

# LaSalle Academy Fractal Workshop – October 2006

## Fractals Generated by Geometric Replacement Rules

### Sierpinski's Gasket

- Press 't'
- Choose 'lssystem' from the menu
- Choose 'sierpinski2' from the menu
- Make the order '0', press 'enter'
- Press 'F4' (This selects the video mode. This mode usually works well. *Fractint* will offer other options if you press the delete key. 'Shift-F6' gives more colors and better resolution, but doesn't work on every computer.)

You should see a triangle.

- Press 'z', make the order '1', 'enter'
- Press 'z', make the order '2', 'enter'
- Repeat with some other orders. (Orders 8 and higher will take a LONG time, so don't choose numbers that high unless you want to wait.)

**Things to Ponder** *Fractint* repeats a simple rule many times to generate Sierpinski's Gasket. The repetition of a rule is called an *iteration*. The order 5 image, for example, is created by performing 5 iterations. The Gasket is only complete after an infinite number of iterations of the rule. Can you figure out what the rule is?

One of the characteristics of fractals is that they exhibit self-similarity on different scales. For example, consider one of the filled triangles inside of Sierpinski's Gasket. If you zoomed in on this triangle it would look identical to the entire Gasket. You could then find a smaller triangle inside this triangle that would also look identical to the whole fractal. Sierpinski imagined the original triangle like a piece of paper. At each step, he cut out the middle triangle. How much of the piece of paper would be left if Sierpinski repeated this procedure an infinite number of times?

## Von Koch Snowflake

- Press 't', choose 'lssystem' from the menu, choose 'koch1'
- Make the order '0', press 'enter'
- Press 'z', make the order '1', 'enter'
- Repeat for order 2
- Look at some other orders that are less than 8 (8 and higher orders take more time)

**Things to Ponder** What is the rule that generates the von Koch snowflake?

Is the von Koch snowflake self-similar in any way? Describe how.

Since it is possible to draw a circle around the von Koch snowflake, it has a finite area. Keeping in mind that the von Koch snowflake is created by an infinite number of iterations, what can you say about the perimeter (the distance around the outside) of the snowflake?

## Fractal Bush

- Press 't', choose 'lssystem', choose 'bush'
- Make the order '0', press 'enter'
- Press 'z', make the order '1', 'enter'
- Look at orders 2, 3, 4, and 5 (Orders 6 and higher take a long time)

**Things to Ponder** Can you describe the rule that generates the bush?

In what way is the fractal bush self-similar?

Can you think of any other objects or processes in nature which exhibit self-similarity on different scales? Can you think of anything that might be formed by the repetition of a rule?

# Iterating Real Functions

## Linear Functions

Consider the function  $f(x) = \frac{1}{2}x$ . This function defines a dynamical system on the number line because every time the function acts, it causes the points on the number line to move in a certain way. We would like to understand the overall effect of the function if we apply it to the number line over and over again. We would like to understand how points on the number line will travel and where they will end up. Applying a function over and over again is called “iteration”.

Let's consider what happens to the number 2 if we apply the function  $f(x)$  to it. Since 2 is our starting number and we have not applied the function to it yet, we say that it is our starting  $x$ , or  $x_0 = 2$ .

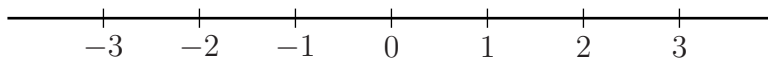
$$\begin{aligned}f(x) &= \frac{1}{2}x \\x_1 = f(2) &= \frac{1}{2}(2) \\&= 1\end{aligned}$$

So our first iteration of the function gives us  $x_1 = 1$ . Let's apply the function a few more times to find more iterates.

$$\begin{aligned}x_2 &= f(1) = \frac{1}{2}(1) = \frac{1}{2} \\x_3 &= f\left(\frac{1}{2}\right) = \frac{1}{2}\left(\frac{1}{2}\right) = \frac{1}{4} \\x_4 &= f\left(\frac{1}{4}\right) = \frac{1}{2}\left(\frac{1}{4}\right) = \frac{1}{8} \\x_5 &= f\left(\frac{1}{8}\right) = \frac{1}{2}\left(\frac{1}{8}\right) = \frac{1}{16}\end{aligned}$$

Plot these iterates on the number line. How are the iterates moving? What would happen if we kept iterating?

Choose some other starting values for  $x_0$ . What happens when you iterate these values? What can you say about the overall dynamic effect of the function  $f(x) = \frac{1}{2}x$ ?



What happens for other linear functions such as  $f(x) = \frac{1}{2}x + 1$ ,  $f(x) = 2x$ ,  $f(x) = -2x$ ,  $f(x) = -\frac{1}{2}x$ ?

