GENERATING A FRACTAL TREE

STUDENT: JINGYI GAO
ADVISOR: MITCHELL NEWBERRY

1. Abstract

Inspired by a beautiful tree pattern carved into a stone screen, we want to generate self-similar tree pattern on any arbitrary shape. In this paper, we will discuss the concept of fractal, Hausdorff dimension, self-similarity, and scaling relationships. We will then explain in what sense this tree is and isn't a fractal, how does the scaling relationship correspond to its physical properties, and how do we determine these relationships of our tree. Finally, we will explain how to build fractal trees from recursive relationships.

2. History and Motivation

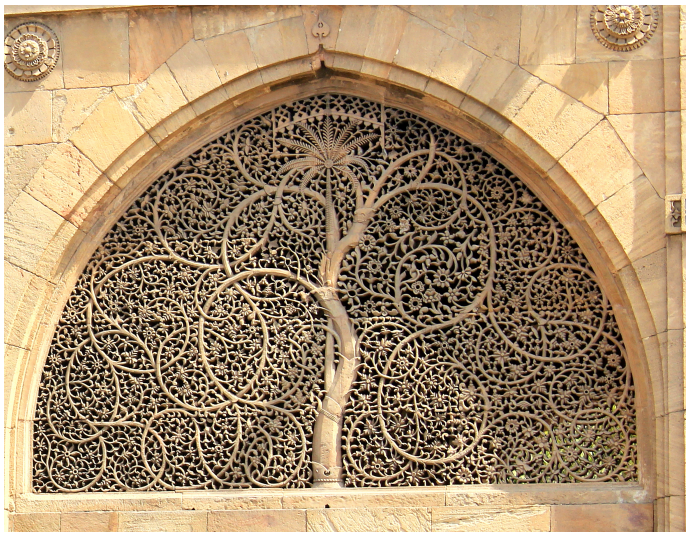


Figure 1. The marble screen of Sidi Saiyyed mosque
Image: CC-BY Vrajesh Jani

The whole project is motivated by this beautiful tree pattern on the marble screen of Sidi Saiyyed mosque in Ahmedabad, Gujarat, India, as shown in Figure 1, which is a notable piece of Islamic architecture, built in 1572AD.

Date: August 25, 2020.

Marveling at these beautiful carve stone windows, we attempted to generate this kind of tree pattern on our own. So the question becomes: if we want to decorate any shape of window, how would be generate a pattern like this? One answer is that we can definitely follow our artistic sense and draw it out using paper and pen. However, if I want to do it more efficiently, can I write an algorithm to help me do this? Or in other words, are there any mathematical ways to describe this kind of tree pattern? Could they be generated by following some specific rules? Indeed, there are former researchers worked on interpreting art in a mathematical way. For example, Lu and Steinhardt[LS07] gave a formula producing quasi-crystalline girih patterns, Li[Li] computed the fractal dimension of Chinese calligraphy in their paper.

Figure 2. A second marble screen of Sidi Saiyyed mosque
Image: CC-BY Vrajesh Jani

In our case, even though Figure 1 and Figure 2 are two different patterns, you can still tell that these two windows belong to the same mosque. Both of them are evenly distributed over the window to fill the space, and the way they branch seems alike. Thus, intuitively, they are sharing some intrinsic properties.

As Mandelbrot said[Man83]

> "We often have a stereotype of geometry as 'cold' and 'dry,' since the Euclidean geometry is unable to describe many of the shapes in nature. For example, clouds are not spheres, mountains are not cones, coastlines are not circles, and lightning doesn't travel in a straight line. However, the arising of fractal geometry allow us to describe many of the irregular and fragmented patterns around us."

The seemingly irregular tree patterns can also be described as a fractal. To see how, let's start with understanding what are fractals.

3. Background

A canonical explanation of fractal is the coastline paradox[Man83]. How long is the British coastline? You might think that it's easy at the first glance—just measure the outline of the country, and you have the answer. However, the truth is that you'd end up getting different results using different measurement units. So what's going on? Obviously, the coast of Britain isn't straight. Every time you look closer to at a line that seems straight, you'll see more squiggles. Thus, the length of the coast seems to approach infinity. However, it doesn't seem to make any sense that we could have a space of finite area surrounded by an infinitely long curve.

This paradox is originated from our stereotype of curve and dimension. Indeed, the British coastline is different from a "typical" curve in the sense that the way it repeatedly generate squiggles on itself is doing some efforts to "fill the space." An analog of this endeavor is that we can knit a single continuous piece of yarn into sweater. Likewise, repeatedly folding an infinitely long curve cause it to almost cover an area. Thus, the British coastline can be considered to be something more than a curve but less than an area.

A classic invariant to describe the difference between curves and areas is dimension. The topological dimension is the general dimension we talk about, defined to be either a non-negative integer or infinity. For example, smooth curves are one-dimensional and smooth areas are two-dimensional. We can think of dimension as a metric to determine "how dense" an object is. The denser an object is, the bigger the dimension it has.

