

Implementing L-systems with Localist Cells

Peter Duggins

University of Waterloo

Computational Neuroscience Research Group

Waterloo, Canada

psipeter@gmail.com

Abstract—L-systems provide a formal grammar for producing branching structures and studying plant development. This project expands the biological plausibility of L-systems by implementing a locality-constrained, cell-based alternative to symbolic production rules. I show that this formalism can reproduce two classic L-systems, the Fractal Binary Tree and Sierpiński’s Triangle, using simple rules for cellular update and reproduction. I conclude by discussing extensions to the framework aimed at modelling the growth of biological plants.

I. INTRODUCTION

L-systems [1] are powerful tools for producing branching and fractal shapes, many of which resemble self-similar patterns observed in natural systems [2]. L-systems consist of (a) an alphabet of symbols, (b) production rules that expand symbols into longer sequences of symbols, and (c) visualizers that convert symbolic sequences into geometric objects [3]. Since their inception in the late 1960s, L-systems have been used to describe the growth of numerous biological structures, particularly in the plant kingdom (trees, branches, leaves, etc.), and are increasingly utilized for research into plant development [4]. For example, a common developmental program utilizing L-systems can be used to generate diverse leaf shapes, reproducing features including the patterning of serrations, lobes, leaflets, vascular systems, and leaf blades [5].

Although L-systems produce lifelike structures using rules inspired by biology, the formalism’s fundamental mechanisms lack biological realism in two respects. Firstly, complex L-systems often include production rules which generate long sequences of symbols in a single step. These symbols may represent physical objects, graphical instructions, or symbolic updates. The complexity and diversity of these sequences contrasts with growth in most biological systems, in which cell behavior (communication, movement, reproduction) occurs either dynamically or in a more discretized fashion. Although the end-result of multiple coordinated

cell divisions may be summarized by a single complex production rule, more fine-grained rules are needed to describe the growth process from the perspective of an individual cell; fine-grained rules also probe the difficulties of overcoming coordination problems in multi-agent systems. Secondly, symbolic sequences created by L-systems must be visualized using biologically-implausible “turtle graphics,” in which a virtual agent performs a feedforward traversal of the sequence and relies on dedicated symbols (computational instructions) to navigate turns and branches. Until this external visualization is performed, the symbolic sequence has no physical interpretation, making it difficult to study subsections of the structure during the growth process.

A. Motivation

The goal of this project is to implement an alternative to L-systems that is constrained to use discrete, local mechanisms that respect biological limitations. In this formalism, the fundamental unit is analogous to a biological cell: the information available to the cell, and the cell’s impacts on the system, are restricted to a local neighborhood. The proposed system should satisfy three constraints: (a) cells’ knowledge of the wider system is gathered from signals communicated by neighboring cells, (b) cells reproduce by birthing children sequentially into the adjacent space, and (c) cells update and reproduce in a random order. I begin by specifying the formalism in more detail, including cell attributes, cell production rules, and global simulation order. I next introduce two classical L-systems, the fractal binary tree and Sierpinski’s triangle, then show that this formalism can reproduce their structure while respecting locality. I conclude by proposing extensions to the formalism that, I argue, could capture more advanced L-system production rules, then discussing how these extensions could be used to model the growth of plant-like structures.

II. METHODS

A. Cell Attributes

Each cell in the simulation is defined by a point in space (x, y) and an orientation θ . Each cell is aware of other cells in its immediate neighborhood, defined by a graph structure that ties parents, children, and interaction partners. When a cell reproduces, it temporarily tracks each child, while each child remembers its parent. When a parent-child pair become spatially separated (e.g. from pushing, see below), these attributes are updated accordingly, preserving physical locality. In addition, each cell has a type (analogous to a symbol in L-systems) which governs its behavior through various (re)production rules (see below).

Most interesting L-systems require a few additional attributes. A cell may perceive the distance d_{type} to the nearest cell of a particular type. This can be viewed as the cell measuring the concentration of a morphogen transmitted by the nearest cell of that type, and is computationally implemented by cell-to-cell communication whenever interaction (birthing or pushing) occurs. A cell may also track its age to allow time-dependent updates.

B. Cell Production Rules

The behavior of a cell is critically determined by its type-specific update and reproduction rules. A cell's update function modifies internal attributes such as type and age. A cell's reproduce function births one or two children, endows the children with the parent's (x, y, θ) , then perturbs the children's attributes by some amount.

