

Analyzing Klein Knots and Torus Knots: Algorithmic Methods for Polynomial Computations

Riddhiman Bhattacharya

July 16, 2023

1 Basics of Klein Links

A knot in mathematics is an embedding of a circle S_1 in \mathbb{R}_3 . We can extend this definition of a knot to the idea of torus links. The (m, n) torus link winds m times around the meridian and n times around the longitude. A known result is that the (m, n) torus link has $\gcd(m, n)$ components.

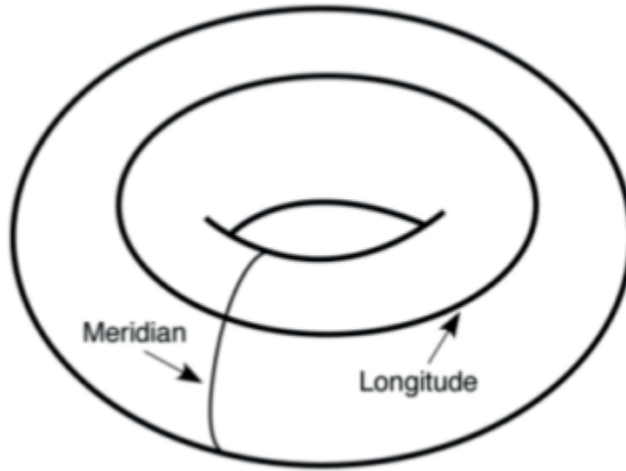


Figure 1: Surface of a torus, showing the meridian and longitude.[1]

If we cut a torus with a torus link on it along the longitude, we can mold the surface into a cylinder. We can then cut the cylinder along the longitude and deform it into a rectangle, with orientation:

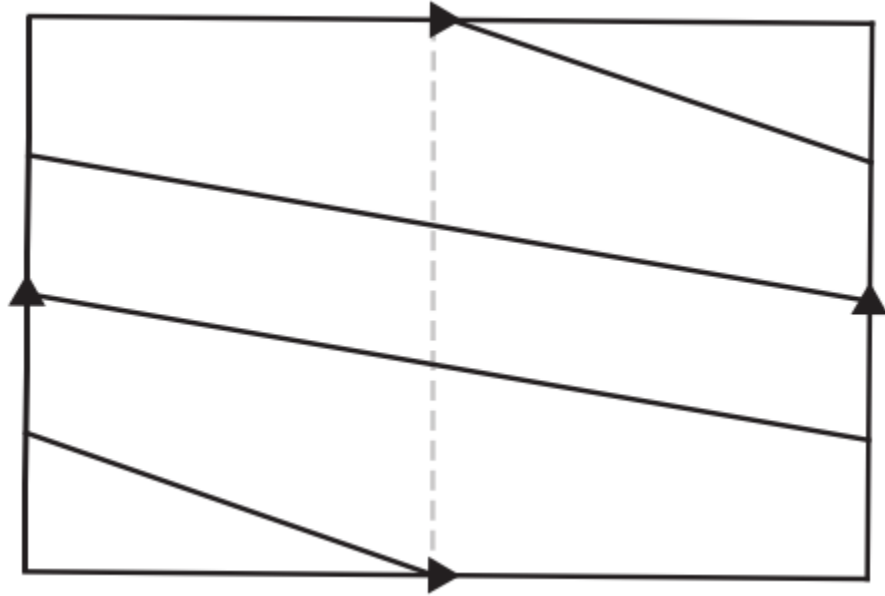


Figure 2: Cutting and deforming a torus with a knot lying on the surface into a rectangle.[1]

We can reverse the orientation of the bottom edge of the rectangle. Gluing back together the vertical sides creates a cylinder with edges in opposite orientations. To ensure that we connect this cylinder back to itself correctly, we must twist one side of the cylinder, by passing the cylinder through itself.

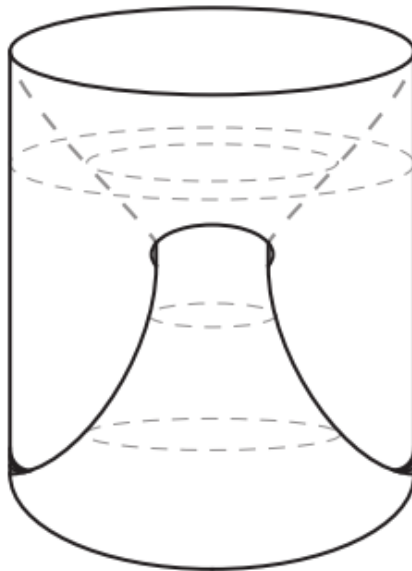


Figure 3: 3-dimensional representation of a Klein bottle. [1]

This results in a hole, because of the intersection of the cylinder with itself. We place this hole in the upper left corner of the rectangle.

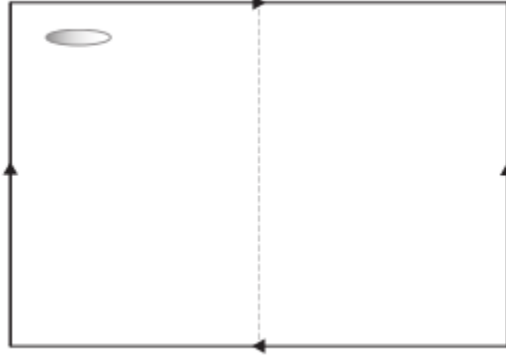


Figure 4: The surface of a Klein bottle, is represented as a rectangle.[1]

For example, the $K(4, 3)$ is shown below on the rectangle.

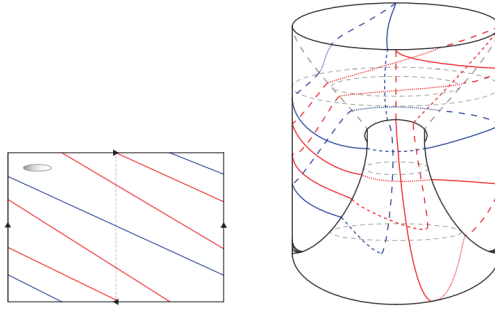


Figure 5: Example of the $K(4, 3)$ Klein link, represented on the rectangle.[5]

We ensure that no strand on the rectangle is on the hole by ensuring that all strands connected to the top edge of the rectangle are above the hole, and all strands connected to the left edge of the rectangle are below the hole.

We can now discuss the braid representation of a Klein link. It is a well-known fact that the braid representation of the (m, n) torus knot is $(\sigma_1 \dots \sigma_{n-1})^m$. In order to represent the (m, n) Klein link, we compose the representation for the torus knot with a 180° half-twist to the left as we look down on the cylinder. This results in $(\sigma_1 \dots \sigma_{n-1})^m \prod_{i=1}^{n-1} (\sigma_{n-1}^{-1} \dots \sigma_i^{-1})$.

