THEORY OF COMPUTING, Volume 11 (6), 2015, pp. 183–219
www.theoryofcomputing.org

How Hard Is It to Approximate the Jones Polynomial?

Greg Kuperberg*

Received August 5, 2009; Revised October 27, 2014; Published June 6, 2015

Dedicated to the memory of François Jaeger (1947-1997)

Abstract: Freedman, Kitaev, and Wang (2002), and later Aharonov, Jones, and Landau (2009), established a quantum algorithm to "additively" approximate the Jones polynomial $V(L,t)$ at any principal root of unity t. The strength of this additive approximation depends exponentially on the bridge number of the link presentation. Freedman, Larsen, and Wang (2002) established that the approximation is universal for quantum computation at a non-lattice, principal root of unity.

We show that any value-distinguishing approximation of the Jones polynomial at these non-lattice roots of unity is #P-hard. Given the power to decide whether $|V(L,t)| < a$ or $|V(L,t)| > b$ for fixed constants $0 < a < b$, there is a polynomial-time algorithm to exactly count the solutions to arbitrary combinatorial equations. Our result is a mutual corollary of the universality of the Jones polynomial, and Aaronson's theorem (2005) that $\mathsf{PostBQP} = \mathsf{PP}$.

Using similar methods, we find a range of values $T(G,x,y)$ of the Tutte polynomial such that for any $c > 1$, $T(G,x,y)$ is #P-hard to approximate within a factor of c even for planar graphs G.

Along the way, we clarify and generalize both Aaronson's theorem and the Solovay-Kitaev theorem.

ACM Classification: F.1.3, G.2.2

AMS Classification: 68Q17, 68Q12, 57M27, 05C31

Key words and phrases: hardness, quantum computation, Jones polynomial, Tutte polynomial

*Partly supported by NSF grants DMS-0606795 and CCF-1013079.

DOI: 10.4086/toc.2015.v011a006

1 Introduction

A well-known paper of Aharonov, Jones, and Landau [5] establishes a polynomial quantum algorithm to approximate the Jones polynomial at any principal root of unity; a more abstract form of this algorithm appeared previously in a paper of Freedman, Kitaev, and Wang [11].

Theorem 1.1 (Freedman, Kitaev, Wang [11]; Aharonov, Jones, Landau [5])**.** *Let $t = \exp(2\pi i/r)$ be a principal root of unity, let L be a link presented by a plat diagram with bridge number g, and let $V(L,t)$ be its Jones polynomial. Then there is a polynomial-time quantum decision algorithm that answers yes with probability*

$$P[\text{yes}] = \left| \frac{V(L,t)}{(t^{1/2}+t^{-1/2})^{g-1}} \right|^2 .$$

(See Burde and Zieschang [9, §2.D] or Section 3.4 for the definition of a plat diagram and its bridge number.)

In the version of the result of Aharonov et al., the algorithm is jointly polynomial time in the r, the order of the root of unity; as well as in the bridge number and the crossing number. They also refine the algorithm to estimate $V(L,t)$ as a complex number rather than just estimating its length. Aharonov et al. describe the error in this algorithm as additive, and note that it would be much harder to provide an algorithm with multiplicative error. Multiplicative approximation (in the sense of the complexity class APX [40]) would mean that $V(L,t)$ or $|V(L,t)|$ can be approximated to within some constant factor $c > 1$.

Another way to distinguish between types of error is to say that the approximation in Theorem 1.1 is *input-dependent*. Given different plat diagrams of the same link L, the error grows exponentially in one of the parameters of the presentation, namely the bridge number. (This additive, input-dependent model of approximating the Jones polynomial was first considered in the converse problem of simulating a quantum computer with the Jones polynomial [7].) An algorithm to approximate the Jones polynomial is only directly useful for topology if the approximation is *value-distinguishing*; i. e., if there is an error bound which is independent of quantities other than the value of $|V(L,t)|$. Multiplicative approximation is one type of value-distinguishing approximation, but it is not the most general kind. For instance, a multiplicative approximation of $\log(1+|V(L,t)|)$ is much weaker than a multiplicative approximation of $|V(L,t)|$ itself, but it is still a value-distinguishing approximation. In general, if an algorithm yield any value-distinguishing approximation of a real-valued function $f(x)$, it means that for each $c \in \mathbb{R}$, there exist real numbers $a < c < b$ such that $f(x) < a$ can be distinguished from $f(x) > b$. (See also Section 2.2.)

Freedman, Larsen, and Wang [13] established that the approximated quantity

$$\left| V(L,t)/\left(t^{1/2}+t^{-1/2}\right)^{g-1} \right|^2$$

in Theorem 1.1 is universal for quantum computation when $r = 5$ or $r \geq 7$. Aharonov and Arad [3] establish an r-uniform version of this result. The exceptions, among principal roots of unity, are $t = \exp(2\pi i/r)$ with $r \in \{1,2,3,4,6\}$. We call these *lattice* roots of unity, because they are the roots of unity for which the ring $\mathbb{Z}[t]$ is a discrete subset of $\mathbb{C}$; the other values of r are *non-lattice* roots of unity.

These results show that even if the approximation is input-dependent, it is computationally valuable for carefully chosen link diagrams.

On the discouraging side, Vertigan [36] showed that it is #P-hard to exactly compute the Jones polynomial $V(L,t)$ except when t is a lattice root of unity. Jaeger, Vertigan, and Welsh [20] established a reduction from the Tutte polynomial of a planar graph to the Jones polynomial of an associated link. Vertigan then showed that the specific values of the Tutte polynomial used in this reduction are #P-hard.

The main result of this article is that the "encouraging" universality result strengthens the "discouraging" hardness result: Any value-distinguishing approximation of a value of the Jones polynomial at a non-lattice root of unity is #P-hard. The argument is a mash-up of three standard theorems in quantum computation: The Solovay-Kitaev theorem [30], the FLW density theorem, and Aaronson's theorem that $\mathsf{PostBQP} = \mathsf{PP}$ [1]. (See also [7] for a different hardness result.)

Theorem 1.2. *Let $V(L,t)$ be the Jones polynomial of a link L described by a link diagram, and let t be a principal, non-lattice root of unity. Let $0 < a < b$ be two positive real numbers, and assume as a promise that either $|V(L,t)| < a$ or $|V(L,t)| > b$. Then it is #P-hard, in the sense of Cook-Turing reduction, to decide which inequality holds. Moreover, it is still #P-hard when L is a knot.*

Theorem 1.2 is proven in Section 3.5 after developing several lemmas. The theorem is stated for the Jones polynomial and only for values where the associated braid group representations are unitary and dense. But the idea applies to many other link invariants and to many non-unitary values of the Jones polynomial. The idea also applies to various functions on graphs or other input data that aren't link invariants. We have no formal statement of a general result, but the basic argument is that if a numerical function can model the execution of a quantum computer sufficiently accurately, then typically multiplicative or value-distinguishing approximation is universal for PostBQP and therefore #P-hard. Here is an example result of this type.

Theorem 1.3. *Let $c > 1$ and let x and y be two real numbers such that $q = (x-1)(y-1) > 4$ and $x, y < 0$, and x and y each have an FPTEAS approximation. Then it is #P-hard to approximate the Tutte polynomial value $T(G,x,y)$ for planar graphs G to within a factor of c.*

Here, a real or complex number has an FPTEAS (fully polynomial-time exponential approximation scheme) if its digits can be computed in FP, for instance if it is an algebraic number (Section 2.2). One interesting ingredient is that we need the Solovay-Kitaev theorem for non-compact Lie groups, Theorem 2.4. (Aharonov, Arad, Eban, and Landau [4] obtained this result for the Lie groups $\mathrm{SL}(d,\mathbb{R})$ and $\mathrm{SL}(d,\mathbb{C})$, which is actually enough for Theorem 1.3.)

We will complete prove Theorem 1.3 in Section 4.5, again after developing some lemmas.

In related results, Aharonov, Arad, Eban, and Landau [4] obtained BQP-universality results about additive approximation to the Tutte polynomial for planar graphs that are clearly related to Theorem 1.3. In particular, as with us, their approach involves a study of non-unitary linear gates. However, their derivation concerns multivariate Tutte polynomials, in which different edges of a graph are allowed different parameters. The value of q must be the same everywhere, but in their version the choice of x (say) is taken from a finite list that satisfies technical conditions. Following Goldberg and Jerrum, we restrict to a single pair of values (x,y).

Goldberg and Jerrum [15] showed that multiplicative approximation of many values of the Tutte polynomial $T(G,x,y)$ is NP-hard (where the reductions are in RP) for non-planar graphs, while some values (those with $q = 4$ and $-1 < y < 0$) are #P-hard. Jaeger, Vertigan, and Welsh [20] also analyzed when $T(G,x,y)$ is #P-hard to compute exactly. They noted that the Jones polynomial $V(L,t)$ of an alternating link L is equivalent to $T(G,x,y)$ for a planar graph G along the curve $xy = 1$. More recently [16], Goldberg and Jerrum also established that many values of the planar Tutte polynomial are NP-hard to approximate. Their new theorems apply to those values of (x,y) in Theorem 1.3 with $q > 5$ (and some other values that we do not analyze), but their constructions are very different. Moreover, we establish #P-hardness, while their planar constructions only establish NP-hardness. On the other hand, we use Goldberg and Jerrum's gadget idea to change from one value of (x,y) to another for a fixed value of q.

Remark 1.4. The first version of this article contained a significant mistake, which the reader may grasp after reading Section 2.5. The author supposed that all of the implementations of quantum gates could have complexity $\mathrm{poly}(1/\varepsilon)$ (or FPTAS approximability) in the proof of both Theorem 1.2 and Theorem 1.3, because this complexity is sufficient to express the complexity class BQP. We actually need complexity $\mathrm{poly}(-\log(\varepsilon))$ (or FPTEAS) to express the complexity class PostBQP, because this class unavoidably needs exponentially small probabilities. Fortunately, the Solovay-Kitaev theorem (Theorem 2.4) satisfies this stringent approximation requirement. See also Lemma 4.3 and Theorem 2.10 for our corrected constructions.

Acknowledgments

The author would like to thank Scott Aaronson, Dorit Aharonov, Leslie Ann Goldberg, and Eric Rowell for helpful discussions. The author would also like to thank the referees for their meticulous remarks.

2 Complexity theory

2.1 Complexity classes

We assume that the reader is somewhat familiar with complexity classes such as P, NP, BQP, #P, and the notation that A^{B} means the class A with oracle B. See the Complexity Zoo [40] and Nielsen and Chuang [30] for a review.

Whereas a problem in the class #P counts the number of witnesses accepted by a verifier in polynomial time, and a problem the class NP reports whether there is an accepted witness, a problem in the class PP reports whether a majority of the witnesses are accepted.

Proposition 2.1. *A problem is* #P*-hard if and only if it is* PP*-hard with respect to Cook-Turing reduction, i. e.,*

$$\mathsf{P}^{\mathsf{PP}} = \mathsf{P}^{\#\mathsf{P}}.$$

Proposition 2.1 is given as an exercise for the reader in the Complexity Zoo [40]. (Hint: Binary search.) It is one reason that we use Cook-Turing reduction in the statement of Theorem 1.2.

A problem which is #P-hard is also hard for the polynomial hierarchy PH, by the deeper theorem due to Toda [34] that

$$\mathsf{PH} \stackrel{\text{def}}{=} \bigcup_{n=1}^{\infty} \underbrace{\mathsf{NP}^{\mathsf{NP}^{\cdot^{\cdot^{\cdot^{\mathsf{NP}}}}}}}_{n} \subseteq \mathsf{P}^{\#\mathsf{P}}.$$

The class NP with a tower of $n-1$ NPs as an oracle is called the nth level of the polynomial hierarchy. One of the standard conjectures in complexity theory is the polynomial hierarchy does not collapse, i. e., that nth level does not equal the $n+1$st level for any n. Thus by Toda's theorem, if a problem is #P-hard, then it is viewed as qualitatively harder than if it is merely NP-hard.

2.2 Approximation classes

The approximation classes listed in the Complexity Zoo [40] that express multiplicative approximation include APX, PTAS, and FPTAS. These classes are defined there for optimization problems, but they can equally well be defined for arbitrary functional problems. Let $f : \Sigma^* \to \mathbb{R}_+$ be a function that takes bit strings x to positive real numbers. Then $f(x)$ is in APX if it can be approximated to within some bounded factor in polynomial time (with fixed-point output); it is in PTAS if it can be approximated to within a factor $1+\varepsilon$ in polynomial time for any $\varepsilon > 0$; and it is in FPTAS if the computation is jointly polynomial time in the bit length $|x|$ and $1/\varepsilon$. (These classes all refer to deterministic computation; there are analogous randomized classes such as FPRAS.)

We will need a stricter version of FPTAS. For many approximate numerical algorithms, although not usually for optimization problems, the computation time is jointly polynomial in $|x|$ and $-\log(\varepsilon)$. We call such an approximation scheme an FPTEAS, or *fully polynomial time, exponential approximation scheme*. In particular every algebraic number has an FPTEAS, using standard numerical algorithms to find its digits.

Indeed, much more is true: The digits of algebraic numbers, and the values of many other elementary functions such as exponentials and logarithms, can be computed in quasilinear time in the RAM machine model [8]. Most numbers that arise in calculus derivations have quasilinear digit complexity; nearly all of them have polynomial digit complexity.

