

Thomas McLoughlin

Registration number 100203952

2021

---

---

# **GPU Accelerated Method for Constructing and Rendering Trees**

---

---

Supervised by Dr. Stephen Laycock



University of East Anglia  
Faculty of Science  
School of Computing Sciences

## **Abstract**

The inclusion of trees in virtual environments when trying to represent nature is essential and the branching structure of trees is vital in representing many other types of foliage. This project covers the research of related work involved with algorithmic generation of trees and the mathematical basis for the discussed approaches. The underlying principles relating to the algorithmic production of fractal patterns helpfully coincide with the requirements for producing believable branching structures.

The project aim is to convert and extend the Lindenmayer-system based tree construction method presented by (Prusinkiewicz et al., 1996) to be used as an independent OpenGL module, the intuitive parametric L-system model was chosen because of its ability to produce diverse results with varying input. This module allows the addition of trees to a real-time environment with minimal user interaction, avoiding the difficulties and expenses of manually producing tree models.

## **Acknowledgements**

I'd like to thank my supervisor, Stephen Laycock for his brilliant help and support throughout this project. Thank you to George Smith and Harry Tucker for their interest in my work and their support during the COVID-19 lockdowns.

## Contents

<b>1. Introduction</b>	<b>5</b>
1.1. Context . . . . .	5
1.2. Related Work . . . . .	5
<b>References</b>	<b>12</b>
<b>A. Parameterised L-system Trees</b>	<b>15</b>

## List of Figures

1.	Example diagram of parent branch $P_AP_B$ bifurcating into child branches $P_BP_1$ with a rotation of $\theta_1$ and $P_BP_2$ with a rotation of $-\theta_2$ . . . . .	6
2.	A ramiform used to create a realistic bifurcation . . . . .	7
3.	A leaf “configuration”: plan view (top) and perspective view (bottom) . . . . .	7
4.	The visualised production rule $p_1$ , axiom and first 3 recursions defined by the snowflake D0L-system from (Prusinkiewicz et al., 1996) pg.12 . . . . .	8
5.	Example of tree parameters . . . . .	10
6.	Space colonisation steps . . . . .	11
7.	Tree structures produced using the proposed L-system described by Prusinkiewicz et al. (1996) . . . . .	15

## 1. Introduction

This section contains a contextual introduction for the project and a collection of related work found during the research phase of development.

### 1.1. Context

Creating and rendering realistic models of trees manually requires advanced expertise with modelling software packages. This limits the ability to produce convincing 3D scenes for small developers with restricted resources.

The purpose of including trees in a natural environment is to provide realism. Trees are common natural structures present in even simple environments throughout the history of computer graphics and have seen many iterations as technology has advanced allowing for more detailed and realistic results.

The aim of this project is to provide a method for creating and rendering trees to be used in a real-time graphics application. This method should be simple to use and implement into an existing OpenGL project.

### 1.2. Related Work

In this subsection various related works will be discussed with respect to how they contribute to the main knowledge required for this project, ordered chronologically. These areas of main knowledge are the branching structure, branch thickness and leaf placement of the constructed trees.

Aristid Lindenmayer proposed an early theory for the development of organisms using a cell's current state and being combined with input that the cell receives from its neighbour (Lindenmayer, 1968a) (Lindenmayer, 1968b). Two new cells are produced from this development to replace the existing cell and the cycle repeats for the two new cells.

This process of generating new outputs recursively to produce larger structures became "Lindenmayer systems" or the abbreviated "L-systems" which became important tools in pattern generation for future applications. L-systems became a common approach for the generation of branching patterns in flora which is where the aim of this project is concerned.

Another early paper that is referenced by many later studies of the subject is by Honda (1971) where he presented one of the earliest algorithmic methods for creating a branching structure. This was done by starting with a parent branch which then bifurcates into two child branches and rotating them by given angles about the termination point of the parent branch, an example of which can be seen in Figure 1. By continuing this method he treated each child branch as another parent and bifurcated them to expand the branching structure which would be continued until a desired result is reached.

