From the Jones Polynomial to Topological Quantum Computation

Jeffrey Epstein

December 1, 2016

Abstract

The Jones polynomial is a polynomial invariant assigned to links. Here I review an algorithm from [1] for efficiently approximating the value of the Jones polynomials of links derived from elements of the braid groups B_n at primitive roots of unity. By solving a **BQP**-hard problem efficiently, this algorithm demonstrates the computational power of certain representations of the braid groups. This provides the starting point for a brief discussion of anyons and topological quantum computation.

Mathematical Preliminaries

In this section, several mathematical structures will be introduced. On the topological side of things, we'll start with links and the Jones polynomial. Then we'll change course and define the braid group and the Temperley-Lieb algebra, and representations of these in the algebra of bounded operators on a Hilbert space.

Topological Things - Links and the Jones Polynomial

An oriented link is an embedding in $\mathbb{R}^3$ of the disjoint union of a finite number of oriented copies of S^1. Two oriented links L and L' are equivalent, $L \sim L'$, if there is a homotopy between them that is an embedding for all values of the intermediate parameter (an isotopy). Informally, a link is a bunch of loops of string with selected orientations that might be linked together and knotted in non-trivial ways, and two links are equivalent if one can be deformed into the other without cutting any strings.

To each loop L is associated the function $V_L(t)$, a Laurent polynomial in $t^{1/2}$, specified by the following axioms [12][11]:

1. If $L \sim L'$, then $V_L = V_{L'}$.
2. If U is the unknot (a single loop with no crossings), then $V_U \equiv 1$.
3. Let links L_+, L_-, and L_0 have the same structure apart from some region where they have the features pictured below. Then $t^{-1}V_{L_+} - tV_{L_-} = (t^{1/2} - t^{-1/2})V_{L_0}$. This is known as the skein relation.

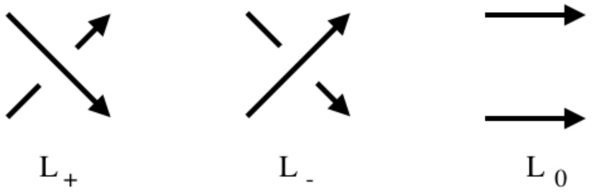

Figure 1:

The first of these axioms says that the Jones polynomial is insensitive to isotopy, and only depends on what we informally think of as "topological" features of the link, i.e. those that cannot be changed without cutting any strings. The second fixes a particular normalization. The real content is in the third axiom, which gives an iterative prescription for computing the Jones polynomial of any link by progressively simplifying the link to relate it finally to the unknot polynomial [11]. This is sufficient to confirm that this axiomatic definiton fully specifies the Jones polynomial. Unfortunately, it is not the case that the Jones polynomial fully characterizes a link - there are known to be pairs of inequivalent links L, L' with $V_L = V_{L'}$.

While the third axiom in principle provides a straightforward procedure for calculating the Jones polynomial given a planar diagram for a link, this method in general requires time at least exponential in the number of crossings, since each application of the skein relation doubles the number of diagrams to keep track of. Therefore, it does not provide a practical algorithm for actually computing the Jones polynomial.

Algebraic Things - Hilbert Space Representations of Braid Groups

All of these objects are defined in [2] (and in many other places). I follow their presentation. For the sake of brevity, I'll refrain from reproducing the many helpful diagrams that are also given in that reference and elsewhere.

Braid Group B_n

The braid group on n strands is simple to describe visually. Its elements are the equivalence classes of ways that n threads may be attached to fixed points on the bottom edge of a square, passed over and under each other, and attached to fixed points at the top edge, where the equivalence relation is that two elements are the same if the strings of one can be moved around, without cutting or untying the fixed ends, to arrive at the configuration of the other. Multiplication of two elements consists of tying the top ends of the strings of one to the botton ends of the corresponding strings of the other. From this picture, it follows naturally that the identity is the element in which there are no string crossings. Formally, this intuition may be captured by defining the group B_n to be the group generated by elements $\{1, \sigma_1, \sigma_{n-1}\}$ with the relations

1. $\sigma_i \sigma_j = \sigma_j \sigma_i$ for $|i - j| \geq 2$
2. $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$

Associating the generator σ_i to the picture in which the i^{th} string crosses under the $i + 1^{\text{th}}$ string, these relations can be verified pictorially. The inverse σ_i^{-1} is the picture in which the i^{th} string crosses over the $i + 1^{\text{th}}$ string.