We do not know of a standard complexity class to express general value-distinguishing approximation, so we define such a class here, APV. Again let $f : \Sigma^* \to \mathbb{R}_+$. Then f is in APV if for every constant $a > 0$, there exists a constant $b > a$ and a polynomial-time algorithm to decide whether $f(x) > b$ or $f(x) < a$, given the promise that one of the two is true. Similarly, we could define a randomized version ARV. Also, both APV and ARV have a variation in which the constant a is an input to one universal algorithm, instead of asking for an algorithm for each value of a.

The following proposition says that if $f(x)$ can be suitably rescaled, then general value-distinguishing approximation becomes equivalent to multiplicative approximation in the sense of APX. Proposition 2.2 and its proof are similar to that of Proposition 2.14, in particular similar to the rescaling of Aaronson [1, Thm 3.4]. We will need the contrapositive of Proposition 2.2 in the proof of Theorem 1.2.

Proposition 2.2. *Suppose that $f(x)$ takes positive real values and is in* APV*, and suppose further that $|\log(f(x))|$ is bounded by a polynomial in the bit length $|x|$ of the input. Suppose that there are constants*

$c > 1$ and $k > 1$ such that for every integer n, there is a reduction $y_n(x)$ such that

$$f(x) < k^n f(y_n(x)) < cf(x),$$

and suppose that this reduction can be computed in joint polynomial time in n and in $|x|$. Then $f(x)$ is in APX.

Proof. Let a and b be some constants such that we can decide by a subroutine whether $f(x) < a$ or $f(x) > b$ in polynomial time. Then we can bound $f(x)$ to within a factor of cb/a. We know by hypothesis that $f(x) > k^{-m}$ and $f(x) < k^m$ for some m which is polynomial in $|x|$. So the strategy is to ask whether $f(y_n(x))$ is less than a or more than b for every $|n| \leq m$. The largest n for which the subroutine reports that $f(y_n(x)) < a$ yields a good estimate of ak^{-n}. The estimate is within a factor of cb/a, even though the subroutine could give a false yes answer when $f(y_n(x)) > b$. □

2.3 Quantum computation

We cannot give a full review of quantum computation in this article. There are many equivalent models of quantum computation, and we would simply like to carefully describe the one that we will use. Let $D : \Sigma^* \to \{\text{yes}, \text{no}\}$ be a decision problem, a function $D(x)$ on bit strings x that takes the values "yes" and "no." In the most standard definition of BQP, we assume a uniform family of quantum circuits C such that x is supplied in input qubits along with ancillas, and one of the output qubits is the output $D(x)$ with good probability. We will use a variation of this definition in which the input is encoded in the circuit rather than in the input to its gates; and the inputs and outputs are all set to 0.

Proposition 2.3. *$D \in$ BQP if and only if there is a quantum circuit $C = C(x)$ with $\mathrm{poly}(|x|)$ unitary gates acting on $n = \mathrm{poly}(|x|)$ qubits, such that C itself can be generated in deterministic polynomial time* FP*, and such that the probability*

$$p(x) = |\langle 0^n|C|0^n\rangle|^2 \tag{2.1}$$

is at least $2/3$ if $D(x) = \text{yes}$ and at most $1/3$ if $D(x)$ is no.

Proposition 2.3 is a well-known result even though it is not the most standard definition. The proof uses the "uncomputation" method.

Proof. We first assume a circuit $C = C(n)$ of the more standard type in which $|x\rangle$ is the input along with $|0\rangle$ ancillas, and one of the qubits is the output. Then we can make a new circuit C' whose input is all ancillas, and that first changes some of the ancillas to $|x\rangle$. One of the outputs $|y\rangle$ of C' agrees with $D(x)$ with probability at least $2/3$; the other outputs are unpredictable. We make a new circuit C'' that applies C', then copies $|y\rangle$ to a fresh ancilla with a CNOT gate, and then applies $(C')^{-1}$. □

2.4 Solovay-Kitaev

In this section we will analyze a central result in quantum computation, the Solovay-Kitaev theorem. Let BQP_Γ be the class BQP defined by some universal finite gate set Γ. If each gate in Γ has at least an FPTAS, then the Solovay-Kitaev theorem implies that BQP_Γ does not depend on the choice of Γ

and can be called BQP. We need some approximability condition here: If the matrix entries of gates in Γ have intractable or uncomputable information, then BQP_Γ also carries intractable or uncomputable information [2, Thm. 5.1].

In this paper we will need the more delicate class PostBQP. As stated in Theorem 2.10, in order to know that $\mathsf{PostBQP}_\Gamma$ is independent of Γ, we need to assume that every gate in Γ has an FPTEAS, and not just an FPTAS. One special case which is widely used in quantum computation and which we need for Theorem 1.2 is gates with algebraic entries; happily, all algebraic gates have an FPTEAS. (Indeed the FPTEAS class is far more general, as explained in Section 2.2.) We also need the Solovay-Kitaev theorem to have polylogarithmic overhead; happily it does.

Finally, for Theorem 1.3 we will need the Solovay-Kitaev theorem for non-compact Lie groups. The theorem was originally proven in the case $G = \mathrm{SU}(d)$. This case is explained in Nielsen and Chuang [30]; as far as we know the proof works without change when G is any compact, semisimple Lie group. Aharonov, Arad, Eban, and Landau [4] derive a version of this theorem for the Lie groups $\mathrm{SL}(d,\mathbb{R})$ and $\mathrm{SL}(d,\mathbb{C})$, which are not compact but still semisimple. Their result is enough for Theorem 1.3; here we show that the traditional argument applies to a more general class of Lie groups.

Theorem 2.4 (Solovay, Kitaev)**.** *Let G be a connected Lie group whose Lie algebra $\mathfrak{g}$ is perfect. Let Γ be a finite set of elements (closed under taking inverses) that densely generates G, and let $g \in G$. Suppose that there is an* FPTEAS *for g and every element of Γ. Then there is a word made from Γ that approximates g,*

$$d(g_1 g_2 \dots g_m, g) \le \varepsilon\,,$$

where the length m and the (deterministic) computation time to find the word are both $\mathrm{poly}(-\log(\varepsilon))$ *(non-uniformly in the choice of G, Γ, and g).*

Before turning to the proof of Theorem 2.4, we discuss some basics of Lie theory. (See Varadarajan [35].)

A Lie group G is a real analytic manifold with a real analytic group law. (Or a smooth manifold or even just a topological manifold; it turns out that the group law induces a unique real analytic structure.) Its Lie algebra $\mathfrak{g} = T_1 G$ is by definition the tangent space at the identity. We assume that our Lie group G is given with some tractable algorithm for computing the group law in real analytic coordinates. For example, G could be a real algebraic group, by definition a Lie group that can be realized (non-uniquely) by polynomial equations in some $\mathrm{GL}(n,\mathbb{R})$.

We can give G a metric to discuss approximation to points in G. The most natural choice is a left-invariant Riemannian metric [31]. Every left-invariant Riemannian metric comes from a positive definite inner product on the Lie algebra $\mathfrak{g}$ of G. Two different inner products on $\mathfrak{g}$ plainly yield different Riemannian metrics on G, but they are they are bi-Lipschitz equivalent. (If d_1 and d_2 are two metrics on a set, then they are *bi-Lipschitz equivalent* if $d_1(p,q) = \Theta(d_2(p,q))$.) A left-invariant metric on $\mathrm{GL}(n,\mathbb{R})$ is not bi-Lipschitz equivalent with Euclidean distance between matrices, but it is equivalent on any bounded set. Thus, any of these choices of metric are equivalent for the purpose of stating Theorem 2.4.

The usual way to understand the structure of a Lie group G is to begin with its Lie algebra $\mathfrak{g}$. A finite-dimensional Lie algebra $\mathfrak{g}$ is *semisimple* if it is a direct sum of non-abelian, simple Lie algebras. It is *perfect* if $\mathfrak{g} = [\mathfrak{g},\mathfrak{g}]$, i. e., $\mathfrak{g}$ is the linear span of all Lie brackets of pairs of its elements. (A semisimple Lie algebra is analogous to a direct product of non-abelian, finite simple groups; a perfect Lie algebra is analogous to a finite perfect group.) The most commonly used Lie algebras, such as $\mathfrak{su}(d)$ and $\mathfrak{sl}(n,\mathbb{R})$, have simple and therefore semisimiple Lie algebras (and are themselves called semisimple groups). Every semisimple Lie algebra is perfect, but there are perfect Lie algebras that are not semisimple. For example, if V is a linear representation of a semisimple Lie group G without any trivial summand, then the Lie algebra of the semidirect product $G \ltimes V$ is perfect.

Every Lie group G has a (real analytic) *exponential map*

$$\exp : \mathfrak{g} \to G$$

defined in polar coordinates by the derivative equation

$$\frac{d}{dt}\exp(tx) = x\exp(tx)$$

for $t \in \mathbb{R}_{\geq 0}$ and $x \in \mathfrak{g}$. In the special case of an algebraic group, it is the usual matrix exponential. We will use three standard results about the derivative map. To state the results, we assume some inner product on $\mathfrak{g}$, and the induced left-invariant metric on G.

Proposition 2.5 ([35, Thm. 2.10.1]). *The exponential map* $\exp$ *is a bi-Lipschitz, diffeomorphic embedding when restricted to a ball $B = B(0,\varepsilon)$ of some radius ε in $\mathfrak{g}$.*

Proposition 2.6 ([35, Thm. 2.10.1]). *Suppose that $\mathfrak{g}$ has a basis $b_1,\ldots,b_k$, and define a function $h : \mathfrak{g} \to G$ by*

$$h\left(\sum_j t_j b_j\right) = \prod_j \exp(t_j b_j)\,.$$

Then f is a bi-Lipschitz embedding when restricted to a ball $B = B(0,\varepsilon)$ of some radius ε in $\mathfrak{g}$. Moreover, we can choose ε and δ so that f is uniformly bi-Lipschitz for any basis $b'_1,\ldots,b'_k$ with $\|b'_j - b_j\| < \delta$.

Proposition 2.6 is less standard than Proposition 2.5, but happily Varadarajan proves a mutual generalization in a single theorem. The last statement about uniform constants if the basis $\{b_j\}$ is perturbed is not in the statement of the theorem, but it follows readily from the proof. Remark: The formula in Proposition 2.6 is a generalization of Euler angles for the group $\mathrm{SO}(3)$.

Proposition 2.7 ([35, Thm. 2.12.4]). *If $[g,h]_G = ghg^{-1}h^{-1}$ is the group commutator and $[x,y]_\mathfrak{g}$ is the Lie bracket, then*

$$[\exp(x),\exp(y)]_G = \exp([x,y]_\mathfrak{g} + O\big(\max(\|x\|,\|y\|)^3)\big)\,.$$

Varadarajan proves Proposition 2.7 with a less uniform error estimate, but the same proof establishes the given formula.

The plan of our proof of Theorem 2.4 is not very different from the standard proof in Nielsen and Chuang [30]: For some constant $r < 1$, we create a set of elements in G that, under the inverse of the exponential map, is a basis of $\mathfrak{g}$ at the scale r^n. In fact, it always approximately the same basis. These bases are formed from commutators at larger scales. Finally, every element $g \in G$ can first be brought within the unit ball of the identity and then whittled away to smaller and smaller scales with these bases.

Since the result is not required to be uniform in g, we do not need a global epsilon net of the Lie group G, only a local one near the identity; a global epsilon net would add extra difficulties in the non-compact case. Another trick that simplifies the derivation is to save the choice of r for the end; it also serves as a fudge factor to enable the construction.

Proof of Theorem 2.4. Let k be the dimension of G. If $\mathfrak{g}$ is a perfect Lie algebra, then it has a basis $b_1,\dots,b_k$ and elements $x_1,\dots,x_k$ and $y_1,\dots,y_k$ such that $[x_j,y_j]=b_j$. We choose some positive definite inner product on $\mathfrak{g}$ and take the induced left-invariant Riemannian metric on G.

By Proposition 2.5, the exponential map $\exp:\mathfrak{g}\to G$ is a bi-Lipschitz diffeomorphism within some radius ε_1. Also, let ε_2 and δ be the constants produced by Proposition 2.6, a radius out to which the map f is a bi-Lipschitz diffeomorphism. Also, since the Lie bracket is bilinear, and by the approximation in Proposition 2.7, we can choose a radius ε_3 within which both the Lie bracket on $\mathfrak{g}$ and the group commutator on G take the ball $B_3=B(0,\varepsilon_3)$ to itself. In other words, both brackets are maps

$$[\cdot,\cdot]_{\mathfrak{g}}:B_3\times B_3\to B_3 \qquad \text{and} \qquad [\cdot,\cdot]_G:\exp(B_3)\times\exp(B_3)\to\exp(B_3)$$

when ε_3 is small enough. Finally we choose

$$\varepsilon_0=\min(\varepsilon_1,\varepsilon_2,\varepsilon_3)$$

to obtain all three properties simultaneously, and we let $B_0=B(0,\varepsilon_0)$.

We take advantage of a subtlety of Proposition 2.6, that the map h only depends on the lines spanned by $\{b_j\}$. We can thus rescale the vectors $\{x_j,y_j,b_j\}$ so that they all lie in B_0, without disturbing the constants used to define B_0.

We can interpret the group commutator $[\cdot,\cdot]_G$ as a map from $B_0\times B_0$ to B_0 via the equation

$$[x,y]_G\stackrel{\text{def}}{=}\log\big([\exp(x),\exp(y)]_G\big)$$

so that we can then say restate Proposition 2.7 as saying that

$$[x,y]_G=[x,y]_{\mathfrak{g}}+O\big(\max(\|x\|,\|y\|)^3\big)\,. \tag{2.2}$$

Without loss of generality, $g\in\exp(B_0)$: Because Γ densely generates G, we can find a word close to g and multiply g by its inverse. Also, we let $r<1$ be a constant that will be chosen at the end of the proof. Again because Γ densely generates G, we can assume for each $n\le 3$ that it contains the set $\{\exp(b_{j,n})\}$ for a basis $\{b_{j,n}\}$ in B_0 such that

$$\|r^{-n}b_{j,n}-b_j\|<\delta \tag{2.3}$$

for every j. Recall again that δ is chosen to match Proposition 2.6.

