9-1-2021

Linking Numbers in Three-Manifolds

Patricia Cahn
Smith College, pcahn@smith.edu

Alexandra Kjuchukova
Max Planck Institute for Mathematics

LINKING NUMBERS IN THREE-MANIFOLDS

PATRICIA CAHN AND ALEXANDRA KJUCHUKOVA

Abstract. We give an explicit algorithm for computing linking numbers between curves in an irregular dihedral p-fold branched cover of S^3. This work extends a combinatorial algorithm by Perko which computes the linking number between the branch curves in the case $p = 3$. Owing to the fact that every closed oriented three-manifold is a dihedral three-fold branched cover of S^3, the algorithm given here can be used to compute linking numbers in any three-manifold, provided that the manifold is presented as a dihedral cover of the sphere. The algorithm has been implemented in Python, and we include the code in an Appendix.

1. Introduction

Non-abelian branched covers are among the oldest and most effective tools for studying knots[1]. One invariant which can be extracted from the non-cyclic branched covers of a knot is the collection, over all covers of a given type, of linking numbers between the branch curves. It is hard to overstate the usefulness of this classical apparatus, in testament of which fact we offer a most abbreviated history. Reidemeister opened the gates by introducing the linking number invariant in [24] and using it to distinguish a pair of knots. Subsequently, Bankwitz and Schumann [2] classified knots of up to nine crossings, primarily by using linking numbers in dihedral covers of two-bridge knots. Perko extended these methods to all knots, which led to the classification to knots of ten and eleven crossings [21]. Burde proved that dihedral linking numbers can tell apart all two-bridge knots [4]. Linking numbers in dihedral covers are also good for analyzing properties of knots, as they provide an obstruction to amphichirality [8], [21] and invertibility [11]. For a more thorough account of the role of linking numbers in knot theory, as well as a couple of illuminating examples, the reader is referred to [23].

Perko obtained the majority of his classification results using linking numbers between the branch curves in dihedral covers; on occasion, other types of non-cyclic covers were needed. His tabulation of linking numbers in three-fold dihedral covers was carried out by designing an algorithm and a computer program for their combinatorial calculation [19]. Since the 1960s, Perko's combinatorial algorithm has been replicated, for example by Thistlethwaite; he gives a self-contained description of the method and illustrates it on a 5-fold cover of

[1]The authors recently learned from [23] that a picture of an irregular dihedral cover of the trefoil appeared in Heegard's thesis!

the knot 7_3 in [26]. To our understanding, the combinatorial approach to linking numbers which this paper extends remains the most efficient known method for computation.

To emphasize, Perko's algorithm serves to compute the linking numbers *between the branch curves*, and this has been the focus of all further work we are aware of, even though theoretical advances call for more general methods for computing linking numbers in branched covers. There are many instances in the literature where linking numbers of non-branch curves play an important role. In [6], Cappell and Shaneson gave a formula, in terms of linking numbers in a cyclic cover of a characteristic knot for α, for the μ invariant of a dihedral cover of α. (Since every oriented three-manifold is a dihedral cover of a knot [13], [18], this method is universal. By the same token, linking numbers in three-fold dihedral covers of S^3 include all linking numbers in all three-manifolds.) The curves whose linking numbers appear in Cappell and Shaneson's formula are lifts to a (cyclic) branch cover of curves which live in the complement of the branching set downstairs, that is, lifts of what we call *pseudo-branch curves* in this article. (Despite the apparent ambiguity, we also use the phrase "pseudo-branch curves" to refer to the lifts themselves.) For a second application, in an unpublished manuscript, Litherland [16] showed that Casson-Gordon gordon invariants of a knot can also be computed using linking numbers of pseudo-branch curves in a branched cover. More recently, Florens et al. have used (twisted) linking numbers in relation with multivariate signatures (a generalization of Tristram-Levine signatures to colored links). As these results show, being able to compute linking numbers of pseudo-branch curves in terms of data in the base bears relevance to many current problems in three-manifolds and knot theory, including questions about slice genus and unlinking numbers of links.

Among the many potential uses of computing linking numbers of pseudo-branch curves, the primary application motivating our work is the following one. In [14], the second author gives a formula for the signature of a p-fold irregular dihedral branched cover $f : Y \to X$ between closed oriented topological four-manifolds X and Y, in the case where the branching set B of f is a closed oriented surface embedded in the base X with a cone singularity described by a knot $\alpha \subset S^3$. This formula shows that the signature of Y deviates from the locally flat case by a defect term, $\Xi_p(\alpha)$, which is determined by the singularity α and can be calculated in part via linking numbers of pseudo-branch curves in a dihedral cover of α. If the base X of the covering map f is in fact S^4, the signature of the cover Y is exactly equal to $\Xi_p(\alpha)$. In particular, an effective method for computing linking numbers between pseudo-branch curves in dihedral covers would help establish the range of signatures of dihedral branched covers of S^4 with singular branching sets. More generally, the same computations play a key role in understanding the classification of branched covers with singular branching sets over any four-manifold base.

With these applications in mind, we set out to produce an algorithm for computing linking numbers between pseudo-branch curves in dihedral covers of S^3. The main idea of our approach, described in detail in Section 3, is a certain refinement of Perko's original algorithm. That Perko's method can be extended to other curves besides the branch curves is noted in Appendix B of [14]. The present work is the first detailed treatment, with applications to concrete examples, of these ideas.

Our algorithm yields the following formula for the linking number $I_{j,k}$ between the k^{th} and j^{th} lifts δ^k and γ^j of the pseudo-branch curves δ and γ, respectively, with branching knot α. (Here a lift refers to a connected component of the preimage of δ or γ under the covering map $f : M \to S^3$.) The h_i are the arcs of δ in an oriented link diagram of $\alpha \cup \delta \cup \gamma$, where $0 \le i \le t-1$ and t is the number of self-crossings of δ plus the number of crossings of δ under γ and α. The function $f(i)$ denotes the number (subscript) of the arc of α, γ, or δ which crosses over h_i; the numbering system is described in Section 3. The sign $\epsilon(i)$ is the sign of the crossing at the head of the i^{th} arc of δ in the diagram. The functions ϵ_5, ϵ_6 and ϵ_7 are used to compute the intersection between a given one-cell and a given two-cell in the cover; exact formulas are given in Section 3.5, and can be computed purely combinatorially from the colored knot diagram. The x_i^j are a solution to an inhomogenous system of linear equations, which describe a 2-chain with boundary γ. This system of equations can also be derived combinatorially from the diagram. A solution exists if and only if γ^j is rationally null-homologous in the branched cover, so, in particular, the x_i^j exist if the linking number in question can be computed. [2]

Theorem 1. *Let $f : M \to S^3$ be a three-fold irregular dihedral cover branched along a knot α, and let $\gamma, \delta \subset S^3 - \alpha$. If the lifts γ^j and δ^k are rationally nullhomologous closed loops in M for $j, k \in \{1, 2, 3\}$, then, in the notation of the above paragraph, the linking number $I_{j,k}$ of δ^k with γ^j is*

$$\sum_{i=0}^{t} C_i,$$

where C_i is given by

$$C_i = \begin{cases} \epsilon(i)\epsilon_5^k(i)x_{f(i)}^j & \textit{if } h_i \textit{ meets an arc of } \alpha; \\ \epsilon(i)\epsilon_6^{j,k}(i) & \textit{if } h_i \textit{ meets an arc of } \gamma; \\ 0 & \textit{if } h_i \textit{ meets an arc of } \delta. \end{cases}$$

We focus here on the case where each pseudo-branch curve lifts to three closed loops because this case is the one we encounter exclusively in our main application (see Section 2). Of course, computations involving pseudo-branch curves whose pre-images under the branched covering map consist of fewer than three connected components can be done by the same methods (see Section 3.6). We also prove a similar formula, Theorem 4, for the linking numbers of the lifts of a pseudo-branch curve with a branch curve.

[2] Of course, both δ^k and γ^j must be rationally null-homologous for the linking number $I_{j,k}$ to be well-defined. We verify this condition by reversing the roles of γ and δ and making sure that each of them bounds a two-chain.

In Section 2, we recall the definition of an irregular dihedral cover as well as that of a characteristic knot, a concept essential both to the construction of dihedral covers of knots and to the main application we have in mind for this work. Section 3 presents our generalization of Perko's algorithm to non-branch curves and the proof of Theorem 1. In the same section we also derive an analogous formula for the linking number between a pseudo-branch curve and a branch curve. Section 4 illustrates our algorithm on a concrete example of a three-fold dihedral cover and several pseudo-branch curves therein. Perko's original method for computing linking numbers of branch curves is recalled in detail in Appendix A. Due to the large number of cells used, computations by hand quickly evolve into an unwieldy task, even for the most resolute and concentrated persons. Our algorithm for calculating linking numbers in branched covers has therefore been implemented in Python, and we include the code in Appendix B. The input used to calculate the example in Section 4 is given in the very short Appendix C.

The authors would like to thank Ken Perko for many helpful discussions.

2. Irregular Dihedral Covers of Knots

Let α be a knot in S^3 and $f : M \to S^3$ a covering map branched along α. We can think of f as being determined by its unbranched counterpart, $f_{|f^{-1}(S^3-\alpha)}$, and therefore by a group homomorphism $\rho : \pi_1(S^3 - \alpha, x_0) \to G$ for some group G. For us, G is always D_p, the dihedral group of order $2p$, ρ is surjective and p is odd. We can now associate to ρ the regular $2p$-fold dihedral cover of (S^3, α); this cover corresponds to the subgroup $\ker \rho \subset \pi_1(S^3 - \alpha, x_0)$. The irregular p-fold dihedral cover of (S^3, α) corresponds to a subgroup $\rho^{-1}(\mathbb{Z}_2) \subset \pi_1(S^3 - \alpha, x_0)$, where $\mathbb{Z}_2$ can be any subgroup of D_p of order 2. The irregular dihedral cover is a $\mathbb{Z}/2\mathbb{Z}$ quotient of the regular one, and different choices of subgroup $\mathbb{Z}_2 \subset D_p$ correspond to different choices of an involution. Recall also that ρ can be encoded by a p-coloring of the knot diagram, where the "color" of each arc indicates the element in D_p of order 2 to which ρ maps the element of the knot group corresponding to the meridian of this arc (see Section 3.2).