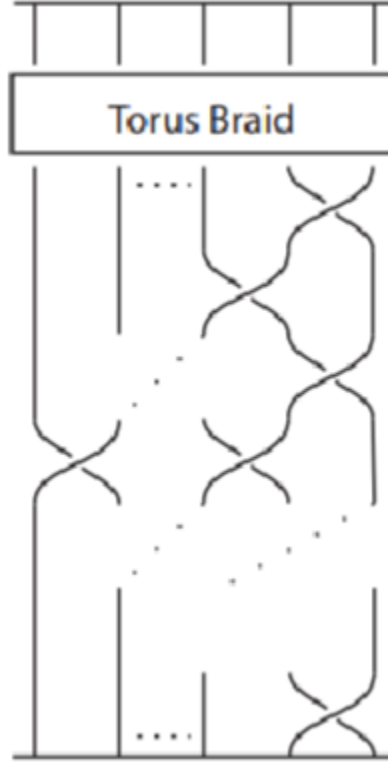


Figure 6: The twist that forms the braid representation of the Klein link.[1]

Given this braid representation of a Klein knot, a connection between $K(m, 2)$ and $T(m - 1, 2)$ becomes clear.

Theorem 1.1. [1] The $K(m, 2)$ Klein link is isotopic to the $T(m - 1, 2)$ torus link.

Proof. This proof is straightforward and can be done by examining their braid representations.

The $K(m, 2)$ Klein link can be represented as $(\sigma_1)^m(\sigma_1^{-1}) = \sigma_1^{m-1}$, while the $T(m - 1, 2)$ torus link is represented as σ_1^{m-1} . They are equivalent. \square

Having reviewed some established properties of Klein Links, we'll now aim to study the genus of Klein Links.

2 Minimal Genus of a Klein Link

To calculate the genus of our Klein Links, we'll need the number of components in each link. It is a well-known fact that the number of components in a (m, n) -torus link is the greatest common divisor of m and n . So, we must simply analyze what the half twist does to the number of components in the link.

2.1 Components in a Klein Link

An interesting but clever proof in Bowen's paper proves the following theorem.

Theorem 2.1. (*Bowen13*) Let $m, n \in \mathbb{N}$. If m is even, then the (m, n) -Klein link has $\lceil \frac{n}{2} \rceil$ components. If m is odd, then the (m, n) -Klein link has $\lceil \frac{n+1}{2} \rceil$ components. We'll denote the number of components $C(m, n)$.

To prove this theorem, we'll set up some machinery.

First, note there is a natural homomorphism from the braid group on n -strands to the symmetry group on n elements: $\pi : B_n \rightarrow S_n$, which maps $\sigma_i \mapsto (i \ i+1)$. So we will define $T_n^m := \pi(T(m, n)) = (1 \ n \ n-1, \dots, 2)^m$ and $K_n := \pi(\Delta_n) = (1 \ n)(2 \ n-1) \dots (\lceil \frac{n}{2} \rceil \lceil \frac{n+1}{2} \rceil)$. Now, that we've redefined our Torus and Klein braids as permutations, we can consider the group action of these permutations on the set $[n] := \{1, \dots, n\}$.

Definition 2.1. Let G be a group and X a set. The **group action** ϕ of G on X is the map

$$\phi : G \times X \rightarrow X : (g, x) \mapsto \phi(g, x)$$

satisfying $ex = x$ for all $x \in X$ and $(gg')x = g(g'x)$ for all $g, g' \in G$ and $x \in X$.

The **fix** of $g \in G$ is the set $\text{fix}(g) = \{x \in X : gx = x\}$. The **orbit** of x in X is the set $\text{orb}_G(x) = \{gx : g \in G\}$

With these tools, we can reframe our problem of the number of components in the language of group actions. We'll consider the group action of $\langle T_n^m K_n \rangle$ (the subgroup of S_n generated by $T_n^m K_n$) on $[n]$. The number of orbits is exactly the number of components of the Klein Link. Using **Burnside's Theorem**, this quantity is easily calculated.

Theorem 2.2. Given a group G acting on a set X , the number of orbits $|X/G|$ is given by

$$|X/G| = \frac{1}{|G|} \sum_{g \in G} |\text{fix}(g)| \quad (1)$$

To prove theorem 2.2, we'll need the order of $\langle T_n^m K_n \rangle$. It is easy to check that the order of $\langle T_n^m K_n \rangle$ is 2 if $n > 2$. With this, we can compute the number of components in a Klein Link.

Proof. First, we consider the $n = 1$ case. Trivially, $|\text{fix}(1)| = 1$, and $|\langle T_1^m K_1 \rangle| = 1$. So, the number of orbits is

$$|[1]/\langle T_1^m K_1 \rangle| = \frac{1}{1}(1) = 1$$

which is indeed $\lceil \frac{1+1}{2} \rceil$ and $\lceil \frac{1}{2} \rceil$.

Now, we consider $n = 2$. In this case, we've $|\text{fix}(1)| = 2$. If m is odd, $\langle T_2^m K_2 \rangle = (1)$, so $|\langle T_2^m K_2 \rangle| = 1$. Then,

$$|[2]/\langle T_2^m K_2 \rangle| = \frac{1}{1}(2) = 2 = \left\lceil \frac{2+1}{2} \right\rceil.$$

If m is even, $\langle T_2^m K_2 \rangle = \{(1), (1\ 2)\}$, so, $|\langle T_2^m K_2 \rangle| = 2$. Then,

$$|[2]/\langle T_2^m K_2 \rangle| = \frac{1}{2}(2 + 0) = 1 = \left\lceil \frac{2}{2} \right\rceil.$$

Now, let's consider the case $n > 2$. As stated before, $|\langle T_n^m K_n \rangle| = 2$, and $fix(1) = n$. So, all that remains is to calculate $|fix(T_n^m K_n)|$. We're looking for $i \in [n]$ such that $T_n^m K_n(i) = i$.

