Random Walk, Diffusion and Cluster Growth

Xinyu Wu, Peifan Liu, Connor Hann and Xiaomeng Jia[1]

[1]*Physics Department, Duke University*

(Dated: March 24, 2016)

Abstract

In this article,

1. 2D RANDOM WALK

A random walk is a mathematical formalization of a path that consists of a succession of random steps. For one dimensional case, a random walker can move one step in $\pm x$ directions with equal probability at each time step. If we make consider the statistical properties of an ensemble of random walkers, the expected value of the position $< x >$ is zero, since the expectation of each step is zero. However, the root-mean-squared distance(RMS) after n steps is:

$$\sqrt{< x_n^2 >} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} < \Delta x_i \Delta x_j >} = \sqrt{n}\Delta x \tag{1}$$

Here Δx is the step unit corresponding to a time unit Δt and a constant velocity v:

$$\Delta x = v \cdot \Delta t \tag{2}$$

Combined with

$$n = \frac{t}{\Delta t} \tag{3}$$

We can show that the motion is diffusive:

$$< x^2(t) >= 2Dt \tag{4}$$

where $D = v \cdot \Delta x/2 = (\Delta x)^2/(2\Delta t)$ is the diffusion constant.

If we generalize the deduction to the 2D case, where the random walkers can move one step in four diections $(\pm x, \pm y)$ with equal probability, we can find that the motion is again diffusive, by evaluating the mean square distance:

$$< r^2(t) >= 2Dt \tag{5}$$

Where $D = (\Delta x)^2/(4\Delta t)$.

This argument can be verified numerically using Python. We write a program to simulate a random walker in 2 dimensions, taking steps of unit length in $\pm x$ or $\pm y$ direction on a discrete square lattice. For up to $n = 100$, by averaging over 10^4 different walks for each $n > 3$, the expected value of the position $< x >$ is zero at all time, and the mean square value $< x^2 >$ has a linear relation with n(remember $n = t/\Delta t$ is proportional to t), as can be seen in Fig.1. A typical 2D random walking pattern is show in Fig.2. Since we choose $\Delta t = \Delta x = 1$ here, the diffusion constant is determined using the slope of $< r^2(t) >$ and n(see Fig.3):$D = 1/4$.