Cappell and Shaneson proved in [7] that the regular and irregular covers of (S^3, α) can be constructed from a p-fold cyclic branched cover of S^3 along an associated knot $\beta \subset S^3 - \alpha$, which they called a *mod p characteristic knot* for α. They also showed that mod p characteristic knots for α, up to equivalence, are in one-to-one correspondence with p-fold irregular dihedral covers of α. For a precise definition, let V be a Seifert surface for α and L_V the corresponding linking form. A knot $\beta \subset V^\circ$ is a *mod p characteristic knot* for α if $[\beta]$ is primitive in $H_1(V; \mathbb{Z})$ and $(L_V + L_V^T)\beta \equiv 0 \mod p$.

It is shown in [14] that the characteristic knots of α play an essential role in the primary application of this work, namely computing the defect to the signature of a branched cover $g : Y^4 \to X^4$ between four-manifolds X and Y if the (closed oriented) branching set of g has a singularity of type α. This application will be explored in further work. For the purposes of this paper, the key property of a characteristic knot $\beta \subset V$ is that every simple closed curve in $V - \beta$ has three disjoint closed lifts in the dihedral cover of α corresponding to β. As a result, we focus here on computations with curves in $S^3 - \alpha$ whose lifts to a three-fold dihedral cover of (S^3, α) have three connected components.

3. The Algorithm

We begin with an overview. Let $\alpha \subset S^3$ be a three-colored knot and $f : M \to S^3$ a dihedral cover of S^3 branched along α. Next, let $\delta, \gamma \subset (S^3 - \alpha)$ be two oriented curves, and denote their lifts to M by γ^j and δ^k for $j, k \in \{1, 2, 3\}$. (We continue to assume that the preimages of δ and γ each have three connected components. Whether this is in fact the case depends on the colors of the arcs of α which these curves pass under, and on the order in which they do so.) To compute the linking numbers $I_{j,k} = lk(\gamma^j, \delta^k)$, we endow M with a cell structure in which the curves γ^j and δ^k are one-subcomplexes. This is achieved by lifting to M an appropriate cell-structure on S^3 in which δ and γ are subcomplexes.

The approach here is a generalization of the one used by Perko in [19]. In a nutshell, Perko computes linking numbers between branch curves by: (1) choosing an appropriate cell structure on S^3; (2) lifting this structure to M; (3) solving a linear system to find explicit two-chains the branch curves bound; (4) computing intersection numbers by adding up the intersection numbers of the one-cells that make up one curve and the two-cells that cobound the other. We carry out an extension to this procedure proposed in [14]. The idea is to subdivide the above cell structure, in a way compatible with its lift to M, so as to incorporate pseudo-branch curves δ and γ into the picture. Precisely, we treat the homomorphism $\rho : \pi_1(S^3 - \alpha) \twoheadrightarrow D_p$ from which the branched cover f arises as a homomorphism of $\pi_1(S^3 - \alpha - \delta - \gamma)$ in which meridians of δ and γ all map to the trivial element.

3.1. **Cell structure on S^3.**

The link diagram $\alpha \cup \gamma$ determines a cell structure on S^3, whose 2-skeleton is the cone on the link and whose single 3-cell is the complement of that cone. More precisely, we have one 0-cell at the cone point, and one 0-cell per crossing in the link diagram; following the terminology of Perko, we have one "horizontal" 1-cell per arc in the link diagram, and one "vertical" 1-cell connecting the cone point to each of the remaining 0-cells; the complement of the 1-skeleton in the cone is a union of 2-cells, one per each arc in the link diagram, attached in the obvious way.

Figure 1 shows the part of the cell structure determined by the cone on α.

The idea is then to use this cell structure to determine whether there is a 2-chain bounding each lift of γ in M, and if so, to write down this two-chain explicitly in terms of lifts of the 2-cells downstairs. Note that what we have described here is a subdivision of the cell structure used by Perko, in which the 2-skeleton consists only of the cone on α. For more details about Perko's convenient and useful naming of the cells, we once again refer the reader to Appendix A.

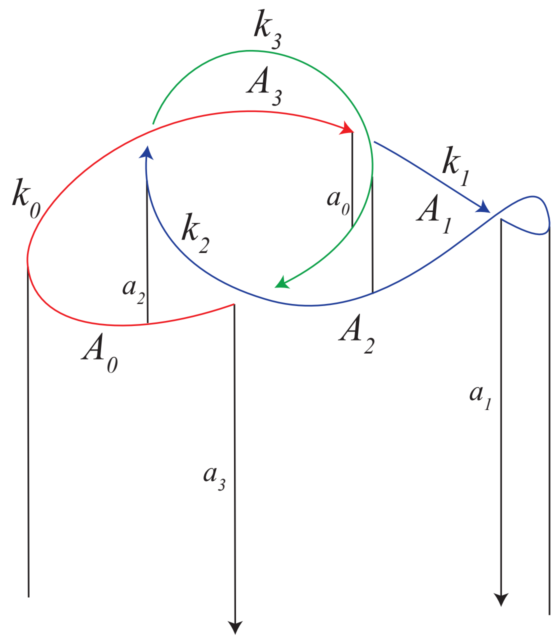

FIGURE 1. The cell structure on S^3 determined by the cone on α.

3.2. **Visualizing the cover near crossings of α.** Fix a 3-coloring of α using the "colors" 1, 2 and 3. The coloring describes a homomorphism $\rho : \pi_1(S^3 - \alpha) \to D_3$ in the usual way: we identify D_3 with S_3, and assign the color 1 to a given arc if and only if ρ sends the corresponding meridian, with either orientation, to (23); similarly for colors 2 and 3. Note that the 3-cell of S^3 has three lifts in M. We number these 3-cells e^3_1, e^3_2 and e^3_3 in such a way that the meridians of the knot act on the subscripts in the way just described.

Consider an inhomogeneous knot crossing – that is, a crossing where three arcs of different colors meet – together with the vertical 2-cells below the knot, as pictured in the upper left corner of Figure 2. Let λ denote the small loop which runs underneath the crossing. The blue loop λ, which we view as based at x_0, has three lifts, all of which are loops. We can subdivide λ into four arcs, with endpoints on the vertical walls, such that the lift of the interior of each arc is contained in a single 3-cell. In Figure 2, the subscripts i of the four corresponding 3-cells e^3_i, for each of the three lifts of λ, starting at x_0 and following the orientation of λ, are: $1, 2, 2, 1$; $2, 1, 3, 3$; and $3, 3, 1, 2$. One can see this directly from the colored crossing downstairs, using the rule that a curve in the j^{th} 3-cell which passes through a lift of the wall under an arc colored $i \in \{1, 2, 3\}$ either stays in 3-cell j if $j = i$, or else enters 3-cell k where $\{k\} = \{1, 2, 3\} - \{i, j\}$. In contrast, the red meridian has two path lifts which are not closed loops and one which is. The lifts fit together to form two closed curves in the cover.

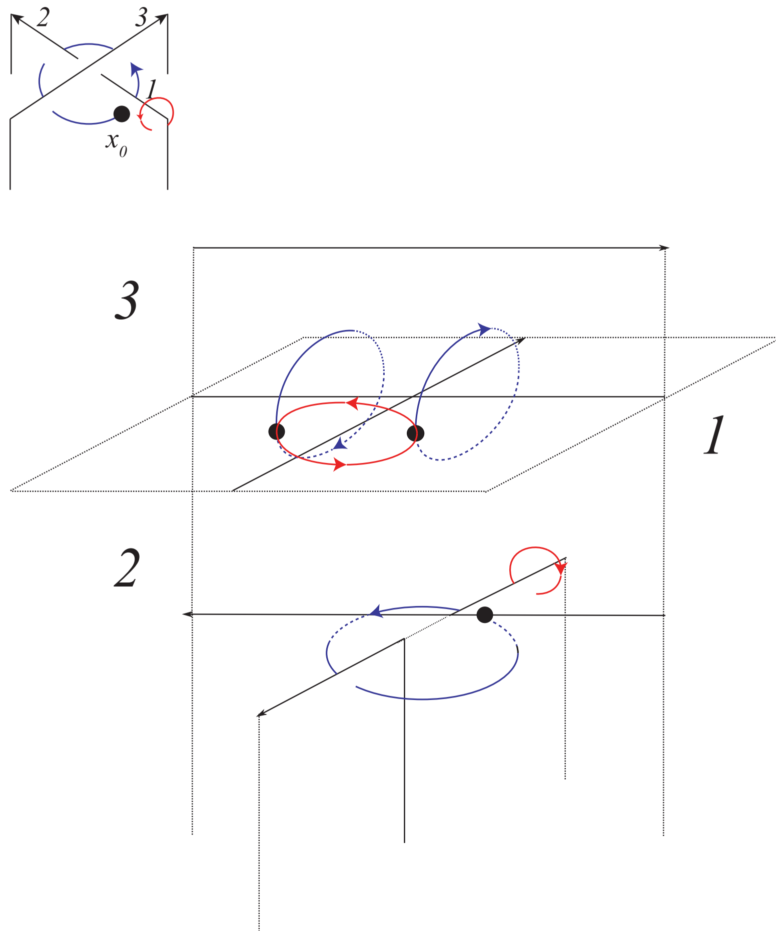

FIGURE 2. The lift of cells below an inhomogeneous crossing, with $p = 3$. In red, visualizing the lifts of loops in $S^3 - \alpha$.

Remark 2. *This approach to visualizing the lifts of crossings generalizes to all odd values of p. For an illustration of the case $p = 5$, see Figure 3.*

3.3. **Cell structure on the branched covering space.** This section serves primarily to establish the notation we use throughout. The arcs of α in the link diagram of $\alpha \cup \gamma$ are labelled $k_0, k_1, \ldots, k_{m-1}$, where m is the sum of the number of crossings of α with itself and the number of crossings of α with γ where α passes under γ. The arcs of γ are labelled $g_0, g_1, \ldots, g_{s-1}$, where s is the number of crossings of γ with itself plus the number of crossings of α with γ where γ passes under α.