C. Simulation

The simulation proceeds as follows. After the initial cells have been created (the "axiom" in L-systems terminology), this simulation loop is repeated for t timesteps:

- 1) for cell in cell list:
 - a) update
 - b) reproduce
 - c) recursively push child (see below)
 - d) append child to cell list
- 2) shuffle cell list
- 3) Plot (x, y) of each cell, or plot lines between parents and each child

An essential part of this algorithm is the recursive push procedure. When a child is birthed and moved according to its new θ , it may come to occupy the same space as an existing cell. To maintain the uniform structure of traditional L-systems, the latter cell must be moved to prevent overlap. The push procedure implements this movement

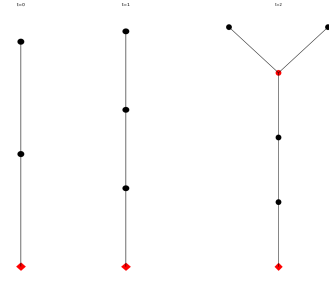


Fig. 1. One iteration of the simulation loop

recursively, so that a child birthed midway through e.g. a line of cells will successively push all downstream cells until the line has been extended by one unit. When a cell pushes another cell, the "pusher" becomes the parent of the "pushed" cell, while the pushed cell replaces the child of the pusher. At the same time, the child (and each cell that is pushed) has its distances d_{type} incremented by one. This is a computationally-efficient means of tracking cells' local neighbors (for the sake of cell-to-cell communication) and relative distances (for the sake of morphogen-specific updates).

Figure 1 shows an example of the simulation running through one full timestep (one evaluation of each cell). As time proceeds from left to right, the middle black cell produces a child and pushes it forward along $\theta = \frac{\pi}{2}$, which causes an overlap and a recursive pushing of the cell on $(0, 2)$. Next, when the pushed cell updates, its d_{red} has been incremented by one, putting it beyond the threshold for a type change. The cell turns from black to red, then runs the update procedure for red, birthing two children at $\pm \frac{\pi}{4}$ relative to the parent.

D. Implementation

The simulation code was written in Python and is available on GitHub. The code is structured such that users can easily modify the cell, update, and reproduce blocks to incorporate additional mechanisms, making extensions to more complex systems straightforward.

III. FRACTAL BINARY TREE

The fractal binary tree (FBT) is a simple branching structure produced by the following L-system:

- $1 \rightarrow 11$
- $0 \rightarrow 1[0]0$

1 draws a line segment, 0 draws a line segment ending in a leaf, and $[]$ branch at $\pm \frac{\pi}{4}$ [3]. The result is a tree in which, after every iteration, line lengths are doubled and leaves bifurcate, as shown in figure 2.

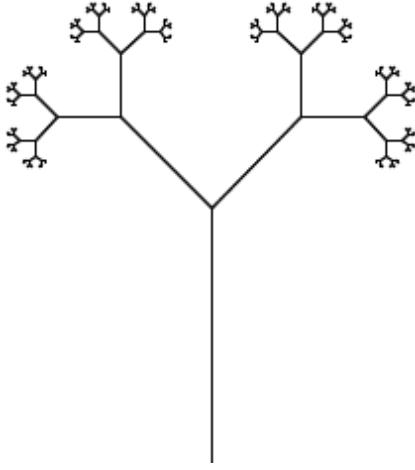


Fig. 2. Fractal Binary Tree produced by an L-system after $n = 7$ iterations [3]

To model this system, I used two types, straight and branch, each represented as 2D points with the following rules:

- reproduce:
 - straight: birth 1 child with $\theta = \theta_{parent}$
 - branch: birth 2 children with $\theta = \theta_{parent} \pm \frac{\pi}{4}$
- update:
 - if $d_{branch} > 1$ and 0 children: straight \rightarrow branch

Figure 3 shows the resulting growth after 2, 4, and 6 timesteps. As intended, the reproduction of new branches on the leaves of the structure bifurcates its leading edge, while continued reproduction of cells in center of the structure grow its length. The result is a tree that grows and divides in a fractal, binary pattern, despite randomized, local cellular update.

IV. SIERPIŃSKI'S TRIANGLE

Sierpiński's Triangle (ST) is an equilateral triangle divided recursively into smaller equilateral triangles (see figure 4). Like the FBT, the ST is a canonical self-similar set, and can be generated in numerous ways: by removing triangles from a solid triangle, by shrinking and duplication, using cellular automata, or using the arrowhead construction. My implementation is based off an L-system that uses the arrowhead construction:

- $A \rightarrow B-A-B$
- $B \rightarrow A+B+A$

where A/B draw line segments and $-/+$ turn by $\frac{\pi}{3}$. The result is an arrowhead in which, after every iteration, straight segments are replaced by smaller arrowheads of different curvatures.

