Algorithms for Normal Curves and Surfaces

Marcus Schaefer[1], Eric Sedgwick[2], and Daniel Štefankovič[3]

[1] DePaul University (mschaefer@cs.depaul.edu)
[2] DePaul University (esedgwick@cs.depaul.edu)
[3] University of Chicago (stefanko@cs.uchicago.edu)

Abstract. We derive several algorithms for curves and surfaces represented using normal coordinates. The normal coordinate representation is a very succinct representation of curves and surfaces. For embedded curves, for example, its size is logarithmically smaller than a representation by edge intersections in a triangulation. Consequently, fast algorithms for normal representations can be exponentially faster than algorithms working on the edge intersection representation. Normal representations have been essential in establishing bounds on the complexity of recognizing the unknot [Hak61, HLP99, AHT02], and string graphs [SSŠ02]. In this paper we present efficient algorithms for counting the number of connected components of curves and surfaces, deciding whether two curves are isotopic, and computing the algebraic intersection numbers of two curves. Our main tool are equations over monoids, also known as word equations.

1 Introduction

Computational topology is a recent area in computational geometry that investigates the complexity of determining properties of topological objects such as curves and surfaces [BE$^+$99, DEG99]. For example, it is known that we can decide whether two curves on a surface are homotopic in linear time if the surface is represented by a triangulation, and the curves as sequences of intersections with the triangulation [DG99].

In 1930 Kneser [Kne30] introduced a representation for curves and surfaces in which these objects are described by their *normal coordinates.* This led to the theory of normal surfaces which was used by Haken in 1961 to show that the unknot could be recognized by an algorithm (which, much later, was shown to run in exponential time). Haken's approach was pushed further by Hass, Lagarias, and Pippenger who showed that the unknot could be recognized in **NP** [HLP99]. To this end they had to verify in polynomial time that a special type of normal surface was an essential disk. The result of [HLP99] was recently extended by Agol, Hass, and Thurston [AHT02]. The main contribution of [AHT02] was a polynomial time algorithm for computing the number of connected components of a normal surface. This immediately implies polynomial time algorithms for checking whether a normal surface is connected, and whether it is orientable.

The theory of normal curves is much simpler than the theory of normal surfaces. Nevertheless, it was one of three essential ingredients in the proof that string graphs can be recognized in **NP**, a problem that had only recently been shown to be decidable at all [SSŠ02].

With this paper we attempt to initiate a more systematic study of algorithms for normal curves and surfaces. We believe that the examples mentioned earlier show that this is a worthwhile attempt. There is also a more theoretical justification of this approach: we get a clearer picture of the computational complexity of a problem if we ensure that the object representations are *succinct*, meaning that they are not compressible. It is easy to show that the representation of an embedded curve as a sequence of intersections with a triangulation is never succinct: if we consider it as a word over an alphabet made up of the edges of the triangulation, it can always be compressed to polylogarithmic size (see Section 4.1). For embedded curves this means that the normal coordinates are a more succinct and natural way of representation.

Among other things we will give efficient algorithms to count components of curves and surfaces, decide whether two curves are isotopic, and compute the algebraic intersection number of a pair of curves.

A final word about technique. We extend the ideas from [SSŠ02] where we based our algorithms on word equations. We believe the connection between word equations and normal curves and surfaces is a very strong one, and, as far as we can tell, has not been observed before. There have, of course, been algorithms based on words in the fundamental group of a manifold, but words in this context are always words in a group. The original observation here is that the recent, and powerful, results about computational aspects of monoids can be applied to embeddings and immersions of curves and surfaces [PR98, GKPR96, Ryt99, DM01]. For example, the algorithm for counting components suggested by Agol, Hass, and Thurston is quite involved, whereas our algorithm follows quite naturally from the decidability of word equations with prescribed lengths by using some topology. A second example is the recognition problem for string graphs. Currently this problem can only be solved with the help of algorithms for trace monoids [DM01, SSŠ02].

