A Rosetta Stone for Quantum Computation

Quantum Algorithms & Beyond

Distributive Quantum Computing

Topological Quantum Computing and the Jones Polynomial

A Quantum Computing Knot Theoretic Mystery

Can be found on my webpage.

---

Samuel J. Lomonaco, Jr.
Dept. of Comp. Sci. & Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD 21250
Email: Lomonaco@UMBC.EDU
WebPage: http://www.csee.umbc.edu/~lomonaco

Topological Quantum Computing and the Jones Polynomial

This work is in collaboration with Louis H. Kauffman

This work is supported by:

- The National Institute for Standards and Technology (NIST)
- The Mathematical Sciences Research Institute (MSRI).
- The L-O-O-P Fund.
- The Institute of Scientific Interchange

This talk is about an algorithm found in:


We will refer to this paper as AJL

and also about the paper


This talk is about an algorithm found in:
**Quantum World**

- Quantum States
- Superposition
- Measurement
- Unitary Evolution
- Entanglement
- Quantum Algorithms

---

**Quantum Algorithms Zoo**

- Deutsch-Jozsa
- Simon’s
- Shor’s
- Q. Hidden Subgroup Algs.
- Quantum Simulation of Q. Systems
- Exponential Speedup
- Related ???
- Grover’s
- Amplitude Amplif. Algs.
- Quadratic Speedup
- Q. Random Walks
- Non-Classical Behavior
- Jones Poly. Alg.
- Trace Estimation
- New
- Adiabatic Algs.
- Other Algs.

---

**Fundamental Questions ???**

What are the limits of Quantum Computation ???

Will future quantum computers be general purpose or special purpose devices ???

---

**Questions ???**

Why are we interested in the Jones polynomial ???

Why do we want to find a fast, i.e. polytime, quantum algorithm for the Jones polynomial ???

Why should the MoD be interested in the Jones polynomial ???

---

**Complexity**

But the Jones polynomial is in the complexity class #P !!!.

---

**So what !!!**

So what ! Who cares ?
The Class \#P

The satisfiability problem (SAT) is NP-complete.

\[ e(x_1, \ldots, x_n) \rightarrow \text{Satisfiable} \]

\[ \begin{array}{c}
\text{Decision Problem} \\
\text{Yes?} \\
\text{No?}
\end{array} \]

The corresponding \#P class is:

\[ e(x_1, \ldots, x_n) \rightarrow \text{Find # of Satisfying Values} \]

\[ \# \{ e \in \{0,1\}^n : e \equiv \text{True} \} \]

Boolean expr in CNF form

Boolean expr in CNF form

The corresponding \#P class is

Counting Problem

Answer: This would mean that P^{#P} is in BQP.

But since NP is in P^{#P}, this implies that NP is in BQP. Hence, a polytime quantum algorithm for the Jones polynomial implies there exists a polytime quantum algorithm for NP-complete problems.

Wow !!! But this is probably not the case!!!

Questions ???

Why do we want to find a fast, i.e. polytime, quantum algorithm for the Jones polynomial ???

Answer: This would mean that P^{#P} is in BQP.

But since NP is in P^{#P}, this implies that NP is in BQP. Hence, a polytime quantum algorithm for the Jones polynomial implies there exists a polytime quantum algorithm for NP-complete problems.

Wow !!! But this is probably not the case!!!

Questions ???

On the other hand ...

Widely held, but unproven belief: There is no quantum algorithm that computes an NP-complete problem in polynomial time.

So can there really be a polytime quantum algorithm for computing the Jones polynomial ??? Probably not !!!

But creating a quantum algorithm for the Jones polynomial might well give us some insight why NP is not in BQP ???

Questions ???

• How can we compute the Jones polynomial on a quantum computer ?

• How fast is the resulting quantum algorithm ?

We will now discuss the polytime algorithm:

\[ \text{Knot } K = \text{\bowtie} \]

AFJKL Algorithm

Jones polynomial

\[ V_K \left( e^{2\pi i/k} \right) \pm \epsilon \]

With exponentially small probability of error
We will now discuss the following theorem:

**Theorem.** (AJL)

For a given braid $\beta$ in $B_n$ with $m$ crossings and for a given integer $k$, there is a quantum algorithm which is polynomial in $n, m, k$, which with all but exponentially (in $n, m, k$) small probability, outputs a result in the closed interval

$$\left[ \Re \left( V_{\beta} \left( e^{\frac{2\pi i}{k}} \right) \right) - \varepsilon d^{-1} , \Re \left( V_{\beta} \left( e^{\frac{2\pi i}{k}} \right) \right) + \varepsilon d^{-1} \right]$$

for $\varepsilon$ which is inverse polynomial in $n, k, m$. The same is true for the imaginary part.

