Parameterized complexity of quantum knot invariants
Clément Maria

To cite this version:
Clément Maria. Parameterized complexity of quantum knot invariants. 2020. hal-02429767

HAL Id: hal-02429767
https://hal.archives-ouvertes.fr/hal-02429767
Preprint submitted on 6 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
PARAMETERIZED COMPLEXITY OF QUANTUM KNOT INVARIANTS

CLÉMENT MARIA

Abstract. We give a general fixed parameter tractable algorithm to compute quantum invariants of links presented by diagrams, whose complexity is singly exponential in the carving-width (or the tree-width) of the diagram.

In particular, we get a $O(N^2 \cdot \text{cw}\cdot \text{poly}(n))$ time algorithm to compute any Reshetikhin-Turaev invariant—derived from a simple Lie algebra $\mathfrak{g}$—of a link presented by a planar diagram with $n$ crossings and carving-width $\text{cw}$, and whose components are coloured with $\mathfrak{g}$-modules of dimension at most $N$. For example, this includes the $N^{th}$-coloured Jones polynomial and the $N^{th}$-coloured HOMFLYPT polynomial.

1. Introduction

In geometric topology, testing the topological equivalence of knots (up to isotopy) is a fundamental yet remarkably difficult algorithmic problem.

A main approach is to compare knots by properties depending on their topological types, called invariants. Starting with the introduction by Jones [15] in the 1980s of a new polynomial invariant of knots, we have witnessed the birth of a new domain of low dimensional topology called quantum topology. From the study of quantum groups [5, 14] in algebra, topologists have designed new families of topological invariants for knots, links, and 3-manifolds, such as the Reshetikhin-Turaev invariants [21]. In practice, these quantum invariants have shown outstanding discriminative properties for non-equivalent knots and links, e.g., in the composition of knot census databases [2], and are at the heart of deep mathematical conjectures in the field [7, 8, 16, 20].

Consequently, efficient algorithms to compute quantum invariants are of strong interest. However, even the simplest quantum invariants, such as the Jones polynomial [13], are #P-hard to compute. A successful approach towards practical implementations has been the introduction of parameterized complexity to low dimensional topology. Independently, computing the Jones polynomial [18] and the HOMFLYPT polynomial [3] have been shown to admit fixed parameter tractable algorithms in the tree-width of the input link diagrams. Note that similar techniques have been applied to quantum invariants of 3-manifolds, such as the Barrett-Westbury-Turaev-Viro invariants of triangulated 3-manifolds [4, 27]. These algorithms led to significant speed-ups in practice.

Contribution. In this article, we give an algorithm to compute quantum invariants derived from ribbon categories [21, 26], taking into account the carving-width of the input link diagram.

**Theorem 1.1.** Fix a strict ribbon category $\mathcal{C}$ of $\mathbb{Z}[q]$-modules, and free modules $V_1, \ldots, V_m \in \mathcal{C}$ of dimension bounded by $N$. The problem:

<table>
<thead>
<tr>
<th>Quantum invariant at $\mathcal{C}, V_1, \ldots, V_m$:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $m$-components link $L$, presented by a diagram $D(L)$,</td>
</tr>
<tr>
<td><strong>Output:</strong> quantum invariant $J^L_C(V_1, \ldots, V_m)$</td>
</tr>
</tbody>
</table>
can be solved in $O(\text{poly}(n)N^{3\sqrt{n}}) \in O(\text{poly}(n)N^{32cw})$ machine operations, with $O(N^{cw} + n)$ memory words, where $n$ and $cw$ are respectively the number of crossings and the carving-width of the diagram $D(L)$.

In particular, this implies that, up to some easily computable re-normalisation, computing any Reshetikhin-Turaev invariant derived from a simple Lie algebra $\mathfrak{g}$ is fixed parameter tractable (complexity class $\text{FPT}$) in the carving-width of the input link diagram. Cases of interests are, in particular, $\mathfrak{g} = \mathfrak{sl}(2, \mathbb{C})$ giving the $N$th-coloured Jones polynomials, and $\mathfrak{g} = \mathfrak{sl}(n, \mathbb{C})$ giving the $N$th-coloured HOMFLYPT polynomials. This algorithm is:

1. the first fixed parameter tractable algorithm, and—considering $cw = O(\sqrt{n})$—sub-exponential time algorithm, for quantum invariants of knots stated in such generality (previously known cases were the (uncoloured) Jones polynomial [18], and the (uncoloured) HOMFLYPT polynomial [3]),

2. an exponential improvement over Burton’s $2^{O(cw \log cw)}\text{poly}(n)$ time algorithm for the uncoloured HOMFLYPT polynomial [3], and generally a low exponent ($\frac{3}{2} cw$) singly exponential algorithm for quantum invariants$^1$.

In Section 2 we recall the definition of quantum invariants derived from ribbon categories, and notions of parameterized complexity. In Section 3 we introduce a high-level parameterized algorithm based on graphical calculus and a tree embedding, then detail in Section 4 the main operation of the algorithm. In Section 5 we develop the implementation of the algorithm in the case of a ribbon category of $\mathcal{R}$-modules, and analyse its arithmetic complexity in Section 6, in the case $\mathcal{R} = \mathbb{Z}[q]$. This last study concludes the proof of the main theorem, and implies more generally that, when the type of invariant is part of the input, computing a quantum invariant is in the complexity class $\text{XP}$.

2. Background

We introduce the necessary notions from knot theory, quantum topology, and parameterized complexity.

Tangles and diagrams. A tangle is a piecewise linear embedding of a collection of arcs and circles into $\mathbb{R}^2 \times [0, 1]$, such that the arcs’ endpoints, called bases, belong to the top or bottom boundaries $\mathbb{R}^2 \times \{0\}$ and $\mathbb{R}^2 \times \{1\}$. A tangle intersecting $i$ times $\mathbb{R}^2 \times \{0\}$ and $j$ times $\mathbb{R}^2 \times \{1\}$ is an $(i, j)$-tangle.