2 Normal Coordinates

In the following M will always be a compact orientable 2-manifold with boundary ∂M, unless stated otherwise. Let b be the number of the boundary components of M and g the genus of M. The Euler characteristic of M is $\chi = 2 - 2g - b$.

A *simple arc* is a homeomorphic image of the interval $[0, 1]$. A simple arc $\gamma : [0, 1] \to M$ such that both its endpoints $\gamma(0), \gamma(1)$ are on the boundary ∂M and the internal points $\gamma(x), 0 < x < 1$ are in the interior of M is called a *properly embedded arc*. A simple closed arc $\gamma : [0, 1] \to M$, $\gamma(0) = \gamma(1)$ such that all the points $\gamma(x), 0 \le x \le 1$ are in the interior of M is called a *properly embedded circle*. A *curve* is an embedded collection of properly embedded arcs and properly embedded circles. Note that a curve cannot be self intersecting, since it is embedded.

Two curves γ_1, γ_2 are *isotopic rel boundary* ($\gamma_1 \sim \gamma_2$) if there is a continuous deformation of γ_1 to γ_2 which does not move the points on the boundary ∂M and during the deformation the curve stays embedded. From now on by isotopy we mean isotopy rel boundary.

Let $T = (V_T, E_T)$ where V_T is a set of points in M and E_T is a embedded set of embedded arcs in M with both endpoints in V_T. T is a *triangulation* of M if every connected component C of $M - E_T$ is homeomorphic to an open disc and in a closed walk along the boundary of C we meet 3 points from V_T (not necessarily distinct). If the manifold M has non-empty boundary and $V_T \subseteq \partial M$ then we call the triangulation T *minimal*.

Let T be a triangulation of M. Let γ be a curve. We say that γ is *normal* w.r.t. T if all the intersections with T are transversal and if γ enters a triangle $t \in T$ via an edge e then it leaves t via an edge different from e (i.e no two consecutive intersections of γ with T belong to the same edge $e \in T$).

Let γ be a curve in M. Let $c \sim \gamma$ minimize the number of intersections with T. If c enters and leaves some $t \in T$ through the same edge e then the number of intersections of c with T can be reduced by pulling the curve c, a contradiction. Hence for each non-trivial curve γ in M there is $c \sim \gamma$ which is normal w.r.t. T. If γ is an arc we can fix one of its endpoints as the initial point of γ, and number the intersection points of γ with T in order (starting with 0). We call this number the *index* of an intersection point along γ.

An isotopy which for each edge $e \in T$ maps points in e to points in e is called a *normal isotopy*. If the triangulation T is minimal then every isotopy is a normal isotopy.

Given a curve γ in normal position w.r.t. T we can write on each edge of the triangulation how many times γ intersects it. Such a representation is called a *representation using normal coordinates*. In each triangle the numbers written on its sides determine the behavior of the curve inside the triangle up to isotopy (for this we need that the curve is properly embedded and in normal position w.r.t. T). Two curves which have the same numbering are normally isotopic (assuming that the positions of the points on the boundary agree).

The size of the representation using normal coordinates is the total bitlength of the labels. The number of intersections of γ with $e \in T$ will be denoted $\gamma(e)$.

The segments into which an edge $e \in T$ is cut by γ are called *ports*. The ports are representatives of points in $M - \gamma$. A port is given by $(u, v) \in T$ on which it occurs and a number from $\{0, \ldots, \gamma(e)\}$ encoding its order on (u, v). Similarly an intersection point of γ with T is given by $(u, v) \in T$ on which it occurs and a number from $\{0, \ldots, \gamma(e) - 1\}$.

3 Results

The goal of this paper is to prove the following results.