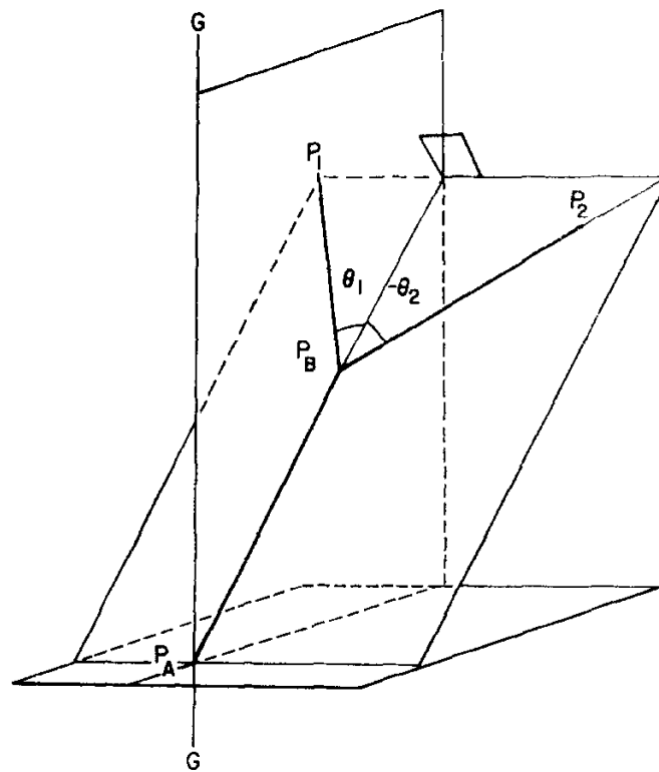


Figure 1.: Example diagram of parent branch  $P_AP_B$  bifurcating into child branches  $P_BP_1$  with a rotation of  $\theta_1$  and  $P_BP_2$  with a rotation of  $-\theta_2$

This paper does not include any information relating to branch thickness or leaf placement as its focus was only on the branching structure. However, this paper provides a groundwork for many later studies that will be discussed in this project.

Bloomenthal (1985) presents his own tree generation process, specifically for a maple tree. He wanted to move beyond simply the branching pattern of a tree which most

efforts before had focused on (Brooks et al., 1974; Marshall et al., 1980; Kawaguchi, 1982; Aono and Kunii, 1984; Smith, 1984). For the construction of the branches and trunk he uses splines to create a tree skeleton, the use of splines rather than straight lines is chosen to produce a more natural structure. He then uses generalised cylinders with varying radii across the splines to create thickness. He also describes the use of ramiforms, shown in Figure 2, to make branch bifurcations realistically curve rather than have an acute separation.

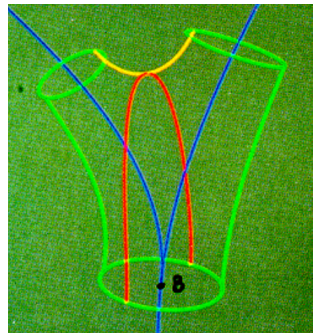


Figure 2.: A ramiform used to create a realistic bifurcation

The method he describes for leaf placement uses polygonal structures to which he applies a leaf texture, these polygons are then brought together in clusters that he calls “configurations” and for each limb that does not exceed a given diameter and that has no outgoing limbs, a leaf configuration is chosen at random, scaled randomly and placed at the limb tip. The interior lines of each polygon correspond to hinge points of the leaves that could be manipulated to show leaf distortion from wind. This method provides a more realistic result than the use of simple primitives (Kawaguchi, 1982; Lintermann and Deussen, 1999; Candussi et al., 2005).

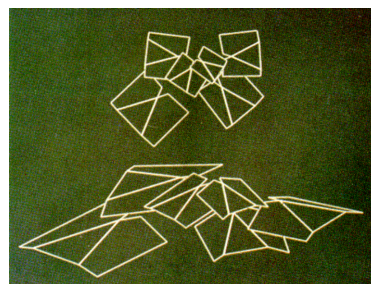


Figure 3.: A leaf “configuration”: plan view (top) and perspective view (bottom)

Prusinkiewicz et al. (1996) presents a method of tree generation using an advanced L-system, developed from the work of Lindenmayer (1968a) and applying that system to the bifurcating parent and child branch trios displayed by Honda (1971). Their chosen method for representing their models is by using turtle graphics in a 3D space and the string generated from the L-system uses symbols that correspond to controlling the turtles orientation in space (Szilard and Quinton, 1979; Prusinkiewicz, 1986).

