

A combinatorial algorithm for visualizing representatives with minimal self-intersection

Chris Arettines

August 25, 2015

Abstract

Given an orientable surface with boundary and a free homotopy class of a closed curve on this surface, we present a purely combinatorial algorithm which produces a representative of that homotopy class with minimal self-intersection.

1 Introduction

Consider Figure 1 which depicts a curve on the torus with one boundary component, obtained by identifying the labeled edges. One can ask whether this curve intersects itself minimally in its free homotopy class. The algorithm described here answers this question and furthermore, if the answer is no, produces a new curve in the same free homotopy class which has the minimal possible number of self-intersections. This work answers a question of Chas first motivated by her study of self-intersections in papers [Chas 2004; Chas 2010]. The input for this algorithm is the edge-gluing pattern for a polygon which determines the surface, and an edge crossing sequence that corresponds to a free homotopy class.

In the past, other algorithms have been found which accomplish similar goals. Some of the earliest published examples can be found in [Chillingworth 1969; Reinhart 1962]. In [Birman and Series 1984], an algorithm is found that determines whether a free homotopy class (given in terms of the same data used here) is simple. In [Cohen and Lustig 1987], the authors discover a method to determine the minimal self-intersection number of a free

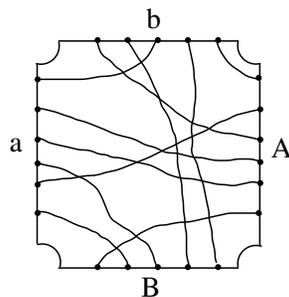


Figure 1: A "typical" curve on the torus with one boundary component.

homotopy class, again with the same data. The algorithm described in this paper finds the self-intersection number via a different method. In addition, it provides a geometric picture of what a minimal curve in the free homotopy class looks like. Other methods such as the flow described in [Hass and Scott 1994] produce minimally intersecting curves, but they have the disadvantage of relying on geometric data which we do not need here. It is a theorem of Hass and Scott, found in [Hass and Scott 1985], that in order to determine if a curve constructed in our manner has the minimal number of self-intersections, one must check for "proper bigons", which are defined in Section 2. The algorithm described here works by converting the problem of finding and removing bigons into a purely combinatorial problem. If a proper bigon is found, then there are combinatorial procedures which mimic the homotopy that removes the bigon. We can then iterate this process until there are no more proper bigons, which by the main theorem in [Hass and Scott 1985] tells us that our curve must have the minimal number of self-intersections.

This paper is structured as follows: In Section 2 we go through some preliminary concepts. In Section 3, we describe the combinatorial procedures which, given the input described in Section 2, are used to:

1. Construct a curve
2. Find bigons, if any
3. Detect if a bigon is proper
4. Remove a proper bigon from the curve

The logical flow of the algorithm is summarized in Section 4 and an example is worked out in full in Section 5. A few remarks: This algorithm runs in polynomial time, where the "units" are comparisons and permutations. The size of the input is the length of the curve in the fundamental group word metric. Thus, it is extremely fast to run on a computer. Secondly, the methods used here are sufficient to find a minimally intersecting configuration for a collection of two or more free homotopy classes, and only need to be slightly modified.

An earlier draft of this paper was the author's undergraduate thesis at Stony Brook, under the guidance of Moira Chas. The author is extremely thankful to her for the many insightful discussions, and for suggesting this interesting subject matter. The author would also like to thank Tony Phillips and Dennis Sullivan for their helpful comments. The author would finally like to thank the anonymous referee for their valuable corrections and suggestions.

2 Basics

We begin with a brief discussion of the basics and terminology used in this paper. Given a set of symbols $S = \{s_1, s_2, \dots, s_n, S_1, S_2, \dots, S_n\}$, we define a *surface word* to be a cyclic sequence of these $2n$ symbols where each symbol in S appears exactly once. Given a surface word $w_1 w_2 \dots w_{2n}$ with each $w_i \in S$, we associate to it a surface with boundary. Beginning with a polygon having $4n$ edges, we label the edges in a clockwise fashion as follows: Choose one edge and label it w_1 , leave the second edge unlabeled, label the third w_2 , the fourth

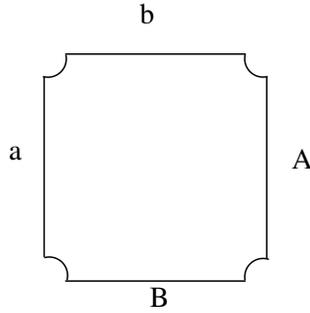


Figure 2: The surface word $abAB$ and the associated polygon. The unlabeled edges are curved to emphasize that they are part of the boundary.

unlabeled, and so on. This polygon, along with the labeling, will be called a *fundamental polygon*. To obtain a surface, we identify the edge labeled s_i with the edge labeled S_i in such a way as to preserve orientability. This identification gives rise to an orientable surface with boundary. In Figure 2, we consider the set of symbols $\{a, b, A, B\}$ with a identified with A , b identified with B . We will use this convention throughout the rest of this paper. The surface word $abAB$ then represents the torus with one boundary component.

After being given a surface word in some alphabet $\{s_1, s_2, \dots, s_n, S_1, S_2, \dots, S_n\}$, the fundamental group for the associated surface is a free group generated by $\{s_1, s_2, \dots, s_n\}$. The inverses of these generators are S_1, S_2, \dots, S_n . Given a word in the group $w_1 w_2 \dots w_n$, where each w_i is a generator, we can consider the word $w_{\sigma(1)} w_{\sigma(2)} \dots w_{\sigma(n)}$, where σ is a cyclic permutation. This collection of words forms an equivalence class called a *cyclic word*. A cyclic word is said to be *reduced* if no element and its inverse are adjacent for all cyclic permutations of the word. Each free homotopy class is given by a unique reduced cyclic word.

