A Follow-up on the Simulation of Lichtenberg Figures

Simon Chen[1,2] and Shan Gao[1,3]

[1] *Department of Mathematics and Statistics, McGill University, Montreal, Canada*
[2] *Department of Physics, McGill University, Montreal, Canada*
[3] *School of Computer Science, McGill University, Montreal, Canada*

Publication date: August 19, 2023

Abstract—Chen and Gao [1] previously developed a breadth-first search (BFS) algorithm meant to simulate the formation of Lichtenberg figures in an insulating material. In this paper, we present a new algorithm based on the dielectric breakdown model that takes into account the stochastic nature of dielectric breakdown.

Keywords—Lichtenberg figures, Diffusion-limited aggregation, Dielectric breakdown model

I. Introduction

Dielectric breakdown occurs when an insulating material suddenly becomes conductive when subjected to a sufficiently strong electric field. During breakdown, electric discharge patterns with fractal and tree-like structures are produced in or on the surface of the insulating material.

To create real-world Lichtenberg figures, a coat of electrolytic solution is first applied to a piece of wood in order to reduce the resistance on its surface. Two electrodes are then placed on either end of the wood, and a high voltage is passed across them. The current generated from the electrodes will cause the surface of the wood to heat up and burn. Because carbonized wood is mildly conductive, the surface will burn in a pattern travelling outwards from the electrodes.

We were interested in replicating the phenomenon of Lichtenberg figures on a wooden surface due to the presence of natural anisotropies in the material, which often leads to visually interesting patterns. Unfortunately, our previous algorithm was unsuccessful in this attempt due to the unphysical mechanism by which the branches propagated and the unrealistic, although aesthetically interesting, end results. The algorithm presented in this paper corrects these issues.

II. Single-Tree Algorithm

a. Methodology

The purpose of the single-tree algorithm was to improve upon the previous BFS algorithm by taking into account the stochastic nature of dielectric breakdown by implementing a probability function based on values obtained using the discrete Laplace equation.

i. Version 3.0.0

As a warm-up, we follow closely the model outlined in section 3 of Tsonis and Elsner [2]. We first initialize an $m \times n$ grid of zeros with periodic boundary conditions on the left and right sides. The value of the top row is fixed to be zero, and the value of the bottom row is fixed to be one. The electric potential $\phi_{i,j}$ at the (i, j)-th entry is given by the value of the grid at that point.

Given these initial and boundary conditions, we compute the potential at every point on the grid by solving Laplace's equation:

$$\nabla^2 \phi = 0 \tag{1}$$

Numerically, the Laplacian can be approximated using the finite difference method, and Laplace's equation can be solved on the grid by iteratively calculating the following equation:

$$\phi_{i,\ j} = \frac{1}{4}\left(\phi_{i+1,\ j} + \phi_{i-1,\ j} + \phi_{i,\ j+1} + \phi_{i,\ j-1}\right) \tag{2}$$

for each i-th row and j-th column.

The initial growth point, from which the discharge pattern will branch out, is set to be the middle entry on the top row. Every immediate neighbour not currently in the discharge pattern is considered a possible candidate for growth.

Suppose that there are k candidate points. The growth probability at the α-th candidate point is given by:

$$p_\alpha = \frac{(\phi_\alpha)^\eta}{\sum_{\beta=1}^{k} (\phi_\beta)^\eta} \tag{3}$$

where ϕ_β is the value of the grid at the β-th candidate point and η is an exponent governing the relation between electric potential and probability.

A growth point is then randomly selected from the set of candidate points with the above weighted probabilities. The selected growth point is added to the discharge pattern, and its grid value is fixed to be zero.

The potential inside the grid is recalculated by iteratively solving Laplace's equation with these new boundary conditions taken into account, and the growth process is repeated until a point touching the bottom row is added to the discharge pattern.

