Appendix A: Analytical Solution to 2D DBM

We can derive the analytical solution for the initial case of 2D DBM in a manner analogous to the 3D derivation of section 3.1. 2D DBM initially contains a small ring of negative charge of radius $R_1 = \frac{h}{2}$, surrounded by a larger ring of positive charge of radius $R_2 = \frac{Nh}{2}$. The potential between the ring and sphere is then defined by the polar Laplacian:

$$\nabla^2\phi(r,\theta) = \frac{\partial^2\phi}{\partial r^2} + \frac{1}{r}\frac{\partial\phi}{\partial r} + \frac{1}{r^2}\frac{\partial^2\phi}{\partial\theta^2} = 0. \quad (1)$$

The positive ring and negative sphere are the Dirichlet boundary conditions:

$$\phi(R_1,\theta) = 1 \quad (2)$$
$$\phi(R_2,\theta) = 0. \quad (3)$$

We observe that the boundary conditions are isotropic, so we can drop the last term in the polar Laplace, reducing the PDE to an ODE.

$$\frac{d^2\phi}{dr^2} + \frac{1}{r}\frac{d\phi}{dr} = 0. \quad (4)$$

This is an Euler equation whose solution is the 2D Green's function:

$$\phi = c_1 + c_2 \log r. \quad (5)$$

We can solve for c_1 and c_2 in terms of the boundary conditions:

$$c_1 = \frac{-\log R_1}{\log R_2 - \log R_1} \quad (6)$$
$$c_2 = \frac{1}{\log R_2 - \log R_1} \quad (7)$$

Unlike in the 3D case, as $R_2 \to \infty$, the 2D Green's function diverges. As the logarithm function increases, any arbitrarily large value of c_1 will be unable to bound it. This can only be prevented if precisely the same amount of positive and negative charge are present in the simulation. For visual rendering purposes, enforcing this constraint is impractical. In its absence, very far away particles will induce an increasingly large potentials, creating 'action at a distance' that is unpredictable, difficult to control, and clearly non-physical.

An analytical solution for cases other than the initial one is difficult, as the polar symmetry of the differential equation is quickly broken. Conformal mapping techniques can be used to solve the Laplace equation over these more difficult domains, but these techniques are outside the scope of this paper.

Numerical Validation

We will now now numerically verify the 2D equations obtained in the previous section. We set $R_1 = \frac{1}{130}$ and $R_2 = \frac{1}{2}$. The initial grid is set to 65 x 65, so that the central ring is only one grid cell across. As the grid is refined, we should approach the values given by the analytical solution.

Figure 1 shows the behavior of the 2-norm and ∞-norm as the simulation resolution is varied from 65^2 to 2048^2. The slope of the two norm is -1.5241, and the slope of the infinity norm is -1.0828, suggesting first order accuracy of the numerical results. Although a second order convergence may be expected due to the second order accuracy of the Laplacian, the inner and outer rings are both represented by a set of point samples that are only first order accurate. As a result, the first order convergence is unsurprising. Numerical results on a polar grid, on the other hand, would probably show second order convergence.

Even at the 2048^2 resolution, we are still able to only obtain accuracy to three significant digits. Even at this high resolution, the inner ring of $R_1 = \frac{1}{130}$ is still coarsely sampled. As this is also the region where the potential function exhibits the most variation, the coarse representation of 2D DBM introduces significant error.

An equivalent convergence analysis for 3D DBM is difficult due to the steep memory requirements of the algorithm. We can only obtain results up to 256^3, and the resulting data set is too coarse for a useful regression to be performed.

