DIFFUSION LIMITED AGGREGATION

PHY612N/412: COMPUTATIONAL PHYSICS

INSTRUCTOR: DR. SUNIL PRATAP SINGH

INDIAN INSTITUTE OF SCIENCE EDUCATION AND RESEARCH BHOPAL

23TH April 2024

SAJAN DAHERIYA, 20239
HRIDHAY AS, 20320

Abstract

In this report, we present a study of the Diffusion-limited aggregation (DLA) process. There we explain some parameters of DLA like dimension of fractal structures, density function, total number function. We are simulating the DLA using two methods

1) Random walk method:

In random walk method based on the classic algorithm proposed by Witten and Sender, and we extend it to consider various scenarios, including general cases, density function and fractal dimension analysis. Moreover, we investigate the effects of different parameters on the DLA process, such as introducing Psnn, varying probabilities on direction, real boundary, 3D DLA, and different seed configurations. To enhance the visualization of our simulation results, we create animations and produce beautiful DLA images

2) Langevin force method.

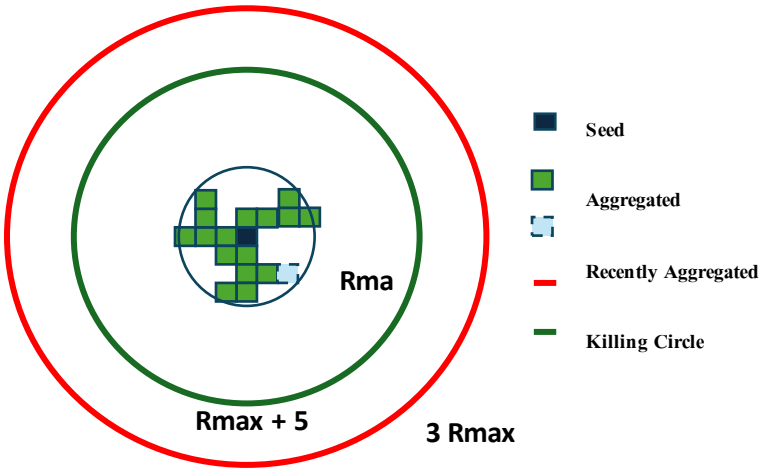
In Langevin force method, we solve Langevin equation with a drag force term to simulate Brownian motion of particles which is more related to real world as it has dependence to T (temperature), eta (Viscosity), R (Radius of the particle) etc. We have used various computational tricks to improve simulation efficiency and reduce computational time. Furthermore, we investigate Dimensionality of fractal like aggregate we obtain from the simulation, density function etc. In this project we have developed a reusable python code, purely done in OOP Python. At last, but not least, we have made some cool animations and aggregate visualisation gifs.

Motivation

Simulating Diffusion Limited Aggregation (DLA) offers insights into pattern formation, growth processes, and fractal geometry. By understanding how complex structures emerge from simple rules, DLA simulations aid in modelling aggregation phenomena observed in materials science and biology. Additionally, studying DLA provides opportunities to explore fractal properties and their implications across scientific disciplines. Simulating Diffusion Limited Aggregation (DLA) not only provides insights into pattern formation, growth processes, and fractal geometry but also addresses real-world phenomena and problems. By uncovering the principles governing DLA, we can optimize materials synthesis processes like electrodeposition, crystal growth and Snowflake Formation, advancing materials science. Additionally, DLA simulations offer insights into biological phenomena such as tumour growth, Wound healing, with potential applications in healthcare and biotechnology. Connecting DLA simulations with real-world challenges enables innovation and practical solutions across diverse domains.

Random Walk Method

The Witter and Sander method uses Monte Carlo techniques on a 2D lattice. Particles move randomly, with a seed at the center and new particles introduced at a distance. They stick to the seed based on probabilities. Nearest neighbour adherence is considered. Particles start at Rmax + 5 distance, with those straying more than 3Rmax away being terminated. Nearest neighbor examination starts when a particle is Rmax + 2 units away. A density function C(r) analyses growth patterns using fractal structures. This density function is then normalized using the total number of particles and the corresponding volume between two spheres of radius $(r - \delta r)$ and $(r + \delta r)$, respectively. To determine the relationship between $\ln(C(r))$ and r, we calculate the density function exponent α using the power law dependency $C(r) \sim r^{\alpha}$. The fractal dimension D can then be obtained using the relationship $D = d - \alpha$, where d is the Euclidean dimension. To calculate α, we first determine the number of particles within different r values, denoted as $N(r)$. We assume that $N(r) = Nr^{a}$ and $C(r) = Cr^{-\alpha}$, and then take the logarithm of both sides to obtain



$$\ln(N(r)) = a\ln(r) + \ln N \quad \text{and} \quad \ln(C(r)) = (-\alpha)\ln(r) + \ln C$$

A schematic depiction of the system is shown in the following figure 1. Different values for the radius of the start-off circle and the distance at which the particle is killed were used, and the corresponding results were the same within a statistical error. The relationship between $N(r)$ and $C(r)$ can be written as

$$N(r) = \int (C(r) \times 2\pi r) dr$$

allowing us to solve for the parameter relationship

$$-\alpha = a - 2 \quad \text{and} \quad \ln C = \ln\left(\frac{a}{2\pi}\right) + \ln N$$

Using a valid range of r values, we fit the logarithm of $N(r)$ and then obtain $\ln(C(r))$. In addition, the fractal dimension $D = d - \alpha$ is just 'α' we calculated before.