A link is a tangle whose connected components are all closed curves (a $(0, 0)$-tangle), and a knot is a 1-component link. An orientation on a tangle is an orientation of each tangle component. Two tangles are equivalent iff they differ by an ambient isotopy of $\mathbb{R}^2 \times [0, 1]$ maintaining the boundary fixed.

A tangle diagram is a projection of the tangle into the plane, induced by a projection of $\mathbb{R}^2 \times [0, 1]$ into $\mathbb{R} \times [0, 1]$, sending $\mathbb{R}^2 \times \{0\}$ and $\mathbb{R}^2 \times \{1\}$ to $\mathbb{R} \times \{0\}$ and $\mathbb{R} \times \{1\}$ respectively. In a tangle diagram, the only multiple points are crossings, at which one section of the tangle crosses under or over another one transversally. We consider diagrams with no base points (i.e., link diagrams) as living on the sphere $S^2$.

Component orientations are pictured with arrow heads, and a $k \in \mathbb{Z}$ framing is pictured by $k$ positive twists if $k > 0$, and $k$ negative twists is $k < 0$. See Figure 1.

We refer to [17] for more details on knot theory.

$^1$Note that previous algorithms [18] are expressed in terms of tree-width, which is proportional but not equal to the carving-width, in consequence exponents are not directly comparable.
Ribbon categories and quantum invariants. We refer to Turaev’s monograph [26] for the categorical formulation of quantum invariants. We only introduce the necessary notions.

Intuitively, a strict ribbon category is an abstraction of the category of modules over a commutative ring, with their usual tensor product. Some morphisms—called braiding, twists, evaluations and co-evaluations—are distinguished in order to establish a connection between topology (tangles and knots) and algebra (morphisms between objects), via graphical calculus.

More precisely, a strict ribbon category \( \mathcal{C} \) is a category with a unit object \( 1 \) and which is equipped, for any objects \( U, V, U', V' \) and morphisms \( f : V \to V', g : U \to U' \), with:

(a) an associative tensor product assigning to \( U \) and \( V \) an object \( U \otimes V \), and to \( f \) and \( g \) a morphism \( f \otimes g : U \otimes V \to U' \otimes V' \),

(b) a natural braiding isomorphism \( c_{U,V} : U \otimes V \to V \otimes U \) with inverse \( c_{U,V}^{-1} \),

(c) a duality associating to any \( V \) a dual object \( V^* \), together with co-evaluation morphisms \( b_V : 1 \to V \otimes V^* \) and evaluation morphisms \( d_V : V^* \otimes V \to 1 \),

(d) a natural twist isomorphism \( \theta_V : V \to V \) with inverse \( \theta_V^{-1} \),

(e) and where \( \text{Hom}_{\mathcal{C}}(1, 1) \) has the structure of a commutative ring \( \mathcal{R} \).

By convention, the “tensor product of zero objects” is equal to \( 1 \). In a strict ribbon category, these objects and morphisms satisfy additional compatibility constraints, that are necessary to state Theorem 2.1 below.

For example, the category of modules over a commutative ring \( \mathcal{R} \) with standard tensor product, and equipped with the trivial braiding \( u \otimes v \mapsto v \otimes u \), forms a strict ribbon category. In this case, the ring \( \mathcal{R} \), seen as a module over itself, is the unit object \( 1 \), and any morphism \( \mathcal{R} \to \mathcal{R} \) is a multiplication by a scalar \( \tau \in \mathcal{R} \). Hence \( \text{Hom}_{\mathcal{C}}(1, 1) \) is isomorphic to the commutative ring \( \mathcal{R} \) itself. For invariants derived from quantum groups, we mainly focus on the category of \( \mathcal{R} \)-modules, generally free of finite dimension but with more complex braidings than the trivial ones. The ring \( \mathcal{R} \) is \( \mathbb{Z}[q] \) (or \( \mathbb{Z}[q, q^{-1}] \)), the ring of one-variable polynomials with integer coefficients. Morphisms between free modules are represented by matrices with \( \mathcal{R} \)-coefficients.

Graphical calculus and coloured tangles. Fix a strict ribbon category \( \mathcal{C} \). A colouring of a link \( L \), with \( m \) ordered components \( L_1, \ldots, L_m \), is an assignment of an object \( V_i \in \mathcal{C}, 1 \leq i \leq m \), to every component \( L_i \) of \( L \).

A link diagram is considered in standard form if it can be decomposed into the following pieces, described in Figure 2: (i) vertical strands, (vi) & (vii) positive and negative crossings, (viii) & (ix) positive and negative right twists, and (x) & (xi) caps and cups. See Figure 3 for a Hopf link in standard position. Any link (or tangle) diagram can be moved into standard form.
Rules (i) to (xi) of Figure 2 gives the conversion from coloured tangle to \( \mathcal{C} \)-morphism, called \textit{Penrose functor}. Specifically, given a coloured link diagram \( D(L) \) in standard form, the Penrose functor turns the diagram into a morphism, following the rules:

(o): A morphism \( f: U \rightarrow V \) in \( \mathcal{C} \) is represented graphically by a box, aligned with \( x \)- and \( y \)-axis, called \textit{coupon}, with incoming vertical \( V \)-coloured strands (top) and outgoing vertical \( U \)-coloured strand (bottom),

(i): reversing a component orientation changes a colour \( V \) to its dual \( V^* \),

(ii): two parallel strands coloured \( U \) and \( V \) are equivalent to a single strand coloured \( U \otimes V \),

(iii): a vertical strand coloured \( V \) is equivalent to the identity morphism \( \text{id}_V \),

(iv): a morphism \( g \) above another one \( f \) is equivalent to there composition \( g \circ f \),

(v): two morphisms \( h_1 \) and \( h_2 \) side by side are equivalent to their tensor product \( h_1 \otimes h_2 \),

(vi) & (vii): a positive crossing is equivalent to a braiding morphism, a negative crossing is equivalent to the inverse of the braiding morphism,

(viii) & (ix): positive and negative twists are equivalent to the twist morphism and its inverse respectively,

(x) & (xi): caps and cups are equivalent to evaluation and co-evaluation respectively,

(xii): the dual morphism \( f^*: V^* \rightarrow U^* \) of a morphism \( f: U \rightarrow V \) is given by the graphical equation (xii) or, equivalently, by:

\[
  f^* = (d_V \otimes \text{id}_{U^*}) \circ (\text{id}_{V^*} \otimes f \otimes \text{id}_{U^*}) \circ (\text{id}_{V^*} \otimes b_U).
\]

The morphisms are applied to the objects colouring the entering and leaving strands. Figure 2 gives the morphism associated to the Hopf link coloured with objects \( U \) and \( V \).

