Homework 4: Diffusion-Limited Aggregation and Fractals

Kory Beach
Department of Physics, Astronomy, and Applied Physics,
Rensselaer Polytechnic Institute
beachk2@rpi.edu

December 11, 2016

1.1 Introduction

Fractals are a ubiquitous phenomenon in nature, mathematics, physics, and even art. The universality with which they arise in these disparate areas is indicative of the remarkable simplicity that underlies their seemingly complex appearance. At their core, all fractals exhibit some kind of self-similarity in structure, and they form naturally (such as in flowers and snowflakes) by means of a very simple set of rules. These underlying rules, and thus the patterns that they form, vary from source to source, but in all fractals they can be studied by means of the fractal dimension, which is a measure of how the complexity of the pattern changes with respect to the scale at which it is measured.

In this report we will discuss the simulation of the growth of fractals by means of a process known as diffusion-limited aggregation (DLA). DLA relies entirely upon the random motion of a particle until it reaches a nucleation site. By successively introducing random-walking particles into such a system, one can form fractals like those grown in crystals in an electrodeposition experiment. We will discuss the implementation of a DLA simulation and analyze the fractal dimensions of the objects we form from this method.

Figure 1.1: Images of fractals formed by DLA in a copper deposition reaction. From [2].

1.2 Methods

1.2.1 The DLA algorithm

The basic algorithm for simulating DLA is quite simple. Starting with an N x N grid of points, i.e. a matrix composed of boolean elements, we can adopt a representation in which elements in the matrix will be 0 if a that location is unoccupied by a particle and 1 if it is occupied. Starting with some configuration of occupied points in the box, meant to represent nucleation sites for the fractals, we then introduce particles one by one from the edge of the box into the system.

For every particle the algorithm undergoes the following procedure:

1. It initializes a new particle on the border of the box.

2. It checks to see if a neighboring cell is occupied by another particle.

   If a neighboring cell is occupied, the particle stops moving and a new particle is initialized.

   Otherwise, the particle moves to a random neighboring cell.

3. Back to step 2.

After the desired number of particles have been randomly diffused into the system, a simple loop to tally up the x- and y-coordinates of each of the occupied matrix elements can be used to plot the final positions of the particles in a Cartesian graph.

1.2.2 The Fractal Dimension

Once a fractal pattern has been "grown" by means of the DLA algorithm, we can analyze it for its fractal dimension. The fractal dimension is a ratio that gives an indication of how the complexity of the pattern changes when the scale at which it is measured changes. In other words, it is a measure of how self-similar or self-dissimilar the pattern is. If it is highly self-similar, it will exhibit the same statistical properties at large scales as at very small scales. If it is self-dissimilar, then the shapes we see at large scales will not be comparable to those at very small scales.

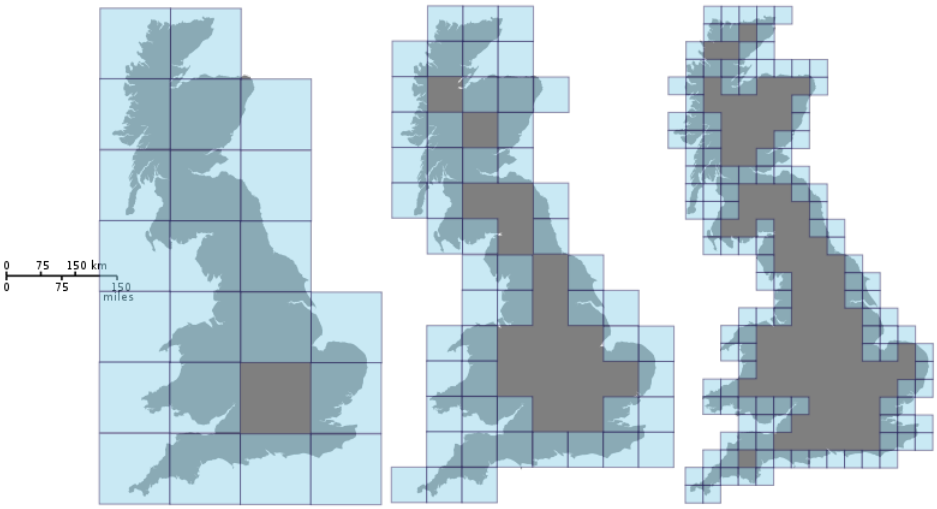

Figure 1.2: Application of the box counting method on an image of Great Britain. Box scale is successively decreased, capturing smaller scale complexity. From [3].