According to previous paragraph, the British coastline is something more than a curve but less than an area, and therefore, ought to have dimension between 1 and 2, which is unable for integral topological dimension to describe. Therefore, several other notions of dimension that are able to give fractional are defined. For example, Hausdorff dimension, similarity dimension, and box dimension. Indeed, British coastline is a fractal and Mandelbrot[Man83] defined fractals being "Every set with a non-integer Hausdorff Besicovitch dimension, or equivalently a set for which the Hausdorff Besicovitch dimension strictly exceeds the topological dimension," which implies that fractals are objects that are denser but not dense enough to get a "+1" increase in dimension.

According to Moran Theorem in Falconer's book [Fal04], Hausdorff dimension, similarity dimension, and box dimension are equivalent for pure self-similar fractals, simply means sets are made of scaled-down copies of themselves. Since Hausdorff dimension is relatively hard to compute, we put more emphasis on similarity dimension.

Definition 3.1 (Similarity dimension)**.** [Str18] The set is consist of m copies of $\frac{1}{r}$ scaled-down version of the set. The similarity dimension D_{sim} is the exponent of the scaling relationship between m and r, $m = r^{D_{sim}}$.

Example 3.2 (Cantor set)**.** *Cantor set is constructed by repeatedly removing the open middle third from an interval, and thus it has topological dimension $D_{top} = 0$ since it is a cloud of points. On the other hand ,it is consist of 2 copies of subset of size $\frac{1}{3}$ of the size of Cantor set. Thus, $m = 2$ when $r = 3$, and $D_{sim} = \log_r m = \log_3 2 \approx 0.63 > D_{top}$. Thus, Cantor set is a fractal.*

4. Determine the diameter relationship of the tree pattern

4.1. **Fractal dimension and scaling relationships.** After looking at notions of dimension, one observation is that dimension is often defined via a scaling relationship, i.e.

$$Y(x) = Y_0 \cdot x^{\alpha}$$

[New05] What's more, it plays the role of exponent in this formula, for example:

(1) intuitively, the dimension of a disk is 2, can be seen from the are formula of disk $A = \pi r^2$;
(2) similarly, the dimension of a ball is 3, can be seen from the volume formula of a ball $V = \frac{3}{4}\pi r^3$;
(3) similarity dimension D_{sim}: $m = r^{D_{sim}}$;
(4) box dimension [Str18] D_{box}: $N(\epsilon) \propto \frac{1}{\epsilon^{D_{box}}}$, where D_{box} is the exponent in the scaling relationship between ϵ and the minimum number of $D-$dimensional cubes of side ϵ needed to cover the set, denoted as $N(\epsilon)$.

4.2. **Alternative of describing fractal.** We've studied many fractals, and we notice the way we identify or generate them conform an identical principle. Recall the coastline paradox: the moment we realize its peculiarity comparing to other smooth curves is when we notice that it would be longer if observing it on a smaller scale. Likewise, for the cantor set, when we scale down, there would be more copies of itself. The key observation here is that for a fractal, there is a scaling relationship between the scale of the figure and the number of "objects" that we could identify. For example, for coastline, "object" stands for squiggle that can increase the total length of measurement; for Cantor set, "object" is the cloud of point looks the same as itself; for Sierpinski triangle [SWC$^+$15], "object" is the number of equilateral triangles. This correlation between scaling relationship and fractal not only coincides with the observation about how dimension is defined in previous section 4, but also gives us a new perspective of understanding fractals other than as a topological space.

Mandelbrot [Man83] in his book mentioned that word frequencies, wealth distribution, and earthquakes are all like fractal not in the topological sense but because they are following some scaling relationship.

In our case, our intuition told us the tree patterns are fractal-like since when we scaled down the pattern, we could see more branches and each branch is following the analogous rule of branching into two copies of itself. However, we don't want to think of tree patterns as "topologically" self-similar. Considering that branches in the tree patterns, as geometric objects, are of great inconsistency and variation comparing with each other in order to fulfill the visual aesthetic, it is improper to describe any branch as a scale-down of the whole tree. Therefore, instead, we want to describe the self-similarity of tree pattern statistically using some scaling relationship. And thus, instead, we sought to a parameter that can reflect the scaling property of tree pattern.

How do we find this parameter? Other than related to a scaling relationship, it's also important for it to be tangible and able to describe the shape of the tree directly for coding purpose. The diameter of branches is an excellent candidate. Indeed, trees as a well-studied self-similar branching system have well-defined relationships between branches.

If we simplify the tree model into bifurcating tree, and the diameters of child branches are equal, a conservation law has been frequently assumed to exist between the diameters of the parent branch d_{parent} and those of the two daughter branches d_{child_1} and d_{child_2} in bifurcating tree structure

$$d_{parent}^{\alpha} = d_{child_1}^{\alpha} + d_{child_2}^{\alpha}, \tag{1}$$

where $d_{parent} > d_{child_1} \geq d_{child_2}$.

