PHYC40600 Physics with Astronomy and Space Science Lab 2; Diffusion-Limited Aggregation

Daragh Hollman
daragh.hollman@ucdconnect.ie

(Dated: October 22, 2023)

The aims of this report were to investigate the Diffusion-Limited Aggregation (DLA) model and make comparisons between the use of a square lattice and a hexagonal lattice. The fractality of such clusters was also investigated including a visual estimate of the fractal dimension. The DLA model was run for approximately 1000 particles on both lattices which each had a size 101 × 101 cells. The resulting clusters did appear to be fractal with dimensions $D(\square) = 1.9$ and $D(\hexagon) = 1.8$ for the square and hexagonal lattice respectively. An investigation into the probability of growing at each growing point was also carried out for the next state after the initial, when the cluster consists of two particles. This resulted in unexpected and counter-intuitive values of probability which are likely due to an error in the implementation of the DLA algorithm or following optimisations.

I. INTRODUCTION

The diffusion-limited aggregation model proposed by Witten and Sander in 1981 [1] has become prominent in the field of computational science and has a wide range of applications in both physical and biological sciences [2]. It has the ability to describe how many objects in nature grow through the random aggregation of smaller pieces, such as snowflakes [3], proteins [4], and topographical drainage models [5].

A. Aggregation

Aggregation is the method describing they way many objects form, from snowflakes [3], to proteins [4], to clouds [6]. It is the clustering of a large number of micro-objects (such as particles), to create a larger macro-object. Typically this occurs through collisions - such as ice molecules sticking together to form snowflakes [3]. The process of aggregation can be easily modelled computationally due to its simplicity.

B. Diffusion

Diffusion describes how particles in a system fill empty space through random movements until equilibrium is reached [7]. A particle in empty space is free to move in all directions, and will do so randomly, as there is no other particles around it. This is useful to approximate how a particle will approach the aggregation cluster. To achieve these random movements, a Monte-Carlo simulation is required.

1. *Monte-Carlo Simulations*

Monte Carlo simulations are used in the modelling of random processes. A good random number generator is required to ensure that no sequential pattern is produced by chance [8]. Computers can generate numbers which seem to be random called pseudo-random numbers. Pseudo-random number generators can vary in complexity and may appear to present truly random numbers on a small scale, however, they will always be a repeating pattern with a period of re-occurrence [9]. Hence, a pseudo-random number generator with re-occurrence period larger than the number of simulation steps taken must be chosen.

2. *Random Walks*

A random walk simulation is a simple type of Monte Carlo simulation. One example of a random walk is the nearest-neighbour walk on a square lattice [2]. The walk starts from an initial position - typically the origin (0,0). For each step in the walk, the new position along the path of the walk is one of the adjacent neighbours to the current position i.e. $(i+1, j), (i, j-1), (i-1, j), (i, j+1)$ for current position (i, j). The chosen neighbour is picked randomly using a psuedo-random number generator [2].

C. Diffusion-Limited Aggregation

The diffusion-limited aggregation model (DLA) was proposed by Witten and Sander in 1981 [1] and expanded upon in 1983 [10]. In this model, a central initial particle is created and a second particle is created at a distance away from the cluster. This particle undergoes random walk diffusion until it eventually moves adjacent to the cluster, at which point the particle sticks, becomes part of the cluster, and this process is repeated.

D. Fractals and Fractal Dimension

Many aggregation models result in fractal structures [11]. The fractal can be characterised by the fractal dimension [12]. The fractal dimension can be calculated using the box-counting dimension [13] using the following equation:

$$D = \lim_{s \to 0} \frac{\ln(N(s))}{\ln(1/s)} \qquad (1)$$

where in each step of the counting, the object is covered by a grid of boxes with side length s. $N(s)$ is the number of boxes which intersect the fractal [13]. This method can be applied computationally to fractals using very small box sizes for good accuracy, however a visual estimate can also be made using much larger box sizes.

