Repeat the experiment many times and plot the frequency distribution of the sums in a histogram. Compare the result to a Gaussian distribution with mean m and variance σ^2,

$$P(s) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[\frac{-(s-m)^2}{2\sigma^2}\right] .$$

Verify the agreement for $N = 10$, 100, and 1000.

Literature

Binder K., Heermann D.W. (1992) Monte Carlo Simulation in Statistical Physics: An Introduction. Springer, Berlin, Heidelberg, New York

Gould H., Tobochnik J. (1996) An Introduction to Computer Simulation Methods: Applications to Physical Systems. Addison-Wesley, Reading, MA

Knuth D.E. (1997) The Art of Computer Programming, Vols. I, II, and III. Addison-Wesley, Reading, MA

Marsaglia G., Zaman A. (1994) Some Portable Very-long Period Random Number Generators. Computers in Physics 8:117

Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P. (1992) Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, Cambridge, New York

5.2 Fractal Aggregates

In Sect. 3.3 we have constructed objects that are more than a line but less than an area. These objects are characterized by a fractional dimension and are called fractals. They are self-similar, a part looks similar to the whole object, and they can be generated according to a simple deterministic rule.

Do such objects exist in nature? Indeed, many structures in our environment can be characterized quantitatively with the help of fractal dimensions. Coastlines, mountain ranges, courses of rivers, blood vessels, nerve cells, surfaces of leaves, variations of stock prices, and many other phenomena can all be described by power laws of the kind "the mass M increases as the length L to the power D." Natural structures, however, are not as strictly regular as the Sierpinski gasket from Sect. 3.3 but are the results of random processes.

Aggregates are a simple example of such fractals arising from the interplay of chance and regularity. If particles diffuse towards a nucleus and attach to it, the result is a loose grainy structure with a fractional dimension between 2 and 3. Using a simple computer model that was proposed and investigated in 1981 by Witten and Sander, we want to demonstrate how such fractal objects can be generated with little effort on the computer.

Physics

In the following, we describe a model of a growth process, in which the undirected diffusion of particles before their attachment plays the decisive role. The deposition of material from an electrolytic solution at very low voltage can serve as an example for this. One only has to make sure that the random motion of the diffusing ions is more important than their drift due to the electric field. Then the deposition of material on the electrode does not lead to a compact object; instead, a filigree-like cluster emerges, which we can describe by a fractal dimension D. The value of D is about 5/6 of the dimension of the space in which the growth process takes place.

The process is called *diffusion limited aggregation* (DLA). Though DLA is easily simulated and analyzed on the computer, to our knowledge there is no analytic theory for it to date. By contrast, there is a well-developed mathematical theory for the diffusion problem, i.e., the random motion of a particle. We want to briefly elaborate on a few statements and results which we will need in the algorithm section.

We consider a random walk on a square lattice or, more generally, a d-dimensional cubic lattice. We designate the lattice vectors by $\boldsymbol{x}$ and the vectors connecting a lattice site to its $2d$ nearest neighbors by $\Delta\boldsymbol{x}_i, i = 1, 2, \ldots, 2d$. The motion of our random walker, which is at the lattice site $\boldsymbol{x}$ at time t, is determined by the following instructions: randomly select one of the nearest neighbor sites $\boldsymbol{x} + \Delta\boldsymbol{x}_i$ and jump there during the next time step Δt. If we now put many particles onto the lattice, all of which jump according to these same random rules, and designate the fraction that will be at the site $\boldsymbol{x}$ at time t by $u(\boldsymbol{x}, t)$, we obtain the following balance equation:

$$u(\boldsymbol{x}, t + \Delta t) - u(\boldsymbol{x}, t) = \frac{1}{2d} \sum_{i=1}^{2d} \left(u(\boldsymbol{x} + \Delta\boldsymbol{x}_i, t) - u(\boldsymbol{x}, t) \right) . \tag{5.8}$$

We have subtracted the term $u(\boldsymbol{x}, t)$ on both sides to make the fact that this is just the discretized version of the diffusion equation more obvious. To see this, we divide (5.8) both by a suitable volume, to obtain a density, and by Δt and let all small quantities, in particular all $\Delta\boldsymbol{x}_i$, go to 0 in proportion to $\sqrt{\Delta t}$. This leads to the equation

$$\frac{\partial u}{\partial t} = \eta \nabla^2 u \tag{5.9}$$

for the probability density, with a diffusion constant η.