A *representative* is a choice of a particular map from the circle S^1 to our surface M .

Definition 2.1. A representative given by a map $f : S^1 \rightarrow M$ is said to contain a *monogon* if there is a closed subarc A of S^1 such that the endpoints of A are mapped to the same point on M and so that $f|_A$ determines a null-homotopic loop. \square

Definition 2.2. A representative given by a map $f : S^1 \rightarrow M$ is said to contain a *bigon* if two conditions hold: (1) there exists two closed subarcs A_1 and A_2 of S^1 such that the endpoints of A_1 are mapped to the same points on M as the endpoints of A_2 are; (2) The images of the two arcs bound a topological disk when we choose appropriate lifts to the universal cover of M . If the arcs A_1 and A_2 are disjoint, we call the bigon *proper*, and call it *improper* otherwise. \square

One would like to have a systematic way of creating representatives with the minimal number of self-intersections possible. This problem has been studied in the past, and in fact there is the following result which seems intuitively obvious but is not trivial to prove:

Theorem 2.3. [Hass and Scott 1985] *If a representative F does not have the minimal number of self-intersections, then it must have a monogon or a proper bigon.*

Once we have any starting representative, by the above theorem we simply need to look for monogons and proper bigons to homotope away. The algorithm to be described takes this "hands on" approach and translates it into a combinatorial method that a computer can easily perform.

3 The Algorithm

3.1 Basic Objects

The input for the algorithm will be a surface word X and a reduced cyclic word W . Consider these fixed for the present discussion. There is a systematic way of constructing a representative curve from these two pieces of data. From X , we build a polygon with the appropriate edge labelings. Next we choose distinct points along the edges of the polygon which our curve will pass through. There is a pair of points (on identified edges of the polygon) for each occurrence in W of a letter or its inverse. We then connect Euclidean line segments between the various chosen points according to the linear ordering given by W , ensuring that we end up with a single connected curve after the edge identifications are made. We call such a representative of the free homotopy class a *segmented representative*. To summarize:

Definition 3.1. A *segmented representative* corresponding to a reduced cyclic word W is a union of oriented Euclidean line segments passing through the interior of the fundamental polygon whose edge-crossing sequence corresponds to W , and whose endpoints are distinct and lie on the boundary of the polygon. \square

A segmented representative may or may not have triple points depending on the precise geometric positioning of the endpoints, but this will not be relevant in the proofs. Before moving on, we note this simple observation that will simplify later arguments:

Proposition 3.2. A *segmented representative* corresponding to a reduced cyclic word W cannot contain a monogon.

Proof. If an intersection x corresponds to a monogon, then we can find a sequence of directed Euclidean line segments starting and ending at x whose edge-crossing sequence is a trivial word. Since the fundamental group of our surface is a free group, there must be a pair of inverses adjacent to each other in this edge crossing sequence. That means there is a pair of inverses adjacent to each other in some cyclic permutation of W , contradicting the fact that W was reduced. \blacksquare

We now describe a way of associating to this representative, two combinatorial objects which encode the intersection structure. Consider the first letter in X , say it is a , and look at the corresponding edge of the fundamental polygon. Label the points along that edge through which the representative passes in a clockwise manner by a_1, \dots, a_k , where k is the total number of appearances of a and A in W . Since our curve is in general position, each point will be distinct. We must also label the paired points on the A edge in clockwise manner by A_k, \dots, A_1 . This is so that a_i and A_i really represent the same point on the surface. Proceed

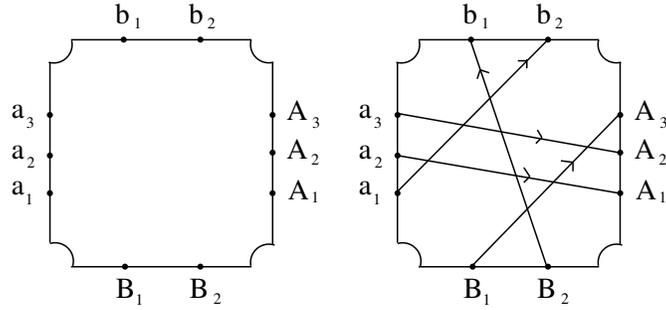


Figure 3: We start by placing the points in P , then connect them according to C to get a representative.

clockwise around the edges of the polygon until all points where the curve intersects an edge are labeled. This procedure leads to the following notions:

Definition 3.3. The cyclic list obtained by reading off the labeled points around the fundamental polygon in clockwise order is called a *point list*, denoted P . The elements in this list will be called *points*. \square

Definition 3.4. After choosing a labeling of points, we follow the orientation of the curve and sequence the pairs of points that are connected by line segments in another cyclic list, called a *segment list*, denoted C . We will call the pairs of points in C *word segments*. \square

Definition 3.5. Points labeled with the identical letter and index but opposite capitalizations are called *inverses* of each other. Hence $a_1^{-1} = A_1$, $B_1^{-1} = b_1$ and so on. \square

For example, if $X = abAB$ and $W = AAAbb$, using Figure 3, we have:

$$P = a_1, a_2, a_3, b_1, b_2, A_3, A_2, A_1, B_2, B_1 ; C = (B_1, A_3)(a_3, A_2)(a_2, A_1)(a_1, b_2)(B_2, b_1)$$

While P will initially be the same for every representative of $AAAbb$, C will vary depending on how these points are connected by line segments. The reader may also notice that the information in C is redundant - for example, a pair (x, A_i) will always be followed by a pair whose first element is a_i . This excess notation for C turns out to be convenient when describing later concepts. Conversely, just using the point list P , the segment list C and X from above, we may construct a segmented representative. Take the first element in P , say a_1 , and choose a point on the a edge of our polygon associated with X to label a_1 . This choice fixes our placement of the point labeled A_1 as well. Proceed clockwise around the edges of the polygon, choosing and labeling points in the correct relative positions on the edges on and the correct sides. Lastly, we use C to connect the appropriate points with line segments to complete the representative. Since the precise geometric positioning of the points to connect with segments was arbitrary (only the relative positioning was important), our new representative may not be exactly identical to the original one we started with. However, it will be shown that any two segmented representations constructed from the same P, C and X have the same number of intersections.