First, note that it follows from the permutation representations of T_n^m and K_n that $T_n^m(i) = i - m \pmod n$ and $K_n(i) = 1 - i \pmod n$. So, we're looking for solutions to the modular equation $T_n^m K_n(i) = i$.

$$T_n^m K_n(i) = T_n^m(1 - i) = 1 - i - m = 1 \pmod n$$

So, we need solutions to the equation:

$$2i = 1 - m \pmod n. \quad (2)$$

If n is odd, $\gcd(2, n) = 1$, so 2 is a unit in \mathbb{Z}_n , so the equation has a single solution, which implies that $|fix(T_n^m K_n)| = 1$. Therefore,

$$|[n]/\langle T_n^m K_n \rangle| = \frac{1}{2}(n + 1) = \left\lceil \frac{n + 1}{2} \right\rceil.$$

If both n and m are even, the $\gcd(2, n) = 2$. Because m is even, $1 - m$ is odd, which means there are no solutions to the equation. Then we've

$$|[n]/\langle T_n^m K_n \rangle| = \frac{1}{2}(n + 0) = \left\lceil \frac{n}{2} \right\rceil.$$

Finally, if n is even and m is odd, we still have $\gcd(2, n) = 2$ and $1 - m$ is even. Then, we've 2 solutions to the above equation, that implies $|fix(T_n^m K_n)| = 2$. So,

$$|[n]/\langle T_n^m K_n \rangle| = \frac{1}{2}(n + 2) = \left\lceil \frac{n + 2}{2} \right\rceil.$$

□

2.2 Minimal Genus

Definition 2.2. A *positive braid* is a braid with only positive crossings.

We state (without proof) the following theorem:

Theorem 2.3. Seifert's algorithm produces a Seifert Surface with minimal genus for the closure of positive braids.

We defined the braid representation of a Klein Link to be $K(m, n) = T(m, n)\Delta_n^{-1}$ and added a "negative" half twist in our definition of the Klein Link, which was convenient. This braid expression contains many negative crossings, so we cannot guarantee that Seifert's algorithm will produce a minimal genus surface. Fortunately, with a bit of manipulation, we can simplify our braid word and in doing so we can eliminate all minimal crossings.

Theorem 2.4. For the $K(m, n)$ ($m \geq n$) a simplified braid word is:

$$K(m, n) = (\sigma_1 \sigma_2 \dots \sigma_{n-1})^{m-n+1} (\sigma_1 \sigma_2 \dots \sigma_{n-2}) \dots \sigma_1 \quad (3)$$

Proof.

$$\begin{aligned} K(m, n) &= (\sigma_1 \sigma_2 \dots \sigma_{n-1})^m (\sigma_{n-1}^{-1} \sigma_{n-2}^{-1} \dots \sigma_1^{-1}) \dots \sigma_1^{-1} \\ &= (\sigma_1 \sigma_2 \dots \sigma_{n-1})^{m-1} (\sigma_{n-1}^{-1} \sigma_{n-2}^{-1} \dots \sigma_2^{-1}) \dots \sigma_{n-1}^{-1} \\ &= (\sigma_1 \sigma_2 \dots \sigma_{n-1})^{m-2} \sigma_1 (\sigma_{n-1}^{-1} \sigma_{n-2}^{-1} \dots \sigma_3^{-1}) \dots \sigma_{n-1}^{-1} \\ &= (\sigma_1 \sigma_2 \dots \sigma_{n-1})^{m-2} (\sigma_{n-1}^{-1} \sigma_{n-2}^{-1} \dots \sigma_3^{-1}) \dots \sigma_{n-1}^{-1} \sigma_1 \\ &= \vdots \\ &= (\sigma_1 \sigma_2 \dots \sigma_{n-1})^{m-n+1} (\sigma_1 \sigma_2 \dots \sigma_{n-2}) \dots \sigma_1 \end{aligned}$$

□

So, now with this new positive braid representation we'll be able to attack the problem of a minimal genus.

Theorem 2.4. The minimal genus g_K of a positive Klein link $K^+(m, n)$ is

$$g_K = 1 - \frac{1}{2} \left(n - (m - n + 1)(n - 1) - \frac{(n - 2)(n - 1)}{2} + C(m, n) \right) \quad (4)$$

Proof. To find the minimal genus, we'll calculate the Euler Characteristic of the surface generated by Seifert's algorithm (which by theorem 2.3 has minimal genus). ***The Euler characteristic of a Seifert surface is the number of Seifert disks, minus the number of bands*** ($\chi = D - B$). So, let's consider an (m, n) - Klein Link (figure 7).

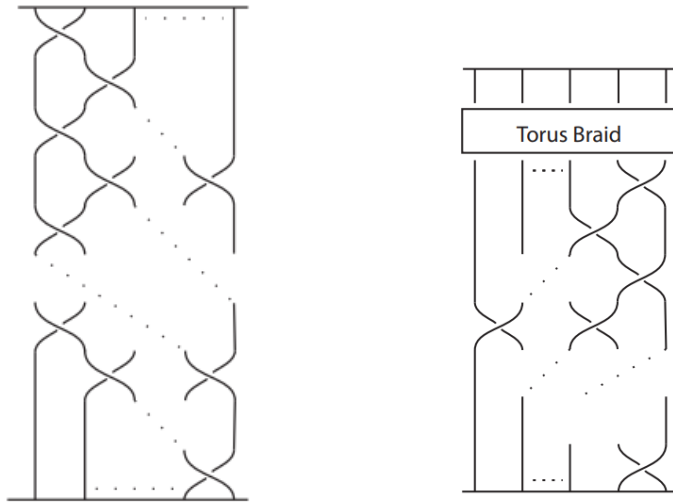


Figure 7: An oriented (m, n) - Klein link (m full twists followed by a half twist on n strings).

To begin Seifert's algorithm, we must orient our knot. Fortunately, the braid representation makes it very clear how the Seifert disks are formed. In altering the crossings, every crossing is eliminated. So we are left with the closure of the identity braid, which means there is a single Seifert disk for each strand of the braid, or $D = n$.

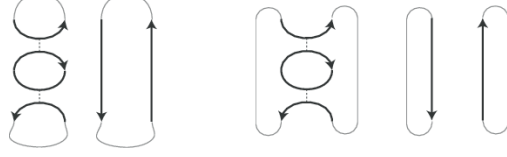


Figure 8: Removing crossings to make Seifert disks.

So, we now must simply count the number of bands. The number of bands is the number of crossings, or equivalently the number of σ_i in the braid word. Recall the braid word for a (m, n) - Klein link is:

$$K(m, n) = (\sigma_1 \sigma_2 \dots \sigma_{n-1})^{m-n+1} (\sigma_1 \sigma_2 \dots \sigma_{n-2}) (\sigma_1 \sigma_2 \dots \sigma_{n-3}) \dots \sigma_1$$

So, the no. of elements in the braid word is:

$$\begin{aligned} B &= (m - n + 1)(n - 1) + \sum_{k=1}^{n-2} k = \\ B &= (m - n + 1)(n - 1) + \frac{(n - 2)(n - 1)}{2} \end{aligned} \tag{5}$$

With this, we can now conclude the genus of $K(m, n)$.

$$\chi = D - B = n - (m - n + 1)(n - 1) - \frac{(n - 2)(n - 1)}{2} = 2 - 2g_k - C(m, n)$$

Rearranging:

$$g_K = 1 - \frac{1}{2} \left(n - (m - n + 1)(n - 1) - \frac{(n - 2)(n - 1)}{2} + C(m, n) \right).$$

□

3 Jones' Polynomial

In this section, we'll discuss finding the Jones polynomial for Klein Links. We explored a few different approaches. We'll first investigate imitating Jones' derivation for the polynomial for a torus knot, and then explore some algorithmic approaches to calculating Jones polynomials for the Klein links.