In the remainder of the proof, we will use asymptotic notation such as $x=O(r)$ to express errors in Lie elements $x\in\mathfrak{g}$. What we mean is that $\|x\|<Cr$, where each constant C does not depend on r or n, but can depend on everything else defined so far.

For each integer $n\ge 1$, we want to define Lie algebra elements $b_{j,n}$, $x_{j,n}$, and $y_{j,n}$, all of them words in Γ made using the group law of G, such that (2.3) holds for all n, and such that

$$x_{j,n}=r^n(x_j+O(r)) \qquad \text{and} \qquad y_{j,n}=r^n(y_j+O(r)) \tag{2.4}$$

also holds for all n. The definition is by an inductive algorithm that makes $x_{j,n}$ and $y_{j,n}$ from $b_{j,n+1}$, and makes $b_{j,n}$ from $x_{j,\lceil n/2\rceil}$ and $y_{j,\lfloor n/2\rfloor}$. So the numbering in n is slightly out of order, but since we have already produced $b_{j,n}$ for $n \le 3$, the induction works.

For each $n \ge 1$, we choose integers $t_j = O(r^{-1})$ so that the expressions

$$\log\left(\prod_j \exp(t_j b_{j,n+1})\right) \tag{2.5}$$

are as close as possible to $r^n x_j$ and $r^n y_j$. We set $x_{j,n}$ and $y_{j,n}$ to be these approximations. We claim that the expressions in (2.5) form an $O(r^{n+1})$)-net of $r^n B_0$. We argue this in stages:

1. The sums $\sum_j t_j r^{n+1} b_j$ are a lattice and an $O(r^{n+1})$-net by rescaling.

2. The sums $\sum_j t_j b_{j,n+1}$ are an $O(r^{n+1})$-net because (2.3) limits the distortion of the lattice.

3. The products $\prod_j \exp(t_j b_{j,n+1})$ are an $O(r^{n+1})$-net because the map h in Proposition 2.6 is Lipschitz on B_0.

4. The logarithms

$$\log\left(\prod_j \exp(t_j b_{j,n+1})\right)$$

are an $O(r^{n+1})$-net because the exponential map exp is inverse Lipschitz on B_0.

Thus, we obtain the error estimates (2.4).

For each $n \ge 4$, we let

$$b_{j,n} = [x_{j,\lceil n/2\rceil}, y_{j,\lfloor n/2\rfloor}]_G\,.$$

If we combine (2.4) with (2.2), we obtain

$$b_{j,n} = r^n(b_j + O(r) + O(r^{3\lfloor n/2\rfloor - n})) = r^n(b_j + O(r)). \tag{2.6}$$

We would like to reconcile (2.6) with (2.3). The relation (2.6) gives us

$$\|r^{-n} b_{j,n} - b_j\| < Cr$$

and we are done provided that $Cr < \delta$. So, at final this stage it is crucial that C does not depend on n or r; we can choose r small enough to make the induction work.

Finally we let $g_0 = g \in \exp(B_0)$. We inductively let

$$h_n = \prod_j \exp(b_{j,n+1})^{t_j}$$

as in (2.5), and then we let $g_{n+1} = h_n^{-1} g_n$. We obtain the estimate

$$\|\log(g_{n+1})\| = O(r^{n+1})\,.$$

It is easy to check by induction that the word length of each $\exp(b_{j,n})$ is $O(n^2)$ (non-uniformly in r, but r is now fixed). Therefore the word length of the product $h_1 h_2 \ldots h_n$ is $O(n^3)$. Also all of the work to find these words is polynomial in n. $\square$

Theorem 2.4 is not uniform in the choice of the group element g and we do not need this uniformity for our purposes. However, the proof shows that it is uniform on any bounded region in G. For completeness, we give a complementary result that in any semisimple algebraic group, any element can be efficiently approximated to within a bounded distance.

Theorem 2.8. *Let G be a semisimple (real) algebraic group which is equipped with a left-invariant Riemannian metric, and which is densely generated by a subset Γ. Let $r > 0$, let $g \in G$, and let $\ell = d(g,1)$. Then there is word made from Γ that approximates g to within a bounded distance,*

$$d(g_1 g_2 \dots g_m, g) < r$$

with $m = O(\ell+1)$ uniformly in g. Moreover, such a word can be found in time $\mathrm{poly}(\ell)$.

Evidently Theorem 2.8 can be combined with Theorem 2.4 to obtain a total word length of

$$m = O(\ell+1) + \mathrm{poly}(-\log(\varepsilon))\,.$$

Note also that the lower bound $m = \Omega(\ell+1)$ follows from the triangle inequality

$$d(1,gh) \leq d(1,g) + d(1,h)$$

and the fact that the finite set Γ has a maximum distance to 1. So Theorem 2.8 is optimal up to a constant factor.

We conjecture that Theorem 2.8 holds for all connected Lie groups. Note that most named Lie groups, such as $\mathrm{GL}(n,\mathbb{R})$, $O(n,\mathbb{C})$, etc., are algebraic groups.

Proof. We assume that G is given as a subgroup of some $\mathrm{GL}(n,\mathbb{R})$ defined by polynomial equations. We review some of the structure theory of semisimple real algebraic groups [31]:

1. G has a maximal compact subgroup K.

2. Every element $g \in G$ has a (canonical) Cartan decomposition $g = \exp(x)k$, where $k \in K$ and $x \in \mathfrak{k}^\perp \subseteq \mathfrak{g}$.

3. The quotient manifold G/K has a G-invariant Riemannian metric; it is then called a *symmetric space of noncompact type*.

4. In the quotient G/K, the unique geodesic connecting $gK = \exp(x)K$ to the identity coset is given by $\exp(tx)K$ with $0 \leq t \leq 1$.

5. Up to a change of basis, $G = G^T$, i. e., G is stable under the transpose map. $K = G \cap O(n)$ is a maximal compact subgroup if and only if $G = G^T$.

6. If $G = G^T$, then the Cartan decomposition $g = \exp(x)k$ coincides with the polar decomposition for matrices, so that x and $\exp(x)$ are symmetric matrices.

Note also that every G-invariant metric on G/K comes from a left-invariant metric on G which also happens to be right-K-invariant. We assume such a metric on G. As a consequence, given any two group elements $g,h \in G$, we have both that

$$d(gK,hK) \leq d_G(g,h)$$

and that equality can be achieved by passing to a different representative $g' \in gK$ or $h' \in hK$. (We need not change both.)

The idea of our proof is to first find a word with all of the desired properties in the symmetric space G/K rather than in the group G. The advantage of working in G/K is that we know how to calculate geodesics and distances, using polar decompositions. Geometrically, the idea is not complicated: We can build a word by taking steps approximately in the direction of the geodesic from $1K$ to gK.

Since Γ densely generates G, and since closed and bounded regions in G are compact, we can assume without loss of generality that Γ contains an $r/2$-net of points inside the closed ball $\overline{B} = \overline{B(1,r)}$ of radius r at the identity. Given $gK \in G/K$, let γ be the unique geodesic that connects $1K$ to gK; we can compute it from the polar decomposition of g. Let hK be the point at which γ exits $\overline{B}K$. Then we know or we can assume that

$$d(1K,hK) = d(1,h) = r \qquad \text{and} \qquad d(hK,gK) = d(h,g) = \ell - r.$$

We can choose $g_1 \in \Gamma$ such that $d(g_1,h) < r/2$. By the triangle inequality,

$$d(g_1,g) = d(1,g_1^{-1}g) < \ell - \frac{r}{2}.$$

Thus, we can let $g' = g_1^{-1}g$ and proceed by induction.

We obtain a word w such that $d(w^{-1}g,K) < r/2$. We are given that K is compact; it follows that there is a finite set of words v in Γ that forms an $r/2$-net of K. So for one of these words,

$$d(wv,g) = d(v,w^{-1}g) < \frac{r}{2} + \frac{r}{2} = r$$

as desired. □

2.5 Postselection

Aaronson [1] defined the class PostBQP as polynomial-time quantum computation with free retries, or postselection. In other words, the computation can output $|\text{yes}\rangle$, $|\text{no}\rangle$, or $|\text{retry}\rangle$. (In Aaronson's formal definition, the outputs are measured as $\langle 00|$, $\langle 01|$, and $\langle 1*|$, respectively; of course the output can equally well be a qutrit whose values are renamed semantically.) If the absolute probabilities are

$$P[\text{yes}] = a \qquad \text{and} \qquad P[\text{no}] = b$$

then the conditional or postselected probabilities are

$$P[\text{yes} \mid \text{yes or no}] = \frac{a}{a+b} \qquad \text{and} \qquad P[\text{no} \mid \text{yes or no}] = \frac{b}{a+b}.$$

An algorithm in $\mathsf{PostBQP}$ is required to output "yes" or "no" with conditional (rather than absolute) probability of at least $2/3$. It is trivially equivalent to say that for some $c > 1$, either $a > cb$ or $b > ca$; all values of c are equivalent because c can be amplified by repeated trials. There is an analogous class $\mathsf{PostBPP}$ for classical randomized computations; it was also defined previously as $\mathsf{BPP}_{\text{path}}$. Aaronson established that $\mathsf{PostBQP} = \mathsf{PP}$. It is not hard to show that $\mathsf{PostBQP}$ is a subset of PP, just as BQP, NP, and a number of other important classes are known to be. (The inclusion $\mathsf{SBQP} \subseteq \mathsf{A_0PP}$ is proved in the same way in Proposition 2.13.) The more surprising fact is that $\mathsf{PostBQP}$ is all of PP.

By contrast, $\mathsf{PostBPP}$ is unlikely to be all of PP. The relevant complexity results are as follows:

1. $\mathsf{PostBPP}$ contains $\mathsf{P}^{||\mathsf{NP}}$ (P with parallel NP queries) [17].

2. $\mathsf{P}^{||\mathsf{NP}}$ equals $\mathsf{P}^{\mathsf{NP}[\log]}$ (P with logarithmically many NP queries) [19, 10].

3. $\mathsf{PostBPP}$ derandomizes to $\mathsf{P}^{||\mathsf{NP}}$. I. e., they are equal if sufficiently good pseudo-random number generators exist [33].

4. Without any derandomization assumption [17],

$$\mathsf{PostBPP} \subseteq \mathsf{BPP}^{\mathsf{NP}} \subseteq \mathsf{NP}^{\mathsf{NP}^{\mathsf{NP}}}.$$

Thus, $\mathsf{PostBPP}$ is known to be in the third level of PH. If we accept derandomization, then it is in the second level.

Another interpretation of $\mathsf{PostBQP}$ or $\mathsf{PostBPP}$ is given by the following proposition:

Proposition 2.9. *Let $c > 1$. Then a decision function D is in* $\mathsf{PostBPP}$ *if and only if there are two randomized, polynomial time algorithms run by Alice and Bob that report "yes" with probabilities a and b, and such that $D(x) = \text{yes}$ when $a > cb$ and $D(x) = \text{no}$ when $b > ca$. The same holds for* $\mathsf{PostBQP}$ *and quantum algorithms.*

Proof. Suppose that we are given a $\mathsf{PostBQP}$ algorithm in the original definition. Then Alice and Bob can both run this algorithm, with the following conversion:

$$\begin{aligned} \text{yes} &\mapsto \text{Alice yes, Bob no}\,, \\ \text{no} &\mapsto \text{Alice no, Bob yes}\,, \\ \text{retry} &\mapsto \text{Alice no, Bob no}\,. \end{aligned}$$

It is easy to check that this satisfies the requirements of the proposition. Conversely, suppose that Alice and Bob have separate algorithms. Then we can combine them into one postselecting algorithm in Aaronson's sense by flipping a coin to decide which of Alice or Bob runs; only one of them runs in a given trial. We can convert according to the following table:

$$\begin{aligned} \text{Alice yes} &\mapsto \text{yes}\,, & \text{Alice no} &\mapsto \text{retry}\,, \\ \text{Bob yes} &\mapsto \text{no}\,, & \text{Bob no} &\mapsto \text{retry}\,. \end{aligned}$$

It is easy to check that this conversion satisfies Aaronson's definition. □

We also need to clarify the definition of PostBQP with regard to different gate sets. Aaronson defines PostBQP using Hadamard and Toffoli gates, on the argument that all choices of gates are equivalent by Solovay-Kitaev. But this is somewhat overstated; we give a more precise equivalence as follows:

Theorem 2.10. *Let Γ be a universal gate set acting on qudits, let* $\mathsf{PostBQP}_\Gamma$ *be* PostBQP *defined with the gate set Γ, and suppose that:*

1. *The matrix entries in each gate have an* FPTEAS.

2. *If $z \neq 0$ is expressible as an integer polynomial in the gate entries with bit complexity* $\mathrm{poly}(n)$ *with exponents written in unary, then*
$$|z| > \exp(-\mathrm{poly}(n)) .$$

Then $\mathsf{PostBQP} = \mathsf{PostBQP}_\Gamma$. *If only condition 1 holds, then* $\mathsf{PostBQP} \subseteq \mathsf{PostBQP}_\Gamma$.

