

A Generalization of the Chaos Game

Tom Bates

Santa Barbara, California, USA; tombagoo@gmail.com

Abstract

The well-known “chaos game” algorithm for generating the Sierpinski gasket stochastically is generalized to any order of regular polygon. A simple formula is derived that gives a value, referred to as the “kissing number,” to produce a crisp analog to the Sierpinski triangle for any order of regular polygon. In addition, a second kissing number is revealed that produces another fractal for each polygon. Finally, some surprising aspects of the algorithm’s behavior are revealed, including that the resulting figures contain a geometrically encoded history of the order of vertices targeted by the algorithm. This work was inspired by a desire to generalize the usual algorithm in a way that could be useful towards artistic ends and of interest for instructional ends at a variety of levels.

The Chaos Game

In the realm of iterated function systems (IFS), there is a procedure called the chaos game [1, 2] that can produce Figure 1. Starting with an equilateral triangle, mark a point inside (or even outside) it. Next, pick one of the vertices of the triangle *at random*. Mark another point halfway from the initial point to the chosen vertex. Repeat this procedure many times and you will get an image of the classic Sierpinski gasket. There are several ways to generate the Sierpinski gasket, but a way that involves hopping halfway to randomly chosen vertices is perhaps a little hard to understand at first. I use the term “cut fraction” to refer to the fraction of the distance from point p_i to the next vertex, which determines the location of point p_{i+1} .

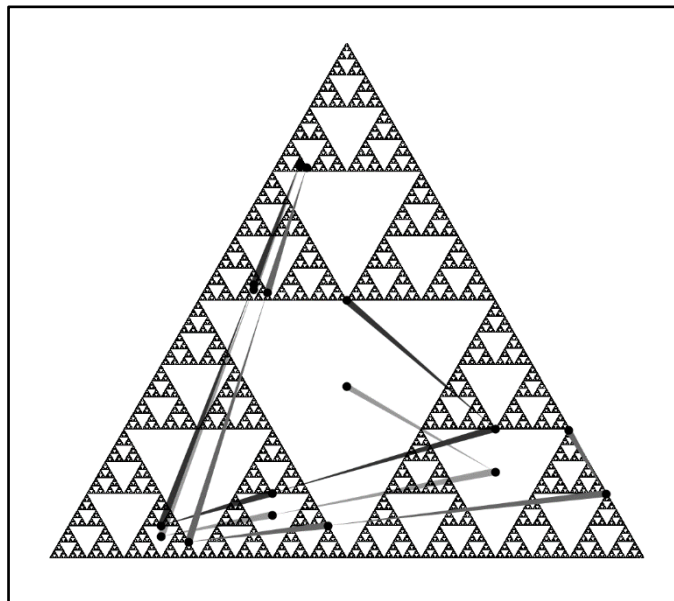


Figure 1: A stochastically generated Sierpinski triangle, with 3 paths connecting their six initial marks. Each subsequent point on a given path reprises the initial point’s geometric location in the parent.

Generalizing the Chaos Game

When I first encountered the chaos game many years ago, I didn't find any discussion of a generalization of the algorithm to any regular polygon. So, I set out to answer this seemingly obvious question for myself.

To get some clues, I wrote a bit of code to play around with the algorithm by varying the order of the polygon, the cut fraction, and several other properties. This revealed that there was always a special value of the cut fraction that creates a crisp fractal for any order, with the seeming exception of the square (Figure 8). But my goal was to find a general formula for the proper cut fraction k , which I expected would depend only on the order n of the polygon. I was looking for the formula:

$$k_n = f(n).$$

If fact, we will see there are two unique values of the cut fraction for each order of regular polygon. Each gives rise to a sharp fractal structure. I call these special cut fractions the interior and exterior kissing numbers, k_n and k_n^* . The reason for this naming will become clear shortly.

