PHYS486-HW5: Stochastic Processes

Nicholas Cemenenkoff

May 25, 2018

Abstract

We use a 2D diffusion-limited aggregation (DLA) random walk model to simulate how snow or soot particles form clusters as they grow. Qualitatively, how the general characteristics of separately initialized clusters compare and contrast is explored, but more quantitatively, a quick-and-dirty fractal dimensionality of each cluster is calculated via linear regression. Through several numerical trials, our results show that clusters grown according to our random walk implementation have fractal dimensions ranging between $0.9 \leq d_f \leq 2$.

1 Problem Statement

Characterize the structure and fractal dimensionality of 2D DLA clusters grown with walkers that begin their walks from a preferred direction (i.e., the walkers begin from lattice points along the x-axis). Contrast this with walkers that begin their walk from no preferred direction, meaning the walkers spawn randomly from a circle encompassing the cluster.

2 Numerical Solution

For purposes of visualization, we think of the growing DLA clusters in this implementation as "soot clumps". There is a seed soot particle and we are moving along with it in its frame, so it appears stationary. Randomly moving smaller surrounding soot particles will eventually bump into the seed particle and stick (causing the clump to grow). The actual code was written with this specific image in mind, hence `gen_clump()` is the name of our code's main function.

Our solution involves a 2D square grid of size n with $20n$ random walkers, each of which can take a maximum number of floor $\left(\sqrt{2}n\right)$ steps. There is a seed cluster state located on the grid represented by a 1. It's up-, down-, left-, and right-neighbor are considered perimeter states and are labeled with a 2. Empty states are labeled with 0s. As the cluster grows, perimeter states change from 2 to 1 when walkers hit the cluster, and after that the perimeter is appropriately expanded by changing up to three of its immediate neighbors from 0s to 2s. The walker spawn type can either be `circle` or `line`, and this represents a 1D range within which new walkers spawn randomly. As the cluster grows, so too does the spawn range so as not to overlap the cluster borders and possibly spawn a walker immediately on the cluster. The starting cluster seed is either centered in the grid for the circular spawn, or it is placed in at the top-middle of the grid for the linear spawn.

Our implementation involves three workhorse functions:

1. `gen_clump()`,
2. `check_in()`, and
3. `find_df()`.

`gen_clump()` performs a diffusion-limited aggregation random walk to simulate how a cluster grows when cluster particles spawn from a preferred direction (`line`) or no preferred direction (`circle`). In both options, particles spawn from a random location on the spawn range. The `line` option has the spawn range just below the cluster so as not to spawn particles directly on top of the cluster. As the cluster grows, the spawn range widens and moves to stay below the cluster dynamically. The `circle` option makes the spawn range a circle that encloses the cluster (without touching it) that expands while the cluster grows. After a walker spawns, a random number r is then drawn from a uniform distribution over $[0, 1)$. If $r \in [0, 0.25)$, the walker steps left, and if $r \in [0.25, 0.5)$, the walker steps right. Similarly, if $r \in [0.5, 0.75)$, the walker steps down. If $r \in [0.75, 1.0)$, the walker steps up. After its step, if the current walker is within the grid and is on a perimeter state, the cluster grows (i.e. a 2 changes to a 1 and up to three 0s change to 2s) and a new walker is started, otherwise the walker keeps walking.

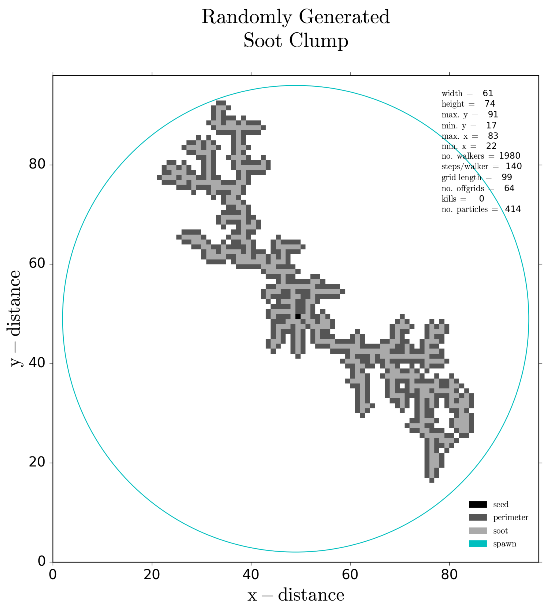

Figure 1: 2D DLA cluster grown with a circular spawn range.