## Quadratic Functions

Consider what happens use this rule starting with the number 1.1.

$1.1^2 = 1.21$	$445.7915685^2 = 198730.1225$
$1.21^2 = 1.4641$	$198730.1225^2 = 3.949366159 \times 10^{10}$
$1.4641^2 = 2.14358881$	$(3.949366159 \times 10^{10})^2 = 1.559749306 \times 10^{21}$
$2.14358881^2 = 4.594972986$	$(1.559749306 \times 10^{21})^2 = 2.432817897 \times 10^{42}$
$4.594972986^2 = 21.11377675$	$(2.432817897 \times 10^{42})^2 = 5.918602921 \times 10^{84}$
$21.11377675^2 = 445.7915685$	$(5.918602921 \times 10^{84})^2 = \text{too big for calculator}$

The numbers become big very quickly. The iterates are “escaping to infinity.”

Now let's repeat this procedure with 0.9.

$0.9^2 = 0.81$	$0.001179018^2 = 0.00000139$
$0.81^2 = 0.6561$	$0.00000139^2 = 1.932334983 \times 10^{-12}$
$0.6561^2 = 0.43046721$	$(1.932334983 \times 10^{-12})^2 = 3.733918487 \times 10^{-24}$
$0.43046721^2 = 0.185302018$	$(3.733918487 \times 10^{-24})^2 = 1.394214727 \times 10^{-47}$
$0.185302018^2 = 0.034336838$	$(1.394214727 \times 10^{-47})^2 = 1.943834705 \times 10^{-94}$
$0.034336838^2 = 0.001179018$	$(1.943834705 \times 10^{-94})^2 = \text{too small for calculator}$

This time, the numbers became smaller. The iterates are “bounded” — they do not escape to infinity. Even though 1.1 and 0.9 are very close together, the replacement rule sends iterates of 1.1 towards infinity and sends iterates of 0.9 towards zero.

## Imaginary and Complex Numbers

Julia sets are created using *complex* numbers. Complex numbers have a real part and an imaginary part added together. Some examples of complex numbers are: **A)**  $-1.2 + 1.01i$ , **B)**  $1 + 2i$ , **C)**  $-1.3 - 0.8i$  **D)**  $0.43 - 0.12i$ . We can plot these complex numbers on a graph by letting the horizontal direction show the real part and the vertical direction show the imaginary part.

See page with graph.

# Julia Sets

- Press ‘x’ - the basic options screen should appear (If it doesn’t, press ‘escape’ to return to the main menu and then press ‘x’)
- Change ‘Inside Color’ to ‘0’ (this will make the inside of the fractals black), press ‘enter’
- Press ‘t’, choose ‘julia’
- Change the real parameter to ‘0.32’ and the imaginary parameter to ‘0.04’ and press ‘enter’

**What’s Going On?** The replacement rule which generates this Julia set is  $new\ z = z^2 + 0.32 + 0.04i$ . We usually call the complex number at the end  $c$  for short.

1. The computer begins with the point in the upper left corner of the screen. It uses the complex number corresponding to that point (might be  $z = -2 + 2i$ , for example).
2. The computer squares this number and adds  $0.32 + 0.04i$  to get the *new*  $z$ .
3. On the next iteration it takes the *new*  $z$ , squares it, and adds  $c$ .
4. The computer repeats this process until the iterates either become “large” (bigger than 2), which means they are escaping to infinity, or until the operation has been performed a predetermined number of times.
5. The computer assigns a color to the point based on how many iterations it takes for the number to escape.
6. The computer then proceeds to the next point on the screen and begins again.
  - Blue background points escaped after the first 2 iterations.
  - Points in the green band escape after 3 iterations.
  - The light blue band shows the points that escape after 4 iterations.
  - Points that are bounded, that never escape, are colored black.

There are as many color bands as there are iterations. *Fractint*’s default number of iterations for Julia sets is 150.

## A Dust Set

- Press ‘z’ to change the  $c$ -value of the Julia set
- This time set the real part equal to 0.3 and the imaginary part to 0.6
- Press ‘enter’

Notice that this fractal has broken up into isolated “island” color bands. This happens because there is no connected black region around which they are centered. To see this better, we need to look at the fractal in black and white.

- Press 'x'
- Change 'Outside Color' to '1', Change 'Maximum Iterations' to '5', press 'enter'
- Press 'F8' to switch to black and white mode

All of the escape bands are now white. The black region represents the points that do not escape within 5 iterations. After 5 iterations, the black region is still connected.

- Press 'x', change 'Maximum Iterations' to '10', press 'enter'
- Repeat this procedure for 20 and 30 iterations

Notice that the black region is no longer connected after 20 iterations. After 30 iterations, the fractal begins to look like dust specks. Eventually, all of the black region will blow away. Julia sets like  $new\ z = z^2 + 0.3 + 0.6i$  are called "dust sets."

