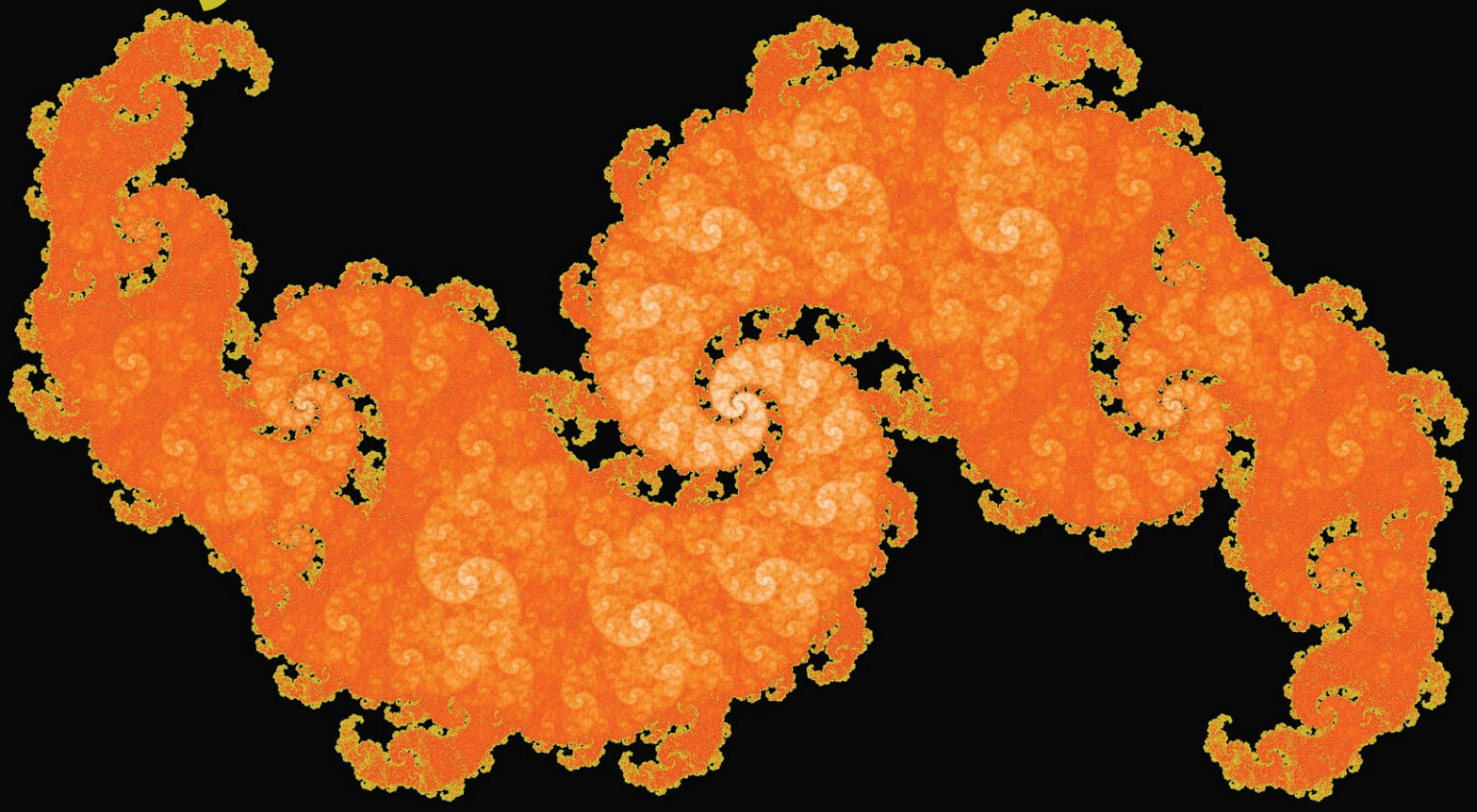
math
HORIZONS

Symmetric Fractals



Seeking *Sangaku*

Ramanujan, Hardy, and Ono

Published by the Mathematical Association of America : : November 2016

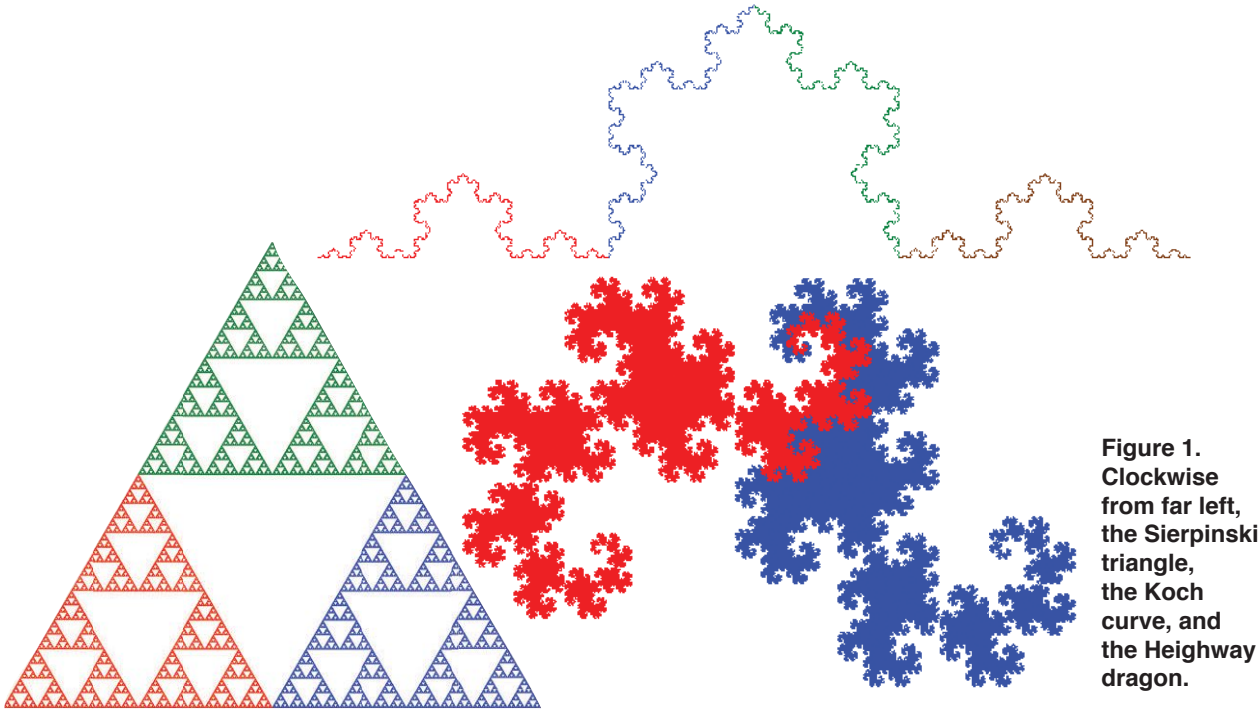


Figure 1. Clockwise from far left, the Sierpinski triangle, the Koch curve, and the Heighway dragon.

Creating Symmetric Fractals

Larry Riddle

Fractals such as the Sierpinski triangle, the Koch curve, and the Heighway dragon, shown in figure 1, are constructed using simple rules, yet they exhibit beautifully intricate and complex patterns.

All three fractals possess *self-similarity*—that is, the fractal is composed of smaller copies of itself. But only the first two fractals have symmetries. In this article we show how to use group theory, which is often used to describe the symmetries of objects, to create symmetric fractals.

Iterated Function Systems

Fractals such as those in figure 1 can be constructed using sets of functions called *iterated function systems* (IFS). The functions in an IFS have a scaling factor less than 1, a rotation, and a translation, which makes them contractive affine transformations.

For instance, the IFS for the Heighway dragon is the set of functions $H = \{h_1, h_2\}$ where h_1 is a scaling by $r = \frac{1}{\sqrt{2}}$ and a counterclockwise rotation by 45° around the origin, while h_2 is also a scaling by r but with a counterclockwise rotation by 135° and a horizontal translation by 1. The dragon is the unique set A, called the *attractor* of the IFS, that satisfies $A = h_1(A) \cup h_2(A)$. In figure 1, $h_1(A)$ is red and $h_2(A)$ is blue. Both subsets are copies of the Heighway dragon scaled by a factor of r.

Bringing in Abstract Algebra

In *Symmetry in Chaos: A Search for Pattern in Mathematics, Art and Nature* (1992, Oxford University Press), Michael Field and Martin Golubitsky showed how to generate a symmetric fractal from a single affine transformation and a *cyclic group* Z_n or a *dihedral group* D_n—groups that students encounter in an abstract algebra course. As we shall see, their method applies equally well when applied to an iterated function system.