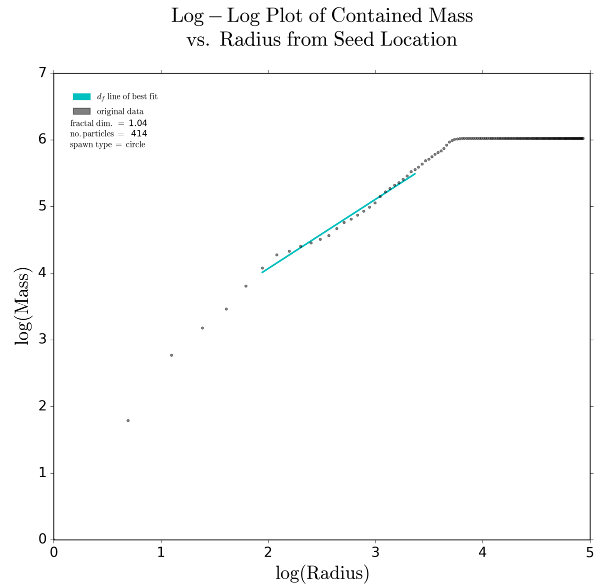

Figure 2: Here we calculate the fractal dimensionality of the circularly grown DLA cluster in Figure 1. Since $d_f = 1.04$, the flake is slightly two-dimensional, and the size of the cracks of the snowflake increase with r.

After the walker either hits the cluster or walks the maximum number of steps, the next walker is sent, and it is in this fashion that the algorithm incrementally generates a cluster. Figure 7 and Figure 8 give a gray scale heat map of walker spawn positions in the `circle` and `line` cases respectively.

Next, `check_in()` checks if a point (x, y) lies within a circle of radius r centered at (h, k) and is a supplementary function within `find_df()`.

Assuming all of a cluster's particles have the same mass, `find_df()` creates successively larger rings emanating from the center of a cluster and counts the number of cluster particles within each ring with `check_in()`. These functions employed in tandem allow us to create a $\log(m)$ vs $\log(r)$ plot wherein we may perform linear regression over a reasonable range to infer how the enclosed mass of a cluster grows with radius from the seed location if the cluster were infinitely large. In other words, `find_df()` uses a linear regression model to estimate the cluster's fractal dimensionality. To understand these results in rough general terms, a curve with fractal dimension very near to 1 (say 1.04) behaves quite like an ordinary line, but a curve with fractal dimension 1.94 winds complicatedly through space much like a surface. Similarly, a surface with fractal dimension of 2.04 occupies space very much like an ordinary surface, but one with a fractal dimension of 2.94 folds and flows to fill space rather nearly like a volume. Note lastly that important parameters like cluster height, width, and number of particles are all saved

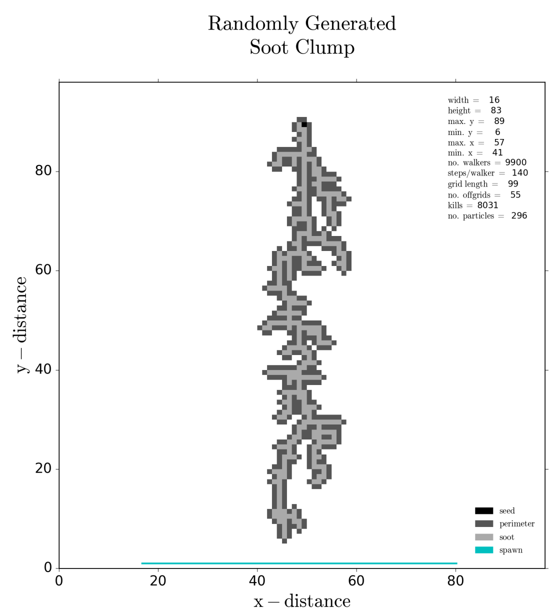

Figure 3: A linearly grown DLA cluster.

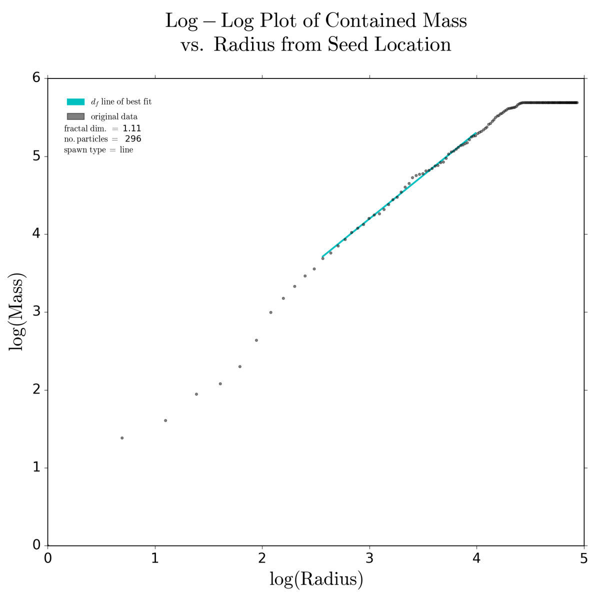

Figure 4: Note the fractal dimension of the linearly grown cluster is 1.11 which is slightly higher than the circularly grown cluster.