The algorithm was first tested on a 5×5 matrix. After initializing and iterating through Laplace's equation 100 times, we obtain the following matrix (blue denotes the initial growth point; red denotes the only candidate point):

$$\begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & 0.25 & 0.25 & 0.25 & 0.25 \\ 0.50 & 0.50 & 0.50 & 0.50 & 0.50 \\ 0.75 & 0.75 & 0.75 & 0.75 & 0.75 \\ 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \end{bmatrix} \tag{4}$$

The algorithm trivially selects the only candidate point as its next growth point and fixes the new point to be zero:

$$\begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.25 & 0.25 & 0.00 & 0.25 & 0.25 \\ 0.50 & 0.50 & 0.50 & 0.50 & 0.50 \\ 0.75 & 0.75 & 0.75 & 0.75 & 0.75 \\ 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \end{bmatrix} \tag{5}$$

Taking into account the new fixed value constraints, we iteratively solve Laplace's equation again and obtain the following matrix (the grid values are approximated for brevity):

$$\begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.203 & 0.158 & 0.00 & 0.158 & 0.203 \\ 0.451 & 0.428 & 0.390 & 0.428 & 0.451 \\ 0.721 & 0.713 & 0.704 & 0.713 & 0.721 \\ 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \end{bmatrix} \tag{6}$$

Taking into account the weighted probabilities, the algorithm selects the left candidate point and fixes it to zero:

$$\begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.148 & 0.00 & 0.00 & 0.148 & 0.181 \\ 0.411 & 0.367 & 0.367 & 0.411 & 0.428 \\ 0.702 & 0.690 & 0.690 & 0.702 & 0.708 \\ 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \end{bmatrix} \tag{7}$$

The function terminates once it reaches a point on the bottom row:

$$\begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.077 & 0.00 & 0.00 & 0.083 & 0.109 \\ 0.198 & 0.00 & 0.00 & 0.225 & 0.274 \\ 0.441 & 0.00 & 0.386 & 0.544 & 0.565 \\ 1.00 & 0.00 & 1.00 & 1.00 & 1.00 \end{bmatrix} \tag{8}$$

ii. Version 3.0.1

The algorithm was modified to also accept any circular 2-dimensional array. The process slightly differs from that of the rectangular grid.

We first initialize an $n \times n$ grid of zeros and apply the midpoint circle algorithm with a radius of $(n-1)/2$ to ensure that the disk defined by the midpoint algorithm remains inside the grid.

The potential outside the circle is fixed to be zero, the potential on the boundary of the circle is fixed to be one, and the potential inside the circle is found by iteratively applying Equation (2).

The initial growth point is set to be the center of the disk, and the growth process is applied as described in the rectangular case until a point touching the boundary is added to the discharge pattern.

b. Results

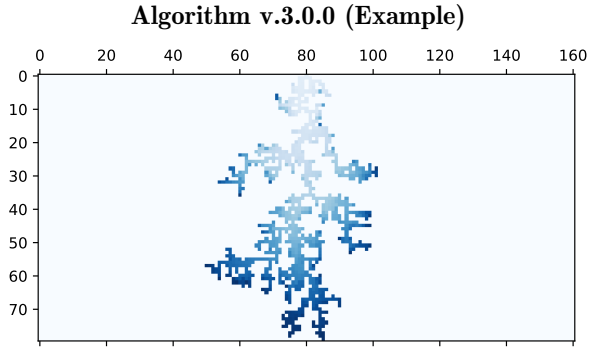

Fig. 1: Figure of the single-tree algorithm traversing an 80×160 homogeneous matrix with $\eta = 1$.

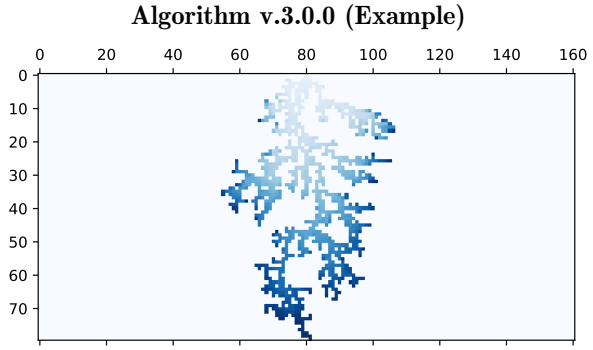

Fig. 2: Figure of the single-tree algorithm traversing an 80×160 homogeneous matrix with $\eta = 1$.