Temperley-Lieb Algebra $TL_n(d)$

This is another object with a pleasing pictorial interpretation. The Kauffman n-diagrams are squares with n marked points on the top and n marked points on the bottom, and n non-intersecting curves starting and ending at distinct marked points. As for the braid group, two such diagrams are equivalent if the curves of one may be continuously deformed, keeping the endpoints fixed and allowing at no point intersections, into the curves of another. A formal vector space over the complex numbers can be defined and equipped with a multiplication that may be realized as a two-step process. The first step is to fuse the top ends of the curves of one element with the bottom ends of the other, and the second is to delete any closed loops that are formed and, for each one, introduce a factor of d. (The zeroth step is to treat the multiplication as distributive so that this process is performed on pairs of Kauffman diagrams.) Again, there is a formal definition of $TL_n(d)$ as the algebra generated by elements $\{1, E_1, \ldots, E_{n-1}\}$ with the relations

1. $E_i E_j = E_j E_i$ for $|i - j| \geq 2$
2. $E_i E_{i\pm1} E_i = E_i$
3. $E_i^2 = dE_i$

where d is a complex number.

The Temperley-Lieb algebras support a trace $\text{Tr}_n : TL_n(d) \to \mathbb{C}$ known as the Markov trace. On the Kauffman n-diagrams, this trace is defined by connecting the n^{th} node on the top of the diagram to the n^{th} on the bottom, the $n-1^{\text{th}}$ to the $n-1^{\text{th}}$, and so on. The resulting diagram will consist of some number a of closed loops, and the trace of the diagram is d^{a-n}. The trace can then be extended linearly to all of $TL_n(d)$. This function satisfies $\text{Tr}_n(\mathbb{1}_n) = 1$ (a normalization) and if $X \in TL_{n-1}(d)$, then $\text{Tr}_n(XE_{n-1}) = d^{-1}\text{Tr}_{n-1}(X)$

(both properties are easy to verify pictorially). This is known as the Markov property. Together, these two conditions uniquely specify Tr_n [2] on $TL_n(d)$ or any of its representations.

Algebra $\mathcal{B}(\mathcal{V}_{n,k})$ of Bounded Operators on the Path Vector Space

Let G_k be the linear graph with $k-1$ vertices. A formal finite-dimensional vector space over $\mathbb{C}$ can be defined with paths of length n starting at the left-most vertex of G_k as generators. Denote this vector space $\mathcal{V}_{n,k}$. Then the bounded operators on it form an algebra $\mathcal{B}(\mathcal{V}_{n,k})$.

Algebra $\mathcal{B}(\mathcal{H}_n)$ of Bounded Operators on the n-Qubit Hilbert Space

This is the standard arena of bounded operators on $\mathbb{C}^{2^n}$ in which quantum computation takes place!

A Unitary Representation of the Braid Group

Here I'll reproduce very briefly a sequence of mappings passing through the algebraic objects defined above. These are all discussed in more detail in [2], which also contains illuminating diagrams. The goal is to obtain maps:

$$B_n \longrightarrow TL_n(d) \longrightarrow \mathcal{B}(\mathcal{V}_{n,k}) \longrightarrow \mathcal{B}(\mathcal{H}_n). \tag{1}$$

For $A \in \mathbb{C}$ such that $d = -A^2 - A^{-2}$, the map from generators of B_n to generators of $TL_n(d)$ defined by

$$\rho_A : \sigma_i \mapsto AE_i + A^{-1}\mathbb{1} \tag{2}$$

may be extended to a homomorphism $\rho_A : B_n \to TL_n(d)$. Note that it must be verified that the map preserves the relations among the σ_i. In other words, ρ_A gives a representation of the group B_n inside the algebra $TL_n(d)$. It was in fact in terms of the Markov trace, defined above, of elements of this representation defined above that Jones originally defined the Jones polynomial, in the context of thinking about representations of the braid group.

In order to define a representation (algebra homomorphism) $\tau : TL_n(d) \to \mathcal{B}(\mathcal{V}_{n,k})$, it suffices to consider the action of τ on the generators E_i of $TL_n(d)$. If E_i is such a generator, then $\tau(E_i) \in \mathcal{B}(\mathcal{V}_{n,k})$ may be specified by its matrix elements $\tau(E_i)_{qq'}$, where q, q' are paths of length n on the linear graph G_k, starting at the left-most vertex.

For some positive integer k, let $d = 2\cos(\pi/k)$, and $\lambda_\ell = \sin(\pi\ell/k)$ for $\ell \in \{1, \ldots, k-1\}$, and define

$$g_\ell = \sqrt{\frac{\lambda_\ell}{\lambda_{\ell-1}}} \qquad h_\ell = \sqrt{\frac{\lambda_\ell}{\lambda_{\ell+1}}}. \tag{3}$$

Now notice that the diagram for E_i divides the top and bottom of the square into $n+1$ regions. Label the regions on the bottom from left to right with the vertices $q_0, q_1, \ldots, q_{n+1}$ of the path q, and those on top from left to right with the vertices of q'. If any of the two-dimensional regions of the square bounded by the curves of E_i has two boundaries with different vertex labels, then $\tau(E_i)_{qq'} = 0$. Otherwise, notice that the diagram has a local maximum and a local minimum of the curves. To each extremum, associate g_ℓ if the convex region bordering the curve is labeled by vertex $\ell + 1$ and the concave region labeled by ℓ, and associate h_ℓ if it's the other way around. Then $\tau(E_i)_{qq'}$ is the product of these two numbers. It may be verified that the $\tau(E_i)$ are Hermitian, and the representation is unitary.

