

University of Nebraska - Lincoln

## DigitalCommons@University of Nebraska - Lincoln

---

MAT Exam Expository Papers

Math in the Middle Institute Partnership

---

7-2006

### Fractals and the Chaos Game

Stacie Lefler

*University of Nebraska-Lincoln*

Follow this and additional works at: <https://digitalcommons.unl.edu/mathmidexppap>



Part of the [Science and Mathematics Education Commons](#)

---

Lefler, Stacie, "Fractals and the Chaos Game" (2006). *MAT Exam Expository Papers*. 24.  
<https://digitalcommons.unl.edu/mathmidexppap/24>

This Article is brought to you for free and open access by the Math in the Middle Institute Partnership at DigitalCommons@University of Nebraska - Lincoln. It has been accepted for inclusion in MAT Exam Expository Papers by an authorized administrator of DigitalCommons@University of Nebraska - Lincoln.

# **Fractals and the Chaos Game**

## **Expository Paper**

**Stacie Lefler**

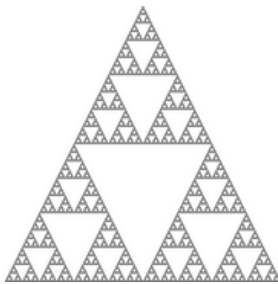
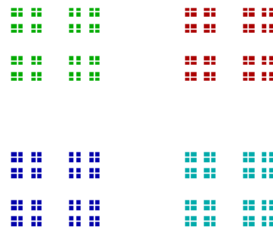
In partial fulfillment of the requirements for the Master of Arts in Teaching with a  
Specialization in the Teaching of Middle Level Mathematics  
in the Department of Mathematics.  
David Fowler, Advisor

July 2006

# Fractals and the Chaos Game<sup>1</sup>

## A. Fractal History

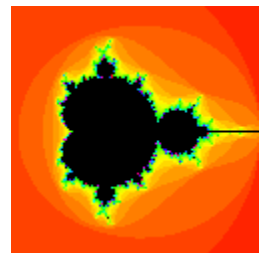
The idea of fractals is relatively new, but their roots date back to 19<sup>th</sup> century mathematics. A fractal is a mathematically generated pattern that is reproducible at any magnification or reduction and the reproduction looks just like the original, or at least has a similar structure. Georg Cantor (1845-1918) founded set theory and introduced the concept of infinite numbers with his discovery of cardinal numbers. He gave examples of subsets of the real line with unusual properties. These Cantor sets are now recognized as fractals, with the most famous being the Cantor Square.



Waclaw Sierpinski (1882-1969), a Polish mathematician, worked in set theory, point set topology, and number theory. He is known for the Sierpinski Triangle.

However, there are many other Sierpinski fractals, such as the Sierpinski Carpet.

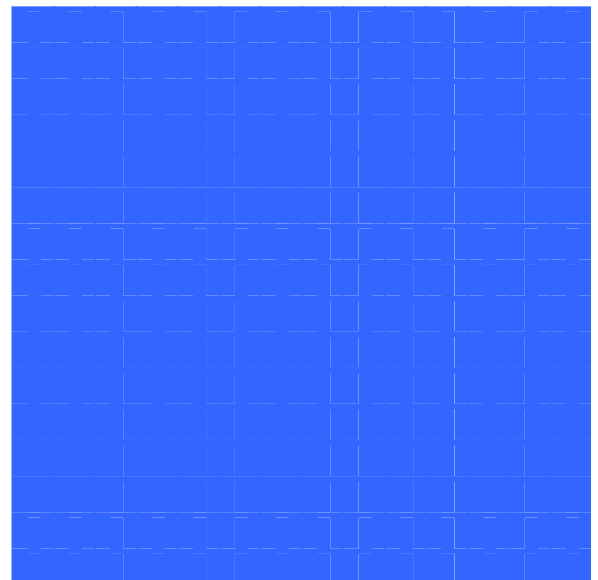
The term ‘fractal’ was coined in 1975 by Benoit Mandelbrot (1924 - ) from the Latin *fractus*, meaning “broken” or “irregular.” This term was used to describe shapes that have the characteristic of self-similarity, i.e. that when you magnify any part it looks just like (or has the same structure) as the original. He is widely known for the Mandelbrot set.



## B. Basic Fractals

Now, let’s try to create some basic fractals using functions on the plane. We can start with a square with corners at  $(0, 0)$ ,  $(1, 0)$ ,  $(0, 1)$ , and  $(1, 1)$ .

We will call our initial image of a square  $S_0$ .



<sup>1</sup> Sections A, B, and C created in collaboration with Sandi Snyder.

We are interested in what happens to our square when we consider the functions

$f_1(x, y) = \left(\frac{x}{2}, \frac{y}{2}\right)$ ,  $f_2(x, y) = \left(\frac{x}{2} + \frac{1}{2}, \frac{y}{2}\right)$ , and  $f_3(x, y) = \left(\frac{x}{2} + \frac{1}{4}, \frac{y}{2} + \frac{1}{2}\right)$ , and evaluate them at the vertices of our square.

When we evaluate  $f_1(x, y) = \left(\frac{x}{2}, \frac{y}{2}\right)$  at the vertices of  $S_0$  we get the following:

$$\begin{aligned} f_1(0, 0) &= (0, 0) \\ f_1(1, 0) &= \left(\frac{1}{2}, 0\right) \\ f_1(0, 1) &= \left(0, \frac{1}{2}\right) \\ f_1(1, 1) &= \left(\frac{1}{2}, \frac{1}{2}\right) \end{aligned}$$

Notice that this takes  $S_0$  and shrinks it to half of its original x length and half of its original y height.

When we evaluate  $f_2(x, y) = \left(\frac{x}{2} + \frac{1}{2}, \frac{y}{2}\right)$  at the vertices of  $S_0$  we get the following:

$$\begin{aligned} f_2(0, 0) &= \left(\frac{1}{2}, 0\right) \\ f_2(1, 0) &= (1, 0) \\ f_2(0, 1) &= \left(\frac{1}{2}, \frac{1}{2}\right) \\ f_2(1, 1) &= \left(1, \frac{1}{2}\right) \end{aligned}$$

Note that this is the same image as we get from  $f_1$ , but it is shifted  $\frac{1}{2}$  unit to the right.