---

**Quick Overview to Knot Theory**

**Planar four valent graph with labeled vertices**

**Reidemeister Moves**

- $R0$
- $R1$
- $R2$
- $R3$

*These are local moves!*

**Comment:** $R1$ is not a Physical Move

*A Discontinuous Move!*
Each point of a curve in 3-space is naturally associated with a 3-frame, called the Frenet frame.

\[ T = \text{tangent} \]
\[ B = \text{binormal} = T \times N \]
\[ N = \text{normal} \]

In other words, given a smooth curve \( x(t) = (x_1(t), x_2(t), x_3(t)) \) in 3-space, there is a naturally associated curve \( A(t) \) in \( SO(3) \).

Theorem (Reidemeister). Two knots (or links) diagrams represent the same knot (or link) iff one can be transformed into the other by a finite sequence of Reidemester moves.

Comment: R1 is not a Physical Move

Example of Application of Reidemeister Moves
What is a knot invariant?

Def. A knot invariant $I$ is a map $I: Knots \rightarrow Mathematical\,\,Domain$ that takes each knot $K$ to a mathematical object $I(K)$ such that

\[ I(\sim K) = I(K) \]

Consequently,$\quad I(\sim K) \neq I(K) \Rightarrow

The Jones polynomial is a knot invariant.

What Is the Braid Group $B_n$ ???

Why is the braid group important for Q Comp ?

* The representations of the Symmetric $S_n$ are the basic building blocks for the representation of the unitary group $U$ used in quantum mechanics,

* The braid group $B_n$ "sits above" the symmetric group $S_n$, i.e., there is a natural epimorphism $B_n \rightarrow S_n$,

* Thus, new representations of the braid group $B_n$ will give us new representations of the unitary group $U$, i.e., quantum gates,

* Claim: These quantum gates can be implemented in quantum systems that are resistant to decoherence because of topological obstructions, e.g., in terms of the fractional quantum Hall effect, anyonic systems

A Braid

Hat Box

3 Strand braid $\beta$

Two Equal Braids

$\beta_1 = \beta_2$

Two Unequal Braids

$\beta_1 \neq \beta_2$
**Shorthand Notation**

- **Hat Box**
- **3 Strand braid β**

**Product of Braids**

\[ \beta_1 \cdot \beta_2 = \beta_3 \]

**Inverse of a Braid**

\[ \beta \cdot \beta^{-1} = 1 \]

To construct the inverse of a braid, take the mirror image of each crossing, and then reverse the order of the crossings.

**Generators of the Braid Group \( \mathbb{B}_n \)**

The braid group \( \mathbb{B}_n \) is generated by

\[ b_1, b_2, \ldots, b_{n-1} \]

**Relations Among the Generators of \( \mathbb{B}_n \)**

- \( b_i b_{i+1} b_i = b_{i+1} b_i b_{i+1}, 1 \leq i < n \)
- \( b_i b_j = b_j b_i, \text{ for } |i - j| \geq 2 \)

**Reidemeister 3 Move**
A Presentation of the Braid Group $B_n$

\[
\begin{align*}
   b_i b_j &= b_j b_i, \text{ for } |i - j| \geq 2 \\
   b_i, b_2, \ldots, b_{n-1}: \\
   b_i b_i &= b_i b_i, 1 \leq i < n-1 \\
   b_i b_j &= b_j b_i, |i - j| > 1, 1 \leq i, j < n-1 \\
   b_i b_j &= b_j b_i, |i - j| > 1, 1 \leq i, j < n-1 \\
\end{align*}
\]

Generators

Complete set of Relations

A Braid Is "Almost" a Permutation

\[
B_n = \left\{ b_i b_j = b_j b_i, 1 \leq i < n-1 \right\}
\]

\[
S_n = \left\{ b_i b_i = b_i b_i, 1 \leq i < n-1 \right\}
\]

Natural Epimorphism

Braids as Words

Every braid $\beta$ in $B_n$ can be written as a product of braid generators $b_i, b_{i+1}, \ldots, b_{n-1}$ and their inverses $b_i^{-1}, b_{i+1}^{-1}, \ldots, b_{n-1}^{-1}$.