Theorem 1. *Let M be a compact 2-manifold. Let T be a triangulation of M. Let γ, δ be curves in M given by normal coordinates w.r.t. T. Let p, q be ports of γ in T. Let r be an intersection point of γ with T. The following problems can be solved in polynomial time.*

(a) *Find the normal coordinates of the connected component γ containing r.*
(b) *Count the number of connected components of γ.*
(c) *List all the non-isotopic connected curves $\gamma_1, \ldots, \gamma_n$ which occur in γ. For each γ_i find the number of occurrences of γ_i in γ.*
(d) *Decide whether γ and δ are isotopic (assuming γ, δ are normal isotopy disjoint).*
(e) *Decide whether ports p, q are in the same connected component of $M - \gamma$. If they are in the same connected component find an embedded arc in $M - \gamma$ connecting them.*
(f) *Compute the algebraic intersection number of γ, δ.*

(g) *If γ is a properly embedded arc, we can for each intersection point with T compute its index along γ (arbitrarily declaring one of γ's endpoints the first intersection).*

Theorem 2. *Let M be a compact 2-manifold. Let T be a triangulation of M. Let γ, δ be curves in M given by normal coordinates w.r.t. T. The following problems can be solved by a Las Vegas algorithm (an algorithm solving the problem in expected polynomial time with zero probability of error).*

(a) *Decide whether γ and δ are isotopic (assuming $\partial M \neq \emptyset$).*
(b) *Locate the intersection point with index n of γ with T.*

Problem (b), and therefore (a) (by reduction), can be solved in polynomial time, but the proof is more complicated, and we do not include it here. Part (b) of Theorem 1 can be extended to normal surfaces; that is, we can count the number of connected components of a surface given in normal coordinates in polynomial time. This result was first shown in [AHT02], however, our proof is simpler, and is based on ideas developed independently in [SSŠ02]. For lack of space, we do not include the proof in this version.

4 Word Equations

Let Σ be an *alphabet*. Words in Σ^* are represented by *straight line programs* (SLP), a special type of context free grammar. A *straight line program P* of length n is a sequence of assignments $x_i = \text{expr}$ for $1 \leq i \leq n$ where expr is either a symbol from Σ or $x_j x_k,\ 1 \leq j, k < i$. For a word w given by SLP we denote the length n of the program by $|SLP(w)|$. Note that there are strings (e.g. a^m) for which $|SLP(w)|$ is exponentially smaller than $|w|$.

Lemma 1 ([GKPR96]). *Let p, t be words represented by SLP's. The following two problems can be solved in time polynomial in $|SLP(p)|$ and $|SLP(t)|$. How many times does p occur in t? What is the position of the first occurrence of p in t?*

Lemma 2. *Let t be a word given by an SLP. The following problem can be solved in time polynomial in $|SLP(t)|$. Given two positions $i \leq j$ in t find an SLP for the substring $t[i \dots j]$ of t.*

Let Θ be an alphabet of variables disjoint from Σ. A *word equation* $u = v$ is a pair of words $(u, v) \in (\Sigma \cup \Theta)^* \times (\Sigma \cup \Theta)^*$. The *size of the equation* $u = v$ is $|u| + |v|$. A *solution of the word equation* $u = v$ is a morphism $h : (\Sigma \cup \Theta)^* \to \Sigma^*$ such that $h(a) = a$ for all $a \in \Sigma$ and $h(u) = h(v)$ (h being a morphism means that $h(wz) = h(w)h(z)$ for any $w, z \in (\Sigma \cup \Theta)^*$). The *length of the solution h* is $\sum_{x \in \Theta} |h(x)|$. A *word equation with specified lengths* is a word equation $u = v$ and a function $f : \Theta \to \mathbb{N}$. The solution h has to respect the lengths, i.e. we require $|h(x)| = f(x)$ for all $x \in \Theta$. Let $h : (\Sigma \cup \Theta)^* \to \Sigma^*$ be a solution of an equation $u = v$. The SLP encoding of h is the sequence of SLP encodings of $h(x)$ for all $x \in \Theta$. The size of the encoding is $|SPL(h)| = \sum_{x \in \Theta} |SLP(h(x))|$. The usefulness of SLP encoding for word equations is demonstrated by following result.