We want to discuss two characteristic solutions of this diffusion equation. The first is the solution of an initial-value problem which contains information about the manner in which a random walk spreads out on average and the velocity with which that happens. The second is a stationary solution from which we can obtain an estimate of the fractal dimension D of the DLA cluster to which the diffusing particles attach.

Equation (5.9) with the initial condition $u(\boldsymbol{x}, t_0) = \delta(\boldsymbol{x} - \boldsymbol{x}_0)$ and the boundary condition $u \to 0$ for $|\boldsymbol{x}| \to \infty$ can be solved by a Fourier transformation. For $t \geq t_0$ one obtains

$$u(\boldsymbol{x}, t) = \left[4\pi\eta(t - t_0)\right]^{-d/2} \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{x}_0|^2}{4\eta(t - t_0)}\right) . \tag{5.10}$$

From this we see that, if the jumping particle is at the point $\boldsymbol{x}_0$ at the time t_0, then the probability of finding it at $\boldsymbol{x}$ at a later time t only depends on the distance $r = |\boldsymbol{x} - \boldsymbol{x}_0|$, not on the direction. This does not come as a big surprise, as it is just a reflection of the isotropy of the diffusion process which we have put into the equation by treating all directions identically. The mean value $\langle(\boldsymbol{x} - \boldsymbol{x}_0)^2\rangle$ is

$$\left\langle(\boldsymbol{x} - \boldsymbol{x}_0)^2\right\rangle = 2d\eta(t - t_0) , \tag{5.11}$$

which means that the average size of the random path increases as $\sqrt{t - t_0}$. Except for the proportionality factor, this law is independent of the dimension d of the space in which the random motion takes place.

The model of Witten and Sander describes a diffusion limited growth process which proceeds according to the following rules. One starts with a minimal cluster, usually a single particle at the origin of the coordinate system. Then, another particle starts diffusing freely at a great distance and at a randomly selected direction, until it hits the nucleus at the origin and attaches to it. Then the next particle starts far out, again at a randomly selected direction, and so on. When averaged over many realizations of this experiment, the central cluster will grow in a radially symmetric way. For any given realization, we designate the maximum radial size of the cluster by $R_{\text{max}}(t)$ and determine its fractal dimension D from the relationship between $R_{\text{max}}(t)$ and its mass $M(t)$ which is given by the number of particles attached by the time t:

$$M \propto R_{\text{max}}^D . \tag{5.12}$$

If we want to describe this model by a probability density $u(\boldsymbol{x}, t)$ of the diffusing particles, we have to take into account the boundary condition that the density vanishes at the newly created cluster. Additional particles are deposited on the cluster with a probability that is proportional to the particle current. We can, however, approximate this by assuming that all particles that get to within a certain capture radius R_{c} are removed by absorption. But according to Witten and Sander, R_{c} is proportional to R_{max}, and we do not commit an error in the following dimensional consideration if we replace R_{c} by R_{max}. Moreover, we can assume that the time that characterizes the diffusion process, e.g., the time it takes a diffusing particle to move from one site to the next, is much smaller than the time that characterizes the growth of the cluster. In other words, we are looking for a radially symmetric, time-independent solution of the diffusion equation (5.9) with the boundary condition $u(R_{\max}) = 0$. The equation

$$\Delta u = \frac{1}{r^{d-1}} \frac{\partial}{\partial r} r^{d-1} \frac{\partial}{\partial r} u = 0 \, , \tag{5.13}$$

however, is easily integrated. For $d > 2$, we get the solution

$$u\,(r) = u_0 \left(1 - \left(\frac{R_{\max}}{r} \right)^{d-2} \right) \, , \tag{5.14}$$

and u is proportional to $\ln(r/R_{\max})$ for $d = 2$. In any case, the radial component of the current density, $j_r = -\eta \partial u / \partial r$, is proportional to $R_{\max}^{d-2}/r^{d-1}$. By integrating this over the surface we obtain the result that the total particle current J absorbed by the cluster is proportional to $R_{\max}^{d-2}$. On the other hand, J is proportional to the increase in mass, which leads to the following equation:

$$R_{\max}^{d-2} \propto J = \frac{\mathrm{d}M}{\mathrm{d}t} = \frac{\mathrm{d}M}{\mathrm{d}R_{\max}} \frac{\mathrm{d}R_{\max}}{\mathrm{d}t} \propto R_{\max}^{D-1} v \, . \tag{5.15}$$