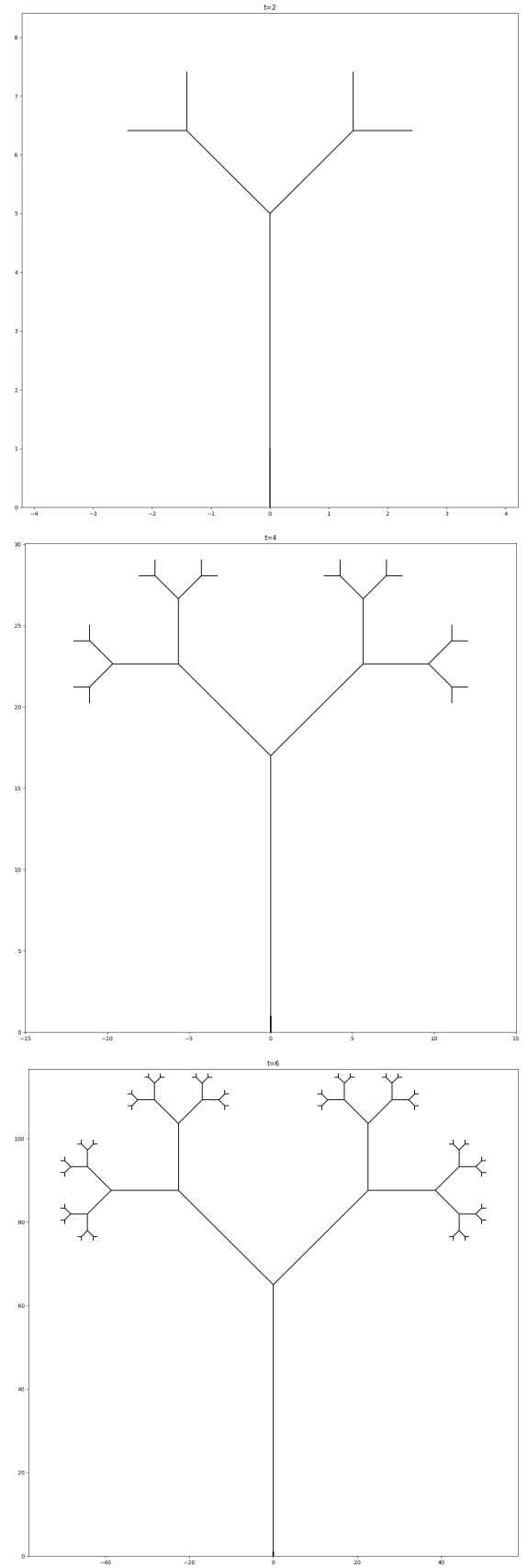


Fig. 3. Fractal Binary Tree produced by local cellular update after $t = 2, 4, 6$ timesteps

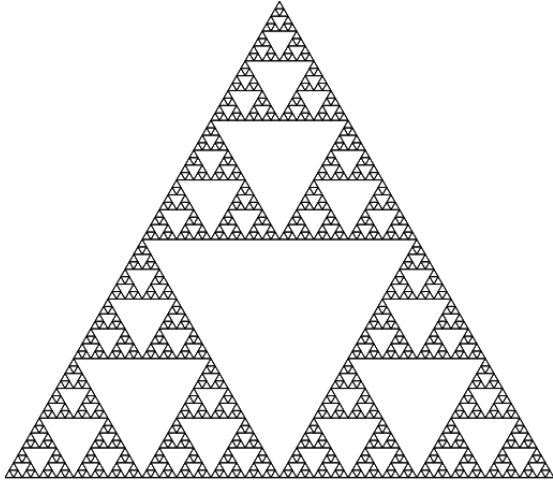


Fig. 4. Sierpiński Triangle produced by an L-system after $n = 6$ iterations [3]

To model this system, I use two types, A and B, each represented as a line segment with a left and right edge, that run the following rules:

- reproduce:
 - A: birth 2 children:
 - * B on left with $\theta = \theta_{parent} + \frac{\pi}{3}$
 - * B on right with $\theta = \theta_{parent} - \frac{\pi}{3}$
 - B: birth 2 children:
 - * A on left with $\theta = \theta_{parent} - \frac{\pi}{3}$
 - * A on right with $\theta = \theta_{parent} + \frac{\pi}{3}$
- update:
 - none (no changes to type or d_{type})

Figure 5 shows the resulting growth after 2, 4, and 6 timesteps. As intended, the reproduction rules introduce more curvature as time progresses. The result is a curve that approximates the fractal shape of the true ST despite randomized, local cellular update.

