Thomas Kagan
PHYS 4041
Prof. Jorge Vinals
10/31/20

Comparing Isotropic Properties of Two Diffusion Limited Aggregation Simulation Techniques

Introduction

"Of all the possible pathways of disorder, nature favors just a few."
— James Gleick, Chaos: Making a New Science[1]

From snowflakes, to coral reefs, to lightning strikes, to string theoretical models of subatomic particles[5], Diffusion Limited Aggregation is a subtle yet ubiquitous mechanism underlying physical systems at every scale. And, being simple to describe physically, DLA is an accessible way to explore the fine line between randomness and emergent order—and so it has been a subject of great interest in the study of chaos[5]. In this paper, I present the physical conditions for which this particular flavor of natural and computational beauty emerges; and I present two models of gas particle diffusion: *Simulation A*, a traditional model described by a Brownian model of the motion of particles, and *Simulation B*, a technique I propose that derives from the diffusion equation in an attempt to improve parallelizability of the algorithm; and I present findings pertaining to the symmetry of the shapes generated by each Simulation to evaluate their respective fitness at application to a variety of domains.

Particles *diffuse* as they follow a chaotic, fractal path. Any path they happen to follow is equally likely, but there are uncountably many such paths of arbitrary resolution. This phenomenon is *Brownian motion.* To simulate DLA, begin with a single particle stationary at the center of the coordinate plane, called the *seed*. Then, introduce a new particle far away from the seed to undergo a random walk of small, discrete steps in time and space around the coordinate plane. If the particle contacts the seed, it sticks, forming an *aggregate*. Repeat for any number of desired particles. This will cause the aggregate to grow out in tendrils. A peninsula has a lot more surface area than its respective arc of the circle, so the longer tendrils grow, the more likely a particle will land on a tendril. It is through this mechanism that fractal growth develops, where many small tendrils surround larger tendrils, which in turn branch off into many small tendrils, a few of which eventually grow into larger tendrils, and so forth.

This report is organized in 6 sections.

- This *Introduction* section which you have just read.
- The following *Objective* section describes the purpose of this report, and what kind of results are intended from the process.
- The *Approach* section presents the algorithms for both Simulations as well as the technique that will be used to measure the aforementioned kind of symmetry. Each algorithm is prefaced with its analytical basis and a mention to relevant numerical accuracy/stability considerations.
- The *Results* section includes Figures of sample aggregates generated and reports the numerical data of each across three Trials for each Simulation.
- The *Conclusion* section summarizes the findings of this report and my interpretation of them. It explains the significance of the results to DLA simulation, and contextualizes their ultimate usefulness.
- The *Future Work* section offers the interested reader with research suggestions, whose results are needed to complete this line of inquiry.
- A bibliography section.

Objective

This report compares the aggregate forms from two different methods of 2-dimensional simulation: conventional DLA simulation of square cellular automata as outlined by Landau[6] (*Simulation A*), and a technique I propose that assists the simulation, and is derived from the diffusion equation (*Simulation B*). **This report aims to evaluate if Simulation *B* is a valid method of performing DLA by using conventional technique Simulation *A* as a control**. If the proposed technique can demonstrate that its generated structures are similar to those generated by the conventional Simulation *A*, then potentially it can be used as a high performance substitution.

DLA is a stochastic process, and therefore no two sufficiently large aggregates will be perfectly equivalent. However, there are several topological metrics of interest that will converge for any realistic DLA-generated structures, such as a fractal dimension of approximately 1.71[5]. This report selects the topological metric of *anisotropy*, the property of how much aggregates prefer to acccumulate mass in some directions rather than others, to use for comparison. In nature, without external influence, DLA has no preference for direction, in principle. So in this report, I quantify and compare the anisotropy of each Simulation with the aim of evaluating if *B*'s anisotropy is sufficiently small or comparable to *A*'s and so may be used as a high performance substitution for *A*. This report will also present figures of the aggregates generated by each Simulation to compare them qualitatively.