Consequently, for a category \( \mathcal{C} \), the Penrose functor associates to any coloured link a morphism \( 1 \rightarrow 1 \). More generally, it associates to a coloured \((i,j)\)-tangle a morphism \( U_1 \otimes \ldots \otimes U_i \rightarrow V_1 \otimes \ldots \otimes V_j \), for the \( \mathcal{C} \)-objects \( U_k \)s and \( V_\ell \)s colouring the bottom and top bases respectively.

If the ordered components of a link \( L \) are coloured \( V_1, \ldots, V_m \), this morphism is written:

\[
  J^\mathcal{C}_L(V_1, \ldots, V_m) \in \text{Hom}_\mathcal{C}(1, 1).
\]

Strict ribbon category produce topological invariants, called \textit{quantum invariants}:

\textbf{Theorem 2.1} ([21, 26]). Let \( D(L) \) be a diagram of an \( m \)-components link \( L \) on \( S^2 \), and let \( \mathcal{C} \) be a strict ribbon category. Let \( V_1, \ldots, V_m \) be a colouring of the components of \( L \). The quantity \( J^\mathcal{C}_L(V_1, \ldots, V_m) \) produced by the Penrose functor is invariant by ambient isotopy of \( S^2 \) and Reidemeister moves on \( D(L) \). It is consequently a topological invariant of the coloured link \( L \).

When \( \mathcal{C} \) is the category of \( \mathcal{R} \)-modules, \( J^\mathcal{C}_L(V_1, \ldots, V_k) \in \text{Hom}(1, 1) \cong \mathcal{R} \) is identified to a scalar in \( \mathcal{R} \).

Graph parameters. The \textit{carving-width}, also known as \textit{congestion}, is a graph parameter introduced by Seymour and Thomas [25].

\textbf{Definition 2.2}. Let \( G = (V, E) \) be a graph on \( n \) vertices, with loops and multiple edges. Let \( T \) be an unrooted binary tree, with all internal nodes of degree 3, and with \( n \) leaves.
An embedding $\phi$ of $G$ into $T$ is a bijective mapping between the nodes of $G$ and the leaves of $T$. Every edge $e$ of $T$ induces a partition of the vertices of $G$ into two sets, $V = U_e \sqcup V_e$, inherited from the partition of $T \setminus e$ into two trees. Let $w(e)$ denote the number of edges in $G$ between $U_e$ and $V_e$, called the weight of $e$.

The congestion of an embedding $(T, \phi)$ is the maximal weight of a tree edge:

$$\text{cn}(T, \phi) = \max_{e \text{ edge of } T} w(e),$$

The carving-width $\text{cn}(G)$ of a graph $G$ is the minimal congestion over all its embeddings into binary trees. The carving-width $\text{cn}(D(L))$ of a link diagram $D(L)$ is the carving-width of the 4-valent planar graph it realises. The carving-width $\text{cn}(L)$ of a link $L$ is the minimal carving-width of any of its diagrams.

The carving-width of a graph is closely related to its tree-width [22], which plays a major role in combinatorial algorithms.

**Figure 2.** Graphical calculus induced by Penrose functor.
Theorem 2.3 (Theorem 1 of [1]). Let $G$ be a graph of maximal degree $\delta$. Then,
\[
\frac{2}{3}(\text{tw}(G) + 1) \leq \text{cng}(G) \leq \delta(\text{tw}(G) + 1).
\]
For tangle diagrams $\delta \leq 4$.

Carving-width has several advantages over tree-width, and has been successfully used in low dimensional topology [11, 12, 19, 24].

First, optimal tree embeddings of planar graphs can be realised topologically, as stated below. A bridge in a connected graph $G$ is an edge of $G$ whose removal splits $G$ into more than one connected component. A tree embedding $(T, \phi)$ of $G$ is bond if the two vertex sets $U_e$ and $V_e$ from the cut associated to an edge $e$ of $T$ induce connected sub-graphs in $G$.

Theorem 2.4 ([25, Theorem 5.1]). Let $G$ be a simple connected bridgeless graph with more than two vertices. If $G$ has carving-width $\text{cw}$ then there exists a bond tree embedding of $G$ of congestion $\text{cw}$.

Up to a subdivision of multiple edges, which does not increase carving-width, a link diagram can be made simple, as a graph. Being 4-valent, it is bridgeless, and, if connected, it consequently admits a bond tree embedding of minimal congestion. We interpret a bond tree embedding of a planar graph (on the sphere $S^2$) as a collection of disjoint Jordan curves $\lambda_e \subset S^2$, one for each edge $e$ of $T$, realising the cut $U_e \cup V_e$ [24].

For planar graphs, a bond tree embedding of minimal congestion can be computed in polynomial time [10, 25]. By the planar separator theorem, the carving-width of a planar graph with $n$-vertices is in $O(\sqrt{n})$.

3. Fixed parameter tractable algorithm via graphical calculus

Let $\mathcal{C}$ be a strict ribbon category, and let $L$ be an oriented link with $m$ components $L_1, \ldots, L_m$. Let $D(L)$ be an oriented link diagram of $L$, where each link component $L_i$ is coloured by an object $V_i$ from the category $\mathcal{C}$, such that the Penrose functor gives an isotopy invariant of $L$ associated to its colouring, as described in Theorem 2.1.