When we evaluate  $f_3(x, y) = \left(\frac{x}{2} + \frac{1}{4}, \frac{y}{2} + \frac{1}{2}\right)$  at the vertices of  $S_0$  we get the following:

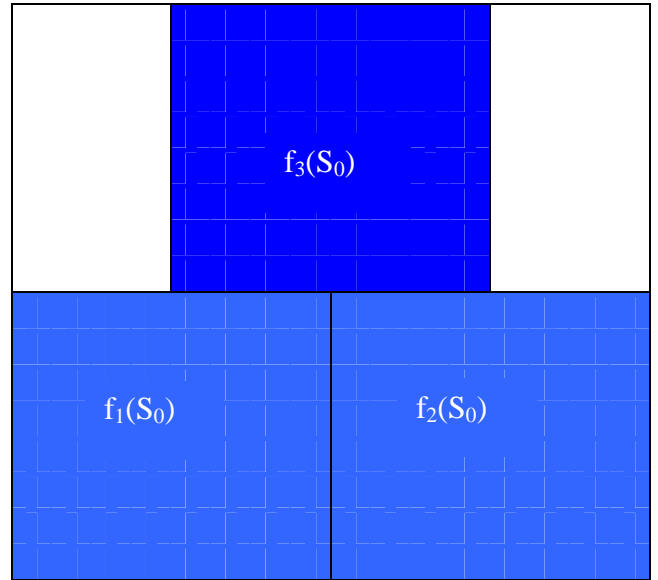
$$\begin{aligned} f_3(0, 0) &= \left(\frac{1}{4}, \frac{1}{2}\right) \\ f_3(1, 0) &= \left(\frac{3}{4}, \frac{1}{2}\right) \\ f_3(0, 1) &= \left(\frac{1}{4}, 1\right) \\ f_3(1, 1) &= \left(\frac{3}{4}, 1\right) \end{aligned}$$

Note that this, too, is the same image as we get from  $f_1$ , but it is shifted  $\frac{1}{4}$  unit to the right and  $\frac{1}{2}$  unit up.

We define the new function formed as  $F(S) = f_1(S) \cup f_2(S) \cup f_3(S)$ .

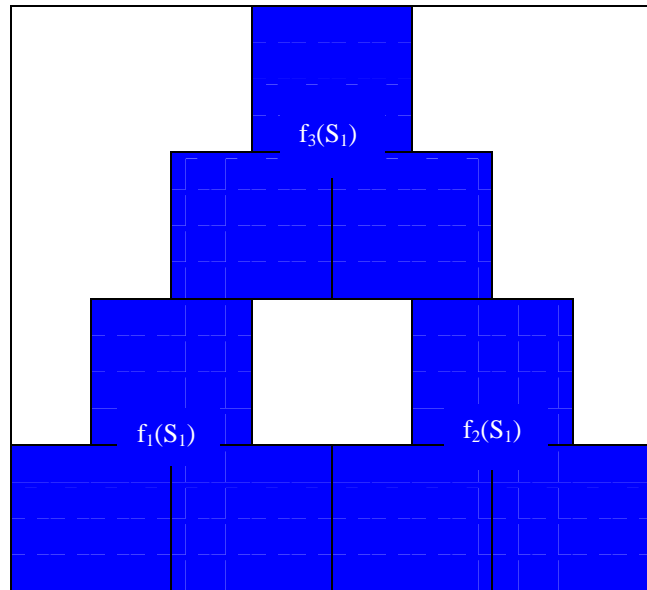
Here is  $F(S_0)$ :

We will call this new image  $S_1$ .



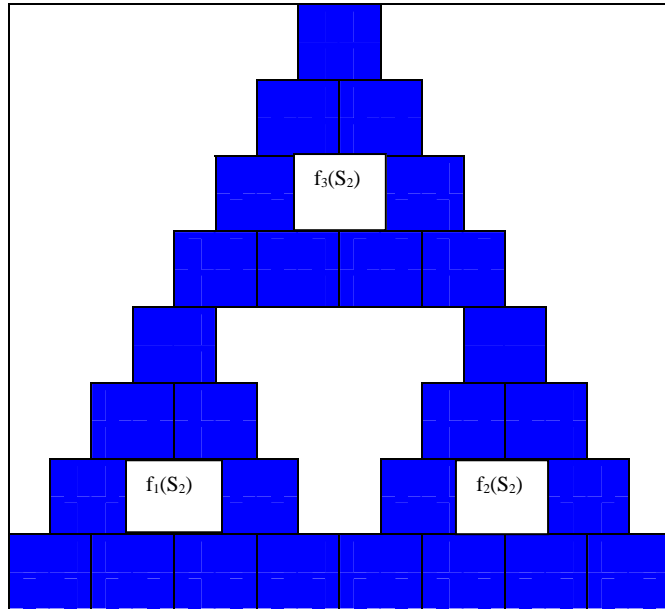
We can now iterate our image again by following the same pattern. To get  $S_2$  we can start by evaluating  $f_1$  at the vertices of  $S_1$  which would shrink  $S_1$  to half of its original x length and half of its original y height. Next, we would evaluate  $f_2$  at the vertices of  $S_1$ . This simply gives a  $\frac{1}{2}$  to the right translation of  $f_1(S_1)$ . Last, we would evaluate  $f_3$  at the vertices of  $S_1$ . This gives us a translation of  $f_1(S_1)$ , too. This one is shifted  $\frac{1}{4}$  unit left, and  $\frac{1}{2}$  unit up.

We'll call this image  $S_2$ .



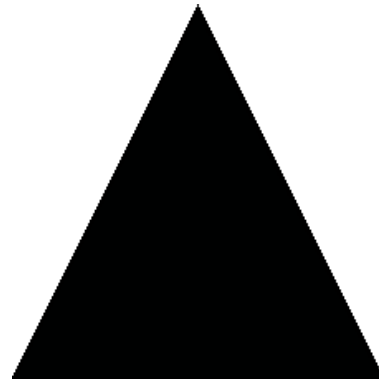
Iterate the image again by evaluating the same three functions at the vertices of  $S_2$ . The first function,  $f_1$ , will shrink the image,  $f_2$  will translate the shrunken figure to the right  $\frac{1}{2}$ , and  $f_3$  will translate the shrunken figure to the right  $\frac{1}{4}$ , and up  $\frac{1}{2}$ .

We'll call this image  $S_3$ .



This is starting to look like the Sierpinski Triangle. We might wonder if our three functions gave us the Sierpinski Triangle because we started with four corners of a square. Let's see what happens if we start with an isosceles triangle instead of a square. We start with an isosceles triangle with corners at  $(0, 0)$ ,  $(1, 0)$ , and  $(\frac{1}{2}, 1)$ .

The initial image is the isosceles triangle. We will call this image  $S_0$ .