The advantages of *B*, in principle, are that:

- it is a formulation of the time evolution of many particles, described by a linear system of many equations. So the optimization "Successive Over Relaxation" may be applied[10].
- the simulation is formally a cellular automata[6]. This means by definition that the time and space evolution of a point only depend on the value of adjacent points in time and space, so parallel processing may be applied.

The purpose of this report, though, is not to measure optimal speed of execution, although the below *Future Work* section will suggest this line of research. This work instead will first establish if the aggregates generated by *B* are similar to those of *A* in terms of anisotropy, as well as if figures of sample aggregates qualitatively appear visually similar. I hypothesize that, due to *B*'s square lattice, anisotropic artifacts will be notable[3]. If that is indeed the case, I suggest an approach to confront this problem in the *Future Work* section.

Algorithm

Simulation A

Einstein has written many famous formula, including one that demonstrated the existence of atoms[2]:

$$P=\frac{e^{-s^2/(4Dt)}}{2\sqrt{\pi Dt}}$$

This expresses that the probability P that a particle under Brownian motion will move a given distance s, in each direction, in a time interval t is governed by a Gaussian distribution. Assume arbitrary units so the diffusion coefficient $D=1$.

It would be inefficient, not to mention inacurrate, to numerically integrate the Gaussian across every polar coordinate. Fortunately, we can analytically aggregate the Gaussians for the x and y coordinates using statistical theory. This is done easily with the observation that a linear combination of Gaussian random variables is itself a Gaussian random variable[4]. So, the path of a particle in simulation *B* will be the sum of succesive computations of two random variables, one for each independent *x* and *y* axes for the particle to move:

$$x = \int_{-1}^{1} G_1(x)\cos x\,dx \qquad y = \int_{-1}^{1} G_2(y)\sin x\,dx$$

Where G_1 and G_2 are Gaussian random variables. Therefore x and y are themselves Gaussian random variables. This fact offers a strategy to simulate *A*. And, this strategy is founded in Brownian statistical theory.

The procedure for *A* is well known. Here it is as described by both Landau[6], chapters 5 and 13, and Wirtz[3]:

1. Start with a 2-D lattice containing a seed particle in the middle
2. Draw a circle around the particle, and place another particle on the cirumference of the circle at some random angle.
3. Release the second particle and have it execute a random walk but restricted to vertical or horizontal jumps between lattice sites. In accordance with the reasoning above, let the length of each step vary according to a random Gaussian distribution.
    - As an optimization, the particle moves in larger steps farther away.

4a. If at some point during its random walk, the particle encounters another particle within one lattice spacing, they stick together and the walk terminates.

- This occurs with *sticking probability* ρ .

4b. If instead the particle passes outside a radius twice that of the circle from which it was released, it is lost forever.

6. Repeat steps 2-4b as much as desired.

Simulation B

State transitions in cellular automata only depend on local states. That is, a cell's transition depends only on the state of itself and its adjacent neighbors; a particle that travels one one side of the aggregate likely will never interact with a particle that travels on the other side at the same time. So it is likely that most particles' paths will have no effect on one another, and can be computed simultaneously. The particles that cannot can just be ignored.

To achieve simultaneous simulation of every particle's random path in the system, we formulate a state transition for diffusion. Fortunately, the diffusion equation allows us to derive a cellular automata[8]. The diffusion equation in two dimensions is:

$$\frac{\partial U}{\partial t} = D\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) \quad [8]$$

Discretizing and setting $D = 1$, it becomes:

$$\frac{u_{i,j}^{(n+1)} - u_{i,j}^{(n)}}{\Delta t} = D\left[\frac{u_{i+1,j}^{(n)} - 2u_{i,j}^{(n)} + u_{i-1,j}^{(n)}}{\Delta x^2} + \frac{u_{i+1,j}^{(n)} - 2u_{i,j}^{(n)} + u_{i-1,j}^{(n)}}{\Delta y^2}\right]$$