First, we need some basic facts about Z_n and D_n. The group Z_n consists of the rotational symmetries of a regular n-sided polygon. We will let the n elements of Z_n correspond to counterclockwise rotations about the origin (which corresponds to the center of the polygon) through angles that are integer multiples of $360°/n$.

The group D_n consists of all $2n$ symmetries of a regular n-sided polygon. We take these to be the n rotations in Z_n and reflections about n lines through the origin that meet in angles that are integer multiples of $180°/n$. Figure 2 shows the lines of symmetry for an equilateral triangle and a square. The group D_2 is an exception since there is no regular polygon with two sides. It is known as the *Klein four-group*, and the four elements are the identity, a vertical reflection, a horizontal reflection, and a 180° rotation.

Given any IFS, we can form a new one by composing each element of the group (Z_n or D_n) with each function in the IFS. For instance, let $Z_2 = \{e, g\}$, where e is the identity and g is the 180° rotation about the origin. Composing elements of Z_2 with elements in the IFS for the Heighway dragon, H, yields the IFS

$$\{f_1 = e \circ h_1 = h_1, f_2 = e \circ h_2 = h_2, f_3 = g \circ h_1, f_4 = g \circ h_2\}.$$

All four functions in this IFS have the same scaling factor $\frac{1}{\sqrt{2}}$ because the only change—if any—is a 180° rotation. So, in addition to the original counterclockwise rotations of 45° and 135° from f_1 and f_2, the elements f_3 and f_4 have counterclockwise rotations of 225° and 315°. Also, f_2 and f_4 include horizontal translations by 1 and -1, respectively. The unique attractor for this new IFS, B, which we call the Z_2 Heighway dragon, satisfies $B = f_1(B) \cup f_2(B) \cup f_3(B) \cup f_4(B)$. See figure 3.

Rotating B by 180° is the same as applying g from Z_2 to the set B. Because $g \circ g = e$ is the identity and $g \circ e = g$, we have

$$\begin{aligned} g(B) &= g\big(f_1(B) \cup f_2(B) \cup f_3(B) \cup f_4(B)\big) \\ &= g \circ h_1(B) \cup g \circ h_2(B) \cup g \circ g \circ h_1(B) \cup g \circ g \circ h_2(B) \\ &= g \circ h_1(B) \cup g \circ h_2(B) \cup e \circ h_1(B) \cup e \circ h_2(B) \\ &= f_3(B) \cup f_4(B) \cup f_1(B) \cup f_2(B) = B. \end{aligned}$$

This shows that the Z_2 Heighway dragon has twofold (or 180°) rotational symmetry.

In figure 3 we see only three scaled versions of the attractor, but four sets are in the union. Because f_1 and f_3 have the same scaling factor, and they rotate B by 45° and 225° counterclockwise, respectively, and B has 180° rotational symmetry, then $f_1(B) = f_3(B)$. This is the red set.

Coloring a Fractal with Pixel Counting

One of the methods for generating the attractor of an iterated function system on a computer is to use a random algorithm known as the *chaos game*. First, we choose an initial point x_0 in the attractor. Then we choose a function f from the IFS at random (with a certain probability), and then we plot $x_1 = f(x_0)$, which lies in the attractor.

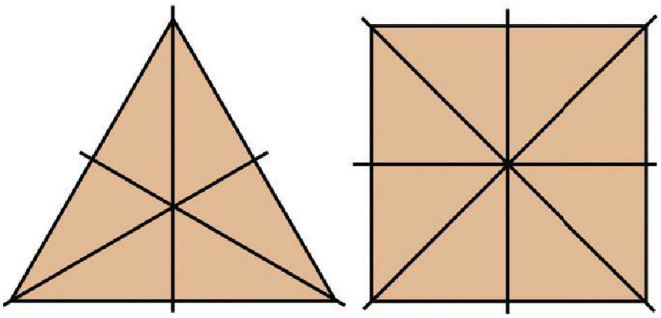


Figure 2. The groups D_3 and D_4 consist of rotations about the centers of the equilateral triangle and the square, respectively, and reflections about the lines of symmetry.

Next, we choose a function from the IFS at random again, apply it to x_1, and obtain another point in the attractor, x_2. Repeat this process of randomly choosing a function, plugging in the previous point, then plotting the new point—millions of times. The sequence of points fills out the attractor.

