Alexander polynomial, linking number and algebraic links

Carlos Sández García

Mathematics Bachelor Thesis

Universidad de Zaragoza

Advisors:

Enrique Artal and José Ignacio Cogolludo

28th June 2017

Prologue

Mathematics has a very long tradition in the history of mankind, and many fields have matured magnifically. In fact, it is not uncommon for them to require years of study and understanding of complex machinery to get up to speed with the latest developments. The mathematical study of knots and links, or *Knot Theory*, is a rare breed. Questions are easy to state, in part due to the geometrical nature of the field. Most importantly, the learning curve required is not so steep, and with basic knowledge of topology, algebra and differential geometry, many results can be grasped. The main factors that drew me to choose this topic were simplicity and accessibility, as I read my first ever paper on the topic [19].

Knot theory has been around for a mere 100 years, although the more interesting results have come in the last 50. In part, this is owed to the strong connection with Topology, as Knot theory is a branch of it. While knots have been used since the beginning of civilisation, only the shape of the knots will be of study and this is where topology plays its role. Gauss was the first to actually consider knots mathematically, inventing a useful invariant which will be key in this study, the *linking number*. However, much of the early research was focused on combinatorial approaches and empirical results. It wouldn't be until the next century that the approach would be systematised with the help of algebraic topology. Further milestone achievements in the field would be the discovery of the Alexander polynomial in 1928 and the Jones polynomial in 1984 [2].

Knots have applications in many fields, and although they have motivated development in areas of mathematics, the most telling examples come from the relation with more applied sciences. In biology, topoisomerases are enzymes that manipulate the DNA strands, performing operations akin to those in knot theory. Scientist try to study this operations by using circular DNA and thus obtaining knots. Virus that inject their DNA to the host may introduce further twists and the study of the resulting DNA is simplified in the framework of knot theory. Again in biology, stereoisomers, or molecules with the same graph but arranged differently in space, are the subject of continuous study. The reason is that this arrangement might provide new unknown properties to the molecules. Artificial knotted molecules have been synthesized to study the change in properties. For instance, the chirality, or whether the molecule is equivalent to its mirror image, is specially relevant. In physics, the modelling of statistical mechanics, or systems with many particles, has been related to the combinatorial ways of computing some polynomial knot invariants. It has only been afterwards that fields like this have been able to connect. For instance, the Yang-Baxter equation, fundamental in statistical mechanics, can be represented as one type of move that doesn't change the knot equivalence class. The original motivation by Gauss to study knots must be mentioned as well. He was trying to determine the change in the magnetic field when there were loops in the path of the moving pole.

The objective is to classify a special type of knots called *algebraic knots*. They have a particular relevance because of their connection with algebraic and analytic varieties. There are many strong results for them that allow a thorough study. To expand a little the scope, multiple components will be included as links, as well as the disjoint union of algebraic links. The results will still hold.

The classification will be oriented to telling links apart, rather than tabulating all the union of disjoint algebraic links. To achieve this, a basic set of concepts will be introduced, along with explanations and remarks. Important results that affect the classification will be presented as well, and the procedures to apply the results will be sketched. These algorithms are shown in the appendix, fully implemented in `Sagemath`. Some practical examples of usage will be done in the conclusion, to demonstrate the applicability of the mathematical construct developed. A more detailed summary of this work follows.

In Chapter 1 braids are defined. The braid group B_n with a presentation is given. This is necessary because later on, links will be expressed by a word in the generators of B_n whose closure is that link. Then, knots are introduced, specifying that they have to be tame. Orientability is studied and the trefoil is given as an example. In part, it is to illustrate how to give a regular projection of a knot, which is explained afterwards. In part, it is because the object of our study, algebraic knots, have the property to be invertible. They are also chiral, so chirality is defined as well. Now torus knots are explained, specifying how to represent them with tuples, and what the choice of longitude is. The crossing number is given, as well as the previous assertion that they are reversible. To further expand the concept of torus knots, more components are needed, so links are introduced in order to explain torus links. After that, the way to iterate is presented to finally reach iterated torus links. Some measures to check if a link is an iterated torus link are given, like whether it's hyperbolic or if the crossing number surpasses a given bound. Another way of iterating is given, in the form of satellite knots and cables. An important remark on how to properly define iterated torus links allows for the torus core to be left intact when iterating. With iterated torus links properly developed, it is time to bring algebraic links to the spotlight. They are first expressed in terms of the complex curve, and then shown as iterated torus links with the change of variables specified. The notation for representing them is explained, with the Puiseux and Newton pairs. There are some restrictions for the iterated torus links tuples to originate from algebraic knots. Some of the less intuitive workings are then given to conclude the introduction to algebraic links, like the relevance of terms in the curve and its coefficients.

In Chapter 2 the Alexander polynomial is introduced in two different ways. The first one is with the Alexander module, and the ideals of principal ideal domains constructed over the fundamental group of the complement of the knot. This gives more information and can sometimes be used when there are some unknowns about the knot. The second one is through the Burau representation. It is easier to understand, but requires a word whose closure is the knot. This is very similar to the fundamental group in the first one, the main difference being a more hands-off approach and the ability to normalise it, even though it is symmetric. The Alexander polynomial of the right-hand trefoil is calculated with this last method. The issues with the polynomial are stated as well, namely that it does not differentiate between noninvertible knots and that the Kinoshita-Terasaka knot has trivial Alexander polynomial. However, the important result, and the reason that it is being used to classify knots is presented. Indeed, the Alexander polynomial is a complete invariant for algebraic knots. After this introduction, the attention is focused on the Alexander polynomial of algebraic knots specifically. So the general form of an iterated torus link is given, and proved using its Burau representation. Then, an idea of what happens for torus links is presented and quickly abandoned, since that will not be relevant for the objective that has been set. Instead, splice diagrams are explained along with their connection with the Alexander polynomial of an iterated torus knot. The pseudocode for reverse-engineering the Alexander polynomial of an algebraic knot is given, along with restrictions to avoid ones that aren't from algebraic knots. The result is the splice diagram, from which the iterated torus knot can be found, and thus its Puiseux pairs. Actually, the result is a minimal splice diagram, where no redundant iterations are present. This can be used to give canonical Puiseux pairs. To conclude, there is a remainder for the case of algebraic links, yet unsolved. It is also stressed that some iterated torus links aren't algebraic.

In Chapter 3 the linking number is finally defined to complete the gap in the classification of algebraic links. It is shown that it is, in fact, an invariant and also how it can be used to distinguish non-algebraic links from others that are algebraic. Some interesting types of links in terms of the linking number are given. After this, there are several algorithms in pseudocode that complete the classification of algebraic links. The first finds the components, after which the pairwise linking numbers and respective Alexander polynomials can be given. This allows for the determination of the equivalence class of an algebraic link, given as a word. There is also an alternative way to compute the linking number from just the equation of the algebraic curves. Some examples follow. These examples aren't all algebraic links, but instead try to showcase the way that the tools developed can bring clarity when deciding whether a given link is algebraic or not.

The goal of the work is then achieved.

Theorem (Lê, Zariski-Lejeune)**.** An algebraic link is determined by the Alexander polynomials of the individual components and their pairwise linking numbers.

Under the guidance of this theorem, two main classification problems are solved in this work:

1. Starting from a braid whose closure is the disjoint finite union of algebraic links we obtain:
   a) The disjoint algebraic links and their components (See A.3)
   b) The Alexander polynomial of each component (See A.1 and A.2)
   c) The pairwise linking number of the components (See A.4)
   d) The iterated torus knot that each component is equivalent to (See A.5 and A.6)
2. Starting from the equation of an algebraic link we obtain:
   a) The iterated torus knot of each branch of the plane algebraic singularity (See (1.5) and (2.7))
   b) The pairwise linking number of the components (See 3.11)

So we have developed a very robust procedure to give necessary conditions for a given link to be algebraic. If it is algebraic indeed, the invariants that completely determine it are given. Finding whether the braid representatives are conjugate to the actual algebraic links with these invariants is not solved, but the way to do so is already prepared.

Resumen

El objetivo de este estudio es clasificar un tipo especial de nudos llamados *nudos algebraicos*. Tienen una relevancia especial debido a su conexión con variedades analíticas y algebraicas, como indica su nombre. Hay muchos resultados importantes para estos nudos, que permiten un estudio profundo. Para aumentar un poco el ámbito de estudio, se tendrán en cuenta distintas componentes en forma de enlaces, así como la unión finita de enlaces algebraicos disjuntos. También se aplicarán a éstos los resultados mencionados previamente.

La clasificación estará eminentemente enfocada a diferenciar enlaces, en lugar de crear una tabla de las uniones de enlaces algebraicos disjuntos. Para conseguir esto se introducen conceptos básicos de teoría de nudos, junto con explicaciones y notas. También se añaden los resultados importantes que afectan a la clasificación y se muestran brevemente los procedimientos para aplicar estos resultados. Los algoritmos están en el apéndice, implementados en `Sagemath` de manera completa. Hay algunos ejemplos resueltos en la conclusión para demostrar la aplicabilidad de la construcción matemática desarrollada. A continuación se da un resumen más detallado del trabajo.

En el Capítulo 1 se definen las trenzas y se da el grupo de trenzas B_n. Hace falta darlo porque más adelante los enlaces se construyen al identificar los extremos correspondientes de la trenza. Una vez hecho esto se introducen los nudos y se especifica que tienen que ser *tame*. Se estudia la orientabilidad, con el trébol como ejemplo. Por un lado se da el ejemplo para ilustrar cómo es la proyección regular de un nudo, lo cual se explica un poco después. Por otra parte, es porque los enlaces algebraicos, objeto de este estudio, son invertibles. También son quirales, así que se define la quiralidad. A continuación se explican los nudos tóricos, especificando cómo representarlos con tuplas y la elección de longitud que se hace. Se da el número de cruce y también lo que se mencionaba sobre que son reversibles. Hace falta generalizar el concepto de nudos tóricos, por lo que se introducen más componentes que permiten hablar de enlaces tóricos. Después de esto, se da la manera de iterar. Así se alcanzan, finalmente, los enlaces tóricos iterados. Se ven algunas técnicas que permiten apreciar si un enlace es tórico iterado. Por ejemplo, si es hiperbólico o si el número de cruce supera una cota dada son condiciones necesarias. También hay otra manera de iterar, dada por nudos satélites y cableados. Aquí se da una nota importante sobre enlaces tóricos iterados, donde se explica que está permitido que el alma de un toro se mantenga al hacer la iteración. Con los enlaces tóricos iterados definidos satisfactoriamente ya, es hora de sacar a los enlaces algebraicos a la luz. Primero se expresan en términos de una curva compleja y después como enlaces tóricos iterados con un cambio de variables correspondiente. La notación para representarlos también se explica, dando así los pares de Puiseux y Newton. Hay algunas restricciones sobre qué enlaces tóricos iterados pueden provenir de algebraicos, que se determinan en función de sus tuplas. También se muestran algunas propiedades poco intuitivas, entre ellas la relevancia de los términos de la ecuación de la curva y sus coeficientes.

En el Capítulo 2 se introduce el polinomio de Alexander de dos maneras distintas. La primera es con el módulo de Alexander y los ideales del dominio de ideales principales construido sobre el grupo fundamental del complemento de un nudo. Esta forma da más información y en ocasiones puede servir para reconstruir el polinomio a partir de datos incompletos. La segunda es a través de la representación de Burau, la cual es más fácil de entender. Sin embargo, hace falta una palabra que represente al nudo. Realmente, encontrar esta palabra es un problema esencialmente equivalente a hallar el grupo fundamental de la forma anterior. La única diferencia es que es más automática y permite normalizar el polinomio, que por cierto es simétrico. Con este último método se calcula el polinomio del trébol. Tam-

bién se mencionan los problemas que tiene el polinomio, como que no diferencia nudos no invertibles y que el del nudo de Kinoshita-Terasaka es trivial. Llegado este punto, se enuncia el resultado importante, la razón por la que se está utilizando este polinomio para clasificar nudos. Esto es, el polinomio de Alexander es un invariante completo para los nudos algebraicos. Una vez dicho esto, se centra la atención en el polinomio de nudos algebraicos específicamente. Se muestra la forma general del polinomio de enlaces tóricos, la cual se prueba con la ayuda de la representación de Burau. Luego se da una idea de lo que sucede para los enlaces tóricos y se abandona rápidamente, ya que no será relevante para el objetivo fijado. Seguidamente, los *diagramas splice* se explican para detallar la conexión de un nudo tórico iterado con su polinomio de Alexander. Se da el pseudocódigo para hallar el nudo algebraico del cual proviene un polinomio de Alexander, junto con restricciones para evitar los que no sean de nudos algebraicos. También se señala como encontrar el nudo tórico iterado y sus pares de Puiseux a partir de este diagrama, el cual es mínimo y no tiene elementos redundantes. Esto se puede utilizar para dar una forma canónica de los pares de Puiseux. Para concluir, hay un recordatorio de que el caso de los enlaces algebraicos aún sigue sin resolver. También se subraya que algunos enlaces tóricos iterados no son algebraicos.

