Problems in Statistical Mechanics VII. Elsevier, Amsterdam, New York, 69–110

Greene J.M. (1979) A Method for Determining a Stochastic Transition. J. Math. Phys. 20:1183

3.3 Fractal Lattices

Consider a simple game of dice and a simple geometric iteration. Both lead to a strange structure that is less than an area, but more than a line – a so-called fractal with a dimension $D = \log 3/\log 2 = 1.58\ldots$. Surprisingly, fractals are formed very frequently in nature. Coastlines, mountain ranges, blood vessels, courses of rivers, variations of stock prices, and water levels in rivers, can all be described by a fractional dimension. In addition to the geometric game, in this section we will learn about aggregates and percolation clusters as examples of these remarkable self-similar objects.

We first want to deal with the definition of the fractal dimension. Then, we will present two programs that use different methods to construct the Sierpinski gasket, which is often presented as a standard example of fractals.

Physics

The spatial dimension D of an object can be determined from the relation between the number of its components that make up its mass M and its linear dimension L:

$$M \propto L^D \, . \tag{3.23}$$

Thus, by comparing the masses of two similar cubes with sides L and $2L$, one obtains

$$\frac{M_2}{M_1} = \left(\frac{2L}{L}\right)^D = 2^D \tag{3.24}$$

with $D = 3$. For squares, the result is obviously $D = 2$, and for lines $D = 1$. L can also be the radius of spheres or disks, or any other characteristic length of the object under consideration; of course the mass is proportional to the number of particles. In any case we have

$$D = \lim_{L\to\infty} \frac{\log M}{\log L} \, . \tag{3.25}$$

There is yet another way of determining the dimension D. For this method, we completely cover the object with small cubes whose edges have the length ε. Let $N(\varepsilon)$ be the smallest number of these cubes needed. Then $N(\varepsilon) \propto \varepsilon^{-D}$, and consequently

$$D = -\lim_{\varepsilon\to 0} \frac{\log N(\varepsilon)}{\log \varepsilon} \, . \tag{3.26}$$

To cover a cube, we need $(L/\varepsilon)^3$ boxes, whereas the number is $(L/\varepsilon)^2$ for a square and $(L/\varepsilon)^1$ for a line.

So, if we want to determine the length of a border on a map, for example, we can take a small ruler of length ε and determine the number $N(\varepsilon)$ of steps we need to measure the borderline. Then, we would expect

$$L = \varepsilon N(\varepsilon) \tag{3.27}$$

to be a measure of that length, as we think of a border as a one-dimensional object with $D = 1$. By this method, the Spaniards have obtained a value of 987 km for the length of their country's border with Portugal, whereas the Portuguese have determined the same border to be 1214 km long. A smaller ruler ε apparently requires more steps than L/ε to cover the border, therefore, D cannot be equal to 1. Consequently, (3.27) will yield a value for L that depends on ε. A more detailed analysis of the data reveals the following: A border is a fractal with $1 < D < 2$; therefore, (3.27) yields $L \propto \varepsilon^{1-D}$, i.e., a length that seems to diverge as $\varepsilon \to 0$. The border does not have a well-defined length at all! Of course, this can only be true for scales ε that are larger than the smallest (e.g., the surveyor's rods used) and smaller than the largest length involved (e.g., the size of Portugal).

Another simple example, in which the fractal dimension D does not agree with the embedding dimension d, that is, the dimension of the space in which the object is located, is a random motion in d-dimensional space. This *random walk*, as it is commonly called, is also discussed as a simple model of a polymer molecule that consists of a large number of monomers. We assume in this discussion that the distance between adjacent monomers is constant, but that the angles between three consecutive monomers are independent and randomly distributed. Then, the vectors $\boldsymbol{r}_i$ connecting adjacent monomers are random vectors, whose properties are easily calculated. If $\langle \ldots \rangle$ denotes the average over many molecular configurations, we have:

$$\langle \boldsymbol{r}_i \rangle = 0 \, , \quad \langle \boldsymbol{r}_i \cdot \boldsymbol{r}_j \rangle = \langle \boldsymbol{r}_i \rangle \cdot \langle \boldsymbol{r}_j \rangle = 0 \ \text{ for } i \neq j \, , \quad \text{and } \ \langle \boldsymbol{r}_i^2 \rangle = a^2 \, ,$$

where a is the distance between adjacent monomers. For the vector $\boldsymbol{R}$ between the beginning and end of a molecule consisting of $N + 1$ monomers, we have

$$\boldsymbol{R} = \sum_{i=1}^{N} \boldsymbol{r}_i$$

and therefore

$$\langle \boldsymbol{R} \rangle = \sum_{i=1}^{N} \langle \boldsymbol{r}_i \rangle = 0$$

and

$$\langle \boldsymbol{R}^2 \rangle = \sum_{i=1}^{N} \sum_{j=1}^{N} \langle \boldsymbol{r}_i \cdot \boldsymbol{r}_j \rangle = \sum_{i=1}^{N} \langle \boldsymbol{r}_i^2 \rangle + \sum_{i \neq j} \langle \boldsymbol{r}_i \cdot \boldsymbol{r}_j \rangle = N a^2 \, .$$