Before proving Theorem 2.10, here are three remarks. First, the class BQP only requires a weaker version of condition 1, namely that each gate in Γ has an FPTAS, in order to enable the Solovay-Kitaev theorem. We need FPTEAS because PostBQP relies on exponentially small probabilities. Without exponentially good approximation, Solovay-Kitaev would still give us a circuit reduction, but the reduction would be relative to P/poly rather than relative to P. Second, we conjecture that if only condition 1 holds, then PostBQP and $\mathsf{PostBQP}_\Gamma$ are not always equal. Third, we do not know whether postselected quantum computation is gate-independent with a time bound of $\tilde{O}(n^\alpha)$ for some fixed exponent α, because the Solovay-Kitaev theorem could change the exponent.

Proof. Condition 1 and Theorem 2.4 together imply that $\mathsf{PostBQP} \subseteq \mathsf{PostBQP}_\Gamma$. The traditional gate set consisting of Hadamard and Toffoli gates can be approximated using gates in Γ; how good of an approximation is sufficient? It is easy to check that the Hadamard and Toffoli gates satisfy condition 2, so the strength of approximation that we need is $\exp(-\mathrm{poly}(|x|))$. This is precisely how much Theorem 2.4 gives us with polynomial overhead, if each gate in Γ has an FPTEAS.

The same argument works in reverse, but we must add condition 2 explicitly, since it is not guaranteed in general. □

We will not strictly need the following proposition, but it helps for understanding Theorem 2.10. It shows that any gate set with algebraic matrix entries automatically satisfies condition 2.

Theorem 2.11. *Let $t_1,\ldots,t_k$ be a finite list of algebraic numbers in $\mathbb{C}$, and let p be an integer polynomial in k variables with bit complexity* $\mathrm{poly}(n)$ *with exponents written in unary. Then*

$$|p(t_1,\ldots,t_k)| > \exp(-\mathrm{poly}(n))$$

(non-uniformly in the choice of $\{t_j\}$), assuming that the value is non-zero.

Proof. We first reduce to the case $k = 1$. The numbers $\{t_j\}$ all lie in some finite-degree field extension $K \supseteq \mathbb{Q}$. It is a theorem of Galois that every such field has a generator t. We thus obtain that each $t_j = p_j(t)$ is some rational polynomial in t, and by rescaling t, we can make each p_j an integer polynomial; these fixed polynomials can be composed with the polynomial p in the proposition. Thus, without loss of generality, we can take $k = 1$ and $t = t_1$.

Next we consider the case that $t = a/b \in \mathbb{Q}$ is rational. In this it is enough for p to have degree $\mathrm{poly}(n)$, because we immediately get

$$|p(t)| > b^{\deg p}.$$

In the general case, let d be the degree of the field K, and let $z = p(t)$. Then $z = z_1$ has a list of Galois conjugates $z_1, z_2, \ldots, z_d$. Moreover, if we choose some basis of the ring of integers of K, then t has rational coordinates $s_1, \ldots, s_d$, and we can write

$$\prod_{j=1}^{d} z_j = q(s_1, \ldots, s_d)$$

for a polynomial q with $\deg q = d(\deg p)$. Thus by the rational case we obtain

$$\left|\prod_{j=1}^{d} z_j\right| > \exp(-\mathrm{poly}(n)).$$

At the same time, because of the degree bound on p and because each coefficient of p is bounded by $\exp(\mathrm{poly}(n))$, we obtain

$$|z_j| < \exp(-\mathrm{poly}(n)).$$

By dividing through, we obtain

$$|z| = |z_1| > \exp(-\mathrm{poly}(n)).$$ □

It is important to compare PostBQP and PostBPP to three other complexity classes: $\mathsf{A_0PP}$, or one-sided almost wide PP, defined by Vyalyi [37]; SBP, or small-bounded probabilistic P [6]; and a quantum class that we will call SBQP. All three classes depend on a real-valued function $f(x)$ in FP (expressed in fixed-point arithmetic, say), where x is the input to the decision problem, and a constant $c > 1$. The classes SBP and SBQP are defined in the same way as the Alice-Bob definition of PostBPP and PostBQP, except with a different model for Bob. As in Proposition 2.9, Alice executes a randomized algorithm in the case of SBP and a quantum algorithm in the case of SBQP and has success probability a. Meanwhile Bob's value $b = f(x)$ is computed directly in FP, as a real number in fixed-point arithmetic. In both SBP and SBQP, the answer is "yes" when $a > cb$ and "no" when $b > ca$.

Finally, $\mathsf{A_0PP}$ is a non-quantum class that is closely related to PP and is defined similarly to SBP. Like SBP, a decision function $D \in \mathsf{A_0PP}$ has a function $b = f(x)$ which lies in FP, and a randomized algorithm whose success probability is a. When $D \in \mathsf{A_0PP}$, we require that

$$D(x) = \text{yes} \implies a > cb + \frac{1}{2} \qquad \text{and} \qquad D(x) = \text{no} \implies \frac{1}{2} \le a < b + \frac{1}{2},$$

which again is like SBP but has an extra $1/2$ term.

Lemma 2.12. *Without loss of generality, the function $f(x)$ in the definition of $\mathsf{A_0PP}$, SBP, SBQP can be taken to be $2^{-p(|x|)}$ for some p; and all values of the constant c are equivalent.*

Proof. The constant c is irrelevant by the usual technique of amplification by repeated trials. This is immediate in the case of SBP and SBQP. It is not very difficult in the case of $\mathsf{A_0PP}$, and was established by Vyalyi [37].

To argue that $f(x)$ can be set to $2^{-p(|x|)}$ (in the cases of SBP and SBQP), first choose p so that $f(x) > 2^{-p(|x|)}$. Then Alice can compute $f(x)$ and reduce her success probability by a factor of $2^{p(|x|)}f(x)$. The argument in the case of $\mathsf{A_0PP}$ is essentially the same and was also explained by Vyalyi [37]. □

Proposition 2.13. $\mathsf{SBQP} = \mathsf{A_0PP}$.

Proof. The proof is almost the same as Aaronson's proof that $\mathsf{PostBQP} = \mathsf{PP}$ [1, Thm. 3.4]. We can also define $\mathsf{A_0PP}$ as a counting class in which, for each certificate y of length n, the computation produces a value $f(y) = \pm 1$, and these values are summed to produce $A(x)$. For a decision problem $D \in \mathsf{A_0PP}$, we require that

$$D(x) = \text{yes} \implies A(x) > 2^n Cb \qquad \text{and} \qquad D(x) = \text{no} \implies 0 \le A(x) < 2^n b\,.$$

First, let $L \in \mathsf{SBQP}$ be computed by a quantum circuit that consists of Hadamard and Toffoli gates. It is convenient to change the counting model of $\mathsf{A_0PP}$ slightly to let the values be ± 1 or 0. Then we obtain an $\mathsf{A_0PP}$ algorithm by multilinear expansion of the effect of these gates on density matrices. The matrix entries of a Toffoli gate, in its effect on a density matrix, are 0 and 1; the corresponding matrix entries of a Hadamard gate are $\pm 1/2$. The final probability is given by a partial trace of the output density matrix, and is non-negative and exactly matches the criteria for $\mathsf{A_0PP}$.

Now let $L \in \mathsf{A_0PP}$ and let a be Alice's success probability in the $\mathsf{A_0PP}$ algorithm. We can again slightly re-express the counting model of $\mathsf{A_0PP}$ so that $f(y) \in \{0,1\}$ and its sum $A = A(x)$ is given by $A = 2^n a$.

Then, in the SBQP algorithm, we can quantum-compute the unitary map

$$U_f|y\rangle = |y, f(y)\rangle$$

where the value $f(y)$ is written to an ancilla qubit. We provide the input $|{+}{+}\cdots{+}\rangle$ to U_f, and then postselect on whether the left n qubits of the result are all $|+\rangle$. If they are, then the ancilla qubit has the state

$$|\psi\rangle \propto (1-a)|0\rangle + a|1\rangle\,.$$

If this qubit is measured in the $\pm$ basis, then the probability of $|-\rangle$ is

$$a' = \frac{(2a-1)^2}{1+(2a-1)^2}\,.$$

If we assume that $b > 1/4$ and let $c = 2$ in the $\mathsf{A_0PP}$ algorithm, then

$$0 < a < \frac{1}{2} + b \implies a' < \frac{4b^2}{1+4b^2} < 4b^2 \qquad \text{and}$$
$$a > \frac{1}{2} + 2b \implies a' > \frac{16b^2}{1+16b^2} > 8b^2.$$

So we can let $b' = 4b^2$ and $c' = 2$ in an SBQP algorithm that produces the probability a'. □

Many of the complexity classes discussed here employ the semantic condition that the probabilities of particular outcomes are above one threshold or below another threshold. We can also consider promise versions of these classes in which these conditions hold for some inputs and not others. When they are considered in promise form, SBP- and SBQP-hardness are the same as PostBPP- and PostBQP-hardness. The non-trivial part of this equality (given that $\mathsf{SBP} \subseteq \mathsf{PostBPP}$ and $\mathsf{SBQP} \subseteq \mathsf{PostBQP}$) is the following inclusions:

Proposition 2.14. $\mathsf{PromisePostBPP} \subseteq \mathsf{P}^{\mathsf{PromiseSBP}}$ *and* $\mathsf{PromisePostBQP} \subseteq \mathsf{P}^{\mathsf{PromiseSBQP}}$.

Proof. Suppose that $D \in \mathsf{PromisePostBQP}$ is a decision function and that it is implemented by a quantum circuit. We recall the assumption that

$$\max(a,b) > 2^{-n}$$

where $n = \mathrm{poly}(|x|)$ and x is the input.

The construction is then similar to a rescaling argument in Aaronson's proof that $\mathsf{PostBQP} = \mathsf{PP}$ (explained in [1, Thm. 3.4] in the second half of the main proof). We assume that either $a > 8b$ or that $b > 8a$. Then for each $0 \leq k \leq n$, use PromiseSBQP to compare both a and b to 2^{-k}. If $a > 8b$, then for every k, PromiseSBQP will either reliably report that $a > 2^{-k}$ or that $2^{-k} > b$, and there will exist a k for which it will do both. Meanwhile if $b > 8a$, it will report that $b > 2^{-k}$ or that $2^{-k} > a$, and both for at least one k. These two outcomes are mutually exclusive.

The argument that $\mathsf{PromisePostBPP} \subseteq \mathsf{P}^{\mathsf{PromiseSBP}}$ is the same, but simpler since the lower bound on $\max(a,b)$ is immediate. □

Finally, as noted by Aaronson, linear computation is another interesting interpretation of PostBQP. (This is linear computation in the sense of non-unitary quantum computation, not $\mathbb{Z}/2$-linear circuits or numerical linear algebra!) Post-conditioning allows us to replace unitary gates by subunitary gates, and to rescale subunitary gates arbitrarily. But every linear operator that acts on vector states $|\psi\rangle$ is proportional to a subunitary operator. Thus, PostBQP can also be defined by the class of polynomial-sized circuits with linear gates, without the unitary restriction.

At first glance, the measurement probability (2.1) used for PostBQP still use the Hilbert space structure, even if the gates do not. But this is not entirely true either. If circuits are evaluated in a form such as $\langle 0^n|C|0^n\rangle$, and if the gates need not be unitary, then there is no need to equate the vector $|0\rangle$ with the dual vector $\langle 0|$ using a Hermitian form. We can instead define $\langle 0|$ and $\langle 1|$ to be the dual basis to $|0\rangle$ and $|1\rangle$. The drawback to this computational model is that it does not have a reasonable notion of a mixed state, nor partial trace that makes mixed states from pure states. We may define $\langle 0|$ using both $|0\rangle$ and $|1\rangle$ (using the relations $\langle 0|0\rangle = 1$ and $\langle 0|1\rangle = 0$), but we cannot in general define $\langle\psi|$ or $|\psi\rangle\langle\psi|$ from $|\psi\rangle$.

Indeed, we can more cleanly define linear computation as computation with *libits* (linear bits). By definition, a libit is like a qubit in the sense that it is assigned a 2-dimensional complex state space V. But unlike a qubit, V is just a vector space with no Hilbert space structure, so that there isn't even any way to say whether linear gates acting on libits are unitary. A libit has kets, which are vectors $|\psi\rangle \in V$, and it has bras, which are dual vectors $\langle\psi| \in V^*$. But V and V^* are simply different vector spaces.

3 The Jones polynomial

In this section we review the definition of the Jones polynomial and some theorems about it that lead to a proof of Theorem 1.2. We will define the Jones polynomial using the Kauffman bracket formalism, which in our opinion is one of the simplest and nicest definitions. For background see Kauffman [24]; also previous work by the author [27, §2] has a review of properties of the Kauffman bracket renamed as the "A_1 spider."