It follows from the definition of Penrose functor that the quantum invariant of a separable link $L \cup L'$ is the product of the invariants of $L$ and $L'$, such that they can be computed separately. W.l.o.g. we assume that the diagram $D(L)$ is connected as a graph, and has at least 2 crossings, not all twists.
Figure 4. The four tangles and associated morphisms at the tree leaves. From left to right: Equations (3.1), (3.2), (3.3), and (3.4). The marked bullet point is on the left of each diagram, and is selected such that only these four morphisms are encountered.

3.1. Tree embedding of link diagrams. Let \((T, \phi)\) be a bond tree embedding of the planar graph of \(D(L)\), and root it by subdividing an arbitrary tree edge, picking the centre as the root. All edges of \(T\) have now a parent and child endpoint. By convention, we add a “half-edge” on top of the tree, having the root as child. Every inner node in \(T\) has consequently degree 3, with two edges “going down”, and one edge “going up”.

Let \(e\) be an edge of \(T\) with child node \(x\), and \(X\) the set of crossings mapped to the leaves of the subtree \(T_x\) rooted at \(x\). According to Theorem 2.4, there exists a Jordan curve \(\lambda_e\) separating \(X\) from the rest of the diagram. The diagram being on the sphere, we draw the tangle “inside” the Jordan curve when we represent it on the plane.

To an edge \(e\) corresponds a \((0, w(e))\)-tangle \(T\), spanned by the crossings \(X\) and contained “inside” \(\lambda_e\). We mark an arbitrary but fixed “bullet” point on \(\lambda_e\) and order the bases of \(T\) counter-clockwise from that point. We get a \((0, w(e))\)-tangle by isotopically sliding all bases to the top boundary, such that the first base in the bullet ordering is rightmost on the top boundary. See Figure 4 for examples of \((0, w(e))\)-tangles at the tree leaves, and Figure 5 (Left) for bases ordered by bullet ordering.

In the process of the algorithm below, bullet orderings are assigned on the fly.

3.2. Tree traversal algorithm. Let \(D(L)\) be coloured by objects of the category \(\mathcal{C}\). To every edge \(e\) of weight \(w(e)\) in \(T\), the Penrose functor assigns a \(\mathcal{C}\)-morphism \(f_e: \mathbb{1} \to V_1 \otimes \ldots \otimes V_{w(e)}\) to the associated tangle, where \(V_1, \ldots, V_{w(e)}\) are the colours of the strands intersecting the Jordan curve \(\lambda_e\).

The morphism associated to the half-edge at the root is a \(\mathbb{1} \to \mathbb{1}\) morphism, because the corresponding Jordan curve does not intersect the link diagram. This morphisms gives the invariant \(J_L \in \mathcal{R}\) of Theorem 2.1. All edge morphisms are computed recursively following a depth first traversal of \(T\). We describe the base morphisms assigned to the edges whose child node is a leaf, and we describe an algorithm for inner edges in the next section.

3.3. Morphisms at the leaves. Up to reorientation of the strands, which algebraically consists of dualising colours, we can restrict to four base morphisms:

\[
\begin{align*}
(3.1) & \quad (\text{id}_{U^*} \otimes c_{V,U} \otimes \text{id}_{V^*}) \circ (b_{U^*} \otimes b_V) \\
(3.2) & \quad (\text{id}_{U^*} \otimes c_{V,U}^{-1} \otimes \text{id}_{V^*}) \circ (b_{U^*} \otimes b_V) \\
(3.3) & \quad (\text{id}_{V^*} \otimes \theta_V) \circ b_{V^*} \\
(3.4) & \quad (\text{id}_{V^*} \otimes \theta_V^{-1}) \circ b_{V^*}
\end{align*}
\]

They correspond graphically to the diagrams in Figure 4, where the bullet ordering is chosen to restrict to these four cases.

3.4. Merging morphisms at tree nodes. Every inner node \(x\) of \(T\) is the parent node of two edges \(e_1\) and \(e_2\), and the child of an edge \(e\). Given the morphisms \(f_{e_1}\) and \(f_{e_2}\) for edges \(e_1\) and \(e_2\) respectively, we construct the morphism \(f_e\) for edge \(e\).
Figure 5. Merging two sub-trees. Left: Planar embeddings of the diagram with Jordan curves $\lambda_{e_1}$, $\lambda_{e_2}$ (inner circles) and $\lambda_e$ (outer circle), depending on the position of the bullets for $\lambda_{e_1}$ and $\lambda_{e_2}$. The bold lines connecting the Jordan curves represent multiple parallel strands connecting the corresponding tangles. Right: Coupons for $f_{e_1}$, $f_{e_2}$ and $f_e$ (outer coupon) obtained after plane isotopy. The bullet for $\lambda_e$ is selected so as to restrict to these three cases.

First, note that the bullet ordering of the strands intersecting $\lambda_{e_1}$ and $\lambda_{e_2}$ leads to three configurations when representing morphisms $f_{e_1}$ and $f_{e_2}$ with coupons; see Figure 5 where thick lines represent sets of parallel tangle strands. By hypothesis, morphisms on tree edges have domain 1. The coupons for $f_{e_1}$, $f_{e_2}$, and $f_e$ (the outer coupon) are obtained by a plane isotopy forcing the strands to intersect coupons on their top side, and putting bullets on the coupons’ left sides. The bullet of the outer coupon $f_e$ is selected so as to restrict to the three configurations of Figure 5.

4. Factorisation of morphisms at tree nodes

Given the morphisms $f_{e_1}$ and $f_{e_2}$ in Figure 5, we describe graphically a factorisation scheme to obtain the morphism $f_e$.

4.1. Sliding and canonical form. The canonical form for morphisms to be merged is depicted in the top left corner of Figure 7. It consists of two side-by-side morphisms $g_1$ and $g_2$, bridged by parallel strands coloured $U_1, \ldots, U_k$. All other strands go vertically.