Since the mass M increases proportionally to the number of N of monomers and the linear dimension L is proportional to $\sqrt{\langle \boldsymbol{R}^2 \rangle}$, this means that

$$M \propto L^2$$

and consequently, using (3.23), $D = 2$. Such a random walk, therefore, is always a two-dimensional object, independent of the dimension d of the space in which it is generated. The fractal dimension is reduced, however, if the size of the individual monomers is taken into account. We will discuss this effect in Sect. 5.4.

Algorithm and Result

Now we want to take a closer look at the two games mentioned at the beginning of this section. First, the algorithm for the game of dice:

- Select three points $\boldsymbol{p}_1, \boldsymbol{p}_2$, and $\boldsymbol{p}_3$ and a starting point $\boldsymbol{q}_0$ arbitrarily distributed in the plane.
- Starting from the point $\boldsymbol{q}_n$, construct the next point $\boldsymbol{q}_{n+1}$ by the following method: randomly (e.g., by casting a die) pick one of the three points $\boldsymbol{p}_i$. Then determine the center of the line connecting $\boldsymbol{q}_n$ and $\boldsymbol{p}_i$:

$$\boldsymbol{q}_{n+1} = \frac{(\boldsymbol{q}_n + \boldsymbol{p}_i)}{2} \ .$$

- Iterate the equation above ad infinitum.

What pattern do the points $\{\boldsymbol{q}_0, \boldsymbol{q}_1, \boldsymbol{q}_2, \ldots\}$ generate in the plane? The algorithm, which we write in C for the sake of increased processing speed, is easy to program. We define each point as a structure with two integer variables x and y, representing the pixel coordinates on the screen.

```
struct{int x, int y} qn={20,20}, pw,
      p[3]={{10,10},{MAXX-10,10},{MAXX/3,MAXY-10}} ;
```

The initial values are specified right in the type declaration. Now in each step of the iteration one of the three points p[i] is selected by generating a random number $i \in \{0, 1, 2\}$,

```
pw = p[(3*rand())/RAND_MAX]
```

and a new point is calculated:

```
qn.x = (pw.x + qn.x)/2 ;
qn.y = (pw.y + qn.y)/2 ;
```

Then a pixel is plotted at that position, e.g., on the PC via

```
putpixel(qn.x, qn.y, WHITE);
```

Figure 3.15 shows the result. Surprisingly, the random motion of the point q leads to an obviously regular structure consisting of triangles nested inside each other. The structure is self-similar: all triangles look the same, independent of their size (of course, this is only true for triangles that are much larger than the size of the pixels). But the structure is nearly empty, as each triangle has an infinite number of holes at any scale. After just a few iterations, any point we start with ends up in this strange grid; thus, the framework, also called the *Sierpinski gasket*, is an attractor for the dynamics of the entire plane.

The second geometric game goes as follows:

- Start with a triangle,
- remove a triangle from its center in such a way that three identical triangles remain,
- iterate this ad infinitum for all triangles remaining after each step.

To take advantage of the simple graphics commands, we want to program this in *Mathematica*. Each triangle is described by a list of three points in the plane. The initial triangle is given by:

```
list={{{0.,0.},{.5,N[Sqrt[3/4]]},{1.,0.}}}
```

Each triangle is tripled by first scaling it down by a factor 1/2 and shifting it along two of its edges by the length of the respective edge:

```
triple[d_]:= Block[ {d1,d2,d3},
                    d1={d[[1]],(d[[2]]+d[[1]])*.5,
                               (d[[3]]+d[[1]])*.5};
                    d2=d1+Table[(d1[[3]]-d1[[1]]),{3}];
```

