FLAT KNOTS AND INVARIANTS

By JIE CHEN

A Thesis Submitted to the School of Graduate Studies in Partial Fulfilment of the Requirements for the Degree PhD of Science

Abstract

This thesis concerns flat knots and their properties. We study various invariants of flat knots, such as the crossing number, the u-polynomial, the flat arrow polynomial, the flat Jones-Krushkal polynomial, the based matrices, and the ϕ-invariant. We also examine the behavior of these invariants under connected sum and cabling. We give a matrix-based algorithm to calculate the flat Jones-Krushkal polynomial.

We take a special interest in certain subclasses of flat knots, such as almost classical flat knots, checkerboard colorable flat knots, and slice flat knots. We explore how the invariants can be used to obstruct a flat knot from being almost classical, checkerboard colorable, or slice.

We show that any minimal crossing diagram of a composite flat knot is a connected sum, and we introduce a skein formula for the constant term of the flat arrow polynomial.

A companion project to this thesis is the interactive website, FlatKnotInfo. It provides a curated dataset of examples and invariants of flat knots. It also features a tool for searching flat knots and another tool that crossreferences flat knots with virtual knots. FlatKnotInfo was used to develop many of the results in this thesis, and we hope others find it useful for their research on flat knots. The Python code for calculating based matrices and flat Jones-Krushkal polynomials is included in an appendix.

Acknowledgements

I would like to express my genuine gratitude to my Ph.D. advisor, Dr. Hans Boden, for their invaluable guidance and strong support throughout my masters and doctoral programs. You have not only been a dedicated mentor but also a remarkable role model that has a profound impact on my own growth as a researcher and an individual.

I am also grateful to my thesis committee, Dr. Andrew Nicas and Dr. Adam Van Tuyl, for their constructive feedback and thoughtful suggestions that have significantly enriched the quality of this work. Special thanks are due to Dr. Patricia Cahn for her meticulous reading of this thesis and all the feedback and valuable suggestions she provided. I would also like to thank to Dr. Dror Bar-Natan for his encouragement and for generously sharing his knowledge in calculating Jones polynomials.

My gratitude extends to Dr. David Lozinski for his mentorship and guidance during my first teaching experience.

I sincerely appreciate Dr. Lindsay White, whose pioneering work in tabulating flat knots has been a constant source of inspiration for my own research. I am also thankful to Dr. Homayun Karimi for his generous help and for sharing his knowledge on virtual knots, and to Robin Gaudreau for their feedback and many suggestions that helped to improve this thesis.

A special thanks goes to my fellow peer students, Mr. Elkin Ramirez and Mr. Subhajit Mishra, whose camaraderie and support have made this academic journey not only intellectually stimulating but also enjoyable. Your friendship and collaborative efforts have been a source of motivation and encouragement.

I owe a debt of gratitude for Canada's diverse and inclusive culture. I am more than grateful to the countless individuals, including strangers, who helped me a lot in my daily life here. Their acts of kindness, support, and understanding have taught me humanity and the power of goodwill.

I am also deeply thankful for the generous research funding provided by McMaster University, James Stewart Research Support Awards, Movmi, and Mitacs.

Finally, I want to express my deep gratitude to Xu Cao, Simone Gomes, and Dr. Yushan Zhang for their unwavering support, understanding, and encouragement.

I am truly fortunate to have had such a supportive and inspiring network of mentors, colleagues, and loved ones.

Thank you all for being a part of this incredible journey.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Knots and Gauss diagrams

In *classical knot theory*, the objects of interest are smooth or piecewise-linear embeddings of S^1 in S^3 up to isotopy. A classical knot can be represented as a knot diagram (not uniquely), and an isotopy of knots can be represented by a finite sequence of Reidemseister moves between diagrams. There is another, purely combinatorial way to represent classical knots using Gauss codes or Gauss diagrams. The Reidemeister moves translate into moves between Gauss diagrams, and there is a bijection between these two representations of classical knots.

Given a knot diagram, there is a unique Gauss diagram corresponding to it, but not every Gauss diagram arises from a knot diagram. To address this deficiency, Kauffman introduced a new type of crossing called a *virtual crossing*, and this led to the development of *virtual knot theory* [Kau99]. Virtual knots give a complete theory of Gauss diagrams in that every Gauss diagram can be realized by some virtual knot.

Virtual knots can also be represented as knots in thickened surfaces. Carter, Kamada and Saito gave a one-to-one correspondence between virtual knots and knots in thickened surfaces up to stable equivalence [CKS02]. Thus, the three representations of classical knots extend to virtual knots. Many invariants of classical knots extend to virtual knots such as the Jones polynomial and Khovanov homology (cf. [Kau99, GPV00, MI13]), but virtual knots also exhibit new and unexpected behavior. For example, the operation of connected sum is more complicated; it is only well-defined in the category of long virtual knots. As well, there exists nontrivial virtual knots with trivial Jones polynomial and even trivial Khovanov homology. For classical knots, it is an open problem whether the Jones polynomial detects the unknot, and it is a deep result of Kronheimer and Mrowka [KM11] that Khovanov homology detects the unknot.

1.2 Flat knots

Flat knots arise from considering virtual knots up to crossing change. As is well-known, every classical knot diagram can be transformed into a diagram of the unknot by crossing changes. This property fails for virtual knots, and the Kishino knot (Figure 1.1) gives an example of a virtual knot that cannot be unknotted by crossing changes. Thus, if one is interested in studying virtual knots up to crossing change, one is inexorably led to the study of *flat virtual knots*, which are represented by *flat virtual knot diagrams* in [Kau99]. (In [Tur04], they are called *virtual strings*.) In this thesis, we use the terms flat knots and flat knot diagrams. In a flat knot diagram, the over and under crossings are replaced with self-intersections. Thus, flat knots are equivalence classes of virtual knots modulo crossing change, and a virtual knot can be unknotted by crossing changes if and only if it projects to the trivial flat knot.

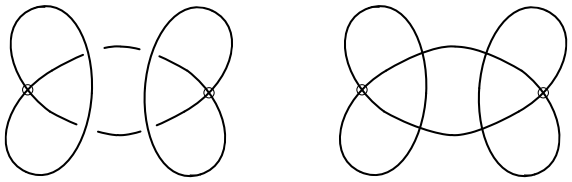


Figure 1.1: The virtual and flat Kishino knot

A central problem in knot theory is the classification problem, which asks when are two knots different and when are they equivalent. The prime decomposition theorem for knots reduces this question to one about prime knots, roughly analogous to tabulating the prime numbers. For virtual knots, the prime decomposition theorem fails and the analogy breaks down; see [Mat12]. Indeed, the Kishino knot is nontrivial even though it is a connected sum of two trivial virtual knot diagrams. For classical knots, connected sum is a commutative operation, whereas for virtual knots, it is not commutative. These properties are true also for flat knots, and they reflect the different behavior present in the monoids of long virtual and flat knots.

In this thesis, we study flat knots using invariants such as the u-polynomial, the flat arrow polynomial, the flat Jones-Krushkal polynomials, primitive based matrices, and the ϕ-invariants.

1.3 Classification and monotonicity

In [Tur04], Turaev developed an algorithm for classifying flat knots. It has been implemented by Gibson [Gib08] and is the basis for FlatKnotInfo [FKI]. One of the advantages of this algorithm is that, under reduction, the number of crossings monotonically decreases until one achieves a minimal crossing diagram. Further, any two minimal diagrams of the same flat knot are related by a sequence of flat Reidemeister 3-moves. (The corresponding statement is not true for classical knots.

In general, one may need to increase the number of crossings in any sequence of Reidemeister moves relating two minimal crossing diagrams of the same classical knot. It is an open problem to find a good bound on this number.)

An alternative approach for classifying flat knots was developed by Chu [Chu13], but her methods apply to long flat knots, and the canonical diagrams that are used as representatives do not always have minimal crossing number.

The monotonicity result for classifying flat knots and links has a somewhat storied history. Hass and Scott showed that any curve on a surface can be reduced to minimal intersection number monotonically in [HS94]. In [Kad03], Kadokami claimed that any flat link diagram can be reduced in a monotonically decreasing way until one obtains a minimal crossing diagram. However, Gibson found counterexamples to Kadokami's claim in [Gib08]. Cahn showed that Kadokami's claim is true for flat knots in [Cah17]. A complete solution was provided by Freund [Fre22], who proved Kadokami's claim for nonparallel flat links.

Monotonicity provides the following general scheme for classifying flat knots. Given a flat knot diagram, we first apply Reidemeister moves to reduce the crossing number. After a finite number of reductions, one obtains a minimal crossing diagram. Next, one determines all diagrams related to the minimal one by Reidemeister 3-moves. Since any two minimal diagrams of the same flat knot are related by Reidemeister 3-moves, this set of minimal diagrams can be used to completely classify the flat knot type.

This scheme can be implemented as an algorithm for tabulating flat knots up to n-crossings. The first step is to construct all flat knot diagrams up to n-crossings. Each flat knot diagram is reduced to a minimal crossing diagram, which is possible by monotonicity. The next step is to determine the Reidemeister 3 orbit of each minimal crossing diagram and record a unique representative for each one. This step uses a linear ordering on the set of flat knot diagrams and results in a unique "name" for each flat knot. The last step is to validate the results by calculating enough invariants of the flat knots to distinguish each pair of flat knots in the table. The majority of non-distinguished flat knots are pairs of permutant flat knots. (Permutant knots are defined in Section 2.8.)

1.4 FlatKnotInfo

As previously mentioned, Gibson applied this method to tabulate flat knots up to 4 crossings in [Gib08]. He represented flat knots as nanowords (cf. [Tur06]) and used the u-polynomial, ϕ-invariants, and the 2-parity projection to distinguish the flat knot types. This approach works for flat knots with up to 4 crossings, but the invariants are not sufficiently powerful to distinguish flat knots with five or more crossings.

Inspired by the famous knot theory website KnotInfo [KI], we created a website for flat knots called FlatKnotInfo [FKI]. We also borrowed the idea from the table

of virtual knots [Gre04] to include a page for each flat knot. In tabulating flat knots [FKI], we used the following modified approach. Firstly, we represent Gauss diagrams using Lyndon words with matchings, which is more efficient than using nanowords, see Figure 1.2. It is especially useful in generating the tables of checkerboard colorable

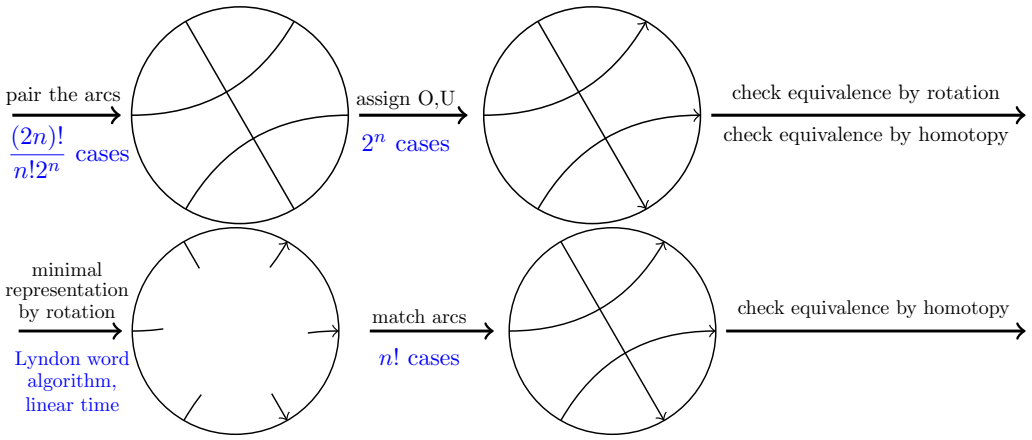


Figure 1.2: Two approaches (i) nanowords (Gibson, Turaev) and (ii) Lyndon words (Boden, White)

flat knots and almost classical flat knots. For those flat knots, the tables include higher crossing flat knots by using the "OU"-pattern: Along the flat knot Gauss diagram we label the singular points $1, 2, \ldots, 2n$. If two singular points are matched as the head and tail of an arrow, then they should have even and odd numbers and be non-consecutive. We assume the tails are odd points and pair them up with the even points. We can always assume the first arrow has the least head-tail difference. This allows us to consider much fewer than $(n-3)((n-1)!)$ cases. On each pattern, we can flip the head and tail of the arrow to obtain a checkerboard colorable diagram, so the case number is multiplied by 2^n. For each one, we apply Reidemeister moves until we obtain a minimal diagram and then we find its 3-orbit and minimal representative (in the linear ordering).

Secondly, we employ a larger set of flat knot invariants to distinguish the flat knot types. This includes the u-polynomial, the flat arrow polynomial, the flat Jones-Krushkal polynomial, and the ϕ-invariants of the flat knots and its cables.

As a result, we are able to tabulate the first 1,289,741 flat knots. We completely distinguish flat knots up to 6 crossings, and our method works for flat knots with 7 crossings leaving only 5 pairs of ambiguities. For flat knots with 8 crossings, the method leaves a total of 511 undistinguished, but most of the ambiguities arise from composite flat knots. Indeed, if we restrict our attention to prime flat knots, the invariants are able to separate all flat knots up to 8 crossings except for one pair with 7 crossings.

We also consider the classification problem for the subclasses of checkerboard colorable and almost classical flat knots. For instance, we tabulate the first 1,379,884

checkerboard colorable flat knots and the first 240,759 almost classical flat knots. For checkerboard colorable flat knots, the invariants are able to completely distinguish them up to 7 crossings and the method works for checkerboard colorable flat knots with 8 crossings leaving only four ambiguities. For almost classical flat knots, the invariants completely distinguish them up to 8 crossings, leaving only 3 pairs of 9-crossing knots undistinguished.

The table in FlatKnotInfo [FKI] lists diagrammatic invariants such as the Reidemeister-3 orbit, the symmetry type, and parity projection, as well as many invariants of flat knots. It also lists concordance information such as the genus of the based matrix and sliceness of flat knots up to 6 crossings with one exception (6.540). There is a tool that cross-references each flat knot to the corresponding virtual knot in Green's table [Gre04], along with a flat knot calculator that finds the minimal diagram, name, and symmetry type of a given flat knot diagram.

FlatKnotInfo [FKI] helped to inform and guide much of the research in this thesis. The code for calculating the based matrix and the Jones-Krushkal polynomials is included in the appendix.

1.5 Roadmap

The rest of the thesis is structured as follows: Basic definitions and conventions are given in Chapter 2. In Chapter 3, we review the based matrix and several invariants derived from it. We show an example of an algebraically slice but not slice flat knot and a calculation of the $(2,1)$-cabled based matrix. In Chapter 4, we discuss the flat arrow polynomial. We use it to give an obstruction to flat knots being checkerboard colorable, and we develop a skein model for the constant term of the cabled flat arrow polynomial. In Chapter 5 we discuss the flat Jones-Krushkal polynomial and its enhanced version. A matrix-based calculation for the Jones-Krushkal polynomial is given, and this invariant is helpful in distinguishing almost classical flat knots with higher crossing number. In Chapter 6, we revisit the notions of sliceness and concordance for long flat knots. We show an example of a non-slice Brunnian link and one 6-crossing, eight 7-crossing flat knots whose sliceness status remains unknown.

The final chapter contains a summary of the key theorems and further discussion of the many open problems. It also presents a table of the numbers of flat knot by rank and type, as well as a list of open problems for future study.

In addition, there are three appendices. The first is a dictionary for translating between Kauffman's terminology (as used here) and Turaev's terminology (as used elsewhere). The other two appendices contain the python code used to compute based matrices and flat Jones-Krushkal polynomials.

Chapter 2

Basic notions

In this chapter, we introduce classical, virtual, and flat knots. We also discuss symmetries of flat knots, Alexander numberings, almost classical flat knots, cabling operations, concordance and connected sum. Throughout this thesis, we use Kauffman's terminology for *virtual and flat knots*. Appendix A provides a dictionary for translating these notions into Turaev's language for *virtual strings*.

Most of the material presented in this chapter is definitional and thus well-known. One exception is Section 2.8, where we prove a subadditivity result for the classical crossing number of composite flat knots.

2.1 Classical knots and their representations

In this section, we introduce basic notions from classical knot theory, including knot diagrams, Reidemeister moves, and Gauss diagrams.

Definition 2.1. A *classical knot*, or just *knot*, is an embedding of S^1 into S^3. Two knots K_1 and K_2 in S^3 are said to be *equivalent* if there is an ambient isotopy carrying K_1 to K_2. A (classical n-component) *link* is the disjoint union $\bigsqcup_n S^1$ embedded in S^3, while the components are allowed to tangle with each other. We will use planar diagrams to represent knots and links.

Definition 2.2. Let $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ be a projection. If $\pi(K)$ has a finite number of singular points, and if they are all transverse double points, then $\pi(K)$ is said to be a *regular projection* of K. The double points in the projection are called *crossings*. A *knot diagram* is a regular projection of a knot with every under-crossing line broken at every double point.

In this thesis, we work with oriented knots, and the orientation is indicated by placing an arrow sign on the knot diagram. A given knot in S^3 have many different diagrams. Therefore, we need to define an equivalence relation between two diagrams.

Definition 2.3. The *Reidemeister moves* are the three moves shown in Figure 2.1.

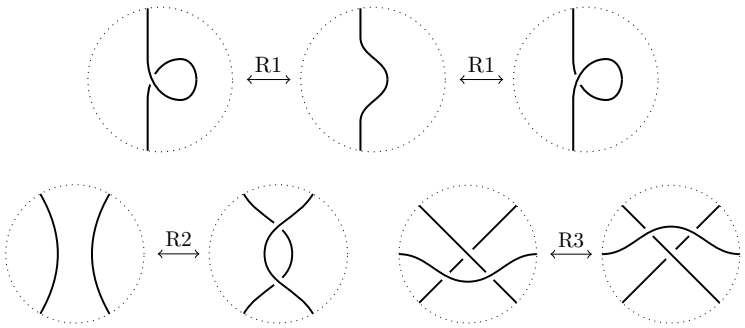


Figure 2.1: Reidemeister moves

Reidemeister proved that two classical knots are equivalent if and only if their diagrams are related by a finite sequence of Reidemeister moves and planar isotopies. The three Reidemeister moves and planar isotopies together generate classical isotopies of knot diagrams.

Aside from knot diagrams, there are other ways to represent knots. For instance, one can represent them using Gauss diagrams.

Definition 2.4. For a knot diagram with n classical crossings, its *Gauss diagram* is a counterclockwise oriented circle with $2n$ points on the circle and n arrows (or "chords") connecting those points. This circle is called the *skeleton* of the Gauss diagram. Every arrow represents a classical crossing with arrow head associated to the under-crossing arc and arrow foot to the over-crossing arc. The arrow head is decorated with a sign $\varepsilon = \pm 1$ according to the writhe of the crossing as in Figure 2.2. The order of the points on the skeleton tell us adjacency of the crossings in the knot diagram.

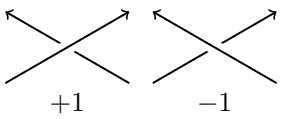


Figure 2.2: Writhe of a crossing

Definition 2.5. A *Gauss code* (or *Gauss word*) is a notation to represent the Gauss diagram. From 12 o'clock of the skeleton in the Gauss diagram, in the order of counterclockwise, assign number $1, 2, \ldots$ to the arrows. At each arrow head (or tail), the point is recorded as "U" (or "O"), followed by this arrow's assigned number and sign. Then from 12 o'clock of the skeleton, go counterclockwise and record every arrow head or tail along the skeleton.

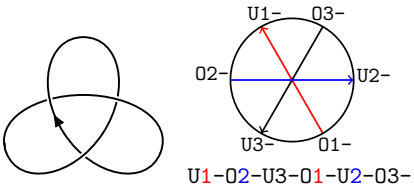


Figure 2.3: Trefoil and its Gauss diagram

The Gauss diagram and Gauss code for the trefoil knot is shown in Figure 2.3. Based on this construction, we can obtain a Gauss diagram. The Gauss code is unique for a given Gauss diagram up to cyclic permutation. The permutation alters the starting point of the Gauss code. If we delete this point from the skeleton, we obtain a new category of knots.

Definition 2.6. A *Gauss diagram of a long knot* is a Gauss diagram of a knot with one point on the skeleton (where no arrow head/tail is located) removed. A *long knot diagram* is a knot diagram with one regular point removed. Two long knot diagrams are said to be equivalent if they are related by finitely many Reidemeister moves and planar isotopy.

Note that classical knot diagrams are completely determined by the associated Gauss diagram, but not every Gauss diagram can be realized as a classical knot diagram.

If the Gauss diagram can be realized as a classical knot, then one can draw all the signed crossings as in Figure 2.2 and connect them by arcs following the adjacency and order given by the Gauss code. Two Gauss diagrams of the same knot are related by the Gauss diagram version of the Reidemeister moves.

2.2 Virtual knots and their representations

In this section, we introduce virtual knots. As previously mentioned, not every Gauss diagram is realized by a classical knot. Virtual knots represent the completion of all Gauss diagrams modulo the Reidemeister moves. Furthermore, the set of classical knots injects faithfully into the set of virtual knots; see [GPV00]. In [Kup03] Kuperberg gave a more general geometric argument showing that the set of classical links injects faithfully into the set of virtual links.

In the last section, we described how to construct a classical knot diagram from a Gauss code by drawing all the crossings and connecting them by arcs. For some Gauss codes, it is not possible to connect the crossings in sequence without introducing additional intersections. In other words, not all Gauss diagrams can be realized by planar knot diagrams. To circumvent these obstacles, we introduce a new type of crossing called a *virtual crossing*, as shown in Figure 2.5. By allowing arcs to intersect

in this new type of crossing (see Figure 2.4), we are able to realize any Gauss code using a virtual knot diagram.

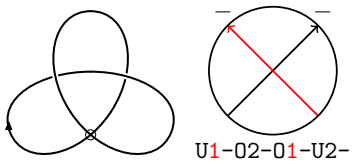


Figure 2.4: Virtual trefoil and its Gauss diagram

Based on this we give the definition of virtual knot diagrams and diagrammatic equivalence.

Definition 2.7. A *virtual knot diagram* is a 4-valent planar graph, but each vertex is now allowed to be a classical crossing or *virtual crossing* as in Figure 2.5.

Figure 2.5: Classical crossing (left) and virtual crossing (right)

There is an equivalence relation defined on virtual knot diagrams.

Definition 2.8. The *virtual Reidemeister moves* are the moves in Figure 2.6. Along with classical isotopies, these virtual Reidemeister moves generate *virtual isotopies* of virtual knot diagrams. Virtual and classical Reidemeister moves are collectively called *generalized Reidemeister moves*.

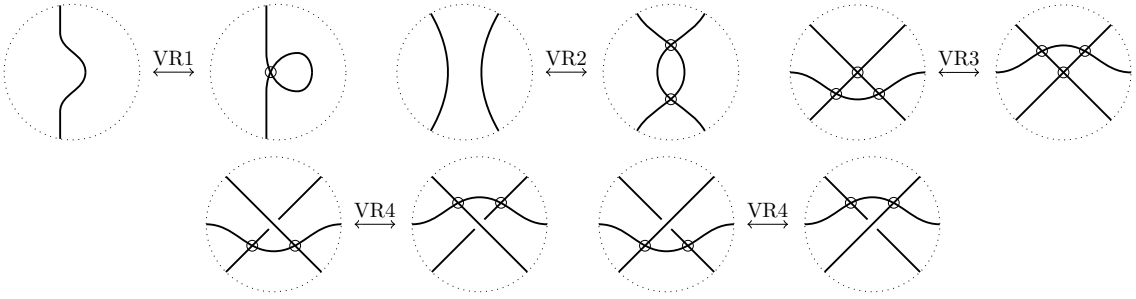


Figure 2.6: Virtual Reidemeister moves

Alternatively, classical isotopies along with detour moves in Figure 2.7 generate virtual knot equivalence. This explains why virtual crossings should be thought to "not really exist". Indeed, virtual crossings are not indicated by the Gauss diagram,

and additionally a Gauss diagram does not change under a detour move. Hence every Gauss diagram represents one virtual diagram up to detour moves and planar isotopies. The *virtual knot type* is an equivalence class of virtual knot diagrams under virtual isotopy.

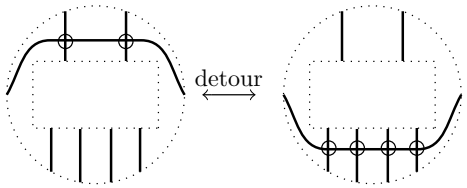


Figure 2.7: Detour move

Recall that classical knots are embeddings of S^1 into S^3, with equivalence given in terms of ambient isotopy. There is an analogous definition for virtual knots.

Definition 2.9. [CKS02] An *embedding representation* of a virtual knot K is an embedding $e_K : S^1 \to \Sigma_g \times I$, where Σ_g is a connected, oriented, closed surface of genus g. Two embedding representations are *stably equivalent* if they are related by a finite sequence of stabilizations, destabilizations and ambient isotopies.

We have explained how knot diagrams and Gauss codes are related. There is also a well-defined map from embedding representations to virtual knot diagrams. A virtual knot diagram can be realized as a simple closed curve embedded in a thickened surface in the following way. If we regard the virtual crossing as in Figure 2.8 (left), and if every classical crossing is realized by thickening the surface as in Figure 2.8 (right), then every virtual knot can be embedded in a thickened surface with boundary. Attaching a 2-disk along each boundary component gives a closed surface which is called the Carter surface of the virtual knot; see [KK00]. This process can be reversed, namely we can obtain a virtual knot diagram from an embedding representation. There is a bijection between virtual knots and knots in a thickened surface up to stable equivalence; see [CKS02] for a detailed explanation.

Figure 2.8: Virtual knots as knots in thickened surfaces

Similarly, we can also obtain a Gauss code directly from the embedding representation of a virtual knot. We can firstly get a "shadow" by projecting $\Sigma_g \times I$ to Σ_g. Then at each double point in the "shadow", the pre-image belongs to two arcs and the arc has a larger coordinate in I is the over-crossing. The rest is the same as in Definition 2.4.

The *genus* $g(K)$ of a virtual knot K is defined to be the minimal genus of the Carter surface over all embedding representations of K. Thus a virtual knot K has genus $g(K) = 0$ if and only if K is classical. Equivalently, a virtual knot K is classical if and only if it can be represented by a virtual knot diagram with no virtual crossings.

2.3 Flat knots and their representations

In this section, we introduce flat knots. They can be described alternatively in terms of flat knot diagrams, flat Gauss diagrams, or immersed curves on surfaces. The flat Reidemeister moves generate an equivalence relation called homotopy. Given a virtual knot diagram, there is an associated flat knot diagram given by flattening all the crossings. This induces a well-defined map from virtual knots to flat knots which is denoted π and called the shadow map.

Definition 2.10. A *flat knot diagram* is a 4-valent planar graph, where each vertex is now allowed to be either a *flat crossing* or virtual crossing as in Figure 2.9.

Figure 2.9: A flat crossing (left) and virtual crossing (right)

Definition 2.11. Two flat knot diagrams are said to be *homotopic* if they are related by a finite sequence of *flat Reidemeister moves* shown in Figure 2.10 and VR1,VR2,VR3 in Figure 2.6 along with ambient isotopies of the surface.

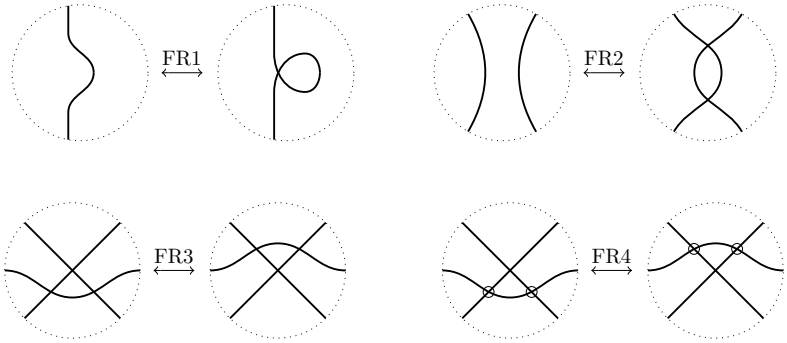


Figure 2.10: Flat Reidemeister moves plus VR1, 2, 3 from Figure 2.6)

Remark 2.12. For oriented diagrams, [Pol10, Theorem 1.1] proved that the Reidemeister moves R1, R2, and R3 in Figure 2.1 assigned with any orientation can be generated by four oriented moves. Their flat knot diagram versions are listed in Figure 2.11.

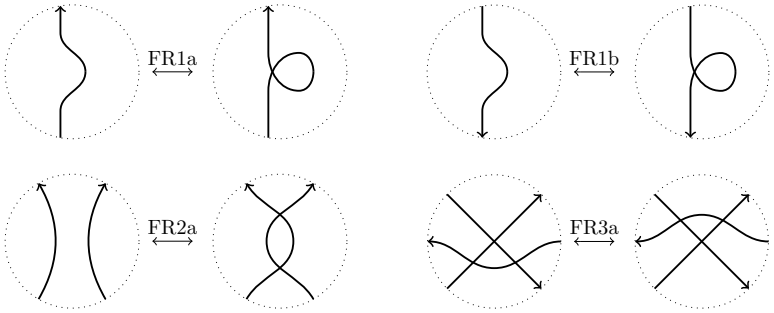


Figure 2.11: Flat Reidemeister moves generating all FR1, 2, 3 moves

Definition 2.13. An *invariant* of (oriented) flat knots is a number, polynomial, group, etc. that can be associated to each flat knot diagram with the property that the object remains unchanged if the diagram is altered by any of the flat Reidemeister moves in Definition 2.11.

The *crossing number* of a flat knot K is denoted $cr(K)$ and defined as follows. Given a flat knot diagram, its crossing number is the number of flat crossings. (Virtual crossings are not counted.) For a flat knot K, $cr(K)$ is defined as the minimum, over all diagrams D for K, of the crossing number of D. Thus, $cr(K)$ is an invariant of the flat knot.

In a similar way, one can define the *virtual crossing number* of a flat knot K as the minimum, over all diagrams D for K, of the number of virtual crossings of D. This also determines an invariant of flat knots.

By Remark 2.12, to check if a flat knot invariant is well-defined, it is enough to check that it is unchanged under the restricted set of moves in Figure 2.11, VR1, 2, 3, in Figure 2.6, and FR4 in Figure 2.6.

The next problem is a major source of motivation for the rest of this thesis.

Problem 2.14. *Find powerful yet computable invariants of flat knots.*

Definition 2.15. For a flat knot diagram with n classical crossings, its *flat Gauss diagram* is a counterclockwise oriented skeleton with $2n$ points on the skeleton and n arrows. Every arrow encodes a crossing as shown in Figure 2.12. The order of the points on the skeleton tell us adjacency of the crossings in the flat knot diagram.

Definition 2.16. A *flat Gauss code* (or *flat Gauss word*) is a notation to represent the flat Gauss diagram. From 12 o'clock of the skeleton in the Gauss diagram, in

the order of counterclockwise, assign numbers $1, 2, \ldots$ to the arrows. At each arrow head (or tail), the point is recorded as "U" (or "O"), followed by this arrow's assigned number. Then from 12 o'clock of the skeleton, go counterclockwise and record every arrow head or tail along the skeleton.

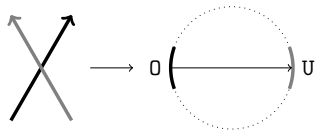


Figure 2.12: Flat crossings to Gauss diagram arrows

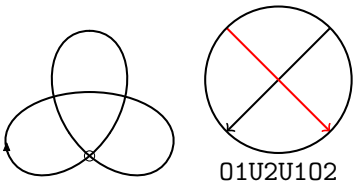


Figure 2.13: Flat virtual trefoil and its Gauss diagram

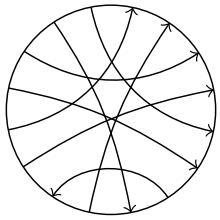
For example, the flat knot in Figure 1.2 has Gauss code O1O2O3U1U3U2. The OU *word* of a Gauss code is the binary string in $\{O, U\}$ obtained by removing the integers. For example, the flat knot in Figure 1.2 has OU word OOOUUU.

Remark 2.17. An OU word is *Lyndon* if it is minimal up to cyclic rotation. Here the ordering has $O < U$. The enumeration algorithm for flat knots [FKI] starts with Lyndon words and then considers matchings on them. One advantage is there is a linear time algorithm for generating Lyndon words due to Duval [Duv83].

For example, the OU word for the flat knot diagram in Figure 2.14 is Lyndon. The Gauss diagram also determines a *matching*, which is a permutation indicating how each "O" is matched to the corresponding "U". In fact, the Gauss diagram is completely determined by its OU word and matching.

To derive the matching, our convention is to first number the O's sequentially going counterclockwise from 12 o'clock in the diagram, and then to record the numbers of the corresponding U's, again going counterclockwise from 12 o'clock. Note that the ordering of crossings in the matching can differ from the ordering in the Gauss word. For example the flat knot in Figure 2.14 has Gauss word O1O2O3O4O5O6U7O8U2O7U4U1U6U3U8U5; in the matching the order of the last two crossings is switched.

Similar to virtual knot diagrams, flat knot diagrams are completely determined by the associated Gauss diagram, up to VR1, VR2, VR3 and FR4 and detour moves. On Gauss diagrams, the corresponding Reidemeister moves are as shown in Figure 2.15. Note that VR1, VR2, VR3, FR4 and detour moves do not change the flat Gauss diagram.



