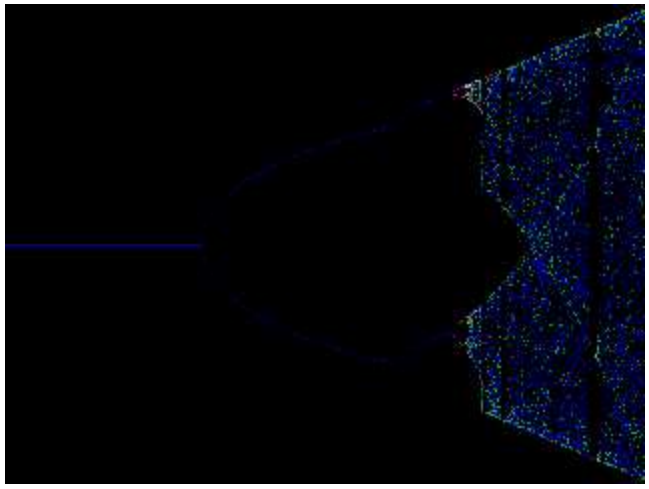

Figure 10. Diagram plotted for the logistic equation

This particular fractal has much more of a relevance to real life than many other fractals. The logistic equation is a model for animal populations with limited resources. The actual equation used is $x_{n+1}=c(1-x_n)$, where x is the population, between 0 and 1, and c is a constant representing the growth rate. Iteration of this equation results in the doubling route to chaos. For c equaling a value between one and three, the population will settle to a fixed value. For values of c just a little greater than 3, however, the population settles into a sustained oscillation. In other words, one year the population is high, causing the population the next year to be low, causing a high population the year after. As the value of c increases, however, the population cycle is suddenly spread out over four years, rather than two. This phenomenon is called *period doubling*. As c continues to increase, the period doubles faster and faster, with thresholds of 3.54, 3.564, 3.569, etc. When this value reaches 3.57, chaos occurs and the population never settles to a fixed point. For most values of c between 3.57 and 4, the population is chaotic.

In a period doubling situation, such as the logistic equation, the ratio of distances between consecutive period-doubling thresholds is called Feigenbaum's constant. Based on computations by Jay Hill and Keith Briggs, the constant has a value of 4.669201609102990671853... The interpretation of this equation is that as you approach chaos, each periodic region is smaller than the previous one by a factor which approaches Feigenbaum's constant. This constant is the same for any quadratic function or system that follows the periodic doubling route which results in chaos. The constant varies for functions of a higher order.

Real-Life Relevance And Importance of Fractals and Fractal Geometry

Fractals continue to be used in many different ways. Both artists and scientists are intrigued by the properties of fractals. Fractals are being used in applications ranging from image compression to finance. We are still only beginning to realize the full importance and usefullness of fractal geometry.

One of the largest relationships with real life is the similarity between fractals and objects in nature. The resemblance between many fractals and their natural counterparts is so strong that it cannot be overlooked. Mathematical formulas are used to model self-similar natural forms. The pattern is repeated at a large scale and patterns evolve to mimic large scale real world objects.

One of the most useful applications of fractals and fractal geometry is in image compression. It is also one of the more controversial ideas. The basic concept behind fractal image compression is to take an image and express it as an iterated system of functions. The image can be quickly displayed, and at any magnification with infinite levels of fractal detail. The largest problem behind this idea is deriving the system

of functions which describe an image.

One of the more trivial applications of fractals is their visual effect. Not only do fractals have a stunning aesthetic value, that is, they are remarkably pleasing to the eye, but they also have a way to trick the mind. Fractals have been used commercially in the film industry, in films such as *Star Wars* and *Star Trek*. Fractal images are used as an alternative to costly elaborate sets to produce fantasy landscapes.

Another seemingly unrelated application of fractals and chaos is in music. Some music, including that of Bach and Mozart, can be stripped down so that it contains as little as 1/64th of its notes and still retain the essence of the composer. Many new software applications are being developed which contain chaotic filters, similar to those which change the speed, or the pitch of music.

Fractal geometry also has an application to biological analysis. Fractal and chaos phenomena specific to non-linear systems are widely observed in biological systems. A study has established an analytical method based on fractals and chaos theory for two patterns: the dendrite pattern of cells during development in the cerebellum and the firing pattern of intercellular potential. Variation in the development of the dendrite stage was evaluated with a fractal dimention. The order in many ion channels generating the firing pattern was also evaluated with a fractal dimension, enabling the high order seen there to be quantized.

Return to the Chaos Theory, Dynamic Systems, And Fractal Geometry main page at ThinkQuest.