Solving for $u_{i,j}^{(n+1)}$ yields

$$u_{i,j}^{(n+1)} = u_{i,j}^{(n)} + D\Delta t\left[\frac{u_{i+1,j}^{(n)} - 2u_{i,j}^{(n)} + u_{i-1,j}^{(n)}}{\Delta x^2} + \frac{u_{i+1,j}^{(n)} - 2u_{i,j}^{(n)} + u_{i-1,j}^{(n)}}{\Delta y^2}\right]$$

For any $[u_{i,j}^{(n)}]$ initial state matrix, the above equation will govern its evolution through time and space, given the stability constraint $\Delta t < \frac{(\Delta x \Delta y)^2}{\Delta x^2 + \Delta y^2}$ [8]. Stability may be enhanced with the "Alternating Direction Implicit Method" technique[11].

However, each $u_{i,j}^{(n)}$ doesn't represent where particles are located; only the probability that a particle is located at that given cell. To form an aggregate, the particles must be localized to check whether they are truly close enough to stick to the aggregate. So in this algorithm, we introduce a *collapse* periodically, every k_t timesteps. A collapse evaluates and updates every $u_{i,j}^{(n)}$ as follows:

$$collapse(u_{i,j}^{(n)}) = \begin{cases} 1 & with\, u_{i,j}^{(n)}\, probability \\ 0 & o.w. \end{cases}$$

For the parameter $k_t = 1$, this algorithm degenerates into precisely a random walk in square cellular automata. $k_t > 1$ generalizes the process to allow optimizations to be applied. When k_t is too large, particles will too easily 'dodge' the aggregate in a way that may subvert the desired fractal structure, and be unrealistic. *B* offers one strategy to get around this, as step 5c; other potential strategies that may get around this issue are offered in the *Future Work* section.

The proposed algorithm is as follows:

1. Create a large odd matrix of 0's, the *diffusion matrix*. Pepper 1's randomly in a ring away from the center, the *spawning ring*.
2. Create the *aggregate set*, a structure to record the points in the aggregate.
3. For k_t steps, evolve the *diffusion matrix* in time according to the diffusion equation described above, applying aforementioned optimizations as desired.
4. Collapse the diffusion matrix. All cells of the matrix are now either 0 or 1.

5a. If a particle with no neighbors is within one lattice spacing of an aggregate particle, for a sticking probability ρ, attach the particle to the aggregate—meaning, subtract it from the diffusion matrix and add it to the aggregate set.

5b. If a particle instead is in the same cell as an aggregate particle, remove it and respawn it in the spawning ring.

5c. If the particle is adjacent to an aggregate particle, but it has direct neighbors (in legal positions), respawn it and its neighbors without appending the aggregate matrix.

- This helps ensure the aggregate particles attach in a realistic order, unbiased by coordinate plane orientation.

6. Repeat steps 3-5c as much as desired.

Measuring Anisotropy

Anisotropy is the tendency of the process to prefer certain directions over others. For instance, if the aggregate is a straight line, the anisotropy will be large. If instead the aggregate is a perfect circle, then the anisotropy should be 0, as mass is distributed equally in all directions. This latter example illustrates why, in the algorithm below, we choose the center of mass of the aggregate as the origin of the coordinate system—all rings are equally anisotropic, no matter which particle on the ring is the seed particle.

Of course, all DLA simulations are expected to distribute mass sharply unequally. Recall from the *Introduction* section that longer tendrils attract more mass, thus making the tendril even longer attracting yet more mass, et cetera. What is of interest is if a Simulation statistically demonstrates that it prefers certain directions across many trials. Both Simulations are square cellular automata, so Wirtz[3] notes that this means they are expected to exhibit anisotropic artifacts related to the shape of the lattice.

To measure the anisotropy of the resulting aggregate,

1. Install a polar coordinate space, centered not on the seed but on the center of mass of the aggregate.
2. Discretize space angularly into n arcs of length $\Delta\theta$, and radially into rings, each a distance Δr from its neighboring rings. The entire space between angles θ_i and $\theta_i + \theta_{i+1}$ is *angular section* i, and the space between ring j and ring $j+1$ is *radial section* j. Ring 0 is the origin and angle $n+1$ is the same as angle n.

