# RESTATEMENT OF L-SYSTEM FRACTALS WITH ANALYTICAL PATH FUNCTIONS IN A CONTINIOUS FRACTAL SPACE

HENK MULDER

Abstract. Using rewrite rules to create L-system fractal tree generates fractal paths that are typically not analytical. We define fractal trees as sets of paths and give the formulation of these paths as smooth analytical functions that take binary trees as one of their arguments. We extend the definition of fractal trees beyond the rewrite limitations of L-systems to create tree fractal with branches defined by arbitrary functions for arbitrary paths. This creates a fractal space with its own fractal geometry.

## 0. Overview

This paper is work in progress. Rather than wait for the entire paper to be complete, we publish for contained sections of this development. The contents to date:

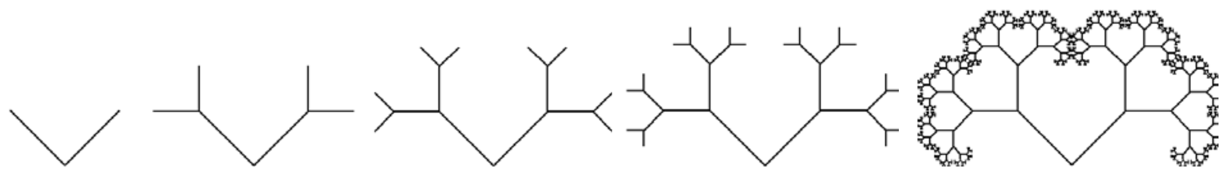
This list will be updated from time to time.

This paper is published on www.geneticfractals.org

## 1. Introduction

In this paper, we are focusing on L-system tree fractals, i.e. fractals that have self similarity and that are generated by a rewriting rule. That rewriting rule is often geometric though that is not a restriction. Below is a common examples of an L-system tree fractal.

The example shows a simple tree: two branches. In the subsequent images, we add a scaled copy of that initial shape to each branch. After 9 iterations, the fractal tree has 1024 branches, which the resolution of image can't very well show.


Initial shape 1st iteration 2nd iteration 3rd iteration 9th iteration

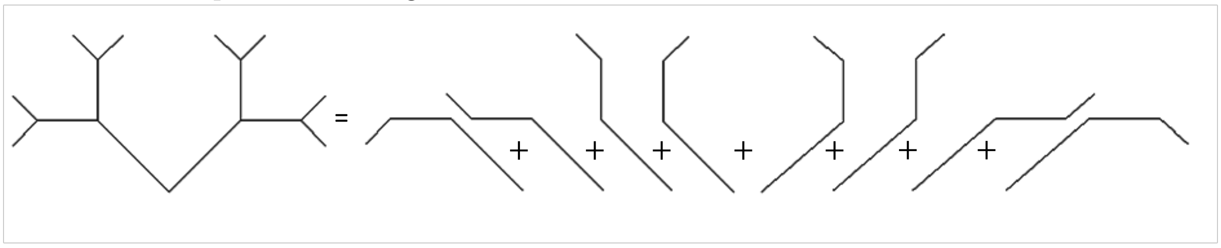
The formal rewriting rule for this fractal is:

Variables: A: add line left π/4, shortened by 2/3, B: add line right π/4, shortened by 2/3
Start: AB
Rules: (A → AB), (B →AB)

Although intriguing, the only analysis we can perform on this fractal is numerical, not functional.

## 2. Fractals as sets of paths

One purpose of this paper is to study the analytical properties of tree fractals and perhaps extend the tools for creating such fractals. We therefore first need a mathematical model of a tree with branches. The model we propose here, defines a tree as a collection of paths that partially coincide at the trunk and some its branches. For example the fractal below left is the union of the paths on the right:



**Definition 2.1.** A tree is a set of paths where all paths coincide at least at start of the trunk and where individual paths that may coincide with other paths for a limited interval from the origin of these paths.

More formally, a tree T is the union of n $\text{paths}_p$ where $\text{path}_a$ (t) = $\text{path}_b$ (t) for any two paths for domain [0..t] and $0<t<\infty$.

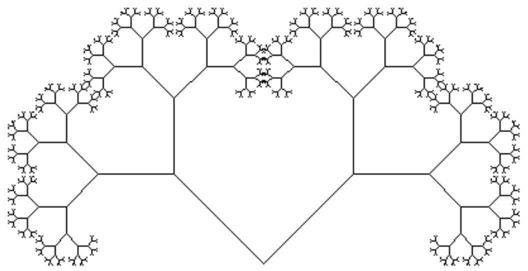
**Definition 2.2.** A L-system tree fractal is a tree constructed by repeating a L-system rewrite rule infinitely many times.

