

# Software to Produce Fractal Tilings of the Plane

Michael Arthur Mampusti

February 28, 2014

In this project, we aimed to produce a program capable of creating fractal tilings of the plane. We did this through the programming package *Mathematica*. In this report we will discuss what tilings of the plane are, some fractal geometry and finally go through step by step how we went about creating fractal tilings of the plane.

## 1 Tilings of the Plane

A *tiling of the plane*, intuitively speaking, is much like a tiling by squares of a bathroom floor. If you think of the floor as  $\mathbb{R}^2$ , then the square tiles which cover it is a tiling of the plane. We make this definition more concrete.

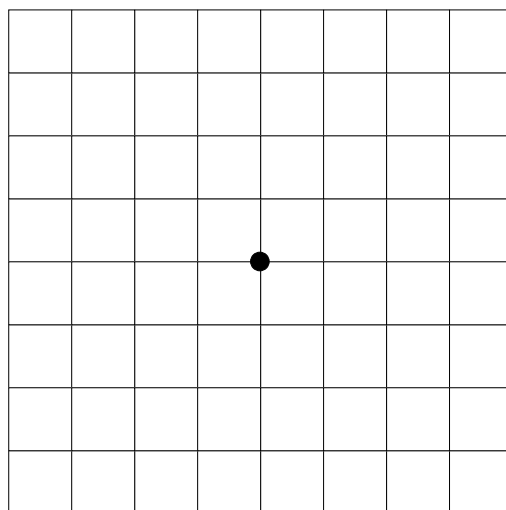


Figure 1: Tiling of the bathroom floor

**Definition 1.** A *prototile*  $p$  is a labeled compact subset of  $\mathbb{R}^2$  containing the origin.

Prototiles are the building blocks of tilings of the plane. A prototile consists of a compact subset of the plane and a label.

At this point, it is useful to define some notation. Given a prototile  $p$ , we denote the translation of  $p$  by a vector  $x \in \mathbb{R}^2$  by  $p + x$ . Furthermore, given some  $\theta \in \mathbb{R}$ , we denote the anticlockwise rotation of  $p$  about the origin by  $\theta$ ,  $R_\theta p$ .

**Definition 2.** Let  $\mathcal{P}$  be a set of prototiles. A *tiling of the plane* is a set of compact subsets  $T := \{t_i : i \in \mathbb{N}\}$ , where each  $t_i$  is a rigid motion of a prototile in  $\mathcal{P}$ . That is, for each  $t_i$ , there exists  $p \in \mathcal{P}$ ,  $\theta \in \mathbb{R}^2$  and  $x \in \mathbb{R}^2$  such that  $t_i = R_\theta p + x$ . Each  $t_i$  is called a *tile*. Furthermore, we require that the following two conditions hold.

1. The tiles cover  $\mathbb{R}^2$ , that is,

$$\bigcup_{i \in \mathbb{N}} t_i = \mathbb{R}^2; \text{ and}$$

2. For each  $i \neq j$ ,

$$\text{int}(t_i \cap t_j) = \emptyset.$$

That is, the intersection of two distinct tiles have empty interior.

In this way we see that the bathroom floor covered with square tiles is a tiling of the plane if the floor and the tiles were extended out to infinity. The tiles cover the floor without any gaps and the tiles do not overlap each other. The set of prototiles for the bathroom floor consists of the one white square. This is because every other tile in the tiling is simply a translation of the white square.

**Remarks 3.** It is worth noting that each element in the set of prototiles is used in the tiling. Each tile in the tiling is a rigid motion of one of the prototiles and for each prototile  $p$  there exists a tile in the tiling which is a rigid motion of  $p$ .

Furthermore, it is clear that every tiling is infinite. This is not the case with the set of prototiles. It is possible that we have a finite or infinite set of prototiles. For this project we only considered tilings made from finitely many prototiles.

**Definition 4.** Given a tiling  $T$ , a *patch* is a finite collection of tiles in  $T$ .

For example, Figure 1 is a patch of the square tiling of the plane. All the figures of tilings throughout this report are patches of tilings.

An important operation that we can apply to tilings is translation. For any tiling  $T$ , we denote  $T + x := \{t_i + x : i \in \mathbb{N}\}$  to be the translation of each of the tiles in  $T$  by the vector  $x \in \mathbb{R}^2$ . This gives rise to another tiling of the plane.

**Definition 5.** A tiling  $T$  is *periodic* if there exists  $x \in \mathbb{R}^2$  such that  $T + x = T$ . If there does not exist such an  $x$ , then we say that  $T$  is *aperiodic*.

The square tiling of the bathroom floor is an example of a periodic tiling since translating the tiling by the side length of a tile horizontally or vertically gives us back the same tiling. For this project, we mainly worked with tilings that are aperiodic. We will introduce some examples of aperiodic tilings in the next section.

## 1.1 Substitution Tilings

An important class of tilings are *substitution tilings*. These are tilings which can be described using *substitution rules* on the set of prototiles. Carefully placing a single prototile at the origin and applying these substitution rules ad in finitum, yields tilings of the plane.

**Definition 6.** Let  $T$  be a tiling and  $\mathcal{P}$  be the set of prototiles for  $T$ . A *substitution rule* on  $\mathcal{P}$ , is made up of:

1. a scaling factor,  $\lambda > 1$ ;
2. a rule  $\omega$  such that, for each  $p \in \mathcal{P}$ ,
  - (a)  $\omega(p)$  is a patch of tiles  $\{t_1, \dots, t_k\}$  with support  $\lambda p$ ;
  - (b) each tile in  $\omega(p)$  is a rigid motion of a prototile in  $\mathcal{P}$ ; and
  - (c)  $\text{int}(t_i) \cap \text{int}(t_j) = \emptyset$  for  $i \neq j$ .

To understand how these substitution rules work, consider the square bathroom floor example. This is an example of a substitution tiling. We have one prototile; the square. To apply the substitution rule, we scale the prototile by a factor of two, that is,  $\lambda = 2$ . We then tile this scaled tile with four of the original squares. Notice that the four squares is supported by the scaled square and that each of the four tiles are translations of the prototile. This process is illustrated in Figure 2.