\[
\beta = b_{j_1}^{\epsilon(j_1)} b_{j_2}^{\epsilon(j_2)} \ldots b_{j_L}^{\epsilon(L)}
\]

where $\epsilon(i) = \pm 1$

Braid word w

Anyons: A Very Brief Overview

Anyons are quantum systems that are confined to two dimensions. They were first proposed by Nobel Laureate F. Wilczek. See for example,


Anyons can be used to explain the fractional quantum Hall effect

Another Perspective

A Braid Represents the Movement of $n$ Holes in a Disc

This braiding can be used to represent Anyon exchanges

Recall: Q.M. = Group Rep. Theory
Anyons Can Also Fuse or Split

Quantum Topology gives us the tools needed to find new unitary representations based on fusing and braiding.

These new unitary transformations are created with an object called a unitary topological modular functor which we call simply an anyon model.

Recall: Q.M. = Group Rep. Theory

Anyons: A Very Brief Overview (Cont.)

Knots from Braids

The Markov trace closure

From Braids to Knots

The Markov Trace Closure of a Braid

Braid $\beta$

Closed Braid $\beta^c$

Can Every Knot Be Constructed from the Closure of Some Braid?

Theorem (Alexander). Every knot is the closure of a braid.

When Does the Closure of Two Braids Produce the Same Link?

Theorem (Markov). Two braids $\beta_1$ and $\beta_2$ produce the same link under Markov trace braid closure iff there exists a finite sequence of Markov moves that transforms one braid into the other.

Definition. Let $\beta$ be a braid in $B_n$. Then the Markov moves are defined as:

- $M_1$: $\beta \mapsto \beta' = b_i^{\pm 1} \beta b_i^{\mp 1}$, $1 \leq i < n$
- $M_2$: $\beta \mapsto \beta \cdot b_{n+1}^{\pm 1} \in B_{n+1}$
The Markov 1 Move

$$\beta' = b_i^{\pm 1} \beta b_i^{\pm 1}, \ 1 \leq i < n$$

The Markov 2 Move

$$\beta' = \beta b_{n+1}^{\pm 1} \in B_{n+1}$$

From Braids to Knots

The Plait Closure of a Braid

Strategy for Computing Jones Polynomial

• We use the Jones representation

$$\rho_A : B_n \to TL_n(d)$$

from the braid group $$B_n$$ to the Temperley–Lieb algebra $$TL_n(d)$$, where $$d$$ is an indeterminate complex number, and where $$A$$ is a complex number defined by $$d = -A^2 - A^2$$.

What is the Temperley–Lieb algebra $$TL_n(d)$$?

???
Diagrammatic Representation of \( TL_n(d) \)

Rectangle

An element \( X \) of \( TL_n(d) \)

Two Equal Elements of \( TL_n(d) \)

\[ X_1 = X_2 \]

Two Unequal Elements of \( TL_n(d) \)

\[ X_1 \neq X_2 \]

Shorthand Notation

Rectangle

Shorthand Notation

Product

\[ X_1 \cdot X_2 = X_3 \]

Product

\[ X_1 \cdot X_2 = X_3 = d \]
Generators of the Temperley-Lieb Algebra $\text{TL}_n(d)$

The Temperley-Lieb algebra $\text{TL}_n(d)$ is generated by

$$1, E_1, E_2, \ldots, E_{n-1}$$

Relations Among the Generators of $\text{TL}_n(d)$

- $E_i E_j = E_j E_i$, for $|i - j| \geq 2$
- $E_i E_{i+1} E_i = E_i$, $1 \leq i < n$
- $E_i^2 = d E_i$, $1 \leq i < n$

The Markov Trace $\text{Tr}_n : \text{TL}_n(d) \to \mathbb{C}$

- $\text{Tr}(1) = 1$
- $\text{Tr}(XY) = \text{Tr}(YX)$
- $X \in \text{TL}_{n+1}(d) \Rightarrow \text{Tr}_{n+1}(XE_n) = \frac{1}{d} \text{Tr}_n(X)$

How to Compute the Jones Polynomial

We use the Jones representation $\rho_A : B_n \to \text{TL}_n(d)$ from the braid group $B_n$ to the Temperley-Lieb algebra $\text{TL}_n(d)$, where $d$ is an indeterminate complex number, and where $A$ is a complex number defined by $d = A^2 - A^{-2}$. 

Let $n$ be an integer, and let $d$ be a complex number.

