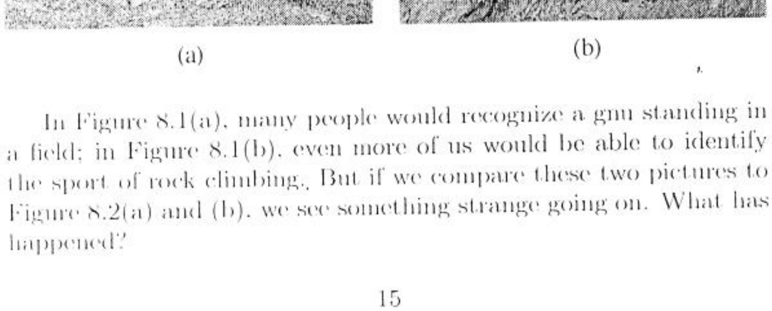
CHAPTER 8

Introduction to Fractal Geometry

THE techniques of perspective are particularly helpful for the realistic drawing of "man-made" objects such as houses, roads, fences, etc. With regard to drawing and understanding the appearance of natural objects such as trees, clouds, mountains and much more, there is another useful branch of mathematics which we introduce in this chapter.

Consider the pictures in Figure 8.1. Can you tell what they represent?



(a) (b)

Figure 8.1. Two pictures from nature.

In Figure 8.1(a), many people would recognize a gnu standing in a field; in Figure 8.1(b), even more of us would be able to identify the sport of rock climbing. But if we compare these two pictures to Figure 8.2(a) and (b), we see something strange going on. What has happened?

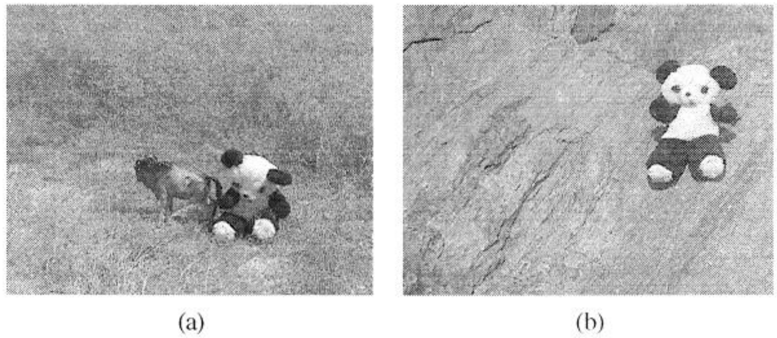


Figure 8.2. A small stuffed panda (a) next to the "gnu" from Figure 1(a), and the same toy panda (b) on the rock where we superimposed the image of a rock climber in Figure 1(b).

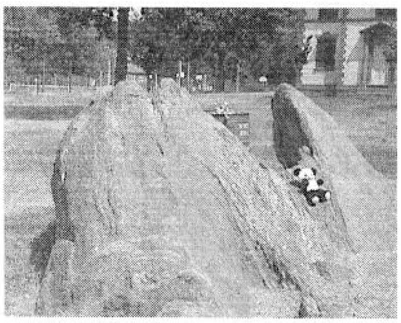


Figure 8.3. The toy panda bear on the "mountain"—a large rock.

In fact, the "field" and the "mountain" were some dry grass and a large rock (Figure 8.3), both near a baseball field at the south end of Franklin & Marshall College; the gnu was a 10-inch plastic toy, and we PhotoShopped the mountain climber in. We used simple tricks to create the effect of larger landscapes than we actually had available.

What made this switch so effective is that many objects look the same at different scales. Dry grass looks somewhat like tall grass. Small rocks look like big rocks, and both look like the side of a mountain. Small pieces of clouds have the same shape and structure as the whole cloud. A small piece of cauliflower (a floret) looks like the whole cauliflower (Figure 8.4).

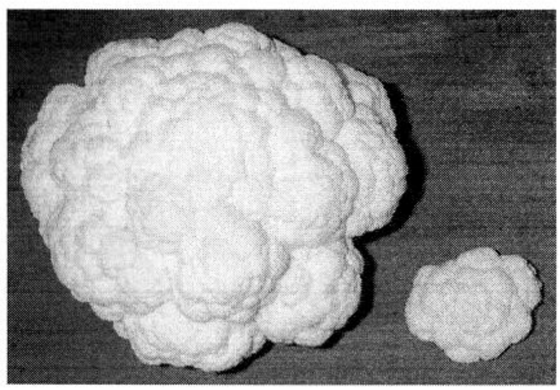


Figure 8.4. A small piece of cauliflower looks like a miniature version of the whole cauliflower. (In fact, a piece like this is called a *floret*, which means a small flower.)

In the 1970s mathematician Benoit Mandelbrot described self-similar objects by coining the name "fractals," from the Latin root *fractus* meaning fractured or broken. That is, if you "fracture" (or break off) a little piece of a fractal, it looks like the whole thing. Many natural objects have this property of "self-similarity," and this property is of interest both artistically and scientifically.

To see how fractals approximate natural forms, and at the same time suggest techniques which an artist might use to render such forms, consider the sequence of pictures in Figure 8.5. The first picture shows a rudimentary tree branch with 5 branch tips. It is not a fractal, because it lacks an important property of fractals called *self-similarity*, which in a broad interpretation means that parts of a shape are exact, or nearly exact, miniatures of the whole shape. In the second picture, each of the 5 branch tips is replaced with an exact miniature of the branch in the first picture. The second branch is not a fractal either, because the 5 sub-branches each have 5 tips, while the whole branch now has $5^2 = 25$ branch tips. The third branch in the sequence is an attempt to remedy this; each of the 5 sub-branches is a 25-tipped miniature of the second branch. Of course, the problem is that the current branch now has $5^3 = 125$ branch tips, so self-similarity has not yet been attained. Indeed, the figure will not become self-similar—a fractal—until the process has in effect been repeated an infinite number of times.

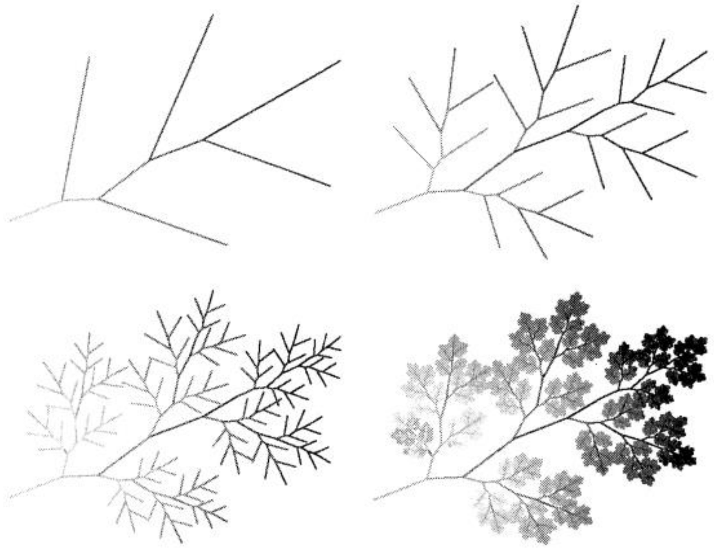


Figure 8.5. A fractal tree branch made by algorithmic drawing.