Given morphisms $f_{e_1}$ and $f_{e_2}$ in Figure 5, we obtain a canonical form by sliding strands, wrapping clockwise around the coupons, under the coupons. For example, in the top right case of Figure 5, we slide strand 1 under the $f_{e_1}$-coupon, and strands $a$ and $b$ under the $f_{e_2}$-coupons.
The details of the operation are depicted in Figure 6, where the V-strand wraps clockwise around the f-coupon, and f is a $1 \rightarrow U \otimes V$ morphism. Sliding the V-strand under the coupon by tangle isotopy produces a positive twist $\theta_V$ and a positive crossing $c_{V,U}$. Decomposing further in Figure 6, let $U = U_1 \otimes \ldots \otimes U_1$ be the tensor product of the colours of $i$ parallel strands, and $V = V_j \otimes \ldots \otimes V_1$ the tensor product of $j$ parallel strands wrapping clockwise around the f-coupon. As depicted in the figure, sliding the $j$ strands under $f$ induces

- a twist $\theta_{V_j}$ on each of the $V_\ell$-coloured strands, $1 \leq \ell \leq j$,
- a sequence of $j(j-1)$ positive and negative crossings of type $c_{V_j,V_1}$, followed by
- a sequence of $ij$ positive crossings of type $c_{V_j,U_k}$.

We obtain the morphisms $g_1, g_2$ of the canonical form (Figure 7) by factorising the morphisms $f_{e_1}$ and $f_{e_2}$ with these sequences of twists and crossings, after the sliding operation.

4.2. Factorisation of the canonical form. Figure 7 pictures two factorisation schemes for side-by-side morphisms $g_1$ and $g_2$ in canonical form, bridged by $k$ parallel strands coloured $U_1, \ldots, U_k$. Denote by $\text{cw}$ the carving-width of the link diagram, and assume the tree embedding $(T, \phi)$ has width $\text{cw}$. We distinguish between two cases:

Small bridge. For $k$ smaller than half the carving-width (Figure 7, Left), we consider first the morphism $d_{U_1 \otimes \ldots \otimes U_k}$ induced by the composition of the evaluation morphisms $d_{U_\ell}$, $\ell = k \ldots 1$. More precisely, the morphism $d_{U_1 \otimes \ldots \otimes U_k} : U_1 \otimes \ldots \otimes U_k \otimes U_k^* \otimes \ldots \otimes U_1^* \rightarrow 1$, is obtained by composing the evaluation morphisms from bottom up:

$$
\begin{align*}
d_{U_1 \otimes \ldots \otimes U_k} : & \quad U_1 \otimes \ldots \otimes U_k \otimes U_k^* \otimes \ldots \otimes U_1^* \rightarrow 1, \\
& = \prod_{\ell = k}^1 \left( \text{id}_{U_1 \otimes \ldots \otimes U_{\ell-1}} \otimes d_{U_\ell} \otimes \text{id}_{U_{\ell-1} \otimes \ldots \otimes U_1^*} \right)
\end{align*}
$$

where $\ell = k$ is the rightmost term of the composition.

The (partial) composition of $d_{U_1 \otimes \ldots \otimes U_k}$ with $g_2$ through $U_k^* \otimes \ldots \otimes U_1^*$ gives the morphism $h$:

$$
\begin{align*}
h : & \quad U_1 \otimes \ldots \otimes U_k \rightarrow W_1 \otimes \ldots \otimes W_j, \\
& = (d_{U_1 \otimes \ldots \otimes U_k} \otimes \text{id}_{W_1 \otimes \ldots \otimes W_j}) \circ (\text{id}_{U_1 \otimes \ldots \otimes U_k} \otimes g_2).
\end{align*}
$$
Finally, the morphism $f_e$ obtained from the merging of $f_{e_1}$ and $f_{e_2}$ is given by the (partial) composition of $g_1$ and $h$, through $U_1 \otimes \ldots \otimes U_k$. Precisely,

$$f_e : 1 \to V_1 \otimes \ldots \otimes V_i \otimes W_1 \otimes \ldots \otimes W_j,$$

$$= (\text{id}_{V_1 \otimes \ldots \otimes V_i} \otimes h) \circ g_1. \tag{4.3}$$

By construction, these operations give the morphism $f_e$ induced by the Penrose functor on the coloured tangle associated to the subtree of $T$ rooted at the child node of edge $e$.

Large bridge. The case $k$ strictly larger than half the carving-width starts by flipping upside-down coupon $g_2$. Precisely, this operation is depicted in Figure 8. Starting with a morphism
Figure 8. Planar isotopy, then factorisation with $g^*$, the dual morphism to $g$.

...
Figure 9. Graphical representation of the seven elementary compositions of morphisms.

**Lemma 5.1.** Consider the elementary morphism compositions in Figure 9 (1), (2), (3), and (4). Let \( U, V, V', W \) be finite dimensional free \( \mathcal{R} \)-modules, with \( \dim U = a \), \( \dim V = b \), \( \dim V' = b' \), and \( \dim W = c \). Then, given the matrices for morphisms \( f, \theta^\pm_{V'}, c^\pm_{V, V'}, b_V \), and \( d_U \), we can compute the matrix for morphism \( h \) in:

- \( O(ab^2c) \) arithmetic operations in \( \mathcal{R} \) for (1),
- \( O(ab^2c) \) arithmetic operations for (2) and (3), and
- \( O(a^2b) \) arithmetic operations for (4).

The memory complexity of the operation does not exceed the size of the output, which is a row or column vector \( h \) containing scalars from \( \mathcal{R} \).

**Proof.** Figure 9 (1), (2), and (3). All three cases consist of the matrix-vector product

\[
h = (\text{id}_U \otimes M \otimes \text{id}_W) \cdot f,
\]

where \( M \) is respectively the \((bb' \times bb')\)-matrix \( c^\pm_{V, V'} \), the \((b \times b)\)-matrix \( \theta^\pm_{V'} \), and the \((b^2 \times 1)\)-matrix \( b_V \).