The Temperley-Lieb algebra $\text{TL}_n(d)$ is the algebra generated by $\{I, E_i, E_{i+1}, \ldots, E_{n+1}\}$ with relations:

$$E_iE_j = E_jE_i \quad \text{for} \quad |i - j| \geq 2$$

$$E_iE_{i+1}E_i = E_i$$

and with an involution $*$ defined by the conjugate linear extension of $(E_iE_{i+1}E_i^*) = E_iE_{i+1}E_i$. 

The Jones Representation

- Let $d$ be an indeterminate complex number, and let $A$ be a complex number such that $d = A^2 - A^{-2}$.

- Let $B_n$ denote the braid group, and let $\text{TL}_n(d)$ denote the Temperley-Lieb algebra.

The Jones representation $\rho : B_n \rightarrow \text{TL}_n(d)$ is the group representation defined by

\[
\begin{align*}
    b_1 &\mapsto AI + A^{-1}E_i = A \uplus A^{-1}U \\
    b_i &\mapsto A^{-1}I + AE_i = A^{-1} \uplus A U
\end{align*}
\]

Strategy for Computing Jones Polynomial

- We use Jones’s definition of the Jones polynomial.

- Let $\beta$ be an element of $B_n$.

- Let $\beta^m$ be the closure of $\beta$.

- Let $\text{Tr}_n : \text{TL}_n(d) \rightarrow C$ be the standard Markov trace into the complex numbers $C$.

Then the Jones polynomial is given by

$$V_{\beta^m}(A^{-4}) = -A^{2w(\beta)} d^{-n} \text{Tr}_n(\rho_A(\beta))$$

Strategy (Cont.)

We now wish to compute a good approximation of the value of the Jones polynomial at $e^{2\pi i/k}$ for $k$ a positive integer.

- Let $G_k$ be the graph with $k-1$ vertices and $k-2$ edges.

The Graph $G_k$

- Adjacency Matrix $M_k = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$

- Eigenvector $\lambda_i = \left( \sin \left( \frac{\pi i}{k} \right) \right)_{i \in \{1, 2, \ldots, k-1\}}$

- Corresponding Eigenvalue $2\cos(\pi i/k)$
**The Graph $G_k$**

- Let $d$ be equal to the eigenvalue $2\cos(\pi/k)$
- Since $d = -A^2 - A^{-2}$, there are four possible choices for $A$, namely
  \[ A = \pm e^{\pm i \pi / 2k} \]
  We choose \[ A = e^{-i \pi / 2k} \]

---

**Strategy (Cont.)**

We will use the graph $G_k$ to construct a unitary representation of the Temperley-Lieb algebra $TL_n(d)$.

- Let $P_{n,k}$ be the set of paths in $G_k$ of length $n$ which start at the vertex 1.
- Let $H_{n,k}$ be the Hilbert space with orthonormal basis \[ \{ |p\rangle : p \in P_{n,k} \} \]

---

**Constructing a Unitary Rep of $TL_n(d)$**

- We now construct a unitary representation of the Temperley-Lieb algebra $TL_n(d)$
  \[ \Phi : TL_n(d) \to \mathbb{C} U \left( H_{n,k} \right) \]

---

**Constructing a Unitary Rep of $TL_n(d)$ (Cont.)**

- To define the rep $\Phi : TL_n(d) \to \mathbb{C} U \left( H_{n,k} \right)$ all that we need do is to specify the images $\Phi_i = \Phi(E_i)$ on each of the generators $E_i$ of the Temperley-Lieb algebra $TL_n(d)$, and then show that the $\Phi_i$'s are compatible with the relations defining the Temperley-Lieb algebra

---

**Constructing a Unitary Rep of $TL_n(d)$ (Cont.)**

- We define each $\Phi_i$ as
  \[
  \Phi_i |p\rangle = \begin{cases} 
  0 & \text{if } p^{i \rightarrow i+1} = 00 \\
  \left( \frac{\lambda_{i,i+1}}{\lambda_{i,i}} \right) |p\rangle + \left( \frac{\lambda_{i+1,i}}{\lambda_{i,i+1}} \right) e^{-i \pi 10^{|p^{i \rightarrow i+1}|}} & \text{if } p^{i \rightarrow i+1} = 01 \\
  \left( \frac{\lambda_{i,i}}{\lambda_{i,i+1}} \right) e^{-i \pi 10^{|p^{i \rightarrow i+1}|}} |p\rangle + \left( \frac{\lambda_{i+1,i}}{\lambda_{i,i+1}} \right) & \text{if } p^{i \rightarrow i+1} = 10 \\
  0 & \text{if } p^{i \rightarrow i+1} = 11 
  \end{cases}
  \]
We define each $\Phi_i$ as

