WXML Final Report: The Translation Surface of the Bothell Pentagon

Jayadev Athreya, Diaaeldin Taha,
Wenbo Gao, Maria Osborne, Matthew Staples

Spring 2016

I Introduction

In the summer of 2015, Casey Mann, Jennifer McLoud-Mann, and David Von Derau of University of Washington Bothell announced that they had discovered a new convex pentagon capable of tiling the plane (hereby referred to as the Bothell Pentagon).

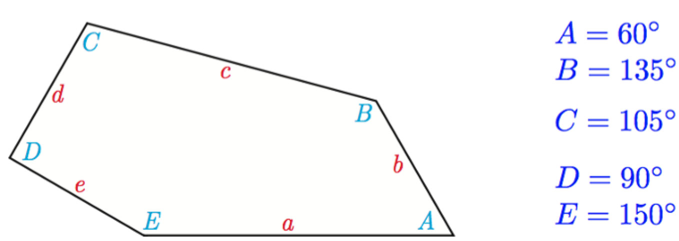

Figure 1: The Bothell Pentagon. Illustration courtesy of Casey Mann

This was an exciting event for the mathematical community, as the Bothell pentagon is only the fifteenth known type of pentagon that can tile the plane. Therefore our initial direction for this project was to study some characteristics and statistics of that tiling. We had intended to look at the gap distribution of the tiling, as well as ideas like the number of vertices and/or pentagons within circles of various radii. The other direction we set out on was enumerating the group elements of the pentagon, with the ultimate goal of determining the genus of its translation surface.

Over the course of the quarter, we ended up focusing more exclusively on studying the translation surface of the Bothell pentagon rather than statistics of the tiling. This led us in the direction of considering billiard paths within the pentagon and the translation surface, as well as the cylindrical decomposition of the surface.

II Theoretical Background

II.i Translation Surface

II.i.(i) Billiard and Reflection Group

The reflection group of a polygon is a finite group of reflected and rotated versions of that polygon that arise from an initial set of reflections. Various permutations of reflections generate elements of the group.
The reflection group is related to the idea of mathematical billiards. Rational billiards refers the the theoretical motion of a point mass in a polygon who angles are rational multiples of π. However, instead of allowing the point mass to bounce off the edge that it contacts, it continues its trajectory in a straight line by flowing into a reflected version of the polygon. Doing enough trajectories results in an exhaustive set of reflections, the reflection group of the polygon. Finding this group allows us to study the surface.

II.i.(ii) Euler's Formula

Euler's Formula is useful in computing the genus of complicated translation surfaces:

$$V - E + F = 2(1 - G)$$

where V is the number of vertices, E the number of edges, F the number of faces, and G the genus, or number of holes, in the surface.

II.ii Cylinders

In simplest terms, cylinders of a surface are the areas bounded by saddle connections. We looked first at determining if our surface decomposed completely into cylinders in a particular direction, and if did, then at the possibility that it might be a Veech surface. A cylinder's modulus is the ratio of its width to its height, and if all the moduli are rationally related in all possible cylindrical decompositions of a surface, then the surface is Veech.

III Algorithm Development

One of our primary goals after finding the group and translation surface of our pentagon by hand was to generalize and automate the process so our group, or anyone else in the mathematical community, could easily do the same. Therefore we wrote code in SageMath that allows the user to compute the reflection group, genus of the translation surface, or explore billiard paths for any polygon, provided that it is a "legal" polygon (which for our purposes means that its internal angles are rational multiples of π and that it is not self-intersecting).

III.i Reflection Group

For a rational polygon, its reflection group is composed of a finite number of generators in $\mathbb{R}^2$, which is called *dihedral group.* The generators are of two kinds: *rotations* (denoted as Rot) and *reflections* (denoted as Ref) with an angle of $\frac{2\pi}{n}$. Their matrix representations are the followings:

$$Rot(\theta) = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix}, Ref(\theta) = \begin{pmatrix} cos\theta & sin\theta \\ sin\theta & -cos\theta \end{pmatrix}$$

where θ is the angle associated with each generator.

The algorithm requires a list of initial generators, i.e., angles that correspond to lines parallel to each side of the polygon. It begins with a rotation by zero and the identity (denoted as id), and then tests compositions of reflections as per the appropriate generators to find elements of the reflection group:

Algorithm 1 Generate Reflection Group

Require: ListOfGenerators

```
FoundElements := {id}
NewElements := {id}
while NewElements != ∅ do
    A := NewElements
    NewElements := {}
    for a ∈ A do
        for g ∈ ListOfGenerators do
            if a * g ∉ FoundElements then
                addset(a * g, FoundElements)
                addset(a * g, NewElements)
            end if
        end for
    end for
end while
```

III.ii Edge Graph and Vertex Graph

The problem of computing the genus for any legal polygon comes down to counting the number of faces, edges, and vertices in the surface. To do so, the algorithm first uses the reflection group code to find the group and thus find the number of faces in the surface.