In fact, it is not necessary to start with a point in the attractor. Any point suffices as long as we wait long enough during the iterative process to start plotting the points. This allows time for the sequence of points to converge toward the attractor.

We can color a point based on which function was used to compute it. This is how we colored the fractals in figure 1. For example, the points in the Heighway dragon computed using h_1 are red, and the

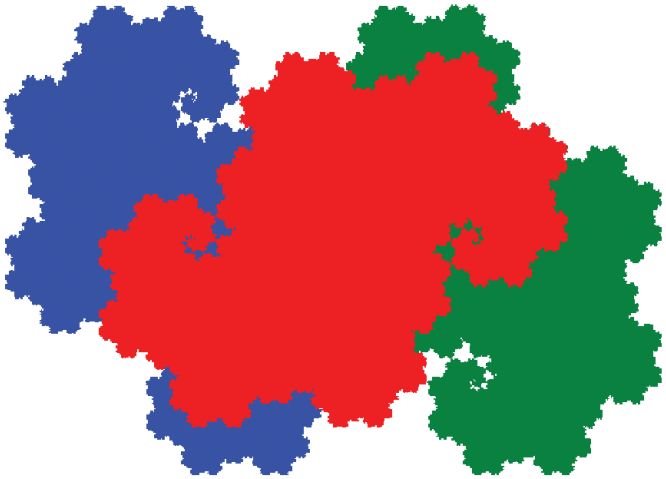


Figure 3. The Z_2 Heighway dragon.

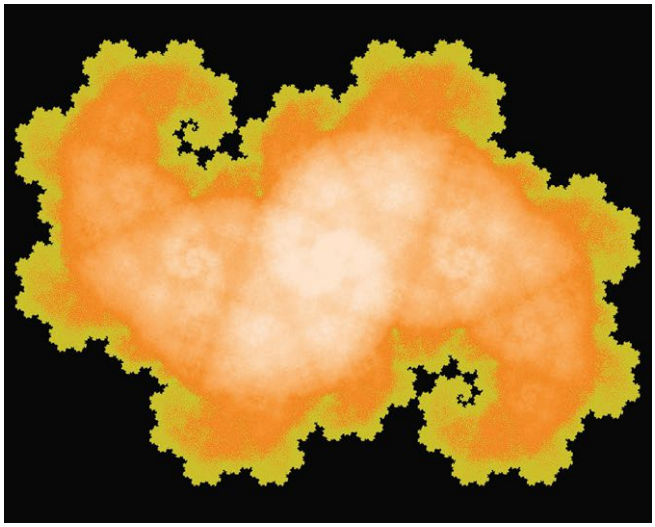


Figure 4. The Z_2 Heighway dragon colored by pixel counting.

points computed using h_2 are blue.

Of course, even the highest resolution computer screen has only finitely many points, or *pixels*. A single pixel can represent infinitely many points in the attractor. An alternative coloring scheme is to color a pixel based on how many points in the random sequence land in that pixel.

Figure 4 shows an image of the Z_2 Heighway dragon colored in this way. The color gradient varies from gold for low pixel counts through various shades of dark orange to pale orange as the pixel counts increase. Some experimentation is needed to find a good color gradient and the corresponding pixel counts. Factors to take into account include the size of the final image, the resolution of the computer screen, the number of points to be plotted, and the probabilities assigned to the functions in the IFS. But

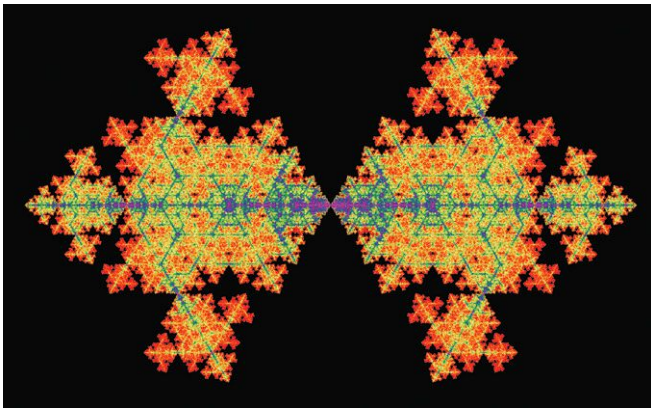


Figure 6. The D_2 Koch curve.