Simulation results:

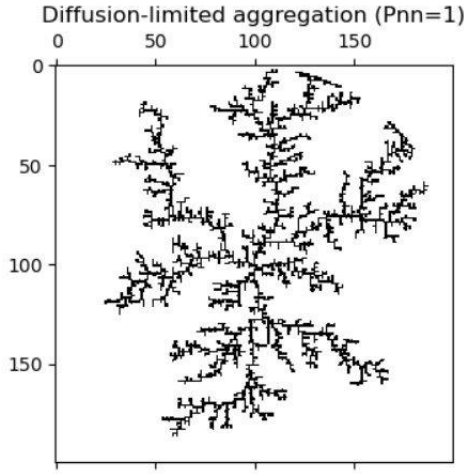


fig.2

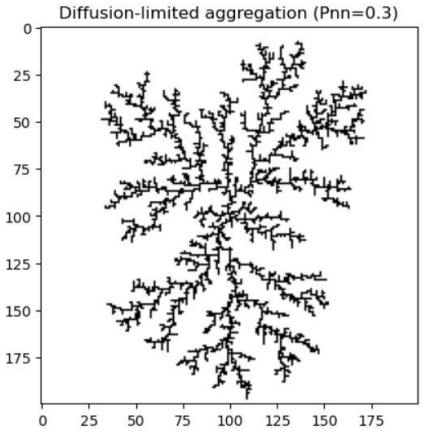


fig.3

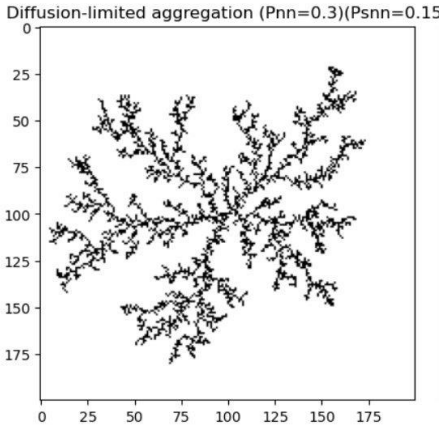


fig.4

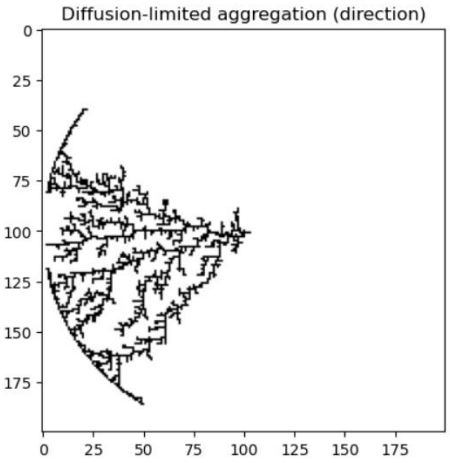


Fig.5

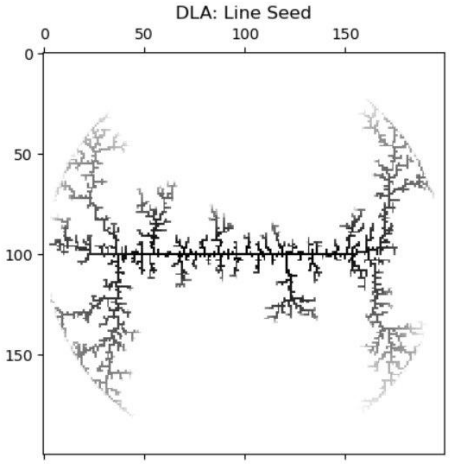
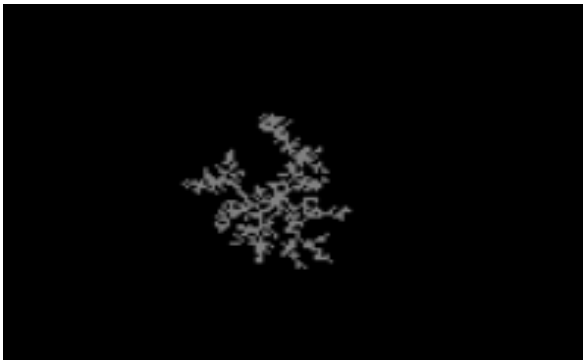