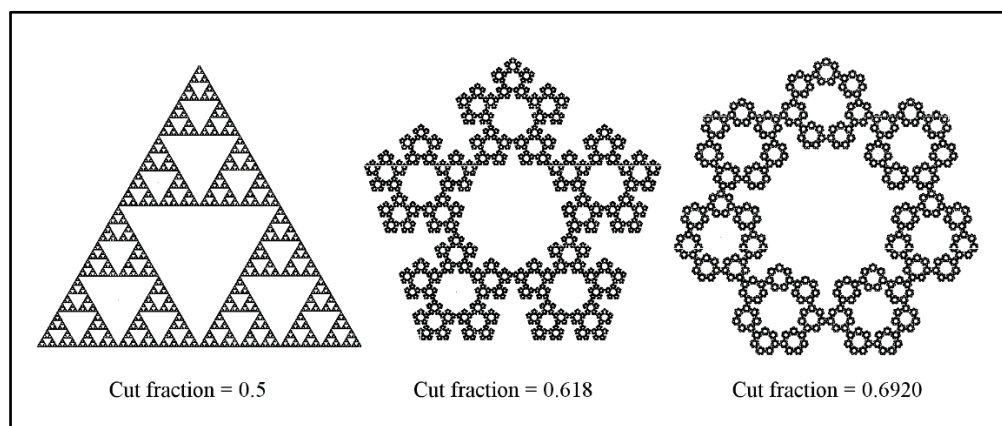


Figure 2: Values of the cut fraction obtained experimentally for three polygon orders.

The typical way of thinking about making the Sierpinski gasket is one of recursive subdivision and subtraction. But a stochastic method, hopping a fixed fraction towards randomly chosen vertices many times, is not so amenable to our imagination's predictive power. The mystery of this process made me want to understand it more deeply, and there were some interesting surprises along the way.

The triangle case is easy to construct by subdivision because you can begin by connecting the midpoints of the parent triangle's edges to generate three children triangles nested inside the parent. But connecting midpoints of adjacent sides for any other order of polygon yields only a scaled down and rotated version of the parent, plus n triangles nested between it and the parent.

A more general way to think of first-level children is that each child starts at a unique parent vertex with infinitesimal radius. These then grow uniformly inside the parent while remaining anchored at their parent vertex. Their growth stops as soon as they just "kiss" their two neighbor siblings, either at a vertex or an edge.

With this idea in mind, we can ask: given a parent polygon of order n and radius R , what child radius r generates n children that just touch their neighbors, either at a vertex or along an entire edge?

Before generalizing to any regular polygon, it is useful to consider n nested circles such that each child circle touches the interior circumference of the parent circle at one point and each of its two neighbors at one point. For a parent of radius R , and child circles of radius r_n , the child centers fall on a circle of radius

$$R - r_n = \frac{R}{1 + \sin\left(\frac{\pi}{n}\right)}.$$

Though simpler, the circle case provides both a guide and a sanity check for any general formula that might be found for the polygon case. Such circles are shown by dashed lines in Figure 3. Contrast these to the circles that circumscribe the child triangles. While the polygon case should approach the circle case for large n , we clearly need a different formula to find the radii of child polygons, and thus where their centers lie relative to the parent center.

In Figure 3, R is the radius of the parent triangle, taken to be 1, r is the radius of the child triangles, and $1 - r$ is the distance from the parent's center to the first-level child centers. For the triangle case $1 - r$ is 0.5, whereas for the circle case the corresponding value is 0.5359. This suggests that a formula similar to the circle formula might yield the proper cut fraction for any regular polygon.