Consider \( M \) to be an \((m \times m')\)-matrix, with coefficients \((M_{i,j})_{1 \leq i \leq m, 1 \leq j \leq m'} \). Matrix \((\text{id}_U \otimes M \otimes \text{id}_W)\) has at most \( m' \) non-zero coefficients per row. We get the formula for the \( i \)th entry of \( h \):

\[
h_{i,1} = \sum_{k=1}^{m'} M_{\beta+1,k} \cdot f_{\alpha cm' + \gamma + (k-1)c,1},
\]

where \( i \) is uniquely written as \( i = \alpha \cdot cm + \beta \cdot c + \gamma \), with \( 0 \leq \alpha \leq a - 1 \), \( 0 \leq \beta \leq m - 1 \), and \( 1 \leq \gamma \leq c \). Computing \( h \) requires \( O(m'|f|) \) arithmetic operation in \( \mathcal{R} \), where \( |f| \) is the length of vector \( f \), storing \( O(|f|) \) scalars from \( \mathcal{R} \).

**Figure 9 (4).** With a similar approach, we get for any \( j, 1 \leq j \leq a^2b \):

\[
h_{1,j} = (d_U)_{1,\alpha a + \gamma} \cdot f_{\beta + 1,1},
\]

where \( j \) is uniquely written as \( j = \alpha \cdot ab + \beta \cdot a + \gamma \), with \( 0 \leq \alpha \leq a - 1 \), \( 0 \leq \beta \leq b - 1 \), and \( 1 \leq \gamma \leq a \). The algorithm has complexity \( O(a^2b) \) and memory usage \( O(a^2b) \). \( \square \)

**Lemma 5.2.** Consider the elementary morphism compositions in Figure 9 (5), (6), and (7). Let \( U, V, W \) be finite dimensional free \( \mathcal{R} \)-modules, with \( \dim U = a \), \( \dim V = b \), and \( \dim W = c \). Then, given the matrices for morphisms \( f \) and \( g \), we can compute the matrix for morphism \( h \) in \( O(abc) \) arithmetic operations in \( \mathcal{R} \), and memory complexity \( O(ab + bc + ac) \) times the size of a scalar in \( \mathcal{R} \).
Proof. Figure 9 (5). Morphism $f$ is a $bc \times 1$ matrix, and morphism $g$ is a $1 \times ab$ matrix. Define $h = (g \otimes \text{id}_W)(\text{id}_U \otimes f)$. Morphism $h$ is a $c \times a$ matrix.

Studying the shape of matrices $(\text{id}_U \otimes f)$ and $(g \otimes \text{id}_W)$, it appears that every one of the $c \times a$ coefficients of the product $h = (g \otimes \text{id}_W)(\text{id}_U \otimes f)$ is a sum of $O(b)$ terms. Precisely, an explicit computation gives us, for any $i, j, 1 \leq i \leq c, 1 \leq j \leq a$:

$$h_{i,j} = \sum_{k=0...b-1} g_{1,(j-1)b+k+1} \cdot f_{kc+i,1}.$$  

Morphism $h$ is a $c \times a$ matrix, and each of its coefficients can be computed in $O(b)$ arithmetic operations in $\mathcal{R}$, leading to the $O(abc)$ time complexity. The memory consumption is the sum of the sizes of the input matrices $f$ and $g$, and the output matrix $h$.

Figure 9 (6). With a similar approach, for any $i, j, 1 \leq i \leq a, 1 \leq j \leq c$:

$$h_{i,j} = \sum_{k=0...b-1} g_{1,kc+j} \cdot f_{(i-1)b+k+1,1}.$$  

Figure 9 (7). With a similar approach, for any $i, 1 \leq i \leq ac$:

$$h_{i,1} = \sum_{k=1...b} g_{\beta,k} f_{ab+k,1},$$  

where we write $i = \alpha c + \beta$, for $0 \leq \alpha \leq a - 1$ and $1 \leq \beta \leq c$. \qed

5.2. Implementation of the algorithm. We implement the algorithm described in Sections 3 and 4 using the elementary composition of Figure 9. Define $N$ a bound on the dimension of the different modules $U_i, V_j, W_k$ colouring the components of the link.

Leaf morphisms. The leaf morphisms described in Equations (3.1-3.1) and Figure 4 are implemented using elementary compositions (1) and (2). By Lemma 5.1, the complexity is at most $O(N^6)$ arithmetic operations in $\mathcal{R}$.

Sliding under a coupon. The sliding operation as presented in Figure 6 composes a morphism $f$ with a sequence of twist and braiding morphisms. Precisely, let $h$ denote the entire morphism in Figure 6. Starting from the $(O(N^{i+j}) \times 1)$ matrix $f$, it is computed iteratively applying $j$ times elementary composition (2) for the twists, then $j(j-1)$ times elementary composition (1) for the braidings between $V_i$- and $V_j$-strands, and finally $ij$ times elementary composition (1) for the braidings between $V_i$- and $U_j$-strands.

During the computation, we maintain a vector of size $(1 \times O(N^{i+j}))$. Applying Lemma 5.1, the sliding operation runs in $O(j(i+j)N^{i+j+2})$ arithmetic operations in $\mathcal{R}$, storing $O(N^{i+j})$ scalar from $\mathcal{R}$. In the algorithm, $i+j \leq cw$, the carving-width of the link diagram. Consequently, we get $O(cw^2 N^{cw+2})$ operations, with memory $O(N^{cw})$.

Construction of evaluations and co-evaluations. The morphism $d_{U_1 \otimes \ldots \otimes U_k}$ appearing in Figure 7 is the result of $k$ elementary compositions of type (4). The morphisms maintained during the computation are of size $(1 \times O(N^{2k}))$. Applying Lemma 5.1, the computation takes a total of $O(kN^{2k})$ arithmetic operations in $\mathcal{R}$, storing $O(N^{2k})$ scalars from $\mathcal{R}$. The case $b_{W_1 \otimes \ldots \otimes W_l}$ is similar.