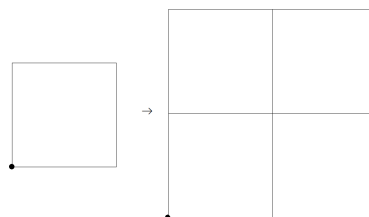


Figure 2: Substitution rule for the tiled bathroom floor

**Example 7** (Penrose Tiling). This is an example of a substitution tiling which is also aperiodic. This tiling consists of four prototiles. In fact, only two up to reflections. In Figure 3, we have the four prototiles (red, blue, yellow and green) and the substitution rules on each. The scaling factor of this substitution tiling is the golden ratio  $\phi$ . Each of the prototiles are inflated and then tiled using the set of prototiles. After applying the substitution rule to the red tile four times, we obtain Figure 4.

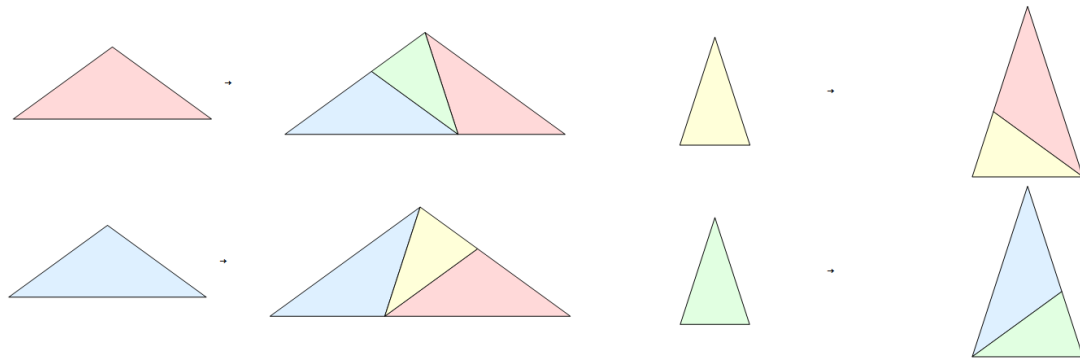


Figure 3: The Penrose substitution rules

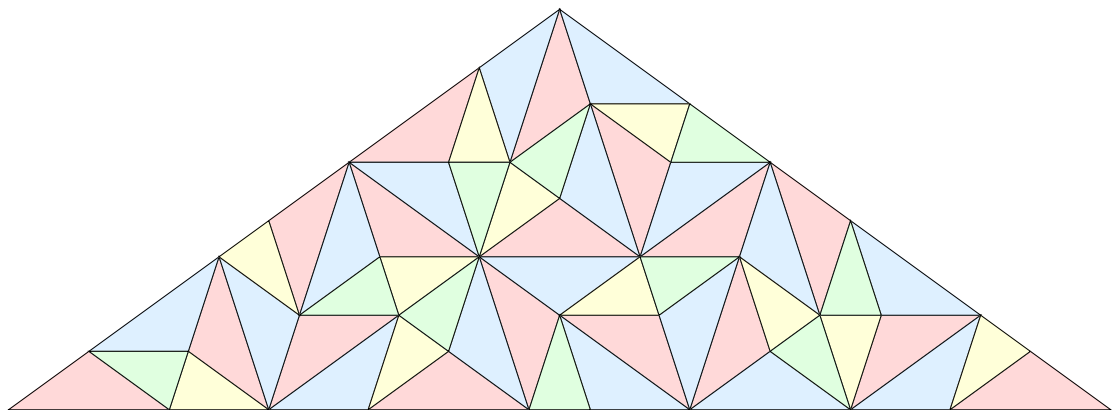


Figure 4: A patch of the Penrose tiling

**Example 8** (Ammann-Beenker Octagonal Tiling). Another substitution tiling which also happens to be an aperiodic tiling of the plane. It consists of three prototiles. These prototiles and their substitutions are found in Figure 5 and a patch of the Ammann-Beenker Octagonal tiling is found in Figure 6. The scaling factor is  $\lambda = 1 + \sqrt{2}$ .