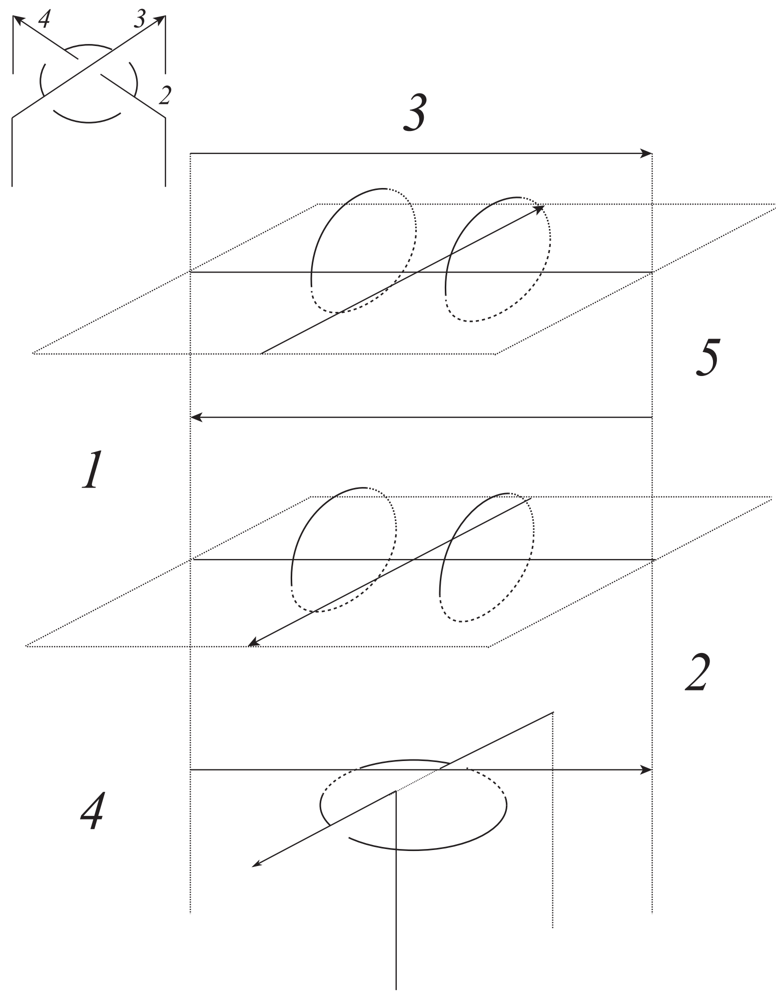

Figure 3. The lift of cells below an inhomogeneous crossing, with $p = 5$.

Denote by $c(i)$ the color of the arc k_i, let $f(i)$ denote the subscript j of the arc (k_j or g_j) passing over crossing i of α, and let $f_\gamma(i)$ denote the subscript j of the arc (k_j or g_j) passing over crossing i of γ. We will sometimes write $f(i)$ rather than $f_\gamma(i)$ to simplify notation, when it is clear we are referring to arcs of γ rather than α.

Let k^1 denote the index-1 branch curve and let k^2 denote the the index-2 branch curve. Each arc k_i has two pre-images under the covering map. Let $k_{1,i}$ denote the index-1 lift of k_i and let $k_{2,i}$ denote the index-2 lift of k_i.

Choose a basepoint x_0 on the arc g_0 of γ. The curve γ has three path lifts under the covering map, γ^1, γ^2, and γ^3, beginning at each of the three preimages of x_0. Assume the γ^i are labelled so that the lift of g_0 which lies in the 3-cell e^3_i is contained in γ^i. The path lifts γ^i can fit together to form either one, two, or three closed loops in the covering space. In general, we want to find surfaces bounding each of these closed loops that is null-homologous. For the rest of this section, however, we assume that each γ^i is a closed loop.

We label the 2-cells below k_i, and their lifts, using the notation of Perko, described in Appendix A. Briefly, A_i is the 2-cell below k_i, and its lifts are $A_{1,i}$, $A_{2,i}$ and $A_{3,i}$. $A_{1,i}$ bounds the index-1 lift of k_i, and the function $w(i)$, also defined in Appendix A, encodes which of the two remaining lifts of A_i is labelled $A_{2,i}$. In order to use Perko's method, we need to assume that the number of self-crossings of α is even; we can do this by performing a Type 1 Reidmeister move.

Let $g_{j,i}$, $j = 1, 2, 3$, denote the lift of g_i which lies in the lift γ^j of γ. Let B_i denote the 2-cell below arc g_i of γ downstairs. Denote by $B_{j,i}$ the lift of B_i which is contained in the lift γ^j of γ; its boundary contains $g_{j,i}$. There are many possible configurations of 2-cells above a self-crossing of γ; see Figure 4 for one example. Let $l^g_j(i)$ denote the number of the 3-cell which contains the lift $g_{j,i}$ of the arc g_i. For example, in Figure 4, $l^g_1(i) = 2$, $l^g_2(i) = 3$, and $l^g_3(i) = 1$.

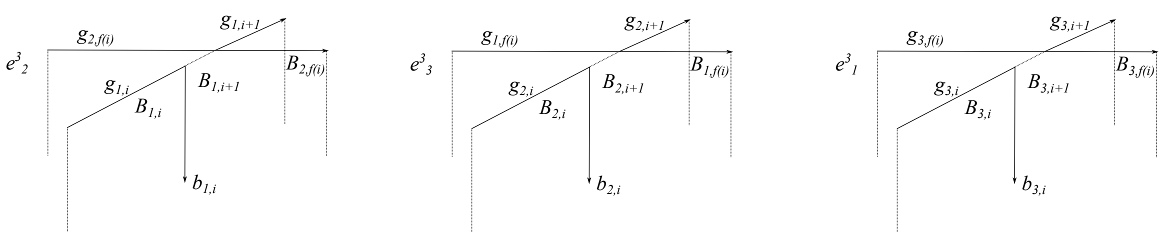

FIGURE 4. One possible configuration of cells lying above a crossing of γ with itself.

When the knot α passes under the pseudo-branch curve γ, there is branching above the crossing. As in the case of crossings of α with itself (described in Appendix A), there are many possible configurations of cells above that crossing, which depend on the value of $w(i)$. One such configuration is pictured in Figure 5.

3.4. **Finding a 2-chain bounded by a pseudo-branch curve.** Now we look for a 2-chain $C^{2,j}$ with $\partial C^{2,j} = \gamma^j$ for fixed j. Since

$$\gamma^j = \sum_{i=0}^{s-1} g_{j,i},$$

each 1-cell $g_{j,i}$ must appear exactly once in the boundary of $C^{2,j}$. Only the cell $B_{j,i}$ can contribute $g_{j,i}$ to $\partial C^{2,j}$, so $C^{2,j}$ must have $\sum_{i=0}^{s-1} B_{j,i}$ as a summand. Note, however, that the sum of the $B_{j,i}$ typically has other boundary components in addition to γ^j, due to

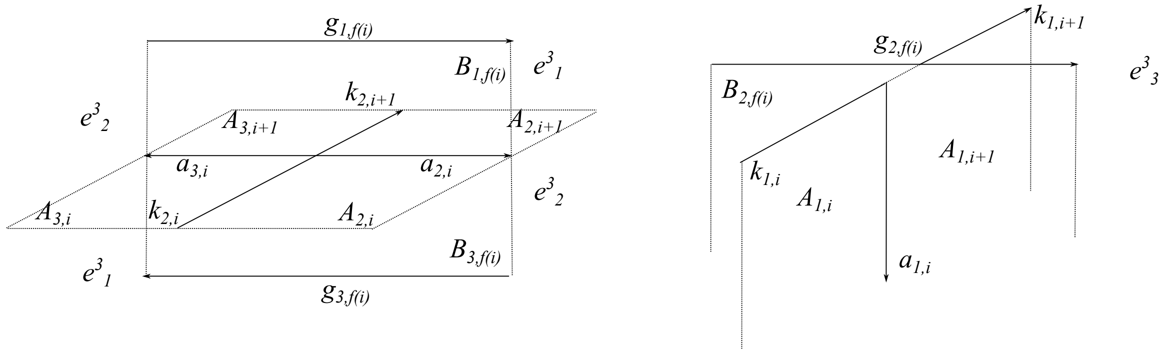

Figure 5. One possible configuration of cells above a crossing where γ passes over α, and the arc k_i is colored 3.

crossings of the knot under arcs of γ, which contribute additional "vertical" 1-cells (lifts of 1-cells which appear vertical in the cone on $\alpha \cup \gamma$) to the boundaries of 2-cells. To cancel these additional boundary components, we write:

$$C^{2,j} = \sum_{i=0}^{s-1} B_{j,i} + \sum_{i=0}^{m-1} x_i^j (A_{2,i} - A_{3,i}). \tag{1}$$

In the above equation, we have used the fact that the coefficients of $A_{2,i}$ and $A_{3,i}$ in $C^{2,j}$ must be negatives of each other because this is the only way to cancel out the "horizontal" cells $k_{2,i}$. (Similarly, we know that cells $A_{1,i}$ can not possibly appear in $C^{2,j}$ because they would introduce extraneous horizontal 1-cells, $k_{1,i}$, to the boundary.)

It remains to find the coefficients x_i^j. To that end, we write down a system of linear equations in the x_i^j, one for each crossing. We obtain three systems of equations, one for each $C^{2,j}$ with $j \in \{1, 2, 3\}$, as follows.

No arc of k^1 or k^2 appears in the boundary of $C^{2,j}$. By examining the configuration of cells above an inhomogeneous self-crossing of α, as in Appendix A, we see that the x_i^j satisfy the equation

$$x_i^j - x_{i+1}^j + \epsilon_1(i)\epsilon_2(i)x_{f(i)}^j = 0, \tag{2}$$

where ϵ_1, ϵ_2 and ϵ_3 (which appears below) are defined as in Appendix A.