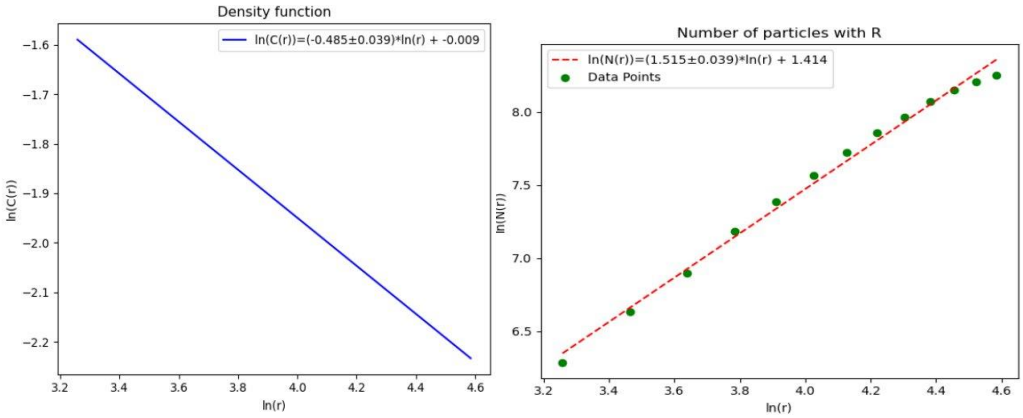
fig.6

Animation of DLA: 2D animation using Py game library in python



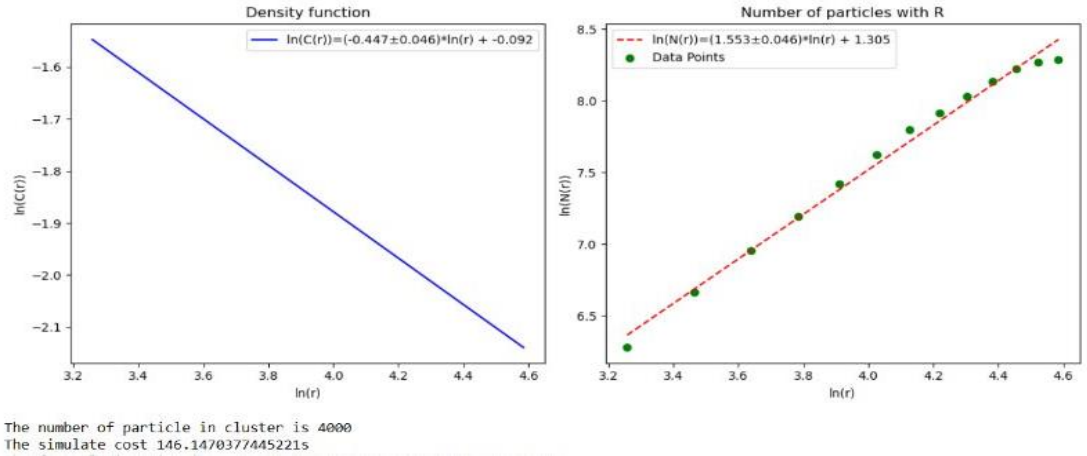
1) Pnn=1(probability for nearest neighbouring sites)

Fig. 2 illustrates a two-dimensional cluster model with N=4000 particles using the algorithm described. The left side shows Density function C(r), while the right side plots N particles (green) and a linear fit function (red). Fractal dimension is calculate d. 200



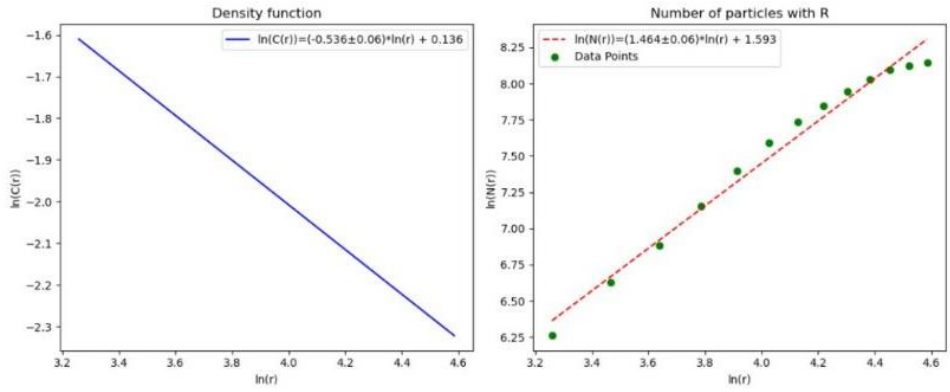
Fractal dimension $=1.515 \pm 0.039$

2)Pnn=0.3: Fig. 3 repeats the simulation with N=4000 particles using the same algorithm but with a reduced Pnn of 0.3. This leads to increased cluster detachment and denser particle aggregation compared to Pnn=1. Fractal dimension = 1.553 ± 0.046



Introduce Psnn(probability for secondary nearest neighbouring sites):