En el capítulo 3 se define finalmente el número de enlace para completar la laguna en la clasificación de los enlaces algebraicos. Se muestra que efectivamente es invariante y también cómo se puede utilizar para distinguir enlaces no algebraicos de otros que lo son. Se dan algunos enlaces en términos del número de enlace. Después de esto, hay varios algoritmos en pseudocódigo que completan la clasificación de enlaces algebraicos. El primero busca las componentes, después de lo cual se pueden dar los números de enlace dos a dos y los polinomios de Alexander respectivos. Esto permite determinar la clase de equivalencia de un enlace algebraico, dado como palabra. También hay una manera alternativa de calcular el número de enlace únicamente a partir de la ecuación de las curvas algebraicas. A continuación se muestran ejemplos. Algunos de estos ejemplos no son realmente enlaces algebraicos, y tratan de mostrar la forma en que las herramientas desarrolladas pueden aclarar si un determinado enlace es algebraico o no.

En el Apéndice A se muestra la implementación de los algoritmos y procedimientos mencionados. Estos algoritmos reciben una trenza y devuelven la matriz de números de enlace dos a dos con los enlaces disjuntos separados por cajas diagonales. También devuelven los polinomios de Alexander de cada una de las componentes y la forma en nudo tórico iterado, si es posible. Esto nos permite dar condiciones necesarias para que un enlace venga de uno algebraico. Que sea efectivamente algebraico, es decir, que la trenza sea conjugada de éste, no está resuelto. Sin embargo, los pasos iniciales ya están señalados, pues solamente habría que comprobar el enlace algebraico con los invariantes dados.

Así se alcanza el objetivo del trabajo.

Teorema (Lê, Zariski-Lejeune)**.** Un enlace algebraico está determinado por los polinomios de Alexander de las componentes individuales y los números de enlace dos a dos.

Con el apoyo de este teorema, resolveremos dos problemas de clasificación:

1. Tomando una trenza que al cerrarse da la unión finita disjunta de enlaces algebraicos se obtiene:
    a) La unión disjunta de enlaces algebraicos y sus componentes (Ver A.3)
    b) El polinomio de Alexander de cada componente (Ver A.1 y A.2)
    c) Los números de enlace dos a dos de las componentes (Ver A.4)
    d) El nudo tórico iterado al cual es equivalente cada componente (Ver A.5 y A.6)
2. Tomando una ecuación de un enlace algebraico se obtiene:
    a) El nudo tórico iterado de cada rama de la singularidad (Ver (1.5) y (2.7))
    b) Los números de enlace dos a dos de las componentes (Ver 3.11)

De esta manera, hemos desarrollado un procedimiento muy robusto que da condiciones necesarias para que un enlace sea algebraico. Si lo es, se determina el enlace algebraico. Discernir si efectivamente las representantes en trenzas son conjugadas de las de los propios enlaces algebraicos no se resuelve.

Contents

Chapter 1

Introduction to algebraic links

1.1. Braids

Definition 1.1. The *configuration space* of n ordered distinct points in the complex plane $\mathbb{C}$ is

$$M_n = \{(z_1, \ldots, z_n) \in \mathbb{C}^n; z_i \neq z_j, \forall i \neq j\}.$$

Simply put, M_n is the set of tuples in $\mathbb{C}^n$ where no coordinates coincide. Note that this has real dimension $2n$. Since it is connected, when defining its fundamental group, any point can be used. A conjugation would be enough to change the chosen point.

Definition 1.2. The *pure braid group* on n strands is

$$PB_n = \pi_1(M_n).$$

So, a pure braid β in PB_n is a closed loop in M_n. Since the points of M_n are all distinct, we can assign each component of β to a separate strand. This assignation is well-defined, and returns a motion of n points where the strands are distinct at every step of the motion. Following this construction, a simpler way of representing β arises: the *geometric braid*. Start by naming the i-th projection of β β_i. These will be called the *strands* of the braid. We can now represent β in $\mathbb{C} \times [0,1]$ as the n strands $(\beta_1(t),t), \ldots, (\beta_n(t),t)$. These strands do not intersect because $\beta(t)$ is in M_n. Now, it is only necessary to fix the base point, which in $\mathbb{C} \times [0,1]$ translates to choosing n distinct points to be the beginning and end of the strands. For the sake of simplicity, the i-th strand will begin at $(i,0)$ and end at $(i,1)$. With this representation in mind it is easy to state when two motions are isotopic. Thus, the endpoints of the strands need to be fixed while being pairwise disjoint. Also, they must intersect $\mathbb{C} \times \{t\}$ exactly once, for all $t \in [0,1]$.

Remark 1.3. It has been previously stated that PB_n is, indeed, a group, and the justification is simple. The neutral element is the motion where all strands are vertical lines. The multiplication is given by joining two braids together and rescaling so as to get a path in $[0,1]$.

We will now expand this notion of *pure braid* to include the ones where the strands don't begin at the same point of $\mathbb{C}$. By introducing the symmetric group $\sum_n$ that permutes the coordinates of M_n, it is possible to give that more general notion.

Definition 1.4. The *configuration space of n unordered distinct points* in the complex plane $\mathbb{C}$ is

$$N_n = M_n \big/ \textstyle\sum_n$$

Now the coordinates are not ordered, so that the only restriction is that they are distinct.

Remark 1.5. Another way to understand N_n is as the complex polynomials of degree n and simple roots. That is, $V_n \setminus D_n$ where $V_n = \{p(t) \in \mathbb{C}[t] | p \text{ monic, } \deg(p) = n\}$ and $D_n = \{p(t) \in V_n | p \text{ has multiple roots}\}$. The map is $M_n \to V_n \setminus D_n$, with $(x_1, \ldots, x_n) \mapsto \Pi_{i=1}^n (t - x_i)$. Now $N_n \cong V_n \setminus D_n$.

Definition 1.6. The *braid group* on n strands is

$$B_n = \pi_1(N_n).$$

This definition is more general because the end of each strand doesn't have to be equal to its start. The geometric construction as above yields braids with i-th strand beginning at $(i,0)$ and end at $(\sigma(i),1)$, where $\sigma \in \sum_n$. This permutation in the components of the point in N_n is the way to understand the monodromy action of B_n. Braids will be considered equal if a homotopy relative to the endpoints exists. The following presentation of B_n was given by Artin in [4]:

$$B_n = \langle \sigma_1, \dots, \sigma_{n-1} | \sigma_i\sigma_j = \sigma_j\sigma_i, |i-j| > 1; \sigma_i\sigma_j\sigma_i = \sigma_j\sigma_i\sigma_j, |i-j| = 1 \rangle \tag{1.1}$$

where σ_i is the twisting of the i-th and $i+1$-th strands by a local negative half Dehn twist, that is, a counter-clockwise rotation of these strands as shown in Fig. 1.1.

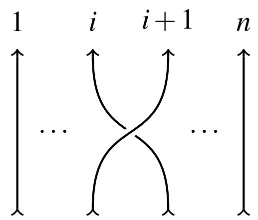


Figure 1.1: Artin generator σ_i

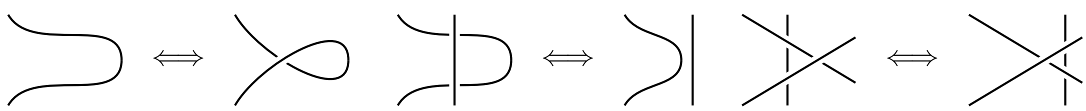
The problem of braid equivalence can be thusly translated to algebraic terms in what's known as the *word problem*. A *word* of a braid is simply an ordered product of σ_i or σ_i^{-1} as shown in (1.1). This problem has been solved [13] so that two words in B_n can be determined to be equal or not through the relations in B_n. Similarly, Garside's work has shown it is possible to solve an extension of the word problem in B_n. Specifically, in [12] he devised a procedure to find when two braids in B_n are conjugate. These results imply that there is no need to study braids through invariants, as we'll do for links.

1.2. Knots

Braids are really just another way of looking at knots and links. Indeed, by identifying the ends of a braid together, we create the object of our study: a link. If it has a single component, it is called a knot. Let us give a more detailed definition.

Definition 1.7. A *knot* is a smooth embedding of $\mathbb{S}^1$ in $\mathbb{S}^3$. Two knots will be considered *isotopic* if there is a diffeomorphism between their ambient spaces.

The notion of isotopy can be understood in a more geometric manner through equivalence and orientation-preserving homeomorphisms. If the focus is on combinatorics, the Reidemeister moves as in Fig. 1.2 are a set of three transformations of the knot which enclose all the possible elementary moves that don't change isotopy class. The Reidemeister moves are specially insightful because there are constructions that extract information from knots and are only invariant for some of the moves. All of this is explored further in [2, 8, 16, 19, 20, 24, 27, 31, 30].



(a) Type-I Move (b) Type-II Move (c) Type-III Move

Figure 1.2: Reidemeister moves

For instance, the inclusion of $\mathbb{S}^1 \subset \mathbb{R}^2$ in $\mathbb{R}^3$ is called the *trivial knot*. For most knots, this change from $\mathbb{S}^3$ to $\mathbb{R}^3$ won't be an issue since only the compactification point is missing. Certainly tame knots will be able to be understood properly here. When talking about this, by an abuse of language, we will refer as knots to both the isotopy class and the image of the embedding. Some knots, called *wild* are troublesome because they are not differentiable. Several examples of wild knots can be seen in [5].

Definition 1.8. A knot is called *tame* if it is equivalent to a piecewise-linear knot.

We will only study tame knots, to avoid the issues that wild ones present. Tame knots can be represented by a finite closed polygonal curve. Knots of this form will be called *polygonal*. Drawing them in a *smooth* manner doesn't mean they lose this property.

In this study, knots will be assumed to be *oriented*. For every knot, two orientations are possible. If both orientations are equivalent, it is said to be *invertible*. Many of the first knots are invertible. Here, first is referring to a classification of knots that will be presented later. For example, the simplest nontrivial knot, the *trefoil knot* is invertible. To see this, simply rotate the diagram of Fig. 1.3 in $\mathbb{R}^3$ along the axis of the affine projection. This idea of projection needs to be explained further.

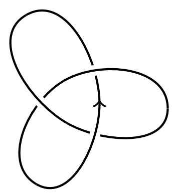


Figure 1.3: The trefoil knot 3_1

Any tame knot can be represented properly as a projection in the following manner.

Definition 1.9. A projection of a tame knot over an affine plane of $\mathbb{R}^3$ is *regular* if it satisfies the following:

a) There is a finite number of multiple points.

b) These multiple points are all double points, that is, the cardinal of their fibers is 2.

c) No vertex is projected over a double point.

Since the multiple points are double points, we can represent which strand goes over by drawing continuously. The minimum number of multiple points of the regular projections of a knot is its *crossing number*. Traditionally knots have been classified according to their crossing number, in the *Alexander-Briggs notation*. For instance, the trefoil is 3_1 in this notation because it has crossing number three. The subindex is simply a convention and doesn't have any inherent information other than to label the knot.