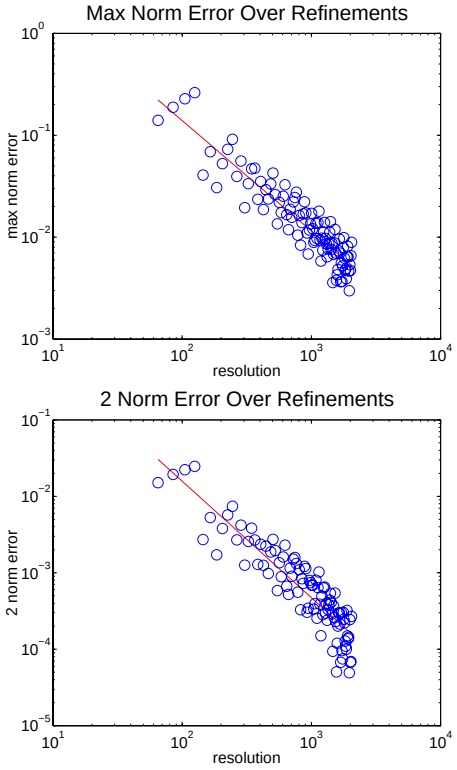

Figure 1: Top: Infinity norm over successive refinements. Bottom: 2 norm over successive refinements. DBM appears to be first order accurate, while the analytical Green's function we use is infinite order accuracy.

Appendix B: Performance for Canonical 2D Case

We simulated 5000 particles on a 256^2 this grid, with $\eta = 1$ for DBM. We set $\eta = 4$ for our algorithm, as this setting gave us results similar to $\eta = 1$ in DBM. Timing results for this setup are given in Figure 2(a). Despite the optimizations to DBM, **Our approach reduced the simulation time by a factor of 5748.** Additionally, since we do not require the allocation of a large uniform grid, **memory consumption was reduced by a factor of 200.**

DLA is far less general than DBM, but excels at generating structures of precisely 1.71D. When generating a 1.71D structure containing 100000 particles, DLA outperformed our algorithm by a factor of 4.23 (Figure 2(b)). However, we would like to stress that DLA *only* excels at generating 1.71D structures, *only* for the canonical configuration. Generating denser structures requires extensive modification to the core algorithm, and the performance quickly degrades. For example, in order to generate a 2D 'Eden cluster' using DLA, we implemented the DLA variant described in [Meakin et al. 1987], which requires the use of several expensive range queries. Over 10000 particles, our algorithm outperformed it by a factor of 34.7 (Figure 2(c)). Overall, we believe our algorithm offers competitive performance with DLA when generating aggregates in the dimensionality range of $2 \leq D \leq 1.71$.

Currently, we know of no modification that allows DLA to generate aggregates of dimension lower than 1.71D. While there exist techniques to generate thinner aggregates, [Nittmann and Stanley 1987] the grid artifacts of the underlying lattice quickly dominate, and comparisons to our algorithm and DBM are no longer apt.

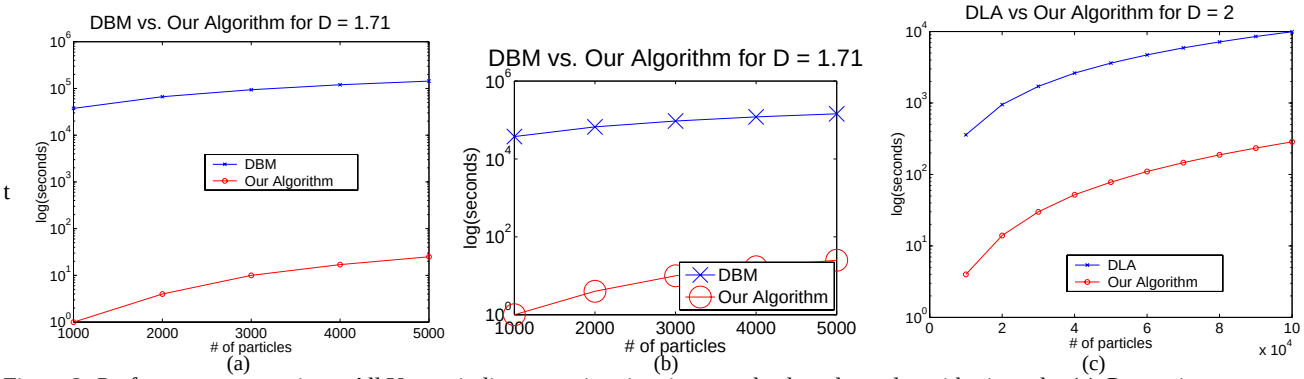

Figure 2: Performance comparison. All Y axes indicate running time in seconds plotted on a logarithmic scale. (a) Generating an aggregate of dimension 1 with 5,000 particles, our algorithm is **5,748** times faster than DBM. (b) Generating an aggregate of dimension 1.71 with 100,000 particles, DLA is 4 times faster than ours. (c) Generating an aggregate of dimension 2 with 100,000 particles, our algorithm is about **35** times faster than DLA.