A useful method for measuring this property in physical patterns is the Minkowski–Bouligand box counting method.[3] If we divide an image with a

fractal pattern into a grid of boxes of dimension ε x ε, as shown in Figure 1.2, we can tally up the number of boxes that cover the fractal set, which in the example shown in the Figure means the number of boxes that contain some part of the coastline. If we define $N(\varepsilon)$ to be the number of boxes covering the fractal for a given box size ϵ, the fractal dimension is given by:

$$D(S) = \frac{\log N(\varepsilon)}{\log(1/\varepsilon)} \tag{1.1}$$

where S is the fractal set in the space $\mathbf{R}^n$.

In practice, D can be obtained for any image containing a fractal or fractal-like pattern. By successively tallying the number of boxes containing more than some predefined threshold of colored pixels for different size boxes, it is easy to determine the fractal dimension of any image.

1.2.3 Algorithmic Considerations

While the fundamentals of the DLA algorithm are quite straightforward, the main challenges that emerge when actually implementing this procedure are matters of scale and optimization. One constraint that must be considered is the fact that, in order to simulate isotropic diffusion, the particles need to start far enough away from the nucleation site(s) that no preferred direction can be seen in the resulting fractal pattern.

It follows, then, that the box needs to be made sufficiently large in order for this condition to be met. However, although we can expect the algorithm to work quickly for a relatively small box, as the matrix size grows we can expect the calculation time to increase by at least $\mathcal{O}(N^2)$ where N x N is the size of the matrix. As such, it is important to take into careful account the parts in our algorithm which are likely to cause significant slowdown.

One source of potential slowdown is in the generation of random numbers within the loop. Whenever the particle needs to select an adjacent box to move to, a new random number is needed. In this implementation, we have allowed for Horizontal, Vertical, and Diagonal (HVD) motion by randomly selecting between -1, 0, and 1, and then adding those numbers to the current x- and y-coordinates separately. To account for falling off the edge we also include a periodic boundary condition in the following manner:

```
x += q;
x -= N*floor(x / N);
y += r;
y -= N*floor(y / N);
```

where **N** is the box dimension and **q** and **r** are randomly selected from {-1, 0, 1}. The generation of these random numbers is potentially expensive if it is performed every time the particle moves. It is therefore beneficial to pre-generate the random numbers in an array of sufficient size (1 million elements works well) and then iterate pseudo-randomly through the array whenever a random number is needed.

Another useful technique for speeding up the simulation is to allow particles to move more than one space when they are sufficiently far away from the nucleation site. While such a technique may undermine the isotropy of diffusion from far distances, this effect can be countered by also randomizing the inital location along the border of the box. By implementing these techniques, the speed of the calculation can be noticeably improved.

1.3 Implementation

This DLA simulation was implemented using C++ on Blue Gene/Q using a g++ compiler. Because the matrix used for this simulation was 512 x 512, particular attention had to be paid, in the interest of optimization, to how data was allocated and retrieved.

The most significant potential for slowdown was in the way the matrix **mat** was called within the loop for assessing and updating the particle location. The difference between calling the matrix or elements of the matrix with **&mat** as opposed to **mat** as inputs to a function cannot be stressed enough. Whereas the former points the function to the memory location of the already stored matrix, the latter copies the entire matrix to a new location, increasing the calculation time by at least a factor of $\mathcal{O}(N^2)$. With the proper dereferencing of variables in function calls the simulations ran smoothly and quickly for large systems.

1.4 Results and Discussion

Two different nucleation site formations were simulated using this implementation. Both simulations involved box sizes of 512 x 512 and 10000 particles. As shown if Figure 1.3, both (a) a single nucleation point at the center of the box and (b) a solid square nucleation site were used. It is clear that both formed fractal patterns, but the shape of the nucleation site leads to changes in the overall structure of the branches. In 1.3a we see that there are six main branches emerging from the center whereas in 1.3b there are only four.