**Definition 2.3.** An approximate L-system tree fractal is a tree constructed by repeating a L-system rewrite rule a finite number of times.

Having defined a fractal as a set of paths, we can now restrict our focus on an arbitrary path.

## 3. PATH FUNCTIONS

Initially we will focus on L-system tree fractal as presented in the introduction.



Considering the rewrite rule, there are two variables that determine the next segment in each path for any rewrite iteration:

$\rho = \frac{\Delta r}{r}$ , i.e. the relative path length between any subsequent line segments

$\theta_k$, the angle to turn to, left or right, relative to the direction of the previous line segment

The relative length is independent from the iteration *k*. Therefore, the length of any path segment after *k* iterations starting from an initial path length $r_0$ is:

$$r_k = r_0 \left(\frac{\Delta r}{r}\right)^k = r_0 \rho^k$$

The angle $\theta_{p,k}$ however is specific for each iteration *k* and for each path *p*. To find the absolute angle $\Theta_{p,k}$ with respect to the first path segment of any line segment after *k* iterations, we have to add them all from the root of the tree $\Theta_k = \sum_{n=0}^{k} \theta_n$.

For a given path, we can write the relative positions of any vertex *k* with respect to the previous vertex *k-1*

$$\Delta v_k = v_k - v_{k-1}$$

$$= \begin{cases} x_k \\ y_k \end{cases} = \begin{cases} r_k \cos \Theta_k \\ r_k \sin \Theta_k \end{cases}$$

$$= \begin{cases} x_k \\ y_k \end{cases} = \begin{cases} r_0 \rho^k \cos\left(\sum_{n=0}^{k} \theta_n\right) \\ r_0 \rho^k \sin\left(\sum_{n=0}^{k} \theta_n\right) \end{cases}$$

Since the absolute position of $v_k$ is

$$v_k = \sum_{m=0\ to\ k}^{k} \Delta v_m.$$

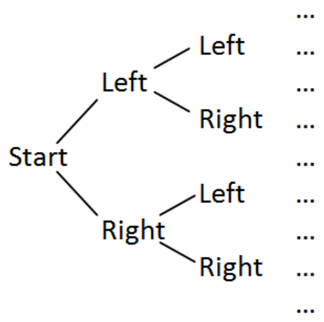
It follows that absolute position of a vertex k on a given path is

**Equation 3.3.**

$$v_k = \begin{cases} x_k \\ y_k \end{cases} = \begin{cases} \sum_{m=0}^{k} r_0 \rho^m \cos\left(\sum_{n=0}^{k} \theta_n\right) \\ \sum_{m=0}^{k} r_0 \rho^m \sin\left(\sum_{n=0}^{k} \theta_n\right) \end{cases}$$

## 4. Fractals and binary trees

For an L-system tree fractal with a rewrite condition involving two branches, we may recognize that the choices of left and right form a binary tree:



We can define such a binary tree as a set of binary paths, representing Left and Right by 0 and 1. Each path is then defined by a binary number, i.e. a sequence of 0's and 1's. The tree above would be given by (after the 'start' node):

00 → Left, Left
01 → Left, Right
10 → Right, Left
11 → Right, Right

A complete binary tree corresponding to a path with for $d$ segments would be given by:

$$B = \{b \middle| 0 \leq b \leq 2^d\}.$$

We will denote a specific vertex on that path is given by an index *i, $B_i$*. So if for a specific path B=1101 then $B_3$=0

Since this type of fractal has a constant directional change angle θ (π/4 in the fractal in the introduction), and if we agree that binary tree has -1 and 1 as its binary values instead of 0 and 1, then we can rewrite the path function as

$$v_k = \begin{cases} x_k \\ y_k \end{cases} = \begin{cases} \sum_{m=0}^{k} r_0 \rho^m \cos\left(\theta \sum_{i=0}^{k} B_i\right) \\ \sum_{m=0}^{k} r_0 \rho^m \sin\left(\theta \sum_{i=0}^{k} B_i\right) \end{cases}$$

What this shows is that L-system tree fractal with two initial branches are a function of a binary path, a constant directional change angle and a relative length. The relative length 'stretches' the fractal whereas the directional change angle determines how we 'skip' through a periodic domain.

### 4.1. Extension to n-ary trees

Fractals may have more than two branches at each node. In this case, θ the direction change angles may take *n* values instead of just 2. If the branches are evenly spaced out, i.e. θ is constant between subsequent branches from the same vertex, we can use n-ary trees instead of binary trees. If the subsequent branches are not evenly spaced out, we can still use this formulation but need to define a function $\theta(b)$ that maps the n-ary state to $\theta$.

