Investigating Dimensional Properties of 2D Fractals from Diffusion-Limited Aggregation

24136
Department of Physics, University of Bath, Bath BA2 7AY, United Kingdom
Date submitted: 23 Mar 2022

Abstract

Using a simulation of square particle, diffusion-limited aggregation, the avoidance of using length parameters in calculating the fractal dimension of simulated DLA clusters was found to give a more precise result. The fractal dimension was 1.72 ±0.03, which was independent of how many particles are in the cluster greater than 500. If the probability of forming a bond between cluster and incident particle is not 1, the fractal dimension of the cluster will exhibit change proportional to the probability at the power of $(5.0 \pm 0.1)x10^{-2}$.

Introduction

All objects have a dimension, d, such as uniform density, 2-dimensional shapes have a d of 2 and 3-dimensional shapes a d of 3 [1]. However, not all structures have an integer d, as 2-dimensional fractals have a d_f, fractal dimension, between 1 and 2 [2].

One process that creates fractals, is diffusion-limited aggregation. DLA occurs when randomly moving particles, due to Brownian motion, collide with each other to form fractal clusters [3].

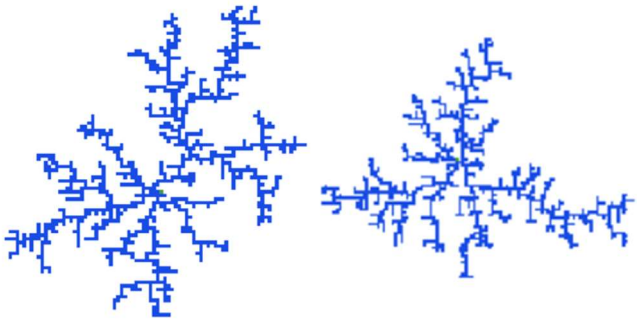

Figure 1. Picture of 2 particle cluster generated through DLA. Labelled, from left to right, cluster 1 and 2.

Due to the random nature of the creation process, accurately measuring the sizes of clusters analytically can be very difficult. However, by using a cluster's fractal dimension, d_f, its radius can be estimated from

$$N_c = (R/a)^{d_f}, \quad (1)$$

where N_c is the number of particles in the cluster, R is the radius of the cluster, and a is a length parameter. The estimation can be refined into a more general expression

$$N_c = (\alpha R)^{d_f} + \beta, \quad (2)$$

where $\alpha$and β are constants. Taking the logarithm of equation (2) and differentiating, it turns into

$$\frac{d(\ln N_c)}{d(\ln R)} = \frac{d_f}{1+\beta/(\alpha R)^{d_f}}. \quad (3)$$

When R is significantly large enough, the right-hand side of equation (3) reduces to be just d_f

DLA as a process appears commonly in nature, such as the growth of viscous finger in Hele-Shaw cells [4]. In physical systems, particles may have a greater energy than usual, creating an elastic collision when colliding with the cluster instead of forming a bond to it. To account for the distribution of possible energies incident particles will have, a sticking probability p can be added to simulations of DLA.

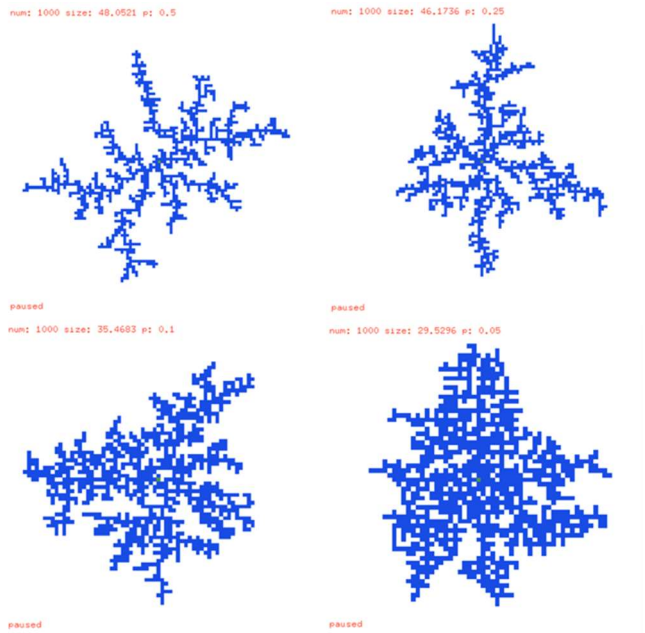

Figure 2. Pictures of particle clusters generated through DLA with a sticking probability, p. From top left to bottom right in reading order, p =0.5, 0.25, 0.1, 0.05.

