CONVEX PRESENTATIONS OF TRANSLATION SURFACES

SIYI CHEN

1. Introduction

A translation surface is a union of polygons with pairs of parallel edges identified by translation. By cutting and pasting a translation surface with identified edges, some surfaces will have a shape of a convex polygon, and such surfaces are said to have a convex presentation.

While whether or not a translation surface has a strict convex presentation is studied thoroughly in genus 2[5], and a test for strict convex presentation of translation surfaces in general is raised by Samuel Lelièvre and Barak Weiss[5], the question is yet unsolved for translation surfaces in genus 3 or higher genus.

One application of translation surfaces is related to rational billiards[8]. A rational polygon is a polygon whose angles are rational multiples of π. The rational billiards studies the straight line flow in rational polygons, where the straight line is reflected when hitting the boundary of the polygon. Instead of considering reflection of straight lines, we may duplicate the polygon "in mirror image on the other side of edge, and let the flow continue straight through the wall into another copy of the polygon"[8]. It turns out that after doing a finite times of mirroring, a translation surface will be formed from the original rational polygon[8].

The application in rational billiards is also connected to solving the illumination problem, where you want to find a place to put only one point light source and light all areas inside a shape. If the shape is a rational polygon and the translation surface formed from the previous process has a strictly convex presentation, then there must be some place to put a point light source lighting all areas inside[2].

The convex presentations of translation surfaces play an important role in areas of pure math such as complex analysis, which will not be introduced here due to the lack of knowledge.

Our main goal in this paper is trying to implement a general Lelièvre-Weiss Convexity Test for strictly convex presentations of square-tiled translation surfaces (i.e., surfaces formed by unit squares) on the computer with the help of the flatsurf package in SageMath. We also manage to enumerate groups of translation surfaces in genus 2 for testing our implementation, which are studied by Samuel Lelièvre and Barak Weiss[5]. In the future, the

Lelièvre-Weiss Convexity Test implemented may be used to study translation surfaces in higher genus.

2. Background

Definition 2.1 (Translation Surface [5])**.** A **translation surface** is defined as a union of polygons with pairs of parallel edges identified by translation, up to cut and paste equivalence, which means the new surface by cutting the surface into 2 pieces along a line and gluing a pair of edges together is the same surface as the original one.

Some translation surfaces have a shape of a convex polygon up to cut and paste, while others do not. Translation surfaces which can be presented as a convex polygon are said to have a **convex presentation**. Figure 1 shows an example of translation surfaces with a convex presentation by cutting and pasting.

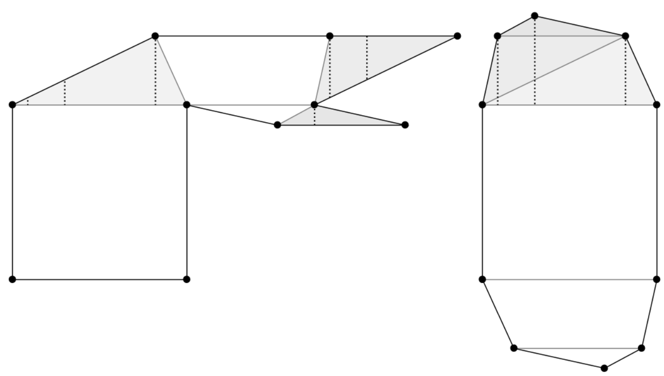

Figure 1. A translation surface with a convex presentation [5]

As shown in Figure 2, for each point p along the boundary of a translation surface, if you start at a point q within a small distance from p and go around p, you will finally get back to q after rotating around p for some angle α (following the arrows 1, 2, . . . , 8). We say α is the overall angle of identifying all sides for the point p. In Figure 2, $\alpha = 6\pi$.

Definition 2.2 (Cone Point and Cone Angle [9, Definition 1.5])**.** A **cone point**, aka **singularity** p_0, on a translation surface is a point whose overall angle of identifying all sides is $2(k+1)\pi$ for some integer $k > 0$. The singularity at p_0 is said to have **cone angle** $2(k+1)\pi$. For example, in Figure 2, p is a cone point with cone angle 6π. Any interior point and any point on a edge of a translation surface has angle 2π with the possible exception of the corners[1].