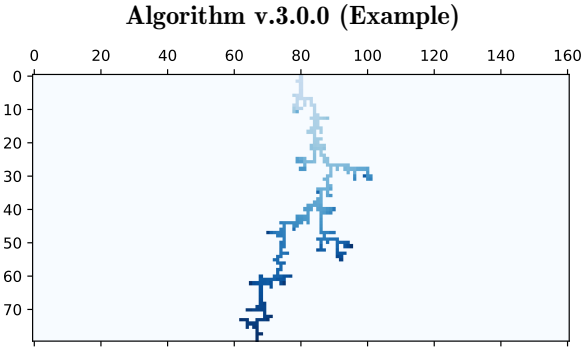

Fig. 3: Figure of the single-tree algorithm traversing an 80×160 homogeneous matrix with $\eta = 2$.

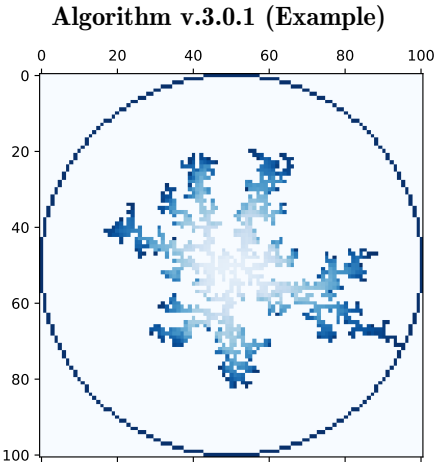

Fig. 4: Figure of the single-tree algorithm traversing an 80×80 homogeneous matrix with $\eta = 1$.

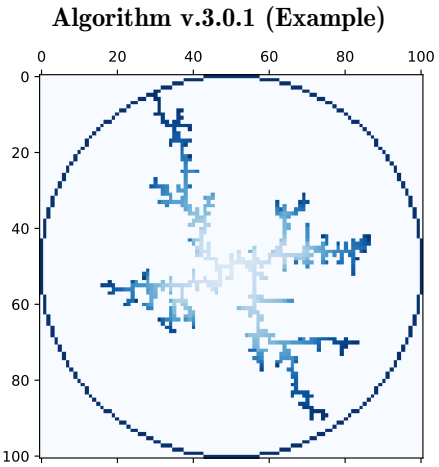

Fig. 5: Figure of the single-tree algorithm traversing an 80×80 homogeneous matrix with $\eta = 2$.

The path taken by each algorithm is recorded in blue. Given that these are static images, a colour gradient was chosen in order to provide a sense of direction: the earliest traversed entries are lighter while the most recent traversed entries are darker. See GitHub for animations of these runs.

c. Discussion

The single-tree algorithm successfully accomplishes its purpose of faithfully replicating lightning discharge patterns originating from a single source; the algorithm's trajectory is determined stochastically and in a physical manner.

i. Version 3.0.0

Unfortunately, this algorithm fails to achieve the main objective of this project since this version lacks analogous substitutes for a cathode and an anode from which real-world Lichtenberg figures are produced through wood burning. This issue is fully resolved in version 4.0.0.

ii. Version 3.0.1

This algorithm is based on the one presented in Niemeyer, Pietronero, and Wiesmann [3]. In their paper, the center of the disk is taken to be a radial electrode from which the discharge pattern emerges, while the boundary of the disk is modeled as the opposite electrode.

Although the development of this version of the algorithm was an interesting exercise, it does not substantially contribute towards the achievement of our primary objective, which is to simulate two simultaneous discharge patterns.