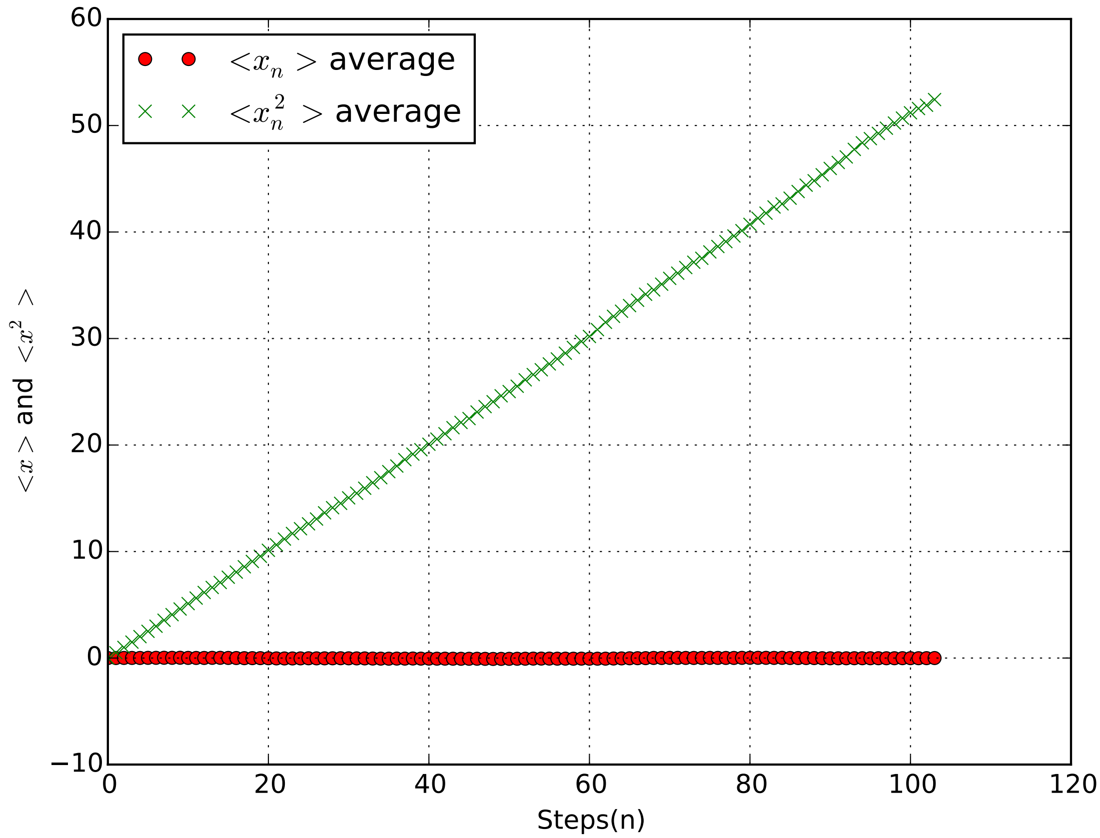

FIG. 1: $< x(t) >$ and $< x^2(t) >$ in 1D random walking, averaging over a 10000 random walker ensemble.

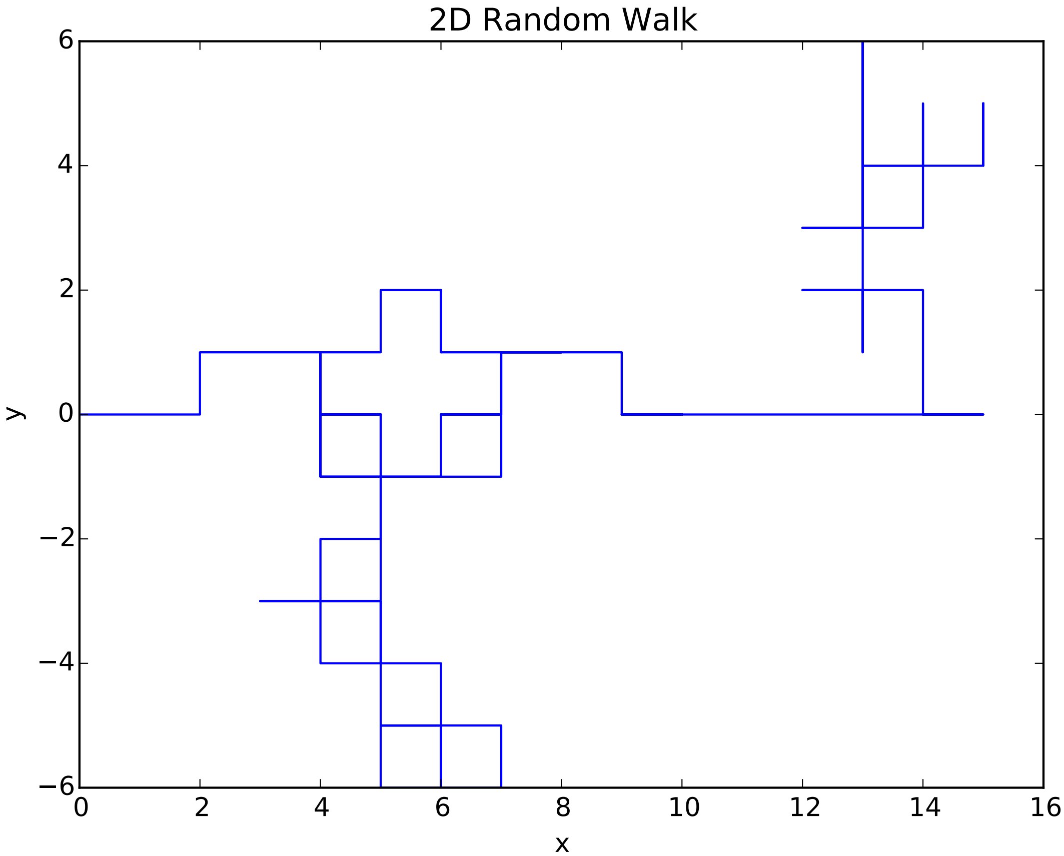

FIG. 2: A tipical 2D random walking pattern, starting from (0,0).