Theorem 3 ([PR98]). *Let $u = v$ be a word equation with lengths specified by a function f. Assume that $u = v$ has a solution respecting f. The $SLP(h)$ of the lexicographically least solution h can be found in polynomial time in the size of the equation and the size of the binary encoding of f.*

4.1 Curve Coloring Equations

In this section we construct a system of word equations with given lengths which will allow us to color connected components of a curve γ on M normal w.r.t. a triangulation T. The alphabet used in the word equations encodes the colors. Let $t \in T$ be a triangle with vertices u, v, w. We add the following twelve variables to the system.

$$x_{t,(u,v)}, x_{t,(v,u)}, x_{t,(u,w)}, x_{t,(w,u)}, x_{t,(v,w)}, x_{t,(w,v)}, y_{t,u}, y_{t,v}, y_{t,w}, y_{u,t}, y_{v,t}, y_{w,t} \tag{1}$$

The variable $x_{t,(u,v)}$ encodes the order in which the colors occur on (u, v). We specify $|x_{t,(u,v)}| = \gamma(u, v)$. The variable $y_{t,u}$ encodes the colors of the (directed) segment of $x_{t,(u,v)}$ whose edges pass from (w, u) to (v, u). See Figure 1.

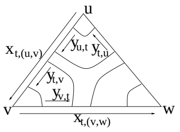

Fig. 1. Some of the variables for triangle t.

The following equations are called *triangle constraints*.

$$\begin{array}{lll} x_{t,(u,v)} = y_{u,t}y_{t,v} & x_{t,(v,u)} = y_{v,t}y_{t,u} & x_{t,(v,w)} = y_{v,t}y_{t,w} \\ x_{t,(w,v)} = y_{w,t}y_{t,v} & x_{t,(u,w)} = y_{u,t}y_{y,w} & x_{t,(w,u)} = y_{w,t}y_{t,u} \end{array} \tag{2}$$

Note that the lengths of the x variables determine the lengths of the y variables, for example $|y_{u,t}| = (|x_{t,(u,v)}| + |x_{t,(u,w)}| - |x_{t,(v,w)}|)/2$. For each edge (u, v) which is contained in two triangles $s, t \in T$ we add the following *edge constraint*.

$$x_{s,(u,v)} = x_{t,(u,v)} \tag{3}$$

Without additional equations the system has a solution in which all words consist of a's only. We can add additional equations specifying colors of some intersection points of γ with T. If the constraints are consistent, the components of γ in the resulting coloring will be monochromatic, otherwise there will not be a solution. (This will be useful when counting connected components later.)

Now we can prove part (a) of Theorem 1. Find the lexicographically smallest solution of the curve coloring equation with r colored by color b over alphabet $\Sigma = \{a, b\}$. Assigning to each edge $(u, v) \in t \in T$ the number of b's in $x_{t,(u,v)}$ yields a normal coordinate representation of γ_r.

Let γ be a properly embedded arc. There is a triangulation T' such that γ is an edge of T and T' is in normal position w.r.t. T. Set up curve coloring equations for the edges of T in the triangulation T' and color each edge of T with different color. In the solution the word x_γ written on γ is the edge intersection representation of γ in T. By Lemma 3 x_γ can be compressed to size polynomial in $\log |\gamma|$ and size of the triangulations T' and T. Hence the edge intersection representation is compressible.

4.2 Region Coloring Equations

We can modify the curve coloring equations (1), (2), (3) to color the connected components of $M - \gamma$. The variable $x_{t,(u,v)}$ will now encode the colors of the ports of γ on (u,v), hence we specify $|x_{t,(u,v)}| = \gamma(u,v)+1$. The variable $y_{t,u}$ will encode colors of regions extending from (w,u) to (v,u). For each $t \in T$ we add a variable z_t which encodes the color of the center region (the region which has ports on all edges of t). The triangle constraints become

$$\begin{array}{lll} x_{t,(u,v)} = y_{u,t}z_t y_{t,v} & x_{t,(v,u)} = y_{v,t}z_t y_{t,u} & x_{t(v,w)} = y_{v,t}z_t y_{t,w} \\ x_{t,(w,v)} = y_{w,t}z_t y_{t,v} & x_{t(u,w)} = y_{u,t}z_t y_{y,w} & x_{t,(w,u)} = y_{w,t}z_t y_{t,u} \end{array} \qquad (4)$$

The edge constraints remain unchanged. Without additional constraints the system has a solution with all regions colored by the same color a. We can add additional equations specifying colors of some ports. In the resulting coloring (if it exists) each region will be monochromatic.