Appendix C: Visual Validation

In Figures 3(a) and 4(a), we have photographs of an Aspen tree and a Maple tree. To show that our algorithm creates visually plausible trees, we used our algorithm to grow trees with similar characteristics as those shown in the photographs. The images of our simulated trees are shown in Figures 3(b) and 4(b) respectively.

Figure 3: **Left:** Photo of an Aspen tree. **Right:** Rendering of an Aspen-like tree growth simulation. Although our algorithm is not specifically geared towards tree growth, it generates visually plausible results.

Appendix D: Additional Images

In Figure 5- 10, we show a wide variety of fractal patterns generated by our algorithm. Each image was generated in less than 2 minutes.

References

MEAKIN, P., FAMILY, F., , AND VICSEK, T. 1987. Viscous fingering simulated by off-lattice aggregation. *Journal of Colloid and Interface Science 117*, 394–399.

NITTMANN, J., AND STANLEY, H. E. 1987. Non-deterministic approach to anisotropic growth patterns with continuously tunable morphology: the fractal properties of some real snowflakes. *Journal of Physics A 20*, L1185–L1191.

Figure 4: **Top:** Photo of a maple tree. **Bottom:** Rendering of our maple tree scene. The stochastic nature of our algorithm produces a tree that differs in shape but is similar in characteristics; it is particularly good at generating complicated, gnarled branches.

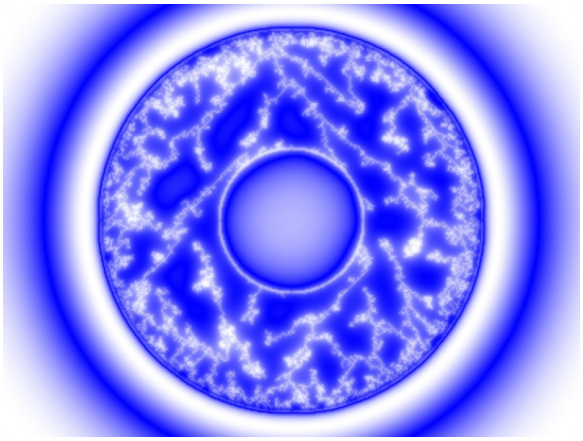

Figure 5: A cluster grown between a ball of negative charge (center) and a ring of positive charge.

Figure 6: A dense cluster grown along an initial cross.

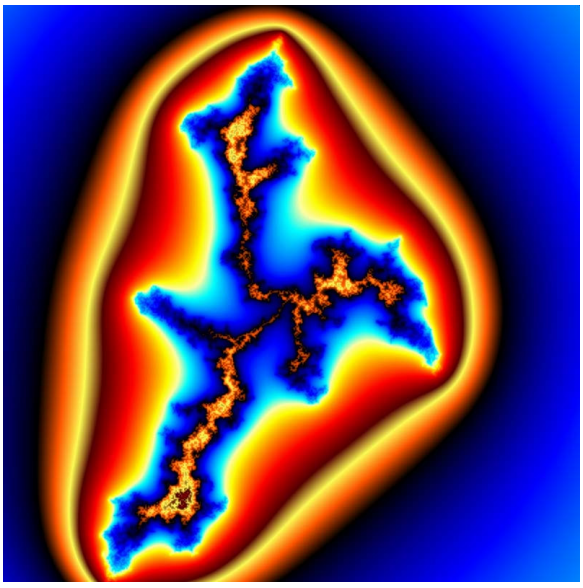

Figure 7: A thin cluster with a rainbow color map.

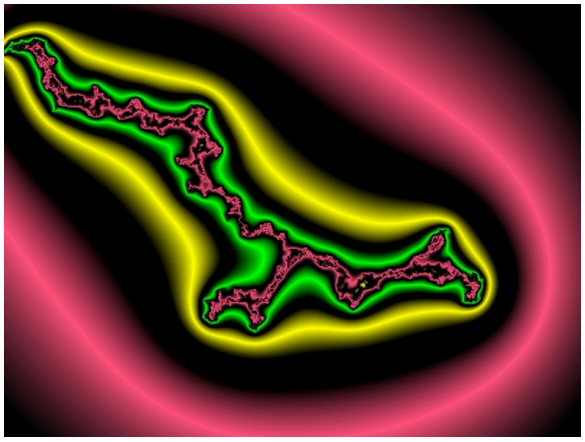

Figure 8: A thin 'fracture-like' cluster that demonstrates the ability of our algorithm to generate fissures.