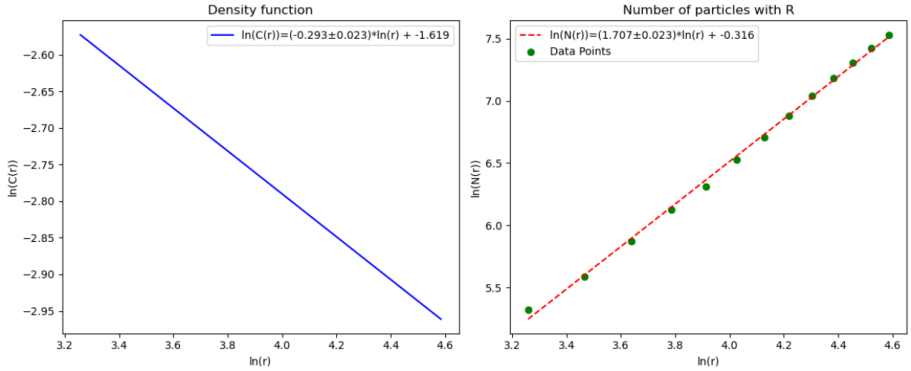
3) Psnn = 0.15, while Pnn=0.3. Fig. 4 shows a DLA simulation with grid size=200, N=4000, Pnn = 0.3, and Psnn = 0.15. The growth pattern is more branched, denser, and contrasts with previous cases. The left side displays Density function C(r), and the right-side plots N particles (green) with a linear fit function (red). Fractal dimension = 1.564 ± 0.06




```
The number of particle in cluster is 4000
The simulate cost 96.15806245803833s
```

4)Biasing Pnn in different directions:

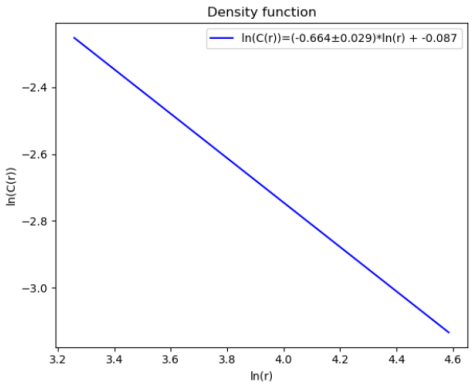
Introducing directional bias in DLA, with Pleft=0.1 and Pright=Pup=Pdown=0.3 for N=4000, leads to particles aggregating more on the left side of the cluster due to higher sticking probability from the right side. This results in a leftward bias in the overall aggregation pattern, depicted in Fig. 5, here Fractal dimension = 1.707 ± 0.023

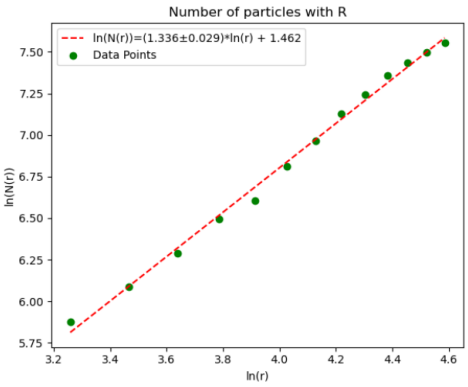



```
The number of particle in cluster is 4000
The simulate cost 69.58704566955566s
```

5)Seed line

Line seed arrangements on the lattice grid result in a growth pattern lacking circular symmetry, making density function and fractal dimension calculation difficult. The simulation's initialization method may constrain middle cluster growth due to particle s on both sides of the line seed configuration, leading to an elongated structure with narrow width, as shown in Fig. 6.



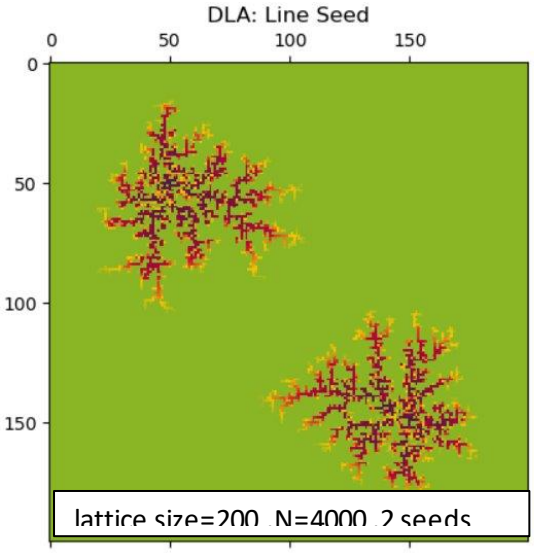


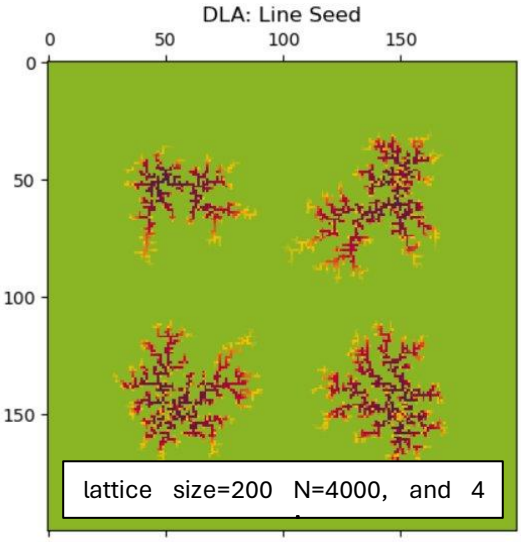

```
The number of particle in cluster is 4000
The simulate cost 63.504164934158325s
```

Fractal dimension $=1.336 \pm 0.029$

Multiple seeds:

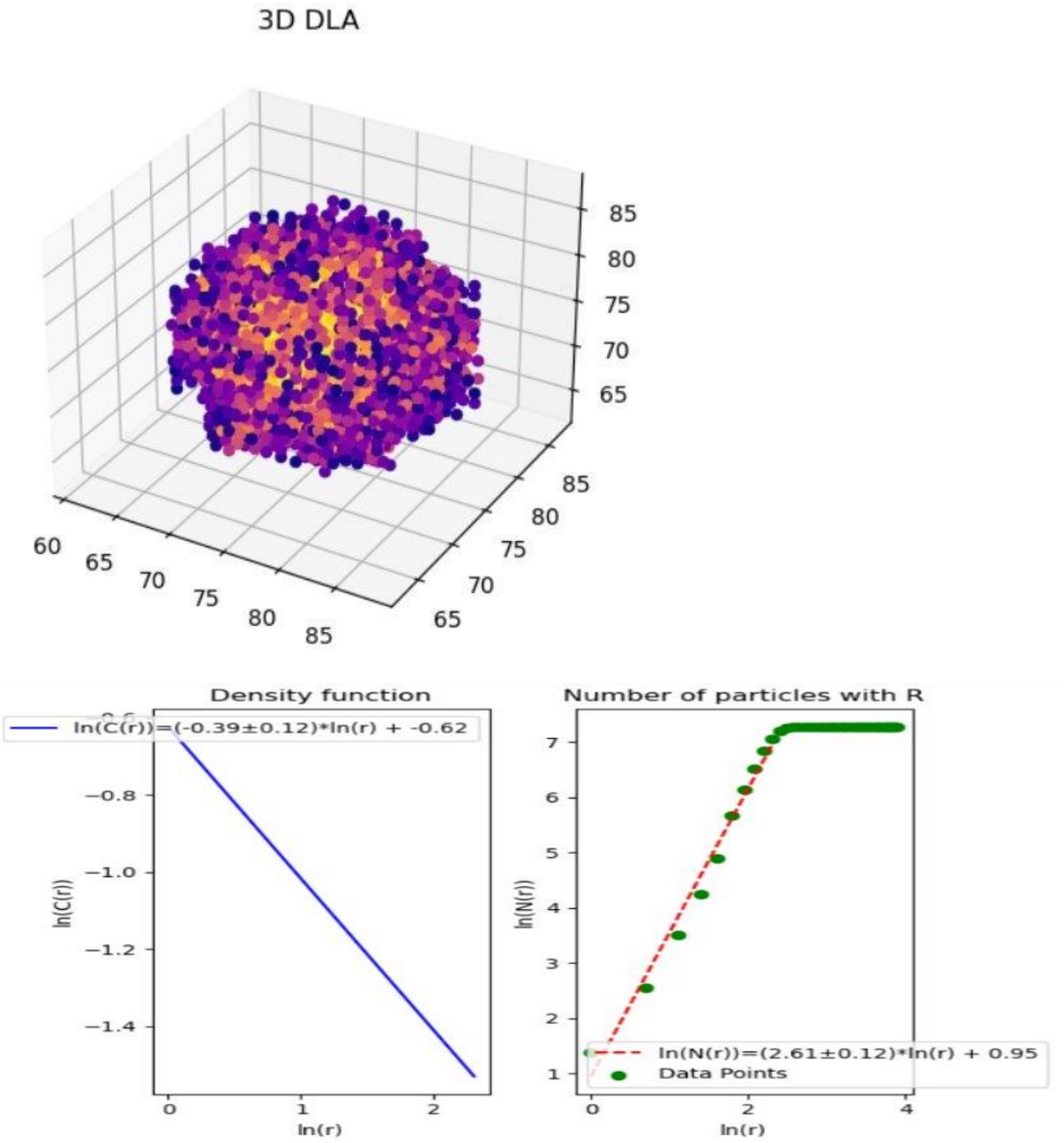
Due to the absence of circular symmetry, calculating density and total particle functions is not meaningful. Instead, we prop ose a new approach: randomly selecting a radius R from two seed configuration centres to generate particles during simulation. For enhanced visualization, we can colour the resulting DLA structure based on the order of particle addition.





Three-dimensional DLA:

In three-dimensional DLA, we downsize the lattice to 150 (150×150×150) to reduce computation times and simulate N=5000 particles per run. Color-coding based on particle aggregation time visualizes the growth pattern, echoing the 2D case. Analyzing density function and fractal dimension in 3D DLA requires adjusting equations used in 2D scenarios. Specifically $N(r) = \int (C(r) \times 4\pi r^2)dr$ Thus, the parameter relationship is modified $-\alpha = a - 3$ $-\alpha = a - 3$ and $\ln C = \ln\left(\frac{a}{4\pi}\right) + \ln(N)$



Size of lattice 150, N=5000, and simulation 10 clusters: Left shows the Density function C(r). Right shows Number of particles with N(green points) and linear fit function of N(r) (red line), Fractal dimension=2.61 ± 0.12