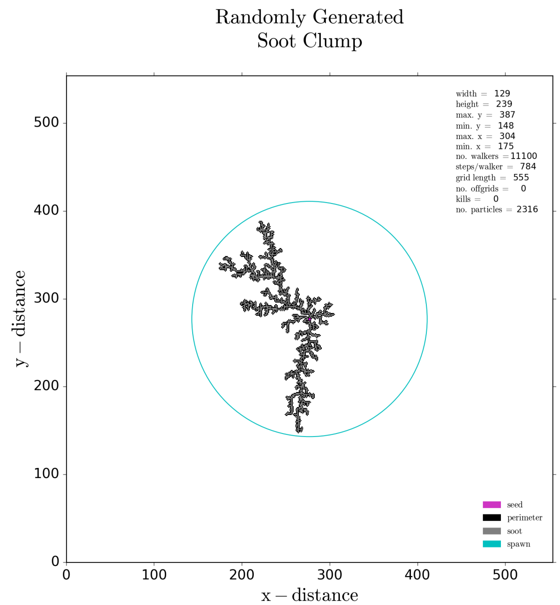

Figure 5: A circularly grown DLA cluster grown on a 555x555 grid.

for separate runs as they provide some quantitative data associated with the physical properties of each cluster. After many runs, it would be possible to investigate these physical characteristics on average, but that is beyond the scope of this paper.

3 Physical Analysis

This random walk scenario is an example of what physicists call Brownian motion. Though each run is unique, it is evident that the linear spawn range causes the clusters to grow more like "icicles", whereas the circular spawn range makes them more like "snowflakes". Initially, since a snowflake seems more two-dimensional than an icicle, we expected the fractal dimension of the cluster resulting from the circular spawn to be larger than the fractal dimension of the linearly generated cluster, but that is not what the data shows. To quote Giordano, "in order for a cluster to have an effective dimensionality d_f, which is not an integer, its mass must increase more slowly than r^2. This means that it must contain certain holes or cracks... [but] the sizes of these open spaces must *increase* with r." It seems then, we may conclude that the cracks in the linearly generated icicle cluster increase faster with r than the snowflake cracks do (a possibility for future research would be to investigate the fractal dimensionality of wisecracks). Interestingly, after many runs, larger circularly grown clusters were found to have close to the same dimension ($d_f = 1.04$) as the circularly grown cluster in Figure 1. One of them is shown in Figure 5. Do their cracks have a similar character? Qualitatively, due to the size difference, it is hard to say, but despite what our intuitions may think, the elementary physical analysis here indicates that perhaps the cracks of circularly grown clusters have the same character no matter the size of the cluster. Does this same character hold for the linearly generated clusters? It seems so according to several numerical trials, but it is evident that there is significant variability in their measured d_f values. Will this difference in fractal dimension between circularly and linearly grown clusters be consistent even for infinitely large clusters? We don't think so, even if it might be a stretch to claim. Given an infinitely large cluster (in all directions), it'd be impossible to understand from which direction most of the growth occurred. Assuming that our numerical implementation of a growing spawn range does not provide any significant bias to the nature of the cluster as it begins to grow, we believe that both the circularly grown and linearly grown DLA clusters that are infinitely large should have the same fractal dimensionality.