If we switch which strand goes over in all the double points, we get the mirror image of the knot. Knots equivalent to their mirror image are *amphichiral*. According to the previously presented classification of knots, 4_1 is the first amphichiral knot. Since 4_1 is also invertible, it is *fully amphichiral*. Knots that are invertible but not amphichiral are called *reversible*. Non amphichiral knots will be referred to as *chiral*.

Definition 1.10. A knot equivalent to one in the revolution torus $\partial\{(x,y) \in \mathbb{C}^2 | \ |x| = 1, |y| \leq 1\} \subseteq \mathbb{S}^3 \subseteq \mathbb{C}^2$ is called a *torus knot*.

Note that $\mathbb{S}^3$ is the union two solid torii glued along their common boundary. Calling the $\mathbb{S}^1$ from $\{(x,1)| \ |x| = 1\}$ the *meridian* curve and the other $\mathbb{S}^1$ from $\{(1,y)| \ |y| = 1\}$ the *longitude* curve, we can count the number of times a given torus knot intersects these curves. Note that the longitude is a meridian in another solid torus. Say the tuple is $\{p,q\}$. We will denote the torus knot with such tuple as $K\{p,q\}$. By imposing that the wrapping of the knot along the torus is counter-clockwise, this tuple defines a single isotopy class. Clockwise wrapping would be represented by negative integers. Changing the sign of only one of the numbers of a torus knot results in its mirror image, while changing both reverses the orientation.

All torus knots with the same tuple as above are equivalent. Moreover, we can consider the tuple as just a set of two numbers, because the torus knots $K\{p,q\}$ and $K\{q,p\}$ are equivalent. It is seen easily simply by deforming the meridian to be the longitude and vice versa. This explains the set notation, used over $K(p,q)$.

Remark 1.11. The choice of orientation is clear, and given by the structure of $\mathbb{C}^2$. The torus knot $K(p,q)$ can be given as $t \mapsto (re^{2i\pi tp}, re^{2i\pi tq})$ by identifying the ends together.

Torus knots can also be given in terms of the braid very easily. For instance, the closure of $(\sigma_1 \cdots \sigma_{p-1})^q \in B_p$ is the torus knot $K(p,q)$.

Theorem 1.12. *The crossing number of a torus knot* $K\{p,q\}$ *is* $\min\{|p|(|q|-1), |q|(|p|-1)\}$.

Note that the crossing number must be less or equal than $\min\{|p|(|q|-1), |q|(|p|-1)\}$ because there exist regular projections with this crossing number. The reason why this is indeed the crossing number is not so immediate, and was proved in [24, 7.5] and in [8, 3.29].

Theorem 1.13. *Nontrivial torus knots are reversible.*

Proof. We need to prove they are both invertible and chiral, that is, $K\{p,q\}$ isotopic to $K\{-p,-q\}$ but not isotopic to $K\{-p,q\}$ or $K\{p,-q\}$. Using the symmetry of the torus, take a reflection from the plane of the longitude followed by a half rotation. This works even for nontrivial torus knots. Analytically, by taking the parametrization in Remark 1.11 and the change of variable $\tilde{t} = -t$, the equivalence is shown. Proof of chirality can be seen in [24, 7.4.2]. □

When attempting to construct a torus knot from the aforementioned set, it is crucial to realise that the numbers must be coprime. Otherwise, the result would be a knot folded on itself.

1.3. Links

Definition 1.14. A *link* is a smooth embedding of the disjoint finite union of $\mathbb{S}^1$ in $\mathbb{R}^3$.

A link can be considered as the union of non-intersecting knots. These distinct knots will be called the *components*. Clearly, the number of components is an invariant of links. If a link is equivalent to having its components separated through planes, it is called *splittable*. When a link is equivalent to the disjoint union of several links, the components in those different links are splittable as well. However, if there is more than one component in any of those links, the original link is not splittable.

Remark 1.15. We can define a torus link of type (dp, dq), $\gcd(p,q) = 1$ as d parallel copies of torus knots with type (p,q) in the same torus. They will, however, not be splittable in general. Consider the case $K\{2,2\}$. This torus link is known as the *Hopf link*. It is not splittable because its linking number is nonzero, as we'll see later on. The converse isn't necessarily true. In particular, there exist some links with components where the linking number is null but which aren't splittable. This is because the other components are involved as well and prevent the splitting from happening. Fig. 3.3 is an example.

Torus knots can be iterated in the following manner from their braid presentations. First construct a torus knot and then consider a cylinder around the strands. Since the knot, by definition, is closed, the cylinder will close and form a second torus. This torus is not the standard torus, but we can work on it just as easily since it is homeomorphic to it. Note that the cylinder might have a very small radius, to avoid self-intersections. On this second torus construct a torus knot.

Definition 1.16. An *iterated torus knot* is a knot constructed over a torus found by iterating as mentioned previously.

Remark 1.17. Clearly, not all knots will be torus knots. By a simple argument with the crossing number of $K\{p,q\}$ we can see this. For instance, there are 7 nonequivalent knots with crossing number 7, but the only possibilities for torus knots are $K\{2,7\}$ and $K\{1,8\}$. The second one is clearly trivial, now we have 6 knots which aren't torus knots. In fact, for crossing number less than 11, only 3_1, 5_1, 7_1, 8_{19}, 9_1 and 10_{124} are torus knots. For comparison, there are 250 knots with crossing number less than 11, so torus knots aren't very common. Reversibility is too broad to characterise iterated torus knot. For instance, the *cinquefoil knot* is a torus knot $K\{5,2\}$. However, the knot 5_2 is reversible as well, but not an iterated torus knot. One way to see that it can't such a knot is because it is a *hyperbolic knot*. This means that a metric of curvature -1 exists for the complement of the knots. Iterated torus knots are non-hyperbolic, so this is actually a better test.

Iterated torus knots can be represented with tuples of integers generalising the torus knot notation, but in order to do that, the longitudes will have to be fixed. Note that the meridians are already canonical. This fixed longitude will be the naive longitude, taken as a parallel curve to the braid representative as in Fig. 1.4. Now, the crossing number is a more delicate question. We will only give a bound for it. First notice that the iterated torus knots $K((p_1,q_1),(p_2,q_2))$ and $K((p_2,q_2),(p_1,q_1))$ are not equivalent. Also not equivalent are $K((p_1,q_1),(p_2,q_2))$ and $K((q_1,p_1),(p_2,q_2))$. This justifies the tuple notation. With a similar argument as before, is is simple to find an upper bound.

Theorem 1.18. *The crossing number of iterated the torus knot* $K((p_1,q_1),\dots,(p_k,q_k))$ *is*

$$cr(K((p_1,q_1),\dots,(p_k,q_k))) \leq \sum_{i=1}^{k}(|q_i|(|p_i|-1)\prod_{j=i+1}^{k}p_j^2) \tag{1.2}$$

Proof. Simply take the braid whose closure is that torus knot and separate all the crossings in stages. This braid is obtained from recursion over the representative shown for one iteration. First consider the crossings from the initial iteration, followed by the second and so on. In the first stage, the strands $q_2 \dots q_k$ are involved in each of the $p_1(q_1-1)$ crossings. In the second one, $q_3 \cdots q_k$ are involved in each side of the $p_2(q_2-1)$ crossings. By induction the bound follows. □

The bound will be reached for the (p_i,q_i) such that the number of iterations is lowest but the knot doesn't change i.e. no iteration is of the type $(1,n)$. Clearly iterations with $(1,n)$ are isotopic to the identity, as they are isotopic to the core of the solid torus.

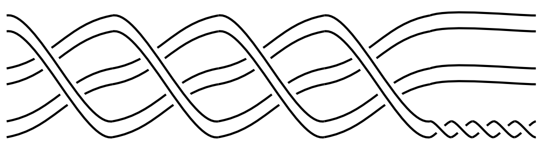


Figure 1.4: Braid with closure $K((3,4),(2,5))$

The notion of iterated torus knots is generalised in that of satellite knot.

Definition 1.19. Let K_1 be a knot inside a solid torus and K_2 a tame knot. Then, its *satellite knot* K_3 is the result of fixing the torus from K_1 inside a tubular neighbourhood of K_2 without self-intersections. In this case, K_2 is called the *companion knot*.

Many new knots can be constructed in this manner. One such example is the *Whitehead double* or *double knot*. This is simply the satellite of the knot whose projection is shown in Fig. 1.5.

The Whitehead double can be twisted in the middle of the torus, creating a whole family of knots, the *twist knots*. In the Alexander-Briggs notation explained previously, twist knots appear right after torus knots.

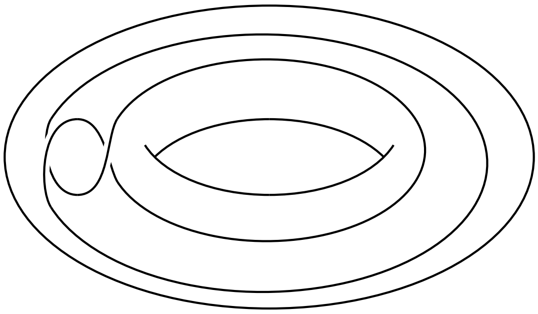


Figure 1.5: Initial knot for the double satellite

Remark 1.20. The knot K_1 doesn't have to be a torus knot, and neither does K_2. But if both are, the resulting knot is an iterated torus knot. This construction concept will be generalised in that of cabling, explained below.

Definition 1.21. If the starting braid is a torus knot $K(p,q)$ with the naive longitude, its satellite knot over the companion knot K is called the (p,q)-*cable* of K.

Remark 1.22. There is one delicate thing here that isn't being explained properly. That is the choice of the longitude. It will be implied that the naive choice of a longitude to the knot is taken. However, many different ones could be taken. One that stands out is that which has a special relation with the torus knot, called *null homologous longitude*. It is harder to express and find, but results in easier representations of invariants. This added difficulty is not necessary in our case, where a more intuitive longitude is both useful and simple to obtain. Null homologous longitudes might be used in other knots, which are constructed over more complex surfaces. Since there is no convention on the choice for longitudes, the reader is encouraged to try and clarify when just the tuples are given. Different choices of longitudes will most likely result in nonequivalent knots, even with equal tuples.

Here the order of p and q does matter, as it does in iterated torus knots with more than one iteration. With the notion of cabling, the result of iteratively taking the (p_i,q_i)-cables starting from the trivial is the iterated torus knot $K((p_1,q_1),\dots,(p_k,q_k))$. In general, cables without a clear choice of longitude will be selected with the null homologous one. However, in the case of torus knots and iterated torus knots, the naive one will be preferred. Whenever the null homologous longitude is chosen, the notation $K[(p_1,q_2),\dots,(p_k,q_k)]$ will be used.

Remark 1.23. Iterated torus link aren't well-defined with just the longitude and the tuples. When iterating, it is needed to stress over which component we are creating a tubular neighbourhood. This problem didn't appear in iterated torus knots simply because there is only one component every time, so the companion knot for the cables is well-defined. This is not so for the case with more components. We could either choose one of the components as the companion and construct the cable over it, but we could just as well construct cables over many components simultaneously. How to specify the construction then? One solution is with the splice diagram for links, which will be presented later. Another one will be to give separate iterated torus knots which are entangled. Either one is a complicated construction and requires a detailed study.

Definition 1.24. An *iterated torus link* is the resulting link from iterative cabling, that allows for the cabling of just one component to be chosen. In these cases, that torus core may not vanish after taking its satellite.

Remark 1.25. The class of iterated torus links is much more general than just allowing non-coprime iteration tuples, precisely for the choice to leave the core in some cases.