III. Multi-Tree Algorithm

a. Methodology

The main objective of this project was to develop a functioning multi-tree algorithm.

The single-tree algorithm successfully simulated the electrical discharge patterns observable in lightning but fell short in reproducing the discharge patterns seen in the formation of Lichtenberg figures on wood.

To accurately represent the discharge originating from both a cathode and an anode, it was necessary to implement a two-tree setup.

i. Version 4.0.0

We first initialize an $m \times n$ grid of zeros, with periodic boundary conditions on the left and right sides. The value of the top row is fixed to be -10, and the value of the bottom row is fixed to be 10. Note that the value of the top row differs from the previous rectangular algorithm; the symmetric initial conditions in this version are required for the unbiased growth of two trees.

Given these initial and boundary conditions, the potential inside the grid is found by iteratively applying Equation (2).

We define two initial growth points: one for the downward-growing tree, and one for the upward-growing tree. The growth point for the former is set to be the middle entry on the top row, and the growth point for

the latter is set to be the middle entry on the bottom row. The immediate neighbours of each growth point are considered possible candidate points for their respective tree.

Without loss of generality, we first consider the formation of the downward-growing discharge pattern. Unlike in the single-tree algorithm where all grid values are non-negative, the presence of negative-valued entries requires shifting the probability values upwards to prevent the trees from clumping up.

Suppose that there are k candidate points for the downward-growing tree. We denote by ϕ_m the candidate point with the lowest grid value. Then the growth probability at the α-th candidate point is given by:

$$p_\alpha = \frac{(\phi_\alpha + |\phi_m|)^\eta}{\sum_{\beta=1}^{k}(\phi_\beta + |\phi_m|)^\eta} \tag{9}$$

Note that, since the potential is negative near the top of the grid and positive near the bottom, the growth probability exhibits a downward bias, towards the upward-growing tree. This physically represents the attraction of opposing charges.

A growth point is then randomly selected from the set of candidate points with the above weighted probabilities. The selected growth point is added to the discharge pattern, and its grid value is fixed to be -10, as opposed to zero, to attract the upward-growing tree.

After a growth point is appended to the downward-growing tree, we repeat the process with the upward-growing discharge pattern. Suppose that there are ℓ candidate points for the upward-growing tree. We denote by ϕ_m the candidate point with the lowest grid value. Then the growth probability at the α-th candidate point is given by:

$$p_\alpha = \frac{(-\phi_\alpha + |\phi_m|)^\eta}{\sum_{\beta=1}^{\ell}(-\phi_\beta + |\phi_m|)^\eta} \tag{10}$$

Note that, since the potential is negative near the top of the grid and positive near the bottom, the negative sign in front of the grid values causes the growth probability to exhibit an upward bias, toward the downward-growing tree. Again, this physically represents the attraction of opposing charges.

A growth point is then randomly selected from the set of candidate points with the above weighted probabilities. The selected growth point is added to the discharge pattern, and its grid value is fixed to be 10 to attract the upward-growing tree.

The potential inside the grid is recalculated by iteratively solving Laplace's equation with these new boundary conditions taken into account, and the growth process is repeated in an alternating manner for both trees until their discharge patterns touch.

ii. Version 4.0.1

For aesthetic purposes, the multi-tree algorithm was modified to exhibit horizontal rather than vertical growth. As such, all "up" terms were changed to "left," and all "down" terms were changed to "right."

b. Results

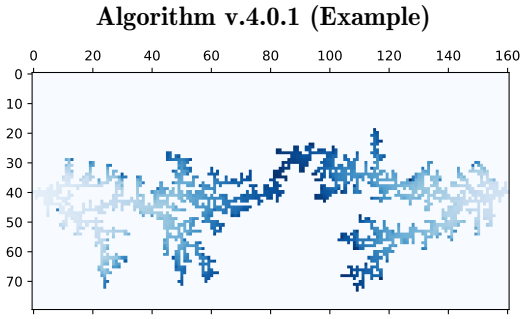

Fig. 6: Figure of the multi-tree algorithm traversing an 80×160 homogeneous matrix with $\eta = 1$.