Method

The simulation was coded in C++, while the data produced was processed analytically using python. Starting with a central point on a grid acting as the origin, the simulation creates a new square particle at a random point along a circle of a set creation radius from the origin. Then, a random number generator determines the particle's movement, with each move being either: up, down, left, or right. This continues until it arrivals at a point adjacent to a previously created particle, wherein its position is added to the grid of the previous particles created. Then, the process repeats itself in a new iteration as a new particle is created with the with the same prerequisites. As N_c increases, the creation radius increases to stay at 1.2 times the size of R.

At certain intervals of N_c, it and R are recorded on a .csv file. To speed up the simulation, if a particle moves far enough away from the origin, 1.7 times the distance as the creation radius, it is assumed lost and is deleted with a new particle being created with the starting conditions [5]. This removed the time waiting for a stray particle to return close to the cluster.

Once the data for many clusters was simulated, it was processed through a python code that calculated: the mean and standard deviation of R for all clusters simulated at each N_c, the logarithm of both R and N_c, and the gradient of logarithms using regression analysis.

For the simulation to be more applicable for physical systems, the code was modified to include p. When a particle ends its move adjacent to another particle, a random fraction is generated. If the fraction is less than or equal to the p, then the particle will stop and follow the same algorithm as before. If the fraction is greater than p, the particle will move next iteration as if it was not adjacent to the neighbour particle. There is fail-safe for if the particle moves into an already occupied space, as the simulation will check to see if the new position is empty before changing the particle's position.

The data from clusters of varying p are collected and are processed through a similar python code as before, calculating the same properties and the logarithm of p and d_f.

Results

The simulation was repeated over 120 times, creating that many individual 1000 particle clusters. The logarithm of N_c was and plotted against the logarithm of the R for the two clusters in figure 1, as well as for the calculated mean cluster.

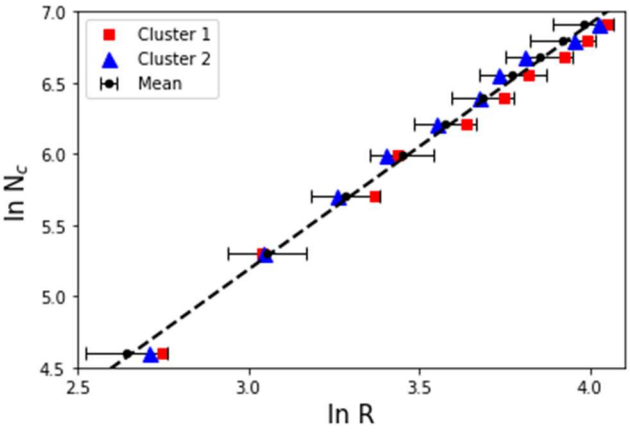

Figure 3. A graph of $ln\, N_c$ against $ln\, R$, with the red squares and blue triangles as the data from clusters 1 and 2 from figure 1 respectively, the black dots as the mean results, the solid lines as the standard deviation of the mean results, and the dotted line as the line of best fit.

Using the data and the line of best fit from figure 3, d_f can be calculated from both equation (1) and equation (3) for the two clusters and the mean cluster, assuming $a = 1$.

Table I: R and d_f , from equation (1) and (3) of the 2 simulated 1000 particle DLA clusters from figure 1 and the mean DLA cluster of 120 runs.

Cluster	R	d_f (equ 1)	d_f (equ 3)
1	57.25	1.707	1.758
2	56.08	1.715	1.769
Mean	53.72	1.73 ±0.15	1.73 ±0.04

Increasing N_c to 3000 allowed d_f to be plotted against N_c.