OU word: OOOOOOUOUOUUUUUU, Matching: [8 2 4 1 6 3 7 5]

Figure 2.14: Lyndon word representation of a flat knot

Remark 2.18. By [Pol10, Theorem 1.1], all the Reidemeister moves for flat Gauss diagrams are generated by FR1(a,b), FR2(a) and FR3(c,d).

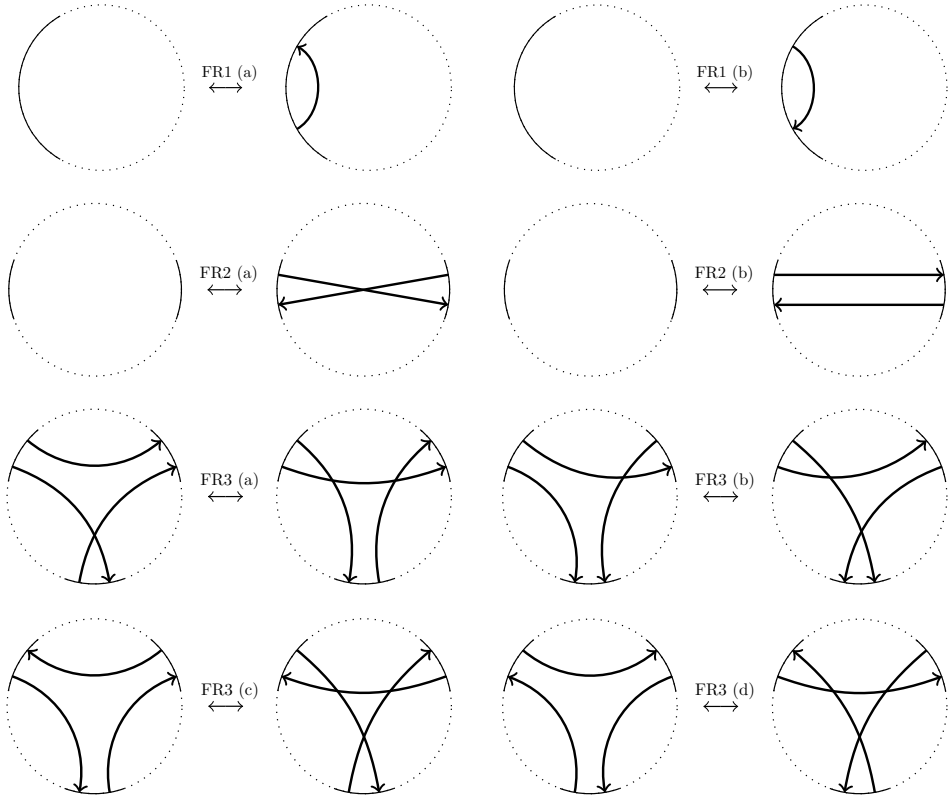


Figure 2.15: Flat Reidemeister moves on Gauss diagrams

Analogous to the embedding representations of virtual knots, flat knots can be represented as immersed curves in Carter surfaces.

Definition 2.19. An *immersion representation* (or a *diagram on surface* Σ_g) of a flat knot α is an immersion $\omega_\alpha : S^1 \looparrowright \Sigma_g$, where Σ_g is a connected, oriented, closed

surface of genus g. Two immersion representations are *stably equivalent* if they are related by a finite sequence of stabilizations, destabilizations, and homotopies. The *flat genus* $g(\alpha)$ of a flat knot α is defined to be the minimal genus over all surfaces admitting an immersion representation for α.

We can get a planar projection of Σ_g so that every double point is transverse and realize all the double points that are not in the immersion as virtual crossings. Thus we get a flat knot diagram. From a flat knot diagram, if we regard the virtual crossing as in Figure 2.16 (left), every flat crossing is realized by thickening the surface as in Figure 2.16 (right), and then attaching 2-disks to that surface along its boundaries to obtain a Carter surface.

Similarly, the "O,U" of the associated flat Gauss code is given by the sign of intersection at the double points. Then chose a starting point, one can travel along the loop and number each arrow by their order. Then the Gauss code is given by listing each arrow's assigned number and their "O,U" in the order of the trip along the loop.

As with virtual knots, there is a bijection between these three representations up to equivalence. We have already seen that there is a correspondence between flat knot diagrams and flat Gauss diagrams which is one-to-one on equivalence classes. When we construct an associated immersion representation from a flat knot diagram, the immersion is not changed under VR1, VR2, VR3, FR4 and detour moves, while FR1, FR2, FR3 give homotopy and/or stabilization of the Carter surface. The genus of the Carter surface can only change under FR2, and the two immersion representations are related by stabilization and hence equivalent. Note that the above equivalence also applies to classical knots, which form a subset of virtual knots.

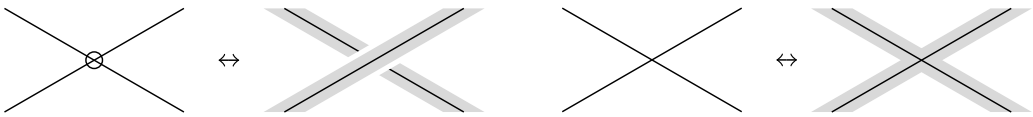


Figure 2.16: Flat knots as immersed loops on surfaces

Throughout this thesis, we will interchangeably use the three representations.

It is worth explaining the relation between classical, virtual and flat knots. Classical knots are a subset of virtual knots. At the level of diagrams, we know that virtual knot diagrams are defined based on classical knot diagrams with introducing a new type of crossing and additional equivalence relations. Two classical knot diagrams are equivalent as classical knots if and only if they are equivalent as virtual knot diagrams.

Definition 2.20. The *shadow map* or *flat projection* is the natural map π sending a virtual knot diagram D to its *shadow diagram*, which is the flat knot diagram obtained by replacing all classical crossings with flat crossings as in Figure 2.17-left.

The shadow map sends Reidemeister moves of virtual knot diagrams to the corresponding flat Reidemeister moves of their shadow diagrams. Therefore, it induces a well-defined map, also denoted π, from virtual knots to flat knots. The shadow map π is obviously a surjection, for any diagram of flat knot α, we can resolve each flat crossing positively, and this produces a virtual knot diagram D with $\pi(D) = \alpha$.

The shadow map π can be described at the level of Gauss diagrams as well. Given a Gauss diagram for a virtual knot, we switch the direction of all the negative arrows and then delete all the signs as shown in Figure 2.17-right. Again, the map is easily seen to be surjective, for a flat Gauss diagram D, we assign positive signs to all the arrows and obtain a Gauss diagram D' of a virtual knot with $\pi(D') = D$.

For geometric representations, from embedding $e_K : S^1 \to \Sigma_g \times I$, we can flatten the Carter surface and allow singular points to occur so that we obtain an immersion $\omega : S^1 \looparrowright \Sigma_g$, which is an immersion representation for the flat knot. For every immersion representation, we can thicken the Carter surface first and lift an arc at each singular point of the loop. Then we obtain an embedding in the thickened Carter surface.

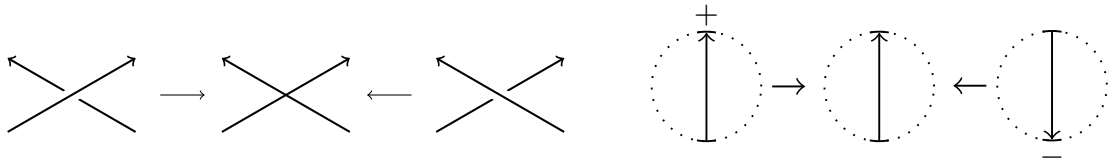


Figure 2.17: Flattening classical crossings and the effect on Gauss diagrams

The shadow map π also has a topological interpretation: a classical knot is an embedding of S^1 into S^3 or $S^2 \times I$, while a virtual knot is an embedding of S^1 into $\Sigma_g \times I$. Equivalently, a flat knot can be represented as an immersion of S^1 in Σ_g. Given a diagram of flat knot α realized as an immersion $\omega_\alpha : S^1 \looparrowright \Sigma_g$, then the immersion can be lifted to an embedding $S^1 \to \Sigma_g \times I$ representing a virtual knot which maps to α under the shadow map.

Note that all classical knots lie in the kernel of π for the simple reason that all classical knots can be unknotted by performing crossing changes. (This is also evident by considering the Δ-move, which is an unknotting operation for classical knots and which maps to $FR3$ under the shadow map.) If K is a virtual knot with genus $g(K) = 1$, then its image $\pi(K)$ under the shadow map is the trivial flat knot. Indeed, since the fundamental group of a torus is commutative, any immersed loop on a torus is homotopic to the shadow of a classical knot (more specifically, a torus knot) and hence trivial as a flat knot.

2.4 Symmetry type of flat knots

In this section, we discuss the symmetries of a flat knot generated by the two involutions.

Let α be a flat knot represented as an immersion $\omega_\alpha : S^1 \looparrowright \Sigma_g$.

1. The *reverse* $-\alpha$ is given by changing the orientation of S^1;
2. The *mirror image* α^* is given by changing the orientation of Σ_g;
3. The *reversed mirror image* $-\alpha^*$.

On flat Gauss diagrams, the operation $\alpha \to -\alpha$ reverses the orientation of the skeleton, whereas the operation $\alpha \to \alpha^*$ reverses all the arrows.

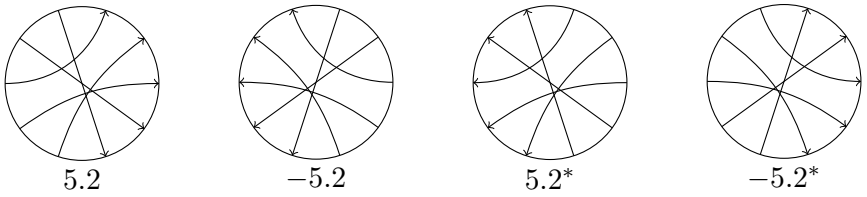


Figure 2.18: Gauss diagrams of 'siblings' of the flat knot 5.2

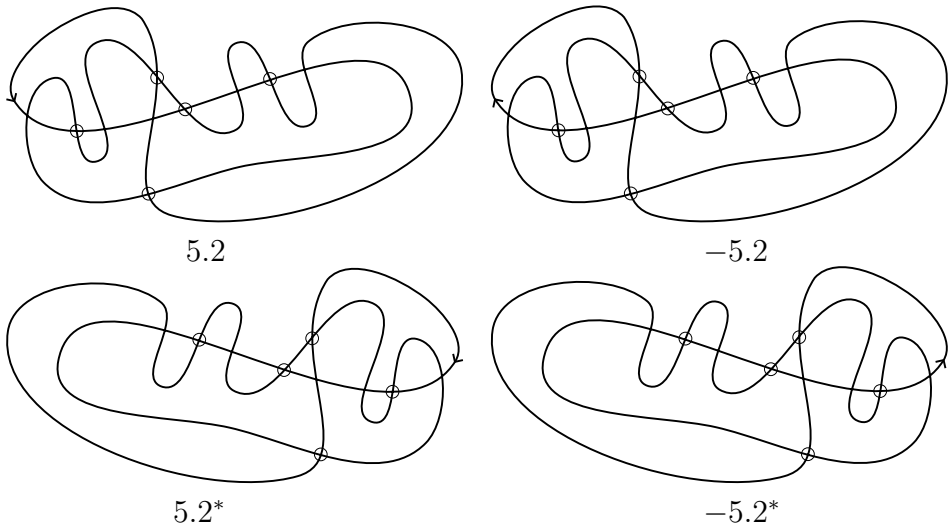


Figure 2.19: Diagrams of 'siblings' of the flat knot 5.2

Definition 2.21. The five symmetry types of flat knots are chiral, reversible, +-achiral, −-achiral and fully-achiral as defined in Table 2.1.

2.5 Arrow index and Alexander numberings

In this section, we introduce the arrow indices, which are integers associated to each crossing in a flat knot diagram. When a flat knot is represented as an immersed curve in a surface, the arrow index can be viewed as an intersection number. A flat

	chiral	reversible	+-achiral	−-achiral	fully-achiral
$\alpha = -\alpha$	No	Yes	No	No	Yes
$\alpha = \alpha^*$	No	No	Yes	No	Yes
$\alpha = -\alpha^*$	No	No	No	Yes	Yes

Table 2.1: Symmetry types of flat knots

knot diagram whose arrow indices are all trivial is said to be *Alexander numberable*, and a flat knot that can be represented by an Alexander numberable diagram is said to be *almost classical* (cf. [BG+17]).

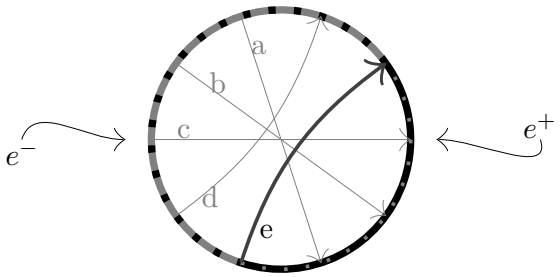


Figure 2.20: Calculation of the arrow index $n(e) = -3$

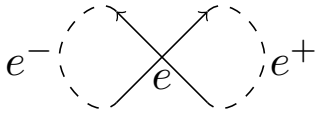


Figure 2.21: Loop associated to e

Let D be a flat Gauss diagram and $\mathrm{arr}(D)$ be its set of arrows. For an arrow $e \in \mathrm{arr}(D)$, let e^O denote its arrow tail and e^U denote its arrow head. Let e^+ denote open interval on the flat Gauss diagram from e^O to e^U and let e^- denote the open interval from e^U to e^O as shown in Figure 2.20.

Definition 2.22. In a flat Gauss diagram, the *index* $n(e)$ of an arrow e is given by the number of arrow tails in e^+ minus the number of arrow heads in e^+.

Now we can calculate the indices of all arrows in Figure 2.20:

$$n(a) = 4, \quad n(b) = 2, \quad n(c) = 0, \quad n(d) = -3, \quad n(e) = -3.$$

Note that given an immersion representation $\omega_\alpha : S^1 \looparrowright \Sigma_g$ of α, e^+ and e^- are associated to two loops intersecting non-transversally at a singular point corresponding to e. Referring to Figure 2.21, the loop going from the darker arc and back to the

lighter arc at the singular point is associated to e^+. Then one can take the homology classes $[e^+], [e^-]$ and consider their intersection number with $[\omega_\alpha(S^1)] \in H_1(\Sigma_g; \mathbb{Z})$.

Next, we define Alexander numberable flat knot diagrams and almost classical flat knots, as well as mod p versions of them.

Definition 2.23. A flat knot diagram D is said to be *Alexander numberable* if $n(e) = 0$ for every arrow e of D. A flat knot α is said to be *almost classical* (AC) if it admits an Alexander numberable diagram.

Definition 2.24. A flat knot diagram D is said to be *mod p Alexander numberable* if $n(e) \equiv 0 \mod p$ for every arrow e of D. A flat knot α is said to be *mod p almost classical* (mod p AC) if it admits a mod p Alexander numberable diagram. When $p = 2$, we say a flat knot is *checkerboard colorable* if it admits a mod 2 Alexander numberable diagram.

The index $n(e)$ is the intersection number of the 1-cycles in the Carter surface associated to the arrow e and the flat knot core element $s = [\omega_\alpha(S^1)]$. Refer to [Tur04, Section 4.2], $s \cup \{[e^+]\}_{e \in \mathrm{arr}(\alpha)}$ is a basis of $H_1(\Sigma; \mathbb{Z})$. Therefore, if $n(e) = 0$ for all $e \in \mathrm{arr}(\alpha)$, then $\omega_\alpha(S^1)$ bounds a surface in Σ. This gives the following equivalent definition of almost classical flat knots.

Definition 2.25 (Version 2 of Definition 2.23)**.** A flat knot α is called *almost classical* if it can be represented as the boundary of an oriented immersed surface in a Carter surface. This immersed surface is referred to as a *flat Seifert surface* for α.

Definition 2.26. If α is an almost classical flat knot, then the minimal genus over all flat Seifert surfaces for α is called the *flat Seifert genus* of α and denoted $g_{ac}(\alpha)$.

If we relax the assumption and require only that the index $n(e) \equiv 0 \mod 2$ for all arrows e, then the immersion bounds an unoriented surface.

Definition 2.27 (Version 2 of Definition 2.24 for $p = 2$)**.** A flat knot α is called *checkerboard colorable* or *mod 2 almost classical*, if it can be represented as a curve bounding an immersed unoriented surface in the Carter surface.

Definition 2.28. Given a flat knot α, its *r-th covering* is denoted $\alpha^{(r)}$ and defined to be the flat knot obtained from a Gauss diagram D of α after deleting arrows $\{e \in \mathrm{arr}(D) \mid n(e) \notin r\mathbb{Z}\}$.

The *r-th covering* $\alpha^{(r)}$ of a flat knot was introduced in [Tur04, Tur08b], where it was shown that the r-th covering is a flat knot invariant representing the lift of ω_α to the r-fold covering of Σ induced by the dual in $H^1(\Sigma; \mathbb{Z}/r)$ of $[\omega_\alpha(S^1)]$.

Manturov gave a purely combinatorial description of the r-th covering of a flat knot in terms of Gaussian parity projection in [Man10]. A detailed explanation of the correspondence between lifts to abelian covers and Gaussian parity projection can be found in [BCG20, Section 5.3].

2.6 Cabling of flat knots

In this section, we introduce the cabling operation for flat knots and show that it is well-defined.

Recall that any flat knot diagram is locally smooth. Therefore, given a flat knot diagram, we obtain a *parallel copy* of the knot by pushing it off itself in the normal direction.

Definition 2.29. The *n-strand cable* α^n of flat knot α is a link consisting of n parallel copies of α as shown in Figure 2.23.

The n-strand cable is well-defined because two homotopic diagrams will have homotopic n-strand cables. For example, the 2-strand cable homotopy in Figure 2.22 are all generated by flat Reidemeister moves. That is to say, flat Reidemeister moves are commutative with cabling.

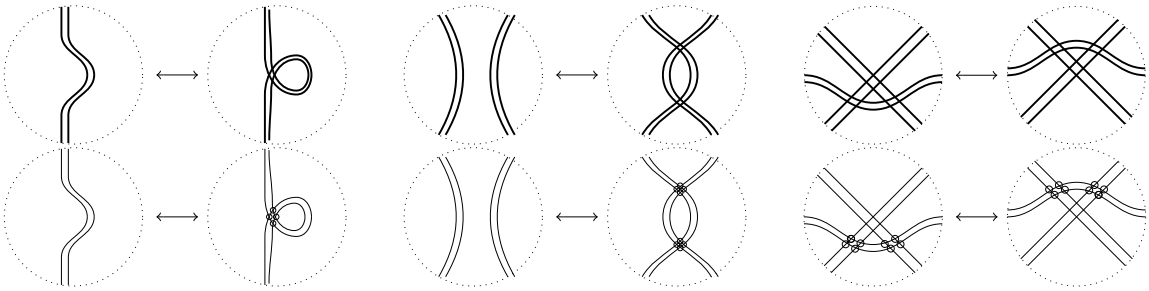


Figure 2.22: Cabled Reidemeister moves are generated by flat Reidemeister moves.

Definition 2.30. The *$(2,1)$-cable* $\alpha^{2,1}$ of a flat knot α is the flat knot obtained by adding a crossing between the components in α^2 as shown in Figure 2.23. The *$(n,1)$-cable* $\alpha^{n,1}$ of a flat knot α is the flat knot is obtained by applying the cyclic permutation $(2,3,\ldots,n,1)$ to n parallel arcs in α^n.

The $(n,1)$-cable is well-defined because the crossings in the permutation $(2,3,\ldots,n,1)$ can pass n-parallel virtual or flat crossings in the flat knot diagram; see the example for $n=3$ in Figure 2.24. If α,β are equivalent flat knot diagrams, then α^n is equivalent to β^n, and $\alpha^{(n,1)}$ is equivalent to $\beta^{(n,1)}$.

2.7 Concordance of flat knots

In this section, we define concordance of flat knots, following Turaev [Tur04].

Definition 2.31. Two flat knots α and β are said to be *concordant* if there exist

(i) immersion representations $\omega_\alpha : S^1 \looparrowright \Sigma$ and $\omega_\beta : S^1 \looparrowright \Sigma'$ for α and β, respectively,

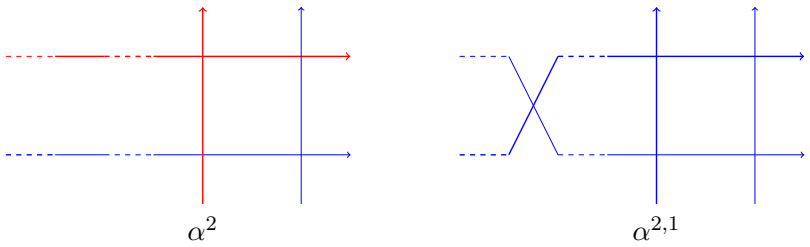


Figure 2.23: The diagrams of the cables α^2 and $\alpha^{2,1}$

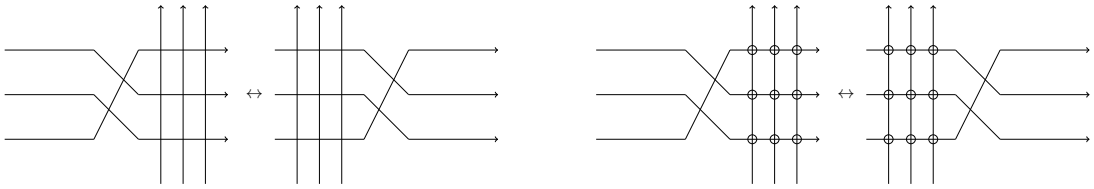


Figure 2.24: The $(n, 1)$-cable is well-defined.

(ii) a compact, connected oriented 3-manifold M with boundary $\partial M = -\Sigma \sqcup \Sigma'$, and

(iii) a properly immersed annulus $C \looparrowright M$ whose boundary is an immersion representation for $-\alpha^* \sqcup \beta$.

The definition of sliceness for flat knots is similar to the definition for classical knots.

Definition 2.32. Let α be a flat knot represented by an immersion $\omega_\alpha : S^1 \looparrowright \Sigma$. Then α is said to be *slice* if there exists a compact oriented 3-manifold M with $\partial M = \Sigma$ and a properly immersed disk $D \looparrowright M$ whose boundary is $\omega_\alpha(S^1)$.

Equivalently, a flat knot α is slice if it is concordant to the trivial flat knot. In Definition 2.31, α is slice if Σ' is S^2 or T^2. Alternatively, a flat knot α is slice if a diagram of α can be transformed to the trivial diagram by a sequence of flat Reidemeister moves, births, deaths and saddle moves shown in Figure 2.25. We further require that the number of saddles is equal to the sum of births and deaths, and that the cobordism surface is connected. Such a sequence is called a *slice movie*. The definition can also be applied to Gauss diagrams. For example, Figure 2.26 shows how one saddle move and Reidemeister moves transform the knot 7.45422 to a trivial flat knot. The first example of a non-slice flat knot was discovered by Carter in [Car91].

We also define the slice genus of a flat knot.

Definition 2.33. Let α be a flat knot represented by an immersion $\omega_\alpha : S^1 \looparrowright \Sigma$. Then α is said to have *slice genus* $g_s(\alpha)$ if there exists a compact oriented 3-manifold

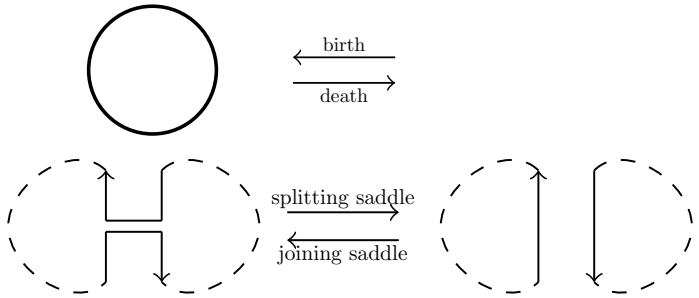


Figure 2.25: Birth, death, and saddle moves

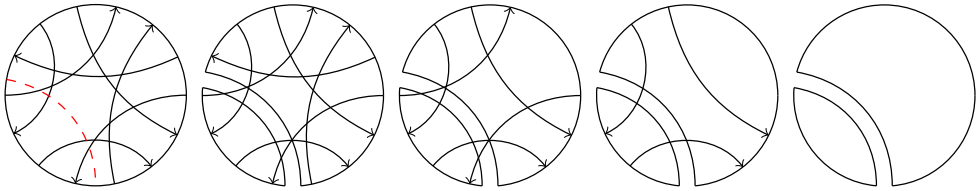


Figure 2.26: A slice movie for the flat knot 7.45422

M with $\partial M = \Sigma$ and a properly immersed oriented surface $F \looparrowright M$ with genus $g_s(\alpha)$ and whose boundary is $\omega_\alpha(S^1)$, and if F has minimal genus among all such surfaces in 3-manifolds M with boundary $\partial F = \omega_\alpha(S^1)$.

A flat knot α is slice if and only if it has slice genus $g_s(\alpha) = 0$.

Note that $g_s(\alpha) < \infty$ for any flat knot α. In fact, we claim that $g_s(\alpha) \leq \frac{1}{2}cr(\alpha)$. To see this, consider the cobordism surface constructed by applying a saddle move at each crossing. Each saddle reduces the number of crossings, and the end result is a knot or link diagram without any crossings. Each component can be capped off with a 2-disk, and the movie describes a cobordism surface bounding the knot. In any sequence of saddle moves, each one is either a splitting saddle or joining saddle. Computing the Euler characteristic of the cobordism surface, one sees that its genus is exactly equal to the number of joining saddles. On the other hand, since no more than half of the saddle moves can be joining saddles, this implies that $g_s(\alpha) \leq \frac{1}{2}cr(\alpha)$. (This last step uses the fact that we started with a flat knot diagram.)

A classical or virtual knot is said to be *ribbon* if it admits a slice movie with only splitting saddles and deaths, no births or joining saddles. Clearly, any classical or virtual knot that is ribbon is also slice. The slice-ribbon conjecture asserts the converse is true, namely that every slice knot is ribbon.

Definition 2.34. A flat knot is said to be *ribbon* if it admits a slice movie with only saddles and deaths.

Clearly, any flat knot that is ribbon is necessarily slice. The following question is an interesting analogue of the slice-ribbon conjecture for flat knots.

Problem 2.35. *Is every flat knot that is slice also ribbon?*

A different notion of ribbon is defined in [Tur04]. Given a flat knot α, we say α is *strongly ribbon* if it admits a Gauss diagram D with $D = -D^*$. In [Tur04], Turaev proved that if α is strongly ribbon, then it is slice. In [SW06], Silver and Williams gave an example of a long flat knot that is slice but not strongly ribbon. It is not difficult to see that the example in [SW06] is ribbon. In general, strongly ribbon implies ribbon, and this can be proved using a nested sequence of saddle moves similar to the slice movie in Figure 6.3. By [SW06], not all ribbon flat knots are strongly ribbon.

2.8 Composite knots and crossing number

In this section, we define connected sum for flat knots and consider the crossing number for composite flat knots. We will show that every minimal crossing diagram of a composite flat knot is a connected sum diagram.

We assume all diagrams in this section are oriented. Given two flat knot diagrams D_1, D_2 and basepoints p_1, p_2 on them, we can form the connected sum. The result is a composite flat knot denoted $D_1^\bullet \# D_2^\bullet$, and it depends on the diagrams and basepoints, cf. Figure 2.27. In particular, the operation of connected sum is not well-defined for virtual or flat knots. (However, as we will see in Chapter 6, connected sum is well-defined for *long* virtual and *long* flat knots.) This example also shows that the decomposition theorem, a theorem stating every flat knot can be uniquely decomposed as a connected sum of nontrivial prime flat knots, does not exist for flat knots.

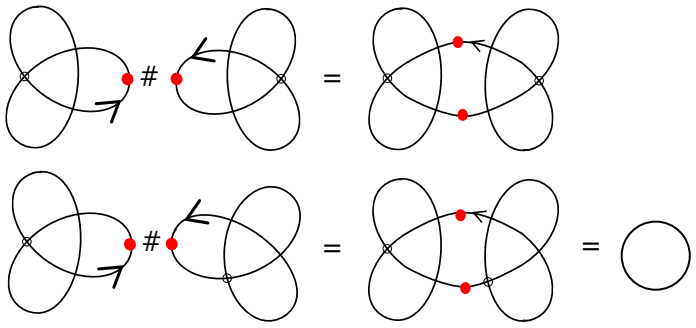


Figure 2.27: Connected sum of flat knots depends on basepoints

Definition 2.36. Let D_1, D_2 be flat knot Gauss diagrams. The *set of the permutant diagrams* is defined to be the set of the connected sum Gauss diagrams

$$[D_1 \# D_2] = \{D_1^\bullet \# D_2^\bullet \mid D_i^\bullet \text{ is the Gauss diagram } D_i \text{ with a choice of basepoint on it}\}.$$

Flat knots obtained as connected sums are called composite flat knots, and a flat knot is prime if it is not composite.

Note that the connected sum of two trivial flat knot diagrams can produce a nontrivial flat knot. For example, the flat knots in Figure 2.28 are nontrivial connected sums of two and three trivial diagrams, respectively. Similarly, one can construct a nontrivial flat knot as the connected sum of n trivial diagrams. These flat knots can be shown to be different by computing their based matrices and associated characteristic polynomials. (These invariants are introduced in the next chapter.)

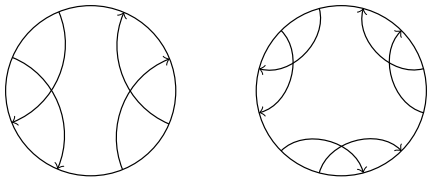


Figure 2.28: Flat knots 4.10 and 6.2064

Proposition 2.37. *Let D_1 and D_2 be two minimal crossing flat Gauss diagrams. Then any permutant diagram $D \in [D_1 \# D_2]$ is a minimal crossing number diagram of its flat knot type.*

Proof. Obviously, $cr(D) = cr(D_1) + cr(D_2)$. For the sake of contradiction, suppose the crossing number of D can be reduced. By monotonicity, the process must involve finitely many FR3 moves in Figure 2.30 followed by a decreasing FR1, FR2 in Figure 2.29.

Suppose FR3 can be applied on three arrows, with one from D_1 and two from D_2. This can only happen in the versions a, b, c and d of FR3, in the left-to-right direction, as shown in Figure 2.30. However, such Gauss diagrams of D_1 or D_2 are not minimal crossing. For example, as shown in Figure 2.31, if the connected sum is formed at the red dashed line, then applying an FR1 on D_1 reduces the crossing number. The same argument applies to FR3(b,c,d) as well. Therefore, no FR3 moves can be applied so that the arrows involved are from both D_1, D_2. Since D_1, D_2 are minimal Gauss diagrams, the diagram obtained after FR3 moves should still be a connected sum diagram of minimal crossing Gauss diagrams.

Now consider the decreasing FR1,2 applied on a connected sum diagram. Any such move applied to D can be also applied to either D_1 or D_2. Then no FR1,2 can be applied to D to reduce its crossing number. □

Lemma 2.38. *Let D be a connected-sum diagram $D = D_1^\bullet \# D_2^\bullet$, where $D_1^\bullet$ and $D_2^\bullet$ are both nontrivial as long flat knots. (Removing the basepoints from $D_1^\bullet, D_2^\bullet$ produces long flat knot diagrams, see Section 6.1 for the precise definitions of long flat knots.) Among all flat Reidemeister moves, only increasing FR2-moves can change D to a non-connected-sum diagram.*

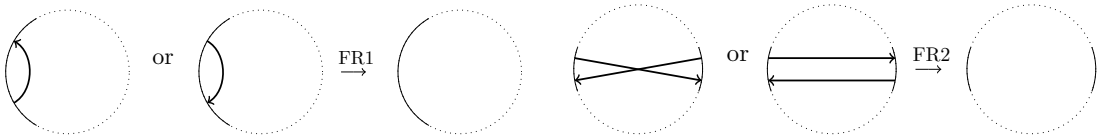


Figure 2.29: The decreasing FR1 and FR2 moves on Gauss diagrams

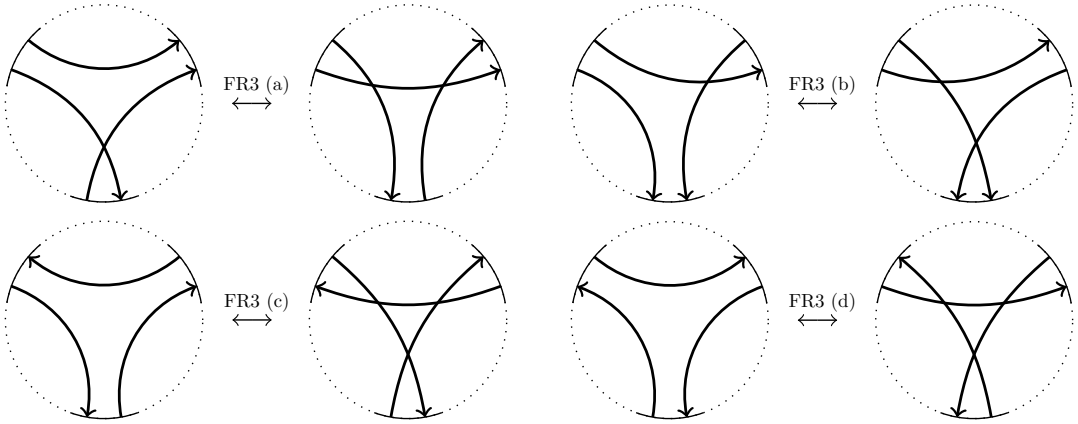


Figure 2.30: FR3 moves on Gauss diagrams

Proof. Observe that if a diagram on the left hand side of a decreasing FR1,2-move in Figure 2.29 is a connected sum, then so is the right hand side. Moreover, no FR3 move can transform a connected-sum diagram to a non-connected sum: as shown in Figure 2.31 if the connected sum is addressed at the red dashed line, then the blue dash-dotted line addresses the connected sum on the right hand side. Other FR3 moves follow similarly. □

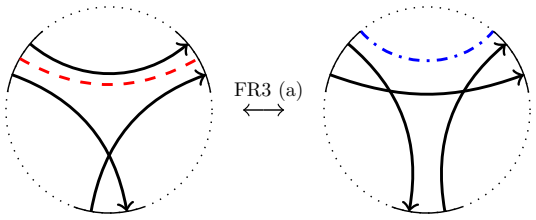


Figure 2.31: FR3(a) move on a Gauss diagram

Theorem 2.39. *Any minimal crossing diagram of a composite flat knot is a connected sum diagram.*

Proof. A composite round flat knot has a Gauss diagram $D = D_1^\bullet \# D_2^\bullet$, where $D_1^\bullet$ and $D_2^\bullet$ are two Gauss diagrams with a choice of basepoint on them, and removing the basepoint from either of them produces a nontrivial long flat knot. Assume D' is a minimal crossing diagram that is flat equivalent to D. By monotonicity, D' is

obtained from D by finitely many FR3-moves and decreasing FR1,2-moves. Then by Lemma 2.38, D' is a connected sum diagram. □

Then we can apply Theorem 2.39 to the tabulation process: if a minimal diagram of a flat knot α is not a connected sum, then α is a prime flat knot. Furthermore, it also implies that the flat knot crossing number is sub-additive for connected sums.