II. COMPUTATIONAL METHODS

A. Square Lattice

An implementation of the Diffusion-Limited Aggregation algorithm was implemented using python. The exact algorithm is as follows: A square grid was initialised with an initial particle in the centre. A new particle was created on a random site on a circle around the cluster with radius $r = R_{\text{max}} + 2$, where R_{max} is the furthest distance to the origin of all the particles in the cluster. For clarity we will refer to this as the placement circle. We can justify the use of this placement circle to represent random aggregation in nature as any particle randomly walking from a position far away from the placement circle will cross the circle at a random position [2]. This particle then undergoes a random walk choosing to move in pseudo-random directions until becoming adjacent to the cluster. An example path of a particle to the cluster is shown by Meakin [2] in figure 1 indicated by S_1. Note that in this figure, they chose to use a placement circle with $r = R_{\text{max}}+5$ instead.

The seed used to generate the pseudo-random numbers for this simulation was was chosen to be the system time resulting in different events on each run. The python library `random` was chosen to generate pseudo-random numbers as it uses the Mersenne Twister algorithm [14] as its generator. It produces 53-bit precision floats and has a re-occurrence period of $2^{19937} - 1$ [15].

1. *Computational Bottleneck*

Through running the simulations it became clear that the script spent most time during the random walk of the new particles. As such, methods were introduced to reduce this bottleneck. One of these methods to save time is to stop the walking particle when it reaches a distance far away from the cluster [2]. If the particle reached a distance of $2R_{\text{max}}$ from the origin it was removed and replaced by a new one on the placement circle. Such a path is indicated by S_2 in figure 1.

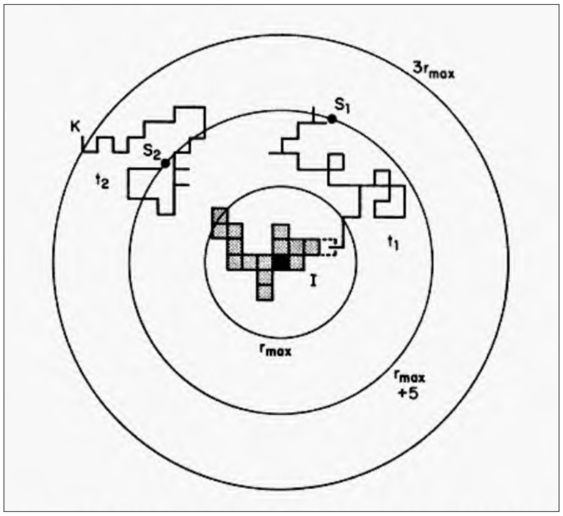

FIG. 1: Two example paths described by Meakin [2]. Here they have used slightly different definitions of the circles with the killing circle being at $3r_{\text{max}}$ and the launching circle at a distance of $r_{\text{max}} + 5$. We see two potential paths, labelled S_1 and S_2. S_1 is created on the launching circle and randomly walks inwards towards the cluster. It moves adjacent to the cluster and sticks, finishing its walk. S_2 is created on the launching circle and randomly walks outwards away from the cluster. When it moves outside of the killing circle it is removed. Note that this diagram does not include an increasing step size as a function of distance as discussed.

Another measure introduced was to increase the size of the steps taken in the random walk when the particle was sufficiently far away from the cluster [2]. If the particle was at a distance of $R > R_{\text{max}}$, a larger walk step of $R - R_{\text{max}} - 1$ was used provided this was larger than 1. This increased walk step ensured that the walker stays close to the cluster as any step towards the cluster from outside R_{max} would bring the walker to a distance of R_{max}. Both of these methods were implemented into the simulation.

Meakin suggests another more accurate and efficient procedure to return any particle which moves outside of the placement circle back to the placement circle [2]. This restricts the particle to within the placement circle while maintaining the random walking nature of the particle. This method was however not implemented into this simulation due to lab time constraints.