Prusinkiewicz et al. produce a variation on a standard L-system known as a “deterministic” L-system or “D0L-system” which produces the same result every time. They demonstrate the application to turtle graphics with the well-known snowflake curve (Mandelbrot, 1982; Koch et al., 1906) shown in Figure 4.

The D0L-system uses an Axiom string and a production rule to recursively develop a longer and more complex string. Once a chosen result is reached, the string is traversed and each character or group of characters corresponds to the movement of the turtle. Fractals give a convenient illustration of the basic workings of an L-system (Prusinkiewicz, 1986; Prusinkiewicz and Lindenmayer, 2012). For example, the D0L-system below represents that of the snowflake fractal where  $F(s)$  corresponds to a movement forward of distance  $s$  and  $+(r)$  or  $-(r)$  correspond to a rotation positively or negatively about the  $x$  axis by  $r$  degrees.

**Axiom  $\omega$ :**  $F(1) - (120)F(1) - (120)F(1)$

**Production  $p_1$ :**  $F(s) \rightarrow F(s/3) + (60)F(s/3) - (120)F(s/3) + (60)F(s/3)$

The axiom draws an equilateral triangle with edges of unit length. Production  $p_1$  replaces each line segment with a polygonal shape where the line segment has been given a convex triangular appendage. The Production  $p_1$  can be visualised as so:

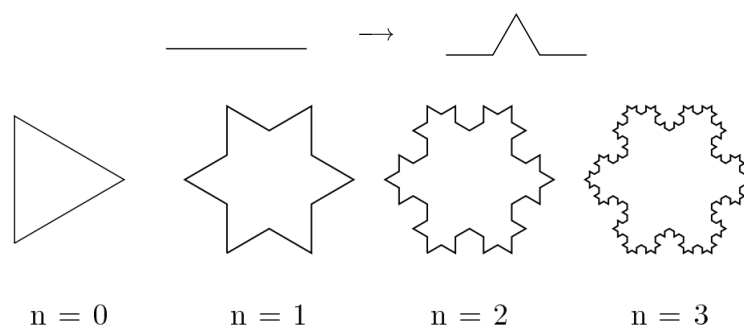


Figure 4.: The visualised production rule  $p_1$ , axiom and first 3 recursions defined by the snowflake D0L-system from (Prusinkiewicz et al., 1996) pg.12



Prusinkiewicz et al. go on to present a complex D0L-system to be used for constructing tree structures by creating a Parametric L-system allowing for input to alter the axiom string producing different results along with using a conditional requirement to decide automatically when to stop recursing.

This L-system is defined below, in this context  $A(s, w)$  refers to a line segment or “apex” of length  $s$  and width  $w$ . Production  $p_1$  begins with the conditional statement to only continue if the length  $s$  of the current apex is greater than or equal to  $min$  which is a chosen minimum length. If the condition is met then  $!(w)F(s)$  draws a line of length  $s$  and width  $w$ .

Lines  $a_1$  and  $a_2$  refer to the bifurcated child apices of the parent apex which are given a rotation of  $\alpha_1$  and  $\alpha_2$  radians about the  $x$  axis and a rotation of  $\phi_1$  and  $\phi_2$  radians about the  $y$  axis. The length of the child apices is calculated by multiplying the parent apex length  $s$  by the corresponding length degradation parameters  $r_1$  and  $r_2$  respectively. A degradation in width is calculated using  $q$  and  $e$ .

$\omega: A(100, w_0)$

$p_1: A(s, w) : s \geq min \rightarrow !(w)F(s)$

$a_1 [+ (\alpha_1) / (\phi_1) A(s * r_1, w * q^e)]$

$a_2 [+ (\alpha_2) / (\phi_2) A(s * r_2, w * (1 - q)^e)]$

This L-system is used to produce various complex tree patterns, shown in section A of the Appendix, ranging from mathematical fractal patterns to some realistic looking structures.

Prusinkiewicz et al. do not directly propose a method of leaf placement as this paper focuses on tree structure and the manipulation of L-systems, they do however provide an example where terminating apices of an L-system could be replaced with a leaf shape to give a rudimentary leaf placement approach.