Fortunately, we don't need to wait forever to see the true fractal branch. The fourth branch in the sequence is the result of repeating the process 3 more times ($5^6 = 15,625$ branch tips), and it is visually indistinguishable from the true fractal limit. This is because the subsequent branch tips which would be added would be so small as to be virtually invisible. Most importantly, notice how the character of the branch has changed in the fourth picture. It is "organic" as opposed to "mechanical" or "geometrical," even though we have used a mathematical algorithm—a rigid step-by-step process—to create it. In this computer rendering, the branch tips have even merged to form leaves, reminding us that not only do trees contain leaves, but leaves also contain trees, in the form of their treelike vein structure. Results like this not only give hope to the artist seeking hints on drawing from nature, but they also suggest ideas about the structure and growth of plants—a notion that plant biologists have taken seriously.[1]

[1]For a readable and fascinating introduction to these ideas, see Lindenmayer, A., and Prusinkiewicz, P., *The Algorithmic Beauty of Plants*, New York, Springer-Verlag, 1996.

Evidence suggests that artists have also taken fractal algorithms seriously, and not just recently. Compare, for example, the Japanese woodblocks in Figure 8.6, which we've paired with modern computer-generated images. Figure 8.6 features three pairs of images. The left member of each pair is a 19th century Japanese woodblock print, and the right member is a computer-generated fractal. Though startlingly similar to their companion images, the woodblocks predate the fractals by more than 100 years! The first image is a detail from the woodblock, *Shono: Driving Rain*, from the series, *The Fifty-Three Stations of the Tokaido*, by Ando Hiroshige (1797–1858). The fractal to its right was generated by a process called an "Iterated Function System," which we discuss later. The second woodblock is *Boats in a Tempest in the Trough of the Waves off the Coast of Choshi* (detail), from the series, *A Thousand Pictures of the Sea*, by Katsushika Hokusai (1760–1849). The fractal to the right of it is called a "quadric Koch island," a name coined by Mandelbrot. The third woodblock is a panel from the triptych, *Short History of Great Japan*, by Ikkasai Yoshitoshi (1839–1892). The accompanying fractal is a mathematical model of two-fluid displacement in a porous medium after Paterson.[2]

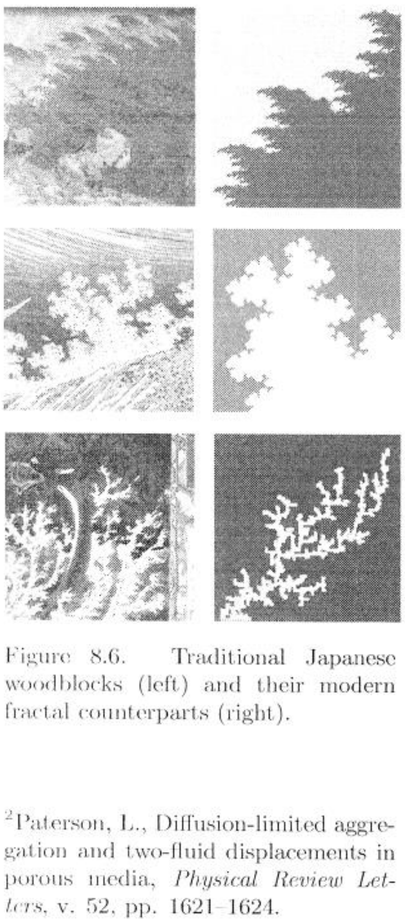


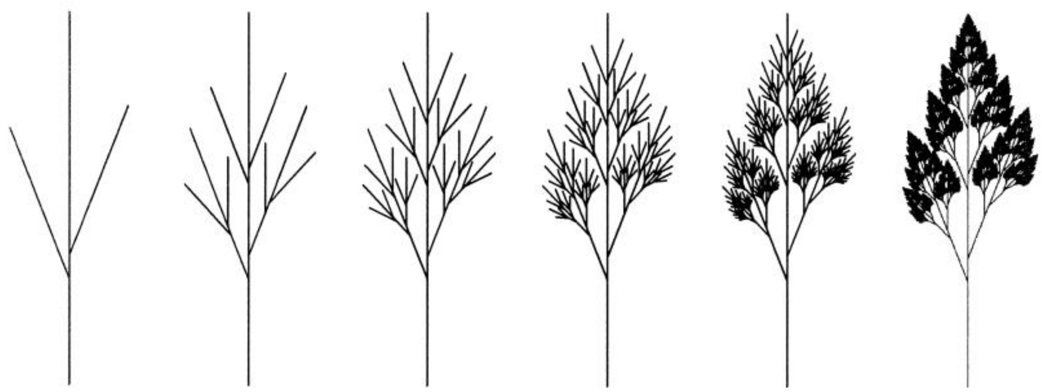
Figure 8.6. Traditional Japanese woodblocks (left) and their modern fractal counterparts (right).

Drawing fractals is much more meditative than drawing in perspective. Patience, repetition, and time are all good attributes to have. Here are some practice exercises that demonstrate this. (Give yourself a half-hour to an hour to do each one; playing good music helps as you work.)

[2]Paterson, L., Diffusion-limited aggregation and two-fluid displacements in porous media, *Physical Review Letters*, v. 52, pp. 1621–1624.

Practice Exercises.

I. Draw a fractal tree. Start with a figure like the one at the left, which consists of a trunk and three main branches (notice that the left branch begins slightly below the right branch). Then turn each of the three branches into a tree by adding two new branches at approximately the same angle as on the orginal tree; you'll now have 9 branches. Repeat this process (on each of the 9 branches, create a tree by adding two new branches; then create trees on the resulting 27 branches, and so on). In Figure 8.7 we show you several initial steps, plus a version that appears after many iterations.



(a) (b) (c) (d) (e) (f)

Figure 8.7. Steps in drawing a fractal tree (a–e), and a computer rendering (f) after many iterations.

II. Use a ruler to draw a fractal "cauliflower." Begin with a horizontal line across the bottom of the paper. From now on, every time you see a line segment, you will add on an inverted "V whose height is approximately 1/3 that of its length, as in Figure 8.