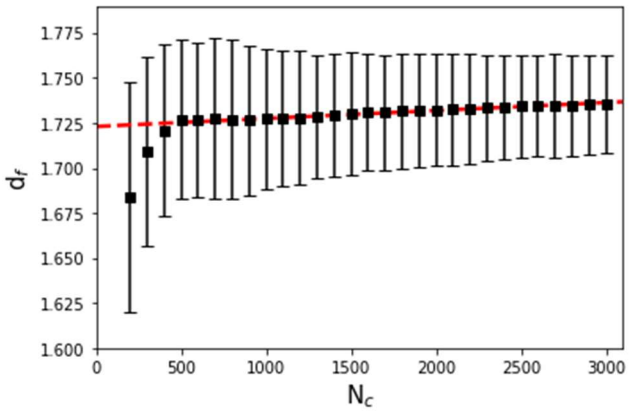

Figure 4. A graph of d_f against N_c for the mean cluster, with the black squares as results, the black solid lines as the error, and the red dotted line as the result's line of best fit.

For N_c greater than 500 particles, d_f has a line of best fit with equation

$$d_f = (4.4 \pm 1.4)x10^{-6}\, N_c + (1.72 \pm 0.03). \qquad (4)$$

When the simulation had p implemented, 50 clusters were made for each p, varying from 1 to 0.05. Using equation (3), d_f was calculated for each p and plotted against each other.

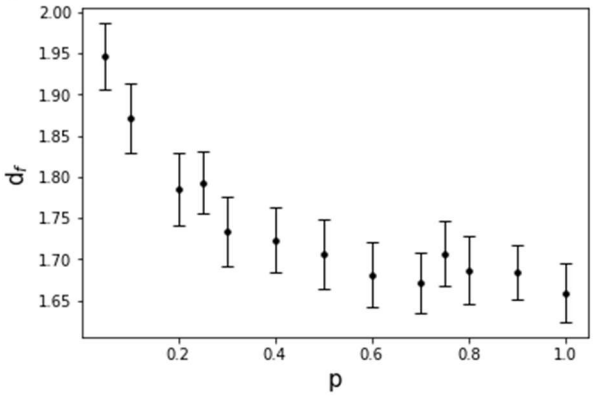

Figure 5. A graph of d_f against p, with the black dots as the points and the black lines as error.

The logarithms of d_f and p were also plotted against each other.

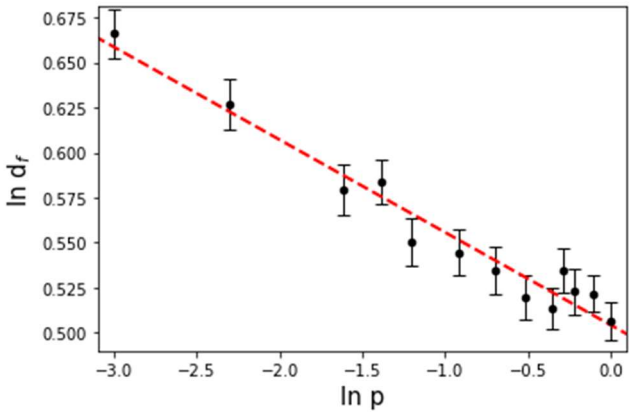

Figure 6. A graph of $ln\ d_f$ against $ln\ p$, with the black dots as the points, the black solid lines as the error and the red dotted line as the line of best fit.

The equation for the line of best fit on figure 6

$$ln\, d_f = -(5.1 \pm 0.1)x10^{-2}\, ln\, p + (0.54 \pm 0.03). \qquad (5)$$

Which can be rearranged into

$$d_f = (1.72 \pm 0.05)p^{-(5.1\pm0.1)x10^{-2}}. \qquad (6)$$

Discussion

Table I shows that the calculated mean cluster gave consistent d_f results to the two simulated clusters, for both equation (1) and (3). Plus, it also showed that the two equations gave the same d_f, with equation (1) having a larger standard deviation. This is due to the equation (1) having a dependence of the length parameter a, which is unknown and unique for each cluster. So, any d_f calculated has a high error associated with it. Whereas equation (3) can be reduced when R is significantly large enough, removing the need to include the unknown constants $\alpha$and β. Conversely, the amount of data and the computational time needed to use equation (1) is much less, it only requires the cluster's information at the end of its creation. While equation (3) needs information at regular intervals during the cluster's formation, then the data must be processed to find the gradient. Nonetheless, equation (3) is favoured over equation (1), as its benefits outweigh its drawbacks.