By the same reasoning, for homogeneous crossings of α the corresponding equation is identical to Perko's equation for the surface bounding the index 1 curve (also in Appendix A):

$$x_i^j - x_{i+1}^j + 2\epsilon_3(i)x_{f(i)}^j = 0. \tag{3}$$

Now we consider crossings of α under γ, as in Figure 5. To capture the combinatorics at play, we associate a function to such crossings as follows:

$$\epsilon_4^j(i) = \begin{cases} -1 \text{ if } l_j^g(i) = w(i); \\ 0 \text{ if } l_j^g(i) = c(i); \\ 1 \text{ otherwise.} \end{cases}$$

For example, in Figure 5, $\epsilon_4^1(i) = 1$, $\epsilon_4^2(i) = 0$, and $\epsilon_4^3(i) = -1$.

The equation associated to a crossing of the knot under a pseudo-branch curve can now be written as:

$$x_i^j - x_{i+1}^j = \epsilon(i)\epsilon_4^j(i). \tag{4}$$

Unlike the previous two, this equation does depend on j; the right hand side will be 1 for one lift, -1 for another, and 0 for the third.

The boundary of $C^{2,j}$ is then, by construction, $\sum_{i=0}^{s-1} g_{j,i} = \gamma^j$. We have thus proved:

Proposition 3. *Let s denote the number of crossings of γ under α plus the number of self-crossings of γ, let m denote the number of crossings of α under γ plus the number n of self-crossings of α. Let $f(i)$ denote the index of the overstrand $k_{f(i)}$ at crossing i, and let the signs ϵ, and ϵ_x for $x = 1, 2, 3, 4$ be as defined above. If the following inhomogeneous system of linear equations*

$$\begin{cases} x_i^j - x_{i+1}^j + \epsilon_1(i)\epsilon_2(i)x_{f(i)}^j = 0 & \text{if crossing } i \text{ of } \alpha \text{ is inhomogeneous} \\ x_i^j - x_{i+1}^j + 2\epsilon_3(i)x_{f(i)}^j = 0 & \text{if crossing } i \text{ of } \alpha \text{ is homogeneous} \\ x_i^j - x_{i+1}^j = \epsilon(i)\epsilon_4^j(i) & \text{if strand } i \text{ of } \alpha \text{ passes under } \gamma \end{cases}$$

has a solution $(x_0^j, x_1^j, \ldots, x_{m-1}^j)$ over $\mathbb{Q}$ then the lift γ^j of γ is rationally nullhomologous and is bounded by the 2-chain

$$C^{2,j} = \sum_{i=0}^{s-1} B_{j,i} + \sum_{i=0}^{m-1} x_i^j(A_{2,i} - A_{3,i}).$$

3.5. **Computing linking numbers and proof of Theorem 1.** To complete the computation, we introduce a second pseudo-branch curve δ into the diagram containing α and γ *without changing the numbering* on the arcs k_i of α or the arcs g_i of γ. We label the arcs of δ $h_0, \ldots, h_t$. (Self-crossings of δ do not contribute anything to the linking number. When numbering arcs of δ for the computer program, we will assign consecutive arcs of δ the same number if they are separated by an overcrossing by another arc of δ, in order to slightly simplify the input.)

We again use $f(i)$ to denote the number of the crossing of the overstrand at the end of arc h_i. Like γ, the curve δ may lift to one, two or three closed loops. We begin with the case where the lifts of δ and γ form three closed loops. Let δ^1, δ^2, and δ^3 denote the three lifts of δ; as before, we choose the superscripts on the δ^i so that the lift of h_0 which is contained in the 3-cell e^3_i is part of δ^i. Let $l^h_k(i)$ denote the number of the 3-cell which contains the lift arc h_i in δ^k.

Having established the notation, it remains to explain how to compute the following intersection numbers:

- The intersection number of γ^j with a surface bounding δ^k, for $j, k \in \{1, 2, 3\}$
- The intersection number of γ^j with the a surface bounding the degree 1 lift of α for $j \in \{1, 2, 3\}$
- The intersection number of γ^j with the a surface bounding the degree 2 lift of α for $j \in \{1, 2, 3\}$

We begin by computing the intersection number $I_{j,k}$ of γ^j with δ^k, which amounts to proving our main Theorem.

Proof of Theorem 1. Assume that we have found a solution $(x^j_0, \ldots, x^j_{m-1})$ to the set of equations discussed in the previous section. Then the 2-chain bounding γ^j is

$$\sum_{i=0}^{s-1} B_{j,i} + \sum_{i=0}^{m-1} x^j_i (A_{2,i} - A_{3,i}).$$

Crossings of δ under both α and γ may contribute to the linking number. Self-crossings of δ do not contribute to the linking number, which is why our numbering system ignores these crossings. One possible configuration of cells above a crossing of δ under γ is show in Figure 6. The lift $h_{j,i}$ will intersect one of the cells $A_{1,f(i)}$, $A_{2,f(i)}$ or $A_{3,f(i)}$. If it intersects $A_{1,f(i)}$, this crossing does not contribute to $I_{j,k}$ because $A_{1,f(i)}$ is never contained in the 2-chain bounding γ^j. If it intersects $A_{2,f(i)}$, the crossing contributes $\epsilon(i)x^j_{f(i)}$ to $I_{j,k}$. If it intersects $A_{3,f(i)}$, the crossing contributes $-\epsilon(i)x^j_{f(i)}$ to $I_{j,k}$. Define ϵ^k_5 as follows:

$$\epsilon_5^k(i) = \begin{cases} 1 \text{ if } l_k^h(i) = w(i), \\ 0 \text{ if } l_k^h(i) = c(i), \text{ and} \\ -1 \text{ otherwise.} \end{cases}$$

The contribution to $I_{j,k}$ of a crossing of δ under α is then $\epsilon(i)\epsilon_5^k(i)x^j_{f(i)}$.

Now consider crossings of δ under γ. The picture in the cover is similar to that of Figure 4, except that the under-crossing arcs are $h_{\cdot,i}$'s rather than $g_{\cdot,i}$'s. The cell $B_{j,f(i)}$ appears in the 2-chain bounding γ^j exactly once, so the contribution of such a crossing to $I_{j,k}$ is $\epsilon(i)$ if the lifts of $h_{k,i}$ and $g_{j,f(i)}$ are in the same 3-cell, and 0 otherwise.

Define ϵ_6 as follows:

$$\epsilon_6^{j,k}(i) = \begin{cases} 1 \text{ if } l_k^h(i) = l_j^g(f(i)), \text{ and} \\ 0 \text{ otherwise.} \end{cases}$$

By construction, crossings of δ under γ contribute $\epsilon(i)\epsilon_6^{j,k}(i)$ to $I_{j,k}$. The theorem follows. □

3.6. **A note on pseudo-branch curves which lift to fewer than 3 loops.** As seen in Figure 2, the pre-image of a pseudo-branch curve γ under the covering map may well have fewer than three connected components. Precisely, the lifts of γ could include two closed loops $\gamma^1 \cdot \gamma^2$ and γ^3, or one closed loop $\gamma^1 \cdot \gamma^2 \cdot \gamma^3$, where each γ^j covers γ and $\cdot$ denotes concatenation of paths.

If some concatenation σ of the γ^i's forms a closed, null-homologous loop, we can still find a 2-chain $C^{2,\sigma}$ with boundary σ using the methods given in the previous Section 3.4 . We do this by writing down the three systems of equations for $j = 1, 2, 3$ listed in Proposition 3. The 2-chain $C^{2,\sigma}$ bounding σ is then:

$$C^{2,\sigma} = \sum_{j \in S} \left(\sum_{i=0}^{s-1} B_{j,i} + \sum_{i=0}^{m-1} x_i^j (A_{2,i} - A_{3,i}) \right).$$

Now let's consider the linking number between two such pseudo-branch curves. Suppose the closed loop σ is a concatination of paths γ^i, where $i \in S \subset \{1, 2, 3\}$, and the closed loop τ is a concatenation of paths δ^i, where $i \in T \subset \{1, 2, 3\}$ and each δ^i is a lift of a second pseudo-branch curve $\delta \subset S^3 - \alpha$. It follows from Section 3.6 that, in the notation of the same section, if σ and τ are rationally null-homologous, their linking number is equal to $\sum_{j \in S, k \in T} I_{j,k}$.

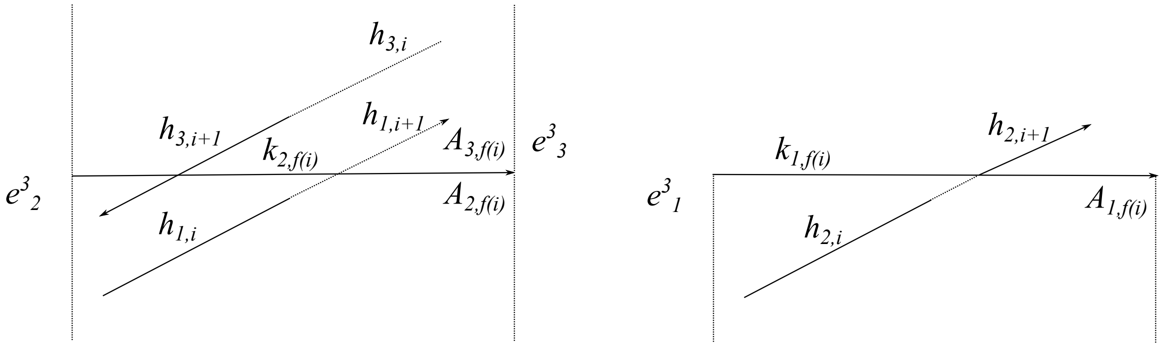

FIGURE 6. One possible configuration of cells above a crossing of δ under α.

3.7. **The linking number of a branch curve with a pseudo-branch curve.** In the same manner, we compute the intersection number of a closed lift of γ with each of the index 1 and 2 branch curves. In this case γ is the only pseudo-branch curve in the picture. Its arcs are again labelled $h_0, h_1, \dots h_t$, where arcs separated by a self-crossing of γ are given the same label h_i and viewed as one arc, and adjacent arcs of γ separated by an overcrossing of α are labelled h_i and h_{i+1}.