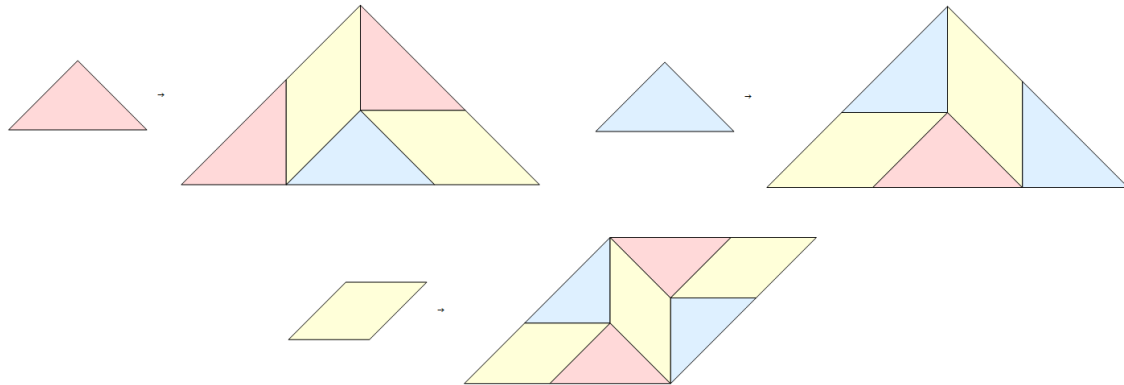


Figure 5: The Ammann-Beenker Octagonal substitution rules

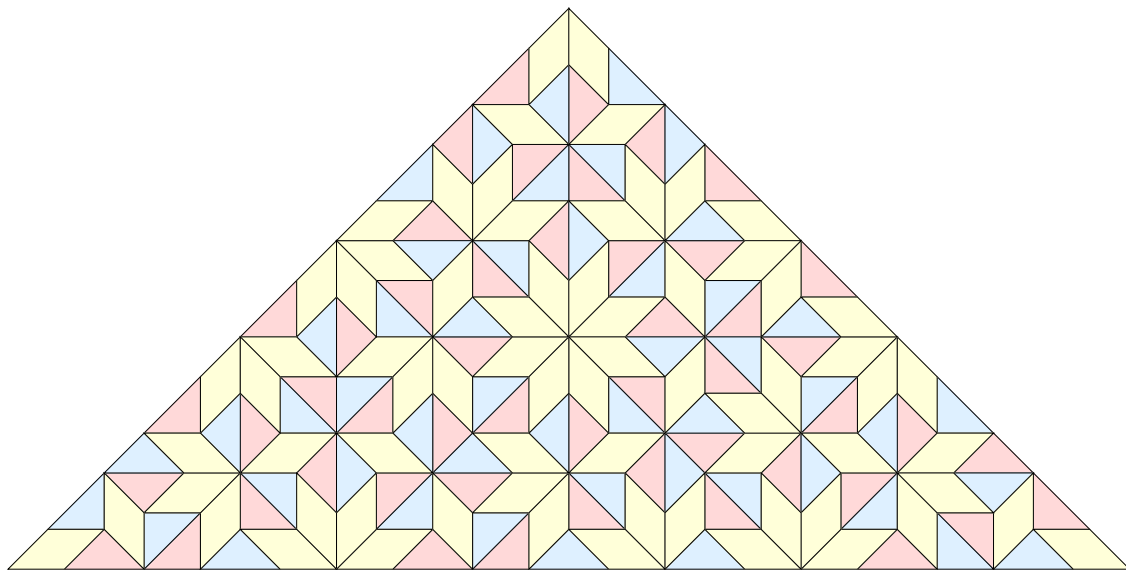


Figure 6: Patch of the Ammann-Beenker Octagonal tiling

### 1.1.1 Programming Substitution Tilings

The first part of this project involved programming already known substitution tilings in *Mathematica*. This involved thinking up a neat way in which to carry out the substitution given the set of prototiles and the substitution rules.

**Problem 1.** How do we store our set of prototiles?

For each prototile, we stored the coordinates for each vertex. For example, the red prototile in the Penrose tiling was stored as

$$\left\{ \{0, 0\}, \{\phi, 0\}, \left\{ \frac{\phi}{2}, \sqrt{1 - \left(\frac{\phi}{2}\right)^2} \right\}, \{0, 0\} \right\},$$

along with other information such as labels and colour.

We then created a vector of prototiles. For the Penrose tiling, this meant a vector of four prototiles. We will see soon why this is useful.

**Problem 2.** How do we apply the substitution rules on the prototiles?

To apply the substitution rules we developed a matrix, called *The Tile Breaker*, which when applied to the vector of prototiles, gives us back the substituted versions. The Tile Breaker is a matrix where each entry is a set of pairs,  $(\theta, x)$ , where  $\theta \in \mathbb{R}$  is a rotation about the origin and  $x \in \mathbb{R}^2$  is a translation. Entry  $(i, j)$  is the set of rigid motions of prototile  $j$  in the substitution of prototile  $i$ . Thus, the  $i$ th row of The Tile Breaker, tells us the substitution rule for prototile  $i$  or how to “break up” prototile  $i$ . If there are  $n$  prototiles, then The Tile Breaker is an  $n \times n$  matrix. The Penrose Tile Breaker is shown in Figure 7.

$$\begin{pmatrix} \left\{ \left\{ \frac{4\pi}{5}, \{\phi^2, 0\} \right\} \right\} & \{\{0, \{0, 0\}\}\} & \{\} & \left\{ \left\{ -\frac{4\pi}{5}, \left\{ \frac{\phi^2}{2}, \phi \sqrt{1 - \left(\frac{\phi}{2}\right)^2} \right\} \right\} \right\} \\ \{\{0, \{1, 0\}\}\} & \left\{ \left\{ -\frac{4\pi}{5}, \left\{ \frac{\phi^2}{2}, \phi \sqrt{1 - \left(\frac{\phi}{2}\right)^2} \right\} \right\} \right\} & \left\{ \left\{ \frac{4\pi}{5}, \left\{ 1 + \frac{\phi}{2}, \sqrt{1 - \left(\frac{\phi}{2}\right)^2} \right\} \right\} \right\} & \{\} \\ \left\{ \left\{ \frac{3\pi}{5}, \{1, 0\} \right\} \right\} & \{\} & \left\{ \left\{ -\frac{3\pi}{5}, \left\{ \frac{\phi-1}{2\phi+1}, \frac{\sqrt{\phi+2}}{2\phi+1} \right\} \right\} \right\} & \{\} \\ \{\} & \left\{ \left\{ -\frac{3\pi}{5}, \left\{ \frac{1}{2}, \frac{\sqrt{\phi+3}}{2} \right\} \right\} \right\} & \{\} & \left\{ \left\{ \frac{3\pi}{5}, \{1, 0\} \right\} \right\} \end{pmatrix}$$

Figure 7: The Tile Breaker for the Penrose tiling

**Problem 3.** How do we apply The Tile Breaker to the vector of prototiles?

Applying The Tile Breaker to the vector of prototiles follows the same idea as, say, multiplying a vector in  $\mathbb{R}^n$  by a real  $n \times n$  matrix. However, instead of getting a vector back, we get a new matrix. Entry  $(i, j)$  in this matrix is a set containing:

1. the set of vertices of prototile  $j$ ; and
2. the set of rigid motions from entry  $(i, j)$  in the Tile Breaker.

Thus, entry  $(i, j)$  of the resultant matrix describes prototile  $j$  as well as its rigid motions in the substitution of prototile  $i$ .

**Problem 4.** How do we get the  $n$ th iteration of the substitution?

To get the first iteration of the substitution, we simply follow the process described in Problem 3. For  $n > 1$ , we must do something different. At first glance, to obtain the  $n$ th substitution, it seems that we could simply multiply the vector of prototiles by The Tile Breaker  $n$  times. This is not the case. After the first application of The Tile Breaker, the resulting output is no longer a vector, but a matrix. Instead, we raise The Tile Breaker to the  $n$ th power and then apply it to our vector of prototiles.