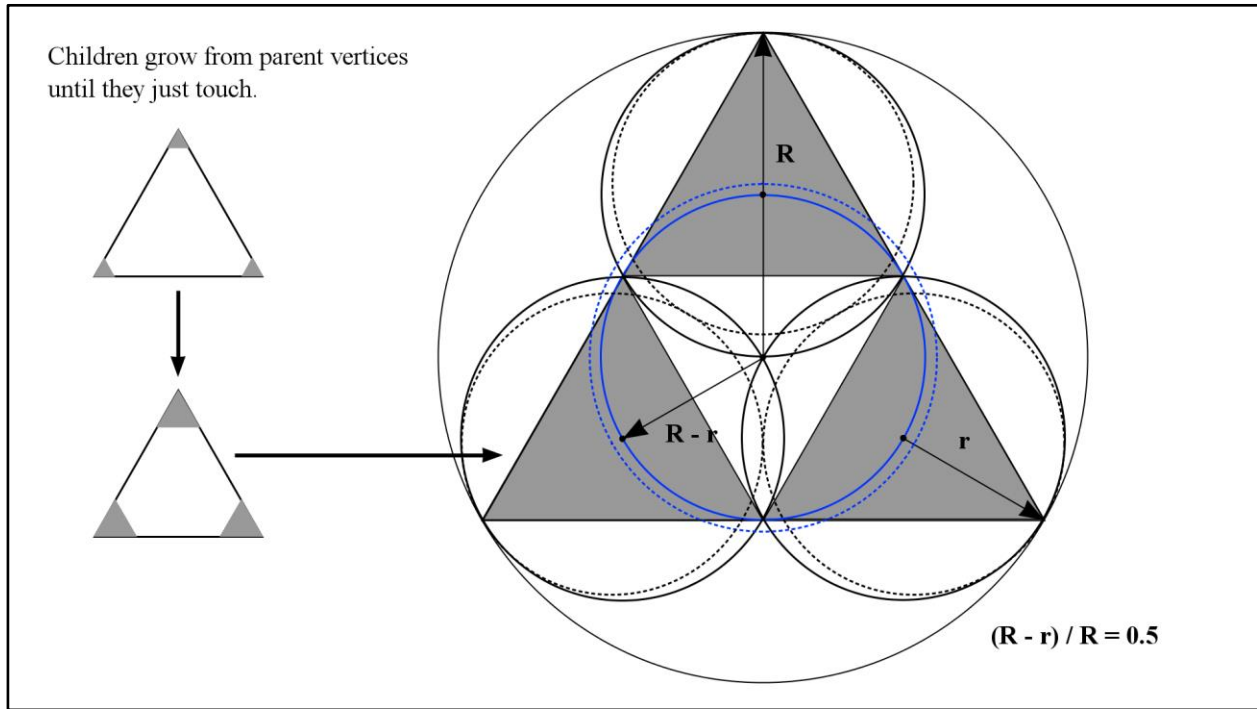


Figure 3: Nested circles are generally incommensurate with nested polygons. The parent circle and triangle on the right both have radius $R = 1$. The three dashed circles are kissing circles. Their centers lie on the dashed circle concentric with the outer circle. The three solid circles circumscribe the kissing child triangles, and their centers lie on the solid circle. They have radii r .

Kissing Polygons Formula Derivation

Compared to the simplicity of nested circles, the complicating issue for polygons are the discrete vertices and edges. The first thing we need is a way to know which vertices of a child, relative to its anchor vertex on the parent, kiss its two neighbors, for any order n .

By looking at the geometry for different n one finds that the kissing vertex v_k is always

$$v_k = \text{Ceiling}(n/4).$$

Figure 4 shows the geometric situation for an enneagon, or $n = 9$, case, for which $\text{Ceiling}(9/4) = 3$. This case is general enough to use for working out a general formula for the distance from the parent center to the children centers as some fraction of the parent radius. And, naturally, this ratio will hold for children and their first level children, etc., all the way down.