Corollary 2.40. *For any $D \in [D_1 \# D_2]$, let $\alpha, \alpha_1, \alpha_2$ be the flat knot types of D, D_1, D_2, respectively. Then the crossing number $cr(\alpha) \geq cr(\alpha_1) + cr(\alpha_2)$*

Proof. The inequality holds if $cr(\alpha_i) = 0$ for $i = 1$ or $i = 2$.

Assume α is a composite knot and $cr(\alpha_i) > 0$ for $i = 1, 2$. Then by Theorem 2.39, a minimal diagram D' of α is a connected sum Gauss diagram. Then $D' \in [D'_1 \# D'_2]$ for some Gauss diagrams D'_1, D'_2 of α_1, α_2, respectively. Then we have

$$cr(\alpha) = cr(D') = cr(D'_1) + cr(D'_2) \geq cr(\alpha_1) + cr(\alpha_2).$$

□

For example, the flat knot diagrams on the left of Figure 2.27 are equivalent to the flat unknot. However, their connected sum, shown on the right top, is a nontrivial flat knot with 4 crossings.

Chapter 3

Based matrices

In this chapter, we introduce based matrices, their equivalence relations and unique representations. We use them to define a notion of algebraic sliceness and give an example of a flat knot that is algebraically slice but not slice. We give formulas for the based matrix of a composite flat knot and a cabled flat knot, and we derive more invariants from based matrices and explore their properties.

3.1 Primitive based matrices

In this section, we define the based matrix associated to a flat knot diagram, as well as elementary reductions and primitive based matrices.

We begin with the abstract definition of a based matrix.

Definition 3.1. A *based matrix over an abelian group* H is a triple $T = (G, s, b)$, where G is a finite ordered set, $\bar{G} = \{s\} \sqcup G$ and $b : \bar{G} \times \bar{G} \to H$ is a skew-symmetric map.

Unless stated otherwise, we take $H = \mathbb{Z}$ here and consider based matrices over the integers.

Definition 3.2. Two based matrices $T = (G, s, b)$ and $T' = (G', s', b')$ are said to be *isomorphic* (written $T \sim T'$) if there is a bijection $i : \{s\} \sqcup G \to \{s'\} \sqcup G'$ with $i(s) = s'$ such that $b'(i(x), i(y)) = b(x, y)$ for all $x, y \in \{s\} \sqcup G$.

We now introduce algebraic analogues of the first and second Reidemeister moves for based matrices.

Definition 3.3. Let $T = (G, s, b)$ be a based matrix. Then we say:

- $x \in G$ is an *annihilating element* if $b(x, y) = 0$ for all $y \in G \sqcup \{s\}$.
- $x \in G$ is a *core element* if $b(x, y) = b(s, y)$ for all $y \in G$.
- $x, y \in G$ are *complementary elements* if $b(x, z) + b(y, z) = b(s, z)$ for all $z \in G \sqcup \{s\}$.

A based matrix $T = (G, s, b)$ is said to be *primitive* if G does not contain annihilating, core, or complementary elements.

An *elementary reduction* of the based matrix $T = (G, s, b)$ is the operation of removing from G an annihilating element, a core element, or a pair of complementary elements. The inverse operation is called an *elementary extension*.

Two based matrices are said to be *homologous* if one can be obtained from the other by a finite sequence of elementary extensions/reductions and isomorphisms.

Referring to [Tur04, Section 6.1], every skew-symmetric square matrix over an abelian group determines a based matrix. Every based matrix is obtained from a primitive based matrix by elementary extensions. Two primitive based matrices are isomorphic if and only if they are homologous.

For a based matrix T, we use $T_\bullet$ to denote the associated primitive based matrix obtained from T under elementary reduction.

Let α be a flat knot, realized both as a Gauss diagram D and as an immersion $\omega_\alpha : S^1 \to \Sigma_g$, where Σ_g is an oriented closed surface of genus g. Let $b : H_1(\Sigma_g, \mathbb{Z}) \times H_1(\Sigma_g, \mathbb{Z}) \to \mathbb{Z}$ be the skew-symmetric form given by the intersection pairing. Let $\mathrm{arr}(D)$ be the set of arrows in D and set $G = \{[e^+] \in H_1(\Sigma_g; \mathbb{Z}) \mid e \in \mathrm{arr}(D)\}$ as in Figure 2.20. We regard G as an ordered set, and define the core element to be $s = [\omega_\alpha(S^1)] \in H_1(\Sigma_g; \mathbb{Z})$.

Definition 3.4. The *based matrix* $T(D)$ associated to the triple (G, s, b) with $|G| = n$ is the $(n+1) \times (n+1)$ skew-symmetric matrix over $\mathbb{Z}$, where the i, j-entry of $T(D)$ is the intersection pairing of the i-th and j-th element of $\bar{G} = \{s\} \sqcup G$.

By convention, the row/column of the based matrix associated to the core element s is called the 0-th row/column, so that the last row/column is called the n-th row/column.

In [Tur04, Section 6.1], Turaev showed that the elementary reductions/extensions correspond to the flat Reidemeister moves on flat knot diagrams. Specifically, a decreasing FR1 move corresponds to deleting a core or annihilating element. A decreasing FR2 move corresponds to deleting a pair of complementary elements. Moreover, the primary based matrix is a flat knot invariant up to isomorphism: for any two diagrams D, D' of the same flat knot, we have $T_\bullet(D)$ is isomorphic to $T_\bullet(D')$.

Turaev in [Tur04, Lemma 4.2.1] gave an algorithm for calculating the based matrix $T(D)$ with respect to the Gauss diagram D:

Algorithm 3.5 (Calculation of based matrix)**.** The input is a Gauss diagram D of the flat knot α with n arrows, and the output is an integral skew-symmetric $(n+1) \times (n+1)$ matrix $T(D)$.

Fix an ordering of the arrow set $\mathrm{arr}(D)$ and use it to order G as in Definition 3.4. Since it is skew-symmetric, the diagonal entries $T(D)_{i,i} = 0$, and it is enough to specify the lower triangular entries $T(D)_{i,j}$ for $i > j$.

Let $s = [\omega_\alpha(S^1)]$ be the core element, and let $[e^+]$ and $[f^+]$ be the i-th and j-th elements of G, respectively.

Then the 0-th column of $T(D)$ has $T(D)_{0,0} = 0$, and for $i > 0$, we have

$$\begin{aligned} T(D)_{i,0} &= b([e^+], s) = n(e) \\ &= |\{f \in \operatorname{arr}(D) \setminus \{e\} \mid f^O \in e^+\}| - |\{f \in \operatorname{arr}(D) \setminus \{e\} \mid f^U \in e^+\}|. \end{aligned}$$

Further, we assume that $i > j > 0$. Then

$$T(D)_{i,j} = b([e^+], [f^+]) = |S_1| - |S_2| + \varepsilon,$$

where

$$S_1 = \{g \in \operatorname{arr}(D) \setminus \{e, f\} \mid g^O \in e^+ \text{ and } g^U \in f^+\},$$
$$S_2 = \{g \in \operatorname{arr}(D) \setminus \{e, f\} \mid g^U \in e^+ \text{ and } g^O \in f^+\}, \text{ and}$$
$$\varepsilon = \begin{cases} +1 & \text{if } f^O \in e^+ \text{ and } f^U \in e^-, \\ -1 & \text{if } f^U \in e^+ \text{ and } f^O \in e^-, \\ 0 & \text{otherwise.} \end{cases}$$

For example, consider the flat Gauss diagram in Figure 3.1 (this is the same diagram as Figure 2.20).

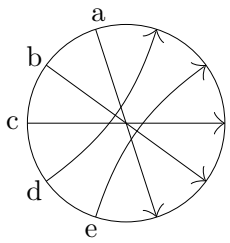


Figure 3.1: A diagram of flat knot 5.1

Then its based matrix with respect to $\bar{G} = \{s, [a^+], [b^+], [c^+], [d^+], [e^+]\}$ is

$$\begin{bmatrix} 0 & -4 & -2 & 0 & 3 & 3 \\ 4 & 0 & 1 & 2 & 4 & 3 \\ 2 & -1 & 0 & 1 & 3 & 2 \\ 0 & -2 & -1 & 0 & 2 & 1 \\ -3 & -4 & -3 & -2 & 0 & 0 \\ -3 & -3 & -2 & -1 & 0 & 0 \end{bmatrix}.$$

By choosing a different ordering of G, one will obtain a different based matrix. Therefore, the based matrix depends on the choice of diagram and ordering of G. To obtain invariants of flat knots that are independent of these choices, Turaev introduced the notions of based matrices, isomorphisms, and homology; see [Tur04].

3.2 Unique representation of based matrices

In this section, we introduce the ϕ-invariants, which were originally defined by Gibson [Gib08]. They record the information in a primitive based matrix.

Given a flat knot diagram D with based matrix $T(D)$, Turaev proved that one can always obtain a primitive based matrix after making finitely many elementary reductions to $T_\bullet(D)$ [Tur04]. He also showed that if D and D' are homotopic flat knot diagrams, then their primitive based matrices $T_\bullet(D)$ and $T_\bullet(D')$ are isomorphic. Thus, the isomorphism class of a primitive based matrix is an invariant of the flat knot up to homotopy.

Note that in Figure 2.15, FR1(a) and FR1(b) generates or removes a core element and an annihilating element, respectively, while FR2(a,b) adds or subtracts a pair of complementary elements. FR3(a-d) does not change the based matrices as long as the ordering of G is not changed. Thus, unless α is a minimal crossing Gauss diagram, its based matrix is not primitive. However, the converse is not true. The based matrix of a minimal crossing Gauss diagram need not be primitive.

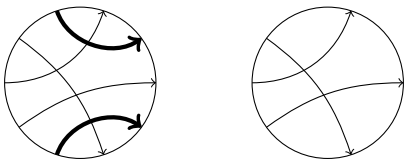


Figure 3.2: Flat knots 5.20 and 3.1

For example, the Gauss diagram of the flat knot 5.20 in Figure 3.2 has based matrix

$$T = \begin{bmatrix} 0 & -1 & -3 & 2 & 1 & 1 \\ 1 & 0 & -2 & 2 & 1 & 1 \\ 3 & 2 & 0 & 3 & 2 & 1 \\ -2 & -2 & -3 & 0 & 0 & 0 \\ -1 & -1 & -2 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 \end{bmatrix}.$$

The 1st and 5th elements in G are a pair of complementary elements. Removing them results in the primitive matrix

$$T_\bullet = \begin{bmatrix} 0 & -3 & 1 & 2 \\ 3 & 0 & 2 & 3 \\ -1 & -2 & 0 & 0 \\ -2 & -3 & 0 & 0 \end{bmatrix}.$$

Notice that $T_\bullet$ cannot be obtained directly from a flat Gauss diagram: The only 3-crossing nontrivial flat knot is 3.1 and its siblings, and none of the associated based

matrices are isomorphic to $T_\bullet$. Moreover, the entries of a based matrix obtained directly from a flat Gauss diagram are strictly less than the crossing number. Thus, not every integral skew-symmetric matrix is the based matrix obtained directly from a flat Gauss diagram. We wonder which integral $(n+1)\times(n+1)$ skew-symmetric matrices can be realized as based matrices obtained directly from flat Gauss diagrams (based matrices obtained by only Algorithm 3.5 and no elementary extension or reduction). More generally, we look for conditions on integral $(n+1)\times(n+1)$ skew-symmetric matrices to be either a primitive based matrix or a based matrix obtained directly from a flat Gauss diagram.

If α is a flat knot with siblings, $-\alpha, \alpha^*, -\alpha^*$, Turaev gave an algebraic method for computing the primitive based matrices of $-\alpha, \alpha^*, -\alpha^*$ in terms of the primitive based matrix of α.

Theorem 3.6 ([Tur04])**.** *Let α be a flat knot with primitive based matrix $T_\bullet$. Then the primitive based matrices of $-\alpha, \alpha^*$ are isomorphic to $(T_\bullet)', -(T_\bullet)'$ respectively, where $(T_\bullet)'_{0,i} = -(T_\bullet)_{0,i}$ and $(T_\bullet)'_{i,j} = (T_\bullet)_{i,j} - (T_\bullet)_{i,0} - (T_\bullet)_{0,j}$ for $i,j \geq 1$.*

Primitive based matrices are only invariants up to isomorphism, and consequently they are difficult to apply. Gibson introduced an integer vector ϕ that records the entries in a primitive based matrix below the diagonal in a canonical way. Note that based matrices are skew-symmetric so the diagonal entries are all zeros. Let D be a flat knot diagram, $T_\bullet(D)$ be a primitive based matrix of α. Define $\varphi(T_\bullet(D))$ to be the entries of the columns below the diagonal of $T_\bullet(D)$.

For example, if

$$T_\bullet(D) = \begin{bmatrix} 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 & -2 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 1 & 2 & 0 & -1 & 0 \end{bmatrix},$$

then $\varphi(T_\bullet(D)) = [-1,0,0,1,-1,0,2,-1,0,-1]$.

Let $\phi_\alpha = \min\{\varphi(T_\bullet) \mid T_\bullet \text{ is a primitive based matrix of a diagram of } \alpha\}$, where the minimum is taken in lexicographic order. For example, $[1,2,3] < [2,2,3] < [2,3,3]$.

Next, we give an algorithm for determining the ϕ-invariant ϕ_α of a flat knot α.

Algorithm 3.7 (Calculation of ϕ_α.)**.** Let $T_\bullet$ be a primitive based matrix of a flat knot α with respect to $\bar{G}$. There is a canonical way to partition G into subsets: Consider the index function (see Definition 2.22) $n : G \to \mathbb{Z}$ with image $\{n_1, \ldots, n_k\} \subset \mathbb{Z}$, where $n_i < n_j$ if $i < j$. Set $P_i = \{x \in G \mid n(x) = n_i\}$. Then each $P_i \neq \varnothing$ and $P_1 \cup \cdots \cup P_k$ determines a partition of G. The ϕ_α can be obtained by applying permutations on each P_i.

Assume $m_i = |P_i|$. Permutations of G preserving the partition $\{P_1, \ldots, P_k\}_{1\leq i\leq k}$ form a subgroup isomorphic to $\mathrm{Sym}(m_1)\times\cdots\times\mathrm{Sym}(m_k)$, with order $\sum_{i=1}^k m_i$. A

permutation $\sigma \in \mathrm{Sym}(m_i)$ acts on $P_i = \left\{x_1^i, x_2^i, \ldots, x_{m_i}^i\right\}$ to give the ordered set $P_i^\sigma = \left\{x_{\sigma(i)}^i, x_{\sigma(2)}^i, \ldots, x_{\sigma(m_i)}^i\right\}$. The algorithm we use will consider the restrictions on $\mathrm{Sym}(m_1) \times \cdots \times \mathrm{Sym}(m_k)$ and narrow it down to a subset. The complexity of the process lies in the ambiguity after ordering by indices; it can be measured by the number of P_i's where $|P_i| > 1$. The simplest case (complexity 0) is when $|P_i| = 1$ for all $1 \leq i \leq k$. The next simplest case is complexity 1.

Set $H_i = \mathrm{Sym}(m_i)$ for $1 \leq i \leq k$.

Let H_1' be the subset of H_1 that minimizes

$$\left\{[b(x_{\sigma(2)}^1, x_{\sigma(1)}^1), b(x_{\sigma(3)}^1, x_{\sigma(1)}^1), \ldots, b(x_{\sigma(m_1)}^1, x_{\sigma(1)}^1)]\right\}_{\sigma \in H_1}$$

in lexicographic order. Then let $H_1 = H_1'$.

If $|H_1| > 1$, then let $p_j := \{j\text{-th element of } P_1^\sigma \mid \sigma \in H_1\}$ for $j = 1, \ldots, m_1$ and repeat the step below until $|H_1| = 1$ or to finish.

For $i = 2, \ldots, k$, let H_i', p_j' be the subset of H_i, p_j respectively, such that it minimizes

$$\left\{[b(x_{\sigma(1)}^i, y), b(x_{\sigma(2)}^i, y), \ldots, b(x_{\sigma(m_i)}^i, y)]\right\}_{\sigma \in H_i, y \in p_j}$$

in lexicographic order. Then let $p_j = p_j', H_i = H_i', H_1 = \{\sigma \in H_1 \mid j\text{-th element of } P_1^\sigma \in p_j\}$.

In reality, the ambiguity is often eliminated after the first few steps. If we stop at this step and if $|H_i| \neq 1$ for some i, we can simply calculate all the ϕ's and choose the smallest one.

To finish the algorithm for a more general case, replace H_1 by H_i and then repeat for $1 \leq i \leq k$ until $|H_i| = 1$. If at the final step $|H_i| \neq 1$ for some i, then it means that there is more than one way to reorder G to reach the same minimum. This occurs, for instance, for periodic flat knot diagrams. For that case, one might as well consider treating periodic cases separately to enhance the efficiency.

We give an example which calculates the ϕ-invariant using this algorithm.

$$T_\bullet = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & -2 \\ -1 & 0 & 1 & 0 & 2 & 0 \\ -1 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 1 \\ 0 & -2 & 0 & -1 & 0 & 0 \\ 2 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

has $|P_1| = |P_2| = 2$. Assume the generators associated with $T_\bullet$ are s, x, y, z, w, v.

Then at the first step we have $b(y, x) = -1$ and $b(x, y) = 1$ so $p_1 = \{x\}$ and $|H_1| = 1$.

For the second step, $P_2^{\mathrm{Sym}(2)} = \{(z, w), (w, z)\}$. We calculate $[b(z, x), b(w, x)] = [0, -2]$ and $[b(z, x), b(z, x)] = [-2, 0]$. Therefore, we obtain $H_1 = \{(x, y)\}$. $H_2 =$

$\{(w, z)\}$. Then the based matrix is

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & -2 \\ -1 & 0 & 1 & 2 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 0 \\ 0 & -2 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 2 & 0 & 0 & -1 & 0 & 0 \end{bmatrix},$$

and $\phi_\alpha = [-1, -1, 0, 0, 2, -1, -2, 0, 0, 0, -1, 1, 0]$.

The based matrix is a very powerful invariant. It can be used to separate symmetric siblings of a flat knot in many cases.

For example, using the flat Gauss diagrams of the flat knot 5.2, we obtain four different ϕ-invariants. in Figure 2.18. They are

$$\begin{aligned} \phi_\alpha &= [-3, -2, -1, 2, 4, 1, 1, 2, 3, 1, 3, 4, 1, 2, 1], \\ \phi_{-\alpha} &= [-4, -2, 1, 2, 3, 1, 3, 2, 4, 2, 1, 3, 0, 1, 0], \\ \phi_{\alpha^*} &= [-3, -2, -1, 2, 4, 0, 1, 3, 4, 0, 1, 2, 2, 3, 1], \\ \phi_{-\alpha^*} &= [-4, -2, 1, 2, 3, 1, 2, 4, 3, 1, 3, 2, 1, 1, 1]. \end{aligned}$$

Therefore, the flat knot 5.2 and its siblings are all distinct.

Using the ϕ-invariant alone, we can distinguish flat knots up to 4 crossings as shown in Table 3.1.

Crossings	# Flat knots	# Non-distinguished by ϕ
3	1	0
4	11	0
5	120	8
6	2086	74
7	46233	1375

Table 3.1: Distinguishing flat knots using ϕ

The first non-distinguished flat knots are the two pairs $\{5.47, 5.65\}$ and $\{5.89, 5.104\}$ of 5-crossing flat knots in Figure 3.3. Their minimal ϕ-invariants, up to symmetry, are given by

$$\phi_{5.47} = \phi_{5.65} = [-2, -1, 0, 1, 2, -1, 1, 1, 3, 1, 0, 1, 0, 1, 0],$$

$$\phi_{5.89} = \phi_{5.104} = [-1, -1, 0, 1, 1, -1, 0, 1, 1, 0, 1, 1, 1, 1, -1].$$

Up to 5 crossings, every flat knot has nontrivial primitive based matrix. The first examples of flat knots with trivial primitive based matrix occur among the 6-crossing flat knots, namely 6.129 and 6.899 in Figure 3.4. These flat knots cannot be distinguished from the unknot by the ϕ-invariant.

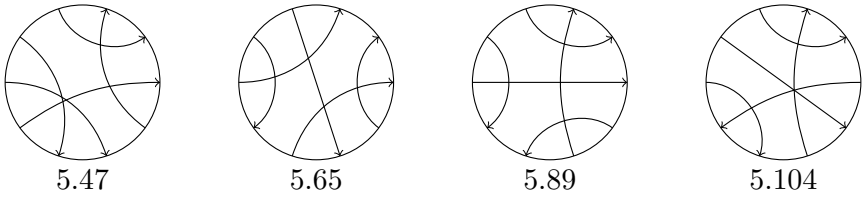


Figure 3.3: Two pairs of non-distinguished flat knots with 5 crossings

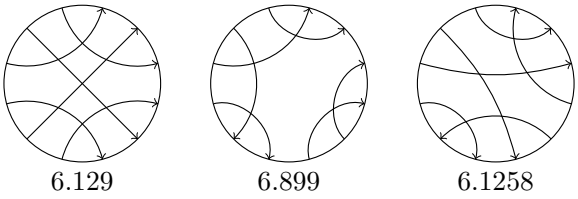


Figure 3.4: Three 6-crossing flat knots with trivial primitive based matrix

Note that in Table 3.1, the invariants of the flat knots with n crossings are compared to those of all flat knots with n or fewer crossings. Specifically, there are three 6-crossing flat knots with trivial ϕ-invariant, and 19 7-crossing flat knots with trivial ϕ-invariant. In particular, these flat knots cannot by distinguished from the unknot using only ϕ-invariants.

3.3 The u-polynomial

In this section, we discuss the u-polynomial of flat knots, which was first introduced by Turaev [Tur04]. In that paper, he gives necessary and sufficient conditions for a polynomial $u \in \mathbb{Z}[t]$ to be the u-polynomial of a flat knot. He also proves that the u-polynomial is additive under connected sum and invariant under concordance of flat knots.

In the previous section, we showed that the 0-th column of the based matrix of Gauss diagram D consists of indices $n(e)$ of all the arrows e in $arr(D)$; see also Definition 2.22.

Definition 3.8 ([Tur04])**.** Let α be a flat knot with Gauss diagram D. Then the u-polynomial of α is defined as

$$u_\alpha(t) = \sum_{e \in \mathrm{arr}(D)} \mathrm{sign}(n(e)) t^{|n(e)|}.$$

To see that the u-polynomial is an invariant of flat knots, notice that it does not change under the flat Reidemeister moves. The reason is that the indices do not change under FR3 in Figure 2.15; FR1 introduces a new arrow of index zero, while

FR2 introduces a pair of arrows with opposite indices. Therefore, we have that the flat Reidemeister moves on Gauss diagrams do not change the u-polynomial. One can define the u-polynomial of a based matrix.

Definition 3.9 ([Tur04])**.** The u-polynomial of an $(n+1)\times(n+1)$ based matrix T is defined as

$$u_T(t) = \sum_{i=1}^{n} \text{sign}(T_{i,0}) t^{|T_{i,0}|}.$$

The u-polynomial is an invariant of the homology class of based matrices: the elementary reductions, extensions and isomorphism do not change the u-polynomial; see [Tur04].

Referring to the flat knot in Figure 2.20, we have calculated its arrow indices, which are $n(a) = 4, n(b) = 2, n(c) = 0, n(d) = -3, n(e) = -3$. Therefore, this flat knot as u-polynomial $u_\alpha(t) = t^4 - 2t^3 + t^2$.

In the next few sections, we will use the u-polynomial as a tool to study sliceness and cabling of flat knots.

3.4 Characteristic polynomials

In this section, we introduce two new polynomial invariants of flat knots: the inner and outer characteristic polynomials of the primitive based matrix. The motivation for defining the characteristic polynomials comes from the complexity of calculating the ϕ-invariant. The characteristic polynomials are weaker invariants but also simpler to compute. As well, they are somewhat analogous to the Alexander polynomial, which for fibered knots can be interpreted as the characteristic polynomial of the monodromy.

Definition 3.10. Let T be a primitive based matrix and set $P_T(t) = \det(T - tI)$, the characteristic polynomial of T. Further, let $\widehat{T}$ be the matrix obtained from T by deleting its first row and column, and set $p_T(t) = \det(\widehat{T} - tI)$, the characteristic polynomial of $\widehat{T}$. Then $p_T(t)$ and $P_T(t)$ are called the inner and outer characteristic polynomials of T, respectively.

The next result shows that inner and outer characteristic polynomials are invariant under isomorphism of primitive based matrices.

Proposition 3.11. *Let T and T' be primitive based matrices. If T and T' are isomorphic, then $P_T(t) = P_{T'}(t)$ and $p_T(t) = p_{T'}(t)$.*

Proof. Recall that isomorphism of primitive based matrices is defined as a congruence by permutation matrices. But two matrices that are congruent by permutation matrices are necessarily conjugate. Since the characteristic polynomial is invariant

under conjugation, it follows that $P_T(t) = P_{T'}(t)$. Note that the permutation sends the first element of the ordered set $\bar{G}$ to the first element of $\bar{G}'$. Therefore, a similar argument shows that $\widehat{T}$ and $\widehat{T}'$ are conjugate, and it follows that $p_T(t) = p_{T'}(t)$. □

By [Tur04], any two primitive based matrices of a flat knot are isomorphic. Therefore, if α is a flat knot with primitive based matrix T, Proposition 3.11 implies that the inner and outer characteristic polynomials of T give well-defined invariants by setting $p_\alpha(t) = p_T(t)$ and $P_\alpha(t) = P_T(t)$.

For example, let U be the flat unknot. It has primitive based matrix [0]. Thus its inner and outer polynomials are $p_U(t) = 1$ and $P_U(t) = t$.

Recall that a flat Gauss diagram D is almost classical if its indices $n(e) = 0$ for all $e \in \mathrm{arr}(D)$. Let $\mathscr{B}_{AC}$ be the set of based matrices whose first row and column have zero entries.

Proposition 3.12. *If $T \in \mathscr{B}_{AC}$, then $P_T(t) = t \cdot p_T(t)$.*

Proof. Let $\widehat{T}$ be the matrix obtained from T removing its first row and column. For $T \in \mathscr{B}_{AC}$, since the first row and column are zero vectors, we have $P_T(t) = \det(T - tI) = t \cdot \det(\widehat{T} - tI) = t \cdot p_T(t)$. □

Remark 3.13. The inner and outer polynomials provide a concise way to encode the information from a primitive based matrix up to permutation. For example, the based matrix obtained from diagram of 4.1 (Figure 4.10) is

$$\begin{bmatrix} 0 & -3 & -1 & 2 & 2 \\ 3 & 0 & 1 & 3 & 2 \\ 1 & -1 & 0 & 2 & 1 \\ -2 & -3 & -2 & 0 & 0 \\ -2 & -2 & -1 & 0 & 0 \end{bmatrix}.$$

This is a primitive matrix. Then its inner characteristic polynomial is

$$p_{4.1}(t) = t^4 + 19t^2 + 1,$$

while its outer characteristic polynomial is

$$P_{4.1}(t) = t^5 + 37t^3 + 11t.$$

There exists no other flat knot up to 7 crossings sharing the same outer characteristic polynomial [FKI].

Problem 3.14. *What polynomials can be realized as inner and outer characteristic polynomials?*

Problem 3.15. *Is there a nice geometric interpretation of the inner and outer characteristic polynomials?*

Problem 3.16. *Can one use the characteristic polynomials to detect sliceness of flat knots?*

3.5 Algebraic sliceness does not imply sliceness

In this section, we recall the genus of a based matrix and use it to define a notion of algebraic sliceness for flat knots, following Turaev [Tur04]. We present an example of a flat knot that is algebraically slice but not slice.

Definition 3.17. Given a based matrix T with respect to the triple (G, s, b), a *filling* $\chi = \{X_i\}_{1\le i\le k}$ is a partition of G such that $G = \bigcup_{1\le i\le k} X_i$ where $X_i \cap X_j = \varnothing$ if $i \neq j$, and $|X_i| \in \{1, 2\}$ for any i. Let $y_i = \sum_{x\in X_i} x$ (as the formal sum in the free module $\mathbb{Z}G$) and $G' = \{y_i\}_{1\le i\le k}$. The intersection form b extends to G' by linearity.

Then we obtain a new triple (G', s, b) from this filling, and a new based matrix $\widehat{T}$ associated to it. The *genus of a based matrix* $\sigma(T)$ is defined to be $\frac{1}{2}\min \operatorname{rank}(\widehat{T})$ over all possible fillings, where the $\operatorname{rank}(\,\cdot\,)$ refers to the rank of the integral matrix. A based matrix T is said to be *null-concordant* if $\sigma(T) = 0$.

When the first row (or column) is the zero vector, we say the based matrix or the flat knot associated to it is *algebraically almost classical* or simply *algebraically AC*.

By [Tur04, Lemma 7.1.1], the genus of the based matrix gives an invariant of flat knots which we call its algebraic genus.

Definition 3.18. The algebraic genus of a flat knot is denoted $g_a(\alpha)$ and given by the genus $\sigma(T)$ of its based matrix (not necessarily primitive).

A flat knot α is *algebraically slice* if $g_a(\alpha) = 0$, namely if its based matrix is null-concordant.

Thus we can visualize the fillings on a Gauss diagram since the generator set of a based matrix corresponds to the arrow set of the Gauss diagram. As shown in Figure 3.5, the based matrix of the diagram is

$$\begin{bmatrix} 0 & -1 & 1 & -2 & 0 & 2 \\ 1 & 0 & 1 & -2 & 0 & 2 \\ -1 & -1 & 0 & -2 & 0 & 2 \\ 2 & 2 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ -2 & -2 & -2 & -2 & -1 & 0 \end{bmatrix}$$

The last five rows correspond to the five arrows from 12 o'clock of the Gauss diagram in counterclockwise order. The filling $[(1, 2), (3, 5), (4)]$, denoted by dashed brown, black, and dotted blue, respectively, gives the algebraic genus zero. When a diagram has a slice movie consisting of only splitting saddles, deaths, FR3 and decreasing FR1 and FR2 moves, the FR1 and FR2 moves determine a filling with $\sigma(T) = 0$. For example, after applying a splitting saddle move to the diagram in Figure 3.5, the arrow pairs (1,2) and (3,5) can be removed by FR2 moves, and the 4th arrow can be removed by a FR1 move.

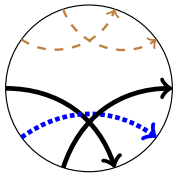


Figure 3.5: Fillings to show $g_a(5.21) = 0$

Every slice flat knot is algebraically slice, and it is natural to ask whether the converse is true. We will give a counterexample to show that it is not true. Recall the r-th covering $\alpha^{(r)}$ of flat knot α in Definition 2.28. In the proof of [Tur04, Corollary 5.17], Turaev showed that if α is slice, then its r-th covering $\alpha^{(r)}$ is also slice. We use this with $r = 3$ to give an example of a flat knot that is algebraically slice but not slice, see Figure 3.6.

Example 3.19. The flat knot 6.464 is algebraically slice but not slice.