**Things to Ponder** What is the highest number of iterations that you can choose so that the black region is still connected?

Give an example of a maximum iteration value that appears to remove all of the black points leaving only the white background.

## Douady's Rabbit

- Press 'z' to change the c-value of the Julia set
- This time set the real part equal to -0.122 and the imaginary part to 0.745, press 'enter'
- Press 'x', set 'Maximum Iterations' to '5'
- Look at other maximum iteration settings

Notice that Douady's Rabbit stays connected no matter how many iterations are performed.

**Things to Ponder** Do you think that this Julia set,  $new\ z = z^2 + -0.122 + 0.745i$ , looks like a rabbit?



## Zooming Around

- Press 'z' to generate a Julia set with a new c-value
- Set the real part equal to any number between -2 and 1.5
- Set the imaginary part equal to any number between -1.25 and 1.25

Is your fractal connected for all maximum iteration settings?

## Putting the colors back

- Press 'x'
- Change 'Outside Color' to 'iter', Change 'Maximum Iterations' to at least 150, press 'enter'
- Press 'F4' to switch to color mode

## To zoom in on your fractal

- Press 'page up' until the box is the right size ('page down' makes the box bigger)
- Use the mouse to move the box to the right place and press 'enter'

*Fractint* makes three passes over the fractal making it less fuzzy each time. It beeps when it is done. The farther you zoom in, the longer this takes. You can either wait for the image to become clear, or you can press 'page up' and zoom in farther before it is done.

To zoom back out, press 'h' until you are back where you want to be. 'Ctrl-h' takes you back in if you zoom out too far.

**Things to Ponder** Is the Julia set that you chose self-similar? That is, do you find the same types of structures as you zoom in? Do you find different structures if you zoom in different places?

## Mandelbrot Set

We have seen that all Julia sets are generated by a rule of the form  $new\ z = z^2 + c$  where  $c$  is a complex number. Some values of  $c$  generate connected Julia sets (like Douady's Rabbit) while others generate dust sets. The Mandelbrot set is a fractal that serves as a sort of "dictionary" for the Julia sets. Among other things, it shows which values of  $c$  yield connected Julia sets.

- Begin with a Julia set on the screen
- Press the space bar to toggle to the Mandelbrot set
- After the beep, press the space bar again to see where in the Mandelbrot set your Julia came from (the cursor shows this). The outline of your Julia set appears in the little box.
- Move the cursor to the middle of the black region, press the space bar and look at the Julia
- Press the space bar once to return to the Mandelbrot set
- After the beep, press the space bar again to display the Julia inset
- Put the cursor on the blue background and look at the Julia
- Put the cursor near the boundary of the black region and examine that Julia

**Things to Ponder** Where on the Mandelbrot set do the dust Julia sets come from? The connected Julia sets? What are the Julia sets near the boundary like?

Move the cursor slowly around the Mandelbrot set, stopping the mouse every so often to look at the shape of the Julia set. Press the space bar to look at the Julia sets more closely. Find your favorite Julia set. Give it a name and record its formula. To find the c-value, press 'z' while the Julia is on the screen. Remember that Douady's Rabbit,  $new\ z = z^2 + -0.122 + 0.745i$ , already has a name!

Name:

Formula:

Try changing the number of iterations to see how your Julia set looks best. Press 'x' for basic options, and change maximum iterations.

Zoom in on your Julia set. Record your favorite window. (Hit tab and copy the numbers of the corners.)

Corners:

## Other Neat *Fractint* Tricks

You can also experiment with changing the inside and outside colors. (Under basic options. Type a number or one of the other options it gives you). Other things to play with are log palette, biomorph color, and decomp option.

For a thrill press '+' or '-' to start color cycling (press Escape to stop color cycling, numbers

on the right key pad change the speed)

Press 'ctrl-a' to have your fractal eaten by the ant automaton.

Press '@' for a list of fractal art.

You can change the color palate used on a particular fractal (press 'c' then 'l'), or edit individual colors of a color palate (press 'e' - 'F1' will show you additional color editing features).

The 3D transforms and overlays are lots of fun too.

Try different settings in each of the options screens ('x', 'y', 'z'), be artistic and creative, have fun!

### **Further Information on *Fractint***

*Fractint* is a freeware program created by the Stone Soup Group. More advanced versions of the program can be found on the Internet as they come out.

*Fractint's* homepage is at **<http://spanky.triumf.ca/www/fractint/fractint.html>**

The *Fractint* web site has a Fractal Screen-Saver, and several add-ons for *Fractint*. It also has information for creating your own formulas, and customizing your copy of *Fractint*.