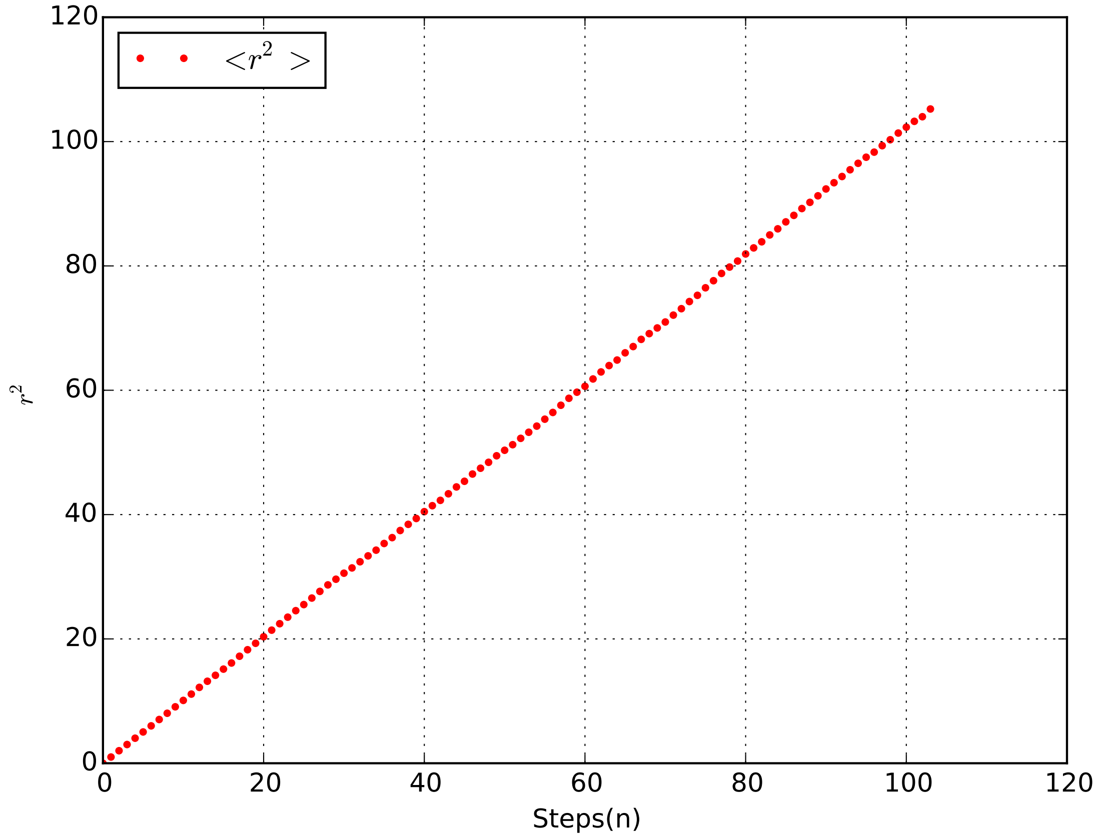

FIG. 3: $< r^2(t) >$ in 2D random walking, averaging over a 10000 random walker ensemble.

2. DIFFUSION EQUATION

In this problem we aim to simulate the diffusion of one substance diffusion in another(sugar in water for instance). Diffusion will start from a box profile with a peak at origin and 10 grid around and diffuse in 1 dimension. We will found out the diffusion will approximate Gaussian distribution at later times and fit will be preformed to verify that $\sigma = \sqrt{2Dt}$ for several snapshot t.

1. First of all we will show analytically that the spatial expectation $\langle x(t)^2 \rangle$ of the 1D normal distribution

$$\rho_{(x,t)} = \frac{1}{\sqrt{2\pi}\sigma(t)} \exp(-\frac{x^2}{2\sigma(t)^2}) \tag{6}$$

is just $\sigma(t)^2$.