In the algorithm, $k$ (or $j$) is smaller than $cw/2$. Consequently, the complexity is $O(cw N^{cw})$ arithmetic operations, storing $O(N^{cw})$ scalars.
Composition of morphisms. Finally, the compositions of morphisms described in Figure 7 are implemented with a constant number of elementary compositions (5), (6), and (7). Considering Lemma 5.2, the product $abc$ of dimensions never exceed $N^2 cw$. Consequently, the compositions of Figure 7 are implemented using $O(N^2 cw)$ arithmetic operations in $R$, storing $O(N cw)$ scalars from $R$.

Overall complexity. In conclusion, we sum up the different steps of the algorithm and its implementation. Let $D$ be a coloured link diagram with $n$ crossings and carving width $cw$, where the dimension of each colouring module is at most $N$. The algorithm first computes an optimal tree embedding in $O(poly(n))$ operations. The tree has size $n$ and width $cw$. W.l.o.g., we assume the diagram has at least one crossing that is not a twist, and consequently $cw ≥ 4$, the maximal degree of the graph. Considering $cw ∈ O(\sqrt{n})$ and $cw + 2 ≤ \frac{3}{2} cw$, the quantum invariant associated to the colouring is computed in:

$$O(n^2 N^2 cw)$$ arithmetic operations in $R$,

storing: $O(n)$ words for the diagram, plus $O(N cw)$ scalars from $R$.

6. Arithmetic complexity and quantum invariants of links

Working with matrices with $R$-coefficients, for a ring $R$, allows the algorithm to be applied in great generality. For example, any complex simple Lie algebra $g$ produces quantum invariants of links, that can be expressed as a composition of morphisms between free $R$-modules, and to which our algorithm can be applied. See [26, Chapter 6] for an explicit construction.

In this case, $R$ is a polynomial ring, and degrees of polynomials as well as values of coefficients may blow-up during intermediate computation. Specifically, both arithmetic operations within $R$ and bit size of $R$-elements may become exponential in $n$.

In this section we describe a solution to control the arithmetic complexity in the case $R = \mathbb{Z}[q]$, which is sufficient for all $J^q_L$ invariants. We also provide detailed complexity bounds for completeness.

6.1. Arithmetic complexity of polynomial invariants. We give coarse, but general, bounds on the degrees and coefficients of a polynomial invariant produced by the algorithm introduced above, that are sufficient for the complexity analysis.

**Proposition 6.1.** Let $C$ be a strict ribbon category of $\mathbb{Z}[q]$-modules, and let $D(L)$ be an $n$-crossings diagram of a link $L$ whose components are coloured with free modules $V_1, \ldots, V_m ∈ C$, of dimension at most $N$.

Let $d_0$ and $C_0$ be respectively a bound on the degree and a bound on the absolute value of coefficients of all polynomials in the matrices $c^\pm_{V_i,V_j}$, $\theta^\pm_{V_i}$, $d_{U_i}$, $b_{U_i}$, for $1 ≤ i, j ≤ m$.

Then the polynomial invariant $J^q_L(V_1, \ldots, V_m) ∈ \mathbb{Z}[q]$ has degree and absolute value of coefficients bounded by $d_n$ and $C_n$ respectively, with:

$$d_n = O(nd_0) \quad \text{and} \quad C_n = 2^{O(n\sqrt{n}\log N + n\log C_0)}.$$  

**Proof.** Consider a tree embedding of graph $D(L)$ where the tree is a path, with leaves attached to it. The minimal congestion over all such embeddings is called the cut-width of the graph, and is $O(\sqrt{n})$ due to the planar separator theorem.

Let $k$ be the cut-width of $D(L)$, and $(P, \phi)$ a minimal embedding of $D(L)$ into a path-tree. Running the algorithm of Sections 3-5 on this path decomposition boils down to computing the product of $O(n)$ matrices $M_{\alpha,n} \cdots M_1$, where all matrices are tensor products of a
\[c_{i,j}^{\pm}, \theta_{i}^{\pm}, d_{U_i}, b_{U_i}, \text{ for some } 1 \leq i, j \leq m, \text{ with identities, and all matrices have size at most } N^{O(k)} \times N^{O(k)}. \text{ Additionally, } M_1 \text{ has 1 column, and } M_{n,n} \text{ has 1 row, to give a scalar in } \mathbb{Z}[q].\]

Tensor with the identity does not change the bounds \(d_0\) and \(C_0\) on degrees and coefficients. Multiplying by a matrix adds at most \(d_0\) to the degree, and multiplies by at most \(N^{O(k)}C_0\) the largest coefficient. We get the global bounds by multiplying the matrices together, and substituting \(O(\sqrt{n})\) for \(k\).

\[\square\]

We give a general algorithm to compute a one-variable, integer coefficient, polynomial invariants, using standard computer algebra techniques and the algorithm of Sections 3-5.

**Proposition 6.2.** Let \(\mathcal{C}\) be a strict ribbon category of \(\mathbb{Z}[q]\)-modules for the one-variable polynomial ring \(\mathbb{Z}[q]\). Let \(L\) be an \(m\)-components link with colours the free modules \(V_1, \ldots, V_m\), and let \(J^c_L(V_1, \ldots, V_m) \in \mathbb{Z}[q]\) be the associated topological invariant. Assume \(L\) is presented by an \(n\)-crossings diagram \(D(L)\) with carving-width \(cw\).