To raise The Tile Breaker to the  $n$ th power, we must make sense of the multiplication of matrices of this form. To do this we must define the multiplication of two entries in the Tile Breaker. Let  $\lambda$  be the scaling factor of our substitution rule and suppose we have  $m$  prototiles. Let  $A$  be our Tile Breaker and  $A_{ij}$  and  $A_{kl}$  denote entries in  $A$ . We define the following multiplication,

$$A_{ij} \cdot A_{kl} := \{ \{ \theta + \psi, R_\theta y + \lambda x \} : \{ \theta, x \} \in A_{ij}, \{ \psi, y \} \in A_{kl} \}. \quad (1)$$

Then the square of matrix  $A$  is given by,

$$(A \cdot A)_{ij} := \bigcup_{k=1}^m A_{ik} \cdot A_{kj}.$$

Given this multiplication, we can now raise The Tile Breaker to the  $n$ th power. That is, take your new matrix  $A^2$  and keep multiplying on the right by  $A$  until you reach  $A^n$ . Then it is simply a matter of applying the resultant matrix to the vector of prototiles to obtain the  $n$ th substitution as we did in Problem 3. This time, however, entry  $(i, j)$  of the resultant matrix describes prototile  $j$  as well as its rigid motions in the  $n$ th substitution of prototile  $i$ .

**Problem 5.** After multiplying our vector of prototiles with the Tile Breaker, how do we then convert this into pictures as in Figures 4 and 6?

Suppose we want to look at the  $n$ th substitution of the  $i$ th prototile. We would raise the Tile Breaker to the  $n$ th power and apply it to our vector of prototiles. Then the  $i$ th row contains all the prototiles and their respective rigid motions in the  $n$ th substitution of prototile  $i$ . Now, the  $j$ th entry in the  $i$ th row contains the vertices of prototile  $j$  as well as a set of rigid motions. For each element in the set of rigid motions, apply it to the vertices of prototile  $j$ . Convert each of these into a graphic via the *Mathematica* function `Polygon` and plot them on the same graph. We do this for each entry in the  $i$ th row. This is the  $n$ th substitution of the  $i$ th prototile.

**Remark 9.** When drawing these substitution tilings, we chose to apply the rotation followed by the translation. We could have similarly chosen to do the translation before the rotation. Had we chosen the latter, The Tile Breaker for the Penrose tiling would not be the same as in Figure 7 and the multiplication in (1) would need to be modified.

## 2 Fractals

Now that we have looked at tilings of the plane, to understand fractal tilings of the plane, we must first delve into fractal geometry. As we are only considering fractal tilings of the plane, we will only discuss fractals from  $\mathbb{R}^2$ .

We consider  $\mathbb{R}^2$  as a metric space with the Euclidean metric. That is,

$$d(x, y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2},$$

where  $x, y \in \mathbb{R}^2$ . This is a complete metric space in the sense that all Cauchy sequences in  $\mathbb{R}^2$  converge in the Euclidean metric. Furthermore, consider the space of compact subsets of  $\mathbb{R}^2$ , denoted by  $\mathcal{H}(\mathbb{R}^2)$ . This is also a complete metric space where the metric is the Hausdorff metric. It is given by,

$$d_{\mathcal{H}}(A, B) = \max\{\min\{d(x, y) : y \in B\} : x \in A\},$$

where  $A, B \in \mathcal{H}(\mathbb{R}^2)$ . Since  $A, B \in \mathcal{H}(\mathbb{R}^2)$ , the metric is well defined. What the metric is doing is finding for each fixed  $x \in A$  the shortest distance to  $B$ , in the Euclidean metric, then taking the largest of these distances. The space  $\mathcal{H}(\mathbb{R}^2)$  is important as it is where our fractals come from.



**Definition 10.** Let  $X$  be a metric space. A map  $f : X \rightarrow X$  is a *contraction on  $X$*  if there exists  $0 < \lambda \leq 1$  such that,

$$d(f(x), f(y)) \leq \lambda d(x, y),$$

for any  $x, y \in X$ .

Consider the map  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  defined by  $f(x) = \frac{x}{3}$ . It is a contraction on  $\mathbb{R}^2$  since

$$d(f(x), f(y)) = d\left(\frac{x}{3}, \frac{y}{3}\right) = \frac{1}{3}d(x, y) \leq \lambda d(x, y),$$

where  $\frac{1}{3} \leq \lambda \leq 1$  for all  $x, y \in \mathbb{R}^2$ .

**Definition 11.** Let  $f_1, \dots, f_n$  be contractions on  $\mathbb{R}^2$ . Then  $\mathbb{R}^2$  with these contractions is an *iterated function system* or an *IFS*.

To define what we consider a fractal, we will need to state the following theorem.

**Theorem 12** (Theorem 7.1, [1]). *Let  $\mathbb{R}^2$  along with  $f_1, \dots, f_n$  be an IFS. Consider the function  $f : \mathcal{H}(\mathbb{R}^2) \rightarrow \mathcal{H}(\mathbb{R}^2)$  defined by,*

$$f(B) := \bigcup_{i=1}^n f_i(B),$$

where  $B \in \mathcal{H}(\mathbb{R}^2)$ . Then,  $f$  is a contraction on  $\mathcal{H}(\mathbb{R}^2)$ . Furthermore, there exists a unique fixed point of  $f$ ,  $A \in \mathcal{H}(\mathbb{R}^2)$ , that is,  $f(A) = A$ , and for any  $B \in \mathcal{H}(\mathbb{R}^2)$ ,

$$A = \lim_{n \rightarrow \infty} f^n(B).$$

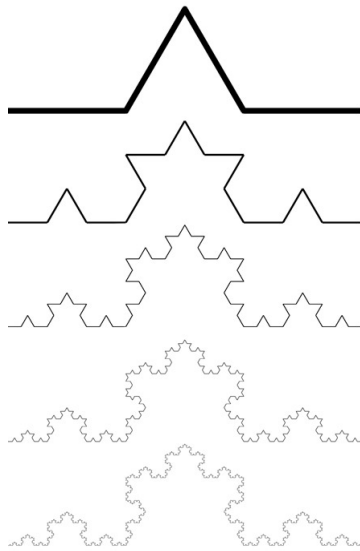
The fixed point  $A \in \mathcal{H}(\mathbb{R}^2)$  is known as the *attractor* of the IFS. For us, the attractor of an IFS will be our definition of a fractal.