3.1 The Kauffman bracket

Let $t^{1/4} \in \mathbb{C}^\times$ be a non-zero complex number. (The reason for this notation is that all of the essential mathematics of the Jones polynomial depends only t, even though it is convenient to choose a fourth root $t^{1/4}$ to define it.) Then the Kauffman bracket is defined as a function on links projections, or *link diagrams*, by the following recursive relations:

$$\begin{aligned} \left|\times\right\rangle_K &= -t^{1/4}\left|)\,(\right\rangle_K - t^{-1/4}\left|\asymp\right\rangle_K, \quad \text{and} \\ \left\langle \bigcirc \right\rangle_K &= -t^{1/2} - t^{-1/2}. \end{aligned} \tag{3.1}$$

Relations of this type are called *skein relations*. (Kauffman writes (3.1) with a bracket $\langle\cdot\rangle$ for all terms, but a "ket" is more consistent with standard quantum notation; see Section 3.2.) What the relations mean is that if three link diagrams L_1, L_2, and L_3 are identical except that they differ in one place as indicated, then their Kauffman bracket values satisfy the given linear relation:

$$\langle L_1\rangle_K = -t^{1/4}\langle L_2\rangle_K - t^{-1/4}\langle L_3\rangle_K\,.$$

The second equation says that if L_1 and L_2 are two link diagrams that are the same except that L_1 has an extra circle, then

$$\langle L_1\rangle_K = -(t^{1/2}+t^{-1/2})\langle L_2\rangle_K\,.$$

The base of the recursion is given by saying that the Kauffman bracket of the empty link diagram is 1. With this normalization, the Jones polynomial is given by

$$V(L,t) = \frac{\langle L\rangle_K}{-(t^{1/2}+t^{-1/2})t^{3w/4}}$$

where w is the writhe of the diagram L, i. e., the number of positive crossings minus the number of negative crossings. It is a remarkable fact, although it is not difficult to check, that the Kauffman bracket is invariant under the second and third Reidemeister moves [9, §1.C], and that the Jones polynomial is invariant under all three Reidemeister moves and is therefore a link invariant.

3.2 Skein spaces

The importance of the skein relations is that they can be extend the Kauffman bracket to a "Kauffman ket" for tangles. Here a *tangle* is an incomplete link, i. e., the intersection of a link and a ball whose boundary is transverse to the link. By definition, the Kauffman ket of a tangle is a vector in a corresponding *skein space*; actually the skein space itself is defined from the tangles. More precisely, given a 3-dimensional ball with $2n$ marked points, let $F(2n)$ be the formal vector space of linear combinations of all tangles that end at the marked points. Then the skein space $W(2n) = F(2n)/\sim$ is by definition the quotient of the vector space $F(2n)$ by the relations (3.1). Any element of $W(2n)$, i. e., any linear combination of tangles modulo the skein relations, is called a *skein*. In this construction, then, the Kauffman bracket $|T\rangle$ of a tangle T is "itself," i. e., the skein that it represents. If $W(2n)$ is a skein space of tangles with $2n$ endpoints, then the Kauffman relations imply that

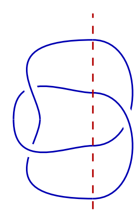
$$\dim W(2n) = C_n = \frac{1}{n+1}\binom{2n}{n}, \tag{3.2}$$

the nth Catalan number, because the planar matchings of the $2n$ endpoints are a basis of the skein space.

When the parameter t is a root of unity, it is more important to look at a certain reduced skein space $X(2n)$. First, we take an explicit model of $W(2n)$ as the skein space of tangles in the right half-plane with end points at the integers $1, 2, \ldots, 2n$ on the vertical number line. Then there is another skein space $W'(2n)$ consisting of tangles in the left half plane and with the same boundary. ($W'(2n)$ is of course equivalent to $W(2n)$, but in more than one way: by reflection, by rotation by 180 degrees, etc.) Then there is a bilinear pairing

$$\langle\cdot,\cdot\rangle_K : W(2n) \times W'(2n) \to \mathbb{C}$$

given by gluing together one tangle on each side and evaluating the Kauffman bracket. For example:

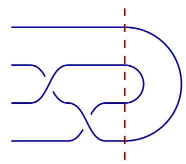
$W'(2n) \ni$  $\in W(2n)$.

Finally,

$$X(2n) \stackrel{\text{def}}{=} W(2n)/(\ker\langle\cdot,\cdot\rangle_K)\,.$$

It is known that $\langle\cdot,\cdot\rangle_K$ is degenerate on $W(2n)$ if and only if t is a root of unity of order $r > 1$ and $n \geq r-1$. Moreover, if $|t| = 1$, then there is a conjugate-linear isomorphism between $W(2n)$ and $W'(2n)$ given by reflecting the tangle across the horizontal line. (The reflection reverses crossings, so we need $|t| = 1$ in order to have $t^* = t^{-1}$ and thus have conjugate linearity.) Thus, if $|t| = 1$, then $\langle\cdot,\cdot\rangle_K$ is a non-degenerate Hermitian form on the quotient space $X(2n)$. It is further known that $\langle\cdot,\cdot\rangle_K$ is positive definite if $t = \exp(2\pi i/r)$ is a principal root of unity. Thus, if t is a principal root of unity, $X(2n)$ is a finite-dimensional Hilbert space, so it and the Jones polynomial become relevant to quantum computation. (See Section 3.3 for references and further explanation.)

The skein spaces $W(2n)$ and $X(2n)$ have an action of the braid group B_{2n} on $2n$ strands. The action is given by attaching the braid to a tangle or skein to make a new tangle or skein:



This is the *Jones braid representation* on $X(2n)$ [13]. In key cases $X(2n)$ is a Hilbert space and the braid representation is unitary (Section 3.3).

A variation of this theme is that if $\sigma \in B_n$ is a braid on n strands, we can simply expand it as a skein in $W(2n)$, with n endpoints on the left and on the right. (Or in $X(2n)$, but for the moment $W(2n)$ is more relevant.) We can also concatenate two elements of $W(2n)$ in the same way that braids are multiplied. I. e., having segregated the $2n$ endpoints into n each on the left and right, we can define a bilinear product map

$$m : W(2n) \times W(2n) \longrightarrow W(2n)$$

where $m(s,t)$ is given by attaching the right endpoints of $s \in W(2n)$ to the left endpoints of $t \in W(2n)$. This makes $W(2n)$ into an associative algebra called the Temperley-Lieb algebra [5]. The Jones braid representation generalizes to a representation

$$\rho : W(4n) \times X(2n) \longrightarrow X(2n)$$

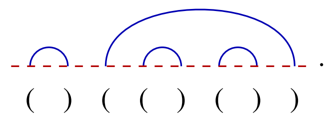
of the Temperley-Lieb algebra $W(4n)$, given by attaching $s \in W(4n)$ to $t \in X(2n)$ along half of the endpoints of the former and all of the endpoints of the latter.

3.3 Other models of skein spaces

There are many ways to define the skein space $W(2n)$ and the reduced skein space $X(2n)$, and the braid group action on them. One of the most important models is that, when t is not a root of unity, $W(2n)$ is the invariant subspace $\mathrm{Inv}(V^{\otimes 2n})$ of the representation $V^{\otimes 2n}$ of the quantum group $U_{\sqrt{t}}(\mathfrak{sl}(2))$, where V is the standard 2-dimensional irreducible representation [23]. This model is well-known to be the equivalent to the Kauffman skein space that we use here [14]. Moreover, it is well-known that as t approaches a principal root of unity, the pairing $\langle \cdot,\cdot \rangle_K$ on $W(2n)$ undergoes a degeneration, that the reduced skein space $X(2n)$ is a Hilbert space, and that the associated braid representation is unitary [26, 38]. In fact, all of these facts are part of a larger theory for all quantum groups $U_{\sqrt{t}}(\mathfrak{g})$ for any simple Lie algebra $\mathfrak{g}$. Unfortunately, it is not practical to give a summarize the theory of quantum groups here.

Since Aharonov, Jones, and Landau [5] use the so-called path model, we want to relate our planar matchings model to that one. In any case, the path model helps to compute the dimension of $X(2n)$, and it yields one proof that it is a Hilbert space. The rest of this section is a summary of calculations based on more advanced points of the Kauffman skein theory [25]. We do not include complete proofs. The results are not needed for our results, other than the one standard fact that $X(2n)$ is a Hilbert space when t is a principal root of unity.

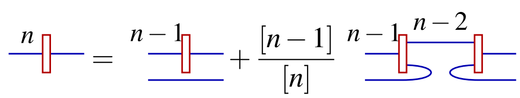
Model $W(2n)$ with planar matchings in the upper half plane. These are equivalent to balanced strings of parentheses, by matching the parentheses:



Then, a balanced string of parentheses of length $2n$ is equivalent to a path from 0 to 0 in the non-negative integers $\mathbb{Z}_{\geq 0}$, given by stepping to the right at each left parenthesis and to the left at each right parenthesis.

It is known that the planar matchings corresponding to the paths that lie in the discrete interval $\{0, 1, \ldots, r-2\}$ are a basis of $X(2n)$, when t is an rth root of unity with $r > 1$. Call these the *admissible* matchings. They are not an orthogonal basis, but their Gram-Schmidt orthogonalization in a natural partial ordering is the path basis used in [5]. (In other words, the admissible matchings are those whose parentheses do not nest beyond a depth of $r-2$.) The partial ordering can be expressed as a relation on paths, that $p \succeq q$ if the path p never crosses to the right of q.

In order to argue these facts, one employs a special skein with $2n$ endpoints called a *Jones-Wenzl projector*, which is given by the following recurrence relation

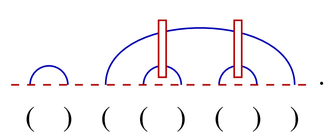


and the rule that the projector of order 1 is a plain strand. Here a strand labeled with n means n strands, and $[n]$ is a *quantum integer* defined by the formula

$$[n] = \frac{t^{n/2} - t^{-n/2}}{t^{1/2} - t^{-1/2}}.$$

The Jones-Wenzl projector exists for all n when t is not a root of unity or $t = 1$, and it exists when $n < r$ when t is a root of unity of order $r > 1$. Also, the projector of order $r-1$ vanishes in $X(2r-2)$. When working with reduced skein spaces $X(2k)$, we can assume, as a new skein relation, that the projector of order $r-1$ vanishes. This new skein relation allows us to express a planar matching whose path reaches $r-1$ in terms of earlier planar matchings. Thus, we can conclude that the admissible matchings are a spanning set of $X(2n)$, and we can ignore the inadmissible matchings.

Then, we can modify a planar matching by inserting a vertical projector between every pair of endpoints:



(The projectors of order 0 and 1 can be omitted, since they are trivial.) Call a skein of this form a *path vector*. By expanding the projectors, one can show that path vectors are related to admissible planar matchings by a triangular matrix. Since admissible matchings span $X(2n)$, so do the path vectors; and if the path vectors are linearly independent in $X(2n)$, so are admissible matchings.

The path vectors, as vectors in $W(2n)$ and $W'(2n)$, have a Gram matrix using the bilinear form on these two spaces. It is not hard to check, using various properties of Jones-Wenzl projectors, that this Gram matrix is diagonal and that the diagonal entries are non-zero. Thus, the path vectors are a basis of $X(2n)$. When t is a principal root of unity, the diagonal entries are also positive real numbers, which implies that $X(2n)$ is a Hilbert space. Finally, the triangular change of basis from admissible matchings to path vectors shows that the latter are the Gram-Schmidt orthogonalization of the former.

3.4 Quantum computation with braids

The idea, first explained by Freedman, Larsen, and Wang [12] is that when t is a principal root of unity, the Hilbert space $X(2n)$ can be interpreted as a quantum memory, and a braid $\sigma \in B_{2n}$ can be interpreted as a quantum circuit. The question then is whether such a model is universal for quantum computation. The well-known answer is yes when t is a non-lattice, principal root of unity, and the main technical tool is the following theorem.

Theorem 3.1 (Freedman, Larsen, Wang [13]). *Let $t = \exp(2\pi i/r)$ with $r = 5$ or $r \geq 7$. Then Jones braid representation of B_{2n} is dense in $\mathrm{PSU}(X(2n))$ for $n \geq 2$, or for $n \geq 3$ in the case $r = 10$.*

Corollary 3.2. *Let $t = \exp(2\pi i/r)$ with $r = 5$ or $r \geq 7$. Let*

$$p(x) > 2^{-\mathrm{poly}(|x|)}$$

be the probability that some polynomial-time quantum algorithm accepts an input x. Then the input x can be encoded as a link $L = L(x)$ with bridge number g, so that

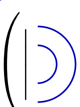
$$p(x) \approx \frac{|\langle L\rangle_K|^2}{|t^{1/2}+t^{-1/2}|^{2g}}\,, \tag{3.3}$$

where "$\approx$" is in the FPTEAS *sense.*

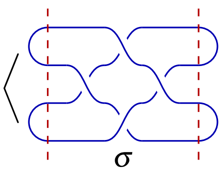
Although Corollary 3.2 is essentially due to Freedman, Larsen, and Wang, we describe one way to prove it, since it is relevant to our result.

Proof of Corollary 3.2. First, $X(4)$ is always two-dimensional and it can be interpreted as a qubit. We can define its computational basis simply by applying the Gram-Schmidt procedure to the basis of planar matchings:

$|0\rangle = \dfrac{1}{t^{1/2}+t^{-1/2}}\Big|$  $\Big\rangle_K,$ (3.4)

$|1\rangle = \dfrac{1}{\sqrt{t+1+t^{-1}}}\Bigg(\Big|$  $\Big\rangle_K + \dfrac{1}{t^{1/2}+t^{-1/2}}\Big|$  $\Big\rangle_K\Bigg).$

Second, by Theorem 3.1 and Theorem 2.4, a quantum circuit C on n qubits can be encoded to exponential tolerance as a braid $\sigma \in B_{4n}$ on $4n$ strands. Third, the amplitude $\langle 0^n|C|0^n\rangle$ is proportional to the Kauffman bracket of a link L, which is the braid σ capped with $2n$ U-turns at both ends:

$$\langle 0^n|C|0^n\rangle \approx \frac{1}{(t^{1/2}+t^{-1/2})^{2n}} \Big\langle$$

$$\Big\rangle_K . \tag{3.5}$$

A diagram of a link L in this form, a braid capped with U-turns, is called a *plat diagram*; the number of U-turns at each end, $g = 2n$ in this case, is its *bridge number*. Finally, by Equation (2.1), we can express the acceptance probability as $|\langle 0^n|C|0^n\rangle|^2$ where C has $n = \mathrm{poly}(|x|)$ qubits and $\mathrm{poly}(|x|)$ gates and can be generated in deterministic polynomial time from x. Combining Equations (3.5) and (2.1), we obtain (3.3), as desired. □

Remark 3.3. In the proof of Corollary 3.2, it is easy to worry about leakage of amplitude into the unused part of the Hilbert space $X(4n)$. But using the plat diagram method, Theorem 3.1 and Theorem 2.4 applied to the unitary group $\mathrm{PSU}(X(8)) \cong \mathrm{PSU}(14)$ controls this leakage along with the intended amplitudes. In some other encodings of quantum computation into the Jones polynomial, one might want a joint denseness version of Theorem 3.1. It isn't needed here, although it is needed in order to prove Theorem 3.1 itself by induction.

