

EE 356

Fractal notes

If you double the length of the side of a square the area increases by a factor of 4. Likewise, if you double the radius of a circle the area goes up by a factor of 4. But if you double the length of a side of a fractal the area increases from some fraction – say 1.6 or so. It is said to have a fractional dimension.

There is a *Fractal Foundation* which is geared to K-12 education and produces a very nice guide to various kinds of fractals along with some equations for producing them. The guide is at <http://fractalfoundation.org/fractivities/FractalPacks-EducatorsGuide.pdf>

The Sierpinski Triangle is given as an example in the guide. It starts with a triangle and continuously removes the inside triangle from all of the triangles that appear. The process of removing the inner triangle can go on forever and that is one of the characteristics of fractals.

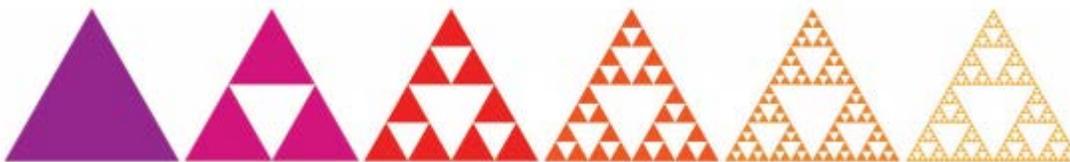


Figure 1

The Sierpinski Triangle fractal.

Mandelbrot Fractal

It's also easy to generate fractals from equations. The most famous of these is the *Mandelbrot Set* which is generated by the equation:

$$z_{n+1} = z_n^2 + c$$

In this equation z and c are complex numbers. To generate an image you work with points (pixels) on the complex plane. Allow $z_0 = 0$. Choose a point on the complex plane and assign its real and imaginary parts to the constant c . Iterate through the equation and monitor the results. The Mandelbrot set is the set of all values for which the equation converges to zero. You can produce an image by assigning colors based on how fast the series diverges to infinity. Suppose we have 64 colors ranging from white (zero) to black (63). Since we can't get to infinity we pick some suitable maximum – say MAX and write a loop like similar to that shown in Figure 2. In this figure, z , $z1$, and c are complex numbers, $z.Mag()$ is a function which provides the magnitude of z , and x and y should be in *world* coordinates – not screen coordinates.

Figure 3 shows the results of creating an image with 64 colors as x goes from -1.5 to +1.5 and y goes from about -0.91 to +0.91. This was created on an image that is approximately 1024 x 768 in screen pixels. You can re-center the screen over certain parts of the Mandelbrot set and expand the view until you eventually run out of numerical precision. Figure 4 shows an expansion of one of the small "bulbs" in Figure 3.

```

for(y=0;y<scrHeight;y++)
  {for(x=0;x<scrWidth;x++)
    {c.Real = x;c.Imag = y;
    count = 0;z = 0;
    while(count < 64 && z.Mag() < MAX)
      {z1 = z*z + c;
      z = z1;
      count++;
      }
    colorNum = count % 64;
    //Save the colorNum in an array
  }
}

```

Figure 2

A loop to generate the Mandelbrot set

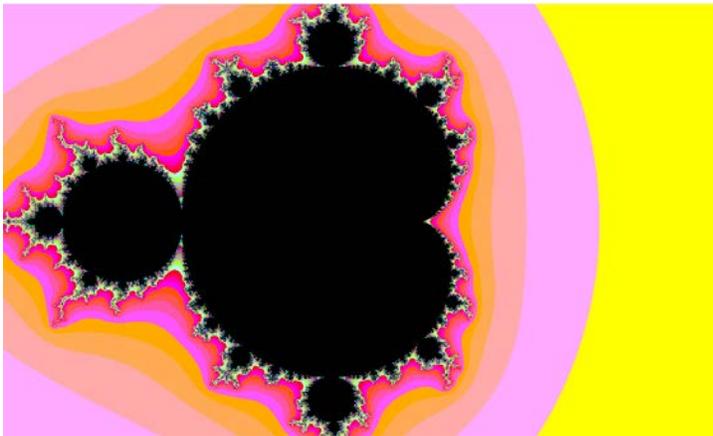


Figure 3

The Mandelbrot set. In this figure, the center is at (0, 0) and the right and left sides are at ± 1.5 in x. The top and bottom are at about ± 0.91 .

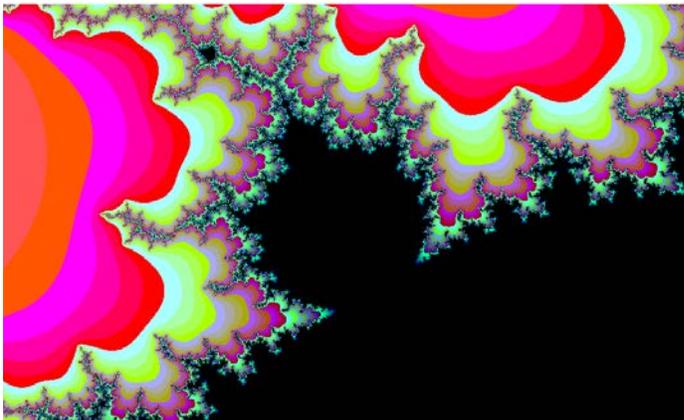


Figure 4

This is an expansion of one of the small "bulbs" at the top left of Figure 3. The width of the figure is about 0.25 in world coordinates.

Julia Fractal

For the Mandelbrot set we used the equation $z_{n+1} = z_n^2 + c$ and c was a coordinate in the x-y plane. The Julia set uses the same equation but c is some constant – usually we take this to be -0.75 and z_n is a coordinate in the x-y plane. Figure 5 shows the Julia fractal set.

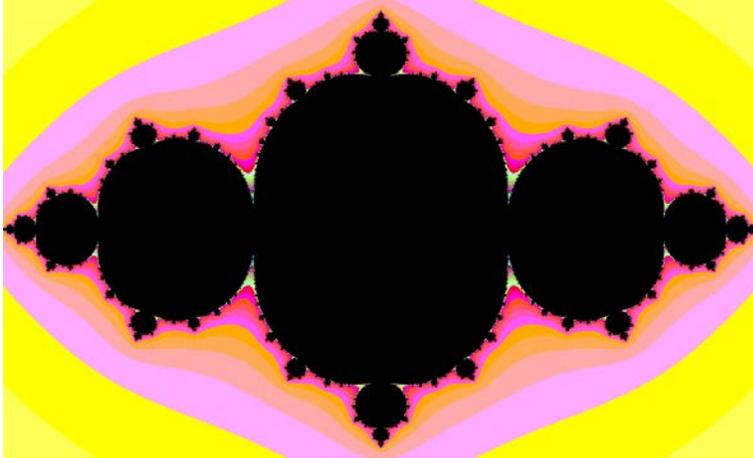


Figure 5
The Julia fractal set.

There are many other fractal types – maybe hundreds. Not all of them are based on escape velocity and a few of them are hard to generate on a computer. The Wikipedia article on fractals is a good place to start.