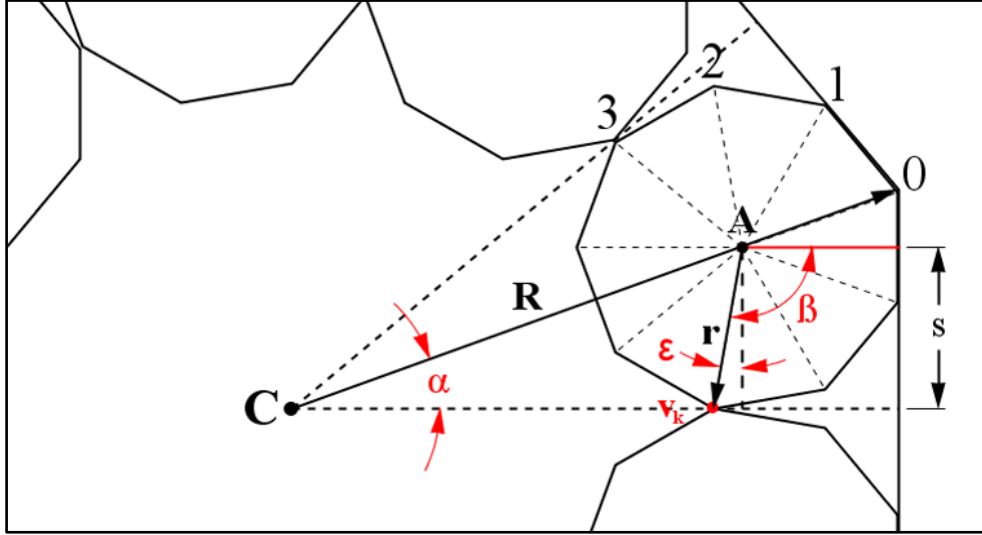


Figure 4: The key angles and lines needed to determine the child radius r from the order of a parent enneagon with radius R . Kissing always occurs on the dashed normal from the center of a parent, C , to the center of an edge. The kissing vertex is v_k . The child vertex numbering begins with 0 at the vertex shared with the parent. Here the kissing child vertices are the third in either direction from the anchor.

From Figure 4, we have the angles

$$\alpha = \pi/n, \beta = 2\alpha (v_k - 0.5), \text{ and } \epsilon = \beta - \pi/2.$$

Looking at the right triangle with CA (i.e. $R - r$) as its hypotenuse and s as its short side we get

$$s = (R - r) \sin(\alpha),$$

while from the smaller triangle with hypotenuse r and long side s defining angle ϵ we get

$$s = r \cos(\epsilon).$$

This is just what we need: two independent equations involving our unknown r :

$$(R - r) \sin(\alpha) = r \cos(\epsilon).$$

Setting $R = 1$ and solving for the child polygon's radius r gives

$$r = \frac{\sin(\alpha)}{\sin(\alpha) + \cos(\epsilon)}. \tag{1}$$

Finally, since the distance from the parent center to the first-level child centers is $1 - r$ we have:

$$k_n = 1 - r = 1 - \frac{\sin(\alpha)}{\sin(\alpha) + \cos(\epsilon)}. \tag{2}$$

This is the general formula for the interior kissing number. This gives the proper cut fraction for creating fractal gaskets analogous to the Sierpinski gasket for any order of a regular polygon. It depends only on n since that is the only dependency for α , β , and ϵ . Table 1 contrasts polygon and circle kissing numbers.

Table 1: Values for the interior kissing numbers for polygons and circles for several values of n .

| n | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|-----------------|--------|--------|--------|---------------|--------|--------|--------|---------------|--------|
| Polygons | 0.5 | 0.5 | 0.6180 | 0.6667 | 0.6920 | 0.7071 | 0.7422 | 0.7639 | 0.7784 |
| Circles | 0.5359 | 0.5858 | 0.6298 | 0.6667 | 0.6974 | 0.7323 | 0.7451 | 0.7639 | 0.7802 |

Notice that for every fourth order starting with 6 the polygon and circle values are the same because β is $\pi/2$ for such orders, which makes the polygon equation for r equal to the circle equation:

$$r = \frac{\sin(\alpha)}{\sin(\alpha) + \cos(\beta - \pi/2)} = \frac{\sin(\alpha)}{1 + \sin(\alpha)}.$$

In general, as n becomes large, the polygons tend toward kissing at a vertex such that β approaches $\pi/2$, and so the polygon values converge toward the circle values, but exactly so only for $n = 6, 10$, etc.