We define one last sign, ϵ_7^k, as follows:

$$\epsilon_7^k(i) = \begin{cases} 1 \text{ if } l_k^h(i) = w(i), \\ 1 \text{ if } l_k^h(i) = c(i), \text{ and} \\ -1 \text{ otherwise.} \end{cases}$$

Theorem 4. *Suppose that the pseudo-branch curve γ lifts to three null-homologous closed loops γ^k for $k \in \{1,2,3\}$. Let $\{x_i^1\}$ and $\{x_i^2\}$ be the solutions to the two systems of equations in Proposition 5. The linking number I_k^1 of γ^k with the index 1 branch curve k_1 is*

$$\sum_{i=0}^{t} C_i$$

where C_i is given by $\epsilon(i)\epsilon_7^k(i)x_{f(i)}^1$.

The linking number I_k^1 of γ^k with the index 2 branch curve k_2 is

$$\sum_{i=0}^{t} C_i$$

where C_i is given by $\epsilon(i)\epsilon_5^k(i)x_{f(i)}^2$.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
x_i^1	0	-1	-1	-1	-1	0	1	0	0	0	1	0	0	1
x_i^2	0	0	1	1	1	1	0	0	-1	0	-1	0	0	-1
x_i^3	0	1	0	0	0	-1	-1	0	1	0	0	0	0	0

TABLE 1. The coefficients x_i^j of A_{2i} in the 2-chain bounding the j^{th} lift of β.

4. EXAMPLES

To conclude, we illustrate the output of the algorithm for a collection of pseudo-branch curves. We begin with a three-colorable knot, α, which is the connected sum of two copies of the trefoil knot. Since the trefoil is a two-bridge knot, the dihedral three-fold cover of S^3 branched along α is $S^1 \times S^2$.

Recall from Section 2 that, corresponding to a given three-coloring of α and a Seifert surface V for α, we have a characteristic knot, $\beta \subset V^\circ$. As explained in Section 2, characteristic knots play a key role in our applications (see [14]), as do essential curves in $V - \beta$. In this example, we let V be the connected sum of two copies of the familiar Seifert surface for the minimal-crossing diagram of the trefoil in two-bridge position, namely a surface consisting of two disks joined by three twisted bands. The characteristic knot β is then the connected sum of two copies of a characteristic knot for the trefoil; it is shown in blue in Figures 8 and 7.

We apply our algorithm to the following pseudo-branch curves: the characteristic knot β, described above; an essential curve ω_1 in $V - \beta$, which has one null-homologous closed lift and two homologically nontrivial closed lifts; and a pseudo-branch curve ω_2 which is a push-off of a curve in $V - \beta$ intersecting β once transversely, and lifts to a single null-homologous closed curve.

Our computer algorithm detects the number of lifts and whether each is null-homologous, and allows us to compute the linking numbers of all pairs of null-homologous lifts. The results of this computation are discussed below. The input used to generate these results is given in Appendix C.

Part I. First, the role of the first pseudo-branch curve, denoted by γ throughout the previous section, is played by the characteristic knot β.

The algorithm finds a 2-chain bounding each closed lift of β. The 2-chain bounding the j^{th} lift of β can be described by a list of coefficients x_i^j of 2-cells $A_{2,i}$, as defined in Section 3. The coefficients for the three lifts of β are given in Table 1.

The matrix of intersection numbers $I_{j,k}$ of a 2-chain bounding the j^{th} lift of β with the k^{th} lift of ω_1 is

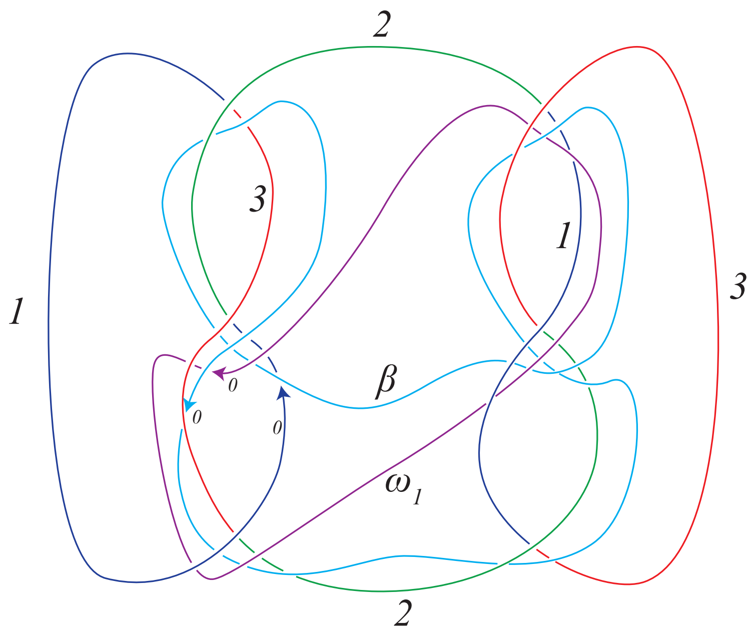

FIGURE 7. The connected sum, α, of two trefoils; a characteristic knot, β, for α; and a curve, ω_1, on a Seifert surface for α which is disjoint from β.

$$(I_{j,k}) = \begin{pmatrix} \mathbf{0} & 0 & 0 \\ -\mathbf{1} & 0 & 1 \\ \mathbf{1} & 0 & -1 \end{pmatrix}.$$

However, we will see in Part II of this example that only the first lift of ω_1 is null-homologous. Thus, the first column of the matrix (in bold) gives the linking numbers of the null-homologous lift of ω_1 with each lift of β. The combinatorial intersection numbers in the second and third columns are not linking numbers.

The matrix of intersection numbers $I_{j,k}$ of a 2-chain bounding the j^{th} lift of β with the k^{th} (path) lift of ω_2 is

$$(I_{j,k}) = \begin{pmatrix} -1 & -1 & 0 \\ -1 & 1 & -2 \\ 0 & -2 & 0 \end{pmatrix}.$$

In this case the 3 path-lifts of ω_2 fit together to form one closed curve – which we call a closed lift – in $S^1 \times S^2$. We will see that the linking numbers of this one closed lift with each of the 3 lifts of β are obtained by summing the rows of the matrix. Hence the linking numbers are each -2.

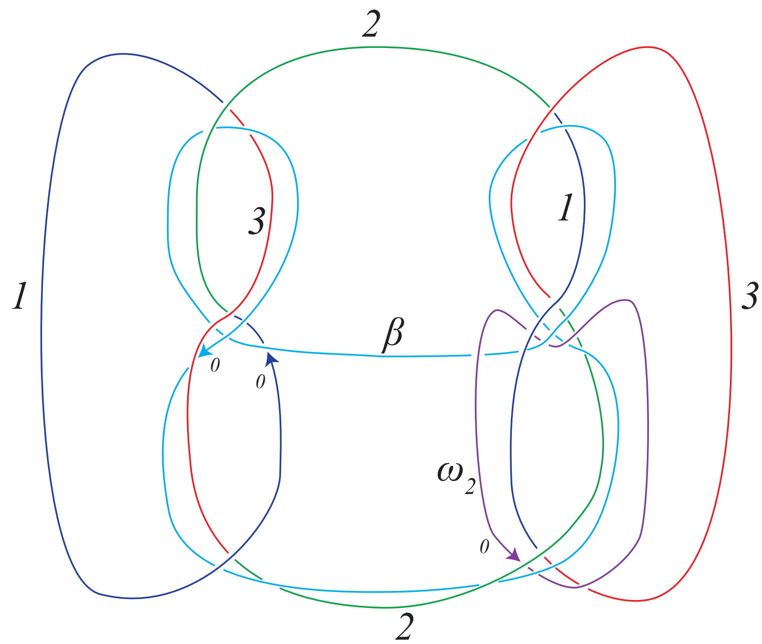

FIGURE 8. The connected sum, α, of two trefoils; a characteristic knot, β, for α; and a push-off, ω_2, of a curve on a Seifert surface for α which intersects β once transversely.

Part II. To complete the example, we let the role of the first pseudo-branch curve be played by ω_1.

The list of coefficients x_i^j of the 2-cells $A_{2,i}$ in the 2-chain bounding lift j of ω_1 is given in Table 2. When $j = 2, 3$ these coefficients are not defined because the corresponding lifts of ω_1 are not null-homologous, and the algorithm detects this, failing to produce a solution for the x_i^j.

The matrix of intersection numbers of the 2-chain bounding the j^{th} lift of ω_1 with the k^{th} lift of β is

$$(I_{j,k}) = \begin{pmatrix} 0 & -1 & 1 \\ x & x & x \\ x & x & x \end{pmatrix}.$$

The program inserts x's into the matrix when the corresponding 2-chain is not defined. The first row of the matrix gives the linking numbers of the null-homologous lift of ω_1 with each lift of β, and we see these numbers agree with the first column of the matrix of

i	0	1	2	3	4	5	6	7	8	9
x_i^1	0	0	0	0	-1	0	1	1	0	0
x_i^2	.	.	.	.	.	.	.	.	.	.
x_i^3	.	.	.	.	.	.	.	.	.	.

Table 2. The coefficients x_i^j of A_{2i} in the 2-chain bounding the j^{th} lift of the curve ω_1 in $V - \beta$. Note that the x_i^2 and x_i^3 are undefined because the corresponding lifts are homologically nontrivial.

intersection numbers of 2-chains bounding lifts of β with lifts of ω_1, confirming our first computation.

5. (Appendix A) Perko's combinatorial method

Our purpose here is to recall Perko's algorithm for computing the linking numbers of the index 1 and 2 branch curves in the 3-fold irregular dihedral branched cover of S^3 along α. The algorithm constructs 2-chains whose boundaries are the degree 1 and 2 branch curves by lifting a cell structure of S^3 to the cover, and finding suitable linear combinations of the cells upstairs. Perko's notation is adhered to, but we introduce a different method of visualizing the lift of the cell structure, which generalizes more easily to the case $p > 3$.

5.1. **The cell structure on S^3 defined by Perko.** We return to the cell structure on S^3 determined by the cone on α, described in Section 3.1, and discuss the labels of the cells. We continue to use Perko's jargon to refer to the cells (calling 1-cells "vertical" or "horizontal" as before and saying a two-cell lies "below" the arc of the knot contained in its boundary) to help guide us through the vast expanse of unavoidable notation.