The spatial expectation

$$
\begin{aligned}
\langle x(t)^2\rangle &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma(t)} x^2 \exp(-\frac{x^2}{2\sigma(t)^2}) dx \\
&= \frac{1}{\sqrt{2\pi}\sigma(t)} \int_{-\infty}^{\infty} x^2 \exp(-\frac{x^2}{2\sigma(t)^2}) dx \\
&= \frac{1}{\sqrt{2\pi}\sigma(t)} \times \sqrt{2\pi}\sigma(t)^3 \\
&= \sigma(t)^2.
\end{aligned}
\tag{7}
$$

So $\langle x(t)^2\rangle = \sigma(t)^2$.

2. We know the diffusion equation

$$\frac{\partial \rho}{\partial t} = D\nabla^2 \rho \tag{8}$$

If we discretize time and position to $t = k\Delta t, x = i\Delta x$, we can get the recursion equation with time,

$$\rho_{i,k+1} = \rho_{i,k} + D\frac{\Delta t}{\Delta x^2}(\rho_{i+1,k} + \rho_{i-1,k} - 2\rho_{i,k}) \tag{9}$$

here we used $\Delta t = 0.002s, \Delta x = 0.1m, D = 2$. The relation $\Delta t = \frac{\Delta x^2}{2D}$ is satisfied to guarantee convergence.

Then, with the initial distribution given, all future state can be solved. We here will solve the equation for 150s and take snapshot at 30s, 60s, 90s, 120s and 150s. Gaussian fit is applied to these snapshot to verify $\sigma = \sqrt{2Dt}$. Plots for the 5 snapshots and fitted Gaussian are shown in Fig. 4.