Gluing all pairs of identified edges of a translation surface together, we would obtain a topological surface, which has certain number of holes, i.e., genus.

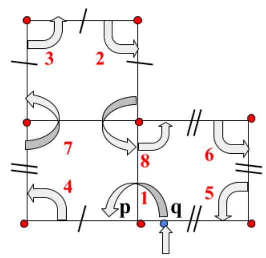

FIGURE 2. Overall angle of identifying all sides for point p.

Definition 2.3 (Genus [3])**.** **Genus** is a topologically invariant property of a surface defined as the largest number of nonintersecting simple closed curves that can be drawn on the surface without separating it. Roughly speaking, it is the number of holes in a surface.

Theorem 2.4. *(Gauβ-Bonnet formula* [6, Proposition 1.14]*) Let x be a finite translation surface of genus g that has n cone angles of* $2(k_i + 1)\pi$ *, i = 1, . . . , n. Then it holds*

$$2g - 2 = \sum_{i=1}^{n} k_i$$

Translation surfaces are usually grouped into strata according to the number of sigularites as well as their cone angles.

Definition 2.5 (Stratum)**.** A **stratum** is a set of all translation surfaces having the same set of cone angles $2(k_i + 1)\pi$, i = 1, . . . , n. A stratum is written as $\mathcal{H}(k_1, k_2, ..., k_n)$, where the order of k_i doesn't matter.

According to Theorem 2.4, translation surfaces in a stratum must have the same genus g. Figure 2 shows a translation surface in $\mathcal{H}(2)$, where all the vertices is a single cone point(i.e., when gluing together, they are the same point) with cone angle 6π.

Definition 2.6 (Saddle Connection)**.** A point, (i.e. a translation surface) in a stratum is specified by a finite number of line segments joining singular points, and such line segments are called **saddle connections**.

Definition 2.7 (Simple Cylinder[5])**.** A **cylinder** on a translation surface is an isometrically embedded copy of a Euclidean cylinder $(\mathbb{R}/c\mathbb{Z})\times(0,\ h)$ whose boundary is a union of saddle connections. A **simple cylinder** on a translation surface is a cylinder with one saddle connection on the top and bottom.

By fixing a minimum number of oriented line segments, possibly not all the line segments, we are able to fix the shape of a translation surface.

Definition 2.8 (Period Coordinates)**.** We can represent a translation surface with vectors representing a minimum number of line segments used to fix the shape of the translation surface. And the coordinates given in the procedure is called **period coordinates**, which allow us to represent linear transformation of shapes more conveniently.

For example, choosing the left two vertical segments and the up two horizontal segments, the leftmost surface in Figure 3 can be represented as $((0,1),(0,1),(1,0),(1,0)) \in \mathbb{R}^8 = dim(\mathcal{H}(2))$.

Based on the period coordinates, how $GL(2,\mathbb{R})$ acts on points in a stratum can be analyzed. For instance, let $U = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, and let x be the leftmost translation surface in Figure 3, Ux, which is the surface obtained after shearing x with U, is shown on the right.

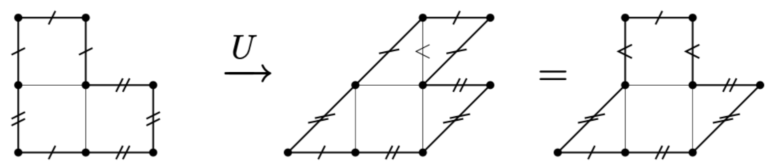

FIGURE 3. U acts on a point in H(2) [4]

$GL(2,\mathbb{R})$ action is found to be a strong tool helping study convex presentations.

Theorem 2.9 ([7])**.** *The property of having no strictly convex presentations is $GL(2,\mathbb{R})$-invariant.*

Hence, $GL(2,\mathbb{R})$ action preserves the property of having strict convex presentations. At the meantime, the method proposed by Samuel Lelièvre and Barak Weiss to check whether or not a translation surface has a strict convex presentation is as follows.