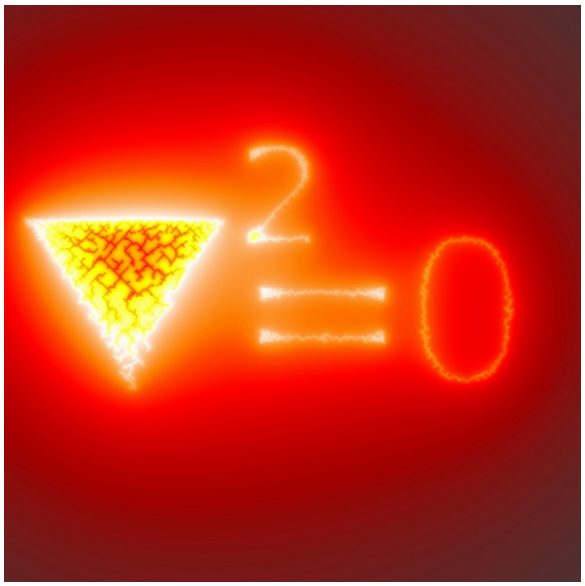

Figure 9: Using point charge configurations derived from simple text, we used our algorithm to grow specific shapes.

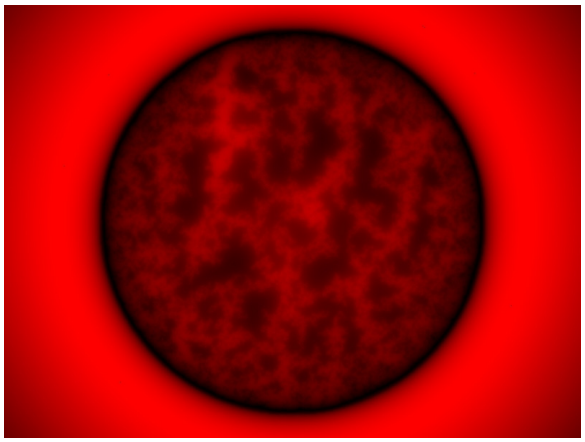

Figure 10: A low η cluster grown with our algorithm

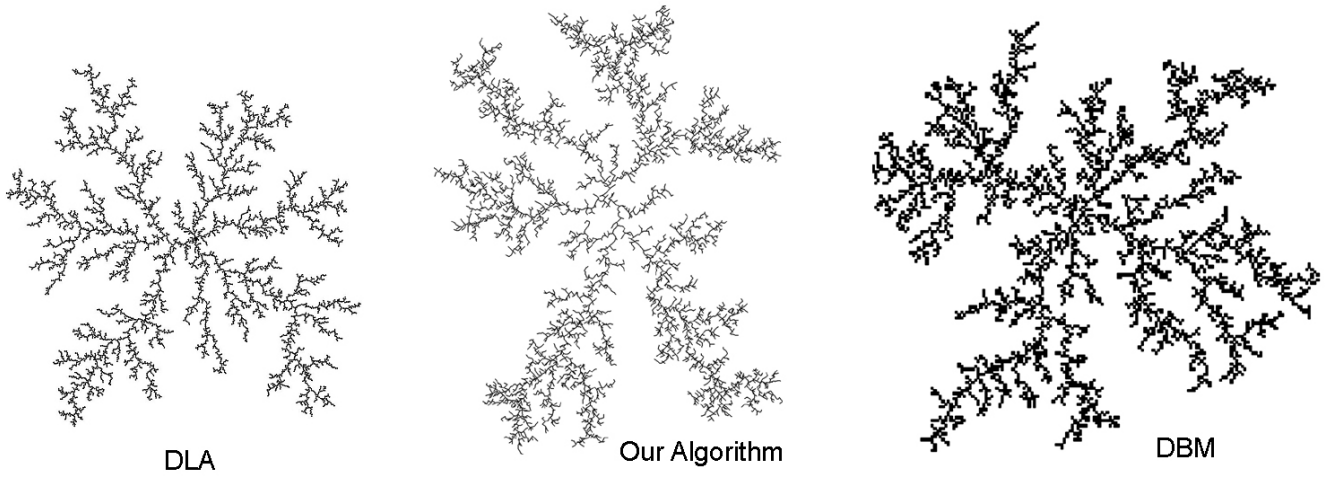

Figure 11: Visual comparison of the results of DLA, DBM, and our algorithm.