When we evaluate  $f_1(x, y) = \left(\frac{x}{2}, \frac{y}{2}\right)$  at the vertices of  $S_0$  we get the following:

$$\begin{aligned} f_1(0, 0) &= (0, 0) \\ f_1(1, 0) &= (\frac{1}{2}, 0) \\ f_1(\frac{1}{2}, 1) &= (\frac{1}{4}, \frac{1}{2}) \end{aligned}$$

Notice that evaluating  $f_1$  with the vertices of  $S_0$  shrinks the image to half of the original length of  $x$  and half the original height of  $y$ .

When we evaluate  $f_2(x, y) = \left(\frac{x}{2} + \frac{1}{2}, \frac{y}{2}\right)$  at the vertices of  $S_0$  we get the following:

$$f_2(0, 0) = \left(\frac{1}{2}, 0\right)$$

$$f_2(1, 0) = (1, 0)$$

$$f_2\left(\frac{1}{2}, 1\right) = \left(\frac{3}{4}, \frac{1}{2}\right)$$

Notice that this gives the same image that we achieved with  $f_1$ , but it has been shifted  $\frac{1}{2}$  to the right.

When we evaluate  $f_3(x, y) = \left(\frac{x}{2} + \frac{1}{4}, \frac{y}{2} + \frac{1}{2}\right)$  at the vertices of  $S_0$  we get the following:

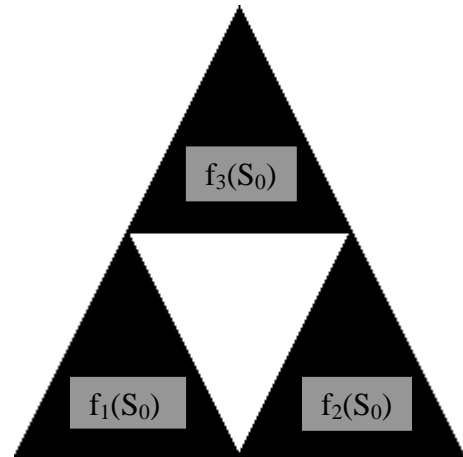
$$f_3(0, 0) = \left(\frac{1}{4}, \frac{1}{2}\right)$$

$$f_3(1, 0) = \left(\frac{3}{4}, \frac{1}{2}\right)$$

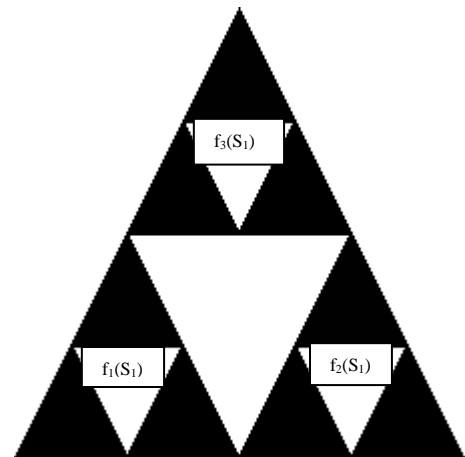
$$f_3\left(\frac{1}{2}, 1\right) = \left(\frac{1}{2}, 1\right)$$

This, too, gives the same image as in  $f_1$ , but it has been shifted  $\frac{1}{4}$  to the right, and  $\frac{1}{2}$  up.

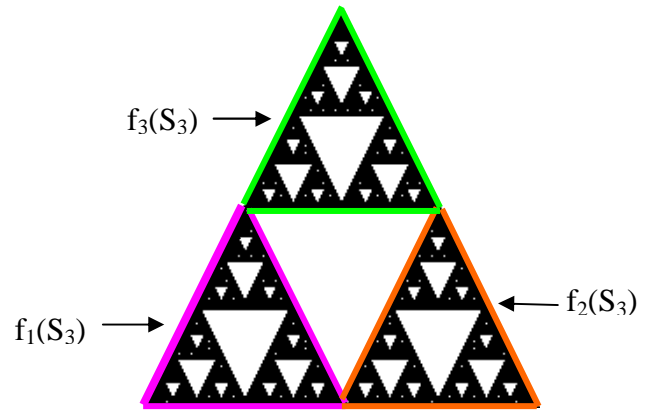
We will call this new image  $S_1$ .



After the second iteration we have the new image,  $S_2$ .



Continue with two more iterations. After the fourth iteration, we have the new image,  $S_4$ .

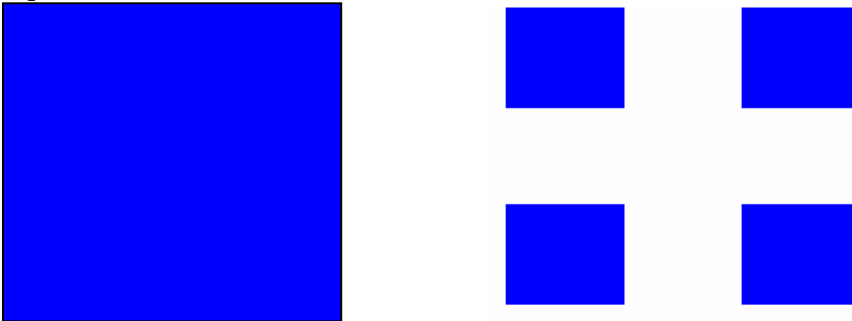


After the fourth iteration, we can see that when we start with a triangle the functions are affecting the image in the same way as when we started with a square. In fact, we could start with any initial image, even a silhouette of Jim Lewis, and after enough iterations using our three functions we would begin to see the Sierpinski Triangle. The final image is actually independent of the initial image. (To learn more about how the final is independent of the initial image, visit [http://www.maths.anu.edu.au/~barnsley/pdfs/V-var\\_super\\_fractals.pdf](http://www.maths.anu.edu.au/~barnsley/pdfs/V-var_super_fractals.pdf).)

If we do enough iterations, the initial image gets smaller and smaller, becoming a dot, and so the final image is in a sense made up of an infinite number of dots. It does not matter what shape we start with, if we apply the same three functions,

$f_1(x, y) = \left(\frac{x}{2}, \frac{y}{2}\right)$ ,  $f_2(x, y) = \left(\frac{x}{2} + \frac{1}{2}, \frac{y}{2}\right)$ , and  $f_3(x, y) = \left(\frac{x}{2} + \frac{1}{4}, \frac{y}{2} + \frac{1}{2}\right)$ , we will get the Sierpinski Triangle.