Two Kissing Numbers for Each Polygon Order

If one uses cut fractions larger than the interior kissing number, the figure initially collapses toward the parent vertices, resembling a Cantor dust with the parent's symmetry as it does so, especially for the square. For cut fractions beyond one the figure blooms anew, larger and mostly outside the parent polygon, and another sharp fractal appears that is significantly larger than the interior fractal, and rather different looking for odd orders (Figure 5). This happens at the exterior kissing number k_n^* . The two kissing number formulas vary only by the sign of r : whereas the interior kissing number is $1 - r$, the exterior kissing number is

$$k_n^* = 1 + r = 1 + \frac{\sin(\alpha)}{\sin(\alpha) + \cos(\varepsilon)}. \tag{3}$$

When one goes beyond the exterior kissing number the figure expands rapidly and at a cut fraction of 2 or more it becomes so large and sparse that it seems to vanish into nothingness.

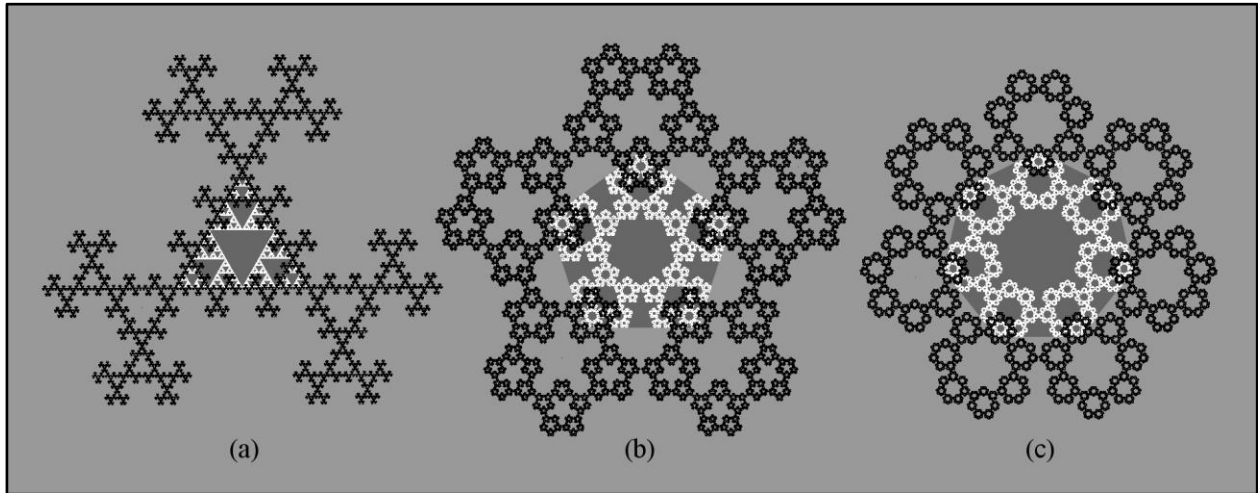


Figure 5: Interior (white) and exterior (black) fractals produced by using both kissing numbers. The interior fractal exactly fills the parent polygon. The values of k_n and k_n^* for each order are (a) 0.5 and 1.5, (b) 0.618 and 1.382, (c) 0.692 and 1.308.

Histories from Geometric Congruence

The magic of k_n is that it takes any point in one level to a geometrically equivalent point in the next child level. This trend continues ad infinitum, *placing only a single mark at each level of scale descent*. Beginning in eventual negative space is the easiest way to see this. But the scale-descending chain of geometric congruency does not depend on where you start. This can be seen in Figure 1, or by overlaying the path taken onto a cutout version of the same order, as in Figure 6.

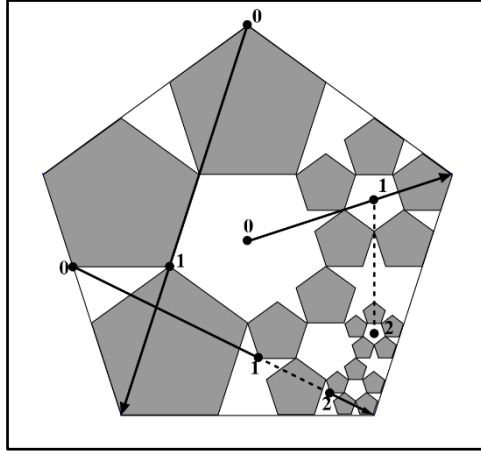


Figure 6: The geometric congruency of placed points shown for three short runs of the pentagonal chaos game. Each run goes from its initial point 0, in or on the parent, toward its first randomly chosen parent vertex (arrowheads). As 0.618 is the pentagonal kissing number higher, numbered points occur 0.618 of the distance from their predecessor to its target vertex.