3.2 Finding Bigons

We may use the point list P and segment list C to determine if our representative has any bigons. Let $C = W_0W_1\dots W_{(n-1)}$ where each $W_i = (w_{i_1}, w_{i_2})$ is a word segment. Consider two word segments W_i and W_j . The associated line segments in a representative intersect if and only if the pair of points w_{i_1} and w_{i_2} separate w_{j_1} and w_{j_2} in the cyclic list P (this corresponds to the line segments being transverse to each other).

Definition 3.6. Two word segments $W_i = (w_{i_1}, w_{i_2})$ and $W_j = (w_{j_1}, w_{j_2})$ form a *combinatorial intersection*, if the pair of points w_{i_1} and w_{i_2} separate w_{j_1} and w_{j_2} in the cyclic list P . We say the two word segments *intersect*, and denote this combinatorial intersection by W_iIW_j \square

Immediately, we see that the number of intersections in a segmented representative determined by P and C depends only on P and C , and not on the specific geometric positioning of the points. We summarize this discussion in the following proposition:

Proposition 3.7. *Intersections in a representative are in one-to-one correspondence with combinatorial intersections.*

Now, given a combinatorial intersection W_iIW_j , we would like to determine if the corresponding point is a vertex of a bigon. To do this pictorially, we start at the intersection, and trace along pairs of line segments through repeated copies of the fundamental polygon in an attempt to identify the two "legs" of the bigon (see the example at the end of the paper to help clarify things). We try this for each of the four possible pairs of line segments until either the pair of line segments we were tracing intersect again, or the pair of segments don't intersect but split to different edges of the polygon. In the first case, we have found a bigon, while in the second case, we know we may stop tracing in that direction, since in the universal cover, the lifts would lead to different fundamental regions. In the case of surfaces without boundary, these lifts could still possibly meet up again to form a closed loop, but in the present case of surfaces with boundary, there is no possibility of the lifts meeting again. If the tracings split in all four directions with no intersection encountered, we conclude that our initial point of intersection cannot possibly be part of a bigon.

This bigon finding procedure just described can easily be done combinatorially. Starting with the intersection W_kIW_l , there are four possible pairs of word segments we may consider next. The four possible pairs of word segments we should look at are W_{k+1} and W_{l+1} , W_{k+1} and W_{l-1} , W_{k-1} and W_{l+1} , and finally W_{k-1} and W_{l-1} (all indices taken mod n). See Figure refCombBigon. To determine if the pair we are looking at intersects, we check the positions of the w_{i_j} s in P as described above. To determine if the pair splits to different sides, we also simply need to check the labelings of the appropriate w_{i_j} s. The ones to be checked should be obvious from Figure 4. If the current pair neither intersects nor splits, we must trace further, advancing or decreasing the indices of the word segments in the same manner we used to get to the current pair of word segments (i.e. if we were looking at W_{k-1} and W_{l+1} for example, we must next look at W_{k-2} and W_{l+2}). Continue checking in the given direction until an intersection is encountered, or no intersection is encountered but the segments split. In the first case, we again have found a bigon. In the second case, we must go back to our original

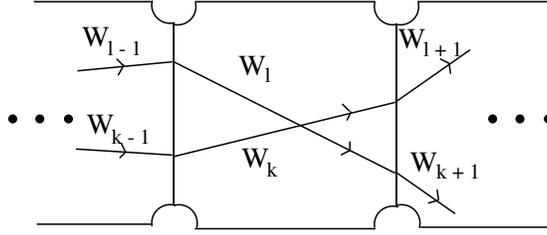


Figure 4: There are 4 directions to check in, but here, two of them split.

pair of word segments W_k and W_l and check in a different direction, until all directions are exhausted. If no direction results in finding another intersection, we conclude that our initial intersection W_kIW_l cannot be a vertex of a bigon.

Starting with the pair P,C, we check all pairs of word segments for potential vertices of bigons until the list of pairs is exhausted or a vertex is found. If no intersections are found then clearly our representative is minimal. If an intersection is found, the procedure described above determines whether the vertex is part of a bigon or not. If it is not, we continue checking the remaining pairs of word segments. If the procedure finds a bigon, the method above gives us two sequences of word segments that were checked, starting with the two segments determining the first intersection and ending with the two word segments determining the second intersection. There are four "orientations" associated with this pair of sequences, depending on if we moved in the forwards orientation for both W_k and W_l (this corresponds to the next pair of word segments being W_{k+1}, W_{l+1}), forwards for the first and reverse for the second (i.e. the next pair of word segments are W_{k+1}, W_{l-1}) and so on. These are denoted $(+, +)$, $(+, -)$, $(-, +)$, and $(-, -)$. Note, however, that any pair of sequences with orientation $(-, -)$ starting with W_k and W_l can be written as a pair of sequences with orientation $(+, +)$ and ending with W_k and W_l . Similarly, a pair of sequences with orientation $(-, +)$ may be considered to be of the form $(+, -)$ simply by switching the order of the two sequences. Thus, any pair of sequences determining a bigon essentially has orientation $(+, +)$ or $(+, -)$. This discussion leads to the following natural definition:

Definition 3.8. A *combinatorial bigon* with orientation $(+, +)$ is a pair of sequences of consecutive word segments in C W_i, W_{i+1}, \dots, W_k and W_j, W_{j+1}, \dots, W_l of equal size L , called the *length* of the bigon, so that:

1. $W_{i+n}IW_{j+n} \iff n = 0$ or $n = L - 1$
2. $w_{(i+n)_2} = w_{(j+n)_2} \forall n$ such that $0 < n < L - 1$

We will abbreviate these two sequences of word segments as $W_{i\dots k}, W_{l\dots j}$ □

These conditions capture the intuitive idea of a pair of line segments intersecting, then fellow traveling without crossing for some time, and finally intersecting again. A similar definition can be given for combinatorial bigons with orientation $(+, -)$, with the indices adjusted to deal with the opposite orientation of the 2nd curve. We summarize this discussion with the following:

Proposition 3.9. *Bigons in a segmented representative are in one-to-one correspondence with combinatorial bigons*

Corollary 3.10. *If a segmented representative has no combinatorial bigons, then it has the minimal number of self-intersections.*

3.3 Removing Bigons

Once we have a bigon, we would like to remove it as illustrated in Figure 5. The homotopy which removes the bigon of the representative induces a permutation on the elements in P . It is easy to see what the permutations are if we break the homotopy into separate steps as in Figure 5. Conversely, certain permutations of points in P correspond to a homotopy in our actual representative. Given a combinatorial bigon $W_{k\dots i}, W_{l\dots j}$, we look at the pairs of word segments W_k, W_l , followed by $W_{k+1}, W_{l\pm 1}$, followed by $W_{k+2}, W_{l\pm 2}, \dots, W_i, W_j$ (with the sign of \pm depending on whether orientation is $(+, +)$ or $(+, -)$) and within each pair of word segments, choose the appropriate points to permute.

Which points to permute clearly depends on which orientation our combinatorial bigon has:

For $m \in \{1, 2, \dots, L - 1\}$ where L is the length of the bigon.

1. $(+, +)$: switch $w_{(k+m)_2}$ with $w_{(l+m)_2}$ and $w_{(k+m)_1}$ with $w_{(l+m)_1}$
2. $(+, -)$: switch $w_{(k+m)_2}$ with $w_{(l+m)_1}$ and $w_{(k+m)_1}$ with $w_{(l+m)_2}$

This does not remove self-intersections in all cases. There are certain conditions on the combinatorial bigon $W_{k\dots i}, W_{l\dots j}$ that determine whether the above permutations remove intersections or not. If no W_{k+m} equals any $W_{l+m'}$, $m, m' \in \{1, 2, \dots, L - 1\}$ (i.e., the two combinatorial legs contain no word segments in common), then we have the following important theorem:

Theorem 3.11. *If the two combinatorial legs of a bigon $W_{k\dots i}, W_{l\dots j}$ contain no word segments in common, then the permutation procedure removes the initial pair of intersections of the bigon and creates no additional intersections*

Proof. We first note that each pair of intermediate segments $W_{k+1}W_{l\pm 1}, \dots, (W_{i-1}W_{j\pm 1})$ becomes crossed, and then uncrossed as the permutations run their course, and in the end all of the intermediate word segments will be swapped (see Figure 5). This clearly contributes no additional intersections. Since the word segments in the combinatorial bigon are distinct, the permutations of the terminal pairs of points do not affect one another, so that W_k no longer intersects W_l , and likewise W_i no longer intersects W_j .

The only thing left to be checked is that no new intersections are formed during this process. To see this, consider Figure 6 where (a, b) and (c, d) represent the terminal segments of the bigon W_k and W_l . We assume our bigon has orientation $(+, +)$ since the argument is nearly identical for the $(+, -)$ case. This means that the final permutation to remove the bigon will swap the locations of the points labeled b and d .

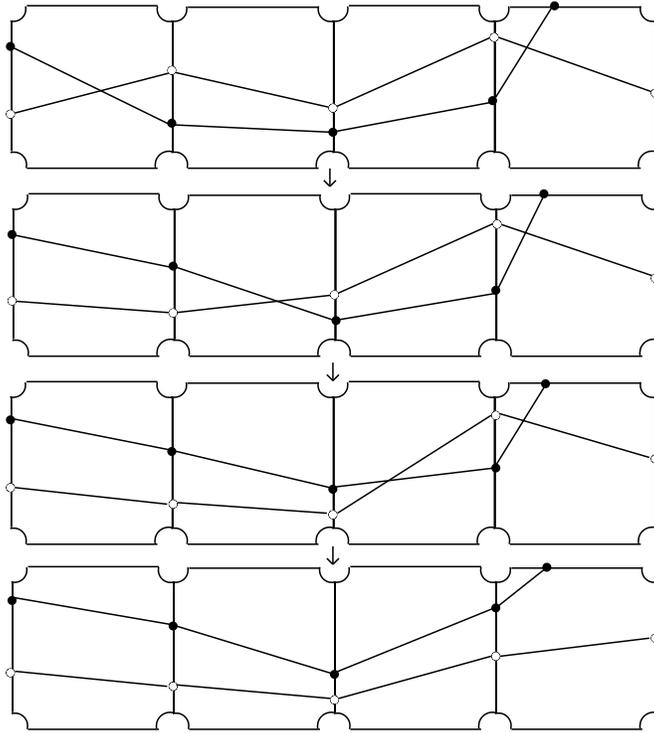


Figure 5: Geometrically, we are homotoping the bigon away. Combinatorially, we are switching the locations of the black and white points in P .