The algorithm uses NetworkX, a package for python, to create edge and vertex "graphs." For the edge graph, it first creates a set of "nodes" that represent each of the polygon's edges under all rotations and reflections as determined by the reflection group. It then creates connections between the nodes of equivalent angle, orientation, and side number.

The Vertex graph algorithm also begins by creating nodes that represent vertices of the polygon under rotations and reflections as determined by the reflection group. It then uses one of NetworkX's cycle finding algorithms to find cycles that make up the basis of the vertex graph, and uses the number of cycles to compute the number of vertices.

Algorithm 2 Generate Edge Graph

Require: reflectionGroup, setOfAngles
  initialize Edge Graph
  index = 0
  for $r \in$ reflectionGroup **do**
    for $s \in \{0, 1, 2, ...,$number of edges in the polygon$-1\}$ **do**
      index = index + 1
      add $(index, \{"generator" : r, "edge" : s\})$ as a node to the Edge Graph
    end for
  end for
  for $i \in \{1, 2, 3, ...,$ number of nodes in the Edge Graph$\}$ **do**
    for $j \in \{i + 1, i + 2, ...,$ number of nodes in the Edge Graph$\}$ **do**
      if $node_i$ matches $node_j$ **then**
        add $(node_i, node_j)$ as an edge to the Edge Graph
        add $(node_j, node_i)$ as an edge to the Edge Graph
      end if
    end for
  end for

Algorithm 3 Generate Vertex Graph

Require: edgeGraph, numEdge(number of edges in the polygon)
  for $i \in$ the set of nodes in the Edge Graph **do**
    add $(i.index, \{"generator" : i.generator, "vertex" : i.edge\})$ as a node to the Vertex Graph
  end for
  for edge $\in$ the set of edges in the Edge Graph **do**
    add (edge.leftVertexIndex, edge.rightVertexIndex) as an edge to the Vertex Graph
    add (edge.leftVertexIndex+1, edge.rightVertexIndex+1) as an edge to the Vertex Graph
  end for

III.iii Billiard Trajectories in Rational Polygons

We wrote code that is capable of drawing billiard trajectories in both the Bothell polygon and legal custom polygons. The algorithm requires a polygon with sides defined by vectors, as well as an initial point and an initial vector to initialize the billiard path. It solves systems of parametric equations to determine if/where the current path intersects with one of the sides. It chooses the first side with which the path intersects (to handle the case of, for example, an L-shaped polygon where a path may potentially intersect with multiple sides). The trajectory terminates if it hits a vertex. Otherwise, the angle at which the trajectory hits and reflects off of a particular side is used to compute the next path. At the present time, the algorithm computes paths up to a inputted number of steps, unless the trajectory terminates by ending in a vertex. It cannot tell when the trajectory is periodic.

IV Result

IV.i Reflection Group

We set side e of the Bothell pentagon to be parallel to the x axis in Cartesian space. The angle generators of the group were therefore 0, $\frac{\pi}{12}$, $\frac{\pi}{6}$, $\frac{\pi}{2}$, and $\frac{5\pi}{6}$:

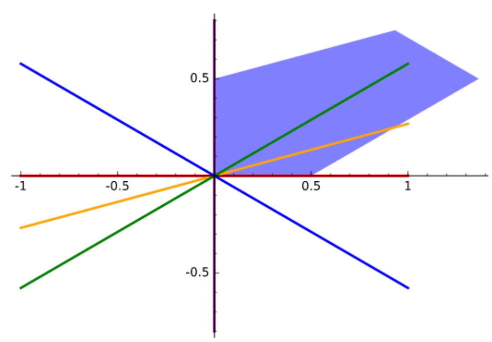

Figure 2: Generators of the Reflection Group

The group for the Bothell pentagon contains 24 elements: 12 reflections and 12 rotations. The angles of rotation are the integer multiples of $\frac{\pi}{6}$ between 0 and 2π, while the angles of reflection are the integer multiples of $\frac{\pi}{12}$ between 0 and π. We found the following to be the most elegant and useful arrangement of the elements of the reflection group, where the yellow pentagon represents the identity:

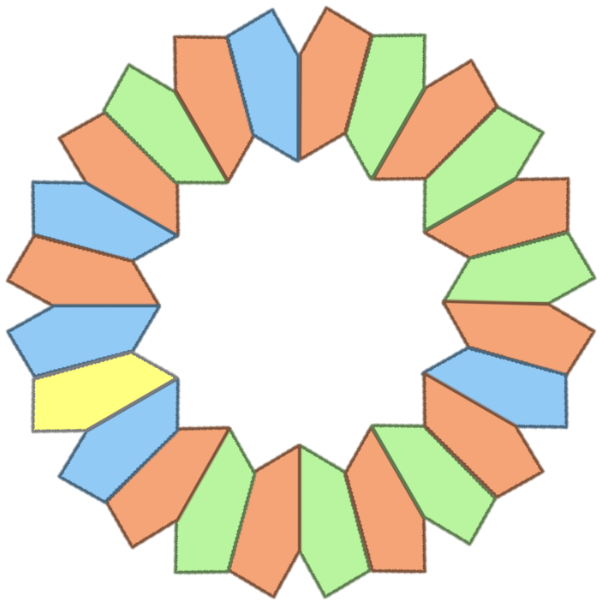

Figure 3: Reflection Group

IV.ii Euler's Characteristic

For the translation surface of the Bothell Pentagon, We needed to find the number of vertices, edges, and faces to compute the genus. The number of faces was simplest to compute, since the surface is made up of 24 pentagons and thus has 24 faces. As shown in the next diagrams, the surface has 60 edges and 16 vertices. Thus Euler's formula gives:

$$16 - 60 + 24 = 2 - 2G \implies G = 11$$

Therefore the genus of the surface is 11. In addition, the surface has 6 vertices where the angle exceeds 2π.

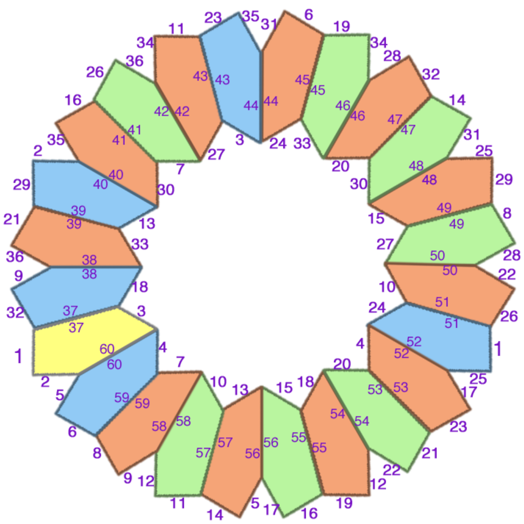

Figure 4: Edge identifications on the translation surface

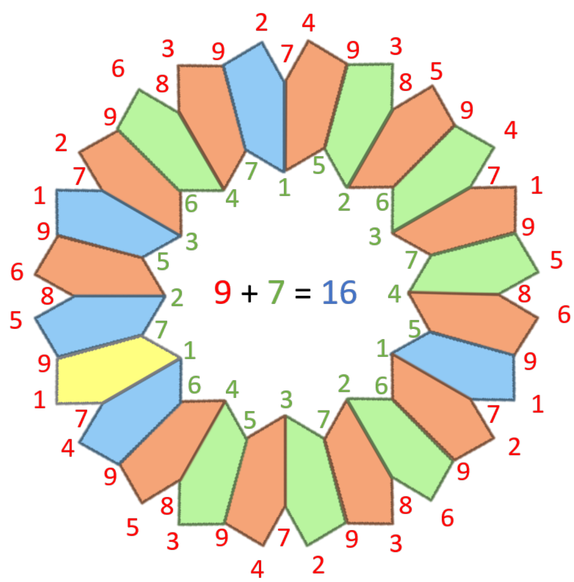

Figure 5: Vertex identifications on the translation surface

IV.iii Billiard Trajectories

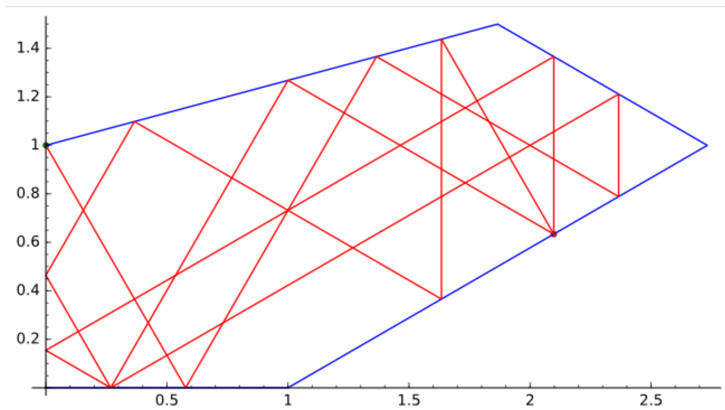

Figure 6: A terminating billiard trajectory

We wrote a code to generate billiard trajectories inside a pentagon which produces the sequence of edges hit in a trajectory as well as the location coordinates of each hit. We intended to study periodic trajectories in the Bothell pentagon but were not able to do so in great detail. However, we found that trajectories which lie inside cylinders in parallel directions are periodic, and those such trajectories which start at a vertex terminate at the same vertex as shown above.