Examples of how other researchers determine this scaling exponent in this conservation law 1 are listed in following subsections.

4.2.1. *Da Vinci's rule and its analog for two dimensional tree.* The original rule [Elo11] states that for a three dimensional tree, the cross-sectional area of branches is preserved, which leads to the formula as follow holds:

$$d_{parent}^{2} = d_{child_1}^{2} + d_{child_2}^{2}$$

i.e., the scaling exponent α now equals to 2.

Then what is the analog of Da Vinci's rule for two dimensional tree? There might be millions of different answers to this question, I personally came up with two.

(1) One analog for two-dimensional tree is that instead of cross-sectional areas are preserved, the cross-sectional length is preserved. i.e. the diameter of parent branch is equal to the sum of the diameter of two children branches,

$$d_{parent}^{1} = d_{child_1}^{1} + d_{child_2}^{1}$$

and thus the scaling exponent $\alpha = 1$.

(2) Another analog is that the same conclusion for three dimensional tree also holds for two dimensional trees! If you think of it as taking a photograph of the tree, it also make sense for the tree on the photo,

which is two-dimensional, to have the same diameter relationship as a three dimensional tree. Personally, both of them make sense.

4.2.2. *Murrays law.* The Murrays law[Mur26] predicts diameter relationship such that total dissipated energy during transportation of nourishment from root to stem and leaves is minimized, wherein $\alpha = 3$, i.e.

$$d_{parent}^3 = d_{child_1}^3 + d_{child_2}^3$$

4.2.3. *Scaling exponent in the conservation law and exponent in the scaling relationship.* The conservation law 1 reflects local property between diameter of parent and children branches, which is useful for coding purpose but cannot directly reflect the fractal-like property of tree patterns. Thus, we studied about whether this conservation law is able to be interpret as a scaling relationship that are able to desbribe the global property of the tree pattern. Recall that the Similarity dimension 3.1 reflects the global density of the shape. We generalized Similarity dimension 3.1 to define an exponent via a scaling relationship that is able to reflect the global density of a tree. Then, we use it to interpret the conservation law.

Generalize the definition of Similarity dimension 3.1, we have following definition for exponent that is able to reflect the global density of trees:

Definition 4.1 (Exponent that is able to reflect the global density of tree)**.** The branch of diameter d_{parent} is branching into m child branches that are of diameter $\frac{1}{r} \cdot d_{parent}$. Then, a parameter β that is able to reflect the global density of the tree is defined via $m = r^\beta$.

Conjecture 4.2. *The exponent in the conservation law 1 is the exponent that is able to reflect the global density of tree.*

Proof of 4.2 for the special case of symmetric trees. Simplify the model into bifurcating tree, and the diameters of child branches are equal. i.e.

$$d_{child_1} = d_{child_2} = d_{child}.$$

Thus, the conservation law

$$d_{parent}^\alpha = d_{child_1}^\alpha + d_{child_2}^\alpha,$$

is equivalent to

$$d_{parent}^\alpha = 2 \cdot d_{child}^\alpha$$

i.e.

$$d_{parent} = 2^{\frac{1}{\alpha}} \cdot d_{child} \tag{2}$$

Using Definition 4.1, we interpret Equation 2 in following way: the diameter of parent branch is composed of 2 copies of itself scaled down by a factor of $2^{\frac{1}{\alpha}}$. Hence, $m = 2$ when $r = 2^{\frac{1}{\alpha}}$, and the exponent β is

$$2 = (2^{\frac{1}{\alpha}})^\beta.$$

And thus,

$$\frac{1}{\alpha} \cdot \beta = 1,$$

i.e.

$$\alpha \equiv \beta$$

□

Given the fact that the exponent in conservation law is not only handy in coding but also able to describe the global density of the tree, determining this exponent becomes the key for us to describe the fractal-like property of the tree pattern. With existing rules of deciding this exponent, it make sense for the tree patterns to have an exponent to be any number between $[1, 3]$.

How to determine it specifically? One way is to Measure each branch of the tree pattern figure we just showed, and solve the relationship of mother branch and children brach for each individual branches, which is very tedious. Alternatively, we can also use the data of branches to fit the power law of number of branches and size of branch using MLE method. To do this, we need to relate the conservation law 1 to a power law governing the number of branches of a given size.