Another famous iteration is known as the Cantor Square. The Cantor Square, in contrast, is an iteration of four functions. We learned that the first two iterations of the Cantor Square look like this:



From that, we were able to determine the functions that generate the fractal. To create the Cantor Square, we begin with a 1 x 1 square. To this image, we apply the following functions:

$$f_1(x, y) = \left(\frac{x}{3}, \frac{y}{3}\right) \qquad f_3(x, y) = \left(\frac{x}{3}, \frac{y}{3} + \frac{2}{3}\right)$$

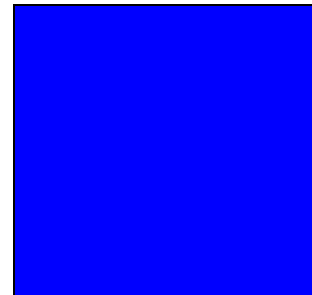
$$f_2(x, y) = \left(\frac{x}{3} + \frac{2}{3}, \frac{y}{3}\right) \qquad f_4(x, y) = \left(\frac{x}{3} + \frac{2}{3}, \frac{y}{3} + \frac{2}{3}\right)$$



We did not know these four functions before creating the fractal. We determined these functions by examining the  $S_0$  and  $S_1$  images. The functions came from discovering the shrinks and translations applied to the initial image,  $S_0$ .

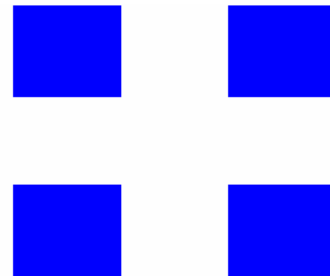
The initial image of the Cantor Square is to the right.

We will call this image  $S_0$ .



$F(S_0)$  is an image that looks like this:

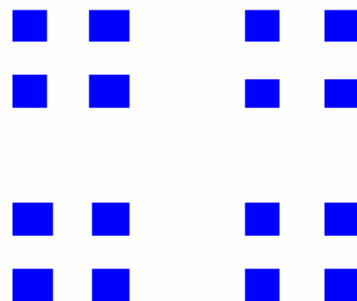
We will call this image  $S_1$ .



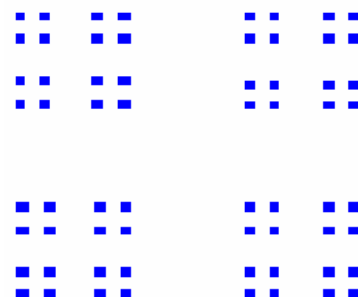
We get these four squares by applying the four functions. The function  $f_1$  simply shrinks the image. The function  $f_2$  translates the shrunken image to the right. The function  $f_3$  translates the shrunken image up, and the function  $f_4$  translates the shrunken image to the right and up.

If we repeat the iteration on the previous image, we get:

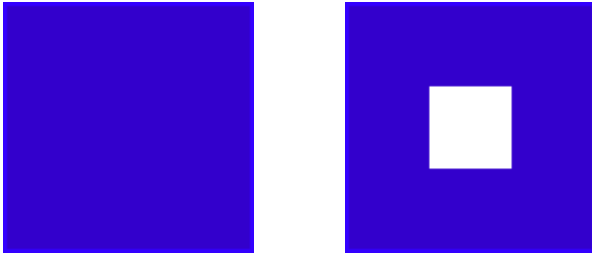
We will call this image  $S_2$ .



And, after one more iteration, we produce this image:  
We will call this image  $S_3$ .



The Sierpinski Carpet is another unique fractal. We learned that the first two images of the Sierpinski Carpet look like the following:



By looking at these images, we determined the eight different functions necessary to generate the Sierpinski Carpet.

$$f_1(x, y) = \left( \frac{x}{3}, \frac{y}{3} \right)$$

$$f_2(x, y) = \left( \frac{x}{3} + \frac{1}{3}, \frac{y}{3} \right)$$

$$f_3(x, y) = \left( \frac{x}{3} + \frac{2}{3}, \frac{y}{3} \right)$$

$$f_4(x, y) = \left( \frac{x}{3}, \frac{y}{3} + \frac{1}{3} \right)$$

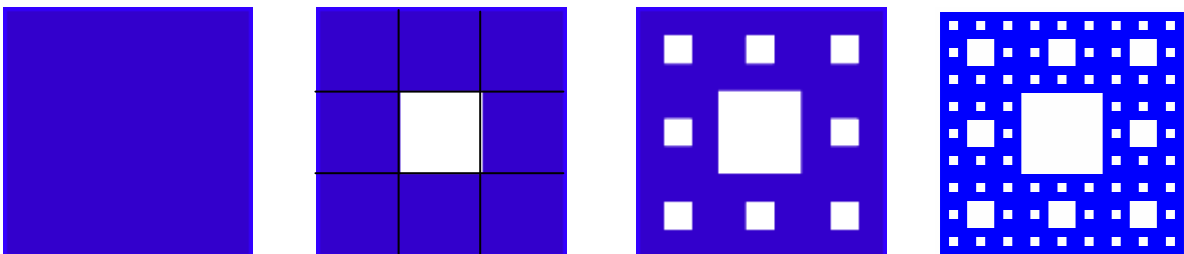
$$f_5(x, y) = \left( \frac{x}{3} + \frac{2}{3}, \frac{y}{3} + \frac{1}{3} \right)$$

$$f_6(x, y) = \left( \frac{x}{3}, \frac{y}{3} + \frac{2}{3} \right)$$

$$f_7(x, y) = \left( \frac{x}{3} + \frac{1}{3}, \frac{y}{3} + \frac{2}{3} \right)$$

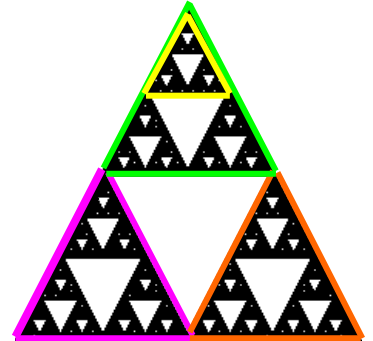
$$f_8(x, y) = \left( \frac{x}{3} + \frac{2}{3}, \frac{y}{3} + \frac{2}{3} \right)$$

We found these eight functions by observing the given images  $S_0$  and  $S_1$  and how  $S_0$  was transformed to achieve  $S_1$ . Each part of  $S_1$  was formed by either shrinking or shrinking and translating  $S_0$ . The first four images are shown below. Notice that the second image is actually eight shrunken copies of the previous image, seven of which are also translated. This is why we have eight functions to create the Sierpinski Carpet.



