Project 13.16 Continuum DLA

(a) In the continuum (off-lattice) version of diffusion limited aggregation. the diffusing particles are assumed to be disks of radius a. A disk executes a random walk until its center is within a distance $2a$ of the center of a disk that is already a part of the DLA cluster. At each step the walker changes its position by $(r\cos\theta, r\sin\theta)$, where r is the step size, and θ is a random variable between 0 and 2π. Modify your DLA program or class `DLAApp` to simulate continuum DLA.

(b) Compare the appearance of a continuum DLA cluster with a DLA cluster generated on a square lattice. It is necessary to grow very large clusters (approximately 10^6 particles) to see the differences.

(c) Use the mass dimension to estimate the fractal dimension of continuum DLA clusters and compare its value with the value you found for the square lattice. ■

Project 13.17 More efficient simulation of DLA

To improve the efficiency of the algorithm, the walker in class `DLAApp` is restarted if it wanders too far from the existing cluster. When the walker is within the distance `startRadius` of the seed, no optimization is used. Because there can be many unoccupied sites within this distance, it is desirable to use an additional optimization technique (see Ball and Brady). The idea is to choose a simple geometrical object (a circle or square) centered at the walker such that none of the cluster is within the object. The walker moves in one step to a site on the boundary of the object. For a circle the walker can move with equal probability to any location on the circumference. For the square we need the probability of moving to various locations on the boundary. To find the largest object that does not contain a part of the DLA cluster, consider coarse grained lattices. For example, each 2×2 group of sites on the original lattice corresponds to one site on the coarser lattice; each 2×2 group of sites on the coarse lattice corresponds to a site on an even coarser lattice, etc. If a site is occupied, then any coarse grained site containing this site is also occupied.

(a) Because we have considered DLA clusters on a square lattice, we use squares centered at the walker. We first find the probability $p(\Delta x, \Delta y, s)$ that a walker, centered on a square of length $l = 2s + 1$, will be displaced by the $(\Delta x, \Delta y)$. This probability can be computed by simulating a random walk starting at the origin and ending at a boundary site of the square. Repeat this simulation for many walkers and then for various values of s. The fraction of walkers that reach the position $(\Delta x, \Delta y)$ is $p(\Delta x, \Delta y, s)$. Determine $p(\Delta x, \Delta y, s)$ for $s = 1$ to 16. Store your results in a file.

(b) We next determine the arrays such that for a given value of s and a uniform random number r, we can quickly find $(\Delta x, \Delta y)$. One way to do so is to create four arrays. The first array lists the probability determined from part (a) such that the values for $s = 1$ are listed first. Call this array `p`. For example, `p[1]` $= p(-1,-1,1)$, `p(2)` $=$ `p(1)` $+ p(-1,0,1)$, `p[3]` $=$ `p[2]` $+ p(-1,1,1)$, etc. The array `start` tells us where to start in the array `p` for each value of `s`. The arrays `dx(i)` and `dy(i)` give the values of Δx and Δy corresponding to `p[i]`. To see how these arrays are used, consider a walker located at (x, y) centered on a square of linear dimension $2s + 1$. Generate a random number r and find `i = start(s)`. If $r <$ `p[i]`, then the walker moves

to `(x + dx(i), y + dy(i))`. If not, increment `i` by unity and check again. Repeat until $r \le$ `p[i]`. Write a program to create these four arrays and store them in a file.

(c) Write a method to determine the maximum value of the parameter s such that a square of size $2s + 1$ centered at the position of the walker does not contain any part of the DLA cluster. Use coarse grained lattices to do this determination more efficiently. Modify class `DLA` to incorporate this method and the arrays defined in part (b). How much faster is your modified program than the original class `DLA` for clusters of 500 and 5000 particles?

(d) What is the largest cluster you can grow on your computer in a reasonable time? Does the cluster show any evidence for anisotropy? For example, does the cluster tend to extend further along the axes or along any other direction? ■

Project 13.18 Cluster-cluster aggregation

In DLA all the particles that stick to a cluster are the same size (the growth occurs by the addition of one particle at a time), and the cluster that is formed is motionless. In the following, we consider a cluster-cluster aggregation (CCA) model in which the clusters do a random walk as they aggregate.

Suppose we begin with a dilute collection of N particles. Each of these particles is initially a cluster of unit mass and does a random walk until two particles become nearest neighbors. They then stick together to form a cluster of two particles. This new cluster now moves as a single random walker with a smaller diffusion coefficient. As this process continues, the clusters become larger and fewer in number. For simplicity, we assume a square lattice with periodic boundary conditions. The CCA algorithm can be summarized as follows:

(i) Place N particles at random positions on the lattice. Do not allow a site to be occupied by more than one particle. Identify the ith particle with the ith cluster.

(ii) Check if any two clusters have particles that are nearest neighbors. If so, join these two clusters to form a single cluster.

(iii) Choose a cluster at random. Decide whether to move the cluster as discussed. If so, move it at random to one of the four possible directions. The details will be discussed in the following paragraphs.

(iv) Repeat steps (ii) and (iii) for the desired number of steps or until there is only a single cluster.

What rule should we use to decide whether to move a cluster? One possibility is to select a cluster at random and simply move it. This possibility corresponds to all clusters having the same diffusion coefficient, regardless of their mass. A more realistic rule is to assume that the diffusion coefficient of a cluster is inversely related to its mass s, for example, $D_s \propto s^{-x}$ with $x \neq 0$. A common assumption is $x = 1$. If we assume that D_s is inversely proportional to the linear dimension (radius) of the cluster, an assumption consistent with the Stokes–Einstein relation, then $x = 1/d$, where d is the spatial dimension. However, because the resultant clusters are fractals, we really should take $x = 1/D$, where D is the fractal dimension of the cluster.