4.3. **Discrete MLE method and empirical observation.**

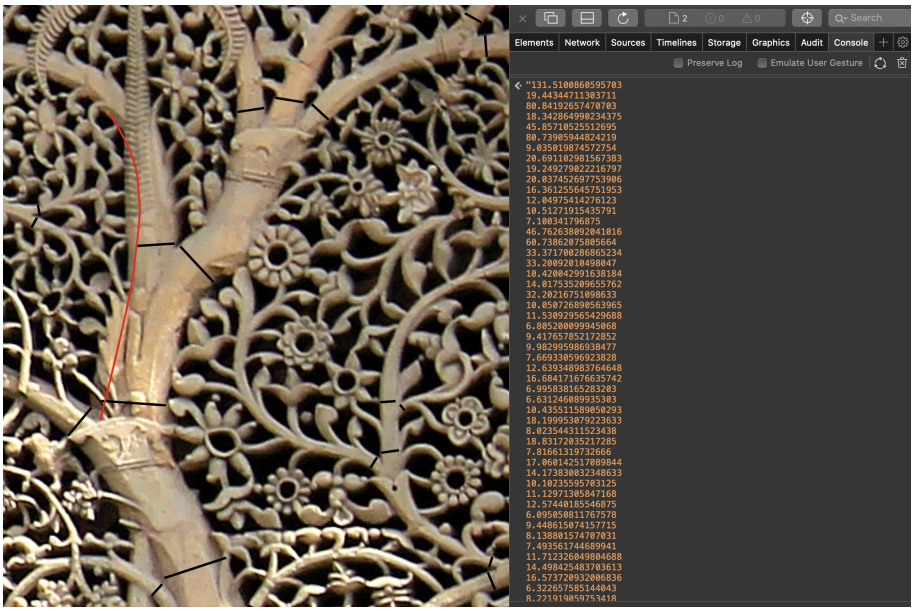


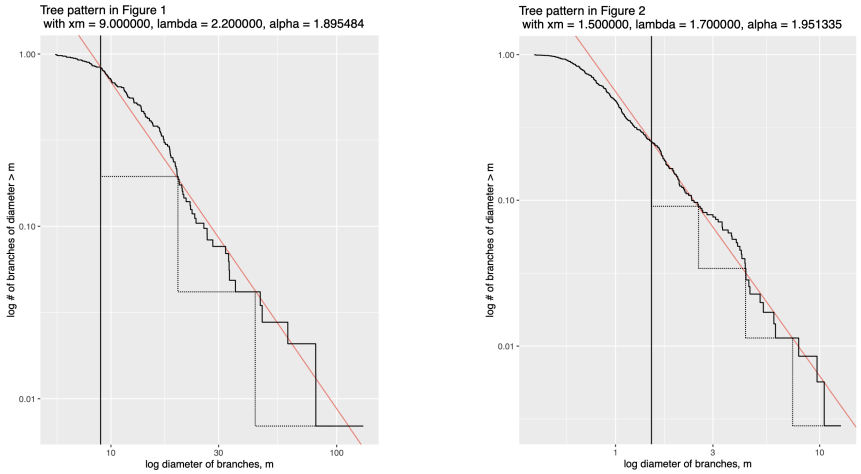
Figure 3. Collecting data

Step 1. As in Figure 3, first of all, we drew hundreds of line segments to measure the length of the diameter of each branches, and wrote a program to get the length of each line segments.

The principles of drawing line segments (i.e. deciding which limb counts for a branch) is as follow:

(1) the limb with only one leaf or flower on it doesn't count for a branch;
(2) a line segment is drew immediately after branching happened, except for the stem, i.e. the first branch;
(3) when the limb is covered by another limb, estimate its diameter by using bezier curve to recover the figure;
(4) a limb originates from a different root doesn't count for a branch.

Step 2. Then, we fed the dataset to the discrete MLE[NS19], which is developed to specifically deal with analyzing the scaling exponent of self-similar objects.



(a) log − log plot of tree pattern in Figure 1 **(b)** log − log plot of tree pattern in Figure 2

Figure 4. Selected log − log plots of tree pattern

In the log − log plot of tail distribution of tree pattern diameters, the $x-$ coordinate stands for the log of diameter of branches, and $y-$ coordinate stands for the log of number of branches of diameter greater than a value.
The parameter xm is the lower bound for which data less than xm are discarded since the power-law relationship cannot hold for these data [CSN09]. And λ is the scaling factor of self-similar object. In this case,

$$d_{parent} = \lambda \cdot d_{child} \tag{3}$$

Parameter $alpha$ is defined via $n = \lambda^{alpha}$, wherein n is the number of child branches a branch has, which is correspond to the definition of exponent that is able to reflect the global density of tree in Definition 4.1.