All of the fractals we have looked at so far share the common characteristic in that they are self-similar. An object is said to be self-similar if it looks "roughly" the same on any scale of magnification. We can choose a small part of the image and it will look very similar to the whole image. For example, consider the Sierpinski Triangle. Zoom in on a section, say the pink region. It is a miniature duplicate or a copy of the whole triangle, as is the yellow region. It is self-similar. In fact, all fractals share this characteristic of self-similarity.

Many objects in nature also have the self-similarity property as we will see later.



### C. Fractals in Nature

This was not Barnsley's only contribution in the area of fractals. Let's say you want to store a picture of a fractal. That would take much storage space if we saved the information by the pixel. Barnsley realized there is no need to store the whole picture, just store the functions needed to create the picture. When you want the picture, run a program that iterates the functions as many times as you choose, and you will get a picture like the fractal. Barnsley found and used the four functions necessary to generate a very natural-looking image of a fern, called the Barnsley fern.



This prompted others to try their hand at generating all sorts of natural images. Believe it or not, the images below are actually computer-generated with the use of fractals, and not taken with a powerful camera.

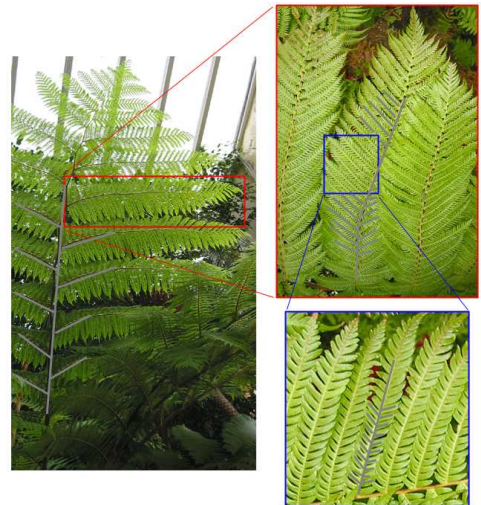




It has also become increasingly popular to use fractal technology to create computer-generated special effects.

Nature holds an array of items that have fractal properties. Among them are a fern and romanesco.

Notice that the left picture is a picture of an entire fern. But, look closer at one stem (in the upper-right picture). Doesn't it look like a miniature version of the whole fern? Then, look at one of the fronds. It, too, looks like a small version of the stem, and an even smaller version of the whole fern. This is an example of how fractals display self-similarity.



Basically, self-similarity is a characteristic of fractals in that a fractal looks the same, or very similar, when you zoom in and magnify it.

To the right is a picture of a broccoli-like plant called a romanesco. You can see how it, too, displays self-similarity as you imagine zooming in on it.



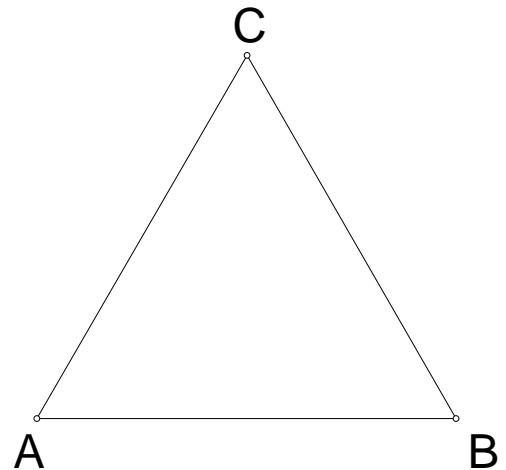
Trees, clouds, snowflakes, mountains, and rivers are some other naturally occurring fractals.

To understand self-similarity further, we can think how self-similarity is displayed in the Sierpinski Triangle. The function  $f_1$  shrinks the exact previous image and places it in the lower-left corner. The function  $f_2$  shrinks the previous image and places it in the lower-right corner. The function  $f_3$  shrinks the image and places it in the upper-middle area. Since the functions are iterated infinitely, it is understandable how we could magnify any section of the Sierpinski Triangle and see the Sierpinski Triangle. For more information about self-similarity, visit <http://math.bu.edu/DYSYS/chaos-game/node5.html>.

#### D. The Chaos Game

Earlier, we created the Sierpinski Triangle using functions. However, can we also create it using a different type of process? Perhaps we can generate the Sierpinski Triangle by random luck. This can be illustrated by the Chaos Game.

To play the Chaos Game, start with an isosceles triangle with the vertices labeled A, B, and C.

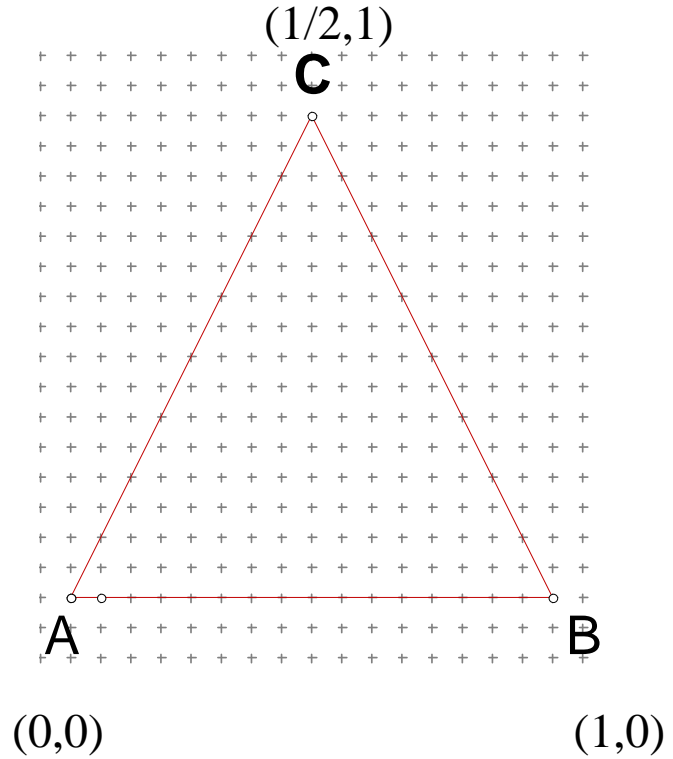


We will now assign each vertex numbers of the die. Vertex A will be the numbers 1 or 2. Vertex B will be the numbers 3 or 4. Vertex C will be the numbers 5 or 6. We first start playing the game by picking a point. Let's make our starting point C. Whatever number