3. Measure the *polar density*◊ $\frac{\Delta a_{i,j}}{\Delta A_{i,j}}$ for all i, j, where $\Delta a_{i,j}$ is the mass of the aggregated material inside each section of area $\Delta A_{i,j}$ (each particle has one unit of mass).

4. Compute the anisotropy, as described below.

ϕ_i is the sum of the polar densities in angular section i. Φ_i is the sum of polar densities for the same angular section for a large number of generated aggregates. For the purposes of this report, the *anisotropy* of the structure is quantified as the variance of the normalized set of data $normalize(\{\Phi_i | for\ all\ i \in 0..n\})$.

Results

"*I always thought the spontaneous emergence of self-organization ought to be part of physics*"
--Doyne Farmer[1], *Chaos*

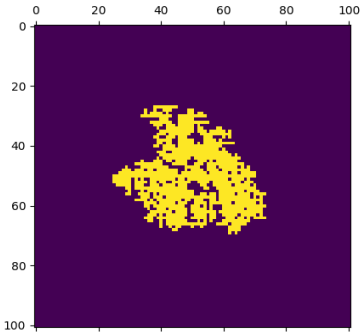

Figure 1a. An example aggregate generated from Simulation A.

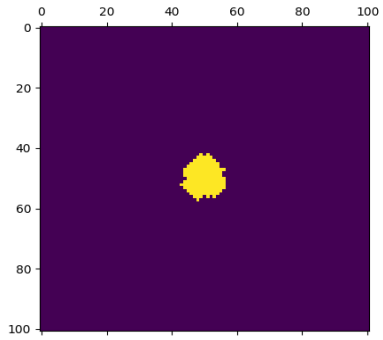

Figure 1b. An example aggregate generated from Simulation B. $k_t = 5$

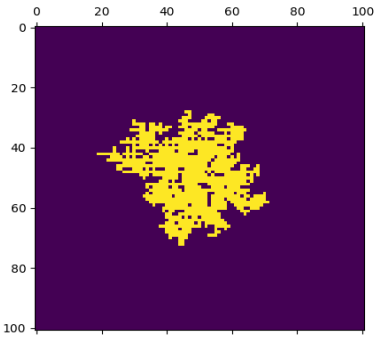

Figure 2a. An example aggregate generated from Simulation A.

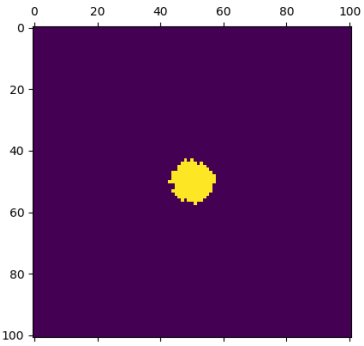

Figure 2b. An example aggregate generated from Simulation B. $k_t = 1$

◊ We use polar density since the area of the polar increases with the square of the radius, thus we expect to find a larger Δa farther away from the center; but speaking to geometric intution, we consider a star with five arms of radius r_α 'equally as anisotropic' as a star of five arms of radius $2r_\alpha$ —mass is an extrinsic property, but isotropy should be intrinsic. *S*o we don't want this property of our coordinate system to influence our calculations.

	Trial A	Trial B	Trial C
Simulation A	1.50137e-06	1.39166e-06	1.36680e-06
Simulation B	1.40432e-06	1.17168e-06	1.31063e-06

Table 1. Anisotropy derived from 100 generated aggregates. Three Trials were performed to yield three values of anisotropy. $k_t = 1$.

Table 1 records the variance of the normalized sum of polar densities taken across 100 generated aggregates per method per trial. The anisotropy between trials within a method does not appear to vary, nor does it between methods, contesting Wirtz' observation. However, we observe that both area and fractal dimension qualitatively differ greatly between Trials. This is regardless of the k_t parameter, so the implementation of Simulation *B* suspiciously does not degenerate to a Gaussian random walk as theorized.

