OBJECTIVE:

The objective of this project has been to generate fractal surfaces with varying fractal dimensions. The variation in fractal dimension with varying number of particles and lattice size has been studied. The surfaces have been generated on a square lattice. This has been done through the use of C and python programming languages

SIMULATION:

The fractal surfaces generated for this project are known as Diffusion limited aggregates (DLA). The methods used here to generate aggregates is based off of the Witten-Sander model with a few modifications to reduce computation time and to make sure that a cluster of appreciable size if formed. The aggregates in this report generated using python, while they have also been written in C it was to plot and make changes using python. The resultant aggregate essentially the x and y co-ordinate of each point. The co-ordinates are essentially indices of a 2D array . The indices of the array can only be positive, so they look similar to a matrix when visualized. Both programs written in C and python have the same code structure.

The simulation takes place on a 2-D square lattice. Such lattice makes it easier to generate points on a circular perimeter. The x and y co-ordinates are generated and stored in two separate 1-D arrays and then a random x co-ordinate of a point and a random y co-ordinate of appoint are selected. They symmetry allows us to select any two random x any y co-ordinates and generate a point on the perimeter. In case of a square boundary we need to Select a particular fixed co-ordinate, x or y and then randomly select a point with the other co-ordinate, and do the same with different combinations of the co-ordinates to get the four sides of the square boundary.

In case of the circular boundary a particular the launching perimeter is at a distance 30 units of lattice constants from the center seed. The DLA overall has three different boundaries, inner (R), outer $(Rout)$ and Maximal cluster radius $(Rmax)$. The inner launching radii is modeled by 30 units every time. equivalently to save run time**.** A particle is allowed to walk only up to a certain distance, in this case its **3 times $\mathbf{Rmax}$** . Each particle is allowed to take no more than 1000 steps. *Rmax* is the largest size up to which a cluster can grow. If even a single point of the cluster reaches this radius , the cluster stops growing, even if there more particles left to be launched and vice versa. The step length of a particle is always kept constant at one lattice unit.

If the particle moves out of this radius its terminated and a new particle is launched. But in order to get a cluster of appreciable size we need to have a larger radius. But a large radius leads to very little to no aggregation , hence we employ a continuously expanding radius. This is easier but takes longer to implement for a circular boundary as we have to compute the radius of the circle the points lie, every single time. The number of particles launched is specified at the beginning. Circle points have been generated using formula for polar co-ordinates **x=Rcosθ** & **y=Rsinθ** Using a single While-loop. Two for-loops would be required to generate points using distance formula. Since the points would be centered around the origin they have to be adjusted to the center of our array. **a = x + (length of our array / 2)** & **b = y + (length of array / 2).** Where a & b are our new x and y co-ordinates.

The fractal dimension (D) is calculated using the formula ,

$$\mathbf{D} = \frac{\log(\text{total No of particles in cluster})}{\log(\text{maximal cluster radius})}$$

RESULTS & DISCUSSION:

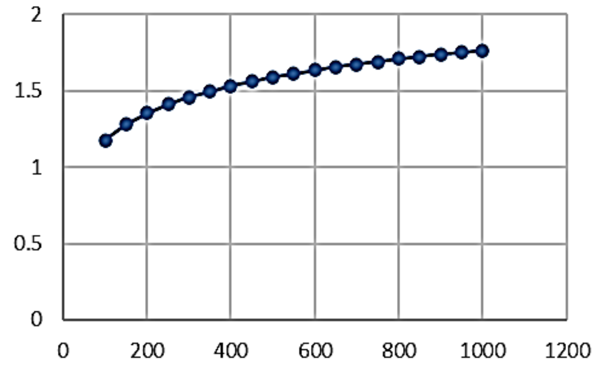
The fractal surfaces considered are only the ones formed using a circular launching boundary, since the ones formed using square launching perimeter turned out to be extremely skewed in directions facing one of the four sides of the square perimeter. It is to be noted that the clusters never grew in the direction of one of the vertices of the square. But these clusters also took less time to form than ones done using a circular boundary.