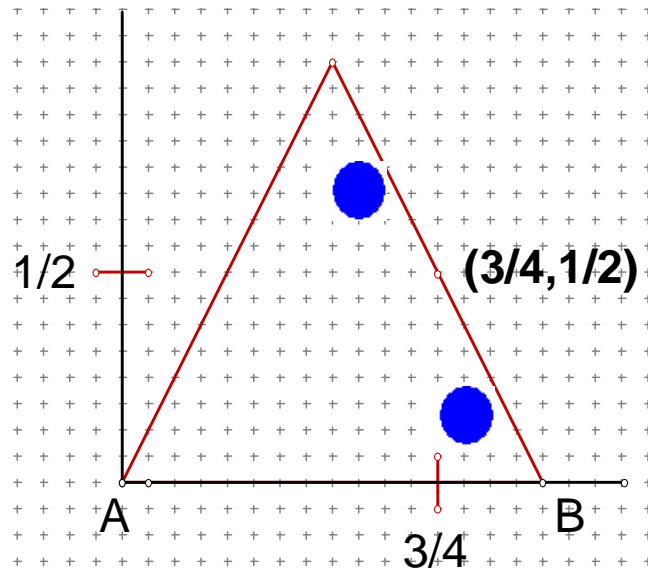
we roll we will move half the distance toward the numbered vertex and plot a point. We will remain at that point and roll the die again. We will then move half the distance toward the numbered vertex and plot a point and remain there. We will repeat the process over and over and over. Actually, we will repeat this process an infinite number of times. We might hypothesize what image we will get in the end. Since the dots are being generated randomly, we may think that we should end up with dots that are scattered chaotically.

We could now sit down and play the game. We will need a die, graph paper, a ruler, and a pencil. Let's pick the starting point **C** and start playing the game.

We can use a 16 x 16 grid with vertices at  $(0,0)$ ,  $(1,0)$ , and  $(1/2,1)$ .



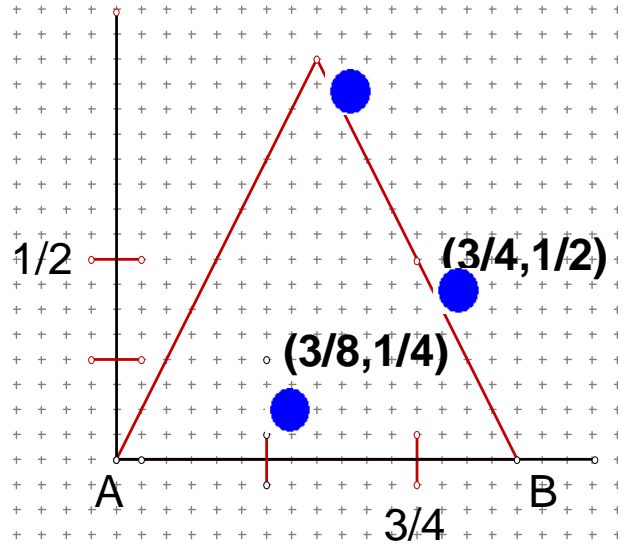
We can now roll the die and take note of what number we rolled. In this case, we rolled a three so we will use a ruler to measure half the distance from C toward B and plot a point and remain there.



Notice that we landed at  $(3/4, 1/2)$ .

We can now roll again and repeat the same process of the Chaos Game.

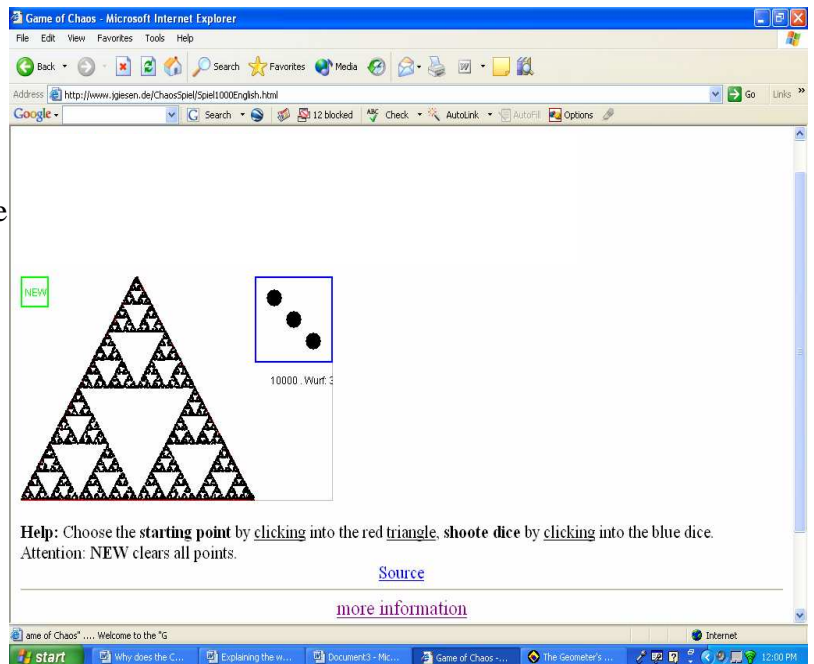
This time we rolled a two and we will measure half the distance toward A and plot a point and remain there. Notice that we landed at  $(3/8, 1/4)$ .



We can now see that this process of rolling the die and measuring with a ruler may take us more time than we have to devote to playing the Chaos Game. In addition to using a ruler to find the midpoint, we could also find the mean of the coordinates of the point we are at and the point we are moving toward. However, this would also take a great deal of time. Even if we roll the die 300 times and plot 300 points, we may not see a pattern. Now is when computers are a great asset to our goal of understanding the Chaos Game. Computers can be programmed easily to plot the points for us and we can take advantage of the available technology. Let's use the computer to jump ahead and see what image we get after 1,000 rolls of the die.

The following web page came from <http://www.jgiesen.de/ChaosSpiel/Spiel1000English.html> .

After 10,000 rolls of the die, we have a surprising image which seems to be approaching the exact Sierpinski Triangle





It's amazing that a random process actually generates a pattern. In fact, this Chaos Game was programmed to generate the Sierpinski Triangle. If we were to play the Chaos Game and roll the die an infinite number of times we would generate exactly the Sierpinski Triangle.

In addition to playing the Chaos Game on-line, we can also program our TI84 calculator to run the Chaos Game for a certain number of times, such as 5,000. Below is the program that we can enter into our calculator.