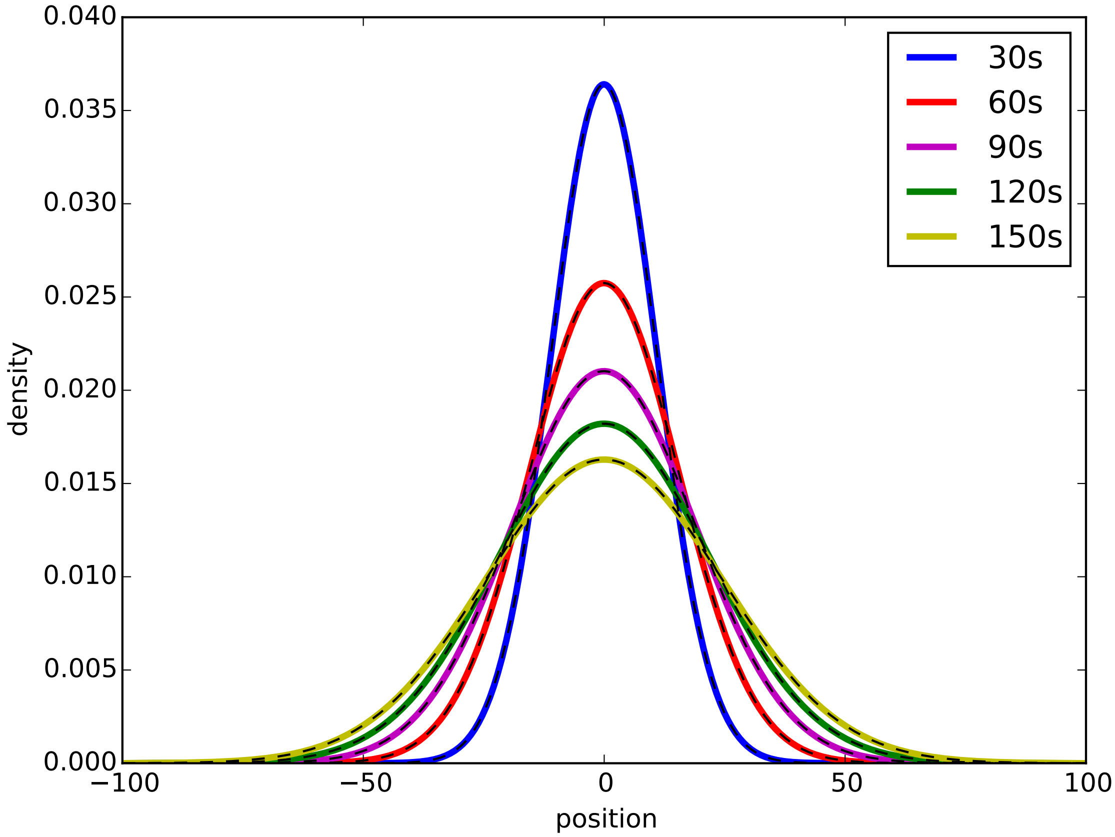

FIG. 4: Snapshots at 30s, 60s, 90s, 120s, 150s and comparison with fitted curve

As for the Gaussian fit, we used the function curve_fit from scipy.optimize. We first define the Gaussian function with input of x, μ, σ and call it in the function _fit to get the optimized σ the comparison of σ with $\sqrt{2Dt}$ are shown in Fig. 5.

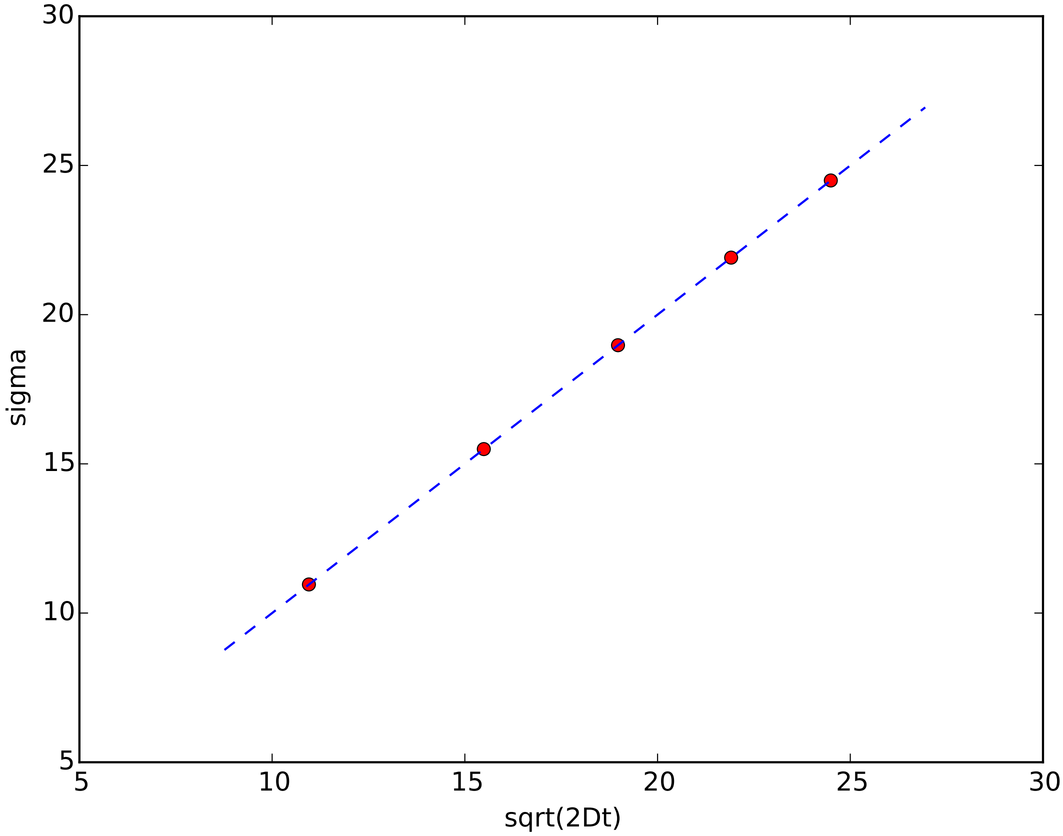

FIG. 5: Comparison of σ and $\sqrt{2Dt}$, the blue dashed line is $y = x$.

Discussion

In this problem, we used the diffusion equation to simulate the diffusion of a initial distribution of a peak(spread out a few grid) and we find that the distribution tend to approximate Gaussian distribution. We also find that the spatial spread expectation $\langle x(t)^2 \rangle = 2Dt$ increase linearly with time.