Yet, even if we start in what is destined to be negative space, the proper figure takes shape, seeming to have been made by not being made! This is entirely due to the limited resolution of our eyes and displays. Notice that for the last few points on the path in Figure 1, it is nearly impossible to see the ever-smaller negative triangles they too are in, so they perceptually contribute to the positive space of the figure by outlining perceptible negative space. Each different starting point, mathematically speaking, yields a completely different non-self-overlapping point set. But beyond the first few points all the resulting figures will look the same. Only zooming in continuously as the algorithm runs could reveal this to the eye.

In principal, such a zoom ability could reveal the history of vertex targeting for a given run purely from the scale descending geometric congruency of each point. Thus, despite random vertex selection, the resulting figure contains a geometrically encoded history of the sequence of vertex targeting. This would involve calculating and recording each point's coordinates to a high enough precision to be able to numerically zoom in on every point and determine the size of the congruent feature they landed in. Sorting these results according to feature size would then reveal the order of vertex targeting. Although not necessarily practical, this is fascinating to me on a conceptual level. It would be interesting to know the point count limit under various assumptions of computational resources and time.

Colorful Histories

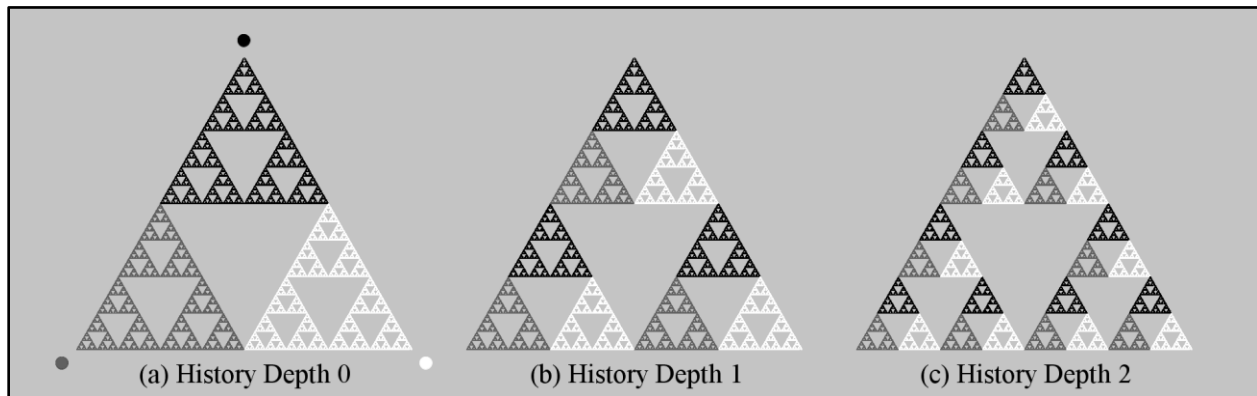


Figure 7: (a) Coloring points according to the last vertex targeted before placement. (b) Coloring according to the next to last vertex targeted. (c) Coloring according to the 2nd to last vertex targeted

Yet another kind of order inherent in the algorithm can be seen by assigning a unique color to each vertex and recording the history of vertex targeting. One can then color each mark according to how deep in their history we look back. Figure 7 shows three cases of such targeting history look-back. For history depth 0 we color a point according to the last vertex targeted, for depth 1 we color according to the vertex targeted one before the last vertex targeted, and for depth 2 the vertex targeted 2 steps back, etc.

Coloring history sheds light on the notion that the chaos game on the square using a cut fraction of 0.5, which is the square's interior kissing number, has no structure. This is dispelled by coloring history, as seen in Figure 8b. Finally, by upping the cut fraction past 0.5, which is the same as reducing the radius of the children relative to the parent, we can see that the square at 0.5 is at the edge of revealing a very visible fractal structure with a Cantor dust nature.