3.5 Proof of Theorem 1.2

Proof. Corollary 3.2 describes a way to approximately (FPTEAS) encode a circuit calculation $\langle 0^n|C|0^n\rangle$ as a plat braid with bridge number $2g$. This type of circuit calculation is BQP-complete by Proposition 2.3. Each gate of the circuit C (say a Toffoli or a Hadamard gate, if these standard generators are used) can be approximated by a braid by Theorem 3.1 (Freedman-Larsen-Wang) and Theorem 2.4 (Solovay-Kitaev). Thus the left side of (3.3) is BQP-complete in additive approximation. But the denominator is exponential in g. This is not by itself a hardness result, but it is a strong indication that Theorem 1.1 does not usually provide information about the Jones polynomial, and that a hardness result should be available.

The first hardness result to obtain is that multiplicative approximation to the Jones polynomial norm $|V(L,t)|$ is #P-hard. Almost by definition (more precisely, by Proposition 2.3), multiplicative approximation to the left side is SBQP-hard, which by Proposition 2.14 is the same as PostBQP-hard. The denominator on the right side is easily computable, so we obtain that multiplicative approximation to the numerator is also PostBQP-hard. This numerator is the Kauffman bracket value $|\langle L\rangle_K|^2$, which equals $|V(L,t)|^2$, which implies hardness of $|V(L,t)|$. Finally, Aaronson's theorem tells us that $\mathsf{PostBQP} = \mathsf{PP}$, and PP-hard implies #P-hard by Proposition 2.1.

To complete the proof, we need to refine the construction in two ways. We need to convert multiplicative approximation to more general value-distinguishing approximation; and we need to change the link L to a knot.

For the first refinement, let $a > b > 0$ be constants as in the statement of Theorem 1.2, and let p and c be the polynomial and the constant in the modified definition of SBQP in Lemma 2.12. By that lemma and Equation (3.3), it is SBQP-complete and therefore #P-hard to determine whether

$$\frac{|\langle L\rangle_K|^2}{|t^{1/2}+t^{-1/2}|^{2g}} \begin{cases} > c2^{-p(|x|)} \\ < 2^{-p(|x|)} \end{cases} .$$

We want to make a modified link L' to make it hard to determine whether $|\langle L'\rangle_K|^2$ is more than a or less than b. Recall that $g = \mathrm{poly}(|x|)$, and note that

$$|t^{1/2}+t^{-1/2}| > 1 .$$

If

$$|t^{1/2}+t^{-1/2}|^{2g} \ll 2^{p(|x|)}$$

when $|x|$ is large, then we can add $m = \mathrm{poly}(|x|)$ copies of the unknot to L so that

$$|t^{1/2}+t^{-1/2}|^{2g+2m}2^{-p(|x|)}$$

is bounded. On the other hand, if

$$|t^{1/2}+t^{-1/2}|^{2g} \gg 2^{p(|x|)}$$

then we can use denseness to first create a link L_0 (say a 2-bridge link corresponding to a 1-qubit circuit) such that $|\langle L_0\rangle_K|$ is a small constant. Then we can add m copies of L_0 to L so that

$$|\langle L_0\rangle_K|^{2m}|t^{1/2}+t^{-1/2}|^{2g}2^{-p(|x|)}$$

is bounded. The constant c in the definition of SBQP can be chosen to overwhelm the bound in either case as well as the specific values of a and b.

Finally, we want to further modify L' into a link L'' that has only one component, i. e., is a knot. The trick for this is that since the braid group is dense, the pure braid group is also dense. Thus we can switch two strands, and then approximately cancel its effect with a pure braid that does not permute any strands. The permutation induced by the braid is thus decoupled from the approximate value of $\langle L''\rangle_K$, so L'' can be chosen so that it has only one component. □

4 The Tutte polynomial

4.1 Tutte and Potts

In order to define the Tutte polynomial, we will first define another graph invariant with equivalent information known as the Potts model. The Potts model of a graph G depends on a positive integer q, the number of colors; and on a variable y. The weight of a coloring of the vertices of G with q colors is defined as y^k if k of the edges of G connect two vertices of the same color. Then the Potts partition function $Z(G,y,q)$ is defined as the total weight of all vertex colorings. The Potts partition function yields the Tutte polynomial $T(G,x,y)$ by the formula

$$T(G,x,y) \stackrel{\text{def}}{=} (y-1)^{-v}(x-1)^{-c}Z(G,y,q)$$

where

$$q = (x-1)(y-1)\,, \tag{4.1}$$

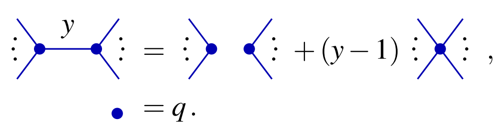
and G has v vertices and c components.

An important variation of the Potts model (or the Tutte polynomial) is the multivariate version, where the weight y can be different for each edge of G, to make a weighted graph $G(\vec{y})$. Then the Potts partition function is defined in the usual way as a multiplicative sum. Namely, the partition function $Z(G(\vec{y}),q)$ is defined as the total weight of all colorings c with n colors; the weight of c is defined as the product of the weights y_e for edges e whose vertices have the same color. Or, as a formula, if $\mathcal{C}$ is the set of colorings and E is the set of edges of G, then

$$Z(G(\vec{y}),q) = \sum_{c\in\mathcal{C}} \prod_{\substack{(j,k)\in E\\ c(j)=c(k)}} y_{(j,k)}\,.$$

Having generalized the parameter y to a weight assigned to each edge, we still want to make use of the parameter x defined from y and q by the relation (4.1). To this end, if we assign a weight y to an edge, we will also assign it the *dual weight* x using (4.1). The dual weight x is simply meant as another notation for the weight y. Since the dual weight x is not the same number as the weight y, we will denote it in the diagrams with parentheses.

The ordinary or multivariate Potts model can also be defined by a contraction-deletion formula, together with the fact that its value for an isolated vertex is q:



and (4.2)

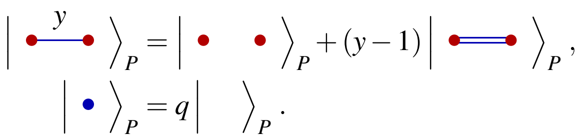
(Tutte's original definition of the Tutte polynomial uses an equivalent contraction-deletion formula.) This second definition is important for two reasons.

First, it shows that the Potts partition function $Z(G,y,q)$ or $Z(G(\vec{y}),q)$ is a polynomial in all of its parameters; it isn't only defined when q is a positive integer. Note that we can only give the Tutte polynomial or the Potts model a complexity if each parameter such as q or y has a computational complexity. To this end, we assume that every parameter is a real number with an FPTEAS. For no essential reason, we do not consider complex values.

Second, the contraction-deletion formula allows us to generalize the Potts model to a skein theory with skein spaces, in the same sense as Section 3.2. More precisely, for each n we let $F(n)_P$ be the vector space of formal linear combinations of weighted planar graphs with n marked boundary points on the outside face. In fact, we would like to allow some of the marked boundary points to be identical, so formally we consider a graph $G(\vec{y})$ together with a function from labels to vertices,

$$f : \{1,\ldots,n\} \to V(G(\vec{y}))$$

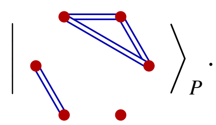
which need not be either injective or surjective. In the diagrams we draw the boundary vertices in red. If a vertex is marked twice or more as a boundary, then it is drawn as multiple vertices connected by double edges to denote that the vertices are equal. Thus (4.2) can be written as follows, also using the ket notation to signify that we are creating a skein theory:



and (4.3)

We then define the skein space to be the quotient $W(n)_P = F(n)_P/\sim$, where the equivalence is given by the relation (4.2).

To review, we have used (4.2) to define skein spaces $W(n)_P$ for *planar* graphs. It is easy to show that one basis of $W(n)_P$ is given by noncrossing partitions of n points arranged in a circle, corresponding to graphs with no edges (other than double edges):



It is well known that the number of noncrossing partitions is the nth Catalan number, so that

$$\dim W(n)_P = C_n = \frac{1}{n+1}\binom{2n}{n}$$

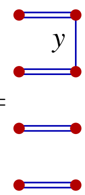
which is the same as the dimension of the Kauffman skein space $W(2n)_K$ as given in (3.2). In fact, the two skein theories are equivalent, and we will make use of this coincidence to prove Theorem 1.3.

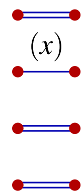
Remark 4.1. What matters the most for a result such as Theorem 1.3 is that the Potts model has *some* skein theory. Although the terminology "skein theory" is not traditional in graph theory, graph theorists have long used the idea of a skein theory, namely local recurrence relations such as the contraction-deletion formula. In particular, if we let $\widetilde{W}(n)_P$ be the skein space of all graphs with n boundary vertices, not just planar graphs, then it is a standard graph theory fact that one basis for it is the set of partitions of n points. The dimension of this skein space is the nth Bell number (by definition, the number of partitions of a set with n elements) rather than the nth Catalan number in the planar case.

4.2 Circuits and braids

In this section, we will define Potts quantum circuits by analogy with the Jones braid representation and its use in the proof of Corollary 3.2. In particular, we will encode the standard quantum circuit evaluation $\langle 0^n|C|0^n\rangle$ in Potts circuit by analogy with (3.5). Just as we did in Section 3.2, we define $W(n)_P$ using graphs in the right half-plane and we denote elements as kets $|\psi\rangle$; we define $W'(n)_P$ using graphs in the left half-plane and we denote its elements as bras $\langle\psi|$. However, we will not define any Hilbert space structures on our skein spaces. Instead, we will just use vector spaces and interpret them using the libit or linear computation model defined at the end of Section 2.5. For concreteness, we define the initial state $|\psi\rangle \in W(n)_P$ to be n disconnected dots, and the final state $\langle\psi| \in W'(n)_P$ to also be n disconnected dots.

Having defined initial and final states for Potts circuits, we still need to define the circuits themselves. We could define a Potts circuits to be any planar graph with left and right boundary vertices. This is the more general possible choice; but we will define more specific quantum gate operators $P(y)$, the parallel gate, and $S(x)$, the series gate. A gate $P(y)$ is an edge with weight y, whose two vertices are both input vertices and output vertices. A gate $S(x)$ is an edge with dual weight x that connects an input vertex to an output vertex. If there are n vertices, then there are $n-1$ positions for $P(y)$ and n positions for the gate $S(x)$; we number them $P(y)_j$ and $S(x)_j$ starting with $j=1$. For example, if $n=4$, then:

$P(y)_1 =$ 

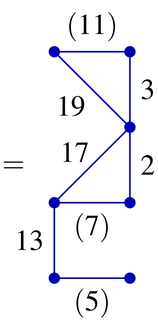
and $S(x)_2 =$ 

.

As an example of the full circuit construction, if $n=4$, we can make a graph G composed of 8 gates so that

$$Z(G(\vec{y});q) = \langle\psi|P(19)_1P(17)_2P(13)_3S(11)_1S(7)_2S(5)_4P(3)_1P(2)_2|\psi\rangle\,.$$

In this example, the graph $G(\vec{y})$ is:

$G(\vec{y}) =$ 

.

To conclude this section, we show that for certain values of the parameters x and y, the gates $P(y)$ and $S(x)$ aren't just analogous to the Jones braid representation; up to scalar factors, they are the Jones braid representation.

Theorem 4.2. *Let q and $t^{1/4}$ be parameters such that*

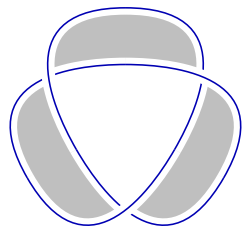
$$q = t + 2 + t^{-1}\,.$$

Then for each n, there is a vector space isomorphism between the planar Potts skein space $W(n)_P$ and the Kauffman skein space $W(2n)_K$ such that the operators $P(-t)$ and $S(-t)$ are proportional to half-twist generators of the Jones braid representation.

Note that $q > 4$ in Theorem 1.3, the corresponding value of t is real and positive in Theorem 4.2, and we can also take $t^{1/4}$ to be real and positive. Thus, in our use of Theorem 4.2, we can do all calculations over the field $\mathbb{R}$.

Theorem 4.2 and its proof are a version of one of the earliest constructions of the Jones polynomial of a link L, as the Potts partition function of an associated graph $G(\vec{y})$ [21, §2]. First, the diagram of L

should be given a checkerboard coloring:



.