Now suppose that there is another segment (x, y) which is fixed by the permutation swapping b and d , which intersects (a, b) after permuting the appropriate points, but not before.

Since (x, y) did not intersect (a, b) before permuting b and d , the points a and b do not separate x from y along the boundary of the fundamental polygon. Since (x, y) intersects (a, b) after permuting b and d , the segment (c, d) must have intersected (x, y) to begin with. At once we see that after permuting b with d , (x, y) no longer intersects (c, d) . Thus, for every intersection involving (a, b) that we create, we remove an intersection with (c, d) . We arrive at a symmetric statement for intersections involving (c, d) and other word segments. A nearly identical argument works using the segments W_i and W_j , and so to avoid repetition we leave it to the curious. Combining the facts we have collected, we arrive at the desired result. ■

If one or more word segments are shared, then the permutations just described may or may not reduce the number of intersections. For instance, consider the situation depicted in Figure 7. The indicated combinatorial legs share a word segment, and upon permuting the points as described above, we do not actually remove any intersections. We call such combinatorial bigons *non-removable*. It will later be shown that non-removable combinatorial

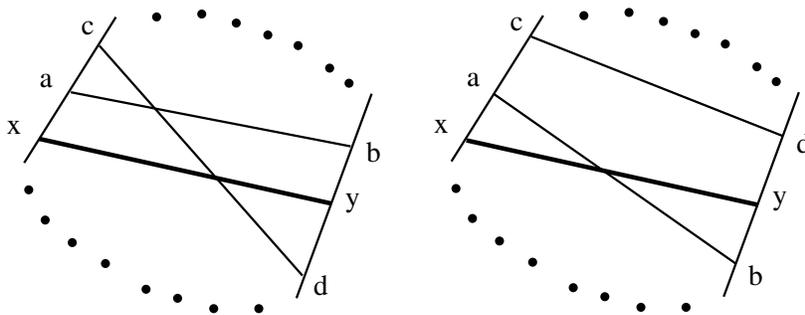


Figure 6: The net number of intersections decreases. The dotted curves represent the extra edges not involved in this particular collection of word segments.

bigons correspond to improper bigons in the representative. Before proceeding further, we present the following propositions which characterize the combinatorial symmetry that bigons with one or more equal segments in both combinatorial legs must have.

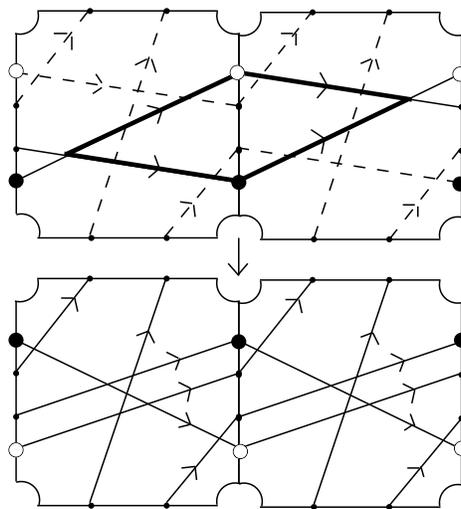


Figure 7: The surface is $abAB$, and the reduced word is $bbAAAA$. Switching the indicated points does not reduce the number of intersections.

Proposition 3.12. *Let $W_{k\dots i}, W_{l\dots j}$ be a combinatorial bigon.*

1. *If the combinatorial bigon has exactly 1 word segment shared by both combinatorial legs, then it must be the case that $W_k = W_j$ or $W_i = W_l$*
2. *If there are exactly 2 word segments shared by both combinatorial legs, then they must either be the first two segments of one combinatorial leg and the last two of the other, or the first and last of both.*

3. *If there are more than 2 word segments shared by both combinatorial legs, then at least the first two word segments of one leg must be equal to the last two of the other leg.*

Proof. Let $f : S \rightarrow M$ be any choice of a segmented representative based on the point list P and segment list C that we have fixed for the moment. The preimages of the points where our representative crosses the edges of the polygon (i.e. the preimages of the points $w_{k_1}, w_{k_2}, w_{l_1}, w_{l_2}$, etc.) partition the circle into N arcs, where N is the length of the reduced cyclic word used to construct our representative. Each of the combinatorial legs can then be associated to N' of these arcs on the circle, where N' is the combinatorial length of the bigon. Let L_1 and L_2 be the arcs on the circle corresponding to the combinatorial legs. With this picture in mind, let us prove statement 1. Since we are assuming that a word segment is shared between the legs of the bigon, L_1 and L_2 must intersect. The two arcs cannot be equal since the two legs of a bigon cannot be identical. It also cannot be the case that one arc is properly contained inside the other either, for then the two combinatorial legs could not have the same length. The only possible is that the intersection must occur at the end of the arcs, which is exactly what statement 1 says. For statements 2 and 3 the argument is again very simple, except now there is the possibility the arcs L_1 and L_2 could wrap around and intersect on opposite sides of the circle. However, this situation is covered in the proposition. ■

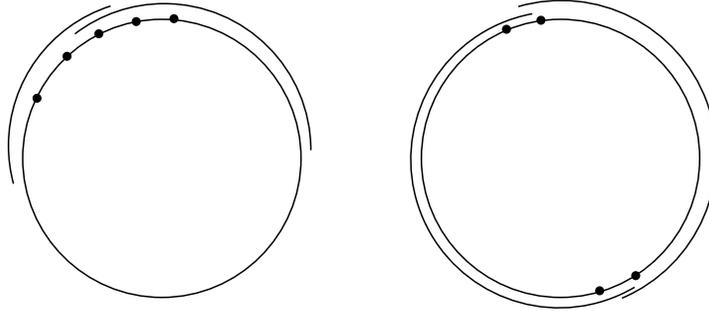


Figure 8: Two possible overlapping configurations. The points on the circle are preimages of the points where the segmented representative crosses an edge of the polygon. The arcs around the circle represent the preimages of the legs of the bigon.