B. Hexagonal Lattice

It is also possible to look at this DLA algorithm with different lattice geometry. One such option is a hexagonal lattice (also called a triangular lattice). To convert from a square lattice to a hexagonal lattice, the same methods and algorithms could be used, with slight adjustments. The square lattice was adjusted by moving every odd row along x by half the cell size. This can be seen in figure 2. This has the effect of each square in the lattice now hav-

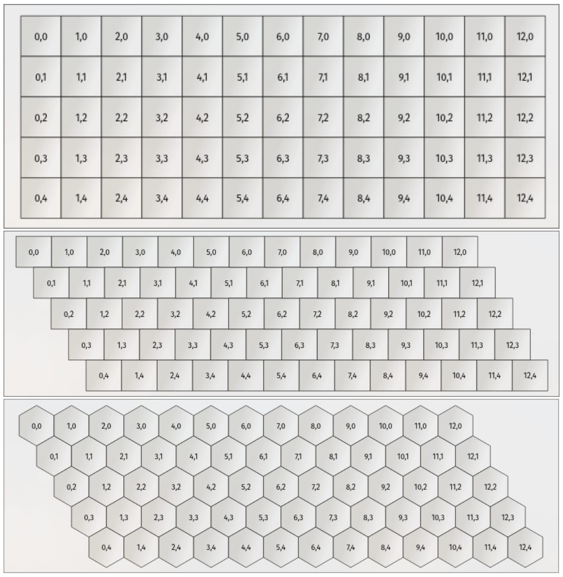

FIG. 2: A collection of figures showcasing how a grid of squares can be transformed to create a grid of hexagons [17]. Here the final step of changing the geometry is being ignored, however, functionally it is the same as keeping the squares in our case.

ing six neighbours instead of four. The determination of these neighbours is obvious visually but should be noted that it non-trivial in practice due to a dependence on the row number. Amit Patel discusses coordinate systems for hexagonal grids [16] and includes a section on neighbour calculation showing its dependence on the parity of the row.

III. RESULTS AND DISCUSSION

A. Square Lattice

A DLA cluster was initialised on a square lattice of shape 101 × 101 cells. The algorithm was left to run, adding approximately 1200 particles to the cluster. The results from this are plotted in figure 3. The result appears fractal-like. Four primary paths emerge from the initial particle which each branch off into smaller branches, which in turn have their own branches. The fractal dimension of this cluster was visually estimated using the box counting method with a box side length of 8 cells. The fractal dimension was found to be approximately 1.9. This is within 12% of the theory value of $D = 1.70$ of a DLA cluster on an open plane [18].

B. Verification of Grow Point Probability

Initially in the system, the four available new growing points each had an equal probability of the particle encountering them, $p_i = \frac{1}{4}$. After a second particle had been added to the cluster, the cluster has six available growing points and is now asymmetrical. A Monte Carlo simulation was set up to test the probabilities for this cluster. This simulation ran for 1000 potential new particles with the results plotted in figure 4. The colour in this figure represents the probability by the number of times the cluster grew from that growing point, normalised to the number of particles tested. A value of 1 means that every new particle grew at that point, and a value of 0 means no new particles grew at that point. We see three points with a higher probability of ≈ 0.19, and three with a lower probability of ≈ 0.14.

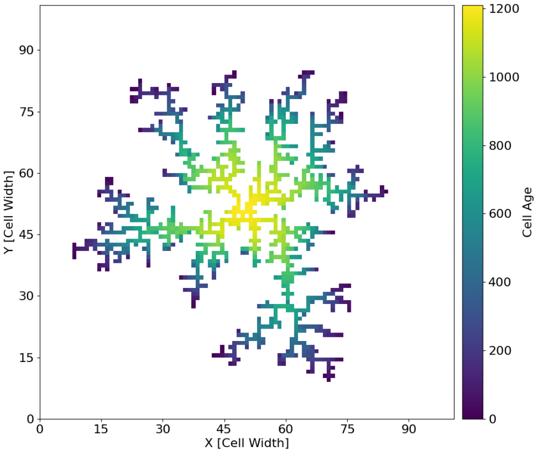

FIG. 3: A DLA cluster with approximately 1200 particles on a square lattice. Here, colour of the cell represents how many time steps since it has been created.