This means that the velocity v with which the cluster size increases is proportional to $R_{\max}^{d-1-D}$. If we add the plausible assumption that v will in no case increase with increasing cluster size, then the exponent $d - 1 - D$ must not be positive, i.e., the relation $d-1 \leq D$ must hold. On the other hand, the DLA cluster is embedded in d-dimensional space, so we obtain the relation

$$d - 1 \leq D \leq d \tag{5.16}$$

which has been confirmed for $d = 2, 3, \ldots, 8$ by numerical simulations.

Algorithm

We want to generate a DLA cluster on a two-dimensional square lattice. To accomplish this, we occupy the site at the center of the lattice, place another particle on the lattice at a distance R_{s}, and let it diffuse until it either attaches to the occupied site or exceeds a distance $R_{\mathrm{k}} > R_{\mathrm{s}}$ from that site. Consequently, the algorithm will be as follows:

1. Start with an empty square lattice and position a particle on the central lattice site.
2. Let the actual size of the aggregate be defined by the radius $R_{\max}$. Place a particle on a randomly selected site at a distance $R_{\mathrm{s}} \gtrsim R_{\max}$ from the origin and let it jump to a randomly selected adjacent site.
3. If the particle reaches a site adjacent to the aggregate, it is added and $R_{\max}$ is increased if necessary. If a particle exceeds the distance R_{k} from the origin ($R_{\mathrm{k}} > R_{\mathrm{s}} > R_{\max}$), it is annihilated ("killed").
4. Iterate steps 2 and 3 and calculate $D = \ln N / \ln R_{\max}$, where N is the number of particles in the aggregate.

The sketch in Fig. 5.3 serves to illustrate the smallest circle around the aggregate, with radius R_{max}, the start circle, and the annihilation circle. However, the introduction of an annihilation circle skews the properties of the aggregate, as it neglects the fact that the diffusing particle can get back to the aggregate even from a distance $R > R_{\text{k}}$. Consequently, our algorithm describes the actual problem only in the limit $R_{\text{k}} \to \infty$. Large values of R_{k}, though, mean long computing times, and we do not expect any significant deviations for finite values of R_{k}. One can analytically calculate the probability that the particle will return to a given position on the start circle and use this to reset the particle in one step. This avoids the annihilation circle. For large aggregates, this is worth the effort, but we choose not to go into this here.

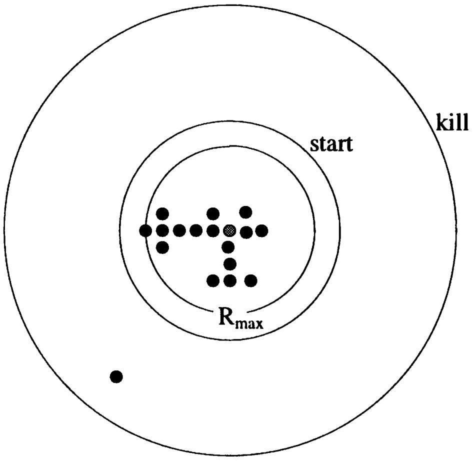


Fig. 5.3. The diffusing particle starts at the middle circle. If it leaves the outer circle, it is annihilated, and a new particle is put on the start circle

Moreover, the algorithm can be accelerated significantly by drawing a circle that just touches the aggregate around any particle that moves away from the cluster. Since, according to (5.10), the particle will get to each point on this circle with the same probability, one can randomly put it on the circle without simulating the time-consuming diffusion. This is repeated until the particle has reached a site adjacent to the aggregate or left the annihilation circle. This leads to the following algorithm:

1. Start with an empty square lattice and position a particle on the central lattice site.
2. Let the actual size of the aggregate be defined by the radius R_{max}. Place a particle on a randomly selected site at a distance $R_{\text{s}} \gtrsim R_{\text{max}}$ from the origin and let it jump to a randomly selected adjacent site.
3. If the particle reaches a site adjacent to the aggregate, it is added and R_{max} is increased if necessary. If the particle exceeds the distance $R_{\text{d}} > R_{\text{s}} \gtrsim R_{\text{max}}$ from the origin – the actual distance is denoted by R – it

is positioned on a randomly selected site on a circle around its current position with radius $(R - R_s)$. If the particle exceeds the distance $R_k > R_d > R_s \gtrsim R_{\max}$, it is annihilated.

4. Iterate steps 2 and 3 and calculate $D = \ln N / \ln R_{\max}$, where N is the number of particles in the aggregate.

Figure 5.4 is intended to clarify this algorithm once more. For a fast simulation of the DLA cluster, we use the programming language C. The square lattice is described by the two-dimensional array xf[rx][ry] whose elements can take the values 1 and 0. The assignment xf[rx][ry]=1 means that there is a cluster particle at the site with the coordinates rx and ry, whereas the sites with xf=0 are available for diffusion. For very large aggregates it would certainly make sense not to store the entire lattice, only the aggregate and boundary sites, but to avoid the bookkeeping that would be necessary in this case, we choose the easier way.

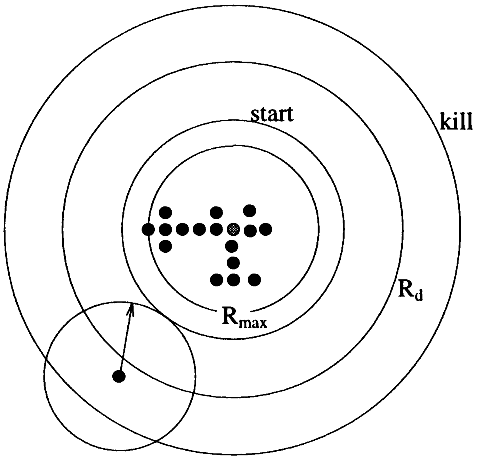


Fig. 5.4. The same situation as in Fig. 5.3, but outside the circle with radius R_d the particle covers the distance $R - R_s$ in one step

The start site on the circle of radius R_s is selected by the function occupy():

```
void occupy()
{
  double phi;
  phi=(double)rand()/RAND_MAX*2.*pi;
  rx=rs*sin(phi);
  ry=rs*cos(phi);
}
```

Jumping to one of the four adjacent sites is controlled via a randomly selected number 0, 1, 2, or 3:

```
void jump()
{
   int r;
   r=random(4);
   switch(r)
   {
     case 0: rx+=1;break;
     case 1: rx+=-1;break;
     case 2: ry+=1;break;
     case 3: ry+=-1;break;
   }
}
```

After each jump, the program has to check whether the particle is annihilated ($R > R_k$), whether it has reached a site adjacent to the aggregate, or whether a short jump ($R < R_d$) or a long "circle jump" ($R \geq R_d$) needs to be performed in the next step. These four "ifs" in step 3 of the algorithm above are tested by the function `check()`:

```
char check()
{
  double r,x,y;
  x=rx;
  y=ry;
  r=sqrt(x*x+y*y);
  if (r > rkill)   return 'k';
  if (r >= rd)     return 'c';
  if (xf[rx + 1 + lmax/2][ry + lmax/2] +
      xf[rx - 1 + lmax/2][ry + lmax/2] +
      xf[rx + lmax/2][ry + 1 + lmax/2] +
      xf[rx + lmax/2][ry - 1 + lmax/2] > 0)
    return 'a';
  else
    return 'j';
}
```