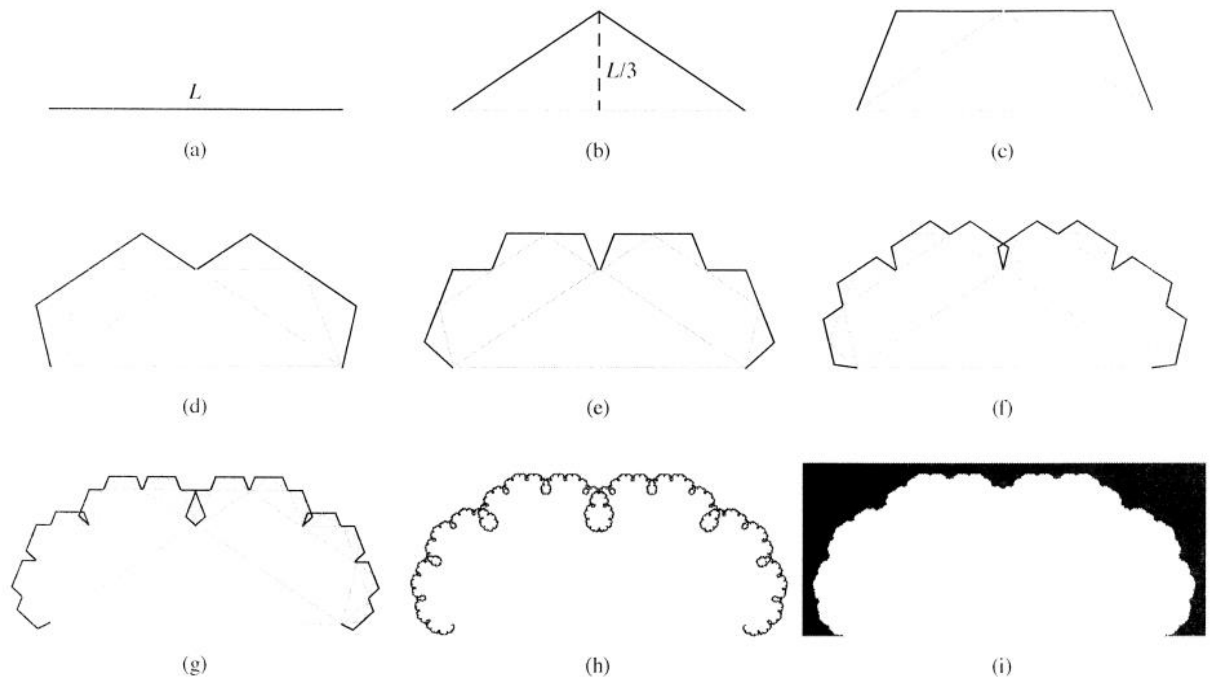


Figure 8.8. Steps in drawing a "cauliflower" (a–g), and a computer rendering (h) after many iterations. Part (i) shows the region above the cauliflower shaded in black.

Here's an example of a slightly different version of the cauliflower-drawing technique applied to lightning bolts. On the right of Figure 8.9 is a detail from a photo of a lightning storm over Boston. We begin by marking four dots A, B, C, D on one of the lightning bolts. In step (i) of the figure, we draw the polygon $ABCD$ as a first approximation to the lightning bolt. In each of the subsequent steps (ii)–(iv), we connect the pairs A, B and B, C and C, D with a scaled copy of the previous approximation. As indicated, the drawing in step (iv) compares pretty well with the actual lightning bolt. This approach isn't completely scientific, but it shows that iterative drawing techniques can lead to nice results. In the next chapter, we'll see how a more scientific study of coastlines helped spur the development of fractal geometry.

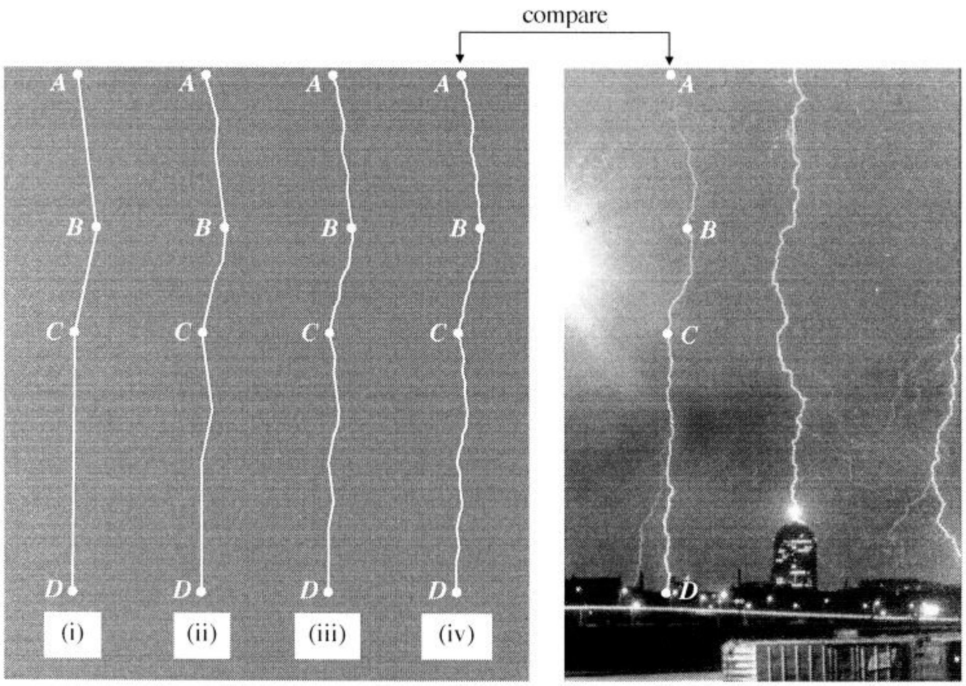


Figure 8.9. Left: Recursive drawing of a lightning bolt. Right: Detail from photograph of a lightning storm over Boston. Step (iv) is a fair approximation of the lightning bolt. Photo: U.S. National Oceanic and Atmospheric Administration. Photographer: *Boston Globe*.

III. A "mathematical-looking" fractal that's easy to sketch is the *Sierpiński triangle* (or Sierpiński gasket), named after Polish mathematician Wacław Sierpiński. In Figure 8.10 we start with an equilateral triangle (a), then draw an upside-down triangle inside it by connecting the midpoints of the larger triangle (b). At first you might want to lightly mark a dot inside this "midpoint triangle" to indicate that you won't draw inside that triangle anymore. In (c) we draw midpoint triangles inside the remaining three (right-side up) triangles and mark dots to keep the new midpoint triangles empty. In (d) we draw midpoint triangles inside the nine remaining triangles. Try this out and keep drawing until the triangles become so small you can't draw inside them anymore. The "limiting shape" you have approximated is the Sierpiński triangle.

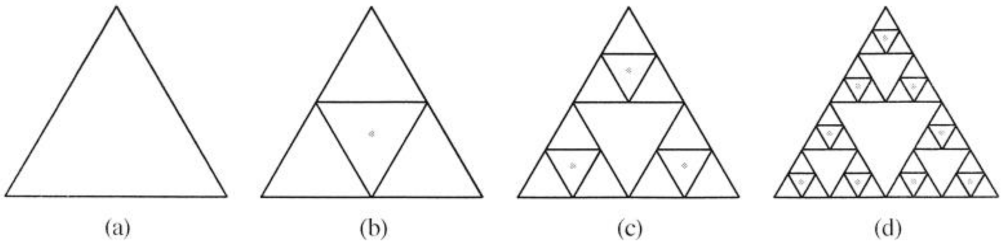


Figure 8.10. Steps in drawing the Sierpiński triangle. The dots indicate triangles that remain empty.