Langevin Force Method

BAOAB Scheme for solving Langevin Equation:

$$m\frac{dv}{dx} = -\gamma v + R(t) + F_{ext}$$

This is the famous Langevin equation, on right side; first term is the drag force term on particle, second term is for random force and the third term is for other deterministic external forces such has electric force. To solve this equation there are vari ous approaches but we have found out that BAOAB is the best amongst them-

In BAOAB Scheme there are steps:

B $v(t) \leftarrow v(t) + \frac{F(t)}{m}\left(\frac{dt}{2}\right)$

A $x(t) \leftarrow x(t) + v(t)\left(\frac{dt}{2}\right)$

O $v(t) \leftarrow e^{-\gamma dt}v(t) + R(t)\sqrt{\frac{K_b T}{m}}\sqrt{1 - e^{-2\gamma dt}}$

Where, $\gamma = \frac{6\pi\eta R_p}{m}$

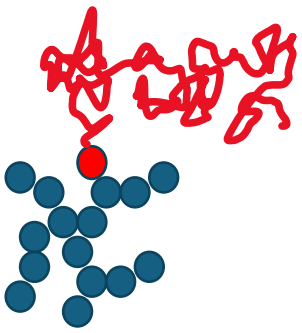
$R(t)$: Gaussian random number with mean zero and standard-deviation 1.

K_b : Boltzmann Constant, T : Temperature, η : viscosity, R_p : Radius of particle, m : Mass of particle

After Implementing this scheme in Python the particle is doing Brownian motion Now we need Aggregation mechanisms and conditions.

Aggregation Scheme:

First, we define a seed point, in our code it is the origin $(0,0,0)$. Then We aggregate the particle to the seed if the distance between them is less than or equals to $2R_p$ and then for next particle we aggregate it to the seed or any aggregated particle if its distance to them is less than or equals to $2R_p$ and the position is chosen in the direction from which particle is approaching that aggregate point.



Computational Tricks and Technical Details

Data Structure for Storing Aggregate Data:

To store aggregate data efficiently, we've created a user-defined data structure called "Cluster". It contains three properties: master (for seed position), children (for connected aggregate points), and id. This nested graph-like structure ensures scalability for future projects.

Checking for Aggregation: To optimize computation, we introduced the "*Size of aggregate*," defined as the maximum distance of an aggregate point from the seed. Using this distance as a radius, we define a sphere centered at the origin (seed). We only check sticking conditions for particles that enter this sphere, reducing unnecessary computation.

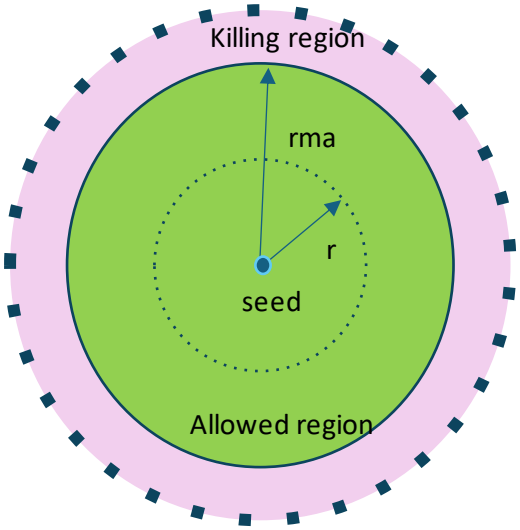
No Collision: Particles do not have any collision and more technically the simulation is performed one particle at a time.

Killing a particle: We kill a particle if it goes farther than 3 times size of aggregate from the origin.

Density Function C(r), which represents the number of particles located at a distance r from the reference particle for all particles in the cluster

Simulation Details

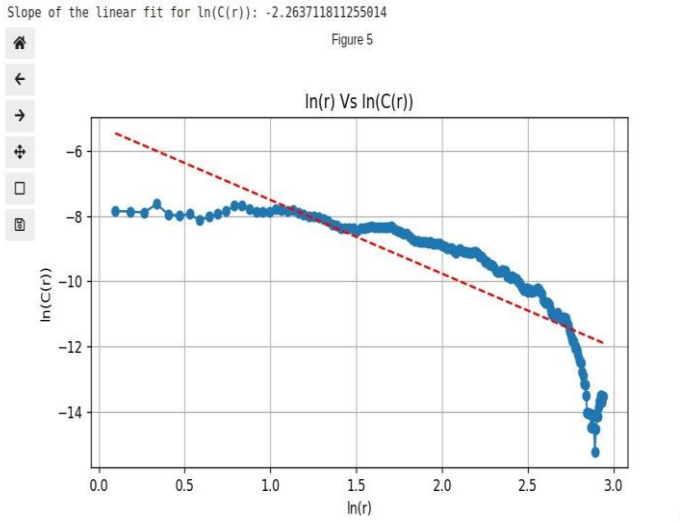
In our simulation of DLA, the idea is to generate a particle at a sufficiently large distance of about $r = 10 \cdot \delta + d_{max}$ and then solve its dynamics using Langevin equation and if in case it goes outside $r_{max} = 20 \cdot \delta + d_{max}$ then killing the particle and repeating until it gets aggregated. This is done in a loop until we have desired number of particles in aggregate.