For the purposes of the algorithm, we require that the diagram of α have an even number of crossings, and we can arrange this to be the case by performing a Type 1 Reidemeister move on α, if necessary. As observed by Perko, this allows for a convenient and consistent labeling of the two-cells in the diagram, reviewed below.

Label the oriented arcs of the projection $k_0, \ldots, k_{n-1}$ where n is the number of crossings of α. These are the "horizontal" 1-cells referred to earlier. The line segment starting at the head of the arc k_i and ending at the cone point p of S^3 is denoted a_i; these are the "vertical" 1-cells. Perko's cell structure is given by: a 0-cell p at the cone point and an additional 0-cell per crossing in the knot diagram; a 1-cells a_i and a one-cell k_i to each arc of the knot diagram; a 2-cells A_i "below" k_i bounded by a_{i-1}, k_i, and a_i (and possibly some other a_j's, in the event that k_i passes over other arcs of the knot); a single 3-cell. The 2-skeleton here is the cone on α; see Figure 9. Lastly, let $f(i)$ denote the number of the

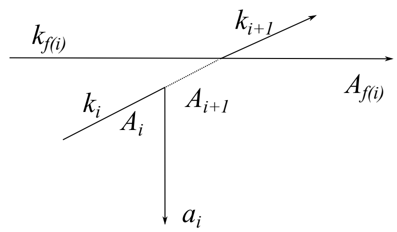

FIGURE 9. Cells at crossing i of α.

arc which passes over k_i, and let $c(i) \in \{1, 2, 3\}$ denote the color of arc i. Let $\epsilon(i) \in \{+, -\}$ denote the sign of crossing i.

Next, lift this cell structure to M. Each k_i has two lifts $k_{1,i}$ and $k_{2,i}$, which are part of the degree one and degree two branch curves, respectively. Each a_i and A_i have three lifts, as does the three cell. Call the three 3-cells e^3_1, e^3_2, and e^3_3 (again, in a way that is consistent with the action of $\pi_1(S^3 - \alpha)$ on the fiber).

Now we describe Perko's method of labelling the lifts of the A_i. For each i, one lift of A_i has boundary meeting the degree 1 branch curve. Call this lift $A_{1,i}$. The other two lifts of A_i share a common boundary segment along the degree 2 curve. These lifts will be called $A_{2,i}$ and $A_{3,i}$. One makes the choice as follows. Let $\vec{A}_i$ be a vector field along k_i tangent to A_i. The vector field $\vec{A} = \cup_{i=0}^n \vec{A}_i$ determines a blackboard framing of k. Now lift $\vec{A}$ to a continuous vector field $\vec{A}_2$ along the degree two lift of k. There are two choices for such a lift. We make a choice arbitrarily along $k_{2,0}$ and this uniquely determines the lift along the entire curve. Call $A_{2,i}$ the lift of A_i to which $\vec{A}_2$ is tangent. Last, we denote by $a_{j,i}$ the lift of a_i which is a subset of the boundary of $A_{j,i}$ for $j = 1, 2, 3$.

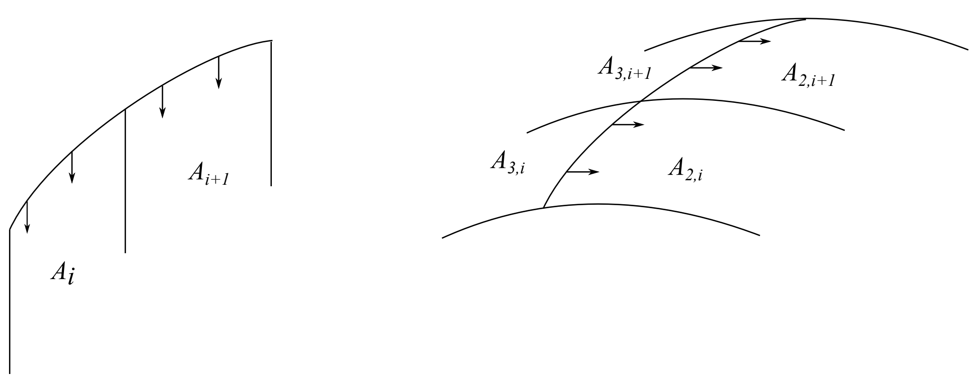

FIGURE 10. A lift of a blackboard framing of α along the degree two curve, which determines the labeling of the lifts of the 2-cells A_i.

Now we discuss the 3-cells e^3_1, e^3_2, and e^3_3 in more detail. The lift of a meridian m_i about k_i has two connected components. One is a meridian of the degree 1 curve $k_{1,i}$ and lies entirely in one 3-cell, namely $e^3_{c(i)}$. The other is a meridian of $k_{2,i}$, is a $2:1$ covering of m_i, and lies in the 3-cells e^3_x and e^3_y, where $\{x, y\} = \{1, 2, 3\} - \{c(i)\}$. It will be convenient to distinguish these 3-cells as follows. Pick a point q on k_{2i}. Let $\vec{v}_{x,q}$ and $\vec{v}_{y,q}$ be vectors in T_qM lying in the half-spaces $T_q(e^3_x)$ and $T_q(e^3_y)$ respectively. One of the two 3-frames $\{\vec{k}_{2,i,q}, \vec{A}_{2,q}, \vec{v}_x\}$ and $\{\vec{k}_{2,i,q}, \vec{A}_{2,q}, \vec{v}_y\}$ is positive, and the other is negative. Let $w(i) \in \{x, y\}$ be the value such that $\{\vec{k}_{2,i,q}, \vec{A}_{2,q}, \vec{v}_{w(i)}\}$ is positive.

Several configurations of 2-cells are possible above a given crossing with prescribed colors. In the case of an inhomogeneous crossing, $w(i)$ equals either $c(f(i))$ or $c(i+1)$, and $w(f(i))$ equals either $c(i)$ or $c(i + 1)$. In addition the crossing may be positive or negative. One possible configuration of cells for a positive inhomogeneous crossing is shown in Figure 11. In this figure, $c(i) = 2$, $c(i + 1) = 1$, and $c(f(i)) = 3$. From this figure one can see that $w(i) = 3$ and $w(f(i)) = 2$.

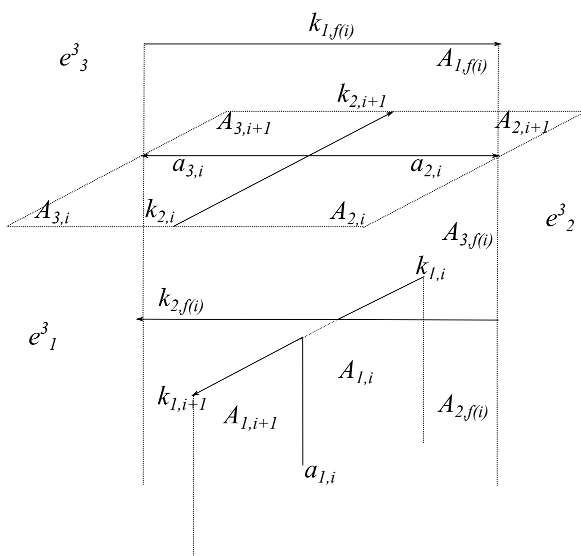

FIGURE 11. One possible configuration of the cells above an inhomogeneous positive crossing i, where k_i is colored 2, k_{i+1} is colored 1, and $k_{f(i)}$ is colored 3.

In the case of a homogenous crossing, the colors $c(i), c(i+1)$ and $c(f(i))$ are all equal, and the 3-cell $e^3_{c(i)}$ is adjacent to the arcs $k_{1,i}$, $k_{1,i+1}$, and $k_{1,f(i)}$. There are, however, multiple possible configurations for the 3-cells near the degree 2 lifts of k; the value of $w(i)$ either coincides with $w(f(i))$, or not. One such configuration, for a positive homogenous crossing, is pictured in Figure 12. For this crossing, all arcs are colored 3, and $w(i) \neq w(f(i))$.

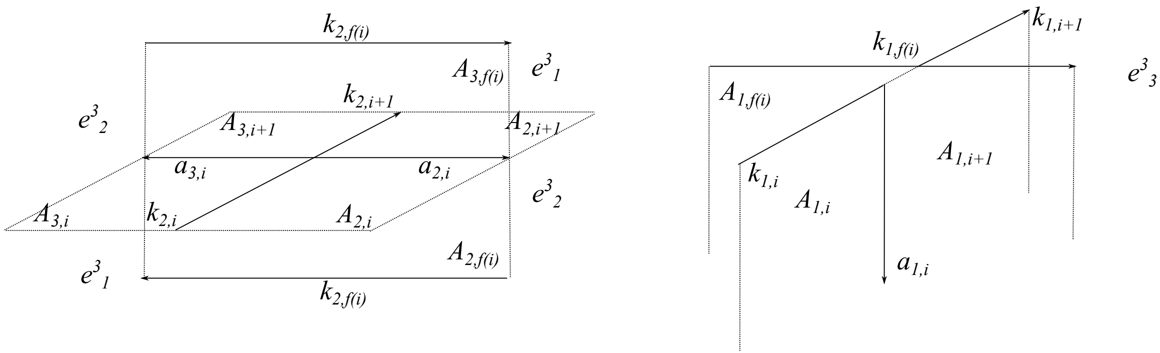

FIGURE 12. One possible configuration of the cells above a homogeneous positive crossing i, where all arcs are colored 3. The two copies of $k_{2,f(i)}$ are identified.

The complete lift of the 2-skeleton may now be constructed by gluing together the configurations above each crossing.