Let $p \in (u,v)$ be a port of γ in T, where (u,v) belongs to triangles $s,t \in T$. We would like to modify the equations to color the connected components of $M-\gamma-p$. It is enough to modify the edge constraints of (u,v) and (v,u). We replace the equation $x_{s,(u,v)} = x_{t,(u,v)}$ by the equations

$$x_{s,(u,v)} = w_1w_2w_3,\ w_1w_4w_3 = x_{t,(u,v)}$$
$$|w_1| = p,\ |w_2| = |w_4| = 1,\ |w_3| = \gamma(u,v) - p.$$

Similarly we can set up equations for coloring components of $M-\gamma$ with polynomially many ports of γ in T removed.

4.3 Intersection Counting Equations

If we are given an edge e, and a number that specifies an intersection of γ along e, we want to compute the index of that intersection.

Again, this problem can be used solving word equations. In a first step we need to determine for each edge (u,v) belonging to a triangle t how many of the intersections of γ along (u,v) enter t, and how many leave t (we think of traversing γ starting at the starting point we fixed). Call these two numbers $\gamma^i(u,v,t)$ and $\gamma^o(u,v,t)$. We know that $\gamma^i(u,v,t)+\gamma^o(u,v,t) = \gamma(u,v)$ for edges (u,v) belonging to t.

We can determine the numbers $\gamma^i(u,v,t)$ and $\gamma^o(u,v,t)$ by setting up the triangle and edge constraints shown in (2) and (3), and specifying that $|x_{t,(u,v)}| = 4\gamma(u,v)$. We then force the starting point of γ to be equal to the string *cabc*. The solution for each edge will be a concatenation of *cabc*'s and *cbac*'s, depending on whether the edge enters or leaves the triangle. Because of Lemma 1 we can count the occurrence of these substrings in polynomial time, and therefore we can compute $\gamma^i(u,v,t)$ and $\gamma^o(u,v,t)$ in polynomial time.

With this information we set up a new set of equations. Let $m = |\gamma|+1$. For each triangle t we take two copies of the variables in (1), one for intersections coming into t, and one for outgoing intersections. Distinguish the two set of variables by superindexing them by i or o. For every edge (u,v) that belongs to triangles s and t we set up the following equations:

$$\begin{array}{lll} x^o_{s,(u,v)}a = ax^i_{t,(u,v)} & ax^o_{s,(v,u)} = x^i_{t,(v,u)}a & x^i_{t,(u,v)} = y^i_{u,t}y^i_{t,v} \\ x^i_{t,(v,u)} = y^i_{v,t}y^i_{t,u} & x^o_{t,(u,v)} = y^o_{u,t}y^o_{t,v} & x^i_{t,(v,u)} = y^o_{v,t}y^o_{t,u} \end{array} \qquad (5)$$

We also specify that $|x^i_{t,(u,v)}| = |x^i_{t,(u,v)}| = m \cdot \gamma^i(u,v,t)$, and $|x^o_{t,(u,v)}| = |x^o_{t,(u,v)}| = m \cdot \gamma^o(u,v,t)$, and set the startpoint of γ to be ba^{m-1}. Equations (5) ensure that at every intersection point the b is moved by one position.

Suppose now we are given a coordinate k along an edge (u,v) of the triangulation. Since we can determine the direction of the intersection, we know the triangle t that γ enters along that point. Using a set of equations like the one we used earlier to count the number of intersections in each direction, we can determine the coordinate k' of the intersection crossing into t. We can then retrieve the string $x^i_{t,(u,v)}[k' * m \dots (k' * m + m - 1)]$, and determine the position of the single b in that string which immediately gives us the index we sought. This proves part (g) of Theorem 1.