Example aggregates are illustrated in the Figures above. At this resolution, they do not appear to qualitatively vary. Most aggregates pick a preference for some direction or another, which makes sense as that is observed in, e.g., corals, physically; but it seems that there is not much system bias overall for either Simulation. Regardless, it is clear that Simulation *B* fails to exhibit stochastic geometry such as that which emerges from Simulation *A*.

Conclusion

This report is interested in whether Simulation *B*, the proposed technique, can be a drop-in replacement for Landau's traditional Simulation *A*[6], and whether both can be considered to be a realistic method for Diffusion Limited Aggregation. This study is primarily concerned with the anisotropy of each technique, useful for the simulation of, perhaps, lightning strikes, or shale oil hydrolic fracturing. In these situations, the direction-dependent effect of electrostatic attraction or gravity, respectively, on the dispersion of material needs to be understood. If there are anisotropic artifacts in a simulation, then these artifacts will obstruct its physical accuracy.

The results above demonstrate that the Simulations do not have anisotropic artifacts. Despite this fact, in regards with the figures above, clearly Simulation *B*, per my description, has proven to be an effectively different process than *A*. In other words, *B* may not be a drop-in substitute of *A*. Of course, the lines of research listed in the Future Work section ought to be investigated to ultimately determine the ultimate viability and usefulness of a technique like Simulation *B*.

Future work

The hypothesis that Simulation *B* with a parameter $k_t=1$ degenerates to Simulation *A* proved to be incorrect. Assuming another technique is found that corrects this discrepency, additional work could:

- measure the fractal dimension of each Simulation. To do so, the box counting method[6] could be applied.
- apply additional optimizations such as parallel processing and evaluate runtime speeds.
- tweak the lattice resolution and particle sizes to see if artifacts emerge at a large scale.
- change the shape of the particle source, which could be useful to achieve deliberately anisotropic shapes such as coral. *B* offers plenty of flexibility to do so.
- discover the optimal balance of the k_t parameter that realizes DLA while maximizing performance.
- develop alternative techniques that allows for a high k_t parameter but still sorts collisions in a statistically realistic order. One could try adopting a hybrid-k_t strategy that enables particles far from the aggregate to behave as if k_t is high, while particles close to the aggregate behave as if k_t is low.

These problems need to be researched to complete this line of inquiry.

Bibliography

[1] Gleick, James. Chaos: Making a New Science. The Folio Society, 2015.
[2] “Brownian Motion.” Encyclopædia Britannica, Encyclopædia Britannica, Inc., www.britannica.com/science/Brownian-motion.
[3] Wirtz, Franz-Josef. “Diffusion-Limited Aggregation and Its Simulation.” Mar. 2010, www.gut-wirtz.de/dla/.
[4] Taboga, Marco. *Linear Combinations of Normal Random Variables*. www.statlect.com/probability-distributions/normal-distribution-linear-combinations.
[5] Halsey, Thomas C. “DIFFUSION-LIMITED AGGREGATION: A MODEL FOR PATTERN FORMATION.” Physicstoday.com, 2000, physicstoday.scitation.org/doi/pdf/10.1063/1.1333284.
[6] Landau, Rubin H. *A Survey of Computational Physics: Introductory Computational Science*. Princeton University Press, 2008.
[7] Anagnostopoulos, Konstantinos N. *Computational Physics: a Practical Introduction to Computational Physics and Scientific Computing (Using C++)*. National Technical University of Athens, 2016.
[8] *The Two-Dimensional Diffusion Equation*, Scipython, 2016, scipython.com/book/chapter-7-matplotlib/examples/the-two-dimensional-diffusion-equation/.
[9] Vinals, Jorge. *The Crank-Nicholson Algorithm for Parabolic Partial Differential Equations*. Github, 30 Oct. 2020, github.umn.edu/vinals/PHYS4041-2020.
[10] “Successive over-Relaxation.” *Wikipedia*, Wikimedia Foundation, 6 Aug. 2020, en.wikipedia.org/wiki/Successive_over-relaxation.
[11] “Alternating Direction Implicit Method.” *Wikipedia*, Wikimedia Foundation, 29 Aug. 2020, en.wikipedia.org/wiki/Alternating_direction_implicit_method.