PROGRAM: SIEPINS	:If 1/3<N and N≤2/3
:FnOff	:Then
:ClrDraw	:.5(.5 +X)→X
:PlotsOff	:.5(1 + Y)→Y
:AxesOff	:End
:0→Xmin: 1→Xmax	:If 2/3<N
:0→Ymin: 1→Ymax	:Then
:rand→X: rand→Y	:.5(1 + X)→X
:For (K, 1, 5000)	:.5Y→Y
:rand→N	:End
:If N≤1/3	:Pt – On (X, Y)
:Then	:End
:.5X→X	:StorePic Pic6
:.5Y→Y	
:End	

Again we see the Sierpinski Triangle being generated on our calculator.

Let's now try to think about why this process generates the Sierpinski Triangle. Remember that I was given the three functions at the beginning of my project. The functions were  $f_1(x, y) = (x/2, y/2)$ ,  $f_2(x, y) = (x/2 + 1/2, y/2)$ , and  $f_3(x, y) = (x/2 + 1/4, y/2 + 1/2)$ . We'll be keeping those functions in mind as we try to figure out why the Chaos Game works.

We will again start with an isosceles triangle with vertex A at the point (0, 0), vertex B at the point (1, 0), and vertex C at the point (1/2, 1). We will choose to start at a vertex, such as the vertex at the point C (1/2, 1) and roll the die. We rolled a three which tells us to plot a point half the distance to vertex B. We will plot the point (3/4, 1/2). Notice that our point is on the Sierpinski Triangle. We can now observe that this is the same point we would have gotten if we would have applied the function  $f_2(x, y) = (x/2 + 1/2, y/2)$  to our starting point.

$$f_2(1/2, 1) = (1/4 + 1/2, 1/2)$$

$$f_2(1/2, 1) = (3/4, 1/2)$$



We can now roll the die a second time. We rolled a 2 so we will plot a point half the distance from our last point to vertex A. Our point lands at  $(3/8, 1/4)$ . Notice again that our point is on the Sierpinski Triangle. We can again observe that rolling a one or a two means that we are applying the function  $f_1(x, y) = (x/2, y/2)$ .

$$\begin{aligned} f_1(x, y) &= (x/2, y/2) \\ f_1(3/4, 1/2) &= (3/8, 1/4) \end{aligned}$$

If I roll a five or six, then I would apply the function  $f_3(x, y) = (x/2 + 1/4, y/2 + 1/2)$  and again land on the Sierpinski Triangle.

$$\begin{aligned} f_3(x, y) &= (x/2 + 1/4, y/2 + 1/2) \\ f_3(3/8, 1/4) &= (3/16 + 1/4, 1/8 + 1/2) \\ f_3(3/8, 1/4) &= (7/16, 5/8) \end{aligned}$$

This process of applying one of the three functions happens every time we roll the die. That is, rolling the die tells us which one of the three functions to apply. If we roll a one or a two, the function  $f_1$  is applied. If we roll a three or a four,  $f_2$  is applied. If we roll a five or a six,  $f_3$  is applied. Using a die gives us a random process of choosing which function to apply. After an infinite number of rolls, each function is used one-third of the time. This guarantees that each part of the Sierpinski Triangle gets filled in. Our process of generating dots is random, but the dots we make are part of the Sierpinski Triangle.

Additionally, another way to convince ourselves why the Chaos Game works is that to notice that if we pick a point in the Sierpinski Triangle and apply one of the three functions we will always get a point in the Sierpinski Triangle. Every point on the Sierpinski Triangle has a pre-determined outcome of the Sierpinski Triangle. Although this is believable, it is very difficult to prove and the proof isn't necessary for our understanding of the Chaos game so we may take it on faith.

Conversely, if we start in a cut out section and roll the die an infinite number of times, we will always land on a cut out section. If we start in the largest cut out triangle, the points will eventually lie in successively smaller removed triangles. Actually, the removed triangles quickly become so small in size that it is essentially invisible. Starting in a cut out section will never allow us to land on the Sierpinski Triangle.

We can also explain the Chaos game in a slightly different way. We saw that when making the Sierpinski Triangle earlier we applied three functions. The first one shrunk the image and put it in the lower left area. The second one shrunk the image and put it in the lower right area. The third one shrunk the image and put it in the top middle area. This is just like the Chaos Game except instead of starting with the image and plotting the image we are starting with a point and plotting the point. Remember that we are using a random process with the die which means that due to probability we will plot points one-third of the time in the lower left area and one-third of the time in the lower right area and one-third of the time in the top middle area. This means that after an infinite number of times of rolling the die and plotting points we would have the Sierpinski Triangle.

Fractals are a branch of chaos theory and we used the Chaos Game to try to understand how chaos theory works. The roots of Chaos Theory date back to about 1900 when the mathematician Henri Poincaré was working with the three-body problem. While working with the three-body problem, he became the first person to discover a chaotic deterministic system. His work laid the foundation for Chaos Theory. In 1960, the meteorologist Edward Lorenz became the first true pioneer to work with Chaos Theory. His interest in Chaos Theory came about accidentally as he was studying weather patterns.

The Chaos Game is a very fast way to generate fractals. For a computer to generate a fractal by applying the functions used to generate the fractal it would take a great deal of time. For example for a computer to generate the Barnsley Fern it would require a full day or more. However, using the Chaos Game to generate the Barnsley Fern requires a second or two to see an image that for all intents and purposes looks like the final Barnsley Fern.

Interestingly, chaos theory is used to make sense of and study natural phenomena that recently was thought to be pattern-less and incapable of being described using mathematical modeling. Some natural phenomena that we now use chaos theory to study are weather patterns, the occurrences of earthquakes, and fluctuations in the stock market.

In Chaos Theory, systems that show mathematical chaos are deterministic which means that they actually have a determined outcome despite the fact that they appear to be complicated and random. Chaos uses determinism which is the philosophical belief that every event is the inevitable outcome of a previous event. Historically, determinism dates back several thousand years while Chaos Theory is fairly recent. With Chaos Theory a random event produces a specific determined outcome. Chaos Theory was used as a central role in the movie with Ashton Kutcher titled *The Butterfly Effect*. The movie got its title from the idea of the butterfly effect which is where the flapping of a butterfly is imagined to produce some effect to the atmosphere that eventually leads to some dramatic event such as a tornado.

Chaos Theory is also portrayed in the book by Ray Bradbury titled *Sound of Thunder* which used Chaos Theory as a central role. This was made into a movie of the same name that was released in 2005. In the book, adventurers pay money to time travel back to the time of dinosaurs for a safari. The time travel company is very strict that the customers do not step off the path. However, at one point one customer steps off the path slightly and squishes a moth. Then, when they return to present time, the world as they knew it is drastically different. For instance the United States is now under the leadership of a ruthless dictator. The outcome of killing that one specific moth produced a specific determined outcome of the dictatorship. This is saying a random event (killing the moth) produces a specific determined outcome (the dictatorship). This is similar to the playing the Chaos Game where the random event of rolling the die an infinite number of times produces a specific determined outcome of the Sierpinski Triangle. Also, every point on the Sierpinski Triangle has a specific determined outcome of the Sierpinski Triangle.