This makes sense intuitively because a particle is more likely to settle on a corner than the center of an edge because the larger surface area of the corner increases the probability of contact. With four corners in the original nucleation site we expect to see four main branches. Once branches start to form, the probability of a particle diffusing around the branch toward a central point becomes even lower; thus, the formation of branches not only increases the probability that successive particles will land on those branches, but also makes unlikely the formation of new branches.

As shown in Figure 1.4, the Minkowski–Bouligand box counting method was then applied to these images. In practice this involved iterating through the final matrix with successively decreasing increment sizes (ε) and tallying up the number of boxes containing particles ($N(\varepsilon)$).

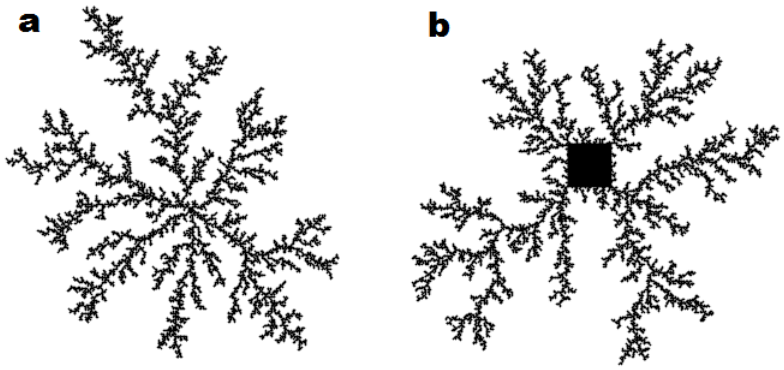

Figure 1.3: DLA fractals with (a) a point nucleation site and (b) a square nucleation site.

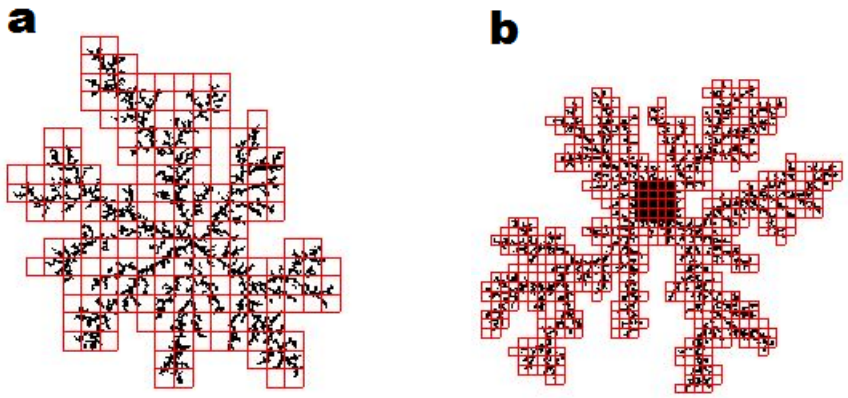

Figure 1.4: Visualization of box counting on fractal patterns.