3. CLUSTER GROWTH WITH THE DLA MODEL

Diffusion-limited aggregation (DLA) is the process whereby particles undergoing a random walk due to Brownian motion cluster together to form aggregates of such particles. This theory, proposed by T.A. Witten Jr. and L.M. Sander in 1981, is applicable to aggregation in any system where diffusion is the primary means of transport in the system. DLA can be observed in many systems such as electrodeposition, Hele-Shaw flow, mineral deposits, and dielectric breakdown.

The implementation of DLA using Python involves the following steps:

1. consider a lattice of points;
2. start with a seed particle at origin;
3. release a particle at a randomly chosen location some distance(here we choose 100 units) from the cluster, let it perform a random walk;
4. judge: if the particle stray too far away (here we set 200) from the original, it is thrown away; if the particle reach a perimeter site(i.e., all the unoccupied sites that are nearest neighbor to occupied sites), it becomes part of cluster;
5. repeat 3-4 multiple times(here we set 10,000 particles).

Typical clusters grown using DLA method are very sparse, i.e., they contain many open spaces and have very irregular perimeters, as can be seen in Fig. 6 and 7.

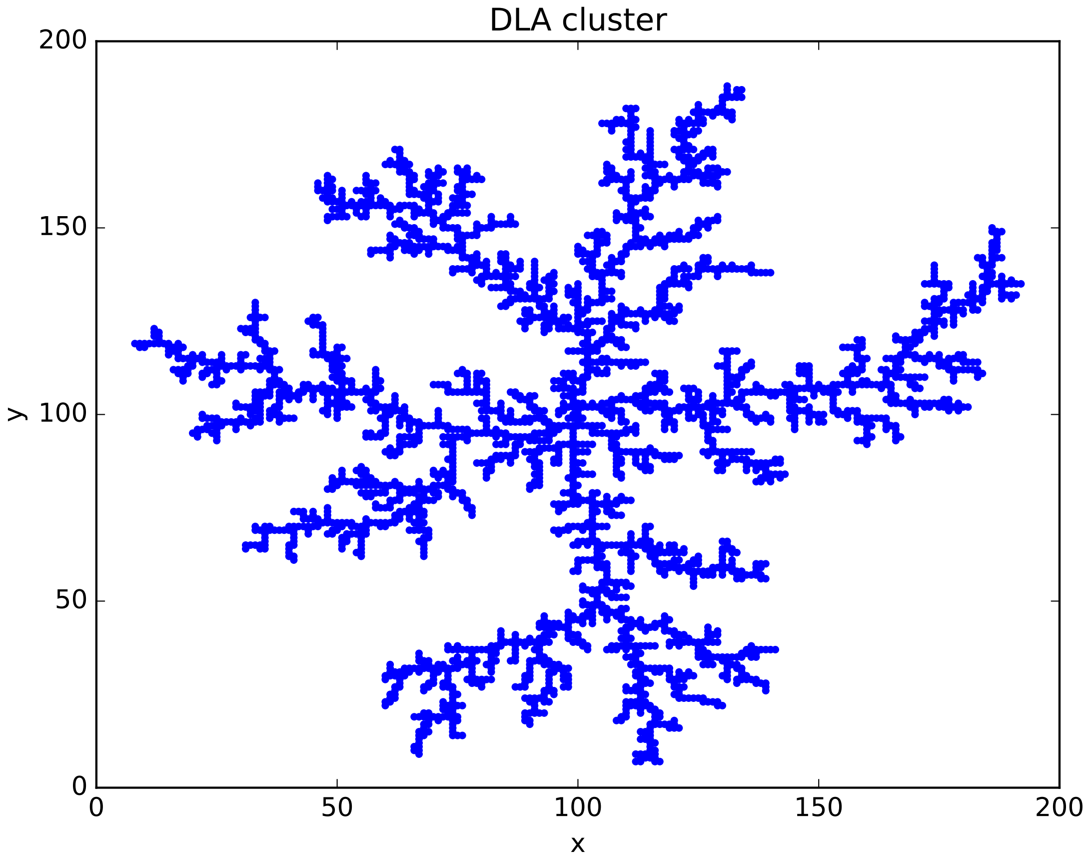

FIG. 6: Cluster grown using DLA method.