Then we can make a weighted graph $G(\vec{y})$ by replacing the gray regions by vertices, and the crossings by edges. There are two types of crossings, checkerboard-positive and checkerboard-negative, and they can be replaced by edges with weight $y = -t^{\pm 1}$ (and therefore dual weight $x = -t^{\mp 1}$):

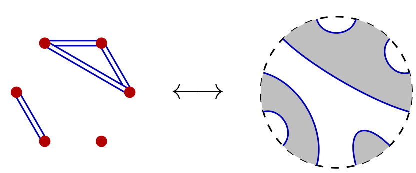
checkerboard positive   checkerboard negative

It turns out that

$$\langle L\rangle_K = t^{u/4}(-t^{1/2} - t^{-1/2})^{-v} Z(G(\vec{y}), q)$$

where u is the number of checkerboard-positive crossings minus the number of checkerboard-negative crossings, and v is the number of black regions of L.

Proof of Theorem 4.2. There is an evident bijection between non-crossing partitions of n points and planar matchings of $2n$ points. Each part of the partition is represented by a polygon with some k sides, and we can replace it by k arcs:



We will use the same symbol m to denote either the partition or its corresponding matching. The vectors $|m\rangle_P$ are a basis of $W(n)_P$, while the vectors $|m\rangle_K$ are a basis of $W(2n)_K$. We identify them using the formula

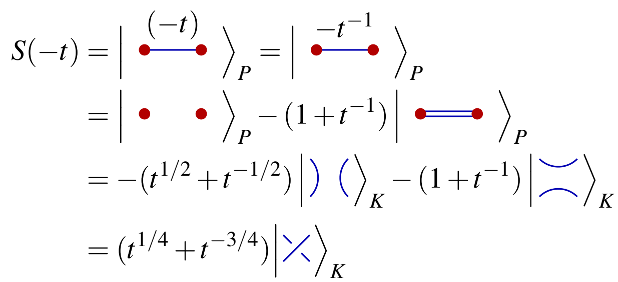
$$|m\rangle_P = (-t^{1/2} - t^{-1/2})^{c(m)} |m\rangle_K$$

where $c(m)$ is the number of components of m as a partition, or the number of black regions of m read as a planar matching.

With this choice of isomorphism, we claim that if R_j is the jth left half-twist operator on $W(n)_K$ in the Jones braid representation, then

$$S(-t)_j = (t^{1/4} + t^{-3/4}) R_{2j-1}, \qquad \text{and}$$
$$P(-t)_j = -t^{1/4} R_{2j}.$$

The first of these relations is established as follows. We do the calculation in terms of kets; the reader can check that it works the same way with operators. We obtain:



using (4.3) and (3.1). (The extra factor of $-t^{1/2}-t^{-1/2}$ in the first term arises from the change of basis from Potts skeins to Kauffman skeins.) The calculation for $P(-t)$ is similar.

Since the braid generators are proportional to the parallel and series operators, the latter generate the same projective representation. □

4.3 Parallel-series compositions

The statement of Theorem 1.3 only allows graphs with the same weight y for every edge. If we want to use the gates $P(y)$ and $S(x)$ universal quantum computation, this is not even enough for the Solovay-Kitaev theorem, if we don't have the inverses of these two gates. In this section we use a technique used by Goldberg and Jerrum in which edges are replaced by subgraph gadgets, to approximately allow any real weight y for any edge [15]. This will give let use the Solovay-Kitaev theorem by the relations

$$P(y)^{-1} = P(y^{-1}) \qquad \text{and} \qquad S(x)^{-1} \propto S(x^{-1})$$

which follow from (4.4) below. It will also make it easier to prove the dense generation criterion that is also needed for the Solovay-Kitaev theorem.

The technique is as follows: If a graph $G(\vec{y})$ has two parallel edges with weight y_1 and y_2, then they are equivalent to a single edge with weight y_1y_2. Meanwhile, if $G(\vec{y})$ has two edges in series with dual weight x_1 and x_2, they are equivalent (up to changing the Potts value Z by a constant factor) to one edge with weight x_1x_2. In other words,

$$P(y_1y_2) = P(y_1)P(y_2) \qquad \text{and} \qquad S(x_1x_2) \propto S(x_1)S(x_2)\,. \tag{4.4}$$

These transformations are called *shift operations*; they are also called *compositions* and *implemented weights*. Note that series and parallel compositions preserve the value of q, and they preserve planarity.

Lemma 4.3. *Consider graphs with the Potts model with q colors and with a single weight y which is an* FPTEAS *number. Suppose that $q > 4$ and that $x, y < 0$. Then all weights $y' \neq 1$ that are* FPTEAS *numbers, can be* FPTEAS *approximated by parallel and series compositions.*

Lemma 4.3 is a refinement of one proved by Goldberg and Jerrum [15]. (The refinement is that they did not establish is the FPTEAS property.)

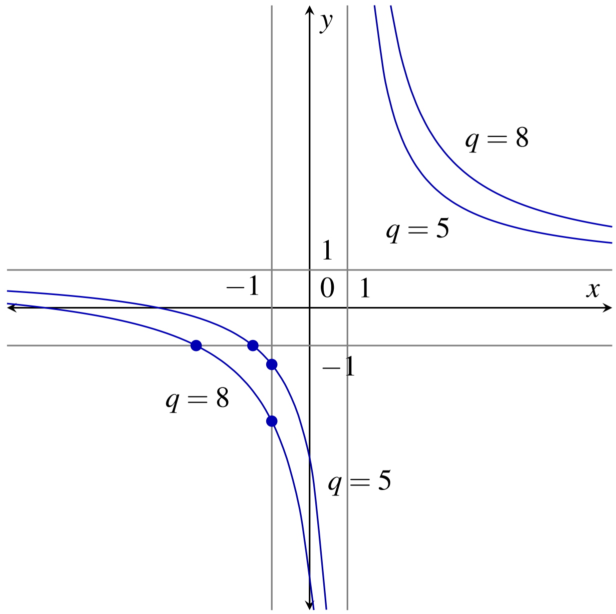


Figure 1: The Tutte plane with level curves of q.

Proof. Figure 1 shows a diagram of curves in the x-y plane (the Tutte plane) with constant values of q. Given that $q > 4$ and $x, y < 0$, we must have either that $x < -1$ or $y < -1$ or both. Parallel composition has the same effect on y as series composition has on x, and vice versa; so we can assume without loss of generality that $x < -1$. As a first step, we can create the dual weight x^n with a series composition with n edges. This creates a sequence of weights y_n that satisfies the estimate

$$\log(y_n) = qx^{-n}(1 + o(1))$$

as $n \to \infty$. Now suppose that $y' > 1$ is some other weight. We claim that we can efficiently approximate y' as a product of weights y_{2n}. Equivalently, we claim that we can efficiently approximate $\log(y')$ as a sum of terms $\log(y_{2n})$:

$$\log(y') = \log(y_{2n_1}) + \log(y_{2n_2}) + \cdots .$$

This can be viewed as a bin packing problem, because both $\log(y')$ and each term $\log(y_{2n})$ are positive. The claim is established by using a greedy bin-packing algorithm. I. e., choose each term $\log(y_{2n_k})$ to be as large as possible, but so that the partial sum does not exceed $\log(y')$. Since the terms $\log(y_{2n})$ decrease exponentially (and no faster), and since the graph complexity of each term is linear in n, the result is a parallel-series composition which is an FPTEAS for the weight y'.

The same bin-packing argument works for $0 < y' < 1$, using the odd-numbered weights y_{2n+1}. So every desired weight $y' > 0$ has an FPTEAS-strength parallel-series composition. In addition, we also have the original weight $y < 0$, so the values of $y' > 0$, $y' = y''y$ with $y'' > 0$, and y itself reach every desired value other than $y' = 0$. Since we also want the remaining weight $y' = 0$, we can at this point achieve its dual weight $x' = 1 - q$ with a series composition with the dual weights $x' = -1$ and $x' = q - 1$. $\square$

4.4 Densely generating $\mathrm{PSL}(W(n)_P)$

In this section, we will prove that if $q > 4$, then there are FPTEAS numbers x, y_1, and y_2, such that the gates $S(x)$, $P(y_1)$, and $P(y_2)$ and their inverses densely generate the group $\mathrm{PSL}(W(n)_P)$ for any $n \geq 2$. Lemma 4.3 says that we can obtain any such gates in FPTEAS approximation using subgraph gadgets. Our argument borrows from the author's previous work [28] and makes crucial use of the Zariski topology on the group $\mathrm{PSL}(W(n)_P)$.

The *Zariski topology* on an algebraic group (or any algebraic variety) is by definition the topology in which the closed sets are solutions to polynomial equations. The Zariski topology on $\mathbb{R}^n$ or on $\mathrm{PSL}(n,\mathbb{R})$ is much coarser than the standard topology, which in this context is called the *analytic topology*. It is easier for a subgroup or a subset to be Zariski dense, and it is easier to prove Zariski denseness in this algebraically adapted topology. In particular:

Theorem 4.4 ([28, Cor. 1.2])**.** *Let $n > 1$ be an integer and let $t > 1$ be real. Then the Jones braid representation of B_{2n} acting on $W(2n)_K = X(2n)_K$ with parameter t is Zariski dense in* $\mathrm{PSL}(X(2n))$*.*

On the other hand, in some circumstances we can get the best of both worlds:

Proposition 4.5 ([28, §3])**.** *A subgroup Γ of a connected, simple Lie group G is analytically dense if and only if it is both analytically indiscrete and Zariski dense.*

(Proposition 4.5 is a baby version of a more famous result known as the Zassenhaus neighborhood theorem [39, 22].)

To finish the construction, let $q = t + 2 + t^{-1}$, let $x = y_1 = -t$ and $y_2 = t^{\sqrt{2}}$. (The the only requirement is that y_2 should be an irrational power of t with an FPTEAS exponent.) Then the gates $P(y_1)$ and $P(y_2)$ generate an indiscrete group by (4.4); their products

$$P(-t)^a P(t^{\sqrt{2}})^b = P((-1)^a t^{a+\sqrt{2}b})$$

for all $a, b \in \mathbb{Z}$ are a dense subset of all $P(y)$. By Theorem 4.2, the gates $S(x)$ and $P(y_1)$ acting on $W(n)_P = W(2n)_K$ generate the Jones braid representation of B_{2n}. By Theorem 4.4, this group action is Zariski dense. With the addition of the gate $P(y_2)$, it is also indiscrete and therefore analytically dense by Proposition 4.5.

Remark 4.6. A self-contained proof of Theorem 1.3 would be simpler if we applied some of the techniques involved in Theorem 4.4 directly to the group generated by gates of the form $P(y)$ and $S(x)$. However, these techniques involve yet another set of mathematical tools that we prefer to relegate to [28].

4.5 Proof of Theorem 1.3

Proof. Following Corollary 3.2 and its proof, let

$$p(x) > 2^{-\mathrm{poly}(|x|)}$$

be the probability that some polynomial-time quantum algorithm accepts an input x. Then

$$p(x) = |\langle 0^n|C(x)|0^n\rangle|^2$$

for some a quantum circuit $C(x)$ that can be generated from x in (classical) polynomial time. We can use the 2-dimensional skein space $W(2)_P$ as a libit, and let $|0\rangle = |\psi\rangle$ be the state of two dots as in Section 4.2. By Lemma 4.3, we can approximate the gates $S(-t)$, $P(-t)$, and $P(t^{\sqrt{2}})$ and their inverses. By Section 4.4, these gates densely generate $\mathrm{PSL}(W(4)_P) \cong \mathrm{PSL}(14,\mathbb{R})$ in the case $n=4$. Then we can apply Solovay-Kitaev, Theorem 2.4, to approximately encode the gates of $C(x)$ as a circuit acting on $\mathrm{PSL}(W(2n)_P)$. Then we finalize the circuit with the states $\langle 0|$, which can also be defined as the state $\langle\psi|$ of two dots.

The result is a graph $G(x)$ such that the Potts value $Z(G(x),q,y)$ satisfies

$$p(x) \approx N(x)|Z(G(x),q,y)|^2$$

where the extra factor $N(x)$ is a polynomial-time computable normalization that depends on the construction of $G(x)$. (The factor of $N(x)$ appears because we are working up a scalar factor in all of our computations. Note also that the x here is the decision problem input and not the Potts parameter.) It follows that for every $c > 1$, multiplicative approximation of $Z(G(x),q,y)$ up to a factor of c is PostBQP-hard, and thus #P-hard. □

5 Final remarks and questions

5.1 Other properties of knots

Theorem 1.2 says that value-distinguishing approximation of certain values of the Jones polynomial are #P-hard even when the link L is taken to be a knot. We conjecture that L could in addition be a prime knot or even an atoroidal knot. (A *prime* knot is one which is not a composite of two knots; an *atoroidal* knot is one which is not a satellite [9, §2.C].) Maybe other such restrictions on the structure of L could be imposed. But without a result such as that distinguishing the unknot (say) is hard, it is not feasible to add arbitrary interesting topological restrictions on L to Theorem 1.2. Maybe recognizing the unknot is in P or BQP. The Jones polynomial would then be easy to compute for knots that are recognized as the unknot or recognized as some other specific knots.

In fact, recognizing the unknot is in NP [18], and in coNP assuming the generalized Riemann hypothesis [29]. Thus, unless the polynomial hierarchy collapses, recognizing the unknot has lower qualitative computational complexity than approximating the Jones polynomial. (But the Jones polynomial could still have competitive *quantitative* complexity, i. e., asymptotic time complexity in a realistic computational model.)

5.2 Other kinds of approximation

There are many other kinds of partial information about the Jones polynomial without any interesting complexity bound to our knowledge. Is the degree of the Jones polynomial intractable? Is it intractable to determine when some value of the Jones polynomial vanishes? What if the Jones polynomial is reduced mod p for some prime p?