The primary objective was to observe the variation of fractal dimension with variation in the different parameters. It is obvious from the formula above that keeping the lattice size and particle numbers same while varying the maximal cluster size would give linear relationship between Rmax and fractal dimension D. Hence our primary observations are that of Fractal dimension for varying lattice sizes with a fixed number of particles and vice versa.

The table below shows values of fractal dimension for different cluster sizes for a fixed lattice of size 310 x 310.

Number of particles	Fractal dimension
100	1.177
150	1.281
200	1.354
250	1.411
300	1.458
350	1.497
400	1.532
450	1.562
500	1.589
550	1.613
600	1.635
650	1.656
700	1.675
750	1.692
800	1.709
850	1.724
900	1.739
950	1.753
1000	1.766

Table .1. The above observations are for a lattice of fixed size 310 x 310, with a maximum cluster radius of 50 units, and maximum walk radius of 150 units.



Graph 1. Fractal dimension (D) vs Cluster size for Table.1

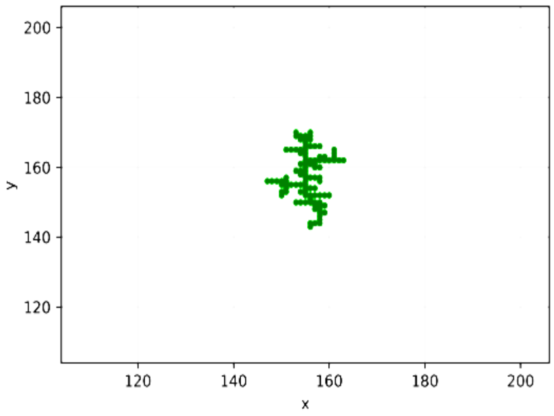


Figure 1. 100 particle aggregate

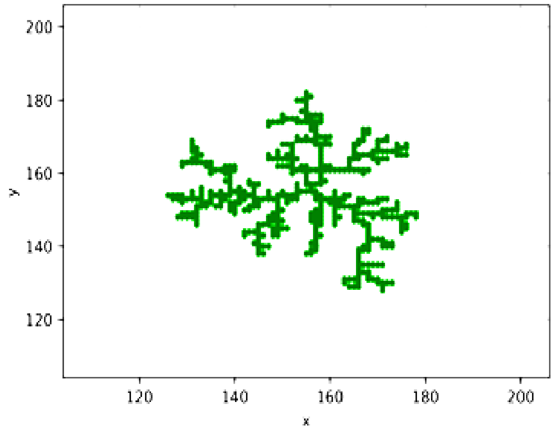


Figure 2. 500 particle aggregate

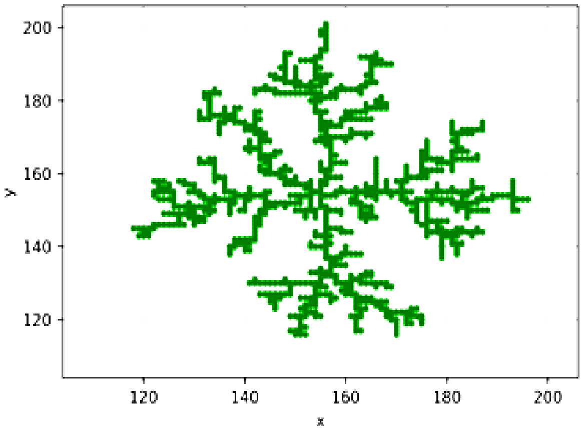


Figure 3. 1000 particle aggregate

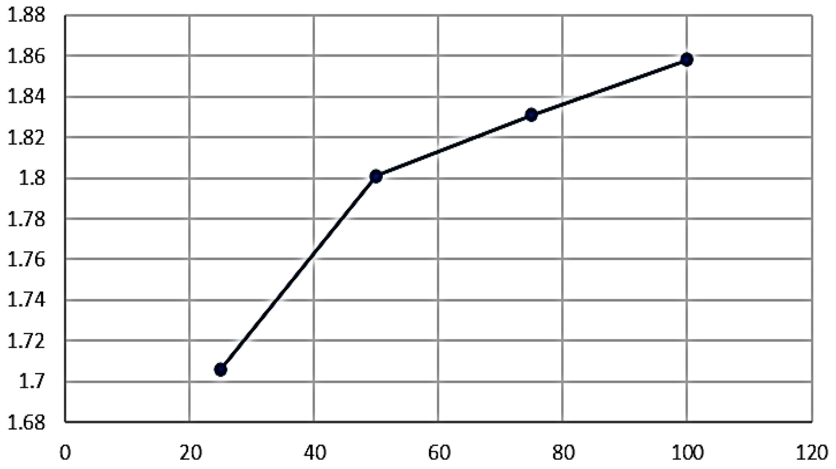
As seen from Table.1 and Graph.1 the fractal dimension goes on increasing linearly for increasing cluster size. In order to accommodate a cluster of size up to 1000 it is required to choose maximal radius and lattice size accordingly. The increasing fractal dimension implies that as the cluster size increases for a large enough value n, as n→ ∞ the fractal dimension approaches the Euclidean dimension of the plane itself.

Unlike as seen for the square boundary where the cluster is skewed in one of the four sides, as particles launched from the corners of the square boundary have a lesser chance of reaching the central part, due to the self-avoiding nature of the walk.

Below are the observations obtained from varying the lattice size Walk radius & cluster radius constant while keeping the number of particles launched constant at 10,000.

Maximal cluster radius	Maximal Random walk Radius	Lattice size	Fractal dimension	Average Cluster size
25	75	150 x 150	1.706 ± 0.019	243.4
50	150	300 x 300	1.801 ± 0.016	1162.5
75	225	450 x450	1.831 ± 0.012	2650.4
100	300	600 x 600	1.858 ± 0.007	5211.5

Table .2 The above observations are for 10,000 particles launched, with varying maximum cluster size , Walk radius and lattice size.