Theorem 2.10 (Lelievre-Weiss Convexity Test[5])**.** *A translation surface has a strict convex presentation if and only if it contains a simple cylinder C such that:*

(1) C is horizontal up to shearing.
(2) C also contains a vertical saddle connection after shearing.
(3) In $\mathcal{H}(2g-2)$, there are $2g-1$ southward pointing vectors from a cone point, each must intersect the top of C.
(4) The last step decomposes the top of C into $2g-1$ many segments, which will return to the bottom of C in reverse order following the southward direction.
(5) Finally, connecting the cone points above C, a convex polygon will be gained.

Manually processing the test for a translation surface would be time consuming. If we manage to implement the convexity test as well as generating translation surfaces in a given stratum with the computer, it will be helpful for us to study the convex presentations of translation surfaces.

For computational achievable, we restrict our study on square-tiled surfaces (origami) with $SL(2,\mathbb{Z})$ action.

Definition 2.11 (n-Origami)**.** An n**-Origami** is a translation surface consisting of n unit squares.

Any n-Origami can be represented using a right permutation and an up permutation, which describes the order of gluing squares together.

Definition 2.12 (Right and Up Permutation)**.** Label n squares in an origami as 1, 2, ..., n. Since only one square can be glued on the right of a given square, hence if we write out the squares on the right of 1, 2, ..., n accordingly, we would gain a permutation of n, which is defined as the **right permutation**. Besides, only one square can be glued on the top of a given square and if we write out the squares on the top of 1, 2, ..., n accordingly, we would also gain a permutation of n, which is defined as the **top permutation**.

Theoretically, since the first step of test in Theorem 2.10 is equivalent that some surface in the $GL(2,\mathbb{R})$ orbit of the given surface will contain a horizontal simple cylinder, hence we would need to check infinitely many directions for shearing to see whether or not there would be a horizontal simple cylinder. Luckily, for any origami, without proof we state that:

Theorem 2.13. *A surface in the $GL(2,\mathbb{R})$ orbit of an origami contains a horizontal simple cylinder if and only if a surface in the $SL(2,\mathbb{Z})$ orbit of the origami contains a horizontal simple cylinder.*

3. Preliminaries

3.1. **Enumeration of n-origami in $\mathcal{H}(2)$.** The first problem is to enumerate origamis in a given stratum. We tackle the problem with origami in $\mathcal{H}(2)$ as a start.

In $\mathcal{H}(2)$, for every square-tiled surface, there are some integer values of h_i, t_i, and u_i, so that the surface is equivalent to the surface in Figure 4 up to cut and paste, where h_i, u_i, t_i are integers and $u_1 < u_2$, $0 <= t_1 < u_1$, $0 <= t_2 < u_2$.

Let S_n denote the collection of all square-tiled surfaces as in the figure where $h_1 u_1 + h_2 u_2 = n$.

For a given integer n, if we fix the group of number h_1, u_1, h_2, u_2, then the varying of t_1 and t_2 describes how the two cylinders are sheared from a rectangle. However, to enumerate surfaces, instead of considering shearing, we can consider the locations of singularities instead.

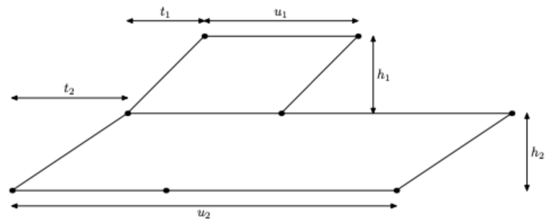

FIGURE 4. A square-tiled translation surface in $\mathcal{H}(2)[5]$

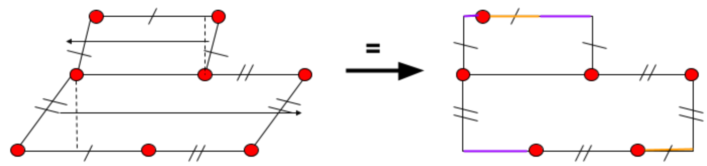

FIGURE 5. Enumerate shearing by enumeration positions of singularities

As shown in Figure 5, by cutting and pasting, any surface in $S(n)$ can be in the shape of a narrow rectangle on the top of a wide rectangle with their left sides aligned.