An easy map of $\mathcal{V}_{n,k}$ into $\mathcal{H}_n$ is as follows. For q a path of length n, define $\phi(q)$ to be the computational basis state of n qubits labeled by the bitstring of length n indicating the directions that the path takes, with 0 indicating a step left and 1 indicating a step right. This map induces a representation of $TL_n(d)$ on $\mathcal{H}_n$ as follows. Let $s = \phi(q)$ for some q. Then $\phi \circ \tau(E_i)$ acts on the i^{th} and $i + 1^{\text{th}}$ qubits, conditioned on the first $i-1$ qubits:

$$\begin{aligned}
\phi\circ\tau(E_i)\left|s_{1:i-1}00s_{i+2:n}\right\rangle &= 0\\
\phi\circ\tau(E_i)\left|s_{1:i-1}01s_{i+2:n}\right\rangle &= \frac{\lambda_{z_i-1}}{\lambda_{z_i}}\left|s_{1:i-1}01s_{i+2:n}\right\rangle + \frac{\sqrt{\lambda_{z_i-1}\lambda_{z_i+1}}}{\lambda_{z_i}}\left|s_{1:i-1}10s_{i+2:n}\right\rangle\\
\phi\circ\tau(E_i)\left|s_{1:i-1}10s_{i+2:n}\right\rangle &= \frac{\lambda_{z_i+1}}{\lambda_{z_i}}\left|s_{1:i-1}10s_{i+2:n}\right\rangle + \frac{\sqrt{\lambda_{z_i-1}\lambda_{z_i+1}}}{\lambda_{z_i}}\left|s_{1:i-1}01s_{i+2:n}\right\rangle\\
\phi\circ\tau(E_i)\left|s_{1:i-1}11s_{i+2:n}\right\rangle &= 0
\end{aligned}\tag{4}$$

where $z_i = 1 + \langle s_{1:i-1}|\, Z_1 + \ldots + Z_{i-1}\, |s_{1:i-1}\rangle$, i.e., it's the index of the i^{th} vertex of the path q. Letting $\phi\circ\tau(E_i)$ act as the identity on the orthogonal complement in $\mathcal{H}_n$ of $\phi(\mathcal{V}_{n,k})$, we have that $\phi\circ\tau(E_i)$ is a unitary operator acting non-trivially on the first $i+1$ qubits.

Composing these maps, we obtain a family of unitary representations $\phi\circ\tau\circ\rho_A : B_n \to \mathcal{B}(\mathcal{H}_n)$ indexed by positive integers k.

Approximating the Jones Polynomial at Roots of Unity

Now that we've defined the Jones polynomial and the braid groups, we are in a position to ask the following question: Given a braid B on n strands with m crossings, can we efficiently find a good approximation for the Jones polynomial $V_{B^{\text{pl}}}$ evaluated at $e^{2\pi i/k}$, for k a positive integer and B^{pl} the plat closure of B, the link obtained by connecting neighboring pairs of string ends on the top and bottom of the braid? We'll see that the structure we've set up in the previous section causes an efficient algorithm [1] for approximating the Jones polynomial fall (more or less) directly into our laps.

Quantum Framing of the Problem

The first step in designing a quantum algorithm to approximate the Jones polynomial is to rewrite the expression we're looking for (which lives on the braid group B_n) in terms of more quantum-flavored quantities (i.e., things that live on the n-qubit Hilbert space $\mathcal{H}_n$). The previous section introduced several algebraic objects and maps between them, the culmination of which was a unitary representation of B_n on $\mathcal{H}_n$. This structure is represented diagrammatically by a commutative diagram, Fig. 2. Now we make use of a series of equalities, passing through the spaces represented in Fig. 2, starting at the top left, moving down to the

$$\begin{tikzcd}
B_n \arrow[r, "V_{\cdot^{\mathrm{pl}}}\left(e^{2\pi i/k}\right)"] \arrow[d, "\rho_A"'] \arrow[dr, phantom, "\text{Jones}"] & \mathbb{C} \\
TL_n(d) \arrow[r, "\mathrm{tr}"] \arrow[d, "\tau"'] \arrow[dr, phantom, "\text{Markov}"] & \mathbb{C} \arrow[u, "f"] \\
\mathcal{B}(\mathcal{V}_{n,k}) \arrow[r, "\phi"] & \mathcal{B}(\mathcal{H}_n) \arrow[u, "\mathrm{Tr}"]
\end{tikzcd}$$

Figure 2: A commutative diagram indicating the equivalent formulation of the Jones polynomial that allows efficient approximation by a quantum computer by providing a unitary representation of the braid group for which the Jones polynomial is closely related to a trace. The names indicate the theorems establishing the commutativity.