Definition 3.1. (*Jones*)

The **Jones polynomial** of a link L , $V(L)$, is a knot-invariant polynomial defined by the following properties:

- Two isotopic knots have the same Jones polynomial.
- The Jones polynomial of the unknot is 1.
- For a link L , there exists the skein relation

$$t^{-1}V(L_+) - tV(L_-) = (t^{1/2} - t^{-1/2})V(L_0)$$

where L_+ , L_- , and L_0 are three oriented diagrams of the link L , with the sole difference being at one crossing of L where L_+ , L_- , and L_0 replace the original crossing with the corresponding crossing drawn below.



Figure 9: L_+ , L_- , and L_0 replace a crossing on an oriented diagram of L with the respective crossing shown above

3.1 Bracket Polynomial

Our first method of approach was to use the **bracket polynomial**, by representing Klein links into the Temperley-Leib Algebra.

Definition 3.2 (3). A **bracket polynomial** of a link L is a polynomial in the variable A , restricted by the following three properties:

- The bracket polynomial of the unknot is 1.
- $\langle \text{red } \times \rangle = A \langle \text{red } | \text{ } | \rangle + A^{-1} \langle \text{red } \text{---} \text{---} \rangle$
- The bracket polynomial of the disjoint union of an unknot and a knot L is equal to $-A^2 - A^{-2}$ times the bracket polynomial of L .

The bracket polynomial for a link L is denoted $\langle L \rangle$.

A key issue, however, arises in that the bracket polynomial is not invariant under type-1 Reidemeister move. To solve this problem, we can use what is called a "normalized" bracket polynomial.

Definition 3.3. The **writhe** of a knot is calculated by assigning 1 to all positive crossings, -1 to all negative crossings, and adding together all of these values.

Definition 3.4 (3). The **normalized bracket polynomial** (also known as the **Kauffman X-polynomial** of a knot K , $\langle \bar{K} \rangle$, is the polynomial obtained from multiplying $\langle K \rangle$ by $(-A^3)^{-W}$, where w is the writhe of the knot.

Theorem 3.1. The normalized bracket polynomial of a knot K is knot-invariant.

Proof. To show that the normalized bracket polynomial is knot-invariant, we show that across all three types of Reidemeister moves, the polynomial doesn't change.

- We begin with the first Reidemeister move. Suppose that we have knot K , and K' is the same knot K , except for one extra Type-1 Reidemeister move added in some place. This changes a straight line into a loop with a crossing. First, let us assume that this crossing is positive.

Let the original knot K have bracket polynomial $\langle K \rangle$ and writhe w . Then, the original knot K has normalized bracket polynomial $\langle \bar{K} \rangle = (-A^3)^{-w} \langle K \rangle$.

The new knot K' has writhe $w + 1$, as we insert one positive crossing. By the second property of a bracket polynomial, we can see that

$$\langle K' \rangle = A(-A^2 - A^{-2})\langle K \rangle + A^{-1}\langle K \rangle = -A^3\langle K \rangle.$$

Thus, the normalized bracket polynomial of K' is

$$\langle \bar{K}' \rangle = (-A^3)^{-(w+1)} - A^3\langle K \rangle = (-A^3)^{-(w+1)+1}\langle K \rangle = (-A^3)^{-w}\langle K \rangle.$$

A similar proof shows that if K' is derived from K but with a single loop from a negative crossing, their normalized bracket polynomials are equal.

Thus, over the Type-1 Reidemeister move, the normalized bracket polynomial is invariant.

- We can combine the cases for the Type-2 and Type-3 Reidemeister moves. As both the writhe and the bracket polynomial are invariant over Type-2 and Type-3 Reidemeister moves, if we derive K' from a knot K (both with writhe w) by performing either a Type-2 or a Type-3 move, $\langle \bar{K}' \rangle = (-A^3)^{-w} \langle K' \rangle = (-A^3)^{-w} \langle K \rangle = \langle \bar{K} \rangle$.

Thus, over both the Type-2 and Type-3 Reidemeister moves, the normalized bracket polynomial is invariant.

Because the polynomial is invariant across all three Reidemeister moves, the normalized bracket polynomial must be knot-invariant. □

From the bracket polynomial, we must find a connection to the Jones polynomial. This is easily done by comparing the skein relations of the Jones polynomial and the bracket polynomial.

Theorem 3.2. [3]

The Jones polynomial can be found by calculating the normalized bracket polynomial, then substituting $A = t^{-1/4}$.

Proof. The skein relation of the regular bracket polynomial is

$$\langle \textcolor{red}{\times} \rangle = A\langle \textcolor{red}{|} \textcolor{red}{|} \rangle + A^{-1}\langle \textcolor{red}{\rule{0.5em}{0.4em}} \rangle.$$

Similarly, we can conclude that

$$\langle \textcolor{red}{\times} \rangle = A \langle \textcolor{red}{\sqcup} \rangle + A^{-1} \langle \textcolor{red}{\sqcap} \rangle$$

If we multiply the first line by A and the second line by A^{-1} , then subtract the second from the first, we get

$$A \langle \textcolor{red}{\times} \rangle - A^{-1} \langle \textcolor{red}{\times} \rangle = \langle \textcolor{red}{\sqcup} \rangle - \langle \textcolor{red}{\sqcap} \rangle + A^2 \langle \textcolor{red}{\mid \mid} \rangle - A^{-2} \langle \textcolor{red}{\mid \mid} \rangle$$

which simplifies to

$$A \langle \textcolor{red}{\times} \rangle - A^{-1} \langle \textcolor{red}{\times} \rangle = (A^2 - A^{-2}) \langle \textcolor{red}{\mid \mid} \rangle$$

This is nearly the same as the skein relation for the Jones polynomial. Now, let us assume that we have an arbitrary link L for which we are calculating the Jones polynomial; consider this skein relation acting on one of the crossings of L , and orient the link L such that each of the skein relation diagrams flows from bottom to top.