1.4. Algebraic links

Now we will change the focus to a subset of torus links where all the wrapping is in the positive direction. The starting premise is to take a complex curve $\{(x,y) \in \mathbb{C}^2; f(x,y) = 0\} = C_f$ and intersect it with a sphere $\{(x,y) \in \mathbb{C}^2; |(x,y)| = \varepsilon\} = \mathbb{S}^3_\varepsilon$ of radius sufficiently small. By a theorem from Milnor [22], this intersection is transversal, so $C_f \cap \mathbb{S}^3_\varepsilon = k_f$ is a well-defined link.

$$f(x,y) \in \mathbb{C}[x,y] \ni f(0,0) = 0, \gcd(f, f') = 1, x \nmid f \tag{1.3}$$
$$\exists \varepsilon, \delta \ni 0 < \delta \ll \varepsilon, f(x,y) \neq 0 \text{ if } |y| = \varepsilon, |x| \leq \delta \tag{1.4}$$

We will now introduce additional, nonintrusive restrictions that will allow us to operate over cartesian coordinates, instead of the stereographic projection. By taking our polynomial as in (1.3), we can find a rectangular sphere $B(0,\varepsilon) \times B(0,\delta)$ from (1.4). This simplifies the working coordinates to visualize the algebraic link, seen in a standard torus in $\mathbb{R}^3$.

Definition 1.26. The resulting curve K_f from the above procedure is an *algebraic link*.

Algebraic links constructed in this manner are well-defined. The choice of excluding multiple roots is to ensure transversality, which will be explored further. Moreover, the zero locus of f doesn't change with this exclusion. The reason for x to not be a factor is a matter of allowing $\partial(B(0,\varepsilon) \times B(0,\delta))$ to intersect C_f, instead of having to rely on $\mathbb{S}^3$.

By construction, all algebraic links are iterated torus links. In fact, we can exchange equations for iterated torus knots with the null homologous longitude as indicated in Remark 1.22. The reverse is also presented for simplicity purposes.

$$y = x^{\frac{m_1}{n_1}} + x^{\frac{m_2}{n_1 n_2}} + \cdots + x^{\frac{m_k}{n_1 \cdots n_k}} \Leftrightarrow$$
$$K\left[(n_1,m_1),(n_2,m_2+m_1n_2(n_1-1)),\ldots,\left(n_k, m_k + m_{k-1}n_k(n_k-1) + \sum_{i=1}^{k-2}[m_i(n_i-1)n_k \prod_{j=i+1}^{k-1} n_j^2]\right)\right] \tag{1.5}$$

$$K[(p_1,q_1),\ldots,(p_k,q_k)] \Leftrightarrow y = x^{\frac{q_1}{p_1}} + x^{\frac{q_2+q_1p_2(1-p_1)}{p_1p_2}} + \cdots + x^{\frac{q_k + \sum_{i=1}^{k-1}[q_i(1-p_i)\prod_{j=i+1}^{k} p_j]}{p_1 \cdots p_k}} \tag{1.6}$$

Remark 1.27. This clearly justifies the choice of naive longitude, since the conversion is much more simple, $y = x^{\frac{m_1}{n_1}} + x^{\frac{m_2}{n_1n_2}} + \cdots + x^{\frac{m_k}{n_1\cdots n_k}} \Leftrightarrow K((n_1,m_1),\ldots,(n_k,m_k))$. The issue with this longitude is that it requires a braid representative and is not canonical. On the other hand, the null homologous longitude is canonical.

The plane curve singularity from the equation $f(x,y) = 0$ may have several branches. Each of them can be expressed after a change of coordinates as $x = t^n$, $y = h(t)$ and so $y = h(x^{\frac{1}{n}})$. Then, every equation represents a component of the algebraic link. The restrictions for the equation form are $\frac{m_1}{n_1} < \frac{m_2}{n_1n_2} < \ldots < \frac{m_k}{n_1\cdots n_k}$ where $m_i, n_i \in \mathbb{N}$ such that $\gcd(m_i, n_i) = 1$. In general, there would be finitely many equations of this type. Notice that some manipulation might be needed to make the n_i appear in the appropriate denominators.

The pairs $(n_1,m_1),\ldots,(n_k,m_k)$ are called *Puiseux pairs*. Some conditions apply to the coefficients in (1.5) and (1.6) because of the following decomposition $y = x^{\frac{s_1}{r_1}}(1 + x^{\frac{s_2}{r_1r_2}}(1+\ldots))$. We can express s_i in terms of Puiseux pairs as $s_i = m_i - n_i m_{i-1}$. This decomposition is called the *Newton decomposition* and its respective pairs (r_i, s_i), the *Newton pairs*. The Newton pairs are important because they give the successive cabling that results in the appropriate iterated torus knot. Since the iterations have to be positive integers, some restrictions for (1.6) can be obtained. Indeed, the positivity of the Newton pairs implies that $q_n > p_n p_{n-1} q_{n-1}$. Iterated torus knots that don't satisfy this aren't algebraic. For instance, the closure of the braid in Fig. 1.4 is $5 \not> 2 \cdot 3 \cdot 4$, so it's not an algebraic knot. Also notice that if the Puiseux pairs are coprime, the Newton pairs will be as well. With the simple restriction that $p_1 < q_1$, there is a canonical representation with iterated torus knots.

Remark 1.28. The algebraic knot from f might be equivalent to that from g, even if f and g aren't equal. For example, the equations $y = x^{\frac{1}{2}} + x^{\frac{3}{4}}$ and $y = x^{\frac{5}{2}}$ yield equivalent algebraic knots. To see this, obtain the respective iterated torus knots with the procedure in (1.5). The knots are $K((2,1),(2,5))$ and $K\{2,5\}$. These are seen to be equivalent by eliminating the redundant iteration $(2,1)$. In general not all terms are relevant. The algebraic knot from $y = (x + \cdots + x^{g_0}) + a_1 x^{\frac{q_1}{p_1}}(1 + \cdots + x^{g_1}) + a_2 x^{\frac{q_2}{p_1 p_2}}(1 + \cdots + x^{g_2}) + \cdots + a_k x^{\frac{q_k}{p_1 \cdots p_k}}(1 + \dots)$ with polynomials in the parenthesis is equivalent to the one from just $y = x^{\frac{q_1}{p_1}} + x^{\frac{q_2}{p_1 p_2}} + \cdots + x^{\frac{q_k}{p_1 \cdots p_k}}$. Moreover, the last parenthesis can even be a convergent series. This can be checked by realising that in each deleted step, no new strands are being added, so the isotopy to the original strand can be taken. This is similar to what happened to the iterations of the form $(r,1)$. Analytically, these terms will only affect to some local deformations that for ε sufficiently small don't alter the equivalence class.

Remark 1.29. When multiple components are involved, the coefficients of the equations are no longer irrelevant for the isotopy type. They will determine the linking number, as will be explored further in Chapter 3. Note that in general, $y(x^{\frac{1}{n}})$ will be equal to $y(\xi x^{\frac{1}{n}})$, with $n = p_1 \dots p_k$ and ξ a root of the unit.

Algebraic links can then be studied as iterated torus links of through their invariants. We have already seen link invariants, like invertibility, chirality, number of components, crossing number or splittability. The focus will be on on splittability, expanded in the notion of linking number and used in conjunction with the Alexander polynomial. Iterated torus links will be used to classify them, once determined.

Chapter 2

Alexander polynomial

With the fundamental group of the complement of a given knot K, a canonical $\mathbb{Q}[t,t^{-1}]$-module can be found. Calling the submodule M_K and the ring R, since R is a principal ideal domain, M_K can be decomposed as a direct sum over some ideals. The decomposition will be of the form $M_K = \bigoplus_{i=1}^{r} \frac{R}{(p_i)} \oplus R^n$. The ideals of the decomposition are then used to define a polynomial. An equivalent definition will be given later.

Definition 2.1. The *Alexander polynomial* of a knot K is

$$\Delta_K(t) \doteq \begin{cases} 0 & n > 0 \\ \Pi_{j=1}^{r} p_j & n = 0 \end{cases} \tag{2.1}$$

with p_j the ideals as above and n the multiplicity of the non-cyclic element in the decomposition.

So the Alexander polynomial being defined through a $\mathbb{Q}[t,t^{-1}]$-module means that this is only defined up to units of $\mathbb{Q}[t,t^{-1}]$. In this ring, the units are of the form $\pm t^n$, $n \in \mathbb{Z}$. This ambiguity is contained in the notation $\doteq$, which represents $p \doteq q \iff \frac{p}{q} = \pm t^n$, $n \in \mathbb{Z}$.

Remark 2.2. Notice that, since the polynomial comes from the complement of the knot, there is no orientation taken into account and thus noninvertible knots are assigned the same invariant. This is the first and more obvious issue that appears, but as will be seen later, there are still some more, like the one with the Kinoshita-Terasaka knot.

The algebraic approach to the Alexander polynomial is general, and can be obtained in some cases where the fundamental group is not known. When it is known, however, a more comprehensive way of finding the Alexander polynomial can be given using the Burau representation and a braid whose closure is equivalent to the knot. This representation is not faithful for B_n, $n \geq 5$ but it won't be an issue for our purposes.

Definition 2.3. The (non-reduced) *Burau representation* of B_n is given by the following map

$$\begin{aligned} \tau : \quad \mathbb{B}_n \quad &\to \quad M(n, \mathbb{Z}[t,t^{-1}]) \\ \sigma_1 \quad &\mapsto \left(\begin{array}{cc|ccc} 1-t & t & & 0 & \\ 1 & 0 & & & \\ \hline & & 1 & & 0 \\ & 0 & & \ddots & \\ & & 0 & & 1 \end{array}\right) \end{aligned}$$

Remark 2.4. The matrices in the non-reduced Burau representation have $(1,t,\dots,t^{n-1})$ as eigenvector of eigenvalue 1. This eigenspace has dimension 1 and will be needed to be taken out in order to properly operate with this representation. The result will be the reduced Burau representation.

More information about the Burau representation is available in [6, 3.2, Example 3] and in [20, 8.4].

Definition 2.5. The *reduced Burau representation* is the quotient of the non-reduced space by the eigenspace of 1. As seen in [15, §2], it is given by $\tau : \mathbb{B}_n \to M(n-1, \mathbb{Z}[t,t^{-1}]$ with the following images:

$$\tau(\sigma_1) = \begin{pmatrix} -t & 0 & \\ -1 & 1 & \\ & & I_{n-3} \end{pmatrix}, \tau(\sigma_i) = \begin{pmatrix} I_{i-2} & & & & \\ & 1 & -t & 0 & \\ & 0 & -t & 0 & \\ & 0 & -1 & 1 & \\ & & & & I_{n-i-2} \end{pmatrix}, \tau(\sigma n-1) = \begin{pmatrix} I_{n-3} & & \\ & 1 & -t \\ & 0 & -t \end{pmatrix}$$

There is a relation between the reduced Burau representation and the Alexander polynomial.

$$\Delta_{\tilde{\beta}}(t) = \frac{\det(I_{n-1} - \tilde{\tau}(\beta))(t-1)}{t^n - 1} \tag{2.2}$$

Remark 2.6. Note that (2.1) and (2.2) define the same polynomial, considering the constraints in representation of (2.1). We shall normalise the polynomial so that no factors of the form $\pm t^n, n \in \mathbb{Z}$ exist. This way, it is a polynomial, not a Laurent polynomial and the highest exponent is as low as possible, while maintaining positive exponents. There are other ways of calculating the polynomial, taking a more combinatorial approach.

Example 2.7. The Alexander polynomial of the right-hand trefoil knot is $\Delta_{3_1}(t) \doteq t^2 - t + 1$. A braid with closure 3_1 is $\beta = (\sigma_1\sigma_2)^2$. First, we find with Definition 2.5 the reduced Burau representative of β, $\tilde{\tau}(\beta) = \begin{pmatrix} 0 & -t^3 \\ t & -t^2 \end{pmatrix}$. So $\det(I_{n-1} - \tilde{\tau}(\beta)) = 1 + t^2 + t^4$ and by (2.2), $\Delta_\beta(t) = \frac{(1+t^2+t^4)(t-1)}{(t^3-1)} = t^2 - t + 1$. Since the closure of $\tilde{\beta} = \sigma_1^3$ is the trefoil as well, it can be checked that $\Delta_\beta(t) = \Delta_{\tilde{\beta}}(t)$ is consistent with what is known. Proceeding similarly, $\tilde{\tau}(\tilde{\beta}) = (-t^3)$ and $\Delta_{\tilde{\beta}}(t) = \frac{(t^3-1)(t-1)}{(t^2-1)} = t^2 - t + 1$. In particular, $K(2,3) = K(3,2)$.