Proof. Consider the flat knot 6.464 in Figure 3.6. Its based matrix can be shown to have genus 0 since there exists a filling as in Figure 3.7-left. Thus, the based matrix is null-concordant. Therefore 6.464 has $g_a(\alpha) = 0$ and is algebraically slice. On the other hand, its 3-fold covering is the flat knot -4.2. By our calculation [FKI], the based matrix of 4.2 has genus 1, all the corresponding fillings of minimal diagrams are listed in Figure 3.7-right. Thus 4.2 is not algebraically slice and not slice. Therefore, 6.464 is also not slice. □

An alternative argument to show that the flat knot 4.2 is not slice is to use the fact that its u-polynomial is $-t^3 + t^2 + t$ and to recall that any flat knot that is slice must have trivial u-polynomial. By [FKI], up to 6 crossings, 6.464 is the only known flat knot that is algebraically slice but not slice. The only other potential example is 6.540, which is algebraically slice but not known to be slice. In fact, 6.540 is the only flat knot up to 6 crossings whose slice state is unknown.

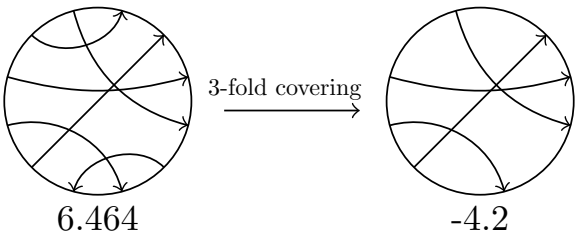


Figure 3.6: The 3-fold covering of the flat knot 6.464

A slice obstruction arising solely from based matrices is regarded as a *primary obstruction*. In [Tur04, Question 2], Turaev asks if one can detect non-slice flat knots using the *secondary obstructions*, see [Tur04, Section 8.4]. Note that the flat knot

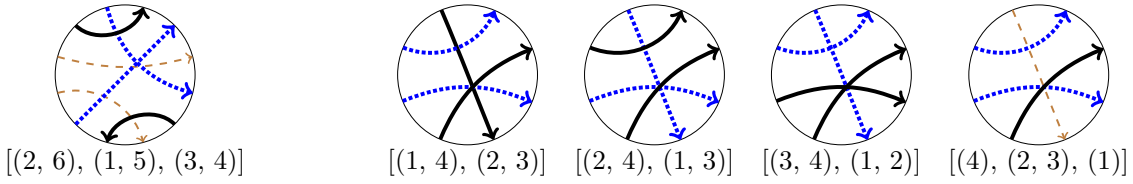


Figure 3.7: Fillings to show $g_a(6.464) = 0$ (left) and $g_a(4.2) = 1$ (right).

6.464 gives an example, and it is actually the first one in the tabulation of [FKI]. By [Tur04, Lemma 8.4.1], the arrows annihilated by the core element should form a slice flat knot. Therefore, the flat knot 6.464 is seen to be non-slice by the secondary obstructions.

There are additional examples of flat knots which are algebraically slice but not slice in Figure 3.8. Each is seen to be non-slice using parity projection, and one of them even has trivial primitive based matrix.

Example 3.20. The flat knot 7.25725 has trivial primitive based matrix, so it is algebraically slice but not slice.

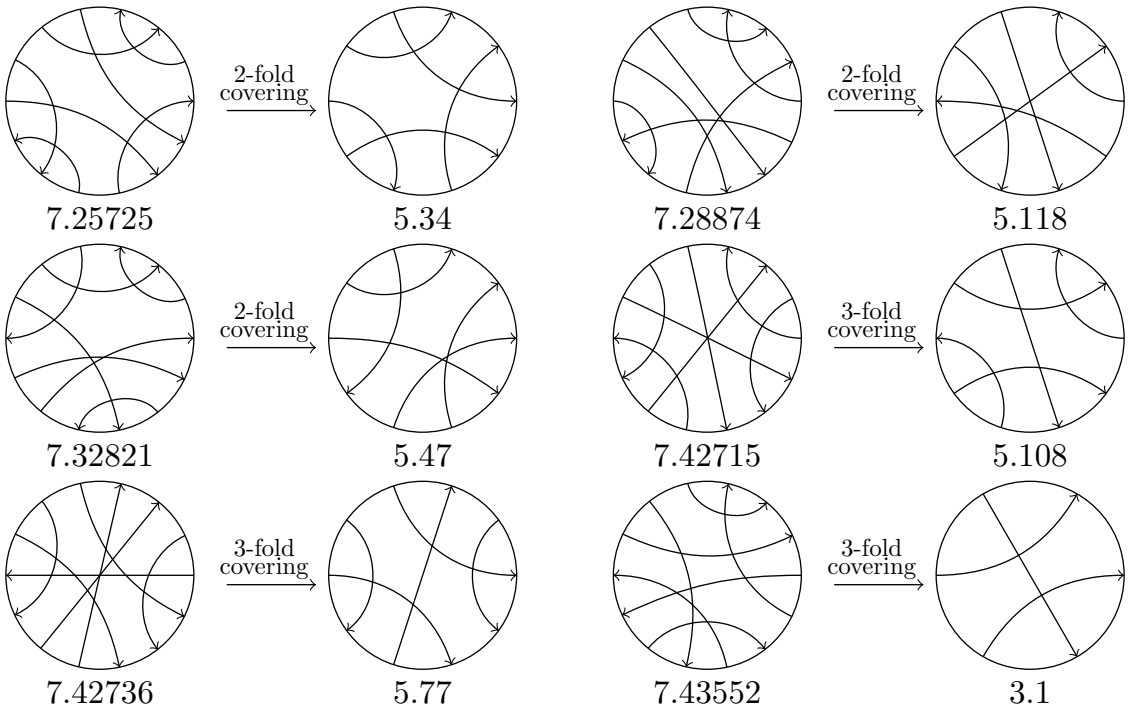


Figure 3.8: Six algebraically slice flat knots that are not slice

3.6 Based matrices and cabling

In this section, we study the based matrices of the (p, q)-cables of flat knots.

Recall the (p, q)-cable of a flat knot is well defined up to flat knot equivalence (see Definition 2.30). When p, q are coprime, the (p, q)-cable $\alpha^{p,q}$ is again a flat knot (rather than a flat link). Hence the primitive based matrices of $\alpha^{p,q}$ are well-defined flat knot invariants of α. We call such a matrix the (p, q)-cabled based matrix of α.

Proposition 3.21. *If the flat knots α and β have isomorphic based matrices from their diagrams, then $\alpha^{2,1}$ and $\beta^{2,1}$ also have isomorphic based matrices.*

Proof. By reordering the crossings, we assume α, β have the same based matrix. In $\beta^{2,1}$, the crossings either come in quadruple as A,B,C,D or alone as E in shown in Figure 3.9. Let $s, x, y \in H_1(\Sigma_g)$ be corresponding to the whole knot, the quadruples A,B,C,D and M,N,P,Q.

We use the same Carter surface Σ_g as an immersion representation for $\beta^{2,1}$. Then we obtain $[E^+] = s \in H_1(\Sigma_g)$, $[A^+] = [D^+] = x \in H_1(\Sigma_g)$, $[M^+] = [O^+] = y \in H_1(\Sigma_g)$, $[B^+] = [C^+] = x + s \in H_1(\Sigma_g)$, $[N^+] = [P^+] = y + s \in H_1(\Sigma_g)$. Therefore,

$$\begin{aligned}
b([A^+],[E^+]) &= b([B^+],[E^+]) = b([C^+],[E^+]) = b([D^+],[E^+]) = b(x,s),\\
b([A^+],[M^+]) &= b([D^+],[O^+]) = b(x,y),\\
b([A^+],[N^+]) &= b([D^+],[N^+]) = b([A^+],[P^+]) = b([D^+],[P^+])\\
&= b(x,y) + b(y,s),\\
b([B^+],[N^+]) &= b([C^+],[N^+]) = b([B^+],[P^+]) = b([C^+],[P^+])\\
&= b(x,y) + b(x,s) + b(s,y).
\end{aligned}$$

Let $t \in H_1(\Sigma_g)$ be corresponding to the whole knot $\alpha^{2,1}$. Then we have $t = 2s$ and hence $b([A^+], t) = 2b([A^+], s)$. Then, if flat knots α and β have isomorphic based matrices from their diagrams, there exist a based matrix as constructed above that belong to $\alpha^{2,1}$ and $\beta^{2,1}$. Then they have isomorphic based matrices as well. □

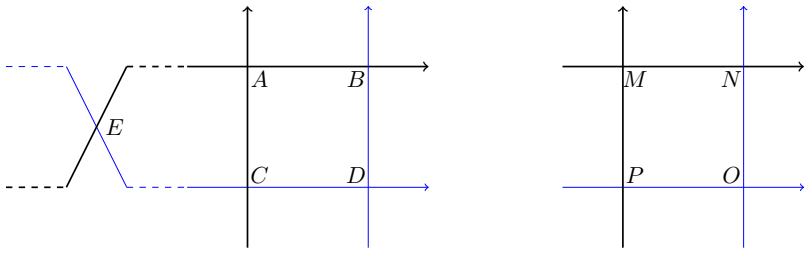


Figure 3.9: Crossings in $(2, 1)$-cable

Remark 3.22. Once we know a based matrix for α, then a based matrix for $\alpha^{2,1}$ can be constructed easily, as in shown in Figure 3.10.

0	$-C^T$
C	H

$\mathcal{C} = [C, \ldots, C]$, where $\dim(\mathcal{C}) = \dim(H)$.

0	$-2C^T$	$-2C^T$	$-2C^T$	$-2C^T$	0
$2C$	H	$H+\mathcal{C}$	$H+\mathcal{C}$	H	C
$2C$	$H-\mathcal{C}^T$	$H+\mathcal{C}-\mathcal{C}^T$	$H+\mathcal{C}-\mathcal{C}^T$	$H-\mathcal{C}^T$	C
$2C$	$H-\mathcal{C}^T$	$H+\mathcal{C}-\mathcal{C}^T$	$H+\mathcal{C}-\mathcal{C}^T$	$H-\mathcal{C}^T$	C
$2C$	H	$H+\mathcal{C}$	$H+\mathcal{C}$	H	C
0	$-C^T$	$-C^T$	$-C^T$	$-C^T$	0

Figure 3.10: Based matrix for α (left) and based matrix for $\alpha^{2,1}$ (right)

For example, we calculated the based matrix of the flat knot 4.1 in Remark 3.13. Its $(2,1)$-cable has based matrix

$$\begin{bmatrix}
0 & -6 & -2 & 4 & 4 & -6 & -2 & 4 & 4 & -6 & -2 & 4 & 4 & -6 & -2 & 4 & 4 & 0 \\
6 & 0 & 1 & 3 & 2 & 3 & 4 & 6 & 5 & 3 & 4 & 6 & 5 & 0 & 1 & 3 & 2 & 3 \\
2 & -1 & 0 & 2 & 1 & 0 & 1 & 3 & 2 & 0 & 1 & 3 & 2 & -1 & 0 & 2 & 1 & 1 \\
-4 & -3 & -2 & 0 & 0 & -5 & -4 & -2 & -2 & -5 & -4 & -2 & -2 & -3 & -2 & 0 & 0 & -2 \\
-4 & -2 & -1 & 0 & 0 & -4 & -3 & -2 & -2 & -4 & -3 & -2 & -2 & -2 & -1 & 0 & 0 & -2 \\
6 & -3 & 0 & 5 & 4 & 0 & 3 & 8 & 7 & 0 & 3 & 8 & 7 & -3 & 0 & 5 & 4 & 3 \\
2 & -4 & -1 & 4 & 3 & -3 & 0 & 5 & 4 & -3 & 0 & 5 & 4 & -4 & -1 & 4 & 3 & 1 \\
-4 & -6 & -3 & 2 & 2 & -8 & -5 & 0 & 0 & -8 & -5 & 0 & 0 & -6 & -3 & 2 & 2 & -2 \\
-4 & -5 & -2 & 2 & 2 & -7 & -4 & 0 & 0 & -7 & -4 & 0 & 0 & -5 & -2 & 2 & 2 & -2 \\
6 & -3 & 0 & 5 & 4 & 0 & 3 & 8 & 7 & 0 & 3 & 8 & 7 & -3 & 0 & 5 & 4 & 3 \\
2 & -4 & -1 & 4 & 3 & -3 & 0 & 5 & 4 & -3 & 0 & 5 & 4 & -4 & -1 & 4 & 3 & 1 \\
-4 & -6 & -3 & 2 & 2 & -8 & -5 & 0 & 0 & -8 & -5 & 0 & 0 & -6 & -3 & 2 & 2 & -2 \\
-4 & -5 & -2 & 2 & 2 & -7 & -4 & 0 & 0 & -7 & -4 & 0 & 0 & -5 & -2 & 2 & 2 & -2 \\
6 & 0 & 1 & 3 & 2 & 3 & 4 & 6 & 5 & 3 & 4 & 6 & 5 & 0 & 1 & 3 & 2 & 3 \\
2 & -1 & 0 & 2 & 1 & 0 & 1 & 3 & 2 & 0 & 1 & 3 & 2 & -1 & 0 & 2 & 1 & 1 \\
-4 & -3 & -2 & 0 & 0 & -5 & -4 & -2 & -2 & -5 & -4 & -2 & -2 & -3 & -2 & 0 & 0 & -2 \\
-4 & -2 & -1 & 0 & 0 & -4 & -3 & -2 & -2 & -4 & -3 & -2 & -2 & -2 & -1 & 0 & 0 & -2 \\
0 & -3 & -1 & 2 & 2 & -3 & -1 & 2 & 2 & -3 & -1 & 2 & 2 & -3 & -1 & 2 & 2 & 0
\end{bmatrix}.$$

Corollary 3.23. *If the flat knots α and β have homologous based matrices, then the cabled knots $\alpha^{2,1}$ and $\beta^{2,1}$ also have homologous based matrices.*

Proof. By same notation in the proof of Proposition 3.21, if x is an annihilating

element in G_α, then $[B^+], [C^+]$ are a pair of complementary elements in $G_{\alpha^{2,1}}$ since $b([B^+], \cdot) + b([C^+], \cdot) = 2b(s, \cdot) = b(t, \cdot)$, and $[A^+], [D^+]$ are annihilating elements in $G_{\alpha^{2,1}}$. If x is a core element in G_α, then $[A^+], [D^+]$ is a complementary pair in $G_{\alpha^{2,1}}$ since $b([A^+], \cdot) + b([D^+], \cdot) = 2b(s, \cdot) = b(t, \cdot)$, while $[B^+], [C^+]$ are two core elements in $G_{\alpha^{2,1}}$ since $b([B^+], \cdot) = b([C^+], \cdot) = 2b(s, \cdot) = b(t, \cdot)$. If x, y are a complementary pair in G_α, then $[A^+], [N^+]$, $[B^+], [O^+]$, $[D^+], [P^+]$ and $[C^+], [M^+]$ are four pairs in $G_{\alpha^{2,1}}$. □

Note that by this argument, we can construct the based matrix of $\alpha^{2,1}$ solely from a based matrix of α. On one hand, this also means that cabled based matrices are no more powerful than the based matrix as a flat knot invariant. On the other hand, this gives us some results on the cabling behavior of flat knots. Recall that the r-th covering of a flat knot α, realized as $\omega_\alpha : S^1 \to \Sigma$, is the lift of α to the r-fold covering of Σ induced by the dual in $H^1(\Sigma; \mathbb{Z}/r)$ of $[\omega(S^1)]$.

Corollary 3.24. *For any flat knot α and any odd integer q, the indices of $\alpha^{2,q}$ are in $2\mathbb{Z}$. Hence the 2-fold covering of $\alpha^{2,q}$ is $\alpha^{2,q}$ itself.*

Corollary 3.25. *For any flat knot α, if $2p$ and q are coprime, then the cabling $\alpha^{2p,q}$ is checkerboard colorable.*

Corollary 3.26. *Let p and q be coprime integers and let D_1 and D_2 be Gauss diagrams of the two flat knots α and β. If the based matrices of D_1 and D_2 are homologous, then $\alpha^{p,q}$ and $\beta^{p,q}$ also have homologous based matrices.*

Proof. We can calculate the based matrices of $D_1^{(p,q)}$ and $D_2^{(p,q)}$ as in the proof of Proposition 3.21. If D_1 and D_2 have isomorphic based matrices, then the matrices obtained by cabling are isomorphic.

The argument in the proof of Corollary 3.23 can be generalized to (p, q)-cable. Note that the (p, q)-cabling gives a permutation in the p strands. Take one strand as the 1-st strand, we can name the remaining strands $2, \ldots, p$-th strands in the order of the permutation. An arrow $e \in \mathrm{arr}(D)$ will give p^2 arrows $\{e_{i,j}\}_{1 \le i,j \le p} \subset \mathrm{arr}(D^{(2,1)})$, such that the tail of the arrow $e_{i,j}$ is on the i-th strand and the head of the arrow $e_{i,j}$ is on the j-th strand; see Figure 2.12 for the correspondence between arcs and arrow heads/tails.

Let x, y, s be $[e^+], [f^+]$ and the core element in $\bar{G}_D$ of D. Then $[e^+_{i,j}] = x + ks$, where $k \equiv j - i \mod p$ and $0 \le k \le p - 1$. If x is an annihilating element in $\bar{G}_D$, then $[e^+_{i,j}] + [e^+_{j,i}] = ns$ for $i \ne j$ and $[e^+_{i,j}] + [e^+_{j,i}] = 0$ for $i = j$. The two elements $[e^+_{i,j}], [e^+_{j,i}]$ are either a pair of complementary elements or two annihilating elements.

If x is a core element in $\bar{G}_D$, then for each $k = 1, 2, \ldots, n$, there are n elements in $\left\{[e^+_{i,j}]\right\}$ that is equal to ks, so $\left\{[e^+_{i,j}]\right\}$ consists of $\frac{n(n-1)}{2}$ pairs of complementary elements and n core elements.

If $x + y = s$ then $[e^+_{i,j}] + [f^+_{j,i}] = ns$, so the two elements $[e^+_{i,j}], [f^+_{j,i}]$ form a pair of complementary elements. □

Proposition 3.27. *The u-polynomial of $\alpha^{2,1}$ is given by $u_{\alpha^{2,1}}(t) = 4u_\alpha(t^2)$.*

Proof. By the based matrix of $\alpha^{2,1}$ given in Remark 3.22, we obtain that $u_{\alpha^{2,1}}(t) = 4u_\alpha(t^2)$. □

For example, since the flat knot 4.1 in Remark 3.13 has u-polynomial $t^3 - 2t^2 + t$, Proposition 3.27 applies to show that its $(2,1)$-cable has u-polynomial $4t^6 - 8t^4 + 4t^2$.

3.7 Realization of based matrices

In this section, we study the realization problem for based matrices of almost classical flat knots.

Recall that in Figure 2.27, we saw that the connected sum of round flat knot diagrams is not well defined. Recall also the set of permutant diagrams in Definition 2.36. In general, permutant diagrams need not have isomorphic based matrices. For example, the diagrams D_3, D_4 in Figure 3.11 are permutant diagrams with primitive based matrices given by:

$$T_\bullet(D_3) = \begin{bmatrix} 0 & -1 & 1 & -2 & 0 & 2 \\ 1 & 0 & 1 & -2 & 0 & 2 \\ -1 & -1 & 0 & -2 & 0 & 2 \\ 2 & 2 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ -2 & -2 & -2 & -2 & -1 & 0 \end{bmatrix}, \quad T_\bullet(D_4) = \begin{bmatrix} 0 \end{bmatrix}.$$

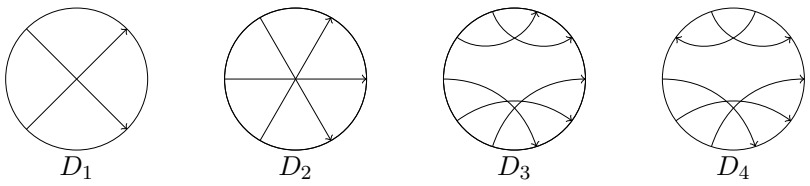


Figure 3.11: $D_3, D_4 \in [D_1 \# D_2]$

The next theorem below shows that permutant almost classical flat knots have isomorphic based matrices. In particular, based matrices are unable to separate permutant pairs of almost classical flat knots.

Proposition 3.28. *Let D_1, D_2 be two almost classical diagrams with based matrices T_1, T_2, respectively. Then every permutant diagram in $[D_1 \# D_2]$ has based matrix isomorphic to $\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}$.*

Proof. Since every almost classical Gauss diagram has all its indices equal to zero, the based matrices T_1 and T_2 both have zeros in the first rows and columns.

The intersection number $b(e, f)$ is equal to zero if e, f are associated to arrows from D_1, D_2, respectively. Otherwise, the intersection number before and after the connected sum are the same. [1] Therefore, every permutant diagram in $[D_1 \# D_2]$ has based matrix isomorphic to $\begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix}$. □

Consequently, the based matrices of the Gauss diagrams in Figure 3.12 are identical.

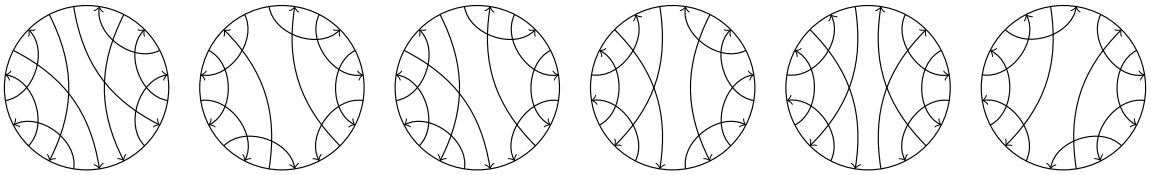


Figure 3.12: Six distinct almost classical permutant flat knots with the same based matrix.

Problem 3.29. *What matrices can be realized as based matrices or primitive based matrices for (almost classical) flat knots?*

In Chapter 6, we recall Turaev's construction of $\mathscr{FG}$, the algebraic concordance group of flat knots. Elements of $\mathscr{FG}$ are algebraic concordance classes of long flat knots, and each can be represented by (primitive) based matrices. This situation is roughly analogous to that of classical knots, where algebraic concordance classes can be represented by Seifert matrices. To gain a better understanding of topological concordance of flat knots, it would be helpful to know exactly which based matrices occur, or to have necessary and sufficient criteria for a based matrix to occur as that of a flat knot.

Let $\mathscr{B}$ be the set of all skew-symmetric matrices over $\mathbb{Z}$. Recall that in Section 3.4, we defined $\mathscr{B}_{AC}$ to be a subset of $\mathscr{B}$ of matrices whose first row and column have all zero entries. Let $\mathscr{B}^0_{AC}$ denote the subset of $\mathscr{B}_{AC}$ consisting of matrices such that the sum of entries in each column is equal to zero.

Example 3.30 shows there are matrices in $\mathscr{B}^0_{AC}$ that cannot be realized as based matrices of almost classical flat knots. It is an intriguing problem to better understand the subtle conditions satisfied by based matrices of almost classical flat knots. Any flat knot realizing the based matrix in Example 3.30 below is not algebraically slice. In particular, if the matrix is realized by an almost classical flat knot, then that would be an almost classical flat knot that is not slice and would thus answer Problem 6.26. On the other hand, the evidence from the knot tables supports the conjecture that every almost classical flat knot is slice.

[1] Referring to [Tur04, Section 12.5], the general case is reviewed in Equation (6.1).

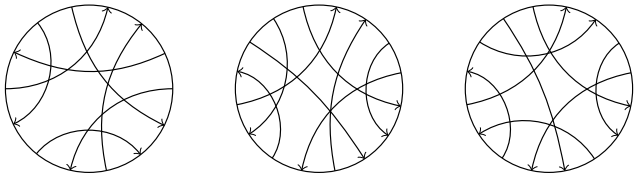
Example 3.30. There exists a based matrix in $\mathscr{B}^0_{AC}$ that is not null-concordant.

Proof. The matrix below is a primitive based matrix in $\mathscr{B}^0_{AC}$, but it is not algebraically slice because its algebraic genus is 2.

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
0 & 0 & 1 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 \\
0 & 1 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & -1 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\
0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & -1 & 1 \\
0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 & 0 & 1 & 0 & -1 & -1 & 1 & 0 \\
0 & 1 & -1 & 1 & 0 & 0 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & -1 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -1 & 1 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}$$

□

Example 3.31. There exists $T \in \mathscr{B}^0_{AC}$, such that T is a primitive based matrix of a flat knot that is not almost classical. Moreover, there exists $T \in \mathscr{B}_{AC}$, such that T is a primitive based matrix of a flat knot that is neither almost classical nor slice.



7.45422 8.1038682 8.1058826

Figure 3.13: Three non-AC flat knots with primitive based matrices in $\mathscr{B}_{AC}$

Proof. There are 5 non-AC flat knots with 7 crossings whose primitive based matrices are in $\mathscr{B}_{AC}$, and 42 non-AC flat knots with 8 crossings whose primitive based matrices are in $\mathscr{B}_{AC}$. This includes the three flat knots in Figure 3.13. For the flat knot 7.45422, its based matrix and primitive based matrix are

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
-1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \text{ and } \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0
\end{bmatrix},$$

respectively. Note that the primitive based matrices for the flat knots 7.45422, 8.1038682 and 8.1058826 lie in $\mathscr{B}_{AC}$ but not in $\mathscr{B}^0_{AC}$. Figure 2.26 shows that 7.45422 is a slice knot. On the other hand, the genus of the based matrices for 8.1038682 and 8.1058826 are both 1, so these flat knots are not algebraically slice. □

Remark 3.32. We list criteria for a matrix to occur as the based matrix of a flat knot; see [Tur04] for more details. For a based matrix $T \in \mathscr{B}$ to occur as a based matrix of a flat knot:

1. the sum of the entries in the first row and column of T should equal zero,
2. every entry of T should be strictly less than $\dim(T) - 1$.

For a matrix $T \in \mathscr{B}_{AC}$ to occur as the based matrix of an almost classical flat knot:

1. the sum of the entries in any row or column of T should equal zero,
2. every entry of T should be strictly less than $\frac{1}{2}(\dim(T) - 1)$.

Conjecture 3.33. *If $T \in \mathscr{B}^0_{AC}$ is realized as a (primitive or not) based matrix of a flat knot (not necessarily almost classical), then T is null-concordant.*

The core of this question is to find finer criteria for based matrices to be realized from flat knots. If this conjecture is true, then the matrix in Example 3.30 is not a based matrix of any flat knot.

Chapter 4

The flat arrow polynomial

In this chapter, we define the flat arrow polynomial, and we discuss a skein theoretic formula for the leading term in the flat arrow polynomial. We also combine cabling with the flat arrow polynomial to produce more powerful invariants of flat knots.

4.1 Definition and basic properties

In this section, we define the flat arrow polynomial of a flat knot α. It is closely related to the arrow polynomial of virtual knots, which was originally introduced by Dye and Kauffman as a powerful generalization of the Jones polynomial defined for virtual knots, see [Jon85, Kau99, DK09].

The flat arrow polynomial is defined by applying the skein relation of Figure 4.1(i) to each flat crossing of a flat knot diagram D. If α has n crossings, then this gives 2^n states. The solid triangles in Figure 4.1(ii) are called *cusps* and follow the rule of reduction as shown. For each state S, after reduction, each loop in S can be assigned as in Figure 4.1(iii). Denote by $\langle S \rangle$ the product of the image in polynomial ring $\mathbb{Z}[K_1, K_2, \ldots, K_n]$ of all loops in S.

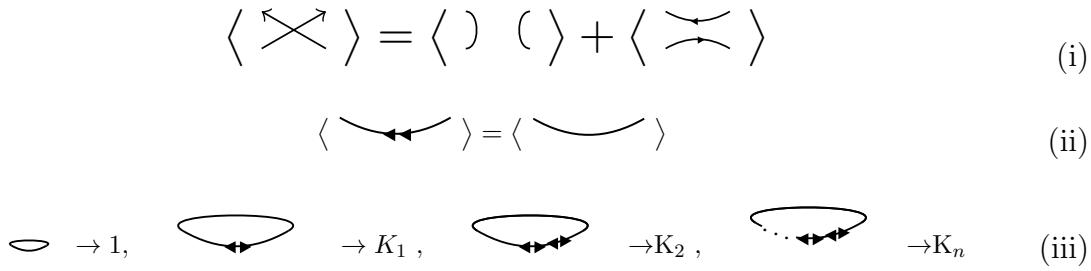


Figure 4.1: State sum model for the flat arrow polynomial

Definition 4.1. For a flat knot or link diagram D, the *flat arrow polynomial* is

defined to be

$$A(D) = \sum_S (-2)^{|S|-1}\langle S\rangle,$$

where the sum is over all states S and $|S|$ denotes the number of loops in S.

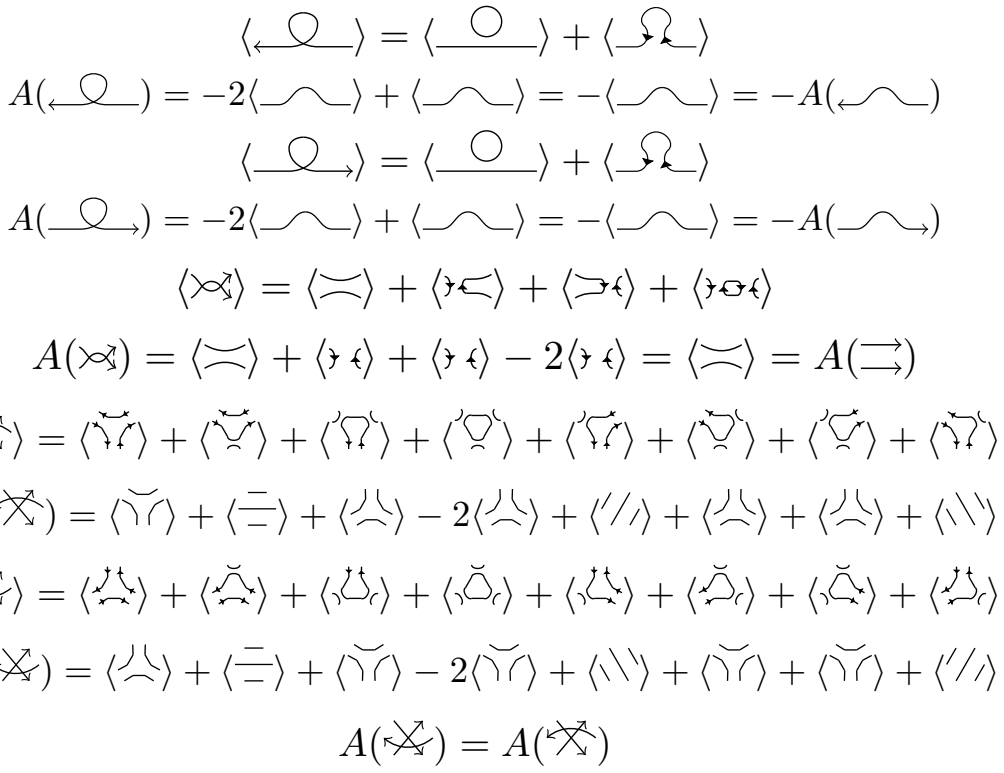
The *(normalized) arrow polynomial of flat knot* α is defined to be

$$\bar{A}(\alpha) = (-1)^{cr(D)}A(D),$$

where D is a flat knot diagram of α.

Proposition 4.2. *The normalized flat arrow polynomial is an invariant of flat knots.*

Proof. If two flat knot diagrams are homotopic then they are related by a finite sequence of flat Reidemeister moves. Observe that VR1,2,3 and FR4 do not change the calculation. As mentioned in Remark 2.12, we only need to show the flat arrow polynomial is invariant under FR1(a,b), FR2(a) and FR3(a) for oriented flat knot diagrams. The calculation details are shown below.



Therefore, we conclude that the normalized flat arrow polynomial is an invariant of flat knots. □

Proposition 4.2 can also be proved using the arrow polynomial of virtual knots and shadow projection from virtual knots to flat knots. The arrow polynomial is an invariant of virtual knots taking values in $\mathbb{Z}[a, a^{-1}, K_1, K_2, \ldots, K_n]$ [DK09]. Let K be a virtual knot and let π be the shadow map from virtual knots to flat knots (cf. Definition 2.20). Hence the normalized flat arrow polynomial is an invariant of flat knots.

Proposition 4.3. *The following statements are true.*

1. *An n crossing diagram of the trivial flat knot has flat arrow polynomial $(-1)^n$.*
2. *Let K be a virtual knot with arrow polynomial $f(K) \in \mathbb{Z}[a, a^{-1}, K_1, K_2, \ldots, K_n]$ (c.f. [DK09]). Then $\bar{A}_{\pi(K)} = f(K)|_{a=1}$.*
3. *For any flat knot α, we have $\bar{A}(\alpha)|_{K_i=1} = 1$.*

Proof. The first claim follows immediately from Proposition 4.2.

The shadow projection π equalizes the over and under crossings in virtual knot diagrams. Observe that in the definition of arrow polynomial of virtual knots, changing the over crossing to under crossing only changes the a to a^{-1}. By the definition of flat knot arrow polynomial, by setting $a = 1$ of the arrow polynomial of a virtual knot K, we obtain $\bar{A}(\pi(K))$.