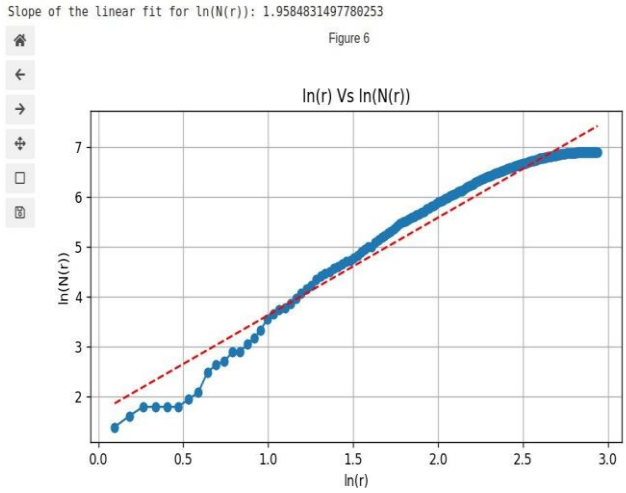


Here, δ : Mean free path, d_{max} : Size of Aggregate

Simulation 1

Parameters: $m = 0.4;\ R = 0.2;\ \eta = 1;\ K_bT = 3;\ dt = 0.1;\ N = 1000$



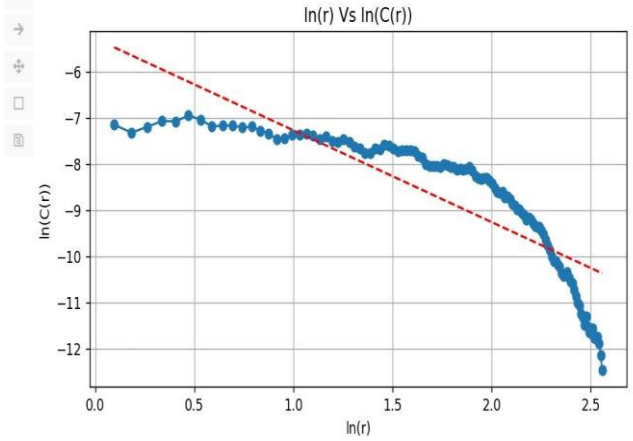


Simulation 2

Parameters: $m = 0.4;\ R = 0.2;\ \eta = 0.01;\ K_bT = 3;\ dt = 0.1;\ N = 1000$

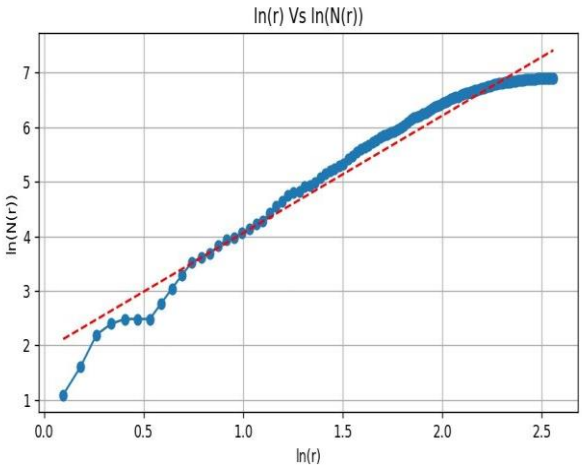
Slope of the linear fit for ln(C(r)): -1.9902874934222774

Figure 7



Slope of the linear fit for ln(N(r)): 2.1486920481680074

Figure 8

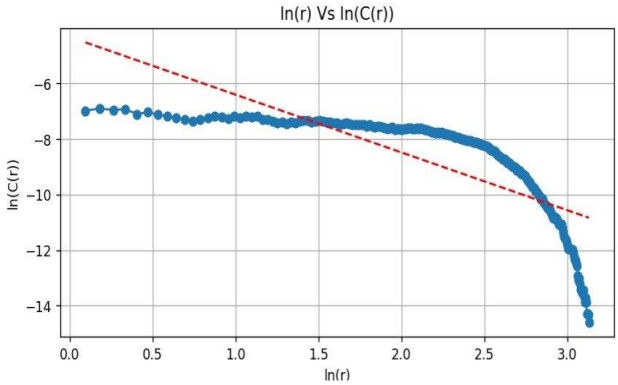


Simulation 3

Parameters: $m = 0.2;\ R = 0.4;\ \eta = 0.01;\ K_bT = 2;\ dt = 0.1;\ N = 5000$

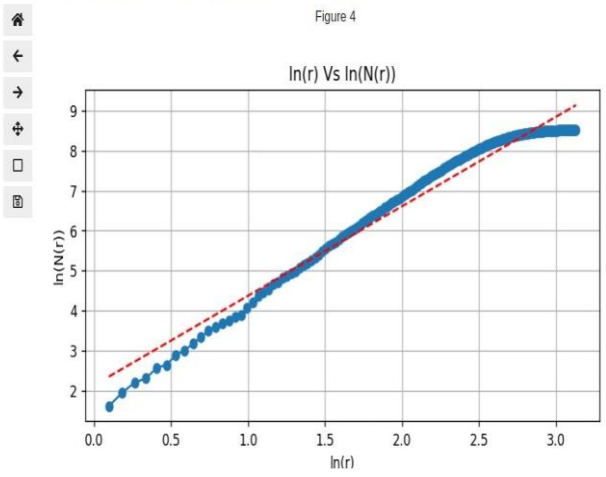
Slope of the linear fit for ln(C(r)): -2.075589890198033