When playing the Chaos Game it looks complicated and random, but the outcome is pre-determined. Chaotic systems are complicated and their behavior appears random, but they are really deterministic. Chaotic systems don't always have to produce a pattern, though they do in the end in the Chaos Game as it generates the Sierpinski Triangle. Essentially, we used the Chaos Game to understand how Chaos Theory works.

### Summary

Fractals and the Chaos Game was an amazing topic for me. I gained an incredible amount of knowledge working through the problems. I am also very excited to use much of what I learned with my students. For example, I have an idea of having my students play a year-long Chaos Game in which each class rolls the die two times a day for a total of six rolls in a day.

In prior years I have had my students create the Sierpinski Triangle using the process of connecting midpoints and cutting out triangles. Before working with my research problem, I had no connection between functions and fractals. Through my research I learned to use functions to generate fractals. I also learned that a specific fractal would be generated no matter what shape you initially started with as long as you apply the specific functions. I am also now able to determine functions for some simple fractals.

When I stumbled upon some fractal landscapes I was amazed that the images were not actual pictures of nature. Fractal "art" became an interesting topic to me that I would like to share with my students. The fact that fractals that are naturally occurring in nature is also something that I had not thought much about prior to working with my research problem. Fractals truly became beautiful images to me through creating fractals and looking at fractal landscapes and fractals in nature.

When I first started working with the Chaos Game, I was blown away by its magical qualities. It seemed an impossible feat to prove that it wasn't working due to magic. However, after hours upon hours of toiling with the game, I began to see how the functions used to create the Sierpinski Triangle were being used. My understanding of why the game works was solidified when I thought of the functions being applied to the points rather than when we first created the Sierpinski Triangle and applied the functions to the image.

At that point, I thought I was completely done. However, for some reason I decided to start thinking about Chaos Theory. Again, I was blown away. The wonderful part about delving into understanding Chaos Theory was that I came to understand how working with the Chaos Game is a useful tool for understanding Chaos Theory. Also, working to understand Chaos Theory helped me to understand that every point on the Sierpinski Triangle has a specific determined outcome of the Sierpinski Triangle. While working to understand Chaos Theory, I was really pleased with the use of Chaos Theory in Ray Bradbury's *Sound of Thunder* and in the movie titled *Butterfly Effect*. Thinking about this book and this movie helped me to understand what Chaos Theory is about. I even

was able to think about how Chaos Theory has played out in my own life and how every event in my life has produced the outcome of where I am now at this point in time.

Overall, the experience of researching fractals and the Chaos Game was excruciating but incredibly wonderful. When it all came together for me I felt like I had never felt before, superhuman.

## References

- Applications of fractals*. Retrieved on July 17, 2006, from <http://library.thinkquest.org/26242/full/ap/ap12.html>
- Barnsley, M., Hutchinson, J., & Stenflo, O. *V-variable fractals and superfractals*. Retrieved on July 19, 2006, from [http://www.maths.anu.edu.au/~barnsley/pdfs/V-var\\_super\\_fractals.pdf](http://www.maths.anu.edu.au/~barnsley/pdfs/V-var_super_fractals.pdf)
- Beck, Alan. *What is a fractal?* Retrieved on July 18, 2006, from [http://www.glyphs.com/art/fractals/what\\_is.html](http://www.glyphs.com/art/fractals/what_is.html)
- Bourke, Paul. *Fractals, chaos*. Retrieved on July 17, 2006, from <http://astronomy.swin.edu.au/~pbourke/fractals/>
- Brown, Adam. (2005). *Fractal landscapes*. Retrieved on July 11, 2006, from <http://www.fractal-landscapes.co.uk/images.html>
- Burger, E. B. and Starbird, M. (2000). *The heart of mathematics: an invitation to effective thinking*. Emeryville: Key College Publishing
- Chaos Theory*. Retrieved on July 22, 2006, from [http://www.reference.com/browse/wiki/Chaos\\_theory](http://www.reference.com/browse/wiki/Chaos_theory)
- Collins, W, et al. (1998). *Glencoe algebra 2: integration applications connections*. Westerville: The McGraw-Hill Companies.
- Fractal ferns*. Retrieved on July 20, 2006, from <http://www.home.aone.net.au/~byzantium/ferns/fractal.html>
- Fractals*. Retrieved on June 25, 2006, from <http://webweevers.com/fractals.htm>
- Fractals*. Retrieved on July 17, 2006, from [http://math.youngzones.org/Fractal%20webpages/fractal\\_applications.html](http://math.youngzones.org/Fractal%20webpages/fractal_applications.html)
- Fractals*. Retrieved on July 16, 2006, from [http://motivate.maths.org/conferences/conf77/c77\\_investigating\\_fractals\\_harder.shtml](http://motivate.maths.org/conferences/conf77/c77_investigating_fractals_harder.shtml)
- Fractals and the chaos game*. Retrieved on July 17, 2006, from <http://wps.aw.com/wps/media/objects/52/54247/lessons/frac3.html>
- Panorama of fractals and their uses*. Retrieved on July 18, 2006, from <http://classes.yale.edu/fractals/Panorama/Nature/NatFracGallery/NatFracGallery.html>

*Playing the chaos game in class.* Retrieved on July 18, 2006, from <http://math.bu.edu/DYSYS/chaos-game/node4.html>

*Self-similarity.* Retrieved on July 19, 2006, from <http://math.bu.edu/DYSYS/chaos-game/node5.html>

*Sierpinski gasket.* Retrieved on July 17, 2006, from <http://planetmath.org/encyclopedia/SierpinskiTriangle.html>

Spencer, Philip. (1999). *Fractals and their history.* Retrieved on June 25, 2006, from <http://www.math.toronto.edu/mathnet/questionCorner/fracthist.html>

*The game of chaos.* Retrieved on July 15, 2006, from <http://www.jgiesen.de/ChaosSpiel/ChaosEnglish.html>

Walker, John. (2005). *Fractal food.* Retrieved on July 19, 2006, from <http://www.fourmilab.ch/images/Romanesco/>

Winter, Dale. *Fractals.* Retrieved on July 20, 2006, from <http://www.math.lsa.umich.edu/mmss/coursesONLINE/chaos/chaos7/index.html>