IV. There's a right-angled version of the Sierpiński triangle that you can build up using squares instead of triangles. Figure 8.11(a) shows a square on a grid, and Figure 8.11(b) shows the three half-sized squares you draw in the next step. One half-sized square goes on top of the big square, one goes in the lower left corner of the big square, and goes immediately to the right of the big square as shown. (The grid is for precision.) Figure 8.11(c) shows how to proceed next. The rule is this: at the end of each completed stage, focus only on the smallest squares you just drew. For each such square, draw a square half as big on top of it as indicated, a square half as big in the lower left corner, and a square half as big immediately to the right of it. Use the grid lines to be precise, and keep going. Now start over in Figure 8.12 (or a photocopy of it). Be sure to count squares to keep your drawing accurate. You'll see that the fractal is in a sense a union of squares, but it doesn't really look like it when it's fully developed!

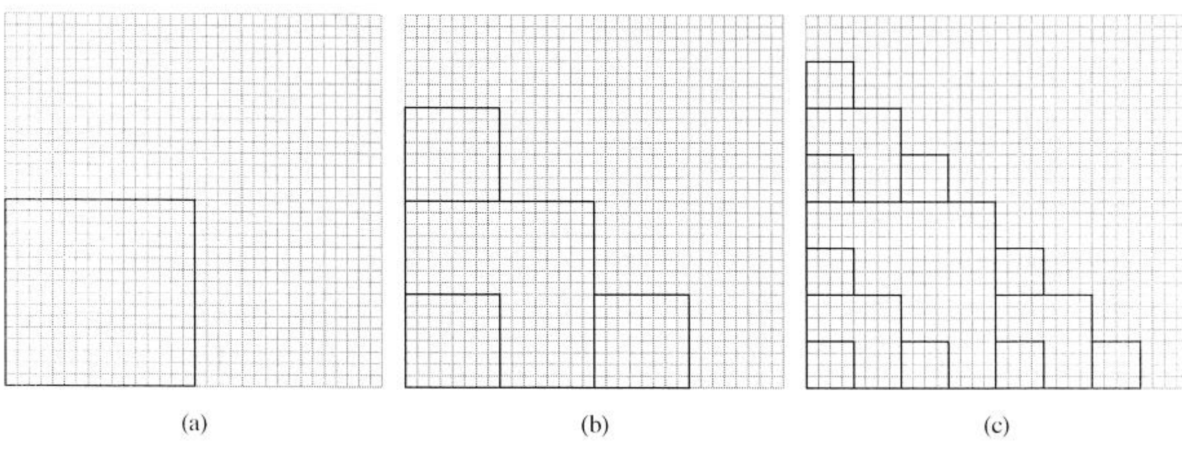


Figure 8.11. Guide for drawing a right-angled Sierpiński triangle using squares.

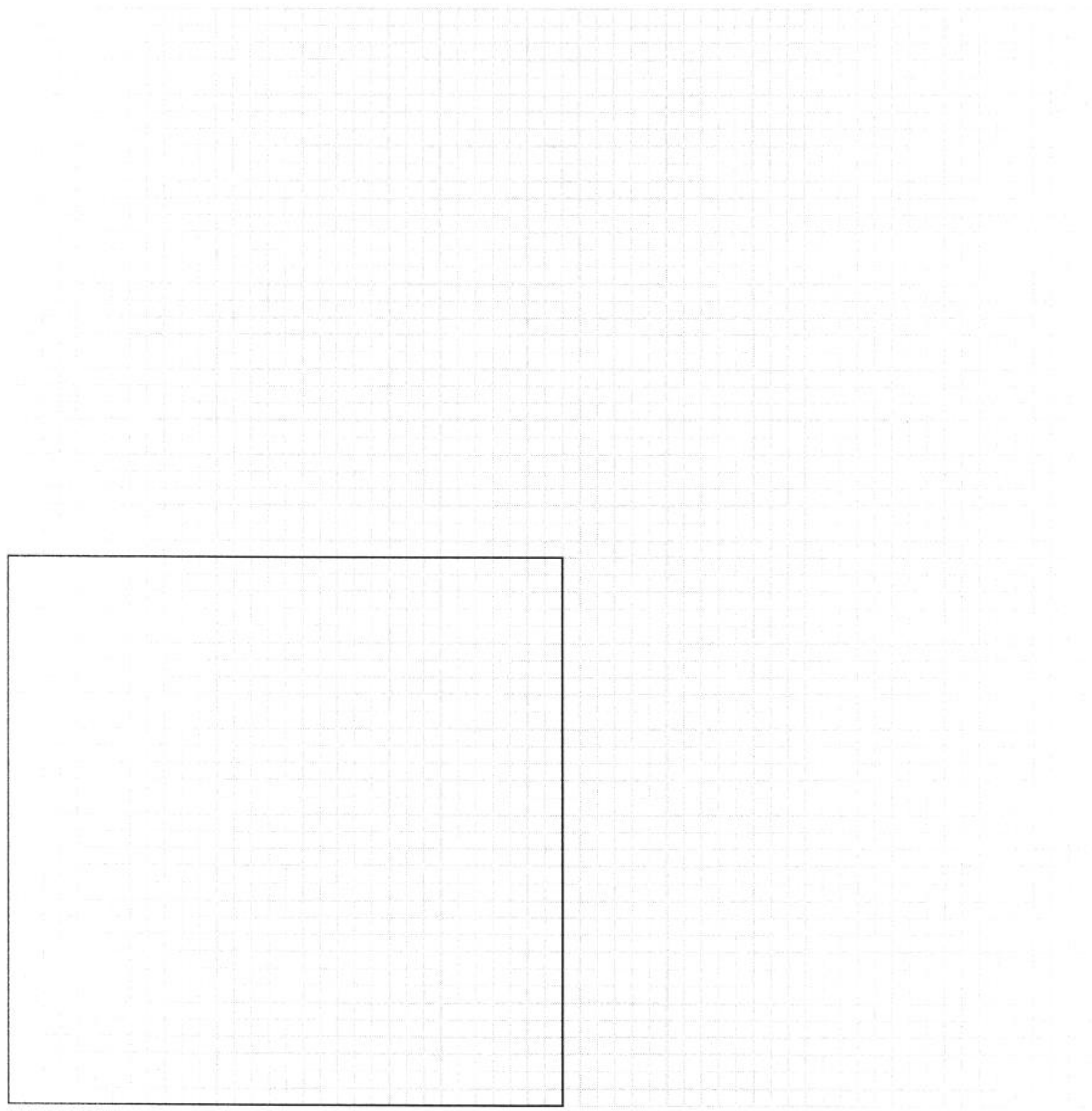


Figure 8.12. Template for drawing the right-angled Sierpiński triangle.

Another classic fractal attributed to Sierpiński is the *Sierpiński carpet*, whose construction appears in Figure 8.13. This time we make the fractal white on a black background.