bottom left and then over to the bottom right:

$$
\begin{aligned}
V_{B^{\mathrm{pl}}}(A^{-4}) &= V_{C^{\mathrm{tr}}}(A^{-4}) \\
&= (-A)^{3w(B^{\mathrm{pl}})} d^{n-1} \mathrm{Tr}_M(\rho_A(C)) \\
&= (-A)^{3w(B^{\mathrm{pl}})} d^{n-1} \mathrm{Tr}_n(\tau \circ \rho_A(C)) \\
&= (-A)^{3w(B^{\mathrm{pl}})} d^{n-1} N^{-1} \lambda_1 \mathrm{Tr}(\tau \circ \rho_A(C)) \qquad (5) \\
&= (-A)^{3w(B^{\mathrm{pl}})} d^{n-1} N^{-1} \lambda_1 \mathrm{Tr}(\Pi_{n,k} \phi \circ \tau \circ \rho_A(C) \Pi_{n,k}) \\
&= (-A)^{3w(B^{\mathrm{pl}})} d^{n-1} N^{-1} \lambda_1 \mathrm{Tr}(\Pi_{n,k} \phi \circ \tau \circ \rho_A(B) \phi \circ \tau \circ \rho_A(\Gamma) \Pi_{n,k}) \\
&= (-A)^{3w(B^{\mathrm{pl}})} d^{n-1} d^{n/2} N^{-1} \lambda_1 \mathrm{Tr}(\Pi_{n,k} \phi \circ \tau \circ \rho_A(B) \left|\alpha\right\rangle \left\langle\alpha\right| \Pi_{n,k}) \\
&= (-A)^{3w(B^{\mathrm{pl}})} d^{3n/2-1} N^{-1} \lambda_1 \left\langle\alpha\right| \phi \circ \tau \circ \rho_A(B) \left|\alpha\right\rangle .
\end{aligned}
$$

the first equality is just a matter of noticing that the plat closure of a braid can be deformed to show its equivalence to the trace closure of the same braid multiplied by a series of "cap-cups" Γ (on the level of the representation in $TL_n(d)$). The second is the original definition of the Jones polynomial in terms of the Markov trace in the Temperley-Lieb algebra. The third passes into the path representation of the Temperley-Lieb algebra, which admits a Markov trace of its own that must be equivalent to the trace Tr_M by uniqueness. The rest of the equalities use properties of this trace to show equivalence to an expectation in the state $\left|\alpha\right\rangle = \left|1010\ldots\right\rangle$.

Since calculating the writhe of a link is efficient (in fact, linear in the number of crossings), an efficient algorithm for approximating the expectation of $\phi \circ \tau \circ \rho_A(B)$ in the state $\left|\alpha\right\rangle$ gives an efficient algorithm for approximating the Jones polynomial.

The Hadamard Test - Efficient Approximation of Quantum Expectations

Suppose that for Q a unitary operator, we are able to implement a controlled Q gate and that we are able to prepare the state $\left|\psi\right\rangle$. Then the following circuit outputs a classical random variable $r = \pm 1 \pm i$ with expectation $\mathbb{E}[r] = \left\langle\psi\right| Q \left|\psi\right\rangle$. This is known as the Hadamard test:

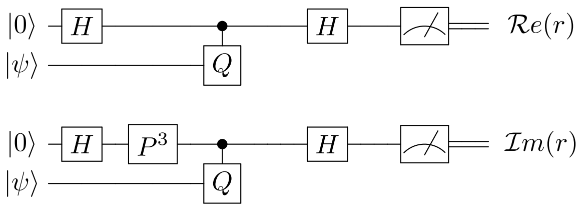

Figure 3: A circuit that implements the Hadamard test, requiring two copies of $\left|\psi\right\rangle$ and two controlled applications of the unitary Q to output a random variable $r = \pm 1 \pm i$ with expectation $\mathbb{E}[r] = \left\langle\psi\right| Q \left|\psi\right\rangle$. The measurements are of the operator Z, so that $\left|0\right\rangle$ gives 1 and $\left|1\right\rangle$ gives -1.

Running the Hadamard test many times, we can estimate the expectation $\mathcal{R}e \left\langle\psi\right| Q \left|\psi\right\rangle$ by taking the sample mean of the real and imaginary parts. How precise is this estimate? The Chernoff-Hoeffding bound tells us that the sample mean of N i.i.d. random variables with mean μ is bounded within δ of μ with probability scaling as $\mathcal{O}(\exp -N\delta^2)$. Introducing the parameter n, we conclude that for $N \sim \mathrm{poly}(n)$ trials, the sample mean is bounded within $\delta \sim 1/\mathrm{poly(n)}$ of μ with probability $p \sim 1 - e^{-\mathrm{poly}(n)}$. Therefore, the Hadamard test, whose outputs are exactly such i.i.d. random variables, gives us a way to efficiently estimate $\left\langle\psi\right| Q \left|\psi\right\rangle$ to polynomial accuracy with exponentially small probability of failure.