The solid stair-shaped line is drew by original data and the dashed line is drew by binned data, i.e. dataset ignoring relatively little noise of original data. In the figure, $\log \lambda$ also stands for the "width" of the stair step, since when we take log on each side of equation 3, we have

$$\log d_{parent} = \log \lambda + \log d_{child}$$

i.e.

$$\log d_{parent} - \log d_{child} = \log \lambda$$

The stair-shaped plot depict the discrete-scale property of the diameter dataset. Meanwhile, it also reflect whether the dataset holds a self-similar property by whether the "stair step is evenly height." It is because that if the original dataset has the self-similar property, then there exist a power law relationship between the number of

branches of diameter d_{parent}, denoted as $n_{d_{parent}}$, and the number of $k_{th}-$generation child branches of diameter d_{child}, denoted as $n_{d_{child}}$, i.e.

$$n_{d_{child}} = 2^k \cdot n_{d_{parent}}.$$

Thus, the "height" of a stair step is

$$\begin{aligned}
&\log(n_{(d>d_{child})}) - \log(n_{(d>d_{parent})}) \\
&= \log(\frac{n_{d_{child}} \cdot (1 - \frac{1}{2}^k)}{1 - \frac{1}{2}}) - \log(\frac{n_{d_{parent}} \cdot (1 - \frac{1}{2}^k)}{1 - \frac{1}{2}}) \\
&= \log 2 + \log(n_{d_{child}}) + \log(1 - \frac{1}{2}^k) \\
&\quad - (\log 2 + \log(n_{d_{parent}}) + \log(1 - \frac{1}{2}^k)) \\
&= \log(n_{d_{child}}) - \log(n_{d_{parent}}) \\
&= \log(2^k \cdot n_{d_{parent}}) - \log(n_{d_{parent}}) \\
&= k \cdot \log 2 + \log(n_{d_{parent}}) - \log(n_{d_{parent}}) \\
&= k \cdot \log 2
\end{aligned}$$

which is a constant, and thus, each stair step should be even if the dataset has the self-similar property.

By testing for different value of xm and λ, we found that the "stair steps" are almost evenly height but not perfectly, and thus, it is reasonable to claim that the tree pattern has self-similar property, and thus counts for a fractal-like set. Moreover, *alpha* always falls into $[1.8, 2.2]$ when xm and λ vary in reasonable range, suggests a relative stable and smaller range for the exponent.

Notice that this possible range of the scaling exponent of diameter, $[1.8, 2.2]$, is included in the intuitive interval of exponent in conservation law 1 of the tree pattern, $[1, 3]$ as mentioned in Section 4.2.3, which is consistent with the hypothesis in Conjecture 4.2 and suggesting that the scaling exponent of diameter and exponent in conservation law are closely related for the asymmetric tree patterns in Figure 1 and Figure 2.

5. Algorithm

Our code is public and on Gitlab. This is the link for those interested [GN].

5.1. **How to generate shapes recursively.** The ultimate goal of our project is to generate self-similar tree pattern on arbitrary shape of window. To start, let's talk about how to generate a self-similar tree using recursive function. There is a snippet of pseudocode to generate a very simple "v"

like tree, and my code to generate other complicated trees are similar just adding more other parameters.

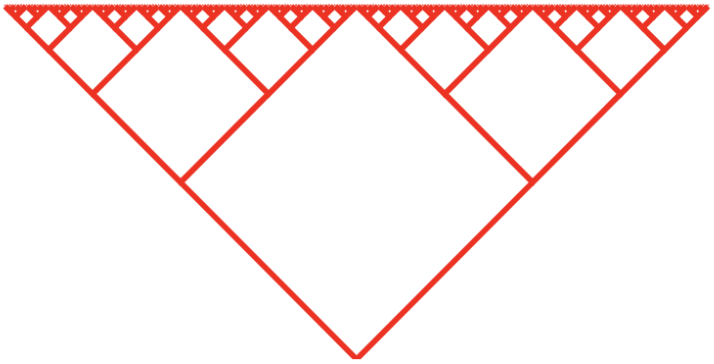


Figure 5. The V tree

Step 1. Create a recursive function with parameter x coordinate, y coordinate, branch length l.
Step 2. Check whether the length of the branch is less than 1/9. If it's true, break.
Step 3. Create a path object that draw a "v" with each branch of length 1/2 of the input length.
Step 4. Call the function on the corresponding x coordinate and y coordinate of the *left* end point of the "v" just drew.
Step 5. Call the function on the corresponding x coordinate and y coordinate of the *right* end point of the "v" just drew.

There are other trees that we generated, by adding the parameters to change the branching angles and diameters.

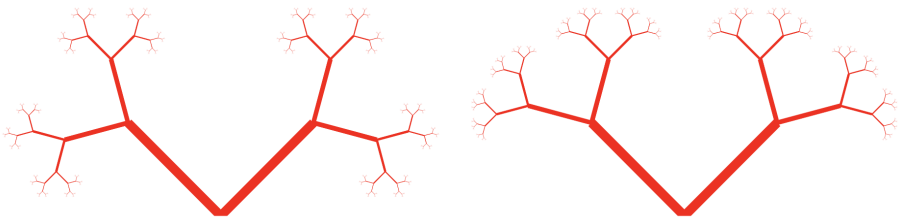


Figure 6. Trees we generated that are able to change the branching angles and diameters

5.2. **Generate a self-similar tree on an arbitrary convex polygon.** Our goal is to generate self-similar tree pattern on arbitrary shape of window. To simplify the model, we started from dealing with convex polygon. As in Figure 7, our algorithm is able to generate different self-similar tree pattern depends on different input of polygon. The polygon below the board is the shape of the input window, and the orange tree is the desired tree pattern. Other dots on the figure are points generated via computation process, which are preserved only for the purpose of future debugging and testing.