This is not the result that was expected. It was predicted that as the cluster now consisted of two particles, there would be two cluster-adjacent lattice locations which were closer to the placement circle than the other four available. Given the equal probability of the new particle occurring from any direction on the placement circle, it is intuitive that the closer points would be encountered more often than the others, given a random walk. The issue responsible for this is not immediately obvious. Given more time this would have been investigated further, however, we can assume this unexpected result to be the cause of one of two potential issues. It is possible that the DLA algorithm was implemented incorrectly which could result in such misbehaviour. Another possibility is that an incorrect implementation of any of the time saving measures during the random walk could have unintended effects which might also cause this.

C. Hexagonal Lattice

Similarly to the square lattice, a DLA cluster was initialised on a hexagonal lattice with shape 101 × 101 cells. The algorithm was left to run, adding approximately

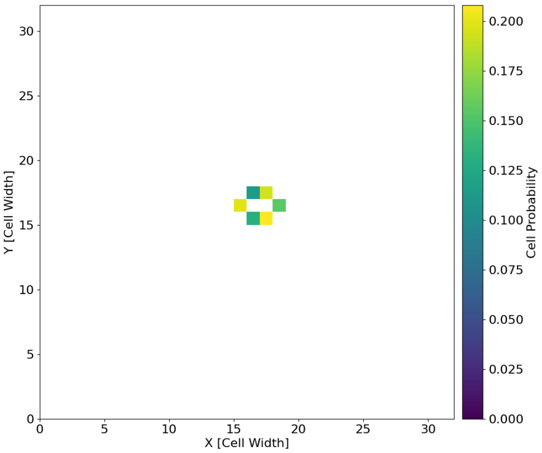

FIG. 4: A representation of the probability of particle placement after the cluster reaches mass 2. Here the colour represents the ratio of particles which landed in that cell over the course of 1000 runs.

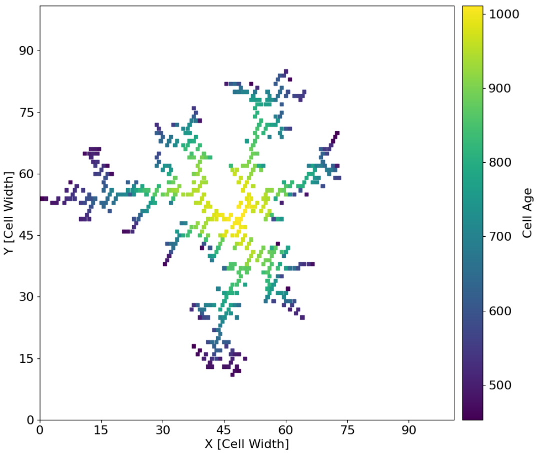

FIG. 5: A DLA cluster with approximately 1000 particles on a hexagonal (triangular) lattice. Here again, colour of the cell represents how many time steps since it has been created.

1000 particles to the cluster. The results from this are plotted in figure 5. The result again appears fractal-like, this time with six primary paths emerging from the initial particle. The fractal dimension of this cluster was again visually estimated using the box counting method with a box side length of 8 cells. The fractal dimension was found to be approximately 1.8. This is within 6% of the theory value of $D = 1.70$ [18].

IV. CONCLUSION

The aims of this report were to investigate the Diffusion-Limited Aggregation model and make comparisons between the use of a square lattice and a hexagonal lattice, and investigate the fractality of such clusters including a visual estimate of the fractal dimension. The DLA model was run for approximately 1000 particles on both lattices which each had a size 101×101 cells. The resulting clusters did appear to be fractal with dimensions $D(\square) = 1.9$ and $D(\text{⬡}) = 1.8$ for the square and hexagonal lattice respectively. These both compare well to the theory expected value of $D = 1.7$ [18]. An investigation into the probability of growing at each growing point was also carried out for time $t = 2$, when the cluster consists of two particles. This resulted in unexpected and counter-intuitive values of probability which are likely due to an error in the implementation of the DLA algorithm or following optimisations.

[1] T. A. Witten and L. M. Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. *Physical review letters*, 47(19):1400–1403, 1981.

[2] Paul Meakin. *Fractals, scaling and growth far from equilibrium*, volume 5;5.;. Cambridge University Press, Cambridge, 2011.