Due to the different shearing, the positions of singularities will vary along the top and bottom of the translation surface. Fixing the shearing of the top and bottom cylinder is equivalent to fixing the positions of the singular points on the top and bottom of the origami.

In conclusion, the algorithm to enumerate all possible square-tiled surfaces in S_n is as follows:

(1) Enumerate all possible integers (h_1, u_1, h_2, u_2) such that $h_1u_1+h_2u_2 = n$ and $u_1 < u_2$.
(2) For each group of integers (h_1, u_1, h_2, u_2), fix the shape of the square-tiled surface to be a narrower rectangle on the top of a wider rectangle with their left sides aligned.
(3) Label each square in the surface as 1, 2, 3, ..., n from left to right, top to bottom. And fix the inner right and up permutations.
(4) Enumerate all possible positions for cone points to be on the top and bottom edges of the surface, which fixes the way of gluing cone points together and thus gives the complete right and up permutations.
(5) For each pair of right and up permutations constructed, create an origami with the flatsurf package in SageMath, and add it to the set of $S(n)$.

Then, we can study origamis in $\mathcal{H}(2)$ as different groups of n-Origamis.

3.2. **Generating $SL(2,\mathbb{Z})$ Orbits.** Given any square-tiled surface x, the **Veech group** of x is defined as $\mathrm{SL}(x) =: \{g : g \in \mathrm{SL}(2,\mathbb{Z}), gx = x\}$. There is a minimal finite collection $\{h_1, ..., h_n\}$ of elements of $\mathrm{SL}(2,\mathbb{Z})$, called **right coset**, so that $\mathrm{SL}(2,\mathbb{Z}) = \bigcup_{i=1}^{n} h_i \mathrm{SL}(x)$. Then the $\mathrm{SL}(2,\mathbb{Z})$ orbit of x is: $\mathrm{SL}(2,\mathbb{Z}) \cdot x = \bigcup_{i=1}^{n} h_i \cdot x$.

By Theorem 2.13, for a square-tiled surface x, if up to shearing by elements in $GL(2,\mathbb{R})$, it contains a horizontal simple cylinder, one of the surface in its $SL(2,\mathbb{Z})$ orbit will contain a horizontal simple cylinder. If any such surfaces in its $SL(2,\mathbb{Z})$ orbit passes steps 2, 3, 4, and 5 of the Lelievre-Weiss Convexity Test, then we can conclude the original surface as well as all the surfaces in its $SL(2,\mathbb{Z})$ orbit have a strict convex presentation.

Our first step is to output the $SL(2,\mathbb{Z})$ orbit of a given origami. The main algorithm is as follows:

(1) Find the left coset of the origami, which is a built-in function in the flatsurf package in SageMage.
(2) For each element in the left coset, find its inverse matrix. And all those inverse matrices form the right coset, which is desired.
(3) Let each element in the right coset act on the origami to gain the orbit.

The main problem to implement the algorithm generating $SL(2,\mathbb{Z})$ orbits is step 3: implementing the $SL(2,\mathbb{Z})$ action on an origami.

3.3. **Implementing $SL(2,\mathbb{Z})$ Action on Origamis.** Let $R = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ be the rotation matrix, $S = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ be the shearing matrix, x be an origami. Rx and Sx are implemented in the flatsurf package.

Theorem 3.1. *The matrices S and R generates $SL(2,\mathbb{Z})$.*

With the built-in functions of flatsurf and Theorem 3.1, the problem is then reduced to factorizing any marix in $SL(2,\mathbb{Z})$ with S and R, after which we can apply the S and R actions in series to complete the action of the original matrix. The factorization problem can be solved using the following result:

(1) If $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, then we are done.
(2) Else, if $a \times c + b \times d > 0$, let $M = S^{-1}M$
(3) Else, if $a \times c + b \times d < 0$, let $M = R^{-1}M$

It turns out for any M in $SL(2,\mathbb{Z})$, the above process will always make M become an identity matrix after finite steps, giving $A_n M = I_2$, where A_n is a series of S^{-1} and R^{-1} multiplying together. Then $M = A_n^{-1}$ is a series of S and R multiplying together.