4 Error Analysis

Due to the random nature of this problem, most of the numerical error analysis we conduct involves corroborating the random nature of our method of

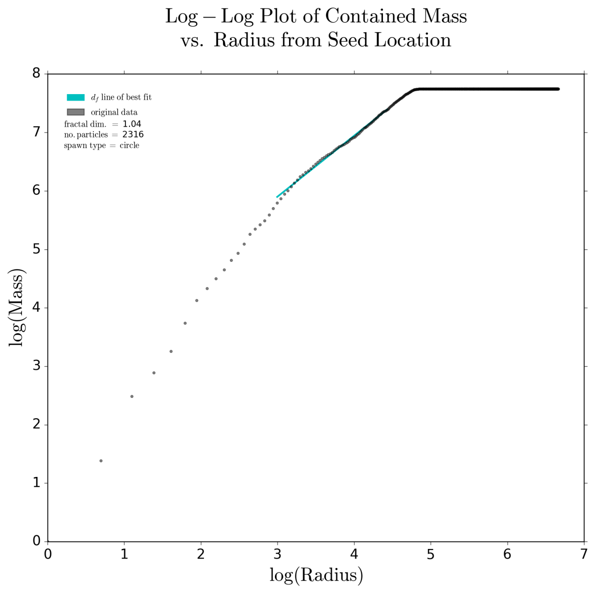

Figure 6: Note the fractal dimension of this larger circularly grown cluster is 1.04 which is the same as the fractal dimension of the cluster found in Figure 1.

cluster generation. Beyond debugging flow control in `gen_clump()`, we find the easiest way to show the random nature of the growing clump is to track the spawn positions of each walker. Figure 7 and and Figure 8 both show a roughly uniform random sampling over successive 1D circular and linear spawn regions respectively. Because Figure 7 is tracking the spawn positions along several nested circles, it maps out a uniformly sampled disk. Figure 8 maps out a more complicated shape, but it is still evident that the spawn range increases monotonically and that the area shown is roughly uniformly sampled.

5 Conclusions

Diffusion-limited aggregation random walks provide the means of easily implementing computational models for clusters that physicists believe grow according to Browning motion. The structures that result from our model exhibit a fractal-like nature, with linearly grown clusters generally having a larger fractal dimensionality than circularly grown clusters, though this is only a preliminary conclusion. Given the sampling in our numerical implementation is random and uniform, we can effectively conclude that our method lays down a robust baseline for further DLA cluster research.

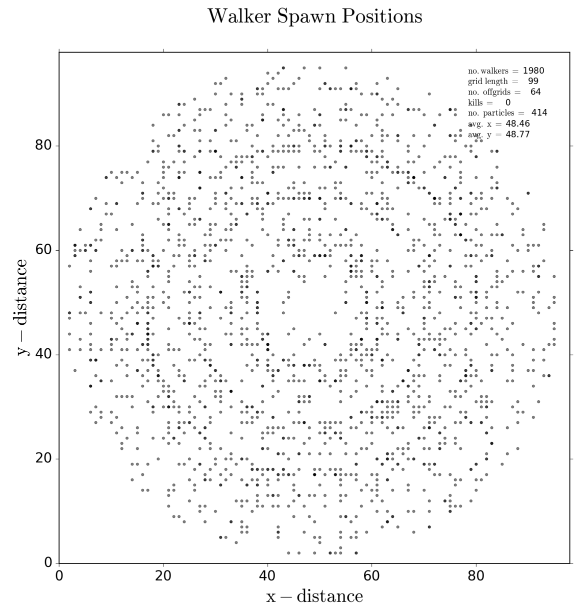

Figure 7: Walker spawn positions for a Figure 1. Note how the area enclosed by the growing spawn range is close to a uniformly randomly sampled disk. This helps to corroborate the random nature of the particle spawn range.

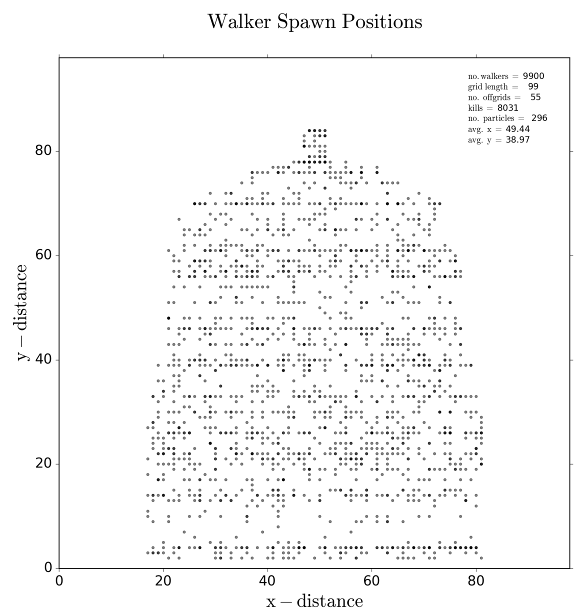

Figure 8: Walker spawn positions for the linearly grown DLA cluster found in Figure 4. Interestingly, the shape mapped out looks similar to a bell jar.