`lmax` is the size of the drawing window. The following function adds the particle to the aggregate:

```
void aggregate()
{
  double x,y;
  xf[rx+lmax/2][ry+lmax/2]=1;
  x=rx;y=ry;
  rmax= max(rmax,sqrt(x*x+y*y));
  if(rmax>lmax/2.-5.) {printf("\7");stop=1;}
  circle(4*rx+340,4*ry+240,2);
}
```

For the circle jump, a random vector of length $R - R_s$ is added to the position of the particle:

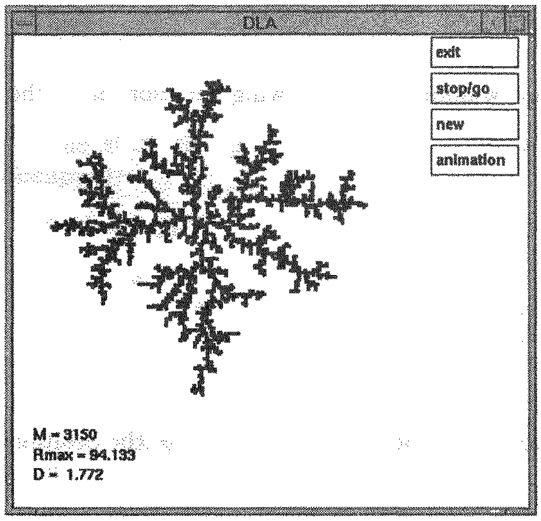
```
void circlejump()
{
  double r,x,y,phi;
  phi=(double)rand()/RAND_MAX*2.*pi;
  x=rx; y=ry; r=sqrt(x*x+y*y);
  rx+=(r-rs)*sin(phi);
  ry+=(r-rs)*cos(phi);
}
```

The particle's coordinates rx and ry and the aggregate array xf[rx][ry] are declared globally before the main routine main(), so all functions have access to them. With the five functions above, the essential part of the main routine can then be written as:

```
while(stop==0)
 {
  if(kbhit()) stop=1;
  switch(check())
   {
    case 'k':occupy();jump();break;
    case 'a':aggregate();occupy();jump();break;
    case 'j':jump();break;
    case 'c':circlejump();break;
   }
 }
```

Result

In Fig. 5.5 the result of the simulation can be seen. On the workstation, 3150 particles have attached themselves to a fractal aggregate within a minute.



```
M = 3150
Rmax = 94.133
D = 1.772
```

Fig. 5.5. DLA cluster on a square lattice

With a maximum extent of 94 lattice constants, one obtains the estimate $D = \ln 3150 / \ln 94 \simeq 1.77$ for the fractal dimension.

Additionally the simulations show that, thanks to the introduction of the long jumps for $R > R_\mathrm{d}$, the processing time does not increase significantly if the radius of the annihilation circle is enlarged.

Exercises

Improved dimensional analysis. In determining the fractal dimension we have, so far, neglected the fact that the cluster being generated is not complete yet. While the "arms" of the cluster continue to form, additional particles can still reach its interior regions. By arbitrarily stopping the simulation as soon as a certain maximum radius is reached, we neglect those particles. The actual cluster should be slightly denser, the fractal dimension slightly higher.

Calculate the mass of the cluster neglecting all particles whose distance from the origin is greater than r. Plot the mass as a function of r in a log–log diagram and attempt to interpret the behavior.

Varying the model parameters. It is interesting to observe in which way the form and fractal dimension of the cluster change if the original DLA model is varied slightly.

Introduce an additional parameter p_stick into the program `dla.c`, which represents the probability that a particle which arrives at a site adjacent to the cluster will actually stick to it.

If the particle is not attached, it shall continue the diffusion process until it hits the cluster again, at which point it will again have a probability p_stick of getting stuck. In continuing the diffusion process one has to make sure, of course, that the particle is prohibited from jumping to a site that is already occupied.

Literature

Gould H., Tobochnik J. (1996) An Introduction to Computer Simulation Methods: Applications to Physical Systems. Addison-Wesley, Reading, MA

Sander E., Sander L.M., Ziff R.M. (1994) Fractals and Fractal Correlations. Computers in Physics 8:420

5.3 Percolation

In an ideal crystal, all atoms are arranged on a lattice in a regular way. In this case we can use Fourier transformations to describe the periodic structures as