Now we show which types of combinatorial bigons always give rise to improper bigons in any representative, regardless of how the points are spaced on the fundamental polygon. By Theorem 2.3 we can then simply skip all combinatorial bigons of these types when searching for removable bigons. It happens that only a relatively small number of essential cases remain, and it will be shown that the permutations strictly reduce the number of intersections in these situations. By Theorem 3.11, non-removable bigons must have at least one word segment shared by both combinatorial legs. Furthermore, any such bigon must have orientation $(+, +)$, for otherwise, the segments shared by both are seen to inherit opposite orientations at the same time - clearly impossible.

Theorem 3.13. *Let $W_{k\dots i}, W_{l\dots j}$ be a combinatorial bigon. If one of the following conditions holds, then the corresponding bigon in any segmented representative with the same P and C*

is improper:

1. At least two word segments are shared by both combinatorial legs
2. One segment is shared by both combinatorial legs, say $W_k = W_j$ and w_{l_1} is between w_{i_1} and w_{k_1} , and w_{i_2} is between w_{l_2} and w_{k_2} in P .

Proof. First choose a segmented representative and a fundamental polygon based on P and C . Let one leg of the geometric bigon be denoted L_1 and the other L_2 . The preimages of all the points where our representative crosses the edges of the polygon (i.e. the preimages of the points $w_{k_1}, w_{k_2}, w_{l_1}, w_{l_2}$, etc.) partition the circle into N arcs, where N is the length of the reduced cyclic word used to construct our representative. Suppose that there are at least two word segments shared by both combinatorial legs. This is equivalent to saying that the intersection $I = f^{-1}(L_1) \cap f^{-1}(L_2)$ is not entirely contained in a single partitioning arc. The only deformations of our segmented representative allowed are sliding the endpoints of segments along the edges of the fundamental polygon without changing P . Any such homotopy of the line segment positions leaves the combinatorial description of the bigon unchanged. Therefore we still have that I is not contained within a single partitioning arc. With our construction, the endpoints of the preimages of L_1 and L_2 must be contained in the interiors of the dividing arcs - otherwise we would have two line segments emanating from the same point on an edge of the polygon. Since the endpoints of $f^{-1}(L_1)$ and $f^{-1}(L_2)$ must be contained in the interiors of dividing arcs, and since I is not contained within a single dividing arc, I cannot be empty. Figure 8 shows this situation.

This covers all cases except when exactly two word segments are shared and that they occur at the ends of both combinatorial legs i.e. $W_k = W_j$ and $W_i = W_l$. From this we see that our bigon really only has one vertex, and thus that our bigon must be improper. For the

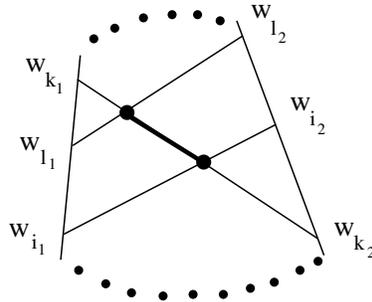


Figure 9: A configuration depicting an improper bigon. The dotted curves represent the extra edges not involved in this particular collection of word segments.

2nd condition above, our representative locally looks like Figure 9. The only way to remove the overlap is to translate one line segment over another, which clearly changes P . Thus our combinatorial condition guarantees that the bigon will be improper. ■

Theorem 3.14. *Let $W_{k\dots i}, W_{l\dots j}$ be a combinatorial bigon with $W_k = W_j$, and no other segments shared. Then the bigon is removable if the 2nd condition from Theorem 3.13 is not satisfied.*

Proof. As in Theorem 3.11 we only need to consider the 6 points in the terminal word segments, W_k, W_l , and W_i , since the intermediate segments are swapped. We consider all possible relative positions of those 6 points within P that simultaneously realize:

1. $(w_{k_1}, w_{k_2})I(w_{l_1}, w_{l_2})$
2. $(w_{k_1}, w_{k_2})I(w_{i_1}, w_{i_2})$
3. w_{k_2} and w_{l_2} on the same edge
4. w_{k_1} and w_{i_1} on the same edge.

These four conditions are necessary in order for our pair of sequences to actually be a combinatorial bigon. There are 12 configurations of points satisfying these conditions, and only 6 that need be considered once symmetry is taken into account. To see this, consider an oriented circle representing the cyclic ordering of P. We imagine placing the initial line segment, say (w_{k_1}, w_{k_2}) which then divides the circle into two arcs. Next, there are two choices for the placement of the line segment (w_{l_1}, w_{l_2}) , corresponding to which arc of the circle we wish to place the point w_{l_1} . For a given placement of (w_{l_1}, w_{l_2}) , the circle is divided into four arcs. The point w_{i_1} may be placed on any arc except the one between the points w_{k_2} and w_{l_2} (that would imply w_{i_1} is not on the same edge as w_{k_1} or that w_{k_2} and w_{l_2} are on the same edge, contrary to conditions 3 or 4. Once w_{i_1} is placed, there are two choices for w_{i_2} that give $(w_{k_1}, w_{k_2})I(w_{i_1}, w_{i_2})$. Counting all combinations, we get $2 * 3 * 2 = 12$ total, but we can eliminate half of the sequences, since one ordering and its reverse ordering are equivalent for the purpose of determining the effect of the permutations.