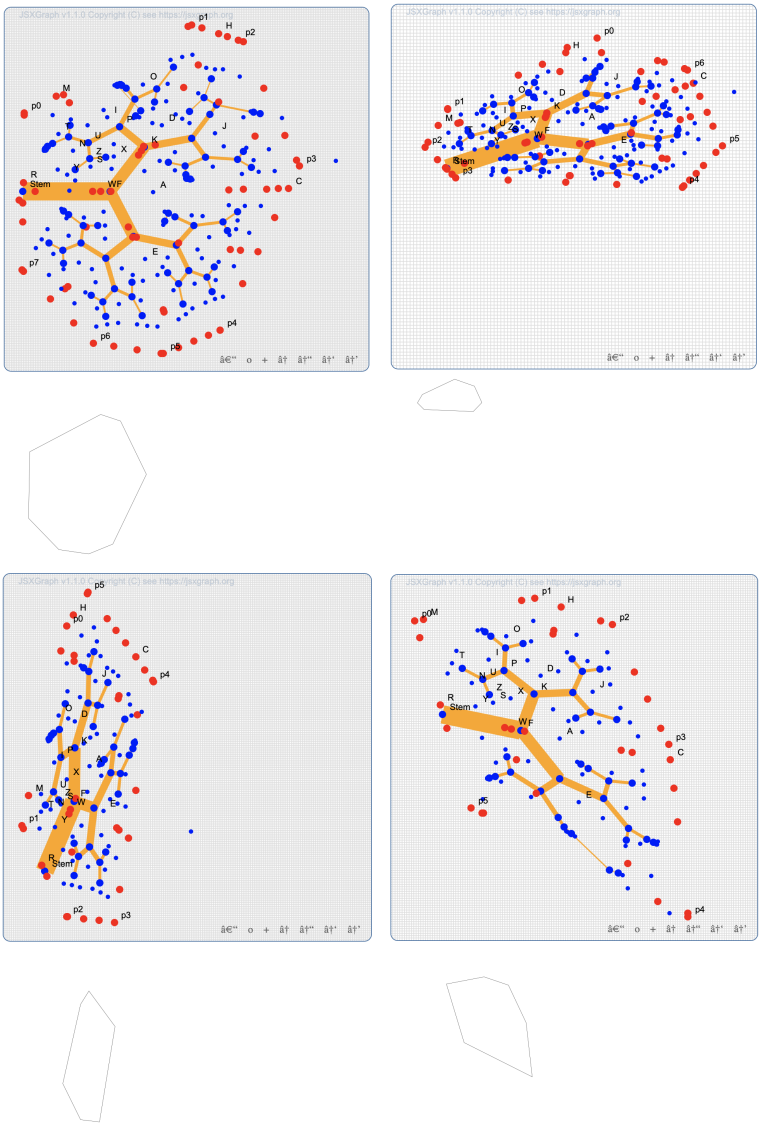


Figure 7. The algorithm is able to deal with any arbitrary convex polygon

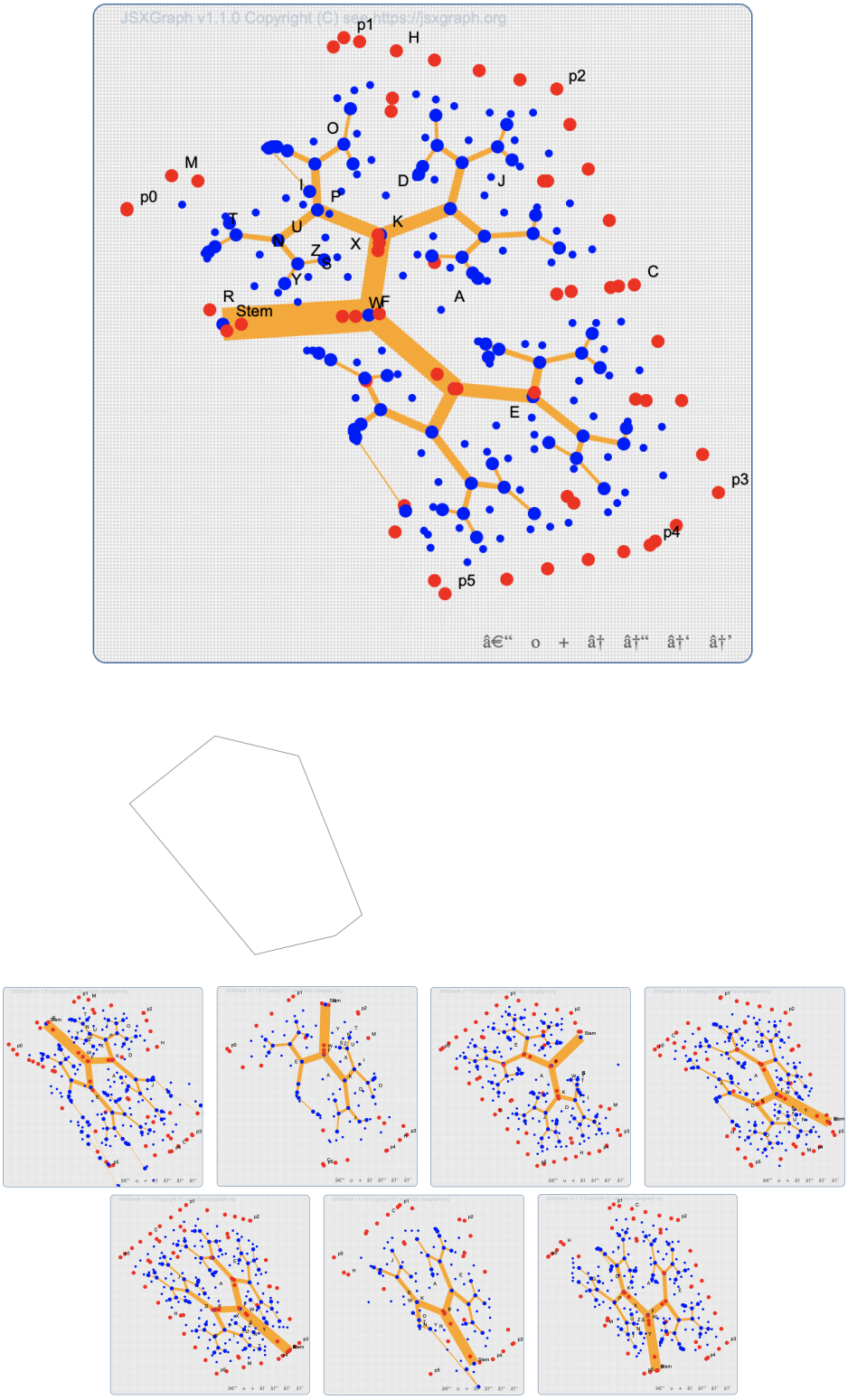


Figure 8. The algorithm is able to cater to different choices of initial position of the tree's stem

What's more, for an identical shape, our algorithm is also able to cater to different choice of initial position of the tree's stem to generate different trees. As shown in Figure 8.

The algorithm basically works as follows:

Step 1. Setup: draw a convex polygon in Inkscape, an open source vector graphics software, as a SVG, aka scalable vector graphics, path object, and get the "d" attribute of the path, which depict the coordinate of vertices of the polygon. If it is depicted by using relative coordinate system (i.e. starting with lower case m), use the convertToAbsolute function to convert it into absolute coordinate system. Use the information of vertices to create a polygon object in JSXGraph[JSX], a JavaScript library;