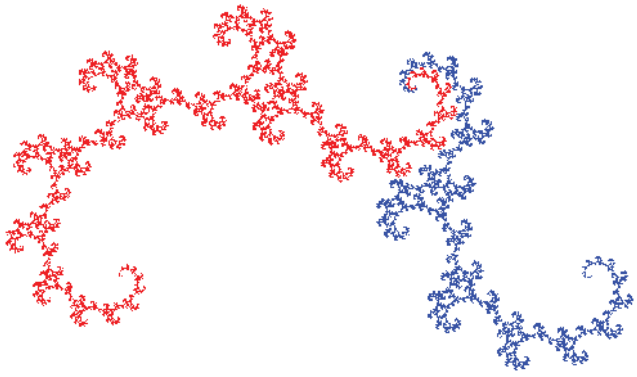
an appropriate choice helps to vividly illustrate the symmetry of the attractor.

Dragons and More

The Heighway dragon is just one example of a larger class of dragon fractals. Another example is the golden dragon curve, so named because its fractal dimension is the golden ratio. Figure 5 shows the golden dragon (formed from an IFS consisting of two functions) on the left and the attractor obtained by composing the cyclic group Z_2 with this IFS. We used the same pixel coloring as for the Z_2 Heighway dragon. This attractor has 180° rotational symmetry.

Figure 6 shows the attractor obtained by composing D_2 with the IFS for the Koch curve, colored from red to violet using pixel counting. The group D_2 gives the fractal 180° rotational symmetry and reflective symmetry across the horizontal and vertical lines through its center.

Figure 7 shows the fractals we obtain when we compose Z_4 and D_4 with the IFS for the Koch curve. The former has 16 functions that produce an attractor



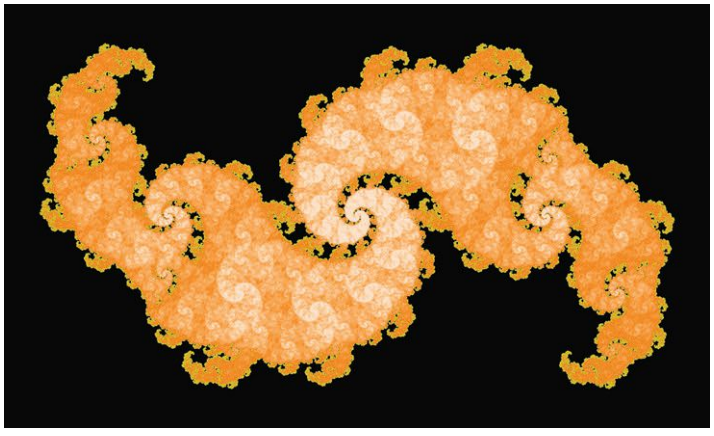


Figure 5. Golden dragon (above) and Z_2 golden dragon (right).