2.1. Properties

Theorem 2.8. *The Alexander polynomial is symmetric. In other words,*

$$\Delta_K(t) \doteq \Delta_K(t^{-1}).$$

Proof. This is done with the introduction of *Fox derivatives* and dual presentations of the fundamental group of the complement of the knot. The knot is then partitioned into segments with only over-crossings or under-crossings. With this partitions, the presentation of the group is changed to account for the words. One presentation will have the additional relations from the over-crossings and the other, the ones from the under-crossings. For the full proof see [11]. □

In this form, $\Delta_{3_1}(t) = t - 1 + t^{-1}$. However, the Alexander polynomial is not a complete invariant. There exist some nontrivial knots whose polynomial equals 1.

If we compute the Alexander polynomial of the knot in Fig. 2.1, the resulting polynomial is 1, the same as $\mathbb{S}^1$ in $\mathbb{R}^3$. Nonetheless, this polynomial will be useful for us because of the following result.

Theorem 2.9. *The Alexander polynomial is a complete invariant for algebraic knots.*

Proof. This is proved checking the roots, which are all of the unit, and seeing the relation they have with the Puiseux pairs. For the complete proof, see [17, Lemma 2.5.1]. Alternatively, Algorithm 2.10 proves this too. □

Thus, we conclude that the Kinoshita-Terasaka knot is not algebraic.

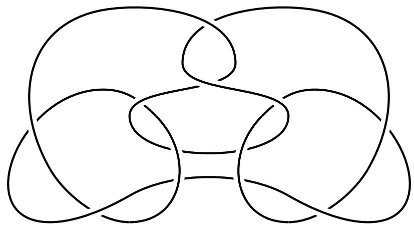


Figure 2.1: Kinoshita-Terasaka knot

2.2. Iterated torus knots

To simplify the calculations, we will now try to generalise the form of the Alexander polynomial of iterated torus knots. First start with torus knots. They will be of the form $K\{p,q\}$ with $\gcd(p,q)=1$. Their Alexander polynomial will be

$$\Delta_K(t) = \frac{(t^{pq})(t-1)}{(t^p-1)(t^q-1)} \tag{2.3}$$

This can be proved from the braid representative $(\sigma_1 \dots \sigma_{p-1})^q$. The reduced Burau representative of $\sigma_1 \dots \sigma_{p-1}$ is the matrix with -1 in the lower sub-diagonal and $(-t)^n$ in the $n-$th entry of the first row. Any power of this matrix will have some diagonal crossing that makes the computation of the determinant with principal minors immediate.

If $\gcd(p,q) \neq 1$, that is, if it were a torus link, the formula doesn't hold. Let $\gcd(p,q) = d$, then

$$\Delta_K(t) = \frac{(t^{\frac{pq}{d}}-1)^d(t-1)}{(t^p-1)(t^q-1)} \tag{2.4}$$

There will be really no use for this last formula, since the focus will be on the Alexander polynomial of each of the components of the torus link. In this way, if $K\{dp,dq\}$ with $\gcd(p,q)=1$, each of the components has the same roots in its Alexander polynomial than $K\{p,q\}$. This applies to iterated torus knots too.

It is still left to find the general formula for an iterated torus knot. In order to do that, we shall give the following graph representation of a given iterated knot. It is only for knots, links with more components would have more complicated constructions, with more arrows in the end, one for each component. This representation of a link is called a *splice diagram* and was introduced in [1, 10, 1.2]. Actually this is a more compact version than the splice diagrams from [10], with just the information needed for the Alexander polynomial.

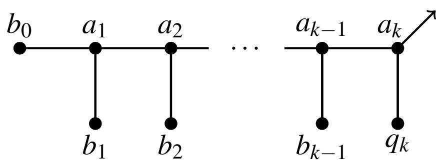


Figure 2.2: Graph representation of an iterated torus knot

If the graph in Fig. 2.2 represents the iterated torus knot $K[(p_1,q_1),\dots,(p_k,q_k)]$, then $a_i = p_i b_i$. So we only need to specify the b_i in terms of p and q with the naive longitude. The first elements are:

$$b_0 = \prod_{i=1}^{k} p_i, \quad b_1 = q_1 \prod_{i=2}^{k} p_i, \quad b_2 = q_2 \prod_{i=3}^{k} p_i \tag{2.5}$$

$$b_n = q_n \prod_{i=n+1}^{k} p_i \tag{2.6}$$

From this representation it is very simple to express the Alexander polynomial because it coincides with the characteristic polynomial of the monodromy action.

$$\Delta_{K[(p_1,q_1),\ldots,(p_k,q_k)]}(t) = \frac{t-1}{t^{b_0}-1}\prod_{i=1}^{k}\frac{t^{p_i b_i}-1}{t^{b_i}-1} \tag{2.7}$$

Since we are only working with the simple to obtain, naive longitude, a change of variable will be needed after the previous formulae. With (2.7) and the fact that $b_n > b_{n-1}$ there is a constructive way of finding some p_i, q_i that represent an algebraic knot in the form of iterated torus knot from its given Alexander polynomial. The procedure is simple:

Algorithm 2.10. (Iterated torus knot from Alexander polynomial)

1. Check that the polynomial is product of cyclotomics. Otherwise the polynomial isn't from an algebraic knot.
2. Take the root of $\Delta_K(t)$ with the highest order and least argument, ξ_{a_k}.
3. From the a_k-th roots of the units, find the highest order q_k that is missing.
4. Reset the polynomial to $\tilde{\Delta}_{k-1}(t^{p_k}) = \frac{\Delta_K(t)(t^{q_k}-1)(t^{p_k}-1)}{(t-1)(t^{a_k}-1)}$, saving p_k as $\frac{a_k}{q_k}$.
5. Proceed recursively until all p_i and q_i are found. Stop when $\tilde{\Delta}(t) = 1$.

Remark 2.11. The recursion ensures that the q_i found in every iterative step is indeed q_i, instead of b_i. Notice how this is true for the last b, b_k. Since we are reducing the polynomial each time, it is then clear that q_i is obtained.

The original knot K is equivalent to $K[(p_1,q_1),\ldots,(p_k,q_k)]$. This procedure will yield no pairs of the form $(1,r)$ which are redundant, as stated previously. Moreover, it will give the minimum number of iterations of torus knots that result in an equivalent knot. The minimality comes from the minimality of this splice diagram. This minimal splice diagram can always be found, and the one presented for iterated torus knots is minimal. For a justification of the general case, see [10, 8.2]. In fact, Algorithm 2.10 will give a canonical splice diagram for iterated torus knots.

Remark 2.12. This takes care of the problem of iterated torus knots, but what happens for iterated torus links is still not solved. For that, we need further information. The Alexander polynomial could be generalised, so that it had a variable for each component of the torus link. For this approach see [33, theorem A].

It is necessary to stress that we are only focusing on algebraic links. Not all iterated torus knots are algebraic links. For instance, the torus knot $K\{3,2\}$ is algebraic but its mirror image, the torus knot $K\{3,-2\}$ isn't. From Theorem 2.9, if two algebraic knots are equivalent, their Alexander polynomials must coincide. Conversely, if two nonequivalent knots have the same Alexander polynomial, both cannot be algebraic. Since the Alexander polynomial of $K\{3,-2\}$ is $\Delta(t) = t^2 - t + 1$, the same as $K\{3,2\}$, and $K\{3,2\}$ is algebraic, its mirror image cannot be algebraic. So in particular, $K\{3,-2\}$ isn't an iterated torus knot. For more than one iteration, neither $K((3,2),(7,-2))$ nor its mirror image are algebraic. This is the reason why the restriction to clockwise wrapping, or positive integers in the iteration tuples, is given.

Chapter 3

Linking number

It has been seen that the Alexander polynomial alone isn't enough to completely characterise algebraic links. The problem comes from the multiple components and the ways that they can interact. We have already given an approximation to this issue in the notion of splittability. However, this was a concept that wasn't specific enough, as more than just the interaction between two components is involved. There is the need then, for an invariant which inputs two components and gives enough information so that, together with the respective Alexander polynomials, it gives a complete characterisation of algebraic links.

Definition 3.1. Let J and K be two oriented knots in $\mathbb{R}^3$ and consider the regular projection of $J \cup K$. The integer $lk(J,K)$ constructed by adding or subtracting in the intersections where J crosses over K as shown in Fig. 3.1 will be called the *linking number* of J and K, $lk(J,K)$.

This definition, at first glance, depends on the projection chosen. However, the linking number is much stronger than a simple projection invariant. It will be proved in Theorem 3.3 that indeed it is an invariant of links and thus doesn't depend on the projection.

Remark 3.2. There are many other ways to define it, but this is the simplest and most intuitive. This definition can be expanded to include oriented links with two components. If the two components of L are J and K, then $lk(L) = lk(J,K)$. This is well-defined because $lk(J,K) = lk(K,J)$. The equality follows from the close relation between the multiplicities in the multi variant Alexander polynomial and the linking number, at least for algebraic knots. In general the argument is more complicated but still true. So the commutativity is inherited.

$+1$ -1

Figure 3.1: Crossing sign convention

3.1. Properties

The linking number is useful when $lk(J,K) \neq 0$. In these cases, J and K are really linked, and cannot be separated by an isotopy. This is because the linking number is preserved by isotopy and if the components were equivalent to being split, the null linking number would be preserved, reaching a contradiction.

Theorem 3.3. *The linking number is invariant by isotopy. That is, if L and L′ are oriented links with two components, then*

$$L \sim L' \Rightarrow lk(L) = lk(L')$$

Proof. This is a matter of checking that it is invariant under Reidemeister moves as shown in Fig. 1.2. Let's check the first two moves. Type-I is clearly invariant because the crossing affects the same component. For Type-II there are two possibilities. Either the strands belong to the same component, in which case the linking number is locally null or they are different. In this second case, the linking number won't change either because it goes up on one crossing and down on the other, no matter the orientation chosen. □

Note that J, K can be linked even when $lk(J,K) = 0$. Fig. 3.2 is an example of this.

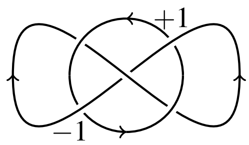


Figure 3.2: Whitehead's Link

In particular, Whitehead's link is not an iterated torus link. The fixed orientation in the wrapping of the strands around the torus in each of the iterations ensures that the linking number between the different components will be a positive integer. This orientation is clearly shown when the braid representative of the iterated torus link is of the type of Fig. 1.4.

Remark 3.4. In fact this allows us to ascertain when a link cannot be an iterated torus knot. For instance, Fig. 3.3 shows a three-component link. The pairwise linking numbers are $lk(K_0,K_1) = 1$, $lk(K_0,K_2) = 1$ and $lk(K_1,K_2) = 0$. Since some unsplittable components have null linking number, it is not an iterated torus link nor the disjoint union of iterated torus links. Indeed, the components in nontrivial iterated torus links have strictly positive linking number. This is just a necessary condition, though. Sufficiency is a harder matter that will only be checked by conjugating the iterated torus braid.

This argument applies to *Borromean rings* not being iterated torus links. This is generalised in *Brunnian links*. Brunnian links have components such that, if one is removed, they are all splittable. Borromean rings are a particular case of these, where the components are trivial knots.

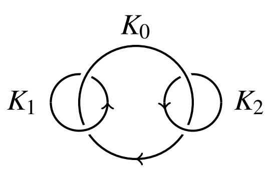


Figure 3.3: Not an iterated torus link

3.2. Algorithms

We will now give an algorithm to obtain the linking number between two given components of an iterated torus link. The starting point will be the iterated torus link. Specifically, the braid whose closure is that link. This way, we essentially have a word that represents the link. First, we'll find the components with this procedure.

Remark 3.5. This algorithm is only a way to give the permutation of the epimorphism $B_n \to \Sigma_n$.

Algorithm 3.6. (Components from word)

1) Substitute σ_i and σ_i^{-1} by the transposition $(i, i+1)$. Note that this doesn't specify which strand goes over.