**Definition 13.** A *fractal* is an attractor of an IFS.

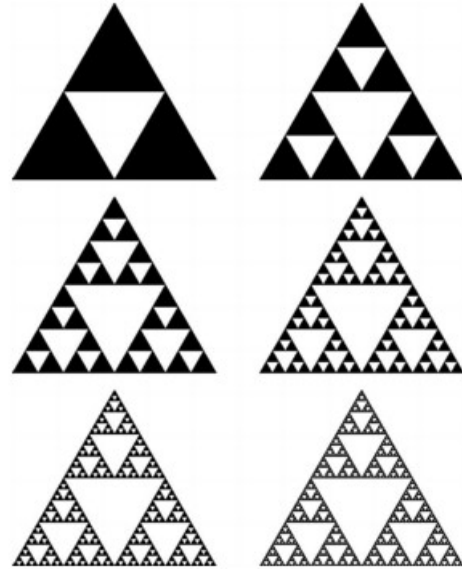
**Remark 14.** This definition of a fractal will suit our purposes, but it may not fit in with other definitions of a fractal. For example, self-similarity is a property of fractals which we have not mentioned.

## 2.1 Examples

In this section, we will look at two popular examples of fractals, the Koch curve and the Sierpinski gasket, both of which can be described as attractors for certain IFSs.



(a) The Koch Curve



(b) The Sierpinski Gasket

Figure 8: Examples of fractals

**Example 15** (The Koch curve). The Koch curve, seen in Figure 8a, is a fractal built from four contractions of  $\mathbb{R}^2$ . They are given by,

$$\begin{aligned}
 f_1(x_1, x_2) &= \left( \frac{x_1}{3}, \frac{x_2}{3} \right); \\
 f_2(x_1, x_2) &= \left( \frac{x_1 - x_2\sqrt{3} + 3}{6}, \frac{x_1\sqrt{3} + x_2 + \sqrt{3}}{6} \right); \\
 f_3(x_1, x_2) &= \left( \frac{x_1 + x_2\sqrt{3} + 2}{6}, \frac{-x_1\sqrt{3} + x_2}{6} \right); \text{ and} \\
 f_4(x_1, x_2) &= \left( \frac{x_1}{3} + \frac{2}{3}, \frac{x_2}{3} \right).
 \end{aligned}$$

Each of these functions involve scaling by a third. This is why they are contractions. Functions  $f_2$  and  $f_3$  also involve a rotation of  $\frac{\pi}{3}$  anticlockwise and clockwise, respectively. We then build the function  $f : \mathcal{H}(\mathbb{R}^2) \rightarrow \mathcal{H}(\mathbb{R}^2)$  defined by,

$$f(B) := \bigcup_{i=1}^4 f_i(B),$$

for all  $B \in \mathcal{H}(\mathbb{R}^2)$ . By Theorem 12, there exists a unique fixed point  $A \in \mathcal{H}(\mathbb{R}^2)$  which is the Koch curve.

**Example 16** (The Sierpinski gasket). The Sierpinski gasket, see in Figure 8b, is a fractal built from three contractions of  $\mathbb{R}^2$ . They are given by,

$$\begin{aligned} f_1(x_1, x_2) &= \left( \frac{x_1}{2}, \frac{x_2}{2} \right) \\ f_2(x_1, x_2) &= \left( \frac{x_1}{2} + \frac{1}{2}, \frac{x_2}{2} \right) \\ f_3(x_1, x_2) &= \left( \frac{x_1}{2} + \frac{1}{4}, \frac{x_2}{2} + \frac{\sqrt{3}}{4} \right). \end{aligned}$$

As was the case with the Koch curve, each of these functions involve scaling by a half. Again, by Theorem 12, the function,  $f : \mathcal{H}(\mathbb{R}^2) \rightarrow \mathcal{H}(\mathbb{R}^2)$  defined by,

$$f(B) = \bigcup_{i=1}^3 f_i(B),$$

for  $B \in \mathcal{H}(\mathbb{R}^2)$ , has a unique fixed point,  $A \in \mathcal{H}(\mathbb{R}^2)$ , being the Sierpinski gasket.

### 3 Fractal Tilings of the Plane

We have looked at tilings of the plane and have been through some basics in fractal geometry. We are now able to discuss fractal tilings of the plane. These are simply tilings of the plane where each tile has fractal edges. These fractal edges are the attractors for a generalized IFS. The main goal of this project was to create software which generates fractal tilings of the plane.

### 3.1 The Method

Our fractal tilings are built from already known substitution tilings. We will go through the process we went through to do this and in the next section, we will see how we implemented it in *Mathematica*. We use a specific example to illustrate our process.

The Half-Hex tiling is a tiling of the plane consisting of one prototile. Figure 9 shows the prototile on the left as well as the substitution rule.

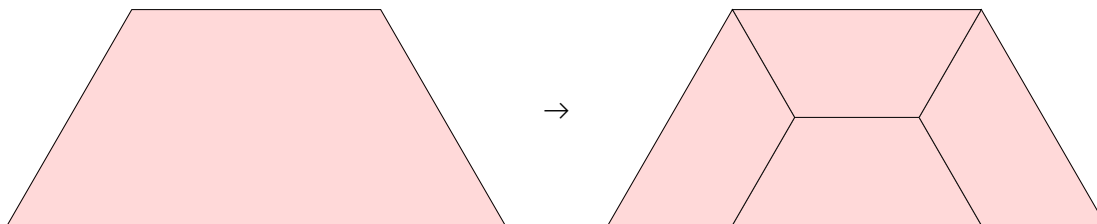


Figure 9: The Half-Hex tiling prototile

We begin by inscribing a graph in each prototile, called the *dual graph*. This means we have a central vertex in each prototile and an edge from the central vertex to the centre of each side. For the Half-Hex prototile we have four sides which means we have four edges, these are the blue edges in Figure 10.