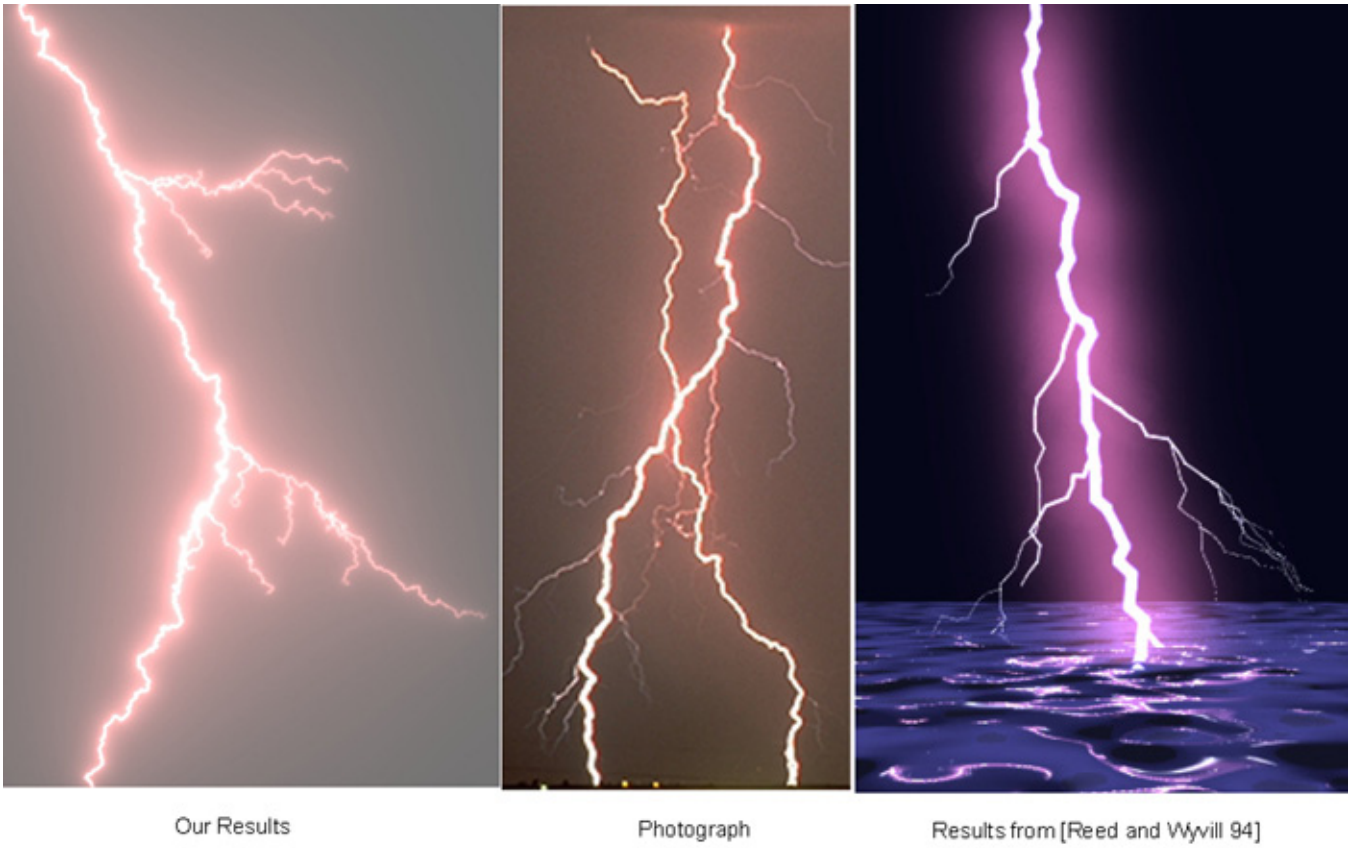

Figure 12: Visual comparison of our lightning with a photograph and the results of Reed and Wyvill.