2) Multiply all the transpositions, making sure to include single strands.

3) Count the number of cycles, including those with length one. That is the number of components of the link.

Remark 3.7. The indices of each cycle are the strands in that component.

Now to find the linking number between two given components first we need to isolate the crossings between only the ones of interest. Since the focus of these algorithms is on obtaining all the pairwise linking numbers of the components in a given link, we will obtain all at the same time. In fact, we will expand the area of study to include the disjoint finite union of algebraic links, or plane algebraic singularities that lie on different points in a straight line. This amounts to adding a constant in the Puiseux expansion. When we find the linking matrix of these links, there is a permutation that takes the matrix to diagonal box form. This permutation will also give the disjoint algebraic links. Two components will then be in the same algebraic link if their pairwise linking number is strictly positive.

Algorithm 3.8. (Pairwise linking numbers from components and word)

1. Enumerate the r components $K_1, \dots, K_r$ and change the indices so that i is assigned to j if $i \in K_j$.

2. Create the symmetric matrix $L = M(r, \mathbb{N}_0)$ that contains the linking numbers.

3. If σ_i changes the i-th index named j to s and $j \neq s$, add one to $L_{j,s}$.

4. If σ_i^{-1} changes the $i+1$-th index named j to s and $j \neq s$, substract one from $L_{s,j}$.

5. Proceed until the end of the word, reorder L with the permutation $\rho \in \sum_r$ so that $\tilde{L}$ is diagonal by boxes.

6. If the boxes of $\tilde{L}$ aren't of the type $\begin{pmatrix} 0 & * & \cdots & * \\ * & 0 & \cdots & * \\ \vdots & \vdots & \ddots & \\ * & * & \cdots & 0 \end{pmatrix}$ with $* \neq 0$, the word doesn't come from a disjoint union of algebraic links.

7. The pairwise linking numbers are $lk(K_j, K_s) = \tilde{L}_{\sigma(j),\sigma(s)}$.

The permutation ρ should be saved, so that the Alexander polynomials can be grouped in their respective links later on. In fact this is exactly what is done in the coded implementation. A similar process to Algorithm 3.8 could be followed to obtain the reduced words of each component, although the condition in step 3. would need to be changed. With these reduced words, the Alexander polynomial can be obtained as in (2.2).

Algorithm 3.9. (Alexander polynomials from components and word)

1. Change the indices as in step one of Algorithm 3.8.

2. Create an empty array of r words.

3. If σ_i doesn't change the i-th index and it is the s-th index called j, multiply the j-th word by σ_s.

4. Proceed until the end of the word.

5. Find the reduced presentation in Definition 2.5 and apply equation (2.2) to each of the r reduced words to obtain an array of r polynomials.

6. The Alexander polynomial of the i-th component is the i-th polynomial of this array.

Remark 3.10. Algorithms 3.8 and 3.9 can be combined to run simultaneously, being careful with ρ.

There is yet another way to obtain the linking number, this time from the different developments of an algebraic link. If we have an algebraic link given by the polynomials $y_1 = a_1x^{\frac{q_1}{p_1}} + a_2x^{\frac{q_2}{p_1p_2}} + \cdots + a_rx^{\frac{q_r}{p_1\cdots p_r}}$ and $y_2 = b_1x^{\frac{\tilde{q}_1}{\tilde{p}_1}} + a_2x^{\frac{\tilde{q}_2}{\tilde{p}_1\tilde{p}_2}} + \cdots + a_rx^{\frac{\tilde{q}_r}{\tilde{p}_1\cdots\tilde{p}_s}}$, the linking number between these two components can be found.

Theorem 3.11. *If K_1 and K_2 are two components of an algebraic knot given as above, then*

$$lk(K_1,K_2) = \sum_{\xi_1^{n_1}=1,\xi_2^{n_2}=1} ord(y_1(\xi_1 x) - y_2(\xi_2 x)) = n_1 \sum_{\xi^{n_2}=1} ord(y_1(x) - y_2(\xi x)) \tag{3.1}$$

where $ord(p(x))$ is the lowest exponent of $p(x)$, $n_1 = p_1\cdots p_r$ and $n_2 = \tilde{p}_1\cdots\tilde{p}_s$.

Remark 3.12. This is constructive, so Theorem 3.11 gives an algorithm for obtaining the linking number when the equations are given.

3.3. Classification theorem

Theorem 3.13 (Lê, Zariski-Lejeune)**.** *An algebraic link is determined by the Alexander polynomials of the individual components and their pairwise linking numbers.*

Proof. We'll give an idea of the proof of the case with two components, the rest follows inductively. Let $((p_1,q_1),\ldots,(p_r,q_r))$ be the Puiseux pair of the two components, obtained from their Alexander polynomials. Denote $p_1\cdots p_k = n$. Let $lk \in \mathbb{N}$ be their linking number. Then, if y_1 and y_2 give the equations of the respective algebraic curves, the linking number is $\sum_{\xi_1^n=1,\xi_2^n=1} \mathrm{ord}(y_1(\xi_1 x) - y_2(\xi_2 x)) = lk$. There will be some exponent α in y_1 and y_2 uniquely determined by the Puiseux pairs and lk. The uniqueness comes from the fact that with the Puiseux pairs fixed, the map $\alpha \mapsto lk$ is strictly increasing and thus injective. This is taken to be the first distinct term i.e. the coefficients are all equal up to x^α, up to multiplication by ξ with $\xi^n = 1$. There are several cases for the position of this factor in the polynomials, $\alpha \in (0, \frac{q_1}{p_1}) \Rightarrow \alpha \in \mathbb{N}$, $\alpha \in [\frac{q_i}{p_1\cdots p_i}, \frac{q_{i+1}}{p_1\cdots p_{i+1}}) \Rightarrow p_1\cdots p_i\alpha \in \mathbb{N}$ and $\alpha \in [\frac{q_r}{p_1\cdots p_r}, \infty) \Rightarrow p_1\cdots p_r\alpha \in \mathbb{N}$. So the links with these fixed pairs and linking number are equivalent. This proof follows the steps of [18]. □

3.4. Exercises

Example 3.14. We will now apply our algorithms to classifying the link given by the word

$$\begin{aligned}\beta = {}& \sigma_4\sigma_6\sigma_5\sigma_4\sigma_6\sigma_8\sigma_2\sigma_7\sigma_3\sigma_4\sigma_5\sigma_2^{-1}\sigma_5\sigma_6\sigma_5\sigma_4\sigma_5\sigma_7\sigma_4\sigma_7^{-1}\sigma_6^{-1}\sigma_7\sigma_6\\ & \sigma_2^{-1}\sigma_6^{-1}\sigma_4^{-1}\sigma_6\sigma_4\sigma_6^{-1}\sigma_4^{-1}\sigma_2\sigma_6\sigma_2^{-1}\sigma_4\sigma_2\sigma_6^{-1}\sigma_2^{-1}\sigma_6\end{aligned}$$

After Algorithm 3.6, we get $(1)(2,4,6,7,3)(5,8,9)$. There are three components, $K_1 = (1)$, $K_2 = (2,4,6,7,3)$ and $K_3 = (5,8,9)$. So it is already known that K_1 is trivial because it only has one strand. After Algorithms 3.8 and 3.9 we obtain the matrix with the pairwise linking numbers and the Alexander polynomials of the components.

$$L = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 4 \\ 0 & 4 & 0 \end{pmatrix},\ \Delta_{K_1}(t) = 1,\ \Delta_{K_2}(t) = t^4 - t^3 + t^2 - t + 1,\ \Delta_{K_3}(t) = 1$$

Indeed, $\Delta_{K_1} = 1$ as seen previously. The third component is trivial as well. The only non-trivial component is the second one. Since we know for certain that it comes from some union of disjoint algebraic links, we can now ensure that there are two algebraic links. The first is a trivial knot, K_1. The second is an algebraic link with a trivial component K_3 and an iteration $(2,5)$ on the other, with $lk(K_2,K_3) = 4$. Applying Algorithm 2.10 returns $K\{2,5\}$ as the only torus knot without trivial iterations and with polynomial $\Delta_{K_2}(t)$. So the link from the closure of β is indeed a candidate for algebraic link.

Example 3.15. A significantly harder link is given by

$$\begin{aligned}\beta = {}& \sigma_5^{-1}\sigma_7^{-1}\sigma_6^{-1}\sigma_8^{-1}\sigma_7^{-1}\sigma_8^{-1}\sigma_6^{-1}\sigma_4^{-1}\sigma_5^{-1}\sigma_9^{-1}\sigma_6^{-1}\sigma_8^{-1}\sigma_{10}^{-1}\sigma_9^{-1}\sigma_7^{-1}\sigma_8^{-1}\sigma_9^{-1}\sigma_6^{-1}\sigma_{11}^{-1}\sigma_{10}^{-1}\\ & \sigma_9^{-1}\sigma_7^{-1}\sigma_6^{-1}\sigma_8^{-1}\sigma_5^{-1}\sigma_7^{-1}\sigma_6^{-1}\sigma_2^{-1}\sigma_1^{-1}\sigma_5^{-1}\sigma_2^{-1}\sigma_3^{-1}\sigma_2^{-1}\sigma_4^{-1}\sigma_1^{-1}\sigma_3^{-1}\sigma_5^{-1}\sigma_2^{-1}\sigma_4^{-1}\sigma_1^{-1}\\ & \sigma_3^{-1}\sigma_2^{-1}\sigma_6^{-1}\sigma_7^{-1}\sigma_5^{-1}\sigma_8^{-1}\sigma_6^{-1}\sigma_9\sigma_4\sigma_8\sigma_5\sigma_7\sigma_9\sigma_6\sigma_5\sigma_7\sigma_8\sigma_4\sigma_9\sigma_7\sigma_{10}\sigma_3\sigma_9\sigma_2\sigma_{11}\sigma_8\sigma_4\sigma_1\sigma_{10}\sigma_3\sigma_9\sigma_2\\ & \sigma_5\sigma_4\sigma_3\sigma_8\sigma_7\sigma_4\sigma_5\sigma_6\sigma_5\sigma_4\sigma_5\sigma_7\sigma_8\sigma_6\sigma_7\sigma_5\sigma_6\sigma_5\sigma_3\sigma_4\sigma_5\sigma_2\sigma_9\sigma_3\sigma_{10}\sigma_1\sigma_4\sigma_8\sigma_{11}\sigma_2\sigma_9\sigma_3\sigma_{10}\sigma_7\sigma_9\\ & \sigma_4\sigma_8\sigma_7\sigma_5\sigma_6\sigma_9\sigma_5\sigma_7\sigma_4\sigma_8\sigma_9\sigma_3\sigma_1\sigma_8\sigma_5\sigma_{10}\sigma_2\sigma_6\sigma_9\sigma_{10}\sigma_9\sigma_4\sigma_{10}\sigma_9\sigma_3\sigma_8\sigma_5\sigma_7\sigma_6\sigma_7\sigma_2\sigma_4\sigma_9\sigma_5\sigma_8\\ & \sigma_9\sigma_4\sigma_3\sigma_4\sigma_{10}\sigma_2\sigma_1\sigma_{10}\sigma_{11}\sigma_2\sigma_8\sigma_{10}^{-1}\sigma_3\sigma_9\sigma_4\sigma_5\sigma_{10}\sigma_6\sigma_7\sigma_3\sigma_5\sigma_4\sigma_6\sigma_5\sigma_6\sigma_4\sigma_3\sigma_4\sigma_8\sigma_9\sigma_2\sigma_7\sigma_1\sigma_8\sigma_6\sigma_3\sigma_5\sigma_{10}\end{aligned}$$

This braid gives the permutation $(1,8,2,9,3,6)(4,11,12,5,7,10)$, so there are only two components $K_1 = (1,8,2,9,3,6)$ and $K_2 = (4,11,12,5,7,10)$. They have linking number 23, but when taking the Alexander polynomials something strange happens. The second component ir a torus link $K\{2,9\}$, but $\Delta_{K_1}(t) = t^{20} - t^{19} + t^{16} - t^{15} + 2t^{13} - 5t^{12} + 9t^{11} - 11t^{10} + 9t^9 - 5t^8 + 2t^7 - t^5 + t^4 - t + 1$, which is not cyclotomic. So K_1 couldn't be an algebraic knot.