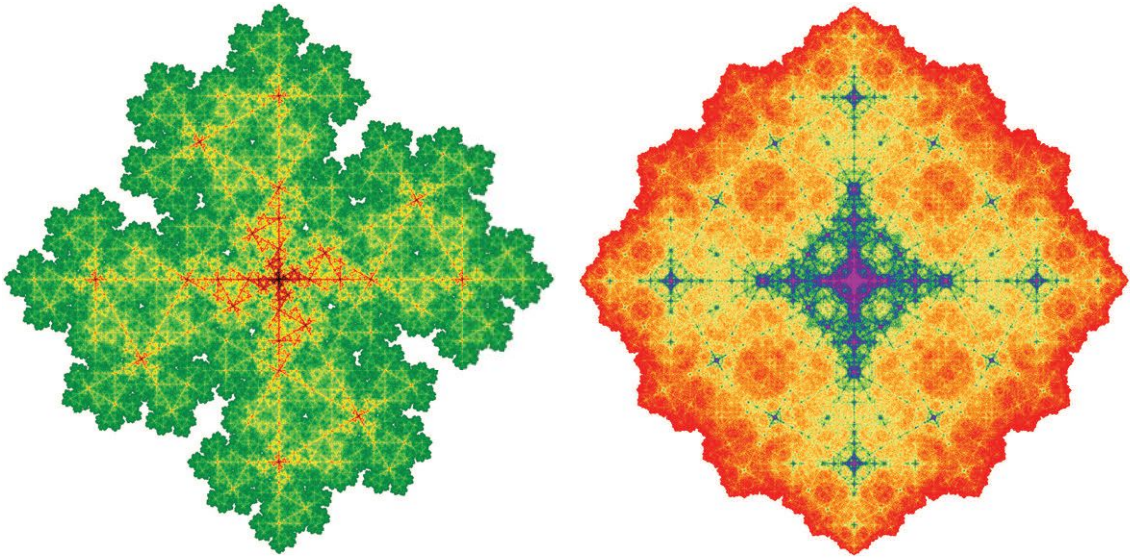


Figure 7. The Z_4 Koch curve (left) and the D_4 Koch curve (right).

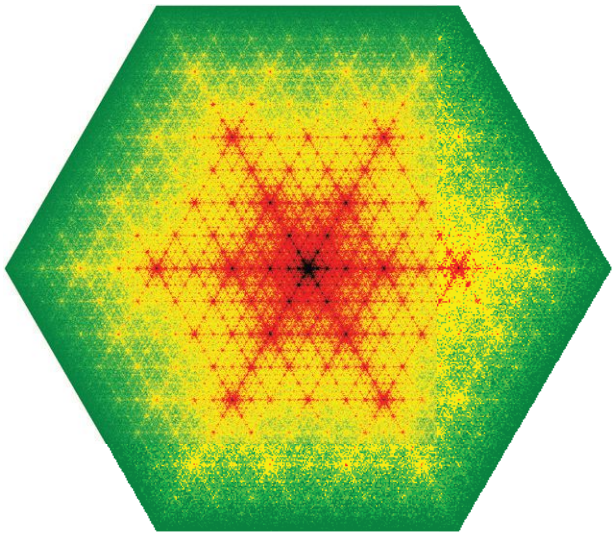


Figure 8. The Z_3 Sierpinski triangle.

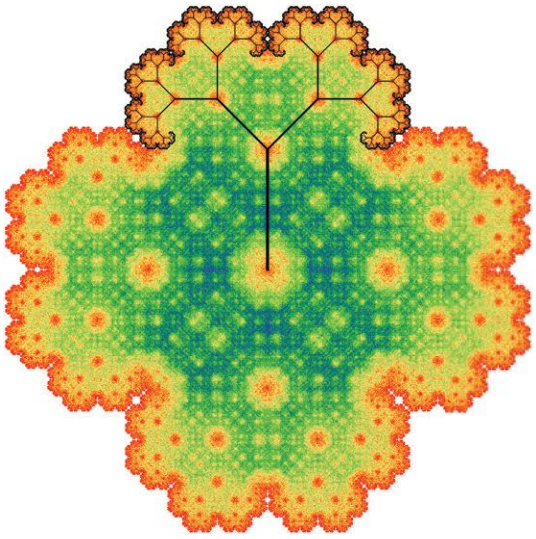


Figure 9. The Z_4 symmetric binary tree.

with fourfold rotational symmetry. The latter has 32 functions with an attractor displaying the D_4 symmetry of a square.

Finally, figure 8 and 9 illustrate that if the original attractor already has some symmetry, then the attractor constructed by composing with Z_n or D_n may produce additional symmetries beyond those guaranteed from the group used to build it.

Figure 8 comes from composing Z_3 with the Sierpinski triangle IFS. It has D_6 symmetry because the attractor is a filled-in hexagon. The initial IFS used for figure 9 generated the self-contacting symmetric binary tree with angle 45°, which is the black tree superimposed on the color fractal. We composed Z_4 with the IFS to form an IFS whose attractor is shown. It has D_4 symmetry.

Further Reading

For more on symmetric fractals, see "Symmetric Fractals" (chapter 7) in Field and Golubitsky's *Symmetry in Chaos*; an updated second edition of their book came out in 2009 (SIAM).

More details can also be found at the author's website, *ecademy.agnesscott.edu/~lriddle/ifs/ifs.htm*.

All fractal images shown here were drawn with the Windows program IFS Construction Kit available at *ecademy.agnesscott.edu/~lriddle/ifskit/*. ■

Larry Riddle teaches mathematics at Agnes Scott College, where he creates digital images of symmetric fractals as well as cross-stitch embroidery fractal artwork.

http://dx.doi.org/10.4169/mathhorizons.24.2.18