$$ \Phi_i |p\rangle = \begin{cases} 
0 & \text{if } p^{(i-1)} = 00 \\
\text{Zig-Zag} & \text{if } p^{(i-1)} = 01 \\
(\text{Bla}) |p\rangle + (\text{Bla}) |p^{(i-1)}10p^{(i-1)}\rangle & \text{if } p^{(i-1)} = 10 \\
\text{Zag-Zig} & \text{if } p^{(i-1)} = 11 
\end{cases} $$

We then verify that $\Phi_i$ is actually a representation by checking that it is compatible with the following Temperley-Lieb algebra identities:

- $E_i E_j = E_i, |i-j| > 2$
- $E_i E_i = E_i$
- $E_i^2 = dE_i$

i.e., the following hold:

- $\Phi_i = 0$
- $\Phi_i = \Phi_i$
- $\Phi_i^2 = d\Phi_i$

**Strategy (Cont.)**

- We now seek to construct a trace $\text{Tr}: \text{Im}(\Phi) \to \mathbb{C}$ which is compatible with the Markov trace on $T_{l_n}(d)$, i.e., a trace $\text{Tr}$ such that the following diagram is commutative:

  $$ T_{l_n}(d) \xrightarrow{\Phi} \text{Im}(\Phi) \xrightarrow{\text{Tr}} \mathbb{C} $$

**The Unitary Rep $\Phi$ of $T_{l_n}(d)$**

- Recall that we define each $\Phi_i$ as

  $$ \Phi_i |p\rangle = \begin{cases} 
0 & \text{if } p^{(i-1)} = 00 \\
(\text{Bla}) |p\rangle + (\text{Bla}) |p^{(i-1)}10p^{(i-1)}\rangle & \text{if } p^{(i-1)} = 01 \\
(\text{Bla}) |p^{(i-1)}01p^{(i-1)}\rangle + (\text{Bla}) |p\rangle & \text{if } p^{(i-1)} = 10 \\
0 & \text{if } p^{(i-1)} = 11 
\end{cases} $$

**Lemma.** The representation $\Phi: T_{l_n}(d) \to \mathbb{C}(H_{s,k})$ maps each ket $|p\rangle$ to a linear combination of kets each labeled by a path of the same length and the same endpoint as $p$.

- Let $P_{n,k,m} = \text{set of paths in } G_k \text{ of length } n \text{ which start at the vertex } 1 \text{ and end at vertex } m$.
- Let $H_{n,k,m}$ be the Hilbert space with orthonormal basis

  $$ \{ |p\rangle : p \in P_{n,k,m} \} $$

**Lemma.** The representation $\Phi: T_{l_n}(d) \to \mathbb{C}(H_{s,k})$ maps each ket $|p\rangle$ to a linear combination of kets each labeled by a path of the same length and the same endpoint as $p$.

**Theorem.**

$$ \Phi = \bigoplus_{m=1}^{l_n} \Phi^{(m)}: T_{l_n}(d) \to \bigoplus_{m=1}^{l_n} \mathbb{C}(H_{n,k,m}) \subset \mathbb{C}(H_{s,k}) $$

where

$$ \Phi^{(m)}: T_{l_n}(d) \to \mathbb{C}(H_{n,k,m}) $$

Hence,

$$ \text{Im}(\Phi) \subset \bigoplus_{m=1}^{l_n} \mathbb{C}(H_{n,k,m}) $$
Recall $M = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}$

This gives us a lot of latitude in choosing a trace. We can choose from among all possible linear combinations of the standard traces $\{ Tr(\Phi^m(X)) : 1 \leq m < k \}$

**Proving Our Trace is Compatible**

We prove each of the following:

- $\widetilde{Tr} \circ \Phi(1) = 1$
- $\widetilde{Tr} \circ \Phi(XX) = \widetilde{Tr} \circ \Phi(YX)$
- $X \in TL_{n+1}(d) \Rightarrow \widetilde{Tr} \circ \Phi(XE_{n+1}) = \frac{1}{d} \widetilde{Tr} \circ \Phi(X)$

Since the Markov trace is the unique trace satisfying the above three conditions, we have that the following diagram is commutative.