Example 3.16. Obtaining the Alexander polynomial and linking number is easy with the methods described above. Now we will give a practical example, where we will apply our methods to L, closure of the braid $\beta = (\sigma_3\sigma_4\sigma_5\sigma_2\sigma_3\sigma_4\sigma_1\sigma_2\sigma_3)^2(\sigma_4\sigma_5)^{13}$. This time, we will show the procedure more in depth, to be able to glimpse the inner workings of the algorithms. The first Algorithm is 3.6.

$$\begin{aligned}&(1)(2)(3)(4)(5)(6)(3,4)(4,5)(5,6)(2,3)(3,4)(4,5)(1,2)(2,3)(3,4)\cdots\\ &\cdots(3,4)(4,5)(5,6)(2,3)(3,4)(4,5)(1,2)(2,3)(3,4)(5,6) = (1)(2)(3)(4,5,6)\end{aligned}$$

There are four components, $K_1 = (1)$, $K_2 = (2)$, $K_3 = (3)$ and $K_4 = (3,4,5)$. Let's find the pairwise linking numbers by applying Algorithm 3.8. The indices are:

$$\begin{aligned}&(1,2,3,4,4,4) \to_{(3,4)} (1,2,4,3,4,4) \to_{(3,4)} (1,2,4,4,3,4) \to_{(3,4)} (1,2,4,4,4,3) \to_{(2,4)}\\ &(1,4,2,4,4,3) \to_{(2,4)} (1,4,4,2,4,3) \to_{(2,4)} (1,4,4,4,2,3) \to_{(1,4)} (4,1,4,4,2,3) \to_{(1,4)}\\ &(4,4,1,4,2,3) \to_{(1,4)} (4,4,4,1,2,3) \to_{(4,1)} (4,4,1,4,2,3) \to_{(4,2)} (4,4,1,2,4,3) \to_{(4,3)}\\ &(4,4,1,2,3,4) \to_{(4,1)} (4,1,4,2,3,4) \to_{(4,2)} (4,1,2,4,3,4) \to_{(4,3)} (4,1,2,3,4,4) \to_{(4,1)}\\ &\qquad (1,4,2,3,4,4) \to_{(4,2)} (1,2,4,3,4,4) \to_{(4,3)} (1,2,3,4,4,4)\end{aligned}$$

The linking numbers are:

$$lk(K_1,K_2) = lk(K_1,K_3) = lk(K_2,K_3) = 0 \text{ and } lk(K_1,K_4) = lk(K_2,K_4) = lk(K_3,K_4) = 3$$

Notice that here we have an issue similar to the one in the link from Fig. 3.3. So the closure of β is not an iterated torus link. Now for the Alexander polynomials. Clearly the first three components are trivial and it is only necessary to calculate $\Delta_{K_4}(t)$. After running Algorithm 3.9, the corresponding reduced word of K_4 is $(\sigma_1\sigma_2)^{13}$ and the Alexander polynomial is

$$\tilde{\tau}\left((\sigma_1\sigma_2)^{13}\right) = \begin{pmatrix} -t^{13} & t^{14} \\ -t^{12} & 0 \end{pmatrix}, \quad \det\left(I_2 - \tilde{\tau}((\sigma_1\sigma_2)^{13})\right) = t^{26}+t^{13}+1, \quad \Delta_{K_4}(t) = \frac{(t^{26}+t^{13}+1)(t-1)}{t^3-1}$$
$$\Delta_{K_4}(t) = t^{24}-t^{23}+t^{21}-t^{20}+t^{18}-t^{17}+t^{15}-t^{14}+t^{12}-t^{10}+t^9-t^7+t^6-t^4+t^3-t+1$$

The only non-trivial component is K_4, and its iterated torus form is $K\{3,13\}$.

Example 3.17. After properly examining the obtention of the braid in Example 3.15, another braid is given, this time presumed to be algebraic. The new braid is:

$$\begin{aligned}\beta = {} & \sigma_6\sigma_7\sigma_3\sigma_4\sigma_8\sigma_2\sigma_5\sigma_3\sigma_9\sigma_4\sigma_1\sigma_2\sigma_8\sigma_1\sigma_3\sigma_{10}\sigma_5\sigma_6\sigma_5\sigma_4\sigma_7\sigma_{11}\sigma_5\sigma_6\sigma_5\sigma_8\sigma_7\sigma_9\sigma_8\sigma_9\sigma_{10}\sigma_9\sigma_8\sigma_6\sigma_7\sigma_{11} \\ & \sigma_8\sigma_{10}\sigma_2\sigma_9\sigma_4\sigma_3\sigma_2\sigma_4\sigma_5\sigma_1\sigma_6\sigma_2\sigma_3\sigma_4\sigma_2\sigma_3\sigma_2\sigma_5\sigma_4\sigma_3\sigma_{10}\sigma_8\sigma_7\sigma_8\sigma_6\sigma_7\sigma_5\sigma_6\sigma_8\sigma_1\sigma_2\sigma_9\sigma_4\sigma_{10}\sigma_8\sigma_{11}\sigma_9\sigma_7 \\ & \sigma_3\sigma_5\sigma_1\sigma_8\sigma_{10}\sigma_4\sigma_9\sigma_2\sigma_8\sigma_1\sigma_6\sigma_5\sigma_7\sigma_6\sigma_8\sigma_7\sigma_8\sigma_{10}\sigma_3\sigma_4\sigma_5\sigma_3\sigma_4\sigma_3\sigma_2\sigma_3\sigma_4\sigma_6\sigma_5\sigma_1\sigma_4\sigma_2\sigma_3\sigma_9\sigma_4\sigma_2\sigma_8\sigma_{10}\sigma_7 \\ & \sigma_{11}\sigma_6\sigma_{10}\sigma_9\sigma_8\sigma_{10}\sigma_9\sigma_{10}\sigma_7\sigma_8\sigma_5\sigma_6\sigma_5\sigma_{11}\sigma_7\sigma_4\sigma_5\sigma_6\sigma_5\sigma_{10}\sigma_3\sigma_8\sigma_1\sigma_2\sigma_1\sigma_4\sigma_9\sigma_3\sigma_5\sigma_2\sigma_8\sigma_4\sigma_3\sigma_7\sigma_6\sigma_5\sigma_7 \\ & \sigma_6\sigma_8\sigma_7\sigma_8\sigma_6\sigma_4\sigma_5\sigma_9\sigma_6\sigma_8\sigma_{10}\sigma_9\sigma_7\sigma_8\sigma_9\sigma_6\sigma_{11}\sigma_{10}\sigma_9\sigma_7\sigma_6\sigma_8\sigma_5\sigma_7\sigma_6\sigma_2\sigma_3\sigma_1\sigma_4\sigma_2\sigma_5\sigma_3\sigma_1\sigma_4\sigma_2\sigma_3\sigma_2\sigma_5 \\ & \sigma_1\sigma_2\sigma_6\sigma_7\sigma_5\sigma_8\sigma_6\sigma_7\sigma_9\sigma_{10}\sigma_{11}\sigma_6\sigma_9\sigma_8\sigma_7\sigma_9\sigma_{10}\sigma_8\sigma_6\sigma_9\sigma_5\sigma_4\sigma_6\sigma_8\sigma_7\sigma_8\sigma_6\sigma_7\sigma_5\end{aligned}$$

The closure of this braid has three components, all of them the iterated torus knot $K((2,3),(2,7))$. Their pairwise linking numbers are $lk(K_1,K_2) = lk(K_1,K_3) = 26$ and $lk(K_2,K_3) = 28$. So it can be said that the algebraic link from which they come is the one with these iterations and linking numbers.

Example 3.18. Now will be shown why the Alexander polynomial alone isn't a complete algebraic link invariant.

$$\begin{aligned}\beta = {} & \sigma_1\sigma_3\sigma_5\sigma_7\sigma_2\sigma_3\sigma_6\sigma_7\sigma_1\sigma_4\sigma_3\sigma_5\sigma_4\sigma_3\sigma_5\sigma_6\sigma_5\sigma_4\sigma_7\sigma_6\sigma_5\sigma_1\sigma_3\sigma_4\sigma_3\sigma_7\sigma_2\sigma_1\sigma_3\sigma_2\sigma_4 \\ & \sigma_3\sigma_5\sigma_4\sigma_3\sigma_5\sigma_6\sigma_5\sigma_7\sigma_6\sigma_2\sigma_5\sigma_1\sigma_2\sigma_3\sigma_2\sigma_7\sigma_5\sigma_1\sigma_7\sigma_2\sigma_3\sigma_2\sigma_1\sigma_5\sigma_2\sigma_6\sigma_7\sigma_5\sigma_6\sigma_5\sigma_3\sigma_4 \\ & \sigma_5\sigma_3\sigma_4\sigma_2\sigma_3\sigma_1\sigma_2\sigma_7\sigma_3\sigma_4\sigma_3\sigma_1\sigma_5\sigma_6\sigma_7\sigma_4\sigma_5\sigma_6\sigma_5\sigma_3\sigma_4\sigma_5\sigma_3\sigma_4\sigma_1\sigma_7\sigma_6\sigma_3\sigma_2\end{aligned}$$

The previous link has $lk(K_1,K_2) = 16$ with components $K((2,3),(2,17))$ and $K((2,5),(2,11))$. The Alexander polynomial of the whole link is $\Delta_L(t) = -t^{92}+t^{84}-t^{82}-t^{78}+t^{74}+t^{70}-t^{68}-t^{63}-t^{61}+t^{60}+t^{55}-t^{51}-t^{49}-t^{47}+t^{45}+t^{43}+t^{41}-t^{37}-t^{32}+t^{31}+t^{29}+t^{24}-t^{22}-t^{18}+t^{14}+t^{10}-t^8+1$.

Example 3.19. Now consider the following braid.

$$\begin{aligned}\beta = {} & \sigma_6\sigma_2\sigma_1\sigma_3\sigma_5\sigma_7\sigma_2\sigma_6\sigma_1\sigma_5\sigma_4\sigma_3\sigma_4\sigma_7\sigma_5\sigma_4\sigma_6\sigma_1\sigma_5\sigma_7\sigma_6\sigma_4\sigma_3\sigma_4\sigma_5\sigma_4\sigma_7\sigma_3\sigma_2\sigma_3\sigma_1 \\ & \sigma_2\sigma_4\sigma_3\sigma_4\sigma_5\sigma_4\sigma_6\sigma_5\sigma_7\sigma_6\sigma_3\sigma_5\sigma_2\sigma_1\sigma_3\sigma_2\sigma_3\sigma_7\sigma_1\sigma_5\sigma_7\sigma_3\sigma_2\sigma_3\sigma_1\sigma_2\sigma_5\sigma_3\sigma_6\sigma_5\sigma_7\sigma_6 \\ & \sigma_4\sigma_5\sigma_4\sigma_3\sigma_4\sigma_2\sigma_1\sigma_3\sigma_2\sigma_3\sigma_7\sigma_4\sigma_5\sigma_4\sigma_3\sigma_4\sigma_6\sigma_7\sigma_5\sigma_1\sigma_6\sigma_4\sigma_5\sigma_7\sigma_4\sigma_3\sigma_4\sigma_5\sigma_1\end{aligned}$$

The components are $K((2,3),(2,15))$ and $K((2,5),(2,13))$, so it's not equivalent to the one in Example 3.18. But the Alexander polynomial of the link is the same as the one in Example 3.18. Hence, it is proved that the Alexander polynomial of the link is not a complete invariant of the whole link. Here the calculation is done with the Burau representative of the link. For more information on the obtention of these specific braids, see [3, 7].

Example 3.20. Taking the equations from Remark 1.28 result in the braids $\beta_1 = \sigma_2\sigma_3\sigma_1\sigma_2\sigma_3^3$, $\beta_2 = \sigma_1^5$ and $\beta_3 = (\sigma_1\sigma_2\sigma_3\sigma_4)^2$. Applying Algorithms 3 and 3.9 shows that the knots are indeed equivalent.