IV.iv Cylinders

The surface of the Bothell pentagon is able to decompose completely into cylinders. We studied cylinders in the vertical direction for our surface, however, as is apparent because of the bilateral and radial symmetry of this particular arrangement of the reflection group, a similar decomposition is possible in the horizontal direction as well as in angular rotations by integer multiples of $\pi/6$.

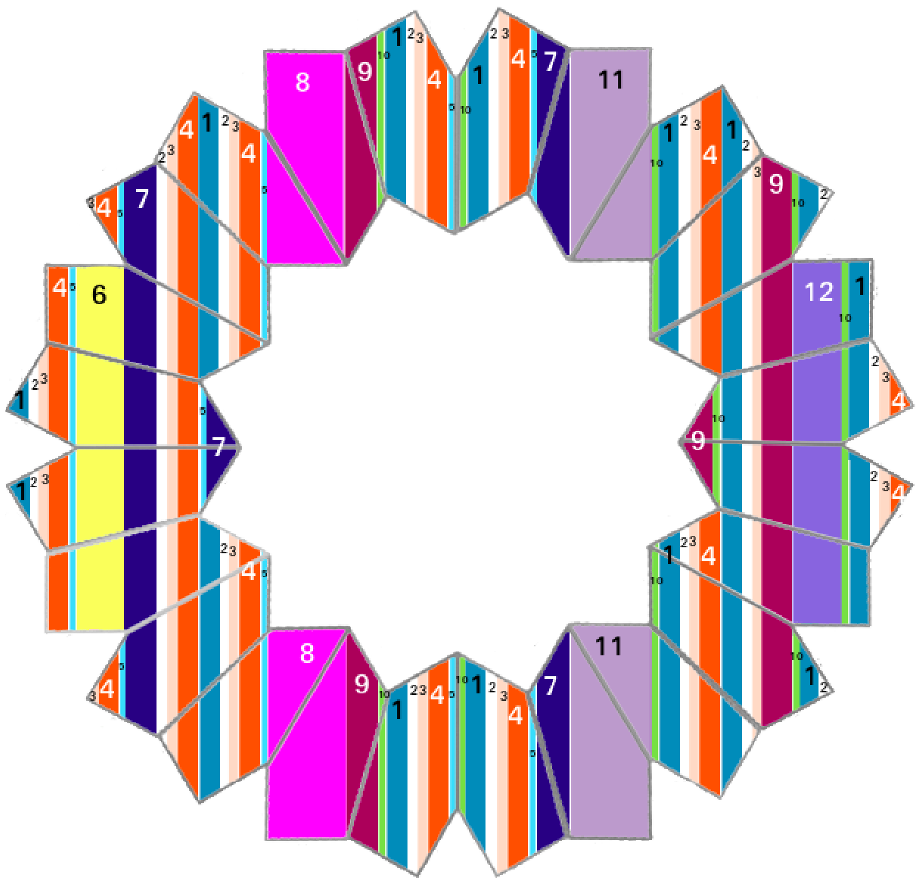

Figure 7: Cylindrical Decomposition in the Vertical Direction

In computing the dimensions of the cylinders, we set the lengths of sides d, e, and b to 1 for simplicity; thus the length of side c is $\frac{\sqrt{2}}{\sqrt{3}-1}$ and the length of side a is 2. Using this scale, the dimensions of the cylinders are as follows (note that each cylinder has identical dimensions as exactly 1 other cylinder):

Cylinder no.	Width	Height
1, 4	$\frac{2\sqrt{3}-3}{2}$	$20+10\sqrt{3}$
2, 3	$\frac{2-\sqrt{3}}{2}$	$22+12\sqrt{3}$
5, 10	$\frac{2-\sqrt{3}}{2}$	$12+6\sqrt{3}$
6, 12	$\frac{3-\sqrt{3}}{2}$	$3+\sqrt{3}$
7, 9	$\frac{\sqrt{3}-1}{2}$	$6+4\sqrt{3}$
8, 11	1	$2+2\sqrt{3}$

There are many cylinders whose moduli are not rationally related, therefore we have proved that this is not a Veech surface. However, the fact that each cylinder has the same dimensions as another suggests that our current surface my be a double cover of a different surface.

V Future directions

One direction to explore is patterns in sequences of edges hit in billiard trajectories. These sequences may show if there are common characteristics in periodic and terminating trajectories other than running parallel to the cylinders we found. There may also be more cylinder decompositions to discover as we only found those in the six directions discussed.

Now that the all processes for finding the translation surface for the Bothell pentagon have been automated, the translation surface for other convex pentagons, and convex polygons in general, can be found easily. From there, the same directions that we took in cylinders and billiards can be taken or completely different questions can be explored.