Efficient Application of Braid Group Unitaries

It's certainly efficient to prepare $|\alpha\rangle$ with local unitaries, assuming we can initialize the system in some product state for free. Therefore, to make efficient use of the Hadamard test to estimate the expectation value $\langle\alpha|\, \phi \circ \tau \circ \rho_A(B)\, |\alpha\rangle$, it remains only to show that the unitary operators $\phi \circ \tau \circ \rho_A(B)$ may be applied efficiently for $B \in B_n$.

Consider a generator σ_i of the braid group B_n. Recall that it's representative in the Temperley-Lieb algebra $TL_n(d)$ is $\rho_A(\sigma_i) = AE_i + A^{-1}\mathbb{1}$. Then it's representative on the image of $\mathcal{V}_{n,k}$ in $\mathcal{H}_n$ is $\phi \circ \tau \circ \rho_A(\sigma_i) = A\phi \circ \tau(E)_i + A^{-1}\mathbb{1}$. We saw above that in the computational basis states, $\phi \circ \tau(E_i)$ acts nontrivially on the i^{th} and $i+1^{\text{th}}$ qubits, for $i = 1, \ldots, n-1$, conditioned on the final index of the path encoded by the first $i-1$ qubits. Then we can make the same statement about $\phi \circ \tau \circ \rho_A(\sigma_i)$. Since there are only $k-1$ possible final indices, we can store the value of this index in the state of $\log 2k$ auxiliary qubits (or any number larger than $\log(k-1)$). Moreover, we can actually compute this value ℓ_{i-1} by moving along the path, for each of the $\mathcal{O}(n)$ qubits $i = 1, 2, \ldots, i-1$ performing the unitary operation

$$|b\rangle\, |\ell\rangle \mapsto |b\rangle \left|\ell + (-1)^b \mod 2k\right\rangle. \tag{6}$$

This can be done using poly(k) local operations (by the Solovay-Kitaev theorem), so $\mathcal{O}(n)$ iterations uses poly(n, k) elementary gates. Finally, remember that to apply the representative of an arbitrary braid with m crossings, we have to apply this procedure m times, so that $\phi \circ \tau \circ \rho_A(B)$ may be applied with poly(m, n, k) elementary gates, i.e., efficiently in these parameters.

The Algorithm

Now we're done. For $B \in B_n$ a braid on n strands with m crossings, the following algorithm estimates $V_{B^{\text{pl}}}(e^{2\pi i/k})$ to within an additive factor of $\epsilon d^{3n/2}/N$ for $\epsilon \sim 1/\text{poly}(m, n, k)$ using poly(m, n, k) elementary gates, with probability of failure $1/\exp(m, n, k)$:

1. Generate poly(m, n, k) copies of the n-qubit state $|\alpha\rangle = |1010\ldots10\rangle$ and use the Hadamard test to obtain i.i.d. instances of the random variables $r_i = \pm 1 \pm i$ with $\mathbb{E}[r_i] = \langle\alpha|\, \phi \circ \tau \circ \rho_A(B)\, |\alpha\rangle$.

2. Let r be the mean of the r_i. Output $(-A)^{3w(B^{\text{pl}})} d^{3n/2-1} N^{-1} \lambda_1 r$.

The correctness of this algorithm follows from the analysis in this section, noting that multiplicative constants may be absorbed into the polynomial error and that the factor of $(-A)^{3w(B^{\text{pl}})}$ is a complex number of modulus one, so does not affect the magnitude of the error.

This outcome is not quite as good as it seems at first glance. First, we don't know a priori the order of magnitude of the Jones polynomial evaluated at the root of unity, so the error, as a fraction of the true value, may be very high. Second, this additive error grows as d^n, and $d \to 2$ for large k. N is a factor that grows asymptotically as 2^n, as can be seen by invoking reflection principle arguments and looking at asymptotic expansions of binomial coefficients [9][10]. Therefore the total error $\epsilon d^{3n/2}/N$ grows exponentially in general. This is disappointing, but we'll see that the algorithm is still interesting.