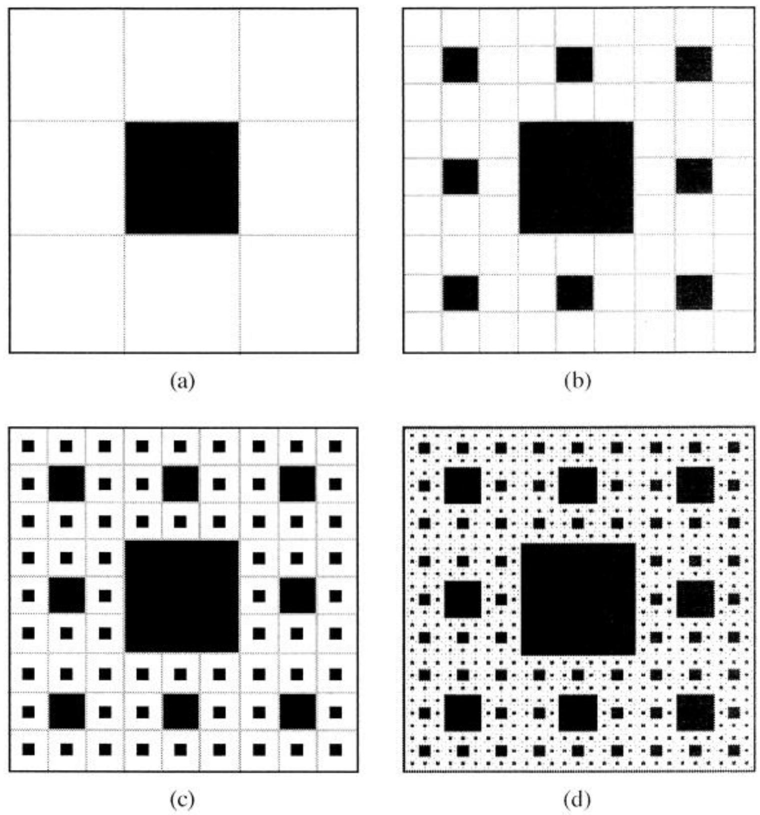


Figure 8.13. The first three steps in drawing the Sierpiński carpet (a–c), and the appearance of the actual fractal (d). The fractal is the remaining white part after the (infinitely many) holes have been subtracted.

To draw the Sierpiński carpet we start with a white square in Figure 8.13(a), use lines to divide it into nine equal-sized squares, and then blacken in the "middle ninth" square. In (b) we apply the same procedure to the remaining 8 white squares, and in (c) we apply it to the remaining 64 white squares. Part (d) shows the fractal (the white part) after the procedure has been carried out indefinitely.

While the Sierpiński triangle and the Sierpiński carpet may look more "mathematical" and not as natural as the fractal tree and the fractal cauliflower, they nevertheless contain essential features of patterns in nature. As an example, we modify the construction of the Sierpiński carpet in Figure 8.13 to model cratering patterns in Figure 8.14.

In Figure 8.14(a) we punch a "giant" crater of width (diameter) w into the surface, much like punching the square hole in Figure 8.13(a). In Figure 8.14(b) we add 8 randomly-placed "large" craters of width $(1/3)w$, much like we added the 8 square holes in In Figure 8.13(b). In Figure 8.14(c) we randomly place $8^2 = 64$ "medium" craters of width $(1/3)^2 w = (1/9)w$, much like we added the 64 square holes in Figure 8.13(c). In Figure 8.14(d) we randomly place $8^3 = 512$ "small" craters of width $(1/3)^3 w = (1/27)w$, and in Figure 8.14(e) we randomly place $8^4 = 4096$ "tiny" craters of width $(1/3)^4 w = (1/81)w$. Stopping here, we compare the result with a photograph of the surface of the Moon in Figure 8.14(f).

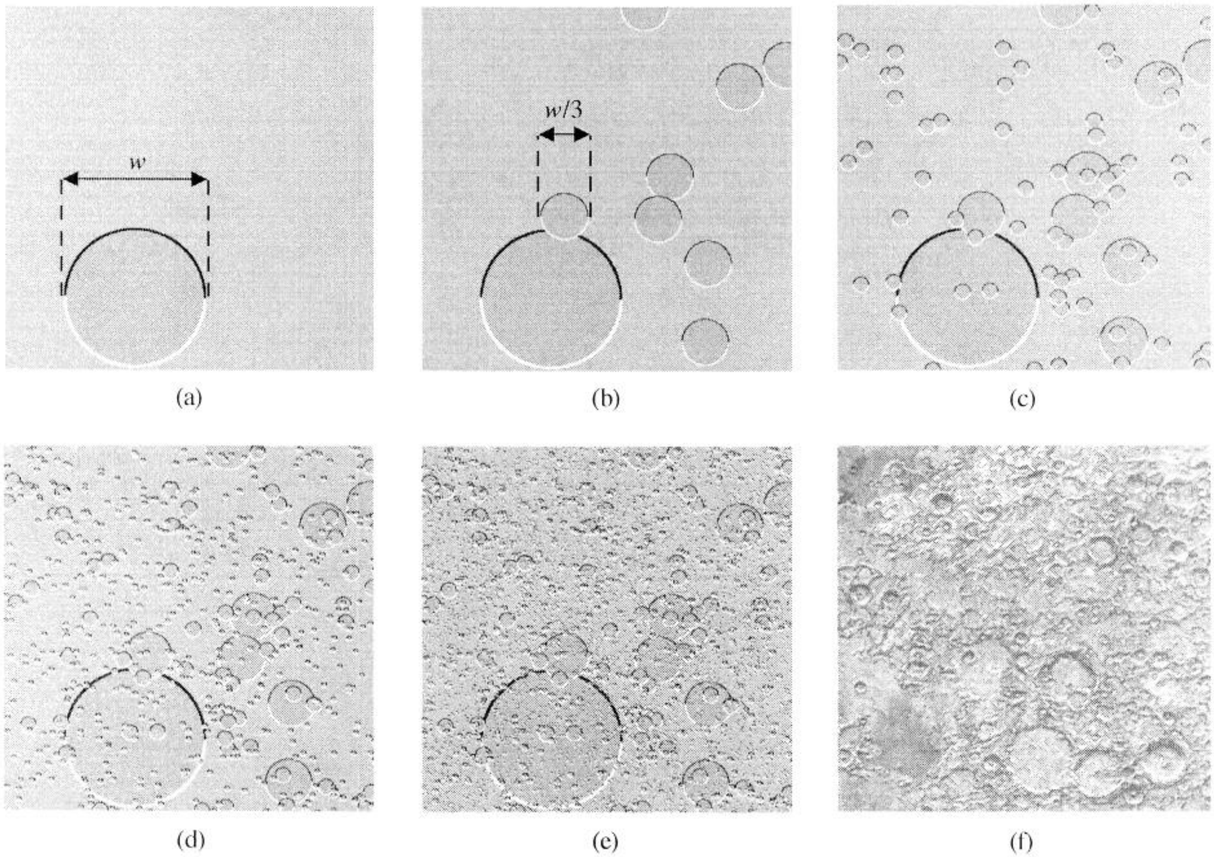


Figure 8.14. Mimicking the construction of the Sierpiński carpet, we start with one "giant" crater (a) of width w; add 8 "large" randomly-placed craters, each 1/3 the size of the giant one (b); then $8^2 = 64$ "medium" craters, each 1/3 the size of the large ones (c); then $8^3 = 512$ "small" craters, each 1/3 the size of the medium ones (d); and $8^4 = 4096$ "tiny" craters, each 1/3 the size of the small ones (e). At each step we use a computer program to randomly place the craters in the gray square. Part (f) is a photograph of the surface of the Moon.