**Where are we ???**

We have $B_n \xrightarrow{\phi} TL_n(d) \xrightarrow{\Phi} U(H_{2n})$

**Compatible Trace**

We have $B_n \xrightarrow{\phi} TL_n(d) \xrightarrow{\Phi} U(H_{2n})$

**Jones Rep**

**Markov Trace**

Moreover, we have the Jones polynomial $V_{\beta^p}(t) = -A d^{n-1} \widetilde{Tr}(\rho_A(\beta))$
Where are we ???

We need to compute $\text{Tr}_n(\rho_4(\beta))$

But

$$\text{Tr}_n(\rho_4(\beta)) = \text{Tr}_n\left(\left(\Phi^m \circ \rho_4\right)(\beta)\right)$$

$$= \frac{1}{N} \sum_{i=1}^{N} \lambda_i \text{Tr}_n\left(\left(\Phi^m \circ \rho_4\right)(\beta)\right)$$

Thus, all we need to do is to compute the traces of the unitary transformations

$$U^{(m)} = (\Phi^m \circ \rho_4)(\beta) \quad \text{for} \quad 1 \leq m < k$$

Where are we ???

All we need to do is to compute the traces of

$$U^{(m)} = (\Phi^m \circ \rho_4)(\beta) \quad \text{for} \quad 1 \leq m < k$$

Let

$$\beta = \prod_{i=1}^{k} b_{i}^k$$

Let

$$U_j^{(m)} = (\Phi^m \circ \rho_4)(b_j) \quad \text{for} \quad 1 \leq m < k, \quad 1 \leq j < n$$

Then

$$U^{(m)} = (\Phi^m \circ \rho_4)(\beta) = \prod_{i=1}^{k} U_j^{(m)}(b_i^k)$$

Why are we ???

Thus, all reduces to the problem of computing the traces of

$$U^{(m)} = \prod_{i=1}^{k} U_j^{(m)}(b_i^k)$$

for $1 \leq m < k$.

where

$$U_j^{(m)} = (\Phi^m \circ \rho_4)(b_j)$$

Overview

The algorithm consists of two phases:

Phase 1: Compilation Phase – We compile Mathematics into hardware.

(Braid Word = Computer Program)

Phase 2: Execution Phase – We run our compiled program

Compilation Phase for Estimating $V_p(e^{2\pi i/k})$

Execution Phase for Estimating $V_p(e^{2\pi i/k})$
The Two Quantum Subroutines QRe & QIm

- **QRe** - Used to compute the real part of the trace, i.e., \( \text{Re}(Tr(U)) \)

- **QIm** - Used to compute the imaginary part of the trace, i.e., \( \text{Im}(Tr(U)) \)

The Quantum Subroutines QRe

\[ U \rightarrow \text{QRe} \rightarrow \text{Re}[Tr(U)] \]
Wiring Diagram for Unitary Transformation

$U : H \rightarrow H$

$n$-Dimensional Hilbert Space

Controlled - $U$

Contr. - $U : H_2 \otimes H \rightarrow H_2 \otimes H$

Control-Qubit

2n-Dimensional Hilbert Space

Output = $\begin{pmatrix} |0\rangle \otimes |\psi\rangle \\ |1\rangle \otimes |\psi\rangle \end{pmatrix}$

Do Zilch if control = 0

Apply $U$ if control = 1

Estimating Trace($U$)

Hadamard Transform

$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

We have used Quantum Mechanics to engineer a probability distribution!!

Engineering a Probability Distribution

Output = 0 or 1

if we repeat many times, then

Thus, we have constructed a probability distribution for computing the real part of

$Trace(U) = \sum \langle \bar{p} | U | p \rangle$
The Quantum Subroutines $QRe$

$$U \xrightarrow{\text{In}} \text{QRe} \xrightarrow{\text{Out}} \text{Re}[Tr(U)]$$

The Quantum Subroutines $QIm$

$$U \xrightarrow{\text{In}} \text{QIm} \xrightarrow{\text{Out}} \text{Im}[Tr(U)]$$

Summary

We have constructed a polytime quantum algorithm:

Knot $K$ =

Integer $k$ → AFJKL Algorithm → Jones polynomial $V_K(e^{2\pi i/k}) \pm \epsilon$

with exponentially small probability of error

Question

Can we transform the AFJKL quantum algorithm using polynomial interpolation into a polytime quantum algorithm that computes the Jones polynomial $V_K(t)$ with exponentially small probability of error?

Answer: No!