Step 2. Bisect polygon: set the stem point, $Stem$ (i.e. the origin of tree set by user on the edge of polygon), as one endpoint, and find another endpoint on the edge of the polygon such that the line segment between these two points bisect the polygon. Denote the resulting two polygons as $\mathcal{P}_{left}$ and $\mathcal{P}_{right}$.

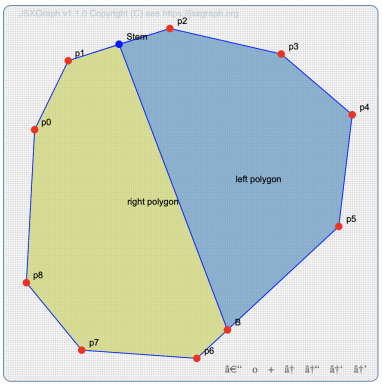


Figure 9. Bisect polygon

Step 3. Find centroid: find the centroid, aka center of mass, of $\mathcal{P}_{left}$ and $\mathcal{P}_{right}$ respectively, denoted as C_l and C_r.

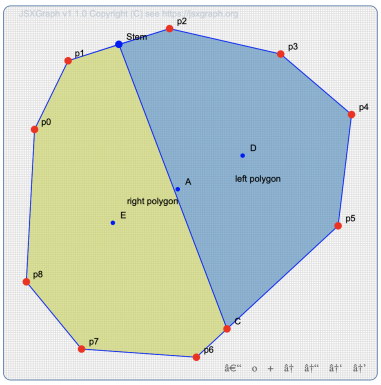


Figure 10. Find the centroid of left polygon, D, and right polygon, E, respectively.

Step 4. Create branch: pick $Stem$, C_l, and C_r as vertices to formulate a triangle. Find the centroid of this triangle, denoted as $newStem$, literally play the role of the new stem point of C_l and C_r.