Complexity Results

Despite this disappointment, this algorithm is still worth thinking about because it turns out to solve a hard problem. The class **BQP** is the class of "problems that a quantum computer can solve efficiently". It is defined in terms of decision problems, but decision problems and estimation problems are closely linked, since by a simple search procedure an exponentially good estimate can be obtained via polynomially many iterations of a decision algorithm. The following theorem from [2] captures the difficulty of evaluating the Jones polynomial in the language of decision problems:

> *Theorem: Let $k > 4$, $k \neq 6$ be an integer, and $t = exp(2i\pi/k)$ its corresponding root of unity. Let $b \in B_n$ be a braid with $m = poly(n)$ crossings, and b^{pl} its plat closure. Finally, Let $V_{b^{pl}}(t)$ be its Jones polynomial, and $\Delta = (2\cos(\pi/k))^{n/2-1}$. Then given a promise that either $|V_{b^{pl}}(t)| \leq \frac{1}{10}\Delta$ or $|V_{b^{pl}}(t)| \geq \frac{9}{10}\Delta$, it is* ***BQP****-hard to decide between the two.*

Then if we can demonstrate an efficient quantum algorithm to estimate the Jones polynomial to high enough precision, we will have shown that this problem is in fact **BQP**-complete. Notice that the necessary precision allows an additive error growing exponentially in n. We can examine the ratio of the additive error of the algorithm presented above to the gap Δ in the theorem:

$$\frac{\epsilon d^{3n/2}/N}{d^{n/2-1}} = \epsilon d^n/N \sim \frac{\epsilon(2-\delta_k)^n}{2^n} \longrightarrow 0 \tag{7}$$

where all constants in n are absorbed into ϵ, a $1/\text{poly}(n)$ factor. This establishes that the algorithm gives an estimate with high enough precision to solve the decision problem with high probability. Since the algorithm requires polynomial time, this establishes by construction the **BQP**-completeness of estimating the Jones polynomial at (most) roots of unity, even with exponentially large error. What this means is that any problem we can solve efficiently on a quantum computer can be turned into a braid without too much overhead, and the answer obtained by applying the Jones estimation algorithm. A machine that could solve this problem would be a universal quantum computer.

Topological Quantum Computation

Hopefully the previous section suggests that it is valuable to think about representations of the braid group. We've now seen that at least one of these is powerful in the sense that it gives the computational model consisting of string braiding a power equivalent to the gate model of quantum computation. In other words, if we had a machine that applied the appropriate unitary operations to some quantum system when we braiding strands around each other, this would be a universal quantum computer.

Of course, we tend to think about quantum mechanics as living on Hilbert space. But what if there were a physical system on which the braid group had a group action on the Hilbert space of some quantum system via a unitary representation? Then we wouldn't have to deal with implementing the braiding operators via gates on qubits, and could think of the braiding of strands as a (quantum) computational model in its own right. This seems like a different perspective from the circuit model of quantum computation, but in some sense it is the same. We tend to think of quantum circuits as *being* unitary operators on Hilbert space, but it would be perfectly acceptable (and, in fact, models the actual situation in a lab better) to consider instead the group (or even semigroup, since while on the level of Hilbert space you can undo a gate, you can't undo *having applied it*) of circuits built from some elementary gate set, and look for unitary representations.

In fact, there is a physical basis for thinking about quantum computation directly in terms of braiding. The (mostly still theoretical) particles known as *anyons* generalize the classification of indistinguishable particles into bosons and fermions by allowing richer exchange statistics. Anyons live in two dimensions, so their worldlines trace out braids and winding numbers of trajectories become topologically well-defined. These braidings act on an internal Hilbert space via a unitary representation, providing exactly the physical implementation of the braid computation model that we want. This model is known as *topological quantum computation*. The following is what I've managed to learn (so far) from several sources about this subject. It is heavily condensed, incomplete, and presumably idiosyncratic, but hopefully not actually wrong. I've qualified my language to try to make it clear when I'm not sure about something.

Anyons

Mathematically speaking, anyons are the simple objects of *unitary modular tensor categories* (UMTCs) [8]. I won't try to describe exactly what this means, because I don't understand it completely. The essential content of this statement, however, is that any object in the theory may be decomposed into the sum of a number of anyons, which cannot themselves be decomposed, and there is a tensor product structure giving a way to combine objects. There is also a braiding isomorphism. Together, these features capture the physical notions of fusing anyons (really just bringing them close enough together to make measurements on the joint system) and moving them around each other. To specify such a theory, we need a few ingredients [5][4]:

1. a finite set of anyon types $\{\mathbb{1}, a, b, c, \ldots\}$ including a "vacuum" particle $\mathbb{1}$ with the property that $\mathbb{1} \otimes a = a = a \otimes \mathbb{1}$ (at least up to isomorphism)

2. fusion rules $a \otimes b = \oplus_c N^c_{ab} c$ with a, b, c anyon types (categorically speaking, I think these count the number of 2-isomorphism classes of morphisms in $\mathrm{Hom}(a \otimes b, c)$)
3. braiding isomorphisms $a \otimes b \to b \otimes a$ giving rise to the R-matrices R^{ab}_c
4. associativity isomorphisms $(a \otimes b) \otimes c \to a \otimes (b \otimes c)$ giving rise to the F-matrices F^{abc}_d.