V. DISCUSSION

The examples above demonstrate that the proposed formalism can reproduce the branching and fractal shapes generated by two well-known L-systems. The production rules in these systems lend themselves well to a localist, cell-based description, making this translation relatively intuitive. In the remainder of the paper, I discuss how to capture more complicated L-systems using this formalism, paying particular attention to models of plant growth and development.

A. L-system Extensions

Extensions to L-system productions may be classified into three categories. With stochastic grammars, more

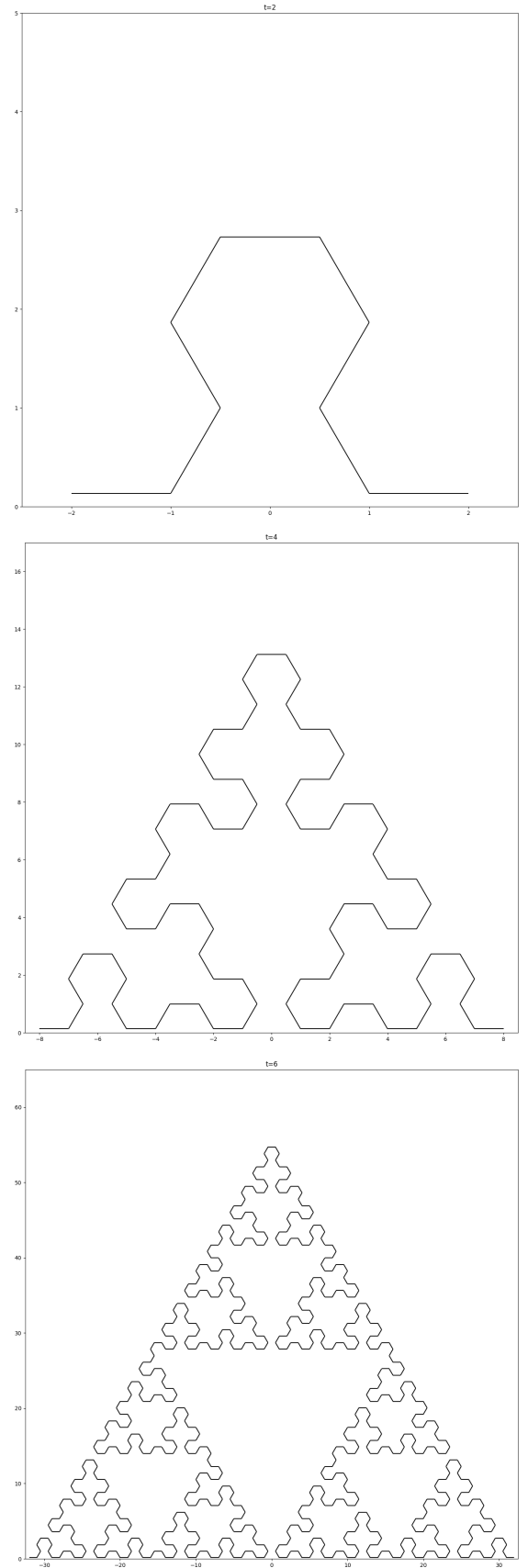


Fig. 5. Sierpiński Triangle produced by local cellular update after $t = 2, 4, 6$ timesteps

than one production rule may be associated with a given symbol, and a non-deterministic procedure is used to choose which production is applied. Although stochastic productions are unnecessary for producing perfectly symmetric or fractal geometric patterns, they are important in capturing the pseudo-randomness observed in biological structures, such as the three-dimensional structure of blood vessels [6]. Stochasticity could be incorporated into the proposed formalism by using random-number generators in the update and reproduce functions. Specifically, if θ_{child} was drawn from a probability distribution instead of a fixed list (e.g. a normal distribution with $\mu = \theta_{\text{parent}}$ and parameterized σ^2), growth and branching angles would include a controllably-random element that could be tuned to capture the particular geometric structure of various plant species.