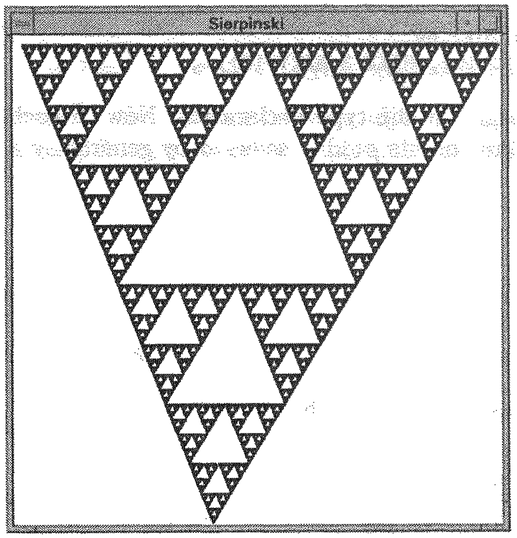


Fig. 3.15. Points in the plane, generated by the game of dice described in the text. The corners of the Sierpinski triangle are given by the points p[0], p[1], and p[2]

```
                d3=d1+Table[(d1[[2]]-d1[[1]]),{3}];
                {d1,d2,d3}
               ]
```

The command Map is used to repeatedly apply this function to the entire list of triangles. Finally, the command Polygon turns the triangles into graphics objects which are drawn by Show:

```
plot1:= Block[ {listtwo,plotlist},
                    listtwo=Map[triple,list];
                    list=Flatten[listtwo,1];
                    plotlist=Map[Polygon,list];
                    Show[Graphics[plotlist],
                         AspectRatio -> Automatic]
              ]
```

Figure 3.16 shows the result. We obviously obtain the same structure as in the previous game of dice, but now, we can easily convince ourselves that the self-similar framework is a fractal in the limit of infinitely many iterations. If we cut in half an edge of the triangle of length L, its mass is obviously reduced by a factor 3. After t steps, we have

$$M_t = 3^{-t}M \ , \quad L_t = 2^{-t}L \ .$$

This leads to

$$D = \lim_{t\to\infty} \frac{\ln M_t}{\ln L_t} = \lim_{t\to\infty} \frac{(-t)\ln 3 + \ln M}{(-t)\ln 2 + \ln L} = \frac{\ln 3}{\ln 2} \ .$$

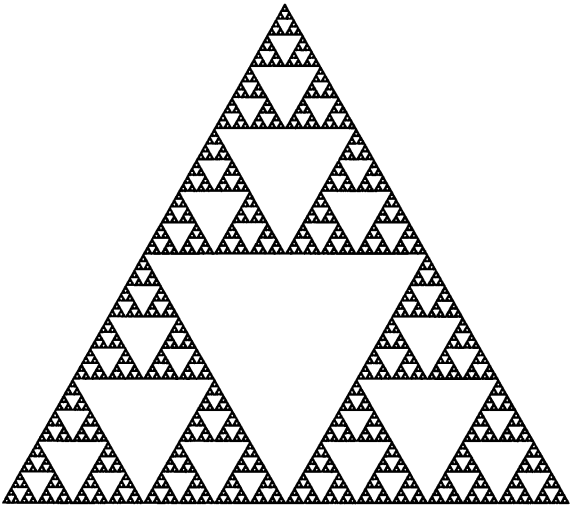


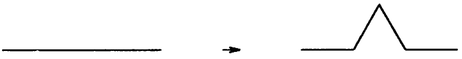
Fig. 3.16. By repeated removal of triangles, a fractal of dimension $D \simeq 1.58$ is generated

Thus, the Sierpinski gasket has the fractional dimension $D = 1.58\ldots$. While it does not fill an area, it is indeed more than a line. Its density $\rho = M/L^2$ is zero, and any triangle, however small it may be, looks the same as a big one.

It is remarkable that nature uses fractals in order to effectively supply blood to every part of the body, for example, or to generate spacesaving direct connections between 10^{11} communicating nerve cells in the brain. So far, technology has not been able to make much use of fractals. The book by M. Schroeder, however, does describe the use of fractal reflectors in acoustics.

Exercise

A Koch curve is defined by the following iteration: The central third of a line is cut out and replaced by a protrusion whose two edges each have the same length as the segment just removed, as shown in the following drawing.



Start with an equilateral triangle, and iterate the construction above as often as possible. As a result, you will obtain the Koch snowflake.

Literature

Ott E. (1993) Chaos in Dynamical Systems. Cambridge University Press, Cambridge, New York

Peitgen H.O., Saupe D. (1988) The Science of Fractal Images. Springer, Berlin, Heidelberg, New York

Schroeder M. (1991) Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise. W.H. Freeman, New York

Schuster H.G. (1995) Deterministic Chaos: An Introduction. VCH, Weinheim, New York

Wagon S. (1991) Mathematica in Action. W.H. Freeman, New York

3.4 Neural Networks

Can modern computers simulate the human brain? Anybody who thinks about it will probably reach the conclusion that the answer to this bold question must be a definite no. While a lot of facts are known nowadays about the way our brain works, it is still unknown how the exchange of information between 10^{11} cells (neurons) and their 10^{14} contact points (synapses) can result in processes as commonplace to us as learning, recognition, and thinking.