By [Dye16, Equation 10.8, 10.9], the Jones polynomial of a virtual knot is obtained from its arrow polynomial by setting all K_i's to be 1. Moreover, it is well known that assigning $a = 1$ sends all Jones polynomials to $(-2)^{n-1}$, where n is the number of the components; see [Dye16, Theorem 10.1]. Therefore, we have $\bar{A}(\alpha)|_{K_i=1} = 1$. □

The operation of connected sum for virtual and flat knots is introduced in Section 2.8. The following example shows that the flat arrow polynomial is not multiplicative under connected sum. For example, the flat knots 4.5 and 6.132 in Figure 4.2 are connected sums of two diagrams of the flat unknot. However, their flat arrow polynomials are

$$\begin{aligned}\bar{A}(4.5) &= -4K_1^2 + 2K_2 + 3,\\ \bar{A}(6.132) &= -16K_1^4 + 8K_1^2K_2 + 8K_1^2 + 1.\end{aligned}$$

If the flat arrow polynomial were multiplicative, then we would have $\bar{A}(4.5) = \bar{A}(6.132) = 1$. Since that is not the case, we conclude that $\bar{A}(\alpha)$ is not multiplicative under connected sum.

Moreover, permutant diagrams can have different flat arrow polynomials. The flat knots 6.139 and 6.549 in Figure 4.2 are permutant since both can be realized as $D\#D'$, where for minimal crossing diagrams D and D; of -3.1 and 3.1, respectively. However, their flat arrow polynomials are

$$\begin{aligned}\bar{A}(6.139) &= 4K_1^2K_2 - 4K_1K_3 + K_4,\\ \bar{A}(6.549) &= 1.\end{aligned}$$

Observe that the flat arrow polynomial of D is nontrivial, indeed $\bar{A}(3.1) = 2K_1^2 - K_2$. The same is true for D'. However, their connected sum $6.549 = D\#D'$ has trivial flat arrow polynomial. These examples also show that the constant term of the flat arrow polynomial is not multiplicative under connected sum.

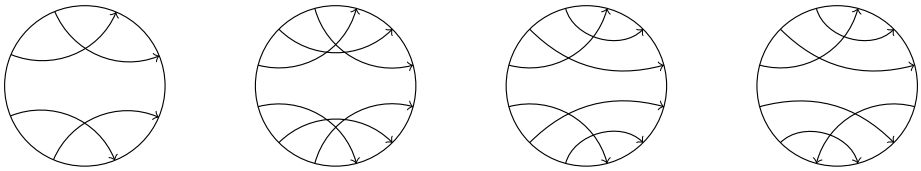


Figure 4.2: Flat arrow polynomial is not multiplicative under connected sum.

4.2 Free equivalence, checkerboard colorability, and almost classicality

In this section, we relate checkerboard colorability and almost classicality of flat knots. When the flat knot is almost classical, we show that its flat arrow polynomial is trivial, and when it is checkerboard colorable, we show its flat arrow polynomial has odd constant term.

We begin by defining the notion of free equivalence of flat knots.

Definition 4.4 ([Tur08a])**.** Two flat knots are said to be *free-equivalent* if they are related by the following relation. The set of *free knots* consists of flat knots modulo free-equivalence.

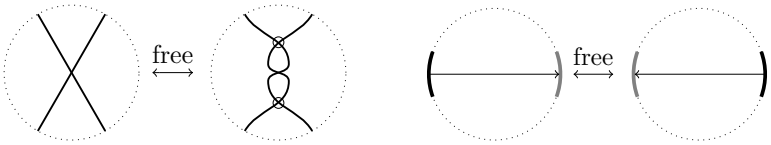


Figure 4.3: Free-equivalence on a flat knot diagram and a flat Gauss diagram

Note that all flat knots up to crossing number 4 are free-equivalent to the unknot. In [Man12], Manturov defined a non-negative integer-valued invariant L of free knots (cf. [CFG+20, Section 6]). It is an obstruction to the knot being *freely slice*. There are examples of free knots with nontrivial L-invariant. Figure 4.4 shows two flat knots with 5 crossings that are nontrivial as free knots. In fact, they are not freely slice.

Definition 4.5. A flat Gauss diagram is said to be of *alternating pattern* if its underlying OU-word is "OUOU $\cdots$ OU".

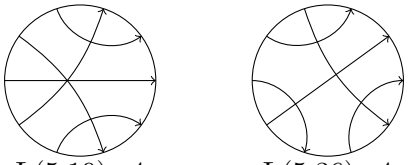


Figure 4.4: Flat knots that are not freely trivial

Note that this definition is adapted from the definition of alternating pattern for virtual knot in [Kar18].

Lemma 4.6 ([Kar18])**.** *Every flat knot with alternating pattern is almost classical.*

Lemma 4.7. *Any flat knot free-equivalent to a checkerboard colorable flat knot is also checkerboard colorable.*

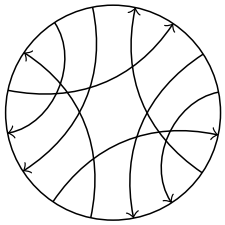
Proof. We consider the effect on the arrow index $n(f)$ of applying a free-equivalence to an arrow e. Suppose first that $e \neq f$. If f^+ contains both e^O and e^U, then the index $n(f)$ does not change. The same is true if f^+ does not contain either e^O or e^U. If f^+ contains only one of e^O and e^U, then the index $n(f)$ changes by ± 2. In case $e = f$, it is easy to verify that the index $n(e)$ changes by sign. Therefore, the parity (even /odd) of an arrow is preserved under free-equivalence, and whether a chord is even or odd is well-defined for free knots. In particular, if two flat knot diagrams D_1, D_2 are free-equivalent and if all the arrows of D_1 are even (i.e., if $n(e) \equiv 0 \mod 2$ for all $e \in \mathrm{arr}(D_1)$), then the same must be true for D_2. □

Lemma 4.8. *A flat knot is almost classical if and only if it has a diagram (not necessarily minimal) of alternating pattern.*

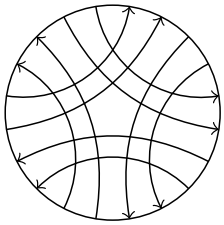
Proof. An almost classical flat knot represents an immersed loop bounding an immersed oriented surface F in a Carter surface. At the cost of increasing the crossing number, we can turn F to a disk attached by finitely many bands, where crossings only occur in quadruples when a band crosses another band. In this way, F has only one side facing to the positive side of the Carter surface. By this construction, the crossings are alternating. Conversely, by Lemma 4.6, a Gauss diagram with alternating pattern "OUOU $\cdots$ OU" has only chords of index zero and hence is almost classical. □

The alternating pattern cannot always be chosen to have minimal crossing number. For example, the almost classical flat knots ac8.16 and ac10.1088 in Figure 4.5 do not have minimal crossing alternating pattern diagrams.

Lemma 4.9. *A flat knot is checkerboard colorable if and only if it has a Gauss diagram that is free-equivalent to a diagram of alternating pattern.*



0102U304U2U1050 3U6U708U507 06U4U8



0102U3U40506U7U80403U9U1008 07U2U1010 09U6U5

Figure 4.5: Almost classical flat knots ac8.16 and ac10.1088.

Proof. A flat checkerboard colorable diagram lifts to checkerboard colorable virtual diagrams. By [Kam02, Lemma 7], a checkerboard colorable virtual diagram can be made alternating by crossing changes, which does not change the flat knot type it projects to. Therefore, for every flat checkerboard colorable diagram, there exists an alternating virtual diagram D projecting to it. Apply free-moves at all negative crossings of flat diagram $\pi(D)$, we obtain a flat diagram of alternating pattern. □

Lemma 4.10. *If flat knot α is almost classical, then $A(\alpha) = 1$.*

Proof. We know α can be represented as the boundary of an immersed Seifert surface $F \looparrowright \Sigma$ and we can alter the singular points so that we get an embedding $F \to \Sigma \times I$, whose boundary represents a virtual knot, say K. By this construction, we have K is also almost classical (also called null-homologous) $\pi(K) = \alpha$, where π is the shadow projection. By [Mil23, Theorem 3.21], if the virtual knot $K$is almost classical then its arrow polynomial is the same as its Jones polynomial. Then by Proposition 4.3-(3), $\bar{A}(\alpha) = 1$. □

We use $C(D)$ to denote the constant term of $A(D)$ for a flat knot diagram D, and we use $\bar{C}(\alpha)$ for the constant term of $\bar{A}(\alpha)$ for a flat knot α. If D is a diagram with n crossings representing α, then these are related by $\bar{C}(\alpha) = (-1)^n C(D)$.

Theorem 4.11. *If the flat knot α is checkerboard colorable, then $C(\alpha) \equiv 1 \mod 2$.*

Proof. By Lemma 4.8, Lemma 4.9 and Lemma 4.10, α is obtained from some almost classical Gauss diagram D with $C(D) = \pm 1$ by free-move or (arrow-change in Gauss diagram). Observe that when we apply one free-move in Figure 4.3, the state resolution described in Figure 4.1 does not change except that the cusps in (1) are in opposite directions. If the state S has more than two loops, then $(-1)^n(-2)^{|S|-1}\langle S\rangle$ either has zero constant term or $(-2)^{|S|-1}$. If the state S has only one loop, then the two cusps are in the same loop, then either the number of cusps after deduction is either changed by at most ± 4. However, by [Mil23, Theorem 3.31], the cusps number before and after a free-move can be only be $8n$: since before and after the move the flat knot remains checkerboard colorable and thus they both lift to some

checkerboard colorable virtual knots which has only K_{4n} for single-loop states mapping to monomials in their arrow polynomials. Therefore, the constant term of an checkerboard colorable knot remains an odd number. □

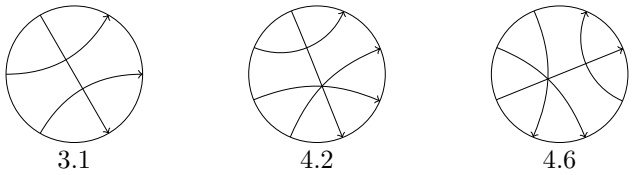


Figure 4.6: Three non-checkerboard-colorable flat knots

The flat knots shown in Figure 4.6 have $\bar{C}(3.1) = 0$, $\bar{C}(4.2) = 2$, $\bar{C}(4.6) = 2$. Since these are all even, Theorem 4.11 applies to show that these flat knots are not checkerboard colorable.

Remark 4.12. For all flat knots up to 7 crossings, one can check that $\bar{C}(\alpha)$ is odd whenever α is slice (refer to Definition 2.32. for the definition of sliceness). However, there exist flat knots β that are slice such that $\bar{C}(\beta)$ is even. For example, the flat knot 8.11946 in Figure 4.7 is slice and has $\bar{A}(8.11946) = 12K_1^3 + 4K_1^2K_2 + 4K_1^2K_3 - 12K_1^2 + 4K_1K_2^2 - 20K_1K_2 - 4K_1K_3 - 4K_2K_3 + 6K_2 + 4K_3 + K_4 + 6$ and hence $\bar{C}(8.11946) = 6$.

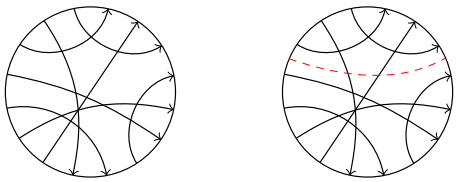


Figure 4.7: Flat knot 8.11946 and saddle move to slice it

4.3 The cabled flat arrow polynomial

In this section, we combine the cabling operation with the flat arrow polynomial to produce stronger invariants for flat knots. These resulting invariants are analogues of the colored Jones polynomials for classical knots. Cabling can often be used to produce powerful knot and link invariants, and we will see that is true for flat knots.

We give two examples to illustrate how cabling produces more powerful invariants. First consider the flat knot $\alpha = 4.4$. It has trivial flat arrow polynomial but its 2-strand cable α^2 has flat arrow polynomial $A(\alpha^2) = -128K_2^4 + 128K_2^2K_4 - 32K_4^2 + 30$.

Now consider the flat knots $\alpha = 4.1$ and $\beta = 4.3$. Both of these flat knots have flat arrow polynomial $A(\alpha) = A(\beta) = -2K_1K_2 + K_1 + K_3 + 1$. Their 2-strand cables have flat arrow polynomials

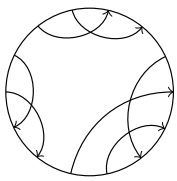
$$\begin{aligned} A(\alpha^2) = {} & 128K_1^2K_2^2 + 48K_1^2K_2 - 88K_1^2 + 64K_1K_2^3K_3 + 224K_1K_2K_3 \\ & + 16K_1K_3K_4 - 32K_2^4 - 64K_2^2K_3^2 - 8K_2^2K_4^2 + 8K_2^2K_4 - 54K_2^2 \\ & + 16K_2K_3K_5 + 8K_2K_4K_6 - 88K_3^2 - 12K_4^2 - 2K_6^2 + 90, \end{aligned}$$

and

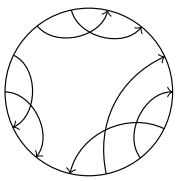
$$\begin{aligned} A(\beta^2) = {} & -72K_1^2 + 96K_1K_2K_3 + 48K_1K_3K_4 - 8K_2^2K_4^2 + 40K_2^2K_4 \\ & - 86K_2^2 + 8K_2K_4K_6 - 72K_3^2 - 44K_4^2 - 2K_6^2 + 90. \end{aligned}$$

Although the flat knots α and β are not distinguished by the flat arrow polynomials, they are distinguished by their 2-strand cabled flat arrow polynomials. These examples show that under cabling, the flat arrow polynomial produces a more powerful invariant of flat knots.

However, the complexity of calculating the flat arrow polynomial is exponential in the crossing number, and the crossing number increases rapidly under cabling. We use the divide-and-conquer method described in [KAT1] by Bar-Natan to calculate the 2-strand cable. However, it is impractical to calculate the flat arrow polynomial of higher cables. An alternative which is much faster and quite effective is to calculate only the constant term $\bar{C}(\alpha)$ of the flat arrow polynomial. As an invariant, $\bar{C}$ of the cables is quite powerful. For example, by calculating the constant term $\bar{C}$ of the 3-strand cables, we are able to distinguish the flat knots in Figure 4.8.



$\bar{C}(\alpha^3) = 74297683$



$\bar{C}(\beta^3) = 75299554$

Figure 4.8: Permutant pair with same ϕ-invariants, $\bar{A}(\alpha)$, $\bar{A}(\alpha^2)$ but separated by $\bar{C}(\alpha^3)$.

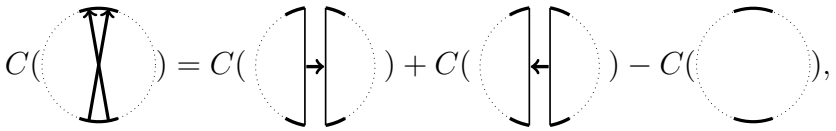
Recall the attempt we have made to relate the flat arrow polynomial or its constant term to sliceness in Remark 4.12. The calculation we have done in [FKI] suggests the conjecture below.

Conjecture 4.13. *If the flat knot α is slice, then $\bar{C}(\alpha^2) \equiv 2 \mod 4$.*

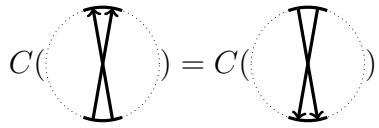
4.4 A skein formula

In this section, we derive a skein formula for the constant term in the flat arrow polynomial.

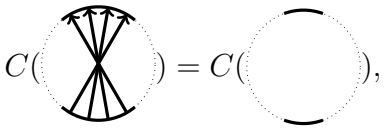
Proposition 4.14. *For flat knot (or link) Gauss diagrams, it holds that*



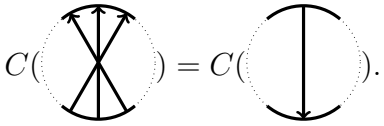
(4.1)



(4.2)

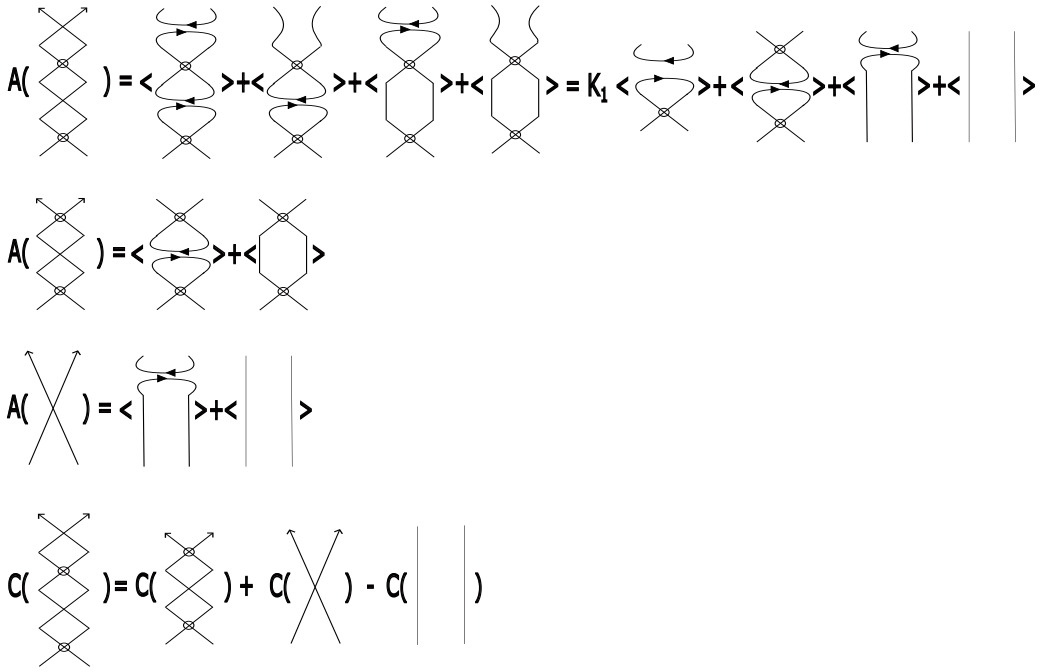


(4.3)



(4.4)

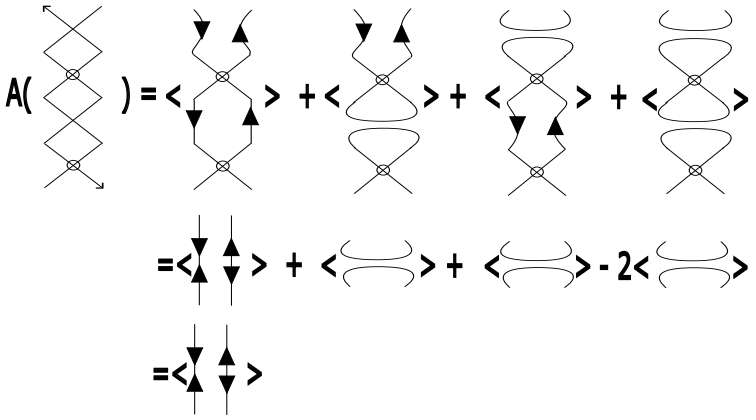
Proof. The state sum for the Gauss diagrams in Equation (4.1) is given below. Since $C(D) = A(D)|_{K_i=0}$, we obtain the equation in the fourth line. Representing the fourth line in terms of Gauss diagrams gives Equation (4.1).



If we replace the left hand side of (1) by $C($  $)$, the right hand side stays the same, which gives Equation (4.2).

To obtain Equation (4.3) and (4.4), we can switch two arrows (also called chords) on  and  . Then apply FR2 (a). □

The flat arrow polynomial of Gauss diagram $A($  $)$ can be calculated as below (in terms of flat knot diagram).



Corollary 4.15. *The $C(D)$ of flat link Gauss diagram in Figure 4.9 (left) is equal to $n \mod 2$.*

Corollary 4.16. *The $C(D)$ of flat Gauss diagram in Figure 4.9 (right) is given in the following table.*

$C(D)$	$n \equiv 0 \mod 4$	$n \equiv \pm 1 \mod 4$	$n \equiv 2 \mod 4$
$m \equiv 1 \mod 2$	-1	-1	1
$m \equiv 0 \mod 2$	1	0	1

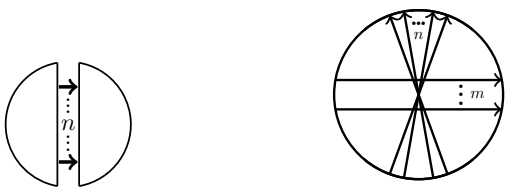


Figure 4.9: Gauss diagrams

For example, as shown in Figure 4.10, the flat knot 4.1 has $m = 2$, $n = 2$, so $\bar{C}(4.1) = 1$; the flat knot 5.1 has $m = 2, n = 3$, so $\bar{C}(5.1) = 0$; the flat knot 5.3 has $m = 3, n = 2$, so $\bar{C}(5.3) = -1$; the flat knot 5.12 has $m = 4, n = 1$, so $\bar{C}(5.12) = 0$.

Along with Definition 4.1, we can calculate $\bar{C}(\alpha)$ in much fewer steps than $A(\alpha)$.

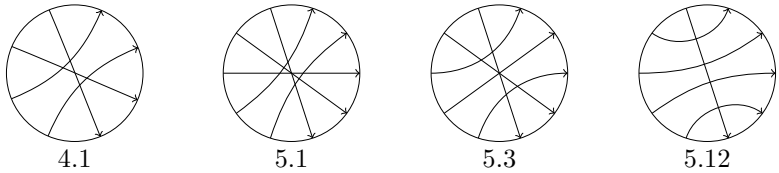


Figure 4.10: Flat knots 4.1, 5.1, 5.3, and 5.12

4.5 Distinguishing flat knots

Using 2-strand cabled arrow polynomials alone, we can distinguish flat knots up to 4 crossings completely. Combined with the ϕ-invariant, we can further distinguish flat knots completely up to 6 crossings and with just 6 pairs of 7-crossing knots not separated, see Table 4.1.[1]

Crossings	# Flat knots	ϕ	$A(\alpha)$	$A(\alpha^2)$	$A(\alpha^2)$, $C(\alpha^3)$ and ϕ
3	1	0	0	0	0
4	11	0	10	0	0
5	120	8	111	2	0
6	2086	74	1919	10	0
7	46233	1375	42163	—	12

Table 4.1: Number of non-distinguished flat knots using the invariant(s)

Specifically, there are two 4-crossing flat knots, eight 5-crossing flat knots, 106 6-crossing flat knots, and 674 7-crossing flat knots that are not distinguished from the unknot by $A(\alpha)$.

The first pair of flat knots that cannot be distinguished by the 2-strand cabled arrow polynomial are 5.112 and 5.113:

$$A(5.112^2) = A(5.112^2) = -64K_1^4 + 144K_1^2K_2 - 80K_1^2 - 56K_2^2 + 54$$

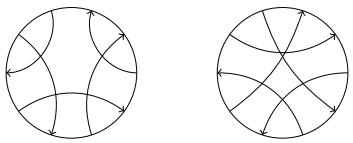
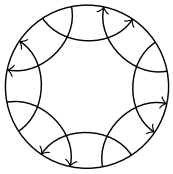


Figure 4.11: Flat knots 5.112 and 5.113

[1]When $cr(\alpha) \geq 7$, we only calculated $A(\alpha^2), C(\alpha^3)$ if they are not distinguished by other invariants. Some calculation of $C(\alpha^3)$ of some 7-crossing flat knots are not finished due to the workload of the calculation.

We know that the arrow polynomial and cabled arrow polynomials of any almost classical flat knot are trivial. To distinguish almost classical knots, we rely on the primitive based matrices. However, there exist nontrivial flat knots with trivial primitive based matrices. For example, the flat knot ac8.19 in Figure 4.12.

In fact, every invariant of flat knots studied up to now is trivial for the flat knot ac8.19 (also called 8.1241248). This flat knot, O1O2U3U1O4O3U5U4O6O5U7U6O8O7U2U8, is the 19-th 8-crossing almost classical knot and 1241248-th 8-crossing flat knot; see Remark 2.17 and Figure2.14 for how the flat knots are ordered in the tabulation. One can find the knot on [FKI] by 8.1241248.) The monotonicity algorithm is the only way to deduce that ac8.19 is nontrivial as a flat knot.



O1O2U3U1O4O3U5U4O6O5U7U6O8O7U2U8

Figure 4.12: Almost classical flat knot ac8.19 has trivial primitive based matrix.

Problem 4.17. *Find a flat knot invariant that distinguishes* ac8.19 from the flat unknot.

In the next chapter, we will introduce new invariants of flat knots and use them to address this question.

There are other examples of almost classical flat knots with trivial primitive based matrices as shown in Table 4.2.

Almost classical name	Gauss word
ac10.174	O1O2U3O4U2O5U4O6U7O3U6U8O9O7U5U1O0O8U1O1O0U9
ac10.175	O1O2U3O4U2O5U6O7U5O3U7U8O9O6U4U1O0O8U1O1O0U9
ac10.561	O1O2U3O4U2O5U4U6O7O8U5O9U8O3U9U1O0O6U1O1O0U7
ac10.760	O1O2U3O4U5O3U6U1O7O6U4U8O9O5U1OU7O8O1OU2U9
ac10.1103	O1U2O3U4O2U5O6U1O4U6O7U8O9U7O1OU3O8U1O0O5U9
ac10.1160	O1U2O3U4O2U5O6U3O7U8O5U7O9U1O8U1O0O4U9O1OU6
ac10.1170	O1U2O3U4O2U5O4U6O7U3O5U8O9U1O6U9O1OU7O8U10
ac10.1310	O1U2O3U4O2U5O4U6O5U7O6U8O9U3O7U1O0O8U1O1O0U9
ac10.1454	O1U2O3U4O2U5O4U6O7U8O5U9O8U3O9U1O0O6U1O1OU7
ac10.1643	O1U2O3U4O2U5O6U7O4U8O5U9O7U1O8U1OO9U3O1OU6
ac10.1657	O1U2O3U4O5U6O2U7O4U8O6U9O7U1O0O8U1O9U3O1OU5

Table 4.2: Almost classical knots with 10-crossings and trivial primitive based matrix

The invariants introduced in the next chapter can also be used to distinguish these flat knots from the flat unknot.

4.6 Open questions

We conclude this chapter with a few open problems.

Problem 4.18. *Is there a bigraded invariant that categorifies the flat arrow polynomial, cf.* [DKM11]*?*

Problem 4.19. *Can one use the flat arrow polynomial to extract slice obstructions for flat knots?*

Problem 4.20. *Which polynomials can be realized as flat arrow polynomials of flat knots?*

Problem 4.21. *Is there a polynomial-time algorithm for computing the constant term $C(D)$ via skein theory; see Section 4.4?*

Chapter 5

The flat Jones-Krushkal polynomial

In this chapter, we define the flat Jones-Krushkal polynomial, as well as a normalization and enhancement of it. We give a matrix-based algorithm to calculate these polynomials, and we apply them to the problem of distinguishing flat knots.

5.1 Definition

We begin by introducing the flat Jones-Krushkal polynomial, which is defined for flat knot diagrams on closed surfaces. It is closely related to the homological Jones polynomial for links in thickened surfaces, which was introduced by Krushkal in [Kru11] and further studied in [BK22].

Let D be a flat knot diagram with n crossings on a closed surface Σ. By applying the skein relation of Figure 5.1(1) to each flat crossing of D, we obtain 2^n states. The states can be indexed by a map $\{1, \ldots, n\} \to \{0, 1\}$. Denote this set of states by $\mathfrak{S}$.

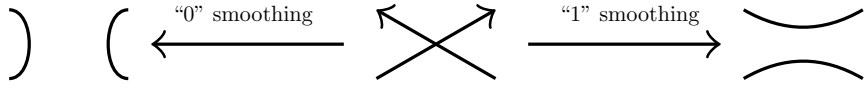


Figure 5.1: Two types of smoothings

Each state $S \in \mathfrak{S}$ contains simple closed loops embedded in Σ. The embedding induces a map $i_* : H_1(S; \mathbb{Z}/2) \to H_1(\Sigma_g; \mathbb{Z}/2)$.

Definition 5.1. The *homological Kauffman bracket* of state S is denoted by $\langle D \,|\, S \rangle_\Sigma$ and given by

$$\langle D \,|\, S \rangle_\Sigma = (-2)^{k(S)} z^{r(S)},$$

where

$$k(S) = \dim(\ker(i_* : H_1(S; \mathbb{Z}/2) \to H_1(\Sigma_g; \mathbb{Z}/2))),$$

$$r(S) = \dim(\mathrm{im}(i_* : H_1(S; \mathbb{Z}/2) \to H_1(\Sigma_g; \mathbb{Z}/2))).$$

Given a flat knot diagram D on the closed surface Σ, the *flat Jones-Krushkal polynomial* is defined by

$$J_D(z) = (-1)^{cr(D)} \sum_{S \in \mathfrak{S}} \langle D \mid S \rangle_\Sigma = (-1)^{cr(D)} \sum_{S \in \mathfrak{S}} (-2)^{k(S)} z^{r(S)}.$$

Proposition 5.2. *If D is checkerboard colorable, then $2|J_D$. If D is not checkerboard colorable, then $z|J_D$.*

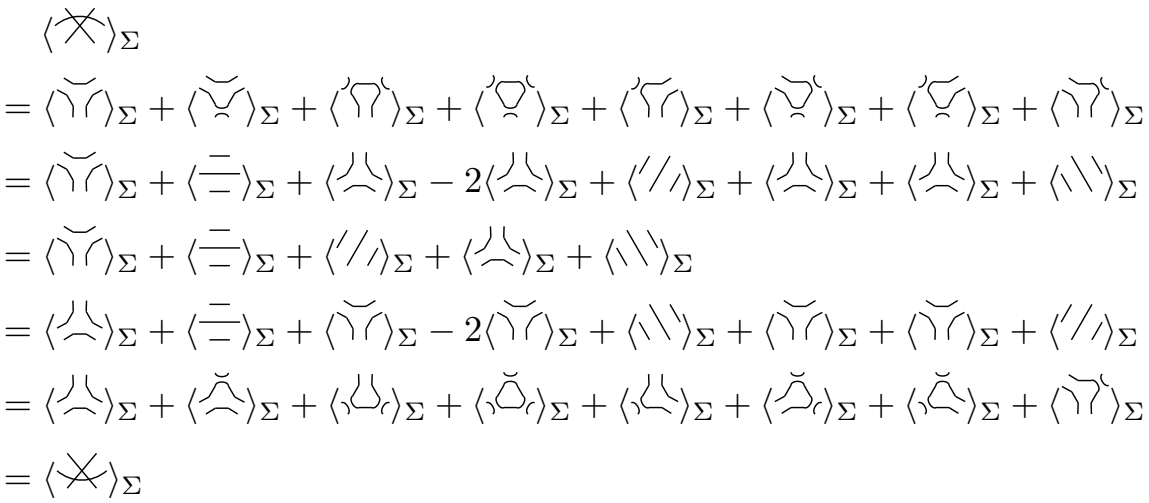
Proof. Let $\omega_D(S^1)$ be the flat knot diagram on Σ. Then the sum of loops in each state S of $\omega_D(S^1)$ is equal to $[\omega_D(S^1)] \in H_1(\Sigma; \mathbb{Z}/2)$. Therefore, if D is checkerboard colorable, then $k(S) > 0$ for each state. If D is not checkerboard colorable, then $r(S) > 0$ for each state. □

Note that when $z|J_D$, the flat Jones-Krushkal polynomial has zero constant term, i.e., $J_D(0) = 0$.

Now we show $J_D(z)$ is an invariant under FR3 moves in Figure 2.10 for flat diagrams on the same surface.

Proposition 5.3. *Let D, D' be two flat diagram on surface Σ, if D, D' are related by one FR3 move, then $J_D(z) = J_{D'}(z)$.*

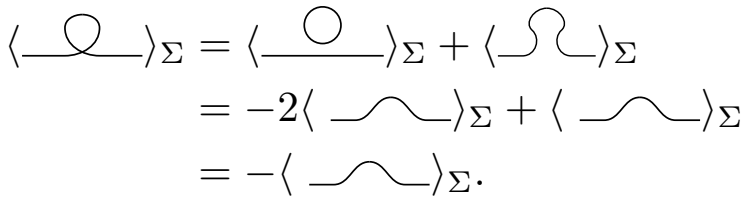
Proof. The calculation details are given below. □



Proposition 5.4. *Let D, D' be two minimal genus diagrams of flat knot α. Then $J_D(z) = J_{D'}(z)$.*

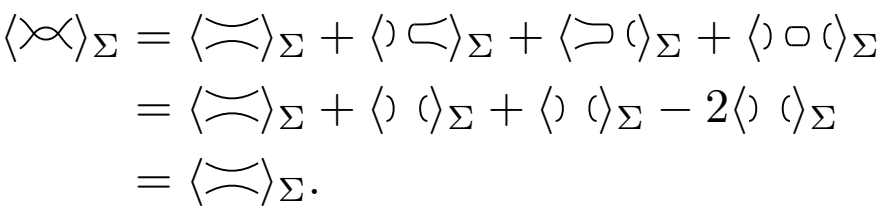
Proof. By [IMN11, Theorem 3.2], any two minimal genus diagrams are related by homotopy on the surface, namely, the FR1,FR3-moves and the FR2 moves that do not change the genus.

We can check that the FR1 move changes the sign of the homological Kauffman bracket:



Since the FR1-move also changes the number of the crossings by one, we conclude that the flat Jones-Krushkal polynomial does not change under the FR1 move.

For the FR2 move, we can calculate the homological Kauffman bracket as below.



Therefore, the flat Jones-Krushkal polynomial is an invariant for minimal genus diagrams of flat knots. □

Definition 5.5. The *flat Jones-Krushkal polynomial* $J_\alpha(z)$ of flat knot α is defined as $J_D(z)$ where D is a minimal genus diagram of α.