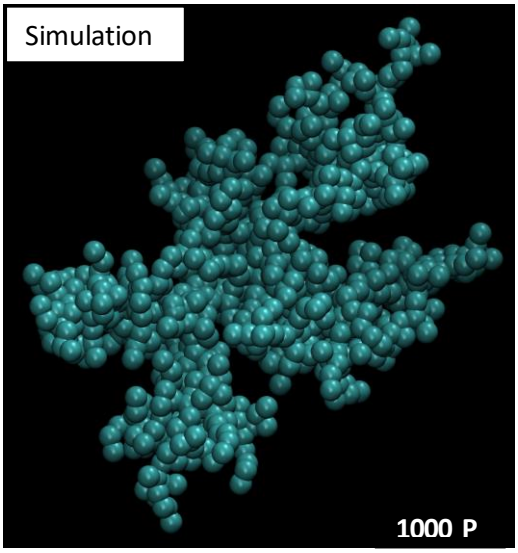
Figure 3

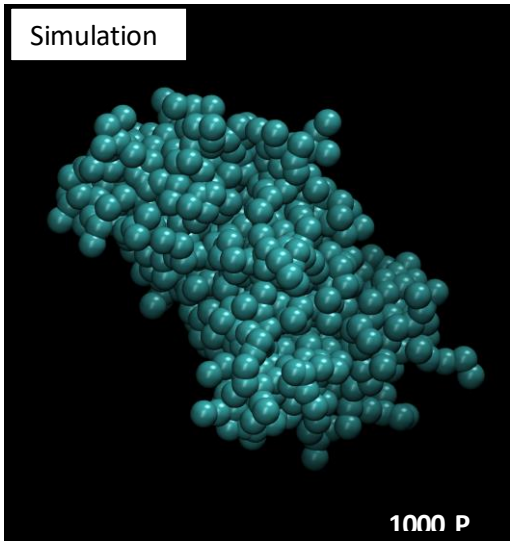


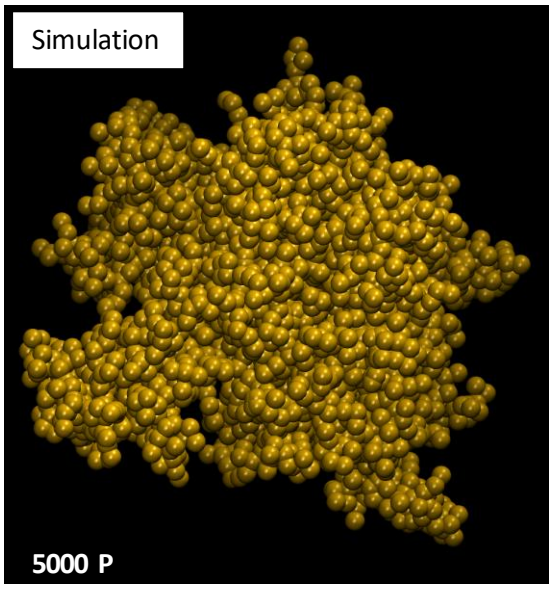
Slope of the linear fit for ln(N(r)): 2.2387427454413973

Figure 4









Discussion On Simulation Result

In three simulations, two with 1000 particles each but different viscosity values and one with 5000 particles:

1. Comparing Sim 1 and Sim 2: Sim 1 has viscosity of 1 and fractal dimension of about 1.95, while Sim 2 has viscosity of 0.01 and fractal dimension of 2.15. Lower viscosity leads to lower fractal dimension due to reduced drag force and increased particle mean free path, resulting in a denser and more complex aggregate.

2. Comparing Sim 2 and Sim 3: Sim 3 has 5000 particles and a fractal dimension of 2.24 with viscosity set at 0.01. Increasing particle count from 1000 to 5000 causes a slight increase in dimension by about 0.09.

Results and Conclusions

From Langevin Method:

We conclude that aggregates formed in less viscous fluids tend to exhibit more surface area, increased complexity, and higher density, as observed in Simulations 1 and 2. Additionally, increasing the number of particles under the same conditions resul ts in only a slight increase in fractal dimension, as seen in Simulations 2 and 3.

From Random Walk Method:

In random walk method, we simulated DLA in 2D in different condition like pnn=1, pnn=0.3.and we additionally added secondary sticking probability, psnn=0.15. Then we saw that the fractal density is higher when we are considering that. After that we considered probability higher in one direction. Then we got higher fractal density in that specific direction. We saw animate d simulation of DLA using *pygame*. After we analysed the multiseed particles using different colours at different time of aggregation and studied about the DLA

References:[1] T. A. Witten and L. M. Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. Phys. Rev. Lett., 47:1400–1403, Nov 1981