5 The Algorithms

5.1 Counting Connected Components

Let T be a triangulation of M, where τ is the number of triangles in T. Let γ be a curve in M given by normal coordinates w.r.t. T. The number of components k of γ can be exponential in the input size. However it is known that the components fall into few normal isotopy classes [Kne30].

Lemma 3. *The components of γ fall into at most 6τ normal isotopy classes.*

We omit the proof. We can now prove parts (b) and (c) of Theorem 1. The algorithm for computing the number of components of γ works as follows. Pick an edge (u,v) in T which is intersected by γ. Let r be the intersection of γ and (u,v) closest to u. Compute the normal coordinates of the connected component γ_r of γ which contains r. Using binary search find the intersection point ℓ of γ with (u,v) whose connected component γ_ℓ is normally isotopic to γ_r. The components of points between r and ℓ are exactly the components of γ which are normally isotopic to γ_r. Hence we can increase the component count by $(\ell - r + 1)$ and run the algorithm for the curve $\gamma - (\ell - r + 1)\gamma_r$. Lemma 3 implies that the number of repetitions of the algorithm is bounded by 6τ.

The described algorithm also finds non-normally-isotopic $\gamma_1, \dots, \gamma_n$ which occur in γ and their counts. Using part (d) of Theorem 1 we can merge the counts for isotopic γ_i.

5.2 Deciding Isotopy I

Let R be a union of connected components of $M - \gamma$ given as a solution of the region coloring equations, where R is colored with color b and $M - R$ is colored with color a. We want to compute the Euler characteristic of R. We will cut R by γ and T into regions and use the formula $\chi = V - E + F$. To compute V, E, F we just need to compute these quantities for each triangle $t \in T$ with appropriate weights and then sum them together. The edges in $\partial t - \partial M$ get weight $1/2$ and the vertices which are also vertices of T get weight one over the number of triangles they occur in.We will only need to compute the numbers of occurrences of b, ab, bb, ba in each $x_{t,(u,v)}$ which can be done using Lemma 1.

Now we can prove part (d) of Theorem 1. Because of part (c) it is enough to consider the case when γ is connected. It is known (see [FM97]) that two *disjoint* properly embedded arcs are isotopic iff they bound a disc in M. Let $\alpha = \gamma + \delta$, the curve whose normal coordinates is the sum of the coordinates of γ and δ. Since γ and δ are normal isotopy disjoint the curve α contains both γ, δ as components. We color the port between starting points of γ and δ with b and check that the resulting region has Euler characteristic 1. The case where γ and δ are properly embedded circles is similar, we only have to check whether they co-bound an annulus.

5.3 Connecting Two Points

In this section we prove part (e) of Theorem 1. First we want to decide whether ports p and q are in the same connected component of $M - \gamma$. Add an equation coloring p with b to the region coloring equations. The ports p, q are in the same connected component of $M - \gamma$ iff port q has color b in the lexicographically smallest solution of the equations.

Suppose that p and q are in the same connected component. We want to find a curve connecting them. Let $t \in T$ and let $r_{t,1}, r_{t,2}, r_{t,3}$ be the ports of the center region of t. A shortest path connecting p and q does not enter the center region of t twice and hence one of the ports $r_{t,1}, r_{t,2}, r_{t,3}$ can be removed while keeping p, q in the same component. We find the port which can be removed, remove it, and move on to another triangle which does not have a port removed. After removing a port of the center region of each triangle the components of $M - \gamma - \{r_{t,?}; t \in T\}$ are either discs or annuli. Finally we remove both p, q and color just one side of p with color b. For one of the sides of p the coloring will reach q. Let $x_?$ be the solution of this system of equations. The colored component which reached from p to q is is a sequence of rectangular regions. The curve connecting p, q which runs in the middle of the colored component intersects edge $(u, v) \in T$, $\#_b x_{t,(u,v)}$ times.