These are required to satisfy several consistency conditions, such as the pentagon and hexagon identities, that I will not go into here. UMTCs have the property that the morphisms $\mathrm{Hom}(X, Y)$ have the structure of Hilbert spaces [5]. In the Fibonacci theory [7][5][8], for example, there is one type of non-trivial anyon τ with the fusion rule $\tau \otimes \tau = \mathbb{1} \oplus \tau$, and the following fusion trees represent orthogonal basis vectors:

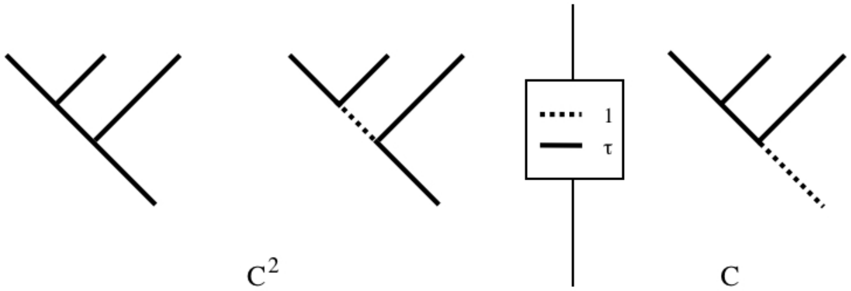

Figure 4: These represent basis vectors for the spaces $\mathrm{Hom}((\tau \otimes \tau) \otimes \tau, \tau)$ (first and second diagrams) and $\mathrm{Hom}((\tau \otimes \tau) \otimes \tau, \mathbb{1})$ (third diagram). I think the same diagram could correspond to multiple basis vectors in the case $N^c_{ab} > 1$, i.e. when $\mathrm{Hom}(a \otimes b, c)$ contains non-isomorphic elements.

Since morphisms compose, it seems reasonable (and is in fact the case!) that the braiding isomorphisms induce transformations on these Hilbert spaces. Using the associativity isomorphisms captured by the F-matrices, the actual matrix elements of these transformations (the R-matrices) can be computed. They are seen to be unitary representations of the braid group. Note that it is *not* the case that the Hilbert space of the anyonic theory is the tensor product of local, single-anyon Hilbert spaces, so they should not be thought of as physical realizations of qudits. This reflects the indistinguishability of the particles. Instead, quantum information is encoded in the outcomes of fusion events. In the Fibonacci theory, three anyons may encode a qubit, since there are two possible ways for them to fuse to form a τ, corresponding to observing a τ or a $\mathbf{1}$ upon fusing the first two. The correspondence of Hilbert spaces to Hom sets physically corresponds to a superselection rule reflecting conservation of anyonic charge, since $\mathrm{Hom}(a, b)$ contains only the zero morphism if $a \neq b$ [5]. In a condensed matter setting in which anyons are quasiparticles rather than elementary particles, I think this means that no high-energy operations are being performed that might effect some change in the topological properties of the system, i.e. creating a single vortex out of the vacuum.

The Power of Topological Quantum Computation

The basic template for using anyons to do topological quantum computation is as follows:

1. Prepare an anyonic state.
2. Implement a braid group transformation by moving anyons around each other.
3. Measure the outcome successively fusing anyons and observing the resulting anyon types.

It remains to determine what the power of this model is, given a particular set of anyons with their fusion rules and braiding statistics. It is clear that some theories possess no computational power whatsoever. Consider, for example, the trivial theory with only the vacuum anyon $\mathbb{1}$. Other theories yield universal quantum computation, such as the Fibonacci anyons [7]. The representation of the braid group used above to compute the Jones polynomial also yields universal quantum computation and arises from an anyonic theory, although seeing this takes some work [6].

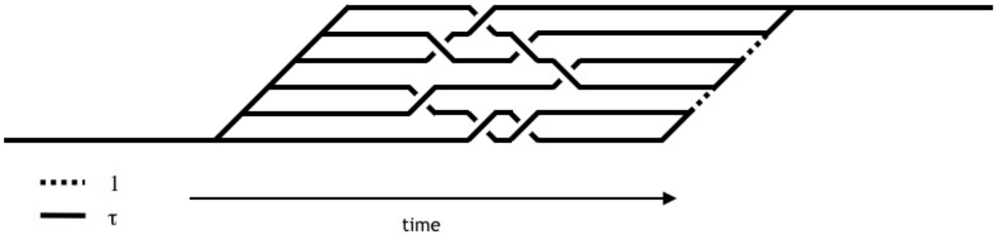

Figure 5: A quantum computation consisting of state preparation, braiding transformation, and measurement of fusion outcomes.

Of course, building a topological quantum computer isn't as simple as going down to the corner store and ordering a few Fibonacci anyons. Physically, these particles arise as exotic excitations of two-dimensional systems - mathematically, in the description of low-energy states of certain field theories. It would be nice, then, to have some intrinsic characterization of the computational power of an anyon model. Showing universality of anyon models relies on reductions: in the case of the Jones representation, to the problem of approximating the Jones polynomial, and in the case of the Fibonacci theory to the problem of simulating quantum circuits (equivalently, by explicitly giving braids corresponding to a universal gate set).