Weber and Penn (1995) produced a model for creating and rendering trees that emphasises overall geometric structure of the tree rather than strictly following botanical principles which they credit similarity to the aims of Reeves and Blau (1985). They admit that their model resembles that proposed by Oppenheimer (1986) which used fractals with parameters such as “branching angle” and “helical twist” to produce trees. However, Weber et al. propose that the use of fractal theory self similarity by Oppenheimer is not necessary and restricts results to only a small number of basic trees.

Weber et al. used cylinders and cones as branch segments, each child branches base circle coinciding with the top circle of the parent branch. Transformation of the child

branches being controlled through the intentionally intuitively named parameters that can be edited by the user to produce a desired result.

Figure 5 shows some of the many parameters that can be used to affect tree generation, *0SegSplits* defines the number of splits at each branch termination, *0SplitAngle* controls the angle at each split, *0CurveRes* refers to the taper given to each branch to slowly reduce thickness of each branch, and *1Rotate* defines the y axis rotation of the branch. The number in front of each parameter refers to a branch, 0 being the root branch, the values of parent branch parameters are passed on to the child branches.

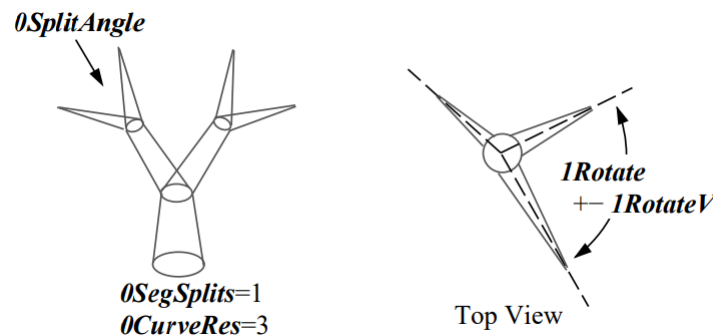


Figure 5.: Example of tree parameters

Leaf placement is handled using a “Level” system that refers to the number of recursions from the root branch. If a parameter *Leaves* is non-zero then the final level of recursion is used to place leaves on the final set of child branches, using a set of leaf specific parameters to control the rotation of each leaf. The shape of the leaves is controlled using the parameter *LeafShape* allowing for different tree species to be represented by adding different leaf shapes. Weber and Penn (1995) served as a groundwork for many future approaches and advances in tree rendering (Remolar et al., 2004; Wesslén and Seipel, 2005). Weber himself went on to produce a simulation focused method of tree generation including some extra features such as branches avoiding obstacles (Weber, 2008).

Another branch structure method, that differs uniquely from methods mentioned above, is that of space colonisation (Prusinkiewicz et al., 2007). The steps of this method can be seen in Figure 6 and works by producing a three-dimensional envelope to be used for the tree crown containing a set of “attraction points” which are used as targets for branch growth. The tree skeleton iteratively expands by producing new branches in the direction of the attraction points. Once an attraction point is reached by a branch, the

point is removed and the branch terminates. This process is continued until all attraction points are reached, when no new branches are within attraction radius of remaining points or when a user inputted number of iterations has been reached.