Assume that the dimensions of the free modules \(V_1, \ldots, V_m\) are at most \(N\), and that the polynomial \(J^c_L(V_1, \ldots, V_m)\) has degree bounded by \(d_n\) and largest coefficient in absolute value bounded by \(C_n\). Then \(J^c_L(V_1, \ldots, V_m)\) can be computed in:

\[
O \left( d_n (d_n + \log C_n) \cdot \mathrm{Ar} (\log(d_n \log d_n + \log C_n)) \times n N^{\frac{3}{2}cw} + d_n^2 (d_n \log d_n + \log C_n)^2 + d_n^2 \mathrm{Ar}(d_n \log d_n + \log C_n) \right)
\]

machine operations, using:

\[
O \left( \log (d_n \log d_n + \log C_n) N^{cw} + nd_n(d_n \log d_n + \log C_n) + d_n^2 \mathrm{Ar}(d_n \log d_n + \log C_n) \right)
\]

bits. Here, \(\mathrm{Ar}(l) \in \tilde{O}(l)\) is the arithmetic complexity of operations \(+, - , \times, \div\) on integers encoded on at most \(l\) bits, which is linear in \(l\) up to a poly-logarithmic factor.

**Proof.** The algorithm relies on evaluation and interpolation. For short, denote \(J^c_L(V_1, \ldots, V_m)\) by \(P(q) \in \mathbb{Z}[q]\).

**Evaluation.** We evaluate \(P(q)\) on integer points \(q \in \{0, 1, \ldots, d_n\}\). Fix \(q_0\) in this set, and substitute \(q_0\) for \(q\) in matrices \(c_{i,j}^{\pm}, \theta_{i}^{\pm}, d_{U_i}, b_{U_i}\). The algorithm of Sections 3-5 is consequently a succession of matrix multiplications, where all matrices have integer coefficients, and the resulting \(P(q_0)\) is an integer of absolute value less than:

\[
C_0^{d_n+1} \leq 2^{(d_n+1) \log_2 d_n + \log_2 C_n} = 2^{O(d_n \log d_n + \log C_n)}
\]

For a fixed \(q_0\), we perform computation modulo the first \(r\) prime numbers \(2 = p_1, \ldots, p_r\) successively, such that the product \(p_1 \cdots p_r\) is larger than \(|P(q_0)|\). We then reconstruct \(P(q_0)\) using the Chinese Remainder Theorem. The product \(p_1 \cdots p_r\) is of order \(2^{r \log r}\) [23]. We take an appropriate \(r\) such that \(r \log r \in \Theta(d_n \log d_n + \log C_n)\), which gives \(r \in O(d_n + \log C_n)\).

Reconstructing the value \(P(q_0)\) from all the \((P(q_0) \mod p_i)\), \(1 \leq i \leq r\), can be computed in \(O(r^2 \log^2 r) = O((d_n \log d_n + \log C_n)^2)\) machine operations [9, Theorem 5.8].

Additionally, the values of all primes \(p_i\), \(i \leq r\), are in \(O(r \log (r \log r)) = O(r \log r) = O(d_n \log d_n + \log C_n)\) [23].

Denote by \(\mathrm{Ar}(l)\) the computational complexity of performing arithmetic operations \(+, -, \times\) on integers encoded on at most \(l\) bits, in \(\mathbb{Z}/w\mathbb{Z}\), for an integer \(w \leq 2^l\). The best known estimate for \(C(l)\) is:

\[
C(l) = O(l \log^2 (l) 2^{O(\log^+ l)}) = \tilde{O}(l),
\]
where $\log^*$ denotes the iterated logarithm, and the $\tilde{O}$-notation hides poly-log factors. This describes the complexity of performing the extended Euclidean algorithm [9] using Fürer’s method [6].

Interpolation. We reconstruct polynomial $P(q) \in \mathbb{Z}[q]$ of degree bounded by $d_n$ using Lagrange interpolation. Lagrange interpolation gives directly a formula for $P(q)$, computable in $O(d_n^2 \log(d_n) \log^2 d_n)$ machine operations [9, Theorem 5.1].

Summing up the complexity of evaluating polynomial $P(q)$ on the first $d_n + 1$ non-negative integers using the modulo reconstruction approach and running the algorithm of Sections 3-5, and the complexity of evaluating the interpolation formula, gives the complexity of the proposition.

We conclude by proving the main Theorem:

Proof. [of main Theorem 1.1] Fixing the category $C$ and the colours $V_1, \ldots, V_m$, of dimension at most $N$, makes $N$ constant, as well as the quantities $d_0$ and $C_0$ bounding degrees and coefficients of polynomials in the matrix for braidings, twists, and (co)evaluations. It enforces $d_n = O(n)$ (the bound on degree of the output polynomial), and $C_n = 2^{O(n \sqrt{n})}$ (the bound on absolute value of coefficients of the output invariant) in the complexity analysis. Substituting values gives the result of Theorem 1.1.

Remark 6.3. Note that quantum invariants are usually defined in the category of $\mathbb{Z}[q, q^{-1}]$-modules. Multiplying the braiding, twist, and (co)evaluation matrices by $q^a$ for $a$ large enough, and re-normalising the output, allows us to restrict the algorithm to the case of $\mathbb{Z}[q]$-modules.

Note that we get the following parameterized complexity result for the more general problem of quantum invariant computation, where the invariant is part of the input:

Theorem 6.4. The problem:

**General quantum invariant problem:**

**Input:** $C, V_1, \ldots, V_m$, presented by braiding, twist, evaluation and co-evaluation matrices, and $m$-components link $L$, presented by a diagram $D(L)$,

**Output:** quantum invariant $J_C^L(V_1, \ldots, V_m)$

can be solved in $O(\text{poly}(n, d_0, \log C_0) N^{2 \text{cw}})$ machine operations, where $n$ and $\text{cw}$ are respectively the number of crossings and the carving-width of the diagram $D(L)$, and $d_0$ and $C_0$ are respectively the maximal degree and maximal absolute value of coefficients of any polynomial in the input matrices.

In other words, when the polynomials in the matrices are encoded with their lists of coefficients, the input size is $\Omega(\text{poly}(N, d_0, \log C_0) + n)$, and the general quantum invariant problem is in the parameterized complexity class XP.

References


INRIA Sophia Antipolis-Méditerranée, FRANCE

E-mail address: clement.maria@inria.fr