The minimal genus is essential for $J_D(z)$ to be well-defined. For example, both the Gauss diagrams below represent the unknot. But the left one has $J_D(z) = 1$ and the right one has $J_{D'}(z) = 2z + 2$.

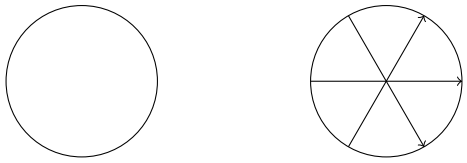


Figure 5.2: Two Gauss diagrams of the unknot

Corollary 5.6. *Let D, D' be two minimal crossing Gauss diagrams of the same flat knot α. Then $J_D(z) = J_{D'}(z) = J_\alpha(z)$.*

Proof. By [IMN11, Corollary 3.1], a minimal crossing Gauss diagram of α achieves its minimal flat genus, and two minimal crossing diagrams D, D' of the same flat knot are related by a finite sequence of FR3 moves. Therefore, their flat Jones-Krushkal polynomials satisfy $J_D(z) = J_{D'}(z)$. □

In [FKI], each flat knot type is represented by a minimal crossing Gauss diagram. One can directly use that Gauss diagram to calculate the flat Jones-Krushkal polynomial.

Note that by [BR21], a minimal crossing flat link diagram also achieves its minimal flat genus. In Section 1.3 we referred to the results that collectively showed monotonicity hold for nonparallel flat links. Therefore, one can calculate the flat Jones-Krushkal polynomial of nonparallel flat links using minimal crossing diagrams.

5.2 Matrix-based calculation

One can compute the flat Jones-Krushkal polynomial by drawing all the states on the Carter surface realizing a minimal diagram of α; refer to [BK22] for detailed calculations on surfaces. In this section, we give a matrix-based algorithm to calculate the flat Jones-Krushkal polynomial, which is easy to code up and more time-efficient even for hand calculation.

In Definition 3.4, we defined the based matrix of a representation $\omega_D : S^1 \looparrowright \Sigma$ of a Gauss diagram D. The set $\bar{G}$ in Definition 3.4 is a generator set of $H_1(\Sigma; \mathbb{Z}/2)$; see [Tur04, Section 4.2]. Notice that the set $\bar{G}$ may contain redundant elements. Let $L \subset \bar{G}$ be a reduced generating set of $H_1(\Sigma; \mathbb{Z}/2)$, and with respect to which we obtain a mod 2 based matrix, say M.

Now consider a parallel copy $\omega_{D'}(S^1)$ of the $\omega_D(S^1)$ as in Figure 5.3. Assume $\omega_D(S^1)$ has n crossings, then so does $\omega_{D'}(S^1)$. The crossings cut $\omega_{D'}(S^1)$ in to $2n$ arcs, denoted by $\mathrm{arc}(D')$. Every arc in $\mathrm{arc}(D')$ has an intersection number with the loops in L.

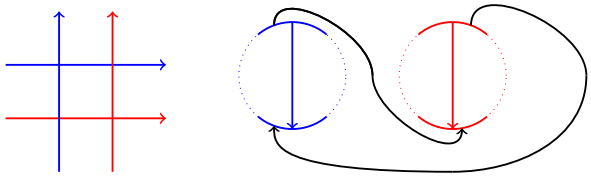
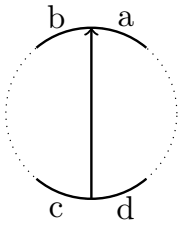


Figure 5.3: A parallel copy (blue) of $\omega_D(S^1)$ (red)

There are two types of smoothings: the oriented and non-oriented smoothings in Figure 5.1. Then at each arrow on a Gauss diagram we can apply the smoothing accordingly. If the smoothing in Figure 5.4 is oriented then the arcs a,d are connected and so are the arcs b,c. If the smoothing is non-oriented, then the arcs a,c are connected and so are the arcs b,d.

We obtain a set of loops for each state S by applying smoothing at all crossings accordingly. Each loop consists of some arcs in $\mathrm{arc}(D')$. Then we can get the intersection numbers of these loops with L. This gives a mod 2 matrix N representing these loops in terms of the dual set of L. Then $r(S) = \mathrm{rank}(N)$, $k(S) = \mathrm{nullity}(N)$.



Oriented smoothing → (a,d),(b,c).

Non-oriented smoothing → (a,c),(b,d).

Figure 5.4: An arrow of a Gauss diagram

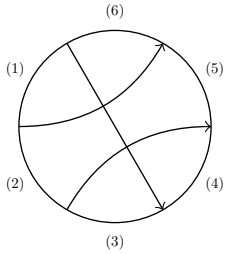


Figure 5.5: Six arcs on the minimal diagram of 3.1

For example, the flat knot 3.1 in Figure 4.6 has flat genus 2. Label the arrows from 12 o'clock by e_1, e_2, e_3. Then the minimal diagram represents an immersion in a genus 2 surface Σ, so L contains 4 elements: the core element and the loops starting and ending at each crossing. The mod 2 intersection matrix is

$$M = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

Then we label the arcs $(1), \ldots, (6)$ as shown in Figure 5.5. Make the parallel copy as in Figure 5.3. Then the mod 2 intersection number between arcs and the generator loops follows similarly the way to calculate the based matrix: the intersection number is given by the number of arrows starting in the arc and ending in the loop minus the number of arrows starting in the loop and ending in the arc. For example, in Figure 5.6, applying the rule described in Figure 5.3 gives that there is only one arrow starting from and no arrow ending in the arc (1) and the arrow starting from (1) does not end in the loop $[e_1^+]$. Therefore, the mod 2 intersection number between arc (1) and $[e_1^+]$ is 0.

Then the mod 2 intersection numbers between these arcs and the generating loops

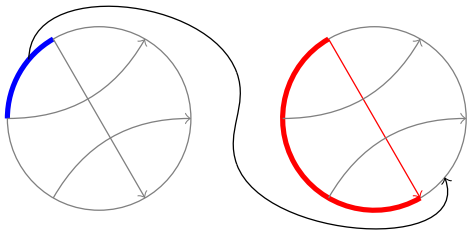


Figure 5.6: The mod 2 intersection number between arc (1) and $[e_1^+]$ is 0.

is given by

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Denote the oriented and non-oriented smoothing as "0" and "1", respectively. Recall that in Figure 5.4, we illustrated how to obtain the loops in S as sum of the arcs. For example for the state 011, the arcs $(1), (3), (5)$ and arcs $(2), (4), (6)$ form two loops. Then the sum of the $1, 3, 5$-th rows gives the mod 2 intersection number $[0, 1, 0, 1]$ with $s, [e_1^+], [e_2^+], [e_3^+]$. Similarly, $2, 4, 6$ rows give $[0, 1, 1, 0]$.

Follow the same manner, we obtain that for state 000, 001, 010, 011, 100, 101, 110, 111, the loop are represented in terms of dual generators by

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix},$$

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}.$$

Therefore, we obtain $r(S) = 2, k(S) = 0$ for state 000. $r(S) = 1, k(S) = 0$ for state 001. $r(S) = 1, k(S) = 0$ for state 010. $r(S) = 2, k(S) = 0$ for state 011. $r(S) = 2, k(S) = 0$ for state 100. $r(S) = 1, k(S) = 0$ for state 101. $r(S) = 1, k(S) = 0$ for state 110. $r(S) = 1, k(S) = 0$ for state 111.

Then the flat Jones-Krushkal polynomial $J_{3.1} = -3z^2 - 5z$.

We wrap up the process of calculation as below.

Algorithm 5.7 (Calculation of $J_D(z)$)**.** The input is a flat knot Gauss diagram D. The output is a polynomial in $\mathbb{Z}[z]$.

1. Calculate L.
   - Calculate the mod 2 based matrix T_D of D.

- Find the subset $L \subset \bar{G}$ generating $H_1(\Sigma; \mathbb{Z}/2)$.

2. Calculate A.
   - Name the arcs as in Figure 5.5.
   - Calculate the intersection matrix A of the arcs and L.
3. Let $J = 0$ and for each state S:
   - For each state apply the rule depicted in Figure 5.4.
   - For each state S, produce a new matrix N whose rows (one per loop in S) are obtained by summing the rows of A corresponding to the arcs in each loop.
   - Calculate $r(S) = \text{rank}_{\mathbb{Z}/2}(N)$ and $k(S) = |S| - r(S)$.
   - Add $(-2)^{k(S)} z^{r(S)}$ to J.
4. $J_D(z) = (-1)^{cr(D)} J$.

Proposition 5.8. *Let D be a flat Gauss diagram obtained from another flat Gauss diagram D' by flipping some arrows (i.e., chords). If D, D' have the same mod 2 based matrix, then $J_D(z) = J_{D'}(z)$.*

Proof. As shown above, the adjacency of the arcs does not change following the rule in Figure 5.4. If the mod 2 based matrix stays the same, flipping arrows in the Gauss diagram does not change the flat Jones-Krushkal polynomial. □

5.3 Normalization and enhancement

In this section, we present normalized and enhanced versions of the flat Jones-Krushkal polynomial. The enhancement is a stronger invariant, and it keeps track of the number of homologically nontrivial loops in each state. We also discuss the properties of these invariants.

Definition 5.9. The *normalized flat Jones-Krushkal polynomial* of flat knot α diagram D on a closed surface Σ is defined by

$$\bar{J}_\alpha(z) = \frac{J_D(z)}{\varepsilon_D},$$

where D is a minimal diagram of α, $\varepsilon_D = -2$ if $[\omega_D(S^1)] = 0 \in H_1(\Sigma; \mathbb{Z}/2)$, and $\varepsilon_D = z$ otherwise.

Then the flat Jones-Krushkal polynomial $\bar{J}_{3.1}(z) = -3z - 5$.

Proposition 5.10. *For any given flat knot α, the normalized flat Jones-Krushkal polynomial satisfies $\bar{J}_\alpha(-2) = 1$.*

Proof. As mentioned in Proposition 4.3, assigning $a = 1$ sends all Jones polynomials to $\sum_S (-2)^{|S|-1} = 1$. Then we have

$$\bar{J}_\alpha(-2) = \sum_S (-2)^{|S|}/(-2) = \sum_S (-2)^{|S|-1} = 1.$$

□

Empirically, we noticed something curious about the roots of $\bar{J}_\alpha(z) - 1$, which is that every almost classical flat knot up to 10 crossings satisfies:

$$\bar{J}_\alpha(-1) = 1.$$

Conjecture 5.11. *For any almost classical flat knot α, the normalized flat Jones-Krushkal polynomial satisfies $\bar{J}_\alpha(-1) = 1$.*

If true, the condition provides a useful criterion for a flat knot to be almost classical. Among all the flat knots up to 7 crossings, only 6 satisfy the condition $\bar{J}_\alpha(-1) = 1$ but are not almost classical.

Definition 5.12. The *enhanced flat Jones-Krushkal polynomial* of a flat knot diagram D realized as immersion on Carter surface of genus g is defined by

$$J^{en}_D(w, z) = (-1)^{cr(D)} \sum_{S \in \mathfrak{S}} \langle D \mid S \rangle_\Sigma \; w^{m(S)},$$

where

$$m(S) = |S| - \#\text{ of null-homologous curves in } S.$$

Note that Algorithm 5.7 and Proposition 5.8 hold for the enhanced flat Jones-Krushkal polynomial accordingly.

Proposition 5.13. *If D, D' are two flat knot diagrams on a surface Σ related by FR3 moves, then $J^{en}_D(w, z) = J^{en}_{D'}(w, z)$.*

Proof. It is enough to verify the claim for two diagrams related by one FR3 move, and the details are similar to the calculation in Proposition 5.3. □

Definition 5.14. The *enhanced flat Jones-Krushkal polynomial* of a flat knot α is defined to be $J^{en}_D(w, z)$, where D is a minimal genus diagram of α.

Take 3.1 as example, $m(S) = 2$ for state 000. $m(S) = 1$ for state 001. $m(S) = 1$ for state 010. $m(S) = 2$ for state 011. $m(S) = 2$ for state 100. $m(S) = 1$ for state 101. $m(S) = 1$ for state 110. $m(S) = 1$ for state 111. Then the enhanced Jones-Krushkal polynomial

$$J^{en}_{3.1}(w, z) = -3w^2z^2 - 5wz.$$

Proposition 5.15. *Let α, β be two flat knots. If $J^{en}_{\alpha}(w,z) = J^{en}_{\beta}(w,z)$, then $J_{\alpha}(w,z) = J_{\beta}(w,z)$.*

Proof. By definition, we have $J^{en}_{D}(1,z) = J_D(z)$. Thus if two knots have the same enhanced flat Jones-Krushkal polynomials, so they must also have the same flat Jones-Krushkal polynomials. □

The converse of Proposition 5.15 is not true. For example, the flat knots 3.1 and 5.1 have $J_{3.1}(z) = J_{5.1}(z) = -3z^2 - 5z$, but their enhanced flat Jones-Krushkal polynomials are not equal:

$$J^{en}_{3.1} = -3w^2z^2 - 5wz,$$
$$J^{en}_{5.1} = (-4w^4 + 6w^3 - 5w^2)z^2 - 5wz.$$

Proposition 5.16. *If α is checkerboard colorable, then $2|J^{en}_{\alpha}$. If α is not checkerboard colorable, then $z|J^{en}_{\alpha}$ and $w|J^{en}_{\alpha}$.*

Proof. Let $\omega_{\alpha}(S^1)$ be the flat knot diagram on Σ. Then the sum of loops in each state S of $\omega_{\alpha}(S^1)$ is equal to $[\omega_{\alpha}(S^1)] \in H_1(\Sigma; \mathbb{Z}/2)$. Therefore, if α is checkerboard colorable, then $k(S) > 0$ for each state. If α is not checkerboard colorable, then $m(S) \geq r(S) > 0$ for each state. □

Definition 5.17. The *normalized enhanced flat Jones-Krushkal polynomial* of a flat knot diagram D realized as immersion on a Carter surface Σ is defined by

$$\bar{J}^{en}_{\alpha}(w,z) = \frac{J^{en}_{D}(z)}{\varepsilon_D},$$

where D is a minimal diagram of α, $\varepsilon_D = -2$ if $[\omega_D(S^1)] = 0 \in H_1(\Sigma; \mathbb{Z}/2)$, and $\varepsilon_D = z$ otherwise.

Proposition 5.15 also holds for the enhanced version, and the converse is not true:

$$\bar{J}^{en}_{3.1} = -3w^2z - 5w,$$
$$\bar{J}^{en}_{5.1} = (-4w^4 + 6w^3 - 5w^2)z - 5w.$$

The following example shows that the flat Jones-Krushkal polynomial of flat knots is not multiplicative under connected sum. The flat knot 4.5 in Figure 4.2 is a connected sum of two diagrams of the unknot. However, the flat Jones-Krushkal polynomial of 4.5 is $-4z + 9$, which is nontrivial.

Problem 5.18. *Is the flat Jones-Krushkal polynomial multiplicative under connected sum at the level of diagrams? I.e. is $J_D(z) = J_{D'}(z) \cdot J_{D''}(z)$ for any $D \in [D' \# D'']$? What about the enhanced flat Jones-Krushkal polynomial, is $J^{en}_{D}(z) = J^{en}_{D'}(z) \cdot J^{en}_{D''}(z)$ for any $D \in [D' \# D'']$?*

5.4 The cabled flat Jones-Krushkal polynomial

In this section, we will define the cabled flat Jones-Krushkal polynomial and give examples to show how it can help to distinguish flat knots.

Recall that the flat arrow polynomial of the 2-strand cable (see Definition 2.30) can be used to distinguish flat knots. Here we use the $(2,1)$ cable for the flat Jones-Krushkal polynomial because the formula to get the $(2,1)$-cabled based matrix is given in Remark 3.22. It is easy to code up the calculation combined with Algorithm 5.7.

Note that the minimal flat genus of α is the same as $\alpha^{2,1}$; see [SW13] for a proof for virtual knots. Making a parallel copy and adding a crossing between the two copies is the same as replacing S^1 in $S^1 \looparrowright \Sigma$ with a Mobius band. This gives us a diagram of $\alpha^{2,1}$ on the minimal genus surface Σ. Following the steps we did for flat knot 3.1, we can calculate $\alpha^{2,1}$.

However, similar to the flat arrow polynomial, the number of states of $\alpha^{2,1}$ is $2^{4\mathrm{cr}(\alpha)+1}$, which increases exponentially when $\mathrm{cr}(\alpha)$ increases.

Referring to [KAT1], a divide-and-conquer strategy can simplify the calculation and reduce the workload for the computer. To boost efficiency, one can choose the order in which the crossings are smoothed so that the arcs are more likely to form loops; see the function "KB1" in [KAT1].

As was done for the flat arrow polynomial, to decrease the workload of calculation, we can only calculate the constant term of the flat Jones-Krushkal polynomial. Applying divide-and-conquer, the calculation of the constant term is much less complex: once we see a nontrivial loop, then the calculation in the branch can be ended. It is worth pointing out that the constant term of the non-normalized flat Jones-Krushkal polynomial of a non-checkerboard colorable flat knot is always zero. However, the $(2,1)$-cable of any flat knot is checkerboard-colorable. Therefore, we can also apply this fast calculation for constant terms of normalized flat Jones-Krushkal polynomials of $(2,1)$-cables.

We give an example to show how cabling the flat Jones-Krushkal polynomial helps to distinguish flat knots. The two flat knots below both have $\bar{J}_{4.1} = \bar{J}_{4.3} = z+3$. but after cabling, we have

$$\bar{J}_{4.1^{(2,1)}} = 36z^2 + 109z + 73,$$
$$\bar{J}_{4.3^{(2,1)}} = 114z^2 + 343z + 229.$$

5.5 Distinguishing flat knots (reprise)

The flat Jones-Krushkal polynomial and its enhancement are powerful tools for distinguishing flat knots when the minimal flat genus is known. For example, the flat knots in Figure 5.8 have $\bar{J}_{7.46142} = -7z^2 - 21z - 13$ and $\bar{J}_{7.46230} = -13z^2 - 39z - 25$,

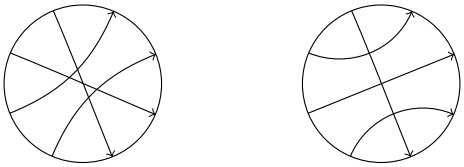


Figure 5.7: Flat knots 4.1 and 4.3 with identical flat Jones-Krushkal polynomial but distinguished by cabled flat Jones-Krushkal polynomial

while other invariants such as u-polynomial, flat arrow polynomial, n-strand cabled flat arrow polynomial and based matrix of the pair are all identical.

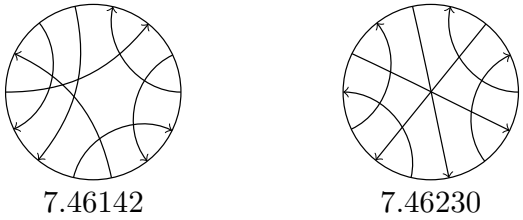


Figure 5.8: Two flat knots not distinguished by cabled arrow polynomials and based matrices

Notice that the pair above are almost classical. We know that all the flat arrow polynomials and cabled flat arrow polynomials of almost classical knots are trivial. Additionally, there are almost classical knots with trivial primitive based matrix. These flat knots are very difficult to separate from the unknot. For example, the flat knot in Figure 4.12 has 8 crossings, and it is the first almost classical knot with trivial primitive based matrix. It can be distinguished from the unknot only by using the flat Jones-Krushkal polynomial, which is given by $\bar{J}_{\text{ac8.19}} = 24z^2 + 72z + 49$.

There are almost classical flat knots having trivial primitive based matrix with higher crossing number. For instance, there are several examples with 10 crossings, but none with 9 crossings. Each such 10-crossing almost classical flat knot can be shown to be nontrivial by computing its flat Jones-Krushkal polynomial.

When combined with other flat knot invariants, the enhanced flat Jones-Krushkal polynomial distinguishes flat knots up to 6 crossings, leaving only 5 pairs of 7 crossing knots not separated, see Table 5.1. The five pairs of non-distinguished 7-crossing flat knots are depicted in Figure 5.9.

Up to 8 crossings, most of the non-distinguished flat knots are permutant and nearly all of them are composite. Indeed, restricting attention to *prime* flat knots, up to 8 crossings, there is only one pair of flat knots with 7 crossings that are not separated by the invariants, namely the two prime flat knots 7.21134 and 7.32153 in Figure 5.9, and the only non-permutant pair.

Crossings	# Flat knots	$A(\alpha^2)$, $C(\alpha^3)$, ϕ	$J^{en}_{\alpha}, A(\alpha^2), C(\alpha^3), \phi$
3	1	0	0
4	11	0	0
5	120	0	0
6	2086	0	0
7	46233	12	10
8	1241291	513	511

Table 5.1: Distinguishing flat knots using $J^{en}_{\alpha}, A(\alpha^2), C(\alpha^3)$, and ϕ-invariants.

For checkerboard colorable knots, the invariants $J^{en}_{\alpha}, A(\alpha^2), \phi$ enable us to distinguish all checkerboard colorable flat knots up to 7 crossings, see Table 5.2. There are four checkerboard colorable flat knots with 8 crossings with the same $J^{en}_{\alpha}, A(\alpha^2), \phi$, see Figure 5.10. Thus, there is a quadruple of 8-crossing checkerboard colorable flat knots that are non distinguished.

Crossings	# Flat knots	$A(\alpha^2), \phi$	$J^{en}_{\alpha}, A(\alpha^2), \phi$
4	1	0	0
5	5	0	0
6	33	0	0
7	347	2	0
8	4451	5	4

Table 5.2: Distinguishing checkerboard colorable flat knots using $J^{en}_{\alpha}, A(\alpha^2), \phi$.

As mentioned in the last chapter, the flat arrow polynomial is trivial for every almost classical flat knot. Thus, the only tools for distinguishing almost classical flat knots are the ϕ-invariant and enhanced flat Jones-Krushkal polynomial. Using J^{en}_{α}, ϕ, we are able to distinguish almost classical knots up to 8 crossings, see Table 5.3. The first almost classical flat knots that are not distinguished are the three pairs of flat knots with 9 crossings shown in Figure 5.11.

Crossings	# Flat knots	ϕ	J^{en}_{α}, ϕ
5	1	0	0
6	1	0	0
7	6	2	0
8	28	1	0
9	190	26	6
10	1682	175	39

Table 5.3: Distinguishing almost classical flat knots using J^{en}_{α}, ϕ

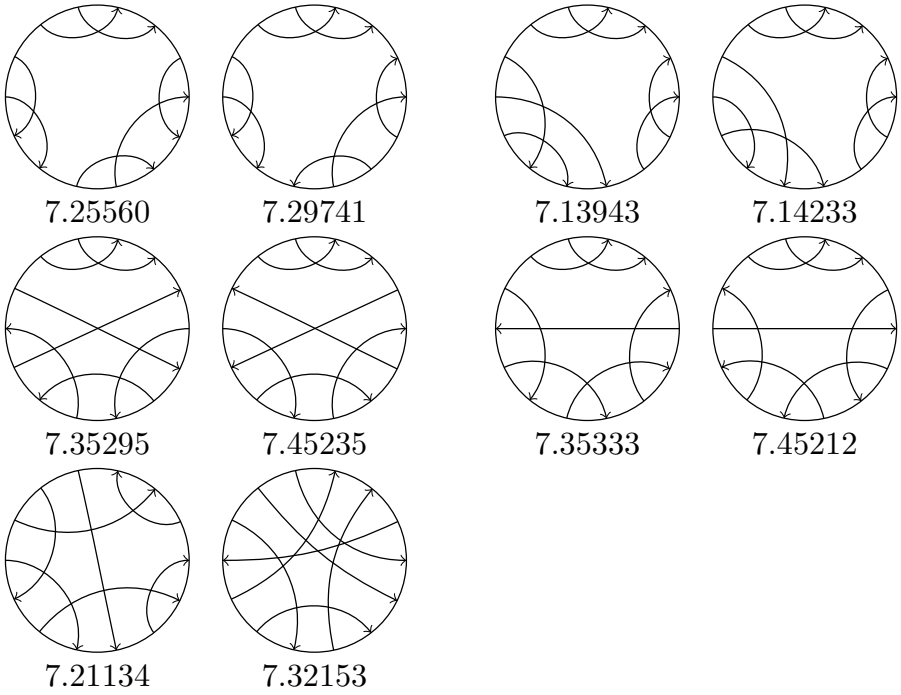


Figure 5.9: Five pairs of flat knots not distinguished

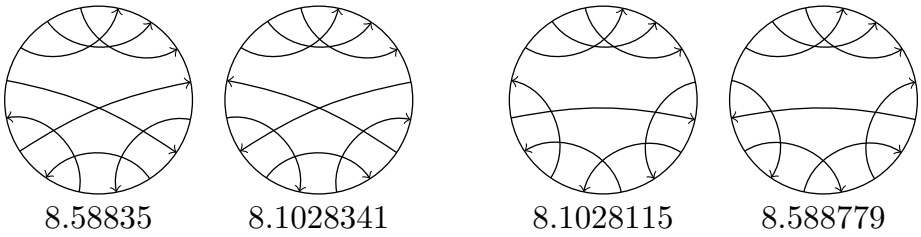


Figure 5.10: Four checkerboard colorable flat knots not distinguished

5.6 Open questions

We conclude this chapter with a few open problems.

Problem 5.19. *Are there bigraded or triply graded invariants that categorify the flat Jones-Krushkal polynomial or its enhancement?*

Problem 5.20. *Can one use the flat Jones-Krushkal polynomial or its enhancement to extract slice obstructions for flat knots?*

Problem 5.21. *Which polynomials can be realized as flat Jones-Krushkal polynomials of flat knots? Which polynomials can be realized as enhanced flat Jones-Krushkal polynomials of flat knots?*

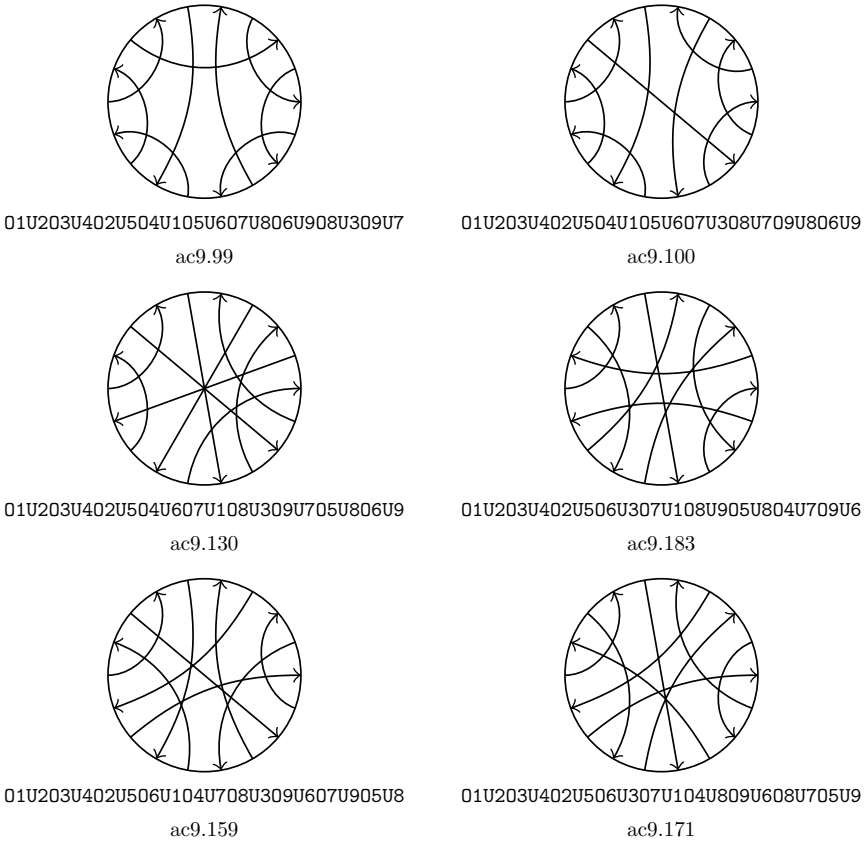


Figure 5.11: Three pairs of almost classical flat knots that are not distinguished.

Problem 5.22. *Referring to* [Mil23]*, does the* homological arrow polynomial *lead to an invariant of flat knots? How powerful is it?*

Chapter 6

The concordance group of flat knots

In this chapter, we introduce long flat knots and use them to construct the concordance group $\mathscr{FC}$ of flat knots. We also introduce graded based matrices and use them to construct the concordance group $\mathscr{FG}$ of based matrices. Given a long flat knot, we can associate a graded based matrix, and this induces a homomorphism $\mathscr{FC} \to \mathscr{FG}$.

We prove two upper bounds of slice genus of flat knots. We construct a non-slice flat link such that each component is an unknot, and pairwise intersection numbers are all zero. At the end, we pose two questions regarding sliceness and discuss possible examples.

6.1 Long flat knots

In this section, we present the definition of long virtual and flat knots and discuss ways to represent them using linear Gauss diagrams and knots on surfaces.

Definition 6.1. An n-tangle planar diagram is a map $\mathbb{R}^1 \times \{1, \dots, n\} \to \mathbb{R}^2$ with only transverse singular points, where outside the square with vertices $(\pm n, \pm n)$, the image of the i-th component coincides with the line $y = i$.

A *long virtual knot diagram* is an oriented 1-tangle planar diagram with only classical crossings and virtual crossings. Two long virtual knot diagrams are said to be *equivalent* if their diagrams are related by finitely many virtual Reidemeister moves and planar isotopy. A *long virtual knot* is an equivalence class of virtual knot diagrams.

Equivalently, we can use a linear Gauss diagram to represent a long virtual knot.

Definition 6.2. A *linear Gauss diagram* consists of an oriented line segment, the linear skeleton, with $2n$ points connected by n arrows. Each arrow is decorated with a sign $(+/-)$.

Definition 6.3. A *long flat knot diagram* is an oriented 1-tangle planar diagram with only flat crossings and virtual crossings. Two long flat knot diagrams are said to be *equivalent* if their diagrams are related by finitely many flat Reidemeister moves and planar isotopy. A *long flat knot* is an equivalence class of flat knot diagrams.

Equivalently, we can use Gauss code or diagram to represent a long flat knot.

Definition 6.4. A *flat linear Gauss diagram* consists of an oriented line segment with $2n$ points connected by n arrows. Along the Gauss diagram by the direction of x-axis, assign number $1, 2, \cdots$ to the arrows. At each arrow head (or tail), the point is recorded as "U" (or "O"), followed by this arrow's assigned number. The *linear flat Gauss code* records every arrow head or tail along the skeleton; see Figure 6.1 for example.

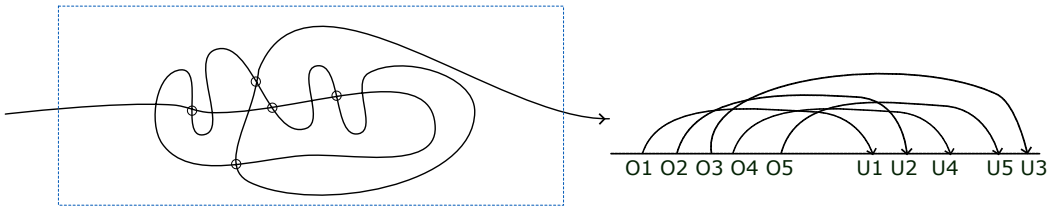


Figure 6.1: A long flat knot diagram with its Gauss diagram and code

Definition 6.5. The *closure* of a long virtual knot is the virtual knot obtained by replacing the curve outside the disk of radius 2 centered at origin in $\mathbb{R}^2$ by $2e^{i\theta}$ for $0 \leq \theta \leq \pi$. The *closure* of a long flat knot is defined similarly. Both operations can be realized in terms of closing Gauss diagrams by taking the closure of the skeleton.

If α is a long flat knot, then we will use $\overline{\alpha}$ to denote its closure. Any invariant of flat knots induces a well-defined invariant of long flat knots by evaluating it on the closure. For example, we can define the u-polynomial of a long flat knot α to be $u_{\overline{\alpha}}(t)$.

Definition 6.6. An *embedding representation* of a long virtual knot K is a proper embedding $e_K : I \to \Sigma^\times \times I$, where $I = [0, 1]$, and $\Sigma^\times$ is the *deleted Carter surface* $\Sigma^\times = \Sigma \setminus D$, where D is a disk and $e_K(0), e_K(1) \in \partial\Sigma^\times \times I = \partial D \times I$. Two embedding representations are *stably equivalent* if they are related by stabilization and destabilization of Carter surfaces and ambient isotopy. The closure $\bar{K}$ of K has an embedding representation $e_{\bar{K}} : S^1 \to \Sigma \times I$, where $e_{\bar{K}}$ extends e_K across $D \times I$, connecting $e_K(0)$ and $e_K(1)$ by a trivial arc in $D \times I$.

Definition 6.7. An *immersion representation* of a long flat knot α is a proper immersion $\omega_\alpha : I \looparrowright \Sigma^\times$, where $I = [0, 1]$, and the *deleted Carter surface* $\Sigma^\times = \Sigma \setminus D$ such that the endpoints $\omega_\alpha(0), \omega_\alpha(1) \in \partial\Sigma^\times = \partial D$. Two immersion representations

are *stably equivalent* if they are related by stabilization and destabilization of Carter surfaces and homotopy on Carter surfaces. The closure $\bar{\alpha}$ of α has an immersion representation $\omega_{\bar{\alpha}} : S^1 \looparrowright \Sigma$, where $\omega_{\bar{\alpha}}$ extends ω_α across D, connecting $\omega_\alpha(0)$ and $\omega_\alpha(1)$ by a trivial arc in D.