Let the resolved crossing L_0 replace the crossing of L (or, if there is naturally no crossing there, do nothing); and let the writhe of the resulting link be w . Then, applying the multiplication of $(-A^3)^{-w}$ for the appropriate writhe value,

$$\begin{aligned} (-A^3)^{w+1} A \langle \textcolor{red}{\times} \rangle - (-A^3)^{w-1} A^{-1} \langle \textcolor{red}{\times} \rangle &= (-A^3)^w (A^2 - A^{-2}) \langle \textcolor{red}{\mid \mid} \rangle \\ (-A^3)^1 A \langle \textcolor{red}{\times} \rangle - (-A^3)^{-1} A^{-1} \langle \textcolor{red}{\times} \rangle &= (-A^3)^0 (A^2 - A^{-2}) \langle \textcolor{red}{\mid \mid} \rangle \\ -A^4 \langle \textcolor{red}{\times} \rangle + A^{-4} \langle \textcolor{red}{\times} \rangle &= (A^2 - A^{-2}) \langle \textcolor{red}{\mid \mid} \rangle \\ A^4 \langle \textcolor{red}{\times} \rangle - A^{-4} \langle \textcolor{red}{\times} \rangle &= (A^{-2} - A^2) \langle \textcolor{red}{\mid \mid} \rangle \end{aligned}$$

From this point, it is obvious that if we let $A = t^{-1/4}$, then we have exactly the skein relation of the Jones polynomial. Thus, finding the normalized bracket polynomial then letting $A = t^{-1/4}$ gives us the Jones polynomial, as all of its properties are met. □

3.1.1 Temperley-Leib Algebra

Definition 3.5. An **algebra** over a commutative ring R is an abelian group A , which has the structure of both a ring and an R -module which satisfies, for all $r \in R$ and $a, b \in A$,

$$r(xy) = (rx)y = x(ry)$$

.

Definition 3.6. Let R be the ring $\mathbb{Z}[A, A^{-1}]$, the ring of all polynomials in terms A and A^{-1} with integer coefficients. Let us fix $\tau \in R$. The **Temperley-Lieb Algebra** $TL_n(\tau)$ is the R -algebra generated by $n - 1$ generators $\{e_1, \dots, e_{n-1}\}$, subject to the following properties:

- $e_i^2 = \tau e_i$

- $e_i e_j = e_j e_i \quad |i - j| > 1$
- $e_i e_{i\pm 1} e_i = e_i$

A **tangle** is an element of a $TL_n(\tau)$ algebra.

The i th generator of a TL_n algebra connects the top row to the bottom row as an identity mapping, except for the i th and $(i + 1)$ th nodes. Instead, the top two i th and $(i + 1)$ th node are connected, and the bottom two i th and $(i + 1)$ th node are connected. The generators for a TL_n algebra can be diagrammatically shown below.



Figure 10: The element on the left is e_1 and the element on the far right is e_{n-1} . [2]

These generators can be multiplied via the operation of concatenation. More specifically, when considering $\sigma_i \sigma_j$, we take the bottom row of σ_i and the top row of σ_j and connect them. Both rows have nodes numbered 1 to n ; we connect the rows respectively.

To connect TL_n with our braids, we must correctly pick τ such that the union of an unconnected unknot with the rest of a tangle will be equivalent to left-multiplying the tangle by τ , a valid operation of a module. Thus, we let $\tau = -A^2 - A^{-2}$, in order to agree with the properties of a bracket polynomial. .

With these generators explicitly described, the three properties described above can be shown to be true. These properties are shown on a TL_n tangle for small n , but it is clear that these properties will hold even when n is increased. First, we have $e_i^2 = \tau e_i$:

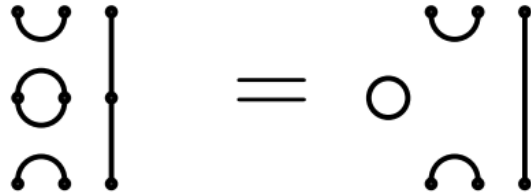


Figure 11: $e_i^2 = \tau e_i$. [2]

Next, we've $e_i e_{i\pm 1} e_i = e_i$. Only $e_i e_{i+1} e_i = e_i$ is shown here; the reader can imagine, however, how the $e_i e_{i-1} e_i = e_i$ property is true, just by flipping this diagram.

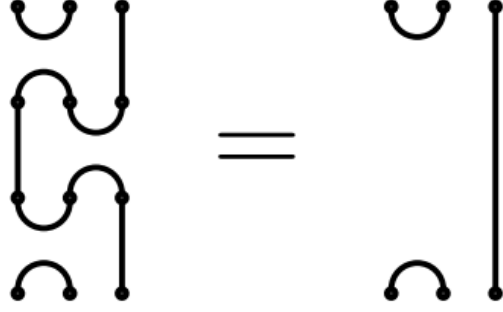


Figure 12: $e_i e_{i+1} e_i = e_i$. [2]

Lastly, we have $e_i e_j = e_j e_i$ when $|i - j| > 1$. This diagram specifically shows this property for $|i - j| = 2$, but by including straight strands in between e_i and e_j , it is clear that this property holds for all $|i - j| > 1$.

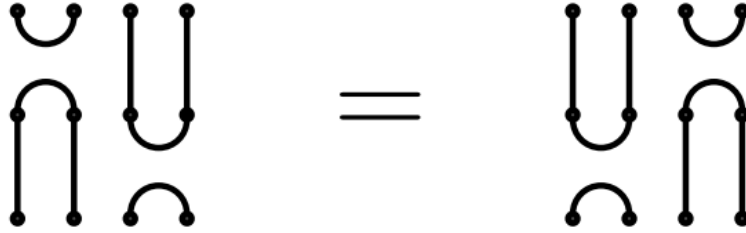


Figure 13: $e_i e_j = e_j e_i$ for $|i - j| > 1$. [2]

As the identity of $TL_n(\tau)$ is just the tangle with no generators, it is equivalent to an n -unlink, and thus is equal to the identity of the braid group. We name this identity 1_n .

Now that we have defined Temperley-Leib Algebra and seen some of its properties, we can represent n -strand braids (and, specifically, the braid representation of a Klein knot) in TL_n .

Definition 3.7. A **representation** of a group G to an R - algebra A is a homomorphism $\rho : G \rightarrow A$.

In our case, we wish to represent the braid group B_n in TL_n . We thus define a homomorphism $\rho_n : B_n \rightarrow TL_n$, where $\rho_n(\sigma_i) = A1_n + A^{-1}e_i$ and $\rho_n(\sigma_i^{-1}) = A^{-1}1_n + Ae_i$. Note that behavior of the identity is as expected; $1_n^j = 1_n$ for any $j \geq 1$, and $1_n e_i = e_i 1_n = e_i$ for any generator e_i . Most of the time, we will omit the 1_n when writing our representation; it is understood that any term without a generator is simply the 1_n term.

To show that this homomorphism is a representation of the braid group B_n , we must show that the properties of the braid group are preserved; that is, the following relations are preserved through the homomorphism:

$$\begin{cases} \sigma_i \sigma_i^{-1} = 1 & i = 1, \dots, n-1 \\ \sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1} & i = 1, \dots, n-2 \\ \sigma_i \sigma_j = \sigma_j \sigma_i & |i-j| > 1 \end{cases}$$

We tackle these cases one at a time.