Graph 2. Fractal dimension (D) vs Maximum cluster radius for Table.2

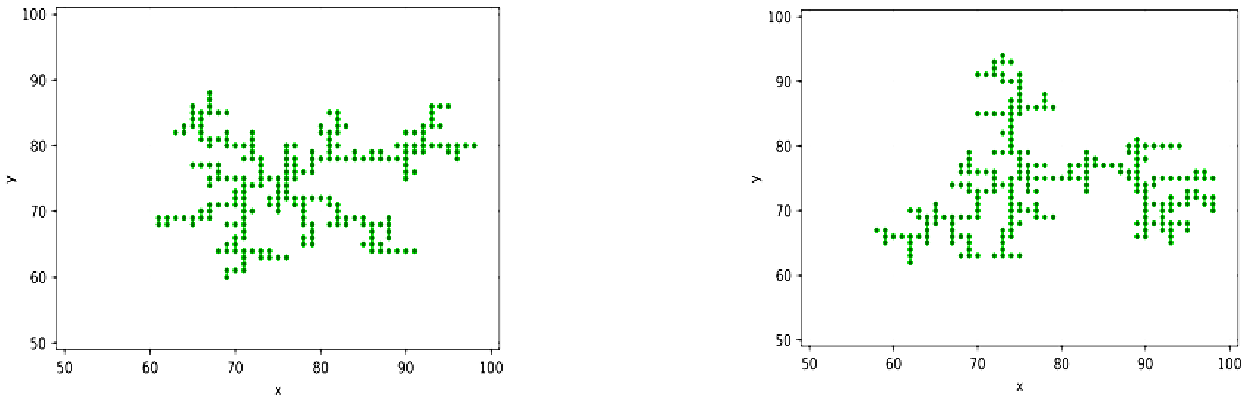


Figure 4 & 5. Aggregates for lattice size 150 x 150

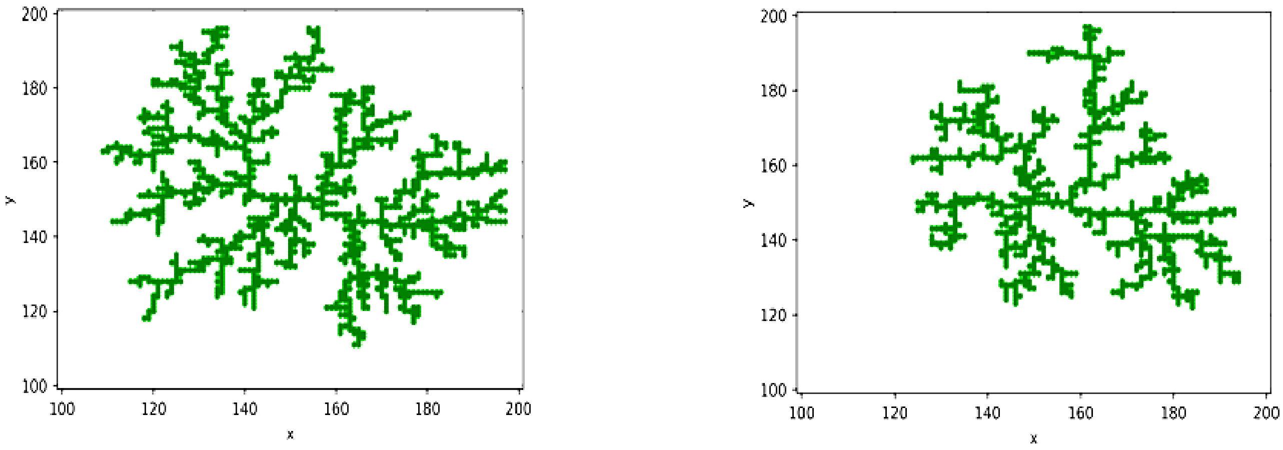


Figure 6 & 7. Aggregates for lattice size 300 x 300

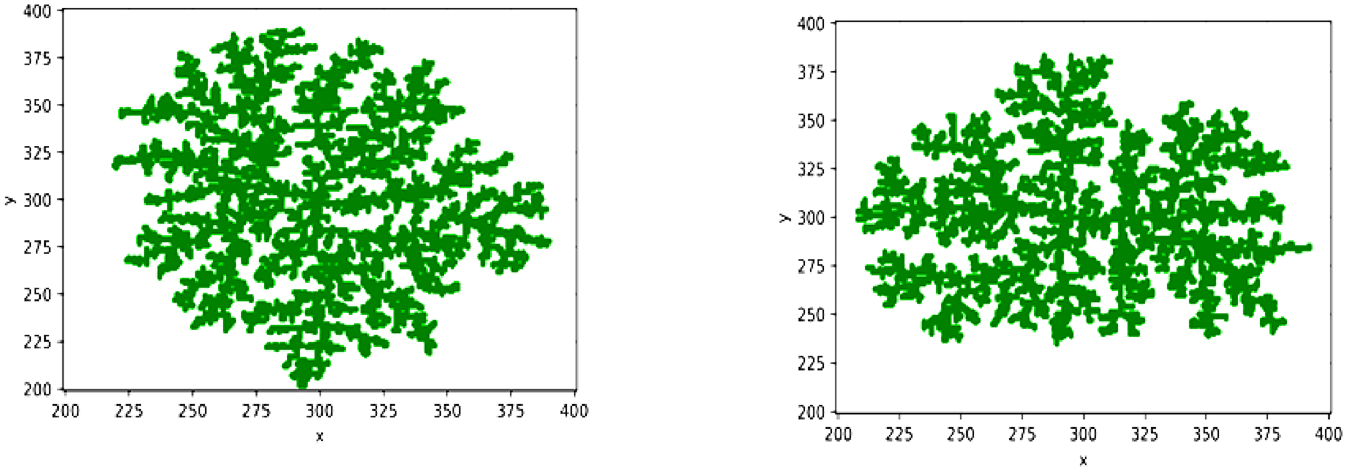


Figure 8 & 9. Aggregates for lattice size 600 x 600

As seen from Graph.2 The fractal dimension is linearly proportional to the lattice size, as lattice size increases the particles path length increases giving them a greater chance of sticking to the interior regions of the cluster. Also, the cluster obtained for lattice size 600 x 600 is fairly isotropic , due to the seed being equidistant to every point on the launching boundary.