Based on the definition above, there are three involutions for long flat knots:

1. The *reverse* $-\alpha$ is given by changing the orientation of I;
2. The *mirror image* α^* is given by changing the orientation of $\Sigma_g^\times$;
3. The *reversed mirror image* $-\alpha^*$.

Note that any involution above is a composition of the other two.

Example 6.8. For long flat knots, there are two types of crossings according to whether the arrow points forward or backward. This leads to a partition $\mathrm{arr}(\alpha) = \mathrm{arr}^+(\alpha) \sqcup \mathrm{arr}^-(\alpha)$. (Note that the subsets $\mathrm{arr}^+(\alpha)$ and $\mathrm{arr}^-(\alpha)$ are not assumed to be proper.)

Using this partition, Turaev introduced the following refinements of the u-polynomial for long flat knots. For $k \geq 1$, set

$$a_k^\pm(\alpha) = \#\{e \in \mathrm{arr}^\pm(\alpha) \mid n(e) = k\} - \#\{e \in \mathrm{arr}^\mp(\alpha) \mid n(e) = -k\} \in \mathbb{Z}.$$

Define $u_\alpha^\pm(t) = \sum_{k\geq 1} a_k^\pm(\alpha) t^k$. Then $u_\alpha^+(t) + u_\alpha^-(t)$ is equal to the u-polynomial of the closure $\bar{\alpha}$. Turaev showed that $u_\alpha^\pm(t)$ are homotopy invariants of long flat knots (see [Tur04, Section 12.2]). The example below illustrates the effect of moving the basepoint on $u_\alpha^\pm(t)$. Each basepoint determines a long flat knot. Since $u^\pm$ are homotopy invariants of long flat knots, it follows that there exist inequivalent long flat knots with equivalent closures.

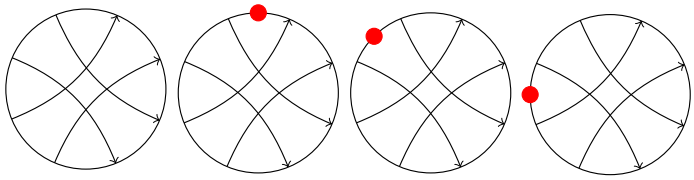


Figure 6.2: Long flat knots with closure 4.4

In addition, for long flat knots, any state will include one long component in addition to several round components. This can be used to refine the flat arrow and Jones-Krushkal polynomials.

6.2 The concordance group of flat knots

In this section, we discuss the connected sum and concordance for long flat knots. Using these notions, we introduce the concordance group $\mathscr{FC}$ of flat knots, cf. Turaev [Tur04].

We begin by defining the connected sum of two long virtual and flat knots. (Recall that connected sum is not well-defined on round flat knots.)

Definition 6.9. The *connected sum* of two long virtual knots K_1, K_2, denoted $K_1 \# K_2$, is the virtual knot obtained by connecting the end point of K_1 to the beginning of K_2. The *connected sum* of two long flat knots α_1, α_2, denoted $\alpha_1 \# \alpha_2$, is the flat obtained by connecting the end point of α_1 to the beginning of α_2.

A long flat knot α is said to be *composite* if it can be represented as the connected sum $D_1 \# D_2$, where neither D_1 nor D_2 is equivalent to α. A long flat knot is *prime* if it is not composite.

Definition 6.10. A flat knot Gauss diagram D is said to be a *connected sum Gauss diagram* if D is the obtained by attaching two nontrivial flat linear Gauss diagram at their ends.

Note that the prime decomposition theorem does not hold for virtual knots; it does hold for $\mathbb{Z}/2$ homologically knots in thickened surfaces, see Matveev [Mat12]. Since connected sum is better behaved on long virtual and flat knots, it is expected that the prime decomposition theorem holds for them. As far as we know, the proof of this result (if true) has not yet been written down.

Next, we define concordance of long flat knots, and here we remind the reader that Definition 2.32 introduces the corresponding notion for round flat knots.

Definition 6.11. A long flat knot α is *slice* if its closure $\bar{\alpha}$ is slice. Two long flat knots α, β are *concordant* if $-\alpha^* \# \beta$ is slice.

For any long flat knot α, Turaev showed in [Tur04] that $-\alpha^* \# \alpha$ is slice; see Figure 6.3 for the gist of the argument. Thus, every long flat knot has an inverse up to concordance.

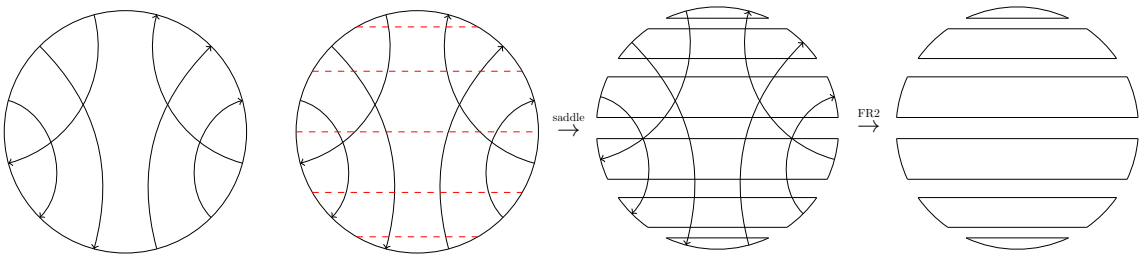


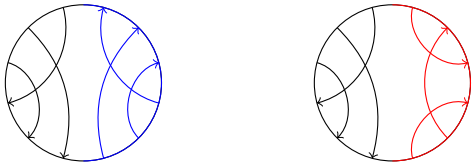
Figure 6.3: Slice movie of $\overline{-\alpha^* \# \alpha}$

Definition 6.12. The concordance group of long flat knots is denoted $\mathscr{FC}$ and consists of concordance classes of long flat knots with group operation given by connected sum.

Since the flat knot 4.4 is slice, all the long knots in Figure 6.2 are slice too. It follows that $u^{\pm}_{\alpha}(t)$ are not invariant under concordance of long flat knots, as claimed in [Tur04, p.2521].

Example 6.13. There exist long flat knots with equivalent closures which are not concordant.

Proof. Consider the permutant knots in Figure 6.4 in $[3.1\#-3.1^*]$. In both diagrams, the black arrows form a long flat knot with closure 3.1. The blue and the red arrows form long flat knots with closure -3.1^*. However, the flat knot 6.1913 on the left is slice by Figure 6.3, whereas the flat 6.1205 has algebraic genus 1, and hence it is not slice. It follows that the two long flat knots formed by the red and blue arrows are not concordant as long flat knots. □



slice knot 6.1913 nonslice knot 6.1205

Figure 6.4: Two permutant diagrams in $[3.1\# - 3.1^*]$

The u-polynomial of long flat knot is defined to be the u-polynomial of its closure.

Proposition 6.14. *The concordance group of long flat knots $\mathscr{FC}$ is not finitely generated.*

Proof. Recall from [Tur04, Section 3.4,5.1,12.3] that the u-polynomial is additive under connected sum and is invariant under concordance. Thus if α, β are long flat knots, then $u_{\alpha\#\beta}(t) = u_{\alpha}(t) + u_{\beta}(t)$, and it induces a homomorphism $\mathscr{FC} \to \mathbb{Z}[t]$. It follows that $\deg u_{\alpha\#\beta}(t) \leq \max\{\deg u_{\alpha}(t), \deg u_{\beta}(t)\}$.

For the sake of contradiction, suppose that $\mathscr{FC}$ is generated by the finite set $\{\alpha_1, \ldots, \alpha_m\}$ and set $n = \max\{\deg u_{\alpha_1}(t), \ldots, \deg u_{\alpha_m}(t)\}$. Then it follows that $\deg u_{\beta}(t) \leq n$ for any $[\beta] \in \mathscr{FC}$. However, by [Tur04, Theorem 3.4.1], an integral polynomial $u(t) \in \mathbb{Z}[t]$ occurs as the u-polynomial of some flat knot if and only if $u(0) = 0 = u'(1)$. In particular, there is a flat knot β with $\deg u_{\beta}(t) > n$, which is a contradiction. □

The concordance group of virtual knots is denoted $\mathscr{VC}$ and defined in terms of concordance classes of long virtual knots with group operation given by connected

sum. The concordance group of classical knots embeds faithfully into $\mathscr{VC}$ [BN17], and the shadow map induces a surjective homomorphism $\mathscr{VC} \to \mathscr{FC}$. Interestingly, $\mathscr{VC}$ is known to be non-commutative [Chr22], but commutativity of $\mathscr{FC}$ is an open question.

Problem 6.15. *Determine whether the concordance group $\mathscr{FC}$ of flat knots is commutative.*

6.3 Concordance of based matrices

In this section, we introduce graded based matrices and use them to construct the concordance group $\mathscr{FG}$ of based matrices.

According to Definition 3.4, a based matrix over an abelian group H is a triple (G, s, b), where G is a finite set, $\bar{G} = \{s\} \sqcup G$, and $b : \bar{G} \times \bar{G} \to H$ is a skew-symmetric map. Next we recall the definitions of a graded based matrix over an abelian group and the directed sum of graded based matrices, two notions originally introduced by Turaev in [Tur04].

Definition 6.16. A *graded based matrix* is a based matrix $T = (G, s, b)$ together with a partition $G = G^+ \sqcup G^-$ into two disjoint subsets. (Neither G^+ nor G^- is assumed to be a proper subset of G.)

Definition 6.17. The *annihilating elements, core elements, and complementary elements* of a graded based matrix $T = (G^{\pm}, s, b)$ are defined as in Definition 3.3. In addition, we require annihilating elements belong to G^+, core elements belong to G^-, and any pair of complementary elements has one in G^+ and the other one in G^-. Thus, following Definition 3.3, we have the definition of *homology* of graded based matrices.

Definition 6.18. Given two graded based matrices $T_1 = (G_1^{\pm}, s_1, b_1)$ and $T_2 = (G_2^{\pm}, s_2, b_2)$ over H, the *directed sum* is the graded based matrix $(G^{\pm}, s, b) = T_1 \boxplus T_2$, where $G^{\pm} = G_1^{\pm} \sqcup G_2^{\pm}$ and the skew-symmetric mapping $b : G \times G \to H$ is defined as follows. For $g \in G_i$ with $i = 1, 2$, set $b(s, g) = b_i(s_i, g)$, $b(g, s) = b_i(g, s_i)$, For any $g \in G_i$, $h \in G_j$ for $i, j \in \{1, 2\}$, set

$$b(g,h) = \begin{cases} b_i(g,h) & \text{if } i = j, \\ \varepsilon(g) b_j(s_j, h) - \varepsilon(h) b_i(s_i, g) & \text{if } i \neq j, \end{cases} \tag{6.1}$$

where $\varepsilon(g) = 1$ if $g \in G^-$ and $\varepsilon(g) = 0$ if $g \in G^+$.

There is a canonical way to map a long flat knot diagram to a graded based matrix: Let D be the associated linear Gauss diagram, then the generator $[e^+]$ associated with arrow $e \in \mathrm{arr}(D)$ is assigned to G^+ if the arrow points from left to

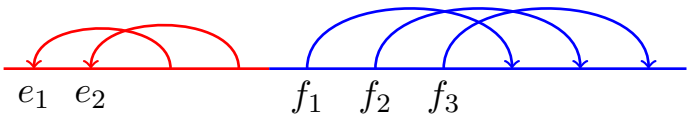


Figure 6.5: Gauss diagram for a long knot

right, and to G^- otherwise. Then one can check that the graded based matrix of the connected sum $D_1 \# D_2$ of two linear Gauss diagrams D_1, D_2 is given by the directed sum $T_1 \boxplus T_2$, where T_1, T_2 are based matrices of D_1, D_2.

For example, the long flat knot in Figure 6.5 is a connected sum of two long flat knots. Let D_1 and D_2 be the flat knot diagrams with the red and blue Gauss diagrams. Then D_1 has graded based matrix

$$T_1 = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

with $G_1^+ = \{[e_1], [e_2]\}$, $G_1^- = \varnothing$, while D_2 has graded based matrix

$$T_2 = \begin{bmatrix} 0 & -2 & 0 & 2 \\ 2 & 0 & 1 & 2 \\ 0 & -1 & 0 & 1 \\ -2 & -2 & -1 & 0 \end{bmatrix}$$

with $G_2^+ = \varnothing$, $G_2^- = \{[f_1], [f_2], [f_3]\}$. Therefore, for any arrow e_i of D_1 and any arrow f_j of D_2, we have $\varepsilon([e_i]) = 1$ and $\varepsilon([f_j]) = 0$, so $b([e_i], [f_j]) = b_2(s_2, [f_j])$. We obtain the graded based matrix for $D_1 \# D_2$:

$$T_1 \boxplus T_2 = \begin{bmatrix} 0 & -1 & 1 & -2 & 0 & 2 \\ 1 & 0 & 1 & -2 & 0 & 2 \\ -1 & -1 & 0 & -2 & 0 & 2 \\ 2 & 2 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ -2 & -2 & -2 & -2 & -1 & 0 \end{bmatrix},$$

where $G^+ = \{[e_1], [e_2]\}$ is associated to the arrows in color red and $G^- = \{[f_1], [f_2], [f_3]\}$ is associated to the arrows in color blue. One can check the based matrix of the round flat knot $\overline{D_1 \# D_2}$ in Figure 3.5 is also $T_1 \boxplus T_2$.

Definition 6.19 ([Tur04])**.** Two graded based matrices $(G_1^\pm, s_1, b_1)$ and $(G_2^\pm, s_2, b_2)$ are said to be *concordant* if their directed sum is null-concordant.

Observe that for any graded based matrix $(G^\pm, s, b)$, we have $(G^\pm, s, b) \boxplus (G^\pm, s, -b)$ is null-concordant.

Definition 6.20. Let $\mathscr{FG}$ be the concordance group of graded based matrices over $\mathbb{Z}$ with group operation given by directed sum $\boxplus$. The concordance inverse of the graded based matrix $(G^{\pm}, s, b)$ is given by $(G^{\pm}, s, -b)$.

The concordance group of based matrices $\mathscr{FG}$ is clearly commutative, cf. Problem 6.15. Given a long flat knot α, we can associate its graded based matrix. This association induces a homomorphism

$$\varphi : \mathscr{FC} \to \mathscr{FG}. \tag{6.2}$$

Turaev proved that two concordant long flat knots have concordant graded based matrices [Tur04]. Further, the graded based matrix of any slice long flat knot is null-concordant. The concordance inverse of long flat knot α is given by $-\alpha^*$, and their based matrices are $T, -T$ respectively. Further, the map f preserves the group structure and hence is a group homomorphism.

Problem 6.21. *Is the homomorphism φ in (6.2) surjective?*

6.4 Sliceness of almost classical flat knots

In this section, we discuss sliceness and algebraic sliceness for almost classical flat knots.

To begin, we define sliceness for flat links.

Definition 6.22. Let ℓ be a flat link with n-components represented by an immersion $\omega_\ell : \bigsqcup_{i=1}^n S_i^1 \looparrowright \Sigma$. Then the flat link ℓ is said to be *slice* if there exists a compact oriented 3-manifold M with $\partial M = \Sigma$ and properly immersed disks $\bigsqcup_{i=1}^n D_i \looparrowright M$ having boundary $\partial D_i = \omega_\alpha(S_i^1)$ for $i = 1, \ldots, n$.

Every almost classical flat knot bounds an immersed Seifert surface, which one can realize as a disk with bands attached as in Figure 6.6. Thus, any almost classical flat knot admits a diagram where the crossings occur in either quadruples of flat crossings or quadruples of virtual crossings as in Figure 6.7.

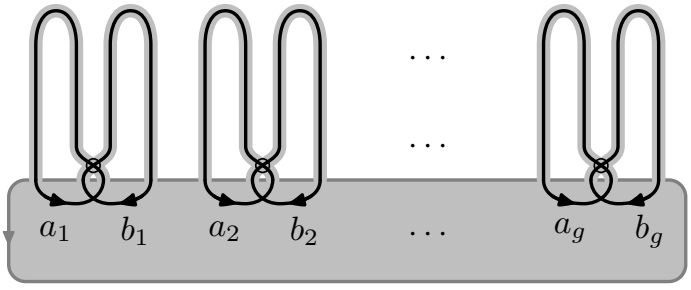


Figure 6.6: Disk-band model

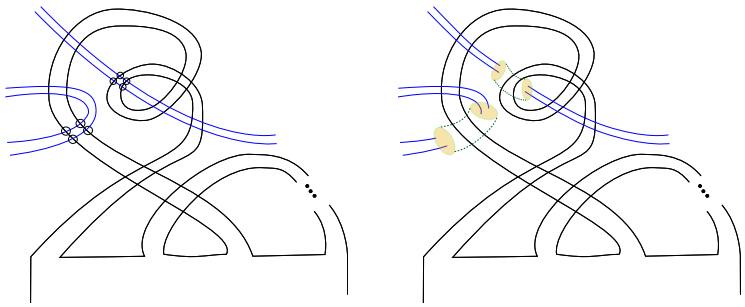


Figure 6.7: Flat Seifert surface (left) and Seifert surface in Carter surface (right)

One can always lift a flat knot diagram to a virtual knot diagram by replacing the flat crossings with classical crossings. If the original flat knot diagram is almost classical, then the virtual knot diagram obtained is also almost classical.

Proposition 6.23. *For every almost classical flat knot α, there exists a virtual knot $K \in \pi^{-1}(\alpha)$, where*

1. *on its virtual knot diagram, double points occur in quadruples, either as band over/under crossings or as band virtual crossings,*
2. *the pair of Seifert matrices corresponding to the disk-band surface for K has all diagonal entries equal to zero.*

Proof. Consider the disk and band model of an immersion. We can choose the Carter surface Σ to have the same orientation as the disk. It suffices to show that each band can be altered so that it has no twists. A full twist of a band is formed by two half twists, corresponding to two flat crossings. A flat crossing can pass both the virtual and flat band crossings by FR3 or FR4 moves, so any two adjacent half twists on a band can be removed by a FR2 move. The zero diagonals can be achieved by adding a full band twist so that the linking number of the generator and the lift of itself is increased or decreased by 1. Repeat this step until the linking number eventually reaches zero. □

Proposition 6.24. *Let α be an almost classical flat knot. If α has a Seifert surface F such that $\{\gamma_i\}_{i=1,\dots,2g(F)}$ generates $H_1(F;\mathbb{Z})$ and $\{\gamma_i\}_{i=1,\dots,g(F)}$ forms a slice link, then $\{\gamma_i\}_{i=1,\dots,g(F)}$ gives a set of surgery curves for a 3-manifold in which α bounds a slice disk.*

Proof. Let $\omega_\alpha(S^1) \looparrowright \Sigma$ be the immersion representation of an almost classical flat knot α. Then $\omega_\alpha(S^1)$ bounds an immersed oriented Seifert surface F of genus g.

Now take the Carter surface Σ as $\Sigma \times \{0\}$ and push if off to $\Sigma \times I$ such that the push-off of $\omega_\alpha(S^1)$ is an embedded loop bounding an embedded Seifert surface F' where by collapsing I to 0, we can regain $\omega_\alpha(S^1)$ and F.

Take an embedded loop $\ell \subset F' \subset \Sigma \times I$ representing a generator of $H_1(F;\mathbb{Z})$. Let $N(\ell)$ be the tubular neighborhood of ℓ. The new 3-manifold $\Sigma \times I \setminus N(\ell)$ has boundary $\Sigma \times \{0,1\} \sqcup \partial N(\ell)$. Take a parallel copy of ℓ on $\partial N(\ell)$ and attach a 2 handle to its collar neighborhood. (On $\partial N(\ell)$, the result can be seen as cutting along the longitude on $\partial N(\ell)$ and attaching two disks.) This surgery gives a 3-manifold bounded by $\Sigma \times \{0,1\} \sqcup S^2$. Attach a ball to S^2 so that we obtain a 3-manifold M bounded by $\Sigma \times \{0,1\}$. Inside M, F is an embedded surface of genus $g-1$. Repeat this step until the genus is reduced to zero which gives the slice disk of α. The slice disk is in a 3-manifold bounded by two copies of Σ. □

Example 6.25. All almost classical flat knots up to 10 crossings are algebraically slice.

Proof. The fillings that give the algebraic sliceness are recorded in [FKI]. □

Problem 6.26. *Is every almost classical flat knot slice? Is every almost classical flat knot algebraically slice?*

We have checked the almost classical knots up to 8 crossings and verified that they are all slice. We do not know whether every almost classical flat knot is slice, or even whether they are all algebraically slice. For example, the flat knot in Figure 6.8 is almost classical, but we do not know whether it is slice or even algebraically slice.

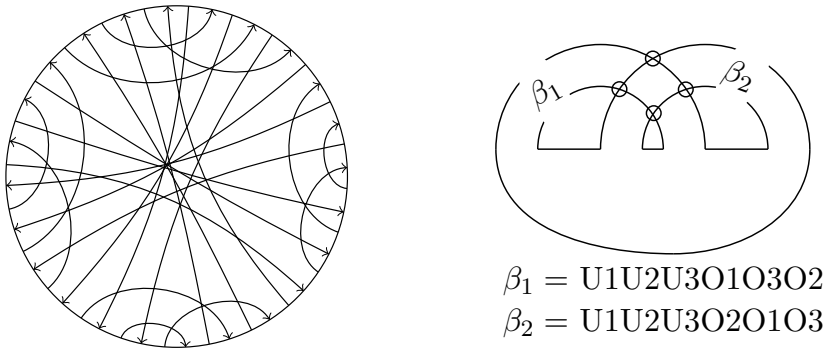


Figure 6.8: A Gauss diagram and disk-band representation of an almost classical knot of 24 crossings

We conclude by presenting two approaches to addressing Problem 6.26 and some examples that shed further light on it.

One approach to answering Problem 6.26 would be to consider the effect of the band-virtualization move in Figure 6.9. If the concordance class were invariant under this move, then that would enable one to show that all almost classical flat knots are slice. Example 6.28 shows however that the concordance class is not preserved under band-virtualization.

Another approach to Problem 6.26 would be to consider *Brunnian flat links*, which are defined to be flat links such that each individual component is unknotted and whose pairwise intersection numbers are all zero. If one could show that all flat Brunnian links were slice, then that would lead to an argument showing that all almost classical flat knots are slice. Example 6.27 shows however that not all flat Brunnian links are slice. Thus Problem 6.26 remains tantalizingly unsolved.

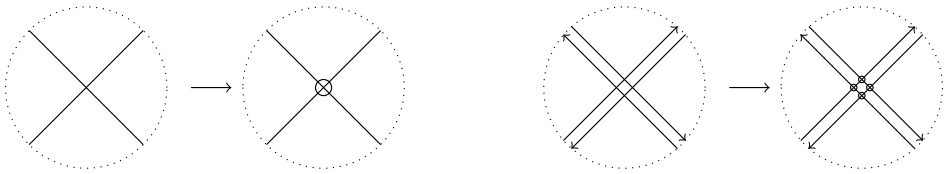


Figure 6.9: Virtualization move and band-virtualization move

Example 6.27. There exists a non-slice flat Brunnian link.

Proof. Consider the flat knot in Figure 6.10-(1), where the band takes the pattern of α. Now on each flat band crossing, assign the virtual crossings as in Figure 6.10-(2) so that we obtain a round flat knot diagram of the closure of $-\alpha\#\alpha$. Apply Reidemeister moves and splitting saddle moves as in Figure 6.10-(2,3,4).

Observe on (4), no flat crossing occurs between the same colored curves. When green components are removed, the blue and red components can be unknotted by Reidemeister moves. Let α be the long flat knot such that its closure is the flat knot 3.1 so that the flat diagram we obtain in Figure 6.10-(2) is the flat knot 6.139; see Figure 6.12. Based on the Gauss diagram of 3.1, we can construct the diagram at the state of (4) as shown in Figure 6.11 where the green curve represents a pair of parallel loops. Note that there are two pairs of parallel loops not drawn in Figure 6.11, which are obtained similarly from band crossings as depicted in Figure 6.10-(2,3,4). Then the flat link diagram depicted in Figure 6.10-(4) has seven components where each component alone is a trivial flat knot and the pairwise intersection number is equal to zero.

However, we know that 6.139 is not slice by computing the algebraic genus of its based matrix, hence the link diagram depicted in Figure 6.10-(4) cannot be slice. □

Example 6.28. The concordance class of a flat knot is not preserved under band-virtualization in Figure 6.9.

Proof. We use the same example taking α as a long flat knot whose closure is 3.1. Then the diagram depicted in Figure 6.10-(3) is not slice. In Figure 6.10-(4), the blue and red curves form a trivial knot diagram. However, the two diagrams can be transformed to each other by one band-virtualization. □

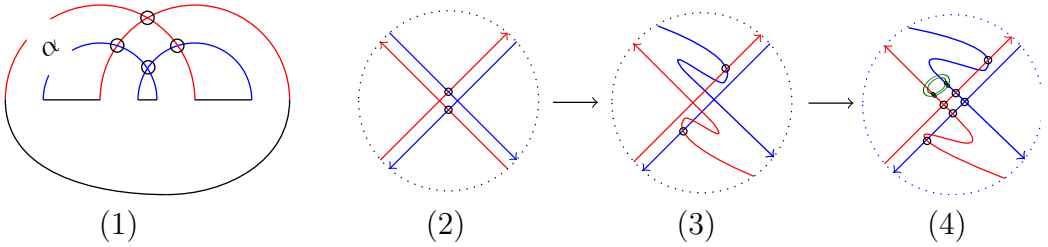


Figure 6.10: A flat concordance movie

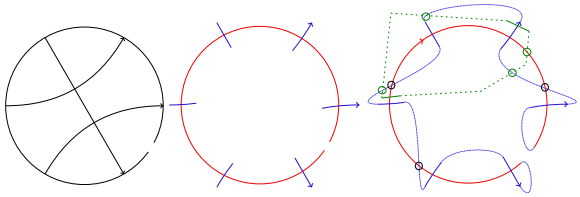


Figure 6.11: Gauss diagram of α determines the diagram of flat link obtained after saddle moves on $\alpha\#(-\alpha)$

6.5 Slice genus and crossing number

In this section, we explore the relationship between the slice genus of a flat knot and its minimal crossing number.

We know that the virtualization move in Figure 6.9 is an unknotting operation of flat knots since a flat knot diagram with only virtual crossings is trivial. There is a metric $v(\alpha)$ on flat knots (and long flat knots), which is defined to be the minimal number of virtualization moves to unknot α. We immediately get that $v(\alpha) < cr(\alpha) - 2$ for flat knots and $v(\alpha) < cr(\alpha) - 1$ for long flat knots, corresponding to the fact that any flat knot with $cr(\alpha) < 3$ is trivial, and any long flat knot with $cr(\alpha) < 2$ is trivial.

Proposition 6.29. *Let α be a long flat knot. Then*

(a) the slice genus $g_s(\alpha) \leq v(\alpha)$, and
(b) the slice genus $g_s(\alpha\#(-\alpha)) \leq cr(\alpha)$.

Proof. We can apply two saddle moves at each classical crossing, one splitting and one joining. Then we obtain a link diagram with only virtual crossings, which is trivial. Therefore, the slice genus $g_s(\alpha) \leq v(\alpha)$. This proves part (a).

Applying combining saddle moves on the diagram in Figure 6.10-(4) on the green loop pair makes the new green component bound a disk and hence can be removed by a death move. There are $cr(\alpha)$ crossings in α so there are as many pairs of crossings in the diagram for $\alpha\#(-\alpha)$ depicted as in Figure 6.10-(4). Since each combining saddle move may increase the genus of the slicing surface by one and after removing all green components we obtain a trivial knot, the slice genus of $g_s(\alpha\#(-\alpha)) \leq cr(\alpha)$. □

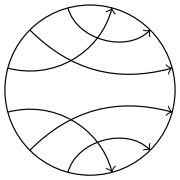


Figure 6.12: Flat knot 6.139 as connected sum $3.1\#(-3.1)$

Let π be the flat projection from virtual knots to flat knots as described in Figure 2.17. It is obvious that if a virtual knot K is slice then its flat projection $\pi(K)$ is also slice. One may ask whether the converse is true.

Problem 6.30. *If α is a flat knot that is slice, then does there exist a virtual knot K with $\pi(K) = \alpha$ that is also slice? What if in addition, we require $cr(K) = cr(\alpha)$?*

This is true for all known slice flat knots up to 6 crossings except for the flat knot 6.540. I have not found a way to unknot 6.540 by saddling the minimal diagram. Though it is algebraically slice, all fillings associated to a minimal diagram fail in giving a slice operation. There are 32 virtual knots of 6 crossings projecting to 6.540, and they are all non-slice. It is also worth mentioning that the *secondary obstruction* (see [Tur04, Section 8.4]) cannot detect the sliceness of 6.540, leaving us longing for more concordance invariants for flat knots. There are several recent papers on concordance invariants of virtual knots, such as [DKK17, BCG20, BCK22, BC21, BK21, BK23]. Those invariants are not yet adapted to flat knots. We do not

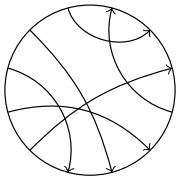


Figure 6.13: Flat knot 6.540

know if 6.540 is slice or not, which raises the following question. For classical knots, any slice knot admits a sequence of births, deaths, and saddle moves such that the number of saddle moves equals the sum of the numbers of births and deaths. Such a sequence is called a *slice movie*. One can similarly give slice movies for flat knots, and so it is natural to ask if there is a slice movie for 6.540?

Furthermore, there are only eight flat knots with 7 crossings whose slice status remains unknown. They are displayed in Figure 6.14. This is the result of filtering out all flat knots that are not algebraically slice, as well as any flat knot that maps to a non-slice flat knot as in Figure 3.8. Then we search for saddle moves using the fillings. This approach was successful in slicing many of the remaining flat knots,

and Figure 6.14 shows the residual set of eight flat knots where this method was inconclusive. Each of these 7 crossings is algebraically slice, but none of them is known to be slice.

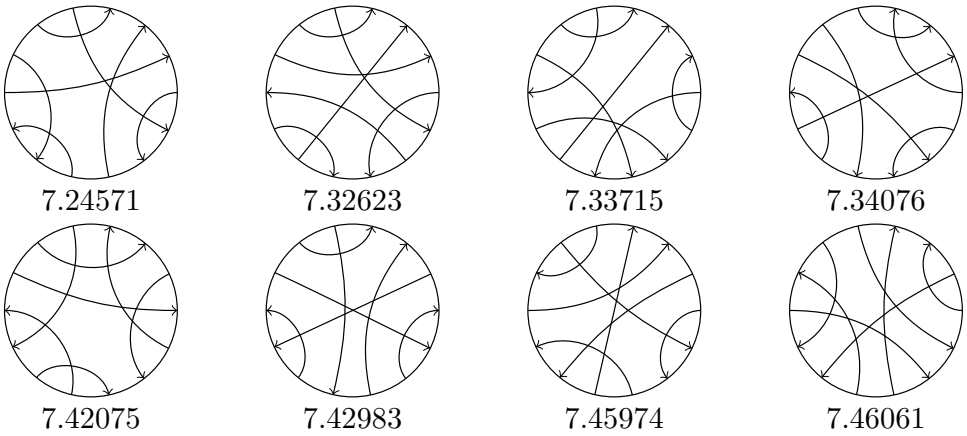


Figure 6.14: Eight flat knots of unknown slice status

Chapter 7

Conclusion

One of the primary goals in this thesis was to develop a better understanding of flat knots, their invariants, and their concordance properties. The list of open questions in [Tur04, Section 13] was a major source of inspiration. In this section, we review some of the questions from [Tur04], and we discuss the progress made on them.

1. *Which matrices can be realized as primitive based matrices for a flat knot?*

   New criteria and a partial solution are given in Remark 3.32.

2. *Can one detect non-slice strings using the secondary obstructions; see* [Tur04, Section 8.4]*?*

   An example of a flat knot that is algebraically slice but not slice is given in Example 3.19.

3. *Classify all strings (flat knots) of small rank (say, ≤ 6) up to homotopy and/or up to concordance.*

   We tabulated flat knots up to 8 crossings, checkerboard colorable flat knots up to 10 crossings, and almost classical flat knots up to 12 crossings as shown in Table 7.1. FlatKnotInfo [FKI] includes all flat knots up to 8 crossings and their pre-computed invariants. The invariants completely distinguish all flat knots up to 6 crossings and leave as unresolved only 5 pairs (see Figure 5.9) of flat knots with 7 crossings.

Along the way, we obtained a number of other results, and here we provide a brief summary. We showed that any minimal diagram of a composite flat knot is a connected sum diagram. By the monotonicity of the algorithm for tabulating flat knots, this enables us to detect composite flat knots. We gave formulas for based matrices of cabled knots and permutant diagrams of almost classical knots. We also developed criteria for the realization of a based matrix, and we gave an example of a

null-concordant based matrix that is not realized by a slice knot, and an example of an almost classical type based matrix that is not null-concordant.