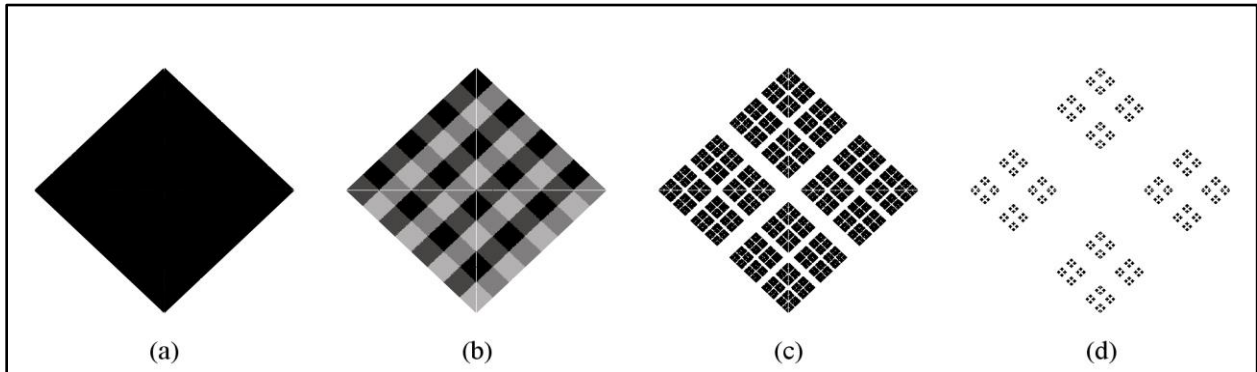


Figure 8: Aspects of the square: (a) using the interior kissing number of 0.5 and a single color, (b) using history coloring with depth 2, (c) using a cut fraction of 0.55, (d) using 0.67 as the cut fraction.

From Theory to Art

Moving on to a more artistic aesthetic, below are a few images derived from this work.