Even with the simplistic "craters" we were drawing, the comparison is not too bad. In fact, for many bodies in the solar system the observed mathematical relationship between the number of craters in a given region and their diameters bears a strong relationship to the number and sizes of the holes in the Sierpiński carpet, the Sierpiński triangle, and other fractals.[3] In this case we have modeled the surface of the Moon as a kind of random Sierpiński carpet with round holes.

[3]Benoit Mandelbrot made this comparison in *The Fractal Geometry of Nature*, New York, W. H. Freeman, 1983, pp. 302–303.

We will make this comparison more precise in the next chapter. For now, we simply wish to note that even highly regular, geometric fractals like the Sierpiński carpet can embody important aspects of the rugged and apparently irregular forms of nature. By comparing fractals with nature, we can more fully understand nature's patterns and thus do a better job of drawing them. For example, the previous comparison immediately shows that when drawing craters, we should draw only a few large craters, many more medium-sized craters, and many, many more small craters.

Iterated Function Systems.[4] How do computers draw fractals? They often uses a process called an "Iterated Function System," or IFS. For our purposes we will think of a *function* as a rule for transforming shapes (sets) in the xy-plane into other shapes. For example, Figure 8.15(a) shows the unit square P—the square with corners at $(0,0)$, $(1,0)$, $(1,1)$ and $(0,1)$. We have labeled it with a big "P" to keep track of the way it gets transformed. A function is usually denoted by a letter, so let us define a function f by the way it transforms P into the shape labeled $f(P)$ (read "f of P") in Figure 8.15(b). The new shape $f(P)$ is called the *image* of P. The image $f(P)$ is a square half the size of P in each direction; it has been flipped (reflected) so that we see the "back side" of P; it has been rotated 90° so that it lies on its side; and it has been moved so that its lowest side stretches from $1/2$ to 1 on the x-axis. By showing the flipped-over letter P as being undistorted in Figure 8.15(b), we mean to suggest that f does not distort the interior of the unit square as it shrinks it and moves it around.[5]

[4] This section is optional.

[5] Such a function is often called a *contracting similarity transformation* because any shape inside the unit square, say a little triangle, gets transformed into a smaller (contracted) triangle that is similar to the original.

Having defined how f transforms the unit square and its interior, we now have a function that takes its inputs as shapes inside the unit square and transforms them to other shapes inside the unit square in a well-defined way. For example, Figure 8.15(c) shows a cat-like shape C; to see how it is transformed by f into its image $f(C)$, we imagine it being painted onto the unit square and going along for the ride, as in Figure 8.15(d). The transformed cat $f(C)$ is half the size of the original, it's located in the same relative position inside the transformed square, and of course it gets flipped around backwards with the square, so its tail appears to point to the opposite side of its body.

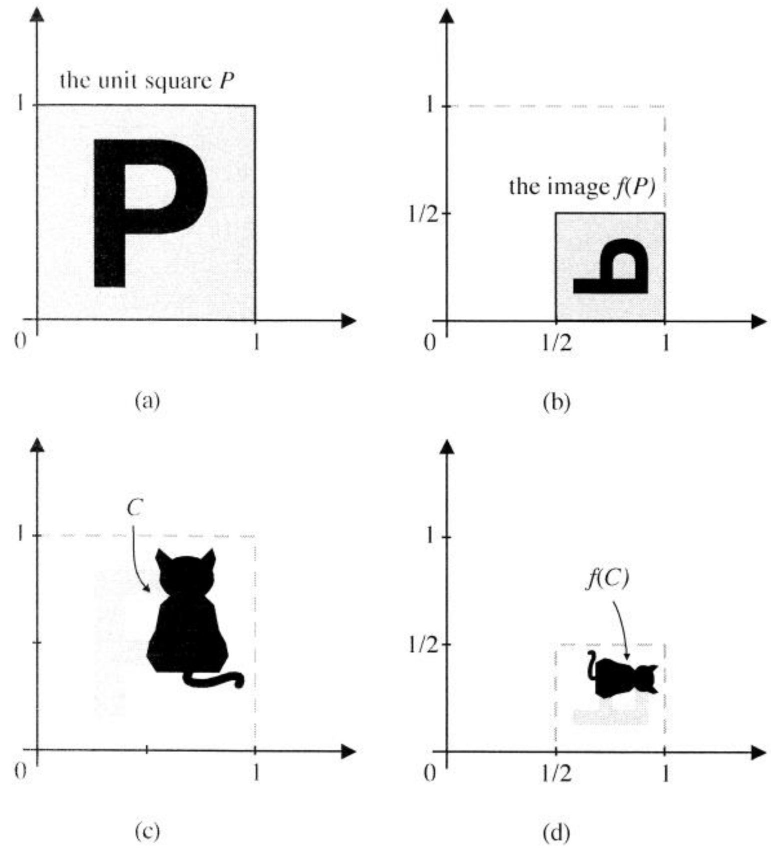


Figure 8.15. How the function f transforms the unit square P and the cat C. Notice that because the unit square gets flipped (reflected) the cat does too, so its tail points in the opposite direction.

We'll deal with several functions at once (that's why these are called "Function Systems"), and we'll repeatedly apply (iterate) the functions over and over (that's why this section is called "Iterated Function Systems"). By this we mean that we will apply a function to a shape, then apply the same or another function to the resulting shape, and so on.

An example of the repeated application of functions appears in Figure 8.16. Parts (a) and (b) illustrate a new function g that also transforms shapes inside the unit square. The function g transforms the unit square P by shrinking it towards the origin by a factor of $1/2$. In Figure 8.16(c) we then transform $g(P)$ with the function f of Figure 8.15 to obtain the set $f(g(P))$ ("f of g of P"), which is also written as $f \circ g(P)$. That's because we can think of this application of two functions as a single function denoted by $f \circ g$, called the *composition* of two functions.