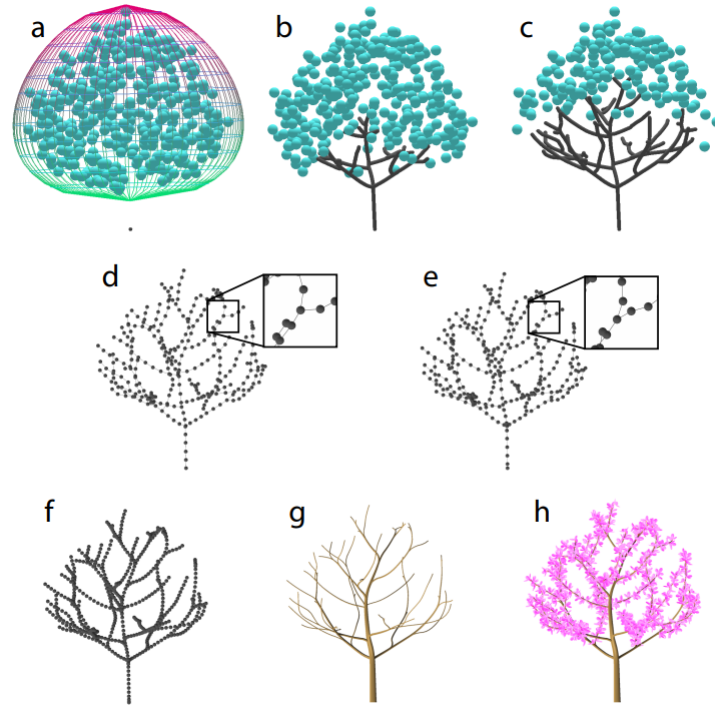


Figure 6.: Space colonisation steps

Once a tree skeleton has been finalised they determine the width of the limbs using the pipe model (Shinozaki et al., 1964) and the tree geometry is modeled using generalised cylinders (Bloomenthal, 1985).

A leaf placement method is not provided as part of this paper due to it focusing solely on branch construction. However, they do suggest that the placement density of leaves or flowers could be controlled using absolute distance from the root position to give a realistic dispersion of leaves throughout the tree crown.

## References

- Aono, M. and Kunii, T. L. (1984). Botanical tree image generation. *IEEE Computer Graphics and Applications*, 4(5):10–34.
- Bloomenthal, J. (1985). Modeling the mighty maple. *ACM SIGGRAPH Computer Graphics*, 19(3):305–311.
- Brooks, J., Murarka, R. S., Onuoha, D., Rahn, F. H., and Steinberg, H. A. (1974). An extension of the combinatorial geometry technique for modeling vegetation and terrain features.
- Candussi, A., Candussi, N., and Höllerer, T. (2005). Rendering realistic trees and forests in real time. In *Eurographics (Short Presentations)*, pages 73–76. Citeseer.
- Honda, H. (1971). Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of theoretical biology*, 31(2):331–338.
- Kawaguchi, Y. (1982). A morphological study of the form of nature. *SIGGRAPH Comput. Graph.*, 16(3):223–232.
- Koch, H. et al. (1906). Une méthode géométrique élémentaire pour l'étude de certaines questions de la théorie des courbes planes. *Acta mathematica*, 30:145–174.
- Lindenmayer, A. (1968a). Mathematical models for cellular interactions in development i. filaments with one-sided inputs. *Journal of Theoretical Biology*, 18(3):280–299.
- Lindenmayer, A. (1968b). Mathematical models for cellular interactions in development ii. simple and branching filaments with two-sided inputs. *Journal of theoretical biology*, 18(3):300–315.
- Lintermann, B. and Deussen, O. (1999). Interactive modeling of plants. *IEEE Computer Graphics and Applications*, 19(1):56–65.
- Mandelbrot, B. B. (1982). *The fractal geometry of nature*, volume 1. WH freeman New York.
- Marshall, R., Wilson, R., and Carlson, W. (1980). Procedure models for generating three-dimensional terrain. *SIGGRAPH Comput. Graph.*, 14(3):154–162.

- Oppenheimer, P. E. (1986). Real time design and animation of fractal plants and trees. *ACM SIGGRAPH Computer Graphics*, 20(4):55–64.
- Prusinkiewicz, P. (1986). Graphical applications of l-systems. In *Proceedings of graphics interface*, volume 86, pages 247–253.
- Prusinkiewicz, P., Hammel, M., Hanan, J., and Mech, R. (1996). L-systems: from the theory to visual models of plants. In *Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences*, volume 3, pages 1–32. Citeseer.
- Prusinkiewicz, P., Lane, B., and Runions, A. (2007). Modeling trees with a space colonization algorithm. *NPH*, 7:63–70.
- Prusinkiewicz, P. and Lindenmayer, A. (2012). *The algorithmic beauty of plants*. Springer Science & Business Media.
- Reeves, W. T. and Blau, R. (1985). Approximate and probabilistic algorithms for shading and rendering structured particle systems. *ACM siggraph computer graphics*, 19(3):313–322.
- Remolar, I., Rebollo, C., Chover, M., and Ribelles, J. (2004). Real-time tree rendering. In Bubak, M., van Albada, G. D., Sloot, P. M. A., and Dongarra, J., editors, *Computational Science - ICCS 2004*, pages 173–180, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Shinozaki, K., Yoda, K., Hozumi, K., and Kira, T. (1964). A quantitative analysis of plant form-the pipe model theory: I. basic analyses. *Japanese Journal of ecology*, 14(3):97–105.
- Smith, A. R. (1984). Plants, fractals, and formal languages. *SIGGRAPH Comput. Graph.*, 18(3):1–10.
- Szilard, A. L. and Quinton, R. (1979). An interpretation for dol systems by computer graphics. *The Science Terrapin*, 4(2):8–13.
- Weber, J. and Penn, J. (1995). Creation and rendering of realistic trees. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, page 119–128, New York, NY, USA. Association for Computing Machinery.