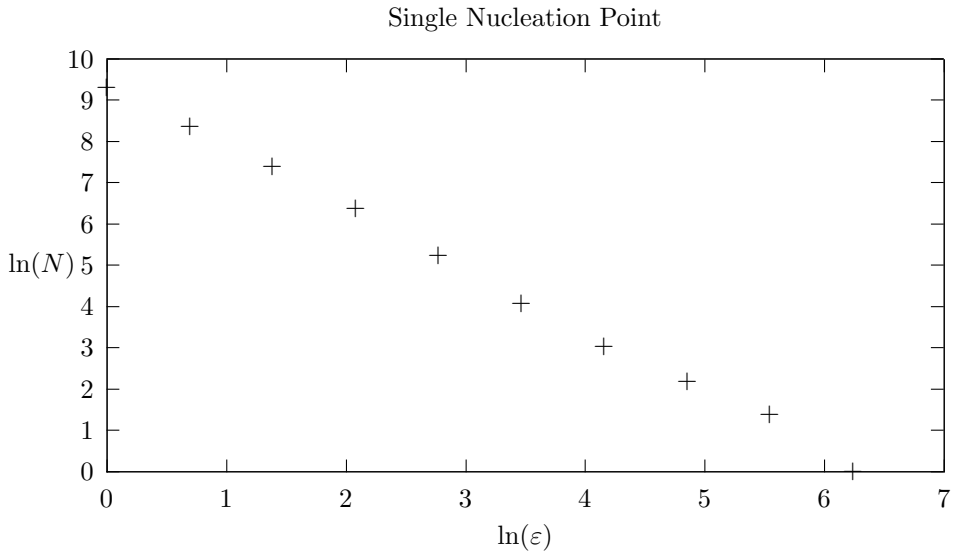

Figure 1.5: Linear fit gives a slope (fractal dimension) of -1.49.

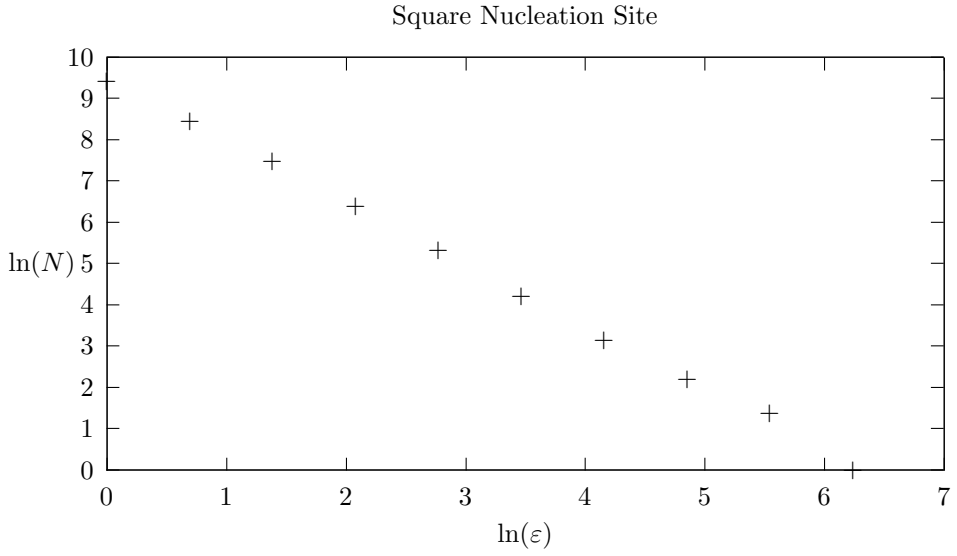

Figure 1.6: Linear fit gives a slope (fractal dimension) of -1.50.

In Figure 1.5 and Figure 1.6 we plot the logarithms of these two parameters for the single nucleation point and the square nucleation site respectively. As is evident in the plots, there is a roughly linear relationship between $\ln(N)$ and $\ln \varepsilon$, meaning that we can use the slope of the line to determine the fractal dimension D. We find that a linear fit for both plots gives a fractal dimension of $D \approx 1.5$, which is fully consistent with what we would expect from such a pattern. For comparison, several well known fractal patterns including the Weierstrass function and the Quadratic von Koch Curve exhibit about the same fractal dimension. [3]

1.5 Conclusion

We have shown that through careful implementation of the DLA algorithm, we can efficiently produce fractals through the purely stochastic process of particle diffusion. This exercise sheds light not only on the underlying mechanics of fractal formation in nature, but also on the rich mathematical principles of self-similarity and fractal dimension upon which it is based. We have also shown that the choice of nucleation site is important for such systems, which when applied to systems in science and nature can have important physical implications.

Bibliography

[1] Meunier, Vincent. Advanced Computational Physics Lecture Slides; Lectures 9 and 10: Fractals

[2] Chishty S. Q, Farooqui M, Khizer M, Rajeshaikh B. B. Fractal Aggregation of Copper Particles using Electroless Cell. Orient J Chem 2013;29(3).

[3] Wikipedia. Artwork by Prokofiev. "Minkowski Bouligand dimension". 2016. https://en.wikipedia.org/wiki/Minkowsk_Bouligand_dimension