We defined several polynomial invariants of flat knots: the inner/out characteristic polynomial, the flat arrow polynomial, and the flat Jones-Krushkal polynomial and its enhanced version. For the constant term of the flat arrow polynomial, we developed a skein formula and also proved a parity result for checkerboard colorable flat knots. We constructed an example of a non-slice Brunnian flat link.

As a companion to this thesis, we tabulated flat knots up to 8 crossings, checkerboard colorable flat knots up to 10 crossings, and almost classical flat knots up to 12 crossings as shown in Table 7.1. Specifically, the data including the diagram and pre-calculated invariants of flat knots up to 7 crossings are uploaded to [FKI]. The information in the tabulation served to inform and guide the research conducted here. The tabulation implements the algorithm for enumerating flat knots described in Section 1.3, and it is based on a vast number of calculations of flat knot invariants that completely distinguish all flat knots up to 6 crossings. The invariants distinguish flat knots with 7 crossings up to 10 undistinguished flat knots, consisting of 5 ambiguous pairs. For flat knots with 8 crossings, there are a total of 511 undistinguished flat knots. For prime flat knots up to 8 crossings, only two 7-crossing flat knots remain undistinguished.

FlatKnotInfo [FKI] includes diagrammatic information, such as Reidemeister-3 orbit, the symmetry type, and parity projections, as well as flat knot invariants such as the u-polynomial, the flat arrow polynomial, the ϕ-invariant, and the inner/outer characteristic polynomials. It provides concordance information such as algebraic genus and sliceness. It also gives diagrammatic illustrations of the fillings that indicate the saddle moves needed for constructing an oriented surface of minimal genus. In theory, this should allow one to compute the slice genus of any flat knot.

Crossings	# Flat knots	# Checkerboard colorable	# Almost classical
3	1	0	0
4	11	1	0
5	120	5	1
6	2086	33	1
7	46233	347	6
8	1241291	4451	28
9		71404	190
10		1303643	1682
11			18002
12			220849

Table 7.1: The number of tabulated flat knots.

Included in FlatKnotInfo [FKI] is a flat knot calculator, which is a real-time

Gauss code based search tool that determines the minimal diagram and symmetry type of a given flat knot diagram. It also fetches the page for the knot and lists all the diagrammatic information and flat knot invariants.

The flat knot table includes a cross-reference to Green's table of virtual knots [Gre04]. This allows one to easily compare information and invariants between flat and virtual knots.

The backbone of FlatKnotInfo is a suite of programs that compute flat knot invariants and compare them for the representative flat knots in the tabulation. The appendix below contains two sample programs, both written in python. One of them computes the based matrix of a flat knot and the other its flat Jones-Krushkal polynomial.

This thesis contains many open problems, and we provide a brief summary of them here. Problem 2.14 served as a starting point for our investigation of a number of different flat knot invariants, including based matrices, ϕ-invariants, characteristic polynomials, flat arrow polynomials, and flat Jones-Krushkal polynomials. For each of them, we considered the question of realization, which is the problem of finding necessary and sufficient conditions for the invariants to occur, see Problems 3.29, 3.14, 4.20, and 5.21. Answering the realization problem is an important step for potentially converting questions about flat knots into purely algebraic questions. There is also the possibility of finding even more powerful invariants through categorification, and Problems 4.18 and 5.19 ask specifically about categorification for the flat arrow polynomial and the flat Jones-Krushkal polynomials, respectively. Problem 5.22 concerns a proposed invariant of flat knots derived from the homological arrow polynomial of Miller [Mil23], which is a virtual knot invariant obtained as a powerful refinement of the arrow polynomial that encodes homological information as is done in the Jones-Krushkal polyomial. Problem 3.15 asks for a geometric interpretation of the characteristic polynomials, analogous to the Alexander matrix in classical knot theory.

This thesis also discusses sliceness and concordance of flat knots, and again there are many interesting questions and open problems. One overriding problem is to use the flat knot invariants studied here to discover new obstructions to sliceness for flat knots. (See Problems 5.20 and 4.19.) Progress here could have important applications to the concordance group $\mathscr{FC}$ of flat knots: Is it commutative? Does it contain torsion? Are either of the flat knots in Figures 6.13 and 6.14 slice? Note that, if the flat knot 6.540 in Figure 6.13 is slice, then that would give a negative answer to Problem 6.30 relating sliceness of a flat knot to that of virtual knots lying over it.

Another potential application is toward addressing Problem 6.26, which asks whether every almost classical flat knot is slice. For instance, perhaps one can use new invariants to find an example of almost classical flat knot that is not slice, assuming such examples exist. On the other hand, if no such examples exist, then that would indicate that the based matrix in Example 3.30 is not realizable, at least

not from almost classical flat knots. This would suggest that $\mathscr{FC} \to \mathscr{FG}$ is not surjective (cf. Problem 6.21). A positive answer to Problem 6.26 could also lead to progress on Problem 6.15.

We conclude this thesis with a few more open problems.

1. Are all flat almost classical knots slice, or can one find an example of a flat almost classical knot that is not slice?
2. Are all checkerboard colorable free knots slice, or can one find an example of a checkerboard colorable free knot that is not slice?
3. Is the concordance group of flat knots commutative?
4. Tabulate flat knots up to concordance.
5. Tabulate free knots, up to equivalence and/or concordance of free knots.

Appendix A

Dictionary

The objects of interest in this thesis are referred to in the literature either as flat virtual knots [Kau99, MI13, Chu13, Dye16] or as virtual strings [Tur04, Tur06, Tur08a, Gib08, Cah17, Fre22]. We will mainly use Kauffman's terminology, but for the benefit of the reader, we have provided dictionary to translate between the two terminologies.

Kauffman's terminology	**Turaev's terminology**
flat knot diagram	virtual string
Reidemeister moves and isotopy	homotopy of virtual string
diagram on Σ	immersion on Σ
long flat knot diagram	open string
(signed) Gauss diagram	arrow diagram
crossing number of flat knot diagram	rank of string
crossing number of flat knot type	homotopy rank
support genus of flat knot diagram	genus of string
flat (virtual) genus	homotopy genus
concordance	cobordism
null-concordant based matrix	hyperbolic based matrix
reverse $-\alpha$	opposite string α^-
mirror image α^*	inverse string $\bar{\alpha}$

Appendix B

Python code for computing based matrices

```
import copy
import itertools as itt
import numpy as np
import pandas as pd
import sympy as sym
t = sym.symbols('t')

def gcode2theta(gcode):
    if '10' in gcode:
        gcode=gcode.replace('10', 'x')
    rank_num = int(len(gcode)/4)
    assert rank_num == len(gcode)/4
    # now we make a list for tails:
    tails = []
    for i in range(1,rank_num+1):
        if i < 10:
            tails.append(int(gcode.rindex("O%i" %i)/2+1))
        elif i ==10:
            tails.append(int(gcode.rindex("O%s" %'x')/2+1))
    # now we make a list for heads:
    heads = []
    for i in range(1,rank_num+1):
        if i < 10:
            heads.append(int(gcode.rindex("U%i" %i)/2+1))
        elif i ==10:
            heads.append(int(gcode.rindex("U%s" %'x')/2+1))
    # now calculate theta, which represent the lower triangle of the matrix:
    # firstly, we calculate the first column n(e_i)=B(e_i,s)
    theta = []
    for i in range(0,rank_num):
        # i is 1 less than the arrow number
        if tails[i]<heads[i]:
            substr = gcode[(tails[i]*2): (heads[i]*2-2)]
```

```
        else:
            substr = gcode[(tails[i]*2):]+ gcode[:(heads[i]*2-2)]
        theta.append( substr.count("O")- substr.count("U") )
        # n=#tails-#heads
    # now we calculate B(e_i,e_j), where i>j:
    for j in range(0,rank_num):
        for i in range(j+1,rank_num):
            if tails[i]<heads[i]:
                ei=gcode[(tails[i]*2): (heads[i]*2-2)]
            else:
                ei=gcode[(tails[i]*2):]+ gcode[:(heads[i]*2-2)]
            if tails[j]<heads[j]:
                ej=gcode[(tails[j]*2): (heads[j]*2-2)]
            else:
                ej=gcode[(tails[j]*2):]+ gcode[:(heads[j]*2-2)]
            dotprod=0
            for m in range(0,len(ei),2):
                if ei[m] == "O" and ("U%s" %ei[m+1] in ej):
                    dotprod += 1
                if ei[m] == "U" and ("O%s" %ei[m+1] in ej):
                    dotprod += -1
            if tails[i]<heads[i]:
                epsilon=int(tails[i]<tails[j]<heads[i])-int(tails[i]<heads[j]<heads[i])
            else:
                epsilon=int(tails[i]<tails[j])+int(tails[j]<heads[i])-\
                int(tails[i]<heads[j])-int(heads[j]<heads[i])
            theta.append(dotprod+epsilon)
    assert len(theta)*2 == rank_num*(rank_num+1)
    return theta

def theta2matrix(theta):
    """
    input a theta, output is a based matrix(no sorting not necessary primitive
    """
    theta_new=copy.copy(theta)
    rank_num=int(np.sqrt(2*len(theta)))+1
    k=0
    i=rank_num
    theta_new.insert(k,0)
    while k<int(rank_num*(rank_num+1)/2)-1:
        k += i
        i -= 1
        theta_new.insert(k,0)
    assert len(theta_new)==int(rank_num*(rank_num+1)/2)
    triu = np.triu_indices(rank_num)
    # Find upper right indices of a triangular nxn matrix
    tril = np.tril_indices(rank_num, -1)
    # Find lower left indices of a triangular nxn matrix
    arr = np.zeros((rank_num,rank_num))
    arr[triu] = theta_new # Assign list values to upper right matrix
    arr[tril] = -arr.T[tril] # Make the matrix symmetric
```

```
    return -arr.astype(int)

def inv(input_arr):
    """
    simply calculate the primitive matrix for orientation reversed diagram.
    """
    rank_num=input_arr.shape[0]
    arr= np.zeros((rank_num,rank_num))
    for i in range(1,rank_num):
        arr[i] =input_arr[i]-input_arr[0]
    for i in range(1,rank_num):
        arr[:,i]-=input_arr[:,0]
    arr[0]=-input_arr[0]
    arr[:,0]=-input_arr[:,0]
    return arr.astype(int)

def cal_matrix(gcode):
    """
    Gauss code to all matrices and phi
    """
    theta=gcode2theta( gcode)
    arr=theta2matrix( theta)
    phi = min_phi(primitive(arr))
    prim_matrix = theta2matrix(phi)
    inv_phi = min_phi(inv(prim_matrix))
    bar_phi = min_phi(-inv(prim_matrix))
    invbar_phi = min_phi(-(prim_matrix))
    sym_type=[0,0,0]
    if inv_phi == phi:
        if invbar_phi!=bar_phi:
            print('something went wrong!',phi,inv_phi,bar_phi,invbar_phi)
        assert invbar_phi==bar_phi
        sym_type[0]=1
    if bar_phi == phi:
        if invbar_phi!=inv_phi:
            print('something went wrong!',phi,inv_phi,bar_phi,invbar_phi)
        assert invbar_phi==inv_phi
        sym_type[1]=1
    if invbar_phi == phi:
        if bar_phi!=inv_phi:
            print('something went wrong!',phi,inv_phi,bar_phi,invbar_phi)
        assert bar_phi==inv_phi
        sym_type[2]=1

    if sym_type==[0,0,0]:
        if bar_phi==inv_phi or bar_phi==invbar_phi or inv_phi==invbar_phi:
            print('something went wrong!',phi,inv_phi,bar_phi,invbar_phi)
        assert bar_phi!=inv_phi and bar_phi!=invbar_phi and inv_phi!=invbar_phi
    in_poly = sym.latex(
            sym.Matrix(prim_matrix[1:,1:]).charpoly(t)
            ).split("left( ")[1].split(", ")[0]
```

```
    out_poly = sym.latex(
            sym.Matrix(prim_matrix).charpoly(t)
            ).split("left( ")[1].split(", ")[0]
    is_prim = (arr.size==prim_matrix.size)
    assert sum(arr[0])==0
    assert sum(prim_matrix[0])==0

    return pd.Series({
        "bsMtxNoSrt": arr,
        "primMatrix": prim_matrix,
        "isPrim": is_prim,
        "phi": phi,
        "phi_sym": min([phi, inv_phi, bar_phi , invbar_phi]),
        "inPoly": in_poly,
        # "outPoly": out_poly,
        "inv_phi" : inv_phi ,
        "bar_phi" : bar_phi ,
        "invbar_phi" : invbar_phi ,
        "sym_type" : sym_type
        })

def sort_2nd_deep(arr):
    if arr.shape[0]==1:
        print("caution of size!")
        return [0]
    vals = arr[:,0][1:]
    ind=range(1,len(vals)+1)
    pdata=pd.DataFrame({'vals':vals,'ind':ind})
    indnodup=list(set(vals) )
    indnodup.sort()
    possible_sub=[[]]
    # now deal with P_1
    perms1=list(itt.permutations(pdata[pdata.vals==indnodup[0]]['ind'].tolist()))
    min_vector=arr[perms1[0][1:],perms1[0][0]]
    min_list=[perms1[0]]
    if len(perms1)>1:
        for perm in perms1[1:]:
            if arr[perm[1:],perm[0]].tolist()<  min_vector.tolist():
                min_list=[perm]
                min_vector=arr[perm[1:],perm[0]]
            elif arr[perm[1:],perm[0]].tolist() == min_vector.tolist():
                min_list.append(perm)
    p1list= list(set([ list(perm) [0] for perm in min_list]))
    possible_sub =np.array(min_list )
    # now deal with P_i
    for candidate in indnodup[1:]:
        permsi=list(itt.permutations(pdata[pdata.vals==candidate]['ind'].tolist()))
        min_vector=arr[permsi[0],p1list[0]]
        min_list=[a+list(permsi[0])
            for a in possible_sub[possible_sub[:,0]==p1list[0]].tolist()]
        if len(permsi)>1 or len(p1list)>1:
```

```
            for perm in permsi:
                for p1 in p1list:
                    if arr[perm ,p1].tolist()<  min_vector.tolist():
                        min_list= [ a+list(perm)
                            for a in possible_sub[possible_sub[:,0]==p1].tolist()]
                        min_vector=arr[perm ,p1]
                    elif arr[perm ,p1].tolist() == min_vector.tolist():
                        min_list += [a+list(perm)
                            for a in possible_sub[possible_sub[:,0]==p1].tolist()]
        p1list= list(set([ list(perm)[0] for perm in min_list]))
        possible_sub =np.array(min_list )
    return possible_sub


def min_phi(arr ):
    """
    input is a matrix, some orderings
    """
    if arr.shape[0]==1:
        return [0]
    possible_sub=sort_2nd_deep(arr)
    candidate=[]
    for sort_wanted in possible_sub.tolist():
        new_arr=arr[[0]+sort_wanted][ :,[0]+sort_wanted]
        phi = -new_arr[np.triu_indices(new_arr.shape[0],+1)]
        candidate.append(phi.tolist())
    return min(candidate)


def primitive(base_matrix):
    """
    input any based matrix
    output a primitive based matrix
    """
    len0=base_matrix.shape[0]
    len1=0
    m1=base_matrix
    while len1<len0:
        len0=m1.shape[0]
        m1=primitive1(m1)
        m1=primitive2(m1)
        m1=primitive1(m1)
        m1=primitive2(m1)
        len1=m1.shape[0]
    return m1

def primitive1(base_matrix):
    """
    R1 type elementary reduction
    """
    m2=base_matrix
```

```
    if len(np.where(
        ~base_matrix[:].any(axis=0)
        )[0])>1 and 0 in np.where( ~base_matrix[:].any(axis=0))[0]:
        m2 = np.delete(
                np.delete(
                    base_matrix, np.where(~base_matrix[:].any(axis=0))[0][1:],axis=0),
                np.where(~base_matrix[:].any(axis=1))[0][1:],
                axis=1)
    elif len(np.where(
        ~base_matrix[:].any(axis=0)
        )[0])>0 and not 0 in np.where( ~base_matrix[:].any(axis=0))[0]:
        m2 = np.delete(
                np.delete(
                    base_matrix, np.where(~base_matrix[:].any(axis=0))[0][:],axis=0),
                np.where(~base_matrix[:].any(axis=1))[0][:],
                axis=1)
    m3 = np.delete(
        np.delete(m2, np.where( np.all(m2[:]==m2[0],axis=1))[0][1:],axis=0),
        np.where( np.all(m2[:]==m2[0],axis=1))[0][1:],
        axis=1)
    return(m3)

def primitive2(base_matrix):
    """
    R2 type elementary reduction
    """
    m3=base_matrix
    for i in range(1,m3.shape[0]):
        for j in range(i+1,m3.shape[0]):
            if np.all(m3[i]+m3[j]==m3[0]):
                return(np.delete(np.delete(m3, [i,j],axis=0), [i,j], axis=1))
    return(m3)

def cal_filling(ma):
    """
    get the genus and the fillings
    """
    assert type(ma)==np.ndarray
    recordRank=ma.shape[0]
    mFilling=[]
    for par in reversed(sorted(fastpartitions(list(range(1,ma.shape[0]))))):
        ma1=copy.copy(ma)
        toBeDeleted=[]
        for item in par:
            if len(item)>1:
                ma1=np.append(ma1,
                        np.expand_dims((ma1[list(item)[0],:] +ma1[list(item)[1],:]),0),
                        axis=0)
                ma1=np.append(ma1,
                        np.expand_dims((ma1[:,list(item)[0]] +ma1[:,list(item)[1]]),1),
                        axis=1)
```

```
                toBeDeleted+=list(item)
            ma1=np.delete(np.delete(ma1,toBeDeleted,axis=0),toBeDeleted,axis=1)
            if np.linalg.matrix_rank(ma1)<recordRank:
                mFilling=[par]
                recordRank=np.linalg.matrix_rank(ma1)
            elif np.linalg.matrix_rank(ma1)==recordRank:
                mFilling.append(par)
        assert recordRank %2 == 0
        genus=int(recordRank/2)
        return(pd.Series({
            "mFilling(w/Gcode)": mFilling,
            "genus": genus
            }))



def fastpartitions(set_):
    if not set_:
        yield []
        return
    for i in range(int(2**len(set_)/2)):
        if ("{:0%ib}" % len(set_) ).format(i).count('0')<=2:
            parts = [set(), set()]
            for item in set_:
                parts[i%2].add(item)
                i//=2
            for item in fastpartitions(parts[1]):
                # Delete those who have >2 subset:
                #assert len(max([parts[0]]+b, key=len))<=2
                yield [parts[0]]+item
```

Appendix C

Python code for computing flat Jones-Krushkal polynomials

```
import numpy as np
import pandas as pd
import sympy as sym
import networkx as nx
import copy

z=sym.Symbol('z')
t=sym.Symbol('w')


def cal_jk_en_poly( strFlat,mstr ):
    if type(mstr)==np.ndarray:
        is_ac=(~mstr.any(axis=1))[0]
        mod2bsdmtr=np.remainder(mstr,2)
    if type(mstr)==str:
        is_ac=(~np.array(eval(
            mstr.replace('␣␣',',').replace('␣–', ',').replace('\n', ',')
            )) .any(axis=1))[0]
        mod2bsdmtr=np.remainder(eval(
            mstr.replace('␣␣',',').replace('␣–', ',').replace('\n', ',')),2 )
    is_cc = (~mod2bsdmtr.any(axis=1)) [0]
    if '10' in strFlat:
        strFlat=strFlat.replace('10', 'x')
    rank_num = int(len(strFlat)/4)
    assert rank_num == len(strFlat)/4
    # now we make a list for tails:
    tails = []
    for i in range(1,rank_num+1):
        if i < 10:
            tails.append(int(strFlat.rindex("O%i" %i)/2+1))
        elif i ==10:
            tails.append(int(strFlat.rindex("O%s" %'x')/2+1))
```

```
    # now we make a list for heads:
    heads = []
    for i in range(1,rank_num+1):
        if i < 10:
            heads.append(int(strFlat.rindex("U%i" %i)/2+1))
        elif i ==10:
            heads.append(int(strFlat.rindex("U%s" %'x')/2+1))
    arrneg=(np.array(heads)<np.array(tails)).astype(int)
    ht=np.array([sorted(x) for x in list(np.array([heads,tails]).transpose())])
    arcs=[]
    for i in range(rank_num*2):
        if strFlat[i*2]=='O':
            node=heads[int((strFlat[i*2+1] ).replace('x', '10') )-1]
            consider_before0= (  ht[:,0] <=node)*(ht[:,1]>node ).astype(int)+arrneg
            consider_before=np.append(np.array([1]), consider_before0, axis=0)
        else: consider_before = np.zeros(rank_num+1 ,dtype=int)
        if strFlat[(i*2+2)%len(strFlat)]=='U':
            node=tails[int((strFlat[(i*2+3)%len(strFlat)]).replace('x', '10') )-1]
            consider_after0= ( ht[:,0] <node)*(ht[:,1] >=node).astype(int)+arrneg
            consider_after=np.append(np.array([1]), consider_after0, axis=0)
        else: consider_after = np.zeros(rank_num+1 ,dtype=int)
        arcs.append((consider_before+consider_after) %2 )
    arc_int_matrix = np.array(arcs)
    polycoef=[0]*(rank_num*2)
    jkpoly_normed=0
    enpoly=0
    for i in range(2**rank_num):
        stanum=("{:0%ib}" %rank_num).format(i) # 7 is '00111' for 5 crossing knot
        hom_rank,hom_ker,hom_trivial=cal_jk_poly_one_state(
            stanum,tails,heads,rank_num,arc_int_matrix,mod2bsdmtr)
        polycoef[hom_rank]+=(-2)**hom_ker
        jkpoly_normed +=(-2)**(hom_ker-is_cc)*z**(hom_rank-1+is_cc)
        enpoly +=(-2)**hom_ker*z**hom_rank*t**(hom_ker+hom_rank-hom_trivial)
    return(pd.Series({
        "jk_coef_unnorm": polycoef,
        "jk_poly_normed": jkpoly_normed,
        'en_poly_unnorm':enpoly,
        'is_cc':is_cc,
        'is_ac':is_ac
    }) )

def cal_jk_poly_one_state(stanum,tails,heads,rank_num,arc_int_matrix,mod2bsdmtr):
    asplit=[j for j in range(len(stanum)) if stanum.startswith('0', j)]
    bsplit=[j for j in range(len(stanum)) if stanum.startswith('1', j)]
    G=nx.Graph( )
    G.add_nodes_from(range(0,rank_num*2))
    for i in asplit:
        G.add_edge(((heads[i]-2)%(rank_num*2) ),((tails[i]-1)%(rank_num*2)))
        G.add_edge(((heads[i]-1)%(rank_num*2) ),((tails[i]-2)%(rank_num*2)))
    for i in bsplit:
        G.add_edge(((heads[i]-1)%(rank_num*2) ),((tails[i]-1)%(rank_num*2)))
```

```
            G.add_edge(((heads[i]-2)%(rank_num*2) ),((tails[i]-2)%(rank_num*2)))
        arcmatrix=[]
        components=nx.connected_components(G)
        for value in components:
            arclst=[0]*(rank_num*2)
            for i in value:
                arclst[i]=1
            arcmatrix.append(arclst)
        arc_in_gen=np.matmul(np.array(arcmatrix),arc_int_matrix) %2
        arc_sum_int=np.matmul(arc_in_gen,mod2bsdmtr)%2
        hom_rank=sym.Matrix(( arc_in_gen)).rank(iszerofunc=lambda x: x % 2 == 0)
        hom_ker=len(arcmatrix)-hom_rank
        hom_trivial=np.sum(~arc_in_gen.any(1))
        assert hom_rank<10
        return(hom_rank,hom_ker,hom_trivial)


def cal_jk_from_gcode(gcode):
    return cal_jk_en_poly(gcode,theta2matrix(gcode2theta(gcode)))
```

References

[BG+17] Hans U. Boden, Robin Gaudreau, Eric Harper, Andrew J. Nicas, and Lindsay White, *Virtual knot groups and almost classical knots*, Fund. Math. **238** (2017), no. 2, 101–142. ↑18

[BN17] Hans U. Boden and Matthias Nagel, *Concordance group of virtual knots*, Proc. Amer. Math. Soc. **145** (2017), no. 12, 5451–5461. ↑79

[BCG20] Hans U. Boden, Micah Chrisman, and Robin Gaudreau, *Signature and concordance of virtual knots*, Indiana Univ. Math. J. **69** (2020), no. 7, 2395–2459. ↑19, 86

[BC21] Hans U. Boden and Micah Chrisman, *Virtual concordance and the generalized Alexander polynomial*, J. Knot Theory Ramifications **30** (2021), no. 5, 2150030, 35pp. ↑86

[BR21] Hans U. Boden and William Rushworth, *Minimal crossing number implies minimal supporting genus*, Bull. Lond. Math. Soc. **53** (2021), no. 4, 1174–1184. ↑63

[BCK22] Hans U. Boden, Micah Chrisman, and Homayun Karimi, *The Gordon-Litherland pairing for links in thickened surfaces*, Internat. J. Math. **33** (2022), no. 10-11, 2250078, 47pp. ↑86

[BK21] Hans U. Boden and Homayun Karimi, *Concordance invariants of null-homologous knots in thickened surfaces*, 2021. ArXiv 2111.07409, to appear in Comm. Anal. Geom. ↑86

[BK22] ———, *The Jones-Krushkal polynomial and minimal diagrams of surface links*, Ann. Inst. Fourier (Grenoble) **72** (2022), no. 4, 1437–1475 (English, with English and French summaries). ↑60, 63

[BK23] ———, *Mock Seifert matrices and unoriented algebraic concordance*, 2023. ArXiv 2301.05946. ↑86

[Cah17] Patricia Cahn, *A generalization of Turaev's virtual string cobracket and self-intersections of virtual strings*, Commun. Contemp. Math. **19** (2017), no. 4, 1650053, 37pp. ↑3, 92

[Car91] J. Scott Carter, *Closed curves that never extend to proper maps of disks*, Proc. Amer. Math. Soc. **113** (1991), no. 3, 879–888. ↑21

[CFG+20] Zhiyun Cheng, Denis A. Fedoseev, Hongzhu Gao, Vassily O. Manturov, and Mengjian Xu, *From chord parity to chord index*, J. Knot Theory Ramifications **29** (2020), no. 13, 2043004, 26. ↑50

[Chr22] Micah Chrisman, *Milnor's concordance invariants for knots on surfaces*, Algebr. Geom. Topol. **22** (2022), no. 5, 2293–2353. ↑79

[Chu13] Karene Chu, *Classification of flat virtual pure tangles*, J. Knot Theory Ramifications **22** (2013), no. 4, 1340006, 17pp. ↑3, 92

[CKS02] J. Scott Carter, Seiki Kamada, and Masahico Saito, *Stable equivalence of knots on surfaces and virtual knot cobordisms*, J. Knot Theory Ramifications **11** (2002), no. 3, 311–322. ↑1, 10

[Duv83] Jean-Pierre Duval, *Factorizing words over an ordered alphabet*, J. Algorithms **4** (1983), no. 4, 363–381. ↑13

[Dye16] Heather A. Dye, *An invitation to knot theory*, CRC Press, Boca Raton, FL, 2016. Virtual and classical. ↑49, 92

[DK09] Heather A. Dye and Louis H. Kauffman, *Virtual crossing number and the arrow polynomial*, J. Knot Theory Ramifications **18** (2009), no. 10, 1335–1357. ↑47, 49

[DKM11] Heather A. Dye, Louis H. Kauffman, and Vassily O. Manturov, *On two categorifications of the arrow polynomial for virtual knots*, The mathematics of knots, Contrib. Math. Comput. Sci., vol. 1, Springer, Heidelberg, 2011, pp. 95–124. ↑59

[DKK17] Heather A. Dye, Aaron Kaestner, and Louis H. Kauffman, *Khovanov homology, Lee homology and a Rasmussen invariant for virtual knots*, J. Knot Theory Ramifications **26** (2017), no. 3, 1741001, 57. ↑86

[FKI] Jie Chen, *FlatKnotInfo: Table of Flat Knot Invariants*, https://www.flatknotinfo.com. ↑2, 3, 4, 5, 13, 36, 38, 39, 54, 58, 63, 83, 88, 89

[Fre22] David Freund, *Complexity of virtual multistrings*, Commun. Contemp. Math. **24** (2022), no. 6, 2150066, 10pp. ↑3, 92

[Gib08] Andrew Gibson, *On tabulating virtual strings*, Acta Math. Vietnam. **33** (2008), no. 3, 493–518. ↑2, 3, 30, 92

[GPV00] Mikhael Goussarov, Michael Polyak, and Oleg Viro, *Finite-type invariants of classical and virtual knots*, Topology **39** (2000), no. 5, 1045–1068. ↑1, 8

[Gre04] Jeremy Green, *A table of virtual knots* (2004), http://www.math.toronto.edu/drorbn/Students/GreenJ. ↑4, 5, 90

[HS94] Joel Hass and Peter Scott, *Shortening curves on surfaces*, Topology **33** (1994), no. 1, 25–43. ↑3

[IMN11] Denis P. Il′yutko, Vassily O. Manturov, and Igor M. Nikonov, *Parity in knot theory and graph links*, Sovrem. Mat. Fundam. Napravl. **41** (2011), 3–163 (Russian, with Russian summary); English transl., J. Math. Sci. (N.Y.) **193** (2013), no. 6, 809–965. ↑62

[Jon85] Vaughan F. R. Jones, *A polynomial invariant for knots via von Neumann algebras*, Bull. Amer. Math. Soc. (N.S.) **12** (1985), no. 1, 103–111. ↑47

[Kad03] Teruhisa Kadokami, *Detecting non-triviality of virtual links*, J. Knot Theory Ramifications **12** (2003), no. 6, 781–803. ↑3

[Kam02] Naoko Kamada, *On the Jones polynomials of checkerboard colorable virtual links*, Osaka J. Math. **39** (2002), no. 2, 325–333. ↑52

[KK00] Naoko Kamada and Seichi Kamada, *Abstract link diagrams and virtual knots*, J. Knot Theory Ramifications **9** (2000), no. 1, 93–106. ↑10

[KAT1] Dror Bar-Natan and Scott Morrison, *The Knot Atlas: The Jones Polynomial*, http://katlas.org/wiki/The_Jones_Polynomial. ↑54, 69

[Kar18] Homayun Karimi, *Alternating virtual knots*: *PhD thesis, McMaster University*, 2018. ↑51

[Kau99] Louis H. Kauffman, *Virtual knot theory*, European J. Combin. **20** (1999), no. 7, 663–690. ↑1, 2, 47, 92

[KM11] Peter B. Kronheimer and Tomasz S. Mrowka, *Khovanov homology is an unknot-detector*, Publ. Math. Inst. Hautes Études Sci. **113** (2011), 97–208. ↑1

[KI] Charles Livingston and Allison H. Moore, *KnotInfo: Table of Knot Invariants*, https://www.knotinfo.math.indiana.edu. ↑3

[Kru11] Vyacheslav Krushkal, *Graphs, links, and duality on surfaces*, Combin. Probab. Comput. **20** (2011), no. 2, 267–287. ↑60

[Kup03] Greg Kuperberg, *What is a virtual link?*, Algebr. Geom. Topol. **3** (2003), 587–591. ↑8

[Man10] V. O. Manturov, *Parity in knot theory*, Mat. Sb. **201** (2010), no. 5, 65–110 (Russian, with Russian summary); English transl., Sb. Math. **201** (2010), no. 5-6, 693–733. ↑19

[Man12] Vassily O. Manturov, *Parity and cobordisms of free knots*, Mat. Sb. **203** (2012), no. 2, 45–76. ↑50

[MI13] Vassily O. Manturov and Denis P. Ilyutko, *Virtual knots. The state of the art*, Series on Knots and Everything, vol. 51, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2013. Translated from the 2010 Russian original; With a preface by Louis H. Kauffman. ↑1, 92

[Mat12] Sergei V. Matveev, *Roots and decompositions of three-dimensional topological objects*, Uspekhi Mat. Nauk **67** (2012), no. 3(405), 63–114. ↑2, 77

[Mil23] Kyle A. Miller, *The homological arrow polynomial for virtual links*, J. Knot Theory Ramifications **32** (2023), no. 1, 2350005, 42pp. ↑52, 73, 90

[Pol10] Michael Polyak, *Minimal generating sets of Reidemeister moves*, Quantum Topol. **1** (2010), no. 4, 399–411. ↑12, 14

[SW06] Daniel S. Silver and Susan G. Williams, *An invariant for open virtual strings*, J. Knot Theory Ramifications **15** (2006), no. 2, 143–152. ↑23

[SW13] ———, *Virtual genus of satellite links*, J. Knot Theory Ramifications **22** (2013), no. 3, 1350008, 4. ↑69

[Tur04] Vladimir Turaev, *Virtual strings*, Ann. Inst. Fourier (Grenoble) **54** (2004), no. 7, 2455–2525. ↑2, 19, 20, 23, 28, 29, 30, 31, 34, 35, 36, 37, 38, 39, 44, 46, 63, 76, 77, 78, 79, 80, 81, 86, 88, 92

[Tur06] ———, *Knots and words*, Int. Math. Res. Not. (2006), Art. ID 84098, 23pp. ↑3, 92

[Tur08a] ———, *Cobordisms of words*, Commun. Contemp. Math. **10** (2008), no. suppl. 1, 927–972. ↑50, 92

[Tur08b] ———, *Cobordism of knots on surfaces*, J. Topol. **1** (2008), no. 2, 285–305. ↑19