5.4 Deciding Isotopy II

Let T be a triangulation of M and let γ be a curve in M given by normal coordinates w.r.t. T. We will show how to compute normal coordinates of $\gamma' \sim \gamma$ w.r.t. a triangulation T' which is a modification of T. The modifications we will consider are called bistellar moves (flip, add, drop) shown on the Figure 2. The moves can be applied only when all the triangles shown in Figure 2 are distinct.

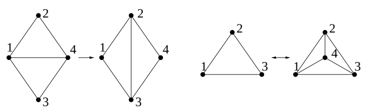

Fig. 2. The bistellar moves.

The new coordinates after the add move can be chosen to be $\gamma(14) = 0$, $\gamma(24) = \gamma(12)$, $\gamma(34) = \gamma(13)$. For the flip move $\gamma(23) = (\gamma(12) + \gamma(24) + \gamma(13) + \gamma(34))/2 + |\gamma(12) - \gamma(24) + \gamma(34) - \gamma(13)|/2 - \gamma(14)$.

For the drop move we can w.l.o.g. assume that the curve γ inside the triangle 123 looks as the one at the Figure 3. The e curves around the point 4 can be removed because they are homotopic to a point. We also need to pull the f curves since they are not in normal position w.r.t. T'.

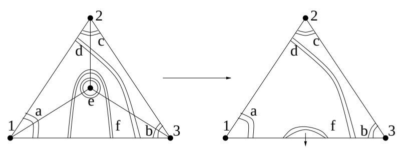

Fig. 3. The drop move.

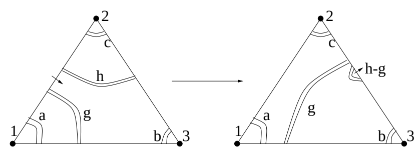

Fig. 4. Pulling the curve to a normal position (w.l.o.g. $h \geq g$).

The pulling might propagate to neighboring triangle as shown in Figure 4. We cannot follow the propagation directly because it can happen exponentially many times.

When both g and h from Figure 4 are non-zero we say that the pulled curves split. For each triangle $t \in T$ at most once split occurs in t (otherwise the curve γ would have to self-intersect).

The algorithm for pulling the curves will work as follows. Using word equations we will find the segment of the pulled curves until the first split. We pull the curves, perform the split and repeat. The number of splits and hence repetitions is bounded by the number of triangles in T.

To find the first split we extract the outermost of the f pulled curves, call it δ. Then we cut δ in t. If δ is still connected we cut it once more at its midpoint (using part (b) of Theorem 2). We obtain two embedded arcs α, β starting in t. The first split occurs when α, β choose different route inside a triangle t. The split can be easily found using binary search.

Lemma 4. *Let M be a 2-manifold with non-empty boundary. Let T be a triangulation of M. Let γ be a curve in M given by normal coordinates w.r.t. T. We can find a minimal triangulation T' of M and normal coordinates of $\gamma' \sim \gamma$ w.r.t. T'*

Proof. Using bistellar moves and by possibly adding new triangles we can remove vertices of T which are not on the boundary ∂M of M. Let $v \in T$ be a vertex in the interior of M. If deg $v = 3$ then there are three distinct triangles neighboring v and we can apply the drop move to eliminate v. If deg $v > 3$ then we can apply a sequence of flip moves to decrease the degree of v to 3 and then eliminate it with a drop. If deg $v = 2$, then we have two triangles t_1, t_2 glued together along two adjacent edges. If either t_1 or t_2 is glued to another triangle, we can apply a flip to increase the degree to 3 and then flip. If not, then M is a disk consisting of two triangles, and we will need to add another triangle to the boundary, and proceed as in the following case. The last case is when deg $v = 1$. This implies that a single triangle is glued to itself. If it is also attached to another triangle, then we can apply a flip to increase the degree of v to 2. If it is not, then M is a disk consisting of a single triangle and we must attach an additional triangle, and then apply the flip.

Two curves given by normal coordinates w.r.t. a minimal triangulation are isotopic iff the coordinates are equal (and the points on ∂M agree). Hence as a corollary of Lemma 4 we obtained a proof of part (a) of Theorem 2.