5.2. **Finding surfaces bounding the degree 1 and 2 lifts of α.** Now we recall Perko's method for finding surfaces with boundary equal to the index 1 or 2 branch curves, denoted α_1 and α_2. Such surfaces will be linear combinations of the 2-cells $A_{1,i}$, $A_{2,i}$, and $A_{3,i}$. That is, we seek integers x_i, y_i and z_i such that

$$\partial\left(\sum_i z_i A_{1,i} + x_i A_{2,i} + y_i A_{3,i}\right) = \sum_i k_{1,i} \text{ or } \sum_i k_{2,i}.$$

The strategy is as follows: the 1-chains $a_{1,i}$, $a_{2,i}$ and $a_{3,i}$ appear only above crossing i. Depending on the configuration of cells above the crossing, they may appear in the boundaries of any of the 2-cells $A_{1,i}$, $A_{2,i}$, $A_{3,i}$, $A_{1,f(i)}$, $A_{2,f(i)}$ or $A_{3,f(i)}$. These 2-cells appear x_i, y_i, z_i, $x_{f(i)}$, $y_{f(i)}$, or $z_{f(i)}$ times in the 2-chain $\sum_i z_i A_{1,i} + x_i A_{2,i} + y_i A_{3,i}$.

We begin with k_1. In this case $z_i = 1$ for all $i \in \{0, \ldots, n-1\}$, because k_1 must appear exactly once in the boundary. In addition $y_i = -x_i$ for all $i \in \{0, \ldots, n-1\}$, because none of the $k_{2,i}$ may appear in the boundary.

Now define two signs which we associate to an inhomogenous crossing:

$$\epsilon_1(i) = 1 \text{ if } c(i) \neq w(f(i)), \text{ and } \epsilon_1(i) = -1 \text{ if } c(i) = w(f(i)), \text{ and}$$
$$\epsilon_2(i) = 1 \text{ if } c(f(i)) = w(i), \text{ and } \epsilon_2(i) = -1 \text{ if } c(f(i)) \neq w(i).$$

Therefore at each inhomogenous crossing we get the following equation:

$$x_i(a_{2,i} - a_{3,i}) - x_{i+1}(a_{2,i} - a_{3,i}) + \epsilon_1(i)\epsilon_2(i)x_{f(i)}(a_{2,i} - a_{3,i}) = \epsilon(i)\epsilon_1(i)(a_{2,i} - a_{3,i}).$$

The first two terms are the contribution of the boundaries of $A_{2,i}$, $A_{3,i}$, $A_{2,i+1}$ and $A_{3,i+1}$. The third term is the contribution of the boundary of either $A_{2,f(i)}$ or $A_{3,f(i)}$, depending on which one contains $a_{2,i} - a_{3,i}$ in its boundary (this is determined by the signs ϵ_1 and ϵ_2). The right hand side of the equation is the contribution of the boundary of $A_{1,f(i)}$. Factoring out $a_{2,i} - a_{3,i}$ gives

$$x_i - x_{i+1} + \epsilon_1(i)\epsilon_2(i)x_{f(i)} = \epsilon(i)\epsilon_1(i).$$

Next we look at the homogeneous crossings. Associate the following sign to a homogeneous crossing:

$$\epsilon_3(i) = 1 \text{ if } w(i) \neq w(f(i)) \text{ and } \epsilon_3(i) = -1 \text{ if } w(i) = w(f(i)).$$

By a similar argument as above, we get the following equation:

$$x_i - x_{i+1} + 2\epsilon_3(i)x_{f(i)} = 0.$$

To find the x_i, and hence a 2-chain with boundary k_1, one now simply sets up an inhomogenous matrix equation using the equations above.

Now we turn to k_2. This case is not explicitly written in Perko's thesis, but we need it in order to compute linking numbers of arbitrary curves with the degree 2 branch curve. In this case, the $k_{1,i}$ cannot appear in the boundary, so all the z_i are zero. Also, because each $k_{2,i}$ appears exactly once in the boundary, we have $y_i = 1 - x_i$ for all i. In this case, the equations to solve are

$$x_i - x_{i+1} + \epsilon_1(i)\epsilon_2(i)x_{f(i)} + \frac{\epsilon_2(i)}{2}\left(\epsilon(i) - \epsilon_1(i)\right) = 0$$

for inhomogeneous crossings, and

$$x_i - x_{i+1} + 2\epsilon_3(i)x_{f_i} - \epsilon_3(i) = 0$$

for homogeneous crossings.

In summary, we have the following:

Proposition 5 (Perko [19]). *Let n denote the number of crossings in a diagram for the knot α, let $f(i)$ denote the index of the overstrand $k_{f(i)}$ at crossing i, and let the signs ϵ, and ϵ_x for $x = 1, 2, 3$ be as defined above. If the following inhomogeneous system of linear equations*

$$\begin{cases} x_i - x_{i+1} + \epsilon_1(i)\epsilon_2(i)x_{f(i)} = \epsilon(i)\epsilon_1(i) & \textit{if crossing } i \textit{ is inhomogeneous} \\ x_i - x_{i+1} + 2\epsilon_3(i)x_{f(i)} = 0 & \textit{if crossing } i \textit{ is homogeneous} \end{cases}$$

has a solution $(x_0, x_1, \ldots, x_{n-1})$ over $\mathbb{Q}$ then the degree 1 branch curve is rationally null-homologous and a multiple of it is bounded by the 2-chain

$$\sum_{i=0}^{n-1} A_{1,i} + x_i(A_{2,i} - A_{3,i}).$$

Similarly if the following system

$$\begin{cases} x_i - x_{i+1} + \epsilon_1(i)\epsilon_2(i)x_{f(i)} = \frac{\epsilon_2(i)}{2}\left(\epsilon_1(i) - \epsilon(i)\right) & \textit{if crossing } i \textit{ is inhomogeneous} \\ x_i - x_{i+1} + 2\epsilon_3(i)x_{f(i)} = 0 & \textit{if crossing } i \textit{ is homogeneous} \end{cases}$$

has a solution $(x_0, x_1, \ldots, x_{n-1})$ over $\mathbb{Q}$ then the degree 2 branch curve is rationally null-homologous and a multiple of it is bounded by the 2-chain

$$\sum_{i=0}^{n-1} x_i A_{2,i} + (1 - x_i)A_{3,i}.$$

6. (Appendix B) The computer program

```
def wallcolorchange(oldcolor,wallcolor):
    if oldcolor==wallcolor:
        newcolor=oldcolor
    elif oldcolor != wallcolor:
        s=set()
        s.add(1)
        s.add(2)
        s.add(3)
        s.discard(oldcolor)
        s.discard(wallcolor)
        newcolor=s.pop()
    return newcolor

def pseudolifts(subknotcolors,p1overnums,p1overtypes):
    l=len(p1overnums)
    lift1=[1]
    lift2=[2]
    lift3=[3]
    for i in range(0,l-1):
        if p1overtypes[i]=='k':
            newcell1=wallcolorchange(lift1[i],subknotcolors[p1overnums[i]])
            lift1.append(newcell1)
            newcell2=wallcolorchange(lift2[i],subknotcolors[p1overnums[i]])
            lift2.append(newcell2)
            newcell3=wallcolorchange(lift3[i],subknotcolors[p1overnums[i]])
            lift3.append(newcell3)
        else:
            lift1.append(lift1[i])
            lift2.append(lift2[i])
            lift3.append(lift3[i])
    return lift1, lift2, lift3

def pseudo2lifts(subknotcolors, p2overnums, p2overtypes):
    l=len(p2overnums)
    lift1=[1]
    lift2=[2]
    lift3=[3]
    for i in range(0,l-1):
        if p2overtypes[i]=='k':
            newcell1=wallcolorchange(lift1[i],subknotcolors[p2overnums[i]])
            lift1.append(newcell1)
            newcell2=wallcolorchange(lift2[i],subknotcolors[p2overnums[i]])
```

```
            lift2.append(newcell2)
            newcell3=wallcolorchange(lift3[i],subknotcolors[p2overnums[i]])
            lift3.append(newcell3)
        else:
            lift1.append(lift1[i])
            lift2.append(lift2[i])
            lift3.append(lift3[i])
    return lift1, lift2, lift3

def subwhereisA2(subknotcolors,subknottypes,subknotovernums):
    l=len(subknottypes)
    if subknotcolors[0]==1:
        where=[2]
    else:
        where=[1]
    for j in range(0,l-1):
        if subknottypes[j]=='k':
            where.append(wallcolorchange(where[j],subknotcolors[subknotovernums[j]]))
        elif subknottypes[j]=='p':
            where.append(where[j])
    return where

def xingsign1(i,subknotcolors,subknottypes,subknotovernums):
    if subwhereisA2(subknotcolors,subknottypes,subknotovernums)[subknotovernums[i]]
!=subknotcolors[i]:
        s=1
    else:
        s=-1
    return s

def xingsign2(i,subknotcolors,subknottypes,subknotovernums):
    if subwhereisA2(subknotcolors,subknottypes,subknotovernums)[i]
!=subknotcolors[subknotovernums[i]]:
        s=-1
    else:
        s=1
    return s

def xingsign3(i,subknotcolors,subknottypes,subknotovernums):
    if subwhereisA2(subknotcolors,subknottypes,subknotovernums)[i]
==subwhereisA2(subknotcolors,subknottypes,subknotovernums)[subknotovernums[i]]:
        s=-1
    else:
        s=1
```

```
    return s

def p1surfacecoefmatrix(subknotcolors,subknottypes,subknotovernums,subknotsigns,
                        p1signs,p1overnums,lift):
    n=len(subknotcolors)
    coefmatrix= [[0 for x in range(n+1)] for x in range(n)]
    for i in range(0,n):
            coefmatrix[i][i]+=1
            coefmatrix[i][(i+1)%n]-=1
            if subknottypes[i]=='k' and
                      subknotcolors[i]!=subknotcolors[subknotovernums[i]]:

                coefmatrix[i][subknotovernums[i]]+=
                 xingsign1(i,subknotcolors,subknottypes,subknotovernums)
                 *xingsign2(i,subknotcolors,subknottypes,subknotovernums)

            elif subknottypes[i]=='k' and
                      subknotcolors[i]==subknotcolors[subknotovernums[i]]:

                coefmatrix[i][subknotovernums[i]]+=
                   xingsign3(i,subknotcolors,subknottypes,subknotovernums)*2

            elif subknottypes[i]=='p' and
                         lift[subknotovernums[i]]==subknotcolors[i]:
                coefmatrix[i][n]=0

            elif subknottypes[i]=='p' and
                         lift[subknotovernums[i]]==
                  subwhereisA2(subknotcolors,subknottypes,subknotovernums)[i]:

                coefmatrix[i][n]=-subknotsigns[i]

            elif subknottypes[i]=='p' and lift[subknotovernums[i]]!=
                subwhereisA2(subknotcolors,subknottypes,subknotovernums)[i]:

                coefmatrix[i][n]=subknotsigns[i]
    return coefmatrix

def solvefor2chain(matrixofcoefs, numcrossings):
    M=Matrix(matrixofcoefs)
    pivots=M.rref()[1]
    numpivots=len(pivots)
    RR=M.rref()[0]
    x=[0 for j in range(numcrossings)]
```

```
    if numcrossings in pivots:
        return 'False'

    else:
        for i in range(0,numpivots):
            x[pivots[i]]=-RR[i,numcrossings]

        return x

total=[[0,0,0],[0,0,0],[0,0,0]]
l=len(p2overnums)
p2lifts=pseudo2lifts(subknotcolors, p2overnums, p2overtypes)
p1lifts=pseudolifts(subknotcolors,p1overnums,p1overtypes)
coeflists=[]
coeflists.append(solvefor2chain(p1surfacecoefmatrix(subknotcolors,subknottypes,
               subknotovernums,subknotsigns,p1signs,p1overnums,p1lifts[0]),len(subknotsigns)))
coeflists.append(solvefor2chain(p1surfacecoefmatrix(subknotcolors,subknottypes,
               subknotovernums,subknotsigns,p1signs,p1overnums,p1lifts[1]),len(subknotsigns)))
coeflists.append(solvefor2chain(p1surfacecoefmatrix(subknotcolors,subknottypes,
               subknotovernums,subknotsigns,p1signs,p1overnums,p1lifts[2]),len(subknotsigns)))
where=subwhereisA2(subknotcolors,subknottypes,subknotovernums)
for s in range(0,3): # Three lifts of 1st pseudo-branch curve
    if coeflists[s]!='False':
        for i in range(0,l):# Arcs of the second pseudo-branch curve
            if p2overtypes[i]=='k': # If the knot passes over the 2nd pseudo-branch curve
                for j in range (0,3):
                    if p2lifts[j][i]==subknotcolors[p2overnums[i]]:
                        total[s][j]+=0
                    elif p2lifts[j][i]==where[p2overnums[i]]:
                        total[s][j]+=coeflists[s][p2overnums[i]]

                    elif p2lifts[j][i]!=where[p2overnums[i]]:
                        total[s][j]-=coeflists[s][p2overnums[i]]

            if p2overtypes[i]=='p':
                for j in range (0,3):
                    if p2lifts[j][i]!=p1lifts[s][p2overnums[i]]:
                        total[s][j]+=0

                    elif p2lifts[j][i]==p1lifts[s][p2overnums[i]]:
                        total[s][j]+=p2signs[i]
```

```
    else:
        total[s][0]='x'
        total[s][1]='x'
        total[s][2]='x'

return total
```