3.4. **Equivalent Lelièvre-Weiss Convexity Test.** According to Theorem 2.10, in step 2 of the Lelièvre-Weiss Convexity Test, the simple horizontal cylinder is sheared to maintain horizontal while having a vertical saddle connection. And steps 3, 4, and 5 are based on finishing step 2.

However, the matrix used to shear in step 2 may be in $SL(2,\mathbb{R})$ instead of $SL(2,\mathbb{Z})$, which will give computation difficulty.

Noticing that the effect of shearing on step 3 and 4 is that after shearing: in step 3, the vectors will point directly southward instead of having a non-zero angle with the vertical direction; in step 4, the intervals on the top of the cylinder will return to the bottom of the cylinder in the southward direction.

Let the shearing matrix applied be $H = \begin{pmatrix} 1 & h \\ 0 & 1 \end{pmatrix}$, and the original direction of vectors as well as return maps be $d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$. Since $Hd = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$, which is in the southward direction, hence $d = \begin{pmatrix} h \\ -1 \end{pmatrix}$

Then, after we know which shearing matrices can be applied to shear the cylinder to have a vertical saddle connection, we can process step 3 and 4 following the according direction instead. The method to find all shearing matrices making a simple cylinder to have a vertical saddle connection is discussed in the next section.

Besides, for step 5, whether shearing the origami or not will not affect the convexity of the shape above the cylinder.

3.5. **Finding All Shearing Matrices Giving a Vertical Saddle Connection of a Simple Cylinder.** As shown in Figure 6, if a horizontal simple cylinder C of height y and width x in a translation surface already contains a vertical saddle connection, then after shearing the translation surface by the matrix $\begin{pmatrix} 1 & \frac{x}{y} \\ 0 & 1 \end{pmatrix}$, the horizontal simple cylinder C in the new surface would still contain a vertical saddle connection.

The two surfaces may not pass the last few steps of the Lelièvre-Weiss Convexity Test at the same time, which means, as long as the two surfaces are not the same surface, we would need to test both of them with the last few steps of the Lelièvre-Weiss Convexity Test.

Theorem 3.2. *Let $M = \begin{pmatrix} 1 & \frac{x}{y} \\ 0 & 1 \end{pmatrix}$, where x, y are integers, and t be an origami. There exists a finite $k \in \mathbb{N}$ such that $M^k t = t$, which is equivalent to M^k being the Veech group of t.*

The proposed algorithm to find all possible shearing matrices making a horizontal simple cylinder have a vertical saddle connection is as follows:

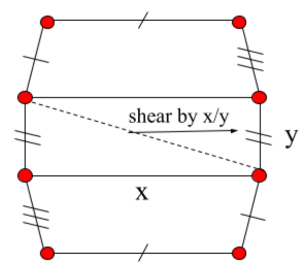

FIGURE 6. Shearing while maintain a vertical saddle connection

(1) Let the list of all possible shearing matrices be $S = [\,]$. Let t be the translation surface containing a horizontal simple cylinder C with height y and with x.
(2) Find any shearing matrix S_0 that will make the C have a vertical saddle connection. Append S_0 into S.
(3) let $M = \begin{pmatrix} 1 & \frac{x}{y} \\ 0 & 1 \end{pmatrix}$.
(4) For $k = 0, 1, 2, 3, \ldots$, if M^k is in the Veech group of $S_0 t$, then stop. Otherwise, append $M^k S_0$ into S and keep increasing k.
(5) Output S.

3.6. **Implementing Lelièvre-Weiss Convexity Test.** Now we are ready to implement the equivalent Lelièvre-Weiss Convexity Test for surfaces in $\mathcal{H}(2g-2)$.