LIIMITATIONS:

Most of the restrictions in this this project are due to the limitations of the hardware or computation power. A simulation, of cluster size more than 10,000 leads to the algorithm having computational complexity that is far outside the capacity of a consumer grade computer. The moving seed model cannot be implemented due to this exact reason, as it would need to iterate over the entire 2D array every time a particle is attached to the cluster, making the algorithm even more complex.

Also, the aggregate formed using Square launching boundary, for a large number of particles does not yield the same result as in the work done by others. They tend to grow in one of the directions allowed for the random walk to take. Changes would be made so as to get the desired results for a large cluster. The below given examples are for 50,000 particles launched for a lattice of size 600 x 600, giving us a cluster size between 2000-3000.

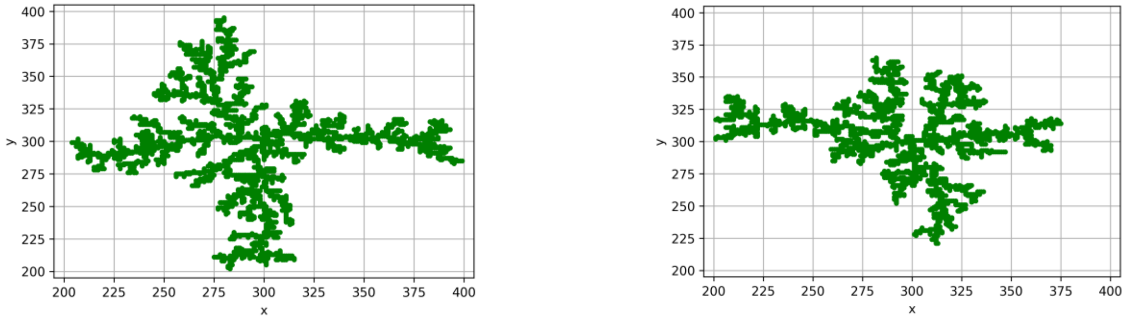


Figure 10 & 11. Aggregates generated using a square launching boundary for lattice size 600 x 600

FUTURE WORK:

While this report just serves as an introduction to fractals the actual application of this concept is far too many to explain in this project alone. In terms of DLA alone there are many applications. The DLA clusters generated here are only the most basics of structures. While further expanding on the work done here it is to be seen that clusters are generated using more novel methods with the aid of more computing power.

Certain observations such as aggregation efficiency i.e. the relationship between the number of particles lunched to the cluster size , requires particles to be launched in the millions in order to form clusters of a large size(in thousands). As of now this has not been done due to the limitations stated above.

Minor changes to the code can be added to create the algorithm for the moving seed model, namely by iterating through the entire lattice (2-D array) every time a new particle is added and randomly choose a particle from the growing cluster. While this can be easily done using the two-for loops to iterate through the array, it takes a significant amount of power to compute. So, a better alternative is to be sought.

Random deposition, Balistic deposition and will be studied in the future, referencing the previous works of authors, with emphasis on studies based on heterogenous catalysts, monomer dimer reactions,LH and ER mechanisms. Algorithms for these programs will include the parameters such as sticking probability and mean field parameter which is lacking in the basics program written for this report. These future projects would be studied using analytical tools such as multifractal analysis and dynamic scaling.

- Python code:

https://colab.research.google.com/drive/114x0U0a0w7NqEDqknp7LGWfcJqc3lyyB?usp=sharing

- C code:

https://drive.google.com/file/d/1DbM9N75XxIPgArMjYK9LHl9fMsDhP5Le/view?usp=sharing