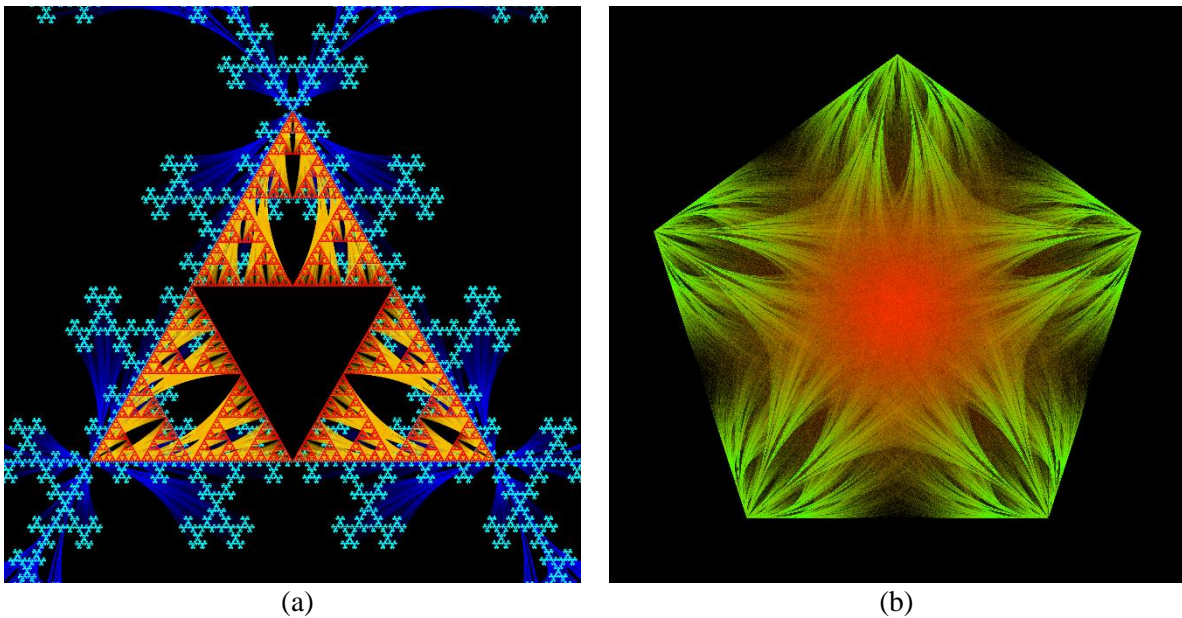


Figure 9: These images are composites of sweeping the cut fraction over some range, thus they are like the frames of an animation superimposed upon each other: (a) a triangle going from interior to exterior kissing numbers, (b) a pentagon with the cut fraction running from 0 to 1.

In Figure 9 and 10, images have been made by compositing many instances of the same figure, but with the cut fraction changing slightly between images so each point generates a curved path over a sequence of images, as if what might best be viewed as an animation has been collapsed onto a single frame.

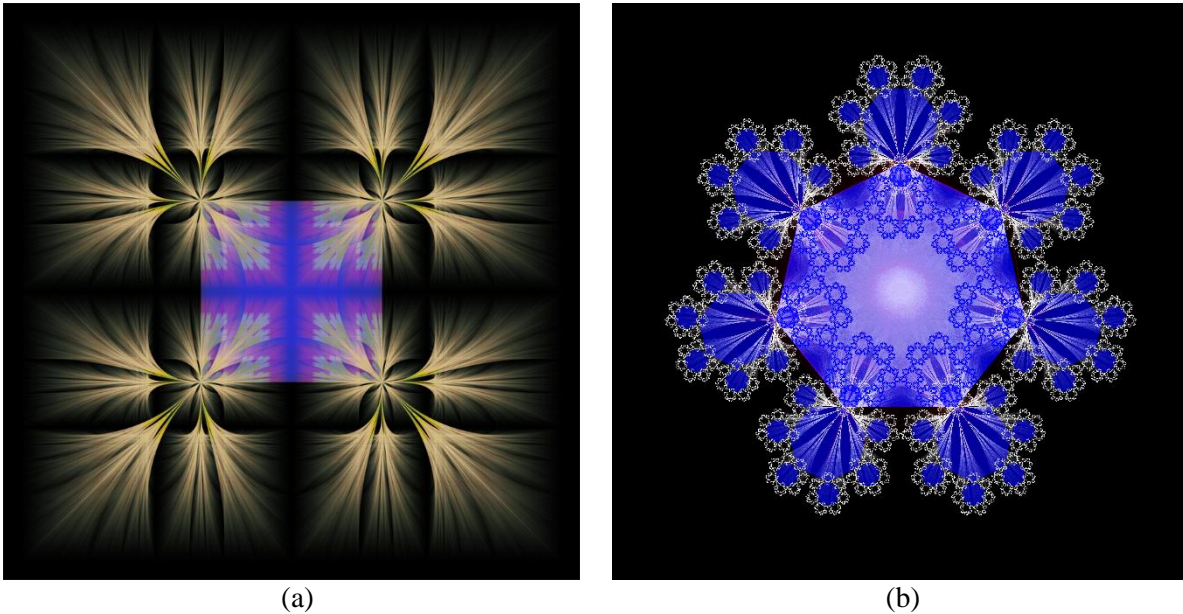


Figure 10: (a) a square with the cut fraction ramped from 0 to its exterior kissing number 1.5, (b) a heptagon ramped from 0 to its exterior kissing number 1.308.

Conclusion

In setting out on my small quest to generalize the chaos game, I had no idea of the insights I would gain about what it actually does. I believe that playing via code is a fantastic way to explore geometric and visual topics such as this one. I have created a basic web demonstration that allows anyone to play with the generalized chaos game. It calculates both kissing numbers for any order and allows you to try other values as well. The demonstration is at http://www.betweenartandscience.com/chaosgame_demo.html.

References

- [1] Barnsley, Michael. *Fractals Everywhere*. 1993.
- [2] Peitgen, Heinz-Otto, & Saupe, Dietmar, eds., *The Science of Fractal Images*. 1988.