In context-sensitive grammars, an L-system production depends not only on the symbol it modifies, but on the symbol's left- and right-hand neighbors. This addition allows L-systems to handle local interactions, partially addressing the concerns about locality that motivated this project. Unsurprisingly, context-sensitive grammars are used in many models of plant growth to account for interactions within the structure itself [7]. However, this extension does lend itself well to long-range or graded interactions within the structure. In contrast, the formalism used here, in which cells communicate with their neighbors and track their relative proximity to morphogen-producing types, allows update and reproduce procedures with arbitrary dependencies on other cells present in the structure. This should allow the formalism to easily capture any context-sensitive L-system productions. Note that incorporating multi-directional growth would require a more sophisticated update of intercellular distances than the currently-implemented in the push procedure. These updates could be computed using partial differential equations in a continuous-space layer underlying the graphical layer; such approaches have been quite successful in modelling morphogen diffusion in other developmental models [8].

Finally, in parametric grammars, each symbol has an associated list of parameters that can be used by production rules or the visualizer. Parametric grammars are similar to the formalism used here, in which cells' attributes govern reproduction, plotting, and updating of internal attributes. Parametric grammars are often used to model the age of a symbol for the purposes of time-dependent updates such as dieoff (removal of the symbol from the structure), which plays a large role in biological development [4].

B. Automation

As argued above, the proposed formalism should account for all types of L-system production rules. This raises the question: can any L-system be implemented using these techniques, and if so, is it possible to automate the translation between an L-system ruleset and cellular attributes, updates, and reproductions?

Although I believe the answer to the first question is “yes,” I’m skeptical that an automatic procedure for this translation is possible. Firstly, the locality constraint I impose in this paper is not a necessary feature of the cell-based implementation: a reproduce rule could technically spawn any number of cells in an arbitrary spatial configuration. Therefore, in translating to a localist cellular ruleset, the ambiguous notion of “locality” must be interpreted by the modeller with reference to the system at hand. Unless this interpretation can be systematically codified, a subjective element in translation will remain, preventing automation by a non-human system.

Furthermore, it can be difficult to translate long production rules of intermixed graphical, computational, and physical symbols into sequences of discrete operations by different cell types. In the ST example, I leveraged the fact that both production rules reproduced the original symbol ($A \rightarrow B-A-B$) to rephrase the rule into successive angle-dependent reproductions. However, a slightly different rule ($A \rightarrow B-B-B$) would require both a type update and a consecutive string of B reproductions. Engineering this sequence in a manner that respects locality/asynchronicity (and doesn't catastrophically interfere with the system's other production rules) requires a degree of creativity. For instance, the solution might require the creation and usage of cell attributes that do not appear in the original L-system (such as d_{branch} in the FBT example). This type of generative, trial-and-error model specification is likely beyond the scope of current translation algorithms.

C. Plant Growth

The original inspiration for this project was to model the development of plant structures using a biologically-plausible variant of the L-systems methodology. Having demonstrated the utility of the proposed framework for L-systems generally, I can now discuss how it may be applied to model biological growth. As mentioned in the introduction, a large body of work has applied L-systems to researching plant development, including many of the models available through Algorithmic Botany.

Although the form and subject of these models differ widely, there are several common features worth noting.

Firstly, many of these models use three-dimensional space to simulate growth; while some plant substructures may be considered two-dimensional (e.g. leaves), an appreciation for higher dimensionality when modelling e.g. the branching of tree trunks is a necessity for realistic simulation. The proposed formalism is consistent with higher spatial dimensions and with volumetric cells: the only required change would be the addition of cell attributes z and polar angle ϕ .

The other essential feature for plant growth is interaction with a simulated environment. The behavior of biological cells is strongly influenced by environmental cues such as atmospheric conditions (wind, temperature, particulate concentrations) and sunlight. In many models of plant growth, the direction of cell propagation is towards a simulated light source [4]. Cells that were once exposed to light, but are now occluded by newer segments of the tree structure, may experience simulated aging and death, leading to bottom-up pruning of the structure [9]. Cells may also compete for space, with overcrowding also leading to developmentally-critical cell death [10]. When combined with stochastic growth, the resulting structures bear a strong resemblance to biological growth, as shown in figure 6. Using the golden ratio to determine branching angles would also be an interesting extension motivated by both biology and mathematical models [11].

Due to time constraints, these extensions were beyond the scope of the current project. Future work would incorporate a third spatial dimension, stochastic reproduction rules, a simulated light source, cell aging, light-dependent dieoff, and additional nodes types such as veins, branches, and chlorophyll cells.