A step towards showing that the Fibonacci anyons are universal for quantum computation is to note that the dimension Hilbert space $\mathrm{Hom}(\tau^{\otimes n}, \tau)$ grows exponentially in n. This is just a combinatorial problem - there are exponentially many fusion trees from n to one τ anyons compatible with the fusion rules. In fact, the sequence of numbers of such trees is precisely the Fibonacci sequence [7]. This is of course not enough to show that the model is universal, but it at least means that an exponentially large problem instance can be encoded.

Another necessary condition for an anyon theory (UMTC) to support universal quantum computation is that it provides representations of the braid groups B_n dense in the unitary group acting on the relevant Hilbert space. This is certainly *not* the case if the image of the braid group is finite. This motivates the following definition and conjecture [8]:

> *Definition: A UMTC $\mathcal{C}$ has property F if the associated representations of B_n on the centralizer algebras End($V^{\otimes n}$) (the hom sets Hom($V^{\otimes}n, V^{\otimes n}$), which become algebras because endomorphisms may be composed) have finite image for all objects V and all $n \in \mathbb{Z}$.*
>
> *Conjecture: A UMTC $\mathcal{C}$ has property F if and only if $\mathcal{D}(\mathcal{C})^2 \in \mathbb{N}$.*

Here, $\mathcal{D}$ is the total quantum dimension of the anyon model defined by

$$\mathcal{D}(\mathcal{C}) = \sqrt{\sum_i d_i^2} \qquad d_a d_b = \sum_c N_{ab}^c d_c \tag{8}$$

with the sum running over all anyon types, i.e. over all simple objects of the category, and the d_i are known as the quantum dimensions of the anyon types. If this conjecture is indeed true, it provides a powerful tool for studying the complexity of computational models based on anyons, as it provides a simple criterion for non-universality given just the fusion rules.

For the Fibonacci theory, we have $d_\tau^2 = d_\tau + 1$ (for any UMTC $d_{\mathbb{1}} = 1$), giving $d_\tau = (1 + \sqrt{5})/2$, the golden ratio. Then $\mathcal{D}^2 = (5 + \sqrt{5})/2$, which is clearly not a natural number. A non-integer value for d_τ seems strange, but serves to reinforce the lesson that it is incorrect to think about the Hilbert space of the theory as a tensor product of single-particle spaces. This isn't so strange - even for fermions or bosons, the Hilbert space on which the representation of the permutation group acts is not the tensor product of single-spin spaces, but rather a one-dimensional space carrying either the trivial or the parity representation. The non-integer dimensions are a consequence of indistinguishability.

The difficulty of assessing the computational power of anyon theories is tied to the difficulty of classifying UMTCs, an effort that is still in its early stages [8]. Luckily, there seem to be many reasons to care about these objects. The computational point of view has been discussed here. From the condensed matter point of view, anyons arise as excitations of topological quantum field theories (TQFTs), which can be put into correspondence with UMTCs [8]. These field theories, in turn, can be used to model exotic phases of matter. From the point of view of quantum gravity, Baez has pointed out in [14] that TQFTs round out the set of theories, with general relativity and quantum field theory on flat spacetime, that achieve two of three of his physical principles of quantumness, background independence, and local degrees of freedom propagating causally, suggesting that there may be insights to be gained by studying these objects on the way to a theory of quantum gravity that unites all three.

References

[1] Aharonov, Jones, Landau. *A Polynomial Quantum Algorithm for Approximating the Jones Polynomial.* 2005.

[2] Aharonov, Arad. *The BQP-hardness of approximating the Jones Polynomial.* 2006.

[3] Panangaden, Paquette. *A categorical presentation of quantum computation with anyons.* 2011.

[4] Rowell. *From Quantum Groups to Unitary Modular Tensor Categories.* 2006.

[5] Blass, Gurevitch. *On quantum computation, anyons, and categories.* 2015.

[6] Delaney, Rowell, Wang. *Local unitary representations of the braid group and their applications to quantum computing.* 2016.

[7] Trebst, Troyer, Wang, Ludwig. *A Short Introduction to Fibonacci Anyon Models.* 2009.

[8] Epelbaum, Lorgat. *Computation in a Topological Quantum Field Theory.* 2015.

[9] Zhao. *Bidirectional Ballot Sequences, Random Walks, and a New Construction of MSTD Sets.* 2009.

[10] Elezovic. *Asymptotic Expansions of Gamma and Related Functions, Binomial Coefficients, Inequalities and Means.* 2015.

[11] Jones. *The Jones polynomial for dummies.* 2014.

[12] Jones. *The Jones polynomial.* 2005.

[13] Pachos. *Introduction to Topological Quantum Computation.* 2012.

[14] Baez. *Higher-Dimensional Algebra and Planck-Scale Physics.* 1999.

[15] Freedman, Larsen, Wang. *A Modular Functor Which is Universal for Quantum Computation.* 2002.