Below, the 6 essential orderings and total number of intersections they determine are given, along with their ordering and intersections after the permutations are performed:

1. $w_{k_1}-w_{i_1}-w_{l_2}-w_{k_2}-w_{i_2}-w_{l_1}$ 3 $w_{i_1}-w_{k_1}-w_{k_2}-w_{l_2}-w_{i_2}-w_{l_1}$ 1
2. $w_{k_1}-w_{i_1}-w_{l_2}-w_{k_2}-w_{l_1}-w_{i_2}$ 2 $w_{i_1}-w_{k_1}-w_{k_2}-w_{l_2}-w_{l_1}-w_{i_2}$ 0
3. $w_{k_1}-w_{l_2}-w_{i_2}-w_{k_2}-w_{l_1}-w_{i_1}$ 3 $w_{i_1}-w_{k_2}-w_{i_2}-w_{l_2}-w_{l_1}-w_{k_1}$ 1
4. $w_{k_1}-w_{i_2}-w_{l_2}-w_{k_2}-w_{l_1}-w_{i_1}$ 2 $w_{i_1}-w_{i_2}-w_{k_2}-w_{l_2}-w_{l_1}-w_{k_1}$ 0
5. $w_{k_1}-w_{i_2}-w_{l_2}-w_{k_2}-w_{i_1}-w_{l_1}$ 3 $w_{i_1}-w_{i_2}-w_{k_2}-w_{l_2}-w_{k_1}-w_{l_1}$ 1
6. $w_{k_1}-w_{l_2}-w_{i_2}-w_{k_2}-w_{i_1}-w_{l_1}$ 2 $w_{i_1}-w_{k_2}-w_{i_2}-w_{l_2}-w_{k_1}-w_{l_1}$ 2

Case 6 satisfies the second condition in Theorem 3.13 and so determines an improper bigon, so it is not surprising that the permutations do not reduce intersections. The only thing that remains to be proven is that for each of the 5 "good" cases, no additional intersections are produced. To make things easier to keep track of, let us relabel the word segments as follows:

$W_k = (a, b), W_l = (c, d)$, and $W_i = (e, f)$. We'll prove the theorem for the first case above by following a line of reasoning similar to that in Theorem 3.11. Careful consideration of Figure 10 yields the proof, but we give some of the details here.

Suppose a word segment $(x, y) \neq (a, b), (c, d)$ nor (e, f) , does not intersect (a, b) initially, but does after permuting. Then (x, y) must have intersected one of (e, f) or (c, d) beforehand. To see this, note that one of the points x or y must lie in the "top" arc between e and d , while the other point must have been either between a and e , or d and b . However, after the permutations, it cannot intersect either of them, as seen from the figure and the argument in the previous sentence. Now suppose (x, y) intersects (c, d) after permuting but not before. Since the point c is unchanged by a permutation, we see that (x, y) must have intersected both (a, b) and (e, f) before the permutations. But after the permutations, (x, y) could not possibly intersect (a, b) , since then (x, y) would have intersected (c, d) to begin with. Finally, suppose (x, y) intersects (e, f) after the permutations but not before. The point f is fixed by the permutations, and we see that (x, y) must have intersected both (a, b) and (c, d) before the permutations. By the exact same reasoning as in the previous situation, (x, y) can no longer intersect (a, b) after the permutations.

In every situation, if an intersection is introduced, there is a corresponding intersection that is removed. Thus there is no net gain of intersections between (x, y) and the word segments $(a, b), (c, d)$ and (e, f) . Since the word segment (x, y) was arbitrary, we proved the theorem for case 1. The proofs for the other 4 "good" cases are nearly identical, so we leave them to the curious. ■

4 Summary

Once given a surface word and a reduced word, we construct the point list P and segment list C in whatever appropriate manner we prefer. Then we examine P and C to determine if there are any combinatorial bigons. If there are none, then by Proposition 3.9 and Theorem 2.3 our current representative is minimal, and we are done. If we find a bigon which satisfies one of the conditions Theorem 3.13, then we skip over it and search for more bigons. If every bigon is of one of the types described in Theorem 3.13 then again we are done, by the earlier

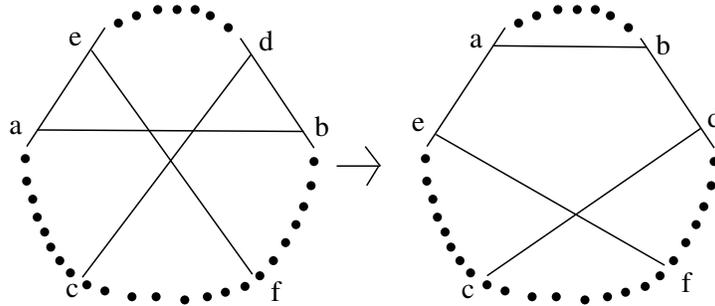


Figure 10: The new configuration after permuting the appropriate points. The dotted curves represent the irrelevant edges of the fundamental polygon.

propositions and Theorem 2.3. Finally, if we find a bigon not of the types in Theorem 3.13, then by Theorem 3.14 we know we may apply the permutations and reduce the total number of intersections encoded in P and C . This process must terminate so we eventually end up with a modified point list P , which along with our original segment list C , encodes a minimal segmented representative.