VI. CONCLUSION

Growth in biological systems involves a complex interplay between cellular and environmental processes. This project builds on L-systems' generation of fractal shapes by aligning symbolic production rules with locality-respecting cellular mechanisms. I showed that certain examples of self-similar growth could be produced using this framework, then speculated how the inclusion of additional cell rules and environmental interactions could be used to model the growth of biological structures such as trees.

REFERENCES

[1] A. Lindenmayer, "Mathematical models for cellular interactions in development i. filaments with one-sided inputs," *Journal of theoretical biology*, vol. 18, no. 3, pp. 280–299, 1968.

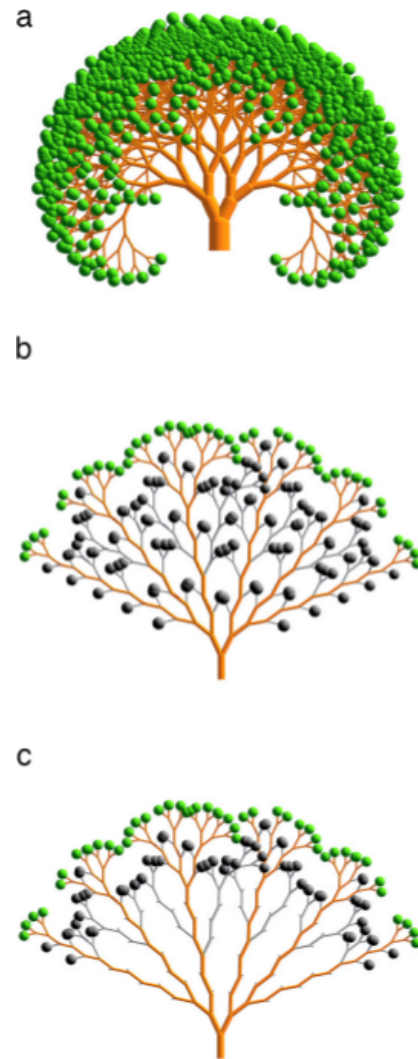


Fig. 6. A two-dimensional tree modelled using the methods described in section V-C. Image reproduced from [4]. In (a), branching growth is unchecked. In (b), occluded leaves die, become black balls that do not reproduce. In (c), branches attached to dead leaves are removed.

[2] P. Prusinkiewicz and A. Lindenmayer, *The algorithmic beauty of plants*. Springer Science & Business Media, 2012.

[3] Wikipedia contributors, "L-systems — Wikipedia, the free encyclopedia," 2004, [Online; accessed 24-July-2018]. [Online]. Available: <https://en.wikipedia.org/wiki/L-system>

[4] P. Prusinkiewicz, M. Cieslak, P. Ferraro, and J. Hanan, "Modeling plant development with l-systems," in *Mathematical Modelling in Plant Biology*. Springer, 2018, pp. 139–169.

[5] A. Runions, M. Tsiantis, and P. Prusinkiewicz, "A common developmental program can produce diverse leaf shapes," *New Phytologist*, vol. 216, no. 2, pp. 401–418, 2017.

[6] M. A. Galarreta-Valverde, M. M. Macedo, C. Mekkaoui, and M. P. Jackowski, "Three-dimensional synthetic blood vessel generation using stochastic l-systems," in *Medical Imaging 2013: Image Processing*, vol. 8669. International Society for Optics and Photonics, 2013, p. 866911.

[7] W. Kurth, B. Sloboda *et al.*, "Growth grammars simulating trees—an extension of l-systems incorporating local variables

and sensitivity.” 1997.

- [8] A. F. Marée and P. Hogeweg, “Modelling dictyostelium discoideum morphogenesis: the culmination,” *Bulletin of mathematical biology*, vol. 64, no. 2, pp. 327–353, 2002.
- [9] W. Palubicki, K. Horel, S. Longay, A. Runions, B. Lane, R. Měch, and P. Prusinkiewicz, “Self-organizing tree models for image synthesis,” in *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3. ACM, 2009, p. 58.
- [10] P. Prusinkiewicz and P. Barbier de Reuille, “Constraints of space in plant development,” *Journal of experimental botany*, vol. 61, no. 8, pp. 2117–2129, 2010.
- [11] R. A. Dunlap, *The golden ratio and Fibonacci numbers*. World Scientific, 1997.

VII. APPENDIX

The simulation code was written exclusively by the author, and is available, alongside the results, on GitHub: https://github.com/psipeter/plant_growth. My initial attempts at extending the model to three dimensions, stochastic growth, and cell aging may also be viewed there.