- First, we show that the property that $\sigma_i \sigma_i^{-1} = 1$ is held through the homomorphism. We have

$$\begin{aligned} \rho_n(\sigma_i \sigma_i^{-1}) &= \rho_n(\sigma_i) \rho_n(\sigma_i^{-1}) \\ &= (A + A^{-1}e_i)(A^{-1} + Ae_i) \\ &= 1_n + A^2e_i + A^{-2}e_i + e_i^2 \\ &= 1_n + (A^2 + A^{-2})e_i + \tau e_i \\ &= 1_n + e_i[(A^2 + A^{-2}) + (-A^2 - A^{-2})] \\ &= 1_n + e_i \cdot 0 \\ &= 1_n \end{aligned}$$

We can see that this first property holds true.

- Next, we show that $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$.

$$\begin{aligned} \rho_n(\sigma_i \sigma_{i+1} \sigma_i) &= \rho_n(\sigma_i) \rho_n(\sigma_{i+1}) \rho_n(\sigma_i) \\ &= (A + A^{-1}e_i)(A + A^{-1}e_{i+1})(A + A^{-1}e_i) \\ &= A^3 + Ae_i + Ae_{i+1} + A^{-1}e_{i+1}e_i + Ae_i + A^{-1}e_i^2 + A^{-1}e_ie_{i+1} + A^{-3}e_ie_{i+1}e_i \\ &= A^3 + Ae_i + Ae_{i+1} + A^{-1}e_{i+1}e_i + Ae_i + A^{-1}\tau e_i + A^{-1}e_ie_{i+1} + A^{-3}e_i \\ &= A^3 + (A + A + A^{-1}\tau + A^{-3})e_i + Ae_{i+1} + A^{-1}e_{i+1}e_i + A^{-1}e_ie_{i+1} \\ &= A^3 + [2A + A^{-1}(-A^2 - A^{-2}) + A^{-3}]e_i + Ae_{i+1} + A^{-1}(e_{i+1}e_i + e_ie_{i+1}) \\ &= A^3 + (2A - A - A^{-3} + A^{-3})e_i + Ae_{i+1} + A^{-1}(e_{i+1}e_i + e_ie_{i+1}) \\ &= A^3 + Ae_i + Ae_{i+1} + A^{-1}(e_{i+1}e_i + e_ie_{i+1}) \\ &= A^3 + A(e_i + e_{i+1}) + A^{-1}(e_{i+1}e_i + e_ie_{i+1}) \end{aligned}$$

$$\begin{aligned}
\rho_n(\sigma_{i+1}\sigma_i\sigma_{i+1}) &= \rho_n(\sigma_{i+1})\rho_n(\sigma_i)\rho_n(\sigma_{i+1}) \\
&= (A + A^{-1}e_{i+1})(A + A^{-1}e_i)(A + A^{-1}e_{i+1}) \\
&= A^3 + Ae_{i+1} + Ae_i + A^{-1}e_ie_{i+1} + Ae_{i+1} + A^{-1}e_{i+1}^2 + A^{-1}e_{i+1}e_i + A^{-3}e_{i+1}e_ie_{i+1} \\
&= A^3 + Ae_{i+1} + Ae_i + A^{-1}e_ie_{i+1} + Ae_{i+1} + A^{-1}\tau e_{i+1} + A^{-1}e_{i+1}e_i + A^{-3}e_{i+1} \\
&= A^3 + (A + A + A^{-1}\tau + A^{-3})e_{i+1} + Ae_i + A^{-1}e_ie_{i+1} + A^{-1}e_{i+1}e_i \\
&= A^3 + [2A + A^{-1}(-A^2 - A^{-2}) + A^{-3}]e_{i+1} + Ae_i + A^{-1}(e_ie_{i+1} + e_{i+1}e_i) \\
&= A^3 + (2A - A - A^{-3} + A^{-3})e_{i+1} + Ae_i + A^{-1}(e_ie_{i+1} + e_{i+1}e_i) \\
&= A^3 + Ae_{i+1} + Ae_i + A^{-1}(e_ie_{i+1} + e_{i+1}e_i) \\
&= A^3 + A(e_{i+1} + e_i) + A^{-1}(e_ie_{i+1} + e_{i+1}e_i) \\
&= A^3 + A(e_i + e_{i+1}) + A^{-1}(e_{i+1}e_i + e_ie_{i+1})
\end{aligned}$$

Again, we can see that this second property holds true.

- Lastly, we show that for $|i - j| > 1$, $\sigma_i\sigma_j = \sigma_j\sigma_i$.

$$\begin{aligned}
\rho_n(\sigma_i\sigma_j) &= \rho_n(\sigma_i)\rho_n(\sigma_j) \\
&= (A + A^{-1}e_i)(A + A^{-1}e_j) \\
&= A^2 + e_j + e_i + A^{-2}e_ie_j
\end{aligned}$$

$$\begin{aligned}
\rho_n(\sigma_j\sigma_i) &= \rho_n(\sigma_j)\rho_n(\sigma_i) \\
&= (A + A^{-1}e_j)(A + A^{-1}e_i) \\
&= A^2 + e_j + e_i + A^{-2}e_j e_i \\
&= A^2 + e_i + e_j + A^{-2}e_ie_j
\end{aligned}$$

We can see that this third property holds true. Note that $e_j e_i = e_i e_j$ as this is true for $TL_n(\tau)$ when $|i - j| > 1$, which is part of our initial assumptions.

Before we may begin analyzing the bracket polynomial, we must define two more terms.

Definition 3.8. Let E be some product (concatenation) of generators of TL_n ; let the tangle represented by this product of generators be T_E . The **closure** of T_E occurs by connecting the n top ends of the tangle with the n bottom ends of the tangle correspondingly. We connect the top row of the first generator in E with the bottom row of the last generator in E , in an operation similar to concatenation.

By taking the closure of a tangle, we get a disjoint union of some number of unknots. Note that there are no crossings in any of the generators, and the concatenation operation does not create any crossings; thus, in the tangle, there are no crossings. Similarly to concatenation, the closure of a tangle does not create any tangles, as it simply creates n loops. It must then be the case that the closure of a tangle is the disjoint union of some number of unknots.

Definition 3.9. If T_E is a tangle, then the **trace** of T_E is the operation $tr(T_E)$ that counts the number of unknots in the tangle of the closure of T_E .

It may be noted that through the three properties of TL_n , the trace of a tangle does not change; in other words, the trace is tangle-invariant.

3.1.2 Bracket Polynomial, Revisited

We can now return to the bracket polynomial and evaluate the bracket polynomial of a tangle $T_E \in TL_n$. We must, however, note a few things.