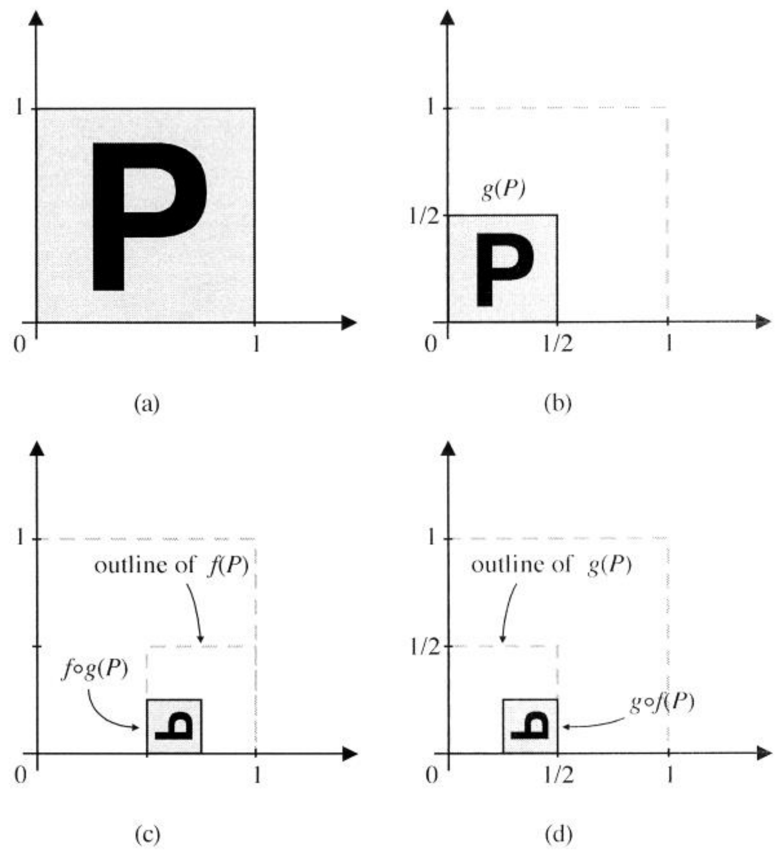


Figure 8.16. Parts (a) and (b) illustrate the function g. Parts (c) and (d) illustrate compositions involving f and g, where f is the function illustrated in Figure 8.15. Notice that the images $f \circ g(P)$ and $g \circ f(P)$ are not the same. Can you sketch $g \circ g(P)$ and $f \circ f(P)$?

Notice that the notation $f \circ g(P)$ tells us to first apply g to P, then apply f to the result. In Figure 8.16(d) we apply the functions in the reverse order to obtain $g \circ f(P)$; notice that $g \circ f(P)$ and $f \circ g(P)$ are different, hence $g \circ f$ and $f \circ g$ are different functions. Can you sketch $g \circ g(P)$ and $f \circ f(P)$?

In our modern world we use function compositions all the time. Let's say you take a digital photo T of a tree and you turn the camera sideways to get the whole thing in the photo. When you transfer it to your computer, the photo T appears sideways, so you rotate it with a function r provided by your computer's image processing software. Now you can enjoy looking at the corrected photo $r(T)$ on your computer. Later you want to email the photo to a friend, but it's too big—it takes up too much memory—so you shrink it to a smaller size with another function s. What you send your friend is the image $s \circ r(T)$.

Now let's look at an Iterated Function System. Consider the three functions f_1, f_2, f_3 illustrated in Figure 8.17. The first iteration in Figure 8.17(b) shows that all three functions shrink the unit square P to half its size and then move it somewhere. The function f_1 moves the reduced version of P to the top center of the square, f_2 moves it to the lower left corner, and f_3 moves it to the lower right corner.

In Figure 8.17(c)—the second iteration—we have labeled the sets $f_1 \circ f_1(P)$ and $f_1 \circ f_3(P)$. Which square is the set $f_3 \circ f_1(P)$? Which is $f_2 \circ f_3(P)$? In the third iteration in Figure 8.17(d) are even smaller squares, representing sets like $f_1 \circ f_3 \circ f_3(P)$—which square is that? The tiny squares in the fourth iteration in Figure 8.17(e) represent images of P under fourfold function compositions, such as $f_2 \circ f_1 \circ f_1 \circ f_3(P)$.

Now we can see the value of functions and function compositions in fractal geometry. As we use more and more complex compositions in Figures 8.17(a)—(e), the resulting collection of squares more and more closely resembles an isosceles Sierpiński triangle whose altitude and base are equal. Figure 8.17(f) depicts the 6th iteration. It consists of $3^6 = 729$ images of P, each $(1/2)^6 = 1/64$ the width of P. All the images were easily created in the computer drawing program Lineform, as they could have been in, say, Adobe Illustrator. (There is also another important way computers generate fractals with Iterated Function Systems, called the *Random Iteration Algorithm*, described in detail in Michael Barnsley's book *Fractals Everywhere*[6]. It makes use of the formulas that define functions like f_1, f_2, and f_3.)

[6]Barnsley, M., 1988, *Fractals Everywhere* (San Diego: Academic Press).

Notice how closely Figure 8.17(f) resembles the isosceles Sierpiński triangle A in Figure 8.18. The set A just fits inside the unit square, so its altitude and base both have length 1. Using more iterations would have increased the resemblance further. The set A is called the *attractor* of the IFS; you can see how the successive iterations are "attracted" to it. In terms of functions, the true fractal A consists of three smaller copies of itself; it is the union of the sets $f_1(A)$ (the top part), $f_2(A)$ (the bottom left part), and $f_3(A)$ (the bottom right part). In set notation, $S = f_1(A) \cup f_2(A) \cup f_3(A)$.

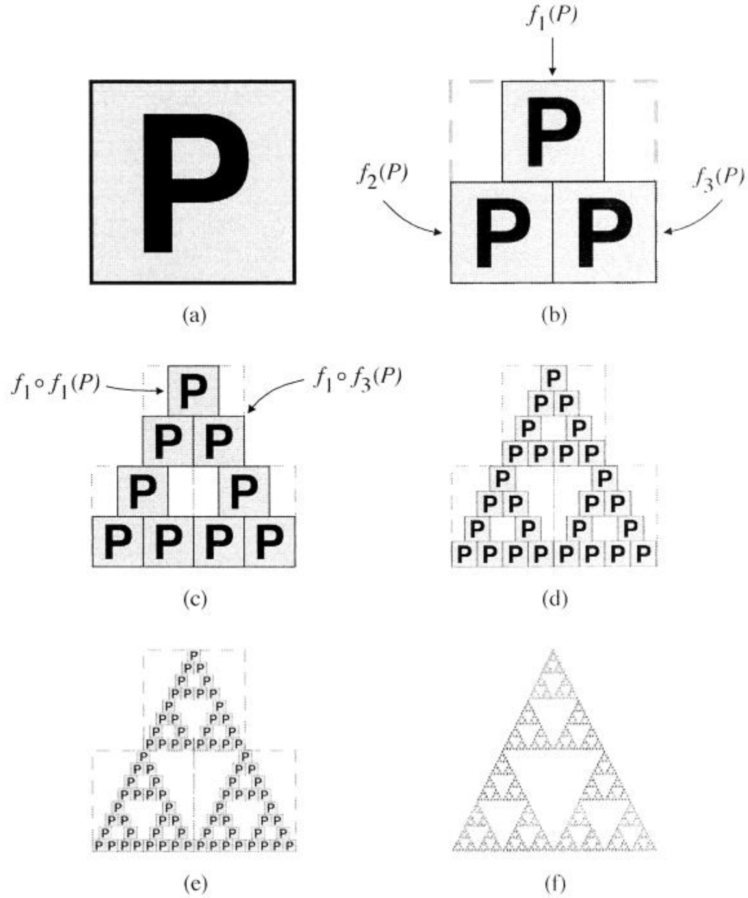


Figure 8.17. Iterating functions to approximate an isosceles Sierpiński triangle with successively smaller and more numerous images of the unit square P. After 1 iteration (b) there are $3^1 = 3$ images of P, each $(1/2)^1 = 1/2$ the width of P. After 2 iterations (c) there are $3^2 = 9$ images of P, each $(1/2)^2 = 1/4$ the width of P. After 3 iterations (d) there are $3^3 = 27$ images of P, each $(1/2)^3 = 1/8$ the width of P. After 4 iterations (e) there are $3^4 = 81$ images of P, each $(1/2)^4 = 1/16$ the width of P. Part (f) shows the result after 6 iterations: $3^6 = 729$ images of P, each $(1/2)^6 = 1/64$ the width of P. All the images were created in a standard computer drawing program.