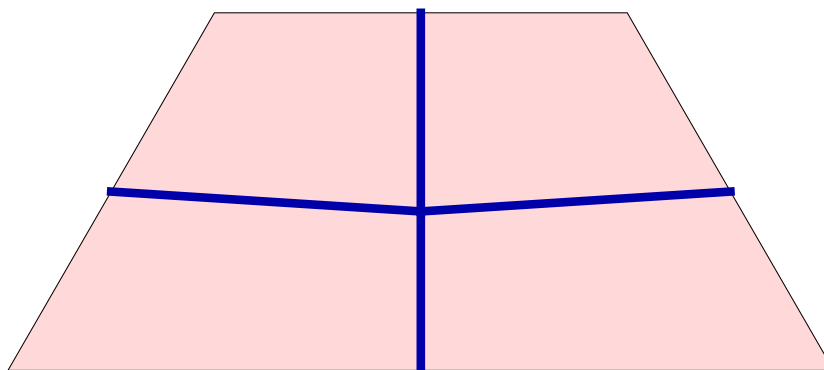


Figure 10: Half-hex dual graph

We then take some number of substitutions of the original substitution tiling, but with the graphs inscribed each tile. For this example, we will use two substitutions.

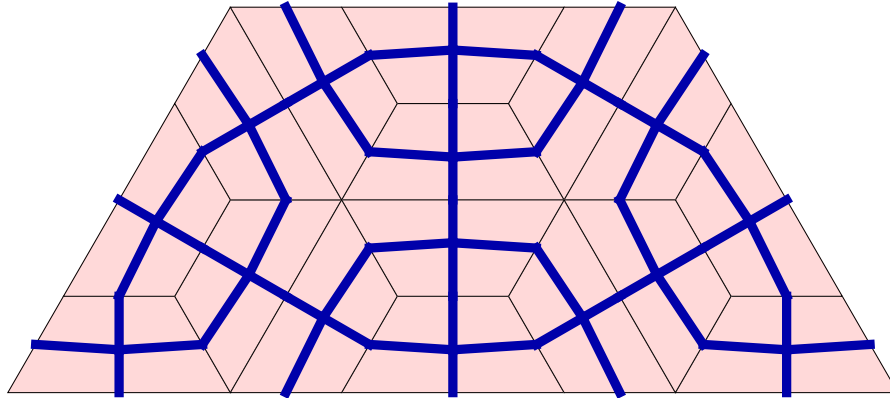


Figure 11: Half-hex prototile twice substituted with graphs inscribed

From here, we delete edges from this substitution so that the remaining edges are graph homeomorphic to the original dual graph. In our example, we delete edges so that the remaining edges correspond to a central vertex with four edges going to each side. Figure 12 gives one possible choice of edges. Each set of edges coming from the central vertex corresponds to an edge in the original dual graph.

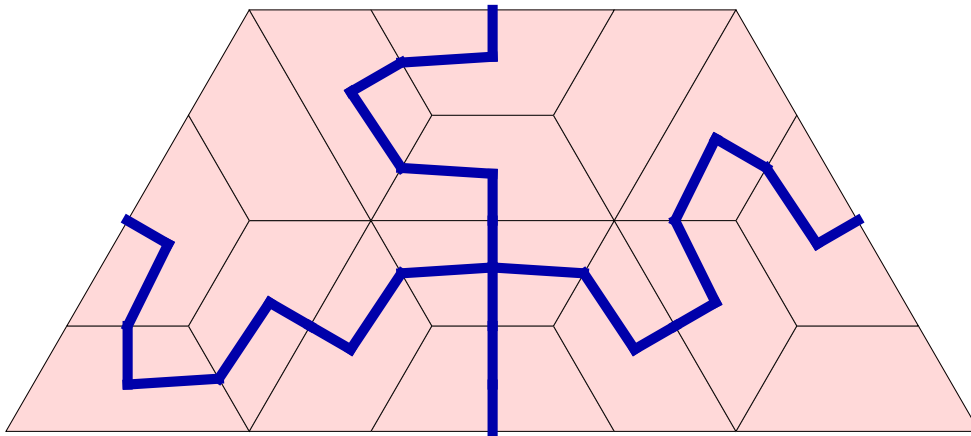


Figure 12: Choosing edges in the half-hex prototile

Following this, we iterate the edges infinitely. What we have here is something like a substitution rule, except defined on edges. Let us label the edges in the original graph, Figure 10, by numbers 1 to 4 starting from the edge going to the left and then

proceeding clockwise. In a similar way, label the edges in Figure 12 with numbers 5 to 8. Then the substitution is done by replacing every edge 1 with edge 5, every edge 2 with edge 6, every edge 3 with edge 7 and every edge 4 with edge 8. This is exactly what we did to get from Figure 10 to Figure 12. To further illustrate our point, let us observe what happens when we iterate one more time on edge 5.

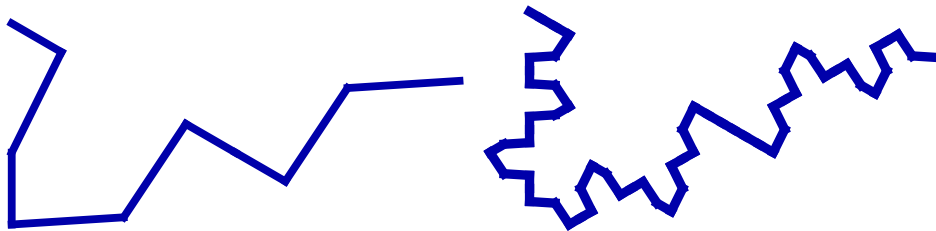


Figure 13: Edge 5 substitution

In figure 13, we have on the left, edge 5, and on the right, the substitution of edge 5. To get from left to right, notice that edge 5 is made up of a combination of edges 1 to 4. Thus, we apply the substitution to each of these parts. This process is done infinitely. However, in practice, this is obviously not possible.

After a few iterations of the edges, we shrink them down and inscribe them within the original prototiles. For our example, we have done this in Figure 14.