- Weber, J. P. (2008). Fast simulation of realistic trees. *IEEE Computer Graphics and Applications*, 28(3):67–75.
- Wesslén, D. and Seipel, S. (2005). Real-time visualization of animated trees. *The Visual Computer*, 21(6):397–405.

## A. Parameterised L-system Trees

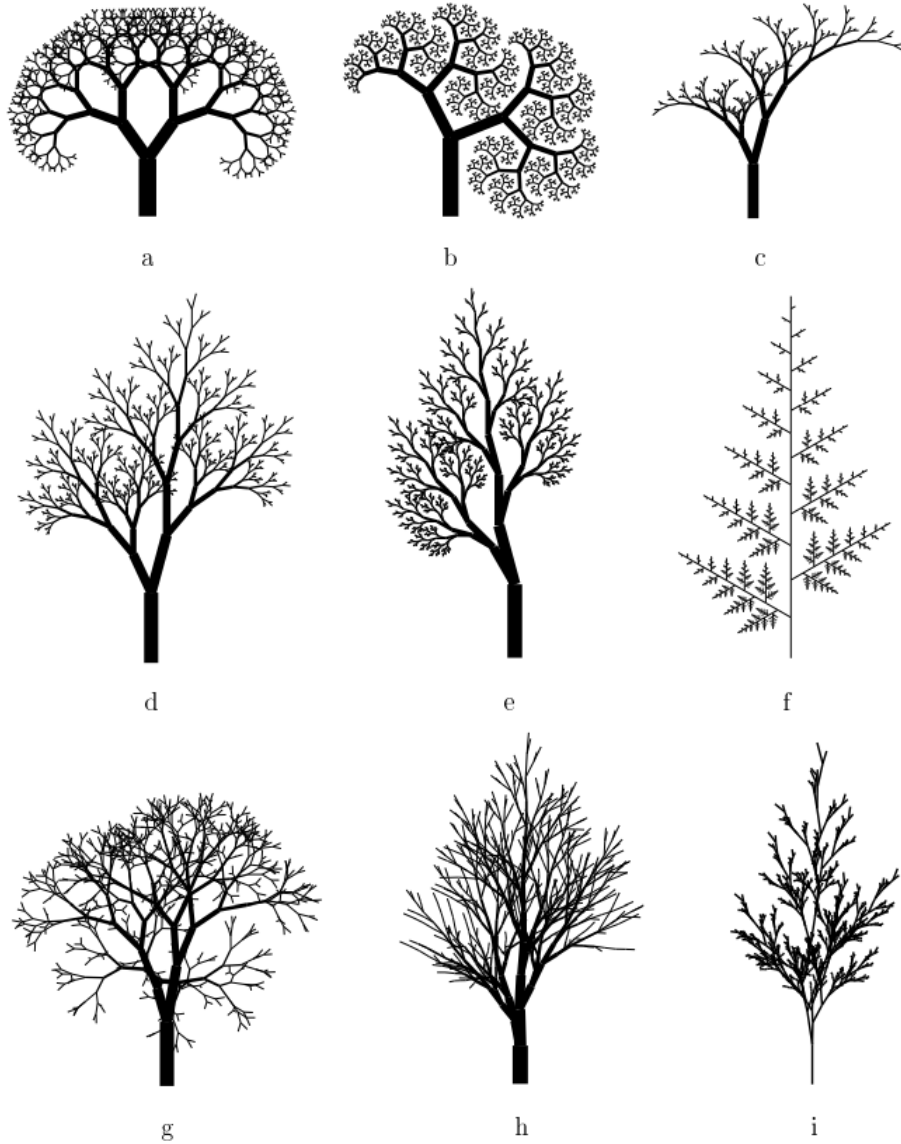


Figure 7.: Tree structures produced using the proposed L-system described by Prusinkiewicz et al. (1996)