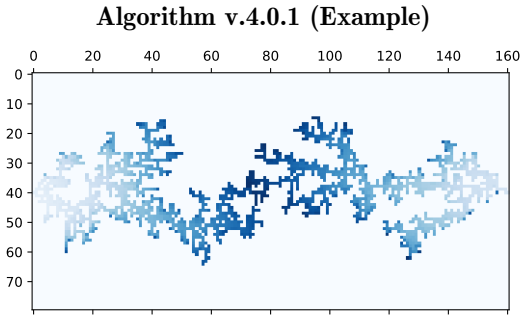

Fig. 7: Figure of the multi-tree algorithm traversing an 80×160 homogeneous matrix with $\eta = 1$.

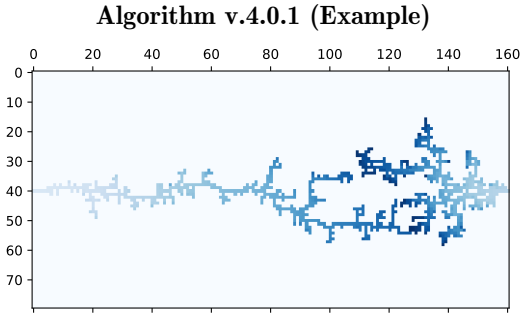

Fig. 8: Figure of the multi-tree algorithm traversing an 80×160 homogeneous matrix with $\eta = 2$.

The path taken by each algorithm is recorded in blue. See GitHub for animations of these runs.

c. Discussion

The multi-tree algorithm successfully replicates, to the best of our knowledge, the formation of Lichtenberg figures from a cathode and an anode in an insulating material. The multi-tree algorithm addresses the limitations of the previous algorithms mentioned in Chen and Gao [1] and earlier in this paper.

It is important to note that, given the increased complexity of the multi-tree algorithm, a larger number of iterations is required to achieve satisfactory convergence

in the application of the discrete Laplacian. The results for algorithms v4.0.0 and v4.0.1 were computed using 500 iterations instead of 100.

Additionally, we have observed that the multi-tree algorithm occasionally generates an electric discharge pattern in which one tree is more spread out than the other. This characteristic is particularly noticeable when $\eta = 2$ and $\eta = 1/2$ (see Figure 8). It is unclear why this bias exists, and understanding this behaviour could help improve some of the algorithm's shortcomings.

IV. Conclusion

The main objective of this project was to create an algorithm capable of accurately simulating the path of an electric current traversing through an insulating material. We are confident that the multi-tree algorithm presented in this paper has achieved this objective.

We find that version 4.0.1 is, to date, the algorithm that most closely mimics the formation of Lichtenberg figures on wood as seen in online video demonstrations.

Future improvements will likely revolve around optimizing the algorithm's efficiency. For example, implementing the method presented by Kim et al. [4] could greatly reduce the runtime of the discrete Laplacian calculations.

References

[1] S. Chen and S. Gao, "Simulating lichtenberg figures," 2022, Unpublished manuscript. [Online]. Available: https://www.researchgate.net/publication/366029117_Simulating_Lichtenberg_Figures

[2] A. A. Tsonis and J. B. Elsner, "Fractal characterization and simulation of lightning," *Beitr. Phys. Atmosph.*, vol. 60, no. 2, pp. 187–192, 1987.

[3] L. Niemeyer, L. Pietronero, and H. J. Wiesmann, "Fractal dimension of dielectric breakdown," *Phys. Rev. Lett.*, vol. 52, no. 12, pp. 1033–1036, 1984.

[4] T. Kim, J. Sewall, A. Sud, and M. C. Lin, "Fast simulation of laplacian growth," *IEEE Computer Graphics and Applications*, vol. 27, no. 2, pp. 68–76, 2007.