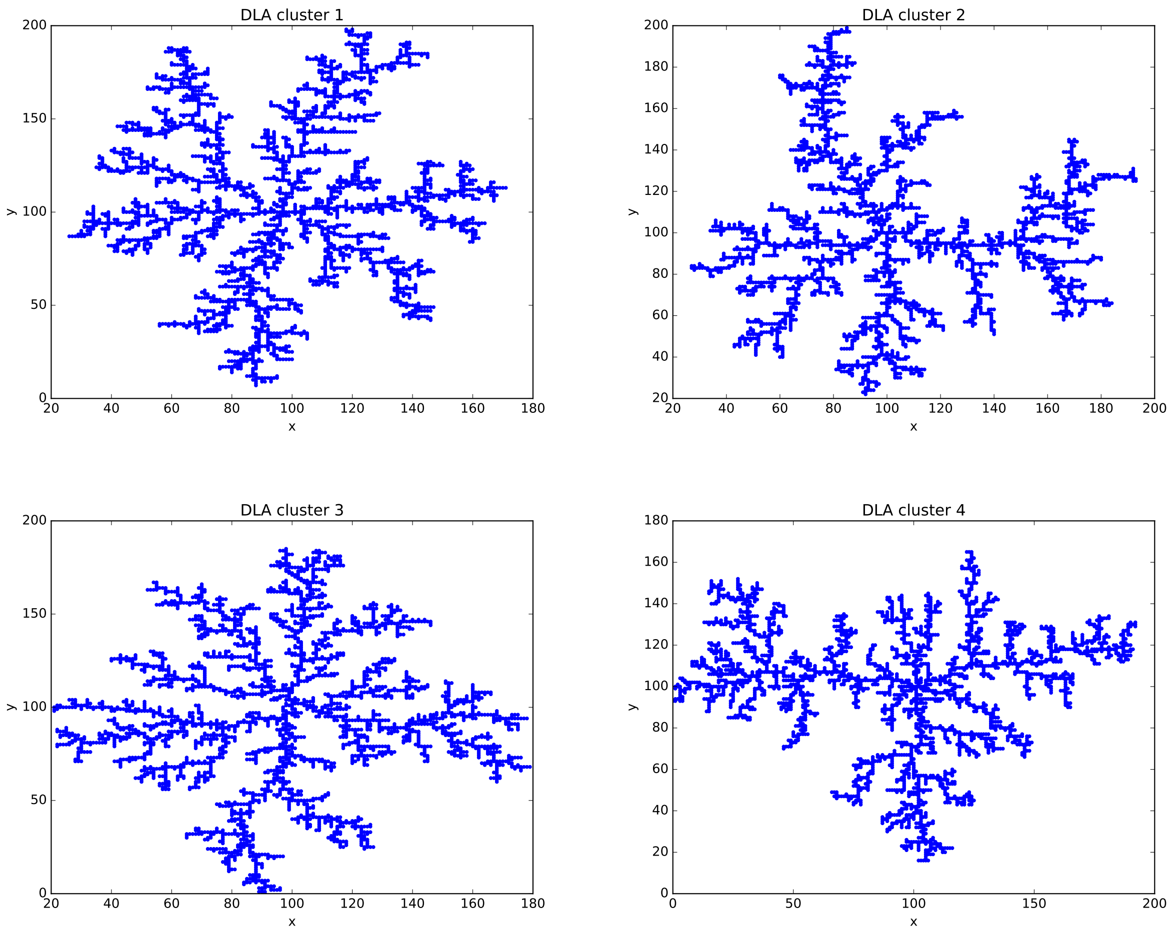

FIG. 7: Other typical clusters grown using the DLA method

Next we want to extract the fractal dimension of the cluster by plotting the "mass" (how many points a cluster contains) as a function of its radius. The fractal dimensionality d_f is defined as:

$$m(r) \sim r^{d_f} \tag{10}$$

In the 2D case, the largest dimensionality comes from the most compact geometry—a uniform disk: $d_f(disk) = 2$. A line, by comparison, has $d_f(line) = 1$. The d_f for DLA cluster will fall in between the two. By counting the points inside various radii r, then determining the slope of $\log(r)$ vs. $\log(m)$ graph, one can calculate the value of d_f. This procedure is illustrated in Fig. 8. In order to obtain a more accurate d_f value, we calculated d_f for 10 distinct DLA clusters and averaged the results. These results are shown in Table I. From these results we calculate $d_f = 1.68 \pm 0.7$, which is in excellent agreement with the literature value of $d_f = 1.65$.

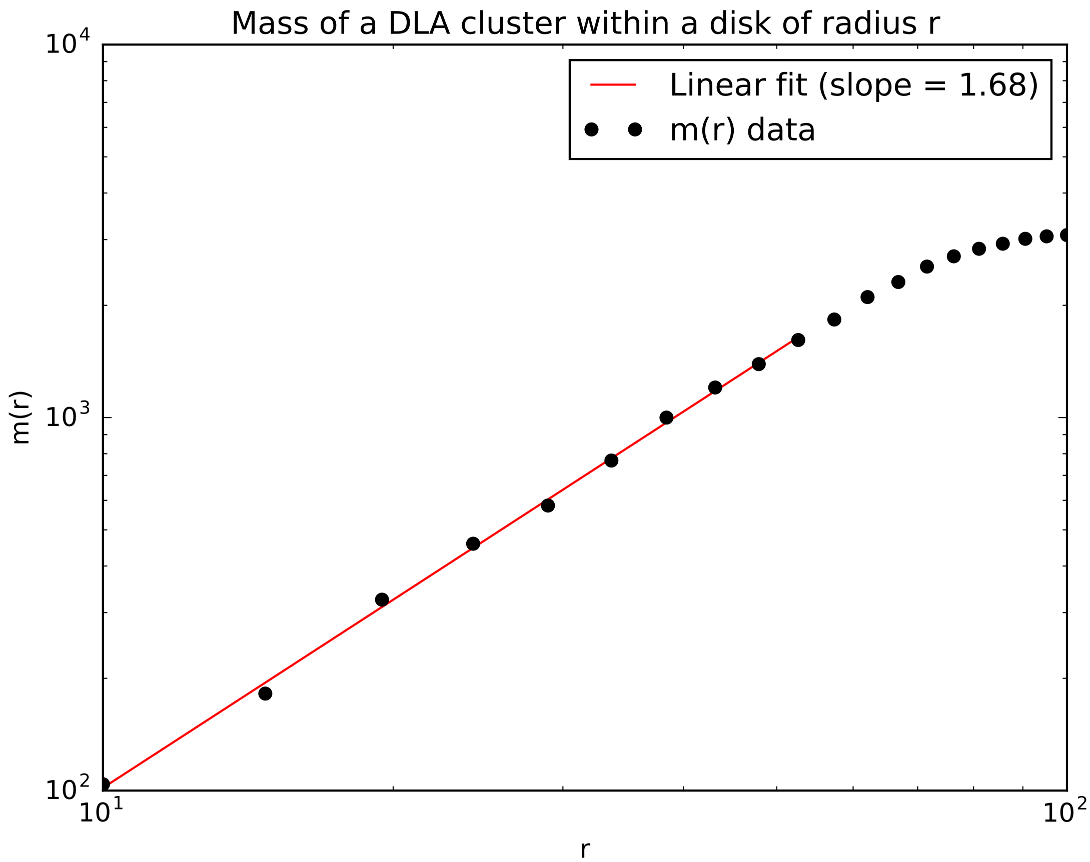

FIG. 8: Extraction of d_f from the relation of the cluster "mass" and radius.

TABLE I: Computed d_f values for 10 different DLA clusters

Cluster	1	2	3	4	5	6	7	8	9	10
d_f	1.74	1.63	1.77	1.55	1.66	1.58	1.64	1.74	1.74	1.75