The complete algorithm is as follows:

(1) For a given origami O, find its $SL(2,\mathbb{Z})$ orbit.
(2) For each origami in $O's$ orbit, find those with horizontal simple cylinders and all of their horizontal simple cylinders. If no such origami exist, output False.
(3) For each surface with simple horizontal cylinders and for each simple cylinder it contains, find all the possible directions for vectors to from singularities.
(4) For each possible direction, check whether there are $2g-1$ arrows intersecting with the top of the cylinder. If no such directions exist, output False.
(5) For each direction having $2g-1$ arrows intersecting with the top of the cylinder, check whether the return map of gained intervals on the top of the cylinder to the bottom of the cylinder is in reverse order. If no such return map exist, output False.
(6) For each origami in $O's$ orbit having a simple cylinder C passing all the above steps, check whether all cone points above C form a convex polygon. If no such convex polygon exist, output False.

(7) Output True.

4. Results

The Lelièvre-Weiss Convexity Test implemented is tested with surfaces in genus 2, which are studied by Samuel Lelièvre and Barak Weiss [5].

Samuel Lelièvre and Barak Weiss gives all 7 types of non-strictly-convex origamis in $\mathcal{H}(2)$. These types are classified as:

(1) $D = 9$, a representative of which is $X(0, 2, 1, 1)$
(2) $D = 16$, a representative of which is $X(0, 3, 1, 1)$
(3) $D = 36$, a representative of which is $X(1, 4, 2, 2)$
(4) $(D, \epsilon) = (25, 0)$, a representative of which is $X(0, 6, 1, 3)$
(5) $(D, \epsilon) = (25, 1)$, a representative of which is $X(0, 6, 1, 2)$
(6) $(D, \epsilon) = (49, 1)$, a representative of which is $X(0, 12, 1, 4)$
(7) $(D, \epsilon) = (81, 1)$, a representative of which is $X(0, 20, 1, 4)$

Where $X(a, b, c, \lambda)$ denotes the translation as shown in Figure 7 surface[5].

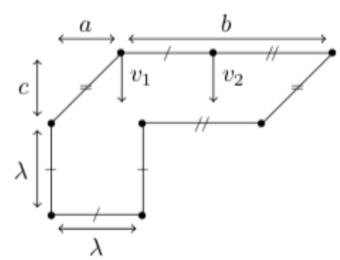

Figure 7. $X(a, b, c, \lambda)$[5]

Our implementation gives correct results for all typical testing surfaces of above types.

Besides, our implementation gives correct results for typical surfaces with a convex presentation, such as the octagon in Figure 8.

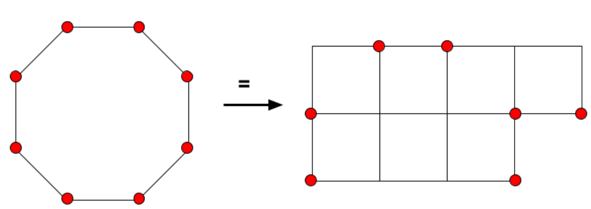

Figure 8. An oracle constructed by unit squares

References

[1] Paul Apsia and Chaya Norton. Log(m): Convex presentations of translation surfaces exercises: 01-26-2021. page 1, 2021.

[2] Paul Apsia and Alex Wright. Marked points on translation surfaces. *arXiv:1708.03411v3 [math.DS].*, pages 14–23, 2021.

[3] A. Gray. Modern differential geometry of curves and surfaces with mathematica. *Boca Raton*, page 635, 1997.

[4] Pascal. Hubert and Samuel Leliévre. Square-tiled surfaces in h(2). *https://www.researchgate.net/publication/228799422.*, pages 10–12, 2013.

[5] Samuel Lelièvre and Barak Weiss. Translation surfaces with no convex presentation. *arXiv:1306.3606v1 [math.DS].*, pages 1–30, 2013.

[6] Anja Randecker. Geometry and topology of wild translation surfaces. *Scientific Publishing.*, page 18, 2016.

[7] W. A. Veech. Geometric realizations of hyperelliptic curves. *Plenum.*, 1995.

[8] Alex Wright. Topic proposal: Translation surfaces and teichmuller theory. page 3, 2010.

[9] Alex Wright. Translation surfaces and their orbit closures: an introduction for a broad audience. *arXiv:1411.1827v1 [math.DS].*, pages 2–7, 2014.