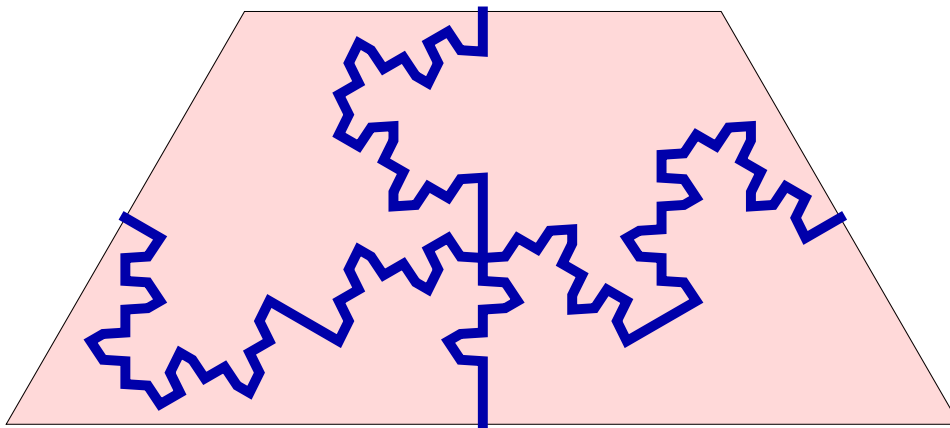


Figure 14: Final fractal edges

We then tile the plane using the original substitution tilings with the fractal edges inscribed within the tiles as in Figure 15.

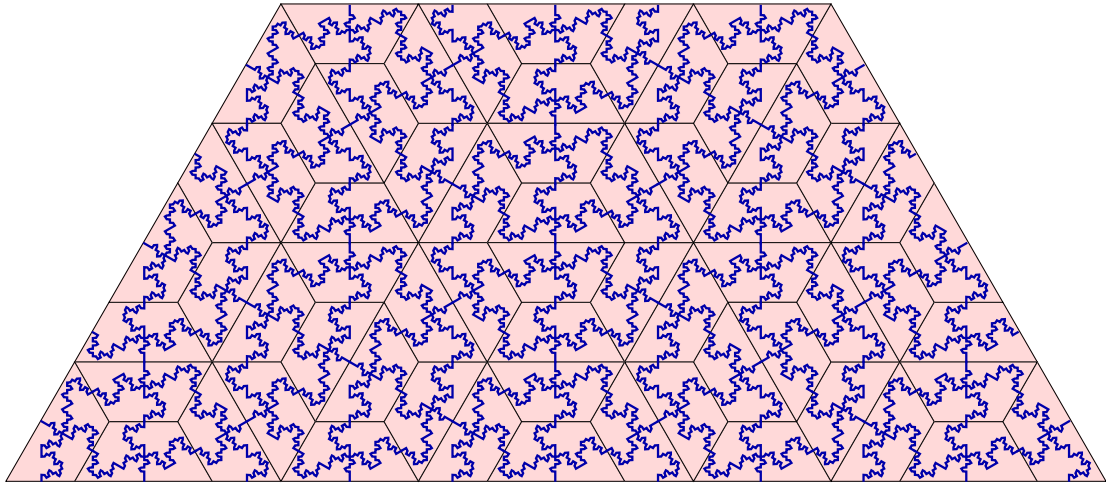


Figure 15: Half-hex tiling with fractal edges inscribed

Finally we then drop the original tiling leaving only the fractal edges. We are left with a fractal tiling of the plane. Figure 16 illustrates the fractal tiling of the plane from our example.

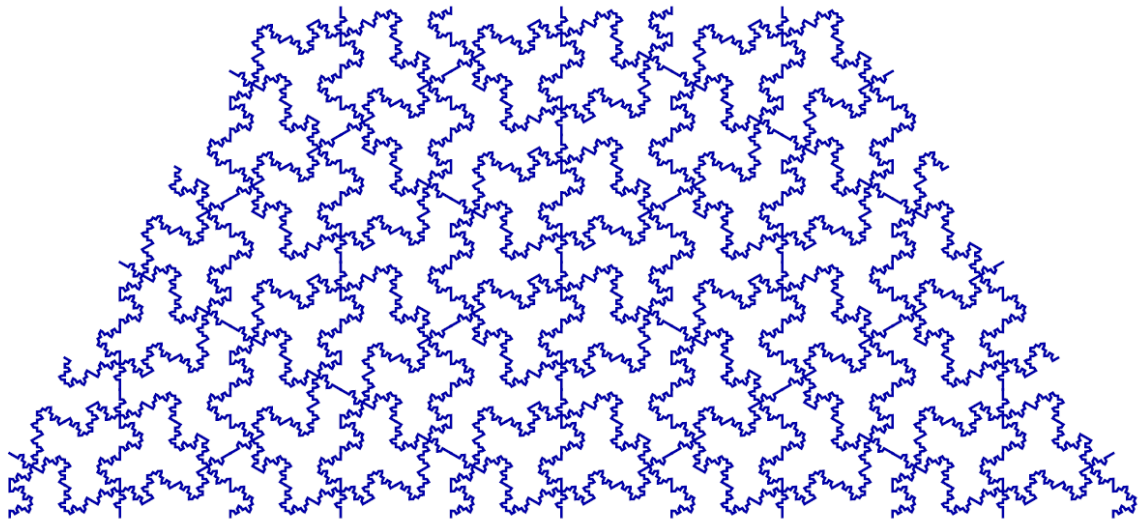


Figure 16: A fractal tiling of the plane

**Remarks 17.** Choosing the dual graph such that the edges meet the centre of each side of the prototile was done to ensure that when we take some substitution with the graph inscribed, the edges meet up in a nice grid as seen in Figure 11.

Furthermore, in Figure 12, where we deleted edges, again the remaining edges meet each side of the prototile in the centre. Again, this ensures that when we tile the plane with the final fractal edges, the fractal edges will meet up and form tiles.

### 3.2 Programming Fractal Tilings

In the previous section, we outlined the method used to obtain fractal tilings from existing substitution tilings. Our program follows the same process mentioned above.

**Problem 6.** How did we store the edges?

Similar to how we stored prototiles when programming substitution tilings, we stored our edges as a vector. For the half-hex example in the previous section, we had four edges and thus our vector will have four components. We describe our edges by their endpoints along with labels as we did with the prototiles.