### 4.2. Self similarity

An important consequence of this presentation of such branching fractals is that we can see why these fractals have self-similarity: n-ary trees of infinite lengths repeat themselves at every vertex and therefore any function derived from such n-ary trees will also have self similarity although the geometry may have been deformed by the specific path function.

**Proposition 4.2.1**. A fractal derived from a n-ary tree has analytical self similarity

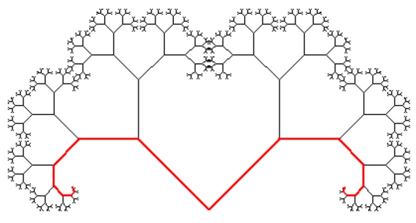
### 4.2. Convergence and boundary of the fractal

As can be seen from Equation 2.3, the maximum radius of the fractal tree is given by the geometric series $\sum_{m=0}^{k} r_0\rho^m$ which converges for $\rho < 1$. Therefore

**Proposition 4.2.2**. An L-system tree fractal is bounded by a circle centred on the root of the fractal with radius $r_0(1-\rho)^{m+1}/(1-\rho)$

## 5. Reference paths

Looking at a fractal tree in the introduction, we see that there two paths that stand out: paths where θ= -π/4 or θ= π/4 at all vertices. These paths are marked in red below.



These paths are special and we'll refer to them as *reference paths*. Since these are mirror images in this example, we will focus on one only. Using the previous formulation, these paths are given by:

$$v_{ref} = \begin{cases} x_k \\ y_k \end{cases} = \begin{cases} \sum_{m=0}^{k} r_0\rho^m \cos(m\theta) \\ \sum_{m=0}^{k} r_0\rho^m \sin(m\theta) \end{cases}$$

By inspecting the symmetry between sibling branches we can see that all the information regarding L-system tree fractal is captured by the reference path: the progression of the relative lengths of the path segment and the absolute angles in which they turn. When this fractal was derived from a binary tree, we can construct the entire fractal just from its reference path. If the fractal is based on an n-ary tree, it will be enough to know its n reference paths. In fact, the same holds true for any of the fractal paths, as long as you know the binary path that was followed.

**Proposition 4.2.1**. A tree fractal fully is defined by any of its paths and the associated binary path

**Proposition 4.2.2**. A tree fractal is fully defined by its reference path(s)

Formally, a tree fractal is defined by a set of paths *p*:

$$v_p = \begin{cases} x_k \\ y_k \end{cases} = \begin{cases} \sum_{m=0}^{k} r(m) \cos\theta(m) P_p \\ \sum_{m=0}^{k} r(m) \sin\theta(m) P_p \end{cases}$$

Where

$r(k)$ is the relative length of the path segments

$\theta(k)$ is the directional change angle

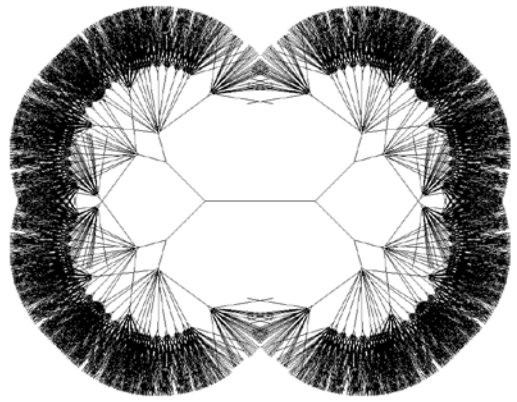
$P_p$ is the summation of the binary path p $\sum_{i=0}^{k} B_{i,p}$

This is a remarkable result because for all the geometric complexity of a tree fractal, its constituent is a just a simple reference path function $r(k)\cos\theta(k)$, $r(k)\sin\theta(k)$. The summation of the binary tree is also straight forward: it is the of number of 1's minus 0's in its path (which represents -1's and 1's, remember).

In this formulation $r(k)$ determine how the fractal expands and $\theta(k)$ how it wiggles. Although we have used simple proportional functions in the example before: $r(k) = r_0\rho^k$ and $\theta(k) = k\theta$ we are free to model more complex behaviour.

Below is a regular tree fractal with the parameters:

- $r(k) = (0.25 + \sin(2\pi\rho/5))^k$, i.e. the relative length varies periodically
- $\theta(k) = 2\pi/(k+1)$, i.e. the branch angles get progressively smaller, starting with $2\pi$
- $k = 0..11$



Due to the geometric distortion it is not obvious that this is a fractal but it is in fact a tree fractal from its definition: it is based on a binary tree.