5 Example

We will now show how this algorithm works for the surface word \mathbf{abAB} and the reduced cyclic word \mathbf{bbAAA} , the same surface and curve as in Figure 3. Recall that in Figure 3, we have chosen $P = a_1, a_2, a_3, b_1, b_2, A_3, A_2, A_1, B_2, B_1$ and $C = (B_1, A_3)(a_3, A_2)(a_2, A_1)(a_1, b_2)(B_2, b_1)$ as our initial representative. We now check through pairs of word segments until we find a pair that intersect. Upon inspection, we see that $(a_3, A_2)I(B_2, b_1)$. The next step is to see if this vertex can possibly be part of a bigon. There can be no bigon starting at this vertex with a $(+, +)$ orientation, since the segments spit to the A and b sides. Likewise for the other 3 orientations. We conclude that this particular intersection cannot be a vertex of a bigon and move on until we find another pair of segments that intersect. Suppose the next pair of segments we find to intersect are (B_2, b_1) and (a_1, b_2) . For this particular intersection, we see that it may be a vertex of a bigon with a $(+, +)$ orientation, since the points b_1 and b_2 are on the same edge. The next pair of segments we compare are (B_1, A_3) and (B_2, b_1) , which also intersect. Thus we have a combinatorial bigon $\{(B_2, b_1)(B_1, A_3); (a_1, b_2), (B_2, b_1)\}$. Since (B_2, b_1) is shared by both sequences, we must use Theorem 3.14 to check if we can remove this bigon to reduce the number of self-intersections. We see that this particular bigon is of type (5) from the proof of Theorem 6 in reverse, and so we may proceed. We need to switch b_1 with b_2 and B_1 with B_2 , as shown in Figure 11.

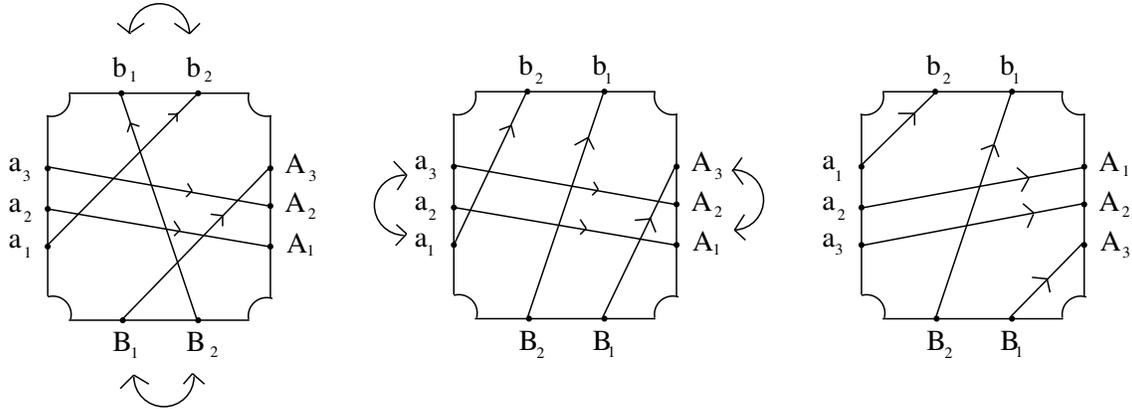


Figure 11: Sequence of permutations removing all of the bigons.

Now, we have $P = a_1, a_2, a_3, b_2, b_1, A_3, A_2, A_1, B_1, B_2$, while C remains the same. We must again compare pairs of segments until we find an intersection. We see that $(a_1, b_2)I(a_2, A_1)$, so we check to see if this intersection can be a vertex of a bigon. A $(-, -)$ orientation is ruled out, since b_2 and A_1 are on different edges, but we see that we may check for a $(+, +)$ orientation.

Doing so results in the combinatorial bigon $\{(a_1, b_2), (a_2, A_1), (a_3, A_2); (a_2, A_1), (a_3, A_2), (B_1, A_3)\}$. This bigon has two segments shared by both sequences, so it is guaranteed to be improper by Theorem 3.13. By Theorem 2.3 there must be a proper bigon if our representative does not have minimal self-intersection, so we continue to look for a different bigon to potentially remove. Suppose we next find that $(a_1, b_2)I(a_3, A_2)$. We find that it is the vertex of the bigon $\{(a_1, b_2), (a_2, A_1); (a_3, A_2), (B_1, A_3)\}$. We switch a_1 with a_3 , and A_1 with A_3 , as shown in the second step of Figure 11. Finally, we once again check all pairs of segments using the permuted P , and determine that there are no more proper bigons, and thus by Theorem 2.3 our representative has the minimal number of self-intersections possible. The final output of the algorithm is $P = a_3, a_2, a_1, b_2, b_1, A_1, A_2, A_3, B_1, B_2$ and $C = (B_1, A_3)(a_3, A_2)(a_2, A_1)(a_1, b_2)(B_2, b_1)$, which determines our representative.

References

- [1] J. Birman and C. Series, An algorithm for simple curves on surfaces, J. London Math. Soc. Vol 29 Iss 2 (1984), 331-342.
- [2] M. Chas, Combinatorial Lie Bialgebras of curves on surfaces, Topology Vol 43 Iss 3 (2004), 543-568
- [3] M. Chas, Minimal intersection of curves on surfaces, Geometriae Dedicata Vol 144 (2010), 25-60
- [4] D. Chillingworth, Simple closed curves on surfaces, Bull. London Math. Soc. Vol 1 (1969), 310-314
- [5] M. Cohen and M. Lustig, Paths of geodesics and geometric intersection numbers I, Combinatorial Group Theory and Topology, Ann. of Math. Studies Vol 111 (1987), 479-500.
- [6] J. Hass, P. Scott, Intersections of curves on surfaces, Israel J. Math. Vol 51 (1985), 90-120
- [7] J. Hass, P. Scott, Shortening curves on surfaces, Topology Vol 33 No 1 (1994), 25-43
- [8] B. L. Reinhart, Algorithms for Jordan curves on compact surfaces, Annals of Math. Vol 75 (1962), 209-222