7. (Appendix C) Applying the program to pseudo-branch curves for α

We used the following input to generate the results presented in Section 4:

7.1. **The characteristic knot β is the first pseudo-branch curve.** The list of overstrand numbers $f(i)$ for α is $(7, 0, 12, 7, 6, 10, 3, 5, 6, 3, 2, 0, 0, 3)$.

The corresponding list of signs for the knot α is $(-1, -1, 1, 1, -1, 1, -1, 1, 1, 1, 1, 1, 1, 1)$.

The list of crossing types is $(p, p, k, k, p, k, p, k, p, p, p, k, p, k)$.

The list of colors is $(1, 1, 3, 2, 1, 1, 3, 3, 2, 2, 2, 2, 3, 3)$.

The list of overstrand numbers for the first pseudo-branch curve $\gamma = \beta$ is $(12, 0, 10, 6, 5, 7, 5, 0, 12, 3)$.

The corresponding list of signs is $(1, -1, 1, -1, -1, 1, -1, -1, 1, -1)$.

The list of crossing types is $(k, k, k, p, k, k, k, p, k, k)$.

The program returns the matrix

$$[[0, -1, -1, -1, -1, 0, 1, 0, 0, 0, 1, 0, 0, 1], [0, 0, 1, 1, 1, 1, 0, 0, -1, 0, -1, 0, 0, -1],$$
$$[0, 1, 0, 0, 0, -1, -1, 0, 1, 0, 0, 0, 0, 0]],$$

which is the list of coefficients x_i^j of the 2-cells $A_{2,i}$ in the 2-chain bounding lift i of β. These coefficients are organized in Table 1.

β is the first pseudo-branch curve and ω_1 is the second pseudo-branch curve. The list of overstrand numbers for the second pseudo-branch curve ω_1 is $(0, 12, 0, 5, 6, 7)$.

The corresponding list of signs is $(-1, 1, -1, 1, 1, -1)$.

The list of crossing types is (p, k, k, k, p, k).

The output of the program is $[[0, 0, 0], [-1, 0, 1], [1, 0, -1]]$. The interpretation of this matrix is given in Section 4.

β is the first pseudo-branch curve and ω_2 is the second pseudo-branch curve.

The list of overstrand numbers for the second pseudo-branch cuve ω_2 is $(10, 3, 6, 5)$.

The corresponding list of signs is $(1, -1, -1, -1)$.

The list of crossing types is (k, p, p, k).

The output of the program is $[[-1,-1,0],[-1,1,-2],[0,-2,0]]$. The interpretation of this matrix is given in Section 4.

7.2. **The curve ω_1 in $V-\beta$ is the first pseudo-branch curve.** The list of over-crossing numbers $f(i)$ for the subdivided knot diagram is $(0,9,5,3,7,4,3,2,0,2)$.

The list of crossing types is (p,k,k,p,k,k,p,p,k,k).

The list of colors is $(1,1,2,1,1,3,2,2,2,3)$.

The list of signs is $(-1,1,1,1,1,1,-1,1,1,1)$.

The program returns the matrix

$$[[0,0,0,0,-1,0,1,1,0,0],'False','False'],$$

which is the list of coefficients x_i^j of the 2-cells $A_{2,i}$ in the 2-chain bounding lift i of β is given in Table 2. The $'False'$ entries signal that the second and third lifts of ω_1 are not zero-homologous.

ω_1 is the first pseudo-branch curve and β is the second pseudo-branch curve.

The list of overstrand numbers for the second pseudo-branch curve β is $(9,0,2,7,3,4,5,3,4,0,9,2)$.

The corresponding list of signs is $(1,-1,-1,1,1,-1,1,1,-1,-1,1,-1)$.

The list of crossing types is $(k,k,p,k,p,k,k,p,k,p,k,k)$.

The output of the program is $[[0,-1,1],['x','x','x'],['x','x','x']]$. The interpretation of this matrix is given in Section 4.

References

1. James Alexander, *Note on Riemann spaces*, Bulletin of the American Mathematical Society **26** (1920), no. 8, 370–372.
2. Carl Bankwitz and Hans Georg Schumann, *Über viergeflechte*, Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg, vol. 10, Springer, 1934, pp. 263–284.
3. Gerhard Burde, *On branched coverings of s3*, Canad. J. Math **23** (1971), 84–89.
4. ______, *Links covering knots with two bridges*, Kobe journal of mathematics **5** (1988), 209–220.
5. Sylvain Cappell and Julius Shaneson, *Branched cyclic coverings*, Knots, Groups, and 3-Manifolds (1975), 165–173.
6. ______, *Invariants of 3-manifolds*, Bulletin of the American Mathematical Society **81** (1975), no. 3, 559–562.
7. ______, *Linking numbers in branched covers*, Contemporary Mathematics **35** (1984), 165–179.
8. Ralph Fox, *Metacyclic invariants of knots and links*, Canad. J. Math **22** (1970), 193–201.
9. Ralph H Fox, *Covering spaces with singularities*, Algebraic Geometry and Topology: A Symposium in Honor of S. Lefschetz, 1957, pp. 243–257.
10. Richard Hartley, *Metabelian representations of knot groups*, Pacific Journal of Mathematics **82** (1979), no. 1, 93–104.
11. ______, *Identifying non-invertible knots*, Topology **22** (1983), no. 2, 137–145.

12. Richard Hartley and Kunio Murasugi, *Covering linkage invariants*, Canad. J. Math **29** (1977), 1312–1339.
13. Hugh Hilden, *Every closed orientable 3-manifold is a 3-fold branched covering space of S^3*, Bulletin of the American Mathematical Society **80** (1974), no. 6, 1243–1244.
14. Alexandra Kjuchukova, *On the classification of irregular dihedral branched covers of topological four-manifolds*, arXiv preprint arXiv:1608.03329 (2016).
15. Paolo Lisca, *Lens spaces, rational balls and the ribbon conjecture*, arXiv preprint math/0701610 (2007).
16. R Litherland, *A formula for the casson–gordon invariant of a knot*, preprint (1980).
17. José María Montesinos, *A representation of closed orientable 3-manifolds as 3-fold branched coverings of S^3*, Bulletin of the American Mathematical Society **80** (1974), no. 5, 845–846.
18. ———, *4-manifolds, 3-fold covering spaces and ribbons*, Transactions of the American mathematical society **245** (1978), 453–467.
19. Kenneth Perko, *An invariant of certain knots*, Undergraduate Thesis (1964).
20. ———, *On covering spaces of knots*, Glasnik Mat **9** (1974), no. 29, 141–145.
21. ———, *On the classification of knots*, Proc. Am. Math. Soc **45** (1974), 262–266.
22. ———, *On dihedral covering spaces of knots*, Inventiones mathematicae **34** (1976), no. 2, 77–82.
23. Kenneth A Perko, *Historical highlights of non-cyclic knot theory*, Journal of Knot Theory and Its Ramifications **25** (2016), no. 03, 1640010.
24. Kurt Reidemeister, *Knoten und verkettungen*, Mathematische Zeitschrift **29** (1929), no. 1, 713–729.
25. Robert Riley, *Homomorphisms of knot groups on finite groups*, mathematics of computation **25** (1971), no. 115, 603–619.
26. Morwen B Thistlethwaite, *Knot tabulations and related topics*, Aspects of topology (1985), 1–76.