In figure 4, we see the effect of N_c has on the mean cluster's d_f. Up to 500 particles, d_f increases with N_c. However, for clusters larger than 500 particles, d_f levels off to have equation (4) as a line of best fit. Its gradient is very small compared to the error of its constant term, so it can be approximated to be zero, effectively making d_f independent of N_c and to be equal to 1.72 ±0.03.

Figure 5 shows an inverse power relationship between p and d_f, when the simulation was edited to better

simulate physical systems. For sticking probabilities greater than 0.3, d_f changed little. This is evident in the clusters of figure 1, $p = 1$, and the $p = 0.5$ cluster in figure 2 having indistinguishable differences between them. However, d_f increases significantly when p is less than 0.3. The other clusters in figure 2 become denser with particles, forming structures that begin to look more like uniform density, 2-dimensional shapes as p reduces in size. The relation is apparent in figure 6, the logarithms of d_f and p give a straight line of best fit, giving equation (5) as the linear expression.

The reason for this occurrence is due the increase in likelihood of particles reaching the centre of the cluster without being caught by one of the cluster's arms. For unmodified DLA simulation, clusters form long arms of particles. These effectively shield the centre of the cluster from growth, as incident particles are more likely to be caught by the arms than make it to the centre, directing the growth of the cluster to the outer edges. This is still the case for the modified simulation, the path of a particle will hit one of the clusters arms multiple times before reaching the centre. So, sticking probabilities higher than 0.3 will still not be low enough for the particles to make it past the arms without being caught. If p is low enough, particles can reliably reach the centre of the cluster, reducing R and therefore increase d_f.

The major limitation of the simulation used throughout, is how the random numbers are generated, both for the particle movement and sticking to the cluster. It is powered by a Linear Congruential Generator, LCG, which is an algorithm that generates a periodical sequence of numbers [6]. Even though the periods can be tremendously large, of order of 2^{32}, it is not completely random. So, using this pseudorandom method will always have error when using it to simulate random physical systems. The only way to reduce this limitation is to use as long a sequence of numbers as possible, as to avoid the periodic nature of a LCG from occurring in simulations.

Another limitation is the shape of the particles being simulated and the movement directions. Square particles can only be adjacent to other particles at 4 angles, as the particle can only move in 4 directions. This limits the number of possible interactions particles can experience, as it could allow particles to keep moving when in reality they would have collided with the cluster. To improve this aspect, increasing the number of movement directions or making the particle's shape more complex will reduce the number of missed collisions, as there are more angles for particles to be considered adjacent.

Conclusion

Overall, avoiding the use of length parameters in calculating the fractal dimension of simulated DLA clusters gave more precise results. For the square particle DLA fractals simulated, fractal dimension was calculated to be is 1.72 ±0.03, which was independent of how many particles are in the cluster greater than 500. Modifying the simulation to include a non-zero probability of forming a bond between cluster and incident particle, showed the fractal dimension of the cluster will exhibit change proportional to the probability at a power of $(5.0 \pm 0.1)x10^{-2}$.

References:

[1] Duda R. The Origins of the Concept of Dimension. *Colloquium Mathematicum*. 1979; 42(1): pp. 95-110.

[2] Landau RH, Páez MJ, Bordeianu CC. *Computational physics: problem solving with Python*. 3rd ed. Weinheim: Wiley-VCH; 2015.

[3] Halsey T. Diffusion-Limited Aggregation: A Model for Pattern Formation. *Physics Today*. 2000; 53(11): pp. 36-41.

[4] Nittmann J, Daccord G, Stanley HE. Fractal growth viscous fingers: quantitative characterization of a fluid instability phenomenon. *Nature*. 1985; 314(6007): pp. 141-144.

[5] Witten TA, Sander LM. Diffusion-limited aggregation. *Physical Review B*. 1983; 27(9): pp. 5686-5697.

[6] Knuth DE. *The art of computer programming – volume 2 seminumerical algorithms*. 3rd ed. Reading, Mass.: Addison-Wesley; 1998.