5.5 Finding the n-th Intersection Point

There is a simple randomized (Las Vegas) algorithm to find the n-th intersection point of an oriented embedded arc γ with the triangulation T. The intersection points are numbered by $0, \dots, |\gamma|$.

Pick a random intersection point r of γ with T. Using the curve coloring equations color one side with b's and the other side with a's. We obtain two embedded curves γ_1, γ_2 which when glued together at r yield γ. If $|\gamma_1| < n$ then find the $n - |\gamma_1|$th intersection point of γ_2 with T, otherwise find nth intersection point of γ_1 with T. The expected number of repetitions is $O(\log |\gamma|)$.

5.6 Computing the Algebraic Intersection Number

The *algebraic intersection number* of two oriented curves γ, δ in an orientable surface is defined as follows. For each intersection of γ and δ in which δ crosses from left to right add $+1$, if it crosses from right to left add -1. The algebraic intersection number is invariant under isotopy.

Since the algebraic intersection number of γ and δ is invariant under isotopy, we can fix a drawing of the curves. For each edge (u, v) of the triangulation we choose which of the curves will intersect (u, v) in the half closer to u, the other one will intersect (u, v) in the half closer to v. Now we draw the curves so that the segments of γ, δ in each triangle are geodesics.

For a triangle $t \in T$ with vertices u, v, w we need to compute the number of segments of γ oriented from (u, v) to (u, w). Take the curve coloring equations for the curve 4γ. Using the same idea we saw earlier, color the copies *cabc* (in that order) at one endpoint. The occurrences of *cabc*, and *cbac* show the orientation of γ along an edge. Counting the number of occurrences of *cabc*, and *cbac* then gives us the result.

References

[AHT02] I. Agol, J. Hass, and W. Thurston. 3-manifold knot genus is NP-complete. In *Proceedings of the 33th Annual ACM Symposium on Theory of Computing (STOC-2002)*, 2002.

[BE+99] M. Bern, D. Eppstein, et al. Emerging challenges in computational topology. ACM Computing Research Repository, September 1999.

[DEG99] T. Dey, H. Edelsbrunner, and S. Guha. Computational topology. In B. Chazelle, J.E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223 of Contemporary Mathematics. American Mathematical Society, 1999.

[DG99] T. Dey and S. Guha. Transforming curves on surfaces. *JCSS: Journal of Computer and System Sciences*, 58(2):297–325, 1999.

[DM01] V. Diekert and A. Muscholl. Solvability of equations in free partially commutative groups is decidable. In *ICALP 2001*, pages 543–554, 2001.

[FM97] A. Fomenko and S. Matveev. *Algorithmic and computer methods for three-manifolds*. Kluwer, 1997.

[GKPR96] L. Gąsieniec, M. Karpinski, W. Plandowski, and W. Rytter. Efficient algorithms for Lempel-Ziv encoding. *in Proceedings of SWAT'96, LNCS 1097*, pages 392–403, 1996.

[Hak61] W. Haken. Theorie der Normalflächen. *Acta Mathematica*, 105:245–375, 1961.

[HLP99] J. Hass, J. Lagarias, and N. Pippenger. The computational complexity of knot and link problems. *Journal of ACM*, 46(2):185–211, 1999.

[Kne30] H. Kneser. Geschlossene Flächen in dreidimensionalen Mannigfaltigkeiten. *Jahresbericht der Deutschen Mathematikver-Vereinigung*, pages 248–260, 1930.

[PR98] W. Plandowski and W. Rytter. Application of Lempel-Ziv encodings to the solution of words equations. In *Automata, Languages and Programming*, pages 731–742, 1998.

[Ryt99] W. Rytter. Algorithms on compressed strings and arrays. In *Proceedings of 26th Annual Conference on Current Trends in Theory and Practice of Infomatics.*, 1999.

[SSŠ02] M. Schaefer, E. Sedgwick, and D. Štefankovič. Recognizing string graphs in np. In *Proceedings of the 33th Annual ACM Symposium on Theory of Computing (STOC-2002)*, 2002.