Because the writhe of a knot is the number of positive crossings minus the number of negative crossings, and each knot can be represented as a braid, the writhe of a braid is similarly defined. But since every crossing in a braid is purely identified by the braid generators, the writhe of a braid is just the sum of exponents of the braid generators. For example, the braid $\sigma_1\sigma_2^{-2}\sigma_4$ has writhe 0.

As the normalized bracket polynomial is knot-invariant, it must also be braid invariant, as each knot can be represented as a braid according to a theorem by Alexander.

The bracket polynomial of a knot (or a braid) is calculated by taking each crossing, considering the different knots/links (and polynomials) obtained by resolving the crossing in two separate ways, then multiplying those polynomials by A or A^{-1} depending on the direction of the resolution. By doing so to a braid, after all crossings have been resolved, we get elements of TL_n .

In fact, in performing the representation ρ_n by mapping any σ_i to $(A1_n + A^{-1}e_i)$ and any σ_i^{-1} to $(A^{-1}1_n + Ae_i)$, we are resolving crossings in two directions, in the same fashion as the property of the bracket polynomial.

Thus, for each concatenation of TL_n generators in the representation ρ_n of a braid B , we calculate the bracket polynomial of the closure of those generators, then multiply by the corresponding coefficients of powers of A and A^{-1} . Because the closure of any tangle in TL_n is simply a disjoint union of some number of unknots, it follows that any bracket polynomial of the closure of some number of generators of TL_n must be a power of $\tau = (-A^{-2} - A^2)$.

Theorem 3.3. If E is a product of some number of generators of TL_n , then bracket polynomial of E is equal to $\tau^{tr(E)-1}$.

Proof. The bracket polynomial of the unknot is 1, and the unknot has trace 1 when represented in TL_n . This agrees with the formula of the bracket polynomial, $\tau^{tr(O)-1} = \tau^0 = 1$.

Consider a tangle T in TL_n with $tr(T) \geq 1$. Let us assume that $\tau^{tr(T)-1}$ correctly represents the bracket polynomial of T . Now consider $T' \in TL_n$, where $tr(T') = tr(T) + 1$. By the formula, the bracket polynomial of T' is $\langle T' \rangle = \tau^{tr(T')}$. By the properties of a bracket polynomial, as T' is just the disjoint union of T with an unknot, Thus, the bracket polynomial of T' is equal to $\langle T' \rangle = (-A^{-2} - A^2)\langle T \rangle = \tau \cdot \tau^{tr(T)-1} = \tau^{tr(T')}$.

The bracket polynomial derived from the formula is equal to the bracket polynomial derived from its properties; thus, by the method of induction, the bracket polynomial of a tangle $T \in TL_n$ for any n is $\langle T \rangle = \tau^{tr(T)-1}$.

□

After we obtain the bracket polynomial, we must multiply by $(-A^3)^{-w}$, where w is the writhe of the braid. As explained earlier, this value is simply the sum of the exponents over all generators in the braid polynomial.

We now have the normalized bracket polynomial $\langle \bar{K}(m, n) \rangle$ of a Klein knot; however, we must find the corresponding Jones polynomial. As shown earlier, we can simply let $A = t^{-1/4}$ to find the Jones polynomial.

Example 3.1. For an example, let us consider the Klein knot represented by $K(4, 2)$; we determined earlier that this knot is isotopic to the trefoil. The $K(4, 2)$ Klein knot has braid representation

$$B_{4,2} = (\sigma_1)^4 \sigma_1^{-1} = \sigma_1^3.$$

The TL_2 representation of $B_{4,2}$ is

$$\rho_2(B_{4,2}) = (A + A^{-1}e_1)^3 = A^3 + 3Ae_1 + 3A^{-1}e_1^2 + A^{-3}e_1^3.$$

In TL_2 , the identity element has trace 2, e_1 has trace 1, e_1^2 has trace 2, and e_1^3 has trace 3. These four can be checked by simply drawing each tangle, then counting the number of unknots. Thus, the bracket polynomial of the trefoil is

$$\langle K(4, 2) \rangle = \tau A^3 + 3A + 3\tau A^{-1} + \tau^2 A^{-3} = -A^5 - A^{-3} + A^{-7}.$$

(The writhe of the $K(4, 2)$ Klein knot is 3, since it's represented by σ_1^3 . Thus, to obtain $\langle \bar{K}(4, 2) \rangle$, we must multiply $\langle K(4, 2) \rangle$ by $(-A^3)^{-3} = -A^{-9}$, so

$$\langle \bar{K}(4, 2) \rangle = A^{-4} + A^{-12} - A^{-16}$$

Letting $A = t^{-1/4}$, we can see that the Jones polynomial of the $K(4, 2)$ knot is

$$V(K(4, 2)) = t + t^3 - t^4.$$

This agrees exactly with the confirmed Jones polynomial for the trefoil.

The largest drawback to this method of bracket polynomials and Temperley-Leib algebra is that for a given Klein knot K , if the braid representation of K contains a product of m braid generators, the polynomial $\langle K \rangle$ will have 2^m terms before simplification. Though this was not an issue for our small example of $K(4, 2)$, this can easily pose a problem for larger, more complex Klein links.

Instead of having to calculate this by hand, we created a computer program to calculate this polynomial for us. It is composed of an object that represents TL_n algebra, allowing one to input n as an argument to the object initiation. Then, given the order of TL_n generators, the object has a method to exactly determine the trace of the tangle. The program then uses these capabilities to determine the bracket (and subsequently, the Jones polynomial) of a given $K(m, n)$ Klein knot.

The hardest part of writing this program was coming up with a way to represent a ring with a non-abelian multiplicative structure, as using sympy and other regular tools assumes that your polynomials are abelian. The PolyPart class solves this issue, along with a special multiplication function to ensure that the generators e_i are kept in the correct order.