5.3 Denseness may be more than necessary

It is easiest to see that a set of gates is universal for linear computation if they densely generate an appropriate Lie group. For instance, they might generate $\mathrm{PSL}(2^n,\mathbb{C})$ if they act on n libits, or $\mathrm{PSL}(2^n,\mathbb{R})$ inside it. But dense generation is more than necessary for certain types of universality. For example, k-libit gates with integer matrices always generate a discrete group, even when acting on $n > k$ libits. Nonetheless, both the Hadamard and Toffoli gates are proportional to integer gates, and they are universal for quantum computation. Thus, multiplicative approximation of amplitudes in linear computation with integer gates is #P-hard. We do not know the right criteria on linear gates to establish #P-hardness results.

5.4 Solovay-Kitaev without inverses

It is a long-standing open problem to generalize the celebrated Solovay-Kitaev theorem to gate sets that are not closed under inverses. This problem could be peripheral in the context of designing actual quantum computers or realistic quantum algorithms. However, it could be important for the purpose of establishing hardness results.

5.5 Morse algorithms may be optimal

It is common practice to compute the Jones polynomial by a strategy known variously as a Morse algorithm, dynamic programming, a scanline algorithm, or a divide-and-conquer algorithm. (Morse theory in geometric topology is a theory of analyzing a topological object by dividing it into horizontal slices.) For a knot in a plat diagram, the strategy is to numerically compute the action of the braid group on the skein space. This type of algorithm requires simple exponential time and space in the number of strands of the braid, or for other kinds of knot diagrams, the width of the diagram. This is much better than a direct recursive evaluation of the Jones polynomial using a finite set of skein relations; the time complexity of any such direct algorithm is instead exponential in the number of crossings.

It is natural to wonder whether there are other clever algorithms that can compute the Jones polynomial even faster. The proof of Theorem 1.2 could be evidence that Morse algorithms are essentially optimal for many kinds of knot diagrams. In short, if braids are evaluated using the Jones polynomial at the dense roots of unity of Theorem 1.2, then they are a model of general planar quantum circuits.

In more detail, consider a typical hard search problem based on classical circuits, and an analogous problem based on quantum circuits. For instance, let $(z,w) = C(x,y)$ be a reversible circuit whose input (x,y) and output (z,w) are each divided into two registers of equal length. Then it is NP-hard to determine whether there is a solution to $(z,0) = C(x,0)$. We conjecture that there are linear-depth, planar circuits C for which this problem requires exponential time in $|x|$, in other words that full cryptography can be achieved with linear depth, planar circuits.

Using denseness at a non-lattice root of unity and Solovay-Kitaev, Theorem 2.4, this circuit problem can be encoded in a braid with polynomial overhead. (Again, the Solovay-Kitaev theorem has poly-logarithmic overhead for BQP, but polynomial overhead for PostBQP.) We conjecture that this extra polynomial overhead is not essential for hardness. We have in mind that there could be cryptographic methods to make linear-depth plat diagrams of knots, for which the Jones polynomial requires exponential time in the bridge number g to estimate at a non-lattice root of unity. (Note that the depth of a braid is not the same as its length; to calculate the depth, commuting half-twists can be applied in parallel.) Such conjectures are very difficult to prove unconditionally, because they would imply that #P is not contained in FP. Nonetheless, if there were a believable theory of cryptography for the Jones representation of linear-depth braids, then one would also believe that Morse algorithms to compute or estimate the Jones polynomial are essentially optimal.

References

[1] SCOTT AARONSON: Quantum computing, postselection, and probabilistic polynomial time. *Proc. Royal Soc. A*, 461(2063):3473–3482, 2005. Preliminary versions in arXiv and ECCC. [doi:10.1098/rspa.2005.1546] 185, 187, 194, 198, 199

[2] LEONARD M. ADLEMAN, JONATHAN DEMARRAIS, AND MING-DEH A. HUANG: Quantum computability. *SIAM J. Comput.*, 26(5):1524–1540, 1997. [doi:10.1137/S0097539795293639] 189

[3] DORIT AHARONOV AND ITAI ARAD: The BQP-hardness of approximating the Jones polynomial. *New J. Phys.*, 13:035019, 2011. [doi:10.1088/1367-2630/13/3/035019, arXiv:quant-ph/0605181] 184

[4] DORIT AHARONOV, ITAI ARAD, ELAD EBAN, AND ZEPH LANDAU: Polynomial quantum algorithms for additive approximations of the Potts model and other points of the Tutte plane, 2007. [arXiv:quant-ph/0702008] 185, 189

[5] DORIT AHARONOV, VAUGHAN JONES, AND ZEPH LANDAU: A polynomial quantum algorithm for approximating the Jones polynomial. *Algorithmica*, 55(3):395–421, 2009. Preliminary version in STOC'06. [doi:10.1007/s00453-008-9168-0, arXiv:quant-ph/0511096] 184, 202, 203

[6] ELMAR BÖHLER, CHRISTIAN GLASSER, AND DANIEL MEISTER: Error-bounded probabilistic computations between MA and AM. *J. Comput. System Sci.*, 72(6):1043–1076, 2006. Preliminary version in MFCS'03. [doi:10.1016/j.jcss.2006.05.001] 197

[7] MAGNUS BORDEWICH, MICHAEL H. FREEDMAN, LÁSZLÓ LOVÁSZ, AND DOMINIC WELSH: Approximate counting and quantum computation. *Combin. Probab. Comput.*, 14(5-6):737–754, 2005. [doi:10.1017/S0963548305007005, arXiv:0908.2122] 184, 185

[8] RICHARD P. BRENT: Fast multiple-precision evaluation of elementary functions. *J. ACM*, 23(2):242–251, 1976. [doi:10.1145/321941.321944] 187

[9] GERHARD BURDE AND HEINER ZIESCHANG: *Knots*. Volume 5 of *De Gruyter Studies in Mathematics*. Walter de Gruyter, 1985. 184, 200, 214

[10] SAMUEL R. BUSS AND LOUISE HAY: On truth-table reducibility to SAT. *Inform. and Comput.*, 91(1):86–102, 1991. [doi:10.1016/0890-5401(91)90075-D] 195

[11] MICHAEL H. FREEDMAN, ALEXEI KITAEV, AND ZHENGHAN WANG: Simulation of topological field theories by quantum computers. *Comm. Math. Phys.*, 227(3):587–603, 2002. [doi:10.1007/s002200200635, arXiv:quant-ph/0001071] 184

[12] MICHAEL H. FREEDMAN, MICHAEL J. LARSEN, AND ZHENGHAN WANG: A modular functor which is universal for quantum computation. *Comm. Math. Phys.*, 227(3):605–622, 2002. [doi:10.1007/s002200200645, arXiv:quant-ph/0001108] 204

[13] MICHAEL H. FREEDMAN, MICHAEL J. LARSEN, AND ZHENGHAN WANG: The two-eigenvalue problem and density of Jones representation of braid groups. *Comm. Math. Phys.*, 228(1):177–199, 2002. [doi:10.1007/s002200200636, arXiv:math/0103200] 184, 202, 204

[14] IGOR FRENKEL AND MIKHAIL KHOVANOV: Canonical bases in tensor products and graphical calculus for $U_q(\mathfrak{sl}_2)$. *Duke Math. J.*, 87(3):409–480, 1995. [doi:10.1215/S0012-7094-97-08715-9] 202

[15] LESLIE ANN GOLDBERG AND MARK JERRUM: Inapproximability of the Tutte polynomial. *Inform. and Comput.*, 206(7):908–929, 2008. Preliminary version in STOC'07. [doi:10.1016/j.ic.2008.04.003, arXiv:cs/0605140] 185, 211

[16] LESLIE ANN GOLDBERG AND MARK JERRUM: Inapproximability of the Tutte polynomial of a planar graph. *Comput. Complexity*, 21(4):605–642, 2012. [doi:10.1007/s00037-012-0046-4] 186

[17] YENJO HAN, LANE A. HEMASPAANDRA, AND THOMAS THIERAUF: Threshold computation and cryptographic security. *SIAM J. Comput.*, 26(1):59–78, 1997. Preliminary version in ISAAC'93. [doi:10.1137/S0097539792240467] 195

[18] JOEL HASS, JEFFREY C. LAGARIAS, AND NICHOLAS PIPPENGER: The computational complexity of knot and link problems. *J. ACM*, 46(2):185–211, 1999. Preliminary version in FOCS'97. [doi:10.1145/301970.301971, arXiv:math.GT/9807016] 214

[19] LANE A. HEMACHANDRA: The strong exponential hierarchy collapses. *J. Comput. System Sci.*, 39(3):299–322, 1989. Preliminary version in STOC'87. [doi:10.1016/0022-0000(89)90025-1] 195

[20] FRANÇOIS JAEGER, DIRK L. VERTIGAN, AND DOMINIC WELSH: On the computational complexity of the Jones and Tutte polynomials. *Math. Proc. Cambridge Philos. Soc.*, 108(1):35–53, 1990. [doi:10.1017/S0305004100068936] 185, 186

[21] VAUGHAN F. R. JONES: On knot invariants related to some statistical mechanical models. *Pacific J. Math.*, 137(2):311–334, 1989. Available online at Project Euclid. 209

[22] MICHAEL KAPOVICH: *Hyperbolic Manifolds and Discrete Groups*. Modern Birkhäuser Classics. Birkhäuser, 2010. Available online at Springer. 213

[23] CHRISTIAN KASSEL: *Quantum Groups*. Volume 155 of *Graduate Texts in Mathematics*. Springer, 1995. 202

[24] LOUIS H. KAUFFMAN: Spin networks and knot polynomials. *Internat. J. Modern Phys. A*, 5(1):93–115, 1990. [doi:10.1142/S0217751X90000039] 200

[25] LOUIS H. KAUFFMAN AND SOSTENES L. LINS: *Temperley-Lieb Recoupling Theory and Invariants of 3-Manifolds*. Annals of Math. Studies. Princeton Univ. Press, 1994. 202

[26] ALEXANDER A. KIRILLOV, JR.: On an inner product in modular tensor categories. *J. Amer. Math. Soc.*, 9(4):1135–1169, 1996. [doi:10.1090/S0894-0347-96-00210-X, arXiv:q-alg/9508017] 202

[27] GREG KUPERBERG: Spiders for rank 2 Lie algebras. *Comm. Math. Phys.*, 180(1):109–151, 1996. [doi:10.1007/BF02101184, arXiv:q-alg/9712003] 200

[28] GREG KUPERBERG: Denseness and Zariski denseness of Jones braid representations. *Geom. Topol.*, 15(1):11–39, 2011. [doi:10.2140/gt.2011.15.11, arXiv:0909.1881] 213

[29] GREG KUPERBERG: Knottedness is in NP, modulo GRH. *Adv. Math.*, 256:493–506, 2014. [doi:10.1016/j.aim.2014.01.007, arXiv:1112.0845] 214

[30] MICHAEL A. NIELSEN AND ISAAC L. CHUANG: *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000. 185, 186, 189, 190

[31] PAUL-ÉMILE PARADAN: Symmetric spaces of the non-compact type: Lie groups. In *Géométries à Courbure Négative ou Nulle, Groupes Discrets et Rigidités*, volume 18 of *Sémin. Congr.*, pp. 39–76. Soc. Math. France, Paris, 2009. Available on HAL archives ouvertes. 189, 193

[32] ERIC C. ROWELL: Two paradigms for topological quantum computation. In *Adv. in Quantum Comput.*, volume 482 of *Contemp. Math.*, pp. 165–177. Amer. Math. Soc., 2009. [doi:10.1090/conm/482/09418, arXiv:0803.1258]

[33] RONEN SHALTIEL AND CHRISTOPHER UMANS: Pseudorandomness for approximate counting and sampling. *Comput. Complexity*, 15(4):298–341, 2006. Preliminary version in CCC'05. [doi:10.1007/s00037-007-0218-9] 195

[34] SEINOSUKE TODA: PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991. Preliminary version in FOCS'89. [doi:10.1137/0220053] 187

[35] VEERAVALLI SESHADRI VARADARAJAN: *Lie Groups, Lie Algebras, and Their Representations*. Volume 102 of *Graduate Texts in Mathematics*. Springer, 1984. Reprint of the 1974 edition. 189, 190

[36] DIRK L. VERTIGAN: The computational complexity of Tutte invariants for planar graphs. *SIAM J. Comput.*, 35(3):690–712, 2005. [doi:10.1137/S0097539704446797] 185

[37] MIKHAIL N. VYALYĬ: QMA = PP implies that PP contains PH. *Electron. Colloq. on Comput. Complexity (ECCC)*, TR03-021, 2003. 197, 198

[38] HANS WENZL: C^* tensor categories from quantum groups. *J. Amer. Math. Soc.*, 11(2):261–282, 1998. [doi:10.1090/S0894-0347-98-00253-7] 202

[39] HANS ZASSENHAUS: Beweis eines Satzes über diskrete Gruppen. *Abh. Math. Sem. Hamburg*, 12(1):289–312, 1937. [doi:10.1007/BF02948950] 213

[40] The Complexity Zoo. https://complexityzoo.uwaterloo.ca. 184, 186, 187

AUTHOR

Greg Kuperberg
Professor
University of California, Davis
greg@math.ucdavis.edu
`http://www.math.ucdavis.edu/~greg`

ABOUT THE AUTHOR

GREG KUPERBERG graduated from Harvard University in 1987 and received a Ph. D. in mathematics from UC Berkeley in 1991. His advisor was Andrew Casson. He joined the mathematics department at UC Davis in 1996. He believes that a day without bicycling is not a perfect day.