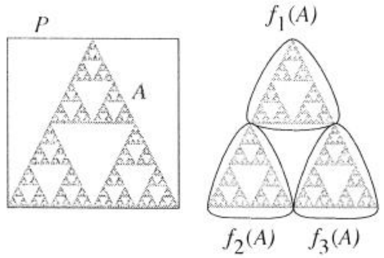


Figure 8.18. The isosceles Sierpiński triangle A on the left just fits inside the unit square P. As suggested on the right, A satisfies

$$S = f_1(A) \cup f_2(A) \cup f_3(A).$$

In the next example, we're going to change one of the functions a little and see what happens. In the previous example, we assumed that the functions preserved orientation and direction: there were no rotations and no flips. In Figure 8.19, we'll forego that assumption, but we'll still have f_1 move the reduced version of P up and center, f_2 move it down and left, and f_3 move it down and right. The difference is that f_1 flips the reduced version of P and rotates it 90°.

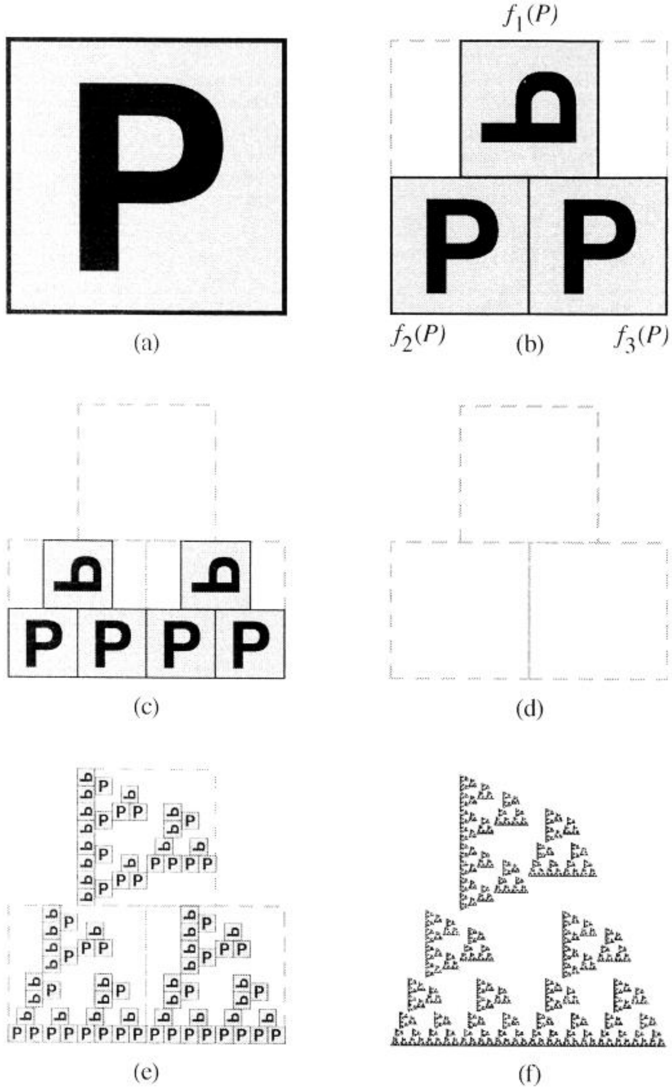


Figure 8.19. Part (a) shows the first iteration of an IFS. Your job is to draw the rest of the second iteration in (c) and all of the third iteration in (d). Part (e) shows the fourth iteration, and part (f) is the attractor of the IFS.

In Figure 8.19 we draw the first iteration (b) and part of the second iteration (c) of the IFS. Your job is to finish the second iteration, and draw the third iteration (d). Part (e) shows the fourth iteration, and (f) shows the attractor of the IFS.

In Figure 8.20 below, we don't tell you the steps we took to draw the fractals; we ask you to figure that out yourself. Each of the pictures in Figure 8.20 is the attractor of an Iterated Function System (each with three functions that send images to the top center, lower left, and lower right). Each attractor is shown inscribed in the unit square. Determine the direction of the P's in the first iteration of the process—for example, as in Figure 8.19(b). Assume that the initial stage is a P in the usual position as in Figure 8.19(a). In cases where there is a lot of symmetry, there can be more than one correct answer.

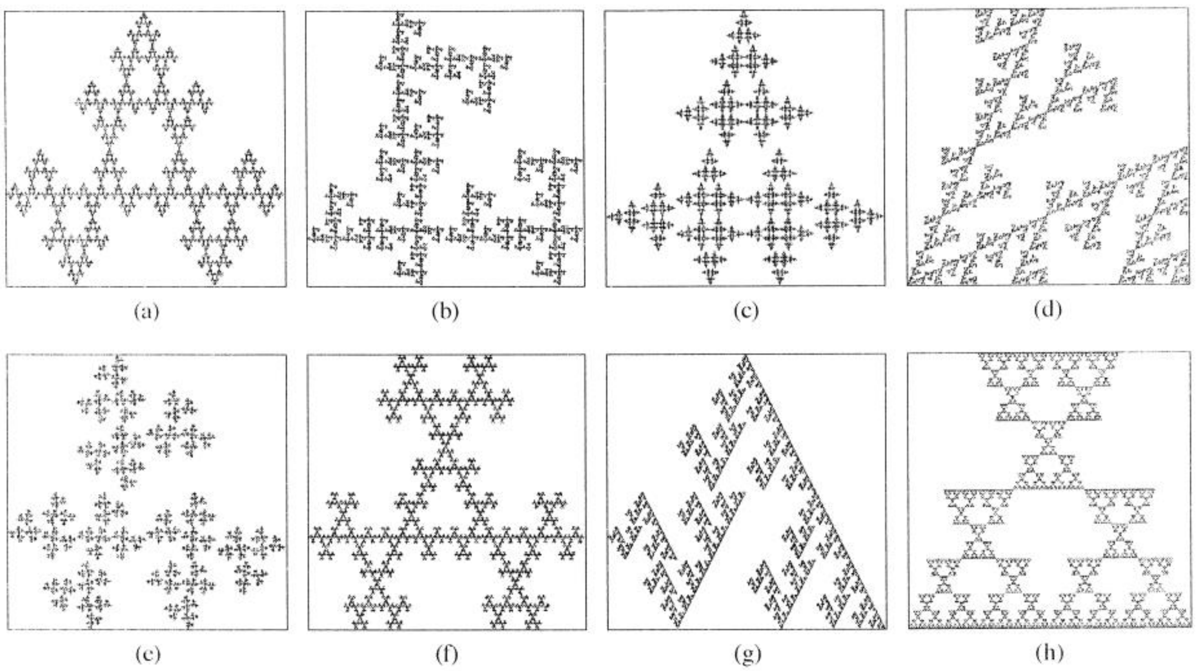


Figure 8.20. Can you guess the IFS in each case?