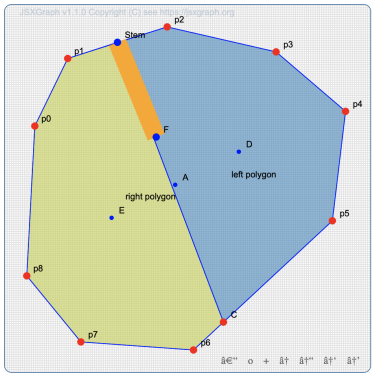


Figure 11. Create branch

Remark 5.1. $newStem$ is not always going to lie on the line segment that bisect the polygon in Step 2, i.e. it is not always going to lie on the edge of C_l and C_r. This is because lines through centroid are not always going to bisect the polygon.
As in Figure 15, A is the centroid of polygon $p0 - p2 - p3 - p4 - p5 - p6 - p7 - p8 - p0$. $\overline{Stem\,C}$ is the line through $Stem$ and bisect the polygon. However A doesn't lie on that line segment.

Step 5. Recursive step: call the function to do Step 1 $\sim$ Step 4 on $\mathcal{P}_{left}$ and $\mathcal{P}_{right}$ using $newStem$ recursively, until the area of current polygon reach the lower bound.

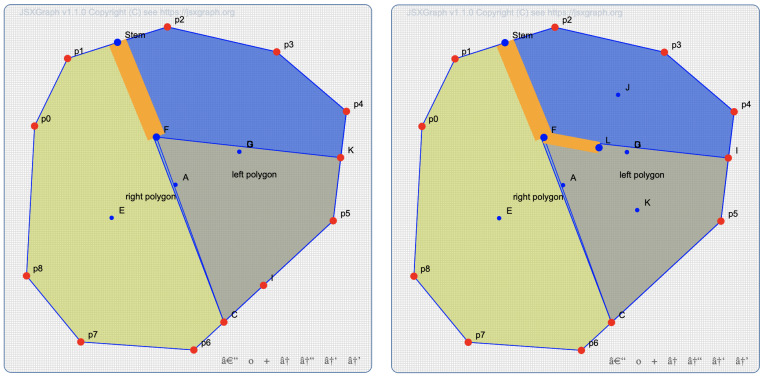


Figure 12. Recursive step (one step on $\mathcal{P}_{left}$)

5.3. **Implementation issues.** There are two parts in the algorithm that can be improved. Firstly, as mentioned in 5.1, since $newStem$ is not always going to lie on the edge of new polygons, the resulting polygon might become non-convex when this error accumulated when the recursive depth become deeper and deeper. Consequently, the centroid of the non-convex polygon might fall outside of the "window," which failed to accomplish the original goal. Secondly, the triangle as an edge case of polygon hasn't been taken into account yet, which might also explain the inability of the algorithm to generate branches when the allowed smallest of current polygon becoming smaller.

6. Acknowledgements

I would like to thank my mentor Mitchell Newberry for his assistance through the research and the initial development of the conjecture. I would also like to thank to Math and Complex System department at the University of Michigan for presenting this opportunity.

References

[CSN09] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[Elo11] Christophe Eloy. Leonardo's rule, self-similarity, and wind-induced stresses in trees. *Physical review letters*, 107(25):258101, 2011.

[Fal04] Kenneth Falconer. *Fractal geometry: mathematical foundations and applications*. John Wiley & Sons, 2004.

[GN] Jingyi Gao and Mitchell Newberry. Gitlab of REU Generating a fractal tree. `https://gitlab.com/jingyig/reu_generating_a_fractal_tree`.

[JSX] JSXGraph. Jsxgraph. `https://github.com/jsxgraph`.

[Li] Yuelin Li. "fractal expression" in ancient chinese calligraphy.

[LS07] Peter J Lu and Paul J Steinhardt. Decagonal and quasi-crystalline tilings in medieval islamic architecture. *science*, 315(5815):1106–1110, 2007.

[Man83] Benoit B. Mandelbrot. *The fractal geometry of nature*. New York, W.H. Freeman and Co., 1983. Rev. ed. of: Fractals. c1977.

[Mur26] Cecil D Murray. The physiological principle of minimum work: I. the vascular system and the cost of blood volume. *Proceedings of the National Academy of Sciences of the United States of America*, 12(3):207, 1926.

[New05] Mark EJ Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[NS19] Mitchell G Newberry and Van M Savage. Self-similar processes follow a power law in discrete logarithmic space. *Physical review letters*, 122(15):158303, 2019.

[Str18] Steven H Strogatz. *Nonlinear dynamics and chaos with student solutions manual: With applications to physics, biology, chemistry, and engineering*. CRC press, 2018.

[SWC+15] Jian Shang, Yongfeng Wang, Min Chen, Jingxin Dai, Xiong Zhou, Julian Kuttner, Gerhard Hilt, Xiang Shao, J Michael Gottfried, and Kai Wu. Assembling molecular sierpiński triangle fractals. *Nature chemistry*, 7(5):389–393, 2015.