Appendix A

Algorithms in Sagemath

A.1. Reduced Burau matrix

```
def burau(n,j):
    C=identity_matrix(LR,n)
    j1=abs(j)
    e1=sign(j)
    if j1==1:
        C[0,0]=-t
        C[1,0]=-1
    elif 1<j1<n:
        C[j1-2,j1-1]=-t
        C[j1-1,j1-1]=-t
        C[j1,j1-1]=-1
    elif j1==n:
        C[j1-2,j1-1]=-t
        C[j1-1,j1-1]=-t
    if e1==1:
        return C
    elif e1==-1:
        return det(C)^-1*C.adjoint()
LR.<t>=LaurentPolynomialRing(QQ)
R=LR.polynomial_ring()
def alex_poly(alex_words,components):
    for i in range(len(components)):
        l=len(alex_words[i])
        if l==0 or l==1:
            alex_words[i]=1
        else:
            n=max(alex_words[i])
            I=identity_matrix(LR,n)
            B=prod([burau(n,j) for j in alex_words[i]])
            pol=det(I-B)
            pol0=normalize(pol)
            alex_words[i]=R(pol0*(t-1)/(t^(n+1)-1))
    return alex_words
```

A.2. Alexander polynomial from reduced Burau representation

```
def ordered_components(braid,dimension):
    cycles=BraidGroup(dimension)(braid).permutation().cycle_tuples()
    max_comp=len(cycles)
    components=[]
    for i in range(max_comp):
        component=list(cycles[i])
        components.append(component)
    return components
def rename_indices(components,dimension):
    ind=[]
    for i in [1..dimension]:
        ind.append(i)
    for i in [0..len(components)-1]:
        for j in [0..len(components[i])-1]:
            ind[components[i][j]-1]=i+1
    return ind
def normalize(pol):
    return pol.polynomial_construction()[0]
```

A.3. Component permutation of disjoint links

```
def order_boxes(matrix):
    dim=matrix.dimensions()[0]
    perm_total=identity_matrix(dim)
    i=1
    while i<dim:
        r=0
        for j in [i+1..dim]:
            if matrix[i-1,j-1]!=0:
                perm=elementary_matrix(dim,row1=i+r,row2=j-1)
                perm_total*=perm
                matrix=perm*matrix*perm
                r+=1
        box=matrix.matrix_from_rows_and_columns([i-1..i+r-1],[i-1..i+r-1])
        box+=identity_matrix(r+1)
        if exists(box.list(),lambda i:i==0)[0]:
            return matrix,'not␣algebraic'
        if i+r+1>dim:
            return matrix,perm_total*vector([1..dim])
        square=matrix.matrix_from_rows_and_columns([i-1..i+r-1],[i+r..dim-1])
        if exists(square.list(),lambda i:i!=0)[0] and square!=[]:
            return matrix,'not␣disjoint'
        i+=r+1
    return matrix,perm_total*vector([1..dim])
```

A.4. Alexander polynomial and linking matrix from braid

```
def alexander_and_linking(braid):
    dimension=max(abs(term) for term in braid)+1
```

```
    components=ordered_components(braid,dimension)
    ind=rename_indices(components,dimension)
    r=len(components)
    L=matrix(r)
    alex_words=[]
    for a in range(r):
        alex_words.append([])
    for a in range(len(braid)):
        i=abs(braid[a])
        j=ind[i-1]
        s=ind[i]
        if braid[a]>0:
            if j==s:
                alex_words[j-1].append(ind[:i].count(j))
            else:
                L[s-1,j-1]+=1
        else:
            if j==s:
                alex_words[j-1].append(-ind[:i].count(j))
            else:
                L[j-1,s-1]+=-1
        ind[i-1],ind[i]=ind[i],ind[i-1]
    box_L,perm=order_boxes(L)
    if perm=='not␣algebraic':
        return alex_poly(alex_words,components),L,'not␣algebraic'
    if perm=='not␣disjoint':
        return alex_poly(alex_words,components),L,'not␣disjoint'
    box_alex_poly=[]
    alex_polys=alex_poly(alex_words,components)
    for i in range(len(alex_polys)):
        box_alex_poly.append(alex_polys[perm[i]-1])
    return box_alex_poly,box_L
```

A.5. Iterations from Alexander polynomial

```
def product_cyclotomics(pol):
    if pol==1:
        return [1,1]
    cyclotomic=pol.is_cyclotomic_product()
    if cyclotomic:
        list=pol.factor()
        res=[]
        for p,j in list:
            res.append((p.is_cyclotomic(certificate=True),j))
        return res
    else:
        return [[0,0]]
def reduce_alex_pol(pol):
    list=product_cyclotomics(pol)
    l1=[_[0] for _ in list]
    a=max(l1)
    if a==0:
```

```
        return [1,'not␣algebraic']
    else:
        b=max(j for j in a.divisors() if j not in 11)
        q=ZZ(a/b)
        pol_reduced=R(pol*(t^b-1)/(t-1)/(t^a-1)*(t^q-1))
        pol_q=vector(pol_reduced.coefficients())*vector([R.gen(0)^(j/q)
                                    for j in pol_reduced.exponents()])
        return [pol_q,q,b]
def alex_to_it(pol):
    it=[]
    if pol==1:
        return [['trivial']]
    while pol!=1:
        reduced=reduce_alex_pol(pol)
        it.insert(0,reduced[1:])
        pol=reduced[0]
    return it
def proper_longitude(it):
    for r in range(len(it)):
        if len(it[r])>1:
            n=[]
            q=[]
            m=[]
            for i in range(len(it[r])):
                n.append(it[r][i][0])
                q.append(it[r][i][1])
            m.append(q[0])
            if len(it[r])==2:
                m.append(q[1]-m[0]*n[1]*(n[0]-1))
            for i in [2..len(it[r])-1]:
                m.append(q[i-1]-m[i-2]*n[i-1]*(n[i-2]-1)-sum([m[j-2]*n[i-1]*
                                        (n[j-2]-1)*prod([n[k-1]^2 for k
                                        in [j..k-2]]) for j in [0..i-3]]))
        for i in range(len(it[r])):
            it[r][i][1]=m[i]
    return it
```

A.6. Iterations and linking matrix from braid

```
def it_and_linking(braid):
    out=alexander_and_linking(braid)
    alex_pols=out[0]
    L=out[1]
    it=[]
    for i in range(len(alex_pols)):
        it.append(alex_to_it(alex_pols[i]))
    if len(out)==3:
        return it,L,out[2]
    return proper_longitude(it),L
```

Bibliography

[1] N. A'CAMPO, Sur la monodromie des singularités isolées d'hypersurfaces complexes, *Inventiones math.*, **20** (1973) 147-169.

[2] C. C. ADAMS, *The Knot Book*, American Mathematical Society, Providence, 2004.

[3] E. ARTAL-BARTOLO, Forme de Seifert des singularités de surface, *C. R. Acad. Sci. Paris, Série I*, **313** (1991) 689-692.

[4] E. ARTIN, Theorie der Zöpfe, *Abh. Math. Sem. Univ. Hamburg*, **4** (1) (1925) 27-72.

[5] E. ARTIN and R. FOX, Some wild cells and spheres in three-dimensional space, *Ann. of Math*, **49** (2) (1948) 979-990, http://www.jstor.org/stable/1969408.

[6] J. S. BIRMAN, *Braids, Links, and Mapping Class Groups*, Princeton University Press, 1975.

[7] P. D. BOIS and F. MICHEL, Sur la forme de Seifert des entrelacs algébriques, *C. R. Acad. Sci. Paris, Série I*, **313** (1991) 297-300.

[8] G. BURDE and H. ZIESCHAN, *Knots*, Walter de Gruyter, Berlin, 2003.

[9] J. I. COGOLLUDO-AGUSTÍN, Braid Monodromy of Algebraic Curves, *Annales Mathématiques*, **18** (1) (2011) 141-209, http://ambp.cedram.org/item?id=AMBP_2011__18_1_141_0.

[10] D. EISENBUD and W. NEUMANN, Three-Dimensional Link Theory and Invariants of Plane Curve Singularities, Princeton University Press, 1985.

[11] R. H. FOX and G. TORRES, Dual presentations of the group of a knot, *Ann. of Math*, **59**, 2 (1954) 211-218.

[12] F. A. GARSIDE, The braid group and other groups, *Quart. J. Math. Oxford Ser.*, **2**, 20 (1969) 235-254, http://www.maths.ed.ac.uk/~aar/papers/garside.pdf.

[13] J. GONZÁLEZ-MENESES, Basic results on braid groups, *Annales Mathématiques*, **18** (1) (2011) 15-59, http://ambp.cedram.org/item?id=AMBP_2011__18_1_15_0.

[14] J. HASS, J. C. LAGARIAS and N. PIPPENGER, The Computational Complexity of Knot and Link Problems, *Journal of the ACM*, **46** (2) (1999) 185-211.

[15] V. F. R. JONES, Hecke Algebra Representations of Braid Groups and Link Polynomials, *Annals of Mathematics*, **126** (2) (1987) 335-388. http://www.jstor.org/stable/1971403.

[16] L. H. KAUFFMAN, *On Knots*, Princeton University Press, USA, 1987.

[17] LÊ D. T., Sur les nœuds algebriques, *Composition Mathematica*, **25** 3 (1972) 281-321, http://www.numdam.org/item/CM_1972__25_3_281_0.

[18] M. LEJEUNE, *Sur l'équivalence des singularités des courbes algebroides planes, Coefficients de Newton*, Centre de Math. de l'Ecole Polytechnique, 1969.

[19] W. B. R. LICKORISH and K. C. MILLETT, The New Polynomial Invariants of Knots and Links, *Mathematics Magazine*, **61**, (1) (1988) 3-23, http://www.jstor.org/stable/2690324.

[20] V. O. MANTUROV, *Knot Theory*, CRC Press, USA, 2004.

[21] S. V. MATVEEV, *Algorithmic Topology and Classification of 3-Manifolds (Algorithms and Computation in Mathematics)*, Springer-Verlag, Secaucus, 2007.

[22] J. W. MILNOR, *Infinite cyclic coverings*, Conf. on topology of manifolds, Weber and Schmidt, 1968.

[23] H. R. MORTON, Threading knot diagrams, *Math. Proc. Camb. Phil. Soc.*, **99** (1986) 247-260.

[24] K. MURASUGI, *Knot theory and its applications*, Birkhäuser, Boston, 1996.

[25] K. MURASUGI, On the braid index of alternating links, *Trans. Amer. Math. Soc.*, **326** (1) (1991) 237-260.

[26] W. NEUMANN, *Topology of hypersurface singularities*, Erich Kaehler: Mathematische Werke de Gruyter, 2003.

[27] V. V. PRASOLOV, *Knots, Liks, Braids and 3-Manifolds*, American Mathematical Society, Providence, 1996.

[28] J. H. PRZYTYCKI, Classical Roots of Knot Theory, *Chaos, Solitions & Fractals*, **9** (4/5) (1998) 531-545.

[29] J. E. REEVE, A summary of results in the topological classification of plane algebroid singularities, *Rendiconti Sem. Math. Torino*, **14** (1954) 159-187.

[30] D. ROLFSEN, *BRAIDS: summer school and workshop*, National University of Singapore, 2007.

[31] D. ROLFSEN, *Knots and Links*, AMS Chelsea Publishing, Providence, 2003.

[32] H. SEIFERT, On the homology invariants of knots, *Quart. J. Math. Oxford*, **1** 2 (1950) 23-32, http://www.maths.ed.ac.uk/~aar/papers/seifert2.pdf.

[33] M. YAMAMOTO, Classification of isolated algebraic singularities by their Alexander polynomials, *Topology*, **23** (3) (1984) 277-287.

[34] O. ZARISKI, General Theory of Saturation and of Saturated Local Rings II: Saturated Local Rings of Dimension 1, *American Journal of Mathematics*, **93** (4) (1971) 872-964, www.jstor.org/stable/2373741

Index