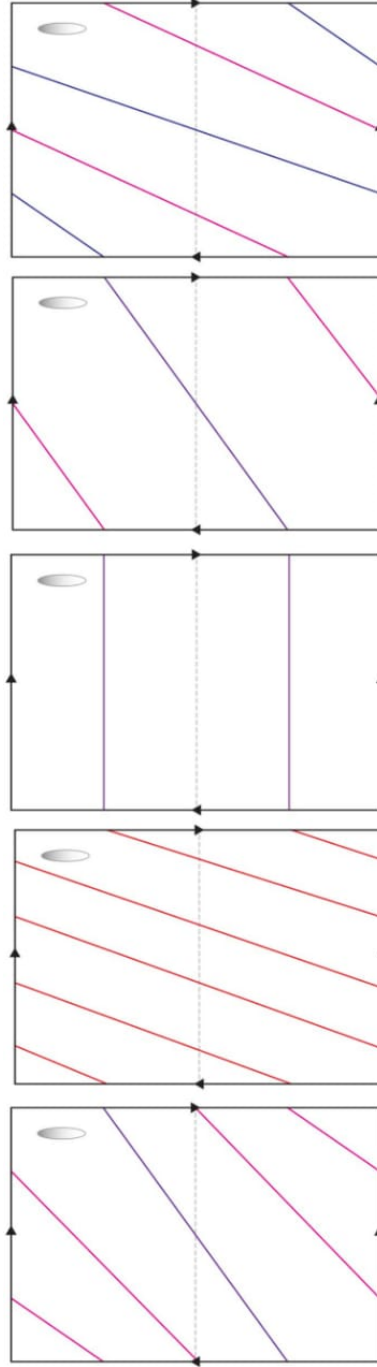


Figure 14: From top to down: $K(3, 2)$, $K(1, 2)$, $K(0, 2)$, $K(4, 2)$, $K(2, 3)$.

Each of these Jones polynomials is correctly computed. The code is attached and clearly commented for the reader.

3.2 Skein Relations

We also tried to directly use the skein relation for the Jones polynomial to calculate the Jones polynomial of a $K(m, n)$ Klein knot.

Theorem 3.4. The Jones polynomial for the $K(0, 2)$ and $K(1, 2)$ Klein knots are 1 and $-t^{1/2} - t^{-1/2}$, respectively.

For a $K(m, 2)$ Klein knot with $m \geq 2$, the Jones polynomial can be calculated by the recursion

$$V(K(m, 2)) = t^2 V(K(m-2, 2)) + (t\sqrt{t} - \sqrt{t}) V(K(m-1, 2)).$$

Proof. As a $K(m, n)$ Klein knot can be represented by the braid representation

$$K(m, n) = (\sigma_1 \dots \sigma_{n-1})^m (\sigma_{n-1}^{-1} \dots \sigma_1^{-1}) \dots \sigma_{n-1}^{-1},$$

a $K(m, 2)$ knot can be simplified into the form

$$K(m, 2) = \sigma_1^{m-1}.$$

The $K(0, 2)$ Klein knot has braid representation σ_1^{-1} , and this is the unknot, so it has Jones polynomial 1.

The $K(1, 2)$ Klein knot has braid representation 1, and this is the 2-unlink. This is known to be $-t^{1/2} - t^{-1/2}$.

To apply the skein relation for $m \geq 2$, let us consider the last σ_1 crossing in the braid representation of $K(m, 2)$. L_+ would be the same as the original braid σ_1^{m-1} , but L_- would turn the last σ_1 into σ_1^{-1} , so the L_- braid would actually be $\sigma_1^{m-3} = K(m-2, 2)$. Similarly, L_0 nullifies the last σ_1 , so the L_0 braid is actually $\sigma_1^{m-2} = K(m-1, 2)$.

The skein relation for the Jones polynomial can be rearranged to be

$$V(L_+) = t^2 V(L_-) + (t\sqrt{t} - \sqrt{t}) V(L_0).$$

We can now replace L_+ , L_- , and L_0 with their corresponding Klein knots to get

$$V(K(m, 2)) = t^2 V(K(m-2, 2)) + (t\sqrt{t} - \sqrt{t}) V(K(m-1, 2)).$$

□

4 Conclusion

Through this paper, we've extended certain properties to Klein links, and we discovered certain properties among Klein links such as their genera, polynomials, and relations between Klein links and torus links. Along the way, we've created an algorithm that can calculate the Bracket and Jones polynomials for a Klein link, and, if parameters are changed, can work for any knot.

4.1 Open Questions

As usual, we chanced upon certain questions to which we did not find answers, but left open-ended.

- Is it possible to extend the Skein relation technique to find the Jones polynomial of a $K(m, 2)$ knot used in Theorem 3.4 to find the Jones polynomial of a general $K(m, n)$ knot? The polynomials for $K(m, 2)$ were easily found by simply realizing that $K(m, 2)$ can be simplified to σ_1^{m-1} , while a simplification of $K(m, 3)$ into $(\sigma_1\sigma_2)^{m-2}\sigma_1$ seems many times more complex, and for larger n , the braid representation for $K(m, n)$ contains more and more terms.
- On the other hand, what is the closed form for Jones polynomials for $K(m, 2)$ links? Given its form as a "linear" recurrence relation (where functions themselves are treated as constants), it's likely that it has a closed form. We tried, however, to calculate the first few Jones polynomials for $K(m, 2)$ links for small m , and there appeared not to be an obvious pattern.
- What is the computational complexity of the algorithm provided? First, what is the complexity of the object and method to calculate trace, and what is the complexity of the function to calculate the polynomial itself? If the algorithm is rather inefficient, what can be done to improve its efficiency?
- It is hard to tell whether there are special cases in which the Jones polynomial is incorrect. By checking with an online catalog of Klein links [6] and knowing what the Jones polynomials are for simple links and knots, the Jones polynomials for small m, n are confirmed to be correct through this algorithm. Originally, we had an issue with $K(m, 0)$ links, but this was resolved by noticing that any $K(m, 0)$ Klein link is simply m unknots and thus its bracket polynomial could be calculated by τ^{m-1} .
- Do there exist more connections between Klein links and torus links? It is possible that there is yet another connection left uncovered between Klein and torus links.

5 Bibliography

- Fruend, David and Smith-Polderman, Sarah. *Klein Links and Braids*. Rose-Hulman Undergraduate Mathematics Journal, Volume 14, No. 1, Spring 2013.
(<https://www.rose-hulman.edu/mathjournal/archives/2013/vol14-n1/paper6/v14n1-6pd.pdf>)
- Abramsky, Samson. *Temperley-Leib Algebra: From Knot Theory to Logic and Computation via Quantum Mechanics*. Mathematics of Quantum Computing and Technology, p415-458, 2007.
(<http://www.cs.ox.ac.uk/samson.abramsky/tambook.pdf>)
- Kauffman, Louis. *State Models and the Jones Polynomial*. Topology, Volume 26, No. 3, p395-407, 1987.
(<http://www.maths.ed.ac.uk/aar/papers/kauffmanjones.pdf>)
- Jones, Vaughan. *The Jones Polynomial*.
(<https://math.berkeley.edu/~vfr/jones.pdf>)
- Bowen, Jennifer. *Klein Link Catalogue*.
(<http://discover.wooster.edu/jbowen/research/klein-links/digital-catalogue/>)
- Bowen, Jennifer et al. *Klein Link Multiplicity and Recursion*.
(<http://discover.wooster.edu/jbowen/files/2013/10/Klein-Link-Multiplicity-and-Recursion.pdf>)