**Problem 7.** How do we generate the grid as in Figure 11?

First of all, we needed to choose some substitution of the original tiling with the edges inscribed, say  $n$ . We then created another matrix which when applied to the vector of edges, gives us the grid we desire. This matrix was the *Fractalizer*. The Fractalizer is built from the  $n$ th power of the Tile Breaker. Suppose our Tile Breaker is an  $m \times m$  matrix. Then we have  $m$  prototiles, say  $p_1, \dots, p_m$ . Furthermore, suppose we have  $k$  edges inscribed in all the prototiles, say  $e_1, \dots, e_k$ . Now, entry  $(i, j)$  in the Fractalizer is obtained by considering edges  $e_i$  and  $e_j$ . If  $e_i$  is inscribed in prototile  $p_s$ , and  $e_j$  is inscribed in prototile  $p_t$ , then entry  $(i, j)$  in the Fractalizer is exactly entry  $(s, t)$  in the Tile Breaker. Applying the Fractalizer to the vector of edges, in the same way we applied the Tile Breaker, gives us the grid of edges.

**Problem 8.** How is deleting edges in the grid encoded in the program?

This process in deleting edges, or choosing our fractal edges, is done in steps; that is, we choose our fractal edges one by one. Consider our example from the previous section. First we choose how edge 1 will substitute. This is exactly edge 5. Thus, we delete all edges in the grid in Figure 10, leaving edge 5. This deletes the corresponding rigid motions of the relevant edges in the first row of the Fractalizer. We follow the same process with all of the edges inscribed within each prototile.



**Problem 9.** How do we iterate the edges?

Suppose we wanted to iterate the edges  $n$  times. Then we simply take the  $n$ th power of the Fractalizer, in exactly the same way we took the  $n$ th power of the Tile Breaker, and apply it to the set of edges.

**Problem 10.** How do we then generate the fractal tiling?

After iterating the edges, we shrink them down so that they will fit in the original prototiles, as we did in Figure 14. We then take the original Fractalizer, without any of the entries deleted, and apply it to our vector of new fractal edges. This yields a fractal tiling of the plane as in Figure 16.

**Remark 18.** These fractal edges are the attractor of a generalized IFS. What this means is we actually have a system of contractions giving us a set of attractors. This system of contractions is encoded in the Fractalizer. Figure 17 illustrates this point.

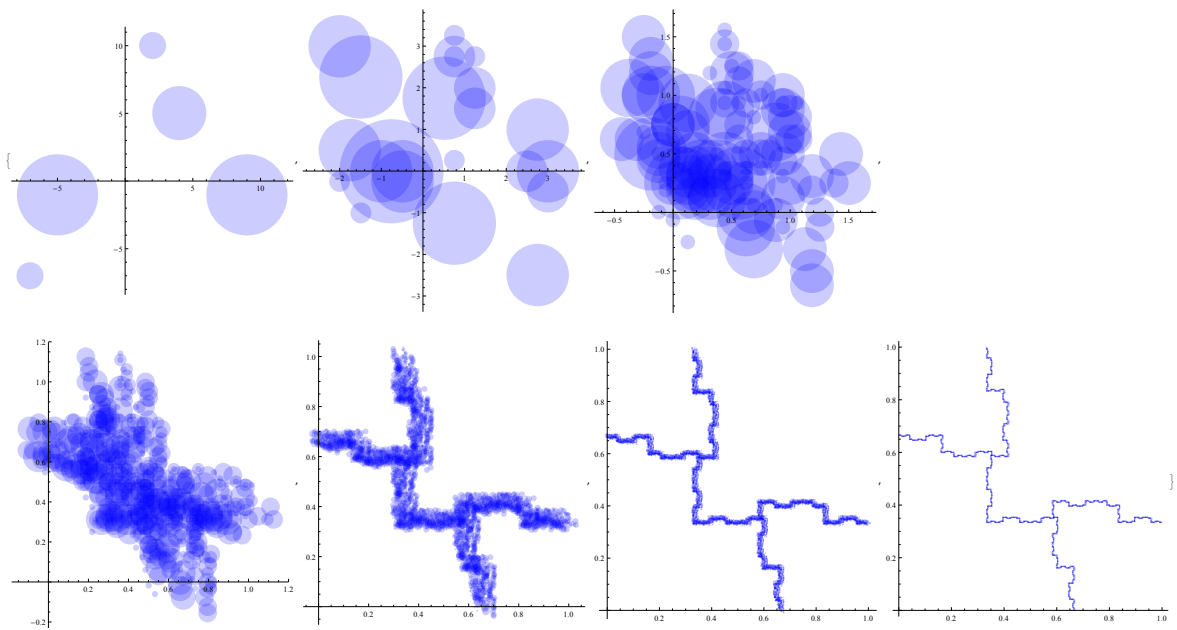


Figure 17: Fractal edges as attractors

The last image in Figure 17 was an example of a set of fractal edges which we had inscribed within a prototile. To see that these edges were attractors of a generalized IFS, we took a set of compact subsets, here we chose closed discs, and applied the

Fractalizer. Each graph in Figure 17 is looking at higher and higher powers of the Fractalizer applied to the set of closed discs. We observe that the discs are converging to the fractal edges as the power of the Fractalizer increases. In this way we see that the fractal edges are the attractors of this generalized iterated function system.

## 4 Future Directions

Our aim of this project was to create fractal tilings of the plane which we were able to attain. However, there is always more work to be done. We hope to explore the following areas.

- Extract the fractiles from the fractal tiling.
- Since these fractal tilings are built from substitution tilings, then they themselves are also substitution tilings. We intend to extract the substitution rules for the set of fractal prototiles from any fractal tiling we build.
- Set up a website where the general public can experiment with our program.

## 5 Acknowledgments

I would like to thank AMSI for granting me the opportunity to undertake this research over the summer. I would also like to thank my supervisors Sam Webster and Michael Whittaker, as well as Natalie Frank, for their guidance and support throughout this research vacation scholarship.

## References

- [1] M. F. Barnsley, *Fractals Everywhere*, Second edition, Academic Press Professional, Boston, 1993.
- [2] N. P. Frank and M. F. Whittaker, *A Fractal Version of the Pinwheel Tiling*, *Math. Intellig.* **33:2** (2011), 7-17.
- [3] L. Sadun, *Topology of tiling spaces*, AMS University Lecture Series, 46, Providence RI, 2008.
- [4] The Tiling Encyclopedia, Web address: <http://tilings.math.uni-bielefeld.de/>