[3] Christenia Westbrook, R. Ball, P. Field, and Andrew Heymsfield. Universality in snowflake formation. *Geophys. Res. Lett.*, 31, 12 2003.

[4] Birgit Strodel. Amyloid aggregation simulations: challenges, advances and perspectives. *Current Opinion in Structural Biology*, 67:145–152, 2021. Theory and Simulation/Computational Methods Macromolecular Assemblies.

[5] Donald L. Turcotte. Self-organized complexity in geomorphology; observations and models. *Geomorphology (Amsterdam)*, 91(3-4):302–310, 2007.

[6] Allison A. Wing, Catherine L. Stauffer, Tobias Becker, Kevin A. Reed, Min-Seop Ahn, Nathan P. Arnold, Sandrine Bony, Mark Branson, George H. Bryan, Jean-Pierre Chaboureau, Stephan R. De Roode, Kulkarni Gayatri, Cathy Hohenegger, I-Kuan Hu, Fredrik Jansson, Todd R. Jones, Marat Khairoutdinov, Daehyun Kim, Zane K. Martin, Shuhei Matsugishi, Brian Medeiros, Hiroaki Miura, Yumin Moon, Sebastian K. Müller, Tomoki Ohno, Max Popp, Thara Prabhakaran, David Randall, Rosimar Rios-Berrios, Nicolas Rochetin, Romain Roehrig, David M. Romps, James H. Ruppert, Masaki Satoh, Levi G. Silvers, Martin S. Singh, Bjorn Stevens, Lorenzo Tomassini, Chiel C. van Heerwaarden, Shuguang Wang, Ming Zhao, and CA (United States) Lawrence Berkeley National Laboratory (LBNL), Berkeley. Clouds and convective self-aggregation in a multimodel ensemble of radiative-convective equilibrium simulations. *Journal of advances in modeling earth systems*, 12(9):e2020MS002138–n/a, 2020.

[7] LibreTexts Physics. Diffusion. available at https://phys.libretexts.org/Bookshelves/University_Physics/Book%3A_Physics_(Boundless)/12%3A_Temperature_and_Kinetic_Theory/12.9%3A_Diffusion, accessed on 18/10/23.

[8] Nicholas T. Thomopoulos. *Essentials of Monte Carlo simulation: statistical methods for building simulation models*, volume 9781461460220. Springer, New York, 1. aufl.;1;2013; edition, 2013;2012;2015;.

[9] D.M. Ceperley D.D. Johnson. Generation of random numbers, atomic-scale simulations, 2001. available at https://courses.physics.illinois.edu/phys466/sp2013/lnotes/random_numbers.html, accessed on 18/10/23.

[10] T. A. Witten and L. M. Sander. Diffusion-limited aggregation. *Physical review. B, Condensed matter*, 27(9):5686–5697, 1983.

[11] Paul Meakin. Fractal aggregates. *Advances in Colloid and Interface Science*, 28:249–331, 1987.

[12] Benoit B. Mandelbrot. *The fractal geometry of nature*. Freeman, New York, updat and augment edition, 1982.

[13] Martin Bouda, Joshua S. Caplan, and James E. Saiers. Box-counting dimension revisited: Presenting an efficient method of minimizing quantization error and an assessment of the self-similarity of structural root systems. *Frontiers in plant science*, 7(2016):149–149, 2016.

[14] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM transactions on modeling and computer simulation*, 8(1):3–30, 1998.

[15] Python Software Foundation. random - generate pseudo-random numbers. available at https://docs.python.org/3/library/random.html, accessed on 18/10/23.

[16] Amit Patel. Hexagonal grids. available at https://www.redblobgames.com/grids/hexagons/, accessed on 22/10/23.

[17] Amit Patel. Grid parts and relationships. available at https://www.redblobgames.com/grids/parts/, accessed on 22/10/23.

[18] J. M. Tenti, S. N. Hernández Guiance, and I. M. Irurzun. Fractal dimension of diffusion-limited aggregation clusters grown on spherical surfaces. *Physical review. E*, 103(1-1):012138–012138, 2021.

Appendix A: Python Script